EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico de Iztapalapa

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INSTITUTO TECNOLÓGICO DE IZTAPALAPA I

INFORME TÉCNICO DE UN MODELO COMPUTACIONAL

TEMA:

LAMBDA CALCULUS

LUGAR DE REALIZACIÓN:

INSTITUTO TECNOLÓGICO DE IZTAPALAPA I

ASESOR INTERNO:

M.C. ABIEL TOMÁS PARRA HERNÁNDEZ

PRESENTA:

AYALA PALMA JOSÉ DE JESÚS, 181080402	25%
GARCÍA SERRANO MARTÍN, 181080130	25%
PICAZO PÉREZ MAYRA PAOLA, 181080131	25%
VÁZQUEZ ROJAS VERÓNICA, 181080202	25%
Total	100%

CIUDAD DE MÉXICO **JUNIO / 2021**

Índice

Resumen 5
Introducción 6
Objetivos 7
Justificación 8
Capítulo 1 9
Marco teórico 10
a) Teoría de la Computación 10
b) Teoría de Computabilidad 11
c) Modelo de Computación (definición y tipos: secuenciales, funcionales y concurrentes) 13
d) Cálculo Lambda 14
e) Temas y conceptos relevantes para su respectivo modelo de computación 17
Capítulo 2 18
Metodología de trabajo 19
Diferencias entre Metodologías Tradicionales y Ágiles 19
Capítulo 3 23
Metodología de trabajo 24
Relación Modelo – Método 24
Relación matemática 24
Resolución por Método de Reducción 25
Resolución por Método de Gauss Jordan 30
Comprobación en Excel 38
Método de Reducción 38
Método de Gauss Jordan 42
Diagramas de Flujo 44
Diagrama Método de Reducción 44
Diagrama Método de Gauss Jordan 45
Código Método de Reducción 46
Código Método de Gauss Jordan 47
Capítulo 4 48
Resultados 49
Método de Reducción 50

Método Gauss Jordan 53
Software del Método de Reducción 55
Software del Método de Gauss Jordan 57
Implementación del menú 60
Conclusiones 66
Fuentes de información 67
Anexos 69
Bitácora 69
Glosario 71

Índice de Figuras

Figura 3.1 “Ejercicio” 25
Figura 3.2 “Obtención de ecuación 3” 25
Figura 3.3 “Obtención de ecuación 4” 26
Figura 3.4 “Multiplicación de las ecuaciones 3 y 1” 26
Figura 3.5 “Obtención de ecuación 5” 27
Figura 3.6 “Multiplicación de las ecuaciones 4 y 5” 27
Figura 3.7 “Suma de las ecuaciones 4 y 5” 28
Figura 3.8 “Obtención de la incógnita Z” 28
Figura 3.9 “Sustitución en ecuación 4” 29
Figura 3.10 “Sustitución en ecuación 1” 29
Figura 3.11 “Ejercicio Método de Gauss Jordan” 30
Figura 3.12 “Valores de Fila 1” 31
Figura 3.13 “Valores de Fila 2” 32
Figura 3.14 “Valores de Fila 3” 32
Figura 3.15 “Nuevo Matriz Obtenida” 32
Figura 3.16 “Nuevos Valores de Fila 1” 33
Figura 3.17 “Nuevos Valores de Fila 2” 34
Figura 3.18 “Nuevos Valores de Fila 3” 34
Figura 3.19 “Nueva Matriz Obtenida” 34
Figura 3.20 “Nuevos Valores de Fila 1” 35
Figura 3.21 “Nuevos Valores de Fila 2” 35
Figura 3.22 “Nuevos Valores de Fila 3” 36
Figura 3.23 “Matriz Final” 36
Figura 3.24 “Ejercicio Excel” 38
Figura 3.25 “Coeficientes Aislados” 39
Figura 3.26 “Funcionamiento” 39
Figura 3.27 “Operaciones” 40
Figura 3.28 “Excel Gauss Jordan” 41
Figura 3.29 “Funcionamiento” 42
Figura 3.30 “Resultado” 43
Figura 3.31 “Diagrama Algoritmo Reducción” 44
Figura 3.32 “Diagrama Gauss Jordan” 45

Figura 3.33 “Código Método de Reducción” 46
Figura 3.34 “Código Método Gauss Jordan” 47
Figura 4.1 “Resultados” 50
Figura 4.2 “Resultados” 51
Figura 4.3 “Resultados” 52
Figura 4.4 “Resultados” 53
Figura 4.5 “Resultados” 53
Figura 4.6 “Resultados” 54
Figura 4.7 “Resultados” 55
Figura 4.8 “Resultados” 56
Figura 4.9 “Resultados” 57
Figura 4.10 “Resultados” 58
Figura 4.11 “Resultados” 59
Figura 4.12 “Resultados” 60
Figura 4.13 “Resultados” 61
Figura 4.14 “Resultados” 62
Figura 4.15 “Resultados” 63
Figura 4.16 “Resultados” 64
Figura 4.17 “Resultados” 65

Resumen

Esta documentación del proyecto, contendrá los detalles de la comparación de la lógica empleada entre dos modelos computacionales, para resolver un mismo ejercicio, los modelos a emplear son: “Máquina de Turing” y el “Cálculo Lambda”.

Para poder lograr apreciar las semejanzas y/o diferencias, emplearemos un mismo ejercicio matemático, usando los métodos de Reducción y Gauss Jordan para su resolución, pero, este será desarrollado por diversas herramientas algunas tradicionales como cálculos hechos a mano, los cuales serán llevados a una interpretación realizada en Excel y finalmente se plasmarán en código, esté último será desarrollado en el lenguaje de programación **C.**

Este planteamiento es con la única finalidad de demostrar que, sin importar la sintaxis, métodos o herramientas empleadas, el conocimiento bien aplicado puede resolver el mismo problema.

Todos los procesos que se lleven a cabo, estarán detallados paso a paso en este documento, así mismo, contendrá las observaciones que sean percibidas por el equipo y según los diferentes resultados arrojados por cada etapa, recordemos que, en algunos cálculos, en especial cuando se trabajan con decimales, podemos obtener una variación en los resultados, lo cual nos lleva a tener un rango de aceptación.

Por último, indicaremos cómo es posible hacer la manipulación de los datos en Excel, para poder ver el comportamiento de los decimales.

Palabras clave:

C, Cálculo Lambda, Ecuación lineal, Excel, Gauss Jordan y Método de Reducción

Introducción

La información que se albergará en este documento tendrá aspectos de interés según el apartado.

Por parte del equipo SNK, se ha decidido dividir el contenido en cuatro capítulos.

El capítulo uno, describe el motivo de elegir dichos métodos para la resolución de matrices, cómo es que se relacionan los métodos de resolución de matrices con los modelos computacionales y se mencionan los primeros pasos realizados con respecto a las operaciones y pruebas de escritorio, que están elaboradas a mano.

El capítulo dos, se centrará en todas las pruebas realizadas en una computadora, en otras palabras, describe las actividades que se realizaron para poder comprobar nuestros primeros resultados y el proceso para desarrollar los modelos computacionales.

El capítulo tres, describirá los aspectos que se tomaron en cuenta al momento de implementar el código en lenguaje de programación C.

Para concluir, en la sección de resultados se profundizará sobre las discrepancias en resultados (en caso de existir) y se hablará respecto a los errores hallados y las soluciones aplicadas.

Objetivos

OBJETIVOS GENERAL

Desarrollar el método de Reducción y Gauss Jordan empleado los modelos computacionales de la Máquina de Turing y el Cálculo Lambda respectivamente.

OBJETIVOS ESPECÍFICOS

Objetivo Específico 1

Demostrar las semejanzas y diferencias entre ambos modelos matemáticos.

Objetivo Específico 2

Elaborar un programa con el cual se pueda resolver cualquier función por medio del método de Reducción y Gauss Jordan .

Objetivo Específico 3

Demostrar que es posible resolver un mismo problema con dos métodos distintos y obtener el mismo resultado.

Objetivo Específico 4

Observar el comportamiento de los decimales y como generan una variación en los resultados de diferentes pruebas.

Justificación

El requisito central del proyecto es comparar si un problema es computable y comprobar que pueda ser elaborado con dos distintos modelos computacionales, el primero es La máquina de Turing y el segundo era un modelo asignado por nuestro profesor, dicho modelo en este caso es Cálculo Lambda.

Por parte del equipo SNK, elegimos un problema matemático a desarrollar, el cual será un sistema de ecuaciones lineales, desarrollado por medio del método de Reducción y Gauss Jordan.

Esta elección se realiza debido a que al ser operado por Gauss Jordan, una matriz y el resultado de la misma, es necesario iterar tantas veces sea necesario para encontrar lo valores de los coeficientes, aquí es donde nos percatamos que estas iteraciones corresponden a la forma en que resuelven los problemas por “Cálculo Lambda”, además, con esto observamos que la reducción de términos se realiza hasta que los coeficientes dan sus propios resultados a través de las múltiples repeticiones.

Después de una lluvia de ideas decidimos que la mejor manera de poder acceder a la comprobación de ambos métodos; es hacer que el sistema cuente con una interfaz gráfica, en la cual el usuario tendrá la posibilidad de elegir qué tipo de método quiere usar para la resolución de las funciones. Al considerar que no todas las computadoras cuentan con todos los recursos necesarios para ejecutar ciertos softwares, decidimos usar el lenguaje de programación **C**, ya que no necesita de muchos recursos de nuestras computadoras para poder funcionar de forma óptima.

Capítulo 1

Marco teórico

a) Teoría de la Computación

En computación teórica y matemáticas, la teoría de la computación es la rama que se ocupa de qué problemas se pueden resolver en un modelo de computación, utilizando un algoritmo, qué tan eficientemente se pueden resolver o en qué grado (por ejemplo, soluciones aproximadas versus soluciones precisas).

El campo se divide en tres ramas principales: la teoría de autómatas y la teoría de los lenguajes formales, la teoría de la computabilidad y la teoría de la complejidad computacional, que están unidas por la pregunta: "¿Cuáles son las capacidades y limitaciones fundamentales de las computadoras?".[1]

La teoría considera distintos modelos de cómputo, como los autómatas finitos (que son los más sencillos), las máquinas de Turing (que son las computadoras usuales de hoy en día) y las computadoras cuánticas (cuyo funcionamiento no es digital). Las lógicas y los lenguajes formales juegan un rol central en la teoría de la computación porque permiten expresar propiedades de los programas y razonar sobre su comportamiento. La teoría de la computación también se encarga de entender el límite entre los problemas computables y los no-computables y, dentro del mundo de lo computable, clasificarlos de acuerdo a su grado de simpleza o dificultad.

En particular, estudiamos lógicas con buen comportamiento computacional, como las lógicas modales, tanto desde el punto de vista de la teoría de modelos como desde el de la teoría de prueba. Analizamos lenguajes eficientes de consultas que permiten razonar sobre distintas estructuras de representación del conocimiento. A un nivel más abstracto, investigamos las propiedades teóricas de los sistemas de reescritura y los modelos de cómputo fuertes, como el cálculo lambda. A la inversa, estudiamos modelos de cómputo débiles, como los autómatas finitos y sus numerosas variantes. Nos ocupamos también de la noción de aleatoriedad en relación a los distintos modelos de cómputo y a los grados de dificultad de los problemas. Por último, introducimos nociones provenientes de la teoría de funciones computables y de la teoría de la aleatoriedad en el procesamiento cuántico de la información y en algunos modelos computacionales que intentan explicar ciertas características de la cognición humana.[2]

---

1 Wikipedia contributors. (2021, 21 abril). Theory of computation. Wikipedia. https://en.wikipedia.org/wiki/Theory_of_computation#cite_note-Sipser-3rd-1

2 Teoría de la Computación. (s. f.). ICC - Instituto de Ciencias de la Computación. Recuperado 16 de junio de 2021, de https://icc.fcen.uba.ar/teoria-de-la-computacion/

b) Teoría de Computabilidad

La teoría de la computabilidad, también conocida como teoría de la recursividad, es una rama de la lógica matemática, la informática y la teoría de la computación que se originó en la década de 1930 con el estudio de las funciones computables y los grados de Turing. Desde entonces, el campo se ha expandido para incluir el estudio de la computabilidad y la definibilidad generalizadas. En estas áreas, la teoría de la computabilidad se superpone con la teoría de la prueba y la teoría de conjuntos descriptiva efectiva. [3]

Las preguntas básicas abordadas por la teoría de la computabilidad incluyen:

> ¿Qué significa que una función sobre los números naturales sea computable?
>
> ¿Cómo se pueden clasificar las funciones no configurables en una jerarquía en función de su nivel de no competencia?

Aunque hay considerable superposición en términos de conocimiento y métodos, los teóricos matemáticos de la computabilidad estudian la teoría de la computabilidad relativa, las nociones de reducibilidad y las estructuras de grado; aquellos en el campo de las ciencias de la computación se centran en la teoría de las jerarquías subrecursivas, los métodos formales y los lenguajes formales.

Computabilidad de Turing

La principal forma de computabilidad estudiada en la teoría de la computabilidad fue introducida por Turing (1936). Se dice que un conjunto de números naturales es un conjunto computable (también llamado conjunto decidible, recursivo computable de Turing) si hay una máquina de Turing que, dado un número n, se detiene con salida 1 si n está en el conjunto y se detiene con salida 0 si n no está en el conjunto. Una función f de números naturales a números naturales es una función (de Turing) computable, o recursiva si hay una máquina de Turing que, en la entrada n, detiene y devuelve la salida f(n). El uso de máquinas de Turing aquí no es necesario; hay muchos otros modelos de computación que tienen la misma potencia de computación que las máquinas de Turing; por ejemplo, las funciones μ recursivas obtenidas de la recursividad primitiva y el operador μ.

Numeración

Una numeración es una enumeración de funciones; tiene dos parámetros, e y x y genera el valor de la función e-ésima en la numeración en la entrada x. Las numeraciones pueden ser parcialmente computables, aunque algunos de sus miembros son funciones computables totales. Las numeraciones admisibles son aquellas a las que se pueden traducir todas las demás. Una numeración de Friedberg (llamada así por su descubridor) es una numeración uno-uno de todas

---

[3] Wikipedia contributors. (2021, 7 mayo). Computability theory. Wikipedia. https://en.wikipedia.org/wiki/Computability_theory

las funciones parciales computables; no es necesariamente una numeración admisible. Investigaciones posteriores se ocuparon también de las enumeraciones de otras clases como clases de conjuntos computablemente enumérales. Goncharov descubrió, por ejemplo, una clase de conjuntos computablemente enumérales para los cuales las numeraciones caen en exactamente dos clases con respecto a los isomorfismos computables.[4]

[4] Wikipedia contributors. (2021, 7 mayo). Computability theory. Wikipedia. https://en.wikipedia.org/wiki/Computability_theory

c) Modelo de Computación (definición y tipos: secuenciales, funcionales y concurrentes)

Un modelo de computación es un modelo que describe cómo se calcula una salida de una función matemática dada una entrada. Un modelo describe cómo se organizan las unidades de cálculos, memorias y comunicaciones. La complejidad computacional de un algoritmo se puede medir dado un modelo de computación. El uso de un modelo permite estudiar el rendimiento de los algoritmos independientemente de las variaciones que son específicas de implementaciones particulares y tecnología específica. [5]

Los modelos de computación se pueden clasificar en tres categorías: modelos secuenciales, modelos funcionales y modelos simultáneos.

Los modelos secuenciales incluyen:

- Máquinas de estados finitos
- Autómatas pushdown
- Máquinas de acceso aleatorio
- Máquinas de Turing

Los modelos funcionales incluyen:

- Cálculo lambda
- Funciones recursivas generales
- Lógica combinatoria
- Sistemas de reescritura abstracta

Los modelos simultáneos incluyen:

- Autómata celular
- Puertas lógicas y circuitos digitales
- Redes de proceso de Kahn
- Redes de Petri
- Flujo de datos sincrónico
- Redes de interacción
- Modelo de actor

5 Wikipedia contributors. (2021b, junio 13). Model of computation. Wikipedia. https://en.wikipedia.org/wiki/Model_of_computation

d) Cálculo Lambda

El cálculo lambda (también escrito como λ-cálculo)es un sistema formal en lógica matemática para expresar la computación basada en la abstracción de funciones y la aplicación utilizando el enlace y la sustitución de variables. Es un modelo universal de computación que se puede utilizar para simular cualquier máquina de Turing. Fue introducido por el matemático Alonzo Church en la década de 1930 como parte de su investigación sobre los fundamentos de las matemáticas. [6]

Explicación y aplicaciones

El cálculo lambda es Turing completo, es decir, es un modelo universal de computación que se puede utilizar para simular cualquier máquina de Turing. Su homónimo, la letra griega lambda (λ), se utiliza en expresiones lambda y términos lambda para denotar el enlace de una variable en una función.

El cálculo lambda puede ser sin tipo o con tipo. En el cálculo lambda con tipo, las funciones solo se pueden aplicar si son capaces de aceptar el "tipo" de datos de la entrada dada. Los cálculos lambda mecanografiados son más débiles que el cálculo lambda sin tipo, que es el tema principal de este artículo, en el sentido de que los cálculos lambda mecanografiados pueden expresar menos que el cálculo sin tipo, pero por otro lado los cálculos lambda mecanografiados permiten que se prueben más cosas; en el cálculo lambda simplemente tipado es, por ejemplo, un teorema que cada estrategia de evaluación termina para cada término lambda simplemente tipado, mientras que la evaluación de términos lambda sin tipo no necesita terminar. Una razón por la que hay muchos cálculos lambda con tipo diferentes ha sido el deseo de hacer más (de lo que el cálculo sin tipo puede hacer) sin renunciar a ser capaz de probar teoremas fuertes sobre el cálculo.

Reducción

El significado de las expresiones lambda se define por cómo se pueden reducir las expresiones.

Hay tres tipos de reducción:

- **α conversión:** variables enlazadas;
- **β reducción :**aplicar funciones a sus argumentos;
- **η reducción**: que captura una noción de extensionalidad.

También hablamos de las equivalencias resultantes: dos expresiones son *equivalentes α,*si se pueden convertir α en la misma expresión. β-equivalencia y η-equivalencia se definen de manera similar.

[6] Wikipedia contributors. (2021b, junio 13). Lambda calculus. Wikipedia. https://en.wikipedia.org/wiki/Lambda_calculus

El término *redex*, abreviatura de *expresión reducible*, se refiere a los subtérminos que pueden ser reducidos por una de las reglas de reducción. Por ejemplo, (λ*x*. *M) N* es un β-redex en expresar la sustitución de *N* por *x* en *M.* La expresión a la que se reduce un redex se llama su *reducción*; la reducción de (λ*x*. *M*) *N* es *M*[*x* := *N*].

Si *x* no es libre en *M*, λ*x*. *M x* es también un η-redex, con una aducción de *M.*

α conversión

α conversión, a veces conocida como cambio de nombre de α, permite cambiar los nombres de las variables enlazadas. Por ejemplo, α-conversión de λ*x*. *x* podría producir λ*y*. *y*. Los términos que difieren solo por α conversión se denominan *equivalentes α.* Con frecuencia, en los usos del cálculo lambda, los términos equivalentes a α se consideran equivalentes.

Las reglas precisas para la conversión α no son completamente triviales. En primer lugar, cuando α una abstracción, las únicas apariciones de variables cuyo nombre se cambia son las que están enlazadas a la misma abstracción. Por ejemplo, una conversión α de λ*x*.λ*x*. *x* podría dar lugar a λ*y*.λ*x*. *x*, pero *no* pudo dar lugar a λ*y*.λ*x*. *y*. Este último tiene un significado diferente del original. Esto es análogo a la noción de programación de sombreado de variables.

En segundo lugar, la conversión de α no es posible si daría como resultado una variable capturada por una abstracción diferente. Por ejemplo, si sustituimos *x* por *y* en λ*x*.λ*y*. *x*, obtenemos λ*y*.λ*y*. *y*, que no es en absoluto lo mismo.

En lenguajes de programación con ámbito estático, la conversión de α se puede utilizar para simplificar la resolución de nombres al garantizar que ningún nombre de variable enmascara un nombre en un ámbito contenedor (consulte α cambio de nombre para que la resolución de nombres sea trivial).

En la notación de índice de Bruijn, dos términos equivalentes a α cualesquiera son sintácticamente idénticos.

Sustitución

La sustitución, escrita *M*[*V* := *N*], es el proceso de reemplazar todas las apariciones *libres* de la variable *V* en la expresión *M* con la expresión *N*. La sustitución en términos del cálculo lambda se define por recursividad en la estructura de términos, de la siguiente manera (nota: x e y son solo variables, mientras que M y N son cualquier expresión lambda):

$x[x := N] = N$

$y[x := N] = y$, si $x \neq y$

$(M_1\ M_2)[x := N] = (M_1[x := N])\ (M_2[x := N])$

$(\lambda x.\ M)\ [x := N] = \lambda x.\ M$

$(\lambda y.\ M)\ [x := N] = \lambda y.\ (M[x := N])$, si $x \neq y$ e $y \notin \mathrm{FV}(N)$

Para sustituirla en una abstracción, a veces es necesario α convertir la expresión. Por ejemplo, no es correcto para $(\lambda x.\ y)[y := x]$ para dar como resultado $\lambda x.\ x$, porque se suponía que la x sustituida estaba libre, pero terminó siendo atada. La sustitución correcta en este caso es $\lambda z.\ x$, hasta α-equivalencia. La sustitución se define de forma única hasta α-equivalencia.[7]

[7] Wikipedia contributors. (2021b, junio 13). Lambda calculus. Wikipedia. https://en.wikipedia.org/wiki/Lambda_calculus

e) Temas y conceptos relevantes para su respectivo modelo de computación

Eliminación de Gauss-Jordan

“No debe confundirse con Método de Gauss-Seidel.

En álgebra lineal, la eliminación de Gauss-Jordan, es un algoritmo que se usa para determinar la inversa de una matriz y las soluciones de un sistema de ecuaciones lineales. Un sistema de ecuaciones se resuelve por el método de Gauss cuando se obtienen sus soluciones mediante la reducción del sistema dado a otro equivalente en el que cada ecuación tiene una incógnita menos que la anterior. El método de Gauss transforma la matriz de coeficientes en una matriz triangular superior. El método de Gauss-Jordan continúa el proceso de transformación hasta obtener una matriz diagonal.” [8]

Método de Reducción

“Consiste en realizar una relación aritmética entre las Ecuaciones Lineales presentes en el Sistema Lineal.

El objetivo de este proceso es reducir al máximo las variables. Y obtener el valor de al menos una de las incógnitas.

Dicho proceso siempre se consigue mediante la simplificación o reducción de valores. Sin embargo, en algunos casos es necesario multiplicar las Variables para conseguir una reducción.” [9]

[8] Colaboradores de Wikipedia. (2021, 16 junio). Eliminación de Gauss-Jordan. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Eliminaci%C3%B3n_de_Gauss-Jordan

[9] Parrales, H. (2018, 12 noviembre). Método de Reducción. Ecuación Lineal. https://ecuacionlineal.com/lineal/metodo-reduccion/

Capítulo 2

Metodología de trabajo

Diferencias entre Metodologías Tradicionales y Ágiles

Se mostrarán las diferencias según la perspectiva desde la cual se desee observar.

Diferencias generales[10]:

Metodologías Tradicionales	**Metodologías Agiles**
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

[10] Arevalomaria, L. T. L. E. (2017, 21 noviembre). Diferencias entre Metodologías Tradicionales y Ágiles #MetodologiasAgiles. Maria Eugenia Arevalo Lizardo. https://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/

Diferencias por etapas y enfoque metodológico[11]

MODELOS RIGUROSOS	ETAPA	MODELOS AGILES
Planificación predictiva y “aislada”	Análisis de requerimientos	Planificación adaptativa: Entregas frecuentes + colaboración del cliente
	Planificación	
Diseño flexible y Extensible + modelos + Documentación exhaustiva	Diseño	Diseño Simple: Documentación Mínima + Focalizado en la comunicación
Desarrollo individual con Roles y responsabilidades estrictas	Codificación	Transferencia de conocimiento: Programación en pares + conocimiento colectivo

[11] Arevalomaria, L. T. L. E. (2017, 21 noviembre). Diferencias entre Metodologías Tradicionales y Ágiles #MetodologiasAgiles. Maria Eugenia Arevalo Lizardo. https://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/

Actividades de control]: Orientado a los hitos + Gestión mini proyectos	Pruebas	Liderazgo-Colaboración: empoderamiento +autoorganización
	Puesta en Producción	

Diferencias por las características del Proyecto[12]

Modelo de Proceso	**Tamaño del Proceso**	**Tamaño del Equipo**	**Complejidad del Problema**
RUP	Medio / Extenso	Medio / Extenso	Medio / Alto
ICONIX	Pequeño / Medio	Pequeño / Medio	Pequeño / Medio
XP	**Pequeño / Medio**	**Pequeño**	Medio / Alto
SCRUM	**Pequeño / Medio**	**Pequeño**	Medio / Alto

[12] Arevalomaria, L. T. L. E. (2017, 21 noviembre). Diferencias entre Metodologías Tradicionales y Ágiles #MetodologiasAgiles. Maria Eugenia Arevalo Lizardo. https://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/

Metodología SCRUM

Scrum es una metodología ágil de trabajo usada por primera vez en 1986 donde fue aplicada para el desarrollo de productos exitosos como cámaras de fotos de Canon, fotocopiadoras de Xerox, automóviles de Honda, ordenadores de HP y otros, si bien su enfoque en un principio era más bien gerencial en el 2002 Ken Schwaber y Jeff Sutherland, lo documentaron en detalle en el libro "Agile Software Development with Scrum", desde entonces es una metodología muy usada en el mundo del desarrollo de software gracias a lo flexible que puede llegar a ser. Historia de Scrum.

Scrum se especializa en proyectos complejos, donde se necesita obtener resultados en un corto lapso de tiempo, los requisitos pueden llegar a cambiar y la innovación es fundamental.[13]

La metodología Scrum se caracteriza por ser muy adaptable, se conforma de buenas prácticas donde uno de los factores más importantes es el trabajo en equipo, buscando obtener el mejor resultado posible del proyecto.

El ciclo de vida de la metodología Scrum a grandes rasgos se compone de iteraciones de 2 a 4 semanas según el tamaño del proyecto, al final de cada iteración se obtiene un resultado completo, se cuenta con una lista de requisitos priorizada. Gracias a esta forma de trabajo todo sigue avanzando, listo para ser entregado al cliente cuando se necesite sin un máximo esfuerzo.[14]

El avance es constante y el trabajo en equipo es primordial, todos los días se realizan cortas reuniones sincronizadas donde se mantiene al tanto a todos los integrantes del equipo del avance y se da feedback en casa de ser necesario es así cómo es posible mostrarle al cliente un avance constante, además de esto la metodología Scrum es bastante flexible ante posibles cambios.

Es por todas estas características que para el desarrollo de este proyecto se utilizó la metodología Scrum, buscando un buen trabajo en equipo donde todos los miembros tienen un papel importante según sus aptitudes cumpliendo su respectivo rol, se busca tener resultados en un corto lapso de tiempo que nos ayuden a evaluar el avance del proyecto y entregar un avance constantemente, estando preparados para posibles cambios que pudiesen llegar a existir. [15]

---

[13] Historia de Scrum. (2017, 23 marzo). Proyectos Ágiles. https://proyectosagiles.org/historia-de-scrum/
[14] Qué es SCRUM. (2018, 9 octubre). Proyectos Ágiles. https://proyectosagiles.org/que-es-scrum/
[15] Colaboradores de Wikipedia. (2021, 28 mayo). Scrum (desarrollo de software). Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)

Capítulo 3

En la asignatura de ingeniería de software se nos ha enseñado que para desarrollar software debemos pasar por 5 etapas, las cuales son:

- Análisis
- Diseño
- Codificación
- Pruebas
- Implementación

Por ello nuestro interés en el análisis será abarcado del capítulo 1 hasta el capítulo 2.

Metodología de trabajo

Como hemos mencionado, por parte del equipo SNK elegimos la metodología de trabajo SCRUM y con base a ello, decidimos realizar una bitácora donde llevamos registro de la mayor parte de nuestras actividades, dicha bitácora se encontrará en el siguiente apartado Bitácora, sin embargo, se profundizará en la descripción de actividades desarrolladas a lo largo de los capítulos.

Relación Modelo – Método

Como primera instancia el día 27/05/2021 nos dimos a la tarea de generar una lluvia de ideas para poder obtener la propuesta que mejor demostrara la relación entre los modelos computacionales y un problema computable, ahí fue cuando nos percatamos que emplear matrices, nos permitiría, representar de forma adecuada dicha relación, pero aun debíamos averiguar qué métodos serían los compatibles para este proyecto.

Relación matemática

Para hablar de la relación matemática que existe entre los modelos computacionales y los métodos de resolución de matrices, primero debemos describir la similitud entre la máquina de Turing y el método de reducción.

La máquina de Turing, es un modelo realizado a través de una secuencia de pasos ordenados, es decir, no podemos ejecutar la siguiente acción sin haber realizado una previa, lo cual nos llevó a relacionarlos con el método de reducción, en el cual se tienen que realizar una secuencia de sumas para poder encontrar nuestro coeficiente y así nos permita despejarla de una manera óptima y posteriormente sustituir los valores de los coeficientes, hasta poder encontrar el valor final de nuestro sistema de matrices.

Con respecto a Cálculo Lambda, tenemos como referencia que el sistema debe ejecutar un conjunto de iteraciones para poder reducir dicho conjunto y así obtener un resultado, esto se puede demostrar de forma exacta cuando empleamos el método de Gauss Jordan, el cual también realiza iteraciones sobre la matriz que se va a reducir por medio de pivotes, los cuales deben ser 1 y avanzan de forma escalonada, transformando a los demás elementos de la matriz en 0, con esto se sabe el valor de cada coeficiente.

Una vez que comprendimos de forma teórica, la forma en como los modelos computacionales se relacionan, daremos inicio con nuestras primeras pruebas realizadas de forma tradicional, es decir, a lápiz y papel, esto con la finalidad de realizar un ejemplo, el cual fue desarrollado paso por paso para un mejor entendimiento.

Resolución por Método de Reducción

Lo primero es identificar con cuantas ecuaciones contamos y cuales serían sus coeficientes, cabe mencionar que solo haremos uso de los coeficientes para hacer más fácil la comprensión del ejercicio.

Figura 3.1 “Ejercicio”

$$
\begin{array}{lll}
4x + 5y + 6z = 24 & \text{— ec } 1 & 4 \quad 5 \quad 6 \quad 24 \\
x + 2y + 3z = 9 & \text{— ec } 2 & 1 \quad 2 \quad 3 \quad 9 \\
3x + y - 2z = 4 & \text{— ec } 3 & 3 \quad 1 \quad -2 \quad 4
\end{array}
$$

El segundo paso es multiplicar la todas los coeficientes de la ecuación 2 por 2, esto con la finalidad de obtener en el primer coeficiente un valor de cero al momento de realizar la sumatoria.

Figura 3.2 “Obtención de ecuación 3”

Multiplicar la ec 2 por -4

$$-4(1 \quad 2 \quad 3 \quad 9) \Rightarrow -4 \quad -8 \quad -12 \quad -36$$

Figura 3.2 Con esta operación se obtiene la ecuación número 3.

Después de eso podemos empezar a realizar la suma de la ecuación 1 y la ecuación 2. Podemos observar que se obtuvo el 0 sin problemas.

Figura 3.3 “Obtención de ecuación 4”

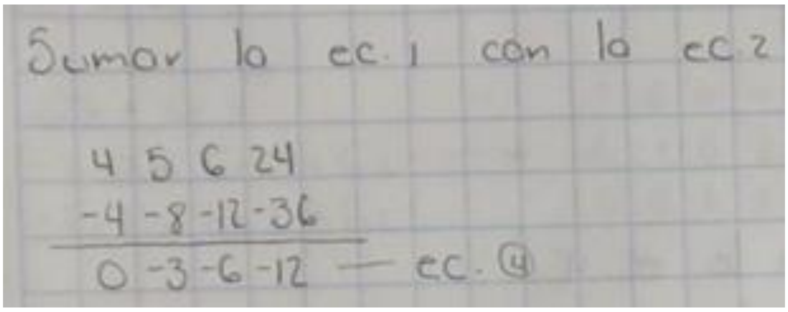


Figura 3.3 Con esta operación se obtiene la ecuación número 4.

Tomando en consideración que el primer coeficiente debe ser reducida a 0, vamos a multiplicar la ecuación 3 por -4 y la ecuación 1 por 3.

Figura 3.4 “Multiplicación de las ecuaciones 3 y 1”

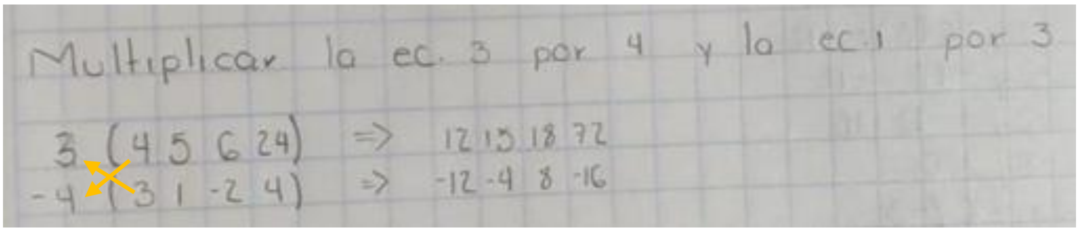


Figura 3.4 como podemos observar son los primeros valores de cada ecuación tienen el mismo valor, pero con signo contrario, es decir, quedan como valores positivo y negativo.

Procedemos a realizar la suma de los coeficientes

Figura 3.5 “Obtención de ecuación 5”

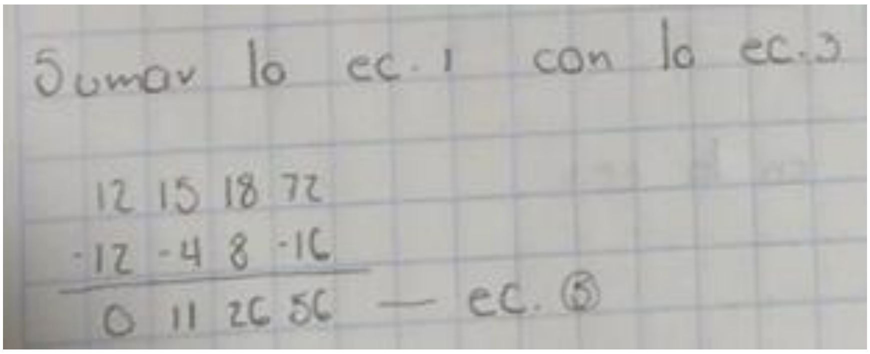


Figura 3.5 Obtenemos 0 en el primer coeficiente y con ello logramos obtener la ecuación número 5.

En este punto repetimos el proceso de multiplicar las ecuaciones por el primer coeficiente correspondiente y acomodando el signo de tal forma que tengamos el mismo valor con signo positivo y negativo.

Figura 3.6 “Multiplicación de las ecuaciones 4 y 5”

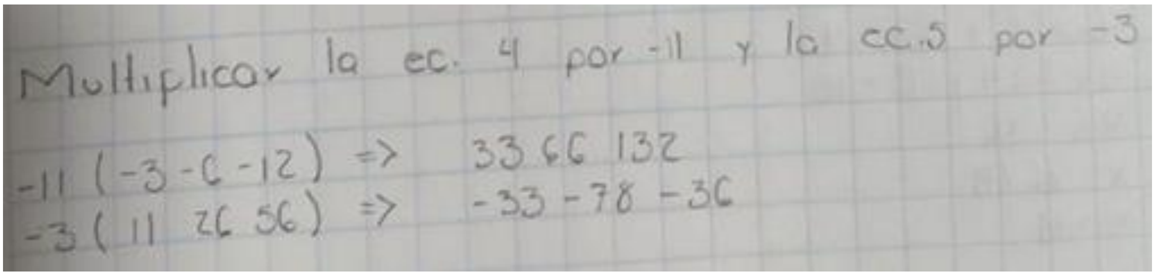
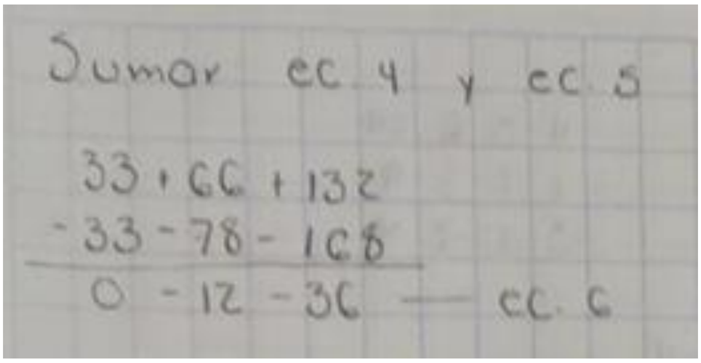


Figura 3.6 Observamos que después de la multiplicación el primer coeficiente sí obtuvo el mismo valor y con signo contrario.

Procedemos a realizar la sumatoria de los nuevos resultados de la ecuación 4 y la ecuación 5.

Figura 3.7 “Suma de las ecuaciones 4 y 5”

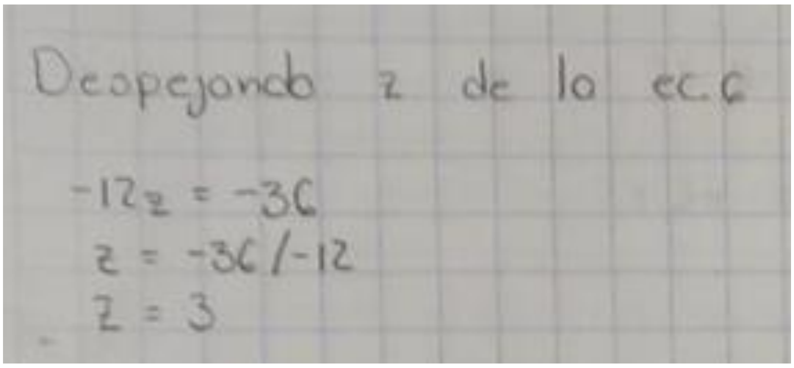

Sumar ec 4 y ec 5

$$\begin{array}{rrr} 33 & +66 & +132 \\ -33 & -78 & -168 \\ \hline 0 & -12 & -36 \end{array} \text{ — ec. 6}$$

Figura 3.7 Apreciamos que se volvió a reducir un término y ahora tenemos una ecuación número 6.

En este punto de la resolución podemos volver a trabajar con los coeficientes e incógnitas, como eliminamos a ***x***, podemos despejar ***z*** de la ecuación número 6.

Figura 3.8 “Obtención de la incógnita Z”


Despejando z de la ec. 6

$$-12z = -36$$
$$z = -36/-12$$
$$z = 3$$

Figura 3.8 Se ha descubierto el valor de ***z***.

Procedemos a realizar la sustitución de ***z*** en la ecuación número 4.

Figura 3.9 “Sustitución en ecuación 4”

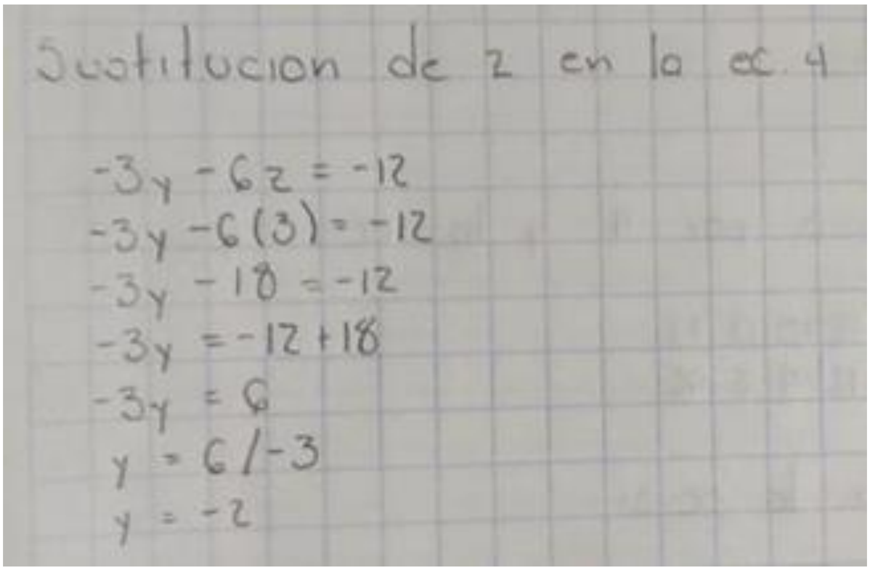


Figura 3.9 Gracias a esta sustitución se obtiene el valor para ***y***.

Con los valores de ***z*** *y* **y**, realizamos la sustitución en la ecuación número 1.

Figura 3.10 “Sustitución en ecuación 1”

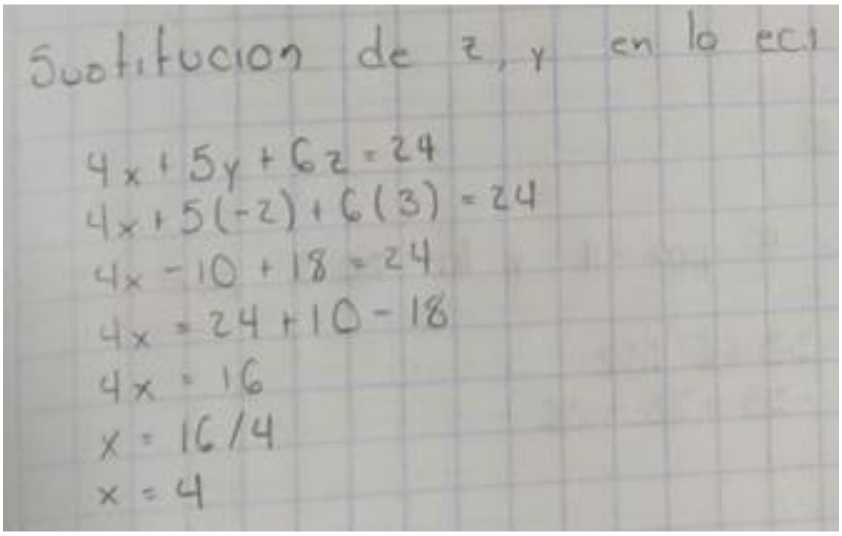


Figura 3.10 Gracias a esta sustitución se obtiene el valor para ***x***.

Los resultados arrojados por el método de reducción fueron los siguientes:

x= 4 y= -2 x= 3

Resolución por Método de Gauss Jordan

Para resolver la matriz haciendo uso de del método de Gauss Jordan, se tomaron en cuenta al inverso multiplicativo que nos va a permitir obtener el pivote (1) de toda la columna y el inverso aditivo, con el cual convertiremos al resto de los elementos de la columna en 0.

Sus respectivas formulas son:

Inverso multiplicativo

$$\left(\frac{1}{x}\right) * x \quad ó \; (\, x^{-1}) * \; x$$

Inverso aditivo

$$-\mathrm{n} \;=\; -1 \;\times\; \mathrm{n}$$

Una vez mencionadas las respectivas fórmulas matemáticas podemos comenzar a resolver nuestro sistema de matrices.

El primer paso es identificar los coeficientes y aislarlos para formar nuestra matriz.

Figura 3.11 “Ejercicio Método de Gauss Jordan”

$$\begin{aligned} 4x + 5y + 6z &= 24 \\ x + 2y + 3z &= 9 \\ 3x + y - 4z &= 4 \end{aligned} \qquad \left|\begin{array}{cccc} 4 & 5 & 6 & 24 \\ 1 & 2 & 3 & 9 \\ 3 & 1 & -4 & 4 \end{array}\right|$$

Iniciamos a realizar el cálculo del inverso multiplicativo con respecto a los valores de la primera columna, se realiza la sustitución en la formula y posteriormente se resuelve la operación.

$$\left(\frac{1}{x}\right) * x$$

$$\left(\frac{1}{4}\right) * 4 = 1$$

Ahora que tenemos nuestro pivote obtendremos nuestro inverso aditivo del resto de elementos, primero se realiza la sustitución en la formula, en ***n*** escribimos el número a convertir en 0 y posteriormente se resuelve la ecuación.

$$-\mathrm{n} \;=\; -1 \;\times\; \mathrm{n} \;\;\text{ó}\;\; \left(-\frac{1}{n}\right)+n$$

$$(-\,1 * 1) + 1 = 0$$

Repetimos el mismo proceso para el siguiente elemento

$$(-\,3 * 1) + 3 = 0$$

Ahora para la fila número 1 vamos a realizar las siguientes operaciones:

$$\left(\frac{1}{4}\right) * 5 = 1.3$$

$$\left(\frac{1}{4}\right) * 6 = 24$$

$$\left(\frac{1}{4}\right) * 24 = 6$$

Figura 3.12 “Valores de Fila 1”

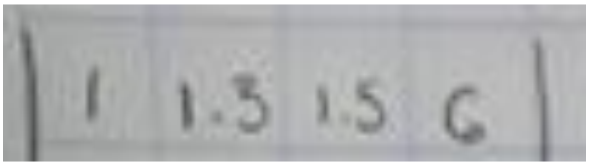


Figura 3.12 Valores de la fila número 1.

Para la fila número 2 vamos a realizar las siguientes operaciones:

$$(-1 * 1.25) + 2 = 0.75$$

$$(-1 * 1.5) + 3 = 1.5$$

$$(-1 * 6) + 9 = 3$$

Figura 3.13 “Valores de Fila 2”

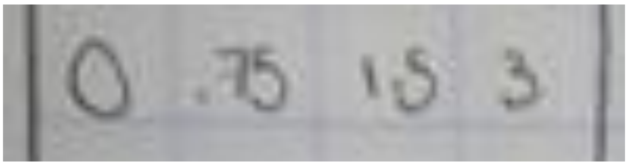


Figura 3.13 Valores de la fila número 2.

Para la fila número 3 vamos a realizar las siguientes operaciones:

$$(-3 * 1.25) + 1 = -2.75$$

$$(-3 * 1.5) + (-2) = -6.5$$

$$(-3 * 6) + 4 = -14$$

Figura 3.14 “Valores de Fila 3”

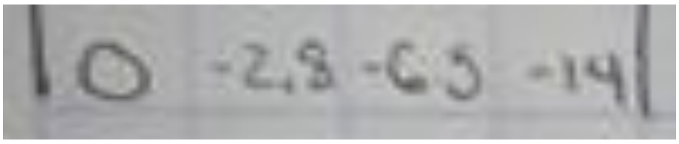


Figura 3.14 Valores de la fila número 3.

Figura 3.15 “Nuevo Matriz Obtenida”

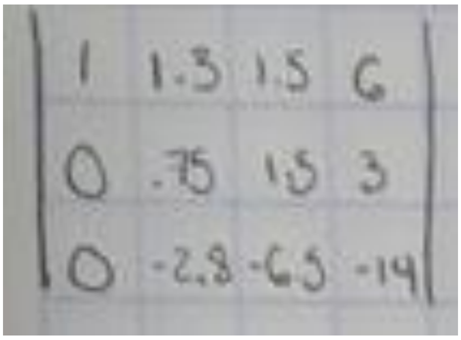


Figura 3.15 Vista de la nueva matriz obtenida.

Ahora vamos a realizar el cálculo del inverso multiplicativo con respecto a los valores de la segunda columna, se realiza la sustitución en la formula y posteriormente se resuelve la operación.

$$\left(\frac{1}{x}\right) * x$$

$$\left(\frac{1}{0.75}\right) * .075 = 1$$

Tenemos nuestro pivote obtendremos nuestro inverso aditivo del resto de elementos, primero se realiza la sustitución en la formula, en ***n*** escribimos el número a convertir en 0 y posteriormente se resuelve la ecuación.

$$-\mathrm{n} \;=\; -1 \;\times\; \mathrm{n} \;\text{ó}\; \left(-\frac{1}{n}\right) + n$$

$$(-\,1.25 * 1) + 1.25 = 0$$

Repetimos el mismo proceso para el siguiente elemento:

$$-(-2.75\ * 1) + (-2.75) = 0$$

Ahora para la fila número 1 vamos a realizar las siguientes operaciones:

$$(-1.25\ * 2) + 1.5 = -1$$

$$(-1.25\ * 4) + 6 = 1$$

Figura 3.16 “Nuevos Valores de Fila 1”

Figura 3.16 Nuevos valores de la fila número 1.

Para la fila número 2 vamos a realizar las siguientes operaciones:

$$\left(\frac{1}{0.75}\right) * 1.5 = 2$$

$$\left(\frac{1}{0.75}\right) * 3 = 4$$

Figura 3.17 “Nuevos Valores de Fila 2”

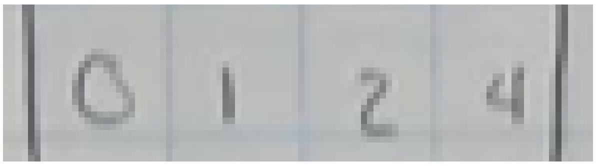


Figura 3.17 Nuevos valores de la fila número 2.

Para la fila número 3 vamos a realizar las siguientes operaciones:

$$-(-\,2.75 * 2) + (-6.5) = -1$$

$$-(-\,2.75 * 4) + (-14) = -3$$

Figura 3.18 “Nuevos Valores de Fila 3”

Figura 3.18 Nuevos valores de la fila número 3.

Figura 3.19 “Nueva Matriz Obtenida”

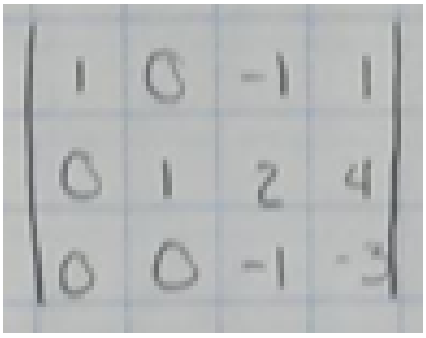


Figura 3.19 Vista de la nueva matriz obtenida.

De nuevo realizamos el cálculo del inverso multiplicativo con respecto a los valores de la tercera columna, se realiza la sustitución en la formula y posteriormente se resuelve la operación.

$$\left(\frac{1}{x}\right) * x$$

$$\left(\frac{1}{-1}\right) * -1 = 1$$

Ahora que tenemos nuestro pivote obtendremos nuestro inverso aditivo del resto de elementos, primero se realiza la sustitución en la formula, en ***n*** escribimos el número a convertir en 0 y posteriormente se resuelve la ecuación.

$$-\mathrm{n} \ = \ -1 \ \times \ \mathrm{n} \ \text{ó} \ \left(-\frac{1}{n}\right) + n$$

$$-(-\,1 * 1) + (-1) = 0$$

Repetimos el mismo proceso para el siguiente elemento:

$$(-2 * 1) + 2 = 0$$

Ahora para la fila número 1 vamos a realizar la siguiente operación:

$$-(-1\ * 3) + 1 = 4$$

Figura 3.20 “Nuevos Valores de Fila 1”

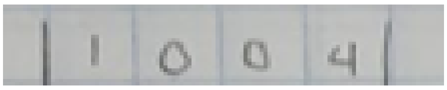


Figura 3.20 Nuevo valor para el último elemento de la fila número 1.

Ahora para la fila número 2 vamos a realizar la siguiente operación:

$$(-2\ * 3) + 4 = -2$$

Figura 3.21 “Nuevos Valores de Fila 2”

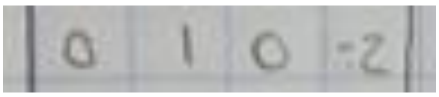


Figura 3.21 Nuevo valor para el último elemento de la fila número 2.

Ahora para la fila número 3 vamos a realizar la siguiente operación:

$$\left(\frac{1}{-1}\right) * -3$$

Figura 3.22 “Nuevos Valores de Fila 3”

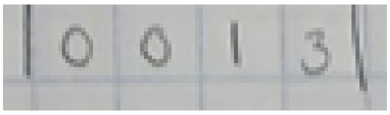


Figura 3.22 Nuevo valor para el último elemento de la fila número 3.

Figura 3.23 “Matriz Final”

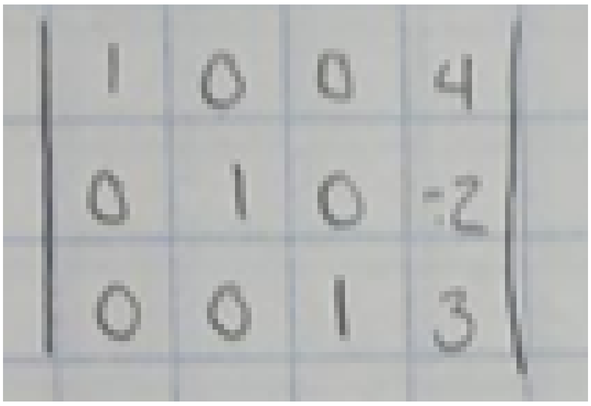


Figura 3.23 Se muestra el resultado final de la matriz.

Los resultados arrojados por el método de Gauss Jordan fueron los siguientes:

x= 4 y= -2 x= 3

Como nos podemos dar cuenta la resolución se convierte en algo repetitivo y al finalizar nuestros resultados son los mismos que en el método de reducción.

Demos analizar el proceso y resultados de esta ecuación de una manera abstracta para poder comprender la relación de los modelos, como se demostró para resolver una matriz por el método de reducción, se tuvieron que seguir una serie de pasos ordenados y esto puede llegar a ser muy repetitivo y tedioso, sin embargo, el método de Gauss Jordan es una forma de optimizar, además es relevante mencionar que, aunque se utilicen ecuaciones, se pueden realizar operaciones más básicas (sin necesidad de despeje), lo que permite ejecutar la simplificación, dicha simplificación no lleva a pensar en la recursividad o en otras palabras en el Cálculo Lambda.

Retomando algunas palabras de cuando iniciamos la ingeniería, en algún momento se nos dijo, que, si bien un programa puede ser elaborado usando 10 líneas de código, la recursividad nos permite reducirlas en 3, esto nos sirve como analogía para poder hablar de los modelos matemáticos, ya que cuando se tiene una base para poder resolver una operación, el proceso puede ser largo y tedioso, sin embargo, una vez que la base está bien definida, las optimizaciones se abren paso para poder facilitar los procesos y por ende la forma de computar los problemas matemáticos.

Comprobación en Excel

Como parte de nuestras buenas prácticas de programación, antes de empezar a codificar, creamos una prueba de escritorio en nuestra herramienta de trabajo Excel, esto para verificar que las fórmulas empleadas y cálculos fueran correctos, además de reducir los errores humanos que pudieran comprometer los resultados.

Método de Reducción

Iniciamos construyendo el Excel que lleva por nombre "Reducción"[16].

Como primer paso incluimos a modo de imagen la ecuación sobre la cual estaríamos trabajando:

Figura 3.24 "Ejercicio Excel"

	A	B	C	D
1				
2	Solucion a un sistema de ecuaciones lineales			
3				
4	Metodo de reduccion para una matriz de 3x3			
5				
6		$x - y - 4z = -4$		
7		$2x + 2y + z = 11$		
8		$x + y + 3z = 13$		
9				

Figura 3.24 Se muestra la ubicación de las ecuaciones para tener una consulta más accesible.

[16] EMMANUEL ASESORÍAS. (2019, 4 septiembre). Método de REDUCCIÓN. Sistema de Ecuaciones de 3x3 [Vídeo]. YouTube. https://www.youtube.com/watch?v=aW4v3ROkTek

Posterior a ello se realiza la extracción de los coeficientes para construir nuestra matriz:

Figura 3.25 "Coeficientes Aislados"

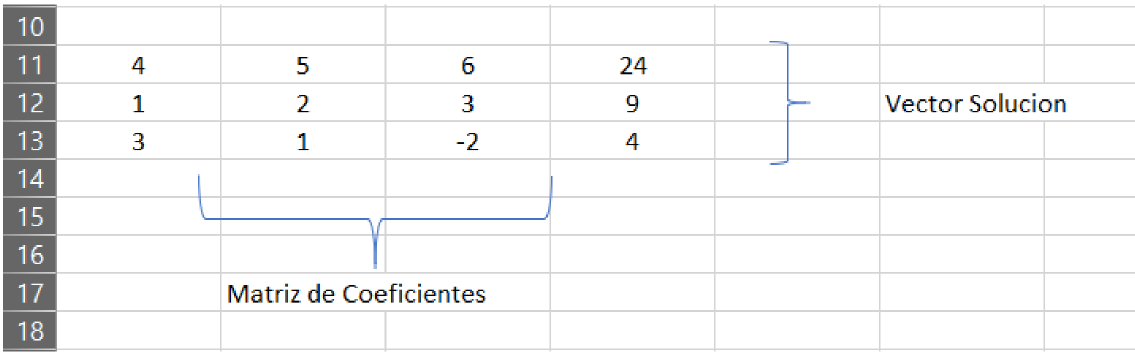


Figura 3.25 Muestra los coeficientes aislados.

Para que este archivo pueda ser utilizado en la resolución de cualquier ecuación lineal, se decidió crearlo dinámico, por ello en un siguiente apartado dentro del mismo libro se utilizó la sentencia **"=A#"**, donde A representa la columna y el # representa la posición dentro de la columna, cabe mencionar que se colocaron las letras del abecedario que correspondientes a cada columna.

La forma en la que funciona es la siguiente:

Del lado derecho observamos que se tienen los mismos valores de los coeficientes, pero al dar clic en esos valores dentro de la casilla nos marca **"=A11"**, esto quiere decir que el valor que este colocado en dicha posición tiene que ser escrito del lado derecho.

Figura 3.26 "Funcionamiento"

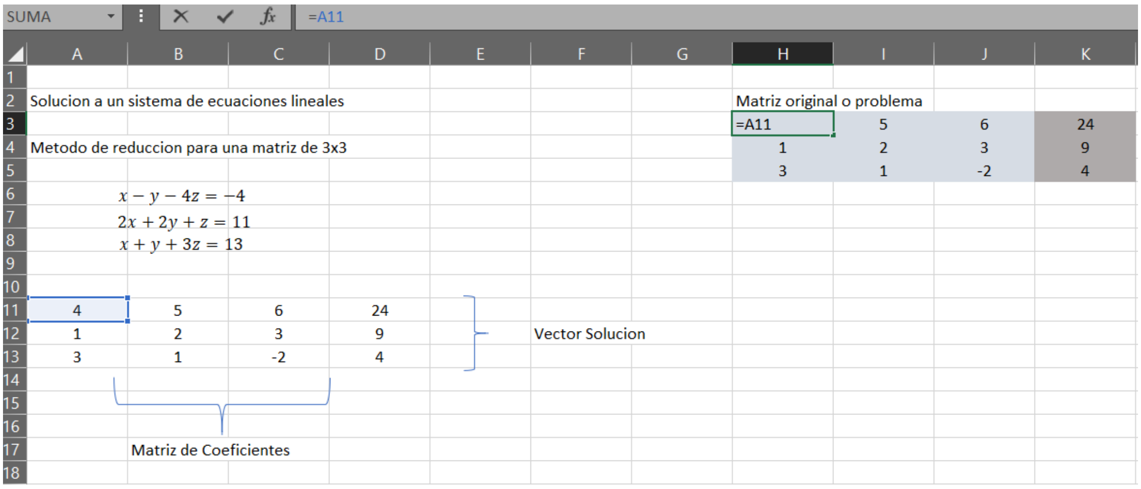


Figura 3.26 Imagen representativa del funcionamiento para recuperar valores.

Para el desarrollo de la multiplicación y suma de ecuaciones se utilizaron las fórmulas de Excel para que en un mismo paso se realizaran las operaciones de multiplicación y suma, las cuales fueron desarrolladas acorde al proceso realizado a lápiz y papel.

Figura 3.27 “Operaciones”

P30

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Solucion a un sistema de ecuaciones lineales							Matriz original o problema				
3								4	5	6	24	Ecuacion 1
4	Metodo de reduccion para una matriz de 3x3							1	2	3	9	Ecuacion 2
5								3	1	-2	4	Ecuacion 3
6		$x - y - 4z = -4$										
7		$2x + 2y + z = 11$						Primer paso (restar ecuacion 1 y ecuacion 2)				
8		$x + y + 3z = 13$						4	5	6	24	
9								-4	-8	-12	-36	
10								0	-3	-6	-12	Ecuacion 4
11	4	5	6	24								
12	1	2	3	9		Vector Solucion		Segundo paso (restar ecuacion 1 y ecuacion 3)				
13	3	1	-2	4				12	15	18	72	
14								-12	-4	8	-16	
15								0	11	26	56	Ecuacion 5
16												
17		Matriz de Coeficientes						Tercer paso (restar ecuacion 4 y ecuacion 5)				
18								0	33	66	132	
19								0	-33	-78	-168	
20								0	0	-12	-36	
21												
22								Cuarto paso (sustitucion de z en la ecuacion 4)				
23								0	-3	-18	-12	
24												
25								Quinto paso (sustitucion de z, y en la ecuacion 1)				
26								4	-10	18	24	
27												

Figura 3.27 Muestra los resultados de las operaciones aritméticas.

Por último, tomamos en cuenta las siguientes operaciones para obtener el valor de las incógnitas:

$$\left(\frac{-36}{-12}\right) = 3$$

$$\left(\frac{(-12 - (-18))}{-3}\right) = -2$$

$$\left(\frac{(24 - 18 - (-10))}{4}\right) = 4$$

Como podemos apreciar los resultados son correspondientes a los calculados a mano.

El siguiente documento de Excel a desarrollar fue el que corresponde a Gauss Jordan[17].

Al igual que en el documento del método de reducción, el primer paso fue incluir a manera de imagen la ecuación sobre la cual estaríamos trabajando:

Figura 3.28 "Excel Gauss Jordan"

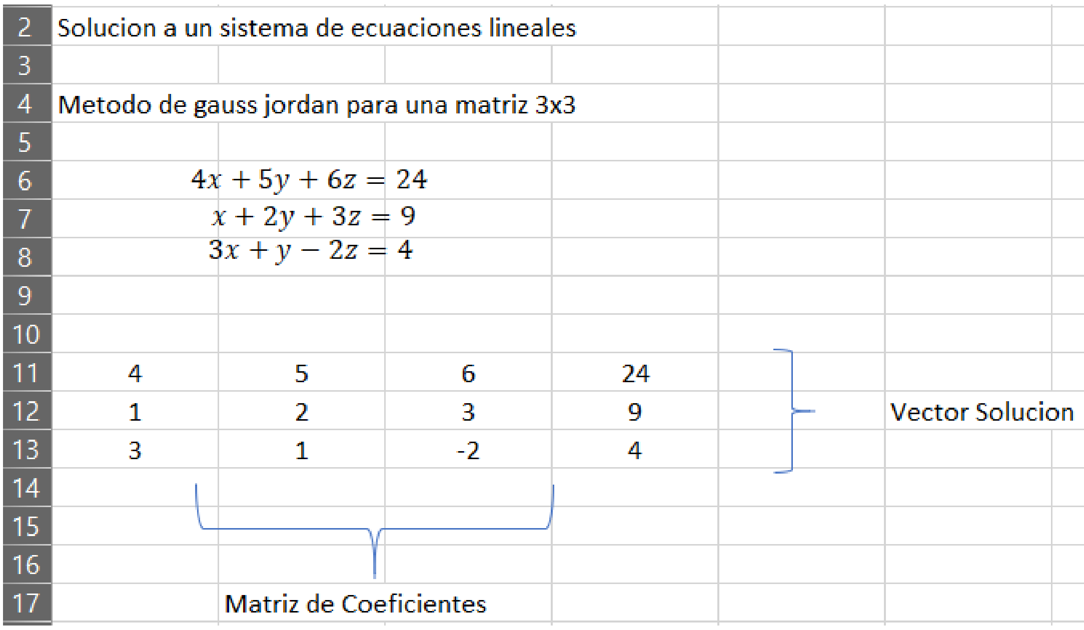


Figura 3.28 Muestra los coeficientes aislados.

[17] Kennya Martinez. (2020, 20 marzo). Reducción Escalonada MAtriz 3x3 Excel [Vídeo]. YouTube. https://www.youtube.com/watch?v=zO2bKRJAeg4

Método de Gauss Jordan

La creación de este archivo de igual manera fue pensada para resolver de forma automática otras ecuaciones, por ello también es dinámico y sigue los mismos principios de la sentencia **“=A#”**, donde A representa la columna y el # representa la posición dentro de la columna, cabe mencionar que se colocaron las letras del abecedario que correspondientes a cada columna.

La forma en la que funciona es la siguiente:

Del lado derecho observamos que se tienen los mismos valores de los coeficientes, pero al dar clic en esos valores dentro de la casilla nos marca **“=A11”**, esto quiere decir que el valor que este colocado en dicha posición tiene que ser escrito del lado derecho.

Figura 3.29 “Funcionamiento”

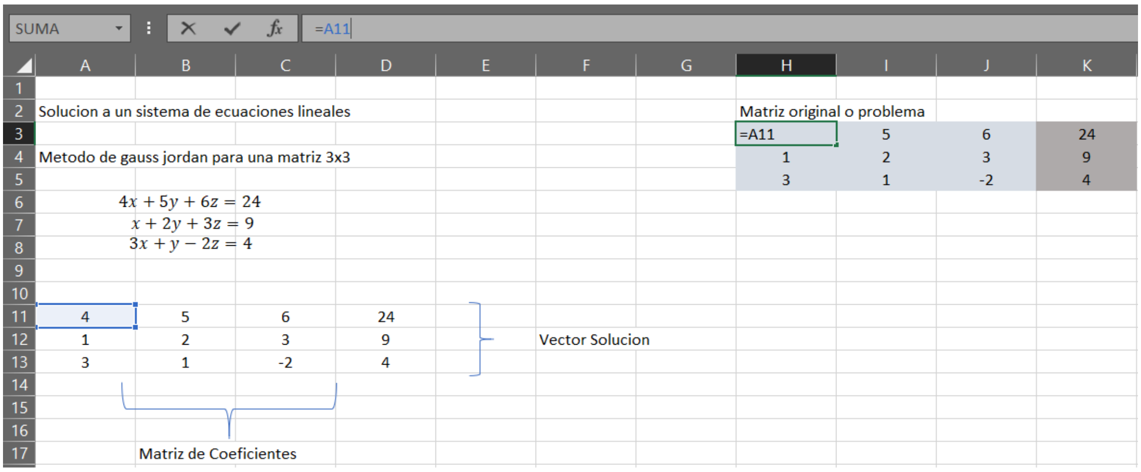


Figura 3.29 Imagen representativa del funcionamiento para recuperar valores.

El desarrollo del ejercicio se llevó a cabo usando las fórmulas del inverso aditivo y multiplicativo, según era el caso, recordemos que la explicación detalla de cada paso fue explicada en el capítulo 1, por ello solo anexaremos la evidencia de cuál fue el resultado obtenido.

Figura 3.30 “Resultado”

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2	Solucion a un sistema de ecuaciones lineales							Matriz original o problema					Operaciones autorizadas		
3								4	5	6	24				
4	Metodo de gauss jordan para una matriz 3x3							1	2	3	9		Inv. Multiplicativo * por el que sera 1		
5								3	1	-2	4		=	(1/$H$3)*H3	
6		$4x+5y+6z=24$													
7		$x+2y+3z=9$						Reduccion primera columna					Inv. Aditivo + el que sera 0		
8		$3x+y-2z=4$						1	1.25	1.5	6		=	(-$H$4*H8)+H4	
9								0	0.75	1.5	3				
10								0	-2.75	-6.5	-14		Incognitas		
11	4	5	6	24									x=	4	
12	1	2	3	9		Vector Solucion		Reduccion segunda columna					y=	-2	
13	3	1	-2	4				1	0	-1	1		z=	3	
14								0	1	2	4				
15								0	0	-1	-3				
16															
17		Matriz de Coeficientes						Reduccion tercera columna							
18								1	0	0	4				
19								0	1	0	-2				
20								0	0	1	3				
21															

Figura 3.30 Muestra el resultado de la matriz final.

En este capítulo demostramos la importancia de tener conocimientos para poder emplear distintas herramientas, ya que pueden fungir como un buen complemento para las actividades a realizar como programadores, además, nos permite enriquecer las pruebas a las que nos sometemos como ingenieros, otro aspecto a destacar es el hecho que al realizar este tipo de pruebas les permite a personas que tengan pocos o nulos conocimientos sobre el tema, comprendan los aspecto principales por medio de la interacción.

Basados en nuestra experiencia como computólogos en formación, hemos aprendido que lo más importante a la hora de diseñar software es dedicar la mayor parte de tiempo y esfuerzo al análisis, por tal motivo hemos empleado aproximadamente el 80% del desarrollo del proyecto a esta etapa, ya que como bien sabemos primero debemos comprender los aspectos teóricos para poder brindar un ejemplo y con razón poder construir dichos ejemplos de forma digital.

Como siguiente paso una vez que realizamos nuestras pruebas de escritorio, elaboramos dos diagramas de flujo, uno por cada método de resolución de matrices, dichos diagramas tenían como principal función ser una guía para poder consultar en caso de presentar trabas en cuanto a la lógica empleada al momento de desarrollar el código en C.

Diagramas de Flujo

Diagrama Método de Reducción

Figura 3.31 “Diagrama Algoritmo Reducción”

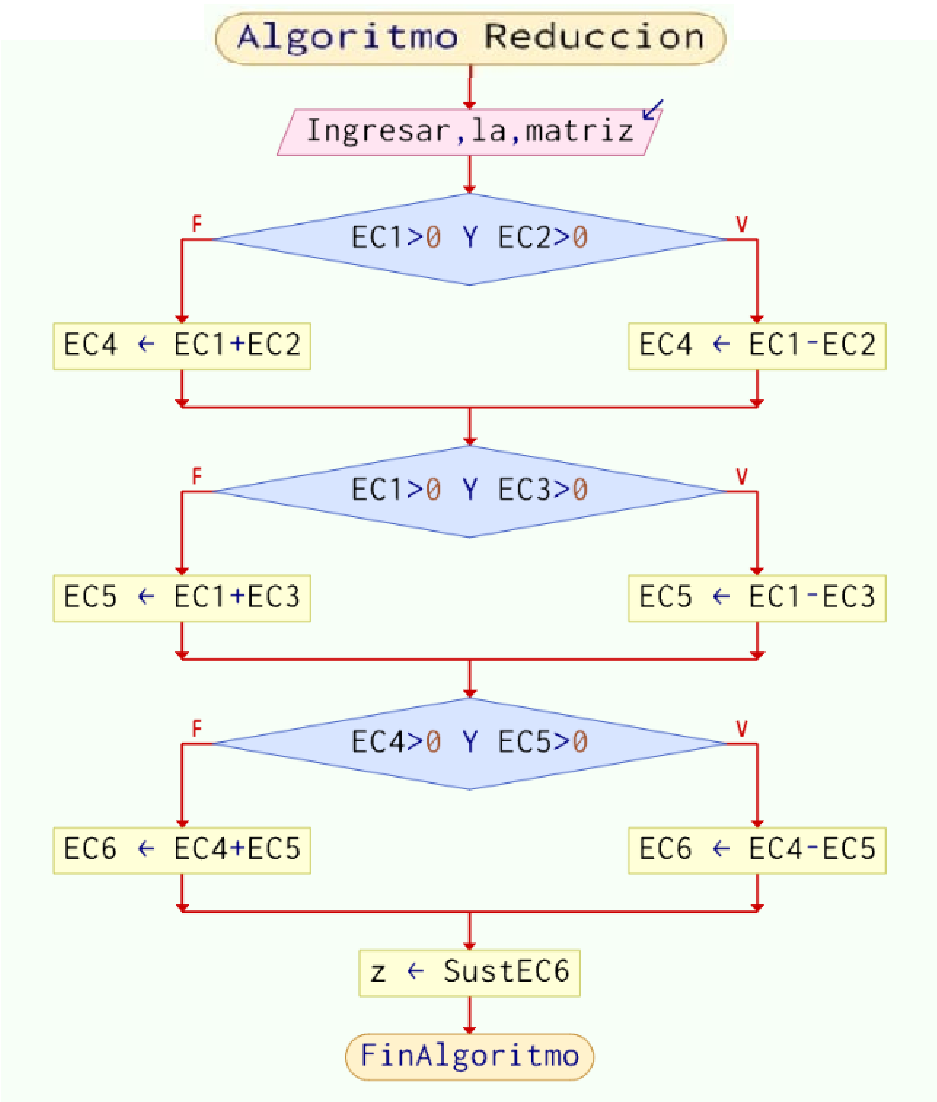


Diagrama Método de Gauss Jordan

Figura 3.32 “Diagrama Gauss Jordan”

Como segunda fase, iniciamos con las ideas para el diseño, por decisión unánime se le mostrará al usuario un menú en el cual pueda decidir con cuál de los dos métodos desea interactuar, pero para construirlo, primero se trabajaron ambos métodos como programas independientes, esto con la finalidad de tener una mejor organización y control.

Dentro de este capítulo se trabajará el diseño de consola (default) que proporciona el entorno IDE, sin embargo, se afinarán los detalles de presentación en la sección de pruebas. Una vez aclarado los puntos de diseño, también daremos inicio con el punto tres del diseño de software, el cual habla de la codificación y los últimos dos puntos (pruebas e implementación) serán explicados a detalle en el apartado de resultados.

Código Método de Reducción

El código empleado, fue desarrollado en cuarto semestre en la materia de Métodos Numéricos con él ingeniero Rene Tocohua, sin embargo, se realizaron modificaciones debido a que el código proporcionaba una resolución para otro sistema de ecuaciones.

Figura 3.33 “Código Método de Reducción”

Reduccion.cpp

```
#include <iostream>
#include <stdio.h>
#include <cstdlib>
#include <math.h>

/* run this program using the console pauser or add your own getch, system("pause") or input loop */
using namespace std; // entrada y salida
 // entrada y salida
int main(int argc, char** argv) {

    system("color 1f");
    float a11, a12, a13, a21, a22, a23, a31, a32, a33, b1,b2,b3;
    float A11, A12, A13, A21, A22, A23, A31, A32, A33, B1,B2,B3;
    float C1, C2, C3, Z1, Z2 ,Z3 , R1, R2, R3;
    float D11, D12, D13, D14, D21, D22, D23, D24 ;
    float X1,X2,X3,X4;
    float z, x, y;
    float E11, E12, E13, E14;
    float F11,F12,F13,F14;




    printf("\n\n\n\t *******************");
    printf("\n\n\n\t PROGRAMA QUE RESUELVE UN SISTEMA DE ECUACIONES LINEALES ");
    printf("\n\n\n\t DE 5x5 POR EL METODO DE GAUSS JORDAN");
    printf("\n\n\n\t a11x + a12y + a13z = b1");
    printf("\n\n\n\t a21x + a22y + a23z = b2");
    printf("\n\n\n\t a31x + a32y + a33z = b3");
    printf("\n\n\n\t *******************");

    do{
    printf("\n\n\n\t ** PARA LA ECUACION 1 **");
    printf("\n\n\n\t INGRESE EL VALOR DE a11: ");
    scanf("%f",&a11);
    printf("\n\n\n\t INGRESE EL VALOR DE a12: ");
    scanf("%f",&a12);
    printf("\n\n\n\t INGRESE EL VALOR DE a13: ");
    scanf("%f",&a13);
    printf("\n\n\t INGRESE EL VALOR DE b1: ");
    scanf("%f",&b1);
}
while (fabs(a11) > fabs(a12) + fabs(a13));
```

Figura 3.33 Fragmento de código del programa “Reducción.cpp”.

Código Método de Gauss Jordan

Con la intensión de optimizar tiempos, tras una investigación obtuvimos de internet el código que resuelve una matriz por el método de Gauss Jordan[18], cabe destacar que dicho código nos permite decidir el tamaño de la matriz.

Figura 3.34 “Código Método Gauss Jordan”

Reduccion.cpp [*] Gaussjordan.cpp

```
#include <iostream>

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

int main(int argc, char** argv) {
     int i,j,k,n; // declare variables and matrixes as
    float a[10][10],b,x[10];
    printf("\nEnter the size of matrix: ");
    scanf("%d",&n);
    printf("\nEnter the elements of augmented matrix (rowwise):\ n");
    for(i=1; i<=n; i++) {
        for(j=1; j<=(n+1); j++) {
            cout << "A[" << i << ", " << j << " ]=";
            cin >> a[i][j];
        }
    }
    //to find the elements of diagonal matrix
    for(j=1; j<=n; j++) {
        for(i=1; i<=n; i++) {
            if(i!=j) {
                b=a[i][j]/a[j][j];
                for(k=1; k<=n+1; k++) {
                    a[i][k]=a[i][k]-b*a[j][k];
                }
            }
        }
    }
    cout<<"\nThe solution is:\n";
    for(i=1; i<=n; i++) {
        x[i]=a[i][n+1]/a[i][i];
        cout<<"x"<<i << "="<<x[i]<<" ";
     return 0;
}
```

Figura 3.34 Programa “GaussJordan.cpp”, tomado de la red.

[18] C++ Program to Implement Gauss Jordan Elimination. (s. f.). Tutorialspoint. Recuperado 21 de junio de 2021, de https://www.tutorialspoint.com/cplusplus-program-to-implement-gauss-jordan-elimination

Capítulo 4

Resultados

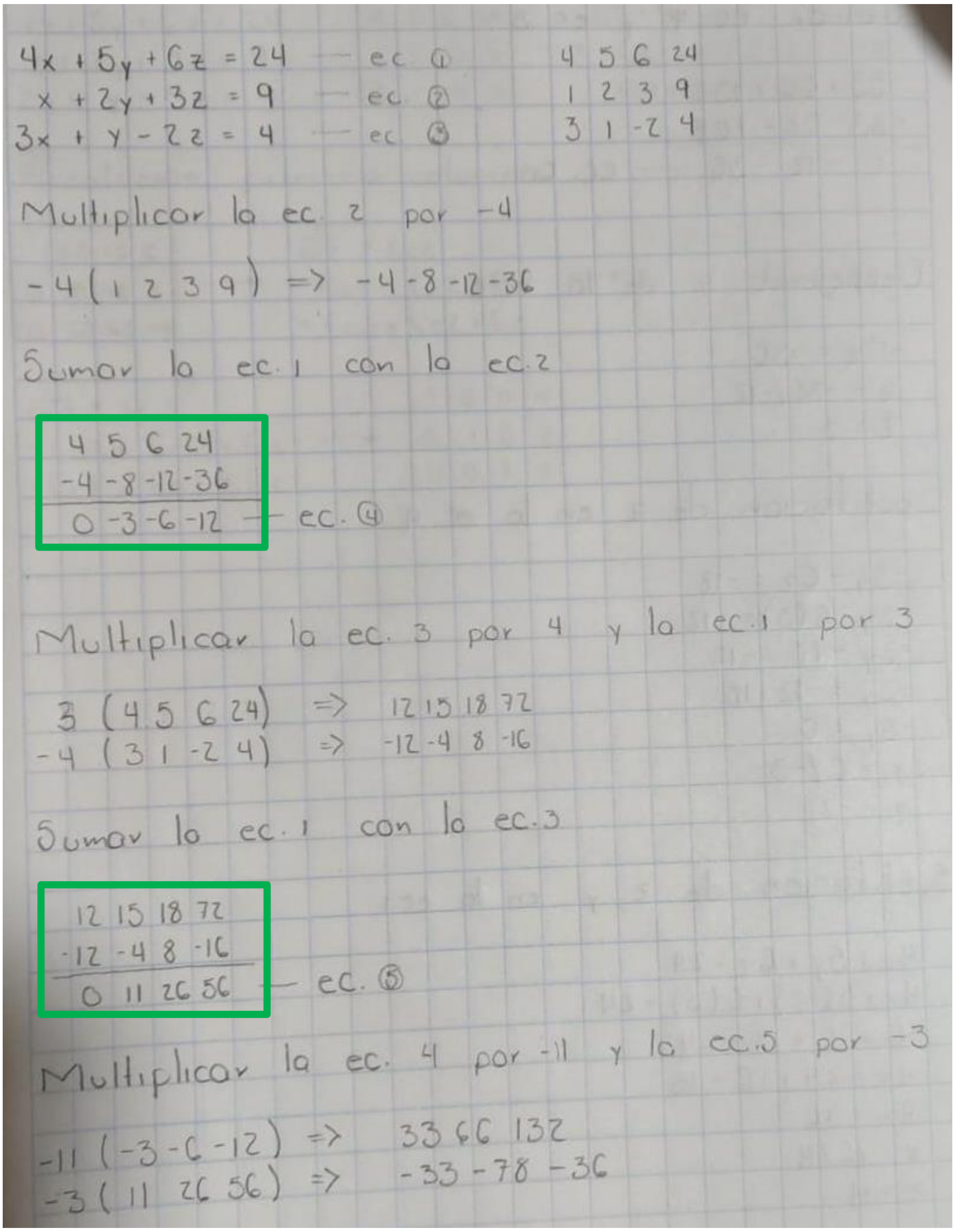
Como primera instancia debemos rectificar que los resultados sean congruentes en lo que respecta a los resultados obtenido a mano y los cálculos de Excel.

A modo de identificador se utilizarán figuras de colores para encerrar los resultados iguales, los cuales se identificarán con un rectángulo de color verde en los casos que serán acertados y en caso de existir una discrepancia que no afecte el resultado se marcará con un círculo u ovalo de color naranja, para los casos que genere un error se marcara con un rectángulo azul.

Es importante realizar este tipo de comparaciones, ya que, basados en nuestra experiencia al confirmar que el cálculo desarrollado en Excel es correcto, facilita la comparación en el código e incluso permite semejar la cantidad de decimales.

Método de Reducción

Figura 4.1 “Resultados”



$4x + 5y + 6z = 24$ — ec ① 4 5 6 24
$x + 2y + 3z = 9$ — ec ② 1 2 3 9
$3x + y - 2z = 4$ — ec ③ 3 1 -2 4

Multiplicar la ec 2 por -4

-4 (1 2 3 9) => -4 -8 -12 -36

Sumar la ec. 1 con la ec. 2

4 5 6 24
-4 -8 -12 -36
0 -3 -6 -12 — ec. ④

Multiplicar la ec. 3 por 4 y la ec. 1 por 3

3 (4 5 6 24) => 12 15 18 72
-4 (3 1 -2 4) => -12 -4 8 -16

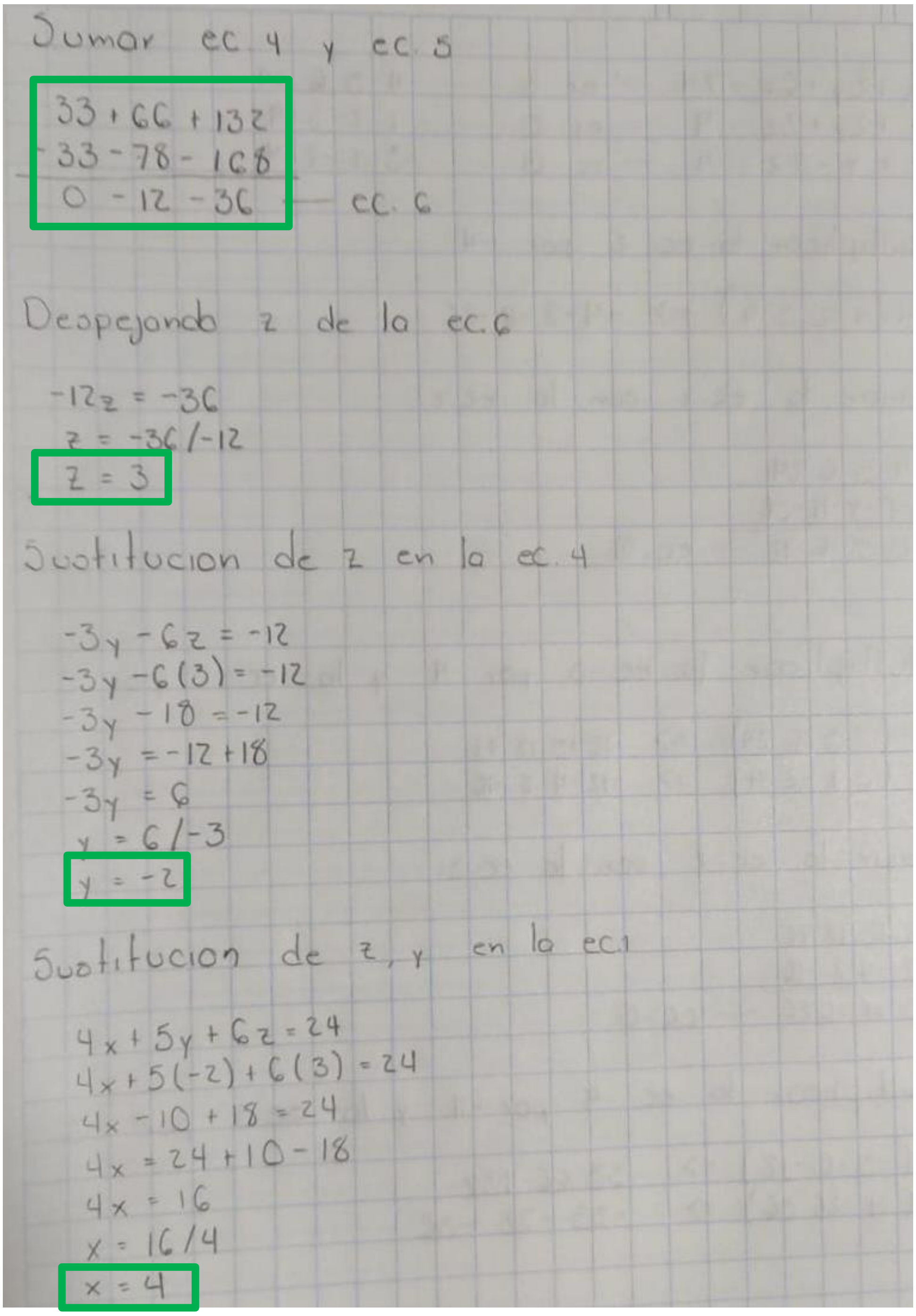
Sumar la ec. 1 con la ec. 3

12 15 18 72
-12 -4 8 -16
0 11 26 56 — ec. ⑤

Multiplicar la ec. 4 por -11 y la ec. 5 por -3

-11 (-3 -6 -12) => 33 66 132
-3 (11 26 56) => -33 -78 -36

Figura 4.2 “Resultados”



Sumar ec 4 y ec 5

$33 + 66 + 132$
$33 - 78 - 168$
$0 - 12 - 36$ ec. 6

Despejando z de la ec.6

$-12z = -36$
$z = -36/-12$
$z = 3$

Sustitucion de z en la ec. 4

$-3y - 6z = -12$
$-3y - 6(3) = -12$
$-3y - 18 = -12$
$-3y = -12 + 18$
$-3y = 6$
$y = 6/-3$
$y = -2$

Sustitucion de z, y en la ec.1

$4x + 5y + 6z = 24$
$4x + 5(-2) + 6(3) = 24$
$4x - 10 + 18 = 24$
$4x = 24 + 10 - 18$
$4x = 16$
$x = 16/4$
$x = 4$

Figura 4.3 “Resultados”

Matriz original o problema						Incognitas	
4	5	6	24	Ecuacion 1		z=	3
1	2	3	9	Ecuacion 2		y=	-2
3	1	-2	4	Ecuacion 3		x=	4
Primer paso (restar ecuacion 1 y ecuacion 2)							
4	5	6	24				
-4	-8	-12	-36				
0	-3	-6	-12	Ecuacion 4			
Segundo paso (restar ecuacion 1 y ecuacion 3)							
12	15	18	72				
-12	-4	8	-16				
0	11	26	56	Ecuacion 5			
Tercer paso (restar ecuacion 4 y ecuacion 5)							
0	33	66	132				
0	-33	-78	-168				
0	0	-12	-36				
Cuarto paso (sustitucion de z en la ecuacion 4)							
0	-3	-18	-12				
Quinto paso (sustitucion de z, y en la ecuacion 1)							
4	-10	18	24				

Haciendo las observaciones y comparaciones pertinentes, podemos decir que el Método de Reducción cumple con los parámetros y no se detectó error alguno.

Método Gauss Jordan

Figura 4.4 “Resultados”

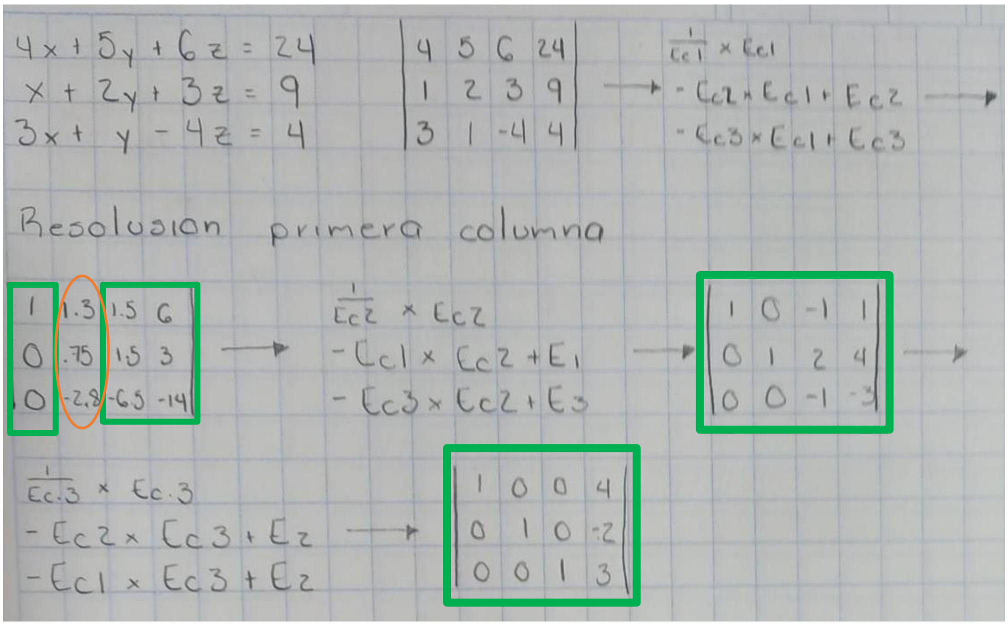


Figura 4.5 “Resultados”

Matriz original o problema			
4	5	6	24
1	2	3	9
3	1	-2	4
Reduccion primera columna			
1	1.25	1.5	6
0	0.75	1.5	3
0	-2.75	-6.5	-14
Reduccion segunda columna			
1	0	-1	1
0	1	2	4
0	0	-1	-3
Reduccion tercera columna			
1	0	0	4
0	1	0	-2
0	0	1	3

Operaciones autorizadas

Inv. Multiplicativo * por el que sera 1
= (1/$H$3)*H3

Inv. Aditivo + el que sera 0
= (-$H$4*H8)+H4

Incognitas	
x=	4
y=	-2
z=	3

Gracias a la identificación por colores, se denota a simple vista el hecho que, en la primera matriz calculada, la segunda columna tiene valores distintos. A modo de prueba se ingresan los valores de nuestros cálculos realizados a mano y este fue el resultado:

Figura 4.6 “Resultados”

H	I	J	K	L	M	N	O
Matriz original o problema					Operaciones autorizadas		
4	5	6	24				
1	2	3	9		Inv. Multiplicativo * por el que sera 1		
3	1	-2	4		=	(1/$H$3)*H3	
Reduccion primera columna					Inv. Aditivo + el que sera 0		
1	1.3	1.5	6		=	(-$H$4*H8)+H4	
0	0.75	1.5	3				
0	-2.8	-6.5	-14		Incognitas		
					x=	4.22222222	
Reduccion segunda columna					y=	-2.22222222	
1	0	-1.1	0.8		z=	3.11111111	
0	1	2	4				
0	0	-0.9	-2.8				
Reduccion tercera columna							
1	0	0	4.22222222				
0	1	0	-2.22222222				
0	0	1	3.11111111				

Como se aprecia los resultados sí tienen variación y aunque nuestros enteros siguen siendo 4, -2 y 3, el hecho de tener decimales ve comprometidos los resultados y en una suposición en la cual los cálculos realizados se emplearan en una cortadora industrial, significaría una pérdida de material, debido a que las piezas ya no serían simétricas y por ende el proceso de ensamblado ya no sería acorde a las necesidades.

Esta es una muestra de la importancia de tener un rango de aceptación para los decimales, sin embargo, por la ecuación que desarrollamos, los resultados deben ser enteros y esto lo comprobamos haciendo la comparación con el método de reducción.

A continuación, se muestran las pruebas que se aplicaron en los softwares, cabe mencionar, que en esta sección se hablara de las etapas de codificación, pruebas e implementación.

Software del Método de Reducción

Cuando se ejecuta el programa y solicita los valores de la matriz, se ve de esta manera:

Figura 4.7 "Resultados"

```
********************

PROGRAMA QUE RESUELVE UN SISTEMA DE ECUACIONES LINEALES

DE 5x5 POR EL METODO DE GAUSS JORDAN

a11x + a12y + a13z = b1

a21x + a22y + a23z = b2

a31x + a32y + a33z = b3

*******************

** PARA LA ECUACION 1 **

INGRESE EL VALOR DE a11: 4


INGRESE EL VALOR DE a12: 5


INGRESE EL VALOR DE a13: 6

INGRESE EL VALOR DE b1: 24
```

La muestra de resultados lo realiza de la siguiente forma:

Figura 4.8 “Resultados”

```
Primer paso (restar ecuacion 1 y ecuacion 2)
4.00x 5.00y 6.00z =24.00
-4.00x -8.00y -12.00z =-36.00
-----------------------------
0.00x +-3.00y +-6.00z =-12.00


Segundo paso (restar ecuacion 1 y ecuacion 3)
12.00x 15.00y 18.00z =72.00
-12.00x -4.00y 8.00z =-16.00
-----------------------------
0.00x 11.00y 26.00z =56.00


Tercer paso (restar ecuacion 4 y ecuacion 5)
-0.00x 33.00y 66.00z =132.00
-0.00x -33.00y -78.00z =-168.00
-----------------------------
-0.00x 0.00y -12.00z =-36.00


Cuarto paso (sustitucion de z en la ecuacion 4)
0.00x -3.00y -18.00z =-12.00


Quinto paso (sustitucion de z, y en la ecuacion 1)
4.00x -10.00y 18.00z =24.00


Nuestras incognitas son
3.00z=
-2.00y=
4.00x=
```

Observamos que en pantalla aparecen las operaciones que se realizaron para hallar los valores de ***x, y*** y ***z***.

Debido a que en ese momento nuestro principal interés era probar la funcionalidad, existieron algunos detalles en diseño, los cuales fueron corregidos al momento de incorporar el programa en el menú.

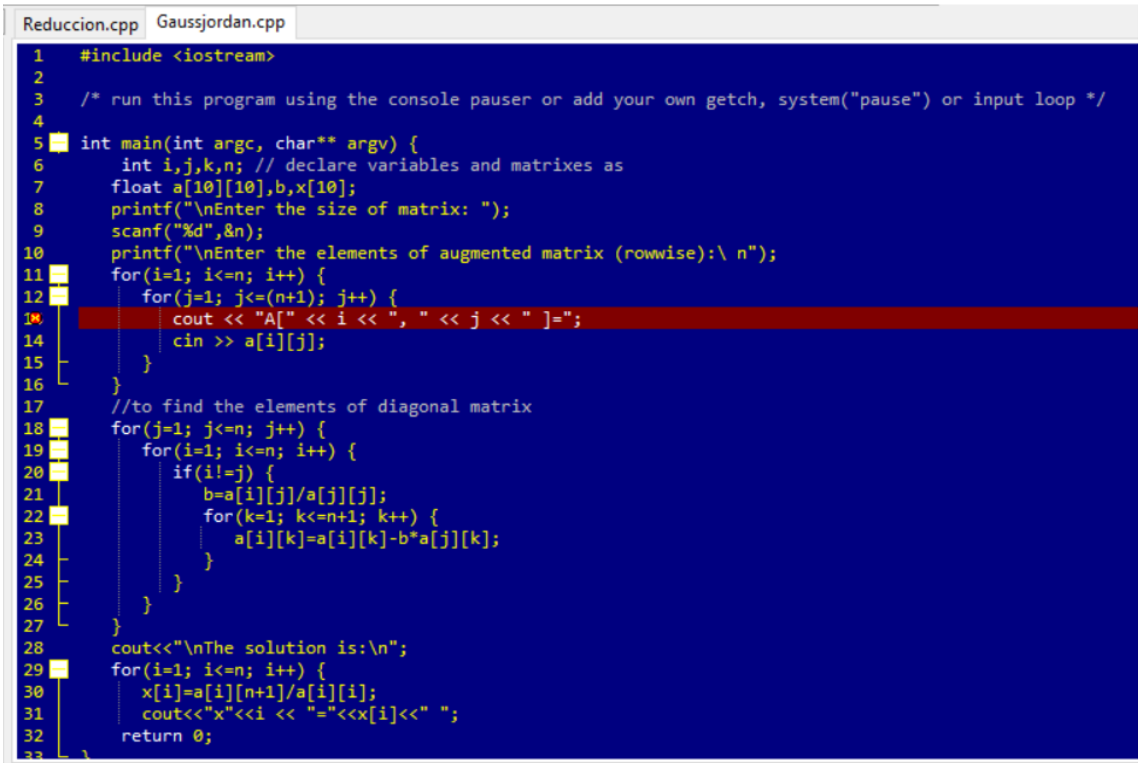
Software del Método de Gauss Jordan

Al momento de probar el código que fue tomado de la red, nos percatamos que existía un error, el cual fue solucionado usando las siguientes sentencias.

#include <math.h>

using namespace std;

Figura 4.9 “Resultados”




```
#include <iostream>

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

int main(int argc, char** argv) {
    int i,j,k,n; // declare variables and matrixes as
   float a[10][10],b,x[10];
   printf("\nEnter the size of matrix: ");
   scanf("%d",&n);
   printf("\nEnter the elements of augmented matrix (rowwise):\ n");
   for(i=1; i<=n; i++) {
      for(j=1; j<=(n+1); j++) {
         cout << "A[" << i << ", " << j << " ]=";
         cin >> a[i][j];
      }
   }
   //to find the elements of diagonal matrix
   for(j=1; j<=n; j++) {
      for(i=1; i<=n; i++) {
         if(i!=j) {
            b=a[i][j]/a[j][j];
            for(k=1; k<=n+1; k++) {
               a[i][k]=a[i][k]-b*a[j][k];
            }
         }
      }
   }
   cout<<"\nThe solution is:\n";
   for(i=1; i<=n; i++) {
      x[i]=a[i][n+1]/a[i][i];
      cout<<"x"<<i << "="<<x[i]<<" ";
    return 0;
}
```

Figura 4.9 Código antes de ser corregido.

Figura 4.10 “Resultados”

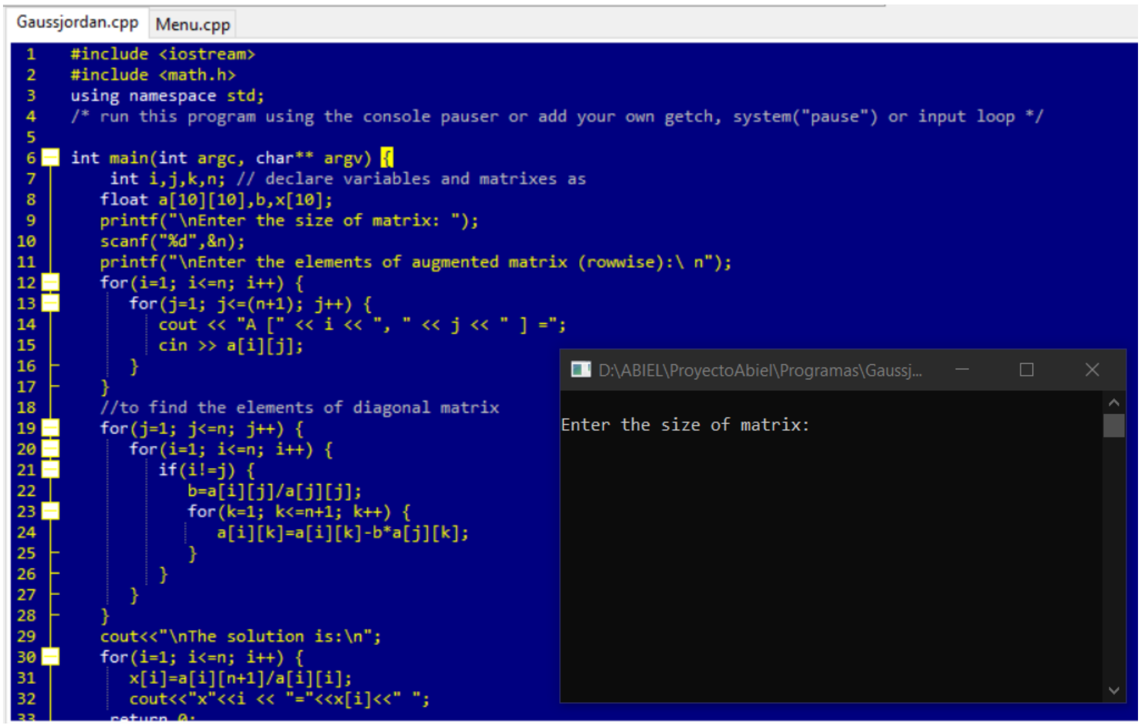


Figura 4.10 Muestra que código se ejecuta correctamente.

Una vez que confirmamos la correcta ejecución del software, ingresamos los valores de la matriz y el resultado es el siguiente:

Figura 4.11 “Resultados”

```
D:\ABIEL\ProyectoAbiel\Programas\Gaussjordan.exe

Enter the size of matrix: 3

Enter the elements of augmented matrix (rowwise): nA [1, 1 ] =4
A [1, 2 ] =5
A [1, 3 ] =6
A [1, 4 ] =24
A [2, 1 ] =1
A [2, 2 ] =2
A [2, 3 ] =3
A [2, 4 ] =9
A [3, 1 ] =3
A [3, 2 ] =1
A [3, 3 ] =-2
A [3, 4 ] =4

The solution is:
x1=4
--------------------------------
Process exited after 27.81 seconds with return value 0
Presione una tecla para continuar . . .
```

Dentro de las observaciones se puede notar la diferencia de idioma, el cual es el inglés, por lo cual, nos dimos a la tarea de realizar la traducción, también nos percatamos que solo arroja el valor para x, sin embargo, es correcto el cálculo, ya que dad como resultado 4.

Implementación del menú

Se realizo por medio de funciones, las cuales son invocadas conforme a la selección del usuario, para el menú se hizo uso de la estructura switch, dentro de la función void().

Figura 4.12 “Resultados”

```
void menu(){
    int op=0;
        printf("\n\n\n\t ****************************************************************");
        printf("\n\n\n\t ELIJA UNA OPCI%cN, POR FAVOR: ",224);
        scanf("%d",&op);
        switch(op)
        {
            case 1:
            gaus();
            break;

            case 2:
                reduccion();
            break;
            case 3:
                exit(3);
            break;
            default: printf("\n\tOpci%cn invalida. Pulse ENTER para continuar", 162);

        }
        menu();
}
```

Una vez que ensamblamos ambos códigos realizamos las pruebas correspondientes y este fue el resultado:

Figura 4.13 “Resultados”

```
C:\Users\veron\AppData\Local\Temp\Rar$DIa8812.16050\Menu.exe

1.-Metodo de Gauss-Jordan
2.-Metodo de Reduccion
3.-Salir
Elije una opcion: 1

Enter the size of matrix: 3

Enter the elements of augmented matrix (rowwise): nA[1, 1 ]=4
A[1, 2 ]=5
A[1, 3 ]=6
A[1, 4 ]=24
A[2, 1 ]=1
A[2, 2 ]=2
A[2, 3 ]=3
A[2, 4 ]=9
A[3, 1 ]=3
A[3, 2 ]=1
A[3, 3 ]=-2
A[3, 4 ]=4

The solution is:
x1=4 x2=-2 x3=3 "pausa" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

1.-Metodo de Gauss-Jordan
2.-Metodo de Reduccion
3.-Salir
Elije una opcion:
```

Figura Resultado 1.13 Muestra el Método Gauss Jordan implementado en el menú.

Como podemos observar ya nos indica los tres valores que son asignados a ***x*, *y*** y ***z***.

Figura 4.14 “Resultados”

```
********************WELCOME******************************

PROGRAMA QUE RESUELVE UN SISTEMA DE ECUACIONES LINEALES

DE 5x5 POR EL METODO DE GAUSS JORDAN

a11x + a12y + a13z = b1

a21x + a22y + a23z = b2

a31x + a32y + a33z = b3

*******************

** PARA LA ECUACION 1 **

INGRESE EL VALOR DE a11: 4


INGRESE EL VALOR DE a12: 5


INGRESE EL VALOR DE a13: 6

INGRESE EL VALOR DE b1: 24
```

Figura Resultado 4.14 Muestra el Método Reducción implementado en el menú.

Se ajusto la manera en la que se muestran los mensajes en la consola, para que a primera vista resulta agradable para el usuario.

Figura 4.15 “Resultados”

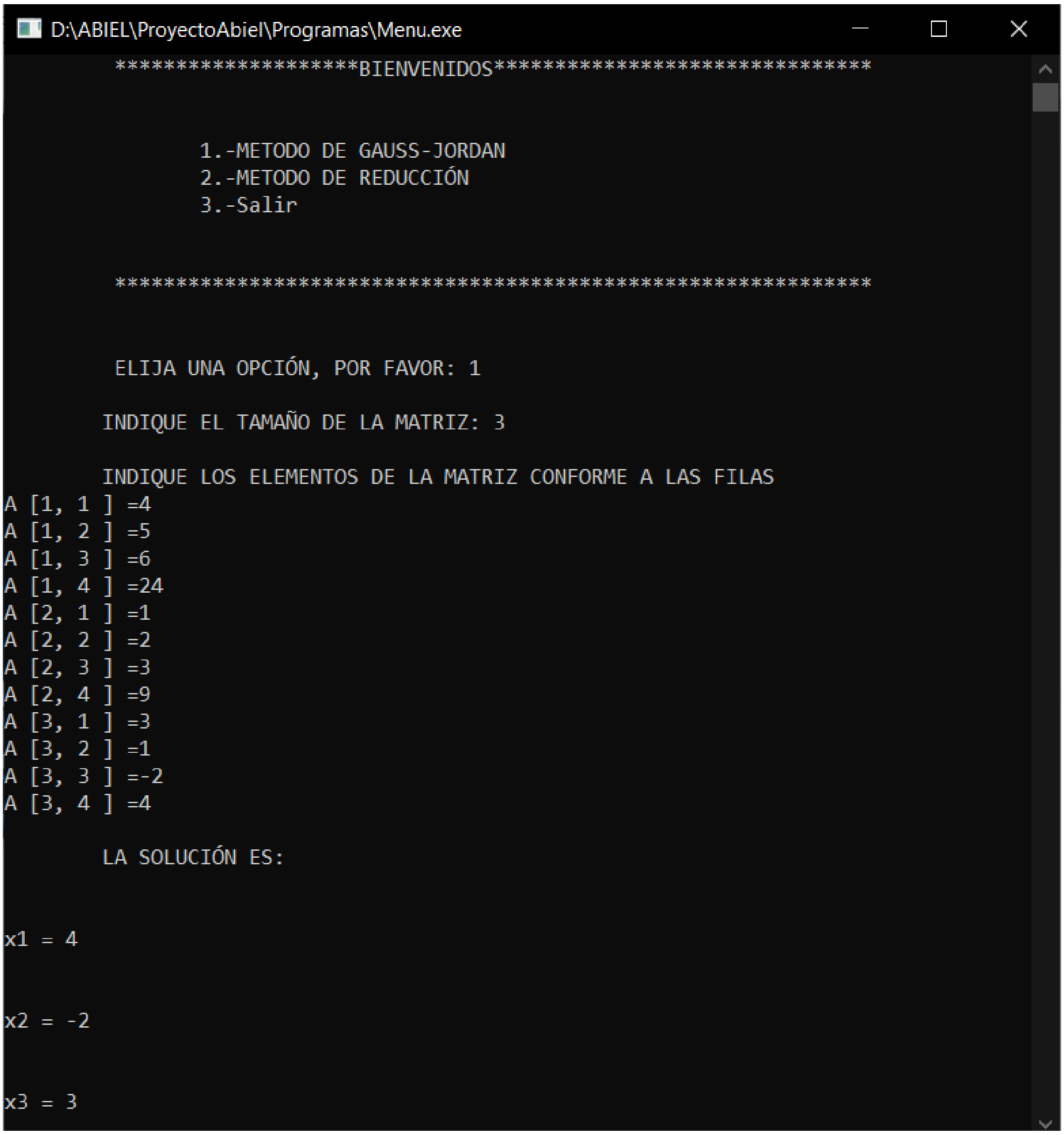


Figura Resultado 4.15 Versión final del Método Gauss Jordan.

Figura 4.16 “Resultados”

```
D:\ABIEL\ProyectoAbiel\Programas\Menu.exe
****************************************************************

TU ELECCIÓN RESUELVE UN SISTEMA DE ECUACIONES LINEALES

DE 3x3 POR EL METODO DE REDUCCIÓN

a11x + a12y + a13z = b1

a21x + a22y + a23z = b2

a31x + a32y + a33z = b3

****************************************************************

** INDIQUE LOS VALORES PARA LA ECUACIÓN 1 **

INGRESE EL VALOR DE a11: 4


INGRESE EL VALOR DE a12: 5


INGRESE EL VALOR DE a13: 6

INGRESE EL VALOR DE b1: 24



** INDIQUE LOS VALORES PARA LA ECUACIÓN 2 **

INGRESE EL VALOR DE a21: 1


INGRESE EL VALOR DE a22: 2
```

Figura Resultado 4.16 Versión final del Método de Reducción.

Figura 4.17 “Resultados”

```
D:\ABIEL\ProyectoAbiel\Programas\Menu.exe

        INGRESE EL VALOR DE a33: -2


        INGRESE EL VALOR DE b3: 4

        El sistema es:
        4.00x 5.00y 6.00z =24.00
        1.00x 2.00y 3.00z =9.00
        3.00x 1.00y -2.00z =4.00


        Primer paso (restar ecuacion 1 y ecuacion 2)
        4.00x 5.00y 6.00z =24.00
        -4.00x -8.00y -12.00z =-36.00
        --------------------------------------------
        0.00x +-3.00y +-6.00z =-12.00


        Segundo paso (restar ecuacion 1 y ecuacion 3)
        12.00x 15.00y 18.00z =72.00
        -12.00x -4.00y 8.00z =-16.00
        --------------------------------------------
        0.00x 11.00y 26.00z =56.00


        Tercer paso (restar ecuacion 4 y ecuacion 5)
        -0.00x 33.00y 66.00z =132.00
        -0.00x -33.00y -78.00z =-168.00
        --------------------------------------------
        -0.00x 0.00y -12.00z =-36.00


        Cuarto paso (sustitucion de z en la ecuacion 4)
        0.00x -3.00y -18.00z =-12.00


        Quinto paso (sustitucion de z, y en la ecuacion 1)
        4.00x -10.00y 18.00z =24.00


        Nuestras incognitas son

        x= 3.00
        y= -2.00
        z= 4.00

"pausa" no se reconoce como un comando interno o externo,
```

Figura Resultado 4.17 Versión final que muestra los resultados del Método de Reducción.

Conclusiones

Al resolver el sistema de ecuaciones por el método de Reducción, percibimos y demostramos de forma satisfactoria que sí tiene una estrecha relación con la máquina de Turing y decimos estrecho porqué se tiene que seguir una estricta serie de pasos como obtener las ecuaciones 4 y 5 para poder saber el valor de las incógnitas, sin embargo, existe la flexibilidad, al momento en que la persona que este desarrollando el ejercicio decida que incógnita desea saber primero, si queremos ver un sentido más estricto, podemos decir que el orden para encontrar los valores de las incógnitas será determinado por orden alfabético, pero al final el resultado siempre deberá ser el mismo.

En cuanto a las semejanzas del Cálculo Lambda con el método de Gauss Jordan, expresamos el uso del inverso aditivo y el inverso multiplicativo, el número de iteraciones que deben ser empleados varia conforme al tamaño de la matriz, ya que, el número de elementos que contendrá será variable. El desarrollar este método en Excel nos permite ver que es repetitivo y escalonado (la posición del pivote).

Hablando en términos de programación, al estudiar el comportamiento del código tomado de la red, este cuenta con dos ciclos for principales o en otras palabras dos ciclos de iteraciones con los cuales se busca reducir el conjunto de números y al final entrega un resultado según la condición otorgada.

Por los valores usados en los coeficientes, no abordamos con detenimiento el comportamiento de los decimales, sin embargo, dejamos ver un ejemplo en el cual se ven comprometidos los resultados.

Para finalizar queremos añadir que la descripción del proceso fue extensa, porqué nos tomamos el tiempo necesario para asegurarnos que todos los términos sean claros y comprensibles.

Fuentes de información

Wikipedia contributors. (2021, 21 abril). Theory of computation. Wikipedia. https://en.wikipedia.org/wiki/Theory_of_computation#cite_note-Sipser-3rd-1

Teoría de la Computación. (s. f.). ICC - Instituto de Ciencias de la Computación. Recuperado 16 de junio de 2021, de https://icc.fcen.uba.ar/teoria-de-la-computacion/

Wikipedia contributors. (2021, 7 mayo). Computability theory. Wikipedia. https://en.wikipedia.org/wiki/Computability_theory

Wikipedia contributors. (2021b, junio 13). Model of computation. Wikipedia. https://en.wikipedia.org/wiki/Model_of_computation

Wikipedia contributors. (2021b, junio 13). Lambda calculus. Wikipedia. https://en.wikipedia.org/wiki/Lambda_calculus

Colaboradores de Wikipedia. (2021, 16 junio). Eliminación de Gauss-Jordan. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Eliminaci%C3%B3n_de_Gauss-Jordan

Parrales, H. (2018, 12 noviembre). Método de Reducción. Ecuación Lineal. https://ecuacionlineal.com/lineal/metodo-reduccion/

Arevalomaria, L. T. L. E. (2017, 21 noviembre). Diferencias entre Metodologías Tradicionales y Ágiles #MetodologiasAgiles. Maria Eugenia Arevalo Lizardo. https://arevalomaria.wordpress.com/2011/11/15/diferencias-entre-metodologias-tradicionales-y-agiles-metodologiasagiles/

Historia de Scrum. (2017, 23 marzo). Proyectos Ágiles. https://proyectosagiles.org/historia-de-scrum/

Qué es SCRUM. (2018, 9 octubre). Proyectos Ágiles. https://proyectosagiles.org/que-es-scrum/

Colaboradores de Wikipedia. (2021, 28 mayo). Scrum (desarrollo de software). Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)

EMMANUEL ASESORÍAS. (2019, 4 septiembre). Método de REDUCCIÓN. Sistema de Ecuaciones de 3x3 [Vídeo]. YouTube. https://www.youtube.com/watch?v=aW4v3ROkTek

Kennya Martinez. (2020, 20 marzo). Reducción Escalonada MAtriz 3x3 Excel [Vídeo]. YouTube. https://www.youtube.com/watch?v=zO2bKRJAeg4

C++ Program to Implement Gauss Jordan Elimination. (s. f.). Tutorialspoint. Recuperado 21 de junio de 2021, de https://www.tutorialspoint.com/cplusplus-program-to-implement-gauss-jordan-elimination

Wikipedia contributors. (2021, 17 junio). C (programming language). Wikipedia. https://en.wikipedia.org/wiki/C_(programming_language)

Ortiz, M. (2021, 11 marzo). Qué es Excel y para qué sirve una hoja de cálculo •. Excel Total. https://exceltotal.com/que-es-excel/

Parrales, H. (2018, 12 noviembre). Método de Reducción. Ecuación Lineal. https://ecuacionlineal.com/lineal/metodo-reduccion/

Anexos

Bitácora

Fecha	Actividad	Comentarios
05/05/2021	Creación del documento de Word que contendrá la información del proyecto	N/A
20/05/2021	El profesor nos asigna el tema del proyecto "Cálculo Lambda" y nos indica los primeros puntos a desarrollar	N/A
27/05/2021	Investigación sobre "Cálculo Lambda"	Por falta de conocimiento, no se presentan los primeros puntos a desarrollar (índice, resumen y justificación). Se realizo una breve investigación para averiguar más sobre dicha teoría y poder comprender como representarla, así mismo se realizó una lluvia de ideas.
28/05/2021	Contacto con el profesor	Se resuelven dudas con el profesor respecto al tema y se mencionan la idea en la cual se basó nuestro desarrollo de proyecto, una vez que se tuvo la aprobación del profesor continuamos labores.
28/05/2021	Delegación de actividades	Asignamos tareas principales según las necedades, sin embargo, todos se vieron involucrados en cada fase. *Ayala Palma: Desarrollo de pruebas de escritorio e implementación en código. *García Serrano: Investigación matemática, desarrollo escrito de los problemas e implementación de código. **Picazo Pérez: Investigación y redacción de documentación. **Vázquez Rojas: Investigación y redacción de documentación.
03/06/2021	Metodología de trabajo	Se inicio con la investigación de la metodología de trabajo que seguiríamos para el desarrollo.
05/06/2021	Revisión de equipo	A través de una conferencia en Meet demostrábamos avances y discutíamos dudas con respecto a los modelos.
05/06/2021	Creación de bitácora	Por acuerdo mutuo del equipo, se decide empezar a llevar un registro formal con todo lo relacionado al proyecto.
10/06/2021	Revisión por parte del profesor	Mostramos el avance obtenido y se tomó nota de los puntos a mejorar.
10/06/2021	Correcciones	Se realiza la modificación de los puntos a mejorar
12/06/2021	Revisión de equipo	A través de una conferencia en Meet, confirmamos que los puntos de

		corrección inmediata fueron realizados de forma correcta y se llevó a cabo la demostración de los primeros pasos del desarrollo de solución de matrices por los métodos de Gauss Jordan y Reducción.
14/06/2021	Cuadro comparativo	Dadas las indicaciones por el profesor se inicia la investigación para presentar un cuadro comparativo con referencia a las distintas metodologías de trabajo
16/06/2021	Marco Teórico	Se da inicio a la construcción del marco teórico.
18/06/2021	Cuadro comparativo	El cuadro es anexado al documento
19/06/2021	Inicio de desarrollo	Se comienza a redactar con respecto a cómo fue el proceso de implementación de los métodos de resolución de matrices y se termina de redactar el capítulo 1.
20/06/2021	Capítulo 2 y 3	Se inicia la redacción del capítulo 2 y 3
20/06/2021	Capítulo 2 y 3	Se finaliza la redacción de ambos capítulos
21/06/2021	Resultados	Se inicio con la redacción de esta sección, fue elaborada con la finalidad de mostrar un análisis más profundo con lo que respecta a los resultados y el proceso de codificación.
21/06/2021	Conclusión	Se redactan las conclusiones elaboradas entre todos los miembros del equipo.
21/06/2021	Revisión	Se envía el documento al profesor para que pueda hacer retroalimentación del contenido.
22/06/2021	Envió	Se envía el contenido al profesor a través de Gmail, dicho contenido ya se encuentra en GitHub y en los perfiles de todos los integrantes.

Glosario

C: es un lenguaje de programación de computadoras de propósito general que soporta programación estructurada.

Cálculo Lambda: es un sistema formal en lógica matemática para expresar la computación basada en la abstracción de funciones y la aplicación utilizando el enlace y la sustitución de variables.

Ecuación Lineal:

Excel: Excel es un programa informático desarrollado por Microsoft y forma parte de Office que es una suite ofimática.

Gauss Jordan : es un algoritmo que se usa para determinar la inversa de una matriz y las soluciones de un sistema de ecuaciones lineales.

Método de Determinantes: es una forma de resolución de sistemas de ecuaciones lineales, al igual que los métodos sustitución e igualación.

Método de Reducción: El método Reducción es una manera de resolver un sistema lineal, pero sólo se puede utilizar en sistemas de resolución que el número de ecuaciones y el número de incógnitas son iguales.

IDE: Entorno de Desarrollo Integrado, es una herramienta que le facilita desarrollar software.