**APLICACIÓN WEB QUE MEDIANTE EL USO DE REDES NEURONALES ARTIFICIALES RESUELVE PROBLEMAS DE OPTIMIZACIÓN NO LINEAL (GerMath.JS)**

POR:

WEYNDER GERMAN AGUIRRE BETANCOUR

COD: 0172010322

UNIVERSIDAD DE LA AMAZONIA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

FLORENCIA-CAQUETÁ

2020

**APLICACIÓN WEB QUE MEDIANTE EL USO DE REDES NEURONALES ARTIFICIALES RESUELVE PROBLEMAS DE OPTIMIZACIÓN NO LINEAL. (GerMath.JS)**

ESTUDIANTE:

WEYNDER GERMAN AGUIRRE BETANCOUR

Estudiante, Universidad de la Amazonia – Florencia, Caquetá

DIRECTOR:

Mg. LUBEIMAR EDUARDO GALLEGO RUIZ

Docente, Universidad de la Amazonia – Florencia, Caquetá

Propuesta de trabajo de grado para optar el título de Ingeniero de sistemas de la Universidad de la Amazonia

UNIVERSIDAD DE LA AMAZONIA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

FLORENCIA-CAQUETÁ

2020

**“El director y los jurados del presente Trabajo de Grado no son responsables de las ideas y conclusiones expuestas en éste; ellas son exclusividad de su autor”**

NOTA DE ACEPTACIÓN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FIRMA DEL JURADO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FIRMA DEL JURADO

**TABLA DE CONTENIDO**

LISTA DE FIGURAS 8

LISTA DE TABLAS 9

LISTA DE ECUACIONES 11

RESUMEN 12

ABSTRACT 14

INTRODUCCIÓN. 16

CAPÍTULO I. DESCRIPCIÓN GENERAL DEL PROYECTO. 18

PLANTEAMIENTO DEL PROBLEMA 18

JUSTIFICACIÓN 22

OBJETIVOS 24

OBJETIVO GENERAL 24

OBJETIVOS ESPECÍFICOS 24

CATEGORIZACIÓN DE VARIABLES 24

VARIABLE INDEPENDIENTE: 24

VARIABLE DEPENDIENTE: 24

CRITERIOS DE BÚSQUEDA 25

POBLACIÓN. 25

MARCO DE REFERENCIA 25

MARCO INSTITUCIONAL 25

Universidad de la Amazonia. 25

MARCO TEÓRICO 26

Problemas de optimización no lineal. 26

Optimización no lineal irrestricta 26

Método del Gradiente 27

Método de Newton 27

Optimización no lineal restricta 27

METODOLOGIAS DE DESARROLLO SOFTWARE. 29

METODOLOGIAS DE DESARROLLO AGIL 30

REDES NEURONALES ARTIFICIALES 34

Neuronas Biológicas: 34

Definición. 34

Perceptrón Multicapas (MLP) 38

Pesos de las Neuronas Artificiales 39

Función de entrada 40

Función de salida (activación) 40

Función de error de red 40

Resilient BackPropagation RProp 41

Aprendizaje Supervisado 42

CAPITULO II. LEVANTAMIENTO DE INFORMACIÓN. 43

DISEÑO METODOLÓGICO GENERAL. 43

*ACTIVIDADES Y HERRAMIENTAS DE LAS FASES.* 44

*FASE 1. ANÁLISIS DE LA INFORMACIÓN.* 44

Actividades de la técnica entrevista: 45

Construcción formato de entrevista. 45

Aplicación de la entrevista. 45

Transcribir la información al formato especificación de requerimientos. 45

Actividades de la técnica arqueología de documentos: 46

Búsqueda de literaturas que aborden los problemas de optimización no lineal. 46

Análisis de documentos. 47

Establecimiento de Requerimientos. 50

*FASE 2 DISEÑO DEL SISTEMA INFORMACIÓN* 54

Elaboración Historias de usuarios 54

CAPITULO III. USO DE HERRAMIENTAS TECNOLÓGICAS PARA EL DESARROLLO DE LA APLICACIÓN. 76

*FASE 3 CODIFICAR EL SISTEMA DE INFORMACIÓN* 76

Diseño de base de datos. 76

Elaborar Back-End y lógica de negocio. 77

Elaborar Front-End. 79

CAPITULO IV. IMPLEMENTACIÓN Y RESULTADOS. 80

*FASE 4 IMPLEMENTACIÓN DE PRUEBAS.* 80

PRODUCTOS ENTREGABLES 84

CRONOGRAMA 85

Cronograma de Desarrollo de la Aplicación Web GerMath.JS 85

RESULTADOS 88

CONCLUSIÓN 88

TRABAJOS FUTUROS 89

REFERENCIAS 90

# LISTA DE FIGURAS

Figura 1. Representación gráfica de la neurona biológica. 34

Figura 2. Modelo de Neurona de McCulloch y Pitts 36

Figura 3. Modelo de comparativo de neuronas (Biológica – Artificial). 37

Figura 4. Ejemplo Gráfico de Perceptrón Multicapa. 39

Figura 5. Logo Microsoft Word 2016. 48

Figura 6. Logo Microsoft Excel 2016. 49

Figura 7. Logo Plataforma Heroku. 49

Figura 8. Logo Plataforma GitHub. 50

Figura 9. Logo Node.JS. 77

Figura 10. Logo Visual Studio. 78

Figura 11. Esquema de carpetas código. 78

Figura 12. Logo de HTML 5, CSS3, JavaScript y Bootstrap. 79

Figura 13. Aplicación Web GerMath.JS en Heroku Running 80

Figura 14. Página de Inicio de Aplicación Web GerMath.JS 81

Figura 15. Página de Inicio de Sesión Aplicación Web GerMath.JS 82

Figura 16. Página de Optimización de Funciones Aplicación Web GerMath.JS. 82

Figura 17. Página de Resultado en Aplicación Web GerMath.JS 83

# LISTA DE TABLAS

Tabla 1. Resumen de los diferentes métodos de optimización no lineal. 28

Tabla 2. Comparativa entre metodologías de desarrollo ágiles y metodologías de desarrollo tradicionales. 31

Tabla 3. Función de activación Tangencial Hiperbólica. 38

Tabla 4. Formato de Historias de Usuario 46

Tabla 5. Literaturas seleccionadas de optimización no lineal 47

Tabla 6. Definición de requerimientos funcionales del proyecto 50

Tabla 7. Relación de requerimientos con la fuente. 52

Tabla 8. Historia de usuario (HU-001), Historia de Usuario Matiz. 54

Tabla 9. Historia de usuario (HU-002-001), Registrar Persona. 55

Tabla 10. Historia de usuario (HU-002-002), Crear Usuario. 56

Tabla 11. Historia de usuario (HU-002-002), Cerrar Sesion. 56

Tabla 12. Historia de usuario (HU-002-005), Desvincular Persona. 57

Tabla 13. Historia de usuario (HU-002-005), Activar Cuenta. 58

Tabla 14. Historia de usuario (HU-003-001), Modificar Usuario. 58

Tabla 15. Historia de usuario (HU-003-002), Listar Usuarios. 59

Tabla 16. Historia de usuario (HU-003-001), Seleccionar Usuario. 60

Tabla 17. Historia de usuario (HU-004-001), Crear Rol 61

Tabla 18. Historia de usuario (HU-004-002), Listar Roles 61

Tabla 19. Historia de usuario (HU-004-003), Establecer Rol. 62

Tabla 20. Historia de usuario (HU-004-004), Eliminar Rol 63

Tabla 21. Historia de usuario (HU-005-001), Graficar Función Ingresada. 63

Tabla 22. Historia de usuario (HU-005-002), Graficar Tabla de Resultados. 64

Tabla 23. Historia de usuario (HU-005-003), Graficar Resultado Individual. 65

Tabla 24. Historia de usuario (HU-005-004), Establecer Tema Visual 65

Tabla 25. Historia de usuario (HU-005-005), Desarrollar un Entorno Visual Intuitivo. 66

Tabla 26. Historia de usuario (HU-006-001), Registrar Función en Segundo Plano. 67

Tabla 27. Historia de usuario (HU-006-002), Modificar Iteraciones. 67

Tabla 28. Historia de usuario (HU-006-003), Modificar Grado de Error. 68

Tabla 29. Historia de usuario (HU-006-004), Limpiar Datos. 69

Tabla 30. Historia de usuario (HU-006-005), Establecer Tiempo Máximo de Respuesta. 69

Tabla 31. Historia de usuario (HU-007-001), Ingresar Función 70

Tabla 32. Historia de usuario (HU-007-002), Derivar Función 71

Tabla 33. Historia de usuario (HU-007-003), Solucionar Ecuaciones. 71

Tabla 34. Historia de usuario (HU-007-004), Soportar Números Simbólicos. 72

Tabla 35. Historia de usuario (HU-008-001), Graficar Datos Iterativamente. 73

Tabla 36. Historia de usuario (HU-008-002), Establecer el Criterio de Aprendizaje. 73

Tabla 37. Historia de usuario (HU-008-003), Aplicar Restricciones a la Función. 74

Tabla 38. Historia de usuario (HU-008-004), Demostrar Resultados. 75

Tabla 39. Resultados de Prueba Obtenidos 83

# LISTA DE ECUACIONES

Ecuacion 1. 27

Ecuacion 2. 27

Ecuacion 3. 36

Ecuacion 4. 36

Ecuacion 5. 38

Ecuacion 6. 40

Ecuacion 7. 40

Ecuacion 8. 40

Ecuacion 9. 41

# RESUMEN

El presente trabajo de grado tiene como objetivo generar una aplicación web que permita optimizar funciones de costo u objetivo, de carácter no lineal con la posibilidad de contener restricciones, haciendo uso de las Redes Neuronales Artificiales (RNA), lo cual favorecería las personas u organizaciones que lo necesiten, esto debido al crecimiento y posterior expansión natural de las industrias, más aún en nuestra región que cada a día avanza hacia el desarrollo tecnológico, en efecto lo que plantea este desarrollo es una ayuda computacional de acceso libre que mejore las ganancias, la productividad, los procesos organizacionales, entre otros.

Para esto se realizó una investigación documentada de los algoritmos, métodos y metodologías necesarias y suficientes para abordar este tipo de problemas de optimización, optando por las redes neuronales artificiales RNA como marco de trabajo, delimitando las restricciones con funciones de penalización, trasformando así la función objetivo original, en una nueva que abarque a cabalidad las restricciones que pudieran presentarse, es de aclarar que para este desarrollo se utilizó la metodología XP(Extreme Programming), siendo el requerimiento fundamental del proyecto la culminación del objetivo antes mencionado.

Ahora bien, la culminación del objetivo mide la eficacia de la RNA, no obstante, es necesario generar un entorno adecuado para la correcta implementación de la aplicación, lo cual es de igual importancia que la RNA misma, por tanto, se utilizaron tecnologías de uso libre tales como JavaScript, Node.Js, MariaDB, Express.JS y HanddleBars Framework para brindar el soporte estructural que requería la aplicación y su correcto despliegue, permitiendo a su vez la gestión de usuarios, la interacción entre las diferentes opciones configurables de la RNA, y el acceso a los contenidos teóricos que fundamentan la misma.

En conclusión se desarrolló una aplicación que hace uso de redes neuronales artificiales para brindar respuesta a un problema de optimización no lineal, que a su vez ayude a la toma de decisiones organizacionales, estableciendo el grado de error del resultado en ; por otra parte y debido a los conjuntos de prueba establecidos en el marco conceptual del desarrollo, (ejercicios propuestos en las bibliografías exploradas) se delimito a 2 la cantidad de variables que puede operar la red neuronal, esto debido al coste computacional y al tiempo de respuesta de la aplicación al aumentar la cantidad de datos operables, de esta manera se garantiza una respuesta veraz, eficiente y rápida al utilizar GerMath.JS.

**Palabras clave:** Redes Neuronales Artificiales, Optimización No Lineal, Optimización por Gradiente

# ABSTRACT

The present work of grade aims to generate a web application that allows optimizing cost or objective functions, of a non-linear nature with the possibility of containing restrictions, making use of artificial neuronal networks (RNA), which would favor people or organizations that they need it, this due to the growth and subsequent natural expansion of the industries, even more in our region that each day progresses towards technological development, in fact what this development raises is a computational help of free access that improves profits, Productivity, organizational processes, among others.

For this, a documented investigation of the necessary and sufficient algorithms, methodologies and methodologies was carried out to address this type of optimization problems, opting for artificial neural networks RNA as a framework, delimiting restrictions with penalty functions, thus transforming the function Original objective, in a new one that fully encompasses the restrictions that could be submitted, is to clarify that for this development the XP methodology (Extreme Programming) was used, the fundamental requirement of the project is the culmination of the aforementioned objective.

However, the culmination of the target measures the effectiveness of the RNA, however, it is necessary to generate an adequate environment for the correct implementation of the application, which is equally important as the RNA itself, therefore, technologies for use were used free such as JavaScript, Node.JS, MariaDB, Express.js and HanddleBars Framework to provide the structural support that required the application and its correct deployment, allowing the management of users, the interaction between the different configurable options of the RNA , and access to the theoretical content that base it.

In conclusion, an application was developed that uses artificial neural networks to provide response to a problem of non-linear optimization, which in turn helps organizational decision making, establishing the degree of result in ; On the other hand, and due to the test sets established in the conceptual development framework, (exercises proposed in the explored bibliographies) it was defined to 2 the number of variables that the neuronal network can operate, this due to the computational cost and at the time of Answer of the application by increasing the amount of operable data, in this way a truthful, efficient and fast response is guaranteed when using GerMath.JS.

**Keywords:** Artificial neural networks, non-linear optimization, gradient optimization

# INTRODUCCIÓN.

El trabajo de grado, “*APLICACIÓN WEB QUE MEDIANTE EL USO DE REDES NEURONALES ARTIFICIALES RESUELVE PROBLEMAS DE OPTIMIZACIÓN NO LINEAL*”, tiene como objetivo fundamental, el resolver problemas de optimización no lineal, ya sea con la presencia o no de restricciones, todo esto enmarcado en una estructura web que soporte el uso asíncrono de la misma, para dar cumplimiento a nuestro objetivo primordial hacemos uso de la metodología eXtreme Programming (xP), la cual nos divide en fases iterables y escalables el desarrollo e implementación de la aplicación, siendo así, se establecieron una serie de capítulos que abarcan a cabalidad las fases, procesos y procedimientos utilizados para alcanzar el objetivo, tenemos entonces:

***CAPITULO I****:* En este capítulo se abordarán los temas relacionados a la base referencial y conceptual del proyecto, entre ellos, las diferentes opciones teóricas a la hora de optimizar funciones no lineales, como también las metodologías de desarrollo adaptables al nuestro proyecto, se realizó una revisión bibliográfica de las metodologías agiles, la optimización no lineal, Matemáticas meta-heurísticas, entre otras. A la par, se realizó una recopilación literal de las implementaciones existentes de las redes neuronales artificiales y su aportación en diferentes países y ciudades de nuestro país, documentándolas y estableciendo el punto de inflexión en donde nuestra aplicación generaría un punto de alto impacto.

***CAPITULO II:***En esta instancia, y siguiendo la metodología escogida(xP), se realizó la gestión de historias de usuario en torno al objetivo fundamental, identificando los requerimientos funcionales y no funcionales inherentes al desarrollo iterativo del proyecto. Es de aclarar que la aplicación surgió como respuesta a una necesidad generalizada de la región, con el fin de brindar una herramienta que les permitiera hacer uso de la optimización en sus procesos, por tanto, el diseño de la interfaz y las opciones adicionales se implementaron para ofrecer una mejor experiencia al usuario, desplegando la información de forma clara y concisa para la futura toma de decisiones.

***CAPÍTULO III:*** Se establecen las herramientas de desarrollo y despliegue que aportan el soporte estructural de la aplicación servida en <https://germathes6.herokuapp.com/>, estando desarrollada en su totalidad con software libre, Programación Orientada a Objetos en JavaScript y EcmaScript6 .

***CAPÍTULO IV:***Siguiendo con los objetivos establecidos se verificó que el sistema cumpliera con su correcta funcionalidad, siendo probado con los ejercicios existentes en las bibliografías de referencia, por tanto, se generó una tabla de resultados con los ejercicios resueltos por la Red Neuronal Artificial “GerMath.JS” notándose una mejoría en los valores existentes anteriormente, en cuanto a los trabajos futuros, se destaca la ampliación del espectro de variables para aumentar el rango de utilidad sin aumentar el tiempo de respuesta.

# CAPÍTULO I. DESCRIPCIÓN GENERAL DEL PROYECTO.

# PLANTEAMIENTO DEL PROBLEMA

Basados en el avance de las tecnologías de la información a lo largo del tiempo, y su auge significativo en los años 80’s y 90’s, siendo determinante en los procesos de creación e innovación de los viajes espaciales, el desarrollo de la robótica, la *Inteligencia**Artificial* **(*IA*)**, la *Investigación de Operaciones*, entre otras. Teniendo en cuenta que la **IO** se considera un campo de estudio bastante extenso, para este desarrollo haremos énfasis en su rama de *Programación no Lineal*más exactamente en los problemas de *Optimización No Lineal* ***(ONL).***

Observamos entonces que. Los problemas de optimización hacen parte fundamental del desarrollo en las organizaciones (Boirivant, 2009), aplicándose en diferentes campos tales como en la administración de recursos o materiales (Hidalgo & Díaz, 2010; Serna, Serna, & Ortega, 2010), la logística (Mejía & Elkin, 2007; Héctor & Briceño, 2011), en las redes de transporte (Trujillo, 2019), en optimización financiera (Jaramillo, Vera, & Barcia, 2018; Ospitia & Cortes, 2019), entre otras, debido a que desde su descubrimiento se generaron importantes y significativos avances frente a la solución óptima de estos problemas, ya que existen algoritmos, metodologías y/o métodos exactos para optimizar funciones de carácter lineal (Ploskas & Samaras, 2014). En contraste con la optimización de funciones no lineales donde no es posible establecer un método que aporte una solución óptima al problema cuando en la función objetivo o en las restricciones de la misma se carece de linealidad, siendo necesario aplicar ***Métodos Heurísticos***[[1]](#footnote-1) de optimización (Gutierrez & Pulido, 2015), los cuales buscan mediante la utilización conjunta de diferentes alternativas brindar una aproximación de gran calidad a la solución óptima.

Teniendo en cuenta lo anterior, considerando el actual y acelerado crecimiento de las organizaciones, se considera indispensable contar con medidas preventivas y correctivas que ofrezcan soluciones eficientes a los problemas de optimización que se pudieran presentar sobre los modelos comerciales establecidos (Henao & Dumar, 2007; Sepúlveda, Avilez, & Jaramillo, 2014; Valdes, Aleaga, & Vidal, 2014), visto de otra forma, es necesario contar con una herramienta que se adapte a las necesidades que puedan surgir de manera esporádica o administrativa en torno a la toma de decisiones, la ejecución de procesos productivos, la minimización de costos de producción, o también para determinar la afectación organizacional debido a cambios en el valor de la materia prima, entre otros. Cabe resaltar que mediante el modelado matemático se pueden generar funciones objetivo o funciones de costo en las cuales intervienen las variables necesarias en los procesos mencionados anteriormente, usualmente estas funciones carecen de linealidad.

Dentro de los algoritmos o métodos reconocidos ***Heurísticamente***[[2]](#footnote-2) (Menna, 2014) para resolver este tipo de problemas (ONL) encontramos los *Algoritmos Genéticos* (Wang, 2014), *Optimización por* *Enjambres* (Lin Li, 2014), la *Optimización Basada en Colonias de Hormigas* (Ínkaya, Kayalıgil, & Özdemirel, 2014). En relación a lo anterior para este desarrollo aplicaremos el concepto de las *Redes Neuronales Artificiales (****RNA)*** buscando aplicar este paradigma en el desarrollo de una aplicación que garantice la correcta solución de los problemas ONL, teniendo en cuenta, tanto la función de objetivo, como las restricciones explicitas e implícitas que pudieran afectar dicha función limitando la zona factible de búsqueda.

Es de notar que las redes neuronales en Colombia ya han sido exploradas, estas se han utilizado en mayor manera para resolver problemas de clasificación, o regresión lineal, también en la predicción de eventos, como es el caso de la ciudad de Cali donde se diseñó e implemento una RNA para predecir la precipitación mensual en su cuenca hídrica principal, de esta manera se logró monitorear mediante 35 estaciones que enviaban datos en tiempo real a la Red, esta clasificaba la información y aportaba una predicción futura en base a los datos analizados, de esta manera se logró determinar mediante simulación a lo largo del tiempo los futuros cambios que debían realizarse y en qué zonas específicas reforzar los planes de contingencia para mitigar de mejor manera los riesgos en la población (Daniel David Montenegro Murillo, 2018).

Otro campo en el cual se ha desempeñado este paradigma en el país es en problemas de clasificación como se evidencia en un proyecto publicado por la universidad Javeriana de Colombia, donde se clasifican 200 empresas según sus aseguradoras de riesgos ARL el número de accidentes ocurridos, las indemnizaciones pagadas a los empleados, las fechas de entrada y retiro de los trabajadores, siendo las entradas de la red los datos existentes en *FASECOLDA*[[3]](#footnote-3) y el Sistema de Riesgos Laborales[[4]](#footnote-4), en este caso se logró determinar que el 85% de las empresas elegidas aleatoriamente realizaban fraude ya sea en la vinculación o en el pago de las debidas indemnizaciones o remuneraciones (Hernández, 2017).

Sin embargo, las aplicaciones antes mencionadas de las RNA se aplican en el campo lineal, clasificando los datos o prediciendo futuros comportamientos, esto conlleva a la necesidad de generar una solución que facilite la optimización de funciones no lineales por parte de las empresas colombianas que en su expansión continua necesiten hacer uso de este paradigma para maximizar sus ganancias, minimizar sus costos u optimizar sus procesos, como por ejemplo los Problemas De Asignación Generalizada (GAP) (Gutierrez C. S., 2019) en donde pueden encontrarse restricciones de capacidad no lineal al momento de optimizar la asignación de recursos, también se encuentran casos no lineales de optimización para los factores ambientales como lo es, la Planificación de los Recursos Hídricos(Villavicencio, Arumí, & Holzapfel, 2011) optando por maximizar el beneficio neto del agua teniendo en cuenta las restricciones físicas, medioambientales y económicas que pudieran surgir. El sector industrial hace uso igualmente de los métodos de optimización no lineal (Mibelli, 2005), definiendo el costo de producción como función objetivo, logrando mediante modelos matemáticos minimizar este valor, del mismo modo el sector agrícola Colombiano para maximizar su producción hace uso de funciones con objetivos múltiples, de las cuales es de destacar el cálculo del riesgo financiero que carecen de linealidad, al igual que las funciones inherentes al terreno de siembra y a la fabricación de fertilizantes (Sarmiento, 2018). Por tanto, es necesario implementar una Aplicación Web de fácil acceso, que ayude a las organizaciones a resolver sus problemas de optimización empresarial aportando soluciones efectivas y eficientes para la toma de decisiones por tanto debemos dar solución al siguiente interrogante.

¿Cómo resolver problemas de optimización no lineal restricta, haciendo uso del paradigma de redes neuronales artificiales?

# JUSTIFICACIÓN

En vista del acelerado crecimiento de la población en Colombia según la revista Forbes (Staff, 2020) la cual pronostica basada en los datos poblacionales del Departamento Administrativo Nacional de Estadística (DANE) que para el año 2020 habrá en Colombia alrededor de 50,3 millones de habitantes, se genera una expansión en el desarrollo industrial, comercial y financiero de las organizaciones, las cuales deben estar preparadas para afrontar estos cambios, ya sea al hacer escalable su capacidad de producción para adaptarse a la demanda cambiante de la población objetivo, o realizando cambios logísticos en la manera como se fabrica el producto. Es de vital importancia contar con herramientas eficaces y eficientes que apoyen la toma de decisiones organizacionales, como la minimización de costos de producción, tiempos de atención al usuario, la maximización de las utilidades por producto, o de las ganancias netas del lote de producción.

A su vez, con el incremento poblacional, la tecnología también se ha visto en la necesidad de evolucionar de manera rápida, supliendo las necesidades que surgieron en esta expansión. De esta forma surgen los Sistemas de Información, que en su abrumante y acelerado cambio se han adaptado rápidamente a todos estos escalones de la modernización, entre ellos tenemos a Las Aplicaciones Web (Gil, 2008) los Servicios Web que para efectos de este desarrollo, nos proporcionaran el soporte estructural en el cual estará desplegada nuestra aplicación, disponible para ser consumida mediante cualquier dispositivo con conexión a internet, teniendo presente el buen manejo de software máquina a máquina que facilita una interoperabilidad asíncrona entre el cliente y el servidor, es por esto que las organizaciones comerciales han optado por migrar sus procesos, inventarios, y ventas a plataformas web que garanticen su expansión comercial, notándose que actualmente para las empresas es cada vez más importante contar con estas alternativas.

Esta investigación tiene como finalidad, desarrollar una aplicación Web, que haciendo uso del paradigma de Redes Neuronales Artificiales resuelva los problemas de optimización no lineal restricta que puedan generarse en estas organizaciones, notándose que no existe un método exacto para optimizar funciones objetivo NO lineales con restricciones, por tanto, para realizar este procedimiento haremos uso de la Heurística, combinando el *Método del Gradiente Descendente* y el algoritmo *Resilient Back Propagation* para así brindar una solución óptima y factible para su posterior análisis, apoyando toma de decisiones gerenciales. De esta manera lo que se busca es contribuir a la expansión empresarial en el país, enfocándonos principalmente en el departamento del Caquetá, siendo este desarrollo una importante herramienta que, con su debida utilización contribuirá a la población antes mencionada en la optimización de sus procesos, costos y utilidades, mejorado la calidad empresarial en la región y por consiguiente a la población en general, la cual será beneficiada por la expansión empresarial y laboral.

# OBJETIVOS

## OBJETIVO GENERAL

Desarrollar una Aplicación Web, que mediante el uso de Redes Neuronales Artificiales resuelva problemas de Optimización no Lineal Restricta.

## OBJETIVOS ESPECÍFICOS

* Determinar a partir de las referencias existentes, los métodos necesarios para resolver Problemas de Optimización No Lineales Restrictos.
* Implementar una Aplicación Web de fácil acceso y uso que resuelva problemas de optimización no lineal Restricta haciendo uso de RNA.
* Evaluar el desempeño de la Red Neuronal Artificial en base a los problemas de optimización más comunes conocidos en las literaturas de referencia.

# CATEGORIZACIÓN DE VARIABLES

## VARIABLE INDEPENDIENTE:

* Optimización de Funciones No Lineales

## VARIABLE DEPENDIENTE:

* Redes Neuronales Artificiales.
* Gradiente Descendente.
* Función de Penalización.

## CRITERIOS DE BÚSQUEDA

* Aplicación Web AND Optimización No Lineal
* Optimización No Lineal con Restricciones
* Gradiente Descendente AND Optimización No Lineal.

# POBLACIÓN.

La población de estudio es de carácter libre, debido a la naturaleza misma de la aplicación web, por tanto, el tipo de muestreo es no probabilístico de carácter intencional, de esta forma cualquier persona que desee hacer uso de la aplicación podrá hacerlo siempre y cuando cuente con una conexión a internet para acceder a la aplicación.

# MARCO DE REFERENCIA

## MARCO INSTITUCIONAL

### Universidad de la Amazonia.

“La universidad de la Amazonia, tiene su sede principal en la Ciudad de Florencia – Caquetá. Es una institución estatal de educación superior, vinculada al Ministerio de educación Nacional y reconocida como Universidad según la resolución No. 6533 de 1983 por el ministerio de Educación Nacional” (Universidad de la Amazonia, 2002).

## MARCO TEÓRICO

### Problemas de optimización no lineal.

La optimización de funciones busca minimizar o maximizar una función objetivo según requerimientos de cada organización, las funciones de carácter lineal poseen métodos exactos para optimizar el valor de las incógnitas y el generado por estas en la función objetivo, garantizando que el valor hallado es el máximo o mínimo (según corresponda) que pudiera encontrarse en la función, estando sujeta o no, a restricciones. Por otra parte, en las organizaciones surgen problemas de optimización que exceden el comportamiento lineal, ya sea en la función objetivo o en las restricciones de la misma, lo cual hace insuficiente la utilización de los métodos tradicionales de optimización como el Método Simplex, (Allende & Bouza, 2005) o el Método Grafico, por tanto se debe optar en estos casos por métodos o algoritmos heurísticos que garanticen la fiabilidad del resultado, por lo regular estos algoritmos aproximan de manera iterativa e incremental los valores de las incógnitas hasta mejorar gradualmente el resultado, no obstante, existen otros algoritmos que utilizan valores de contrapeso para eliminar las restricciones no lineales de la ecuación con el objetivo de tratar la función como netamente lineal.

### Optimización no lineal irrestricta

Se les considera también métodos de descenso, estos métodos de optimización utilizan el Gradiente ( como medida de eficiencia operando hasta que este, o en su defecto el valor de parada del error lo permitan, es de aclarar que aunque ambos métodos se fundamenten en el concepto antes mencionado, la forma como se abordan y recorren las iteraciones hasta el resultado demarca diferencias significativas. El método del gradiente al realizar iteraciones sucesivas refleja un comportamiento ortogonal, este a su vez genera un efecto de zig-zag mientras se acerca al óptimo, por otra parte, el método de newton posee un caso especial al solucionar funciones del tipo cuadrático ya que estas requieren solo de una iteración para alcanzar el óptimo, sin embargo, este método también puede llegar a divergir si su primera iteración se sitúa en un punto lejano del optimo global. (Gely, 2009)

### Método del Gradiente

Este método se enfoca en establecer la dirección máxima de ascenso basándose en el valor del gradiente resultante al derivar dicha función, de igual manera el máximo descenso se obtiene gracias a (-) el gradiente, siendo así este método puede iterar progresivamente en busca de un mínimo o un máximo según se ajuste a las necesidades de la *Función Objetivo*.

###### 

(1)

### Método de Newton

Este método utiliza la segunda derivada de la función objetivo para trazar la dirección en la que se desplazará y aproximará al resultado óptimo, se fundamenta principalmente en aplicar *Series de Taylor* de 2do orden en la función objetivo, luego se minimiza esta aproximación igualando su gradiente a cero, haciendo recursivo el método, de esta manera el grado de error permitido determinará si se realizan o no más iteraciones.

###### 

**(2)**

### Optimización no lineal restricta

Al establecer restricciones funcionales, explicitas o implícitas en los problemas de optimización no lineal, se desestima el hallazgo de algún mínimo local cuyo gradiente sea igual a cero, ( = 0) ya que el punto hallado pudiera estar o no, fuera de la frontera de decisión, ahora bien si su resultado es óptimo respecto a las variables, este sería inviable respecto a las restricciones, por tanto se debe migrar de manera ágil a otros mínimos locales que cumplan a cabalidad con las restricciones de la función objetivo. Teniendo en cuenta lo anterior se han desarrollado métodos de optimización no lineal pertenecientes a las corrientes clásicas de la Investigación de Operaciones, estas tienden a transformar en lineales las restricciones para así poder aplicar los métodos clásicos de optimación lineal.

1. Resumen de los diferentes métodos de optimización no lineal.

Resumen de los diferentes métodos de optimización no lineal

|  |  |  |  |
| --- | --- | --- | --- |
| **Problema** | **Condición Necesaria** | **Condición Suficiente** | **Técnica** |
|  | ( = 0) | convexa | Método del Gradiente |
| Método de Newton |
|  |  | convexa  convexa | LaGrange KKT |
| Multiplicadores de LaGrange |
|  |  | convexa  Lineales | LaGrange KKT |
| Multiplicadores de LaGrange |

Fuente: Adaptado del articulo *Optimización de Funciones No lineales* de la Facultad de Ciencias Físicas y Matemáticas de Universidad de Chile. (Goic, 2005)

## METODOLOGIAS DE DESARROLLO SOFTWARE.

Si hablamos de la metodología de un proyecto tradicional, se basará en planear a cabalidad los detalles rigurosos que pudieran afectar en el desarrollo de las actividades o fases, sin embargo esto tiende a convertirse en un problema en el momento en que surjan cambios específicos, correcciones en los requerimientos, o al modificar el enfoque del proyecto que se esté desarrollando, sumado a esto el paradigma sistemático y pragmático de las nuevas tecnologías hace necesario contar con capacidades de adaptabilidad a cambios súbitos o esporádicos que llegaren a ser requeridos, a diferencia de las metodologías de desarrollo tradicional, estas a causa de su naturaleza iterativa deben completar una iteración antes de establecer los cambios que se deben realizar, luego de esto, planificar nuevamente los costes, duración y alcance del proyecto, esto claramente retrasaría cada una de las fases siguientes del desarrollo, lo cual es catastrófico si hablamos de presupuestos reducidos para afrontar problemas en empresas pequeñas y medianas(pymes).

En respuesta a lo anterior, se han desarrollado metodologías, no tan robustas que brindan mayor libertad y ergonomía al permitir la integración de modificaciones en los requerimientos por parte del cliente sin afectar los diferentes procesos que se estén llevando a cabo. Este es el concepto de metodología ágil de desarrollo que permite la adaptación rápida a cada cambio que pretenda el cliente, debido a que él hace parte del equipo de desarrollo, participando activamente en reuniones, aportando su punto de vista en cada problema tan pronto ocurra, una solución en conjunto en la cual intervienen tanto los desarrolladores, el diseñador y cada uno de los integrantes del equipo de desarrollo.

Lo cual al final logra que al usuario se le entregue lo que realmente busca, ya que fue el mismo quien superviso y aporto las ideas en cada fase del desarrollo, y puesta en marcha de la aplicación, garantizando aun después de su implantación su mejora constante, hasta que todas y cada una de las necesidades o historias de usuario del cliente sean satisfechas, llegando hasta la fase de muerte donde se le otorga total control al cliente y solo se bridaría soporte en caso de necesitarse. Siendo así, las metodologías que se pueden o no utilizar ya dependen de cada desarrollador o grupo de desarrolladores, pero en el campo de la programación practica se utilizan metodologías agiles para proyectos a corto plazo, y en proyectos bastante grandes se recomiendan metodologías tradicionales, dependiendo de cada caso.

### METODOLOGIAS DE DESARROLLO AGIL

Desde 2001 se fundó “*The Agile Aliance”* una organización sin ánimo de lucro dedicada a promover los conceptos relacionados con el desarrollo ágil, integrada en si por 17 expertos en el área del desarrollo de software para entonces, de esta forma en EEUU se empezaron a dar pasos agigantados en vías de construir una sólida metodología para el desarrollo de todo tipo de aplicaciones o proyectos, para esto se creó en ese entonces y se conserva hasta ahora, **El Manifiesto Ágil.** (Beck, 2001)

Enfocados claramente en encontrar la forma de modelar, organizar, y bridar la seguridad organizativa que pudiera encontrarse en la competencia, fue así como paso a paso se tomaron partes funcionales de otras metodologías y se fue creando un consolidado de principios y reglas que se deben seguir, en las cuales se rige esta metodología, donde priman los intereses del cliente, la adaptabilidad, y la flexibilidad a los cambios, donde la integración, factorización y encapsulamiento priman sobre temas arbitrarios como la documentación, donde se puede generarse cambios por petición del cliente, por refactoring organizacional o haciendo reingeniería a un método obsoleto véase Tabla 2.

1. Comparativa entre metodologías de desarrollo ágiles y metodologías de desarrollo tradicionales.

Comparativa entre metodologías de desarrollo ágiles y tradicionales.

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

Fuente: Tomade del artiíulo Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP) de la Universidad Politécnica de Valencia. Pág. 17.

Para este desarrollo escogimos la metodología de desarrollo Extreme Programming de carácter **Ágil**:

**Metodología de Desarrollo XP (Extreme Programming):** esta metodología establece como una característica fundamental el mantener las relaciones interpersonales como factor fundamental en el desarrollo, de esta manera se valora de manera positiva el trabajo en equipo, propiciando un buen clima de trabajo con comunicación directa y fluida entre todos los participantes. XP se caracteriza por su realimentación continua entre el cliente y el equipo de desarrollo, simplificando las soluciones para que sean más fáciles de implementar en caso tal de requerirse un cambio, esta metodología se logra agrupar en tres pilares fundamentales (Beck, 2001):

**Historias de Usuario:** Es la forma como XP identifica los requerimientos, son tarjetas de papel en las cuales el cliente plasma brevemente los requerimientos o características que el sistema deberá poseer, pueden ser requisitos funcionales o no funcionales. Es de aclarar que gracias a la flexibilidad de la metodología en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, se debe delimitar cada historia de usuario de tal manera que los programadores puedan implementarla en unas semanas

**Roles:** Aunque en las diferentes literaturas aparecen algunas variaciones y extensiones de roles XP, describiremos los roles de acuerdo con la propuesta original de Beck

*Cliente***:** Responsable de definir y conducir el proyecto, de igual manera los objetivos.

*Programadores***:** Estiman el tiempo y el coste del proyecto, además del desarrollo.

*Tester***:** Encargado de pruebas.

*Tracker***:** Encargado del seguimiento.

*Big**Boss***:** Gestor del proyecto.

**Proceso**: El proceso cíclico de desarrollo software en resumen consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.

2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las

restricciones de tiempo.

4. El programador construye ese valor de negocio.

5. Vuelve al paso 1.

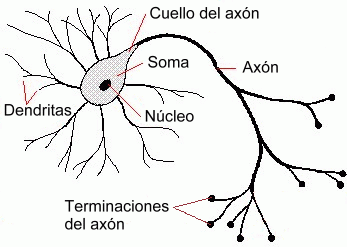
Teniendo en claro la flexibilidad de las fases de esta metodología, es de vital importancia resaltar que en todas las iteraciones que pudieran presentase en un ciclo tanto el cliente como el desarrollador deben establecer las prioridades y riesgos que pudieran surgir mientras avanza el proyecto. Es por esta planificación inicial que no se debe presionar al programador a realizar más trabajo que el estimado, ya que esto supone una pérdida de calidad en el software o un retraso en los tiempos de entrega del proyecto. El ciclo de vida ideal de XP consta de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

## REDES NEURONALES ARTIFICIALES

Neuronas Biológicas: La neurona es la unidad funcional y estructural del sistema nervioso, produce y transmite impulsos eléctricos, estos a su vez propagan y almacenan información para posteriormente ser usada en el razonamiento o en la toma decisiones, véase Figura 1. Se encuentra formada por tres partes: el cuerpo neural o soma, el axón, las dendritas y zonas de conexión entre una neurona y otra llamada, sinapsis. (Boeree, 2009)

1. Representación gráfica de la neurona biológica.

*Representación gráfica de la neurona biológica*



Fuente: *Tomado del libro Atmosferas en el Capítulo VI pág. 3-4* (Universidad de Murcía, 2014)

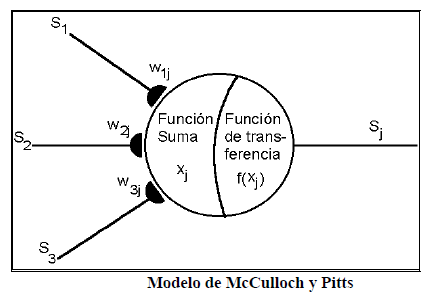
### Definición.

Las Redes Neuronales Artificiales (RNA) son modelos inspirados en la estructura neuronal del sistema nerviosos, debido a esto, intentan replicar su comportamiento, no obstante sus métodos se pueden establecer como la abstracción computacional de una serie de características propias del cerebro humano, las RNA en su principio básico logran aprender mediante la iteración, almacenando la experiencia de eventos pasados para aplicarla a eventos futuros, las RNA pueden ser entrenadas para llevar a cabo tareas específicas, también se puede ejercer sobre ellas un entrenamiento iterativo que como explicamos anteriormente conlleva a un aprendizaje, y posterior a eso la adaptación del conocimiento en una mejora continua, es de notar que existen diferentes algoritmos de aprendizaje como son: el aprendizaje supervisado, el no supervisado y el aprendizaje por refuerzo, (Olabe, 2010) cada uno de ellos con características diferentes en cuanto a la forma de abordar el criterio de error en base a la salida deseada, siendo el aprendizaje no supervisado el que asemeja de manera más lógica el comportamiento biológico del cerebro, sin embargo ambos algoritmos de entrenamiento tienen como objetivo el correcto aprendizaje de la red neuronal en base a el conocimiento impartido.

Las RNA se componen de neuronas conectadas entre sí, estas transmiten información de forma innata, reciben, procesan y envían datos entre sí, estas neuronas interconectadas responden al aprendizaje impartido, retornando una salida con la respuesta a la tarea específica para la cual fue entrenada la red. Uno de los primeros modelos matemáticos de una neurona fue el propuesto por McCulloch y Pitts en 1.943 (Pitts & Mcculloch, 1943) y en él, se basan las redes neuronales actuales, en este modelo cada neurona consta de un conjunto de entradas ,y una sola salida,, cada entrada es afectada por un coeficiente denominado peso y que se representa por la letra . El subíndice refleja que el peso afecta a dicha entrada, y hace alusión a la neurona en ese determinado momento, véase *Figur*a *2*.

1. Modelo de Neurona de McCulloch y Pitts

*Modelo de Neurona de McCulloch y Pitts.*



Fuente: Tomado del Articulo *Redes Neuronales Artificiales* de la Universidad Nacional de Colombia pág. 10 (Torres, 2011)

Entonces: la cantidad referente a la suma del producto de cada entrada multiplicada por su peso respectivo se denomina donde es el indicador de cada uno de los productos antes definidos, luego todos y cada uno de los valores de son multiplicados por la *Función de Transferencia* logrando así la “***Activación****”* de los valores obtenidos, y generando una salida

= es un valor “*umbral”*

###### 

**(3)**

De esta forma:

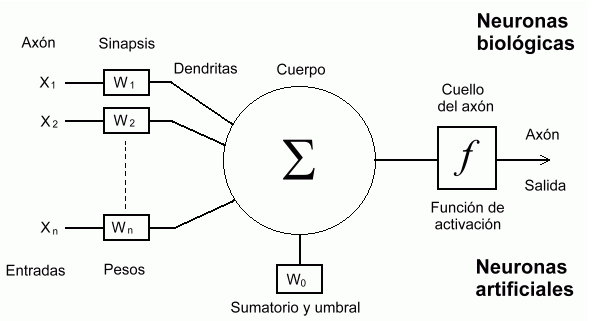
###### 

**(4)**

La Figura 3, expone un ejemplo de modelo neural con entradas, que consta de: Un conjunto de entradas **,** los pesos sinápticos , correspondientes a cada entrada, una sumatoria **,** que almacenara los productos de **,** una función de activación . La cual mantiene el conjunto de valores de salida normalmente entre (0,1) o (-1,1), haciendo uso de la función sigmoidea, véase *Tabla 2*, y una salida .

1. Modelo de comparativo de neuronas (Biológica – Artificial).

Modelo de comparativo de neuronas (Biológica – Artificial)



Fuente: Tomado del Website *Redes Neuronales: una visión superficial* de la Universidad de Sevilla, (Caparrini, 2018).

Las entradas son el estímulo que la neurona ha recibido del entorno, y la salida es la respuesta a tal estímulo. La neurona puede adaptarse y aprender del medio en que se encuentre, modificando el valor de sus pesos sinápticos, por ello son conocidos como los parámetros libres del modelo, ya que pueden ser modificados y adaptados, en este modelo, la salida neuronal está dada por:

###### 

**(5)**

La función de activación establecida de acuerdo con la tarea realizada por la neurona fue la Función de Activación tangencial Hiperbólica.

***Función de Activación Tangencial Hiperbólica:*** esta función de activación surgió como solución al problema de la simetría respecto a los valores negativos presentados en la Función de Activación Sigmoidea, por tanto es una función de activación no lineal, su acotamiento se encuentra en -1 y 1 logrando así generar salidas negativas en base a la función objetivo a ajustar, lo cual a su vez se potencia con el algoritmo del gradiente descendente, haciendo de esta una de las funciones de activación más utilizada en la I.A.

1. Función de activación Tangencial Hiperbólica.

*Función de activación Tangencial Hiperbólica*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Función | Rango | Gráfica |
| Tangente Hiperbólica |  |  |  |

Fuente: Adaptado del artículo *Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento* de la revista *Odeón,* Número 9 pág. 121. (Anzola, 2015)

### Perceptrón Multicapas (MLP)

Es un tipo de red neuronal que tiene una capa de entrada y una capa de salida, y una o más capas ocultas en medio. Cada capa tiene un número de neuronas fijo, pero potencialmente diferente, es decir, será fijo durante todas iteraciones o épocas de la capacitación, después de obtener el resultado esperado, o que el umbral de error sea lo suficientemente pequeño para determinar que la solución es aceptable e idónea, se podrá modificar tanto la cantidad de parámetros de entrada, las épocas de capacitación o el objetivo a resolver. Cada conexión de neuronas tiene un peso, una función de entrada y una función de salida, véase Figura 4.

1. Ejemplo Gráfico de Perceptrón Multicapa.

*Ejemplo Gráfico de Perceptrón Multicapa*



Fuente: Tomado del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad del País Vasco, (Larranaga, Inza, & Moujahid, 2005)

### Pesos de las Neuronas Artificiales

En una red biológica, se envía una señal eléctrica por el axón hasta las dendritas de las neuronas. La fuerza de la señal determina la cantidad de influencia que la neurona encendida tiene sobre las demás neuronas, las RNA son el análogo a este tipo de red biológica, y simula la fuerza de la señal por medio del peso de la conexión de salida, que claramente podemos definir, pero a su vez la red neuronal puede ajustar según sus necesidades. De esta forma el peso se reparte entre la capacidad de ajustarse entre épocas de capacitación, y la fuerza de la señal, que denota la influencia en la red.

### Función de entrada

El tipo de función de entrada más habitual se llama Suma Ponderada. Una neurona de entrada está conectada a neuronas de la capa anterior de la red, la suma ponderada de las entradas para se calcula añadiendo el producto del valor de entrada de cada conexión por veces el peso de la conexión a lo largo de todas las entradas . (Larranaga, Inza, & Moujahid, 2005).

###### 

**(6)**

### Función de salida (activación)

El tipo de función de activación que se utilizo fue la Función de Activación Tangencial Hiperbólica antes mencionada (véase Tabla 2) , tenemos entones que para una suma ponderada en una neurona determinada, el valor tangencial de , es calculado por:

###### 

**(7)**

### Función de error de red

La función de error más usada con las redes MLP es la función de error cuadrático medio. La cual calcula la *"distancia"* media entre el valor real que el programa calcula, y el valor esperado de los datos de entrenamiento, dadas neuronas de salida, para cada suma ponderada de la salida , el programa de capacitación calcula la diferencia entre el valor de los datos de capacitación y el valor de la red, lo eleva al cuadrado, suma esos valores de todas las neuronas de salida y lo divide por el número de neuronas de salida para llegar al error de salida total . (IBM, 2001).

###### 

**(8)**

### Resilient BackPropagation RProp

RProp es una técnica utilizada en el entrenamiento supervisado de redes neuronales artificiales el cual se basa en la búsqueda de la derivada de ; en este caso, establece una relación entre la diferencia arrojada por la salida de la RNA y el valor esperado (Riedmiller, 1994). Se diferencia del algoritmo BackPropagation principalmente por que las derivadas parciales de la función error se usan única y exclusivamente para determinar el sentido en que se deben corregir los pesos de la red, pero no la magnitud del ajuste, otra característica que las diferencia es que BackPropagation modifica el valor del paso o ajuste proporcionalmente al gradiente resultante de la función de error, no obstante si se llega a una condición donde el gradiente tiende a ser plano o muy cercano al 0 el algoritmo avanza proporcional a este valor por tanto su paso será mínimo y su eficiencia se verá afectada.

RProp por otra parte genera un único parámetro que establece la velocidad de avance, este parámetro recorre la función objetivo para todos y cada uno de los pesos de la red neuronal, logrando así un avance constante solo siendo restringido por el umbral de parada al encontrar los parámetros óptimos. Este algoritmo usa la derivada solamente para determinar la dirección en la actualización de los pesos. Gracias a esto y su bajo consumo computacional, converge de manera más eficiente que los algoritmos que se basan en BackPropagation.

###### 

**(9)**

= Función de cambio de signo de la derivada del error en iteraciones (k) y (k+1).

= Tamaño del paso. (Gradiente).

= Tamaño máximo que puede llegar a tomar el paso.

= Tamaño mínimo que puede llegar a tomar el paso.

Si, el signo de la derivada no cambia en las últimas dos iteraciones, se aumenta el tamaño del paso en , , si hay un cambio de signo en la derivada de la función se supera el punto de ajuste mínimo por tanto se debe reducir el tamaño del paso en una cantidad , sin sobrepasar el factor , siendo en caso de ser cero la derivada, no se realizan modificaciones.

### Aprendizaje Supervisado

Ocurre cuando se toma un conjunto de datos muéstrales de ejemplo, y cada ejemplo está formado por una entrada y una salida deseada. Se introducen entradas hasta que la red neuronal arroje un porcentaje de fiabilidad pertinente en la salida deseada.

Entonces tenemos:

1. *alimentar la red con datos conocidos.*
2. *¿La red arroja respuestas correctas?* 
   * *Sí (dentro del porcentaje de fiabilidad): Ir al paso 3.*
   * *No:* 
     1. *Ajustar los pesos de las conexiones de la red.*
     2. *Ir al paso 1.*
3. *Fin.*

# CAPITULO II. LEVANTAMIENTO DE INFORMACIÓN.

# DISEÑO METODOLÓGICO GENERAL.

El desarrollo metodológico de este proyecto se conformó de dos métodos de investigación y un marco de desarrollo, en el cual se establecieron los lineamientos, principios y fases del propias del proyecto, de esta manera se estructuraron los aspectos básicos que debía cumplir la aplicación como objetivo fundamental, además de esto, se contemplaron los aspectos que de manera indirecta intervenían en la realización del mismo, adaptando y encapsulando en fases funcionales estos requerimientos para su posterior manejo, haciendo uso de la maleabilidad e integración continua del marco de desarrollo establecido.

Continuando con lo anterior, es necesario determinar y cuantificar los métodos de carácter investigativo presentes en el proyecto, ya que estos fundamentan la base teórica del desarrollo, logrando de esta forma aumentar el espectro de investigación, tanto del entorno como de la variable de estudio, para contemplar mediante las teorías establecidas y la conjunción de métodos modernos la mejor solución al problema de investigación planteado.

Es así como el entorno y objeto de estudio permiten hacer uso del método **Descriptivo**, con esto se le permite al investigador observar a cabalidad el entorno, ahondando de ser necesario en los estándares, premisas, métodos, información relativa o necesaria para la investigación, esto con el fin de tener una interpretación objetiva del mismo y la afectación de este a los demás entornos que pudieran existir, a su vez, también fue necesario utilizar el método de investigación **Experimental**, esto debido a que la naturaleza del proyecto daba lugar a la experimentación de metodologías y métodos necesarios para llegar a una solución consistente del problema, realizando de ser necesario cambios en el desarrollo de la hipótesis y las variables que en ella intervenían siendo necesarias para mejorar el grado de precisión del resultado final, cabe aclarar que el marco metodológico en su estructura básica emplea la fase de pruebas y adaptación para retroalimentar el proyecto en base a los resultados obtenidos en una mejora constante.

Se empleó la metodología ágil de desarrollo de software *“****XP****”,*donde según (Canós, Letelier, & Penadés, 2012) esta metodología se fundamenta en la agilidad, maleabilidad y adaptación a los cambios en el momento de llevar a cabo el desarrollo y posterior entrega final del producto, es de resaltar que se toman en cuenta las relaciones interpersonales como un potenciador positivo y necesario, integrando tanto al cliente como a los posibles usuarios al equipo de desarrollo, esto favorece el desarrollo en la medida en que el cliente como consumidor final supervisa el producto en sus diferentes etapas, aportando ideas y sugiriendo los cambios que considere necesarios, la metodología establece un tiempo para la realización de dichos cambios, estipulado en las historias de usuario inherentes a la metodología misma, y definidas en el momento en que se identificó el cambio a realizar, por ello es que esta metodología es ideal para los proyectos donde los requisitos son cambiantes.

## *ACTIVIDADES Y HERRAMIENTAS DE LAS FASES.*

A continuación, se detallarán cada una de las fases, actividades y herramientas tecnológicas que se utilizaron en la construcción de este proyecto:

## *FASE 1. ANÁLISIS DE LA INFORMACIÓN.*

Para poder dar cumplimiento al objetivo específico*; “Determinar a partir de las referencias existentes, los métodos necesarios para resolver Problemas de Optimización No Lineales Restrictos”,* se realizó bajo el método de investigación descriptiva, específicamente con la etapa recolección de datos; las técnicas a utilizar para realizar esta actividad son: arqueología de documentos y entrevistas.

## Actividades de la técnica entrevista:

### Construcción formato de entrevista.

Esta actividad se desarrolló mediante la utilización en primera medida de texto manuscrito, posteriormente la utilización del software Word 2016 ya que este permite realizar documentos, formatos, transcripciones, escritos personalizados entre otros.

### Aplicación de la entrevista.

La realización de esta entrevista se hizo presencialmente en los espacios físicos de la Universidad de la Amazonia al docente del área “Modelos Determinísticos” en calidad de asesoría sobre la temática del curso, de donde surgió el interrogante de ¿Cómo resolver un problema de optimización no lineal con restricciones? Además de la aclaración de la ausencia de aplicaciones o teorías absolutas para este fin, como soporte de ello se generó una transcripción manuscrita de la reunión y del problema abordado

## Transcribir la información al formato especificación de requerimientos.

Llevar la información al formato de Historia de Usuario para una clara definición de los requerimientos, de esto depende el éxito o fracaso del proyecto, la metodología aconseja tareas cortas, realizables y cuantificables, y ya que la información es nuestro insumo principal, la mejor practica es levantar los requerimientos de manera clara concisa y concreta.

1. Formato de Historias de Usuario

Formato de Historias de Usuario

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-000 | **Usuario**: |
| **Nombre Historia:** | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** |
| **Puntos estimados:** | **Iteración asignada:** |
| **Tiempo de Entrega:** |  |
| **Fecha de Inicio:** | **Fecha de Terminación:** |
| **Programador Responsable:** | |
| **Descripción:** | |
| **Observaciones:** | |

Fuente: Elaboración Propia, 2020

## Actividades de la técnica arqueología de documentos:

### Búsqueda de literaturas que aborden los problemas de optimización no lineal.

Los problemas de optimización han existido desde tiempos atrás, por ende, es usual encontrar literaturas que abarquen el tema de manera parcial o global, no obstante, estas literaturas carecen de algoritmos de solución para problemas de optimización no lineal, por tanto, en la búsqueda sectorizada que se realizó se tomó como criterio de decisión la inclusión de la optimización no lineal como ítem fundamental. Las literaturas analizadas exponen problemas cotidianos enmarcados en la optimización no lineal, con diferentes formas de solución.

1. Literaturas seleccionadas de optimización no lineal

Literaturas seleccionadas de optimización no lineal

|  |
| --- |
| **DOCUMENTACIÓN ENCONTRADA** |
| The Hybrid Genetic Algorithm with two Local Optimization Strategies for traveling salesman problem. Wang, Y. (2014). ELSEIVER, LXX, 16. |
| Evacuation dynamic and exit optimization of a supermarket based on particle swarm optimization. Lin Li, (2014), Physica A, 23 |
| Ant Colony Optimization based Clustering Methodology. Ínkaya, T., Kayalıgil, S., & Özdemirel, N (2014). ELSEIVER, V, 51. |
| Using Artificial Neural Networks to predict monthly precipitation for the Cali river basin, Colombia. Daniel David Montenegro Murillo. (2018). DYNA UNAL COLOMBIA, 122-130 |
| Redes Neuronales para Clasificación: Una aplicación al caso de Riesgos Laborales en Colombia. Hernández, C. M. (2017), PONTIFICIA UNIVERSIDAD JAVERIANA. |
| Problemas de Asignación Generalizada: modelización, aplicaciones lógicas y métodos de solución. Gutierrez, C. S. (2019). Universidad de Valladolid. |
| Planificación de recursos hídricos en zonas de secano usando un modelo de optimización no lineal. Villavicencio, A., Arumí, J., & Holzapfel, E. (2011). Obras y Proyectos, 10, 73-80 |
| Modelo Matematico de Programacion No-Lineal Para la Optimizacion de los Costos de Fabricacion de Vigas de Concreto Armado. Mibelli, G. (2005). *Escuela de Ingenieria Civil, I*, |
| Modelo de Optimización Multi-Objetivo para la Propagación de la Producción Agrícola a Pequeña Escala en Santander, Colombia. Sarmiento, L. H. (2018). Escuela de Estudios Industriales Y Empresariales, 1117 |
| Resource constrained project scheduling Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for relay Node deployment in Wireless Sensor Networks. Gutierrez, J., & Pulido, A. (2015). The *ELSEIVER, V*, 13. |

Fuente: Elaboración propia.

### Análisis de documentos.

Al realizar el análisis de la bibliografía mencionadas anteriormente y de la bibliografía adyacente a esta, se determinan los métodos experimentales necesarios para dar solución a la pregunta problema, lo que se genera es un algoritmo computacional iterativo capaz de resolver teóricamente problemas de optimización no lineal con y sin restricciones, conformándose de la heurística de las redes neuronales artificiales como entorno, el gradiente descendente como algoritmo de búsqueda, y una función de penalización inversamente proporcional a la violación de las restricciones.

1. Algoritmo Generado para resolver problemas de optimización no lineal

Aquí va el algoritmo urgente

*Fuente. Elaboración Propia.*

Para realizar el seguimiento del desarrollo de software se utilizó las herramientas ofimáticas como Word y Excel.

1. Logo Microsoft Word 2016.

*Logo Microsoft Word 2016*



*Fuente:* [*https://products.office.com/es-co/word*](https://products.office.com/es-co/word)

1. Logo Microsoft Excel 2016.

*Logo Microsoft Excel 2016*



*Fuente:* [*https://products.office.com/es-co/excel/*](https://products.office.com/es-co/excel/)

Para realizar el despliegue de la aplicación se utilizó la plataforma de despliegue Heroku, en conjunción con la aplicación de repositorios GitHub, almacenando en esta última plataforma el código fuente para su compilación y puesta en marcha en la dirección web <https://germathes6.herokuapp.com/>

1. Logo Plataforma Heroku.

*Logo Plataforma Heroku.*

**

Fuente: <https://brand.heroku.com/>

1. Logo Plataforma GitHub.

*Logo Plataforma GitHub.*



Fuente: <https://github.com/logos/>

## Establecimiento de Requerimientos.

Una vez se estableció el *“target”* del desarrollo, se procedió a estructurar de manera secuencial los componentes de la aplicación y la sinergia necesaria para que estos componentes funcionaran correctamente, para ello se generó una historia de usuario matriz que desencadeno a su vez historias de usuario modulares, estas en su realización y acoplamiento total, generaron la solución a la pregunta problema, dando por realizado el objetivo.

1. Definición de requerimientos funcionales del proyecto

*Definición de requerimientos funcionales del proyecto*

|  |  |  |
| --- | --- | --- |
|  | **REQUERIMIENTO FUNCIONAL** | FUENTE |
| RF-001 Requerimiento Matriz | Desarrollar una Aplicación Web que mediante el uso de redes neuronales resuelva problemas de optimización no lineal con restricciones. | USUARIO |
| RF-002 | El sistema debe permitir la creación de usuarios tanto para el registro de la persona como para su posterior acceso, adicional a esto debe permitir sesiones múltiples y acotadas para mejorar la seguridad. | USUARIO |
| RF-003 | El sistema debe permitir la Gestión de Usuarios, de tal forma que sea posible realizar las operaciones CRUD[[5]](#footnote-5), su visualización, individualización y conteo | USUARIO |
| RF-004 | El sistema debe permitir la creación y modificación de roles, siendo administrados y usuario los roles existentes, no obstante se toleraran las operaciones CRUD para un posterior proyecto PREMIUM | USUARIO |
| RF-005 | El sistema dedicara un requisito funcional a la gestión visual de la aplicación siendo así, la aplicación debe graficar las funciones ingresadas al lugar que genera una tabla con las iteraciones respectivas, el valor obtenido y el procedimiento realizado, debe garantizar la comodidad visual así como su entendimiento y correcto funcionamiento | USUARIO |
| RF-006 | El sistema debe permitir la modificación de parámetros tales como el número de iteraciones o el grado de error permitido, almacenara de manera incógnita las funciones ingresadas para su seguimiento, debe ser intuitiva eficiente y plotear el resultado en un tiempo no superior a 30 segundos | USUARIO |
| RF-007 | El sistema debe ser capaz de manejar ecuaciones del tipo no lineal, exponencial y multivariado siendo (x, y, z) las variables que pudieren operar en el sistema, generar números simbólicos y operaciones complejas como derivadas, y operaciones algebraicas. | USUARIO |
| RF-008 | El sistema debe garantizar el aprendizaje iterativo, la mejora constante del resultado y su trazabilidad | SISTEMA |

Fuente: elaboración propia

1. Relación de requerimientos con la fuente.

*Relación de requerimientos con la fuente.*

|  |  |  |  |
| --- | --- | --- | --- |
| **REFERENCIA** | **REQUERIMIENTO FUNCIONAL** | **FUENTE** | ACTOR |
| RF-002-001 | Registrar Persona | RF-002 | Usuario |
| RF-002-002 | Crear usuario | RF-002 | Usuario |
| RF-002-003 | Iniciar Sesión | RF-002 | Usuario |
| RF-002-004 | Cerrar Sesión | RF-002 | Usuario |
| RF-002-005 | Desvincular Persona | RF-002 | Administrador |
| RF-002-006 | Activar Cuenta | RF-002 | Administrador |
| RF-003-001 | Modificar Usuario | RF-003 | Administrador |
| RF-003-002 | Listar Usuarios | RF-003 | Administrador |
| RF-003-003 | Seleccionar Usuario | RF-003 | Administrador |
| RF-004-001 | Crear Rol | RF-004 | Administrador |
| RF-004-002 | Listar Roles | RF-004 | Administrador |
| RF-004-003 | Establecer Rol | RF-004 | Administrador |
| RF-004-004 | Eliminar Rol | RF-004 | Administrador |
| RF-005-001 | Graficar Función ingresada | RF-005 | Sistema |
| RF-005-002 | Graficar Tabla de resultados | RF-005 | Sistema |
| RF-005-003 | Graficar Resultado individual | RF-005 | Sistema |
| RF-005-004 | Establecer tema visual, cómodo visualmente | RF-005 | Desarrollador |
| RF-005-005 | Desarrollar un entorno visual de la aplicación para que esta sea usada de manera intuitiva | RF-005 | Usuario |
| RF-006-001 | Registrar Función en segundo plano | RF-006 | Sistema |
| RF-006-002 | Modificar iteraciones | RF-006 | Usuario |
| RF-006-003 | Modificar grado de error | RF-006 | Usuario |
| RF-006-004 | Limpiar datos | RF-006 | Usuario |
| RF-006-005 | Establecer tiempo máximo de respuesta | RF-006 | Usuario |
| RF-007-001 | Ingresar Función | RF-007 | Usuario |
| RF-007-002 | Derivar Función | RF-007 | Sistema |
| RF-007-003 | Solucionar Ecuaciones | RF-007 | Sistema |
| RF-007-004 | Soportar números simbólicos | RF-007 | Sistema |
| RF-008-001 | Graficar datos iterativamente | RN-008 | Sistema |
| RF-008-002 | Establecer el criterio neural de aprendizaje. | RN-008 | Desarrollador |
| RF-008-003 | Aplicar Restricciones a la Función (CRUD) | RN-008 | Usuario |
| RF-008-004 | Demostrar Resultados | RN-008 | Desarrollador |

Fuente: Elaboración Propia.

## *FASE 2 DISEÑO DEL SISTEMA INFORMACIÓN*

En este punto se optó por generar una estructura web con el framework Express.JS, en el Marco de Node.JS como entorno de desarrollo software, de esta forma se creó una estructura básica para la primer iteración, de esta forma y gracias a la metodología XP podemos avanzar en otros campos como el diseño visual de la aplicación o el gestor de aprendizaje neural, para la siguiente iteración se pueden agregar esos componentes a la estructura, haciéndola cada vez más robusta y estable para el proyecto.

## Elaboración Historias de usuarios

Dado el entorno ágil en el que se ejecuta el desarrollo de software se hace uso de las historias de usuario, que en síntesis son los requerimientos expuestos por el cliente.

1. Historia de usuario (HU-001), Historia de Usuario Matiz.

Historia de usuario (HU-001), Historia de Usuario Matiz.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-001 | **Usuario**: Objetivo General |
| **Nombre Historia:** Desarrollo de un aplicación web que mediante el uso de Redes Neuronales Artificiales resuelva Problemas de Optimización no Lineal Restricta | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 10/10 | **Iteración asignada:** Constante |
| **Tiempo de Entrega:** 4 Meses | |
| **Fecha de Inicio:** 9 de Diciembre de 2020 | **Fecha de Entrega:** 20 de mayo de 2021 |
| **Programador Responsable:** W. German Aguirre B. | |
| **Descripción:** La funcionalidad global de esta aplicación web se enmarca en la consecución de los objetivos del proyecto, por tanto este requerimiento es de importancia vital pues encapsula dentro de él las demás funcionalidades e historias de usuario correspondientes que se generaron a lo largo de su realización | |
| **Observaciones:** Se le asigna el máximo puntaje y una iteración continua ya que al ser el requerimiento global este debe verse reflejado al culminar el desarrollo. | |

Fuente: Elaboración propia.

1. Historia de usuario (HU-002-001), Registrar Persona.

Historia de usuario (HU-002-001), Registrar Persona

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-002-001 | **Usuario**: Cliente |
| **Nombre Historia:** Registrar Persona | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 6 | **Iteración asignada:** 2 |
| **Tiempo de Entrega:** 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema debe permitir el registro de personas, de tal forma serán necesarios los siguientes datos: Número de Identificación, Nombres, Apellidos, Fecha de Nacimiento y Teléfono. | |
| **Observaciones:** En este punto elsistema genera el usuario mediante un disparados automático en la base de datos, por tanto el usuario y la contraseña con el número de identificación, y el apellido respectivamente | |

Fuente: Elaboración Propia

1. Historia de usuario (HU-002-002), Crear Usuario.

Historia de usuario (HU-002-002), Crear Usuario.

|  |  |
| --- | --- |
| Historia de Usuario | |
| **Referencia**: HU-002-002 | Usuario: Cliente |
| **Nombre Historia**: Crear Usuario | |
| **Prioridad en el Proyecto**: Alta | **Riesgo en desarrollo**: Vital |
| **Puntos estimados**: 6 | **Iteración asignada**: 2 |
| **Tiempo de Entrega**: 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema debe generar un usuario para el posterior ingreso al sistema, capaz de soportar sesiones, cookies y tiempo de inactividad. | |
| **Observaciones:** El usuario se genera con los datos ingresados por el cliente en el momento del registro. | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-002-002), Cerrar Sesion.

Historia de usuario (HU-002-002), Cerrar Sesión

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Referencia:** HU-002-004 | **Usuario**: Cliente |
| **Nombre Historia:** Cerrar Sesión | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 6 | **Iteración asignada:** 2 |
| **Tiempo de Entrega:** 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema debe permitir el cierre de la sesión del usuario, o en su defecto cerrar la sesión debido a un tiempo de inactividad o a una hora especificada por administrador. Se aclara que esta es una acción estructural y de seguridad de la aplicación. | |
| **Observaciones:** | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-002-005), Desvincular Persona.

Historia de usuario (HU-002-005), Desvincular Persona.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-002-005 | **Usuario**: Cliente |
| **Nombre Historia:** Desvincular Persona | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 6 | **Iteración asignada:** 2 |
| **Tiempo de Entrega:** 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema debe soportar la desvinculación de una persona, haciendo inactivo su usuario y los datos existentes de esta, no obstante los datos obtenidos de las funciones realizadas serán de uso exclusivo de Germath.JS de carácter anónimo y experimental por tanto, estos datos no serán eliminados, en caso de requerir usar nuevamente la aplicación la persona deberá solicitar al administrador que active nuevamente su cuenta | |
| **Observaciones:** El sistema generara una nueva tarea para el administrador. (Activar Cuenta) | |

Fuente: Elaboración propia.

1. Historia de usuario (HU-002-005), Activar Cuenta.

Historia de usuario (HU-002-006), Activar Cuenta.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-002-006 | **Usuario**: Cliente |
| **Nombre Historia:** Activar Cuenta | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 6 | **Iteración asignada:** 2 |
| **Tiempo de Entrega:** 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema generara una tarea para el usuario *ADMNISTRADOR* de tal forma que le sean indicados los datos de la persona que solicitud de activación para su posterior verificación y activación. | |
| **Observaciones:** | |

Fuente: Elaboración propia.

1. Historia de usuario (HU-003-001), Modificar Usuario.

*Historia de usuario (HU-003-001), Modificar Usuario.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-003-001 | **Usuario**: Cliente |
| **Nombre Historia:** Modificar Usuario | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 6 | **Iteración asignada:** 2 |
| **Tiempo de Entrega:** 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema debe permitir a modificación de los datos del usuario, sin embargo los datos de usuario y contraseña, solo podrán modificarse bajo autorización del administrador, generando la solicitud correspondiente, mediante el formulario *“recuperar contraseña”* | |
| **Observaciones:** Debido a que el número de identificación y el apellido son los datos de acceso al modificar dichos datos se alteraría el algoritmo de reconocimiento causando error por tanto se debe verificar esta acción por la administración con las políticas establecidas. | |

Fuente: Elaboración propia.

1. Historia de usuario (HU-003-002), Listar Usuarios.

*Historia de usuario (HU-003-002), Listar Usuarios.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-003-002 | **Usuario**: Cliente |
| **Nombre Historia:** Listar Usuarios | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 6 | **Iteración asignada:** 3 |
| **Tiempo de Entrega:** 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema debe listar de los datos de los usuarios existentes, con el fin de ejercer control y proyección en base a estos datos. | |
| **Observaciones:** | |

Fuente: Elaboración propia.

1. Historia de usuario (HU-003-001), Seleccionar Usuario.

*Historia de usuario (HU-003-003), Seleccionar Usuario.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-003-003 | **Usuario**: Administrador |
| **Nombre Historia:** Modificar Usuario | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Vital |
| **Puntos estimados:** 6 | **Iteración asignada:** 3 |
| **Tiempo de Entrega:** 2 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema generara un formulario con los datos del usuario seleccionado, incluyendo datos relevantes, personales, y acciones en la aplicación. | |
| **Observaciones:** Esta acción es de carácter especial y no constante. | |

Fuente: Elaboración propia.

1. Historia de usuario (HU-004-001), Crear Rol

Historia de usuario (HU-004-001), Crear Rol.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-003-003 | **Usuario**: Administrador |
| **Nombre Historia:** Crear Rol | |
| **Prioridad en el Proyecto:** Baja | **Riesgo en desarrollo:** Leve |
| **Puntos estimados:** 2 | **Iteración asignada:** 5 |
| **Tiempo de Entrega:** 1 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** el sistema debe soportar una opción para que el Administrador, agregue roles según sea necesario. Los permisos de vistas y acciones están sujetas al rol. | |
| **Observaciones:** | |

Fuente: Elaboración propia.

1. Historia de usuario (HU-004-002), Listar Roles

*Historia de usuario (HU-004-002), Listar Roles.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-004-002 | **Usuario**: Administrador |
| **Nombre Historia:** Listar Roles | |
| **Prioridad en el Proyecto:** Baja | **Riesgo en desarrollo:** Leve |
| **Puntos estimados:** 2 | **Iteración asignada:** 5 |
| **Tiempo de Entrega:** 1 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** el sistema debe permitir que sean listados los roles existentes con sus respectivos permisos y acciones | |
| **Observaciones:** no necesariamente se deben listar los usuarios con roles específicos. Debido a la naturaleza de la aplicación. | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-004-003), Establecer Rol.

Historia de usuario (HU-004-003), Establecer Rol.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-004-003 | **Usuario**: Administrador |
| **Nombre Historia:** Establecer Rol | |
| **Prioridad en el Proyecto:** Media | **Riesgo en desarrollo:** Leve |
| **Puntos estimados:** 3 | **Iteración asignada:** 5 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La aplicación debe soportar el uso de roles específicos como también el establecimiento o modificación del rol actual del usuario. | |
| **Observaciones:** Esta opción solo está habilitada para el administrador global del sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-004-004), Eliminar Rol

Historia de usuario (HU-004-003), Eliminar Rol.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-004-004 | **Usuario**: Administrador |
| **Nombre Historia:** Eliminar Rol | |
| **Prioridad en el Proyecto:** Media | **Riesgo en desarrollo:** Leve |
| **Puntos estimados:** 3 | **Iteración asignada:** 5 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La aplicación debe soportar la eliminación de roles específicos como también el establecimiento o modificación del rol actual del usuario. | |
| **Observaciones:** Esta opción solo está habilitada para el administrador global del sistema | |

*Fuente: Elaboración Propia.*

1. Historia de usuario (HU-005-001), Graficar Función Ingresada.

Historia de usuario (HU-005-001), Graficar Función Ingresada.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-005-001 | **Usuario**: Usuario |
| **Nombre Historia:** Eliminar Rol | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 7 | **Iteración asignada:** 4 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La Aplicación Germath.JS debe graficar en un plano 2D las funciones ingresadas para su posterior solución para de esta forma poder visualizar de manera gráfica el comportamiento de la función manejada y el óptimo encontrado. | |
| **Observaciones:** Componente netamente Visual | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-005-002), Graficar Tabla de Resultados.

Historia de usuario (HU-005-002), Graficar Tabla de Resultados.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-005-002 | **Usuario**: Usuario |
| **Nombre Historia:** Graficar Tabla de Resultados. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 7 | **Iteración asignada:** 4 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La Aplicación Germath.JS debe generar una tabla con las iteraciones, entradas y salidas iterativas de la red y el proceso de aprendizaje y minimización del error, para de esta forma poder visualizar de manera gráfica el comportamiento de la función manejada y el óptimo encontrado. | |
| **Observaciones:** Componente netamente Visual | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-005-003), Graficar Resultado Individual.

Historia de usuario (HU-005-003), Graficar Resultado Individual.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-005-003 | **Usuario**: Usuario |
| **Nombre Historia:** Graficar Resultado Individual. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 7 | **Iteración asignada:** 4 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La Aplicación Germath.JS debe graficar de manera concisa y separada, los valores de solución óptima hallada para mejorar el sentido intuitivo de la aplicación. | |
| **Observaciones:** Componente netamente Visual | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-005-004), Establecer Tema Visual

Historia de usuario (HU-005-004), Establecer Tema Visual

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-005-004 | **Usuario**: Usuario |
| **Nombre Historia:** Establecer Tema Visual. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 7 | **Iteración asignada:** 4 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** En este punto se requiere generar los logos, tema visual de colores y adaptabilidad de fondos para los diferentes formulario, además de seleccionar temas de color neutro, con un enfoque minimalista y sencillo, no obstante eso no merma los detalles visuales y atractivos que presenta la aplicación. | |
| **Observaciones:** Componente netamente Visual | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-005-005), Desarrollar un Entorno Visual Intuitivo.

*Historia de usuario (HU-005-005), Desarrollar un Entorno Visual Intuitivo.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-005-005 | **Usuario**: Usuario |
| **Nombre Historia:** Establecer Tema Visual. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 7 | **Iteración asignada:** 4 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** Es de aclarar que cuando se solicita que la aplicación se desarrolle para funcionar intuitivamente, hacemos alusión a que sea lo más sencilla, concreta y expresiva posible, esto con el fin de, no confundir al usuario con opciones excesivas, o un renderizado cargado de componentes. | |
| **Observaciones:** Componente netamente Visual | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-006-001), Registrar Función en Segundo Plano.

*Historia de usuario (HU-006-001), Registrar Función en Segundo Plano.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-006-001 | **Usuario**: Sistema |
| **Nombre Historia:** Registrar Función en Segundo Plano. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema debe recoger los daros de las unciones ingresadas y los resultados óptimos incluidos de manera anónima, esto con el fin de retroalimentar la aplicación en posteriores despliegues. (Trabajos futuros) | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-006-002), Modificar Iteraciones.

*Historia de usuario (HU-006-002), Modificar Iteraciones.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-006-001 | **Usuario**: Sistema |
| **Nombre Historia:** Modificar Iteraciones | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema permitirá la modificación de las iteraciones máximas que realizara a red neuronal antes de arrojar su mejor valor, se aclara que el número de iteraciones es directamente proporcional a la eficiencia de la red. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-006-003), Modificar Grado de Error.

*Historia de usuario (HU-006-003), Modificar Grado de Error.*

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-006-003 | **Usuario**: Sistema |
| **Nombre Historia:** Modificar grado de Error | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema permitirá la modificación del grado de error mínimo aceptable que tolerara la red neuronal antes de arrojar su mejor valor, se aclara que el grado de error es directamente proporcional a la eficiencia de la red. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-006-004), Limpiar Datos.

Historia de usuario (HU-006-004), Limpiar Datos.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-006-004 | **Usuario**: Sistema |
| **Nombre Historia:** Limpiar Datos | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** El sistema permitirá limpiar la pantalla de datos en un reset de los elementos o modificaciones que hubiere podido sufrir el formulario. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-006-005), Establecer Tiempo Máximo de Respuesta.

Historia de usuario (HU-006-005), Establecer Tiempo Máximo de Respuesta.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-006-005 | **Usuario**: Sistema |
| **Nombre Historia:** Establecer Tiempo Máximo de Respuesta. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La aplicación web Germath.JS no debe superar los 30 segundos en arrojar un resultado, veraz, eficiente y confiable. (con iteraciones y grado de error aceptable) | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-007-001), Ingresar Función

Historia de usuario (HU-006-005), Ingresar F.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-007-001 | **Usuario**: Sistema |
| **Nombre Historia:** Ingresar Función. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La aplicación debe contener un espacio para ingresar el texto, es de aclarar que solo se podrán ingresar la cantidad de letras establecida en la función enmarcadas en X, Y o Z con un numero de caracteres no mayor a 100 para simplificar el trabajo de la red neuronal. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-007-002), Derivar Función

Historia de usuario (HU-007-003), Derivar Función.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-007-002 | **Usuario**: Sistema |
| **Nombre Historia:** Derivar Función. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** La aplicación debe permitir la derivación de la función ingresada, derivando la función respecto a cada variable, siendo su máximo 3 y estando ligado este requerimiento al de referencia HU-007-001 | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-007-003), Solucionar Ecuaciones.

Historia de usuario (HU-007-003), Solucionar Ecuaciones.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-007-003 | **Usuario**: Sistema |
| **Nombre Historia:** Solucionar Ecuaciones | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** debido a la naturaleza del algoritmo y a las ecuaciones colaterales dependientes que se generan en cada iteración, es necesario resolver estas ecuaciones garantizando la fiabilidad del resultado. Por tanto es necesario establecer dentro de nuestra aplicación lun solucionador de ecuaciones con variables en igual número a las incógnitas de la ecuación por iteración. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-007-004), Soportar Números Simbólicos.

Historia de usuario (HU-007-004), Soportar Números Simbólicos.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-007-004 | **Usuario**: Sistema |
| **Nombre Historia:** Soportar Números Simbólicos. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** debido a la naturaleza del algoritmo este usa números simbólicos para establecer el gradiente adecuado, por tanto es necesario que la aplicación soporte este tipo de números con sus respectivas operaciones. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-008-001), Graficar Datos Iterativamente.

Historia de usuario (HU-008-001), Graficar Datos Iterativamente.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-008-001 | **Usuario**: Sistema |
| **Nombre Historia:** Graficar Datos Iterativamente | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** Es necesario que la aplicación evidencieel avance iterativo graficando de manera secuencial estructurada y organizada la información para poder evidenciar la trazabilidad de la misma. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-008-002), Establecer el Criterio de Aprendizaje.

Historia de usuario (HU-008-002), Establecer el Criterio Neural de Aprendizaje.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-008-002 | **Usuario**: Sistema |
| **Nombre Historia:** Establecer el Criterio Neural de Aprendizaje. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** Para abordar el problema en concreto se realizó una minuciosa documentación de los algoritmos necesarios en la aplicación, entre ellos el criterio de aprendizaje, para ello es necesario que la red neuronal retroalimente sus entradas con la salida obtenida, para así generar un aprendizaje sectorizado, enmarcado en el uso de la red neuronal. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-008-003), Aplicar Restricciones a la Función.

Historia de usuario (HU-008-003), Aplicar Restricciones a la Función (CRUD).

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-008-003 | **Usuario**: Sistema |
| **Nombre Historia:** Aplicar Restricciones a la Función (CRUD). | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| **Descripción:** como se ha especificado antes, (CRUD) la aplicación debe permitir el manejo de restricciones en la función objetivo, así como su modificación, eliminación o agregación de una o más restricciones conforme se requiera. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

1. Historia de usuario (HU-008-004), Demostrar Resultados.

Historia de usuario (HU-008-004), Demostrar Resultados.

|  |  |
| --- | --- |
| **Historia de Usuario** | |
| **Numero:** HU-008-004 | **Usuario**: Sistema |
| **Nombre Historia:** Demostrar Resultados. | |
| **Prioridad en el Proyecto:** Alta | **Riesgo en desarrollo:** Alta |
| **Puntos estimados:** 5 | **Iteración asignada:** 6 |
| **Tiempo de Entrega:** 4 días | |
| **Programador Responsable:** Weynder German Aguirre Betancour | |
| * **Descripción:** La aplicación Germath.JS debe generar una solución eficiente y veraz a los problemas de optimización no lineal que en ella sean ingresados, dando respuesta a el Objetivo Específico “*Evaluar el desempeño de la Red Neuronal Artificial en base a los problemas de optimización más comunes conocidos en las literaturas de referencia.”,* por tanto, se debe generar una tabla con los ejercicios escogidos de las literarias de referencia con sus respectivos datos de solución. | |
| **Observaciones:** Componente netamente del Sistema | |

Fuente: Elaboración Propia.

# CAPITULO III. USO DE HERRAMIENTAS TECNOLÓGICAS PARA EL DESARROLLO DE LA APLICACIÓN.

## *FASE 3 CODIFICAR EL SISTEMA DE INFORMACIÓN*

El objetivo específico ***Implementar una Aplicación Web de fácil acceso y uso que resuelva problemas de optimización no lineal Restricta haciendo uso de RNA*** se hará a partir del método de *investigación proyectivo*, específicamente en la etapa elaboración ya que este permite la construcción de los planos y prototipos propuestos en el desarrollo de la investigación a continuación resaltamos las actividades para el desarrollo del mismo.

### Diseño de base de datos.

Considerando el carácter libre de la aplicación y su despliegue, se optó por una base de datos gratuita, MariaDB, siendo consistente con el modelo relacional de datos, y permitiendo en ella la ejecución de disparadores y procedimientos almacenados que agilizan la búsqueda y escritura de los datos en segundo plano, y la herramienta HeidiSQL para el manejo visual de la base de datos y la información almacenada en ella, para la generación del modelo relacional se utilizó WorkBench, por su facilidad en la creación y manipulación del modelo.

El Modelo de la base de datos, y los adyacentes a esta, se encontrarán en los archivos entregables, siendo modificada la misma en las iteraciones de desarrollo, por tanto, se anexaron tantas imágenes como cambios realizados por iteración, esto para mejor comprensión del avance del proyecto, los modelos de las bases de datos representan la estructura del almacenamiento de la información, para esta aplicación web servirán de soporte a la información de los usuarios y las funciones en ella ingresada, todo esto para estricto control y monitoreo del uso de la aplicación.

### Elaborar Back-End y lógica de negocio.

A partir de la definición de programación orientada a objetos, se realizó el desarrollo de la aplicación “*Germath.JS”,* integrando en esta los framework’s y estructuras de código actualizadas al entorno de desarrollo software, como lo es ECMAScript6 y Node.JSen su versión 15.14 con el editor de texto, Visual Studio Code, de esta forma se inició la codificación con el leguaje de desarrollo JavaScript de la arquitectura, Las acciones, las funcionalidades, alertas y validaciones de la aplicación.

Node.JS, Entorno de desarrollo software con el que se realizó la construcción de la aplicación.

1. Logo Node.JS.

*Logo Node.JS.*

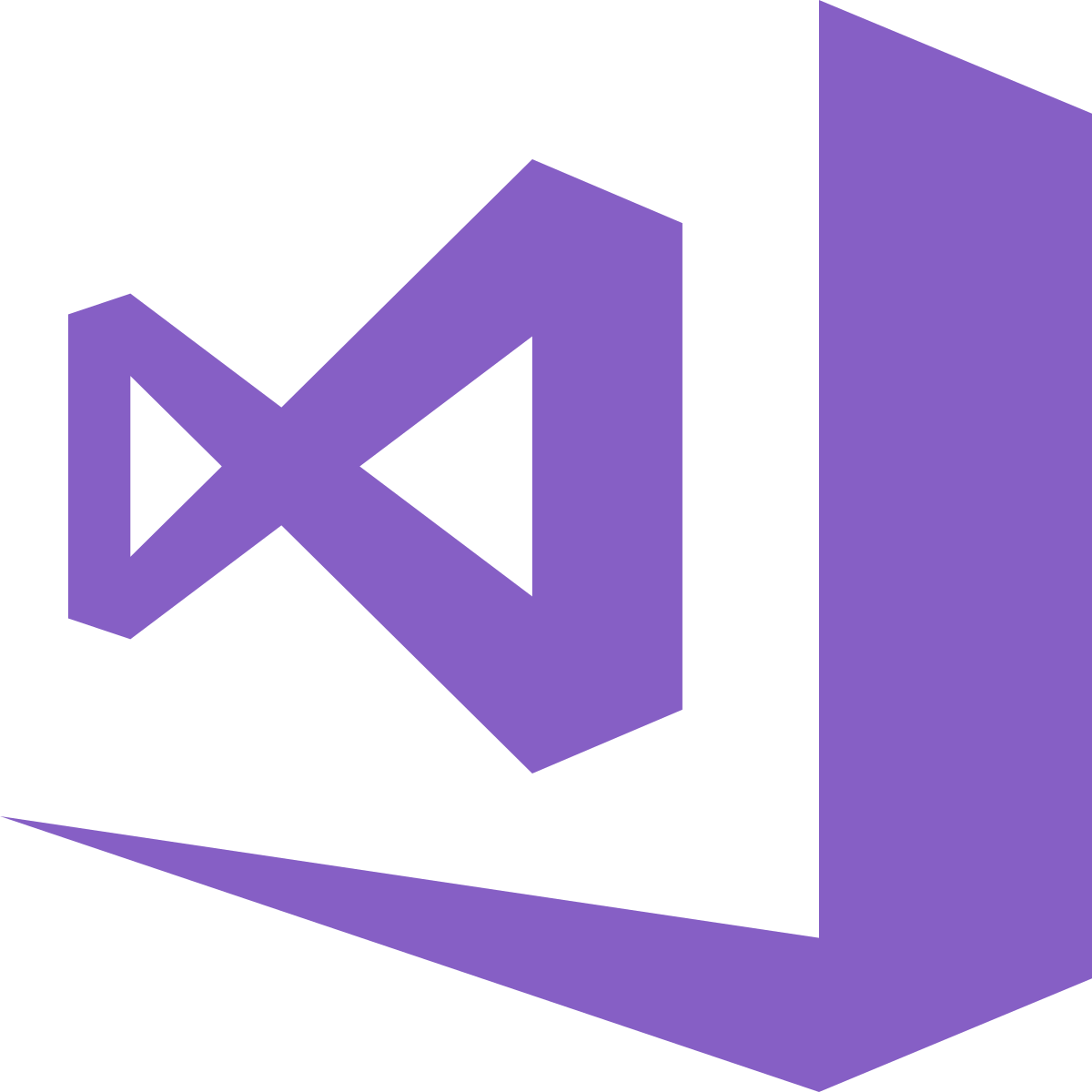


*Fuente:* <https://nodejs.org/es/about/resources/>

Visual Studio, herramienta para el entorno de desarrollo de software.

1. Logo Visual Studio.

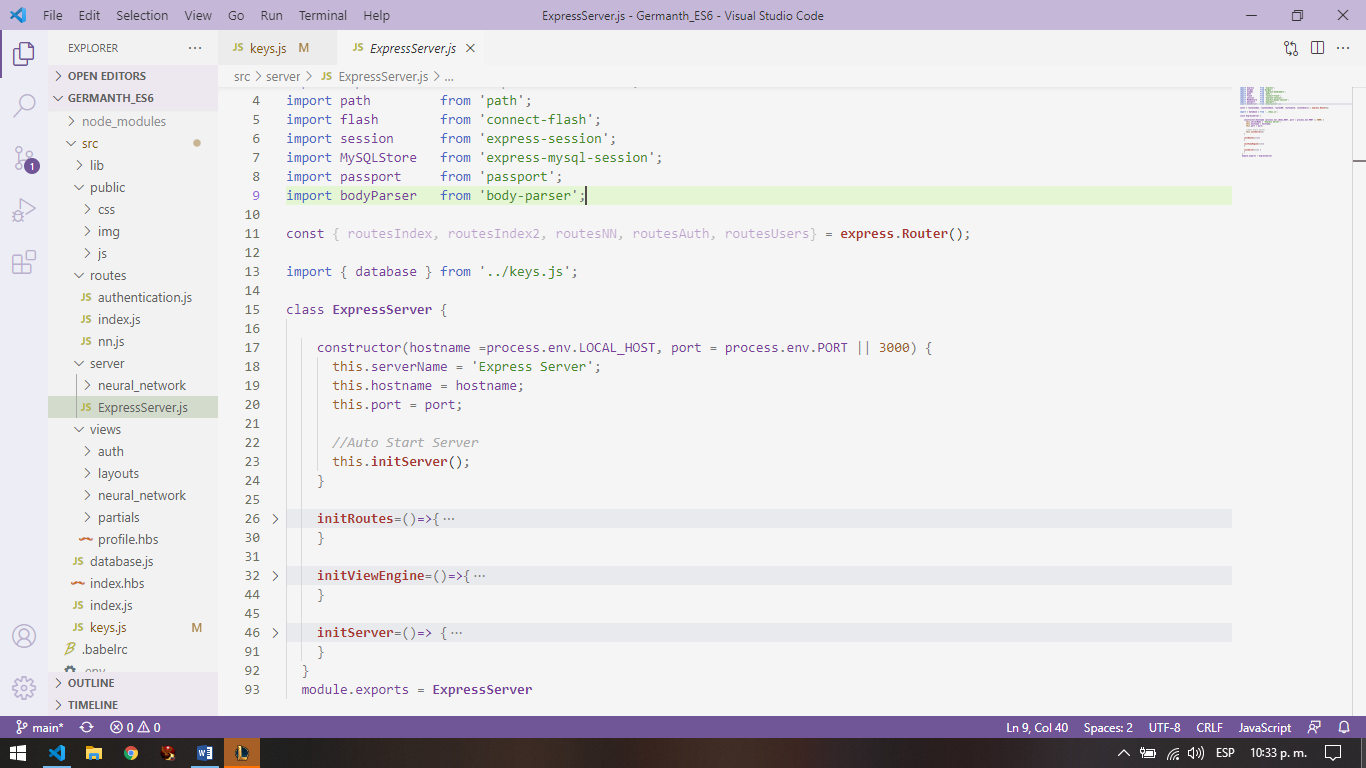
*Logo Visual Studio Code.*



*Fuente:* [*https://visualstudio.microsoft.com/es/*](https://visualstudio.microsoft.com/es/)

1. Esquema de carpetas código.

*Esquema de carpetas código.*



*Fuente:* [*https://visualstudio.microsoft.com/es/*](https://visualstudio.microsoft.com/es/)

### Elaborar Front-End.

Es de resaltar que desde la iteración 2, se dedicaron, historias de usuario, horas de trabajo y puntos de importancia al desarrollo de la plataforma y con ella el concepto visual de la misma, se desplego el front-end del proyecto con el framework de HandleBars.JS y con los leguajes de estructuración del lado del cliente como HTML 5, CSS3 y Bootstrap Se realiza la estructuración visual, validaciones a lado del cliente, las animaciones, las alertas, también se generan conexiones a la persistencia, o a los controladores de la aplicación (Back-End), para realizar las consultas, agrupaciones de datos, el ingreso de los datos o el despliegue de los mismos.

1. Logo de HTML 5, CSS3, JavaScript y Bootstrap.

*Logo de HTML 5, CSS3, JavaScript y Bootstrap.*



*Fuente:* [*https://www.udemy.com/course/sitios-web-multimedia-con-html5-video-background-css3-y-js/*](https://www.udemy.com/course/sitios-web-multimedia-con-html5-video-background-css3-y-js/)

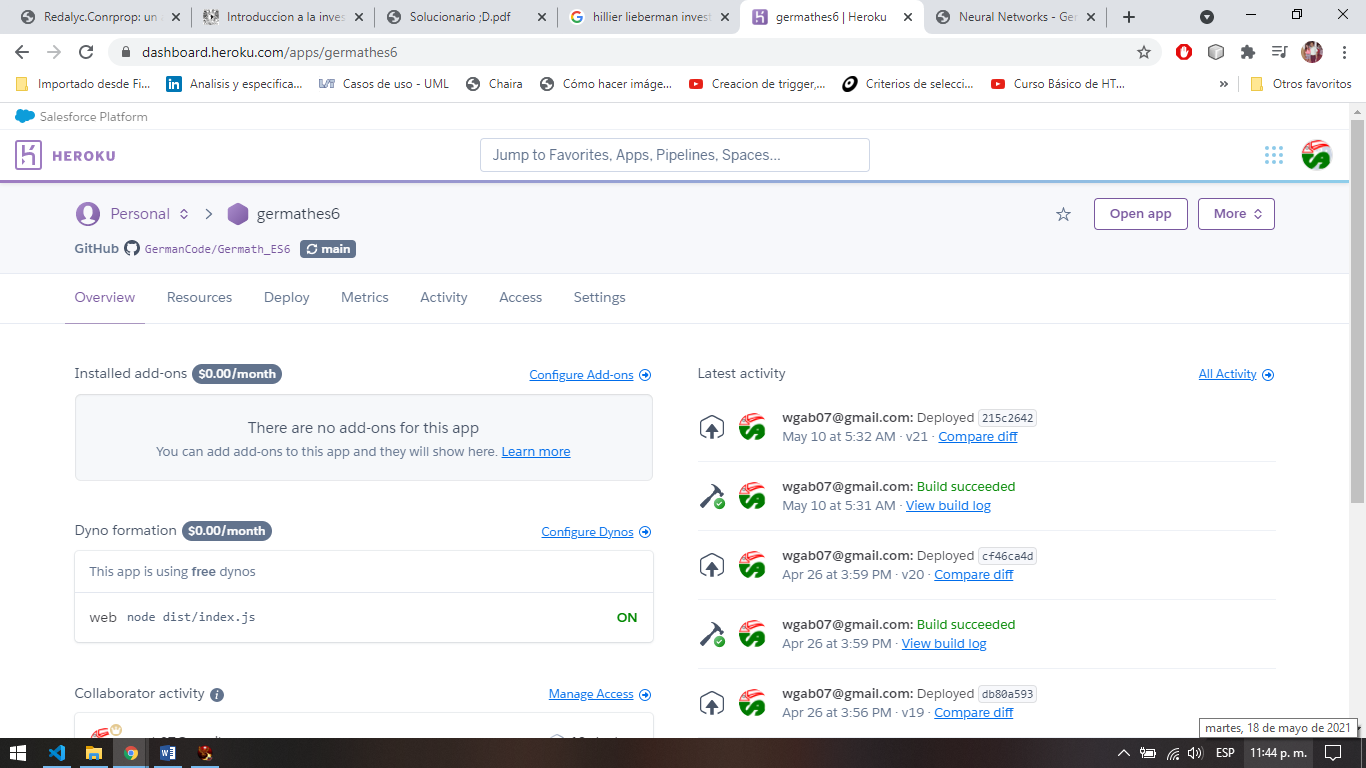
# CAPITULO IV. IMPLEMENTACIÓN Y RESULTADOS.

## *FASE 4 IMPLEMENTACIÓN DE PRUEBAS.*

Es de aclarar que se realizaron diferentes despliegues en el trascurso del desarrollo de la aplicación, estos se encuentran documentados en el repositorio del proyecto, de esta forma como producto final se obtuvo una aplicación web capaz de adaptarse y solucionar problemas de optimización no lineal con y sin restricciones de hasta dos variables, véase figura 18, se obtuvieron resultados positivos respecto a las literaturas de referencia siendo los ejercicios 12.5-4 y 12.7-2 del libro de Hillier Lieberman 7ma Edición los escogidos para la prueba y como puede evidenciarse en la tabla resultados de prueba, véase tabla 40, se mejoró el valor documentado en la literatura.

1. Aplicación Web GerMath.JS en Heroku Running

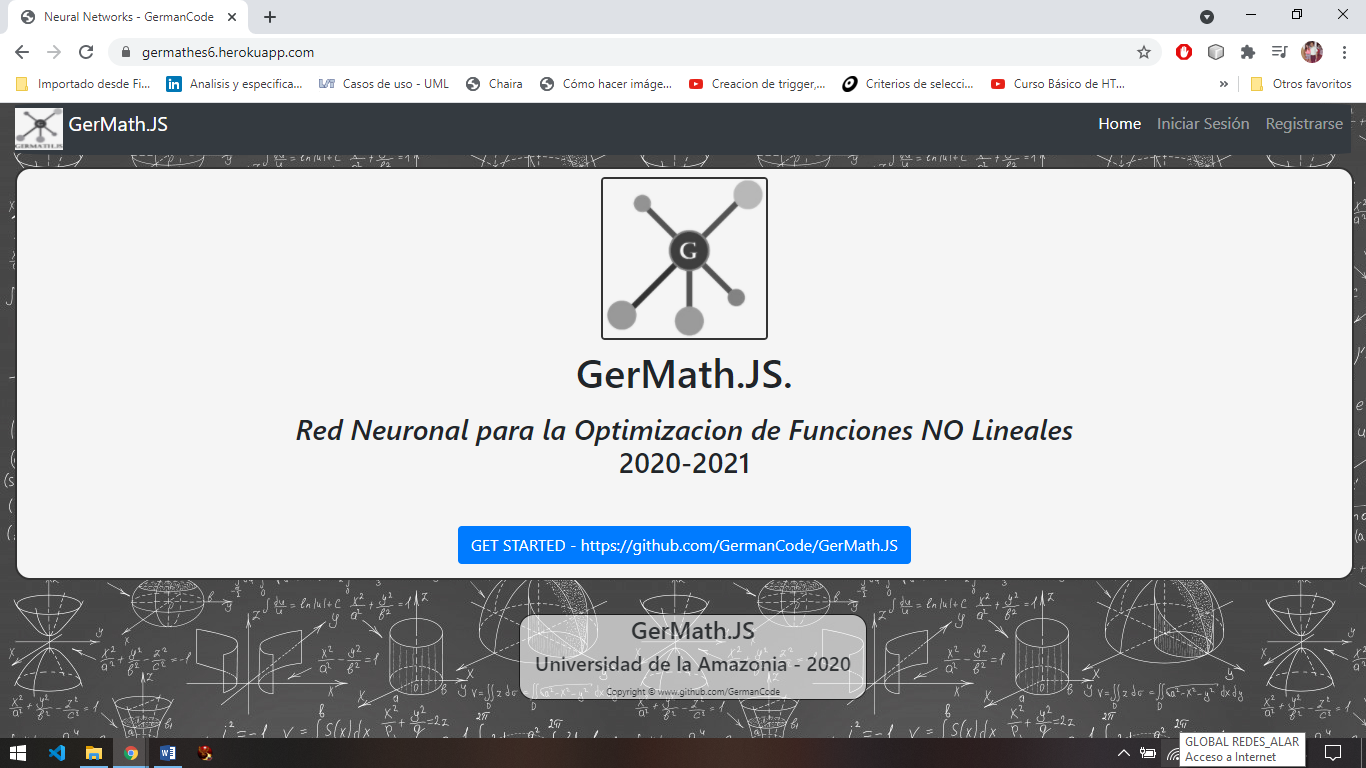
*Aplicacion Wen GerMath.JS en Heroku Running*



Fuente: Elaboración propia

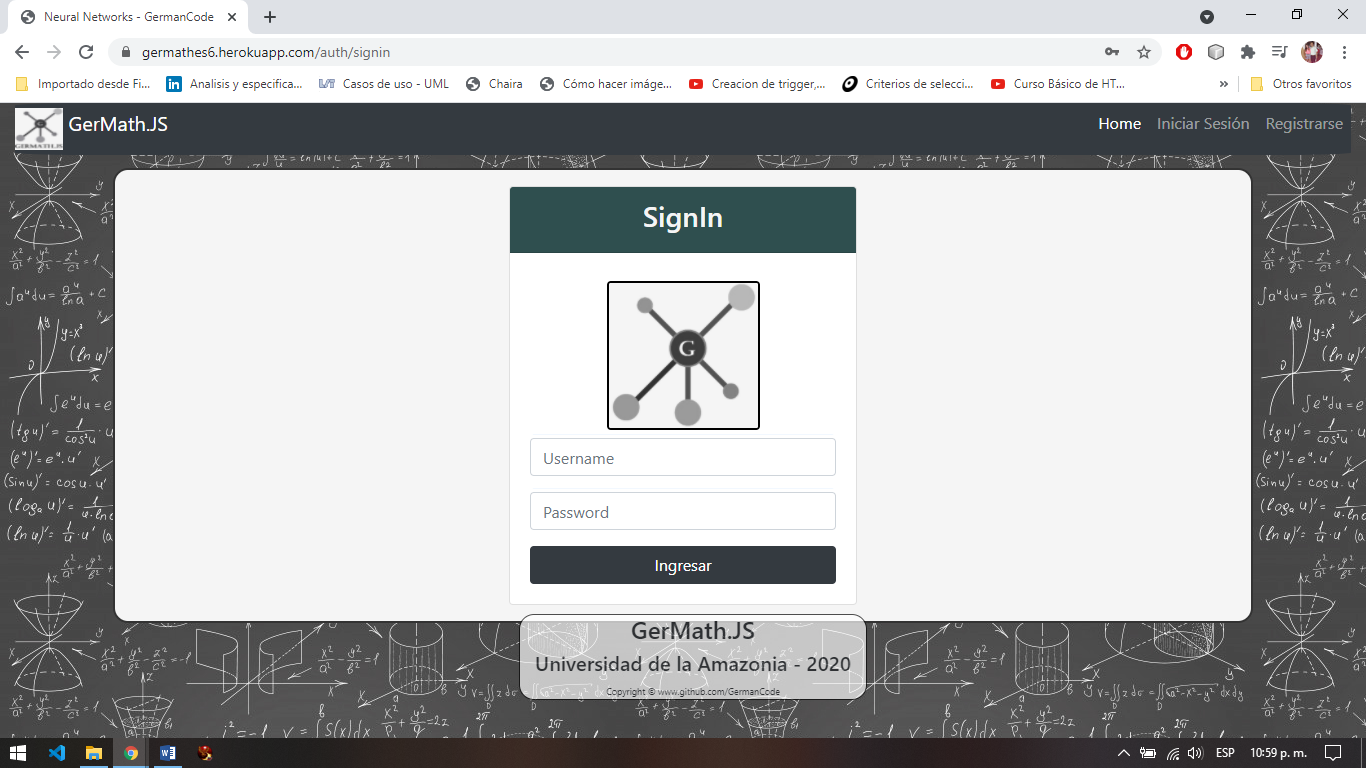
Entonces se fija el prototipo GerMath.JS v. 2.1 garantizando la persistencia web mediante la plataforma Heroku siendo desplegada en la dirección <https://www.germathes6.herokuapp.com> estando disponible para cualquier usuario que necesite hacer uso de ella.

1. Página de Inicio de Aplicación Web GerMath.JS

*Página de Inicio de Aplicación Web GerMath.JS*

Fuente: Elaboración Propia

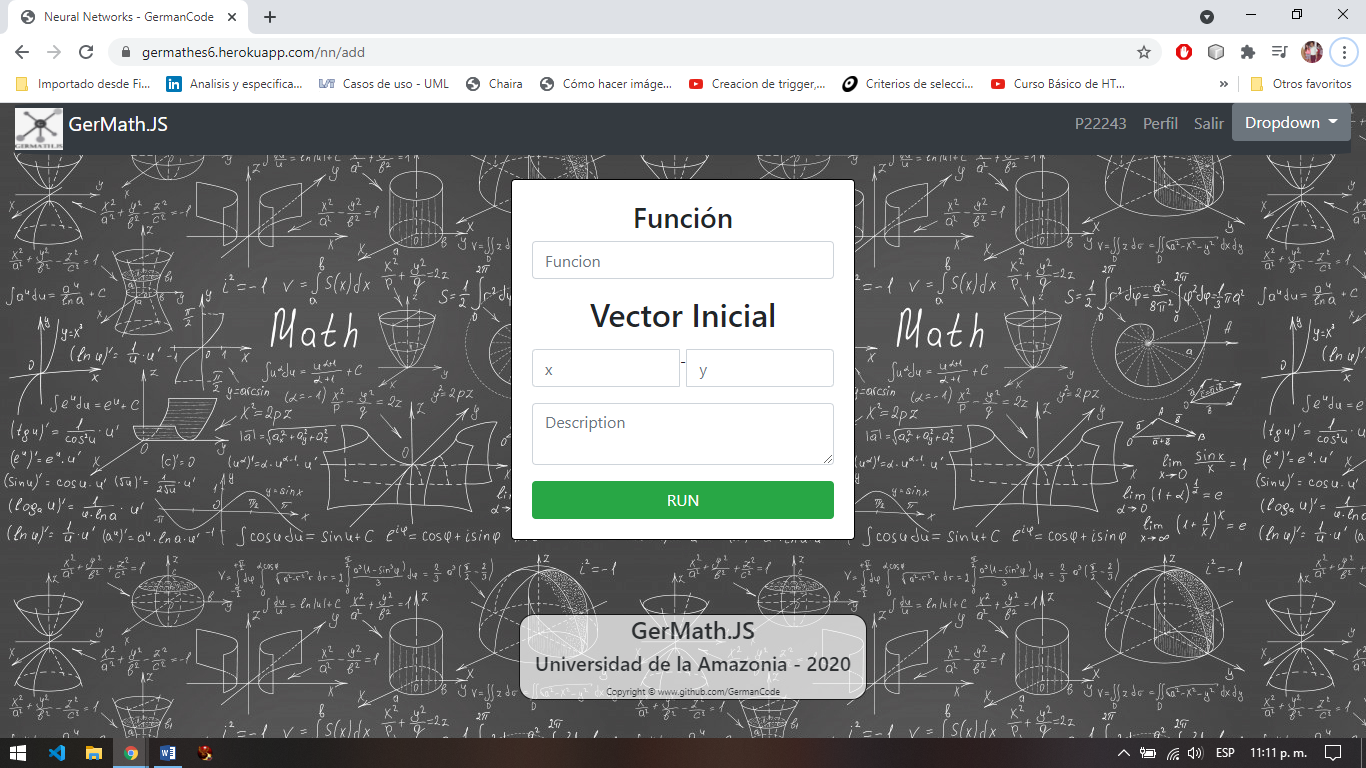
1. Página de Inicio de Sesión Aplicación Web GerMath.JS

*Página de Inicio de Sesión Aplicación Web GerMath.JS*

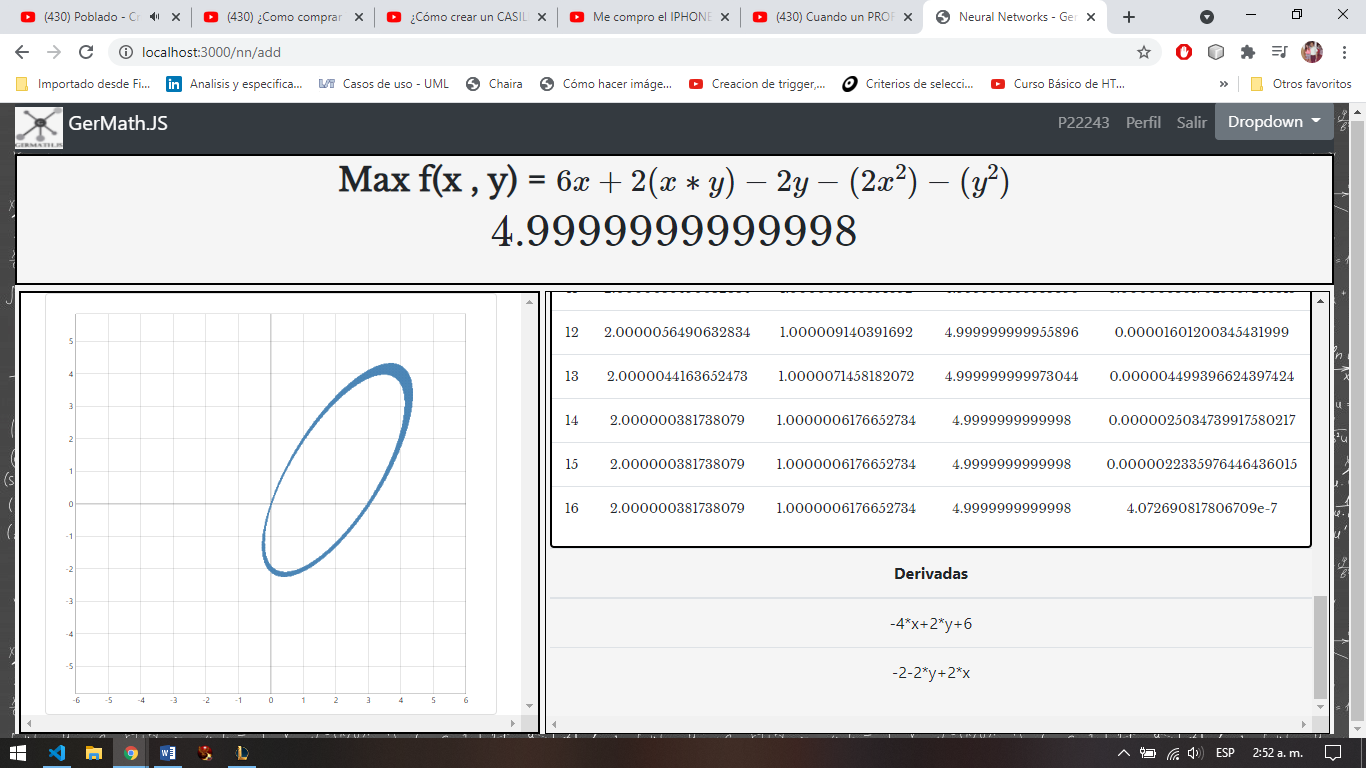
Fuente: Elaboración propia.

1. Página de Optimización de Funciones Aplicación Web GerMath.JS.

Módulo de Optimización de Funciones Aplicación Web GerMath.JS.

Fuente: Elaboración Propia.

1. Página de Resultado en Aplicación Web GerMath.JS

*Página de Resultado en Aplicación Web GerMath.JS*

Fuente: Elaboración propias.

1. Resultados de Prueba Obtenidos

Tabla de resultados obtenidos

|  |  |  |  |
| --- | --- | --- | --- |
| **Problema** | **Inicial** | **Resultado Reportado** | **Resultado Obtenido** |
| Max | [1,1] | [1.994, 0.989]  4.999939 | [1.9999, 0.99999]  4.999999999999 |
| Max | [0,0] | [4,4]  12 | [4.00004, 4.000003]  11.999999998 |

Fuente: Elaboración propia.

# PRODUCTOS ENTREGABLES

Los productos por entregar al final del desarrollo del proyecto son los siguientes:

**Modelo entidad relación:** estructura relacional de la base de datos desarrollada para la aplicación.

**Aplicativo web:** Aplicación web desarrollada en JavaScript, que implementa el framework Express.JS para su funcionamiento, Handdlebars.JS para su renderizacion, y HTML5 para su visualización.

**Documentación de software:** Diagramas, historias de usuario y manual de usuario.

**Manual de usuario:** instrucciones sobre el funcionamiento de la aplicación, y descripciones técnicas de cada componente.

# CRONOGRAMA

### Cronograma de Desarrollo de la Aplicación Web GerMath.JS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **OBJETIVOS** | **TIEMPO DE EJECUCION** | | | | | | | | | | | | | | | |
| **MES 1** | | | | **MES 2** | | | | **MES 3** | | | | **MES 4** | | | |
| **OBJETIVO ESPECIFICO 1** | **SEMANA** | | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Determinar mediante la revisión bibliográfica el algoritmo de solución más apropiado para resolver problemas de Optimización No Lineal con  Restricciones. | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Establecer el tipo de RNA necesaria para abarcar la totalidad del problema |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Identificar el método de aprendizaje óptimo a impartir sobre la red neuronal para solucionar problemas PNL. |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **OBJETIVO ESPECIFICO 2** | **SEMANA** | | | | | | | | | | | | | | | |
| **MES 1** | | | | **MES 2** | | | | **MES 3** | | | | **MES 4** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Planificación: |  | | | | | | | | | | | | | | | |
| Historias de Usuario |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| Revisión del estado de las Historias |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| Confirmación de las Historias |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| Diseño: |  | | | | | | | | | | | | | | | |
| Diseño de la Base de Datos. |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| Revisión, ajustes a la base de datos. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desarrollo: |  | | | | | | | | | | | | | | | |
| Gestión de Usuarios. |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| Gestión de Roles. |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |
| Gestión de Tareas |  |  |  |  |  |  | X | X |  |  |  |  | X |  |  |  |
| Gestor de soluciones |  |  |  |  |  |  |  | X |  |  |  |  | X | X |  |  |
| Analizador de Ecuaciones |  |  |  |  |  |  |  | X | X |  |  |  |  | X | X |  |
| Gestor de aprendizaje neural |  |  |  |  |  |  |  |  | X | X |  |  |  | X | X |  |
| Desarrollo de módulos independientes de PNL |  |  |  |  |  |  |  |  |  | X | X | X |  | X | X |  |
| Solución de PNL con restricciones |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| Solución de PNL sin restricciones |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |
| Despliegue del proyecto. (Beta) |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |
| Pruebas: |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A medida que se desarrolla la Aplicación Web, se aplicaran una serie de pruebas funcionales e iterativas. |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| Documentación: |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Manual de usuario. |  |  | X |  |  |  | X |  |  |  |  | X |  |  |  |  |
| **OBJETIVO 3** | SEMANA | | | | | | | | | | | | | | | |
| **MES 1** | | | | **MES 2** | | | | **MES 3** | | | | **MES 4** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Despliegue e implementación funcional de la aplicación mediante ejercicios de prueba de la documentación |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |

# RESULTADOS

# CONCLUSIÓN

La realización de la aplicación Germath.JS refleja la sinergia especifica entre la utilización de las TIC’S[[6]](#footnote-6), la ingeniería de software y los teoremas matemáticos existentes, esto permitió el desarrollo de una aplicación capaz de resolver problemas de optimización no lineal con y sin restricciones, utilizando el paradigma de las Redes Neuronales Artificiales. Este como propósito fundamental debe apoyar a los estudiantes, empresas o usuarios que requieran optimizar funciones del carácter antes mencionado, ayudando primeramente en la optimización de funciones, recursos, gastos o procesos, favoreciendo la toma de decisiones gerenciales generando un método innovador y eficiente para abordar metaheurísticamente la programación no lineal en la cotidianidad de la región, para esto se establecieron diferentes etapas, el análisis de requerimientos, diseño y desarrollo de la aplicación, así mismo como conceptualización e investigación literaria de los métodos metodologías o algoritmos referentes al problema de estudio, esto para abarcar de mejor manera los conocimientos existentes aplicando la ingeniería en la adaptación de los mismos en una aplicación compacta eficiente y eficaz

Por otra parte, la implementación de la aplicación se realizó en un entorno libre y gratuito haciendo uso de plataformas y tecnologías modernas que soporten el correcto funcionamiento del la aplicación, garantizando la disponibilidad de la misma y de los datos en ella almacenados, es de esta forma como el presente trabajo de grado se ajustó a la necesidad encontrada y documentada en los requerimientos, tanto funcionales como naturales del proyecto, dando solución a todos y cada uno de ellos para concluir de forma satisfactoria este desarrollo.

# TRABAJOS FUTUROS

Dentro del marco de desarrollo e implementación de esta aplicación web se realizaron investigaciones y comprobaciones de teoremas matemáticos, cabe destacar dentro de ellos el algoritmo del gradiente descendente, la función de penalización para las restricciones, y hasta el paradigma de redes neuronales artificiales en la estructura, logrando así resolver problemas de optimización no lineal de hasta dos variables con y sin restricciones, generando módulos de independientes tanto para la red neuronal, como para la infraestructura propia de la aplicación. Esto con el fin de que en un futuro se pueda realizar una ampliación, posterior manejo y trazabilidad al proyecto, esto debido a que como políticas institucionales los derechos de autor de la aplicación antes mencionada se deben ceder a la Universidad de la Amazonia, en síntesis para trabajos futuros que se pretendan implementar en la infraestructura de la Aplicación se tienen: tareas:

* Ampliación del espectro de variables hasta 5 por ejercicio.
* Reutilización de la aplicación como herramienta metodológica para le enseñanza o refuerzo de las asignaturas de componentes de optimización (Métodos Numéricos, Métodos Determinísticos, Métodos Probabilísticos).
* Difundir el uso de la aplicación para así generar datos de trazabilidad de las funciones en ella ingresadas.

# REFERENCIAS

Allende, C. M., & Bouza, C. (05 de 2005). Professor George Bernard Dantzig, Life And Legend. *Revista Investigcion Operacional*, 205-211. Recuperado el 09 de 2019

Anzola, N. S. (2015). Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento usd/copspot intradiario. *ODEÓN*, 113-172.

Arévalo, W., & Atehortúa, A. (2012). Metodología de Software MSF en pequeñas empresas. *ACTIVA, ISSN 2027-8101.*(4), 83-90.

Beck, M. B.-K. (2001). *Manifiesto Ágil*. Obtenido de https://agilemanifesto.org/iso/es/manifesto.html

Blazquez, J. R. (2005). El Perceptron. *Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada*, 16.

Boeree, D. C. (2009). *La Neurona*. (Universidad de Shippensburg) Recuperado el 2019, de http://webspace.ship.edu/cgboer/genesp/neuronas.html

Boirivant, J. A. (2009). La Programación Lineal Aplicación De La Pequeñas y Medianas Empresas. *Reflexiones, 88*, 89 - 105.

Cadavid, A. N., Martínez, J. D., & Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospect, 11*(2), 30 - 39.

Calvo, D. (08 de 12 de 2018). *http://www.diegocalvo.es/perceptron/*. Recuperado el 09 de 2019, de http://www.diegocalvo.es/perceptron/

Canós, J. H., Letelier, P., & Penadés, M. C. (2012). Métodologías Ágiles en el Desarrollo de Software.

Caparrini, F. S. (26 de 12 de 2018). *Redes Neuronales: una visión superficial*. ( Dpto. de Ciencias de la Computación e Inteligencia Artificial) Recuperado el 09 de 2019, de http://www.cs.us.es/~fsancho/?e=72

Daniel David Montenegro Murillo, M. A. (2018). Using Artificial Neural Networks to predict monthly precipitation for the Cali river basin, Colombia. *DYNA UNAL COLOMBIA, 86*(211), 122-130.

Gely, M. C. (2009). Métodos matemáticos de optimización no restringida Búsqueda multivariable. *Universidad Nacional del Centro de la Provincia de Buenos Aires*, 19.

Gil, M. C. (2008). Aplicacion Web. *Departamento de Ingeniería Electrica, Electronica y de control*, 48-58. Recuperado el 04 de 2020

Goic, M. (2005). Optimización de Problemas no lineales. *Universidad de Chile*, 15.

Gutierrez, C. S. (2019). Problemas de Asignación Generalizada: modelización, aplicaciones lógicas y métodos de solución. *Universidad de Valladolid, I*(I), 92.

Gutierrez, J. M., & Pulido, J. A. (2015). The resource constrained project scheduling Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for relay Node deployment in Wireless Sensor Networks. *ELSEIVER, V*, 13.

Hassim, Y. M., & Ghazali, R. (2012). Training a Functional Link Neural Network Using an Artificial Bee Colony for. *ReserchGate*, 7.

Héctor, M., & Briceño, G. (2011). Modelo de Programación Lineal Aplicado a la Red Logística de una Empresa del Sector Plástico. *Corporación Universitaria Minuto De Dios*, 34.

Henao, J. D., & Dumar, M. A. (2007). Modelado Del Precio Del Café Colombiano en la Bolsa de Nueva York Usando Redes Neuronales Artificiales. *Revista Facultad Nacional de Agronomía Medellín, LX*(2), 4129-4144.

Hernández, C. M. (2017). Redes Neuronales para Clasificación: Una aplicación al caso de Riesgos Laborales en Colombia. *PONTIFICIA UNIVERSIDAD JAVERIANA*, 72.

Hidalgo, L. D., & Díaz, H. H. (2010). Aplicación de un modelo de programación lineal en la optimización de un sistema de planeación de requerimientos de materiales (MRP) de dos escalones con restricciones de capacidad. *Ingeniería e Investigación, 30*(1), 168 - 173.

IBM. (01 de 12 de 2001). *Rational Unified Process: Best Practices for Software*. (IBM) Recuperado el 20 de 09 de 2019, de Link

Ínkaya, T., Kayalıgil, S., & Özdemirel, N. E. (2014). Ant Colony Optimization based Clustering Methodology. *ELSEIVER, V*, 51.

Jaramillo, V. H., Vera, D. S., & Barcia, K. (2018). Modelo de Programación Lineal Aplicado a una Empresa PYME de Calzado. *LACCEI - International Multi-Conference for Engineering*, 10.

Larranaga, P., Inza, I., & Moujahid, A. (2005). Tema 8. Redes Neuronales. *Departamento de Ciencias de la Computacion e Inteligencia Artificial, I*, 19.

Lin Li, Z. Y. (2014). Evacuation dynamic and exit optimization of a supermarket based on particle swarm optimization. *Physica A*, 23.

Martínez, A., & Martínez, R. (2010). Guía a Rational Unified Process. *Universidad de Castilla la Mancha*, 15.

Mejía, G., & Elkin, C. (2007). Optimización del proceso logístico en una empresa de colombiana de alimentos congelados y refrigerados. *Revista de Ingeniería - Universidad de los Andes*(26), 8.

Menna, S. (2014). Heurísticas y Metodología de la Ciencia. *Mundo Siglo XXI, IX*(32), 67-77.

Mibelli, G. P. (2005). Modelo Matematico de Programacion No-Lineal Para la Optimizacion de los Costos de Fabricacion de Vigas de Concreto Armado. *Escuela de Ingenieria Civil, I*, 72.

Molina, S. G. (2013). Metodologías Ágiles Enfocadas Al Modelado de Requerimientos. *Universidad Nacional de la Patagonia Austral*(ISSN: 1852 - 4516), 16-17.

Olabe, X. B. (2010). REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES. *Escuela Superior de Ingeniería de Bilbao, EHU, II*, 79.

Ospitia, D. F., & Cortes, L. K. (2019). Optimización de las Utilidades en la Empresa DM&E S.A.S mediante un Modelo de Programación Lineal que permita mejorar su Rendimiento Operacional. *Universidad Piloto de Colombia - Girardot*.

Pitts, W., & Mcculloch, W. S. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Mathematical Biophysics From The University Of Illinois, College Of Medicine, V*, 19.

Ploskas, N., & Samaras, N. (2014). Efficient GPU-based implementations of simplex type algorithms. *ELSEIVER, I*, 19.

Riedmiller, M. (1994). Advanced supervised learning in multi-layer perceptrons — From backpropagation to adaptive learning algorithms. *Elseiver, XVI*(3), 265 - 278.

Sarmiento, L. H. (2018). Modelo de Optimización Multi-Objetivo para la Propagacion de la Produccion Agricola a Pequeña Escala en Santander, Colombia. *Escuela de Estudios Industriales Y Empresariales*, 1117.

Sepúlveda, G. F., Avilez, D. A., & Jaramillo, I. E. (2014). Análisis Del Precio Del Carbón Mediante Redes Neuronales Artificiales (RNA). *Boletín Ciencias de la Tierra*(35), 31-36.

Serna, M. D., Serna, C. A., & Ortega, G. P. (2010). Uso de la programación lineal paramétrica en la solución de un problema de planeación de requerimiento de materiales bajo condiciones de incertidumbre. *Ingenieria E Investigacion, 30*(3), 96-105.

Staff, F. (2020). En 2020 Colombia tendría 50,3 Millones de Habitantes. *Forbes Colombia*, 2.

Torres, L. C. (2011). *Redes Neuronales Artificiales.* Recuperado el 04 de 2020, de https://disi.unal.edu.co/~lctorress/RedNeu/LiRna007.pdf

Trujillo, M. A. (2019). Aproximación A Un Modelo De Programación No Lineal Para La Asignación De Tráfico En La Ciudad De Pereira. *Universidad Libre de Pereira*, 44.

Universidad de Murcía. (2014). Redes neuronales biológicas - [Imagen]. En A. Requena, R. Quintanilla, J. Bolarín, A. Vázquez, A. Bastida, J. Zúñiga, & L. Tomás., *Nuevas Tecnologías y Contaminación de Atmósferas, para PYMEs* (págs. VI - 1). España: Universidad de Murcia. Recuperado el 09 de 2019, de https://www.um.es/LEQ/Atmosferas/Ch-VI-3/C63s4p1.htm

Universidad Nacional Autonoma de Mexico. (2002). Metodologías y procesos de análisis de software. En *Ingenieria de Software* (págs. 11-16). Mexico.

Valdes, M. M., Aleaga, A. M., & Vidal, G. G. (2014). Redes neuronales artificiales en la predicción de insolvencia. Un cambio de paradigma ante recetas tradicionales de prácticas empresariales. *Universidad Tecnológica Equinoccial, V*(2), 21.

Villavicencio, A., Arumí, J. L., & Holzapfel, E. (2011). Planificación de recursos hídricos en zonas de secano usando un modelo de optimización no lineal. *Obras y Proyectos, 10*, 73-80.

Wang, Y. (2014). The Hybrid Genetic Algorithm with two Local Optimization Strategies for traveling salesman problem. *ELSEIVER, LXX*, 16.

1. Método Heurístico: Adj. De la Heurística o relativo a ella. – Método Heurístico. [↑](#footnote-ref-1)
2. Heurística: Conjunto de técnicas, metodologías o métodos utilizados con la intención resolver problemas a través de la creatividad, pensamiento divergente o lateral, siendo el aprendizaje iterativo uno de los métodos más utilizados. [↑](#footnote-ref-2)
3. FASECOLDA: Federación de Aseguradores Colombianos, representa la actividad del sector asegurador frente a las entidades de vigilancia y control [↑](#footnote-ref-3)
4. Sistema de Riesgos Laborales: articula el sistema de prevención de accidentes de trabajo y enfermedades laborales [↑](#footnote-ref-4)
5. **CRUD:**(Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. [↑](#footnote-ref-5)
6. Tecnologías de la Información y las Comunicaciones. [↑](#footnote-ref-6)