



Universidad Católica
San José

ESCUELA DE POSGRADO
MAESTRÍA EN EDUCACIÓN

APLICACIÓN DEL MÉTODO BALTEX PARA
BRINDAR ORIENTACIÓN Y TUTORÍA EN
EL PROCESO DE APRENDIZAJE DE LA
COMPUTACIÓN EN LOS ESTUDIANTES
DEL PROGRAMA NACIONAL BECA 18 EN
EL AÑO 2015

TESIS PRESENTADA POR EL ALUMNO MAESTRISTA
ING. MANUEL VALERIANO BALTA ROSPIGLIOSI PARA OBTENER
EL GRADO ACADÉMICO DE MAESTRO EN EDUCACIÓN, EN LA
MENCION ORIENTACIÓN Y TUTORÍA.

LIMA – PERÚ

2018

AGRADECIMIENTOS

- Agradezco a Dios por haberme dado la misión de ser educador.
- Agradezco al Mg. Edy Dalmiro Barnett Mendoza, por darme las facilidades en mi centro laboral para realizar mis estudios de Posgrado durante los años 2014 y 2015.
- Agradezco al Dr. Carlos Darcourt Huck, docente de la Universidad Católica San José quien se interesó por el tema de mi tesis y me dio pautas para profundizar en mi investigación.
- A mis profesores de la Escuela de Posgrado de la Universidad Católica San José, que compartieron sus experiencias, modelos de enseñanza y guías para brindar orientación y tutoría a nuestros estudiantes.
- A mis compañeros de la Maestría en Educación, mención Orientación y Tutoría que enriquecieron el curso con sus aportes que permitieron, en un sentido ecléctico, aprender de ellos.
- A los estudiantes del Programa Nacional Beca 18 que participaron en esta investigación.
- A mis hermanos Ada, Miguel y Rosa quienes me dieron su apoyo en todo momento para culminar mis estudios de Posgrado.

A mi hijo Alessandro,
a quien amo con todo mi corazón,
cuya sonrisa es mi felicidad.

ÍNDICE GENERAL

Carátula	I
Agradecimientos	II
Dedicatoria	III
Índice General	IV
Índice de tablas	VI
Índice de figuras	VIII
Resumen	X
Abstract	XI
Introducción	XII

CAPÍTULO I PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1	Planteamiento del problema	1
1.2	Formulación del problema	5
1.2.1	Problema general	5
1.2.2	Problemas específicos	5

CAPÍTULO II MARCO TEÓRICO

2.1	Antecedentes	6
2.2	Bases teóricas	9
2.2.1	La orientación educativa	9
2.2.2	Los principios de la orientación educativa	9
2.2.2.1	Principios de prevención	9
2.2.2.2	Principios de desarrollo	10
2.2.2.3	Principios de intervención social	11
2.2.2.4	Principios de fortalecimiento personal	11
2.2.3	La tutoría	12
2.2.4	Proceso de aprendizaje	12
2.2.5	El aprendizaje de contenidos procedimentales	13
2.2.6	Capacidad de reflexión, entendimiento y planificación	15
2.2.7	Capacidad de adaptar un panorama visual	15
2.2.8	Capacidad de redacción de pseudocódigos	16
2.2.9	Capacidad de programación	16
2.2.10	Representaciones algebraicas	16
2.2.11	Algunas representaciones algebraicas	17
2.2.12	Representación de datos en pseudocódigos	18
2.2.13	Lista de pseudocódigos	19
2.2.14	Equivalencias entre pseudocódigos y código c	20
2.2.15	Representación de datos en forma gráfica	21
2.2.16	El Método Baltex	23
2.2.16.1	Definición de la Estructura Secuencial	23
2.2.16.2	Definición de la Estructura de Selección de Datos "SI-SINO"	31
2.2.16.3	Definición de la Estructura de Selección de Datos "SI Anidado"	40
2.2.16.4	Definición de la Estructura de Selección de Datos Múltiple "Cuando sea"	51
2.2.16.5	Definición de la Estructura de Repetición de Datos "Mientras"	61
2.2.16.6	Definición de la Estructura de Repetición de Datos "Para"	71
2.2.16.7	Definición de la Estructura de Repetición de Datos "Hacer-Mientras"	81
2.2.17	Aplicación de los principios de la orientación educativa en el Método Baltex	90
2.2.18	Definición de términos básicos	93
2.2.19	Justificación del estudio	98

CAPÍTULO III OBJETIVOS E HIPÓTESIS

3.1	Objetivos	99
3.1.1	Objetivo general	99
3.1.2	Objetivos específicos	99
3.2	Hipótesis	100
3.2.1	Hipótesis general	100
3.2.2	Hipótesis específicas	100

CAPÍTULO IV METODOLOGÍA DE LA INVESTIGACIÓN

4.1	Diseño de la investigación	101
4.1.1	Diseño cuasiexperimental	101
4.1.2	Alcance explicativo	102
4.1.3	Alcance descriptivo	102
4.2	Enfoque cuantitativo	103
4.3	Población y muestra	105
4.4	Operacionalización de las variables	106
4.5	Técnicas para la recolección de datos	108
4.5.1	Descripción de los instrumentos	108
4.5.2	Validez y confiabilidad de los instrumentos de recolección de datos	109
4.6	Técnicas para el procesamiento y análisis de los datos	110
4.7	Aspectos éticos	110

CAPÍTULO V RESULTADOS DE LA INVESTIGACIÓN

5.1	Resultados descriptivos en el grupo experimental	111
5.2	Resultados descriptivos en el grupo control	116
5.3	Prueba de hipótesis	121

CAPÍTULO VI CONCLUSIONES DE LA INVESTIGACIÓN

Conclusiones	133
--------------	-----

CAPÍTULO VII RECOMENDACIONES

Recomendaciones	134
-----------------	-----

ANEXOS

Anexo 1	Matriz de consistencia	136
Anexo 2	Instrumentos para la recolección de datos	137

REFERENCIAS CONSULTADAS	140
--------------------------------	------------

ÍNDICE DE TABLAS

Tabla 1:	Ejemplo de algunas representaciones matemáticas	17
Tabla 2:	Lista de pseudocódigos	19
Tabla 3:	Equivalencias entre pseudocódigos y código c	20
Tabla 4:	Relación de símbolos del diagrama de flujo. Parte 1	21
Tabla 5:	Relación de símbolos del diagrama de flujo. Parte 2	22
Tabla 6:	Tabla analítica de la estructura secuencial	25
Tabla 7:	Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura secuencial.	27
Tabla 8:	Correspondencia entre cada pseudocódigo y el código C asociado de la estructura secuencial.	28
Tabla 9:	Tabla analítica de la estructura de selección de datos "SI-SINO"	34
Tabla 10:	Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de selección de datos "SI-SINO"	36
Tabla 11:	Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de selección de datos "SI-SINO"	37
Tabla 12:	Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de selección de datos "SI anidado"	47
Tabla 13:	Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de selección de datos "SI anidado"	48
Tabla 14:	Tabla analítica de la estructura de selección de datos múltiple "Cuando sea".	54
Tabla 15:	Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de selección de datos múltiple "Cuando sea".	57
Tabla 16:	Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de selección de datos múltiple " Cuando sea"	58
Tabla 17:	Tabla de seguimiento y control de valores de la estructura de repetición de datos "Mientras"	65
Tabla 18:	Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de repetición de datos "Mientras".	67
Tabla 19:	Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de repetición de datos "Mientras"	68
Tabla 20:	Tabla de seguimiento y control de valores de la estructura de repetición "Para"	75
Tabla 21:	Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de repetición de datos "Para".	77
Tabla 22:	Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de repetición de datos "Para"	78
Tabla 23:	Tabla de seguimiento y control de valores de la estructura de repetición de datos "Hacer-Mientras"	85
Tabla 24:	Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de repetición de datos "Hacer-Mientras".	87
Tabla 25:	Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de repetición de datos "Hacer-Mientras"	88
Tabla 26:	Operacionalización de la variable independiente "Método Baltex", para el grupo experimental	106
Tabla 27:	Operacionalización de la variable independiente "Método Baltex", para el grupo control	106
Tabla 28:	Operacionalización de la variable dependiente "Aprendizaje procedimental", para el grupo experimental	107
Tabla 29:	Resultados de la Prueba de Confiabilidad - Coeficiente Alfa de Cronbach	109
Tabla 30:	Tabla de frecuencias de la variable dependiente: "Aprendizaje procedimental", en el grupo experimental	111
Tabla 31:	Tabla de frecuencias de la Dimensión 1: "Capacidad de reflexión, entendimiento y planificación"	112
Tabla 32:	Tabla de frecuencias de la Dimensión 2: "Capacidad de adaptar un panorama visual"	113
Tabla 33:	Tabla de frecuencias de la Dimensión 3: "Capacidad de redacción de pseudocódigos"	114

Tabla 34:	Tabla de frecuencias de la Dimensión 4: “Capacidad de programación	115
Tabla 35:	Tabla de frecuencias de la variable dependiente: “Aprendizaje procedimental”, en el grupo experimental	116
Tabla 36:	Tabla de frecuencias de la Dimensión 1: “Capacidad de reflexión, entendimiento y planificación”	117
Tabla 37:	Tabla de frecuencias de la Dimensión 2: “Capacidad de adaptar un panorama visual”	118
Tabla 38:	Tabla de frecuencias de la Dimensión 3: “Capacidad de redacción de pseudocódigos”	119
Tabla 39:	Tabla de frecuencias de la Dimensión 4: “Capacidad de programación	120
Tabla 40:	Resultados de la prueba de normalidad	122
Tabla 41:	Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la variable dependiente	123
Tabla 42:	Resultados de la Prueba de Wilcoxon respecto a la variable dependiente en el grupo experimental	123
Tabla 43:	Selección de la prueba estadística para la comparación de muestras independientes respecto a la variable dependiente	124
Tabla 44:	Resultados de la Prueba U Mann-Whitney respecto a la variable dependiente en el postest	124
Tabla 45:	Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 1	125
Tabla 46:	Resultados de la Prueba de Wilcoxon respecto a la Dimensión 1 en el grupo experimental	125
Tabla 47:	Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 1	126
Tabla 48:	Resultados de la Prueba U Mann-Whitney respecto a la Dimensión 1 en el postest	126
Tabla 49:	Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 2	127
Tabla 50:	Resultados de la Prueba de Wilcoxon respecto a la Dimensión 2 en el grupo experimental	127
Tabla 51:	Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 2	128
Tabla 52:	Resultados de la Prueba U Mann-Whitney respecto a la Dimensión 2 en el postest	128
Tabla 53:	Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 3	129
Tabla 54:	Resultados de la Prueba de Wilcoxon respecto a la Dimensión 3 en el grupo experimental	129
Tabla 55:	Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 3	130
Tabla 56:	Resultados de la Prueba U Mann-Whitney respecto a la Dimensión 3 en el postest	130
Tabla 57:	Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 4	131
Tabla 58:	Resultados de la Prueba T-Student para muestras relacionadas respecto a la Dimensión 4 en el grupo experimental	131
Tabla 59:	Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 4	132
Tabla 60:	Resultados de la Prueba T-Student para muestras independientes respecto a la Dimensión 4 en el postest	132
Tabla 61:	Matriz de consistencia	136

ÍNDICE DE FIGURAS

Figura 1:	Enunciado a programar con la estructura secuencial	23
Figura 2:	Diagrama de flujo de la estructura secuencial	26
Figura 3:	Programa en código C de la estructura secuencial	29
Figura 4:	Programa ejecutado en la computadora de la estructura secuencial	30
Figura 5:	Esquema de comprensión de la estructura de selección de datos SI-SINO	31
Figura 6:	Enunciado a programar con la estructura de selección de datos "SI-SINO"	32
Figura 7:	<i>Diagrama de flujo de la estructura de selección de datos SI-SINO</i>	35
Figura 8:	Programa en código C de la estructura de selección de datos "SI-SINO"	38
Figura 9:	Programa ejecutado en la computadora de la estructura de selección de datos "SI-SINO"	39
Figura 10:	Esquema de comprensión del CASO I de la estructura de selección múltiple "SI Anidado"	40
Figura 11:	Esquema de comprensión del CASO II de la estructura de selección múltiple "SI Anidado"	41
Figura 12:	Esquema de comprensión del CASO III de la estructura de selección múltiple "SI Anidado"	42
Figura 13:	Enunciado a programar con la estructura selección múltiple "SI Anidado"	43
Figura 14:	Técnica analítica con la estructura selección múltiple "SI Anidado"	45
Figura 15:	Diagrama de flujo de la estructura selección múltiple "SI Anidado"	46
Figura 16:	Programa en código C de la estructura selección múltiple "SI Anidado"	49
Figura 17:	Programa ejecutado en la computadora de la estructura de selección múltiple "SI Anidado"	50
Figura 18:	Esquema de comprensión de la estructura de selección múltiple "Cuando Sea"	51
Figura 19:	Enunciado a programar con la estructura de selección múltiple "Cuando Sea"	52
Figura 20:	Diagrama de flujo de la estructura selección múltiple "Cuando Sea". Parte 1	55
Figura 21:	Diagrama de flujo de la estructura selección múltiple "Cuando Sea". Parte 2	56
Figura 22:	Programa en código C de la estructura selección múltiple "Cuando Sea"	59
Figura 23:	Programa ejecutado en la computadora de la estructura selección múltiple "Cuando Sea"	60
Figura 24:	Esquema de comprensión de la estructura de repetición de datos "Mientras"	61
Figura 25:	Enunciado a programar con la estructura de repetición de datos "Mientras"	63
Figura 26:	Diagrama de flujo de la estructura de repetición de datos "Mientras"	66
Figura 27:	Programa en código C de la estructura de repetición de datos "Mientras"	69
Figura 28:	Programa ejecutado en la computadora de repetición de datos "Mientras"	70
Figura 29:	<i>Esquema de comprensión de la estructura de repetición de datos "Para"</i>	71
Figura 30:	Enunciado a programar con la estructura de repetición de datos "Para"	73

Figura 31:	Diagrama de flujo de la estructura de repetición de datos “Para”	76
Figura 32:	Programa en código C de la estructura de repetición de datos “Para”	79
Figura 33:	Programa ejecutado en la computadora de repetición de datos “Para”	80
Figura 34:	Esquema de comprensión de la estructura de repetición de datos “Hacer-Mientras”	81
Figura 35:	Enunciado a programar con la estructura de repetición de datos “Hacer-Mientras”	82
Figura 36:	Técnica analítica de la estructura de repetición de datos “Hacer-Mientras”	84
Figura 37:	Diagrama de flujo de la estructura de repetición de datos “Hacer-Mientras”	86
Figura 38:	Programa en código C de la estructura de repetición de datos “Hacer-Mientras”	89
Figura 39:	Programa ejecutado en la computadora de repetición de datos “Hacer-Mientras”	89
Figura 40:	Fotografía es de estudiantes del Programa Nacional Beca 18 donde se aplica el principio de intervención social de la orientación educativa utilizando el Método Baltex	91
Figura 41:	Fotografía de la solución acertada de una pregunta en una evaluación de un estudiante del Programa Nacional Beca 18 donde se aplica el principio de fortalecimiento personal de la orientación educativa utilizando el Método Baltex	92
Figura 42:	Fotografía de estudiantes del Programa Nacional Beca 18 donde se aplica el principio de fortalecimiento personal de la orientación educativa utilizando el Método Baltex	92
Figura 43:	Representación gráfica del diseño de la investigación	104
Figura 44:	Diagrama de la población, muestra y muestreo de la investigación	105
Figura 45:	Gráfico de barras de la variable dependiente: “Aprendizaje procedimental”, en el grupo experimental	111
Figura 46:	Gráfico de barras de la Dimensión 1: “Capacidad de reflexión, entendimiento y planificación”	112
Figura 47:	Gráfico de barras de la Dimensión 2: “Capacidad de adaptar un panorama visual”	113
Figura 48:	Gráfico de barras de la Dimensión 3: “Capacidad de redacción de pseudocódigos”	114
Figura 49:	Gráfico de barras de la Dimensión 4: “Capacidad de programación”	115
Figura 50:	Gráfico de barras de la Variable Dependiente: “Aprendizaje procedimental”, en el grupo experimental	116
Figura 51:	Gráfico de barras de la Dimensión 1: “Capacidad de reflexión, entendimiento y planificación”	117
Figura 52:	Gráfico de barras de la Dimensión 2: “Capacidad de adaptar un panorama visual”	118
Figura 53:	Gráfico de barras de la Dimensión 3: “Capacidad de redacción de pseudocódigos”	119
Figura 54:	Gráfico de barras de la Dimensión 4: “Capacidad de programación”	120

RESUMEN

La presente investigación intitulada “APLICACIÓN DEL MÉTODO BALTEX PARA BRINDAR ORIENTACIÓN Y TUTORÍA EN EL PROCESO DE APRENDIZAJE DE LA COMPUTACIÓN EN LOS ESTUDIANTES DEL PROGRAMA NACIONAL BECA 18 en el año 2015”, tiene como objetivo ser una herramienta profundamente didáctica, para enseñar a programar a personas sin destreza o experiencia suficiente para ello, de modo que, aplicando su metodología resulta muy difícil no realizar un programa correctamente. Esto tiene una enorme utilidad desde diversos puntos de vista. En primer lugar, su aplicación daría lugar a una reducción en el tiempo de desarrollo de programas ya que si bien pueda parecer que la aplicación de cada uno de sus pasos prolongará el tiempo de desarrollo; sin embargo, reducirá mucho las fases de búsqueda de errores y rectificación de las aplicaciones. De lo anterior, se deduce que la calidad de la programación es muy superior al caso de realizarla de una manera menos estructurada. Por ello mismo, también es más fácil revisar y comprender el código desarrollado.

Finalmente, creo que sería muy interesante aplicar su sencillez para enseñar programación a personas de entornos desfavorecidos, lo que podría abrirles una puerta para un futuro laboral mejor e incluso detectar, entre ellos mentes privilegiadas que podrían aprovechar mejor sus capacidades, a pesar de proceder de entornos marginales.

Palabras clave: Computación, herramientas didácticas, enseñanza, programación básica

ABSTRACT

The present investigation entitled "APPLICATION OF THE BALTEX METHOD TO PROVIDE GUIDANCE AND TUTORING IN THE PROCESS OF LEARNING OF THE COMPUTATION IN THE STUDENTS OF THE NATIONAL SCHOLARSHIP PROGRAM 18 in the year 2015", has as objective to be a deeply didactic tool, to teach to program people without skill or sufficient experience for it, so that, applying its methodology it is very difficult not to carry out a program correctly. This has enormous utility from various points of view. In the first place, its application would lead to a wording in the time of program development since, although it may seem that the application of each of its steps will prolong the development time, however, it will greatly reduce the phases of search for errors and rectification of the applications. From the above, it can be deduced that the quality of the programming will be far superior to the case of performing it in a less structured way. For this reason, it will also be easier to review and understand the developed code.

Finally, I think it would be very interesting to apply its simplicity to teach programming to people from disadvantaged backgrounds, which could open a door for a better future for work and even detect, among them privileged minds that could make better use of their capabilities despite coming from environments marginal.

Keywords: Computing, teaching tools, teaching, basic programming

INTRODUCCIÓN

El objetivo de esta investigación es ayudar a las personas interesadas en aprender a redactar programas en código C, sin ninguna experiencia previa en este tipo de actividad.

Es recomendable que los estudiantes tengan conocimientos de Lógica, Aritmética, Álgebra y principios elementales de Geometría para que puedan entender los problemas que vamos a proponer para su análisis y entendimiento.

Antes de redactar un programa para que sea ejecutado por una computadora, es necesario hacer un raciocinio con operaciones aritméticas y lógicas. Este razonamiento matemático se fortalece utilizando herramientas adicionales como la tabla analítica, el diagrama de flujo y el algoritmo utilizando pseudocódigos.

Con la utilización de estas herramientas, en forma minuciosa y con orden, se procederá a escribir el programa en función del algoritmo según el criterio del programador.

El método que llevará al estudiante a la elaboración de programas para ser ejecutado en las computadoras se denomina Método Baltex.

El Método Baltex es un camino seguro para lograr la construcción de programas computarizados de una forma exitosa, coherente y ordenada, basándose en cuatro pasos que son los siguientes:

Paso 1: **B**osquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos.

Paso 2: **A**nalizar y definir la arquitectura del diagrama de flujo.

Paso 3: Leer y traducir el contenido de cada símbolo del diagrama de flujo a pseudocódigos para configurar el algoritmo.

Paso 4: Exportar el pseudocódigo a un lenguaje de programación de computadoras.

El lenguaje de programación que se utiliza en esta tesis, es lenguaje de Programación C, de amplio uso mundial, ya que casi todo el software que se ha creado para ser usado, en las organizaciones, está escrito basándose en este lenguaje y fue instalado en las tablets y en los celulares inteligentes con Sistema Operativo Android, ya que muchos estudiantes no contaban con una computadora personal para estudiar y practicar. La asignación económica otorgada por el Programa Nacional Beca 18 para adquirir una computadora portátil con el software requerido para el curso demoró en llegar en las primeras semanas de estudio. El lenguaje de Programación C es muy parecido al lenguaje de Programación Java, por lo que no hubo ningún impedimento para usarlo como herramienta alternativa complementaria, en cada sesión de tutoría.

El entorno de desarrollo que se ha utilizado para codificar los programas desarrollados es el wxDev-C++ (<http://wxdsgn.sourceforge.net/>), de naturaleza libre creado por Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA.

Todos los programas desarrollados en código c están en el formato Win32 Console, (Ventana de comandos de MS-DOS) por la simplicidad didáctica.

Como los estudiantes del Programa Nacional Beca 18 necesitan abordar los procesos lógicos y aritméticos no se considera en esta investigación el uso de un lenguaje con interfaces gráficas de colores que se pueden elaborar con otros entornos visuales tales como Eclipse Kepler o C Sharp ya que wxDev-C++ requiere de pocos recursos de Hardware que permite elevar el nivel cognitivo de los

estudiantes en la actividad analítica, previa a la codificación del programa en un lenguaje de programación. Cuando un estudiante domine el funcionamiento de las instrucciones de un lenguaje de programación podrá utilizar sin dificultad cualquier software de programación visual.

En la fase de aprendizaje, la persona que emplee el Método Baltex para construir programas en código c, encontrará algún tipo de redundancia en los tres pasos finales, que son necesarios para dar confianza, seguridad a la propuesta planteada para tener la certeza que se están haciendo bien el procedimiento.

Las personas más experimentadas en tópicos de programación no suelen graficar los diagramas de flujo; sin embargo, al plantear el problema, sí detallan una serie de pasos a realizarse que no es comprendida por una persona sin experiencia en realización de este programa. Para evitar confusiones, se recomienda la utilización del diagrama de flujo que visualmente brinda un panorama espacial favorable para esta actividad y facilita la comprensión de todas las acciones que se realizan en un programa de esta naturaleza.

Una persona sin experiencia, desde esta óptica, no debe desarrollar un programa sin realizar un análisis que sea expresado, posteriormente, en un diagrama de flujo, puesto que es una herramienta necesaria que ayuda a plantear una solución.

Cada programa redactado está alineado con la habilidad matemática del programador y ante un problema siempre se hallarán diferentes raciocinios y muchas formas de resolverlo, todos válidos si se llega al objetivo final, que depende de cada punto de vista y que nutre la habilidad personal a través del entrenamiento permanente y que busca la mejora continua.

Esperando que este aporte personal, a través de estas técnicas analíticas pueda contribuir con su formación profesional y que los problemas presentados sean de mucha utilidad.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA DE LA INVESTIGACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

El Programa Nacional Beca 18 se instauró en el mes de octubre del año 2011 por el Gobierno Peruano cuando era Presidente del Gobierno Peruano el Sr. Ollanta Humala Tasso entre los años 2006 y 2011. Este programa ha estado dirigido por el Programa Nacional de Becas y Crédito Educativo (Pronabec) del Ministerio de Educación. Como lo señala COTLER (2016) el Programa tiene por objeto promover la capacitación de jóvenes provenientes de las capas más pobres, a fin de que emprendan un proceso de movilidad social, conjuntamente con sus familiares. Actualmente alrededor de 50 mil jóvenes gozan de esta oportunidad.

De acuerdo con los datos proporcionados por el Ministerio de Educación a través de Estadística de la Calidad Educativa (ESCALE) del año 2014, el porcentaje de la población entre 17 y 18 años no pobre que ha concluido la Educación Secundaria es de 75.7%, mientras que para los pobres no extremos es de 51.9% y para los de pobreza extrema es de 30.3%.

Es importante recalcar que los jóvenes que integran el Programa Nacional Beca 18 pertenecen a la población del Perú que vive en pobreza extrema. Estos estudiantes han logrado finalizar la Educación Secundaria con mucho sacrificio, y serias carencias y con altas calificaciones, en un ambiente caracterizado por altas tasas de deserción escolar, y que radican en un medio rural.

Según las directivas del Ministerio de Educación, el Pronabec y los centros de estudios deben prestarles la ayuda necesaria mediante un sistema de tutorías y servicios psicopedagógicos a los estudiantes de Beca 18 en forma permanente.

LIPSKY (2010) afirma que los tutores y profesores son actores claves en la implementación del contenido educativo de estos estudiantes especiales.

Inicié mi desarrollo profesional en el Departamento de Educación de IBM del Perú en el año de 1995 y en un primer momento, mi responsabilidad fue capacitar en cursos de alto contenido técnico a profesionales que no sabían cómo operar una computadora. No había Internet aún. Es así como los primeros estudiantes eran personas mayores. Tuve la oportunidad de desarrollar con total libertad, técnicas para enseñar conceptos difíciles con figuras, dibujos y analogías que han permitido lograr un aprendizaje significativo en las personas con las que me relacioné. Para esa labor fui contratado.

Desde ese año, he estado vinculado a la Educación Superior y he ido aprendiendo, en el camino, ciertas competencias. Tal como lo señala (TEJADA 2013), el profesor es un profesional que antes de iniciarse en la docencia se formó en otra disciplina distinta a la pedagógica, didáctica o afín de la educación.

Posteriormente, en el año 2003, en la Universidad de San Martín de Porres, inicié mi actividad docente preparando cursos de Computación para las Escuelas Profesionales de Ingeniería de Computación y Sistemas e Ingeniería Industrial donde los estudiantes ya no eran mayores como los de la experiencia IBM, donde contacté directamente con el modelo universitario, respecto de la enseñanza-aprendizaje, desarrollé diversas investigaciones para que los estudiantes asimilen conceptos complejos utilizando nuevas formas de abordar el razonamiento lógico.

En el año 2013, fui seleccionado para enseñar la estructura de la Computación Básica en el Curso de Introducción a la Algoritmia en la Institución Educativa Cibertec para los estudiantes del Programa Nacional Beca 18. Acepté el reto, sabía que podía hacer algo por esos jóvenes. El recibir a estos estudiantes de sectores menos favorecidos que necesitan mucha orientación, tutoría y refuerzo educativo impulsó mi decisión de estudiar la Maestría en Educación en la mención: Orientación y Tutoría en la Facultad de Teología Pontificia y Civil de Lima. El curso de Introducción a la Algoritmia se dicta durante diecisiete semanas. Ello significó aunar esfuerzos en las explicaciones de tópicos y

contenidos lógico-matemáticos que eran difícilmente comprendidos y asimilados por los estudiantes.

En las sesiones de aprendizaje de la asignatura, se emplearon las diapositivas institucionales y los ejercicios propuestos fueron materiales que estaban preparados para estudiantes con experiencia y no para novatos.

Además se observó una carencia de una metodología orientada a los estudiantes del Programa Nacional Beca 18 para comprender el funcionamiento de una instrucción de un lenguaje de programación y lograr la articulación con operaciones lógico-matemáticas para realizar un programa básico que sea ejecutado por la computadora, escrito en un lenguaje de programación.

Según CHECA (2011), profesora de la Universidad de Lima, el proceso de aprender a programar es complejo, es una actividad cognitiva de alto nivel. Por lo tanto, al enseñar se deben desarrollar habilidades del pensamiento lógico. La enseñanza de la programación comúnmente se centró en el desarrollo de algoritmos y en el manejo de la sintaxis de un lenguaje de programación y se descuidó otros importantes dominios de habilidades. En estos cursos, la tasa de desaprobación es alta, la que genera elevados niveles de frustración en los actores educativos.

Asimismo el pensamiento de los investigadores Salgado, A.; Gorina, A. & Alonso, I. (2013) manifiestan que hay necesidad de iniciar a los alumnos en un camino que los lleve a utilizar técnicas de programación eficientes. De aquí la importancia de poner énfasis en el diseño previo que debe hacerse para desarrollar programas en un lenguaje específico. Una estrategia válida, en esta dirección, es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos.

Los estudiantes del Programa Nacional Beca 18 tienen una diversidad de niveles de estudios que hay que nivelarlos. Hay estudiantes notables y disciplinados. También existen otro tipo de estudiantes con deficiencias

cognitivas y que requieren un proceso de orientación y tutoría permanente en diversas etapas del aprendizaje de la Computación.

Se presentan debilidades en el dominio de tópicos de Aritmética, Álgebra, Geometría, Lógica y Razonamiento Matemático, por tanto no realizan actividades y ejercicios que deben ser entendidos y desarrollados por ellos mismos para superar estas carencias.

Se registraron las siguientes dificultades en los estudiantes del Programa Nacional Beca 18 quienes se matricularon en el curso de Introducción a la Algoritmia:

- Elevada demanda de horas de tutoría grupal (Grupo de 8 estudiantes).
- Repetición continua de los mismos temas realizados en horas de tutoría grupal ya desarrollados.
- Dificultad para identificar variables que almacenaran datos ingresados.
- Poco dominio de la técnica para distinguir qué datos se necesitan para elaborar los procesos necesarios para solucionar problemas.
- Desconocimiento para agrupar los datos que iban a ser procesados en la computadora.
- Imposibilidad para identificar los elementos de cada problema.
- Carencia de un planteamiento analítico.
- No tenían el software instalado en su computadora personal.
- Pocas horas de práctica en sus casas o lugares donde se hospedaban.
- Carencia de orden para elaborar el proceso de realización de un programa.
- No recordaban los comandos para programar en el software correspondiente.
- Lentitud para llegar al planteamiento explícito del problema.
- Poca comprensión del problema.

Estas dificultades motivaron la realización de la presente investigación, cuyo objetivo fue diseñar un nuevo método de aprendizaje de la computación básica (Método BALTEX), para que los estudiantes del Programa Nacional Beca 18 puedan adquirir las competencias que ofrece el curso de Introducción a la Algoritmia en 17 semanas y adquirir la habilidades y capacidades, a fin de no perder la beca.

1.2 FORMULACIÓN DEL PROBLEMA

1.2.1 PROBLEMA GENERAL: De acuerdo con lo expuesto anteriormente, se formula la siguiente pregunta:

¿El empleo del Método BALTEX, tiene efectos significativos sobre el aprendizaje procedimental de los estudiantes de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?

1.2.2 PROBLEMAS ESPECÍFICOS

- ¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad de reflexión, entendimiento y planificación de los estudiantes de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?
- ¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad de adaptar un panorama visual de los estudiantes de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?
- ¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad redacción de pseudocódigos de los estudiantes de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?
- ¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad de programación de los estudiantes de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?

CAPÍTULO II: MARCO TEÓRICO

2.1 ANTECEDENTES

Durante muchos años, la enseñanza de la programación, estaba centralizada en la escritura de algoritmos y en la posterior codificación del algoritmo en un lenguaje de programación, sin una estrategia para el aprendizaje didáctico, pedagógico y significativo.

La complejidad de la actividad de aprendizaje de la programación, produce la necesidad de construir un camino dotado de diversas técnicas, para que el estudiante pueda asimilar la experiencia de la actividad de la programación, poniendo énfasis en la arquitectura del diseño y no solo que el estudiante logró elaborar algoritmos y programas bajo el enfoque tradicional.

JOYANES, L. (2008), afirma que: “para poder identificar y definir bien un problema es conveniente responder a las siguientes preguntas:

- ¿Qué entradas se requieren? (tipo de datos con los cuales se trabaja y cantidad).
- ¿Cuál es la salida deseada? (tipo de datos de los resultados y cantidad).
- ¿Qué método produce la salida deseada?
- Requisitos o requerimientos adicionales y restricciones a la solución”. (p.47).

En la cita anterior, se recomienda diseñar el algoritmo y luego la codificación en un lenguaje de programación sin profundizar en la orientación detallada, de los pasos que deben dar los estudiantes que, por primera vez, abordan estos conceptos de naturaleza técnica, como es el caso de los estudiantes del Programa Nacional Beca 18.

VILLADIEGO, A., LÓPEZ, J. & SIERRA, I. (2015), señalan lo siguiente:

“Sin embargo, si se indaga frente a las metodologías para la enseñanza de las formas de creación utilizando la programación de computadores como paradigma, encontramos un consenso entre los estudiantes que las consideran una temática complicada, poco motivadora y sin ninguna aplicación práctica en las demás áreas del conocimiento, ni en su cotidianidad. Es evidente que nuestros estudiantes utilizan diversos medios para obtener información, aprender y entretenerse; los niños, niñas y adolescentes de hoy tienen contacto con el mundo de una forma gráfica, inmediata, y vivencial, usando diversas tecnologías, son capaces de clasificar qué contenidos son relevantes y darles total atención, incluso si eso implica hacer a un lado los contenidos de la escuela. Centrar la atención de los estudiantes para motivarlos hacia la creación usando tecnología puede llegar a plantearse como un reto debido en muchos casos a que se sigue enseñando los principios básicos de la programación de computadores utilizando el texto como único medio, lo cual puede ser frustrante, algunas veces aburrido, o poco motivador para las expectativas de los estudiantes. Incluso se aborda la enseñanza de la programación en las aulas utilizando las mismas estrategias metodológicas y herramientas que se utilizaron durante la formación de pregrado o postgrado de los mismos docentes.” (p.34).

Esta afirmación resalta que, es imperativo, hacer una reestructuración, de las estrategias didácticas de aprendizaje de la Computación.

La revista cubana *Educare* publicó, en el año 2013, un informe de investigación que propone un modelo de la dinámica lógico-algorítmica para la resolución de problemas de programación computacional.

SALGADO, A., GORINA, A. & ALONSO, I. (2013) encontraron en sus investigaciones en la Universidad del Oriente de Cuba que la principal novedad del modelo propuesto para entender un problema, radica en haber revelado la lógica integradora entre la representación matemática de la situación problemática y la generalización de la representación con pseudocodificado.

La investigación se estructuró en cuatro dimensiones:

- a) Construcción lógico-matemática.
- b) Orientación matemático-algorítmica.
- c) Estructuración algorítmico-computacional y
- d) Validación algorítmica-computacional.

Entre sus hallazgos, plantean técnicas para capturar, almacenar, procesar información y lograr la interacción humano-computador.

Adicionalmente, expresaron en su investigación, que el nivel de conocimientos previos que exige que todo curso de programación que asuma el estudiante se enfrente a una ardua disciplina de trabajo autónomo donde es necesario que se dedique buena parte del tiempo a la comprensión de los conceptos básicos enlazando la matemática y la lógica.

2. 2 BASES TEÓRICAS

2.2.1 LA ORIENTACIÓN EDUCATIVA

Según los estudios de PARRAS et al. (2009), la concepción actual de la orientación educativa determina que su función principal es la prevención.

2.2.2 LOS PRINCIPIOS DE LA ORIENTACIÓN EDUCATIVA

Se distinguen los siguientes principios de la orientación educativa: principio de prevención, principio de desarrollo, principio de intervención social y principio de fortalecimiento personal.

2.2.2.1 PRINCIPIO DE PREVENCIÓN

Está basado en la necesidad de preparar a las personas para la superación de las diferentes crisis de desarrollo.

Su objetivo es promocionar conductas saludables y competencias personales como las relacionadas con la inteligencia interpersonal y la intrapersonal, con el fin de evitar la aparición de problemas.

Esta orientación adquiere un carácter proactivo que se anticipa a la aparición de todo aquello que suponga un obstáculo al desarrollo de la persona y le impida superar crisis implícitas en el mismo.

El objetivo de la prevención es desarrollar la competencia funcional y social de la persona, su capacidad para afrontar situaciones y su fortalecimiento interior (*empowerment*).

Las características de las acciones preventivas son las siguientes:

- Proactiva; actúa con anterioridad a la aparición del problema.
- Dirigida a grupos.
- Planteamiento ecológico sistémico, teniendo muy en cuenta el entorno.
- El objetivo es reducir los factores de riesgo e incrementar los elementos que favorecen la defensa y la protección ante la crisis.
- Asume la multiculturalidad.
- Se orienta al fortalecimiento personal.
- Pretende la disminución de la frecuencia y la tasa de incidencia de los problemas en la población.
- Incorpora la colaboración conceptual y procedimental en la intervención, de manera que los destinatarios son agentes activos del cambio.
- Palia las condiciones desfavorables del contexto.

2.2.2.2 PRINCIPIO DE DESARROLLO

En el marco de este principio, la intervención supone un proceso mediante el que se acompaña al individuo durante su desarrollo, con la finalidad de lograr el máximo crecimiento de sus potencialidades.

Tiene un doble objetivo:

Pretende dotar de competencias necesarias a la persona para que pueda afrontar las demandas de las etapas evolutivas (enfoque madurativo).

Ha de proporcionar situaciones de aprendizaje vital que faciliten la reconstrucción y progreso de los esquemas conceptuales del mismo (enfoque constructivista).

2.2.2.3 PRINCIPIO DE INTERVENCIÓN SOCIAL

La intervención orientadora debe estar dirigida tanto a modificar aspectos relativos a la organización y funcionamiento del centro educativo (organización de grupos, sistemas de evaluación, metodología, etc.) como del contexto social del alumno o alumna (becas, ayudas, asistencia social, etc.). Según este autor (PARRAS et al), no deben aceptarse posturas pasivas enmascaradas en argumentos dudosos como la competencia del profesional de la orientación.

La acción orientadora y la intervención, debe seguir dos condiciones:

- Analizar el desarrollo y la conducta de los destinatarios en el marco de los sistemas en los que actúa la persona a través de procesos de socialización en valores, normas, expectativas y metas.
- Diseñar una intervención que tenga como objetivo eliminar los efectos negativos de los ambientes de aprendizaje sobre las personas.

2.2.2.4 PRINCIPIO DE FORTALECIMIENTO PERSONAL (EMPOWERMENT)

Es el proceso mediante el cual las personas, las organizaciones o las comunidades adquieren un dominio sobre los asuntos de interés que le son propios.

Solo se producirá una orientación para el fortalecimiento personal cuando exista un compromiso del orientador u orientadora para cambiar las estructuras y sistemas que están impidiendo el desarrollo de los menos favorecidos.

Se puede establecer un cierto paralelismo entre el significado de *empowerment* en el contexto de la empresa y en el contexto educativo, entendido este como una institución que tiende hacia la calidad y la excelencia. Para la empresa, el *empowerment* es una estrategia de motivación, en el trabajo, que consiste en la delegación o transmisión de mayor capacidad de decisión a los empleados para que se sientan más implicados en la actividad y los objetivos de la empresa.

2.2.3 LA TUTORÍA

Como lo señala el MINISTERIO DE EDUCACIÓN DEL PERÚ & DIRECCIÓN DE TUTORÍA Y ORIENTACIÓN EDUCATIVA (2007), la tutoría es una modalidad de la orientación educativa. De acuerdo con el Diseño Curricular Nacional es concebida como “un servicio de acompañamiento socioafectivo, cognitivo y pedagógico de los estudiantes. Es parte del desarrollo curricular y aporta al logro de los aprendizajes y a la formación integral, en la perspectiva del desarrollo humano” (página 10).

2.2.4 PROCESO DE APRENDIZAJE

Es un proceso que debe ser canalizado cuidadosamente para que el estudiante adquiera un nuevo comportamiento, en su conducta, asimilando conocimientos, facilitados por el docente a través de la lectura de información, desarrollo de competencias con base a la experiencia significativa. El proceso de aprendizaje se da en forma permanente.

Según YAÑEZ (2016), en el proceso de aprendizaje, el apoyo del docente es vital para motivar al estudiante. Se debe tener presente

que el fracaso pudiera ocurrir para aprender de los errores, reflexionar y corregir estrategias para incrementar su crecimiento intelectual.

Todo estudiante debe desarrollarse con interés, atención, dedicación, aprehendiendo las ideas importantes que les serán útiles en su actividad laboral.

Un reto para nosotros, los docentes, es que un estudiante puede tener capacidad crítica, según la asimilación y la aplicación de sus habilidades adquiridas en su centro de formación técnica o universitaria en situaciones en las que se requiera.

2.2.5 EL APRENDIZAJE DE CONTENIDOS PROCEDIMENTALES

ZAVALA, A. (2000), se refiere a los contenidos procedimentales señalando lo siguiente: "un contenido procedimental - que incluye, entre otras cosas, las reglas, las técnicas, los métodos, las destrezas o habilidades, las estrategias, los procedimientos- es un conjunto de acciones ordenadas y finalizadas, es decir, dirigidas a la consecución de un objetivo."

El aprendizaje procedimental se refiere a la adquisición y/o mejora de nuestras habilidades, a través de la ejercitación reflexiva en diversas técnicas, destrezas y/o estrategias para hacer cosas concretas. Se trata de determinadas formas de actuar cuya principal característica es que se realizan de forma ordenada: "Implican secuencias de habilidades o destrezas más complejas y encadenadas que un simple hábito de conducta".

¿Qué condiciones son fundamentales para el aprendizaje de contenidos procedimentales?

- La realización de las acciones que conforman los procedimientos es una condición fundamental para el aprendizaje: se aprende a hablar, hablando; a dibujar, dibujando; a observar, observando.
- La ejercitación múltiple es necesaria para el aprendizaje de una técnica, no basta con realizar alguna vez las acciones del contenido procedimental, hay que realizar tantas veces como sean necesarias las diferentes acciones o pasos de dichos contenidos de aprendizaje.
- La reflexión sobre la misma actividad es un elemento imprescindible que permite tomar conciencia de la actuación. No basta con repetir el ejercicio habrá que ser capaz de reflexionar sobre la manera de realizarlo y sobre las condiciones ideales de su uso. Esto implica realizar ejercitaciones, pero con el mejor soporte reflexivo que nos permita analizar nuestros actos, y por consiguiente, mejorarlos. Para ello hace falta tener un conocimiento significativo de contenidos conceptuales asociados al contenido procedimental que se ejercita o se aplica. Así, por ejemplo, se puede revisar una composición a partir de un conjunto de reglas morfosintácticas que le permitan establecer errores y hacer modificaciones posteriores.
- La aplicación, en contextos diferenciados, se basa en el hecho de que aquello que hemos aprendido será más útil en la medida en que se pueda utilizar en situaciones siempre imprevisibles. Las ejercitaciones han de realizarse en contextos diferentes para que los aprendizajes puedan ser utilizados en cualquier ocasión.

2.2.6 CAPACIDAD DE REFLEXIÓN, ENTENDIMIENTO Y PLANIFICACIÓN

Según DE GUZMAN, M. (2006), la capacidad de reflexión, entendimiento y planificación implica una serie de actividades descritas a continuación:

- Leer tranquilamente el enunciado. Puede ser necesario que se lea varias veces, hasta estar seguro de haberlo entendido y de que no se ha escapado ningún dato interesante. Ha de tener muy claro en qué consiste, qué conoce, qué se le pide, cuáles son las condiciones. Esto es imprescindible para afrontar el problema con garantías de éxito.
- Cuando se está seguro de haber entendido bien el problema y se cuenta con toda la información necesaria, es el momento de elegir una estrategia para resolverlo. Existe una gran variedad de estrategias que conviene conocer y que se practique continuamente para mejorar la capacidad de comprender problemas.
- Cuando se elige una estrategia que parece adecuada, se debe trabajarla con decisión y no se debe abandonar frente a la primera dificultad. Pero si se aprecia que las cosas se complican demasiado y no hay solución, se debe volver al paso anterior y probar con una estrategia diferente. Por lo general, hay varias formas de llegar a la solución y no acertar siempre con la más apropiada al primer intento.

2.2.7 CAPACIDAD DE ADAPTAR UN PANORAMA VISUAL

Muchos problemas de Computación requieren ser analizados desde otra óptica para comprenderlos mejor. A veces, una imagen vale más que mil palabras. En el dibujo o esquema que se haga, se debe

incorporar los datos realmente importantes y prescindir de los demás. El diagrama de flujo es la mejor forma de tener un panorama visual del problema completo que se va a solucionar. Se deben utilizar los símbolos adecuados para representar los datos que se ingresan por el teclado, el proceso de transformación de los mismos hasta la visualización en la pantalla de la computadora de los datos esperados. El objetivo es que sirva de apoyo para avanzar en la construcción del programa.

2.2.8 CAPACIDAD DE REDACCIÓN DE PSEUDOCÓDIGOS

Se refiere a la capacidad de expresar las órdenes que se le dé a la computadora en un lenguaje informal, aceptado por la comunidad de expertos en Computación para describir actividades generales que debe realizar una computadora.

2.2.9 CAPACIDAD DE PROGRAMACIÓN

Es la habilidad para trasladar las órdenes que se le debe dar a una computadora en un lenguaje de programación determinado según sus reglas sintácticas.

2.2.10 REPRESENTACIONES ALGEBRAICAS

El lenguaje algebraico consta principalmente de las letras de alfabeto y algunos vocablos griegos. La principal función de lenguaje algebraico es elaborar un lenguaje que ayude a generalizar las diferentes operaciones que se desarrollan dentro de la aritmética. Por ejemplo, si se restan dos números cualesquiera basta con decir $x - y$; donde la letra x indique que es un número cualquiera de la numeración que conocemos, y de la misma manera que a significa un número cualquiera de la numeración.

Cuando se inicia en el estudio del Álgebra se necesita representar mediante notaciones especiales diversos enunciados matemáticos para elaborar estructuras algebraicas que puedan resolver un problema determinado. Eso se muestra en la Tabla 01.

2.2.11 ALGUNAS REPRESENTACIONES ALGEBRAICAS

Ejemplo de algunas representaciones algebraicas

Tabla 01

ENUNCIADO MATEMÁTICO	REPRESENTACIÓN ALGEBRAICA
Un número cualquiera	X
El doble de un número cualquiera	2Y
Tres números consecutivos	FORMA 1: Z, Z+1, Z+2 FORMA 2: Z-1, Z, Z+1 FORMA 3: Z-2, Z-1, Z
El triple de un número más el doble de otro número cualquiera.	3N

2.2.12 REPRESENTACIONES DE DATOS EN PSEUDOCÓDIGOS

Al igual que el estudio del Álgebra necesita sus representaciones. El estudio de la Computación también requiere sus representaciones considerando que los datos siempre se van a almacenar en variables.

Los datos que se van a representar son de cinco tipos:

- Cadena de texto (palabras o frases)
- Carácter (un símbolo del teclado)
- Lógico (verdadero [1] o falso[0])
- Entero (números enteros entre -32768 y 32767)
- Real (números reales hasta con 15 dígitos de precisión).

En la tabla 02, se muestran las representaciones de datos en pseudocódigos que permitan indicar a la computadora las instrucciones que se deben ejecutar en un programa.

2.2.13 LISTA DE PSEUDOCÓDIGOS

La siguiente lista de pseudocódigos se ha preparado para que los estudiantes encuentren un significado en las representaciones que deben diseñar un programa.

Lista de pseudocódigos

Tabla 02

ENUNCIADOS A REPRESENTAR	PSEUDOCÓDIGO
Una variable que almacenará un dato de tipo texto sin contenido	Forma 1: TEXTO a Forma 2: TEXTO a = " "
Una variable que almacenará un dato de tipo texto con contenido	TEXTO b= "Hola"
Una variable que almacenará un dato de tipo caracter sin contenido	Forma 1: CARACTER b Forma 2: CARACTER b = ' '
Una variable que almacenará un dato de tipo caracter con contenido	CARACTER c= 's'
Una variable que almacenará un dato de tipo entero sin valor inicial	ENTERO d
Una variable que almacenará un dato de tipo entero con valor inicial 16	ENTERO e=16
Una variable que almacenará un dato de tipo real sin valor inicial	REAL f
Una variable que almacenará un dato de tipo real con valor inicial 17.30	REAL g=17.30
Una variable que almacenará un dato de tipo lógico sin valor inicial	LOGICO h
Una variable que almacenará un dato de tipo lógico con valor inicial VERDADERO	LOGICO i=VERDADERO
Una variable que almacenará un dato de tipo lógico con valor inicial FALSO	LOGICO i=FALSO
Capturar un valor ingresado por el teclado (Leer un valor del teclado) de la computadora y almacenarlo en una variable de tipo entero sin contenido. Nota: Se debe de declarar previamente: ENTERO d	LEER d
Visualizar, mostrar o escribir en la pantalla de la computadora el contenido de una variable de tipo texto Nota: Se debe de declarar previamente: TEXTO b="Hola"	ESCRIBIR b
Visualizar, mostrar o escribir en la pantalla de la computadora la palabra "Bienvenido"	ESCRIBIR "Bienvenido"
Salto de línea (Punto y aparte)	SALTOLINEA()
Tabulador	TABULADOR()

2.2.14 EQUIVALENCIAS ENTRE PSEUDOCÓDIGOS Y CÓDIGO C

Las equivalencias permitirán convertir los pseudocódigos en instrucciones de un lenguaje de programación para construir un programa.

Equivalencias entre pseudocódigos y código C

Tabla 03

PSEUDOCÓDIGO	CÓDIGO C
TEXTO a	string a ;
TEXTO a = " "	string a= " ";
TEXTO b= "Hola"	string b="Hola";
CARACTER b	char b ;
CARACTER b = ' '	char b= ' ';
CARACTER c= 's'	char c='s';
ENTERO d	int d;
ENTERO e=16	int e=16;
REAL f	double f;
REAL g=17.30	double g=17.30;
LOGICO h	bool h;
LOGICO i=VERDADERO	bool i=true;
LOGICO i=FALSO	bool i=false;
LEER d	cin>>d;
ESCRIBIR b	cout<<b;
ESCRIBIR "Exitos" (*) (*): El texto que se encuentra entre comillas dobles se está almacenando en una variable interna cuyo nombre no conoce el programador pero sí conoce el editor del lenguaje de programación.	cout<<"Exitos";
SALTOLINEA ()	cout<<endl; o cout<<"\n";

TABULADOR ()	cout<<"\t";
---------------	-------------

2.2.15 REPRESENTACIONES DE DATOS EN FORMA GRÁFICA

Los datos se introducen en variables de dos formas; una cuando el programador asigna directamente los valores de los datos en el momento de redactar el código del programa o cuando se capturan los datos por el teclado. Una herramienta muy importante es el diagrama de flujo.

DIAGRAMA DE FLUJO: Es la representación gráfica de las instrucciones que debe ejecutar la computadora para que funcione un programa de acuerdo con los objetivos del programador.

Tabla 04

RELACIÓN DE SÍMBOLOS DEL DIAGRAMA DE FLUJO. PARTE 1

NOMBRE DEL SÍMBOLO	SÍMBOLO	TEXTO DENTRO DEL SÍMBOLO	SE USA PARA:
RECTÁNGULO CON ESQUINAS REDONDEADAS		INICIO / FIN	Indicar el inicio o el final de un programa
RECTÁNGULO		PSEUDOCÓDIGOS, FÓRMULAS, ASIGNACIONES DE VALORES A VARIABLES	<ul style="list-style-type: none"> • Declarar variables • Establecer fórmulas • Asignar valores a variables.
PARALELOGRAMO		LA VARIABLE O VARIABLES QUE ALMACENARÁN DATOS INGRESADOS POR EL TECLADO, PREVIA DECLARACIÓN DE LAS MISMAS. (entrada de datos)	Capturar datos ingresados por el teclado (Leer datos ingresados por el teclado)

Tabla 05

RELACIÓN DE SÍMBOLOS DEL DIAGRAMA DE FLUJO. PARTE 2

NOMBRE DEL SÍMBOLO	SÍMBOLO	TEXTO DENTRO DEL SÍMBOLO	SE USA PARA:
HOJA DE IMPRESIÓN		LA VARIABLE O VARIABLES CUYO CONTENIDO SE DESEAN QUE SE VISUALICEN, MUESTREN O ESCRIBAN EN LA PANTALLA DE LA COMPUTADORA.	Visualizar, mostrar o escribir el contenido de las variables en la pantalla de la computadora.
ROMBO		PROPOSICIONES LÓGICAS	Establecer condiciones lógicas.
CONECTOR		CONECTORES	<p>Evitar que se crucen las líneas de un diagrama.</p> <p>Al colocarle un número dentro del símbolo indica que el diagrama continúa en la siguiente hoja.</p> <p>El diagrama de flujo en la siguiente hoja debe empezar con el mismo símbolo y con el mismo número.</p>
FLECHAS		NINGUNO. Solo si la flecha sale de un rombo debería incluirse encima de la flecha una letra V si la condición lógica es verdadera o la F si	Indicar el camino que siguen los datos en el diagrama de flujo.

		la condición lógica es falsa.	
--	--	-------------------------------	--

2.2.16 EL MÉTODO BALTEX

El Método Baltex ha sido diseñado para aprender las siete estructuras básicas para iniciar el aprendizaje de la Computación en su primera etapa. Las 7 estructuras básicas son:

- Estructura secuencial.
- Estructura de Selección de Datos “SI-SINO”
- Estructura de Selección de Datos “SI Anidado”
- Estructura de Selección de Datos “Cuando sea”
- Estructura de Repetición de Datos “Mientras”
- Estructura de Repetición de Datos “Para”
- Estructura de Repetición de Datos “Hacer-Mientras”

2.2.16.1 DEFINICIÓN DE LA ESTRUCTURA SECUENCIAL

La estructura secuencial es aquella que se inicia con el ingreso de datos necesarios para transformarlos a través de fórmulas aritméticas o lógicas y convertirlos en un resultado esperado.

EJEMPLO

Hacer un programa en código c que permita ingresar tres números por el teclado y los sume.

Figura 1: Enunciado a programar con la estructura secuencial

CONSTRUCCIÓN DEL PROGRAMA UTILIZANDO EL MÉTODO BALTEX

PASO 1 **B**osquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos. (Actividad creativa)

- El objetivo es construir un programa en código c, que permita ingresar tres números por el teclado y los sume.
- No se hace ninguna referencia si los números son enteros o reales, por lo que el programador debe decidir. En este caso, como los datos reales contienen a los números enteros, tendremos que utilizar cuatro variables para almacenar datos de tipo real. Tres de ellas contendrán los números enteros, ingresados por el teclado y en la cuarta variable se calculará la suma de los tres números enteros.
- Para ingresar cada uno de los tres números reales, se necesitan tres mensajes que permitan capturar esos números y un mensaje para indicar que se está mostrando la suma de los tres números

reales, por lo que se necesitará mostrar cuatro mensajes y se utilizarán cuatro variables para almacenar datos de tipo texto.

TABLA ANALÍTICA DE LA ESTRUCTURA SECUENCIAL

Declaración de variables	Entrada	Variables que se involucran mediante operaciones aritméticas (Proceso)	Salida
REAL N1=0	N1	SUMA=N1+N2+N3	SUMA
REAL N2=0	N2		
REAL N3=0	N3		
REAL SUMA=0			
TEXTO M1="Ingrese primer numero"			
TEXTO M2="Ingrese segundo numero"			
TEXTO M3="Ingrese tercer numero"			
TEXTO M4="La suma es"			

Nota: Las tildes son omitidas en el contenido de las variables de texto, ya que no las reconoce el wxDev-C++, software que se utilizará para redactar el programa en lenguaje C.

Tabla 06

PASO 2 Analizar y definir la arquitectura del diagrama de flujo. (Actividad creativa)

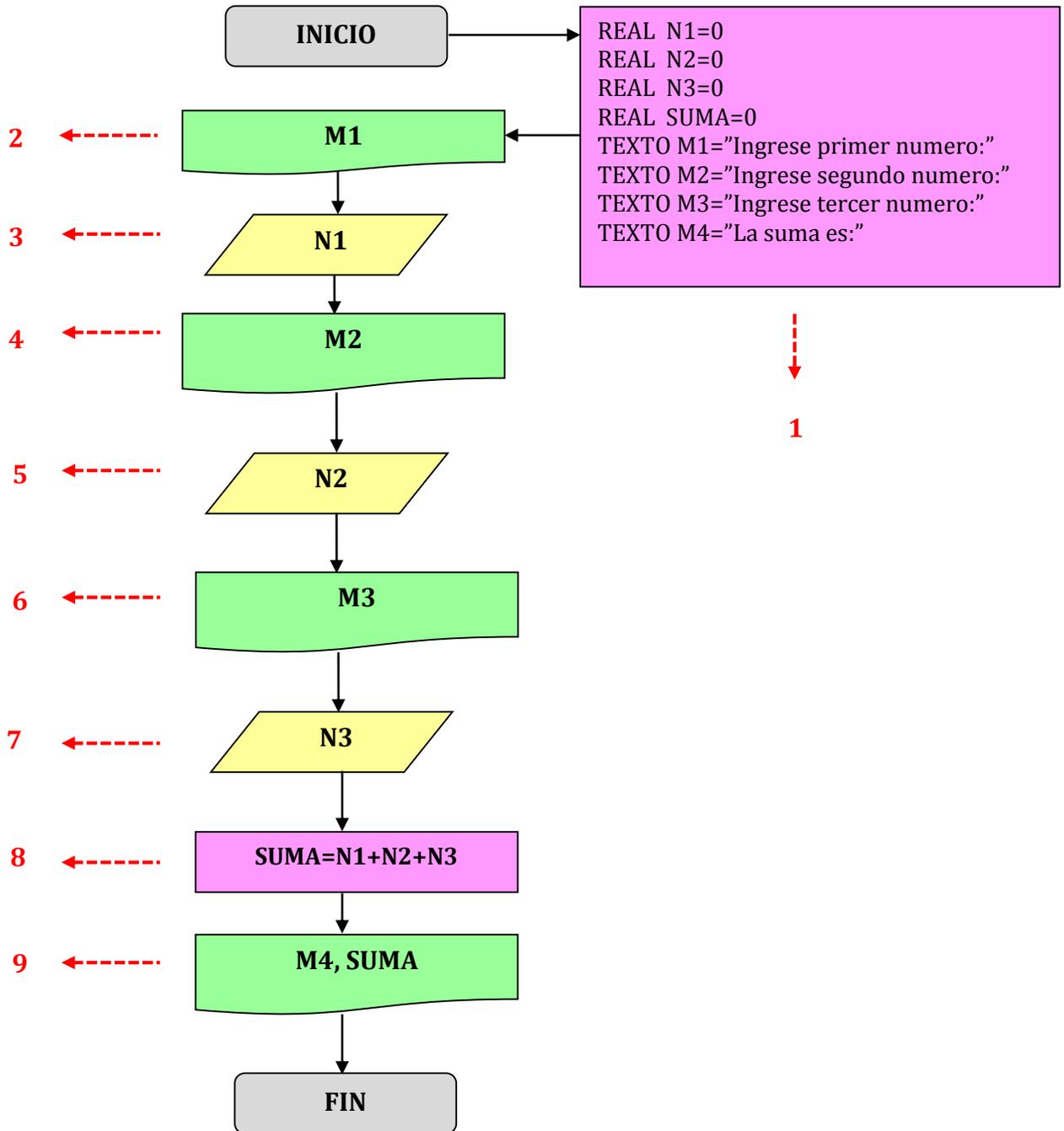


Figura 2: Diagrama de flujo de la estructura secuencial

PASO 3 Leer el contenido de

cada símbolo del diagrama de flujo y traducir a pseudocódigos.

(Actividad mecánica)

REFERENCIAS DEL DIAGRAMA DE FLUJO	ALGORITMO : PrgSuma
	ENTRADA: N1, N2, N3
	SALIDA: SUMA
	INICIO
	//Declaración de variables
1	REAL N1=0
1	REAL N2=0
1	REAL N3=0
1	REAL SUMA=0
1	TEXTO M1="Ingrese primer numero"
1	TEXTO M2="Ingrese segundo numero"
1	TEXTO M3="Ingrese tercer numero"
1	TEXTO M4="La suma es:"
	//Entrada
2	ESCRIBIR M1
3	LEER N1
4	ESCRIBIR M2
5	LEER N2
6	ESCRIBIR M3
7	LEER N3
	//Proceso
8	SUMA= N1+N2+N3
	//Salida
9	ESCRIBIR M4, SUMA
	FIN

Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura secuencial

Tabla 07

PASO 4 Exportar el pseudocódigo a un lenguaje de programación.
(Actividad mecánica)

ALGORITMO : PrgSuma	Programa en código C
ENTRADA: N1, N2, N3	#include <cstdlib>
SALIDA: SUMA	#include <iostream>
	using namespace std;
	int main(int argc, char *argv[])
INICIO	{
//Declaración de variables	//Declaración de variables
REAL N1=0	double N1=0;
REAL N2=0	double N2=0;
REAL N3=0	double N3=0;
REAL SUMA=0	double SUMA=0;
TEXTO M1="Ingrese primer numero"	string M1="Ingrese primer numero";
TEXTO M2="Ingrese segundo numero"	string M2="Ingrese segundo numero";
TEXTO M3="Ingrese tercer numero"	string M3="Ingrese tercer numero";
TEXTO M4="La suma es:"	string M4="La suma es:";
//Entrada	//Entrada
ESCRIBIR M1	cout<<M1;
LEER N1	cin>>N1;
ESCRIBIR M2	cout<<M2;
LEER N2	cin>>N2;
ESCRIBIR M3	cout<<M3;
LEER N3	cin>>N3;
//Proceso	//Proceso
SUMA= N1+N2+N3	SUMA= N1+N2+N3;
//Salida	//Salida
ESCRIBIR M4, SUMA	cout << M4<<SUMA;
FIN	system("PAUSE");
	return EXIT_SUCCESS;
	}

Correspondencia entre cada pseudocódigo y el código C asociado de la estructura secuencial

Tabla 08

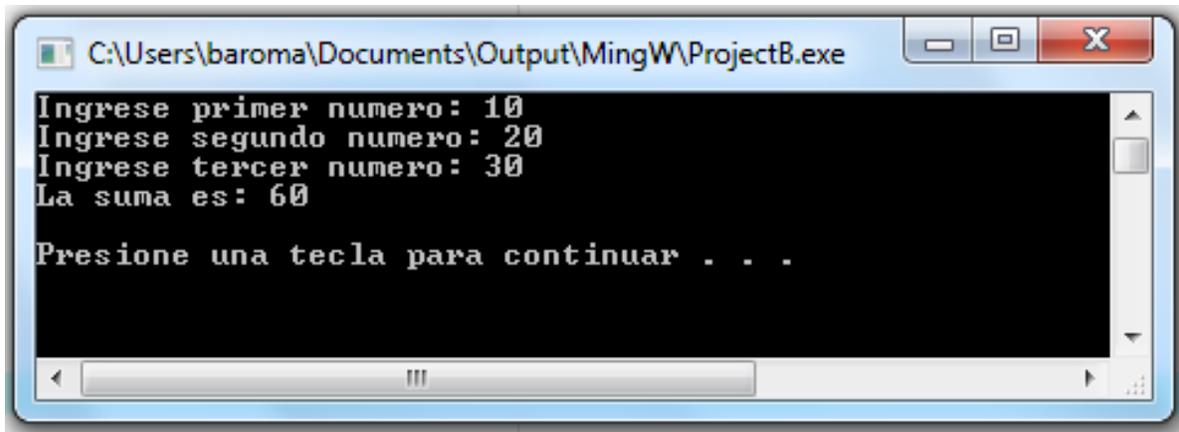
PROGRAMA EN CÓDIGO C REDACTADO EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

```
main.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
double N1=0;
double N2=0;
double N3=0;
double SUMA=0;
string M1="Ingrese primer numero: ";
string M2="Ingrese segundo numero: ";
string M3="Ingrese tercer numero: ";
string M4="La suma es: ";
//entrada
cout<<M1;
cin>>N1;
cout<<M2;
cin>>N2;
cout<<M3;
cin>>N3;
//proceso
SUMA= N1+N2+N3;
//salida
cout << M4<<SUMA<<endl;

system("PAUSE");
return EXIT_SUCCESS;
}
```

Figura 3: Programa en código C de la estructura secuencial

PROGRAMA DESPUÉS DE EJECUTARSE EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++



```
C:\Users\baroma\Documents\Output\MingW\ProjectB.exe
Ingrese primer numero: 10
Ingrese segundo numero: 20
Ingrese tercer numero: 30
La suma es: 60

Presione una tecla para continuar . . .
```

Figura 4: Programa ejecutado en la computadora de la estructura secuencial

2.2.16.2 DEFINICIÓN DE LA ESTRUCTURA DE SELECCIÓN DE DATOS SI-SINO

La estructura de selección de datos SI-SINO permite elegir una alternativa entre dos opciones.

La estructura de selección de datos completa requiere un dato de entrada para que se someta a una evaluación lógica. Si el resultado de esta evaluación lógica es verdadero, el dato seguirá su camino por la ruta marcada como verdadera, en caso contrario, si el resultado de la evaluación lógica es falso, el dato seguirá su camino por la ruta marcado como falso.

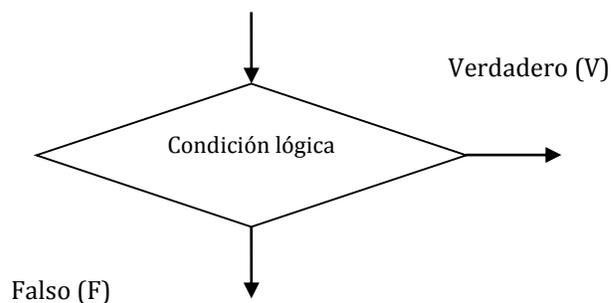


Figura 5: Esquema de comprensión de la estructura de selección de datos “SI-SINO”

Nota: Algunas veces suele considerarse realizar una acción para algunos casos especiales solamente cuando el resultado de la condición lógica es verdadero. Este caso permite utilizar la ESTRUCTURA DE SELECCIÓN DE DATOS SI SIMPLE, ya que no se realizará ningún proceso cuando la condición lógica es falsa.

Al parecer, es mucho más lógico y ordenado utilizar la ESTRUCTURA DE SELECCIÓN DE DATOS SI-SINO.

EJEMPLO

Hacer un programa en código c que permita capturar un número entero ingresado por el teclado y que permita mostrar en la pantalla de la computadora un mensaje si es un número par o impar.

Figura 6: Enunciado a programar con la estructura de selección de datos "SI-SINO"

CONSTRUCCIÓN DEL PROGRAMA UTILIZANDO EL MÉTODO BALTEX

PASO 1 Bosquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos. (Actividad creativa)

- El objetivo es construir un programa que permita capturar el valor entero de un dato ingresado por el teclado y determinar si es un número par o impar.
- Se hace referencia al tipo de dato que se quiere capturar por el teclado, que es de tipo entero.
- Para saber si el número es par se debe calcular el resto al dividirlo entre 2. Si el resto es cero, el número es par, si el resto es diferente a cero, el número es impar.

Ejemplo:

ENTERO a

Sea el número ingresado por el teclado 5

Por lo tanto $a=5$.

Si $a\%2 = 0$, el número es par

en caso contrario,

Si $a\%2 \neq 0$, el número es impar

En nuestro caso, $a=5$ es un número impar.

TABLA ANALÍTICA DE LA ESTRUCTURA DE SELECCIÓN DE DATOS SI-SINO

Declaración de variables	Entrada	Condición lógica	Resultado de la condición lógica		Variables que se involucran mediante operaciones de asignación (Proceso)	Salida
			V	F		
TEXTO vt1="Ingrese un numero entero:" TEXTO vt2=" " ENTERO num=0	num	(num%2=0)	X	---	vt2="es par"	ESCRIBIR (num,vt2)
			---	X	vt2="es impar"	

Nota: Las tildes son omitidas en el contenido de las variables de texto, ya que no las reconoce el wxDev-C++, software que utilizaremos para redactar el programa en lenguaje C.

Tabla 09

PASO 2 Analizar y definir la arquitectura del diagrama de flujo. (Actividad creativa)

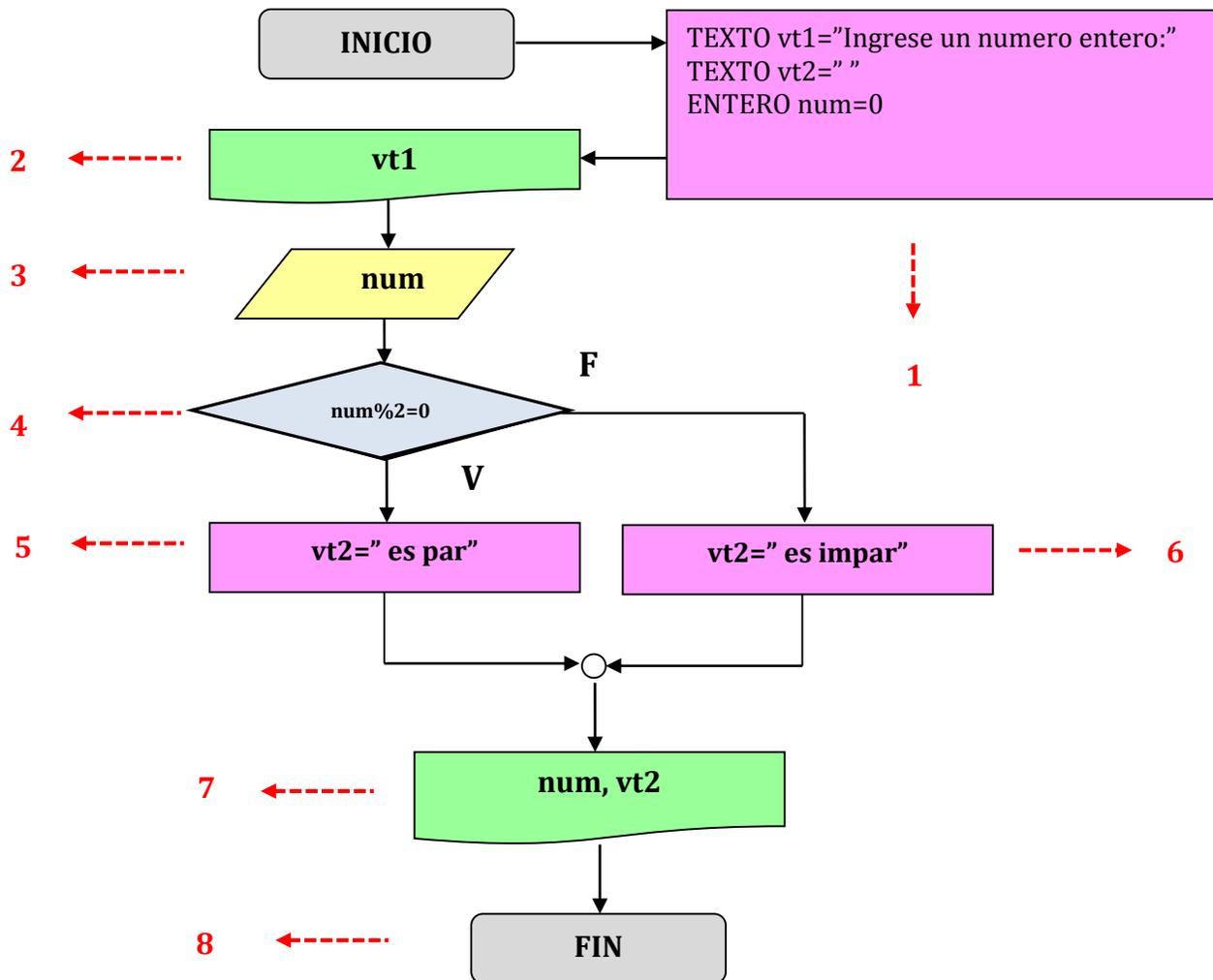


Figura 7: Diagrama de flujo de la estructura de selección de datos “SI-SINO”

PASO 3 Leer el contenido de cada

símbolo del diagrama de flujo y traducir a pseudocódigos. (Actividad mecánica)

REFERENCIAS DEL DIAGRAMA DE FLUJO	ALGORITMO : PrgNum
	ENTRADA: num
	SALIDA: num, vt2
	INICIO
	//Declaración de variables
1	TEXTO vt1="Ingrese un numero entero: "
1	TEXTO vt2=" "
1	ENTERO num=0
	//Entrada
2	ESCRIBIR vt1
3	LEER num
	//Condición lógica
4	SI (num % 2=0) ENTONCES
	//Proceso si condición lógica es V
5	vt2=" es par"
	SINO
	//Proceso si condición lógica es F
6	vt2=" es impar"
	FIN SI
	//Salida
7	ESCRIBIR num, vt2, SALTOLINEA ()
	FIN

Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de selección de datos "SI-SINO"

Tabla 10

PASO 4 Exportar el pseudocódigo a un lenguaje de programación. (Actividad mecánica)

ALGORITMO : PrgNum	Programa en código C
ENTRADA: num	#include <cstdlib>
SALIDA: num, vt2	#include <iostream>
	using namespace std;
INICIO	int main(int argc, char *argv[])
//Declaración de variables	//Declaración de variables
TEXTO vt1="Ingrese un numero entero:"	string vt1="Ingrese un numero entero:";
TEXTO vt2=" "	string vt2=" ";
ENTERO num=0	int num=0;
//Entrada	//Entrada
ESCRIBIR (vt1)	cout<< (vt1);
LEER num	cin>> (num);
//Condición lógica	//Condición lógica
SI (num % 2=0) ENTONCES	if (num % 2==0)
//Proceso si condición lógica es V	//Proceso si condición lógica es V
vt2=" es par"	{ vt2=" es par";}
SINO	else
//Proceso si condición lógica es F	//Proceso si condición lógica es F
vt2=" es impar"	{vt2=" es impar";}
FIN SI	
//Salida	//Salida
ESCRIBIR num, vt2, SALTOLINEA ()	cout<<num<< vt2<<endl;
FIN	system("PAUSE");
	return EXIT_SUCCESS;
	}

Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de selección de datos "SI-SINO"

Tabla 11

PROGRAMA EN CÓDIGO C REDACTADO EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

```
main.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    string vt1="Ingrese un numero entero:";
    string vt2=" ";
    int num=0;
    //entrada
    cout<< (vt1);
    cin>> (num);
    //condicion lógica
    if (num % 2==0)
    //proceso si condicion lógica es V
        { vt2=" es par";}
    else
    //proceso si condicion lógica es F
        {vt2=" es impar";}

    //salida
    cout<<num<< vt2<<endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Figura 8: Programa en código C de la estructura de selección de datos "SI-SINO"

PROGRAMA DESPUÉS DE EJECUTARSE EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

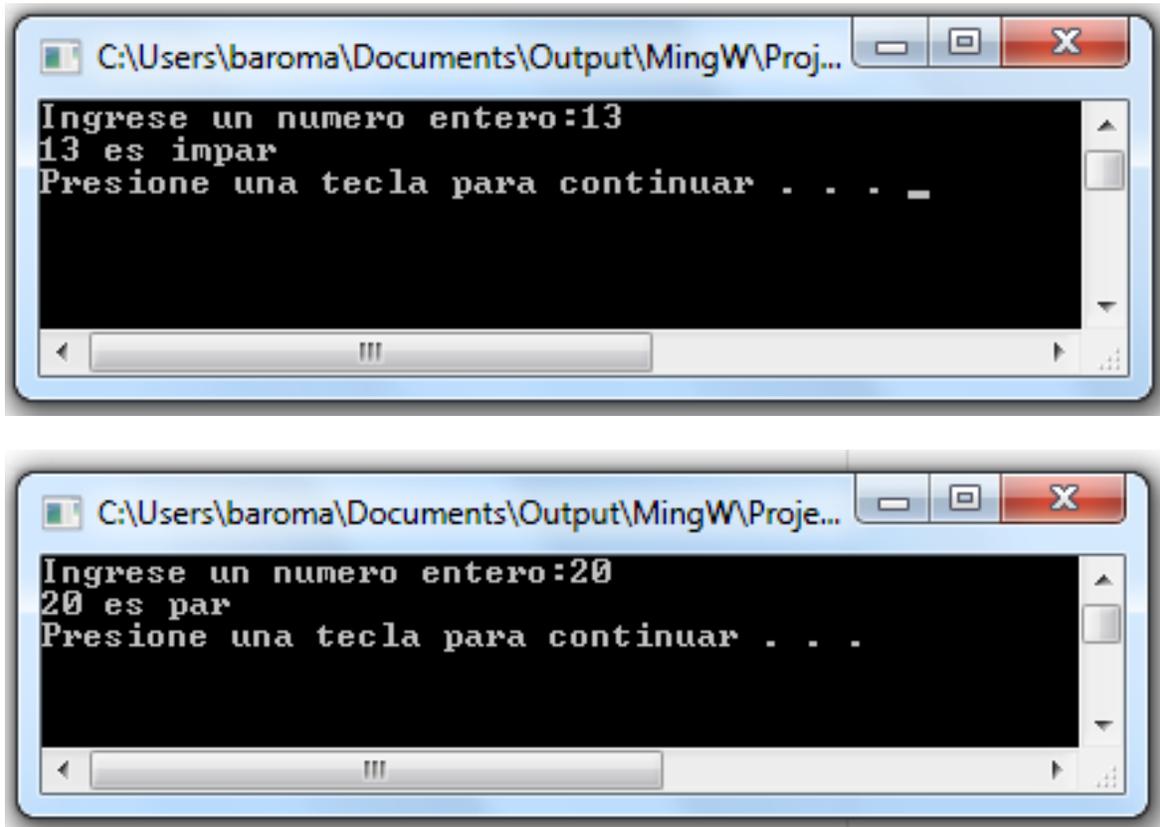


Figura 9: Programa ejecutado en la computadora de la estructura de selección de datos "SI-SINO"

2.2.16.3 DEFINICIÓN DE LAS ESTRUCTURA DE SELECCIÓN DE DATOS SI ANIDADO

La estructura de selección de datos SI ANIDADO permite elegir una alternativa entre más de dos opciones.

La estructura de selección de datos SI ANIDADO necesita un dato de entrada para que sea evaluado por una primera condición lógica.

Recordemos el funcionamiento de una estructura de selección de datos completa: Si el resultado de esta evaluación lógica es verdadero, el dato se trasladará por el camino indicado por el resultado lógico **Verdadero**; en caso contrario si el resultado de la evaluación lógica es falso, el dato se trasladará por el camino indicado por el resultado lógico **Falso**.

Se dan tres casos:

CASO I:

Se considera que es una estructura de selección de datos SI ANIDADO cuando un dato es evaluado por una primera condición lógica y si el resultado de la condición lógica es **verdadero**, el dato se evalúa por una segunda condición lógica, tal como se muestra en la **Figura 10**.

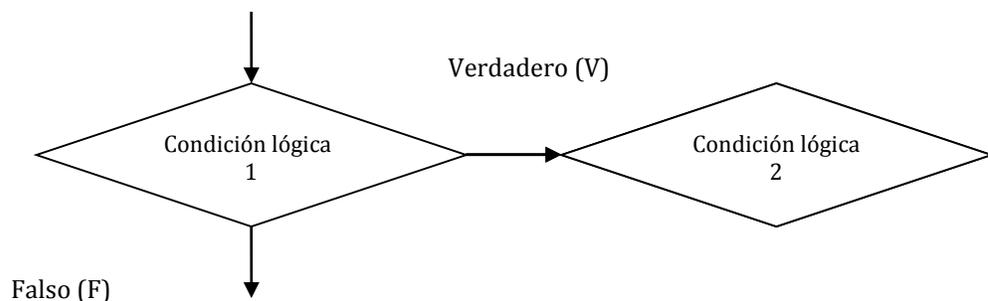


Figura 10: Esquema de comprensión del CASO I de la estructura de selección múltiple “SI Anidado”

CASO II:

Se considera que es una estructura de selección de datos SI ANIDADO, cuando un dato es evaluado por una primera condición lógica y si el resultado de la condición lógica es **falso**, el dato es evaluado por una tercera condición lógica, tal como se muestra en la **Figura 11**.

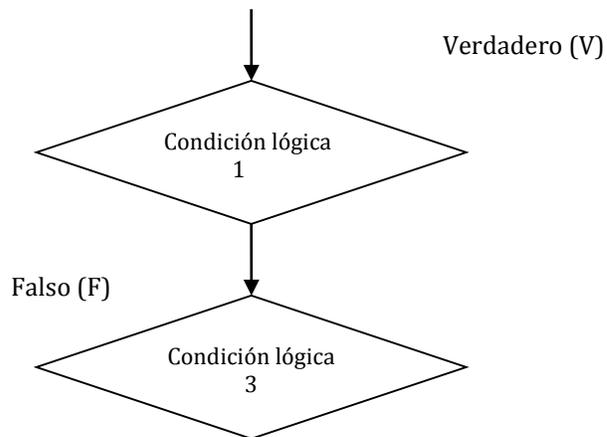


Figura 11: Esquema de comprensión del CASO II de la estructura de selección múltiple “SI Anidado”

CASO III:

Se considera que es una estructura de selección de datos SI ANIDADO cuando un dato es evaluado por una condición lógica inicial.

Si el resultado de la condición lógica es **verdadero**, el dato es evaluado por una segunda condición lógica y si el resultado de la condición lógica inicial es **falso**, el dato se evalúa por una tercera condición lógica, tal como se muestra en la **Figura 12**.

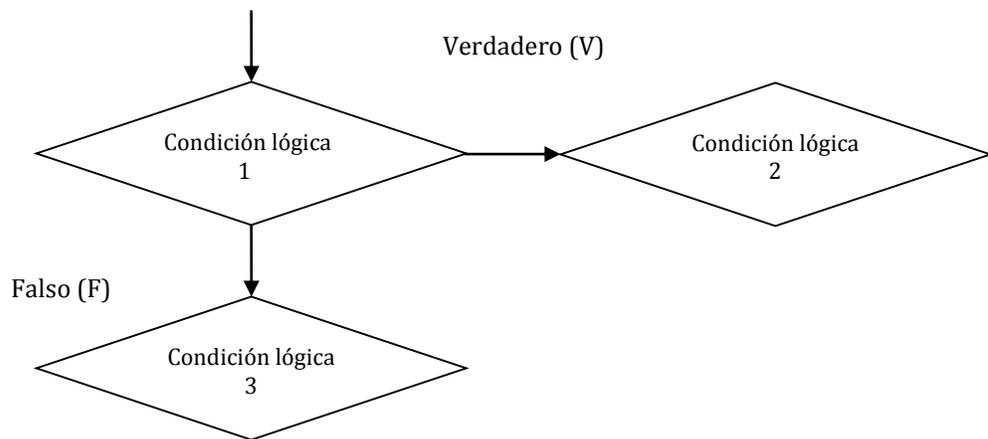


Figura 12: Esquema de comprensión del CASO III de la estructura de selección múltiple “SI Anidado”

EJEMPLO

Hacer un programa en código c que permita a un gerente de una empresa comercial otorgar un porcentaje de incremento en su sueldo por haber cumplido una meta de ventas de acuerdo con la siguiente tabla:

Cantidad de computadoras vendidas	Ingreso adicional al sueldo
Entre 20 y menos que 40	25% del sueldo
40 o más	30% del sueldo

Nota: Se debe ingresar por el teclado el sueldo del trabajador y la cantidad de computadoras vendidas para que el programa en código c muestre el incremento y el sueldo que recibirá el vendedor.

Figura 13: Enunciado a programar con la estructura selección múltiple "SI Anidado"

CONSTRUCCIÓN DEL PROGRAMA UTILIZANDO EL MÉTODO BALTEX

PASO 1 Bosquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos. (Actividad creativa)

- El objetivo es construir un programa en código c, que permita capturar por el teclado el sueldo de un trabajador. No se hace referencia al tipo de datos, por lo tanto, en este caso, se deben de utilizar dos variables reales, una para almacenar el sueldo del empleado y otra para el incremento en el sueldo si se cumplen las metas trazadas por el departamento comercial.
- Para saber si un empleado tiene incremento en su sueldo se debe saber la cantidad de computadoras vendidas, por lo tanto, para almacenar este valor se utilizará una variable entera.
- Luego las variables a utilizarse son las siguientes:

REAL sueld, increm

ENTERO cant

Nota: Las tildes son omitidas en el contenido de las variables de texto, ya que no las reconoce el wxDev-C++, software que se utilizará para redactar el programa en lenguaje C.

TÉCNICA ANALÍTICA DE LA ESTRUCTURA DE SELECCIÓN SI ANIDADO

Para analizar los programas utilizando la estructura de selección de datos completa anidada, se utilizarán la recta numérica y los intervalos abiertos y cerrados según las condiciones del ejercicio.

En este caso:

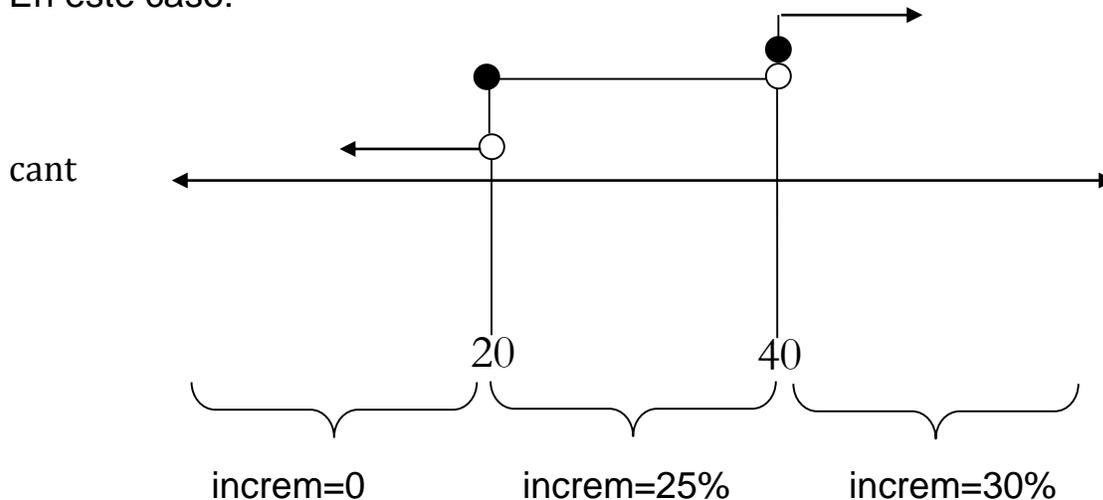


Figura 14: Técnica analítica con la estructura selección múltiple “SI Anidado”

Según la recta numérica trazada, hay tres rangos definidos por los intervalos abiertos y cerrados.

El primer rango ($\text{cant} < 20$) contiene los valores inferiores al intervalo abierto en el número 20. Las cantidades que estén en este rango no tendrán incremento.

El segundo rango ($\text{cant} \geq 20$ y $\text{cant} < 40$) contiene los valores comprendidos entre el intervalo cerrado en el número 20 y el intervalo abierto en el número 40. Las cantidades que estén en este rango tendrán un incremento del 25%.

El tercer rango ($\text{cant} \geq 40$) contiene los valores, en el intervalo cerrado, en el número 40 o superiores. Las cantidades que estén en este rango tendrán un incremento del 30%.

Por lo tanto:

La primera condición lógica será **$\text{cant} < 20$** .

La segunda condición lógica será **$\text{cant} < 40$** .

Adicionalmente, se debe ingresar por el teclado el sueldo del empleado.

PASO 2 Analizar y definir la arquitectura del diagrama de flujo. (Actividad creativa)

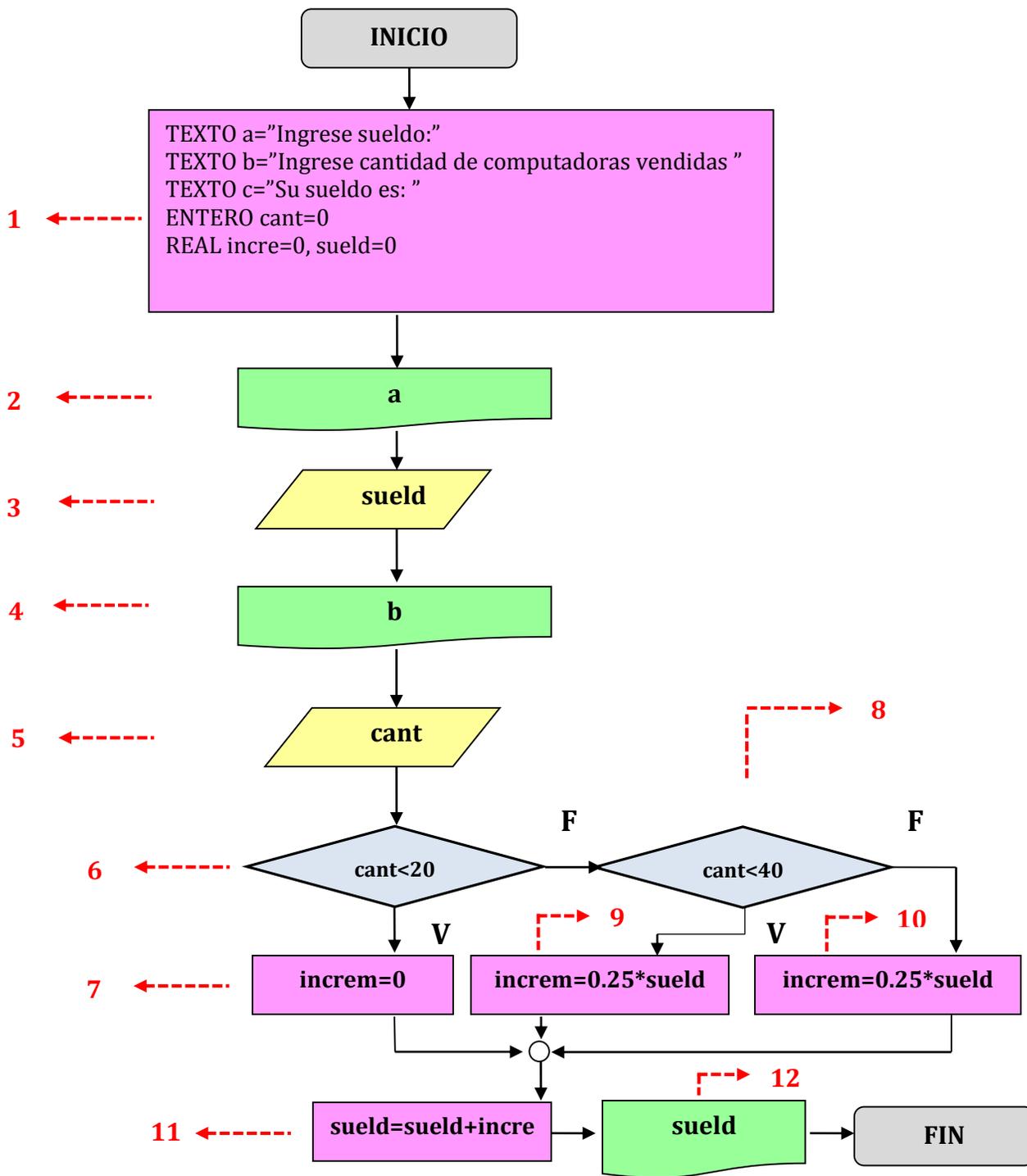


Figura 15: Diagrama de flujo de la estructura selección múltiple "Si Anidado"

PASO 3 Leer el contenido de cada símbolo del diagrama de flujo y traducir a pseudocódigos.
(Actividad mecánica)

REFERENCIAS DEL DIAGRAMA DE FLUJO	ALGORITMO : PrgIfA1
	ENTRADA: a,b
	SALIDA: sueld
	INICIO
	//Declaración de variables
1	TEXTO a="Ingrese sueldo:"
1	TEXTOb="Ingrese cantidad de computadoras vendidas: "
1	TEXTO c="Su sueldo es:"
1	ENTERO cant=0
1	REAL sueld=0, incre=0
	//Entrada
2	ESCRIBIR (a)
3	LEER sueld
4	ESCRIBIR (b)
5	LEER cant
	//Condición lógica
6	SI (cant<20) ENTONCES
	//Proceso si condición lógica es V
7	incre=0
	SINO
	//Proceso si condición lógica es F
8	SI (cant<40) ENTONCES
	//Proceso si condición lógica es V
9	incre=0.25*sueld
	SINO
	//Proceso si condición lógica es F
10	incre=0.30*sueld
	FIN SI
	FIN SI
11	sueld=sueld+incre
12	ESCRIBIR sueld
	FIN

Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de selección de datos SI anidado

Tabla 12

PASO 4 **Ex**portar el pseudocódigo a un lenguaje de programación. (Actividad mecánica)

ALGORITMO : PrgIfA1	Programa en código C
ENTRADA: a,b	#include <cstdlib>
SALIDA: sueld	#include <iostream>
	using namespace std;
INICIO	int main(int argc, char *argv[])
//Declaración de variables	//Declaración de variables
TEXTO a="Ingrese sueldo: "	string a="Ingrese sueldo: ";
TEXTO b="Ingrese cantidad de computadoras vendidas: "	string b="Ingrese cantidad de computadoras vendidas: ";
TEXTO c="Su sueldo es:"	string c="Su sueldo es:";
ENTERO cant=0	int cant=0;
REAL sueld=0, incre=0	double sueld=0, incre=0;
//Entrada	//Entrada
ESCRIBIR (a)	cout<<a;
LEER sueld	cin>>sueld;
ESCRIBIR (b)	cout<<b;
LEER cant	cin>> cant;
//Condición lógica	//Condición lógica
SI (cant<20) ENTONCES	if (cant<20)
//Proceso si condición lógica es V	//Proceso si condición lógica es V
incre=0	{ incre=0;}
SINO	else {
//Proceso si condición lógica es F	//Proceso si condición lógica es F
SI (cant<40) ENTONCES	if (cant<40)
//Proceso si condición lógica es V	//Proceso si condición lógica es V
incre=0.25*sueld	{ incre=0.25*sueld;}
SINO	else
//Proceso si condición lógica es F	//Proceso si condición lógica es F
incre=0.30*sueld	{ incre=0.30*sueld;}
FIN SI	}
FIN SI	
sueld=sueld+incre	sueld=sueld+incre;
ESCRIBIR sueld	cout<<c<<" "<<sueld;
ESCRIBIR SALTOLINEA ()	cout<<endl;
FIN	system("PAUSE"); return EXIT_SUCCESS;}

Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de selección de datos SI anidado

Tabla 13

PROGRAMA EN CÓDIGO C REDACTADO EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

```
[*] main.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    //declaración de variables
    string a="Ingrese sueldo: ";
    string b="Ingrese cantidad de computadoras vendidas: ";
    string c="Su sueldo es:";
    int cant=0;
    double sueld=0, incre=0;
    //entrada
    cout<<a;
    cin>>sueld;
    cout<<b;
    cin>> cant;
    //condición lógica
    if (cant<20)
    //proceso si condición lógica es V
    {   incre=0;}
    else {
    //proceso si condición lógica es F
    if (cant<40)
    //proceso si condición lógica es V
    {   incre=0.25*sueld;}
    else
    //proceso si condición lógica es F
    {   incre=0.30*sueld;}
    }
    sueld=sueld+incre;
    cout<<c<<" "<<sueld;
    cout<<endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Figura 16: Programa en código C de la estructura selección múltiple “SI Anidado”

PROGRAMA DESPUÉS DE EJECUTARSE EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

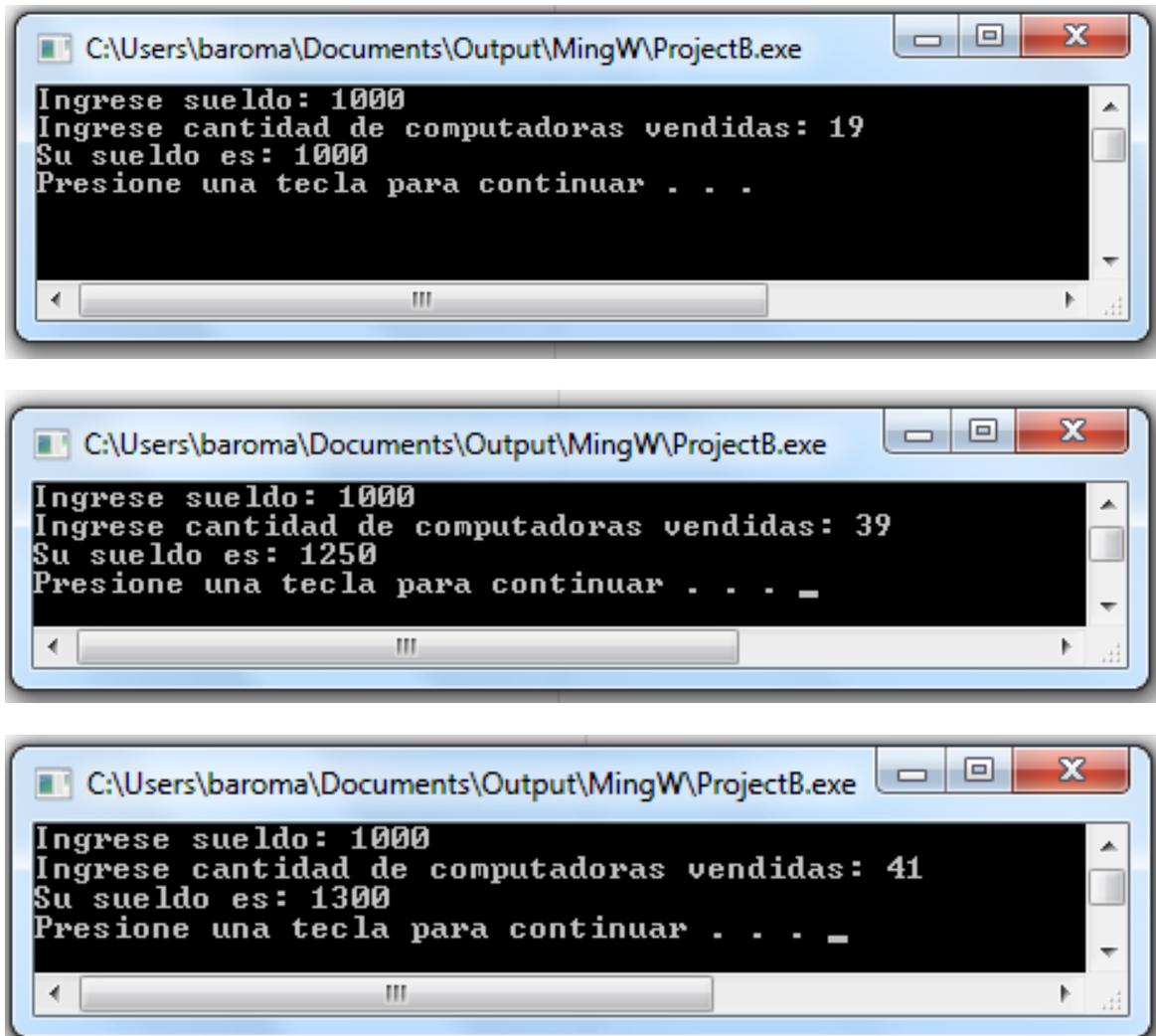


Figura 17: Programa ejecutado en la computadora de la estructura selección múltiple "SI Anidado"

2.2.16.4 DEFINICIÓN DE LA ESTRUCTURA DE SELECCIÓN DE DATOS MÚLTIPLE “CUANDO SEA”

La estructura de selección de datos múltiple “CUANDO SEA” permite seleccionar una alternativa entre más de dos opciones, dependiendo del dato ingresado en una variable. Si el valor del dato coincide con un valor especificado por la condición lógica, en un caso determinado, se ejecutarán solamente las acciones definidas en ese caso. En caso contrario, si el dato no coincide con uno de los valores especificados por condiciones lógicas en otros casos alternativos, se realizará la acción denominada “Otros”, tal como se muestra en la **Figura 18**

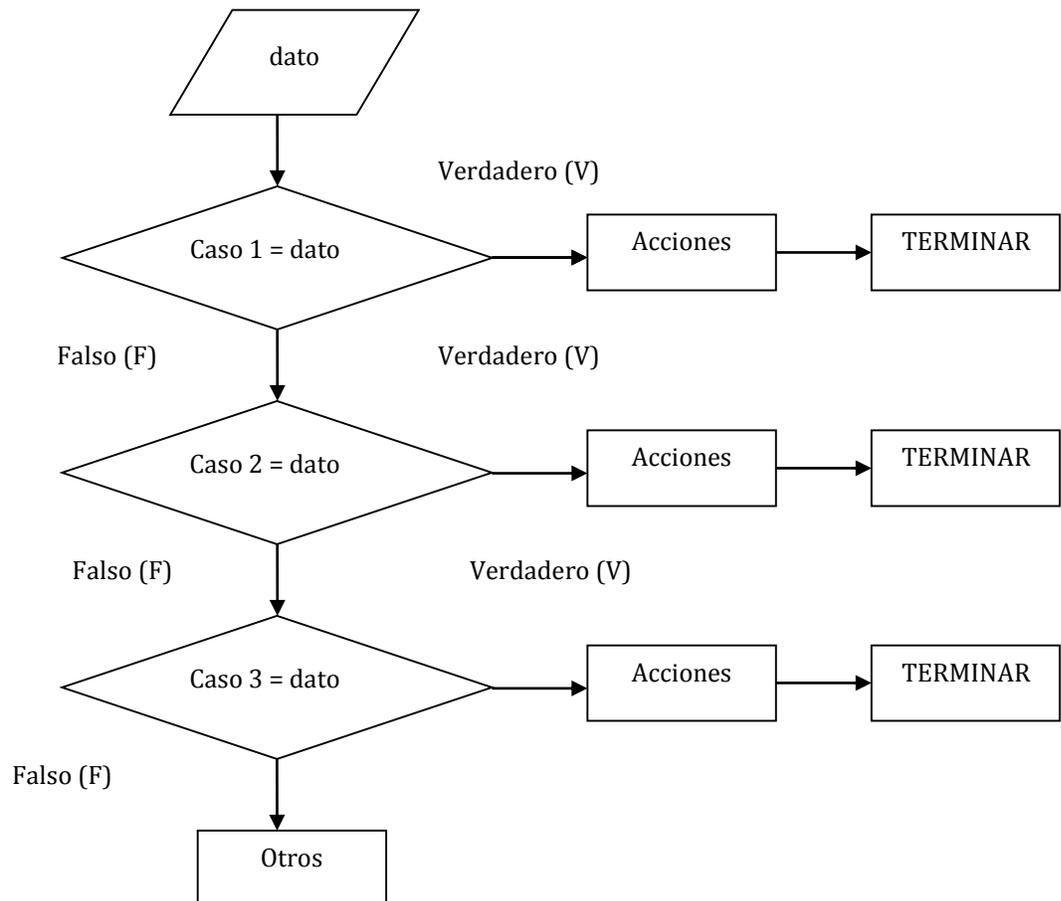


Figura 18: Esquema de comprensión de la estructura de selección múltiple “Cuando sea”

EJEMPLO

Hacer un programa en código c para que el Departamento de Recursos Humanos de una empresa comercial pueda pagar el sueldo correspondiente a sus empleados incluyendo los incrementos de sueldo según disposiciones del Departamento de Ventas de acuerdo con la siguiente tabla

Categoría	Incremento	Tipo de empleado
1	38%	vendedor
2	21%	supervisor
3	12%	jefe
4	8%	gerente

Figura 19: Enunciado a programar con la estructura de selección múltiple "Cuando sea"

CONSTRUCCIÓN DEL PROGRAMA UTILIZANDO EL MÉTODO BALTEX

PASO 1 Bosquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos. (Actividad creativa)

- El objetivo es construir un programa en código c, que permita capturar por el teclado el sueldo y la categoría de un trabajador. No se hace referencia al tipo de datos, por lo tanto, en este caso, se deben utilizar dos variables, una de tipo real para almacenar el valor del sueldo y una variable de tipo entero para almacenar la categoría del empleado.
- En este programa, se supone que después de ingresar el sueldo del empleado por el teclado, se procede a proporcionar el valor de la categoría y en forma automática, el programa calcula el nuevo sueldo del empleado con el incremento incluido especificado en la tabla.
- Luego las variables a utilizar son las siguientes:

REAL sueld, increm

ENTERO categ

TABLA ANALÍTICA DE LA ESTRUCTURA DE SELECCIÓN DE DATOS MÚLTIPLE “CUANDO SEA”.

Para analizar los programas utilizando la estructura de selección de datos múltiple “CUANDO SEA”, se utilizará la Tabla 14.

Para organizar el pensamiento analítico, se va a suponer que el sueldo del empleado ingresado por el teclado corresponde a 100.

Nota: Las tildes son omitidas en el contenido de las variables de texto, ya que no las reconoce el wxDev-C++, software que, se utilizará para redactar el programa en lenguaje C.

Categoría	Sueldo	Porcentaje de incremento	incremento	sueldo= sueldo+incremento
1	100	38% del sueldo	$(38*100)/100$	100+38
2	100	21% del sueldo	$(21*100)/100$	100+21
3	100	12% del sueldo	$(12*100)/100$	100+12
4	100	8% del sueldo	$(8*100)/100$	100+8

Tabla 14

PASO 2 Analizar y definir la arquitectura del diagrama de flujo. (Actividad creativa)

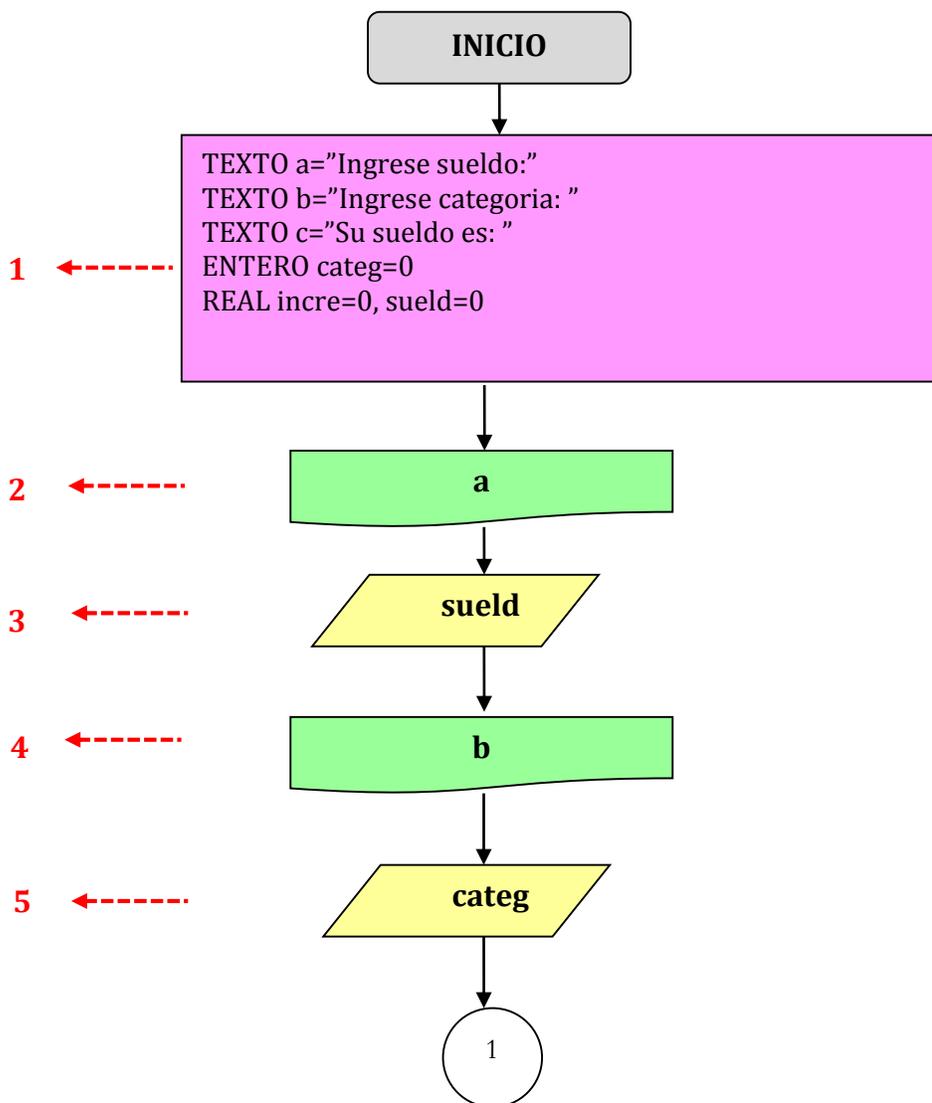


Figura 20: Diagrama de flujo de la estructura selección múltiple “Cuando Sea”. Parte 1

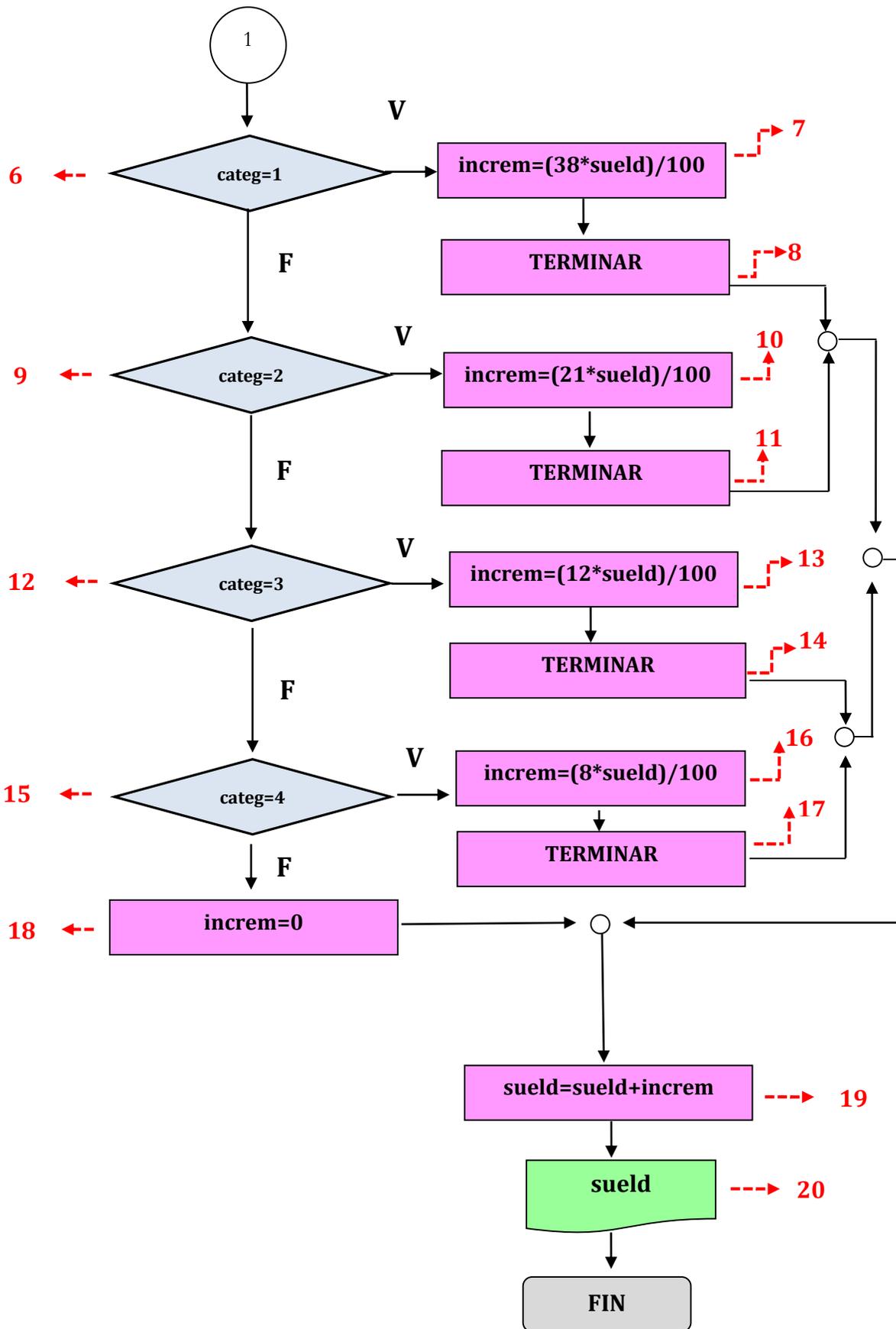


Figura 21: Diagrama de flujo de la estructura selección múltiple “Cuando Sea”. Parte 2

PASO 3

Leer el contenido de cada símbolo del diagrama de flujo y traducir a pseudocódigos.
(Actividad mecánica)

REFERENCIAS DEL DIAGRAMA DE FLUJO	ALGORITMO : PrgSw1
	ENTRADA: sueld, categ
	SALIDA: sueld
	INICIO
	//Declaración de variables
1	TEXTO a="Ingrese sueldo:"
1	TEXTO b="Ingrese categoría: "
1	TEXTO c="Su sueldo es: "
1	ENTERO categ=0
1	REAL incre=0, sueld=0
	//Entrada
2	ESCRIBIR a
3	LEER sueld
4	ESCRIBIR b
5	LEER categ
	//Inicio "CUANDO SEA"
	CUANDO(categ) SEA
6	CASO (categ=1)
7	$incre=(38*sueld)/100$
8	TERMINAR
9	CASO (categ=2)
10	$incre=(21*sueld)/100$
11	TERMINAR
12	CASO (categ=3)
13	$incre=(12*sueld)/100$
14	TERMINAR
15	CASO (categ=4)
16	$incre=(8*sueld)/100$
17	TERMINAR
	OTROS
18	$incre=0$
	FIN CUANDO
	//Fin "CUANDO SEA"
19	sueld=sueld+incre
20	ESCRIBIR sueld , FIN

Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de selección de datos múltiple "Cuando Sea".

Tabla 15

PASO 4 Exportar el pseudocódigo a un lenguaje de programación. (Actividad mecánica)

ALGORITMO : PrgSw1	Programa en código c
ENTRADA: sueld, categ	#include <cstdlib>
SALIDA: sueld	#include <iostream>
INICIO	using namespace std;
	int main(int argc, char *argv[])
//Declaración de variables	//Declaración de variables
TEXTO a="Ingrese sueldo:"	string a="Ingrese sueldo:";
TEXTO b="Ingrese categoria: "	string b="Ingrese categoria: ";
TEXTO c="Su sueldo es: "	string c="Su sueldo es: ";
ENTERO categ=0	int categ=0;
REAL increm=0, sueld=0	double increm=0, sueld=0;
//Entrada	//Entrada
ESCRIBIR a	cout<< a;
LEER sueld	cin>> sueld;
ESCRIBIR b	cout<< b;
LEER categ	cin>>categ;
//Inicio "CUANDO SEA"	//Inicio "CUANDO SEA"
CUANDO(categ) SEA	switch(categ) {
CASO (categ=1)	case 1: {
increm=(38*sueld)/100	increm=(38*sueld)/100;
TERMINAR	break;}
CASO (categ=2)	case 2: {
increm=(21*sueld)/100	increm=(21*sueld)/100;
TERMINAR	break;}
CASO (categ=3)	case 3: {
increm=(12*sueld)/100	increm=(12*sueld)/100;
TERMINAR	break;}
CASO (categ=4)	case 4: {
increm=(8*sueld)/100	increm=(8*sueld)/100;
TERMINAR	break;}
OTROS	default: {
increm=0	increm=0;}
FIN CUANDO //Fin "CUANDO SEA"	} //Fin "CUANDO SEA"
sueld=sueld+increm	sueld=sueld+increm;
ESCRIBIR c, sueld, SALTOLINEA ()	cout<< c<<sueld<<endl;
FIN	return EXIT_SUCCESS; }

Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de selección de datos múltiple "Cuando Sea"

Tabla 16

PROGRAMA EN CÓDIGO C REDACTADO EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

```
main.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    //declaracion de variables
    string a="Ingrese sueldo:";
    string b="Ingrese categoria: ";
    string c="Su sueldo es: ";
    int categ=0;
    double increm=0, sueld=0;
    //entrada
    cout<< a;
    cin>> sueld;
    cout<< b;
    cin>>categ;
    //Inicio de estructura de selección múltiple "CUANDO SEA"
    switch(categ) {
    case 1: {
        increm=(38*sueld)/100;
        break;}
    case 2: {
        increm=(21*sueld)/100;
        break;}
    case 3: {
        increm=(12*sueld)/100;
        break;}
    case 4: {
        increm=(8*sueld)/100;
        break;}
    default: {
        increm=0;}
    }
    //Fin de estructura de selección múltiple "CUANDO SEA"
    sueld=sueld+increm;
    cout<< c<<sueld<<endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Figura 22: Programa en código C de la estructura selección múltiple “Cuando Sea”

PROGRAMA DESPUÉS DE EJECUTARSE EN EL ENTORNO DE
DESARROLLO LIBRE wxDev-C++

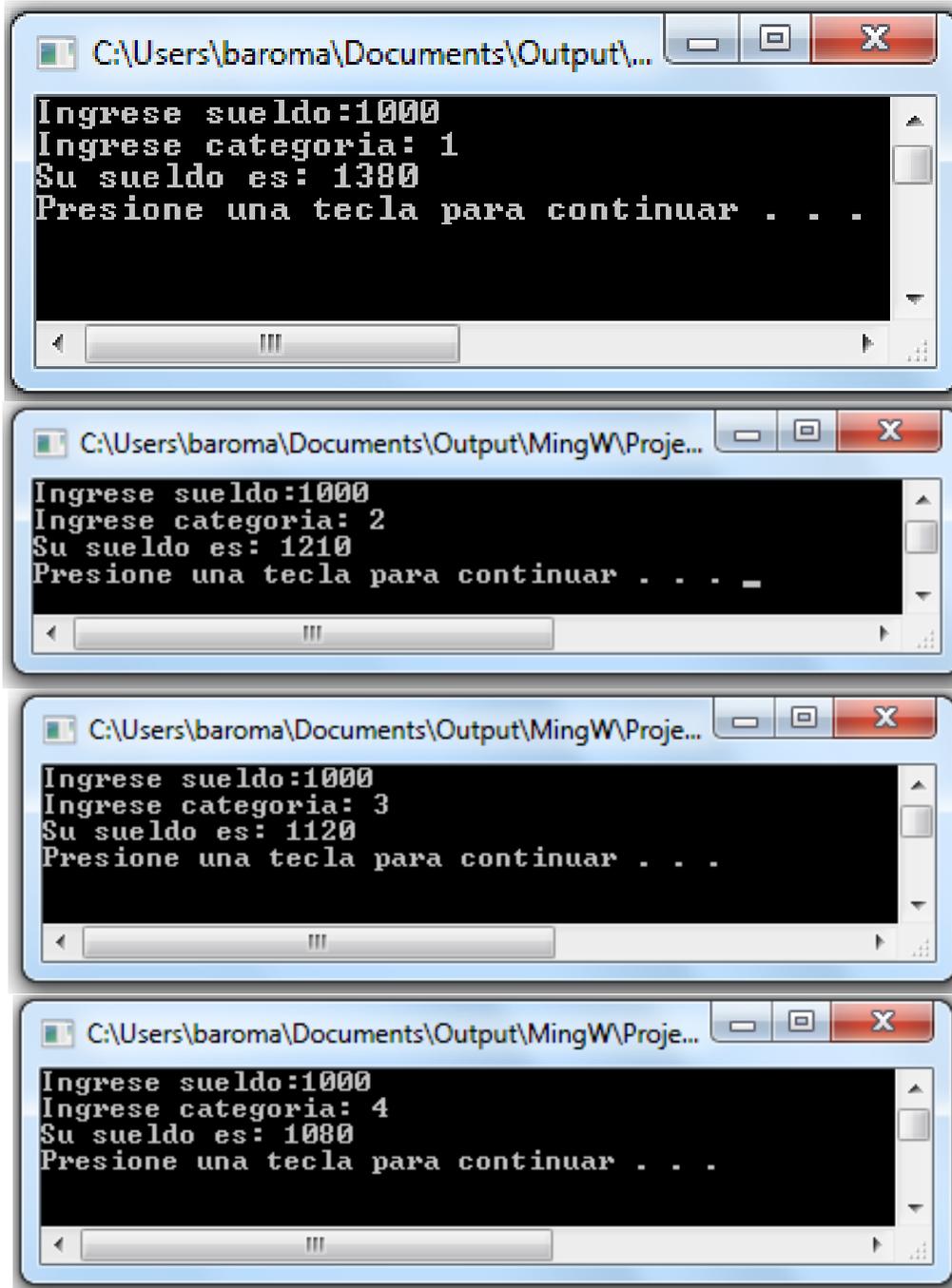


Figura 23: Programa ejecutado en la computadora de la estructura selección múltiple “Cuando Sea”

2.2.16.5 DEFINICIÓN DE LA ESTRUCTURA DE REPETICIÓN DE DATOS “MIENTRAS”

La estructura de repetición de datos “Mientras” necesita el valor de un dato para que sea evaluado por una condición lógica. Si el resultado de esta evaluación lógica es verdadero, se ejecutarán diversas instrucciones en forma repetitiva. Las instrucciones dejarán de repetirse cuando el resultado de la evaluación lógica sea falsa. **(Ver figura 24)**

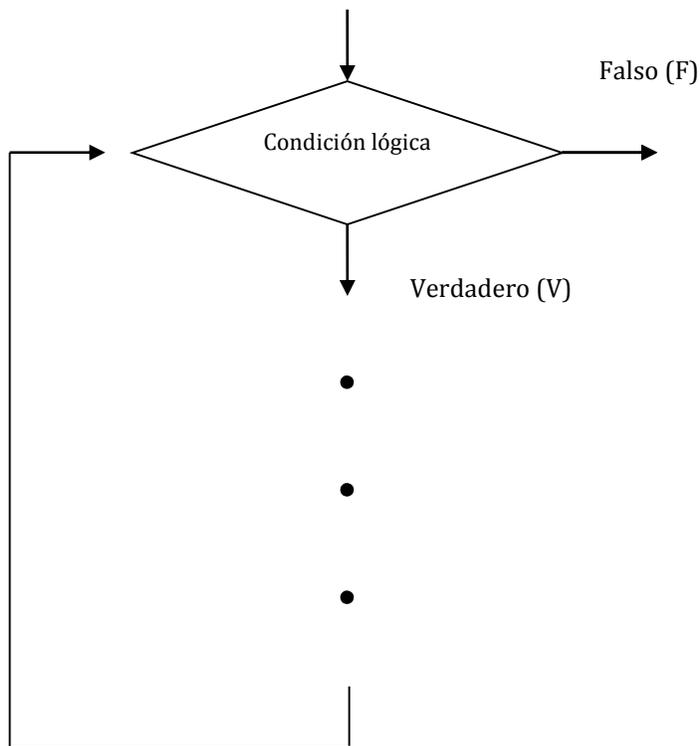


Figura 24: Esquema de comprensión de la estructura de repetición de datos “Mientras”

La estructura de repetición “mientras” se basa en dos postulados.

Postulado 01: “Todo dato generado y visualizado en la pantalla de la computadora se almacenará, temporalmente, en una sola variable. El tipo de dato depende de la naturaleza del dato a mostrar “

Postulado 02: “Cuando el resultado lógico de la estructura de repetición “Mientras” es verdadero, se ejecutan un grupo de instrucciones. La última instrucción por ejecutar es una fórmula de actualización del valor de la variable involucrada en la condición lógica y este nuevo valor se evalúa nuevamente con la condición lógica.”

Hay dos variables especiales que se utilizan en la estructura de repetición “Mientras”; la variable contador y la variable acumulador.

La variable contador sirve para contar las veces que se muestra un dato, que cumple con algunas características especiales definidas en una condición lógica.

Ejemplo: Si se declara la variable entera cont, esta variable debe tener un valor inicial igual a cero.

La fórmula general para implementar un proceso de conteo es: **cont=cont+1**.

Se considera un proceso fijo porque aumenta de uno en uno.

La variable acumulador sirve para almacenar el valor de un dato a través de una operación aritmética de suma o de multiplicación para saber el valor total del contenido de una serie de datos generados.

Ejemplo: Si se declara la variable entera x para almacenar una serie de datos enteros y se desea calcular la suma y el producto de estos valores, se necesitan declarar, adicionalmente, las variables enteras sum y prod.

La variable sum debe tener un valor inicial igual a cero.

La variable prod debe tener un valor inicial igual a uno.

La fórmula general para implementar un proceso de suma de datos es:
sum = sum+x.

La fórmula general para implementar un proceso de multiplicación de datos es:
prod=prod*x.

EJEMPLO

Hacer un programa en código c que genere la serie **2 4 6 8** y mostrarla en la pantalla de la computadora.

Figura 25: Enunciado a programar con la estructura de repetición de datos "Mientras"

CONSTRUCCIÓN DEL PROGRAMA UTILIZANDO EL MÉTODO BALTEX

PASO 1 Bosquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos. (Actividad creativa)

- El objetivo es construir un programa en código c, que permita generar y mostrar la serie 2 4 6 8.
- Según el postulado 1 de la estructura de repetición “Mientras”, los valores de cada término deben almacenarse en una sola variable.
- Como se puede apreciar, los datos de la serie son de tipo entero, por lo tanto, se necesita declarar una variable entera.
- Luego la variable a utilizarse es la siguiente: ENTERO x
- El primer término de la serie es 2, por lo tanto, el primer valor de la variable x debe ser 2.
- El último valor de la serie es 8, por lo tanto, la condición lógica propuesta que cumplen todos los valores de la serie puede ser $x \leq 8$ o $x < 10$ que es el siguiente término posible de la serie. Se puede utilizar, indistintamente, cualquier condición lógica propuesta.
- El siguiente término de la serie se incrementa en dos unidades. La fórmula de actualización del valor de la variable que genera la serie es: $x = x + 2$.

TÉCNICA ANALÍTICA ESTRUCTURA DE REPETICIÓN DE DATOS “MIENTRAS”

Para analizar los programas usando la estructura de repetición de datos se utilizará la siguiente tabla de seguimiento y control de valores de la variable que genera los datos. La condición lógica que se empleará será $x \leq 8$.

TABLA DE SEGUIMIENTO Y CONTROL DE VALORES DE LA ESTRUCTURA DE REPETICIÓN DE DATOS “MIENTRAS”

Valor actual de x	Condición Lógica ($x \leq 8$)	Resultado de la Condición Lógica	Cuerpo de la estructura de repetición “Mientras”	
			ESCRIBIR (x)	$x = x + 2$
2	$2 \leq 8$	Verdadero	2	$x = 2 + 2$
4	$4 \leq 8$	Verdadero	4	$x = 4 + 2$
6	$6 \leq 8$	Verdadero	6	$x = 6 + 2$
8	$8 \leq 8$	Verdadero	8	$x = 8 + 2$
10	$10 < 8$	Falso		

Tabla 17

La cantidad de columnas de la tabla depende del análisis del programador.

IMPORTANTE: En una columna de la tabla, se visualiza la serie a generar.

Cada línea de la tabla cuando la condición es verdadera se llama iteración.

Cuando la condición es falsa, se detiene el proceso de repetición de instrucciones.

Nota: Las tildes son omitidas en el contenido de las variables de texto, ya que no las reconoce el wxDev-C++, software que se utilizará para redactar el programa en lenguaje C.

PASO 2 Analizar y definir la arquitectura del diagrama de flujo. (Actividad creativa)

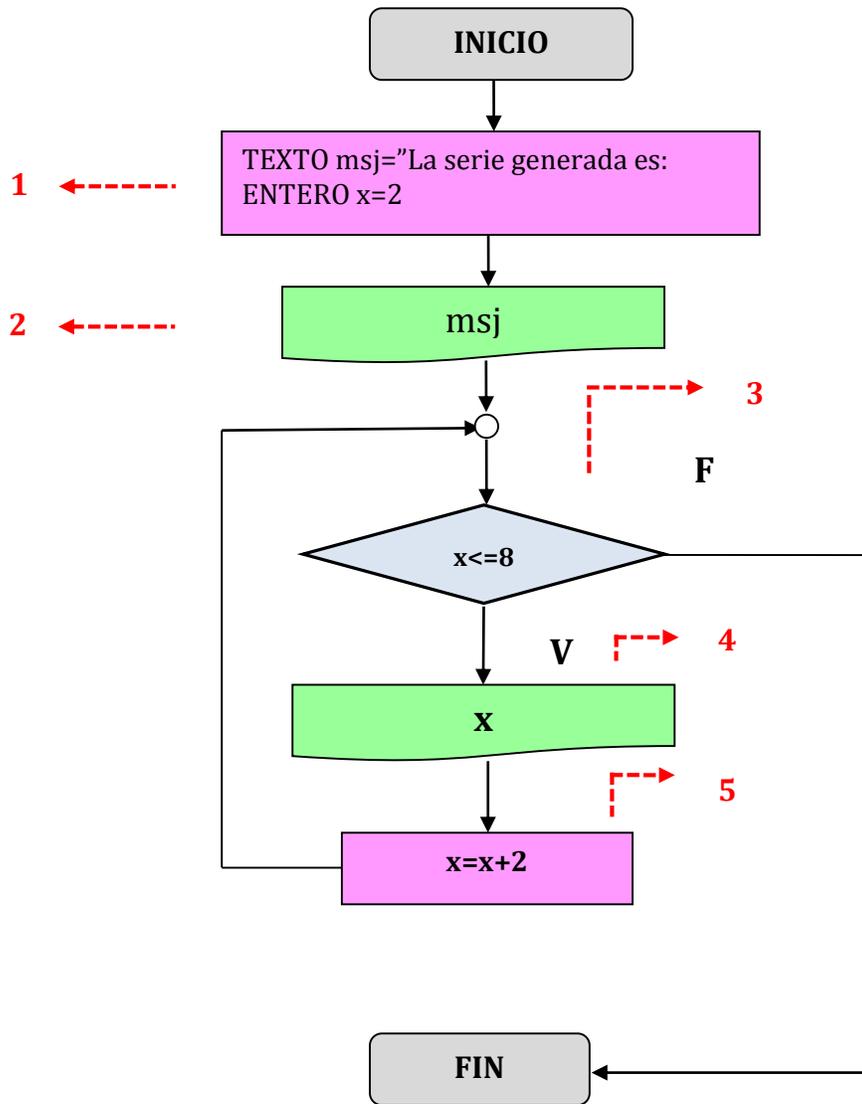


Figura 26: Diagrama de flujo de la estructura de repetición de datos “Mientras”

PASO 3 Leer el contenido de cada símbolo del diagrama de flujo y traducir a pseudocódigos. (Actividad mecánica)

REFERENCIAS DEL DIAGRAMA DE FLUJO	ALGORITMO : PrgSerie1
	ENTRADA: SALIDA: x
	INICIO
	//Declaración de variables
1	TEXTO msj="La serie generada es:"
1	ENTERO x=2
	//Visualizar mensaje
2	ESCRIBIR (msj)
	//Condición lógica
3	MIENTRAS (x<=8) HACER
	//Proceso si condición lógica es V
	//Inicio del cuerpo de la estructura de repetición "Mientras"
4	ESCRIBIR x
5	x=x+2
	//Fin del cuerpo de la estructura de repetición "Mientras"
	FIN MIENTRAS

Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de repetición de datos "Mientras"

Tabla 18

PASO 4 Exportar el pseudocódigo a un lenguaje de programación.
(Actividad mecánica)

ALGORITMO : PrgSerie1	Programa en código c
ENTRADA:	#include <cstdlib>
SALIDA: x	#include <iostream>
	using namespace std;
	int main(int argc, char *argv[])
INICIO	
//Declaración de variables	//Declaración de variables
TEXTO msj="La serie generada es:"	string msj="La serie generada es:";
ENTERO x=2	int x=2;
//Visualizar mensaje	//Visualizar mensaje
ESCRIBIR (msj), SALTOLINEA()	cout<<msj<<endl;
//Condición lógica	//Condición lógica
MIENTRAS (x<=8) HACER	while (x<=8) {
//Proceso si condición lógica es V	//Proceso si condición lógica es V
//Inicio del cuerpo de la estructura de repetición "Mientras"	//Inicio del cuerpo de la estructura de repetición "Mientras"
ESCRIBIR (x)	cout<< x<<" ";
x=x+2	x=x+2;
//Fin del cuerpo de la estructura de repetición "Mientras"	//Fin del cuerpo de la estructura de repetición "Mientras"
FIN MIENTRAS	}
ESCRIBIR SALTOLINEA()	cout<<endl;
FIN	system("PAUSE");
	return EXIT_SUCCESS;
	}

Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de repetición de datos "Mientras"

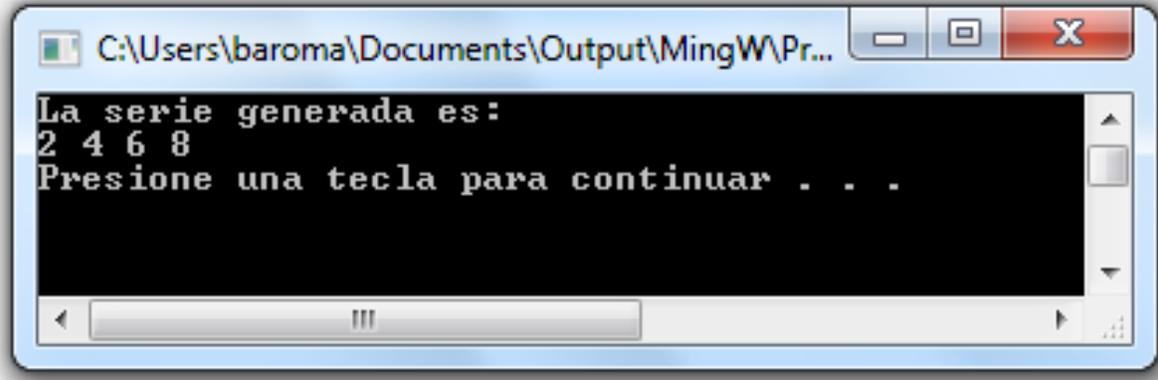
Tabla 19

PROGRAMA EN CÓDIGO C REDACTADO EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

```
main.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    //declaración de variables
    string msj="La serie generada es:";
    int x=2;
    //visualizar mensaje
    cout<<msj<<endl;
    //condición lógica
    while (x<=8) {
        //proceso si condición lógica es V
        //inicio del cuerpo de la estructura de
        //repetición "Mientras"
        cout<< x<<" ";
        x=x+2;
        //fin del cuerpo de la estructura de
        //repetición "Mientras"
    }
    cout<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Figura 27: Programa en código C de la estructura de repetición de datos "Mientras"

**PROGRAMA DESPUÉS DE EJECUTARSE EN EL ENTORNO DE
DESARROLLO LIBRE wxDev-C++**



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\baroma\Documents\Output\MingW\Pr... The window contains the following text:
La serie generada es:
2 4 6 8
Presione una tecla para continuar . . .

Figura 28: Programa ejecutado en la computadora de repetición de datos "Mientras"

2.2.16.6 DEFINICIÓN DE LA ESTRUCTURA DE REPETICIÓN DE DATOS “PARA”

La estructura de repetición de datos “Para” necesita el valor de un dato para que sea evaluado por una condición lógica. Si el resultado de esta evaluación lógica es verdadero, se ejecutarán diversas instrucciones en forma repetitiva. Las instrucciones dejarán de repetirse cuando el resultado de la evaluación lógica sea falso. **(Ver Figura 29)**

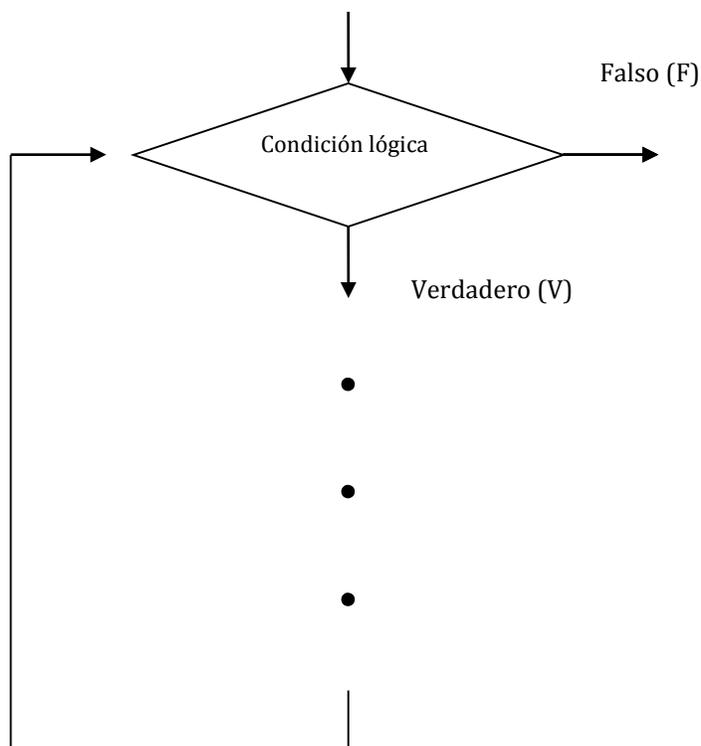


Figura 29: Esquema de comprensión de la estructura de repetición de datos “Para”

La estructura de repetición “Para” se basa en dos postulados.

Postulado 01: “Todo dato generado y visualizado, en la pantalla de la computadora, se almacenará temporalmente en una sola variable. El tipo de dato depende de la naturaleza del dato a mostrar “.

Postulado 02: “Cuando el resultado lógico de la estructura de repetición “Para” es verdadero se ejecutan un grupo de instrucciones. La última instrucción a ejecutar es una fórmula de actualización del valor de la variable involucrada en la condición lógica y este nuevo valor se somete a la condición lógica”.

Hay dos variables especiales que se utilizan en la estructura de repetición “Para”, la variable contador y la variable acumulador.

La variable contador sirve para contar las veces que se muestra un dato que cumple con algunas características especiales, definidas en una condición lógica.

Ejemplo: Si se declara la variable entera **cont**, esta variable debe tener un valor inicial igual a cero. La fórmula general para implementar un proceso de conteo es: **cont=cont+1**. Se considera un proceso fijo porque aumenta de uno en uno.

La variable acumulador sirve para almacenar el valor de un dato a través de una operación aritmética de suma o de multiplicación para conocer el valor total del contenido de una serie de datos generados.

Ejemplo: Si se declara la variable entera x para almacenar una serie de datos enteros y se desea calcular la suma y el producto de estos valores, se requiere declarar, adicionalmente, las variables enteras sum y prod.

El valor inicial de la variable sum debe ser cero.

El valor inicial de la variable prod debe ser uno.

La fórmula general para implementar un proceso de suma de datos es: **sum = sum+x**.

La fórmula general para implementar un proceso de multiplicación de datos es: **prod=prod*x**.

Las estructura de repetición “Mientras” y “Para” son similares en un 98%. Ambas estructuras utilizan el mismo análisis y el mismo diagrama de flujo. Su diferencia

radica en los pseudocódigos utilizados y en el código de programación. Todos los programas realizados con la estructura de repetición “Mientras” pueden programarse con la estructura de repetición “Para”.

Con la intención de demostrar esta afirmación, se resolverá el ejercicio resuelto con la estructura de repetición “Mientras”, con la estructura de repetición “Para”.

EJEMPLO

**Hacer un programa en código c que genere la serie
2 4 6 8 y mostrarla en la pantalla de la
computadora.**

Figura 30: Enunciado a programar con la estructura de repetición de datos “Para”

CONSTRUCCIÓN DEL PROGRAMA UTILIZANDO EL MÉTODO BALTEX

PASO 1 Bosquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos. (Actividad creativa)

- El objetivo es construir un programa en código c, que permita generar y mostrar la serie 2 4 6 8.
- Según el postulado 1 de la estructura de repetición “Mientras”, los valores de cada término deben almacenarse, temporalmente, en una sola variable.
- Como podemos apreciar, los datos de la serie son de tipo entero, por lo tanto, se requiere declarar una variable entera.
- Luego la variable a utilizarse es la siguiente: ENTERO x
- El primer término de la serie es 2, entonces el primer valor de la variable x debe ser 2.
- El último valor de la serie es 8, entonces la condición lógica propuesta que cumplen todos los valores de la serie puede ser $x \leq 8$ o $x < 10$ que es el siguiente término posible de la serie. Se puede utilizar, indistintamente, cualquier condición lógica propuesta.
- El siguiente término de la serie se incrementa en dos unidades. La fórmula de actualización del valor de la variable que genera la serie es: $x = x + 2$.

TÉCNICA ANALÍTICA

Para analizar los programas usando la estructura de repetición de datos se utilizará la siguiente tabla de seguimiento y control de valores de la variable que genera los datos. La condición lógica que se empleará será $x \leq 8$.

TABLA DE SEGUIMIENTO Y CONTROL DE VALORES DE LA ESTRUCTURA DE REPETICIÓN “PARA”

Valor actual de x	Condición lógica ($x \leq 8$)	Resultado de la condición lógica	Cuerpo de la estructura de repetición “Para”	
			ESCRIBIR (x)	$x = x + 2$
2	$2 \leq 8$	Verdadero	2	$x = 2 + 2$
4	$4 \leq 8$	Verdadero	4	$x = 4 + 2$
6	$6 \leq 8$	Verdadero	6	$x = 6 + 2$
8	$8 \leq 8$	Verdadero	8	$x = 8 + 2$
10	$10 < 8$	Falso		

Tabla 20

La cantidad de columnas de la tabla depende del análisis del programador.

IMPORTANTE: En una columna de la tabla, se visualiza la serie por generar.

Cada una de las líneas de la tabla, cuando la condición es verdadera se llama iteración.

Cuando la condición es falsa se detiene el proceso de repetición de instrucciones.

Nota: Las tildes son omitidas en el contenido de las variables de texto, ya que no las reconoce el wxDev-C++, software que se utilizará para redactar el programa en lenguaje C.

PASO 2 Analizar y definir la arquitectura del diagrama de flujo. (Actividad creativa)

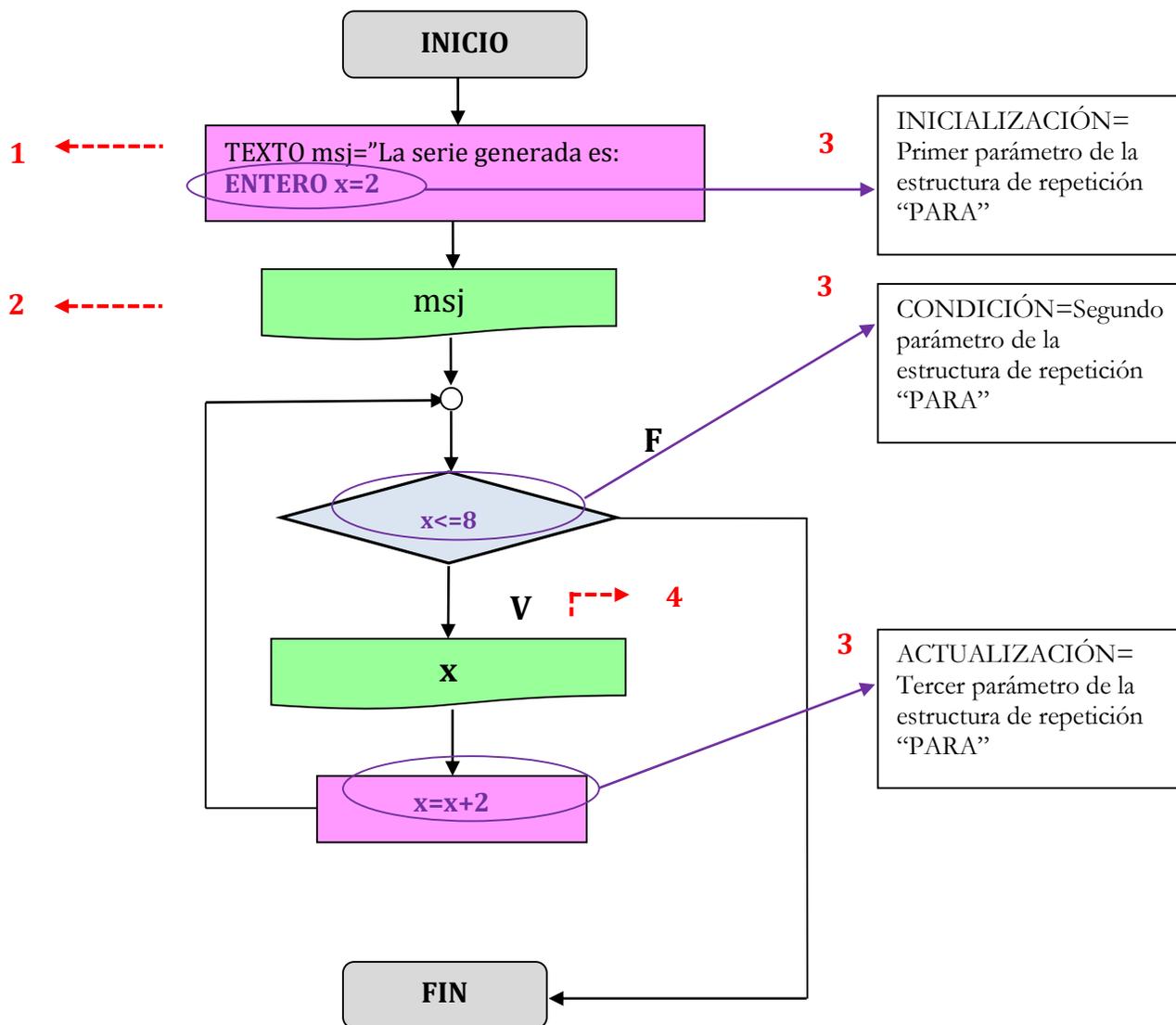


Figura 31: Diagrama de flujo de la estructura de repetición de datos "Para"

PASO 3 Leer el contenido de cada símbolo del diagrama de flujo y traducir a pseudocódigos. (Actividad mecánica)

REFERENCIAS DEL DIAGRAMA DE FLUJO	ALGORITMO : PrgSerieF1
	ENTRADA:
	SALIDA: x
	INICIO
	//Declaración de variables
1	TEXTO msj="La serie generada es:"
1	ENTERO x
	//Visualizar mensaje
2	ESCRIBIR (msj)
	//Condición lógica
3	PARA(x=2, x<=8, x=x+2) HACER
	//Proceso si condición lógica es V
	//Inicio del cuerpo de la estructura de repetición "Para"
4	ESCRIBIR x
	//Fin del cuerpo de la estructura de repetición "Para"
	FIN PARA

Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de repetición de datos "Para"

Tabla 21

PASO 4 Exportar el pseudocódigo a un lenguaje de programación.
(Actividad mecánica)

ALGORITMO : PrgSerieF1	Programa en código c
ENTRADA:	#include <cstdlib>
SALIDA: x	#include <iostream>
	using namespace std;
	int main(int argc, char *argv[])
INICIO	
//Declaración de variables	//Declaración de variables
TEXTO msj="La serie generada es:"	string msj="La serie generada es:";
ENTERO x	int x;
//Visualizar mensaje	//Visualizar mensaje
ESCRIBIR (msj), SALTOLINEA()	cout<<msj<<endl;
//Condición lógica	//Condición lógica
PARA(x=2, x<=8, x=x+2) HACER	for(x=2; x<=8; x=x+2) {
//Proceso si condición lógica es V	//Proceso si condición lógica es V
//Inicio del cuerpo de la estructura de repetición "Para"	//Inicio del cuerpo de la estructura de repetición "Para"
ESCRIBIR (x)	cout<<x<<" ";
//Fin del cuerpo de la estructura de repetición "Para"	//Fin del cuerpo de la estructura de repetición "Para"
FIN PARA	}
ESCRIBIR SALTOLINEA()	cout<<endl;
FIN	system("PAUSE");
	return EXIT_SUCCESS;
	}

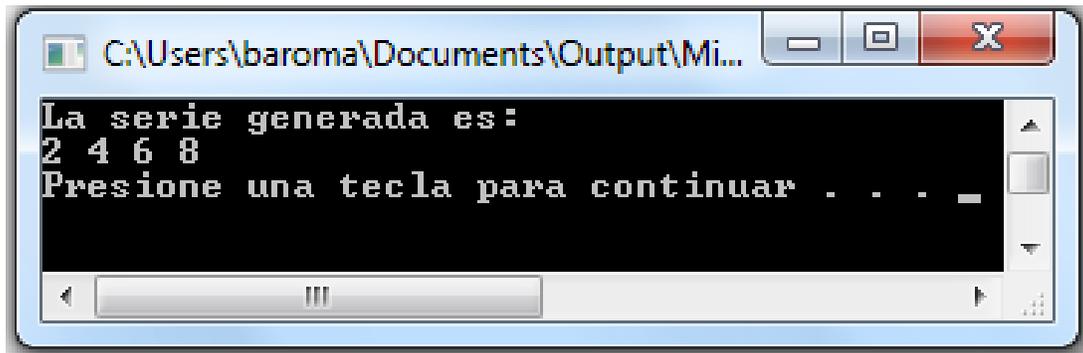
Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de repetición de datos "Para"
Tabla 22

PROGRAMA EN CÓDIGO C REDACTADO EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

```
main.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    //declaración de variables
    string msj="La serie generada es:";
    int x;
    //visualizar mensaje
    cout<<msj<<endl;
    //condición lógica
    for(x=2; x<=8; x=x+2) {
        //proceso si condición lógica es V
        //inicio del cuerpo de la estructura de
        //repetición "Para"
        cout<<x<<" ";
        //fin del cuerpo de la estructura de
        //repetición "Para"
    }
    cout<<endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Figura 32: Programa en código C de la estructura de repetición de datos "Para"

PROGRAMA DESPUÉS DE EJECUTARSE EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++



The image shows a screenshot of a console window from the wxDev-C++ IDE. The window title bar indicates the file path: C:\Users\baroma\Documents\Output\Mi... The console output is as follows:

```
La serie generada es:  
2 4 6 8  
Presione una tecla para continuar . . . -
```

Figura 33: Programa ejecutado en la computadora de repetición de datos "Para"

2.2.16.7 DEFINICIÓN DE LA ESTRUCTURA DE REPETICIÓN DE DATOS “HACER-MIENTRAS”

La estructura de repetición de datos “Hacer-Mientras” permite que se ejecuten diversas instrucciones en forma repetitiva mientras que una condición lógica sea verdadera. Si la condición lógica es falsa no se repiten ese grupo de instrucciones definidas en el cuerpo de la estructura de repetición de datos “Hacer-Mientras”.
(Ver Figura 34)

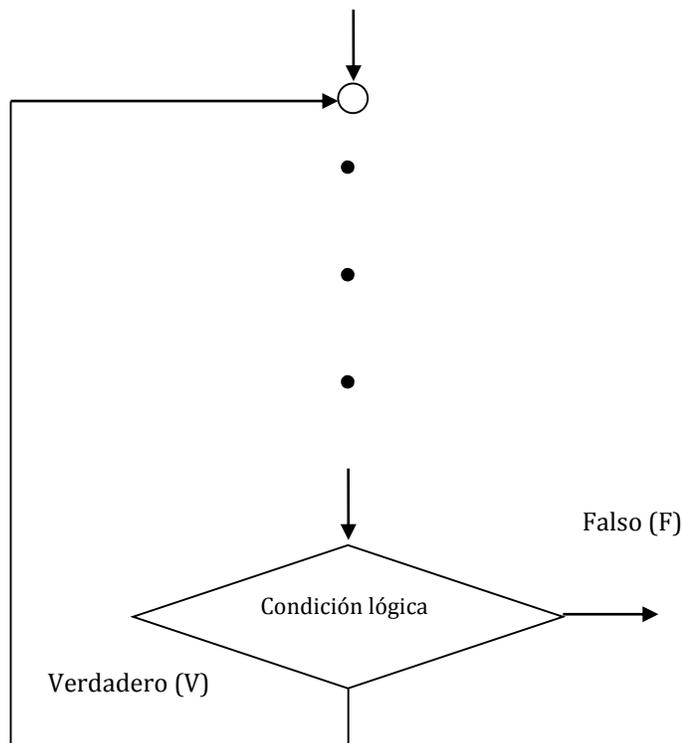


Figura 34: Esquema de comprensión de la estructura de repetición de datos “Hacer-Mientras”

La estructura de repetición de datos se utiliza para garantizar que los datos que necesita un programa estén en un rango de valores que aseguren un dato libre de errores. A esta operación se le conoce como validación de datos.

EJEMPLO

Hacer un programa en código c para que una computadora solamente acepte por el teclado un valor entre tres valores posibles 1, 2 o 3.

Figura 35: Enunciado por programar con la estructura de repetición de datos "Hacer-Mientras"

CONSTRUCCIÓN DEL PROGRAMA UTILIZANDO EL MÉTODO BALTEX

PASO 1 Bosquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos. (Actividad creativa)

- El objetivo es construir un programa en código c, que solo permita el ingreso por el teclado de un solo valor entre tres posibilidades 1, 2 o 3.
- Para almacenar ese valor que se va a ingresar por el teclado, se tendrá que utilizar una variable de tipo entero. En este caso, se utilizará la variable num.
- Se utilizarán la recta numérica y los intervalos abiertos o cerrados para definir la condición lógica que debe cumplir el dato ingresado por el teclado.

TABLA DE SEGUIMIENTO Y CONTROL DE VALORES DE LA ESTRUCTURA DE REPETICIÓN DE DATOS “HACER-MIENTRAS”

Valor ingresado por el teclado y almacenado en la variable num	Condición lógica (num<1 o num>3)	Resultado de la condición lógica	Interpretación lógica
1	(1<1 o 1>3)	Falso	Si la condición lógica es falsa, el número ingresado está en el rango permitido.
	Falso o falso		
2	(2<1 o 2>3)	Falso	Si la condición lógica es falsa, el número ingresado está en el rango permitido.
	Falso o falso		
3	(3<1 o 1>3)	Falso	Si la condición lógica es falsa, el número ingresado está en el rango permitido.
	Falso o falso		
4	(4<1 o 4>3)	Verdadero	Si la condición lógica es verdadera, el número ingresado no está en el rango permitido.
	Falso o verdadero		
0	(0<1 o 0>3)	Verdadero	Si la condición lógica es verdadera, el número ingresado no está en el rango permitido.
	Verdadero o falso		

Tabla 23

Nota: Las tildes son omitidas en el contenido de las variables de texto, ya que no las reconoce el wxDev-C++, software que se utilizará para redactar el programa en lenguaje C.

PASO 2 Analizar y definir la arquitectura del diagrama de flujo. (Actividad creativa)

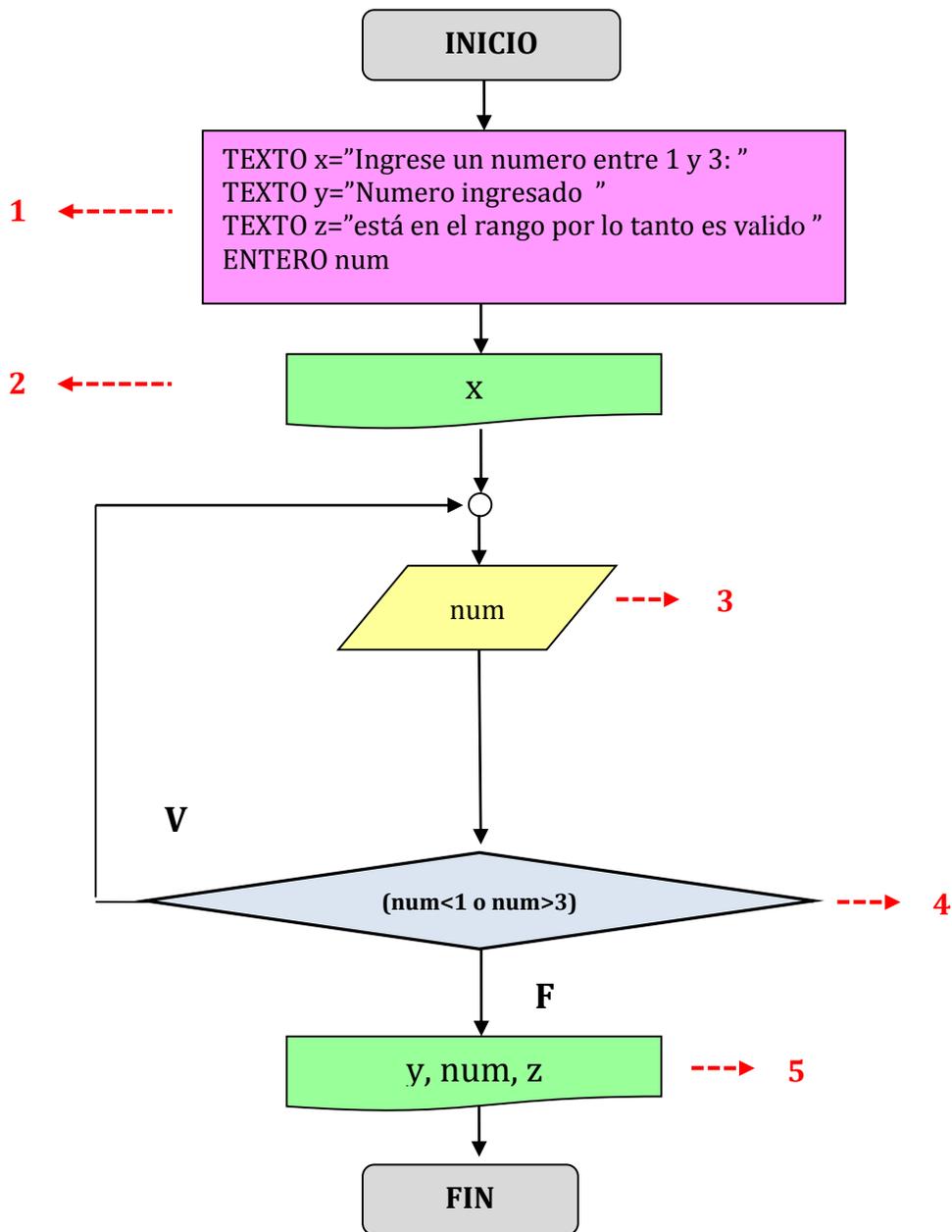


Figura 37: Diagrama de flujo de la estructura repetición de datos "Hacer-Mientras"

PASO 3 Leer el contenido de cada

símbolo del diagrama de flujo y traducir a pseudocódigos.

(Actividad mecánica)

REFERENCIAS DEL DIAGRAMA DE FLUJO	ALGORITMO : PrgDo1
	ENTRADA: num
	SALIDA: y, num, z
	INICIO
	//Declaración de variables
1	TEXTO x="Ingrese un numero entre 1 y 3: "
1	TEXTO y="Numero ingresado "
1	TEXTO z="está en el rango por lo tanto es valido "
1	ENTERO num
	//Inicio del cuerpo de la estructura de repetición HACER-MIENTRAS
	HACER
	//Entrada
2	ESCRIBIR (x)
3	LEER num
	//Fin del cuerpo de la estructura de repetición HACER-MIENTRAS
	//Condición lógica
4	MIENTRAS (num<1 o num>3)
5	ESCRIBIR (y, num, z)
	FIN

Correspondencia entre cada símbolo del diagrama de flujo y el pseudocódigo asociado de la estructura de repetición de datos "Hacer-Mientras"

Tabla 24

PASO 4 Exportar el pseudocódigo a un lenguaje de programación. (Actividad mecánica)

ALGORITMO : PrgDo1	Programa en código c
ENTRADA: num	#include <cstdlib>
SALIDA: y, num, z	#include <iostream>
	using namespace std;
	int main(int argc, char *argv[])
INICIO	
//Declaración de variables	//Declaración de variables
TEXTO x="Ingrese un numero entre 1 y 3: "	string x="Ingrese un numero entre 1 y 3: ";
TEXTO y="Numero ingresado "	string y="Numero ingresado ";
TEXTO z="está en el rango por lo tanto es valido "	string z="esta en el rango por lo tanto es valido ";
ENTERO num	int num;
//Inicio del cuerpo de la estructura de repetición HACER-MIENTRAS	//Inicio del cuerpo de la estructura de repetición HACER-MIENTRAS
HACER	do {
//Entrada	//Entrada
ESCRIBIR x	cout<<x;
LEER num	cin>>num;
//Fin del cuerpo de la estructura de repetición HACER-MIENTRAS	//Fin del cuerpo de la estructura de repetición HACER-MIENTRAS
//Condición lógica	//Condición lógica
MIENTRAS (num<1 o num>3)	} while (num<1 num>3);
ESCRIBIR (y, num, z, SALTOLINEA())	cout<<y<<num<<z<<endl;
FIN	system("PAUSE");
	return EXIT_SUCCESS;
	}

Correspondencia entre cada pseudocódigo y el código C asociado de la estructura de repetición de datos "Hacer-Mientras"
Tabla 25

PROGRAMA EN CÓDIGO C REDACTADO EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++

```
main.cpp
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    //declaración de variables
    string x="Ingrese un numero entre 1 y 3: ";
    string y="Numero ingresado ";
    string z=" esta en el rango por lo tanto es valido ";
    int num;
    //inicio del cuerpo de la estructura de repetición HACER-MIENTRAS
do {
    //ingreso de dato
    cout<<x;
    cin>>num;
    //fin del cuerpo de la estructura de repetición HACER-MIENTRAS
    //condición logica
} while (num<1 || num>3);
cout<<y<<num<<z<<endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Figura 38: Programa en código C de la estructura de repetición de datos “Hacer-Mientras”

PROGRAMA DESPUES DE EJECUTARSE EN EL ENTORNO DE DESARROLLO LIBRE wxDev-C++



```
C:\Users\baroma\Documents\Output\MingW\ProjectBB.exe
Ingrese un numero entre 1 y 3: -3
Ingrese un numero entre 1 y 3: 4
Ingrese un numero entre 1 y 3: 1
Numero ingresado 1 esta en el rango por lo tanto es valido
Presione una tecla para continuar . . .
```

Figura 39: Programa ejecutado en la computadora de repetición de datos “Hacer-Mientras”

2.2.17 APLICACIÓN DE LOS PRINCIPIOS DE LA ORIENTACIÓN EDUCATIVA EN EL MÉTODO BALTEX

En el momento de desarrollar la presente investigación, me desempeñaba como docente de la institución educativa Cibertec. Dicté el curso de Introducción a la Algoritmia, en que los son conocimientos de tipo lógico-analítico, para que los alumnos puedan ejecutar un conjunto de instrucciones en un programa que pueda ser realizado en una computadora.

Los requisitos de la asignatura son los siguientes:

- Demostrar dominio de operaciones aritméticas.
- Poseer conocimientos básicos del Álgebra.
- Realizar operaciones lógicas simples y complejas.
- Poseer buen raciocinio matemático.
- Mostrar disposición para el aprendizaje.

Aplicando el **principio de prevención** elaboré siete recomendaciones básicas antes de iniciar el curso. Son las siguientes:

1. Escuchar con atención las clases teóricas.
2. Practicar las clases de laboratorio en su computadora personal.
3. Aplicar los conocimientos adquiridos a través de ejercicios y tareas.
4. Preguntar al docente orientador, sobre cualquier duda que tuviese el aprendiz.
5. Rendir evaluaciones con eficiencia.
6. Cumplir con las indicaciones del docente orientador, a fin de evitar errores.
7. Reflexionar y aprender de los errores cometidos.

Al aplicar **el principio de desarrollo** se trata de dotar al alumno de habilidades y destrezas para iniciar la elaboración de programas para computadora, en las siete estructuras básicas de aprendizaje de la computación utilizando el Método Baltex.

Al aplicar el **principio de intervención social**, se concentran esfuerzos en las tareas que deben ejecutarse en equipos bajo una metodología definida, para que les dé confianza en la participación en el aula.

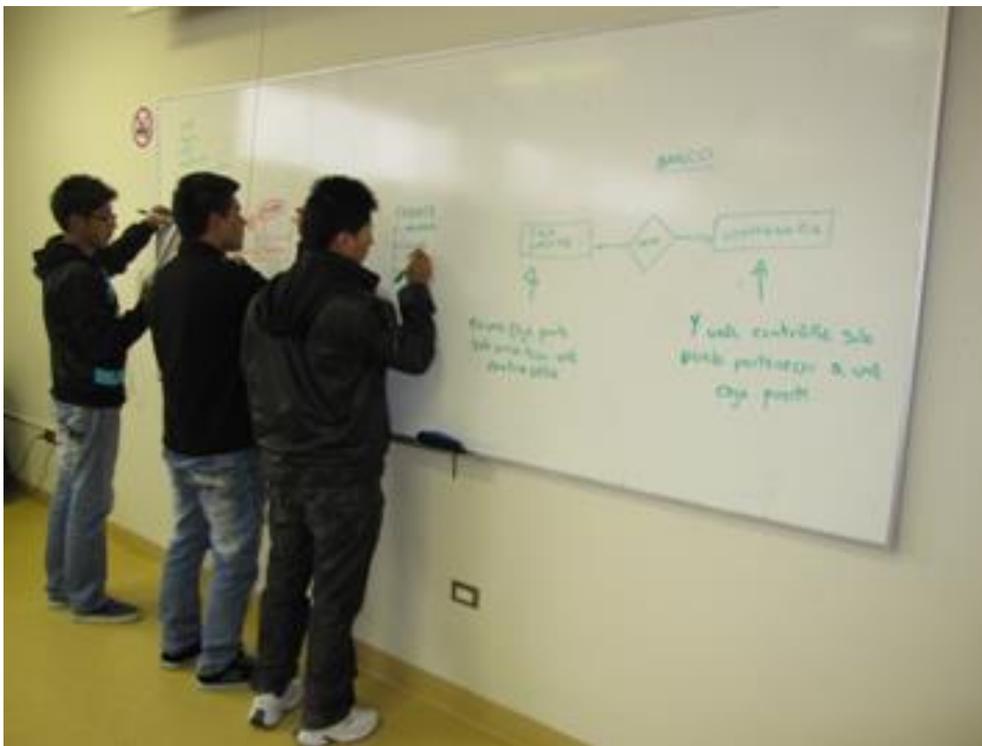


Figura 40: Fotografía de estudiantes del Programa Nacional Beca 18 donde se aplica el principio de intervención social de la orientación educativa utilizando el Método Baltex

Al aplicar el **principio de fortalecimiento personal** se trata de que los alumnos desarrollen habilidades y capacidades para realizar un buen trabajo y reconocer sus logros.

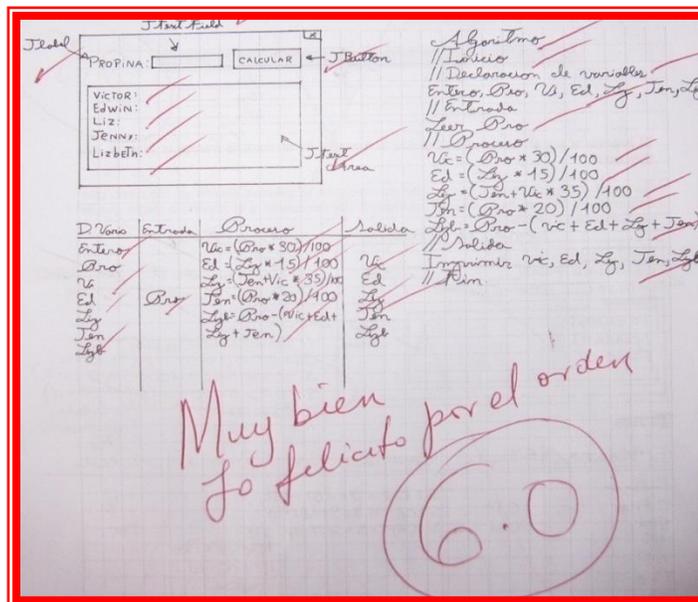


Figura 41: Fotografía con la solución acertada de una pregunta en una evaluación de un estudiante del Programa Nacional Beca 18 donde se aplica el principio de fortalecimiento personal de la orientación educativa utilizando el Método Baltex.



Figura 42: Fotografía con estudiantes del Programa Nacional Beca 18 en que se aplica el principio de fortalecimiento personal de la orientación educativa utilizando el Método Baltex.

2.2.18 DEFINICIÓN DE TÉRMINOS BÁSICOS

- **ALGORITMO:** Constituye una lista completa de instrucciones que se deben ejecutar en pasos secuenciales para resolver un determinado problema.
- **CÓDIGO:** Modo de representar los datos y los programas de modo que puedan ser aceptados y transformados por la computadora.
- **COMPUTACIÓN:** La Computación es el estudio de métodos algorítmicos para representar y transformar la información, en el que se incluyen los conceptos teóricos para realizar el diseño, la implementación y la mejora continua a través de la retroalimentación.

Las bases de la Computación se apoyan profundamente en el razonamiento matemático y la ingeniería. La Matemática coadyuva en el análisis y la ingeniería sustenta el diseño del programa.

- **DATO:** Elemento básico que necesita un programa de computadora para transformarlo con operaciones aritméticas y lógicas para producir un resultado.
- **DIAGRAMA DE FLUJO:** Es la representación gráfica de los pasos de un proceso, que se realiza para entender mejor al mismo, comprender y analizar los hechos.

- **EXPORTAR:** En el argot de la Computación, significa convertir un pseudocódigo en un lenguaje de programación.
- **INSTRUCCIÓN:** Orden que se le asigna a la computadora.
- **LENGUAJE C:** Es un lenguaje de programación de propósito general asociado, de modo universal, al sistema operativo UNIX. Sin embargo, la popularidad, eficacia y potencia de C se concreta porque este lenguaje no está prácticamente asociado a ningún sistema operativo que no sea UNIX, ni a ninguna máquina en especial.
- **LENGUAJE DE PROGRAMACIÓN:** Conjunto de signos y reglas que permite la comunicación con una computadora.
- **LIBRERÍA:** Es un conjunto de recursos y algoritmos prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones.
- **#include<stdlib>** : Es una librería que tiene funciones que sirven para controlar ciertos recursos del sistema operativo.
- **#include<iostream>**: Es una librería que contiene las funciones de ingresar y mostrar datos.
- **#include<Math.h>**: Es una librería que contiene las funciones matemáticas comunes.

- **using namespace std:** Es una sentencia que facilita la captura de datos con la instrucción “cin>>” y la visualización de variables con la instrucción “cout<<”.
- **int main (int argc, char* argv[]); :** Es la función principal de un programa en Lenguaje C, representa el punto de inicio de su ejecución.
- **system(“PAUSE”):** Sentencia para detener el programa.
- **return EXIT_SUCCESS:** Sentencia para saber si la salida de una función o proceso es exitoso.
- **Encabezado básico de un programa en lenguaje C:** Es la escritura de las siguientes instrucciones que indican el inicio de un programa en código C en forma práctica.

```

#include<cstdlib>
#include<iostream>
#include<Math.h>
using namespace std;
int main(int argc, char* argv[ ]);
{

```

- **Pie de página básico de un programa en lenguaje C:** Es la escritura de las siguientes instrucciones que indican el fin de un programa en código C en forma práctica.

```
system ("PAUSE");  
return EXIT_SUCCESS;  
}
```

- **MÉTODO:** Modo ordenado y estructurado de obtener un resultado, descubrir la verdad y sistematizar los conocimientos.
- **MÉTODO BALTEX:** Método para construir programas computarizados, orientado a novatos, diseñado por el Ing. Manuel Valeriano Balta Rospigliosi. Consta de los siguientes cuatro pasos.

Bosquejar el entendimiento del problema mediante un análisis y definir las entradas y salidas de datos.

Analizar y definir la arquitectura del diagrama de flujo

Leer y traducir el contenido de cada símbolo del diagrama de flujo a pseudocódigos para configurar el algoritmo.

Exportar o codificar el algoritmo a un lenguaje de programación de computadoras.

- **MS-DOS:** Sistema operativo Disk Operating System de la empresa Microsoft.
- **POSTULADO:** Proposición que se toma como base para un razonamiento o demostración, cuya verdad se admite sin pruebas.

- **PROCESO:** Constituye una actividad de transformación, cuyo objetivo es combinar los datos ingresados en la computadora que selecciona un programador para obtener datos que representen una respuesta a un requerimiento de automatización.
- **PROGRAMA:** Secuencias de instrucciones detalladas y codificadas que sirven para dirigir la realización de operaciones de una computadora.
- **PSEUDOCÓDIGO:** Es un lenguaje algorítmico informal, que es una imitación de uno o más lenguajes de programación de alto nivel. Un pseudocódigo puede contener símbolos (+, -, *, /, =, etc.), palabras (LEER, ESCRIBIR, etc.) y estructuras de programación (SI, SINO, MIENTRAS, HACER MIENTRAS, PARA).
- **SENTENCIA:** Órdenes especiales que especifican y controlan el flujo de ejecución del programa.
- **VARIABLE:** Es un espacio de memoria de la computadora reservado para almacenar un valor que corresponde a un tipo de dato soportado por el lenguaje de programación.

Los nombres de las variables deben comenzar con una letra, un símbolo subrayado(_) o un símbolo de dólar (\$).

Los nombres de las variables no pueden ser palabras reservadas de un lenguaje de programación.

Las letras mayúsculas y minúsculas se consideran diferentes.

2.2.19 JUSTIFICACIÓN DEL ESTUDIO

En el aspecto teórico, la investigación contribuyó con enfoques para abordar nuevas formas de entender los problemas haciendo énfasis en la comprensión y entendimiento de los tópicos de aritmética y lógica.

En el aspecto metodológico, se han propuesto diversas técnicas analíticas para asimilar, en 17 semanas, conceptos de alta complejidad que se requieren en la actividad de programación.

Desde el aspecto práctico, se consideró que la presente investigación aporta nuevos conocimientos en el proceso de aprendizaje de la computación básica; evitar consecuencias negativas como la frustración en los estudiantes, la deserción y la desmotivación; elevar las vivencias de éxito al comprobar el funcionamiento de los programas elaborados al utilizar el Método Baltex por los estudiantes del Programa Nacional Beca 18, que se matricularon en la sección T1RM en el curso de Introducción a la Algoritmia del primer ciclo de la Carrera Técnica Profesional de Computación e Informática en el semestre 2015-I en la sede de Miraflores de la institución educativa Cibertec.

Asimismo, los resultados del estudio sirven de base para mejorar el método de aprendizaje de futuros estudiantes del Programa Nacional Beca 18 lo que otorgará fortalecimiento académico a las organizaciones de nivel superior que ofrezcan estudios tecnológicos en el Perú.

CAPÍTULO III: OBJETIVOS E HIPÓTESIS

3.1 OBJETIVOS

3.1.1 OBJETIVO GENERAL:

Determinar los efectos del Método BALTEX, sobre el aprendizaje procedimental de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I

3.1.2 OBJETIVOS ESPECÍFICOS:

- Determinar los efectos del Método BALTEX sobre la capacidad de reflexión, entendimiento y planificación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.
- Determinar los efectos del Método BALTEX sobre la capacidad de adaptar un panorama visual de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.
- Determinar los efectos del Método BALTEX sobre la capacidad de redacción de pseudocódigos de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.
- Determinar los efectos del Método BALTEX sobre la capacidad de programación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.

3.2 HIPÓTESIS

3.2.1 HIPÓTESIS GENERAL

El Método BALTEX logra mejoras significativas sobre el aprendizaje procedimental de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.

3.2.2 HIPÓTESIS ESPECÍFICAS

- El Método BALTEX logra mejoras significativas en la capacidad de reflexión, entendimiento y planificación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.
- El Método BALTEX logra mejoras significativas sobre la capacidad de adaptar un panorama visual de los estudiantes del primer de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.
- El Método BALTEX logra mejoras significativas sobre la capacidad redacción de pseudocódigos de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.
- El Método BALTEX logra mejoras significativas sobre la capacidad de programación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I.

CAPÍTULO IV: METODOLOGÍA DE LA INVESTIGACIÓN

4.1 DISEÑO DE LA INVESTIGACIÓN

4.1.1 DISEÑO CUASIEXPERIMENTAL

Respecto a este diseño, HERNÁNDEZ, R., FERNÁNDEZ, C. & BAPTISTA, L. (2014), afirman que:

Los diseños cuasiexperimentales manipulan deliberadamente, al menos, una variable independiente para observar su efecto sobre una o más variables dependientes, solo que difieren de los experimentos “puros” en el grado de seguridad que pueda tenerse sobre la equivalencia inicial de los grupos. En los diseños cuasiexperimentales, los sujetos no se asignan al azar a los grupos ni se emparejan, sino que dichos grupos ya están conformados antes del experimento: son grupos intactos (la razón por la que surgen y la manera cómo se integraron es independiente o aparte del experimento) (p.151).

La presente investigación se desarrolló desde un diseño cuasiexperimental, debido a que se trabajó con un grupo experimental y un grupo de control, y ambos fueron previamente formados en la institución educativa superior Cibertec.

4.1.2 ALCANCE EXPLICATIVO

Acerca de las investigaciones de diseño cuasiexperimental, HERNÁNDEZ, R., FERNÁNDEZ, C. & BAPTISTA, L. (2014) sostienen:

Debido a que analizan las relaciones entre una o más variables independientes y una o más dependientes, así como los efectos causales de las primeras sobre las segundas, son estudios explicativos. Se trata de diseños que se fundamentan en el enfoque cuantitativo y en el paradigma deductivo. Se basan en hipótesis preestablecidas, miden variables y su aplicación debe sujetarse al diseño concebido con antelación; al desarrollarse, el investigador está centrado en la validez, el rigor y el control de la situación de investigación. Asimismo, el análisis estadístico resulta fundamental para lograr los objetivos de conocimiento (p. 150).

Por tanto, debido a que la presente investigación se desarrolló bajo un diseño cuasiexperimental, en el que se realizó un análisis de las mejoras de la aplicación del método Baltex, en el grupo experimental, sobre las capacidades del aprendizaje procedimental, esta tuvo un alcance explicativo.

4.1.3 ALCANCE DESCRIPTIVO

Respecto a las investigaciones de alcance descriptivo, HERNÁNDEZ, R., FERNÁNDEZ, C. & BAPTISTA, L. (2014), afirman que “buscan especificar las propiedades, las características y perfiles de personas, grupos, comunidades, procesos, objetos o cualquier fenómeno. Pretenden medir o recoger información

de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren” (p. 92).

Por tanto, la presente investigación cumplió también con un alcance descriptivo, debido a que se realizaron mediciones de las capacidades de aprendizaje procedimental de estudiantes en diversos momentos, analizando sus estados respectivos.

4.2 ENFOQUE CUANTITATIVO

Respecto a este enfoque, HERNÁNDEZ, R., FERNÁNDEZ, C. & BAPTISTA, L. (2014), señalan:

El enfoque cuantitativo es secuencial y probatorio. Cada etapa precede a la siguiente y no podemos “brincar” o eludir pasos. El orden es riguroso, aunque desde luego, podemos redefinir alguna fase. Parte de una idea que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica. De las preguntas se establecen hipótesis y determinan variables; se traza un plan para probarlas; se miden las variables en un determinado contexto; se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones respecto de la o las hipótesis (p. 4).

La presente investigación tuvo un enfoque cuantitativo, debido a que se siguieron pasos sistemáticos y planificados que comprendieron, entre otros, la formulación

del problema de investigación, el planteamiento de objetivos, la revisión de fuentes de información, la formulación de hipótesis, la ejecución de un trabajo de campo, un diseño de base de datos, un análisis estadístico de datos para obtener descripciones de la muestra, y un conjunto de pruebas estadísticas con el fin de probar las hipótesis y dar respuesta a los problemas de investigación.

El diseño descrito se presenta en el siguiente esquema:

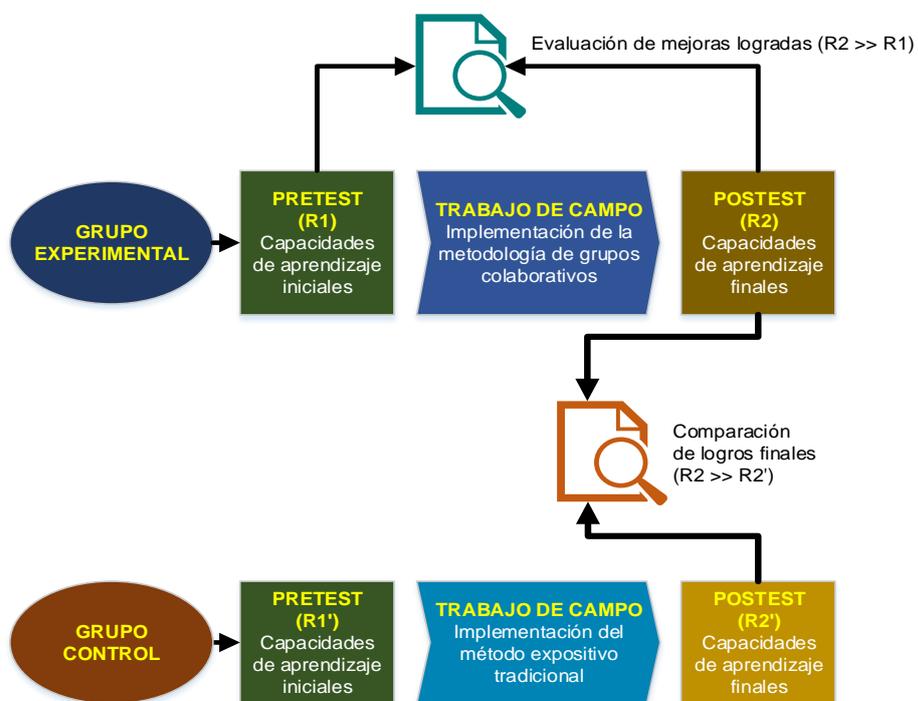


Figura 43. Representación gráfica del diseño de la investigación

4.3 POBLACIÓN Y MUESTRA

- **POBLACIÓN**

La población estuvo constituida por los estudiantes matriculados en la Asignatura de Introducción a la Algoritmia de la carrera técnica profesional de Computación e Informática de la institución educativa superior Cibertec, de la Sede de Miraflores, quienes sumaron un total de 112 estudiantes.

- **MUESTRA**

Para la obtención de la muestra, se aplicó un muestreo no probabilístico por conveniencia, pues la selección de las aulas de clase en las que se realizó el trabajo de campo y la comparación se basó en las disposiciones de las autoridades de la institución educativa Superior Cibertec. Además, se aplicó un criterio de exclusión, con base a una asistencia mínima del 70% de las sesiones de clase. La muestra final se detalla en el siguiente diagrama:

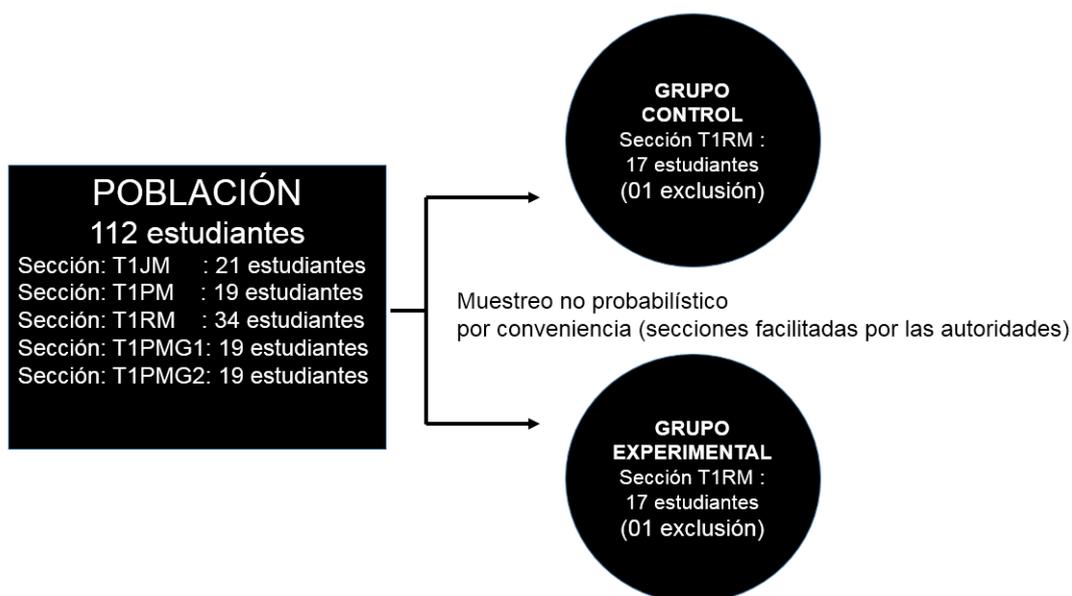


Figura 44. Diagrama de la población, muestra y muestreo de la investigación

4.4 OPERACIONALIZACIÓN DE VARIABLES

Operacionalización de la variable independiente: “Método Baltex” para el grupo experimental

Etapas	Pasos	Instrumentos de control
Planificación	<ul style="list-style-type: none"> • Diseño de materiales • Diseño de sesiones de aprendizaje • Diseño de evaluaciones (pretest y postest) 	
Ejecución	<ul style="list-style-type: none"> • Aplicación de pretest • Desarrollo de sesiones de aprendizaje. • Desarrollo de actividades prácticas • Desarrollo de asignaciones • Desarrollo de tutorías • Utilización de técnicas analíticas. • Realización de diagramas de flujo • Realización de pseudocódigos • Realización de programas 	<ul style="list-style-type: none"> • Lista de cotejo • Plan curricular
Evaluación	<ul style="list-style-type: none"> • Aplicación del postest • Comparación de resultados 	

Tabla 26

Operacionalización de la variable independiente: “Método tradicional” para el grupo control

Etapas	Pasos	Instrumentos de control
Planificación	<ul style="list-style-type: none"> • Diseño de materiales • Diseño de sesiones de aprendizaje • Diseño de evaluaciones (pretest y postest) 	
Ejecución	<ul style="list-style-type: none"> • Aplicación de pretest • Desarrollo de sesiones de aprendizaje. • Desarrollo de actividades prácticas • Desarrollo de asignaciones 	<ul style="list-style-type: none"> • Lista de cotejo • Plan curricular
Evaluación	<ul style="list-style-type: none"> • Aplicación del postest • Comparación de resultados 	

Tabla 27

Operacionalización de la variable dependiente: “Aprendizaje procedimental”

Dimensiones	Indicadores	Ítems	Niveles y rangos
Dimensión 1 Capacidad de reflexión, entendimiento y planificación	Identifica los datos de entrada, proceso y salida.	<ul style="list-style-type: none"> Identifica las variables de entrada. Identifica las variables de salida. Realiza el proceso 	<p>Para la variable Alto (60 - 80) Medio (38 - 59) Bajo (16 - 37)</p> <p>Para las dimensiones Alto (16 - 20) Medio (10 - 15) Bajo (4 - 9)</p>
	Aplica conocimientos de matemáticas y lógica.	<ul style="list-style-type: none"> Realiza la tabla analítica 	
Dimensión 2 Capacidad de adaptar un panorama visual	Usa técnicas necesarias para la práctica de la computación.	<ul style="list-style-type: none"> Identifica los símbolos gráficos de las variables de entrada. Identifica los símbolos gráficos de las variables de salida. Identifica los símbolos gráficos del proceso 	
	Traduce la tabla analítica en un diagrama de flujo.	<ul style="list-style-type: none"> Realiza el diagrama de flujo 	
Dimensión 3 Capacidad de redacción de pseudocódigos	Comprende procesos que soportan los órdenes que se le debe dar a una computadora.	<ul style="list-style-type: none"> Identifica los pseudocódigos de las variables de entrada. Identifica los pseudocódigos de las variables de salida. Identifica los pseudocódigos del proceso. 	
	Traduce los símbolos del diagrama de flujo a pseudocódigos más adecuados.	<ul style="list-style-type: none"> Realiza el pseudocódigo 	
Dimensión 4 Capacidad de programación	Implementa componentes que satisfacen necesidades que se requieren automatizar.	<ul style="list-style-type: none"> Codifica los pseudocódigos de las variables de entrada en el lenguaje de programación. Codifica los pseudocódigos de las variables de salida en el lenguaje de programación. Codifica los procesos secuenciales, selectivos y repetitivos en el lenguaje de programación. 	
	Traduce los pseudocódigos a códigos de un lenguaje de programación.	<ul style="list-style-type: none"> Realiza el programa 	

Tabla 28

4.5 TÉCNICAS PARA LA RECOLECCIÓN DE DATOS

La presente investigación hizo uso de la técnica de evaluación. Esta comprendió el uso de una prueba de evaluación escrita de entrada, un examen final y un cuestionario, detallados a continuación:

4.5.1 DESCRIPCIÓN DE LOS INSTRUMENTOS

- **Prueba de evaluación escrita:** Evaluó las siguientes dimensiones:
 1. Capacidad de reflexión, entendimiento y planificación.
 2. Capacidad de adaptar un panorama visual.
 3. Capacidad de redacción de pseudocódigos.
 4. Capacidad de programación.

- **Cuestionario de actitudes:** Fue aplicado mediante la técnica de encuestas. Evaluó las opiniones del estudiante respecto al método tradicional y al método Baltex considerando la:
 1. Planificación
 2. Ejecución
 3. Evaluación

Esta tuvo respuestas cerradas en Escala Likert, con las siguientes respuestas:

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo.

4.5.2 VALIDEZ Y CONFIABILIDAD DE LOS INSTRUMENTOS DE RECOLECCIÓN DE DATOS

- **Validez:** Los instrumentos fueron validados por juicio de expertos calificados (3 docentes universitarios). Los informes respectivos cuales validan la aprobación requerida, se muestran en el anexo 2.
- **Confiabilidad:**
Previo a la aplicación de los instrumentos de recolección de datos, se realizaron pruebas de confiabilidad de homogeneidad, para asegurar que los datos a recolectar sean fiables. La prueba seleccionada fue la del cálculo del Coeficiente Alfa de Cronbach, considerando un valor mínimo de confiabilidad del 75% (0.75), en un grupo piloto de 15 estudiantes. La prueba dio los siguientes resultados:

Resultados de la Prueba de Confiabilidad - Coeficiente Alfa de Cronbach

Variable / Dimensión	Coeficiente calculado
Variable dependiente: Aprendizaje procedimental	0,979062
Dimensión 1: Capacidad de reflexión, entendimiento y planificación	0,944744
Dimensión 2: Capacidad de adaptar un panorama visual	0,952821
Dimensión 3: Capacidad de redacción de pseudocódigos	0,930176
Dimensión 4: Capacidad de programación	0,974239

Tabla 29

De acuerdo con la tabla 4, el valor del Coeficiente Alfa de Cronbach, tanto para la variable dependiente como sus dimensiones, fue siempre superior al mínimo establecido (0.75). Por tanto, se afirmó que el instrumento por aplicarse fue capaz de recolectar datos fiables.

4.6 TÉCNICAS PARA EL PROCESAMIENTO Y ANÁLISIS DE LOS DATOS

- **Tipo de análisis de datos:** Cuantitativo.

- **Organización de datos:** Clasificación a base de variables y dimensiones. Los datos fueron la base para describir las variables y dimensiones a base de las frecuencias y porcentajes de sus valores posibles.

- **Procesamiento de datos:**
 - Software para procesamiento estadístico: SPSS versión 24.
 - Gráficas estadísticas: gráficos de barra.
 - Prueba de hipótesis: verificación de estado de normalidad. Los resultados determinaron el uso de la Prueba T (muestras relacionadas e independiente), o bien el uso de la Prueba de Wilcoxon y la Prueba U Mann-Withney.

4.7 ASPECTOS ÉTICOS

La investigación cumplió con los principios básicos de equidad de raza, género y credo, puesto que no se realizaron discriminaciones a base de estos criterios. Asimismo, se respetó la confidencialidad de los estudiantes, al no difundir sus identidades. Por otro lado, la investigación respetó todo derecho de autor mediante la citación respectiva durante la elaboración del marco teórico, además de listar sus publicaciones en las fuentes de información. Finalmente, se cumplió con el principio de respeto a la verdad, mediante la no alteración de los datos recolectados.

CAPÍTULO V: RESULTADOS DE LA INVESTIGACIÓN

5.1 RESULTADOS DESCRIPTIVOS EN EL GRUPO EXPERIMENTAL

Resultados respecto a la variable dependiente: “Aprendizaje procedimental”

Tabla 30

Tabla de frecuencias de la variable dependiente: “Aprendizaje procedimental”, en el grupo experimental

Nivel	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (16 - 37)	10	58.82%	0	0.00%
Medio (38 - 59)	6	35.29%	3	17.65%
Alto (60 - 80)	1	5.88%	14	82.35%

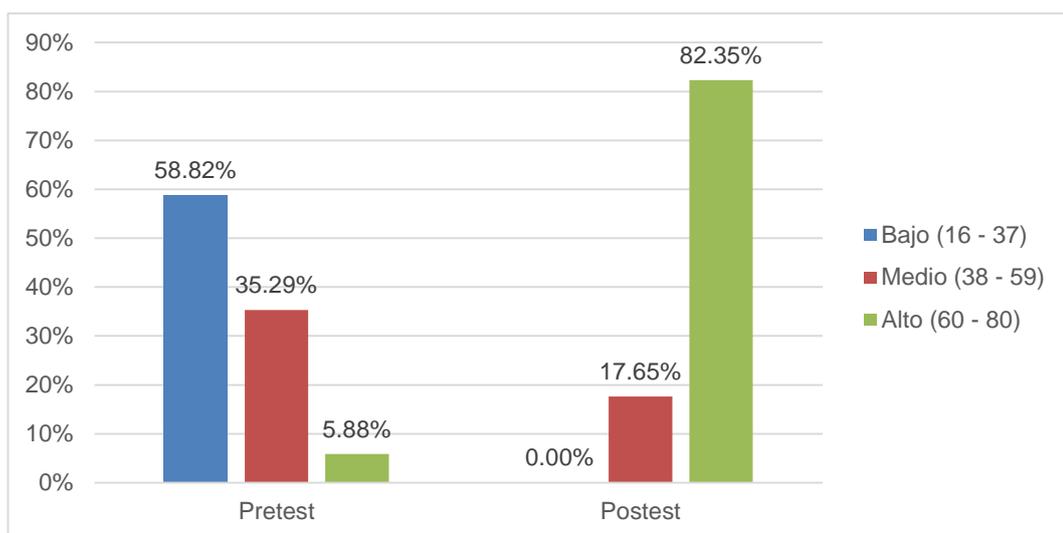


Figura 45. Gráfico de barras de la variable dependiente: “Aprendizaje procedimental”, en el grupo experimental

De acuerdo con la tabla 5 y la figura 1, se determina lo siguiente:

- En el caso del pretest, el 58.82% de los estudiantes abordados indicaron un nivel bajo de aprendizaje procedimental, mientras que el 35.29% indicaron un nivel medio, y el 5.88% indicaron un nivel alto.
- En el caso del postest, el 17.65% de los estudiantes abordados indicaron un nivel medio de aprendizaje procedimental, mientras que el 82.35% indicaron un nivel alto.

Resultados respecto a la dimensión 1: Capacidad de reflexión, entendimiento y planificación

Tabla 31

Tabla de frecuencias de la Dimensión 1: Capacidad de reflexión, entendimiento y planificación

Nivel	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	9	52.94%	0	0.00%
Medio (10 - 15)	7	41.18%	2	11.76%
Alto (16 - 20)	1	5.88%	15	88.24%

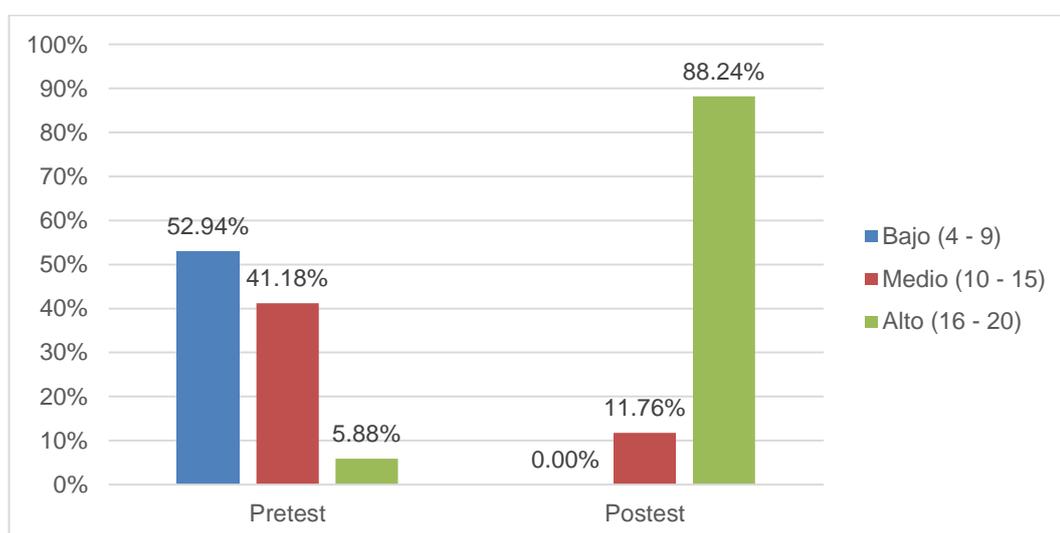


Figura 46. Gráfico de barras de la dimensión 1: Capacidad de reflexión, entendimiento y planificación

De acuerdo con la tabla 6 y la figura 2, se determina lo siguiente:

- En el caso del pretest, el 52.94% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de reflexión, entendimiento y planificación, mientras que el 41.18% indicaron un nivel medio, y el 5.88% indicaron un nivel alto.
- En el caso del postest, el 11.76% de los estudiantes abordados indicaron un nivel medio respecto a su capacidad de reflexión, entendimiento y planificación, mientras que el 88.24% indicaron un nivel alto.

Resultados respecto a la dimensión 2: Capacidad de adaptar un panorama visual

Tabla 32

Tabla de frecuencias de la dimensión 2: Capacidad de adaptar un panorama visual

Nivel	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	11	64.71%	0	0.00%
Medio (10 - 15)	6	35.29%	3	17.65%
Alto (16 - 20)	0	0.00%	14	82.35%

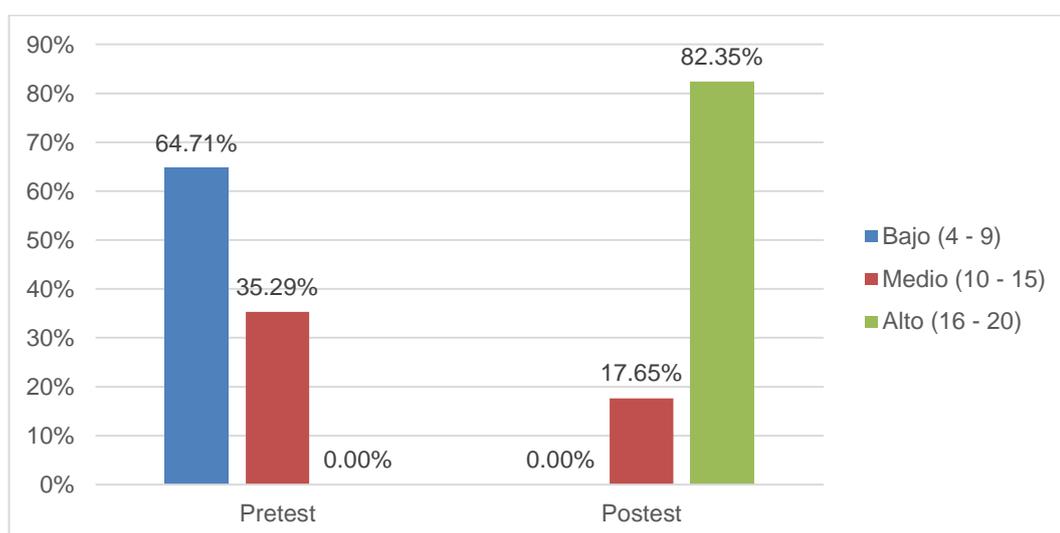


Figura 47. Gráfico de barras de la dimensión 2: Capacidad de adaptar un panorama visual

De acuerdo con la tabla 7 y la figura 3, se determina lo siguiente:

- En el caso del pretest, el 64.71% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de adaptar un panorama visual, mientras que el 35.29% indicaron un nivel medio.
- En el caso del postest, el 17.65% de los estudiantes abordados indicaron un nivel medio respecto a su capacidad de adaptar un panorama visual, mientras que el 82.35% indicaron un nivel alto.

Resultados respecto a la dimensión 3: Capacidad de redacción de pseudocódigos

Tabla 33

Tabla de frecuencias de la dimensión 3: Capacidad de redacción de pseudocódigos

Nivel	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	11	64.71%	0	0.00%
Medio (10 - 15)	4	23.53%	3	17.65%
Alto (16 - 20)	2	11.76%	14	82.35%

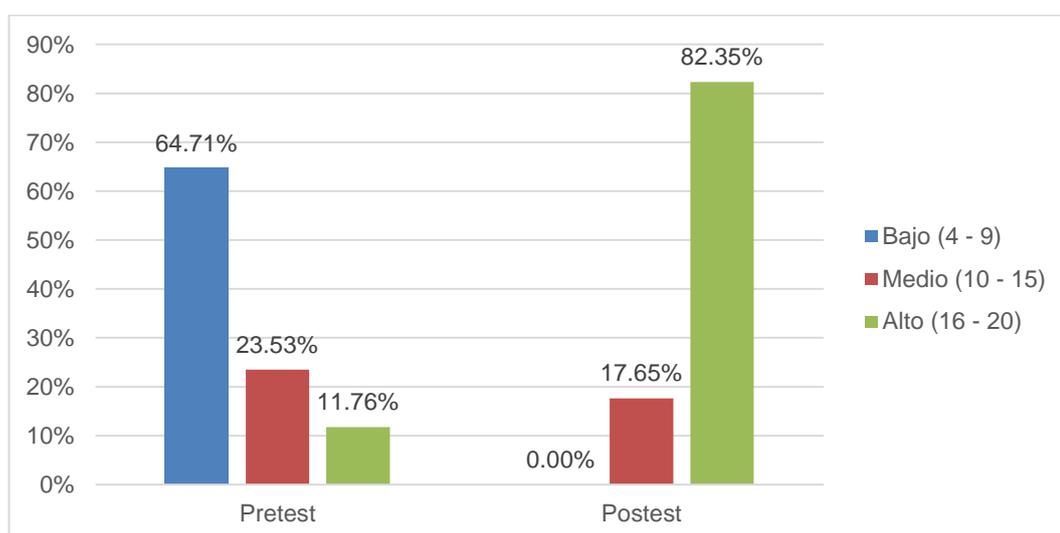


Figura 48. Gráfico de barras de la dimensión 3: Capacidad de redacción de pseudocódigos

De acuerdo con la tabla 8 y la figura 4, se determina lo siguiente:

- En el caso del pretest, el 64.71% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de redacción de pseudocódigos, mientras que el 23.53% indicaron un nivel medio, y el 11.76% indicaron un nivel alto.
- En el caso del postest, el 17.65% de los estudiantes abordados indicaron un nivel medio respecto a su capacidad de redacción de pseudocódigos, mientras que el 82.35% indicaron un nivel alto.

Resultados respecto a la dimensión 4: Capacidad de programación

Tabla 34

Tabla de frecuencias de la dimensión 4: Capacidad de programación

Nivel	Pretest		Posttest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	10	58.82%	0	0.00%
Medio (10 - 15)	6	35.29%	5	29.41%
Alto (16 - 20)	1	5.88%	12	70.59%

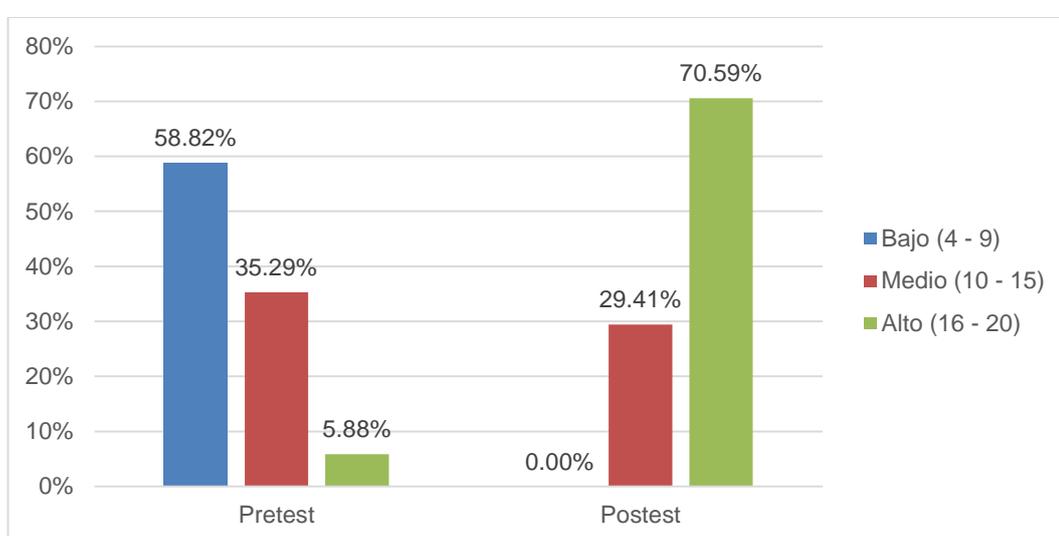


Figura 49. Gráfico de barras de la dimensión 4: Capacidad de programación

De acuerdo con la tabla 9 y la figura 5, se determina lo siguiente:

- En el caso del pretest, el 58.82% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de programación, mientras que el 35.29% indicaron un nivel medio, y el 5.88% indicaron un nivel alto.
- En el caso del posttest, el 29.41% de los estudiantes abordados indicaron un nivel medio respecto a su capacidad programación, mientras que el 70.59% indicaron un nivel alto.

5.2 RESULTADOS DESCRIPTIVOS EN EL GRUPO DE CONTROL

Resultados respecto a la variable dependiente: “Aprendizaje procedimental”

Tabla 35

Tabla de frecuencias de la variable dependiente: Aprendizaje procedimental, en el grupo experimental

Nivel	Pretest		Posttest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (16 - 37)	13	76.47%	1	5.88%
Medio (38 - 59)	3	17.65%	10	58.82%
Alto (60 - 80)	1	5.88%	6	35.29%

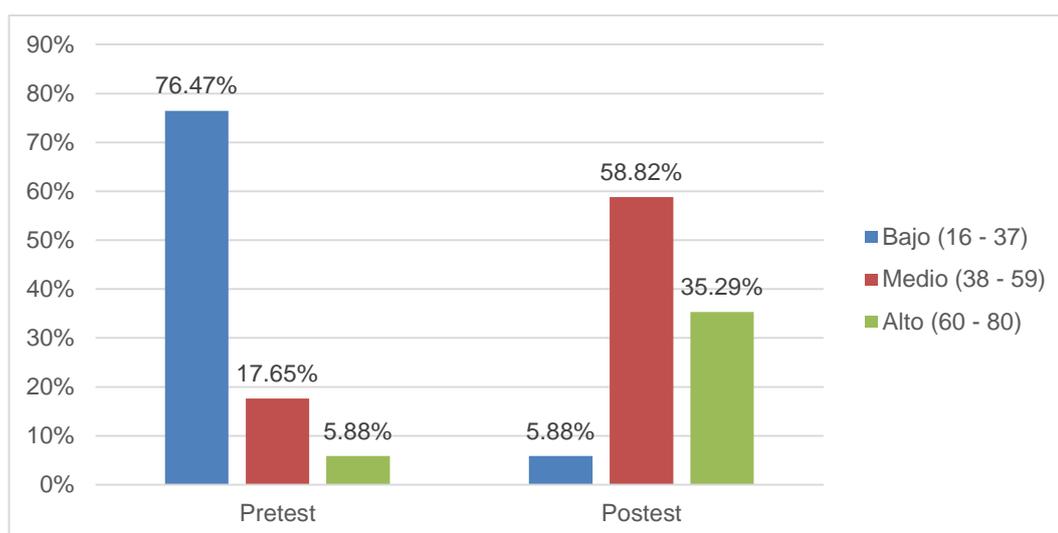


Figura 50. Gráfico de barras de la variable dependiente: Aprendizaje procedimental, en el grupo experimental

De acuerdo con la tabla 10 y la figura 6, se determina lo siguiente:

- En el caso del pretest, el 76.47% de los estudiantes abordados indicaron un nivel bajo de aprendizaje procedimental, mientras que el 17.65% indicaron un nivel medio, y el 5.88% indicaron un nivel alto.
- En el caso del posttest, el 5.88% de los estudiantes abordados indicaron un nivel bajo de aprendizaje procedimental, mientras que el 58.82% indicaron un nivel medio, y el 35.29% indicaron un nivel alto.

Resultados respecto a la dimensión 1: Capacidad de reflexión, entendimiento y planificación

Tabla 36

Tabla de frecuencias de la dimensión 1: Capacidad de reflexión, entendimiento y planificación

Nivel	Pretest		Posttest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	12	70.59%	3	17.65%
Medio (10 - 15)	4	23.53%	9	52.94%
Alto (16 - 20)	1	5.88%	5	29.41%

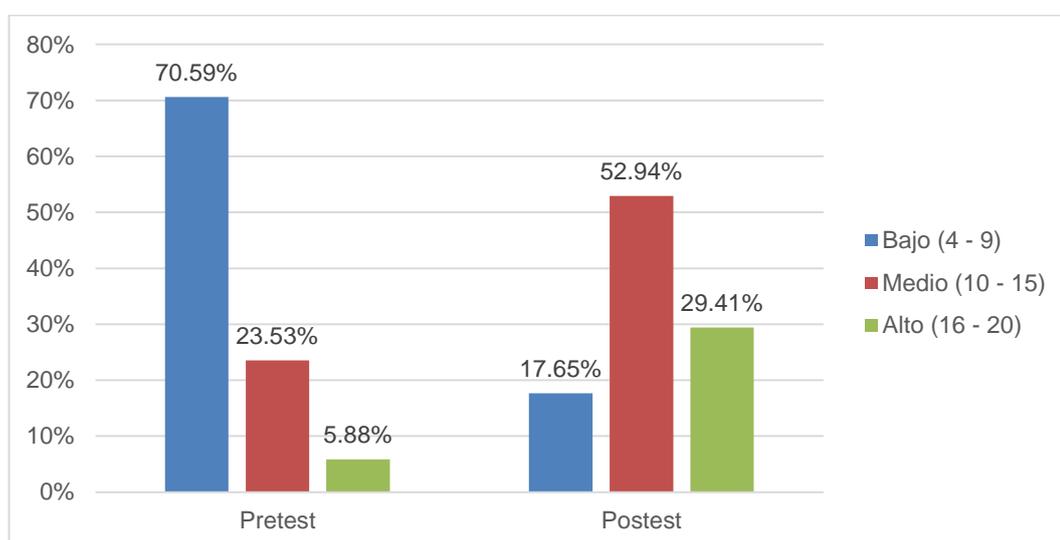


Figura 51. Gráfico de barras de la dimensión 1: Capacidad de reflexión, entendimiento y planificación

De acuerdo con la tabla 11 y la figura 7, se determina lo siguiente:

- En el caso del pretest, el 70.59% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de reflexión, entendimiento y planificación, mientras que el 23.53% indicaron un nivel medio, y el 5.88% indicaron un nivel alto.
- En el caso del posttest, el 17.65% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de reflexión, entendimiento y planificación, mientras que el 52.94% indicaron un nivel medio, y el 29.41% indicaron un nivel alto.

Resultados respecto a la dimensión 2: Capacidad de adaptar un panorama visual

Tabla 37

Tabla de frecuencias de la dimensión 2: Capacidad de adaptar un panorama visual

Nivel	Pretest		Posttest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	13	76.47%	1	5.88%
Medio (10 - 15)	3	17.65%	12	70.59%
Alto (16 - 20)	1	5.88%	4	23.53%

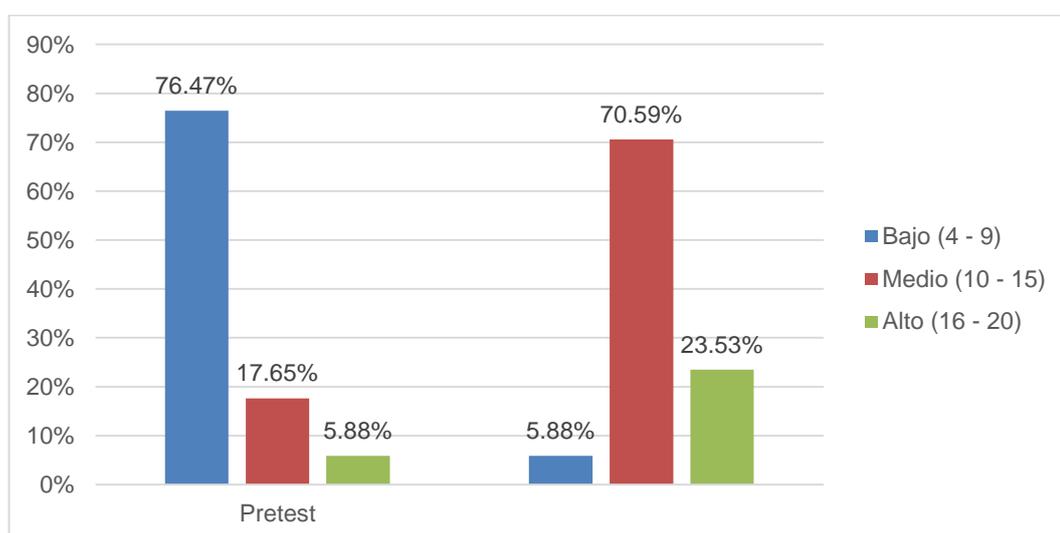


Figura 52. Gráfico de barras de la dimensión 2: Capacidad de adaptar un panorama visual

De acuerdo con la tabla 12 y la figura 8, se determina lo siguiente:

- En el caso del pretest, el 76.47% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de adaptar un panorama visual, mientras que el 17.65% indicaron un nivel medio y el 5.88% indicaron un nivel alto.
- En el caso del posttest, el 5.88% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de adaptar un panorama visual, mientras que el 70.59% indicaron un nivel medio, y el 23.53% indicaron un nivel alto.

Resultados respecto a la dimensión 3: Capacidad de redacción de pseudocódigos

Tabla 38

Tabla de frecuencias de la dimensión 3: Capacidad de redacción de pseudocódigos

Nivel	Pretest		Postest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	12	70.59%	1	5.88%
Medio (10 - 15)	5	29.41%	11	64.71%
Alto (16 - 20)	0	0.00%	5	29.41%

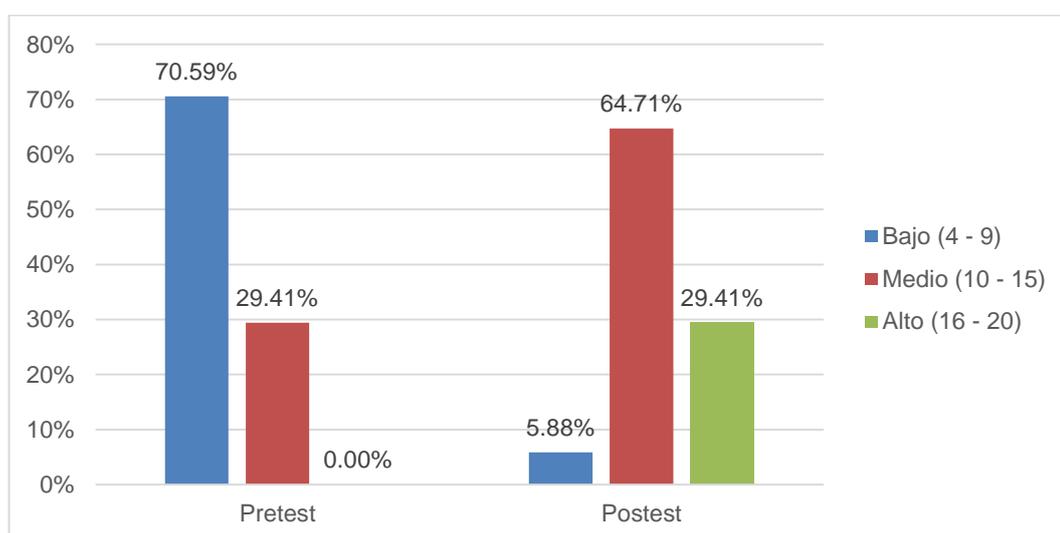


Figura 53. Gráfico de barras de la Dimensión 3: “Capacidad de redacción de pseudocódigos”

De acuerdo con la tabla 13 y la figura 9, se determina lo siguiente:

- En el caso del pretest, el 70.59% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de redacción de pseudocódigos, mientras que el 29.41% indicaron un nivel medio.
- En el caso del postest, el 5.88% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de redacción de pseudocódigos, mientras que el 64.71% indicaron un nivel medio, y el 29.41% indicaron un nivel alto.

Resultados respecto a la dimensión 4: Capacidad de programación

Tabla 39

Tabla de frecuencias de la dimensión 4: Capacidad de programación

Nivel	Pretest		Posttest	
	Frecuencia	Porcentaje	Frecuencia	Porcentaje
Bajo (4 - 9)	14	82.35%	2	11.76%
Medio (10 - 15)	2	11.76%	9	52.94%
Alto (16 - 20)	1	5.88%	6	35.29%

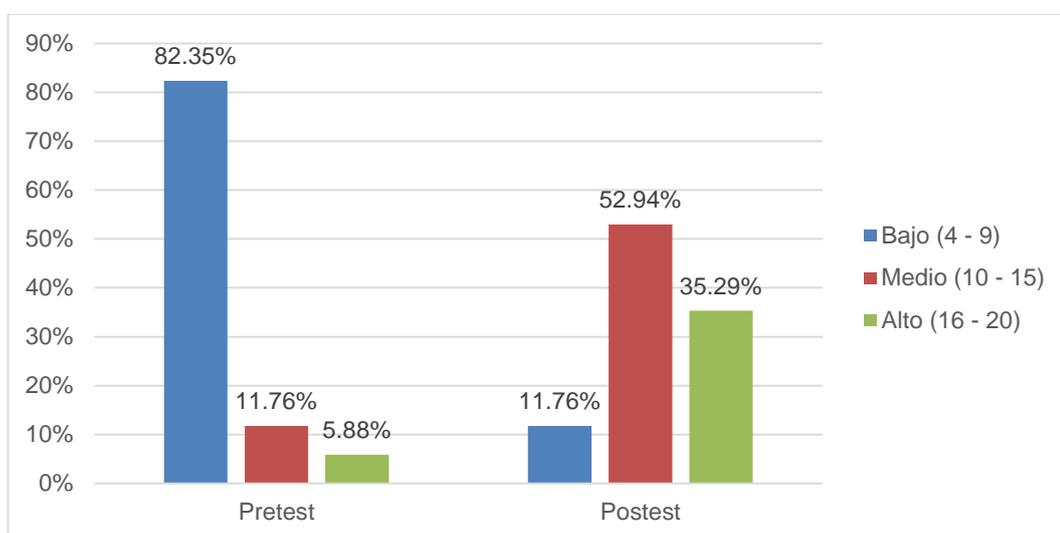


Figura 54. Gráfico de barras de la dimensión 4: Capacidad de programación

De acuerdo con la tabla 14 y la figura 10, se determina lo siguiente:

- En el caso del pretest, el 82.35% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de programación, mientras que el 11.76% indicaron un nivel medio, y el 5.88% indicaron un nivel alto.
- En el caso del posttest, el 11.76% de los estudiantes abordados indicaron un nivel bajo respecto a su capacidad de programación, mientras que el 52.94% indicaron un nivel medio, y el 35.29% indicaron un nivel alto.

5.3 PRUEBAS DE HIPÓTESIS

Para la selección de las pruebas estadísticas necesarias, se consideraron las siguientes:

- **Variable dependiente: Aprendizaje procedimental**
Variable numérica
- **Dimensión 1: Capacidad de reflexión, entendimiento y planificación**
Dimensión numérica
- **Dimensión 2: Capacidad de adaptar un panorama visual**
Dimensión numérica
- **Dimensión 3: Capacidad de redacción de pseudocódigos**
Dimensión numérica
- **Dimensión 4: Capacidad de programación**
Dimensión numérica

Prueba de normalidad

Debido a que la variable dependiente y sus dimensiones fueron numéricas, se realizó una prueba de normalidad, considerando un valor de error inferior al 5% (0.05) para rechazar el supuesto de normalidad. Esta prueba permitió determinar el uso de pruebas paramétricas, o bien pruebas no paramétricas. Además, debido a que la cantidad de estudiantes en los grupos experimental y de control (17) fue menor a 50, se optó por aplicar la prueba de Shapiro - Wilk, obteniéndose los siguientes resultados:

Tabla 40

Resultados de la prueba de normalidad

Variable - dimensión	Grupo	Error	Tipo de distribución
Pretest - Dimensión 1	Experimental	0,240272	Normal
	Control	0,139889	Normal
Pretest - Dimensión 2	Experimental	0,042226	Diferente a la normal
	Control	0,027655	Diferente a la normal
Pretest - Dimensión 3	Experimental	0,057456	Normal
	Control	0,037690	Diferente a la normal
Pretest - Dimensión 4	Experimental	0,208477	Normal
	Control	0,005604	Diferente a la normal
Pretest - Variable dependiente	Experimental	0,159878	Normal
	Control	0,020891	Diferente a la normal
Postest - Dimensión 1	Experimental	0,000078	Diferente a la normal
	Control	0,440880	Normal
Postest - Dimensión 2	Experimental	0,003040	Diferente a la normal
	Control	0,067825	Normal
Postest - Dimensión 3	Experimental	0,001002	Diferente a la normal
	Control	0,022381	Diferente a la normal
Postest - Dimensión 4	Experimental	0,065495	Normal
	Control	0,195283	Normal
Postest - Variable dependiente	Experimental	0,000240	Diferente a la normal
	Control	0,466856	Normal

Los resultados de la tabla 15 fueron usados como guía para la selección de las pruebas estadísticas como se observa en las páginas siguientes.

Prueba de la hipótesis general

El Método Baltex logra mejoras significativas sobre el aprendizaje procedimental de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia en la institución educativa superior Cibertec, semestre 2015-I.

Tabla 41

Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la variable dependiente

Variable	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Aprendizaje procedimental	Experimental	Pretest	Normal	No paramétrica (Wilcoxon)
		Postest	Diferente a la normal	

De acuerdo con la tabla 16, fue necesario aplicar la Prueba de Wilcoxon, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del grupo evaluado. La prueba dio los siguientes resultados:

Tabla 42

Resultados de la Prueba de Wilcoxon respecto a la variable dependiente en el grupo experimental

Tiempo	Media	Diferencia de medias	Error
Pretest	35.53	33.82	0.000292
Postest	69.35		

De acuerdo con la tabla 17, el valor de error calculado (0.000292) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del pretest y postest en el grupo experimental. Es así como este grupo mejoró sus resultados en 33.82 puntos, respecto a lo logrado en el pretest.

Adicionalmente, fue conveniente comparar los resultados finales logrados por el Método Baltex, respecto a la enseñanza tradicional, por lo que se seleccionó una prueba de comparación, se acuerdo a lo siguiente:

Tabla 43

Selección de la prueba estadística para la comparación de muestras independientes respecto a la variable dependiente

Variable	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Aprendizaje procedimental	Experimental	Postest	Diferente a la normal	No paramétrica (U Mann-Whitney)
	Control		Normal	

De acuerdo con la tabla 18, fue necesario aplicar la Prueba U Mann-Whitney, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del postest de los grupos evaluados. La prueba dio los siguientes resultados:

Tabla 44

Resultados de la Prueba U Mann-Whitney respecto a la variable dependiente en el postest

Grupo	Media	Diferencia de medias	Error
Experimental	69.35	16.64	0.000058
Control	52.71		

De acuerdo con la tabla 19, el valor de error calculado (0.000058) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del postest de los grupos experimental y de control. Es así como se puede apreciar que entre estos grupos hay una diferencia de 16.64 puntos, a favor del grupo experimental.

Por tanto, se acepta la hipótesis formulada, resaltando su superioridad en sus resultados, respecto al método tradicional.

Prueba de la hipótesis específica 1

El Método Baltex logra mejoras significativas sobre la capacidad de reflexión, entendimiento y planificación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia en la institución educativa superior Cibertec, semestre 2015-I.

Tabla 45

Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 1

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de reflexión, entendimiento y planificación	Experimental	Pretest	Normal	No paramétrica (Wilcoxon)
		Postest	Diferente a la normal	

De acuerdo con la tabla 20, fue necesario aplicar la Prueba de Wilcoxon, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del grupo evaluado. La prueba dio los siguientes resultados:

Tabla 46

Resultados de la Prueba de Wilcoxon respecto a la Dimensión 1 en el grupo experimental

Tiempo	Media	Diferencia de medias	Error
Pretest	9.00	9.12	0.000286
Postest	18.12		

De acuerdo con la tabla 21, el valor de error calculado (0.000286) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del pretest y postest en el grupo experimental. Es así como este grupo mejoró sus resultados en 9.12 puntos, respecto a lo logrado en el pretest.

Adicionalmente, fue conveniente comparar los resultados finales logrados por el Método Baltex, respecto a la enseñanza tradicional, por lo que se seleccionó una prueba de comparación, de acuerdo con lo siguiente:

Tabla 47

Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 1

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de reflexión, entendimiento y planificación	Experimental	Postest	Diferente a la normal	No paramétrica (U Mann-Whitney)
	Control		Normal	

De acuerdo con la tabla 22, fue necesario aplicar la Prueba U Mann-Whitney, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del postest de los grupos evaluados. La prueba dio los siguientes resultados:

Tabla 48

Resultados de la Prueba U Mann-Whitney respecto a la Dimensión 1 en el postest

Grupo	Media	Diferencia de medias	Error
Experimental	18.12	5.24	0.000204
Control	12.88		

De acuerdo con la tabla 48, el valor de error calculado (0,000204) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del postest de los grupos experimental y de control. Es así como se puede apreciar que entre estos grupos hay una diferencia de 5.24 puntos, a favor del grupo experimental.

Por tanto, se acepta la hipótesis formulada, resaltando su superioridad en sus resultados, respecto al método tradicional.

Prueba de la hipótesis específica 2

El Método Baltex logra mejoras significativas sobre la capacidad de adaptar un panorama visual de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia en la institución educativa superior Cibertec, semestre 2015-I.

Tabla 49

Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 2

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de adaptar un panorama visual	Experimental	Pretest	Diferente a la normal	No paramétrica (Wilcoxon)
		Postest	Diferente a la normal	

De acuerdo con la tabla 24, fue necesario aplicar la Prueba de Wilcoxon, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del grupo evaluado. La prueba dio los siguientes resultados:

Tabla 50

Resultados de la Prueba de Wilcoxon respecto a la Dimensión 2 en el grupo experimental

Tiempo	Media	Diferencia de medias	Error
Pretest	8.29	8.59	0.000281
Postest	16.88		

De acuerdo con la tabla 25, el valor de error calculado (0.000281) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del pretest y postest en el grupo experimental. Es así como este grupo mejoró sus resultados en 8.59 puntos, respecto a lo logrado en el pretest.

Adicionalmente, fue conveniente comparar los resultados finales logrados por el Método Baltex, respecto a la enseñanza tradicional, por lo que se seleccionó una prueba de comparación, de acuerdo con lo siguiente:

Tabla 51

Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 2

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de adaptar un panorama visual	Experimental	Postest	Diferente a la normal	No paramétrica (U Mann-Whitney)
	Control		Normal	

De acuerdo con la tabla 26, fue necesario aplicar la Prueba U Mann-Whitney, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia significativamente diferente entre los resultados del postest de los grupos evaluados. La prueba dio los siguientes resultados:

Tabla 52

Resultados de la Prueba U Mann-Whitney respecto a la Dimensión 2 en el postest

Grupo	Media	Diferencia de medias	Error
Experimental	16.88	4.06	0.000126
Control	12.82		

De acuerdo con la tabla 27, el valor de error calculado (0,000126) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del postest de los grupos experimental y de control. Es así como se puede apreciar que entre estos grupos hay una diferencia de 4.06 puntos, a favor del grupo experimental.

Por tanto, se acepta la hipótesis formulada, resaltando su superioridad en sus resultados, respecto al método tradicional.

Prueba de la hipótesis específica 3

El Método Baltex logra mejoras significativas sobre la capacidad de redacción de pseudocódigos de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia en la institución educativa superior Cibertec, semestre 2015-I.

Tabla 53

Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 3

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de redacción de pseudocódigos	Experimental	Pretest	Normal	No paramétrica (Wilcoxon)
		Postest	Diferente a la normal	

De acuerdo con la tabla 28, fue necesario aplicar la Prueba de Wilcoxon, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia significativamente diferente entre los resultados del grupo evaluado. La prueba dio los siguientes resultados:

Tabla 54

Resultados de la Prueba de Wilcoxon respecto a la Dimensión 3 en el grupo experimental

Tiempo	Media	Diferencia de medias	Error
Pretest	9.12	8.76	0.000347
Postest	17.88		

De acuerdo con la tabla 29, el valor de error calculado (0.000347) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del pretest y postest en el grupo experimental. Es así como este grupo mejoró sus resultados en 8.76 puntos, respecto a lo logrado en el pretest.

Adicionalmente, fue conveniente comparar los resultados finales logrados por el Método Baltex, respecto a la enseñanza tradicional, por lo que se seleccionó una prueba de comparación, de acuerdo con lo siguiente:

Tabla 55

Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 3

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de redacción de pseudocódigos	Experimental	Postest	Diferente a la normal	No paramétrica (U Mann-Whitney)
	Control		Diferente a la normal	

De acuerdo con la tabla 30, fue necesario aplicar la Prueba U Mann-Whitney, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del postest de los grupos evaluados. La prueba dio los siguientes resultados:

Tabla 56

Resultados de la Prueba U Mann-Whitney respecto a la Dimensión 3 en el postest

Grupo	Media	Diferencia de medias	Error
Experimental	17.88	4.70	0.000042
Control	13.18		

De acuerdo con la tabla 31, el valor de error calculado (0.000042) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del postest de los grupos experimental y de control. Es así como se puede apreciar que entre estos grupos hay una diferencia de 4.70 puntos, a favor del grupo experimental.

Por tanto, se acepta la hipótesis formulada, resaltando su superioridad en sus resultados, respecto al método tradicional.

Prueba de la hipótesis específica 4

El Método Baltex logra mejoras significativas sobre la capacidad de programación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia en la institución educativa superior Cibertec, semestre 2015-I.

Tabla 57

Selección de la prueba estadística para la comparación de muestras relacionadas respecto a la Dimensión 4

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de programación	Experimental	Pretest	Normal	Paramétrica (T-Student)
		Posttest	Normal	

De acuerdo con la tabla 32, fue necesario aplicar la Prueba T-Student para muestras relacionadas, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del grupo evaluado. La prueba dio los siguientes resultados:

Tabla 58

Resultados de la Prueba T-Student para muestras relacionadas respecto a la Dimensión 4 en el grupo experimental

Tiempo	Media	Diferencia de medias	Error
Pretest	9.12	7.35	8.1433E-7
Posttest	16.47		

De acuerdo con la tabla 33, el valor de error calculado (8.1433E-7) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del pretest y posttest en el grupo experimental. Es así como este grupo mejoró sus resultados en 7.35 puntos, respecto a lo logrado en el pretest.

Adicionalmente, fue conveniente comparar los resultados finales logrados por el Método Baltex, respecto a la enseñanza tradicional, por lo que se seleccionó una prueba de comparación, de acuerdo con lo siguiente:

Tabla 59

Selección de la prueba estadística para la comparación de muestras independientes respecto a la Dimensión 4

Dimensión	Grupo	Tiempo	Tipo de distribución	Tipo de prueba
Capacidad de programación	Experimental	Postest	Normal	Paramétrica (T-Student)
	Control		Normal	

De acuerdo con la tabla 34, fue necesario aplicar la Prueba T-Student para muestras independientes, considerando un valor de error inferior al 5% (0.05) para asumir una diferencia, significativamente, diferente entre los resultados del postest de los grupos evaluados. La prueba dio los siguientes resultados:

Tabla 60

Resultados de la Prueba T-Student para muestras independientes respecto a la Dimensión 4 en el postest

Grupo	Error calculado con la Prueba de Levene para asumir varianzas diferentes (error inferior al 5% - 0.05)	Media	Diferencia de medias	Error
Experimental	0,590227	16.47	2.65	0.017357
Control	(Se asumen varianzas iguales)	13.82		

De acuerdo con la tabla 35, el valor de error calculado (0.017357) fue menor al establecido (0.05), evidenciando una diferencia significativa entre los resultados del postest de los grupos experimental y de control. Es así como se puede apreciar que entre estos grupos hay una diferencia de 2.65 puntos, a favor del grupo experimental.

Por tanto, se acepta la hipótesis formulada, resaltando su superioridad en sus resultados, respecto al método tradicional.

CAPÍTULO VI : CONCLUSIONES DE LA INVESTIGACIÓN

1. El Método Baltex ha mejorado, significativamente, el aprendizaje procedimental de los estudiantes abordados logrando un incremento de 33.82 puntos.
Además, este método ha demostrado ser superior al tradicional con una mejora de 16.64 puntos, respecto de los resultados finales.
2. El Método Baltex ha mejorado, significativamente, la capacidad de reflexión, entendimiento y planificación de los estudiantes abordados logrando un incremento de 9.12 puntos.
Además, este método ha demostrado ser superior al tradicional con una mejora de 5.24 puntos, respecto de los resultados finales.
3. El Método Baltex ha mejorado, significativamente, la capacidad de adaptar un panorama visual de los estudiantes abordados logrando un incremento de 8.59 puntos.
Además, este método ha demostrado ser superior al tradicional con una mejora de 4.56 puntos, respecto de los resultados finales.
4. El Método Baltex ha mejorado, significativamente, sobre la capacidad de redacción de pseudocódigos de los estudiantes abordados logrando un incremento de 8.76 puntos.
Además, este método ha demostrado ser superior al tradicional con una mejora de 4.70 puntos, respecto de los resultados finales.
5. El Método Baltex ha mejorado, significativamente, la capacidad de programación de los estudiantes abordados logrando un incremento de 7.35 puntos.
Además, este método ha demostrado ser superior al tradicional con una mejora de 2.65 puntos, respecto de los resultados finales.

CAPÍTULO VII: RECOMENDACIONES

1. Continuar con la investigación en futuros escenarios académicos de la misma naturaleza del tópico de estudio.
2. Replicar el uso del Método Baltex en sesiones de aprendizaje o tutorías grupales.
3. Brindar un seguimiento y control en cada tarea propuesta a los estudiantes del curso para garantizar una tutoría eficaz en el aprendizaje del curso de Introducción a la Algoritmia.
4. Generar espacios de capacitación docente y estudiantes del curso con nuevos recursos didácticos y nuevas estrategias metodológicas de los temas inherentes a la asignatura de Introducción a la Algoritmia.

ANEXOS

ANEXO 1: MATRIZ DE CONSISTENCIA

Tabla 61

PROBLEMA	OBJETIVO	HIPÓTESIS	VARIABLES	METODOLOGÍA
<p>Problema general:</p> <p>¿El empleo del Método BALTEX, tiene efectos significativos sobre el aprendizaje procedimental de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?</p> <p>Problemas específicos:</p> <p>¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad de reflexión, entendimiento y planificación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?</p> <p>¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad de adaptar un panorama visual de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?</p> <p>¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad redacción de pseudocódigos de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?</p> <p>¿El empleo del Método BALTEX, tiene efectos significativos sobre la capacidad de programación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I?</p>	<p>Objetivo general:</p> <p>Determinar los efectos del Método BALTEX, sobre el aprendizaje procedimental de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>Objetivos específicos:</p> <p>Determinar los efectos del Método BALTEX, sobre la capacidad de reflexión, entendimiento y planificación estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>Determinar los efectos del Método BALTEX, sobre la capacidad de adaptar un panorama visual de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>Determinar los efectos del Método BALTEX, sobre la capacidad redacción de pseudocódigos de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>Determinar los efectos del Método BALTEX, sobre la capacidad de programación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p>	<p>Hipótesis general:</p> <p>El Método BALTEX logra mejoras significativas sobre el aprendizaje procedimental de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>Hipótesis específicas:</p> <p>El Método BALTEX logra mejoras significativas la capacidad de reflexión, entendimiento y planificación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>El Método BALTEX logra mejoras significativas sobre la capacidad de adaptar un panorama visual de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>El Método BALTEX logra mejoras significativas sobre la capacidad redacción de pseudocódigos de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p> <p>El Método BALTEX logra mejoras significativas sobre la capacidad de programación de los estudiantes del primer ciclo de la asignatura Introducción a la Algoritmia, en la institución educativa superior Cibertec, semestre 2015-I</p>	<p>Variable independiente en el grupo experimental:</p> <p>Método BALTEX</p> <p>Variable independiente en el grupo de control:</p> <p>Método tradicional</p> <p>Variable dependiente:</p> <p>Aprendizaje procedimental</p> <p>Dimensiones:</p> <ul style="list-style-type: none"> • Capacidad de reflexión, entendimiento y planificación. • Capacidad de adaptar un panorama visual. • Capacidad de redacción de pseudocódigos. • Capacidad de programación. 	<p>Enfoque</p> <p>Cuantitativo</p> <p>Alcance</p> <p>Descriptivo y explicativo</p> <p>Diseño</p> <p>Cuasiexperimental</p> <p>Población</p> <p>105 estudiantes del primer ciclo de la asignatura de Introducción a la Algoritmia, de la carrera técnica profesional de Computación e Informática la institución educativa superior Cibertec, semestre académico 2015-I.</p> <p>Muestra</p> <p>Grupo de control: 15 estudiantes</p> <p>Grupo experimental: 15 estudiantes</p> <p>Técnica de recolección de datos</p> <p>Encuesta</p> <p>Instrumento de recolección de datos</p> <p>Cuestionario</p>

CERTIFICADO DE VALIDEZ DEL
 INSTRUMENTO DE MEDICIÓN DE LA VARIABLE INDEPENDIENTE
 "Método tradicional", para el Grupo Control

Marcar con una X, donde proceda la valoración	SI	NO
Planificación		
1. Buen diseño de materiales		
2. Buen diseño de sesiones de aprendizaje		
3. Buen diseño de evaluaciones (pretest y postest)		
Ejecución		
4. Adecuada aplicación del pretest		
5. Buen desarrollo de sesiones de aprendizaje		
6. Buen desarrollo de actividades prácticas		
7. Buen desarrollo de asignaciones		
Evaluación		
8. Adecuada aplicación del postest		
9. Adecuada comparación de resultados		

Valoración

Sí = Cumplió

No = No Cumplió

Observaciones (precisar si hay suficiencia): Sí hay suficiencia

Opinión de aplicabilidad: Aplicable [x] Aplicable después de corregir [] No aplicable []

Nombres y Apellidos del juez validador: José Antonio Cárdenas Martínez

Especialidad del juez validador: Doctor en Educación DNI: 07558536

Fecha: 28 Noviembre del 2017

Firma del experto informante: _____

