

Desarrollo de un prototipo de software predictivo, sobre el consumo de banda ancha en una red que permita predecir la capacidad necesaria en el futuro

Cristian David Salazar Fandiño
Jeisson Arley López Barrientos

Universitaria Agustiniana
Facultad de Ingeniería
Programa de Tecnología en Desarrollo de Software
Bogotá, D.C.
2020

Desarrollo de un prototipo de software predictivo, sobre el consumo de banda ancha en una red que permita predecir la capacidad necesaria en el futuro

Cristian David Salazar Fandiño
Jeisson Arley López Barrientos

Director
Mauricio Alonso Villalba

Trabajo de grado para optar al título de Tecnólogo en Desarrollo de Software

Universitaria Agustiniana
Facultad de Ingeniería
Programa de Tecnología en Desarrollo de Software
Bogotá, D.C.
2020

Agradecimientos

A Dios

Por darme la sabiduría y fortaleza,

Para lograr esta meta.

A mis padres

Por brindarme su apoyo emocional,

Por impulsarme a cumplir mis sueños,

Y permitirme estudiar esta carrera.

Al semillero Opensgroup

Por ser parte de nuestro crecimiento profesional,

Por su apoyo en la realización de este proyecto,

Por la confianza depositada.

Resumen

En este documento se describirá el procedimiento realizado para el desarrollo de un software de inteligencia artificial específico del área Machine Learning. La finalidad de este software es la predicción de la banda ancha consumida dentro de una red, este sistema podría brindarle a las micro y pequeñas empresas información aproximada sobre la capacidad requerida de este recurso a futuro, el desarrollo se hará siguiendo el modelo de prototipos puesto que es perfecto para llevar a cabo pruebas continuas sobre los modelos predictivos desarrollados, estos serán la pieza clave en determinar los patrones complejos dentro de la colección de datos a utilizar y posterior, poder realizar predicciones sobre el futuro.

Palabras clave: machine learning, predicción, banda ancha.

Abstract

This document will describe the procedure carried out for the development of a specific artificial intelligence software in the Machine Learning area. The purpose of this software is the prediction of the broadband consumed within a network, this system could provide micro and small companies with approximate information on the required capacity of this resource in the future, the development will be done following the prototype model set which is perfect for carrying out continuous tests on the predictive models developed, these will be the key piece in determining the complex patterns within the collection of data to be used and later, being able to make predictions about the future.

Keywords: machine learning, prediction, broadband.

Tabla de contenidos

Introducción	10
1. Título del proyecto de software.....	11
2. Planeación del proyecto de grado.....	12
2.1. Objetivos del proyecto.....	12
2.1.1. Objetivo general.....	12
2.1.2 Objetivos específicos.....	12
2.2. Planteamiento del problema vs necesidad	12
2.3 Alcance del proyecto	13
2.4 Metodología del desarrollo de software	13
3. Marco teórico y estado del arte	19
3.1. Marco teórico.....	19
3.2. Estado del arte	21
4. Especificación de Requisitos de Software (IEEE 830)	27
4.1. Perspectiva del producto.....	27
4.2. Funcionalidad del Producto	27
4.3. Características de los Usuarios	27
4.4. Restricciones.....	27
4.5. Suposiciones y Dependencias.....	28
4.6. Requisitos Específicos	28
4.6.1. Actores/Roles	28
4.6.2. Requisitos Funcionales	28
4.6.3. Diagrama de casos de uso.....	29
4.6.4. Especificaciones de los casos de uso.....	29
4.7. Requisitos de Rendimiento.....	33
4.8. Restricciones de Diseño.....	33
4.9. Atributos de Calidad del Software del Sistema	33
5. Diseño del software (ISO -12207-1)	35
5.1. Diseño de la Arquitectura de Software.....	35
5.2. Diseño detallado del software	35

5.2.1. Diagrama de clases	35
5.2.2. Diagrama de paquetes	36
5.2.3. Diagrama de despliegue	37
5.3. Diseño de la interfaz	37
5.3.1. Interfaz gráfica de Usuario.....	37
5.3.2. Interfaces de Entrada.....	38
5.3.3. Interfaces de Salida	41
6. Implementación.....	43
6.1. Plataformas de desarrollo	43
6.2 Base de datos	43
6.3. Infraestructura de hardware y redes.....	44
7. Pruebas del software.....	45
7.1. Pruebas del software	45
7.2. Pruebas de usabilidad	46
Conclusiones	48
Recomendaciones.....	49
Referencias	50
Anexos.....	52

Lista de tablas

Tabla 1. Tabla de actores.....	28
Tabla 2. Estructura de las tablas de los casos de uso	29
Tabla 3. Caso de uso: Iniciar sesión.....	30
Tabla 4. Caso de uso: Recuperar contraseña.....	30
Tabla 5. Caso de uso: Cerrar sesión	31
Tabla 6. Caso de uso: Cargar un archivo de datos	31
Tabla 7. Caso de uso: Hacer una predicción	32
Tabla 8. Caso de uso: Cargar nuevo archivo de datos	32
Tabla 9. Observaciones pruebas del software	45
Tabla 10. Comentarios prueba de usabilidad	47

Lista de figuras

Figura 1. Modelo de construcción de prototipos.....	13
Figura 2. Ciclo de vida de XP	16
Figura 3. Cronograma de actividades.....	18
Figura 4. Paessler – página de inicio.....	22
Figura 5. Tabla de funciones del software PRTG	23
Figura 6. Clouddradar – página de inicio.....	24
Figura 7. Clouddradar – ejemplo de reporte	25
Figura 8. Nagios – página de inicio.....	25
Figura 9. Nagios – características	26
Figura 10. Diagrama de casos de uso	29
Figura 11. Arquitectura de tres niveles	35
Figura 12. Diagrama de clases	36
Figura 13. Diagrama de paquetes	37
Figura 14. Diagrama de despliegue.....	37
Figura 15. Mock up ventana login	38
Figura 16. Mock up ventana principal	38
Figura 17. Captura de pantalla: login	39
Figura 18. Captura de pantalla: recuperar contraseña	40
Figura 19. Captura de pantalla: cargar archivo	40
Figura 20. Captura de pantalla: pestaña datos cargados.....	41
Figura 21. Captura de pantalla: pestaña datos predichos	42
Figura 22. Base de datos	43
Figura 23. Resultados pruebas del software.....	45
Figura 24. Resultado prueba de usabilidad	46

Introducción

Internet se ha convertido en una parte esencial de nuestra vida cotidiana, ha cambiado la forma en que nos comunicamos, la forma en que vivimos y también la forma en la que se desarrollan las actividades dentro de las empresas, para muchas de estas empresas internet es imprescindible al momento de desarrollar sus actividades económicas y es por esto que no contar con un buen servicio siempre termina en pérdidas económicas , a esto se le suma el elevado costo de planes con mayor capacidad de navegación, y el costo de contratar una línea directa al proveedor de servicios, costos que estas empresas no pueden asumir y por tanto, terminan en problemáticas como caídas del servidor, saturación del canal de comunicación, etc. Es por esto que se planteó el desarrollo de un software que permita a las empresas que lo usen el visualizar gráficos estadísticos sobre la capacidad de navegación que se ha consumido últimamente y la esperada a ser consumida en los próximos días, con el fin de darles oportunidad de formar estrategias que permitan evitar este tipo de inconvenientes.

Este proyecto se realiza especialmente para las pequeñas empresas que no cuentan con los recursos económicos necesarios para adquirir un mejor servicio, es una investigación de tipo cuantitativo y su desarrollo se llevará a cabo a partir de los datos obtenidos a través de una aplicación de terceros que llevará registro de todo el tráfico realizado en el aula en un periodo correspondiente a una semana. Se espera culminar con un prototipo funcional el cual consta de un set de datos limpiado previamente, el algoritmo de aprendizaje automático basado en redes neuronales y entrenado con el set de datos y la interfaz web en la cual se mostrará de manera clara y sencilla la información retornada por el modelo predictivo.

1. Título del proyecto de software

Desarrollo de un prototipo de software predictivo, sobre el consumo de banda ancha en una red que permita predecir la capacidad necesaria en el futuro.

2. Planeación del proyecto de grado

2.1. Objetivos del proyecto

2.1.1. Objetivo general.

Desarrollar un prototipo de software predictivo, que muestre información resumida sobre el consumo de banda ancha en una red y permita predecir la capacidad necesaria en el futuro, con el fin de minimizar las problemáticas derivadas de un mal control de este recurso.

2.1.2 Objetivos específicos.

Diseñar una interfaz web amigable para el usuario en la que se mostrará la información retornada por el software.

Testear y probar el algoritmo de aprendizaje que sea más óptimo para el software, luego se entrenará la red neuronal y se corroborarán los resultados.

Integrar las 2 partes del software (interfaz web y modelo predictivo) utilizando un servidor local que permita la ejecución del código Python en segundo plano.

Codificar y probar las peticiones dentro del servidor que serán utilizadas para llevar a cabo la comunicación entre el cliente que utiliza la página web y el modelo predictivo contenido dentro del servidor local.

2.2. Planteamiento del problema vs necesidad

El uso de internet se ha generalizado tanto que se ha convertido en una forma más de trabajo, muchas microempresas como call-centers, agencias de viaje, servicios varios por internet y empresas dedicadas al comercio electrónico dependen en gran medida de este recurso y de su disponibilidad, pero debido a la mala gestión que se le da, este termina en caídas, tráfico lento, saturación de la red, etc.”,A lo largo del día las empresas cambian mucho la naturaleza del uso que se le da a la red [...] todo esto ocasiona que haya picos de velocidad y de lentitud de la conexión”(s.n,2019). Es por esto que resulta importante el diseñar un software que le permita a estas empresas mantener este recurso vital bajo control, un sistema desde el cual puedan visualizar graficas estadísticas sobre la capacidad de navegación promedio utilizada en las últimas semanas y la esperada para los días futuros, con el fin de que se puedan hacer los ajustes pertinentes de personal y políticas de navegación que les permitan prestar un servicio con la mejor calidad posible sin pensar en una costosa ampliación de infraestructura como solución inicial.

2.3 Alcance del proyecto

El alcance del presente proyecto será el salón 115 de la universitaria agustiniana, este salón es utilizado diariamente por una gran variedad de estudiantes por lo que es perfecto para simular el tráfico dentro de una micro empresa.

2.4 Metodología del desarrollo de software

Debido a que la razón de este proyecto es desarrollar un modelo predictivo que tenga asertividad optima, se realiza teniendo en cuenta un enfoque de tipo cuantitativo, por lo que es necesario realizar continuas pruebas de ensayo y error hasta encontrar el algoritmo de aprendizaje que más se ajuste a la necesidad en cuestión.

Se ha decidido optar por una metodología de desarrollo basada en el modelo de prototipos, puesto que es perfecto para llevar a cabo la investigación y codificación de la distinta variedad de algoritmos de aprendizaje que existen actualmente, para luego tomar una decisión correcta al elegir uno de estos basándonos en los resultados obtenidos, rendimiento y complejidad. El modelo de prototipo nos dice Pfleeger (2002a) es:

Un modelo que permite llevar el desarrollo del software o una de sus partes de manera ágil, pudiendo aclarar y comprender mejor los elementos necesarios en el proyecto, eliminando en gran cantidad la incertidumbre, los prototipos son diseños sencillos y funcionales de cada una de las partes que pueden o no componer nuestro producto final, y es la interacción entre el cliente y el programador la que decide los componentes que serán utilizados y las mejoras a realizar.

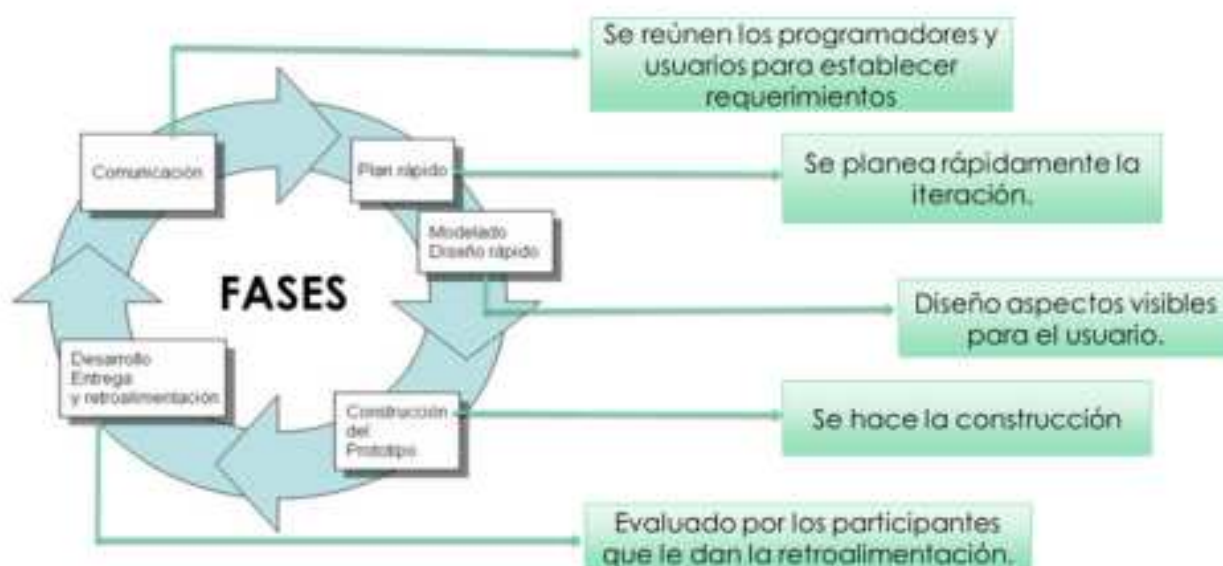


Figura 1. Modelo de construcción de prototipos. Alava (2015).

El modelo de prototipos se compone de 6 etapas:

Comunicación: Es la etapa inicial y el marco de referencia que usará el equipo de desarrollo para completar la tarea encomendada por el cliente, para el modelo de prototipos es importante que el cliente participe en la labor de la construcción del software final puesto que aporta las pautas para llevar a cabo el refinamiento del producto (Pfleeger, 2002b).

En esta etapa se contactó con el docente de la Universitaria Agustiniana, el ingeniero Ramiro Osorio, quien para este proyecto cumple el rol de cliente, en una charla informal expuso su idea de software, los elementos de los cuales se debía componer el software requerido, y los resultados esperados a obtener.

Plan rápido: Para esta etapa se recopila y clasifica la información obtenida en la comunicación, se establecen los requisitos funcionales y no funcionales que debe abarcar el producto de software y se hace una descripción general de los componentes a desarrollar (Pfleeger, 2002c).

Esta etapa comienza reflexionando lo conversado con el cliente, se listaron todos los elementos funcionales y no funcionales que debía poseer el producto de software final, después de estudiar qué instrumentos serían los más correctos a utilizar para llevar a cabo el desarrollo, previa aprobación del cliente, se decidió optar por una interfaz web como medio de visualización de la información retornada por la aplicación y, utilizar el lenguaje Python como lenguaje de programación primario para desarrollar el modelo predictivo gracias a que este lenguaje cuenta con varias herramientas y librerías que facilitan y potencian el desarrollo de redes neuronales y su gran compatibilidad con el Deep learning.

Modelado y diseño rápido: Para esta etapa se revisa toda la documentación de requerimientos con detenimiento y se hace un plan de acción con las especificaciones de software a desarrollar, estas especificaciones se hacen en un nivel de arquitectura alto y se diseña el procesado de los datos (Pfleeger, 2002d).

En esta etapa se llevó a cabo el desarrollo de los wireframes correspondientes a la interfaz web, un wireframes es “Una representación visual en escala de grises de la estructura y funcionalidad de una sola página web o pantalla de aplicación móvil” (Lucidchart, 2019a). Estos después de ser codificados de manera sencilla fueron mostrados al cliente en motivo de que este eligiera cual le resultará más atractivo. También se llevaron a cabo las investigaciones sobre distintas técnicas de machine learning y los distintos modelos predictivos que existen en internet, se eligió uno a uno cada modelo al mismo tiempo que iban cambiando los patrones de aprendizaje.

Construcción del prototipo: Esta es la etapa de desarrollo, se crean, prueban y corrigen los posibles errores del producto consiguiendo bloques de construcción de software que persisten para la creación de nuevos prototipos (Pfleeger, 2002e).

En esta etapa se llevó a cabo la codificación de los distintos wireframes diseñados, paralelo al desarrollo del modelo predictivo para su posterior prueba, también se llevó a cabo la limpieza y recolección de los datos pertenecientes al consumo de internet registrado en el aula 115, siendo guardados dentro de un archivo .csv.

Desarrollo, entrega y retroalimentación: Para esta etapa se planea una reunión con el cliente en la cual se muestra el prototipo desarrollado, este desarrolla pruebas sobre el producto y sugiere modificaciones para lograr cumplir las necesidades reales (Pfleeger, 2002f).

En esta etapa se hizo la correspondiente muestra de los wireframes al cliente, con el fin de poder cumplir sus expectativas o recibir una retroalimentación que nos permitiera llegar a ello, también se llevó a cabo el testeo del modelo predictivo al alimentar este con un 80% de los datos capturados, el 20% de datos restantes fueron utilizados para testear el modelo y calcular la tasa de asertividad de la red neuronal, hasta lograr el 80% planteado inicialmente, en cada retroalimentación se llevó a cabo el desarrollo de un acta con la finalidad de dejar por escrito aquellos puntos positivos y negativos encontrados y una lista con las recomendaciones dadas por el cliente.

Entrega del desarrollo final: Después de haber culminado con el modelo predictivo que logrará satisfacer los requerimientos de nuestro cliente, y diseñar la interfaz de su agrado, se prosiguió con el pulido y desarrollo total de la interfaz, luego comunicamos el modelo predictivo con la interfaz elegida y se realizaron pruebas finales para solucionar errores menores.

Metodología Extreme Programming

Para el desarrollo final del software que compone la creación del sitio web junto con el servidor de peticiones se optó por utilizar la metodología XP (Extreme Programming) pues permite la interacción del grupo de programadores con el cliente y, de ese modo, desarrollar un producto de software que cumpla con los requerimientos y características esperadas.

La metodología ágil de desarrollo XP está centrada en mejorar las relaciones interpersonales al ubicar a los miembros en parejas de programadores, potenciando el trabajo en equipo y alimentando de conocimientos a los desarrolladores, dando lugar a agradables ambientes de trabajo, una comunicación fluida y capacidad para adaptarse a los cambios de requisitos que se pueden presentar a lo largo de la vida del proyecto, es una metodología “Especialmente adecuada para

proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico”(s.n , s.f.).

Roles de la metodología XP

Programador: Es el encargado de desarrollar el producto de software y tomar las decisiones que este proceso requiera, realiza el diseño, implementa el código y lleva a cabo las pruebas.

Jefe de proyecto: Es la persona que asegura las condiciones de un ambiente óptimo en el que se pueda realizar el desarrollo, planea las reuniones entre los programadores y el cliente, guía a los programadores y brinda las herramientas de conocimiento para realizar el software correctamente.

Cliente: Es aquella persona con el requerimiento de software, determina lo que hay que desarrollarse y cuando debe hacerse, decide si aceptar el software o realizar cambios hasta obtener un producto que satisfaga sus necesidades.

Ciclo de vida de la metodología XP

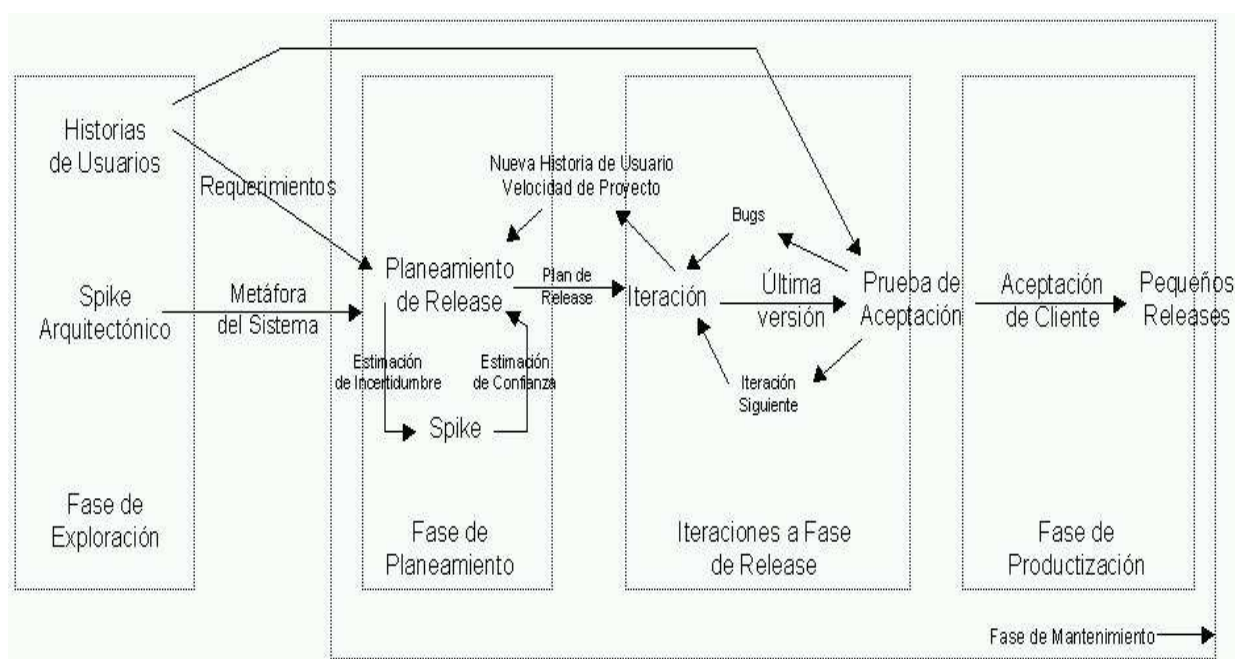


Figura 2. Ciclo de vida de XP, s.n. (s.f.)

La metodología XP se caracteriza por ser un proceso en ciclo en donde el programador y el cliente mantienen una comunicación estrecha y aprenden mutuamente, de esta metodología se pueden distinguir seis etapas (s.n, s.f).

Exploración: Es la etapa inicial donde el cliente plantea el requerimiento de software a grandes rasgos frente a la pareja de programadores y se establecen características que podrían mostrarse en

una primera entrega, es la etapa en donde los programadores se familiarizan con las herramientas o tecnologías que se utilizarán en el proyecto.

Planificación de la entrega: En esta etapa el cliente establece prioridades sobre las características a desarrollar dentro del producto de software y los programadores estiman el tiempo y esfuerzo necesario para realizar cada una, usualmente los tiempos de desarrollo se manejan en semanas ideales de programación.

Iteraciones: Es la etapa de repetición en donde el cliente establece una arquitectura de sistema básica que pueda utilizarse durante todo el desarrollo del proyecto, y en cada iteración se proponen nuevas características que deben ser codificadas y, posteriormente añadidas a la arquitectura básica, se realizan continuamente pruebas de aceptación para conocer si se requieren cambios y así, al final se obtendrá un sistema listo para entrar en la etapa de producción.

Producción: En esta etapa se llevan a cabo pruebas de rendimiento y algunas pruebas adicionales para definir si se requiere la inclusión de nuevas características, por lo que en esta etapa aún se pueden presentar cambios, luego de esto se entrega el producto al cliente para que compruebe su rendimiento.

Mantenimiento. Ya con el producto de software en funcionamiento, se buscan mejoras y la consolidación de una versión estable, esto se logra dando soporte al cliente por un tiempo determinado, para esta etapa se pueden presentar reestructuraciones dentro del equipo de desarrollo puesto que la velocidad de desarrollo es más baja.

Muerte del proyecto: Ocurre cuando el cliente no tiene más opiniones o sugerencias que puedan ser incluidas dentro del producto de software, se generan los respectivos documentos de soporte y se establece que se ha terminado en desarrollo del proyecto.

Cronograma de Actividades

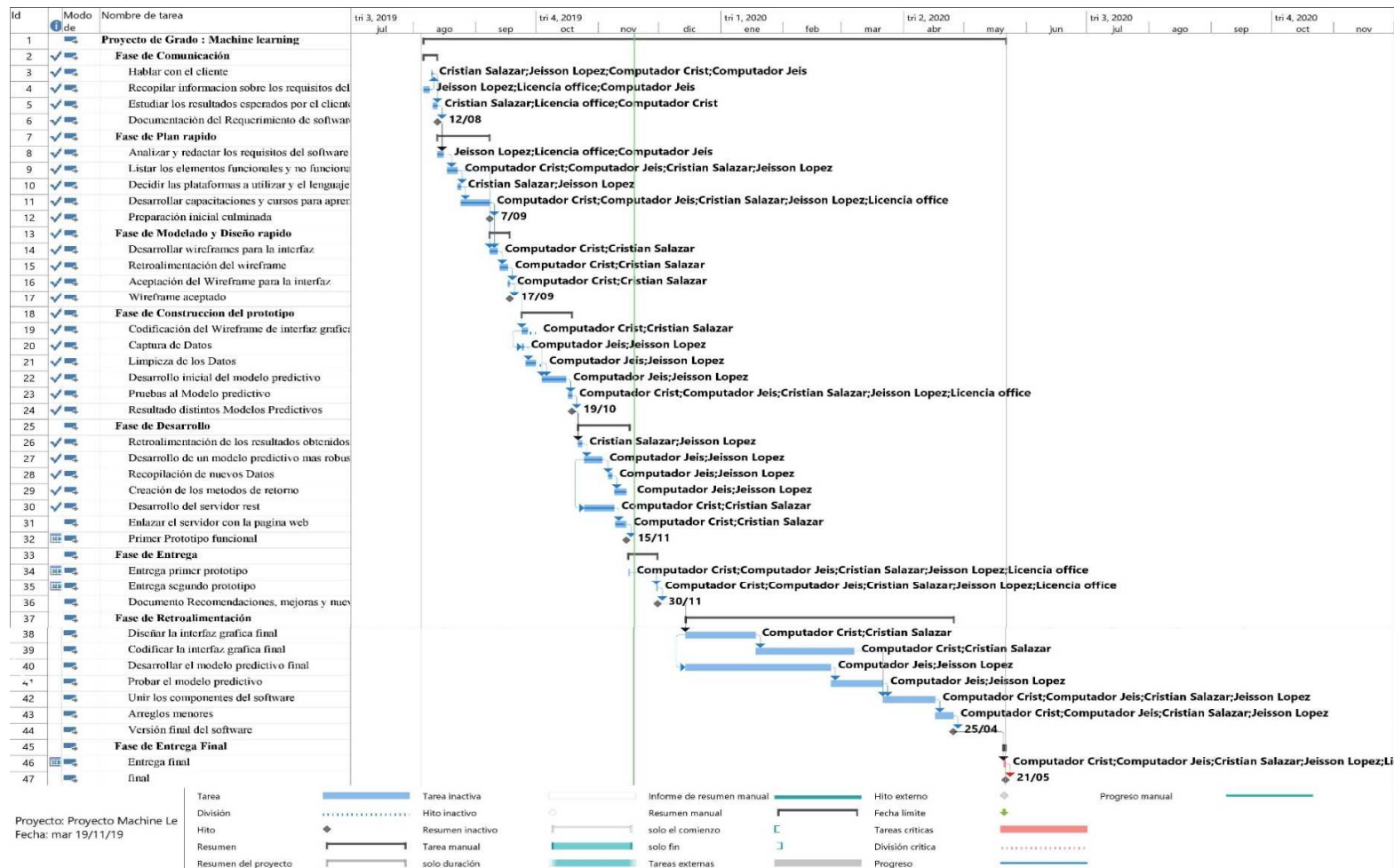


Figura 3. Cronograma de actividades, Elaboración propia.

3. Marco teórico y estado del arte

3.1. Marco teórico

Desde su creación en la década de los 50, internet se ha convertido en una de las herramientas que ha transformado completamente nuestro estilo de vida, permitiendo a tan solo un clic la interacción entre usuarios, en palabras de Castells (2001) “es mucho más que una tecnología. Es un medio de comunicación, de interacción y de organización social”, lo que ha terminado en el surgimiento de nuevas actividades como el comercio electrónico, por comercio electrónico podemos entender todo tipo de ventas, compras o intercambios de bienes y servicios por medio de internet, esto permite a las empresas abarcar un mercado amplio si necesidad de aumentar los costos comerciales y permite a los usuarios realizar compras y transacciones desde la comodidad de la oficina o la casa.

Para las empresas, internet se ha convertido en un canal vital de publicidad con sus clientes objetivo, su costo es mucho menor que cualquier otro medio y permite a cualquier persona, en cualquier lugar del mundo, acceder a la información de productos y servicios ofrecidos, también es una herramienta que ha simplificado las tareas cotidianas a través de sistemas de software que permiten mantener control sobre la organización, documentación, cuentas, logística, etc. Reconocer la importancia de internet dentro de las empresas y la forma en que hace el trabajo más rápido y cómodo nos lleva a preguntarnos ¿cómo controlan las organizaciones este recurso?

Uno de los principales problemas que puede presentarse dentro de la empresa es insuficiencia de ancho de banda debido a una sobrecarga generada sobre el canal de comunicación, ya sea por el sobreuso de terminales conectadas a la misma red o el uso de aplicaciones que no son esenciales para desarrollar las actividades de la empresa, algunos ejemplos de este uso indebido los encontramos en: el uso de redes sociales en horario laboral, servicios de video-streaming, consulta de diarios de noticias.

Cuando se presenta fallas en el servicio de internet dentro de la empresa, aunque sea por pocas horas, puede significar pérdidas en términos económicos, Izquierdo (2015) las divide principalmente en tres categorías:

Pérdida de ventas: Aquellas pérdidas generadas porque el cliente no puede visualizar la información de los productos ofrecidos por la empresa, optando por utilizar los servicios de la competencia.

Pérdida de productividad.: Aquellas pérdidas generadas porque los empleados no pueden realizar sus tareas habituales o, en caso de ser una empresa prestadora de servicios, el tiempo que perderán tus clientes al no tener un acceso estable.

Problemas de Imagen: Es el efecto consecuencia de las dos pérdidas anteriores, los clientes forman una imagen negativa de la empresa.

Una herramienta que podría ayudar a dar solución a este problema conlleva el uso de tecnologías de machine learning para predecir el futuro. El machine learning (en español, aprendizaje de máquinas) es el desarrollo de conceptos de manera automatizada por medio de la utilización de algoritmos de aprendizaje que reconocen patrones dentro de una colección de datos por medio de la experiencia, el Machine Learning es en palabras de Bueno (2017):

Es el desarrollo de procesos que permitan a las máquinas aprender por sí solas a partir de un conjunto de datos que un instructor va introduciendo trabajosamente primero y corrigiendo manualmente después. En este proceso, el ordenador extrae conocimiento a través de experiencia supervisada.

El machine learning se ha convertido en una forma de predecir el futuro como si fuéramos adivinos y ha sido utilizado por grandes empresas para conocer el comportamiento de sus consumidores en pro de no quedar obsoletos y mantenerse como los primeros en el ambiente comercial, utilizando este recurso correctamente y con la debida cantidad de datos relevantes, se hace posible pensar en la aplicación de esta tecnología para hacer predicciones sobre el consumo de ancho de banda dentro del ambiente empresarial como una herramienta más de control sobre este recurso,

El núcleo del machine learning han sido los algoritmos de aprendizaje. Estos son procedimientos que permiten al computador desarrollar la capacidad de aprender sin la programación explícita y constante de los programadores. Los algoritmos de aprendizaje se subdividen en 3 categorías, según el criterio que estos usan para entrenar, estos son: aprendizaje bajo supervisión, aprendizaje sin supervisión y aprendizaje por refuerzo.

Aprendizaje bajo supervisión: En este modelo se le enseña a la máquina a partir de una colección de datos de ejemplo llamados set de datos de entrenamiento, luego de aprender lo suficiente la máquina estará en capacidad de desarrollar predicciones para cualquier entrada.

Aprendizaje sin supervisión: En este modelo se utiliza una colección de datos sin etiquetar o clasificar, es el sistema quien debe agrupar los elementos en categorías.

Aprendizaje por refuerzo: Este modelo se basa en el ensayo y error, se le brinda la capacidad a la máquina de interactuar con el medio ambiente y es esta quien define el comportamiento a desarrollar dentro del contexto establecido.

Para la realización de este proyecto se optó por utilizar una de las tecnologías del machine learning que ha estado revolucionando el campo de las predicciones en los últimos años, el Deep learning y sus redes neuronales, son un tipo de modelo de aprendizaje que simula el funcionamiento del sistema nervioso humano y, más específicamente, la transmisión de información dentro de las neuronas, separando estas en capas, estas aprenden a través del entrenamiento que les brinda el usuario con ejemplos de entrada y la solución que esta debe obtener, en palabras de IBM (2019) son:

Una red neuronal es un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas.

Uno de los elementos que acompaña al Deep learning son los métodos de aprendizaje, el más popular Back Propagación. (en español. propagación hacia atrás), fue introducido por primera vez en la década de los 70 y desde entonces se ha convertido en uno de los algoritmos de entrenamientos más utilizados debido a su eficiencia y rapidez, consta de un algoritmo que, al final de cada salida calcula el error entre el valor obtenido por la capa de salida y lo compara con el resultado esperado, esta diferencia de error modifica las neuronas que generan las salidas empezando desde la capa de salida hasta la capa de entrada ,este proceso es realizado reiteradamente con nuevos valores IBM (2017).

3.2. Estado del arte

Para llevar a cabo el desarrollo del software solicitado se realizó una investigación inicial sobre los productos que existen actualmente en el mercado que comparten similitudes con lo solicitado por el cliente, entre los más destacados se encuentran:

PRTG Network Monitor

Autores: Paessler AG, Dirk Paessler, Jens Rupp

PAESSLER PRODUCTO ▾ PRECIOS RECURSOS ▾ SOPORTE ▾ Blog Empresa ▾ Partners ▾ | 🌐 👤 🔍

Monitoree toda su infraestructura de TI

PRTG monitoriza toda su infraestructura las 24 horas del día, los 7 días de la semana, y le alerta de problemas antes incluso de que los usuarios se den cuenta. Trabaje de forma más rápida e inteligente con PRTG.

PRTG NETWORK MONITOR

DESCARGA GRATUITA

- Versión completa de PRTG de 30 días
- Después de 30 días, se convierte en versión gratuita
- Si le convence, puede adquirir la versión de pago

Descarga	Software de monitoreo de redes Version 19.3.52.3502 (October 1st, 2019)
Idiomas	Inglés, alemán, español, francés, portugués, neerlandés, ruso, japonés y chino simplificado
Monitorización unificada	Dispositivos de red, banda ancha, servidores, aplicaciones, entornos virtuales, sistemas remotos, IoT y mucho más

Figura 4. Paessler – página de inicio. Paessler (2019).

PRTG (*Paessler Router Traffic Grapher*) es un software de monitoreo y clasificación de recursos de red y condiciones del sistema, está desarrollado sobre el lenguaje de programación .NET y su funcionalidad se basa en la recopilación de datos a través de sensores configurados en los distintos dispositivos objetivo a monitorizar, se recomienda su instalación en procesadores de 64bits, con el sistema operativo Windows Server 2019, que tengan instalado .NET Framework 4.7.2 o posterior (Paessler, 2019a).

PRTG cuenta con una gran variedad de sensores que permiten recopilar datos desde distintos protocolos de comunicación como son: Ping, SNMP, WMI, NetFlow, jFlow, sFlow, DICOM o la API REST, así como también permite recoger información propia sobre el funcionamiento de los distintos dispositivos como lo son: los tiempos de respuesta, procesador, memoria, información de la base de datos, temperatura o estado del sistema.

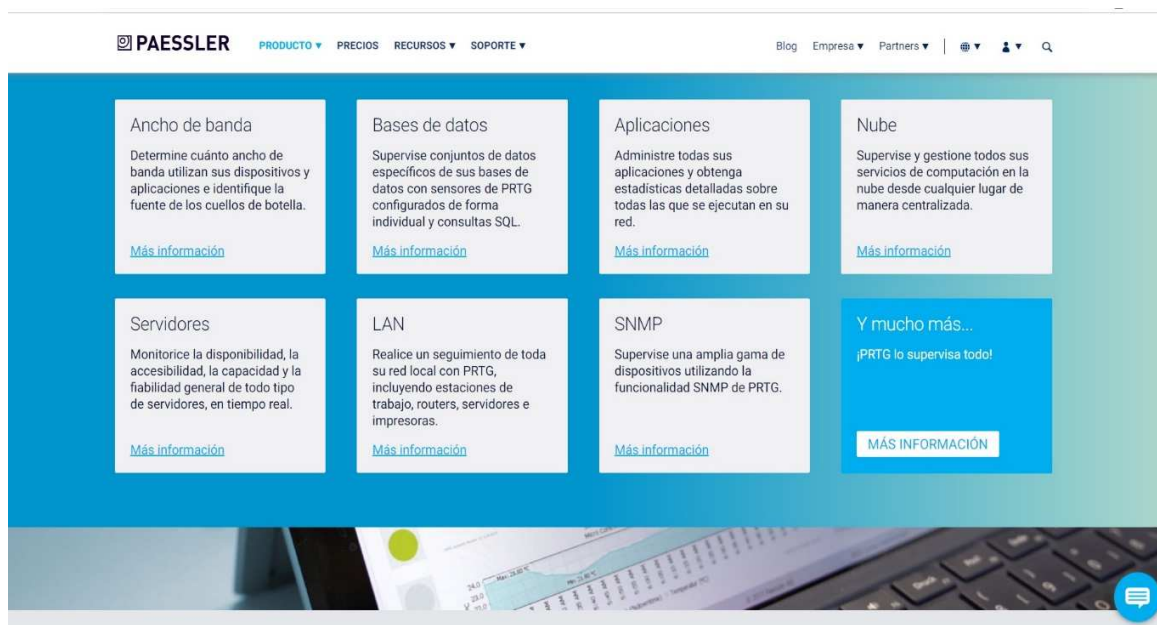


Figura 5. Tabla de funciones del software PRTG. Paessler (2019).

Una de las grandes características que ofrece PRTG es que captura la información de los sensores en tiempo real y permite visualizar la información de maneras distintas, ya sea en una tabla, un diagrama de curvas, etc. Con respecto al ancho de banda, permite visualizar las velocidades de carga y descarga de una red local o wifi y al mismo tiempo clasifica esta información en dispositivos individuales permitiendo conocer la carga que estos ejercen sobre la red, aproximadamente se hacen necesarios 10 sensores en cada dispositivo que se desea monitorizar para obtener toda la información necesaria para generar estos reportes.

Respecto a su costo, la licencia de PRTG se gestiona dependiendo del número de sensores que requiera el cliente, teniendo una licencia gratuita para 100 sensores, mientras que la versión Premium ronda los 1600 dólares en su versión más corta de 500 sensores, hasta los 60 mil dólares con su versión de sensores ilimitados, también está disponible una versión plus ideal para empresas que requieran soporte las 24 horas. (Paessler, 2019b).

Clouddradar

Clouddradar es un software de monitoreo basado en la nube, por lo que no requiere instalación en los equipos para desarrollar la mayor parte de sus funciones, está adaptado para la mayoría de sistemas operativos como son: Windows, Ubuntu, Linux, Windows Server, Debían, etc. Cuenta con un sistema de alertas configurables para alertar al usuario cuando ciertas incidencias ocurran, dichas incidencias se pueden notificar por medio de mensajes de texto o correo electrónico.



Figura 6. Cloudradar – página de inicio. Cloudradar (2019).

Cuenta con funciones de: supervisión de servidores alojados en la web, monitoreo de intranets y el monitoreo de internet, también permite monitorizar el tráfico de clientes de correo electrónico, el tráfico registrado por un sitio web específico, el tráfico de cámaras ip, routers e impresoras conectadas a un cortafuego.

Entre sus características más llamativas están la generación de reportes de ventajas y debilidades, y además permite visualizar la información en forma de tablas y gráficas que permiten ser combinadas entre sí para tener una visión más general del sistema, el costo de la licencia varía de acuerdo al número de dispositivos a monitorizar, con un costo de 1,25 dólares por dispositivo (Cloudradar, 2019).

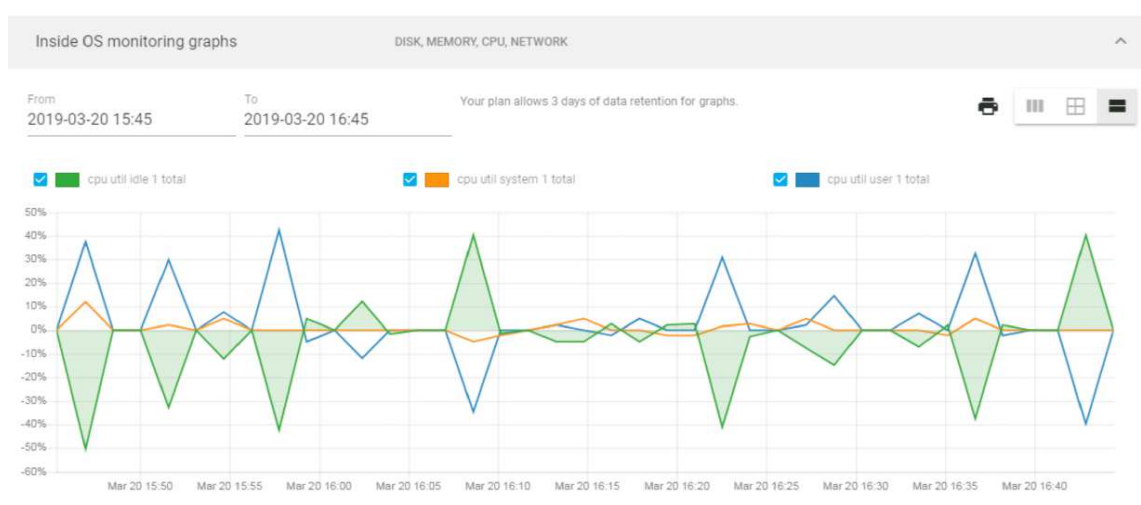


Figura 7. Clouddradar – ejemplo de reporte. Clouddradar (2019).

Nagios Network Analyzer
 Autores: Nagios Enterprise

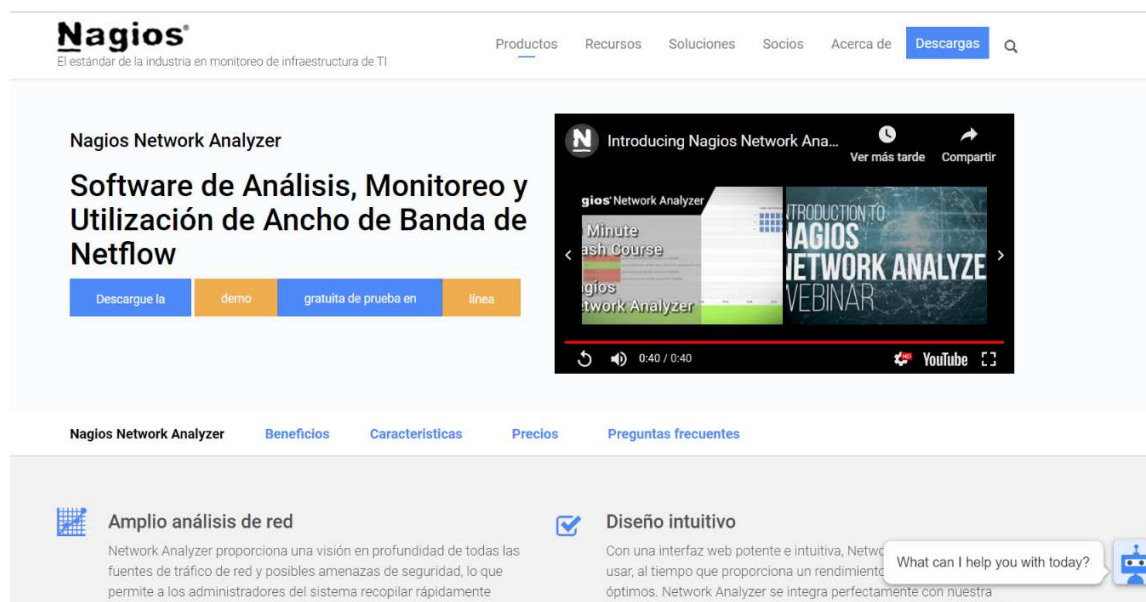


Figura 8. Nagios – página de inicio. Nagios (2019).


Nagios Network Analyzer es un software de monitoreo de tráfico de red basado en el protocolo de red NetFlow desarrollado por Cisco, el software recopila la información retornada por el enrutador con soporte a NetFlow para su posterior almacenamiento y procesamiento, esta información es analizada por el software y mostrada al usuario de forma que le brinde información útil sobre el estado de la red y además cuenta con algoritmos programados que permiten encontrar vulnerabilidades de seguridad en la red.

Cuenta con funciones de alertas y notificaciones para informar al cliente en tiempo real sobre problemáticas que puedan surgir, es un software de fácil manejo y el consumo de recursos sobre el dispositivo es bajo, también permite la descarga de informes con alta flexibilidad en los datos que el usuario quiere visualizar y, también permite la instalación de agentes de monitorización en equipos que no cuenten con soporte a NetFlow.

Cuenta con un tipo de licencia única válida por un año con un costo de 1995 dólares, en la licencia se incluyen 10 sesiones de soporte al cliente en vivo, acceso al foro exclusivo de clientes y actualizaciones gratuitas por un año.


Nagios® [Productos](#) [Recursos](#) [Soluciones](#) [Socios](#) [Acerca de](#) [Descargas](#)

Nagios Network Analyzer [Beneficios](#) [Características](#) [Precios](#) [Preguntas frecuentes](#) [¡Descargar ahora!](#)




Panel de control integral

El panel de control principal proporciona una descripción general de alto nivel de las fuentes, las comprobaciones, los datos de flujo de red y mucho más.



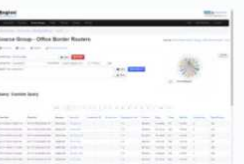
Security and Reliability

Network Analyzer es capaz de alertar a los usuarios cuando se producen actividades sospechosas en la red.




Visualizaciones avanzadas

Las visualizaciones avanzadas proporcionan información rápida y profunda sobre el tráfico de red, el ancho de banda y el estado general de la red.




Supervisión personalizada de aplicaciones

Las consultas, vistas e informes individualizados le permiten supervisar el uso de la red de aplicaciones específicas.




Vistas especializadas

Realice un seguimiento de subconjuntos específicos de información de flujo de red y mantenga datos históricos de flujo de red.




Sistema de alerta automatizado

Reciba alertas cuando tenga lugar una actividad anormal, o cuando el uso del ancho de banda exceda los umbrales especificados.



Integración perfecta con Nagios XI Integre

con Nagios XI para ver los informes de Network Analyzer y quién se comunica con cualquier servidor de su red desde el sistema XI.



Administración avanzada de usuarios

Las opciones avanzadas de usuario permiten a los equipos de TI trabajar juntos de manera eficiente para mantener la red funcionando sin problemas.

Figura 9. Nagios – características. Nagios (2019).

4. Especificación de Requisitos de Software (IEEE 830)

4.1. Perspectiva del producto

Se proyecta desarrollar una primera versión de un software predictivo de banda ancha el cual utilizando un algoritmo de aprendizaje automático y una colección de datos previamente revisados, permita predecir el consumo de internet a partir de datos básicos como la cantidad de datos que entran al dispositivo (Conocido como RX) y la cantidad de datos que salen del dispositivo (Conocido como TX) , también permitirá visualizar la información en formato de tablas y gráficos para su mejor comprensión, es un software independiente ya que no tendrá relación con otros sistemas.

4.2. Funcionalidad del Producto

Las funciones del Software predictivo y estadístico son las siguientes:

Inicio de Sesión: El usuario necesitará unas credenciales para poder acceder al menú principal del software.

Seleccionar archivo de datos: En esta opción el usuario podrá seleccionar un set de datos que contendrá información de tx, rx, ancho de banda y puerto que serán utilizados para entrenar el modelo predictivo, estos deberán estar en un formato preestablecido.

Hacer predicción: Para esta versión inicial del software se utilizará el modelo predictivo para predecir el ancho de banda a partir de los datos de tx y rx del archivo de datos inicial y se compararan los datos predichos con los datos originales.

Cargar nuevo archivo de datos: Esta opción le permitirá al usuario recargar el modelo predictivo con una nueva colección de datos, sin necesidad de cerrar sesión en la aplicación, se deben cumplir unas pautas sobre el formato de los datos a cargar.

4.3. Características de los Usuarios

El software está diseñado para un solo tipo de usuario el cual cumple el rol de administrador, debe poseer un nivel académico de mínimo secundaria, con conocimientos de nivel básico en estadística y manejo de computadoras, con experiencia técnica en manipulación de datos para crear los sets de datos de entrenamiento y retroalimentación.

4.4. Restricciones

Interfaz basada en tecnologías web.

Lenguajes y tecnologías en uso: HTML, PYTHON 3.5.4

La predicción no debe tardar más allá de un tiempo prudencial.

El software deberá tener un diseño intuitivo.

El software deberá tener un rango de asertividad de mínimo 80%.

4.5. Suposiciones y Dependencias

Se advierte que la aplicación desarrollada se encuentra en una versión inicial, por lo cual es propensa a sufrir cambios.

En primera instancia el software será desarrollado para ser ejecutado sobre el sistema operativo Windows 10, para otros sistemas operativos no se garantiza la ejecución correcta del mismo.

4.6. Requisitos Específicos

4.6.1. Actores/Roles

Tabla 1.

Tabla de actores

Nombre	Descripción de Escenario
Administrador	Inicia sesión dentro del software y selecciona el archivo con la colección de datos a utilizar para el entrenamiento, visualiza y descarga los reportes generados por el software y retroalimenta el modelo predictivo.

Nota. Elaboración propia.

4.6.2. Requisitos Funcionales

Se definirán las tareas que debe realizar el software de acuerdo al área que estos pertenezcan-

Inicio de Sesión

Requisito (1) El usuario necesitara credenciales de acceso para ingresar al software.

Requisito (2) Se le permitirá al usuario cambiar su contraseña en caso de que este la olvide.

Selección de archivo de datos

Requisito (3) Se permitirá seleccionar el archivo con la colección de datos debidamente organizados, del cual se mostrarán las gráficas y tablas por parte del software.

Requisito (4) Se permitirá que el usuario seleccione el tipo de gráfico de su agrado y la descarga del mismo.

Hacer predicción

Requisito (5) Para esta primera versión se hará la predicción utilizando los mismos datos de entrada rx, tx para obtener el valor del ancho de banda y se corroborarán los resultados con los datos reales.

Requisito (6) Se generarán las respectivas tablas y gráficos con la información retornada por el modelo predictivo y se permitirá la descarga de estos.

Cargar nuevo archivo de datos

Requisito (7) Se permitirá que el usuario re entrene el modelo predictivo con nuevos datos siempre en cuando estén en un formato establecido.

4.6.3. Diagrama de casos de uso.

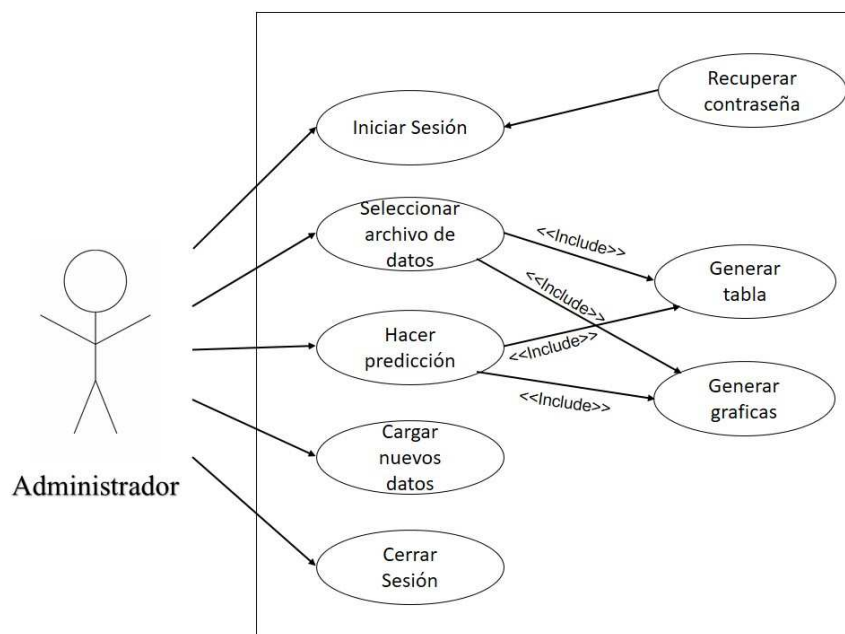


Figura 10. Diagrama de casos de uso. Elaboración propia.

4.6.4. Especificaciones de los casos de uso.

El diagrama de caso de uso es un tipo de gráfico que representa la forma en la que un usuario (denominado actor dentro del diagrama) realiza acciones que permiten la interacción con el producto de software. Sera utilizado el siguiente formato para explicar cada uno de los casos de uso.

Tabla 2.

Estructura de las tablas de los casos de uso

Nombre	Nombre descriptivo que se le otorgará al caso de uso correspondiente.
Autor	Creador del caso de uso.
Fecha	Correspondiente cuando se hizo el caso de uso.
Descripción	Explicación del caso de uso que se están planteando.
Actor	Tipo de usuario del video juego.

Precondiciones	Condiciones que se deberán cumplir antes de determinar el caso.
Flujo Normal	Pasos que debe realizar el actor hasta llegar al caso de uso.
Flujo Alternativo	Pasos que pueden afectar cada FN ya sea por el actor o el sistema.
Post condiciones	Condiciones que se presentan cuando se ejecuta el caso de uso.

Nota. Tabla elaborada por Espinoza y Rodríguez (2020).

Tabla 3.

Caso de uso: Iniciar sesión

Nombre	Iniciar Sesión
Autor	Jeisson López
Fecha	24/10/20
Descripción	El actor entra a la página web Login y se encuentra con un formulario en el cual se le pide ingresar las credenciales de acceso y pulsa el botón iniciar sesión o recuperar contraseña.
Actor	Administrador
Precondiciones	El equipo debe estar funcionando correctamente, el usuario posee los conocimientos básicos para usar un computador.
Flujo Normal	<ol style="list-style-type: none"> 1. Actor ingresa a la página web. 2. Observa el formulario de sesión e ingresa sus credenciales. 3. Pulsa el botón de iniciar sesión o el de recuperar contraseña.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El computador no cumple los requisitos de acceso. 2. El actor no cuenta con las credenciales de acceso. 3. El sitio web deja de funcionar.
Post condiciones	Actor ya se encuentra en la página principal o se encuentra en la página para recuperar su contraseña.

Nota. Elaboración propia.

Tabla 4.

Caso de uso: Recuperar contraseña

Nombre	Recuperar contraseña
Autor	Jeisson López
Fecha	24/10/20
Descripción	El actor entra a la página recuperar contraseña donde se le pedirá que ingrese el correo electrónico, si estos datos son correctos se le enviará un mensaje al correo electrónico donde podrá cambiar su contraseña.
Actor	Administrador
Precondiciones	El actor debió escoger la opción de recuperar contraseña en la página de login.
Flujo Normal	<ol style="list-style-type: none"> 1. Actor ingresa a la página recuperar contraseña. 2. Observa el formulario de recuperación e ingresa su cuenta. de usuario y correo electrónico.

	<ol style="list-style-type: none"> 3. Pulsa el botón de recuperar contraseña. 4. Se le despliega una ventana donde se le pide insertar 2 veces la nueva contraseña. 5. Pulsa el botón confirmar. 6. Vuelve a la página de login.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El computador no cumple los requisitos de acceso. 2. El actor no cuenta con las credenciales de acceso. 3. El sitio web deja de funcionar. 4. la contraseña no se cambia correctamente.
Post condiciones	Actor se encuentra nuevamente en la página de login listo para iniciar sesión correctamente.

Nota. Elaboración propia.

Tabla 5.

Caso de uso: Cerrar sesión

Nombre	Cerrar Sesión
Autor	Jeisson López
Fecha	24/10/20
Descripción	El actor se encuentra en la página principal y pulsa clic sobre el botón Cerrar Sesión, se despliega un cuadro de dialogo de confirmación.
Actor	Administrador
Precondiciones	El actor inicio sesión correctamente en la página web de login.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor se encuentra en la página principal. 2. Pulsa clic en el icono de usuario, luego selecciona Cerrar Sesión. 3. Se abre un cuadro de dialogo en el que se confirma su petición.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El computador no cumple los requisitos de acceso. 2. El sitio web deja de funcionar. 3. Las sesiones no se cierran correctamente.
Post condiciones	Actor se encuentra en la página de login en donde se le pide ingresar sus credenciales de acceso.

Nota. Elaboración propia.

Tabla 6.

Caso de uso: Cargar un archivo de datos

Nombre	Cargar un archivo de datos
Autor	Jeisson López
Fecha	24/10/20
Descripción	El actor se encuentra en la página de carga de datos y se le pide seleccionar el archivo que contiene los datos para alimentar el modelo predictivo.
Actor	Administrador
Precondiciones	El actor inicio sesión correctamente en la página web de login.

Flujo Normal	<ol style="list-style-type: none"> 1. El actor se encuentra en la página principal. 2. Selecciona un archivo de datos desde su ordenador. 3. Se llena la tabla con los datos que fueron cargados, se muestra un histograma y un gráfico de polígono de frecuencias en la parte derecha.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El computador no cumple los requisitos de acceso. 2. El sitio web deja de funcionar. 3. Las tablas y gráficos no se generan. 4. Los datos cargados no cumplen el formato necesario.
Post condiciones	Actor se encuentra en la página principal que muestra los datos solicitados por este.

Nota: Elaboración propia.

Tabla 7.

Caso de uso: Hacer una predicción

Nombre	Hacer una predicción
Autor	Jeisson López
Fecha	24/10/20
Descripción	El actor se encuentra en la página principal y pulsa clic sobre el botón hacer una predicción, se despliega una nueva pestaña con los datos predichos por el modelo predictivo.
Actor	Administrador
Precondiciones	El actor inicio sesión correctamente en la página web de login y cargó correctamente los datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor se encuentra en la página principal. 2. Pulsa clic sobre la opción hacer una predicción. 3. Se activa una nueva pestaña en la que se muestran los datos. predichos por el software y los gráficos correspondientes.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El computador no cumple los requisitos de acceso. 2. El sitio web deja de funcionar. 3. Las tablas y gráficos no se generan. 4. La predicción no se realizó correctamente.
Post condiciones	Al actor se le activa una nueva pestaña “predicción” en la que se visualiza la información retornada por el modelo predictivo.

Nota: Elaboración propia.

Tabla 8.

Caso de uso: Cargar nuevo archivo de datos

Nombre	Cargar nuevo archivo de datos
Autor	Jeisson López
Fecha	24/10/20

Descripción	El actor se encuentra en la página principal y pulsa clic sobre el botón Cargar nuevos datos, se despliega un cuadro de dialogo donde se le solicita seleccionar un archivo de datos desde el ordenador.
Actor	Administrador
Precondiciones	El actor inicio sesión correctamente en la página web de login
Flujo Normal	<ol style="list-style-type: none"> 1. El actor se encuentra en la página principal. 2. Da clic sobre el botón cargar nuevos datos. 3. Se abre un cuadro de dialogo donde se le pide seleccionar un nuevo archivo de datos. 4. Se activa y recarga la pestaña “predicción” donde se visualizan los datos retornados por el modelo predictivo.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El computador no cumple los requisitos de acceso. 2. El sitio web deja de funcionar 3. Las tablas y gráficos no se generan 4. La nueva pestaña no se activa.
Post condiciones	Al actor se le activa una nueva pestaña “predicción” en la que se visualiza la información retornada por el modelo predictivo.

Nota: Elaboración propia.

4.7. Requisitos de Rendimiento

Al realizar la alimentación del modelo predictivo, este no debe tardar más de 5 minutos con un set de datos de 1000 entradas.

Al momento de realizar predicciones, no se debe sobrecargar la CPU y el tiempo de respuesta no debe superar 5 minutos.

La descarga de tablas y graficas no puede tardar más de 30 segundos.

Se espera que la red neuronal sea capaz de soportar una colección de 10000 entradas.

4.8. Restricciones de Diseño.

Para la parte de la interfaz web se tendrán en cuenta las Recomendaciones publicadas por el World Wide Web Consortium (W3C).

4.9. Atributos de Calidad del Software del Sistema

Al momento de que el usuario desee acceder al software se le pedirá un nombre y clave de acceso valida y el software deberá comprobar si se trata de un usuario autorizado, si los datos no corresponden no se le permitirá el ingreso.

El software contará con una guía rápida de instalación y configuración que puede ser encontrada en el anexo 14: Guía de instalación

5. Diseño del software (ISO -12207-1)

5.1. Diseño de la Arquitectura de Software

El desarrollo de este software se realizó siguiendo el patrón de arquitectura de tres niveles.

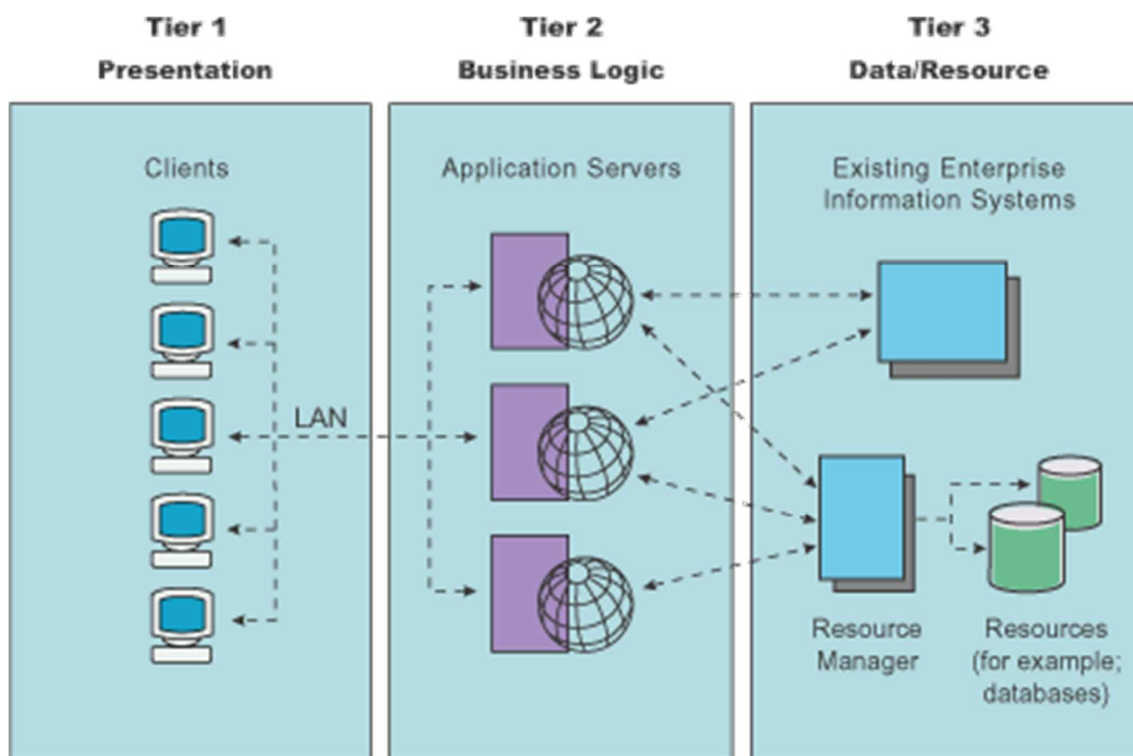


Figura 11. Arquitectura de tres niveles, IBM (s.f.).

5.2. Diseño detallado del software

5.2.1. Diagrama de clases

El diagrama de clase es uno de los diagramas más utilizados al momento de representar el funcionamiento de un producto de software, permite el modelado de las clases, los atributos, las operaciones y relaciones que existen entre los objetos que la componen. (Lucidchart 2019b).

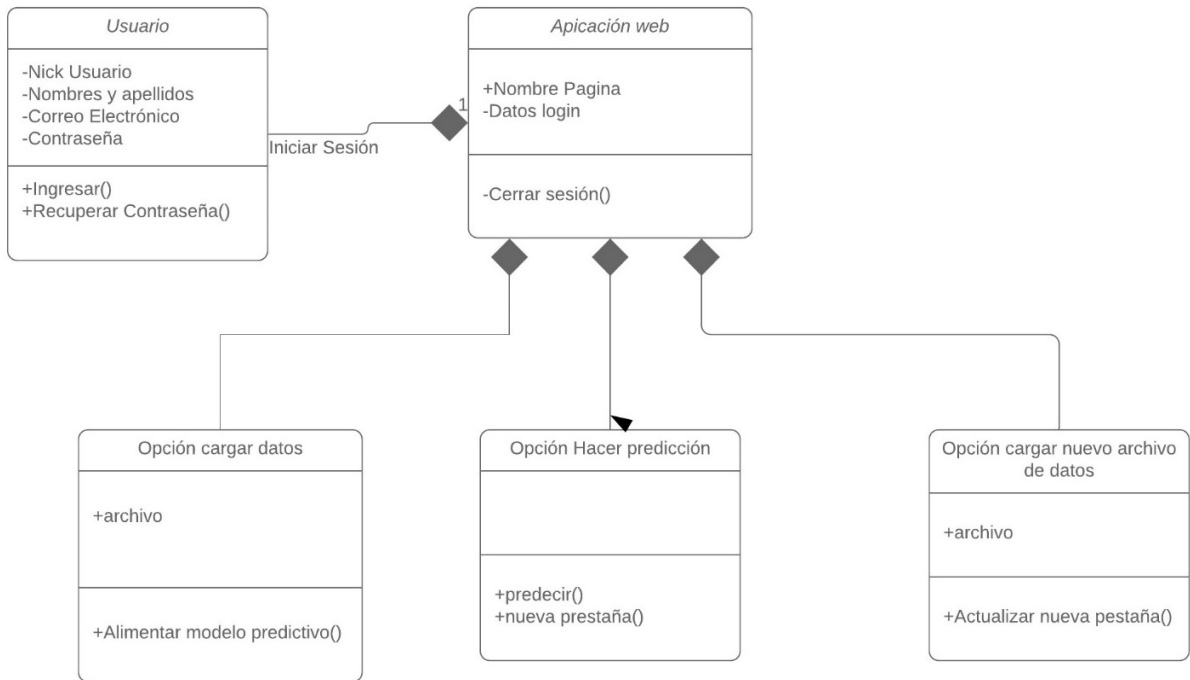


Figura 12. Diagrama de clases, Elaboración propia.

5.2.2. Diagrama de paquetes

El diagrama de paquetes es un diagrama enfocado en representar la estructura del producto de software, los paquetes son un conjunto de elementos que comparten características en común o forman parte de la misma dependencia

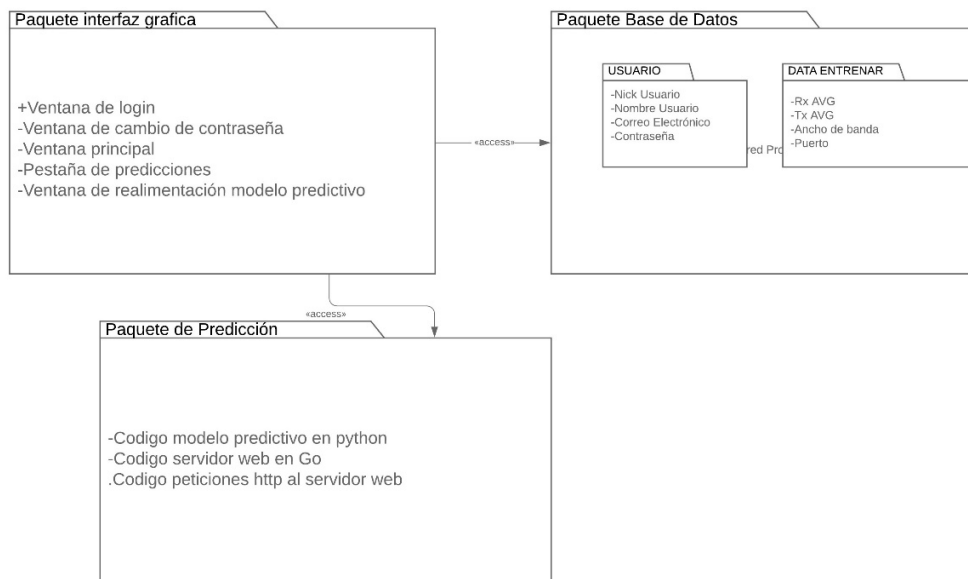


Figura 13. Diagrama de paquetes, Elaboración propia.

5.2.3. Diagrama de despliegue

El diagrama de despliegue es un diagrama que representa la información generada en cada uno de los distintos componentes del software, a esta información se le conoce como artefacto.

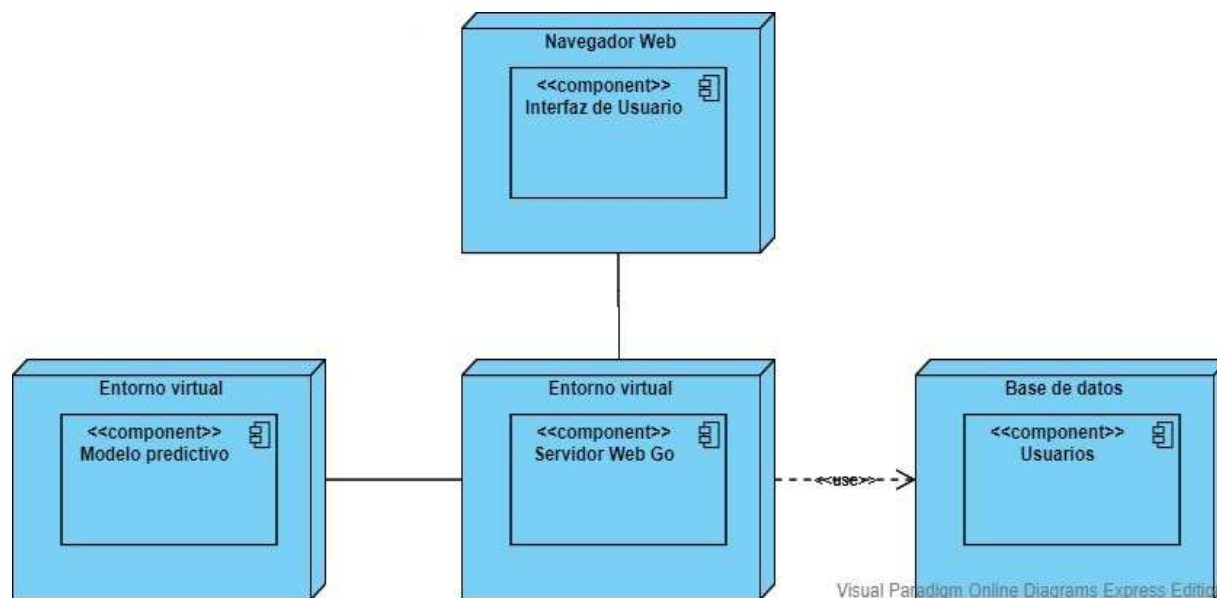


Figura 14. Diagrama de despliegue, Elaboración Propia.

5.3. Diseño de la interfaz

5.3.1. Interfaz gráfica de Usuario

A continuación, se detallarán cada una de las ventanas de las cuales se compone el software y se describirá su forma de uso, se mostrarán algunos Mock ups del diseño preliminar planeado al inicio de desarrollo del software

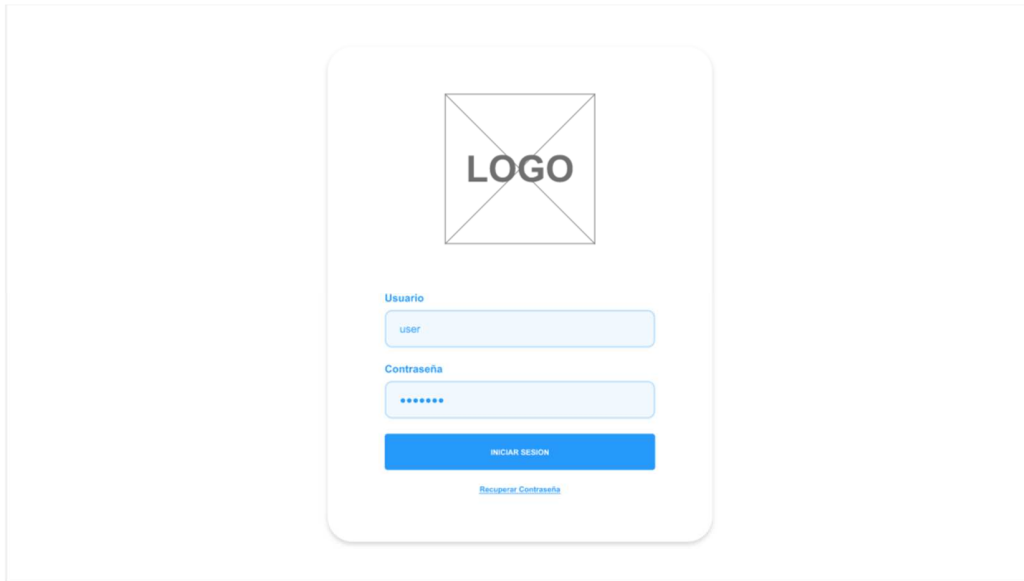


Figura 15. Mock up ventana login, Elaboración propia.



Figura 16. Mock up ventana principal, Elaboración propia.

5.3.2. Interfaces de Entrada

Login

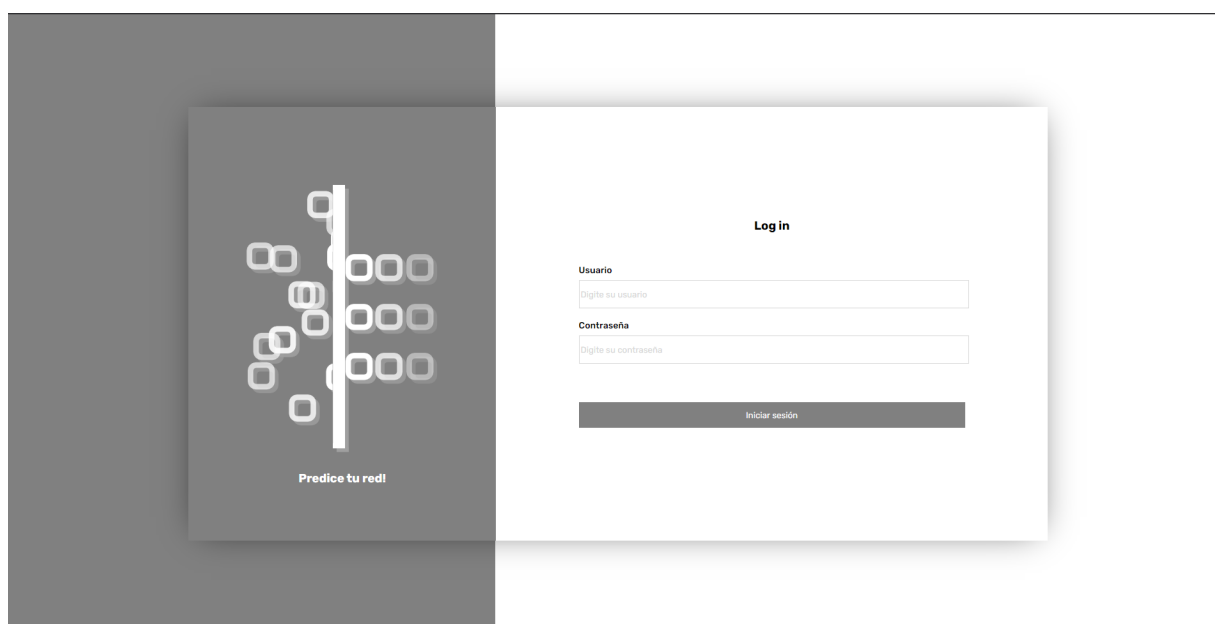


Figura 17. Captura de pantalla: login, Elaboración propia.

Esta es una interfaz de entrada y la página inicial del software, consta de un formulario sencillo en la parte central de la ventana con los campos Usuario y Contraseña, debajo de este se encuentra un botón de color gris con la palabra Iniciar Sesión y una línea de texto con la frase “Recuperar contraseña”, la ventana se compone de un fondo de color blanco, el nombre del software y su logotipo se encuentran en la parte izquierda de la ventana.

Si el usuario pulsa clic sobre el botón Entrar se validarán los campos del formulario y se enviará una petición a la base de datos que validará las credenciales de acceso, si estas son válidas el usuario será dirigido a la página principal.

Recuperar Contraseña

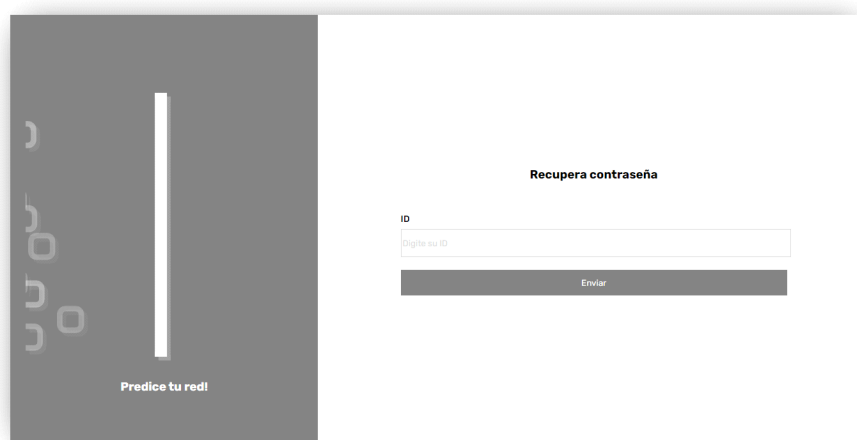


Figura 18. Captura de pantalla: recuperar contraseña, Elaboración propia.

Esta es una interfaz de entrada, consta de un formulario en la parte central con el campo correo electrónico, debajo de este se encuentra un botón de color gris con la frase “enviar” de color gris, la ventana se compone de un fondo de color blanco.

Página Carga de archivos



Figura 19. Captura de pantalla: cargar archivo, Elaboración propia.

Esta es una interfaz de entrada, se compone de un botón “Cargar archivo” que abrirá una ventana del explorador de Windows en la cual el usuario debe seleccionar el archivo con los datos de

entrenamiento para el modelo predictivo, si el archivo se carga correctamente, el usuario será dirigido a la página principal.

5.3.3. Interfaces de Salida

Página principal

Esta es una interfaz de salida, se compone de dos pestañas en la parte superior izquierda, una de ellas llamada Datos cargados y otra llamada datos predicción, que se pondrá en fondo azul cuando alguna de las dos este seleccionada, también encontramos un botón llamado Predecir de color gris que, al ser presionado hará que el modelo predictivo haga una predicción sobre todos los datos ingresados anteriormente y habilitará la pestaña predicción, esta acción puede tardar un poco.

Pestaña Datos cargados

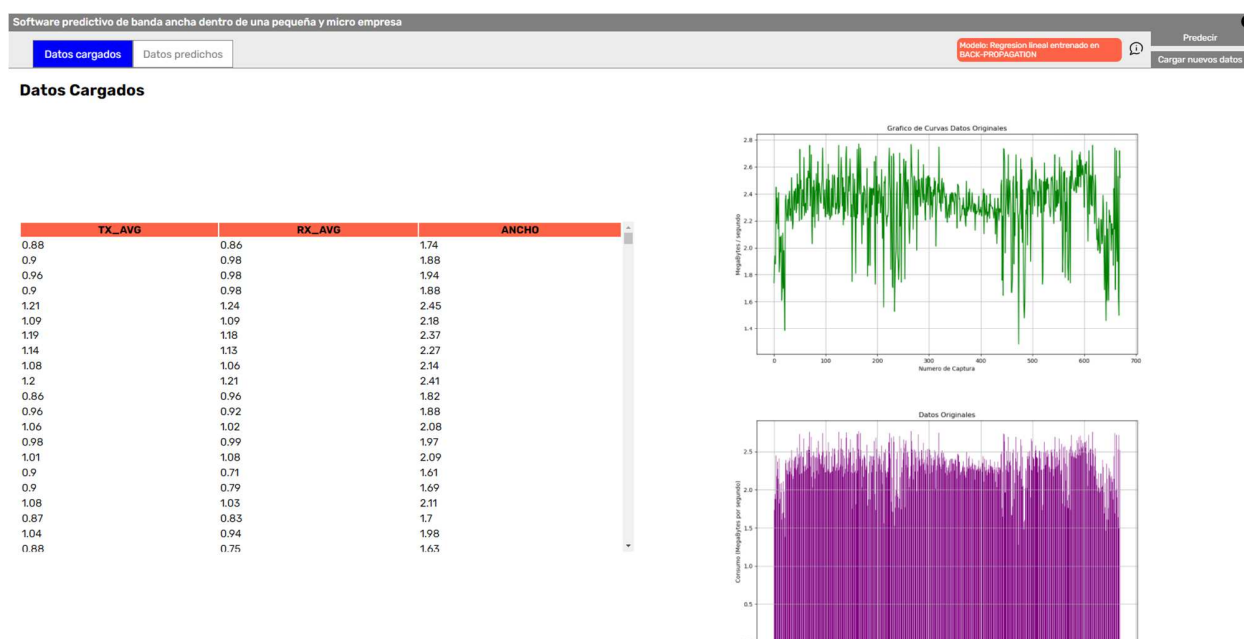


Figura 20. Captura de pantalla: pestaña datos cargados, Elaboración propia.

La pestaña datos se compone de una tabla en la parte central izquierda en la que se muestran los datos contenidos en el archivo de datos ingresado anteriormente, en la parte derecha se encuentran un gráfico en forma de histograma generado con los datos de la tabla y un gráfico en forma de polígono de frecuencia, al dar clic sobre uno de estos gráficos se mostrará una versión de este en pantalla completa y un botón que permite la descarga.

Pestaña Datos predichos

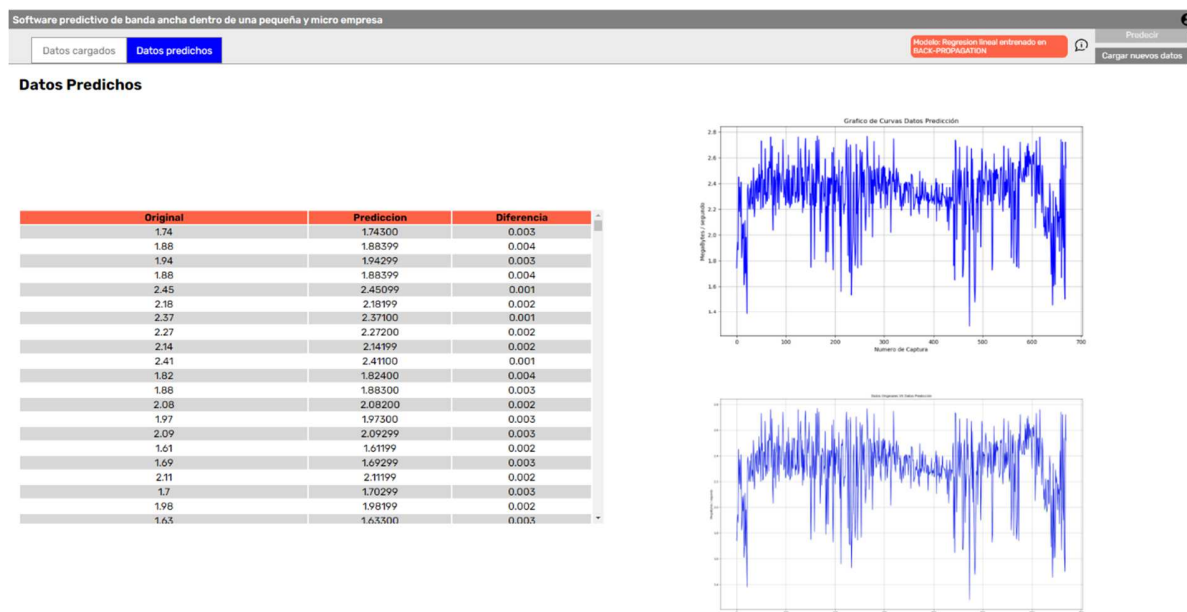


Figura 21. Captura de pantalla: pestaña datos predichos, Elaboración propia.

La pestaña predicción se compone de una tabla en la parte central izquierda en la que se muestran los datos generados por el modelo predictivo producto de hacer una predicción sobre el campo ancho de banda del archivo de datos de entrenamiento, en la parte derecha se encuentran un gráfico en forma de histograma generado con los datos de la tabla, un gráfico en forma de polígono de frecuencia, y un gráfico comparativo entre el polígono de frecuencias generado con los datos originales, versus los datos generados por el modelo predictivo, al dar clic sobre uno de estos gráficos se mostrará una versión de este en pantalla completa y un botón que permite la descarga.

6. Implementación

6.1. Plataformas de desarrollo

Para el desarrollo del proyecto se optó por el uso de un lenguaje dinámico y completo como lo es Python, específicamente en su versión 3.5.4, el cual además cuenta con varias librerías especializadas en el desarrollo de productos de machine learning, se decidió la utilización de las librerías Scikit y Tensorflow las cuales se especializan en el desarrollo de modelos de aprendizaje y. en específico cuentan con herramientas para la creación de redes neuronales basados en la librería Keras.

Para la codificación del lenguaje Python se descargó la herramienta Anaconda el cual es una interfaz gráfica de usuario que por defecto instala las librerías más utilizadas del lenguaje y varios de los IDE más utilizados para codificar lenguaje Python, se optó por la utilización de los notebooks de jupyter que permiten programar en bloques dentro de un mismo archivo Python.

Para la parte del servidor web se utilizó el lenguaje GO el cual permite crear el back-end para aplicaciones web de manera sencilla y potente, permitiendo el desarrollo de servicios rest que serán consumidos desde la parte front-end del producto de software

6.2 Base de datos

Una base de datos se entiende como un sistema de almacenaje de datos de manera electrónica, los datos se guardan de manera ordenada en campos, los campos forman registros, y los registros forman los archivos que son leídos por un sistema gestor de base de datos el cual puede generar información a partir de estos datos, un autor define las bases de datos como “Una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular” Valdez (2019).

Para el despliegue de la base de datos se utilizó el gestor de bases de datos SQLite puesto que permite crear un archivo local pequeño y que el uso de la base de datos solo estará limitado a validar las credenciales de sesión.

	ID	name	email	user	password
	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	user default	admin@admin.com	admin	123456

Figura 22. Base de datos, Elaboración propia.

6.3. Infraestructura de hardware y redes

En el desarrollo del proyecto fueron utilizados un computador de escritorio genérico y un portátil de la línea Asus, con las siguientes especificaciones técnicas.

Computador de escritorio genérico

CPU: Procesador Intel® Core™ i3-6100, caché de 3 M, 3,70 GHz

GPU: Gigabyte Ge Force® GTX 1060 WINDFORCE OC 3G

RAM: Kingston Hyper X Fury DDR4 2400 8 gigabytes

DISCO DURO: Toshiba DT01ACA100

TARJETA MADRE: Gigabyte G1 Sniper B7

Portátil Asus Vivobook X556UR

CPU: Procesador Intel® Core™ Procesador Intel® Core™ i5 6200U

GPU: NVIDIA Ge Force 930MX, 2GB VRAM

RAM: DDR4 2133MHz SDRAM

DISCO DURO: 1TB 5400RPM SATA HDD

SO: Windows 10 Home

7. Pruebas del software

7.1. Pruebas del software

Las pruebas de software son una pieza importante en la determinación de si se ha cumplido o no el alcance establecido al principio del desarrollo del proyecto, en palabras de Ecu Red (2019) “Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo”.

Se analizaron los resultados obtenidos del test de prueba realizado por usuarios, quienes calificaron y dieron su opinión sobre el producto de software de acuerdo a los requisitos funcionales y no funcionales planteados en este documento, se concluye que todos los requisitos fueron cumplidos pero que hay espacios de mejora en los apartados de login e información mostrada en forma de gráficos.



Figura 23. Resultados pruebas del software, Elaboración propia.

Los resultados obtenidos en las pruebas de software muestran que fueron cumplidos cada uno de los requisitos funcionales y no funcionales planteados a lo largo del desarrollo de este proyecto, con algunas observaciones y recomendaciones de mejora para perfeccionar el software.

Tabla 9.

Observaciones pruebas del software

Observaciones	Descripción
1	Deberían agregar una opción para registrarse
2	Quizás más gráficos deje más completo el programa
3	Bastante rápido sabiendo lo que hace

Nota: Elaboración propia.

Las observaciones sugieren la implementación de un formulario que permita a usuarios registrarse de manera general para poder entrar con una cuenta de uso personal, esta característica está pensada para una versión futura, otra de las recomendaciones sugiere la inclusión de una mayor

variedad de gráficos, lo cual fue una característica pensada en el desarrollo del software pero que no fue tomada en esta versión. Para una versión futura se planea investigar otro tipo de gráficos que permitan a los usuarios visualizar de otra manera útil la información retornada por el software.

7.2. Pruebas de usabilidad

Las pruebas de usabilidad son una herramienta que nos permite identificar el desempeño de nuestro producto de software en interés de buscar mejoras en la usabilidad y desempeño de la versión final, definida por la IDF (Interaction Design Foundation, 2019) como “la práctica de probar lo fácil que un diseño es usar en un grupo de usuarios representativos. Por lo general, implica observar a los usuarios cuando intentan completar las tareas y pueden realizarse para diferentes tipos de diseños, desde interfaces de usuario hasta productos físicos”.

Se analizarán los resultados obtenidos de los test de prueba realizadas por usuarios, quienes dieron su opinión y recomendación sobre algunas de las características del software.

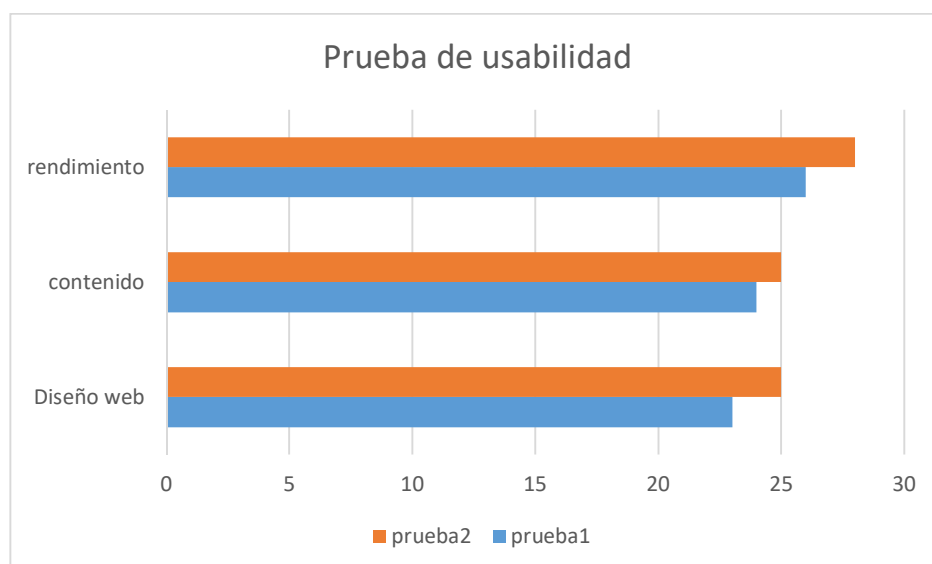


Figura 24. Resultado prueba de usabilidad, Elaboración propia.

Cada categoría cuenta con 30 puntos como límite máximo, los resultados obtenidos muestran que la categoría mejor calificada fue rendimiento, y la peor calificada fue diseño web, aunque el promedio de puntos es superior al 75% en todas las categorías lo que muestra un desarrollo del software estable, las dos características con los puntajes más bajos fueron:

La página muestra de manera visible el título del sitio, o de la sección donde se encuentra: 3 de 5 puntos.

Los colores y el diseño en general resultan atractivos: 3 de 5 puntos.

Tabla 10.

Comentarios prueba de usabilidad

Número	Descripción
1	El diseño de la página es muy bonito pero se ve un poco opaco, se vería mejor con colores más vivos.
2	El título no dice mucho sobre lo que hace el software

Nota: Elaboración propia.

Los comentarios de los usuarios sugieren la introducción de una interfaz más llamativa y brillante, esta característica podría ser analizada en una versión futura, el otro comentario del usuario sugiere que el título del software no es muy explícito, esto puede deberse a que el nombre utilizado para esta versión fue el mismo del título de proyecto, no se pensó en esta versión un nombre comercial o abreviatura.

Conclusiones

Se concluye que la interfaz web diseñada es intuitiva y que las funcionalidades del software son fáciles de entender, se evidencia posibilidades de mejora para una versión posterior.

Se concluye que el desarrollo de la primera versión del software culminó con éxito, se evidencia que el algoritmo de aprendizaje desarrollado cumple su propósito inicial de obtener una asertividad superior al 80% , este dato se obtiene al comparar los datos retornados por el modelo predictivo con los datos originales.

Se evidencia que la incorporación del lenguaje GO cumplió el requisito de conectar la interfaz web de la aplicación con el entorno virtual de Python donde funciona el modelo predictivo, concluimos que es posible hacer una predicción aproximada sobre el ancho de banda utilizado a partir de datos como la cantidad de datos entrantes y salientes dentro de un dispositivo.

Se comprueba que las peticiones solicitadas por la interfaz web funcionan correctamente dentro del servidor local y que el modelo predictivo recibe y envía los datos solicitados por este sin inconvenientes, los desarrolladores a cargo del proyecto evidenciamos un crecimiento personal y profesional al implementar un lenguaje de programación nuevo y aprender sobre la nueva tecnología de Machine Learning.

Recomendaciones

Se recomienda recopilar información sobre el consumo de banda ancha de un lapso de tiempo superior a 1 año con el fin de poder hacer ajustes más precisos al modelo predictivo y desarrollar una versión mucho más confiable.

Se recomienda probar el modelo con datos reales de varios micros empresas el fin de determinar el grado de escalabilidad y adaptabilidad del software.

Se estima que el modelo puede mejorar con la integración de entradas adicionales como el número de terminales en funcionamiento, días de la semana, calendario laboral, entre otros.

Referencias

- Alava, N. (2015). Modelos de procesos prescriptivos. [Imagen]. Ingeniería en software. Recuperado de <https://ingenieriaensoftwarenathalyalava.wordpress.com/2015/04/25/modelos-de-procesos-prescriptivos/>
- Bueno, O. (2017). Inteligencia artificial ¿Qué es y para qué puede servir el ‘machine learning’?. Retina. Recuperado de https://retina.elpais.com/retina/2017/10/19/innovacion/1508392516_816211.html
- Castells, M. (2001). Internet y la Sociedad Red. La factoría. Recuperado de <https://revistalafactoria.org/articulos/2018/6/4/internet-y-la-sociedad-red>
- Clouddradar. (2019). Clouddradar – Ejemplo de reporte. [Imagen]. Clouddradar. Recuperado de <https://get.clouddradar.io/es/server-monitoring#monitoringCards>
- Clouddradar. (2019). Clouddradar - Página de inicio. [Imagen]. Clouddradar. Recuperado de <https://get.clouddradar.io/es/server-monitoring>
- Clouddradar. (2019). Clouddradar – Precio. Clouddradar. Recuperado de <https://get.clouddradar.io/es/server-monitoring#pricing>
- Ecu-Red. (2019). Pruebas de software, Ecured. Recuperado de https://www.ecured.cu/Pruebas_de_software
- Espinoza, D. Rodríguez, C. (2020). *Desarrollo videojuego acerca de la vida de San Agustín*. (Trabajo de grado, Universitaria Agustiniiana), Bogotá, Colombia.
- IBM. (s.f.). Arquitecturas de tres niveles. [Imagen]. IBM. Recuperado de https://www.ibm.com/support/knowledgecenter/es/SS7K4U_9.0.5/com.ibm.websphere.zseries.doc/ae/covr_3-tier.html
- IBM. (s.f.). El modelo de redes neuronales. IBM. Recuperado de https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html
- IDF. (2019). what is Usability Testing?. Interaction Design Organization. Recuperado de <https://www.interaction-design.org/literature/topics/usability-testing>
- Izquierdo, R. (2019). Servidor caído: causas y perjuicios que puede ocasionar a una empresa. Pandorafms. Recuperado de <https://pandorafms.com/blog/es/servidor-caido>

- Jones. M. (2017). A neural networks deep dive. IBM. Recuperado de <https://developer.ibm.com/technologies/artificial-intelligence/articles/cc-cognitive-neural-networks-deep-dive>
- Lucidchart. (2019). Qué es un wireframe para un sitio web. Lucidchart. Recuperado de <https://www.lucidchart.com/pages/es/que-es-un-wireframe-para-un-sitio-web>
- Lucidchart. (2019). Tutorial de diagrama de clases UML. Lucidchart. Recuperado de <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>
- Nagios. (2019). Nagios Network Analyzer – Características. [Imagen]. Nagios. Recuperado de <https://www.nagios.com/products/nagios-network-analyzer/#features>
- Nagios. (2019). Nagios Network Analyzer – Página de inicio. [Imagen]. Nagios. Recuperado de <https://www.nagios.com/products/nagios-network-analyzer>
- Paessler. (2019). Paessler – Página de inicio. [Imagen]. Paessler. Recuperado de <https://www.es.paessler.com>
- Paessler. (2019). Precios. Paessler. Recuperado de <https://www.es.paessler.com/prtg/pricing>
- Paessler. (2019). Requisitos de sistema para PRTG. Paessler. Recuperado de <https://www.es.paessler.com/prtg/requirements>
- Paessler. (2019). Tabla de funciones del software PRTG. [Imagen]. Paessler. Recuperado de <https://www.es.paessler.com/prtg>
- Pfleeger, L. (2002). *Ingeniería de software: teoría y práctica*. (Traducción de: Elvira Quiroga.). México: Editorial Prentice hall.
- s.n. (2019). Los problemas más frecuentes de internet en las empresas. Quois magazine. Recuperado de <https://www.quis.com/blog/los-problemas-mas-frecuentes-internet-las-empresas>
- s.n. (s.f). Ciclo de vida de un proyecto XP. Oness-Sourceforge. Recuperado de <http://oness.sourceforge.net/proyecto/html/ch05s02.html>
- s.n. (s.f). Programación Extrema XP, Ingeniería de software. Recuperado de http://ingenieriadesoftware.mex.tl/52753_xp---extreme-programing.html
- s.n. (s.f.). Programación Extrema (XP). [Imagen]. Ingeniería de Soporte Lógico. Recuperado de <https://sites.google.com/site/ingsoportelogico/home/programacion-extrema-xp>
- Urías, E. (s.f.). Algoritmos de Aprendizaje Automático. Invid. Recuperado de <https://invidgroup.com/es/algoritmos-de-aprendizaje-automatico>

Anexos

Anexo 1: Acta desarrollo proyecto #1

Anexo 2: Acta desarrollo proyecto #2

Anexo 3: Acta desarrollo proyecto #3

Anexo 4: Acta desarrollo proyecto #4

Anexo 5: Acta desarrollo proyecto #5

Anexo 6: Acta desarrollo proyecto #6

Anexo 7: Acta desarrollo proyecto #7

Anexo 8: Acta desarrollo proyecto #8

Anexo 9: Acta desarrollo proyecto #9

Anexo 10: Prueba de software #1

Anexo 11: Prueba de software #2

Anexo 12: Prueba de usabilidad #1

Anexo 13: Prueba de usabilidad #2

Anexo 14: Guía de instalación