~~DESARROLLO DE UNA~~ APLICACIÓN SOFTWARE QUE MEDIANTE EL USO DE ~~USO DE~~ REDES NEURONALES ARTIFICIALES RESUELVA PROBLEMAS DE OPTIMIZACIÓN NO LINEAL~~:~~
(GerMath.JS)

Comentado [u1]: …Resuelve…

Comentado [u2]: Los títulos no terminan en punto

ESTUDIANTE:
WEYNDER GERMAN AGUIRRE BETANCOUR

UNIVERSIDAD DE LA AMAZONIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
FLORENCIA-CAQUETÁ
2020

DESARROLLO DE UNA APLICACIÓN SOFTWARE QUE MEDIANTE EL USO DE USO DE REDES NEURONALES ARTIFICIALES RESUELVA PROBLEMAS DE OPTIMIZACIÓN NO LINEAL.
(GerMath.JS)

ESTUDIANTE:
WEYNDER GERMAN AGUIRRE BETANCOUR
Estudiante, 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

TABLA DE CONTENIDO

1. INTRODUCCIÓN

Los problemas de optimización no lineal en su definición más simple buscan la utilización de métodos, metodologías y/o algoritmos que permitan minimizar o maximizar funciones objetivo. Estas pueden o no, estar sujetas a restricciones según sea el caso [1]. Estos pueden abarcar desde la toma de decisiones en grandes organizaciones, hasta la optimización de los procesos y subprocesos productivos que se lleven a cabo en estas. Es de resaltar como precedente significativo el desarrollo del método Simplex en el año de 1947 por George Dantzig, quien proporciono el método para solucionar problemas lineales restrictos multivariados [2], sin embargo años antes ya se habían realizado avances en cuanto a la optimización, de manera informal y artesanal. Esto debido a la llegada de la Revolución Industrial en 1880, que ocasiono que los pequeños empresarios comenzaran a implementar maquinaria y nuevas tecnologías, logrando un crecimiento en las organizaciones y con ello la necesidad de optimizar los procesos utilizando la Investigación de Operaciones **I-O** [3].

En su campo de acción fundamental la **I-O** se utiliza para solucionar los problemas que surgen en la dirección y/o administración de sistemas de hombres, maquinas, materiales, dinero, en las industrias, en los negocios, en el gobierno, en la defensa entre otros. Notándose entonces que, debido a su enfoque cuantitativo apoyado por las matemáticas, proporciona soluciones eficientes a los problemas que pudieran originarse. Con la aparición de los ordenadores y el continuo avance de estos, se mejoró considerablemente la velocidad y capacidad de procesamiento de grandes cantidades de datos, facilitando así el manejo y solución de estos problemas en las organizaciones.

Tenemos entonces que la ***I-O*** se puede aplicar a cualquier proceso cotidiano logrando modelar las variables que influyen en dicho momento y en el objetivo a analizar, logrando determinar de forma canónica si lo que se desea es minimizar o maximizar el modelo planteado (en la mayoría de casos lo que se busca es maximizar las ganancias o, minimizar los costos), en este desarrollo nos centraremos en la rama de la ***I-O*** que abarca la solución de *Problemas No Lineales* (***PNL***). Los cuales se caracterizan por la ausencia de linealidad, ya sea en la función objetivo, en las restricciones del problema, o en ambos casos. nuestro problema central se establece en ¿cómo solucionar problemas de optimización no lineal restricta?

Para ello utilizaremos el paradigma computacional de las Redes Neuronales Artificiales (**RNA**), el cual se inspira el comportamiento biológico de las neuronas del cerebro humano, desarrollando un sistema computacional conexionista, siendo la neurona la base del procesamiento [4], esta recibe las entradas de datos, luego esta se encarga de procesar esta información, la eficiencia de la **RNA** depende del valor del error estimado en el cálculo, por lo cual, se recalcula el error y se actualiza la red con los nuevos datos de entrada hasta satisfacer EL valor permitido, es entonces que se genera una salida. Sin embargo esto se ve potenciado en la medida que biológicamente las neuronas están diseñadas para funcionar en grupo [5], la salida de una neurona puede ser la entrada de otra. De esta forma la complejidad de la red neuronal es proporcional al problema a resolver, notándose que el número de incógnitas de la función objetivo es el mismo número de entradas en la red, siendo la salida el valor máximo de ganancia o mínimo de perdida buscado.

Comentado [u3]: Punto y seguido

Comentado [u4]: Mismo formato que las anteriores, no-cursiva

Comentado [u5]: Punto y seguido

Comentado [u6]: Utilizar un conector gramatical

Comentado [u7]: Minúscula

Comentado [u8]: coma

Comentado [u9]: …donde esta…

Comentado [u10]: …la…

Comentado [u11]: Punto y seguido

Comentado [u12]: Utilizar un conector gramatical

Comentado [u13]: …a partir de…

2. FORMULACIÓN DEL PROBLEMA

Basados en el avance de las tecnologías de la información a lo largo del tiempo, y su auge significativo de 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* **(*I-A*)**, la *Investigación de Operaciones* **(*I-O*)**, entre otras. Teniendo en cuenta que la **I-O** se considera un campo de estudio bastante extenso, para este desarrollo haremos énfasis en su rama de *Programación* ***NO*** *Lineal* **(PNL)** más exactamente en los problemas de *Optimización* ***(Optimización No Lineal).***

Observamos entonces que~~,~~ los problemas de optimización hacen parte fundamental del desarrollo en las organizaciones [6], aplicándose en diferentes campos tales como en la administración de recursos o materiales [7, 8], la logística [9, 10], en las redes de transporte [11], en optimización financiera [12, 13], entre otras, debido a que desde su descubrimiento se han generado 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 [14], a diferencia ~~de~~ la optimización de funciones **NO** lineales~~, en las cuales~~ 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***[a] de optimización [15] 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 [16, 17, 18], 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 ~~ya sea,~~ a la toma de decisiones, ~~en~~ la ejecución de procesos productivos, ~~en~~ 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***[b] [19] para resolver este tipo de problemas (ONL) encontramos los *Algoritmos Genéticos* [20], *Optimización Por Enjambres* [21], la *Optimización Basada En Colonias De Hormigas* [22], en este desarrollo aplicaremos el concepto de las *Redes Neuronales Artificiales* ***(RNA)*** ~~definido anteriormente, de esta manera~~ buscamos 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

[a] Método Heurístico: Adj. De la Heurística o relativo a ella. – Método Heurístico.
[b] Heurística: Conjunto de técnicas, metodologías o métodos utilizados para resolver un problema.

Comentado [u14]: En

Comentado [u15]: No se aconceja el uso del guíon para separar las palabras. Esta definiendo una sigla y ya. Igual para el resto.

Comentado [u16]: Ya la definió en la introducción.

Comentado [u17]: En

Comentado [u18]: Utilizar otra sigla porque ya está utlizando PNL para "*Problemas No Lineales (**PNL**)*", así esté escrita en letra cursiva. Sobra decir que de aquí en adelante el empleo de la sigla generará confusión ya que habrá claridad respecto a la referencia del término.

Comentado [u19]: No utilizar negrita en partes de la frase, es un documento, no una presentación.

Comentado [u20]: Minúscula

Comentado [u21]: ¿singular o plural?

Comentado [u22]: Punto y seguido

Comentado [u23]: en

Comentado [u24]: Minúscula, sin negrita

Comentado [u25]: Si va a utilizar definiciones en pie de página, pues que estas realmente aporten y se contextualicen con el ejercicio, de lo contrario es mejor omitirlas y que quede a conciencia del lector buscar la referencia. Decir simplemente que los métodos huerísticos son relativos a la heurística, deja un sin sabor. Igual para la definición "b", un método para resolver problemas, pero ¿de qué forma? ¿qué es eso particular que los hace especiales?

Comentado [u26]: Coma antes de "los"

Comentado [u27]: Minúscula

Comentado [u28]: Minúscula

Comentado [u29]: Minúscula

Comentado [u30]: Punto y seguido

Comentado [u31]: Utilizar un conector gramatical

Comentado [u32]: Buscando

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 [23].

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***[c] y el **Sistema de Riesgos Laborales**[d], 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 [24].

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. Por tanto, es necesario implementar una Herramienta Software 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.

Comentado [u33]: ¿por ejemplo?

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

[c] FASECOLDA: Federación de Aseguradores Colombianos, representa la actividad del sector asegurador frente a las entidades de vigilancia y control

[d] Sistema de Riesgos Laborales: articula el sistema de prevención de accidentes de trabajo y enfermedades laborales

3. JUSTIFICACIÓN

En vista del acelerado crecimiento de la población en Colombia según la revista Forbes [25] 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, esto 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 entonces 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.

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 los Servicios Web [26, 27, 28] 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 software soportada en un Servicio Web, que haciendo uso del paradigma de Redes Neuronales Artificiales resuelva los problemas de optimización no lineal que puedan generarse en estas organizaciones, notándose que no existe un método exacto para optimizar funciones objetivo NO lineales, por tanto, para realizar este procedimiento haremos uso de la Heurística, combinando el ***Método del Gradiente Descendente*** y el algoritmo ***Resilent 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, que será beneficiada por la expansión empresarial y laboral.

Comentado [u34]: se

Comentado [u35]: es de

Comentado [u36]: conector gramatical

Comentado [u37]: sin negrita

Comentado [u38]: sin negrita

Comentado [u39]: punto y seguido

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Desarrollar una herramienta software, que mediante el uso de Redes Neuronales Artificiales resuelva problemas de Optimización No Lineal Restricta.

Comentado [u40]: minúscula

4.2 OBJETIVOS ESPECÍFICOS

Realizar una revisión de la literatura concerniente a la solución de Problemas De Optimización No Lineales.

Comentado [u41]: es una actividad, no un objetivo

Determinar los métodos necesarios para Resolver Problemas De Optimización No Lineales Restrictos.

Identificar una metodología de desarrollo software que se ajuste al tiempo planificado para el desarrollo de la herramienta.

Implementar una Servicio Web en el cual se despliegue la Aplicación Software para su posterior utilización.

Evaluar el desempeño de la Red Neuronal Artificial en base a los problemas de optimización conocidos en las literaturas de referencia.

5. MARCO REFERENCIAL

5.1 METODOLOGIAS DE DESARROLLO SOFTWARE.

Si hablamos ~~como tal~~ de la metodología de un proyecto~~, si es~~ tradicional, se basara en planear a cabalidad los detalles rigurosos que pudieran afectar de uno u otro modo el desarrollo de las actividades o fases, sin embargo esto se convierte en un problema, debido al enfoque sistemático y pragmático de las nuevas tecnologías, las cuales causan que las metodologías tradicionales o "pesadas " por llamarlas de algún modo, no se adapten de manera correcta a cambios específicos solicitados por el cliente, de esta forma al realizar una modificación en los requerimientos para tomar otro enfoque, se retrasarían cada una de las fases siguientes del desarrollo, lo cual es catastrófico si hablamos de presupuestos reducidos ~~en una región que apenas empieza a requerir el desarrollo de sistemas de información o aplicaciones software consumibles desde dispositivos móviles~~ 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 el concepto de metodología ágil de desarrollo~~.~~ permitie~~ndo~~ la adaptación rápida a cada cambio que pretenda el cliente~~.~~ (*Tabla 1*~~.~~) debido intrínsecamente a que hace parte del equipo de desarrollo, participando activamente en reuniones, orientando y 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 a la final logra que al usuario final se le entregue lo que realmente ~~estaba~~ busca~~ndo~~, ya que fue el 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 brida~~ría~~ 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 proyecto bastante grandes se recomiendan metodologías tradicionales, dependiendo de cada caso.

5.1.1 METODOLOGIAS DE DESARROLLO TRADICIONAL

Estas metodologías imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace el énfasis en la planificación total de todo el trabajo a realizar, organizando por tareas cronografiadas, dependiendo cada una de la otra, siendo así una cadena de estabones donde el más débil puede arriesgar al resto, no obstantes estas metodologías brindan mecanismos para blindar estos acontecimientos y que el desarrollo no se vea afectado por los cambios o modificaciones que pudieran surgir, es entonces que empieza el proceso de producción del software, la puesta en marcha de la extrema y organizada planificación

Comentado [u42]: ¿en dónde? ¿con qué?

Comentado [u43]: Punto y seguido

Comentado [u44]: La tabla se debe ubicar inmediatamente después al párrafo donde es citada por primera vez. Si le parece que está mejor ubicada dos secciones y más adelante, entonces déjela allá, pero retire la referencia acá.

Comentado [u45]: Signo de puntuación

Comentado [u46]: al

Comentado [u47]: él mismo

Comentado [u48]: plural

Comentado [u49]: ¿Referencia bibliográfica? El Turnitin lo marca entero

Comentado [u50]: Punto y seguido

definida por procesos, roles, actividades, métodos y herramientas que se pueden usar para cumplir el objetivo organizacional del desarrollo, documentando cada cambio rigurosamente mediante UML, llevando el registro del análisis, diseño, implementación, y documentación de sistemas orientados a objetos.

Entre las metodologías tradicionales o "pesadas" podemos citar:

• RUP (Rational Unified Procces) [29]: Esta metodología se caracteriza por ser iterativa e incremental, estar centrada en la arquitectura y basada principalmente por los casos de uso. Incluye el modelo de casos de uso, el código fuente, diagramas de secuencia o de actividades, también se especifican claramente los roles de los usuarios.

• MSF (Microsoft Solution Framework) [30]: Esta metodología se caracteriza por su jerarquía organizacional, notándose que desde la presentación de la propuesta de proyecto se deben cumplir algunos parámetros fundamentales para ser considerado viable o necesario, por otra parte también al ser una metodología tradicional divide sus tareas en procesos y estos en subprocesos, que deben ser documentados como soporte estructural y organizacional del desarrollo software, las fases fundamentales de esta metodología son:

- Visión: es donde el ingeniero de sistemas levanta los requerimientos del proyecto.
- Planificación: es donde se especifica a cabalidad las tareas a realizarse, su duración e impacto.
- Desarrollo: es la fase donde se genera como tal el producto software necesario.
- Estabilización: en esta fase se implanta el software en la organización y se capacita al personal sobre su correcto uso.
- Liberación: es donde se entrega la solución software al cliente, brindando soprte solo en caso de ser necesario.

• Spiral Model [31]: En esta metodología los procesos son representados como una espiral, no hay fases fijas tales como especificación o diseño, como su nombre lo indica, se itera en espiral corrigiendo en cada giro los inconvenientes presentados en ese momento, mitigando los riesgos originados durante el proceso, se llevan a cabo 4 tareas específicas en cada giro:

- Determinar o fijar objetivos: Se identifican objetivos específicos para la fase, también se fijan los productos a obtener, ya sea requerimientos, especificaciones, manual de usuario, se especifican las restricciones y los riesgos inmersos en ese giro.
- Hay una cosa que solo se hace una vez: planificación inicial o previa.
- Análisis del riesgo: son identificados y se realizan actividades para reducir los riesgos clave.
- Desarrollo, verificar y validar: Se escoge un modelo de desarrollo para el sistema, y se prueban las tareas propias de la actividad que se está desarrollando para así evitar pruebas unitarias sobrecargadas.
- Planificar: Se revisa a cabalidad el proyecto, se evalúa el desarrollo parcial, y se toman decisiones sobre la continuidad o por el contrario la mejora en torno a la actividad que se está llevando a cabo, luego de esto se planifica la siguiente fase de la espiral

5.1.2 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 solida 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.** [32]

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 poco a poco se fue creando un consolidado de principios y reglas que se deben seguir, en las cuales se rige esta filosofía metodológica, donde priman los intereses del cliente, la adaptabilidad, y la flexibilidad a los cambios, donde la integración, factorización y encapsulamiento priman sobre temas tan arbitrarios como la documentación, donde se puede cambiar por petición del cliente o por refactoring organizacional, haciendo reingeniería a un método obsoleto. véase Tabla 1.

Tabla 1: Comparativa entre metodologías de desarrollo ágiles y metodologías de desarrollo tradicional. [33]

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

Entre las metodologías ágiles más destacadas hasta el momento se pueden nombrar:

- XP (Extreme Programming) [33]: esta metodología se centra en potenciar las relaciones interpersonales como factor fundamental en el desarrollo, promoviendo el trabajo en equipo, propiciando un buen clima de trabajo con comunicación fluida entre todos los participantes. XP se caracteriza por su realimentación continua entre el cliente y el equipo

Comentado [u51]: A menos que esta referencia sea quien desarrolló la metodología, la referencia no se ubica en la presentación de la sección. Se cita o se realiza la referencia en el apartado (texto) utilizado o inspirado por la misma.

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:

- Historias de Usuario: Es la forma como XP maneja los requerimientos, son tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe 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 de llevarlo a feliz término en cuanto a los rigores del desarrollo.
  - **Tester:** Encargado de pruebas.
  - **Tracker:** Encargado del seguimiento.
  - **Coach:** su papel es orientar y guiar el desarrollo del proyecto.
  - **Big Boss:** Gestor del proyecto.

- Proceso: El ciclo de desarrollo 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.

    En todas las iteraciones que pudieran presentase en un ciclo tanto el cliente como el programador establecen las prioridades y riesgos que surgen conforme avanza el proyecto. Es por esto que no se debe presionar al programador a realizar más trabajo que el estimado, debido a que esto genera una perdida de calidad en el software o no se cumplirán los plazos estimados. El ciclo de vida ideal de XP consta de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Comentado [u52]: ¿referencias bibliográficas?

- DSDM (Dynamic Systems Developmemt Method) [34]: es una metodología de desarrollo creada para entregar la solución correcta en el momento correcto. Utiliza un ciclo de vida iterativo, dividiendo el proyecto en periodos cortos de tiempo, definiendo entregables para cada uno de estos periodos, esta metodología cuenta con roles claramente.

Los principios esta metodología se establecen en la necesidad del negocio como eje central, las entregas a tiempo, el trabajo colaborativo, nunca comprometer la calidad del desarrollo por apresurar las entregas; desarrollar de modo incremental sobre una base sólida, aplicando el desarrollo iterativo, la comunicación clara y continua para llegar así a la demostración de control

- ASD (Adaptive Software Development) [35]: esta metodología se fundamenta la teoría de sistemas adaptativos complejos. Es decir, interpreta los proyectos de software como sistemas adaptativos complejos compuestos por agentes, entornos y salidas, lo cual sería el proyecto terminado. El ciclo de vida de ASD está orientado al cambio y se compone de las fases: especulación, en esta fase se inicia el proyecto y se planifican las características del software, colaboración en esta fase desarrollan las características previamente identificadas determinado los tiempos y costos del desarrollo de cada componente, y aprendizaje en este punto, se revisa su calidad, y se entrega al cliente, la revisión puede hacer necesarias más iteraciones en el desarrollo del producto antes de su entrega final, también posee características propias de las metodólogas del tipo ágil, como ser iterativas, tener marcos de tiempo especificados, además está orientado a los componentes software más que a las tareas y es tolerantes a los cambios.

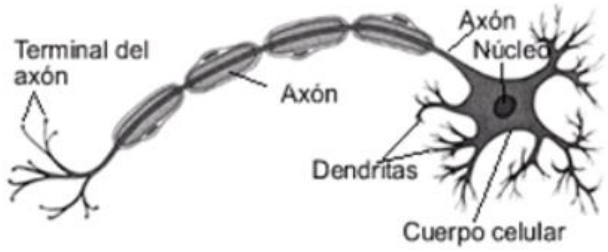
5.2 REDES NEURONALES ARTIFICIALES

5.2.1 Origen.

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. (*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. [36]

Comentado [u53]: Utilizar adecuadamente los signos de puntuación y al menos los mínimos estándares de gramática en la redacción.

Figura 1. Representación gráfica de una neurona biológica. [37]



Estructura de una neurona

5.2.2 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, [38] cada uno con características diferentes apuntando a un mismo objetivo, que sería 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 [39] y en él, se basan las redes neuronales actuales, en este modelo cada neurona consta de un conjunto de entradas S_i. y una sola salida, S_j, cada entrada i es afectada por un coeficiente denominado peso y que se representa por la letra W_{ij}. El subíndice i refleja que el peso afecta a dicha entrada, y j hace alusión a la neurona en ese determinado momento como lo muestra la *figura 2*.

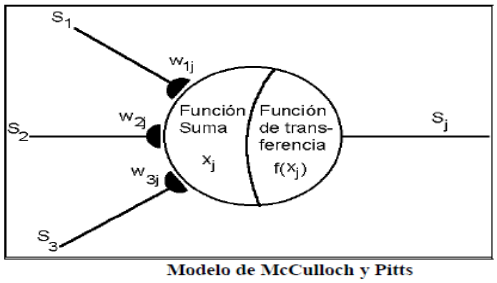
Entonces: ~~la cantidad referente a~~ la suma del producto de cada entrada multiplicada por su peso respectivo se denomina "***Activación***" de la neurona X_j, la salida S_j de la neurona es una función de activación de esta, en resumen:

$$X_j = Sum\ S_i W_{ij} + Q_j \rightarrow Q_j = \text{es un valor "}umbral\text{"} \quad (1)$$

De esta forma:

$$S_j = f(X_j) \quad (2)$$

Figura 2. Modelo de Neurona de McCulloch y Pitts. [39]



Modelo de McCulloch y Pitts

Comentado [u54]: No necesariamente

Comentado [u55]: Las figuras se ubican inmediatamente después al párrafo donde es citada por primera vez. Utilizar el mismo formato en todas las referencia que se utilizan en el texto, como se hizo con la Figura 1.

Comentado [u56]: La suma es la suma, no la activación, esta última ocurre al pasar por la función de activación.

Comentado [u57]: **La definición es confusa, las variables no corresponden a la definición ni la ecuación, prestar mucha atención porque es base fundamental del trabajo que se propone realizar.**

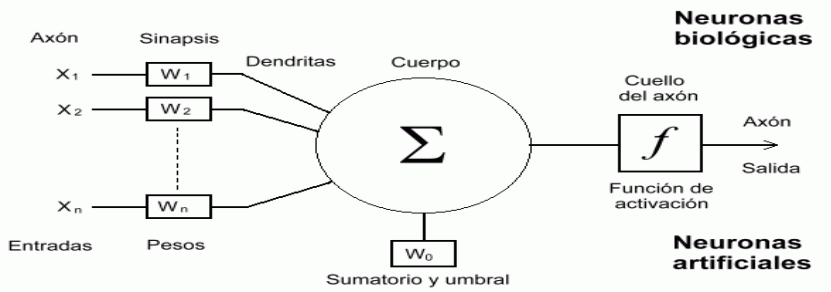
Comentado [u58]: La definición de variables se realiza en otro renglón, no hacen parte de la ecuación

Comentado [u59]: ¿??????????????????

La figura 3, expone un ejemplo de modelo neural con $\boldsymbol{n}$ entradas, que consta de:

Un conjunto de entradas $\boldsymbol{S_i}$, los pesos sinápticos $\boldsymbol{W_i}$, ... , $\boldsymbol{W_n}$, correspondientes a cada entrada, este se va ajustando de forma automática a medida que la red neuronal va aprendiendo, una función de agregación, $\boldsymbol{\Sigma}$, una función de activación, $\boldsymbol{f()}$, mantiene el conjunto de valores de salida normalmente entre (0,1) o (-1,1), la más habitual es la función sigmoidea que se puede observar en la tabla 2, y una salida, $\boldsymbol{Y}$.

Figura 3. Modelo de comparativo de neuronas (biológica – artificial) [40]



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 $\boldsymbol{Y}$ está dada por:

$$\boldsymbol{Y} = f(\sum_{i=1}^{n} \boldsymbol{\omega}_i \, \boldsymbol{x}_i) \quad \textbf{(3)}$$

La función de activación se elige de acuerdo a la tarea realizada por la neurona.

Tabla 2. Diferentes tipos de función de activación. [40]

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = sign(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = tgh(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0,+1]$	
Sinusoidal	$y = A\,\mathrm{sen}(\omega x + \varphi)$	$[-1,+1]$	

Comentado [u60]: **No corresponde a la imagen. Todo estaba marchando bien, pero en este momento estoy empezando a dudar sobre la posibilidad de desarrollar el trabajo, ya que entiendo que no se apropia de la información que está presentando. Sobre todo que está presentado el eje fundamental en la metodología de desarrollo que quiere implementar.**

Comentado [u61]: ¿función de agregación? ¿Qué es esa vaina? **UNA SUMATORIA ¿O ES QUE NO RECONOCE EL SIGNO?**

Comentado [u62]: ¿?????????????

Comentado [u63]: ¿de qué forma?

Comentado [u64]: Esto es una tabla, no una figura, por tanto utilice el edtitor de tablas y genere una con los datos, gráficos, definiciones, ecuaciones, etc., que necesite. NO UTILICE LA IMAGEN DE UNA TABLA PARA HACER REFERENCIA A LA MISMA COMO TAL.

5.2.3 Perceptrón

El perceptrón es la red neuronal más básica que existe, teniendo en cuenta que su aprendizaje es supervisado, se remonta a la década de los años 1.950. El funcionamiento es muy sencillo, simplemente lee las entradas, suma todas las entradas de acuerdo a unos pesos y el resultado lo introduce en una *función de activación* que genera el resultado final, el entrenamiento del perceptrón no es más que determinar los pesos sinápticos y el umbral que haga que la entrada se ajuste a la salida. Para determinar esto, se sigue un proceso adaptativo e iterativo, el proceso da inicio con valores aleatorios y se van modificando estos valores según la diferencia entre los valores deseados y los calculados por el perceptrón utilizando la medida del error para aproximar el próximo resultado o la próxima salida. [41, 42]

1. Inicializar pesos y umbrales
2. Bucle: hasta resultado de pesos sea aceptable
    - Bucle: para todos los ejemplos
        - Leer valores de entrada
        - Calcular error
        - Actualizar pesos según el error
            - Actualizar pesos de entradas
            - Actualizar el umbral

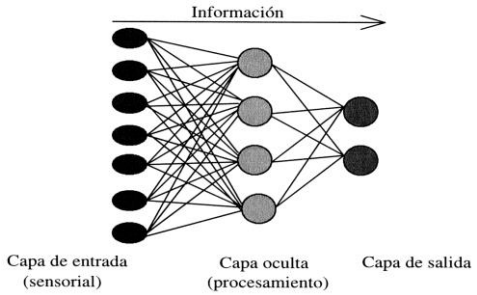
Nota: Solo es capaz de representar funciones lineales debido a que no dispone de capas ocultas.

5.2.4 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 se podrá modificar. Cada conexión de neuronas tiene un peso, una función de entrada y una función de salida~~. (ver~~ figura 4)

Comentado [u65]: ¿??????????????????

Figura 4. Ejemplo de Perceptrón Multicapa. [43]



5.2.5 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.

5.2.6 Función de entrada

El tipo de función de entrada más habitual se llama *Suma Ponderada*. Una neurona de entrada $\boldsymbol{n}$ está conectada a $\boldsymbol{n}$ neuronas de la capa anterior de la red, la suma ponderada $\boldsymbol{A}$ de las entradas para $\boldsymbol{n}$ se calcula añadiendo el producto del valor de entrada de cada conexión por $\boldsymbol{A}$ veces el peso de la conexión $\boldsymbol{\omega}$ a lo largo de todas las entradas $\boldsymbol{n}$. [43]

$$A_N = \sum_{i=1}^{n} a_i \omega_i \qquad \textbf{(4)}$$

5.2.7 Función de salida (activación)

El tipo de función de activación que más habitualmente se usa en las redes MLP es la *Función sigmoidea*. Para una suma ponderada **A** para una neurona determinada, el valor sigmoideo $\boldsymbol{V}$ de $\boldsymbol{A}$, **A** es calculado por:

$$V_A = \frac{1}{1+e^{-A}} \quad \textbf{(5)}$$

Nota: La función sigmoidea es no lineal, lo que la hace muy adecuada para su uso en redes neuronales.

Comentado [u66]: Exponga que hace, que características deben tener, cuando se pueden utilizar unas u otras.

5.2.8 Función de error de red

La función de error más usada con las redes MLP es la *función de error cuadrático medio*. La cual ~~lo que hace es~~ calcula la *"distancia"* media entre el valor real que el programa calcula, y el valor esperado de los datos de capacitación, dadas $\boldsymbol{n}$ neuronas de salida, para cada suma ponderada de la salida $\boldsymbol{A}$, 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 $\boldsymbol{n}$ para llegar al error de salida total $\boldsymbol{E}$. [44]

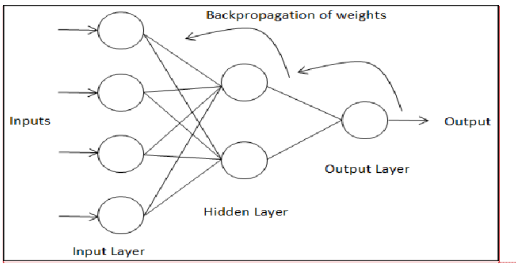
Comentado [u67]: En general ser habla de "entrenamiento".

$$E_o = \frac{1}{n}\sum_{i=1}^{n}(A_i Esperado - A_i Actual)^2 \quad \textbf{(6)}$$

5.2.9 Retropropagación De Error

Cuando una época de capacitación acaba, el programa de capacitación calcula el error de la red neuronal, y modifica los pesos de la conexión a lo largo de la red, empezando en la capa de salida y propagando el error hacia atrás, hacia la capa de entrada, ajustando el peso de cada conexión neuronal mientras recorre la red ver figura 5, a esto se le conoce como retropropagación *de error* y es una técnica ampliamente utilizada para capacitar redes neuronales.

Figura 5. RNA BackPropagation. [45]



Comentado [u68]: Para su trabajo final espero en este punto ver expuesto el algoritmo backpropagation y la respectiva explicación sobre la retropropagación del error y el cálculo de los pesos en cada conexión.

Comentado [u69]: Cuando utilice imágenes que contengan términos en ingles u otro idioma diferente al español, genere una nueva imagen donde aquellos términos se encuentren escritos en español, el idioma en el cual está escribiendo su trabajo. Cuando haga la cita, escribe: traducido de [referencia]

5.2.10 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.*

5.2.11 Aprendizaje NO Supervisado

El entrenamiento no supervisado es aquel en que la red tiene que entender las entradas sin ayuda de la salida, cabe aclarar que en su gran mayoría se prefiere utilizar aprendizaje supervisado; tenemos entonces que la red es alimentada con entradas, pero no con las salidas deseadas, el sistema decidirá las características y atributos que deban tener los datos para ser agrupados, es en este punto donde es imprescindible el cálculo del error, debido a que con la utilización de métodos numéricos podemos aproximar a la salida optima en base al error obtenido, iterando hasta lograr un a medida de error aceptable.

6. MARCO METODOLOGICO

Comentado [u70]: Expone una metodología de desarrollo general, pero no habla de las muestras que piensa tomar para el entrenamiento. Si son de la literatura entonces ¿por qué no están expuestas? Al menos algunas, en este punto ya debe saber a qué se va a enfrentar, sería lo más lógico y sensato. Si son de problemas de las pequeñas y medianas empresas, igual, que tipo de problemas ¿a qué se enfrenta? ¿cuántos tipos de problemas piensa que puede trabajar?

Teniendo en cuenta las diferentes opciones, corrientes y filosofías de desarrollo, es necesario buscar una metodología que brinde seguridad y eficiencia, esta metodología debe cumplir a cabalidad el objetivo propuesto y bridarnos la alternativa de agilizar el proceso de desarrollo siendo un marco ideal y eficiente a la hora de la adaptación a cambios o de reveses definidos por el cliente; Siendo esas las premisas de esta búsqueda, se decidió utilizar la metodología **EXTREME PROGRAMMING (XP)**. (Ver figura 6.)

6.1 CARACTERISTICAS DE LA METODOLOGIA XP

Simplicidad: La simplicidad consiste en desarrollar solo el sistema que realmente se necesita. Implica resolver en cada momento solo las necesidades actuales.

Feedback: Una metodología basada en el desarrollo iterativo de pequeñas partes, con entregas y pruebas frecuentes y continuas, proporciona un flujo de retroinformación valioso para detectar los problemas o desviaciones.

Decisión:

- Implica saber tomar decisiones difíciles.
- Reparar un error cuando se detecta.
- Mejorar el código siempre tras el feedback y las sucesivas iteraciones.

Comunicación: Algunos problemas en los proyectos tienen origen en que alguien no dijo algo importante en algún momento, XP hace casi imposible la falta de comunicación, ya que pone en comunicación directa y continua a clientes y desarrolladores.

Planificación: Se hacen las historias de usuario y se planifica en qué orden se van a hacer y las mini-versiones, la planificación se revisa continuamente.

Versiones Pequeñas: Las versiones simplificadas deben ser lo suficientemente modulares como para poder hacer una en pocas semanas. Deben ser versiones que ofrezcan algo útil al usuario final y no fragmentos de código que no pueda ver funcionando.

Diseño Simple: Hacer siempre lo mínimo imprescindible de la forma más sencilla posible, mantener siempre sencillo el código.

Integración Continua: Debe tenerse siempre un ejecutable del proyecto que funcione y en cuanto se tenga una nueva pequeña funcionalidad, debe recopilarse y probarse. Es un error mantener una versión congelada dos meses mientras se hacen mejoras y luego integrarlas todas de golpe.

Ritmo Sostenible: Se debe trabajar a un ritmo que se pueda mantener indefinidamente. Esto quiere decir que no debe haber días muertos, en que no se sabe qué hacer y no se deben hacer en exceso horas otros días. Hay que trabajar para conseguir el objetivo cercano de terminar una historia de usuario o una mini-versión.

6.2 VENTAJAS Y DESVENTAJAS DE LA EXTREME PROGRAMMING.

6.2.1 VENTAJAS DE LA EXTREME PROGRAMMING:

Una de sus principales ventajas o la más resaltante a la hora de emprender los desarrollos es la gran adaptabilidad que ofrece esta metodología, siendo así escalable, adaptable y elástica, gracias a esto su utilidad y versatilidad se ven inmersas en lo amplio de su campo de acción, otra característica resaltable es la optimización del tiempo de desarrollo, al centrarse solo en los detalles funcionales de la aplicación, dejando de lado detalles o propiedades que no atacan agresivamente la funcionalidad del sistema(Diseño, Documentación, Soporte de Manual).

6.2.2 DESVENTAJAS DE LA EXTREME PROGRAMMING:

Su principal desventaja es que los costes de tiempo y costo de obra de mano, o desarrollo funcional, no pueden ser definidos al inicio del proyecto, debido a su naturaleza iterativa e incremental, que soluciona fallas tan pronto son detectadas, lo cual hace que se aumente el tiempo de desarrollo y el coste, no obstante, también es de aclarar que, en felices términos sin sobresaltos generaría un ahorro sustancial en el tiempo de desarrollo funcional de la aplicación.

7. FASES DE LA METODOLOGIA EXTREME PROGRAMMING.

7.1 FASE DE EXPLORACION

En esta fase se exponen a grandes rasgos las necesidades del cliente en las historias de usuario, para la primer entrega del producto, se prueba las tecnologías que serán utilizadas, y se construye un prototipo con las mismas, evaluando el desempeño y la posible respuesta ante las necesidades venideras, de aquí es donde parte la fase de planeación, sin perder de vista la individualidad de la misma ya que el cliente puede redactar más historias de usuario si lo considera necesario, se recomienda que las historias de usuario no superen las 3 líneas. [33]

7.2 FASE DE PLANEACION DE LA ENTREGA

Ya en esta fase se estable la prioridad de cada historia de usuario o actividad a realizar, esta fase es supremamente importante ya que en términos ingenieriles, puede ser nuestro “Callback”, se iniciaría el desarrollo tan pronto termine esta fase, y a su vez sigue siendo recursivo en cada iteración, debido a que dependiendo del tiempo en semanas que se empleó para llevar a cabo la primer fase del proyecto, incluyendo con el prototipo funcional, sirve de medición para calcular el tiempo en semanas que llevara hacer los cambios solicitados por el cliente, de esa forma el cronometro está en marcha, corriendo en contra, lo que supone que se deba trabajar las 40 horas a la semana, siendo respetuosos con el calendario, con la velocidad del proyecto que es demarcada por la velocidad del equipo de desarrollo y la cantidad de tareas pendientes en esta fase, por lo regular se debe intentar distribuir bien la carga en todos los días de la semana para evitar desiertos o colapsos.

7.3 FASE DE ITERACIONES

Esta fase abarca la serie de iteraciones que deben hacerse antes de entregar el resultado, resolviendo la mayor cantidad de errores que se pueda en cada iteración, lo cual no nos exime que por cuestiones de tiempo o de sobre esfuerzo, se entregue un prototipo funcional con algún tipo de error, este a su vez debe ser tenido en cuenta para trabajar en "Background" o segundo plano haciendo iteraciones individuales para corregir este y los demás errores que se pudieran identificar, sin permitir que dichos erros afecten el comportamiento del prototipo funcional al restarle calidad, siendo un proceso muy transparente de mejora continua; iteraciones de no más de tres semanas para resolver los problemas nuevos como los identificados sin resolver, calculando así con la velocidad del proyecto y la cantidad en semanas que llevara realizar las tareas específicas de las historias de usuario.

7.4 FASE DE PRODUCCION

Pruebas adicionales de funcionabilidad y respuesta, revisiones de rendimiento antes de la migración al entorno del cliente, inclusión de nuevas características dependiendo de las necesidades surgidas por la organización o cambios de última hora, en este punto las entregas se harán cada semana esperando pulir el producto final lo más que se pueda, en caso de funcionalidades extra no documentadas por el cliente a su debido tiempo serán solucionadas en la fase de mantenimiento generando un valor agregado al producto.

7.5 FASE DE MANTENIMIENTO

Mientras la primera versión se encuentra en producción, el sistema debe seguir iterando en la mejora continua, o en la adaptación de nuevas funcionalidades requeridas por el cliente, de esta forma la velocidad de desarrollo disminuirá debido a que se debe dividir el equipo en tanto a soporte como en la programación de las nuevas funciones o la optimización de las existentes.

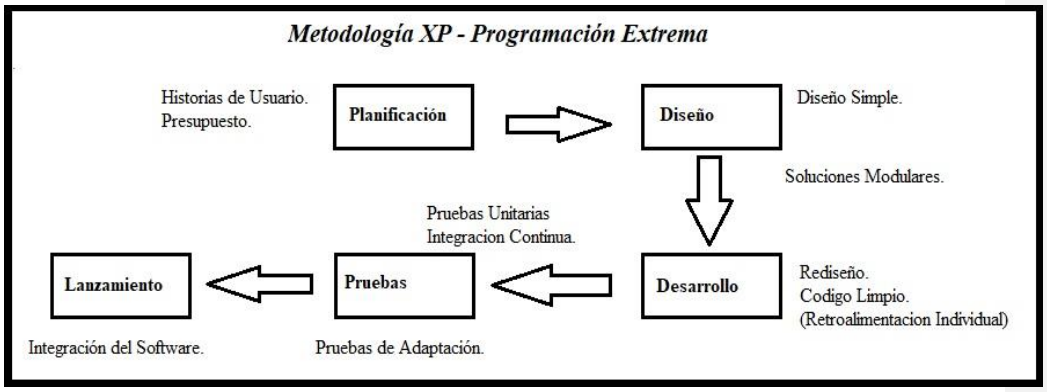
7.6 FASE DE MUERTE

En este punto, el cliente no tendrá más historias para incluir en el sistema, culminando a cabalidad con los requerimientos especificados, esto resulta en que se satisfagan las necesidades del usuario. En otros aspectos como rendimiento y confiabilidad del sistema, abarcar tanto las historias funcionales como las no funcionales, brindando la seguridad, respaldo y estabilidad necesaria de un desarrollo software. Se generará entonces la documentación final del sistema y no se realizan más cambios en la arquitectura siendo este el reléase de esta aplicación.

7.7 ENCAPSULAMIENTO EN FASES FUNCIONALES.

Es de aclarar que en este punto del documento y para facilidad de la abstracción de los diferentes elementos que interactúan en esta metodología, simplificaremos un poco las cosas al nivel de tomar solo 4 fases totalmente funcionales, estas son similares a las etapas de la metodología Rational United Proccess (**RUP)** [44] o cualquiera otra metodología robusta que se base en iteraciones, sin importar su tiempo de retorno.

Figura 6. (Metodología Extreme Programming).



8. CRONOGRAMA

Comentado [u71]: No se ajusta a la metodología presentada, en este caso la crítica es para la presentación de la metodología y no para el cronograma.

OBJETIVOS	TIEMPO DE EJECUCION															
	AGO – 2020				SEP – 2020				OCT - 2020				NOV - 2019			
OBJETIVO ESPECIFICO 1	SEMANA															
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Determinar una metodología ajustable que minimice los costes y tiempos requeridos para el desarrollo de la herramienta software	**X**															
Identificar un lenguaje de programación que mediante la premisa de software libre logre codificar el programa.		**X**														
Establecer el algoritmo de solución más apropiado para dichos problemas.		**X**														
OBJETIVO ESPECIFICO 2	SEMANA															
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
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 3	SEMANA															
	AGO – 2020				SEP – 2020				OCT - 2020				NOV - 2020			
	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.													**X**	**X**		
Pruebas:																
Test: mediante se desarrolla el sistema, cada iteración una prueba.															**X**	
Documentación:																
Manual de usuario.			**X**				**X**					**X**				

OBJETIVO 4	SEMANA															
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Despliegue de la aplicación.																**X**

9. REFERENCIAS

[1] S. P. Hernandez, «Optimización no lineal,» de *OPTIMIZACIÓN Y CONTROL ÓPTIMO*, Cartagena, Colombia, Universidad Politecnica de Cartagena, 2019, p. 29.

[2] www.ecured.cu, «George Bernard Dantzig,» 05 2014. [En línea]. Available: https://www.ecured.cu/George_Bernard_Dantzig. [Último acceso: 09 2019].

[3] M. A. P. Q. Lipicia Munguía Ulloa, Investigacion De Operaciones, Costa Rica: Editorial Universidad Estatal a Distancia, 2005.

[4] L. C. S. Torres, «Redes Neuronales Artificiales,» 2011. [En línea]. Available: https://disi.unal.edu.co/~lctorress/RedNeu/LiRna007.pdf. [Último acceso: 04 2020].

[5] G. Daniel, Principles Of Artificial Neural Networks, World Scientific, 2007.

[6] J. A. Boirivant, «La Programación Lineal Aplicación De La Pequeñas y Medianas Empresas,» *Reflexiones*, vol. 88, pp. 89 - 105, 2009.

[7] L. D. Hidalgo y H. H. T. Díaz, «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*, vol. 30, nº 1, pp. 168 - 173, 2010.

[8] M. D. A. Serna, C. A. Serna y G. P. Ortega, «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*, vol. 30, nº 3, pp. 96-105, 2010.

[9] G. Mejía y C. Elkin, «Optimización del proceso logístico en una empresa de colombiana de alimentos congelados y refrigerados,» *Revista de Ingeniería - Universidad de los Andes*, nº 26, p. 8, 2007.

[10] M. Héctor y G. Briceño, «Modelo de Programación Lineal Aplicado a la Red Logística de una Empresa del Sector Plástico.,» *Corporación Universitaria Minuto De Dios*, p. 34, 2011.

[11] M. A. R. Trujillo, «Aproximación A Un Modelo De Programación No Lineal Para La Asignación De Tráfico En La Ciudad De Pereira,» *Universidad Libre de Pereira*, p. 44, 2019.

[12] V. H. G. Jaramillo, D. S. Vera y K. Barcia, «Modelo de Programación Lineal Aplicado a una Empresa PYME de Calzado,» *LACCEI - International Multi-Conference for Engineering*, p. 10, 2018.

[13] D. F. C. Ospitia y L. K. M. Cortes, «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*, 2019.

[14] N. Ploskas y N. Samaras, «Efficient GPU-based implementations of simplex type algorithms,» *ELSEIVER*, vol. I, p. 19, 2014.

[15] J. M. L. Gutierrez y J. A. G. Pulido, «The resource constrained project scheduling Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for relay Node deployment in Wireless Sensor Networks,» *ELSEIVER,* vol. V, p. 13, 2015.

[16] J. D. V. Henao y M. A. A. Dumar, «Modelado Del Precio Del Café Colombiano en la Bolsa de Nueva York Usando Redes Neuronales Artificiales,» *Revista Facultad Nacional de Agronomía Medellín,* vol. LX, nº 2, pp. 4129-4144., 2007.

[17] G. F. Sepúlveda, D. A. V. Avilez y I. E. V. Jaramillo, «Análisis Del Precio Del Carbón Mediante Redes Neuronales Artificiales (RNA),» *Boletín Ciencias de la Tierra,* nº 35, pp. 31-36, 2014.

[18] A. M. L. A. G. G. V. Marcia M. Lastre Valdes, «Redes neuronales artificiales en la predicción de insolvencia. Un cambio de paradigma ante recetas tradicionales de prácticas empresariales,» *Universidad Tecnológica Equinoccial,* vol. V, nº 2, p. 21, 2014.

[19] S. Menna, «Heurísticas y Metodología de la Ciencia,» *Mundo Siglo XXI,* vol. IX, nº 32, pp. 67-77, 2014.

[20] Y. Wang, «The Hybrid Genetic Algorithm with two Local Optimization Strategies for traveling salesman problem,» *ELSEIVER,* vol. LXX, p. 16, 2014.

[21] Z. Y. Y. C. Lin Li, «Evacuation dynamic and exit optimization of a supermarket based on particle swarm optimization,» *Physica A,* p. 23, 2014.

[22] T. Ínkaya, S. Kayalıgil y N. E. Özdemirel, «Ant Colony Optimization based Clustering Methodology,» *ELSEIVER,* vol. V, p. 51, 2014.

[23] M. A. P. O. V. V. F. Daniel David Montenegro Murillo, «Using Artificial Neural Networks to predict monthly precipitation for the Cali river basin, Colombia,» *DYNA UNAL COLOMBIA,* vol. 86, nº 211, pp. 122-130, 2018.

[24] C. M. Hernández, «Redes Neuronales para Clasificación: Una aplicación al caso de Riesgos Laborales en Colombia,» *PONTIFICIA UNIVERSIDAD JAVERIANA,* p. 72, 2017.

[25] F. Staff, «En 2020 Colombia tendría 50,3 Millones de Habitantes,» *Forbes Colombia,* p. 2, 2020.

[26] W3C, «The World Wide Web Consortium W3C,» W3C, 11 02 2004. [En línea]. Available: https://www.w3.org/TR/ws-arch/. [Último acceso: 05 2020].

[27] GENEXUS DEVELOPER TOOLS, «WEB SERVICES,» 2018. [En línea]. Available: http://library.gxtechnical.com/gxdlsp/pub/GeneXus/Internet/TechnicalPapers/Web_Services.htm. [Último acceso: 04 2020].

[28] IBM, «¿Qué es un servicio web?,» 25 04 2014. [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac55710_.htm. [Último acceso: 04 2020].

[29] A. Martínez y R. Martínez, «Guía a Rational Unified Process,» *Universidad de Castilla la Mancha,* p. 15, 2010.

[30] W. Arévalo y A. Atehortúa, «Metodología de Software MSF en pequeñas empresas,» *ACTIVA, ISSN 2027-8101.*, nº 4, pp. 83-90, 2012.

[31] Universidad Nacional Autonoma de Mexico, «Metodologías y procesos de análisis de software,» de *Ingenieria de Software*, Mexico, 2002, pp. 11-16.

[32] M. B. -. K. Beck, «Manifiesto Ágil,» 2001. https://agilemanifesto.org/iso/es/manifesto.html.

[33] P. Letelier y M. C. Penadés, «Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» *Universidad Politécnica de Valencia.*, p. 17, 2012.

[34] S. G. R. Molina, «Metodologías Ágiles Enfocadas Al Modelado de Requerimientos,» *Universidad Nacional de la Patagonia Austral*, nº ISSN: 1852 - 4516, pp. 16-17, 2013.

[35] A. N. Cadavid, J. D. F. Martínez y J. M. Vélez, «Revisión de metodologías ágiles para el desarrollo de software,» *Prospect*, vol. 11, nº 2, pp. 30 - 39, 2013.

[36] D. C. G. Boeree, «La Neurona,» Universidad de Shippensburg, http://webspace.ship.edu/cgboer/genesp/neuronas.html.

[37] R. Gálvez, «Sistema Nerviosos Humano,» https://weebly.com, sistemanerviosohumano.weebly.com/estructura-neuronal.html.

[38] X. B. Olabe, «REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES,» *Escuela Superior de Ingeniería de Bilbao, EHU*, vol. II, p. 79.

[39] W. Pitts y W. S. Mcculloch, «A Logical Calculus of the Ideas Immanent in Nervous Activity,» *Mathematical Biophysics From The University Of Illinois, College Of Medicine*, vol. V, p. 19, 1943.

[40] F. S. Caparrini, «Redes Neuronales: una visión superficial,» Dpto. de Ciencias de la Computación e Inteligencia Artificial, 26 12 2018. http://www.cs.us.es/~fsancho/?e=72.

[41] D. Calvo, «http://www.diegocalvo.es/perceptron/,» 2018. http://www.diegocalvo.es/perceptron/.

[42] J. R. Blazquez, «El Perceptron,» *Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada*, p. 16, 2005.

[43] P. Larranaga, I. Inza y A. Moujahid, «Tema 8. Redes Neuronales,» *Departamento de Ciencias de la Computacion e Inteligencia Artificial*, vol. I, p. 19, 2015.

[44] IBM, «Rational Unified Process: Best Practices for Software,» IBM, 2001. Link.

[45] DeepAI, «BackPropagation,» DeepAI, 2019. https://deepai.org/machine-learning-glossary-and-terms/backpropagation.

[46] D. Shiffman, The Nature of Code: Simulating Natural Systems with Processing, PapperBack, 2012.