



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS NUMÉRICO COMO HERRAMIENTA
DIDÁCTICA PARA LOS ESTUDIANTES DEL CURSO MATEMÁTICA APLICADA 3 DEL
DEPARTAMENTO DE MATEMÁTICAS DE LA FACULTAD DE INGENIERÍA, USAC**

José Carlos Bautista Mazariegos

Asesorado por el Ing. Javier Estuardo Navarro Delgado

Guatemala, septiembre de 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS NUMÉRICO COMO HERRAMIENTA
DIDÁCTICA PARA LOS ESTUDIANTES DEL CURSO MATEMÁTICA APLICADA 3 DEL
DEPARTAMENTO DE MATEMÁTICAS DE LA FACULTAD DE INGENIERÍA, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JOSÉ CARLOS BAUTISTA MAZARIEGOS

ASESORADO POR EL ING. JAVIER ESTUARDO NAVARRO DELGADO

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, SEPTIEMBRE DE 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Herman Igor Véliz Linares
EXAMINADOR	Ing. José Alfredo González
EXAMINADOR	Ing. Oscar Alejandro Paz Campos
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS NUMÉRICO COMO HERRAMIENTA DIDÁCTICA PARA LOS ESTUDIANTES DEL CURSO MATEMÁTICA APLICADA 3 DEL DEPARTAMENTO DE MATEMÁTICAS DE LA FACULTAD DE INGENIERÍA, USAC

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha febrero 2021



José Carlos Bautista Mazariegos

Guatemala, 05 de mayo de 2022

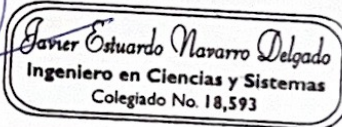
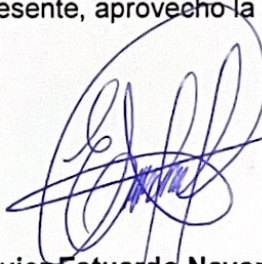
Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por los estudiantes **José Carlos Bautista Mazariegos** con carné **201504231** y CUI **2666 91234 0101** titulado **"IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS NUMÉRICO COMO HERRAMIENTA DIDÁCTICA PARA LOS ESTUDIANTES DEL CURSO MATEMÁTICA APLICADA 3 DEL DEPARTAMENTO DE MATEMÁTICAS DE LA FACULTAD DE INGENIERÍA, USAC"**, lo he revisado y luego de corroborar que el mismo se encuentra concluido y que cumple con los objetivos propuestos en el respectivo protocolo, procedo a la aprobación respectiva.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Javier Estuardo Navarro Delgado
Colegiado No. 18593



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 14 de julio de 2022

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **JOSÉ CARLOS BAUTISTA MAZARIEGOS** con carné **201504231** y CUI **2666 91234 0101** titulado **“IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS NUMÉRICO COMO HERRAMIENTA DIDÁCTICA PARA LOS ESTUDIANTES DEL CURSO DE MATEMÁTICA APLICADA 3 DEL DEPARTAMENTO DE MATEMÁTICAS DE LA FACULTAD DE INGENIERÍA, USAC”** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA

LNG.DIRECTOR.170.EICCSS.2022

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS NUMÉRICO COMO HERRAMIENTA DIDÁCTICA PARA LOS ESTUDIANTES DEL CURSO MATEMÁTICA APLICADA 3 DEL DEPARTAMENTO DE MATEMÁTICAS DE LA FACULTAD DE INGENIERÍA, USAC**, presentado por: **José Carlos Bautista Mazariegos**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”



Msc. Ing. Carlos Gustavo Alonzo
Director

Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, septiembre de 2022



LNG.DECANATO.OI.603.2022

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS NUMÉRICO COMO HERRAMIENTA DIDÁCTICA PARA LOS ESTUDIANTES DEL CURSO MATEMÁTICA APLICADA 3 DEL DEPARTAMENTO DE MATEMÁTICAS DE LA FACULTAD DE INGENIERÍA, USAC**, presentado por: **José Carlos Bautista Mazariegos**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Inga. Aurelia Anabela Cordova Estrada
Decana



Guatemala, septiembre de 2022

AACE/gaoc

ACTO QUE DEDICO A:

- Dios** Por su amor incondicional e infinito, por haberme escuchado en los momentos difíciles y haber encontrado en él, un refugio de paz y guía durante toda mi carrera.
- Mi mamá** Por tus oraciones y amor incondicional, por ser mi confidente, por emocionarte por mis triunfos y ser mi consuelo en mis derrotas. Por tus abrazos y cuidados especiales.
- Mi papá** Por tu amor y ejemplo de integridad, por tus palabras oportunas y consejos. Me enseñas el amor de Dios con tus acciones y me demuestras que la misericordia de Dios nos acompaña y nos bendice
- Mis hermanos** David y Elizabeth Bautista, por su particular manera de apoyarme y estar siempre para mí. Por sus muestras de cariño y por las pequeñas lecciones que me enseñaron. Espero ser siempre un buen ejemplo para ustedes como hermano mayor.

Mis amigos

Por haberme regalado recuerdos inolvidables y momentos trascendentales en esta vida. Por demostrarme lo que significa la esencia de una amistad sólida y duradera. En especial agradezco a Andrés Montufar, Carlos Gil, Daniel Franco, Edgar Cil, Erik Flores, Eyron Steve, Fredy Ramírez, Gabriela Lombardi, Javier Navarro, Jonathan Mairén, Jose Girón, Karina Mejía, Kathleen Aquino, Kimberly Aristondo, Mafer Villagrán, Pablo Rosas.

Mis tías

Por siempre tener cuidados y palabras especiales para mí, apoyarme y celebrar mis triunfos conmigo. Por sus oraciones y detalles que siempre tuvieron conmigo. En especial agradezco a Carmen Bautista y Lily Mazariegos.

AGRADECIMIENTOS A:

Guatemala

La tierra que me vio crecer, espero que mi profesión y mis conocimientos sean útiles para tu crecimiento.

Universidad de San Carlos de Guatemala

Por enseñarme tanto de la vida, por ser mi casa de estudios que me acogió y preparó para convertirme en un buen profesional.

Facultad de ingeniería

Por haberme brindado tantos conocimientos y experiencias de vida.

Mi asesor Ing.

Javier Estuardo Navarro Delgado, gracias por haber compartido tus conocimientos y por todo tu apoyo durante esta odisea.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN	XVII
1. ESTUDIO DE LA HERRAMIENTA EDUCATIVA Y SU IMPACTO EN GUATEMALA	1
1.1. Teoría detrás de la investigación	1
1.1.1. Aplicaciones de la Matemática Aplicada 3.....	1
1.1.2. Retos y desafíos de la Matemática Aplicada 3	2
1.2. Teoría y relación con la herramienta	2
1.2.1. Método de bisección.....	3
1.2.1.1. Algoritmo de bisección.....	4
1.2.2. Método de punto fijo	5
1.2.2.1. Algoritmo de punto fijo	7
1.2.3. Método de Newton.....	9
1.2.3.1. Algoritmo del método Newton- Raphson	10
1.2.4. Método de la secante	12
1.2.4.1. Algoritmo del método de la secante.....	13
1.2.5. Método posición falsa o regla Farsi	15
1.2.5.1. Algoritmo del método de posición falsa	17

1.2.6.	Método de Steffensen	20
1.2.6.1.	Algoritmo del método de Steffenson	21
1.2.7.	Método de Muller.....	23
1.2.7.1.	Algoritmo del método de Muller.....	24
1.2.8.	Polinomios de Lagrange.....	26
1.2.8.1.	Algoritmo del método de polinomios de Lagrange	27
1.2.9.	Método de aproximación de Neville	30
1.2.9.1.	Algoritmo del método de Neville.....	31
2.	IDENTIFICACIÓN DEL PROBLEMA Y LA SOLUCION QUE LA APLICACIÓN PROPONDRA	33
2.1.	Antecedentes	33
2.2.	Usuarios potenciales de la aplicación	35
2.3.	FODA	36
2.4.	Benchmark	38
3.	ARQUITECTURAS EN LA NUBE	39
3.1.	Arquitectura <i>serverless</i>	39
3.1.1.	Proveedores de arquitectura <i>serverless</i>	39
3.2.	Microservicios	41
3.2.1.	Que se considera como microservicios	41
3.2.2.	Arquitectura de microservicios	42
3.2.3.	Ventajas de utilizar microservicios	42
3.3.	Arquitectura SOA	42
3.3.1.	Modelo de servicios.....	44
3.3.2.	Modelos de despliegue Cloud	45
3.3.3.	Orquestación de servicios	46

4.	DOCUMENTACIÓN DE ESPECIFICACIONES, REQUERIMIENTOS Y CRITERIOS DE ACEPTACIÓN DE LA APLICACIÓN WEB	49
4.1.	Objetivo estratégico	49
4.2.	Objetivo general	49
4.3.	Lista de acrónimos y glosario de la aplicación.....	49
4.4.	Alcances.....	50
4.5.	Análisis de riesgos.....	51
4.6.	Arquitectura de la solución	52
4.7.	Descripción de módulos principales	53
4.8.	Requerimientos	54
5.	PROCESO DE DESARROLLO DEL PROYECTO	55
5.1.	Metodología ágil de trabajo	55
5.1.1.	Manifiesto ágil.....	56
5.1.2.	Scrum	57
5.1.3.	Scrumban	58
5.2.	Ciclo de vida del proyecto.....	59
5.2.1.	Fase de inicio del proyecto	59
5.2.2.	Planificación del proyecto	60
5.2.3.	Ejecución del proyecto.....	61
5.2.4.	Supervisión y control de proyectos	63
5.2.5.	Cierre del proyecto	63
5.2.6.	Tiempos de cierre de proyecto	64
5.3.	Tecnologías	64
5.3.1.	Proveedores de servicios en la nube.....	65
5.3.2.	API.....	71
5.3.3.	<i>Framework</i>	75
5.3.4.	<i>Backend</i>	77
5.3.4.1.	<i>Golang</i>	78

	5.3.4.2.	Python	79
	5.3.4.3.	NodeJs	80
	5.3.5.	FrontEnd.....	82
	5.3.5.1.	Angular CLI	83
	5.3.5.2.	React JS.....	84
6.		IMPLEMENTACIÓN DEL PROYECTO.....	87
6.1.		AWS Cloud	87
	6.1.1.	S3 <i>bucket</i>	87
		6.1.1.1. Proceso para la configuración del servicio Amazon S3.....	88
	6.1.2.	Funciones Lambda.....	89
		6.1.2.1. Conceptos básicos de las funciones Lambda	89
		6.1.2.2. Proceso para la configuración de una función Lambda.....	90
		6.1.2.3. Invocar una función Lambda	90
		6.1.2.4. Precios para el uso de las funciones Lambda	91
	6.1.3.	Instancias EC2	91
		6.1.3.1. Proceso para iniciar y configurar una instancia EC2	91
		6.1.3.2. Lanzamiento de una instancia.....	92
6.2.		<i>Backend</i>	93
6.3.		Front-End	94

CONCLUSIONES	97
RECOMENDACIONES	99
BIBLIOGRAFÍA.....	101
APÉNDICES	103

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Representación gráfica del algoritmo del método de bisección	3
2.	Diagrama de flujo que da solución al método de bisección.....	5
3.	Solución grafica del método de punto fijo.....	7
4.	Diagrama de flujo que da solución al método de punto fijo	8
5.	Solución grafica del método de Newton	10
6.	Diagrama de flujo del método de Newton- Raphson.....	11
7.	Representación gráfica del método de la secante con rectas tangentes	13
8.	Diagrama de flujo para el método de la secante	15
9.	Gráfico con la representación de cómo trabaja el método de la posición falsa	17
10.	Diagrama de flujo para el método de posición falsa.....	19
11.	Diagrama de flujo para el método de Steffenson	22
12.	Gráfico con la representación de cómo trabaja el método de Muller ...	24
13.	Diagrama de flujo para el método de polinomios Lagrange	29
14.	Arquitectura propuesta para la solución	52
15.	Comparativo entre empresas que proporcionan los servicios en la nube de cada uno de los escenarios de tipos de instancias	69
16.	Precios bajo demanda que proporcionan las empresas de cada uno de los escenarios de tipos de instancias	69

TABLAS

I.	Método de aproximación de Neville	30
II.	Comparativo de herramientas que se asemejan a la aplicación.....	34
III.	Indicadores clave	38
IV.	Cuadro comparativo de herramientas para desarrollar el <i>frontend</i>	76

LISTA DE SÍMBOLOS

Símbolo	Significado
∞	Infinito
$>$	Mayor que
$<$	Menor que
$\%$	Porcentaje
Σ	Sumatoria

GLOSARIO

Algebra lineal	Se le conoce como la especialización del algebra que trabaja con matrices, vectores, espacios vectoriales y ecuaciones de tipo lineal.
Almacén de datos	Un almacén de datos es una colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que sirve como soporte en la toma de decisiones.
API	API es una sigla que procede de la lengua inglesa y que alude a la expresión <i>Application Programming Interface</i> (cuya traducción es Interfaz de Programación de Aplicaciones). El concepto hace referencia a los procesos, las funciones y los métodos que brinda una determinada biblioteca de programación a modo de capa de abstracción para que sea empleada por otro programa informático.
Función <i>serverless</i>	Es una función programada para un solo propósito. Estas funciones están basadas sobre una infraestructura en la nube. Las empresas en la nube son las que se encargan del mantenimiento y la ejecución del código para que los desarrolladores puedan implementar código nuevo de forma más rápida y sencilla.

RESUMEN

El propósito de la aplicación se desarrolló para darle la posibilidad a los estudiantes de contar con una aplicación intuitiva que agiliza el proceso de resolución de problemas de los métodos que pertenecen al álgebra lineal. En la actualidad existen soluciones para los estudiantes de la Facultad de Ingeniería de la Universidad de San Carlos Guatemala, que implican el uso de una computadora física y un archivo de Excel que haya sido programado, para darle solución a los problemas matemáticos propuestos durante el curso de matemática aplicada 3.

Es importante dar a conocer que los avances tecnológicos en la actualidad, permiten una capacidad de procesamiento superior a la que se solía tener hace unos años. Actualmente los servicios en la nube proveen una solución viable para ejecutar una herramienta en la web capaz de procesar en tiempo real cada una de las operaciones necesarias, esto sin comprometer o tener dependencia de un dispositivo con especificaciones mínimas de procesamiento.

Por ese motivo que se propone una solución que pueda ser de fácil acceso para los estudiantes con tan solo contar con el url del aplicativo se pudiese rápidamente hacer uso de la herramienta, esto tomando en cuenta un buen desempeño y tiempo de respuesta por parte de los servicios en la nube que se configuraron, innovando la implementación de funciones *serverless* en la nube.

La estructura tecnológica con la que fue desarrollada la aplicación y la tecnología detrás de ella forman parte de las nuevas tendencias actuales en cuanto a arquitecturas de software se refiere. Esto sumado al hecho de que la arquitectura está montada sobre una plataforma en la nube, incluyendo el mismo aplicativo y sus servicios de procesamiento, le agrega valor y dota a la herramienta de una capacidad superior a las demás en el mercado de herramientas para el procesamiento de ecuaciones diferenciales.

OBJETIVOS

General

Implementar un sistema de análisis numérico como herramienta didáctica para los estudiantes del curso Matemática Aplicada 3, del Departamento de Matemáticas de la Facultad de Ingeniería, USAC.

Específicos

1. Crear una solución que permita a los estudiantes ingresar con facilidad a la herramienta didáctica del curso Matemática aplicada 3.
2. Disminuir la interacción con otros programas, para optimizar el proceso con la herramienta didáctica.
3. Optimizar los tiempos de respuesta para mejorar la experiencia de usuario.
4. Desarrollar una interfaz que sea amigable para el usuario al momento de trabajar con la herramienta didáctica.

INTRODUCCIÓN

Con el realce de la era digital, el análisis numérico se vuelve más importante para la sociedad. El valor de la información es cada día mejor valorado, y poder tener buenos criterios basados en la matemática dan los datos para poder tomar decisiones o simplemente ampliar los conocimientos sobre una vasta cantidad de temas.

En la actualidad ante una era digital sin precedentes, el estudiante se ha ido inclinando por el estudio de las ciencias exactas ya sea en ingeniería o en una matemática pura. Sabiendo que es necesario el uso de una herramienta que realiza la parte operativa de los algoritmos para el análisis numérico que abarca varios cursos de matemática, es posible desarrollar esta herramienta en una plataforma web.

Los sistemas de información van ligados a los datos y a la matemática, se complementan y sirven de base para realizar modelos, generar bases de datos, solucionar problemas, automatizar procesos, promover actividades eficientes, análisis de comportamiento, hacer confiable y crecer negocios, facilitar interacciones y ser de beneficio al ser humano.

Prácticamente, todo lo relacionado con industria, comercio, construcción, servicios, talento humano, minería, aeroespacial, entre otros. está relacionado con los sistemas de información. En la búsqueda de la mejora continua es importante, porque todo proceso se puede mejorar, lo importante es encontrar un punto de equilibrio. En este momento, ante una era digital sin precedentes, el estudiante se ha ido inclinando por el estudio de las ciencias exactas ya sea

en ingeniería o en una matemática pura. Sabiendo que es necesario el uso de una herramienta que realiza la parte operativa de los algoritmos para el análisis numérico que abarca varios cursos de matemática, es posible desarrollar esta herramienta en una plataforma web.

1. ESTUDIO DE LA HERRAMIENTA EDUCATIVA Y SU IMPACTO EN GUATEMALA

Durante el siguiente capítulo se estará abordando el estudio basado en la teoría matemática que contempla el curso de Matemática aplicada 3. Interconectando la parte teórica con la de la lógica de programación que procesará y dará solución a los problemas matemáticos.

1.1. Teoría detrás de la investigación

Se hace un breve análisis del curso en cuestión al que se le propone el desarrollo del aplicativo.

1.1.1. Aplicaciones de la Matemática Aplicada 3

Una de las actividades más frecuentes en el curso de Matemática Aplicada 3 es interpretar y analizar el manejo de métodos para resolver ecuaciones de una variable y el planteamiento de polinomios de aproximación, es decir, la intersección con el eje de las x o la solución como tal. Por lo que se debe tomar en cuenta que no todas las ecuaciones tienen una sola solución, y que no todas tienen solución, es por eso que se debe tener una idea de la forma que tendrá la representación de la curva de la ecuación antes de comenzar a ejecutar el algoritmo del método para resolver estos sistemas de ecuaciones lineales y no lineales.

1.1.2. Retos y desafíos de la Matemática Aplicada 3

Se enlistan los retos y desafíos que se enfrentan a la hora de desarrollar el aplicativo.

- Reconocer una función lineal por medio del análisis de su tabla de valores, gráfico o ecuación y conociendo uno de los tres modelos anteriores, determinar los otros dos para comprender y predecir variaciones constantes.
- Contrastar la función lineal con la función exponencial para comprender las diferencias entre variaciones constantes y variables.
- Reconocer los conceptos en sus distintas representaciones, procedimientos y métodos matemáticos para la correcta formulación, análisis y resolución de problemas involucrados en ingeniería y ciencias afines, por medio de modelos matemáticos adecuados.

1.2. Teoría y relación con la herramienta

Dentro de este apartado se trabajan las ecuaciones donde solamente se encuentra una variable que se encuentra elevada al exponente uno

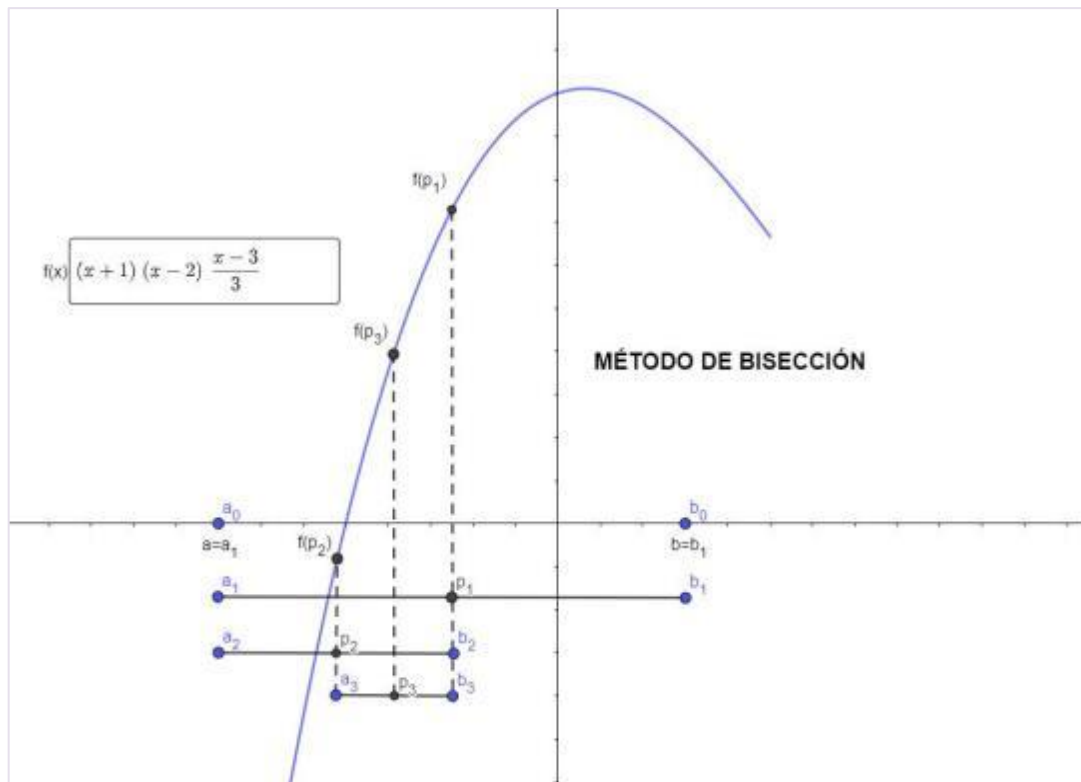
- Ejemplo:

○ $X + 4 = 11$ $3y = 75$ $m - 12 = 90$ $n/7 = 8$

1.2.1. Método de bisección

Este método se basa en el teorema del valor intermedio y parte de la preposición $f(a)$ y $f(b)$, que tienen signos opuestos, tomando en cuenta que el procedimiento funciona bien para los casos en los que existe más de una solución dentro del intervalo $[a, b]$. El método consiste en dividir por la mitad repetidamente los subintervalos de $[a, b]$, y en cada iteración localizar la mitad que contendrá la solución p .

Figura 1. Representación gráfica del algoritmo del método de bisección



Fuente: elaboración propia, empleando software GeoGebra.

1.2.1.1. Algoritmo de bisección

Se detalla el algoritmo de bisección representando el conjunto de pasos para llegar al final que proporciona la respuesta al problema planteado:

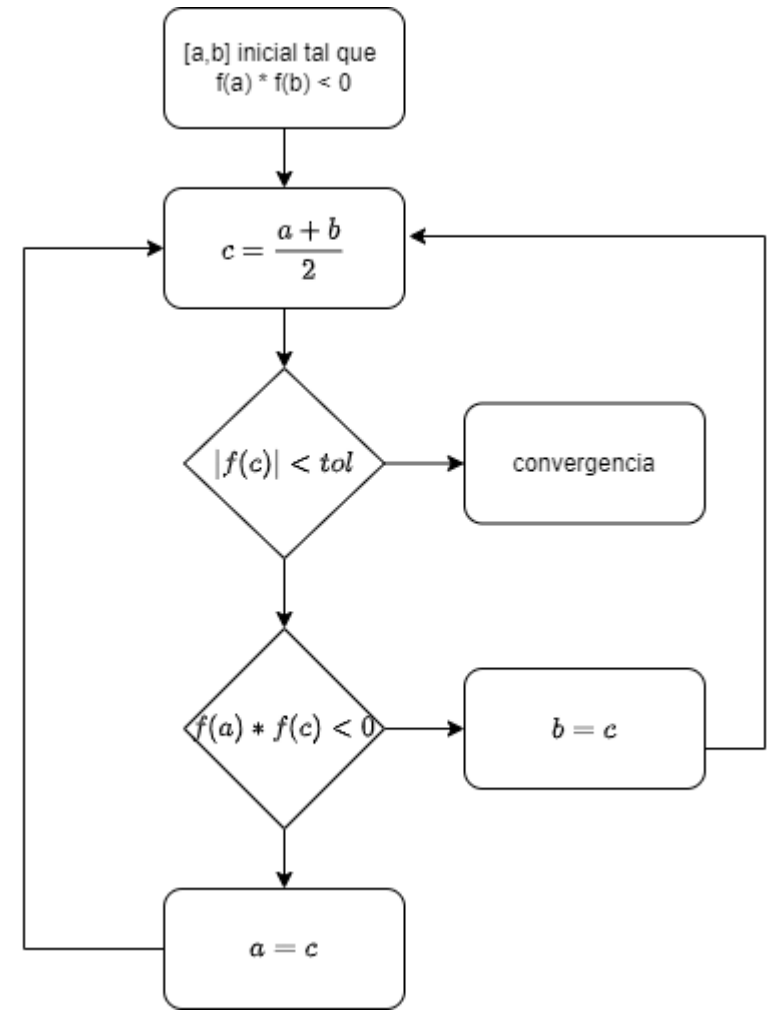
- Paso 1: $f(a)$ y $f(b)$ de signos opuestos
- Paso 2: se calcula la aproximación a la raíz $P = \frac{(a+b)}{2}$
- Paso 3: se calcula el cambio del intervalo
 - Si $f(a) * f(p) > 0$ cambia el valor de [a] por el de [p] en la siguiente iteración se utilizará [p , b].
 - Si $f(a) * f(p) < 0$ cambia el valor de [b] por el de [p] en la siguiente iteración se utilizará [a , p].

- Paso 4: se calcula el error

$$E = \frac{(b - a)}{2}$$

- Paso 5: cálculo del error < tolerancia
 - Si $error < tolerancia$, se encuentra la raíz con el número de cifras que se especificaron.
 - Si $error < tolerancia$, se debe regresar al paso 3 para cambiar el intervalo, y lo que sigue es iniciar otra iteración hasta que el error sea menor a la tolerancia $error < tolerancia$.

Figura 2. Diagrama de flujo que da solución al método de bisección



Fuente: elaboración propia, empleando software Draw.io

1.2.2. Método de punto fijo

El método de punto fijo aplicada a una función w , representa a un número “ q ” tal que $w(q) = q$. El problema que representa es poder encontrar las soluciones de una ecuación $f(x) = 0$ y encontrar los puntos fijos de una función $w(q)$, estos serán equivalentes en los siguientes casos: siguiendo con el

problema de encontrar el conjunto de soluciones de una ecuación $f(x) = 0$, se puede establecer una función w con un punto fijo p .

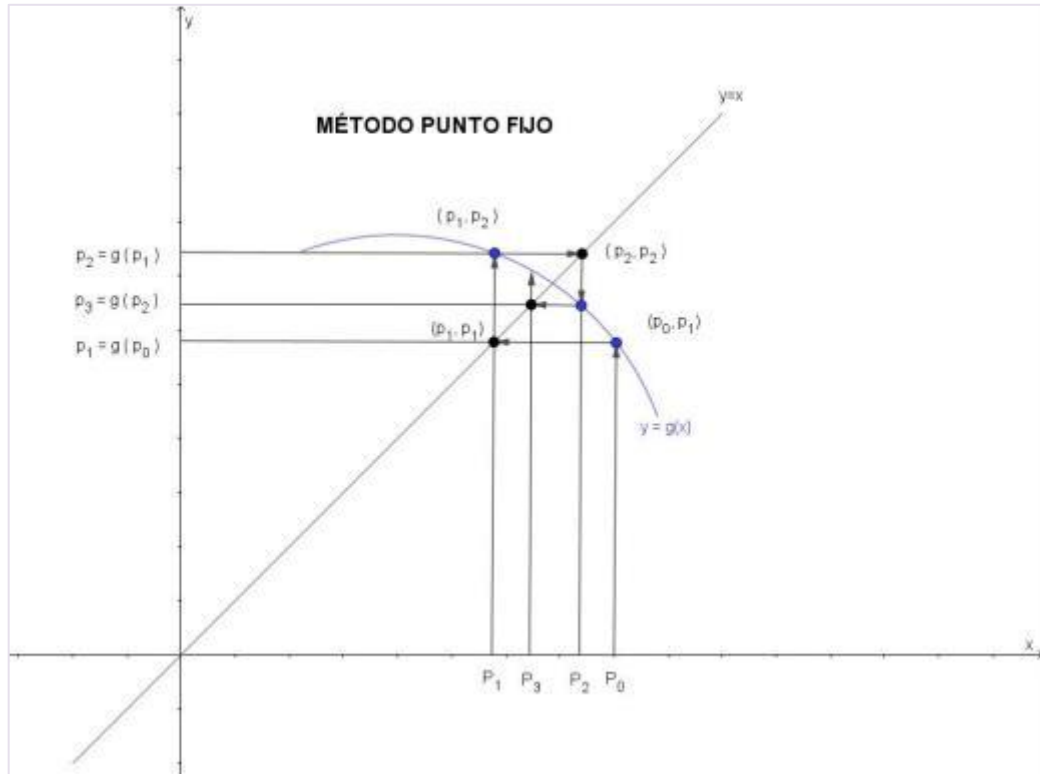
El método de punto fijo se utiliza junto a una ecuación de la siguiente forma $w(x) = x$. El primer paso es definir un valor aleatorio x_0 y luego se aplica la fórmula $x_{n+1} = w(x)$ en el caso que se encuentre $n \geq 0$.

Si existiera $\lim_{n \rightarrow \infty} x_n = q$, entonces si w es continua en ese valor q :

$$\lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} w(x_{n-1}) = w\left(\lim_{n \rightarrow \infty} x_{n-1}\right) = g(p)$$

Con lo antes descrito, se obtendrá una solución a lo que se está planteando.

Figura 3. Solución grafica del método de punto fijo



Fuente: elaboración propia, empleando software GeoGebra.

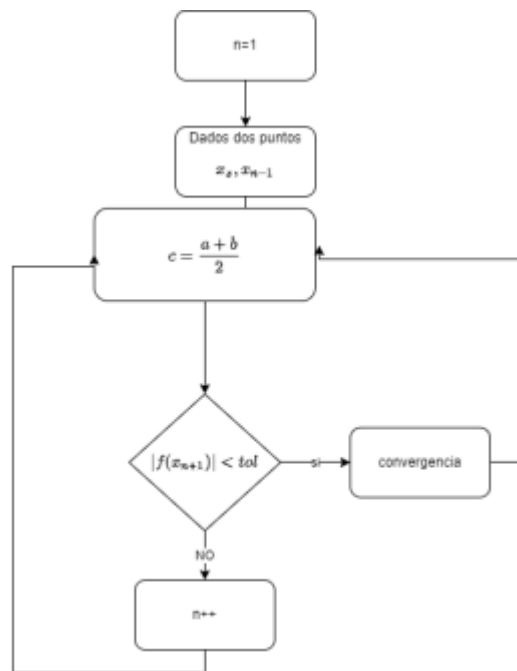
1.2.2.1. Algoritmo de punto fijo

Se detalla el algoritmo de punto fijo, siguiendo una secuencia de pasos ordenados para obtener el resultado esperado.

- Punto 1: se debe proporcionar la función a la cual se debe encontrar la raíz.
- Punto 2: se debe proporcionar un valor inicial aleatorio x_0

- Punto 3: con base en la función $f(x) = 0$, se debe despejar “x”, buscando una nueva función de “x” que será llamada $g(x)$.
- Punto 4: el siguiente paso es derivar la función $g(x)$
 - Si el valor absoluto de la derivación de $g(x)$ es menor a uno, se asegura que el despeje realizado funcione.
- Punto 5: lo siguiente es evaluar $g(x)$, utilizando el valor inicial x_0 . El resultado obtenido se convierte en el nuevo valor de “x” y así sucesivamente hasta que se encuentre la raíz que se está buscando, tomando en cuenta el factor de error.

Figura 4. **Diagrama de flujo que da solución al método de punto fijo**



Fuente: elaboración propia, empleando software Draw.io.

1.2.3. Método de Newton

El método de Newton es una técnica numérica para resolver un problema de búsqueda de raíces $f(x) = 0$ mas efectivas y conocidas. Existen varias maneras de implementarla, y la más común es tomarla de la manera gráfica. Otra solución consiste en derivar como una técnica que permite lograr una convergencia más rápida que la que ofrece otros tipos de iteración funcional.

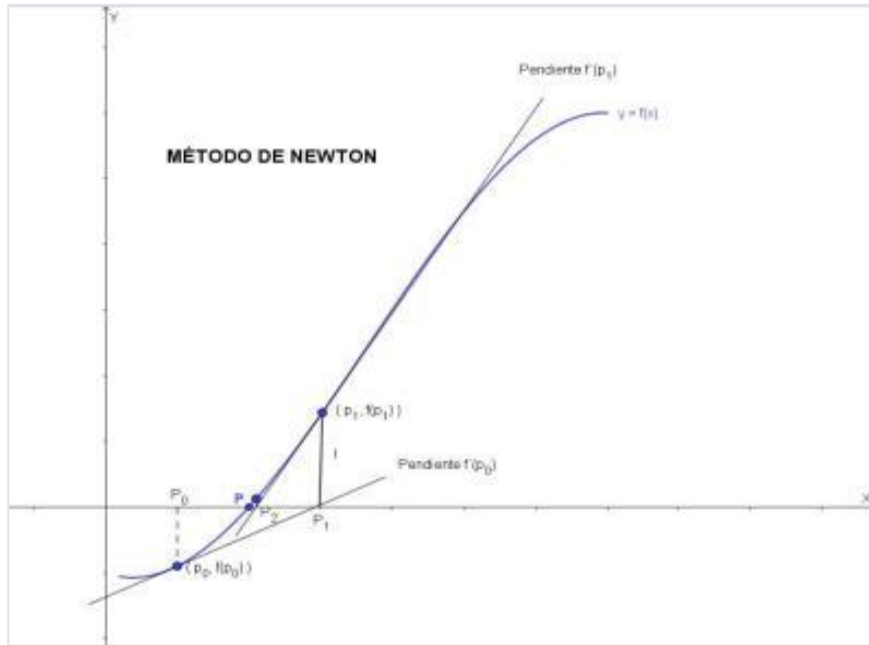
Este método se basa tomando un punto cercano a la raíz y el corte del eje de las "x" de las rectas tangentes sucesivas.

Suponiendo que se tiene una función cualquiera $f(x)$, y un punto x_0 , por consiguiente, con el punto $[x_0, f(x_0)]$ y la pendiente que se obtiene al sacar la primera derivada $f'(x)$. Con estos datos se puede calcular la primera recta tangente en ese punto.

Dándole seguimiento al método Newton-Raphson es importante tomar en cuenta lo siguiente:

- Este no trabaja con intervalos donde no se encuentra la raíz
- El método en algunas ocasiones no converge, sino que se recae en un ciclo infinito.
- Existen casos en los cuales el procedimiento efectuado no proporciona una raíz que se acerca a la raíz buscada.

Figura 5. Solución grafica del método de Newton



Fuente: elaboración propia, empleando software GeoGebra.

1.2.3.1. Algoritmo del método Newton- Raphson

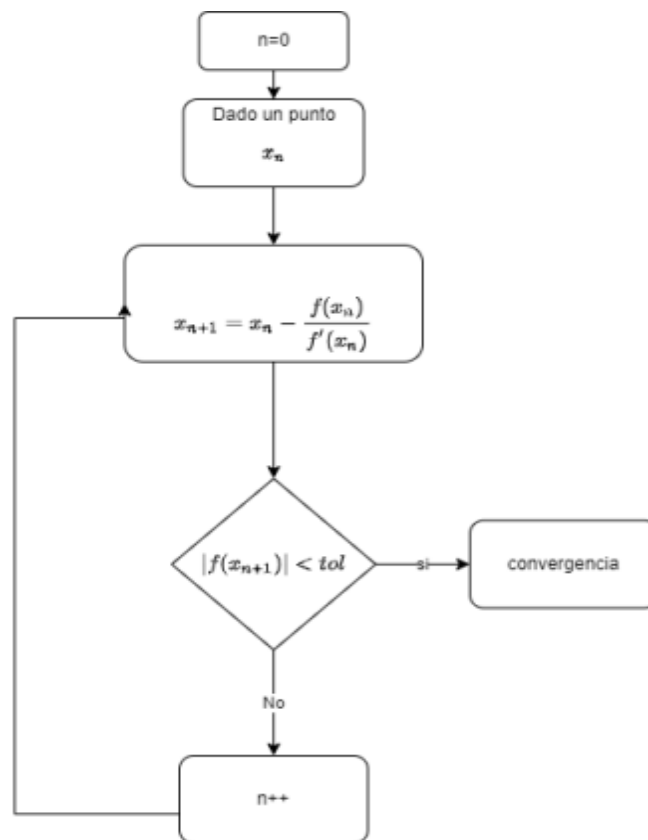
Se describe la serie de pasos ordenados para obtener los factores de tolerancia.

- Paso 1: contar con un punto inicial cercano a la raíz x_0
- Paso 2: calcular $f(x)$
- Paso 3: calcular la aproximación a la raíz por el corte con el eje de las "x" de la recta tangente.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Paso 4: se calcula el factor de error $|x_n - x_{n+1}|$
- Paso 5: según los datos obtenidos y tomando el factor de tolerancia
 - Si el error $<$ tolerancia entonces se encontró la raíz con el número de cifras consecutivas especificada.
 - Si el error $>$ tolerancia entonces se regresa al paso 3 para cambiar punto x_{n+1} y luego se inicia otra iteración hasta que error $<$ tolerancia.

Figura 6. **Diagrama de flujo del método de Newton- Raphson**



Fuente: elaboración propia, empleando software GeoGebra.

1.2.4. Método de la secante

El propósito de este método es poder eliminar el problema que presenta la derivada de la función del método de Newton, ya que en la vida real hay fenómenos físicos que influyen en las funciones, donde la derivada es muy compleja.

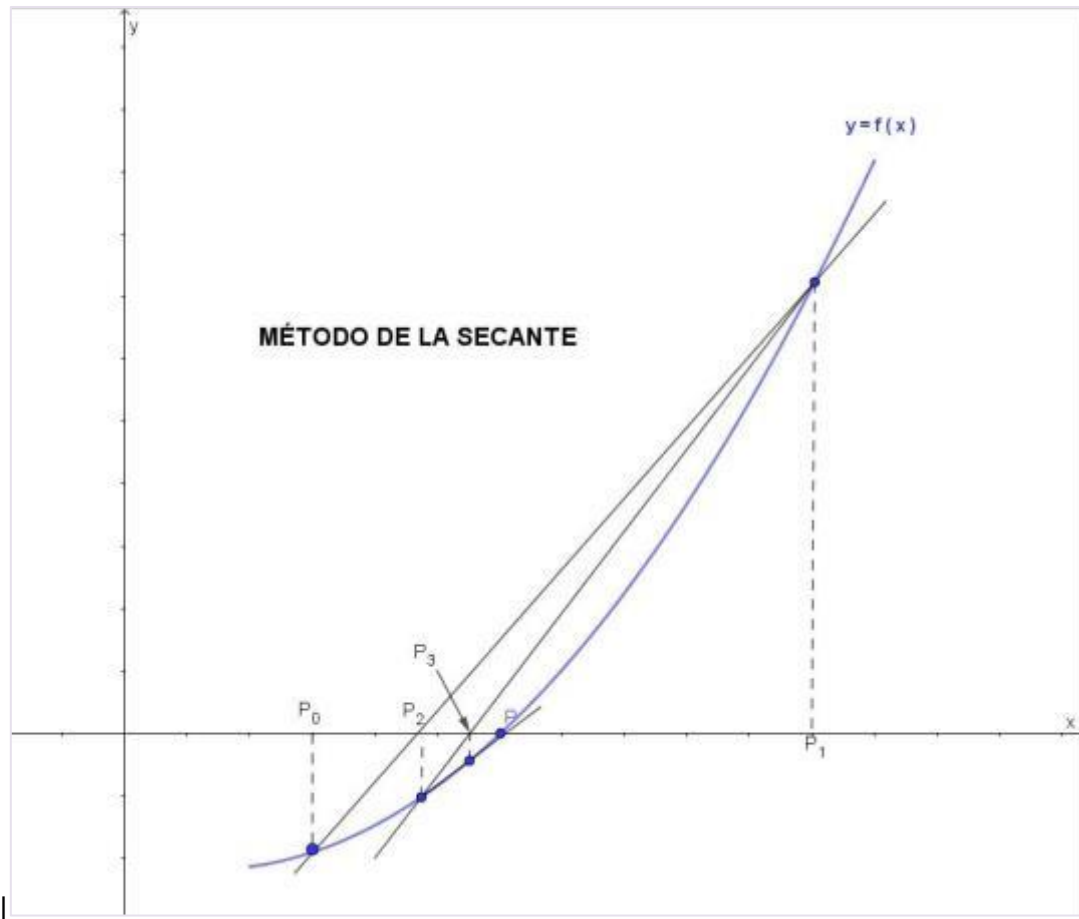
El método de Newton y el de la secante son muy similares pero su diferencia principal está en que el método de la secante no requiere de la primera derivada.

El problema principal del método de Newton radica en que requiere conocer el valor de la primera derivada de la función en un punto. Pero en ocasiones la forma de la función $f(x)$, puede dificultar el cálculo de la derivada. Por eso en estos casos lo mejor es emplear el método de la secante.

La diferencia principal con el método de bisección y Newton, casi nunca se le encuentra falla, solamente requiere 2 puntos al principio, y luego el mismo método se va retroalimentando. Gráficamente se estarían trazando rectas tangentes a la curva de la ecuación que se tiene originalmente y se va a ir verificando la intersección de las rectas con el eje de las X para ver si es la raíz que se está buscando.

Este método funciona por medio de dos puntos sobre el eje X, dentro del intervalo $x = (x_{n-1}, x_n)$, los cuales se evalúan en la función para sacar los puntos correspondientes en el eje de la y, los puntos que se estarían localizando son $(f(x_{n-1}), f(x_{n-1}))$, las coordenadas de los puntos que interceptan a la función son $(x_{n-1}, f(x_{n-1}))$ y $(x_n, f(x_n))$.

Figura 7. **Representación gráfica del método de la secante con rectas tangentes**



Fuente: elaboración propia, empleando software GeoGebra.

1.2.4.1. Algoritmo del método de la secante

Se describe la serie de pasos ordenados para encontrar las raíces tomando en cuenta los factores de tolerancia.

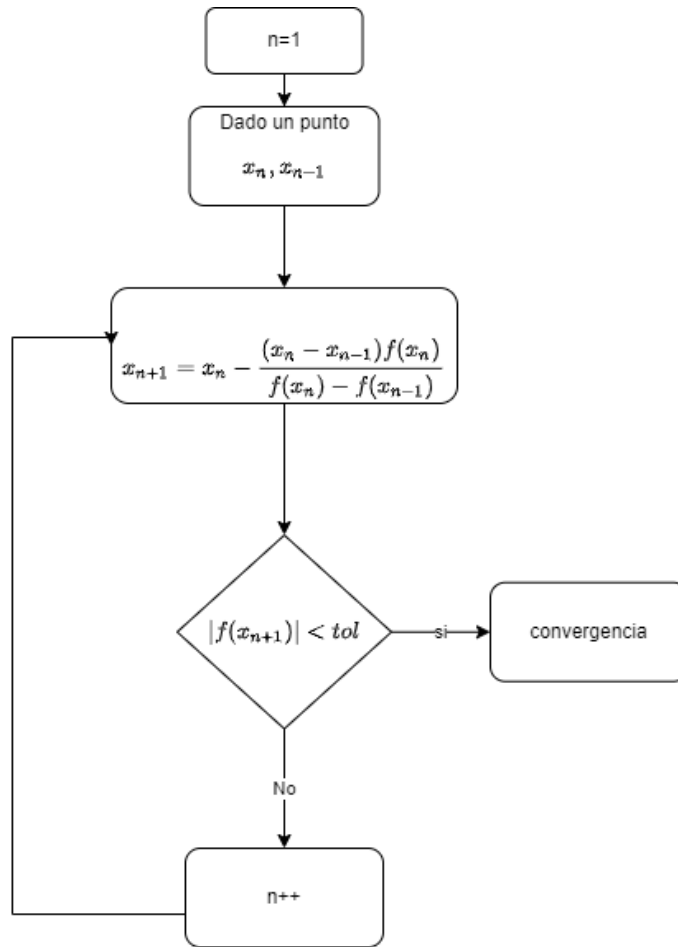
- Paso 1: contar con un punto inicial que encierre a la raíz (x_{n-1}, x_n)

- Paso 2: calcular la aproximación a la raíz por el corte con el eje de las “x” de la recta secante.
 - $$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

- Paso 3: el cálculo del error del método
 - Error = $|x_n - x_{n-1}|$

- Paso 4: según los datos obtenidos y tomando el factor de tolerancia
 - Si el error < tolerancia entonces se encontró la raíz con el número de cifras consecutivas especificada.
 - Si el error > tolerancia entonces se regresa al paso 3 para cambiar punto x_{n+1} y luego se inicia otra iteración hasta que error < tolerancia.

Figura 8. Diagrama de flujo para el método de la secante



Fuente: elaboración propia, empleando software Draw.io.

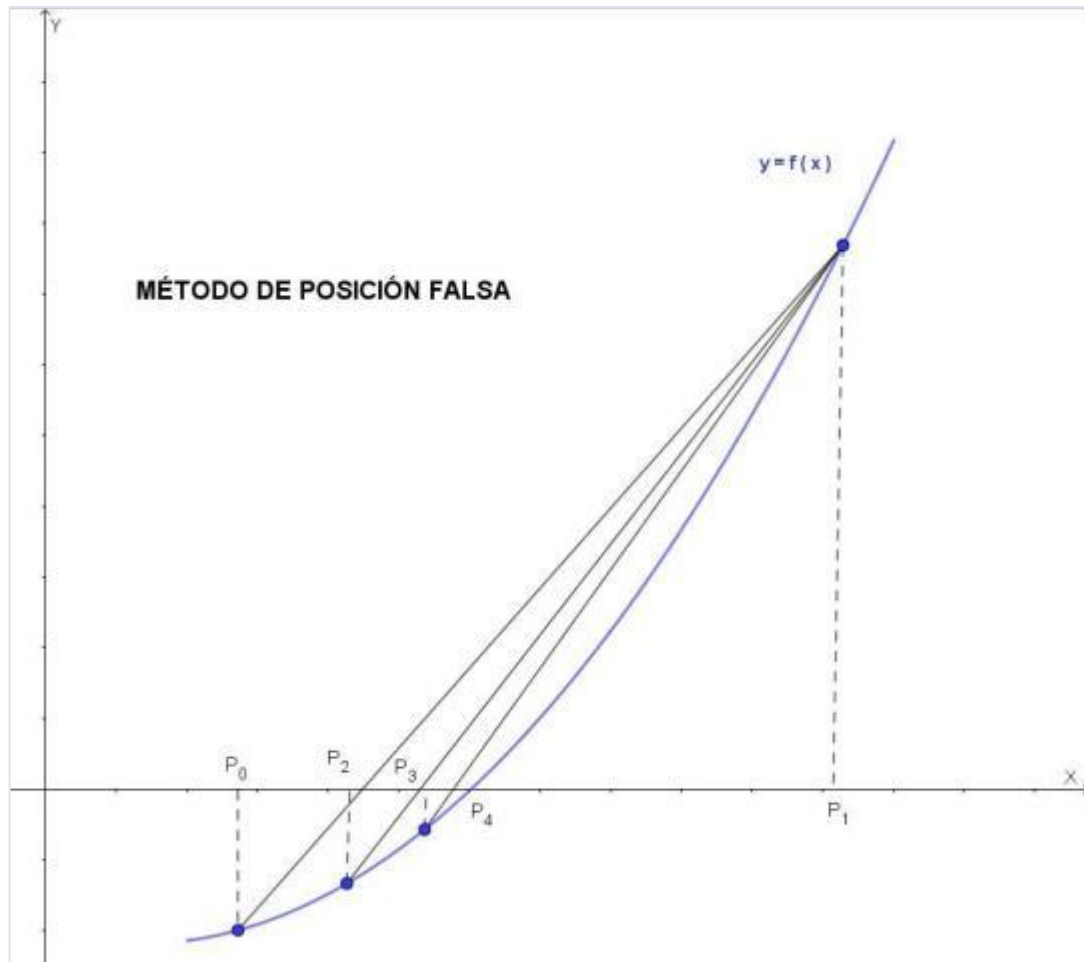
1.2.5. Método posición falsa o regla Farsi

El método de la posición falsa pretende conjugar la seguridad del método de la bisección con la rapidez del método de la secante. Este método, como en el método de bisección, parten de dos puntos que están rodeando a la raíz.

Este método es superior al método de la secante, es capaz de impedir la posibilidad de una divergencia del método. Por otro lado, mejora notablemente la elección del intervalo, no solamente se limita a dividir el intervalo a la mitad. Una desventaja es que el método de la posición falsa tiene una convergencia muy lenta hacia la solución. Una vez que se haya iniciado el proceso iterativo, uno de los extremos del intervalo tiende a no modificarse.

Este método funciona por medio de dos puntos y dentro de un intervalo en "x" (x_{n-1}, x_n) , donde se evalúan en la función para obtener puntos correspondientes en el eje de la y , los puntos que se deben obtener son $(f(x_{n-1}), f(x_n))$, por lo tanto, las coordenadas de los puntos que interceptan a la función son $(x_{n-1}, f(x_{n-1}))$ y $(x_n, f(x_n))$.

Figura 9. **Gráfico con la representación de cómo trabaja el método de la posición falsa**



Fuente: elaboración propia, empleando software GeoGebra.

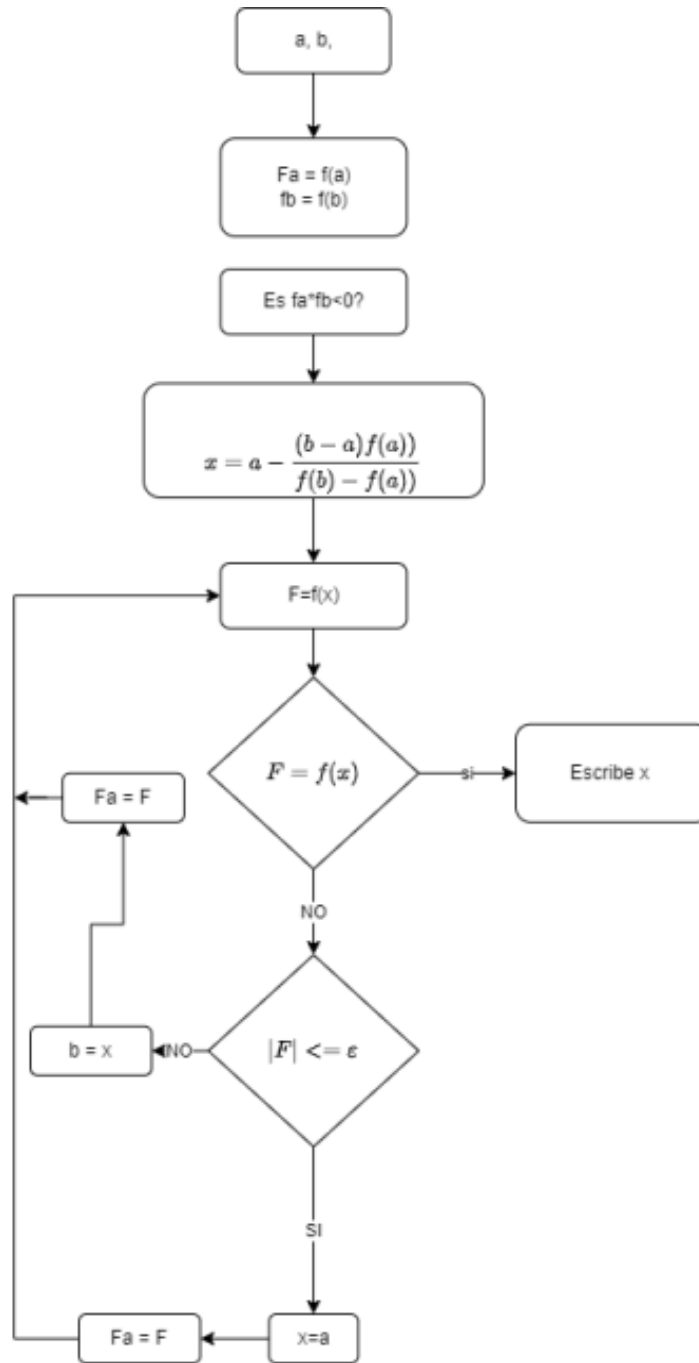
1.2.5.1. Algoritmo del método de posición falsa

Se describe una serie de pasos para encontrar las raíces tomando en cuenta el factor de tolerancia que se aplica al método de posición falsa.

- Paso 1: contar con un punto inicial que encierre a la raíz (x_{n-1}, x_n)
- Paso 2: calcular la primera aproximación a la raíz por el corte con el eje de las “x” por el Método de la Secante.
 - $$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$
 - $$\text{Error} = |x_n - x_{n-1}|$$
- Paso 3: del resultado de la segunda iteración “n” se debe iterar por el método de la posición falsa con la misma fórmula de la secante y dejar fijo el punto (x_{n-1}, x_n) , luego se calcula el error del método $|x_n - x_{n-1}|$
- Paso 4: para los dos pasos 2 y 3 se calcula el error < tolerancia.
 - Si el error < tolerancia entonces se encontró la raíz con el número de cifras consecutivas especificada.

Si el error > tolerancia entonces se regresa al paso 3 para cambiar punto x_{n+1} y luego se inicia otra iteración hasta que error < tolerancia.

Figura 10. Diagrama de flujo para el método de posición falsa



Fuente: elaboración propia, empleando software GeoGebra.

1.2.6. Método de Steffensen

Este método está basado en una función $x = g'(x_0) < 1$ por ende NO ES NECESARIO CALCULARLO, se puede mejorar su comportamiento respecto a la rapidez de convergencia.

Sin usar para ello ninguna derivada, el método de Steffensen proporciona una convergencia cuadrática en la localización de un punto fijo de una función real. Este método puede ser considerado como una simplificación del método de Newton, la diferencia radica en que este método empieza con dos aproximaciones por el método de punto fijo.

Suponiendo que $\{P_n\}_{n=\infty}$ es una sucesión linealmente convergente con un límite de valor p . Luego se supone que los signos de las aproximaciones son: $\{P_n - P\}$, $P_{n+1} - P$, $P_{n+2} - P$ son iguales que “n” es suficientemente grande como para que:

$$\frac{P_{n+1} - P}{P_n - P} = \frac{P_{n+2} - P}{P_{n+1} - P}$$

Por lo tanto:

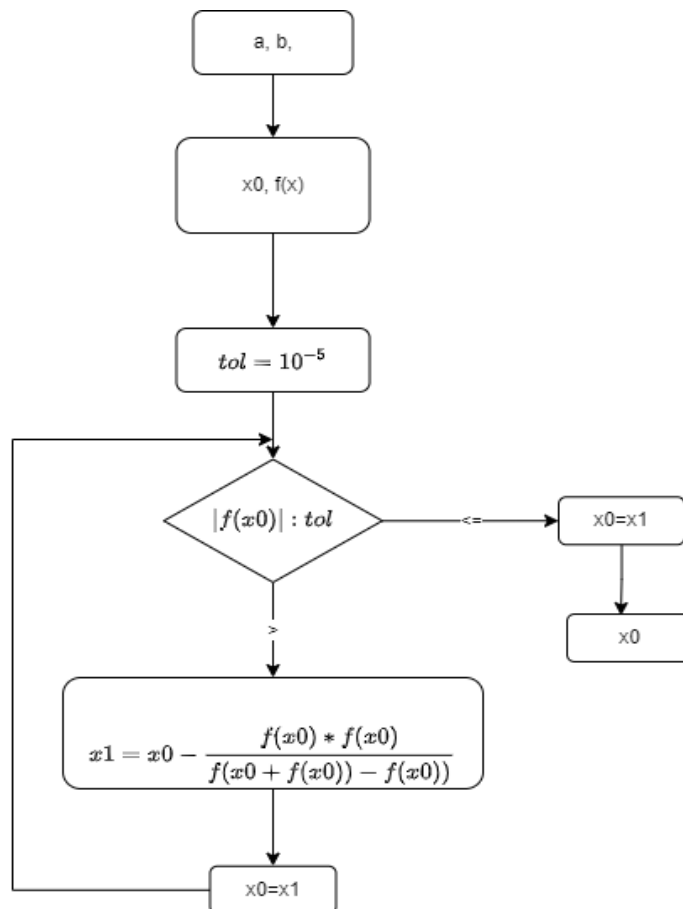
$$(P_{n+1} - P)^2 = (P_{n+2} - P)(P_n - P)$$

Transponiendo al lado izquierdo los términos que contiene “p” se tiene:

$$(P_{n+1}^2 - 2P_{n+1}) * p = P_{n+2} - P_{n+1}^2$$

- Paso 5: se procede a obtener el cálculo del Error = $|P_0 - P_{N+2}|$
 - Si el error < tolerancia, entonces se puede decir que se encontró la raíz con el número de cifras consecutivas especificadas.
 - Si el error > tolerancia, se tendrá que regresar al paso 4 para cambiar el valor de P_0 por el valor de " P_{n+2} " para luego iniciar otra iteración hasta que Error < Tolerancia

Figura 11. Diagrama de flujo para el método de Steffenson



Fuente: elaboración propia, empleando software GeoGebra.

1.2.7. Método de Muller

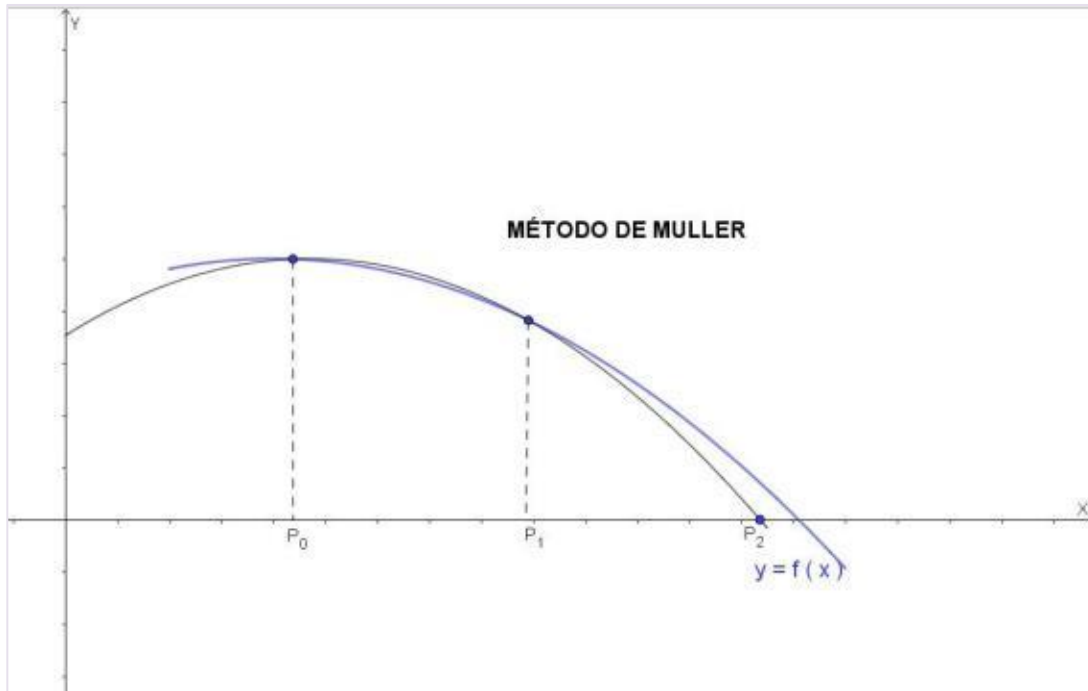
Un polinomio de grado "n" tiene la siguiente forma: $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 + a_0$, si P(x) es un polinomio de grado "n" mayor o igual que 1 con coeficientes reales o complejos, por lo tanto, $P(x) = 0$; tiene por lo menos una raíz que puede ser posiblemente complejas. Si P(x) es un polinomio de grado "n" mayor o igual que 1 con coeficientes reales o complejos, entonces existen constantes únicas $x_1 + x_2 + x_3 \dots x_k$ posiblemente complejas y enteros positivos $m_1 + m_2 + m_3 \dots m_k$ tales que:

- $\sum_{i=1}^k m_i = n$
- $P(x) = a_n (x - x_1)^{m_1} (x - x_2)^{m_2} (x - x_3)^{m_3} \dots (x - x_k)^{m_k}$

Esto establece que el conjunto de ceros de un polinomio es único y que si cada cero X_i se cuenta el mismo número de veces que su multiplicidad m_i , entonces un polinomio de grado "n" tendrá exactamente "n" ceros.

El método de Muller utiliza tres aproximaciones iniciales x_0, x_1 y x_2 (siendo estos valores aproximaciones a una de las raíces) y esto determina la siguiente aproximación x_3 al considerar la intersección del eje "x" con la parábola que atraviese $(x_0, f(x_0)), (f(x_1))$ y $(x_2, f(x_2))$. La deducción del Método de Muller comienza considerando el polinomio cuadrático.

Figura 12. **Gráfico con la representación de cómo trabaja el método de Muller**



Fuente: elaboración propia, empleando software GeoGebra.

1.2.7.1. Algoritmo del método de Muller

Se detalla la serie de pasos para encontrar la raíz que corresponde a la solución del método de Muller.

- Paso 1: se encuentra h_1 y h_2 , aplicando las ecuaciones descritas anteriormente.

$$h_1 = x_1 - x_0$$

$$h_2 = x_2 - x_1$$

- Paso 2: se encuentran las r_1 y r_2

$$r_1 = \frac{f(x_1) - f(x_0)}{h_1}$$

$$r_2 = \frac{f(x_2) - f(x_1)}{h_2}$$

- Paso 3: se procede a encontrar el parámetro d

$$d = \frac{r_2 - r_1}{h_1 + h_2}$$

- Paso 4: luego se encuentra el parámetro b

$$b = r_2 + h_2(d)$$

- Paso 5: se aplica el siguiente criterio

- Si $|b - D| < |b + D|$

- $E = b + D$

- Si no

- $E = b - D$

- $h = \frac{-2 * f(x_2)}{E}$

- $p = x_3 = raiz = x_2 + h$

- Paso 6: se encuentra el error
 - $\text{error} = |h| < \text{tol}$

- Paso 7: se realizan las sustituciones correspondientes
 - $x_0 = x_1$
 - $x_1 = x_2$
 - $x_2 = p = x_3 = \text{raiz}$

1.2.8. Polinomios de Lagrange

En algunas ocasiones, no se cuenta con una función continua sino más bien valores de la función específicos $y(x)$, para una x dada. Estas funciones son conocidas como funciones tabulares.

La interpolación consiste en hallar un dato $f(x)$, dentro de una serie de pares de puntos dentro del plano. Es importante tomar en cuenta que este próximo a uno de sus extremos, puesto que de otra forma no es muy fiable el resultado obtenido.

El polinomio de interpolación de Lagrange se plantea como:

$$P(x) = y_0l_0 + y_1l_1 + y_2l_2$$

Donde los polinomios de Lagrange l_i pertenecen a la tabla de datos y $i \neq 0$. Como se debe satisfacer que $P(x_i) = y_i$, esto cumple solo si $l_0(x_0) = 0$ para toda $i \neq 1$ y así sucesivamente.

Por lo tanto, el polinomio de Lagrange queda definido como:

$$P_i(x) = \sum_{i=0}^n f(x_n) \prod_{i \neq j} \frac{(x - x_j)}{(x_j - x_i)}$$

Desde un punto de vista matricial, se puede ver este polinomio de Lagrange así:

$$P_n(x) = [B][N]^{-1} * [f(x_n)]$$

B = bases de la interpolación o grado del polinomio que se quiere

N = Nodos de la interpolación o puntos del eje de las "x"

1.2.8.1. Algoritmo del método de polinomios de Lagrange

- Para polinomio de grado 1 (se necesitan 2 puntos para formular la recta).

$$P(x) = l_0 f(x_0) + l_1 f(x_1)$$
$$P(x) = \left(\frac{x - x_1}{x_0 - x_1} \right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0} \right) f(x_1)$$

- Para polinomio de grado 2 (se necesitan 3 puntos para formular la parábola).

$$P(x) = l_0 f(x_0) + l_1 f(x_1) + l_2 f(x_2)$$

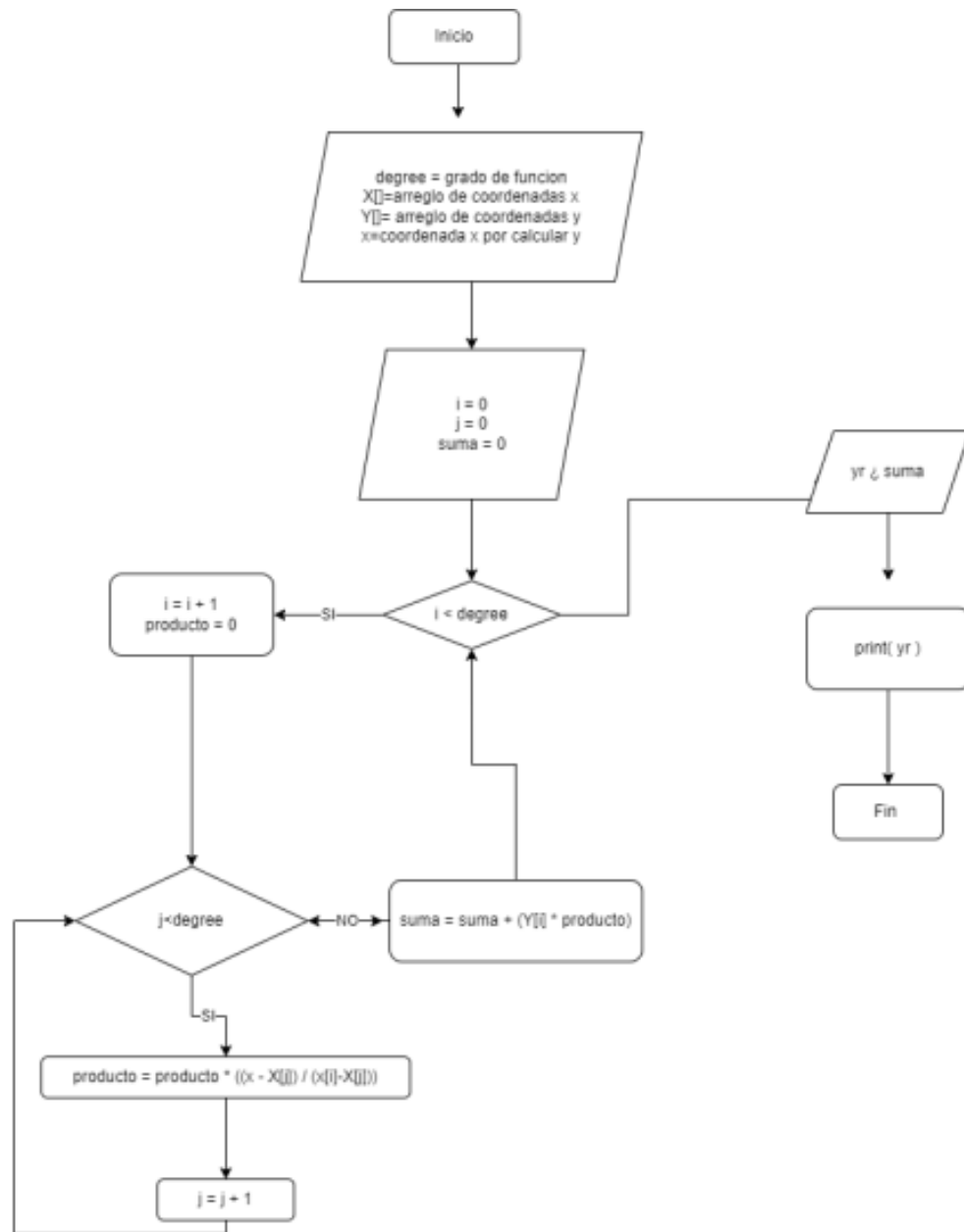
$$P(x) = \left(\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} \right) * f(x_0) + \left(\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \right) * f(x_1) \\ + \left(\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \right) * f(x_2)$$

- Para polinomio de grado 3(se necesitan 4 puntos para formular la parábola).

$$P(x) = l_0f(x_0) + l_1f(x_1) + l_2f(x_2) + l_3f(x_3)$$

$$p(x) = \left(\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} \right) * f(x_0) + \left(\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \right) \\ * f(x_1) + \left(\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} \right) * f(x_2) \\ + \left(\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} \right) * f(x_3)$$

Figura 13. Diagrama de flujo para el método de polinomios Lagrange



Fuente: elaboración propia, empleando software Draw.io.

1.2.9. Método de aproximación de Neville

Este analiza otro método de interpolación y extrapolación, también necesita pares de puntos dentro del plano para poder determinar un valor $f(x)$ que se encuentre dentro de una serie de pares de puntos en el plano. Esta extrapolación consiste en encontrar un dato $f(x)$ que se encuentre fuera de una serie de pares de puntos. Por ello es importante tomar en cuenta que esté próximo a uno de sus extremos.

Este método de aproximación se comporta de la siguiente forma:

Tabla I. Método de aproximación de Neville

X	F(x)	Columna 1	Columna 2	Columna 3
x_0	$y_0=p_0$			
x_1		$p_{0,1}$		
x_2	$y_1=p_1$			
x_3		$p_{1,2}$	$p_{0,1,2}$	
x_4	$y_2=p_2$			
.		$p_{2,3}$	$p_{1,2,3}$	$p_{0,1,2,3,4}$
.	$y_3=p_3$.
.		$p_{3,4}$	$p_{2,3,4}$.
	$y_4=p_4$.	.	.

Fuente: elaboración propia.

1.2.9.1. Algoritmo del método de Neville

Es un procedimiento que involucra la utilización de una interpolación polinómica contando con $n+1$ puntos, hay un polinomio único de grado $\leq n$ que pasa por los puntos dados.

El algoritmo de Neville evalúa este polinomio.

- Columna 1:

$$p_{0,1} = \frac{(x - x_1)y_0 - (x - x_0)y_1}{(x_0 - x_1)}$$

$$p_{1,2} = \frac{(x - x_2)y_1 - (x - x_1)y_2}{(x_1 - x_2)}$$

$$p_{2,3} = \frac{(x - x_3)y_2 - (x - x_2)y_3}{(x_2 - x_3)}$$

$$p_{3,4} = \frac{(x - x_4)y_3 - (x - x_3)y_4}{(x_3 - x_4)}$$

- Columna 2:

$$p_{0,1,2} = \frac{(x - x_2)p_{0,1} - (x - x_0)p_{1,2}}{(x_0 - x_2)}$$

$$p_{1,2,3} = \frac{(x - x_3)p_{1,2} - (x - x_1)p_{3,4}}{(x_1 - x_3)}$$

$$p_{2,3,4} = \frac{(x - x_3)p_{1,2} - (x - x_1)p_{3,4}}{(x_1 - x_3)}$$

- Column 3:

$$p_{0,1,2,3} = \frac{(x - x_4)p_{0,1,2} - (x - x_0)p_{1,2,3}}{(x_0 - x_3)}$$

$$p_{1,2,3,4} = \frac{(x - x_4)p_{1,2,3} - (x - x_1)p_{2,3,4}}{(x_1 - x_4)}$$

- Column 4:

$$p_{0,1,2,3,4} = \frac{(x - x_4)p_{0,1,2,3} - (x - x_0)p_{1,2,3,4}}{(x_0 - x_4)}$$

2. IDENTIFICACIÓN DEL PROBLEMA Y LA SOLUCION QUE LA APLICACIÓN PROPONDRA

Este capítulo tiene como propósito identificar de forma objetiva la problemática que se va estar abordando y la propuesta que vendrá a darle una solución efectiva, tomando en consideración los diferentes casos de uso.

2.1. Antecedentes

El desarrollo de las tecnologías en la última década, ha permitido tener la posibilidad de tener herramientas eficaces para el procesamiento de problemas matemáticos, los cuales hasta hace poco eran imposibles de resolver. Es importante que se entienda que el propósito de las primeras máquinas computacionales era darles solución a problemas complejos matemáticos, procesando una gran cantidad de información en un periodo corto. Actualmente existen herramientas educativas orientadas a la matemática, que permiten trabajar con una buena parte de esta, hasta un nivel universitario. Incluso cabe destacar la potencia para trabajar métodos matemáticos avanzados.

Según la investigación que se realizó, actualmente no existe una herramienta accesible que sea capaz de poder procesar y darle solución a los problemas que están contemplados en el pensum de Matemática Aplicada 3. Dentro de las herramientas que están orientados a la solución de problemas matemáticos, se encuentran las siguientes:

- GeoGebra
- Thatquiz

- Khanacademy
- F(x) Matemática

Estos son ejemplos de softwares o plataformas de matemáticas para todo nivel educativo. Estas herramientas abarcan temas relacionados con geometría, algebra, estadística, cálculo y análisis de ecuaciones. Otras plataformas como Khanacademy ofrecen un conjunto de lecciones que van enfocados a la docencia, brindando material para impartir en los salones de clase.

Tabla II. **Comparativo de herramientas que se asemejan a la aplicación**

La aplicación propuesta	Geogebra
<ul style="list-style-type: none"> • Se busca darle solución a los métodos y teoremas que se ven en la Matemática Aplicada 3 • Proponer una solución tomando en cuenta un numero de iteraciones que se puede configurar. • Se busca proporcionar una respuesta según el formato, raíces evaluadas y resultados en las tablas. • Se propone la opción de generar archivos PDF para descargar. 	<ul style="list-style-type: none"> • Cuenta con una alta gama de funciones y aplicaciones. Sin embargo, no trabaja específicamente teoremas que se ven en Matemática Aplicada 3. • La respuesta puede estar limitada por el algoritmo matemático que le da solución internamente. • Proporciona una amplia gama de aproximaciones representadas en gráficas, teoría, y explicación adicional. • No cuenta con algún medio por el cual se puedan descargar pdf.
La aplicación propuesta	Thatquiz
<ul style="list-style-type: none"> • Una aplicación que cuenta con una interfaz <i>responsive</i> e intuitiva para los usuarios. • No requiere ningún pago adicional, ni algún tipo de usuario. • Da solución a problemas complejos de Matemática Aplicada 3. 	<ul style="list-style-type: none"> • Esta aplicación está más enfocada al área docente de las materias, proporciona material y ejercicios. • Igualmente, no es de pago y tiene varios lenguajes al que esta traducido, ampliando su alcance • Da solución no solo a problemas matemáticos sino de varias ramas de la ciencia

Fuente: elaboración propia.

Dentro de las ventajas destacables para aplicación propuesta se encuentra una aplicación fácil e intuitiva para los estudiantes, que no requiere de un dispositivo específico para poder utilizarla, únicamente es necesario contar con red de internet para poder procesar los problemas, adicionalmente la opción de poder generar un reporte o tarea con los ejercicios que se realizaron, facilitando el proceso de las tareas, e incluso incentivar al estudiante a realizar sus tareas.

2.2. Usuarios potenciales de la aplicación

El objetivo es poder brindar una plataforma fácil e intuitiva para que los estudiantes del curso de Matemática Aplicada 3 sea capaz de darle solución a los problemas. El perfil serán estudiantes de la facultad de ingeniería o matemática, específicamente para los estudiantes que estén cursando Matemática Aplicada 3 y carreras afines que trabajen con los métodos que será capaz el aplicativo de darle solución. El nivel de conocimientos que se espera para poder optar a utilizar sería tener conocimientos básicos de Matemática Aplicada 3.

La accesibilidad es un factor que va a determinar el éxito de la aplicación, se tiene pensado lanzar esta aplicación por medio de una página web *responsive* que se pueda adaptar a cualquier dispositivo desde un teléfono inteligente, una *tablet* y una computadora. El público objetivo de la aplicación es:

- Los estudiantes que cursan Matemática Aplicada 3 en la Facultad de Ingeniería.

- Estudiantes con conocimientos de Cálculo Multivariable y fundamentos de Ecuaciones Diferenciales.
- Personas que tienen interés en la rama de la Matemática Aplicada 3.
- Personas que estén interesadas en demostrar métodos y resultados de aproximaciones.
- Personas que apliquen estos métodos para proyecciones matemáticas.
- Personas de escasos recursos que se les dificulte tener acceso a una calculadora potente o una computadora.

Conociendo el perfil del estudiante de Ingeniería en Ciencias y Sistemas, se entiende que es necesario una plataforma que pueda desplegarse tanto en un dispositivo como *laptop* o *tablet*, pero también sea compatible con un teléfono inteligente. Esto para ampliar la disponibilidad del estudiante y no infringir en gastos extras para poder hacer uso del aplicativo.

2.3. FODA

Se estará realizando un análisis FODA con el objetivo de demostrar la superioridad a nivel educa que otorga el uso de esta nueva aplicación en el Departamento de Matemáticas de la Universidad de San Carlos de Guatemala.

- Fortalezas
 - No existe competencia directa a la propuesta de la aplicación que se quiere desarrollar.

- Capacidad de procesamiento en la nube sin dependencias.
- El listado de métodos se adapta a las necesidades del estudiante que cursa el curso de Matemática Aplicada 3.
- Amplio conjunto de métodos matemáticos que brindan una solución concreta.
- Una interfaz de usuario amigable.
- No requiere inversión, es una herramienta gratuita.
- Redundancia de servicios.
- Oportunidades
 - Accesibilidad desde cualquier dispositivo
 - Alta demanda dentro del mercado objetivo
 - Tendencia entre los consumidores para desenvolverse mejor en entornos virtuales.
 - Herramienta innovadora.
 - El usuario requiere información precisa e inmediata.
- Debilidades
 - Esta aun en una versión beta y esto puede en algunas ocasiones provocar bugs en la aplicación.

- El Alcance está limitado al objetivo de este trabajo de investigación.
- En el largo plazo surgiría una necesidad de financiamiento.
- Amenazas
 - Nuevos competidores
 - Poca confianza de los consumidores para usar la aplicación
 - Canal de distribución puede ser comprometido

2.4. **Benchmark**

Comparador de indicadores claves para medir el rendimiento de la aplicación.

Tabla III. **Indicadores clave**

Aplicación	Disponibilidad	Alcance de funcionalidad	Interfaz de usuario amigable
ThatQuiz	X		
Geogebra	X	X	X
Khanacademy	X		
F(x) Matemática	X	X	X

Fuente: elaboración propia.

3. ARQUITECTURAS EN LA NUBE

Este capítulo tiene como objetivo dar a conocer los servicios en la nube y los proveedores. Además, también se da una definición de la arquitectura en la nube y los componentes que pueden ser parte de esta arquitectura. Junto a estas definiciones también se dará a conocer modos de despliegue en la nube y la orquestación de los servicios activos.

3.1. Arquitectura *serverless*

La propuesta de una arquitectura *serverless* ofrece la posibilidad de trabajar sin un servidor y es una forma de crear y ejecutar servicios sin tener que tomar en cuenta una infraestructura. La aplicación Propuesta aplicación es capaz de continuar ejecutándose, pero AWS se encargará de toda la administración de los servidores. Ya no compete acondicionar, escalar ni mantener los servidores para ejecutar las bases de datos y aplicativos. Por parte de AWS se encuentra la capa de AWS Lambda, que provee esta funcionalidad *serverless*, este es un servicio basado en eventos que permite ejecutar código en respuesta a eventos, todo esto sin gestionar ningún servidor.

3.1.1. Proveedores de arquitectura *serverless*

Hoy en día este tipo de tecnología ha empezado a encontrar un nicho del mercado y los proveedores de servicios en la nube han propuesto nuevas opciones, donde cada uno cuenta con beneficios. Dentro de los proveedores principales de funciones *serverless* se encuentran:

- AWS Lambda
 - Funciona bajo demanda
 - Integración con ecosistema en la nube
 - Capa gratuita

- Google Cloud Functions
 - Escalamiento automático según la carga
 - Provee funciones que integradas de depuración, registro y supervisión.
 - Posibilidad de integrar una red para casos híbridos o de múltiples nubes.

- IBM Cloud Functions
 - Precios Rentables por el tiempo de uso
 - Integración sencilla por medio de una API REST
 - Tiempos de respuesta sin memoria y sin coste

- Azure Functions
 - Infraestructura escalable automáticamente
 - Integrada a un modelo de programación que responde a eventos y al mismo tiempo conecta otros servicios.
 - Varios lenguajes de programación y servicios de hosting integrados para cada escenario que se pueda suscitar ajustado a cualquier tipo de negocio.

- FaaS (Functions as a Service)

Este modelo consiste en delegar la responsabilidad de ejecutar un código solamente cuando sea necesario, por lo tanto, se busca que se cobre el tiempo que la función estuvo activa. Esto evita el acondicionamiento y configuración de una infraestructura o servidor.

3.2. Microservicios

Se estará realizando una breve descripción de lo que son los microservicios, las ventajas y los casos dentro de la propuesta orientada a la nube.

3.2.1. Que se considera como microservicios

Este es un nuevo enfoque relacionado a la arquitectura y la organización que se utiliza para el desarrollo que está compuesto por pequeños servicios independientes que se comunican entre sí por medio de una API de manera independiente y autónoma. Como cada uno de los servicios trabajan independientemente se tiene la posibilidad de trabajar distintos lenguajes de programación según las necesidades que se tengan. Esta arquitectura está del lado del servidor y están enfocadas hacia un numero grande de usuarios y volumen de información.

- Facilidad para escalar, actualizar, testear y realizar el *deploy*
- Altamente disponible
- Descentralizar funciones
- Diversificación de servicios

3.2.2. Arquitectura de microservicios

Tomando en cuenta el contexto de una arquitectura orientada a microservicios se puede determinar una descomposición en subsistemas que puedan trabajar de manera autónoma en forma de microservicios de colaboración y contenedores, donde un microservicio es un contenedor.

Este enfoque indica una implementación de un conjunto de funciones que están estrechamente relacionados. Estos microservicios se desarrollan e implementan como contenedores de forma independiente entre ellos. Cada microservicio tiene su propia base de datos, y esto permite tener más independencia entre ellos.

3.2.3. Ventajas de utilizar microservicios

- Escalabilidad
- Funcionalidad modular y módulos independientes
- Libertad del desarrollador de desarrollar y desplegar servicios de forma independiente.
- Uso de contenedores permitiendo el despliegue y el desarrollo de la aplicación rápidamente.

3.3. Arquitectura SOA

En esta sección se analizará cómo la arquitectura en la nube, y la arquitectura orientada a servicios coexisten. Existe una sincronización entre ambos modelos se explicará como la combinación de estas arquitecturas optimiza e incrementa el alcance, a diferencia de implementaciones individuales.

En la actualidad el propósito de las de las implementaciones basadas en la nube (*cloud computing*), es optimizar las aplicaciones a gran escala. Las aplicaciones deben ser flexibles para que al introducir una arquitectura basada en servicios este pueda ser de bajo acoplamiento. También es importante la escalabilidad, que es una habilidad importante para la sinergia a largo plazo de estas tecnologías.

Dentro de las soluciones que propone un tipo de solución en la nube se encuentran infraestructuras de terceros como Amazon EC2, Google Cloud, entre otros. Esto puede representar una solución total o parte de una arquitectura, una diferencia remarcable es el hecho que los recursos del sistema serán mejor distribuidos.

SOA por otra parte es el proceso que define la solución de la arquitectura, SOA no viene a sustituir a la nube, sino que la mayoría de las soluciones que están basadas en la nube están definidas por SOA. SOA es un patrón de arquitectura, pero también es una estrategia que converge la tecnología y métricas del negocio, esto permite la posibilidad de por parte de la arquitectura para sufrir cambios, teniendo un bajo coste de implementación.

Uno de los ideales de SOA recae en la reutilización de la funcionalidad, y si bien es cierto la escalabilidad es un factor importante en una arquitectura *Cloud computing* es un factor que se debe tomar en cuenta.

- ¿Qué arquitectura se acopla más a este modelo?

Para la elección de la arquitectura, tomando en consideración que la implementación de funciones *serverless* aprovechando que AWS cuenta con una capa gratuita que proporciona una serie limitada de llamadas a las

funciones programables. Además del tema económico otra razón por la que se decidió basarse bajo el esquema de una arquitectura en la nube. La facilidad para consumir estos servicios y la integración con Java para el lenguaje de ejecución de la función *serverless*.

3.3.1. Modelo de servicios

El modelo de servicio más completo es aquél que ofrece el software y el hardware como un servicio conjunto, *software* como servicio o SaaS proporciona toda la infraestructura, software necesario para brindar un servicio completo.

Software as a Service se puede describir como un *software* que esta implementado y desplegado en un servidor en la nube, al que se puede acceder por internet. En este modelo, los usuarios, pagan por el uso del servicio mediante un tipo de suscripción que dependen de la plataforma que proporcione el servicio. Dentro de las características de este modelo se pueden mencionar:

- *Software* as a Service es capaz de soportar múltiples usuarios
- El usuario se desliga de todo lo relacionado a la parte operativa de la infraestructura física de los servidores que almacenan los servicios.
- El *software* cuenta una alta disponibilidad por internet.
- El proceso de facturación puede ser basado en el uso del *software* o por medio de una suscripción.

3.3.2. Modelos de despliegue *Cloud*

- *Cloud privada*: es un conjunto de tecnologías o datacenter que están disponibles para organizaciones o empresas, con infraestructura específicas en base a la demanda. Donde mediante la virtualización se pueden agregar nuevas características y beneficios de la nube tales como un nivel de elasticidad y provisionamiento adicional.

Este tipo de soluciones benefician un cierto grado de garantía de la información y seguridad de esta, al no compartir recursos se asegura la información. Otra característica importante es la capacidad de poder elegir el proveedor de los recursos tecnológicos que se adapten a las necesidades técnicas y económicas.

Este modelo de despliegue implica un costo elevado puesto que, a pesar de su flexibilidad y adaptabilidad, requiere una inversión no solamente inicial sino de mantenimiento. Es por esto que las nubes privadas están especialmente orientadas a organizaciones con una alta concentración de recursos y sistemas tecnológicos. Entre las características más importantes se encuentran:

- Mejora la operación IT interna
- Mejoras en costes mínimas
- Escalabilidad y flexibilidad limitadas
- Riesgo de obsolescencia
- gestión limitada de picos de demanda
- Proyectos a medida

- Cloud público: este tipo de implementación se caracteriza por ofrecer recursos TIC compartidas entre múltiples clientes. Estos recursos son accedidos mediante conexiones VPN. Este tipo de infraestructura cuenta con todas las ventajas de una nube. El usuario es capaz de automatizar el despliegue de máquinas virtuales e integrarlas fácilmente. Estas implementaciones tienen la capacidad de ser escalables y crecer o decrecer según las necesidades del tráfico hacia ese despliegue, esto hace que el cobro sea por el consumo real de los servicios que se usan.

Adicional a esto las nubes pueden complementarse con otros servicios que ofrezcan estas plataformas en la nube, como por ejemplo balanceadores de carga, servicios de *backup*, seguridad y otros servicios adicionales.

3.3.3. Orquestación de servicios

La orquestación de nubes se refiere a la coordinación de tareas automatizadas para racionalizar los flujos de trabajo. La orquestación se produce a nivel global, dando cuenta de cada elemento y proceso y consiguiendo que todo funcione conjuntamente por el flujo de trabajo completo y eficiente.

Se tiene contemplado para la propuesta de la infraestructura que se estará presentando la orquestación secuencial en la nube de la siguiente manera:

- Amazon Web Services
 - Solicitudes de los usuarios
 - manejo de las máquinas virtuales

- configuración de la red para los requerimientos de las aplicaciones de las VM's.
- Despliegue de las funciones Lambda para el procesamiento.

La orquestación en la nube obliga un flujo de trabajo que busca la automatización de las tareas, e implementa factores de seguridad con identificación de accesos para un mejor manejo de las políticas de acceso.

4. DOCUMENTACIÓN DE ESPECIFICACIONES, REQUERIMIENTOS Y CRITERIOS DE ACEPTACIÓN DE LA APLICACIÓN WEB

Este capítulo está dedicado a la documentación y definición de parámetros de aceptación para la propuesta de la aplicación web.

4.1. Objetivo estratégico

Innovar implementando servicios en la nube para colocando la infraestructura que se hará cargo de procesar los datos de los problemas matemáticos.

4.2. Objetivo general

Proporcionar una herramienta para que los estudiantes tengan acceso remoto al sistema mejorado para los estudiantes del curso de Matemática Aplicada 3.

4.3. Lista de acrónimos y glosario de la aplicación

- SaaS: permite a los usuarios conectarse a aplicaciones que se encuentran en la nube por la red o como sus siglas lo indican es un software como servicio.
- PaaS: es un entorno de desarrollo e implementación completo en la nube, esto incluye infraestructura como almacenamiento, servidores y

redes con canales asignados. Este es capaz de sostener un ciclo de vida completo de una aplicación web.

- Compilación
 - Pruebas
 - Implementación
 - Administración
 - Actualización
-
- FaaS: esta propuesta indica un modelo que consiste en poder delegar a un tercero la labor de ejecutar un bloque de código cuando sea necesario. El cobro por estos servicios se determina por el tiempo directo en el que se hace uso de la función.

 - HTML: este es un lenguaje de etiquetas de hipertexto que permite estructurar, crear y diseñar párrafos, encabezados, enlaces, elementos gráficos para páginas web.

 - Internet Gateway: es un componente de una VPC escalable, redundante y disponible que permite la comunicación entre la VPC y el internet.

4.4. Alcances

El alcance previsto para este trabajo de investigación es poder abarcar cada uno de los temas que se trabajan en las unidades que conforman el curso de Matemática aplicada 3.

4.5. Análisis de riesgos

Dentro de los riesgos que pueden llegar a perjudicar gravemente el aplicativo web están:

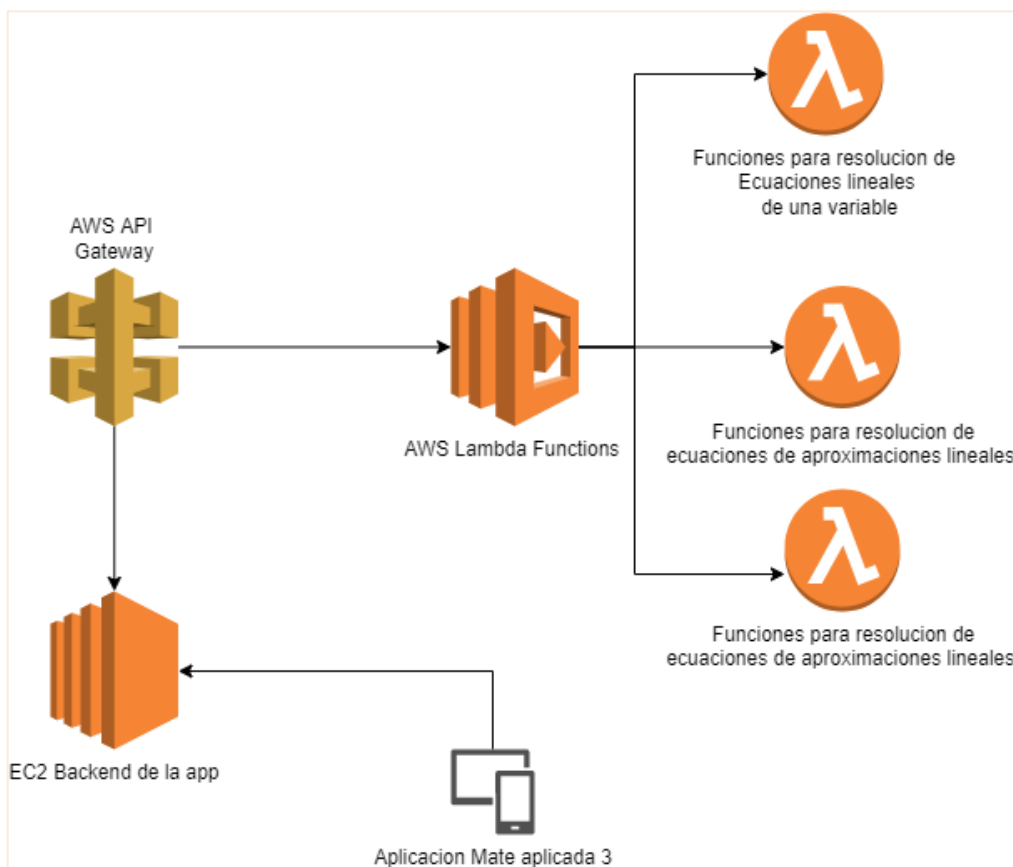
- Perdida de conexión a internet en medio de una transacción de información. El aplicativo necesita internet para conectarse al servicio en la nube que proporciona esta función *serverless* que procesa la función y le da una solución integrada. Se debe contar con una conexión estable a internet es vital, esa dependencia puede ser un riesgo que retrase el proceso de la resolución de un problema. Este riesgo, aunque no es tan probable si cabe dentro de la posibilidad y se propone, un tipo de aplicación de uso local que no necesite una conexión a la red activa.
- El aplicativo no se pueda dar a conocer en el mercado objetivo, por la situación actual, si el aplicativo no tiene un buen recibimiento no se podrán dar a conocer reseña relacionadas con la experiencia del usuario, incluso posibles *bugs* que no sean detectados y solucionados en la etapa de *testing*. Es muy importante poder conocer acerca de la experiencia de usuario. Que el consumidor tenga una buena experiencia de usuario y que pueda incluso recomendar el aplicativo con sus compañeros de estudio. Para mitigar este riesgo se propondría hablar con los catedráticos y que ellos proporcionen esta herramienta.
- Desconocimiento de algún detalle que no se haya tomado en cuenta, y la facilidad de poder dar un dato erróneo. Teniendo presente que las matemáticas son muy precisas, es importante analizar y comprobar cada uno de los teoremas matemáticos. Esto incluye los diferentes algoritmos en programación que le dan la solución a los teoremas y métodos. Para

mitigar este riesgo se busca garantizar que los algoritmos trabajen de una manera correcta.

4.6. Arquitectura de la solución

En la siguiente figura se presenta la arquitectura propuesta para la solución:

Figura 14. **Arquitectura propuesta para la solución**



Fuente: elaboración propia, empleando software Draw.io.

4.7. Descripción de módulos principales

El aplicativo tiene como objetivo, integrar una aplicación de fácil uso e intuitiva para que el usuario tenga una experiencia de usuario simple, pero adecuada. Para ello se estuvo trabajando en los siguientes módulos:

- Métodos para resolver ecuaciones de una variable
 - Método de bisección
 - Iteración de punto fijo
 - Método de Newton
 - Método de la Secante
 - Método de la posición falsa
 - Método de la Secante
 - Método de Muller

- Métodos para resolver ecuaciones multivariables
 - Polinomios de Lagrange
 - Método de Neville
 - Diferencias divididas de Newton
 - Método de Jacobi
 - Método de Gauss-Seidel
 - Punto fijo, varias variables
 - Newton, sistemas no lineales

4.8. Requerimientos

Lo necesario para poder hacer uso de la aplicación es poder contar con:

- Dispositivo inteligente (SmartPhone, *tablet*, computadora)
- Conexión a internet

El proceso es tan fácil como acceder al enlace que se proporcione y navegar por medio del menú, luego de seleccionar el método que se quiera procesar, se despliega el formulario que luego de ser llenado y dar clic en Calcular, esta da los resultados y una tabla con los resultados de las iteraciones.

5. PROCESO DE DESARROLLO DEL PROYECTO

Este capítulo tiene como objetivo dar a conocer las metodologías que se implementaron durante el proceso de desarrollo de la aplicación y la descripción de las fases del ciclo de vida de un proyecto.

5.1. Metodología ágil de trabajo

Es importante para el desarrollo de los proyectos poder implementar una metodología ágil de trabajo que agilice el desarrollo a medida del alcance que tendrá el mismo. Estas metodologías brindan un conjunto de técnicas que se aplican dentro de un entorno de trabajo orientadas a optimizar en gran medida todo el flujo de trabajo. Esto para garantizar un resultado final que cumpla con las expectativas del usuario final o cliente.

Para el proyecto que se está planteando es importante tomar en cuenta que la implementación de una metodología ágil debe ser aplicado desde el inicio del proyecto cuando se documentó el alcance y las historias de los usuarios haciendo énfasis en la necesidad de los estudiantes. Luego para la definición de las funciones que se necesitaban hasta llegar a la elección de los recursos informáticos que se iban a estar utilizando.

En este punto se define un presupuesto y tiempo de desarrollo que tomara el proyecto, para su posterior aprobación.

5.1.1. Manifiesto ágil

Es importante mencionar que para el desarrollo de este proyecto de investigación se aplicaron los principios del manifiesto ágil para el desarrollo de este proyecto. El manifiesto ágil se compone de los siguientes principios:

- Se tiene como principal objetivo satisfacer al cliente, haciendo entregas continuas de valor.
- Se tiene considerado que los requisitos puedan cambiar durante el transcurso del tiempo. Estos procesos ágiles aprovechan cada ventaja competitiva para beneficio del cliente.
- Entregas de software totalmente funcional junto a entregas continuas en periodos de tiempo que rondan en las dos semanas a dos meses.
- Los clientes y desarrolladores trabajan en conjunto durante todo el proyecto siempre teniendo un seguimiento.
- Los proyectos se basan bajo un esquema de individuos motivados, la misión es proporcionar un entorno y el apoyo que se necesita.
- El medio más eficiente para comunicación dentro de un equipo es la conversación cara a cara.
- La principal métrica de avance de un software es cuando es funcional.
- El propósito de los procesos ágiles es incentivar un desarrollo que pueda ser sostenible. Se debe ser capaz de sostener un ritmo constante.

- Proponer un buen diseño y excelencia referente a la parte técnica mejora la agilidad del proyecto.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Ser pragmático o la habilidad de maximizar la cantidad de trabajo no realizado es importante.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
- En un equipo autoorganizado es donde se encuentran las mejores arquitecturas, requisitos y diseños.
- Siempre se busca como llegar a ser más efectivo y se mejora constantemente.

5.1.2. Scrum

Al utilizar la metodología de Scrum para la propuesta de trabajo de graduación, se logra dar visibilidad de todo lo que está pasando mediante a reuniones de planificación y seguimiento. Las revisiones constantes en los entregables y el monitoreo del progreso para identificar y corregir las variaciones no deseadas. Luego aplicar estos ajustes y entregables para la entrega continua.

- Conceptos básicos del Scrum
 - Iterativo incremental
 - Colaboración constante con el cliente
 - Equipos auto gestionados

Scrum se basa en un ciclo de vida iterativo incremental. Cada iteración se denomina “sprint” y tiene una duración fija de entre 1 a 4 semanas. En este caso se define sprint's de 2 semanas. En este caso por tratarse de un trabajo de graduación el asesor asumía la responsabilidad de tomar el cargo de Scrum Master, quien estuvo velando por los entregables semanales del aplicativo y a quien entregaba cuentas de lo que estaba ocurriendo durante el proyecto.

Es importante mencionar que para el desarrollo de proyectos de gran escala los equipos están conformados por 2-10 integrantes como máximo y cada equipo tiene un líder que audita a los demás y proporciona apoyo.

5.1.3. Scrumban

En esta modalidad de la metodología las reuniones se llevan a cabo cuando sea necesario determinar que historias se completaran en la próxima iteración. Esto ahorra tiempo ya que se planifica bajo demanda lo que permite ahorrar tiempo al equipo en relación con las planificaciones recurrentes.

Para mantener bajo el número de iteraciones, se regula el límite de trabajo WP. Al igual que en Kanban, el tablero Scrumban revela cuellos de botella en el flujo de trabajo. Scrumban no requiere ningún número máximo de miembros del equipo o roles específicos como se utiliza Scrum. Los roles que tiene un equipo antes de adoptar Scrumban se mantienen.

Para el proyecto se considera que la metodología de Scrumban se acoplaba a las necesidades que iba a representar el desarrollo de la aplicación, así mismo también se propuso la utilización de un tablero Kanban para llevar un control del estado actual del aplicativo.

5.2. Ciclo de vida del proyecto

El ciclo de vida de un proyecto ayuda a tener un mejor panorama y le proporciona una mejor forma de abordar las diferentes tareas. En este caso se definen cada una de las fases de las que el proyecto estará compuesto desde su inicio hasta el final, estas fases están basadas en las necesidades del cliente.

Dentro de un marco general para definir un ciclo de vida, se puede definir que estas cuentan normalmente con las siguientes características.

- Fases secuenciales
- Cada paso de fase involucra una entrega de resultado
- Cada una de las fases entregables tienen que ser aprobadas antes de comenzar la siguiente fase.

5.2.1. Fase de inicio del proyecto

Durante esta fase se definió el argumento educativo para el aplicativo que se iba a estar desarrollando para luego ser aprobado posteriormente. Durante este tiempo se definieron aspectos como:

- Estudio de viabilidad del aplicativo para los estudiantes, sabiendo el estado socioeconómico del estudiante promedio, la compatibilidad y

accesibilidad de los estudiantes a la herramienta era importante, tomar en cuenta también plataformas en las que fuera capaz el estudiante de poder acceder y que impacto podía llegar a tener esta aplicación en el marco educativo dentro de la universidad.

- Crear un plan de ejecución del proyecto de cómo se tenía planeado abordar cada una de las etapas que estaban contempladas como el tema matemático y de los algoritmos que iban a conformar la base de las funciones que posteriormente iban a ser manejadas en la nube. Luego de haber terminado la parte lógica se iba a empezar a desarrollar el aplicativo y por último las pruebas finales que desplegaran el resultado esperado para cada uno de los métodos.
- Identificar a las partes interesadas que en este caso son los estudiantes y los maestros que son el objetivo, ellos serían los usuarios que estarán utilizando el aplicativo para el cálculo de los métodos que se trabajan en el aplicativo.
- Seleccionar las herramientas de gestión del proyecto que en este caso por el número limitado de integrantes se optó a utilizar una variante de Scrum ban.

5.2.2. Planificación del proyecto

Tomando en cuenta la necesidad de los estudiantes de contar con una herramienta que les pudiese facilitar la resolución de los problemas con los que se topan en Matemática Aplicada 3, se identificaron puntos importantes a tomar en consideración para la maquetación de la aplicación. Temas relacionados a la fácil interpretación de la aplicación, viabilidad del proyecto, estimaciones de

tiempos de entrega en base a módulos, las historias de los usuarios, retroalimentación por parte de los maestros, desarrollo y etapa de pruebas.

Se definió un prototipo que abarcara un conjunto limitado de funciones que el aplicativo fuese capaz de dar solución como una versión de prueba. Proyectando la integración con otras matemáticas que requieren este nivel de procesamiento.

Para este punto los objetivos están dirigidos al desarrollo de una aplicación multidispositivo web App que sea capaz de darle solución a las funciones que se procesan dentro del curso de matemática aplicada 3. Así mismo agregar una integración en la nube conectando servicios universitarios de funciones *serverless*.

Otro punto para destacar será el diseño *responsive* y adaptable a una gran variedad de dispositivos sin exponer la funcionalidad ni la estética de la aplicación.

Según el análisis y la estimación de los entregables se piensa tener un entregable que contempla las funcionalidades necesarias para darle solución a los problemas de Matemática Aplicada 3. Una estimación en tiempo de lo que requirió el desarrollo de este proyecto desde la investigación hasta el cierre del proyecto.

5.2.3. Ejecución del proyecto

Las tareas que se llevaron a cabo durante esta etapa de ejecución de este proyecto fueron:

- Monitoreo y control

Se definió la visión de lo que iba a consolidar la aplicación, tomando en cuenta la tecnología para desarrollarla. Actuando frente a posibles cambios que afectaran el flujo del proyecto.

- Informe de avances

Se les dio seguimiento a los cambios, teniendo al asesor como supervisor o cliente, se procedió a dar un informe de los avances y actualizaciones. Evitando cualquier tipo de desentendimiento haciendo saber de cualquier tipo de inconveniente que fuera sucediendo.

- Concertación de reuniones de seguimiento

De la mano del asesor se estuvo teniendo reuniones semanales para asegurar que se estuviera avanzando en el proyecto. Estableciendo un orden y tomando notas de observaciones de diseño o de implementación de la arquitectura en la nube.

- Gestionar y resolver problemas

- Demoras: fue un obstáculo que se tuvo que abordar por cuestiones laborales el tiempo era limitado. Fue necesario reestructurar entregables y aplazar reuniones.
- Problemas humanos: un inconveniente por el tema de la situación de salud actual que se tuvo fue la comunicación fluida. Por el

aislamiento respecto a las autoridades que se tenía, era en parte difícil concretar una comunicación productiva.

- Problemas de calidad: cuando se hicieron venir los problemas de demoras y problemas de gestión humana, el proyecto tuvo también el problema de corrupción del alcance ya que se ponía en riesgo el alcance que se tenía para la app en cuestión.

5.2.4. Supervisión y control de proyectos

En el siguiente apartado se va a estar definiendo controles para la gestión efectiva de los procesos del proyecto, acá se toma en cuenta el inicio, la planificación, la ejecución y el cierre del proyecto. Una lista de puntos importantes a tomar en cuenta para la supervisión y control del proyecto en curso.

- Medir el rendimiento real con el del plan del proyecto
- Rendimiento en general
- Minimizar o mitigar los riesgos del proyecto
- Mantener la información actualizada relacionada con el proyecto
- Proveer proyecciones para actualizar la información de los costos y el cronograma.
- Control de las versiones de la web App tomando en cuenta los cambios.

5.2.5. Cierre del proyecto

Con el cierre del proyecto lo que se buscaba era dar una propuesta de aplicativo web que pudiese sustituir las opciones actuales, el desarrollo de la

aplicación se estaría concluyendo en esta etapa, alcanzando como objetivos generales:

- Recopilar información y documentación del aplicativo
- Analizar objetivamente el desempeño del proyecto
- Reunir todos los puntos aprendidos para futuros proyectos

5.2.6. Tiempos de cierre de proyecto

La estimación en cuanto al tiempo da una perspectiva que más se asemeje a la realidad. Aunque existen diferentes factores que pueden llegar a afectar el tiempo de ejecución o alguna otra fase del ciclo del proyecto en curso, existen varios tipos de cierre de proyecto.

- El cierre final del proyecto: que indica que el proyecto finalizó en el tiempo establecido cumpliendo los parámetros que se definieron o bien existe la posibilidad de que el proyecto haya sido cancelado.
- El cierre parcial: indica que se entrega una versión que cumplen con determinadas tareas, es conveniente proveer cierres parciales y luego el resto del proyecto que continúe.
- Los cierres intermedios: el ejemplo más claro es el cierre de una fase del ciclo de vida de un proyecto.

5.3. Tecnologías

Las tecnologías seleccionadas para el desarrollo del aplicativo se tomaron en cuenta considerando compatibilidad con los dispositivos, facilidad de integración y tiempos de entrega por módulo.

Estas tecnologías se dividieron en tres partes que conforman el pipeline de tecnologías en las cuales se despliega toda la aplicación. Para la elección de la tecnología se realizó una investigación donde se tomaron en cuenta factores como el presupuesto, que los servicios fueran escalables y la capacidad de procesamiento.

5.3.1. Proveedores de servicios en la nube

La universidad dada situación que actualmente está viviendo, ha tenido que migrar indefinidamente a las plataformas digitales para continuar proveyendo la educación superior a los estudiantes.

Herramientas como Moodle, Google suite con los beneficios a la Universidad de San Carlos de Guatemala de ser una institución educativa y la plataforma para la gestión de notas de los estudiantes (Ej. Portal de ingeniería), conforman la estructura digital en la que se sostiene con base al contexto educativo en estos tiempos de pandemia.

Si bien estaban las herramientas proporcionando una variedad de soluciones para que los estudiantes puedan tener una experiencia lo más apegada a cuando presencialmente, existen muchos déficits que perjudican el proceso educativo.

La inestabilidad de los servidores durante asignaciones a cursos conforma uno de los problemas que más aquejan a los estudiantes. Es evidente que el volumen de estudiantes que intenta ingresar y realizar su asignación a cursos es demasiada carga para el servidor.

Esta propuesta contempla la migración no solamente de la base de datos de la universidad sino la migración de funciones e implementación de procesos de seguridad, así como el respaldo y mitigación de posibles fallas en los servidores, proveyendo a los servicios en la nube, la suficiente potencia para poder procesar rápidamente la cantidad de solicitudes que los estudiantes que están realizando en sus gestiones académicas.

Se describe un análisis de las tres empresas líderes (Amazon, Azure, *Google cloud*), en proveer servicios en la nube, para luego de acuerdo con el análisis se defina una propuesta singular o híbrida de servicios en la nube. Tomando en cuenta factores de precio, capacidad, escalabilidad y disponibilidad según los servicios que podrían utilizar este ente educativo.

- Amazon

Es una plataforma pionera en el tema de los servicios en la nube y el comercio electrónico por Amazon. Actualmente es una de las plataformas de infraestructura como servicio para cualquier tipo de empresas que más se utiliza y que cuenta con más funcionalidades, lo cual habla mucho del constante trabajo por seguir innovando.

AWS ofrece una gran gama de ofertas de SaaS y PaaS para cualquier tipo de clientes. Los usuarios pueden aprovechar estos beneficios de escalabilidad, seguridad y confiabilidad.

Los clientes también cuentan con almacenamiento en la nube de alta seguridad, red de entrega de contenido, entrega de archivos, base de datos administrada y varias otras características.

- Google Cloud Platform

Se establece como un servicio de computación en la nube líder en el mercado. Esta provee una amplia gama de servicios en la nube que se ejecutan en la misma infraestructura que los productos populares de Google.

App Engine, anunciado como una oferta de Google en 2008, es una plataforma para que se pueda desarrollar y alojar aplicaciones web en los sólidos centros de datos de la empresa. Fue la primera oferta de computación en la nube del gigante tecnológico.

- Azure

Este es un proveedor de IaaS que le pertenece a Microsoft e incorpora una amplia gama de servicios y productos. Es un proveedor de *backend* líder para dispositivos móviles, esto por medio de la implementación de características móviles en el que confían las aplicaciones *node.js* y *c#*.

Azure tiene la ventaja de resiliencia de datos, protegiendo la data en diferentes centros de datos con múltiples centros de datos de alta seguridad.

Azure también viene con integración BCDR que garantiza una copia de seguridad y una recuperación confiable.

Para determinar que proveedor era el más adecuado para proporcionar los servicios que se buscan implementar se tomaron en cuenta los factores generales. Basándose en que los tres proveedores cuentan servicios similares se procedió a hacer un análisis de los costos para la toma de la decisión a largo plazo.

A medida que los proveedores de la nube continúan bajando los precios de las instancias en la nube, han agregado opciones de descuento.

La comparación de precios en la nube de AWS, Azure y Google Cloud tiene un nivel de complejidad dado que están en un cambio constante. Haciendo la comparación del recurso de una instancia / máquina virtual, se obtienen los siguientes resultados:

- Criterios de comparación de precios en la nube

Para hacer más justa la comparación de precios se estará considerando la misma región para comparar CPU y sistemas operativos para las instancias.

- Región: Este EE. UU
 - SO: Linux
 - CPU virtuales/núcleos: 4
-
- Tipos de instancias/máquinas virtuales:
 - Propósito general
 - Computación optimizada
 - Memoria optimizada

Figura 15. **Comparativo entre empresas que proporcionan los servicios en la nube de cada uno de los escenarios de tipos de instancias**

Instance type	aws		Azure		Google Cloud	
	AWS Instance	RAM	Azure VM	RAM	Google Cloud VM	RAM
General purpose	t4g.xlarge	16 GiB	B4MS	16 GiB	e2-standard-4	16 GiB
Compute optimized	c6g.xlarge	8 GiB	F4s v2	8 GiB	c2-standard-4	8 GiB
Memory optimized	r5.xlarge	32 GiB	E4a v4	32 GiB	m1-ultramem-40	961 GiB

Fuente: elaboración propia, empleando cast.ai.

En la siguiente figura se muestran los precios bajo demanda por hora de cada uno de los escenarios de tipo de instancia en AWS, Azure y Google Cloud.

Figura 16. **Precios bajo demanda que proporcionan las empresas de cada uno de los escenarios de tipos de instancias**

Instance type	AWS	Azure	Google Cloud	Pricing		
				AWS	Azure	Google Cloud
General purpose	t4g.xlarge	B4MS	e2-standard-4	\$0.1997	\$0.198	\$0.16
Compute optimized	c6g.xlarge	F4s v2	c2-standard-4	\$0.204	\$0.212	\$0.25
Memory optimized	r5.xlarge	E4a v4	m1-ultramem-40	\$0.302	\$0.28	\$6.3039

Fuente: elaboración propia, empleando cast.ai.

Con estos datos se puede determinar que el beneficio del compromiso de un año es similar en AWS y Azure. AWS gana en la categoría de Compute Optimized, que es lo que se busca para procesar un gran volumen de peticiones.

Con la información recabada y comparación de servicios en instancias se llega a la conclusión que el proveedor que cuenta con los servicios y la potencia de infraestructura para soportar el volumen de usuarios es AWS.

- Ventajas de utilizar AWS:
 - Flexibilidad y facilidad de uso: con AWS puedes seleccionar el sistema operativo, el lenguaje de programación, la plataforma de aplicaciones web, la base de datos u otros servicios que necesites.
 - Rentabilidad: AWS ofrece planes de precios por el costo de la potencia de cómputo, el almacenamiento, y demás tipos de recursos que vayas a utilizar sin contratos a largo plazo ni gastos anticipados.
 - Velocidad de organización: con AWS, se puede ganar en agilidad en la organización, se puede reducir el tiempo que dichos recursos tardan en estar disponibles para los desarrolladores.
 - Escalabilidad y elasticidad: gracias al respaldo de la sólida infraestructura de Amazon con herramientas como AWS, Auto Scaling y Elastic Load Balancing su aplicación podrá ampliarse o reducirse según la demanda. Gracias a AWS tendrás acceso a los

recursos informáticos y de almacenamiento siempre que los necesites.

- Seguridad: AWS aplica un enfoque integral para proteger la infraestructura, incluidas medidas físicas operativas y de *software*. AWS es una plataforma tecnológica que cuenta con certificaciones y acreditaciones para administrar la infraestructura de TI de las empresas de manera segura y duradera.

5.3.2. API

En cuanto a los *API* que se conoce como interfaz de programación de aplicación, es un conjunto de protocolos y definiciones que pueden permitir que una aplicación se comuniquen con otra. Esta interfaz que permite la interacción entre varios intermediarios de *software*. Su función principal es el envío y recepción de solicitudes y llamadas. Este permite una vía de acceso a toda la información que permitirá que el *software* y el *backend* puedan tener una vía de comunicación.

Ventajas de utilizar una API:

- Los servicios en un *api* pueden ser moldeadas según las necesidades. Las *API's* podrán ser modificadas o cambiar cuando sea necesario.
- Las *API's* pueden acceder a los componentes de las aplicaciones pueden hacer que la entrega de información y servicios sea más flexible y rápida.
- Las *API's* pueden simplificar la implementación de nuevos modelos para aplicativos innovadores.

- Las API son la mejor forma de integrar servicios para otras plataformas. Una API puede permitir que diferentes aplicaciones, sistemas y plataformas interactúen y compartan información entre sí para llevar a cabo diversos tipos de tareas con facilidad.

Para determinar que tecnología era la más adecuada para proveer los servicios que serían implementados se tomaron en cuenta factores generales. Basándose en que las tres tecnologías proporcionan funciones similares se procedió a hacer un análisis de las características principales de cada uno y la velocidad de respuesta.

Teniendo en cuenta que para el proceso de solicitudes y respuestas se necesita un protocolo rápido y seguro, se definió utilizar el protocolo REST para manejar la comunicación entre el aplicativo y la API en formato JSON.

Así que habiendo definido el proveedor AWS como la plataforma sobre la cual se levantarían servicios, se determinan las siguientes opciones sobre las cuales se podría levantar la API.

Golang: es un lenguaje de programación compilado de código abierto que se utiliza para crear software simple, sistemático y seguro. Fue diseñado por Google en el año 2007 y ha sido adoptado fácilmente por desarrolladores de todo el mundo debido a sus características como seguridad de la memoria, tipificación estructural, recolección de basura y similitud con el lenguaje C. *Golang* se utilizan para escribir interfaces de programación de aplicaciones (API).

Node js: *Node.js* es un ambiente de ejecución de JavaScript. Utiliza un modelo de entrada y salida sin bloqueo controlado por eventos, de esta manera

lo hace un entorno ligero y eficiente. Esta ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones *network* escalables. Gracias a que está construido en base a V8, motor el cual está escrito en C++, prácticamente se puede correr JavaScript en cualquier sistema operativo, desde servidores hasta dispositivos móviles.

Flask Python: es un ambiente de ejecución de *Python*, es un “micro” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC. Esta es una librería ligera que trabaja con ORM/libraries. Flask RESTFull impulsa el uso de buenas prácticas con una configuración mínima.

La comparación de las funcionalidades de las tres propuestas ofrece funcionalidades similares, pero al final la decisión quedo entre la herramienta de NodeJs o *Golang*. Se sabe que Node.js ha sido durante mucho tiempo la opción preferida para el desarrollo *backend*, pero *Golang* ha surgido recientemente como un fuerte oponente.

Analizando individualmente la propuesta que ofrece *NodeJs*, cabe destacar que cuenta con un motor para construir aplicaciones de red simples y complejas. Es escalable y muy eficiente con cualquier aplicación. El entorno de ejecución está basado en JavaScript que ayudan a crear entornos del lado del servidor, así como aplicaciones de red. NodeJs es una excelente opción para crear una aplicación en tiempo real no conexiones bidireccionales donde el servidor no espera a que una API devuelva todos los datos.

Por otra parte, *Golang* es un lenguaje de programación que puede utilizarse para crear diversas aplicaciones, incluidos sistemas de aplicaciones

que son altamente modulares que tienen como objetivo principal proporcionar seguridad y la velocidad.

Lo más importante es que *Golang* soporta la programación de sistemas distribuidos a gran escala y aplicaciones o servicios de red altamente escalables. También es bien utilizado en la nube y en aplicaciones de Big Data.

[insertar pros y contras de cada uno]

Tomando en cuenta los pros y contras de cada una de las tecnologías, se determinó que la herramienta que se iba a estar utilizando para desarrollar la API sería NodeJs por su adaptabilidad y previos conocimientos que incentivaron la elección.

Ventajas de utilizar *NodeJs*:

- La compilación de Node.js se realiza en tiempo de ejecución, *Just In Time* (JIT), esto trae consigo una mayor optimización a las funciones que más veces sean llamadas.
- Mediante *clusters* permite tener una escalabilidad alta.
- Se puede expandir el código añadiendo módulos de forma fácil gracias al *Node Package Manager* (NPM).
- Un alto rendimiento en proyectos donde se necesite ejecución en tiempo real.

5.3.3. Framework

En la actualidad existe una variedad de *frameworks* que pueden proporcionar una interfaz estable para el desarrollo del trabajo de investigación. Actualmente *frameworks* híbridos que permiten manejar un mismo código y compilar para las diferentes plataformas son parte del conjunto de *frameworks* innovadores que con la necesidad de poder optimizar el proceso de desarrollo se han ido desarrollando.

La tecnología híbrida nació para reducir los costes, los tiempos y facilitar el aprendizaje y desarrollo de aplicaciones móviles, aportando también la posibilidad de crear webs con estos desarrollos. Por ello se podría decir que este tipo de aplicaciones son totalmente multiplataforma (celular, *tablet*, computadora, TV...). Con el desarrollo de un código único se pueden generar aplicaciones para las plataformas deseadas. En el caso de móvil, para Android y iOS.

El desarrollo de aplicaciones híbridas tiene una curva de aprendizaje mucho más suave que el desarrollo nativo y esto aportó al proyecto lo que se estaba buscando para desarrollar todo el entorno educativo para que los estudiantes pudiesen interactuar con la aplicación. Dentro de los *frameworks* que se encuentran más atractivos para la propuesta están:

- *ReactJs*
- Angular

ReactJs fue lanzada en 2013, esta es una biblioteca de JavaScript. Este *framework* cuenta con una gran variedad de componentes que proporcionan una gran variedad de herramientas para crear un diseño *responsive* y que sea

más amigable para el usuario. El virtual DOM es un programa de aplicaciones que es equivalente a un documento XML. Tiene forma de árbol con múltiples nodos que representan cada documento. Con *ReactJs* actualiza el DOM virtual rápidamente cada vez que hay un cambio, esto permite una tasa de refrescamiento y actualización del *frame* o componente que se está utilizando impecable.

Tabla IV. **Cuadro comparativo de herramientas para desarrollar el *frontend***

	Angular	ReactJs
Fundador	Misko hevery	Jordan Walke
Ideal para	Creando aplicaciones web interactivas	Aplicaciones web extensas con un manejo de información variable
DOM	Real – DOM incremental, comparado con el previo.	Virtual solamente se compara con el previo y se modifica.
Tamaño de app	Relativamente pequeña	Relativamente pequeña
Desempeño	Alto	Alto
Binding UI	Interfaz de usuario unida a un objeto plano o propiedad	Enlace directo de los estados hacia la interfaz de usuario
Enlace de información	Two-way(bi-direccional)	One-way(mono-direccional)
Curva de aprendizaje	Empinada	moderada
Opiniones generales	Considerablemente menos opinada	Opiniones flexibles
Precio	Open source	Open source

Continuación de la tabla IV.

<p>Por qué debería utilizar esta tecnología</p>	<ul style="list-style-type: none"> • TypeScript • Apoyo de una gran comunidad • Orientado a objetos (OOP) • Html limpio 	<ul style="list-style-type: none"> • Flexible • Gran ecosistema • JavaScript/TypeScript • Permite manejo en equipos pequeños • Bueno para elegir entre las mejores opciones (paquetes)
--	---	---

Fuente: elaboración propia.

5.3.4. Backend

En cuanto al desarrollo de la aplicación dentro del contexto de una web app, se entiende que se debe manejar una capa de acceso a datos que no es directamente accesible por los estudiantes. Dentro del *backend* se encuentra la lógica de la aplicación que procesa los datos.

En este caso las funciones Lambda de AWS otorgarán la lógica a la que el *backend* hará, las consultas y luego de configuraciones de acceso el aplicativo tendrá la función de enviar solicitudes a la espera de una respuesta. Para dar una descripción menos técnica de lo que hace la función que tendrá el *backend*, se puede decir que el usuario en este caso el estudiante al ingresar los datos de una operación y presionar el botón (calcular).

Este enviará una solicitud al *backend* y este procesara la solicitud haciendo la petición a la función *serverless* correspondiente, obteniendo cada uno de los parámetros que componen el método matemático que se esté

trabajando y luego de la función desarrollada este devolverá la respuesta al problema planteando que se procesó.

Se hace un análisis técnico de las tecnologías que se tienen para desarrollar el *backend*.

5.3.4.1. Golang

Al utilizar la tecnología de *Golang* y para la implementación de las funcionalidades necesarias para el aplicativo educativo es necesario utilizar un *framework* para el manejo de las peticiones web. Gin es un *framework* web para Go, actualmente es uno de los más populares, para construir microservicios porque permite manejar pipelines de despliegue donde se pueden conectar en los middlewares.

En cuanto al performance cabe mencionar que utiliza un agrupamiento de rutas optimizado y es ligero, esto quiere decir que las velocidades de respuestas son muy competitivas. En este caso se estaría configurando el *backend* en Go para utilizarlo como un intermediario entre la app y las funciones Lambda.

Se enumeran una serie de ventajas y desventajas que implica la utilización de *Golang* como *framework* para el *backend*.

- Ventajas
 - El lenguaje es de fácil aprendizaje y mucho más cómodo a comparación de otros lenguajes.

- Facilidad de uso y numerosos mecanismos de automatización de procesos que ahorran tiempo.
- Un mismo formato lo que facilita en gran medida su legibilidad y uso.
- Un excelente rendimiento, la velocidad de compilación es evidente y se compara con otros compiladores.
- Desventajas
 - El manejo de errores lo cual le agrega complejidad a la hora de debuggear la aplicación que se esté desarrollando.
 - Por lo novedoso y reciente desarrollo, la refactorización puede traer problemas para el manejo de errores sintácticos.
 - No permite la sobrecarga de funciones.

5.3.4.2. Python

Python permite la integración ligera junto con *flask* para trabajar en un marco web que es pequeño y ligero, este proporciona herramientas útiles para crear aplicaciones web. Una ventaja de utilizar Python junto a *flask* es la flexibilidad y extensibilidad que no fuerza a una estructura concreta ni requiere código complejo para poder iniciar.

Se enumeran una serie de ventajas y desventajas que implica la utilización de Python como framework para el *backend*.

- Ventajas
 - Es un lenguaje de alto nivel que es de fácil interpretación respecto del lenguaje natural.
 - Es polivalente y de paradigmas que puede funcionar para varios propósitos y de múltiples paradigmas admitiendo programación estructurada, funcional, y programación orientada a objetos.

- Desventajas
 - El performance puede llegar a comprometerse debido a su dinamicidad y versatilidad.
 - Consume una gran cantidad de memoria por su flexibilidad de los tipos de datos.
 - No es bueno para el desarrollo móvil

5.3.4.3. NodeJs

NodeJs es ideal por el desarrollo del *backend*, y por su velocidad de respuesta en tiempo real, este destaca en la capacidad de construir aplicaciones rápidas y escalables en la red. Para esto fue necesario utilizar un framework específicamente desarrollado para el procesamiento de peticiones realizadas desde una web app. Express es un framework flexible y ligero para *node.js* que provee una vasta cantidad de métodos de utilidad HTTP y *middleware* para crear una API robusta rápida y fácil.

Luego de un análisis y tomando en cuenta el propósito que iba a cumplir el *backend*, se llegó a la conclusión que la herramienta que se iba a utilizar sería *NodeJs*, por ser una herramienta de fácil implementación con una comunidad que proporciona una documentación y foros detallados.

Se enumeran una serie de ventajas y desventajas que implica la utilización de *NodeJS* como *framework* para el *backend*.

- Ventajas
 - Permite la utilización del lenguaje de JavaScript para servidor y cliente.
 - *NodeJS* está basado en eventos, lo cual permite el manejo asíncrono de operaciones.
 - Proporciona una gran variedad de paquetes por medio de NPM para agregar funcionalidades.

- Desventajas
 - Cuenta con una API inestable en la que entre versiones la compatibilidad difiere.
 - Puede llegar a anular los beneficios del performance por cuellos de botella en la cola de eventos que se ejecutan.

- Estructura de una aplicación utilizando *NodeJs* y *express*

El documento `app.js` es el archivo principal del directorio que es generado al crear la aplicación *backend*, se podría decir que es el cerebro de la aplicación.

El número de puerto para la aplicación está asignado en el directorio `/bin/www` y se puede configurar.

Otro archivo de configuración importante es el manifiesto `package.json` que contiene todos los metadatos acerca del proyecto tal como descripción y dependencias.

- Estructura del folder
 - Views - las vistas del *backend*, ideales para documentar la API
 - Routes – los controladores de las rutas para el procesamiento de las solicitudes http. Aquí es donde la mayor parte del trabajo sucede.
 - Public – Artefactos públicos (imágenes, css, entre otros), se van a almacenar acá.

5.3.5. FrontEnd

El frontEnd está relacionado con todo lo que los estudiantes van a ver, interactuando con los iconos, botones, inputs entre otros. Se debe cumplir con altos estándares de calidad y usabilidad. El objetivo es desarrollar una aplicación amigable que este optimizada para evitar cualquier tipo de cuello de botella que retrase el tiempo de respuesta de la aplicación en uso.

5.3.5.1. Angular CLI

Este *framework* permite crear web Apps totalmente funcionales y su atractivo está en que permite usar *TypeScript*, que es un lenguaje basado en JavaScript, este aporta beneficios enfocados en la escalabilidad y mejor manejo de la memoria. Entre las ventajas y desventajas de utilizar este framework se encuentran:

- Ventajas
 - Permite la integración de pruebas unitarias
 - Está basado en el paradigma de objetos para definir características de este.
 - Cuenta con una estructura basada en componentes.
 - Permite una gran variedad de entornos de desarrollo en los que se puede trabajar.
 - Le ofrece al programador una estructura modular para una mejor organización que ayuda a mejorar la productividad.
- Desventajas
 - Las aplicaciones tienden a ser pesadas, esto por sus paquetes de JavaScript o dependencias que utiliza.

- En primera instancia la aplicación en angular puede llegar a tener un alto nivel de complejidad, por lo que aprender angular implica un gran reto.
- La documentación carece de detalles importantes lo que dificulta la resolución de problemas.

5.3.5.2. React JS

Al final se tomó la decisión de utilizar *React JS* por su integración con herramientas de desarrollo que agilizan el tiempo de desarrollo.

Recientemente por parte del equipo de desarrollo de Facebook que son los precursores de *react* se ha logrado facilitar la manera en la que se genera un nuevo proyecto. Entre las ventajas y desventajas de utilizar este *framework* están:

- Ventajas
 - *React* ofrece una opción rentable para crear aplicaciones multiplataforma.
 - Se requiere menos código para realizar funcionalidades que cuenta con componentes ya listos.
 - Acceso a bibliotecas de funcionalidades totalmente gratuitas.
 - Cuenta con un excelente rendimiento.

- Diseño modular que facilita la creación e integración de aplicaciones por los módulos que pueden llegar a ser reutilizables.
- Desventajas
 - La documentación, aunque hoy en día es bastante completa, por su rápido desarrollo y constante cambio se dificulta el tema de documentar que varios desarrolladores contribuyen.
 - Actualmente no se tiene noción de un estándar de desarrollo.
 - Aunque su curva de aprendizaje es medianamente compleja toma tiempo dominar la integración completa de las funcionalidades de *React Js*.

Tomando en consideración los pro y contras de *React* y *angular*, se considera que *React Js* proporcionaría el conjunto de herramientas como programadores para desarrollar el aplicativo de forma efectiva, los factores que se tomaron en cuenta para la toma de decisión fueron:

- Facilidad de aprendizaje
- El uso del DOM Virtual para manejar los diferentes elementos de la aplicación.
- Gran variedad de librerías de código abierto.
- *Framework* ligero.

- Estructura del directorio básico de un proyecto en *React*

Tomando como referencia MVC (modelo vista controlador), para el proyecto, se estaría proponiendo una organización de la siguiente manera:

- *Controllers*
- *Models*
- *Views*
- *Config*

6. IMPLEMENTACIÓN DEL PROYECTO

En este capítulo se describe el proceso que conllevó el desarrollo de este proyecto, explicando secuencialmente como se abordaba cada fase que involucraba las diferentes tecnologías utilizadas.

6.1. *AWS Cloud*

Una vez tomada la decisión de la nube que se iba a estar utilizando, se definió una secuencia de pasos que se iban a llevar a cabo para poder configurar el entorno dentro del ambiente de AWS, montando las funciones y configurando los servicios necesarios.

6.1.1. *S3 bucket*

Para la propuesta de la aplicación se estaría haciendo uso de este servicio que lo que provee es un contenedor para objetos. Este espacio sería útil para el almacenamiento de los pdf desarrollados del perfil del usuario para el manejo de las tareas. Estos contenedores de objetos son flexibles y aceptan una infinidad de tipos de archivos y metadata.

Para almacenar un objeto, lo primero que se tiene que hacer es, crear un *bucket* y luego subir el objeto. Cuando el objeto está en el contenedor se puede descargar, mover, abrir y darle el uso que se le quiera dar.

6.1.1.1. Proceso para la configuración del servicio Amazon S3

Lo primero es contar con una cuenta registrada dentro del entorno de AWS para acceder a los servicios que esta plataforma en la nube provee.

Tomando en cuenta que el pago será únicamente, se va a estar utilizando y utilizando las herramientas para cálculo de precios, existe además una capa gratuita que siempre y cuando no exceda esos límites es posible usar el servicio de manera completamente gratuita.

Lo siguiente es contar con un usuario IAM o utilizar un servicio root para otorgar estos permisos de uso para los servicios correspondientes a S3 *bucket*.

Luego que ya están conectados a AWS, se dirigen a la consola de gestión para los servicios de S3:

- Se elige crear *bucket*
- Se escoge el *Bucket name*, se ingresa la información del DNS para el *bucket*.
- Se escoge la región donde se desea que el servicio se ubique físicamente.
- Se habilitan o deshabilitan las ACL, eligiendo una de las siguientes configuraciones.

- Por defecto se tiene la configuración del *bucket* restringida, entonces se debe cambiar a público para su correcto funcionamiento.

6.1.2. Funciones Lambda

Para el caso de uso, estas funciones Lambda permitirían lograr tener el poder del procesamiento de las funciones en la nube, esto se lograría gracias a las funciones *serverless* que en este caso serían las que provee la plataforma de AWS. Las funciones Lambda corren bajo el esquema de instancias que procesan eventos. Estos eventos son invocados directamente utilizando una API de Lambda. Estas funciones contienen código que procesan los eventos y estos pueden ser codificados en lenguajes como Node.js, Java, C#, Python, Ruby, Go, PowerShell.

6.1.2.1. Conceptos básicos de las funciones Lambda

- **Function:** son los recursos que puede ejecutar el código dentro de una función Lambda. Estas funciones contienen lógica que recibe una entrada y luego de procesarla devuelve una respuesta.
- **Trigger:** se le conoce como la configuración que invoca una función Lambda. En AWS los *triggers* o disparadores se pueden configurar para invocar funciones y eventos, donde el evento mapeado es un recurso que lee objetos de la cola e invoca una función.
- **Event:** es un documento en formato JSON que contiene la información que la función Lambda procesara.

6.1.2.2. Proceso para la configuración de una función Lambda

Creación de una función Lambda sin utilizar la consola de AWS Lambda

- Navegar al apartado de Funciones Lambda en la consola de AWS
- Seleccionar “create your Lambda Function from scratch
- Seleccionar el lenguaje de programación sobre el cual se estará trabajando la función en tiempo de ejecución, Java 8 por ejemplo.
 - Para el nombre de la función se ingresa el nombre para definir la función.
 - Para el momento de ejecución, confirmar el lenguaje con el que se correr la función.
- Desplegar una función por medio de un api Gateway previamente configurado para el acceso al servicio.

6.1.2.3. Invocar una función Lambda

Para invocar las funciones Lambda usando un evento de información provista, en este caso, por el *backend* que encapsula el conjunto de parámetros que serán procesados.

Para invocar una función:

- Luego de seleccionar la función, se selecciona Test tab

- En la sección del evento, se selecciona *New event*, y se prueba utilizando los parámetros de prueba para las funciones.
- Se guardan los cambios y luego se cierra la pestaña Test. Lambda corre las funciones por sí mismo. El *handler* de la función Lambda recibe y procesa la petición.

6.1.2.4. Precios para el uso de las funciones Lambda

La ventaja de usar AWS es que no existe un recargo por la creación de las funciones Lambda. Se carga por la ejecución de una función y por la transferencia de información entre funciones Lambda u otros servicios AWS.

6.1.3. Instancias EC2

Las instancias EC2 proveen capacidad de computación escalable. Este servicio en específico será utilizado como plataforma para tener levantado el *backend* mencionado anteriormente. Por sus beneficios de configuración de seguridad y de la red, manejo de memoria. AWS permite escalar gradualmente ante cambios en los requerimientos o picos de demanda.

6.1.3.1. Proceso para iniciar y configurar una instancia EC2

El proceso que se debe seguir para crear una instancia EC2 se compone de 3 pasos necesarios para la configuración inicial.

- Contar con una cuenta AWS
- Crear un *key pair* (conjunto de llaves)
 - Abrir la consola de Amazon EC2
 - Dirigirse al panel de navegación y seleccionar *Key pair*
 - Seleccionar “*Create key pair*” e ingresar un nombre *descriptive*
 - Para el *key pair type*, se puede escoger RSA o ED25519
 - Este genera un archivo descargable, es importante que se resguarde este archivo será el permiso para hacer uso de las instancias que se generan por un servicio SSH.
- Crear un grupo de seguridad
 - Abrir la consola de Amazon EC2
 - En la barra de navegación superior seleccionar la región a la que se le desea asignar un grupo de seguridad.
 - Luego se dirige al panel izquierdo y se localiza “*Security Groups*”.
 - Se selecciona “*Create security group*”.
 - Se agregan las reglas que permiten un tráfico en específico a la que pueden tener o no, acceso a las instancias.

6.1.3.2. Lanzamiento de una instancia

Para el caso de uso se estará utilizando una instancia de Linux por su capacidad de procesamiento, dada la facilidad para levantar sus servicios. Se presenta una breve descripción del proceso y configuraciones necesarias para preparar el ambiente de las instancias EC2 linux.

- Desde la consola del Dashboard, se selecciona “Launch instance”

- Se selecciona el tipo de instancia, la configuración del hardware que se va a estar proveyendo a la máquina virtual. En este caso existe una capa gratuita y otros tipos de instancias que representan un recargo mayor, pero con una capacidad de procesamiento superior.
- Se selecciona el *Security group* sobre el cual estará dictaminada la instancia EC2.
- Se selecciona el *key pair* que se creó anteriormente, confirmando que se tenga acceso al *key pair*.
- Todas las instancias se podrán ver en el dashboard de instancias, y se puede visualizar el estado del lanzamiento.

6.2. Backend

En cuanto a lo referente al procesador de funciones, o funciones *serverless* en la nube, se estuvieron trabajando las funciones correspondientes a los métodos de la Matemática Aplicada 3.

Luego de instalar Node.js y contar con la terminal de NPM que es un administrador de paquetes al que se le pueden instalar nuevos módulos. Para propuesta de aplicación, se maneja un archivo `package.json`, este archivo contiene información sobre el proyecto, como versiones de módulos, como ejecutarlo, y comandos npm que sean personalizados.

Luego se instala *express* que fue un módulo, y en el código se hace una llamada a la constante de *express* junto con un puerto. Así fue como se definió la ruta de la API en el servidor.

Lo primero fue definir la estructura inicial de las funciones que se iban a trabajar, tomando en cuenta los parámetros que necesitaban las funciones para poder funcionar de una manera correcta. Luego de definir las variables iniciales se iniciaba un proceso de verificación de los parámetros recibidos para evitar cualquier retorno nulo. Lo siguiente dependiendo del método que se estuviera trabajando sería el procesamiento de los valores de entrada a los algoritmos correspondientes que generan la solución, utilizando las llamadas a las funciones Lambda.

Factores técnicos por tomar en consideración es que el *backend* al ser montado en una máquina virtual EC2, fue provisionada con los más básicos servicios, haciendo uso de la capa gratuita. Además, también se restringieron los puertos que tenían acceso a esta máquina virtual, esto como factor de seguridad del *backend*. Fue importante tomar en cuenta la simplicidad de las funciones que respondían a las peticiones del Frontend, esto para llevar un código limpio que fuese totalmente funcional.

6.3. Front-End

Al tomar la decisión de utilizar *React JS* como *framework* o una biblioteca basada en JavaScript para desarrollar interfaces de usuario/Frontend, se procedió a la instalación inicial para preparar el ambiente de trabajo. Se describe una serie de pasos iniciales para definir las bases de la parte visual del proyecto:

- Agregar CDN o instalar complementos de *Bootstrap*, esto para fomentar el uso de un diseño *responsive*, adaptable a cualquier tipo de dispositivo.

- Definir el directorio de trabajo en subcarpetas, para un mejor control y manejo del código, definiendo una organización estandarizada para los archivos que conformaran.
 - Components
 - Hooks
 - Utils
 - Service
 - Screens
- Uso del manejador de paquetes NPM para instalación de paquetes, y así agregar funcionalidad al proyecto.
- Se definió un contexto para la propuesta de aplicación para manejar estados en toda la aplicación.

Para la parte enfocada al diseño específicamente se procuró llevar un diseño *responsive* que fuese amigable para los estudiantes, implementando una herramienta didáctica sencilla y simple donde simplemente al ingresar se pudiera rápidamente tener acceso a las diferentes funciones.

CONCLUSIONES

1. El origen de la programación se fecunda en la matemática, lenguaje binario para ser más precisos, desde sus inicios con la máquina de Turing hasta las máquinas más potentes que existen ahora. El avance tecnológico por partes individuales ha permitido segmentar las tecnologías según su rubro expandiendo su alcance, reduciendo costos y recursos humanos para gestionar estos sistemas modulares que realizan operaciones complejas como en este caso de operaciones complejas de matemática lineal.
2. Los servicios en la nube cada día cobran más relevancia, la adherencia de los servicios a estas tecnologías es inminente en el futuro, esto va a permitir no solo un mejor servicio sino mejores costos de implementación, mejor tiempo de integración, monitoreo y mantenimiento.
3. Las herramientas didácticas ayudan al estudiante en su proceso de aprendizaje, la tecnología es un facilitador que permite recortar los tiempos de cálculos manuales y dar una respuesta certera en un periodo de tiempo corto, aprovechar la tecnología para presentar una propuesta con una interfaz amigable y de fácil beneficia de gran manera al estudiante.
4. Los *frameworks* híbridos hoy en día están tomando relevancia que permiten con un solo código compilar para diferentes plataformas o dispositivos (WEB, IOS, Android). Esto acorta los tiempos de Desarrollo

de una herramienta. En el caso de *React JS* es un *framework* con un excelente rendimiento, un diseño modular y rentable que es de código abierto. Así mismo su acoplamiento con su variante *React Native* permite el desarrollo de aplicaciones móviles (IOS y Android).

5. El performance de *React* y de *Node JS* son muy similares y los dos son capaces de utilizar un virtual DOM, es posible manejar componentes dinámicos para las vistas. Los dos son excepcionalmente rápidos

RECOMENDACIONES

1. Realizar mejoras en su funcionamiento, para la integración y optimización de más funcionalidades relacionadas a la Matemática Aplicada 3, así como implementar una herramienta didáctica para la resolución del procedimiento paso a paso.
2. Llevar a cabo la integración de una funcionalidad para generar una plantilla de tarea en base lo procesado en la aplicación. Sería posible incluso integrar un análisis de la información de uso de la aplicación que pueda proporcionar información para la toma de decisiones.
3. Mejorar mientras se le dé uso a la aplicación, los tiempos de respuesta de la misma, a través de optimizar los algoritmos de las funciones *serverless*.
4. Adecuar el sistema a las clases magistrales de los catedráticos para mejorar la experiencia de los estudiantes durante el curso de Matemática Aplicada 3. Esto resultaría en un mejoramiento en la entrega de tareas y poder mitigar la falta de una herramienta de fácil acceso para que los estudiantes realicen sus operaciones

BIBLIOGRAFÍA

1. AWS. *Comience a crear con AWS hoy mismo*. [en línea]. <https://aws.amazon.com/es/?nc2=h_lg>. [Consulta: 1 de enero de 2022].
2. ECHEVERRI, Manuela. *Método de punto fijo*. [en línea]. <<https://sites.google.com/site/procesosnumericos20161procesos/home/metodo-de-punto-fijo/>>. [Consulta: 25 de febrero de 2022].
3. ESPINO, Carlos. *Análisis predictivo: técnicas y modelos utilizados y aplicaciones del mismo*. España: Universidad de Cataluña, 2017. 135 p.
4. FITZGERALD, Anna. *Funciones sin servidor: el nuevo mejor amigo de su sitio web*. [en línea]. <<https://blog.hubspot.com/website/serverless-functions>>. [Consulta: 27 de octubre de 2021].
5. GUTIÉRREZ, Lorena. *Métodos numéricos para resolver ecuaciones y problemas de optimización no lineales*. Tesis de Ing. Petrolera. Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2016. 100 p.
6. LLEDÓ, Pablo. *Profesional ágil*. Argentina: Pablolledo, 2020. 25 p.

7. LUCHANINOV, Yuri. *Uso de Node.js para el desarrollo web backend en 2022*. [en línea]. <<https://mobidev.biz/blog/node-js-for-backend-development/>>. [Consulta: 8 de diciembre de 2021].
8. MITCHELL, Beca. *Investopedia*. [en línea]. <<https://www.investopedia.com/terms/s/software-as-a-service-saas.asp>>. [Consulta: 1 de junio de 2021].
9. PADILLA, Jorge. y GONZÁLEZ, Floreth. *Método de bisección*. [en línea]. <<https://jorgeyfloreth.wordpress.com/2017/02/03/biseccion/>>. [Consulta: 1 de enero de 2022].
10. React. *Una biblioteca de JavaScript para construir interfaces de usuario*. [en línea]. <<https://es.reactjs.org/>>. [Consulta: 1 de enero de 2022].
11. SaM Solutions. *AWS frente a Azure frente a Google Cloud: ¿cuál es mejor?* [en línea]. <<https://sam-solutions.us/aws-vs-azure-v-s-google-cloud-which-is-better/>>. [Consulta: 21 de enero de 2022].

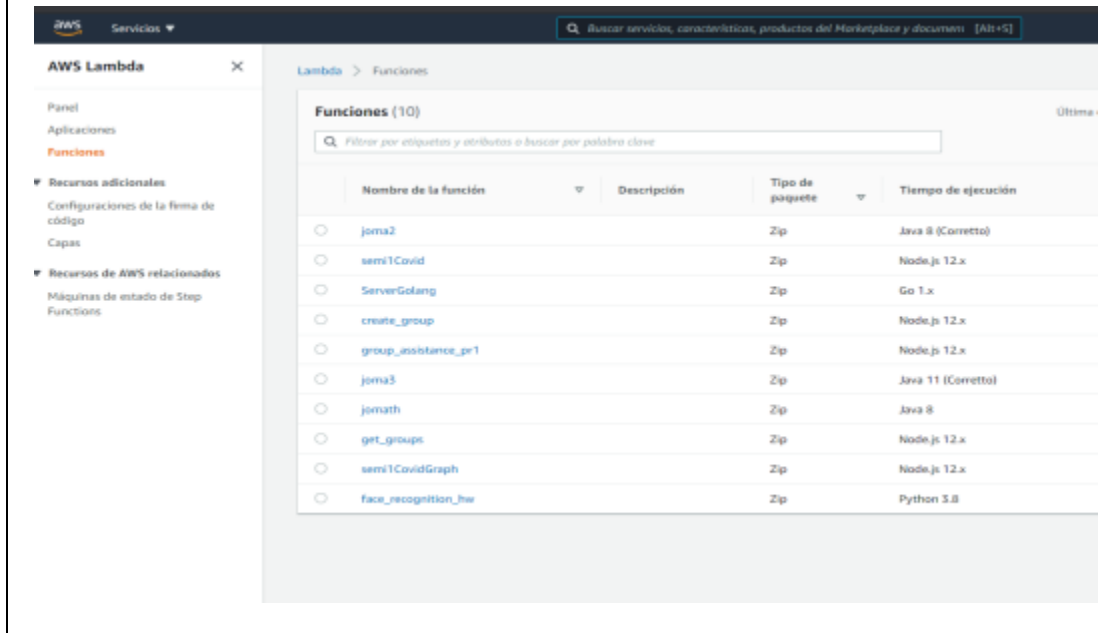
APÉNDICES

Apéndice 1. Utilización de Amazon AWS para implementación de las funciones Lambda

Interfaz gráfica para la creación de una función Lambda

Para acceder a las herramientas que ofrece Twitter a los desarrolladores es necesario llenar un formulario, que inicia solicitando al aplicante que defina la razón por la que desea aplicar, tal como se observa en la figura.

AWS ofrece estas funciones *serverless* o *Lambda functions* que básicamente proporciona toda la infraestructura para procesar información en la nube. Como se puede observar en la imagen se muestra un listado de funciones que se utilizaron en los métodos matemáticos.



Continuación del apéndice 1.

Posteriormente el proceso indica al crear una nueva función, un formulario de configuraciones iniciales y el lenguaje con el que se quiere trabajar.

Lambda > Funciones > Crear una función

Crear una función info

Seleccione una de las siguientes opciones para crear la función.

- Crear desde cero** Empiece con un sencillo ejemplo "Hello World".
- Utilizar un proyecto Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.
- Imagen de contenedor Seleccione un contenedor de imagen para la función.

Información básica

Nombre de la función
Escriba un nombre para describir el propósito de la función.

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

Tiempo de ejecución info
Elija el lenguaje que desea utilizar para escribir la función. Tenga en cuenta que el editor de código de la consola solo admite Node.js, Python y Ruby.

Permisos info
De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante.

► Cambiar el rol de ejecución predeterminado

► Configuración avanzada

Luego es necesario que se cree un recurso, implementando una API Gateway que está asociada a la función Lambda. Aquí se encuentran los métodos de un proceso, y devuelven el resultado esperado.

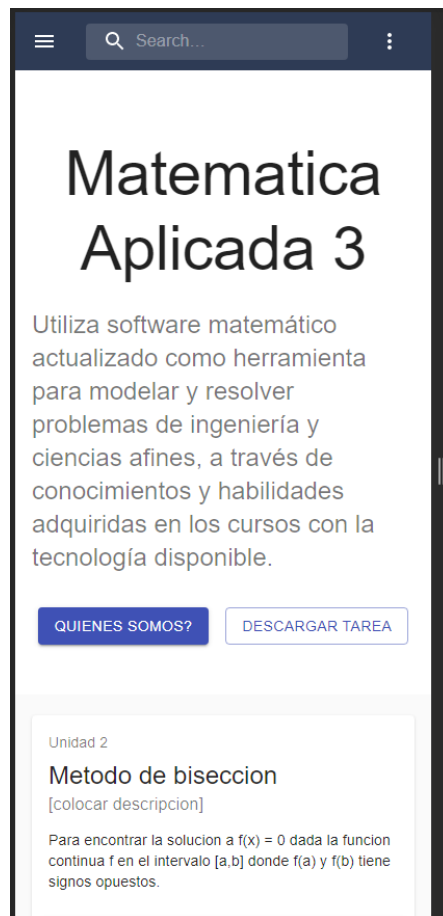
Fuente: elaboración propia, empleando software GeoGebra.

Apéndice 2. Captura de la interfaz de usuario



Fuente: elaboración propia, empleando software GeoGebra.

Apéndice 3. **Captura de vista *responsive* o adaptable a cualquier dispositivo**



Fuente: elaboración propia, empleando software GeoGebra.

Apéndice 4. **Vista del formulario que el cliente completa para realizar los procedimientos necesarios**

Unidad 2

Metodo de biseccion

[colocar descripcion]

Para encontrar la solucion a $f(x) = 0$ dada la funcion continua f en el intervalo $[a,b]$ donde $f(a)$ y $f(b)$ tiene signos opuestos.

Desplegar Calculador ▲

Ingrese funcion $f(x)$	Ingrese Tolerancia
Limite Inferior a	Max. de iteraciones
Limite Superior a	No. Decimales

CALCULAR

Fuente: elaboración propia, empleando software GeoGebra.

