

PREDICCIÓN DE PROBLEMAS REFERENTES A TIC EN ORGANIZACIONES DE SERVICIOS GENERALES UTILIZANDO REDES NEURONALES

Prediction of problems related to TIC in general service organizations using a neural network

Christian Jesús Pinzón Casas¹, Adriana Marcela Vega Escobar²

¹Universidad Distrital Francisco José de Caldas, Estudiante Maestría en las Ciencias de Información y Comunicaciones, Grupo de Investigación GESETIC. ²Docente de la Universidad Distrital Francisco José de Caldas, Grupo de Investigación GESETIC. Email: ¹cjpinzonc@correo.udistrital.edu.co, ²avegae@udistrital.edu.co

(Recibido 23 de junio de 2022 y aceptado 15 de julio de 2022)

Resumen

Este artículo muestra la eficiencia de implementar una aplicación de mesa de ayuda con un modelo predictivo mediante redes neuronales para definir un catálogo de los servicios, con la ejecución de buenas prácticas para la gestión de servicios tecnológicos mediante un estándar de calidad para garantizar dichos servicios. Por ello, la empresa de servicios generales pretende resolver las solicitudes y requerimientos de los usuarios por medio de dicha aplicación, manteniendo centralizado el ciclo de vida de cada incidente. Este sistema de *Help Desk* se desarrolló mediante el lenguaje de programación PHP, con la estructura HTML5, sus estilos CSS por Bootstrap y con su motor de bases de datos MySQL. Con la creación de un modelo entidad – relación adecuada para las necesidades de la empresa, un módulo de gestión de tickets, módulo de usuarios, módulo de manuales y un módulo de análisis descriptivo de los tickets de la base.

Palabras clave: mesa de ayuda, ITIL, tecnologías de la información y comunicaciones (TIC), ISO 20000, PQRSD, redes neuronales.

Abstract

This article shows the efficiency of implementing a Help Desk Application with a predictive model for the definition of a catalog of services, with the execution of good practices for the management of technological services through a quality standard to guarantee these services. Based on that, the general services company attempts to resolve the requests and requirements of users through application, keeping the life cycle of each incident centralized. This Help Desk system was developed using the PHP programming language, with the HTML5 structure, its CSS styles defined by Bootstrap and its MySQL database engine. Thanks to the creation of an entity-relationship model suitable for the needs of the company, a ticket management module, user module, manual module and a descriptive analysis module of the tickets found in the base.

Key words: help desk, ITIL, information and communication technology (ICT), ISO 20000, HPCC, neural networks.

1. INTRODUCCIÓN

La propuesta se fundamenta en aplicar una metodología basada en *Help Desk* para la solución de los diferentes inconvenientes que se presentan en las Tecnologías de Información y Comunicaciones (TIC), al realizar una estructura que tome las mejores prácticas y procedimientos que definen las políticas de la Biblioteca de Infraestructura de Tecnologías de Información (ITIL) en su versión actualizada (V4) en conjunto con los procesos que se realizan en las Peticiones, Quejas, Reclamos, Sugerencias y Denuncias (PQRSD); basados en la normatividad ISO 20000 para que se brinde un servicio con altos estándares de calidad.

Lo anterior, con la finalidad de realizar un soporte de una manera organizada que permita la solución oportuna de los incidentes que se reportan mediante un proceso estandarizado, dando un diagnóstico asertivo con la ayuda de un modelo de análisis estadístico de la información que se recibe de las diferentes problemáticas reportadas, para la ayuda de la toma de decisiones [1] por parte del personal de sistemas, de manera que, se implemente la mejor solución a los incidentes que se les presenta a los usuarios.

Para ello, se pretende centralizar la información de las diferentes solicitudes realizadas por los usuarios para cuantificar y catalogar los inconvenientes presentados mediante un modelo de Procesamiento de Lenguaje Natural (PLN) mediante la librería spaCy [2]. De esta manera, permitirá mejorar los tiempos de respuesta y facilitar la gestión tanto para el área de sistemas, como centralizar las solicitudes que involucren otras áreas; al implementar consigo el proceso de PQRSD. En seguida, con la clasificación de eventos se procede a realizar un desarrollo de mesa de servicio conforme a los requerimientos funcionales y no funcionales enfocados a las necesidades de la empresa y al cumplimiento de los estándares de calidad basados en las buenas prácticas definidas.

Además, es importante llevar un seguimiento a las solicitudes que se reciben mediante un tablero de control que contenga el proceso realizado para su solución en donde se detalle el comportamiento de estos casos para proceder de manera óptima gracias a una medida preventiva de solución, la cual estará establecida en manuales de usuario, de tal manera que, se pueda autogestionar. Como consecuencia, la finalidad de contar con un repositorio de problemas solucionados permitirá la predicción de los mismos y así ofrecer un servicio adecuado para cada trámite.

2. METODOLOGÍA

La calidad de los Sistemas de Gestión del Servicio (SGS) es un factor que tiene gran importancia para atender los diferentes incidentes presentados tanto en las empresas de servicios generales como en las empresas en general. Para alcanzar dicha calidad se debe implementar los principios descritos en las ITIL, en su versión 4, con la normatividad ISO 20000:2018, de esta manera, se podrá crear una estructura con las mejores prácticas para la gestión de servicios de tecnologías de la información [3] todo esto alineado con los objetivos y metas de la organización.

La empresa utiliza herramientas para gestionar sus procesos con mayor eficiencia mediante las Tecnologías de la Información y Comunicaciones (TIC), como lo son: los servidores, una red interna con enrutamiento y control de firewall, estaciones de trabajo con su respectivo computador, impresoras, aplicaciones para el control de acceso, software especializado ERP para el manejo modular de las áreas de la empresa, etc. Es así que, estas herramientas deben tener un seguimiento para su funcionamiento adecuado, en otras palabras, se crea la necesidad de contar con un software capaz de manejar y gestionar los incidentes de una manera ágil [4], como resultado, se evitarán los tiempos muertos y la ineficiencia de los procesos, al no poder realizar/ejecutar las funciones que se deben realizar en el quehacer organizacional.

2. 1. Panorama Inicial

Es necesario establecer cuáles son los usuarios que van a interactuar con el sistema de mesa de servicio para la identificación de sus necesidades; el primer grupo se compone por los usuarios internos, que son el personal de todas las áreas articuladas al organigrama de la empresa; el otro grupo de usuarios, son los clientes y el personal operario encargado de realizar el aseo general que tiene la empresa, tal y como se refleja en la Tabla 1.

Tabla 1. Usuarios que componen el organigrama de la empresa.

| Área | Tipo_Usuario | #_Personas |
|---------------------------|--------------|------------|
| Auditoría | Interno | 2 |
| Ciente | Externo | 9 |
| Contabilidad | Interno | 6 |
| Contratación y Licitación | Interno | 3 |
| Gerencia | Interno | 2 |
| Inventario | Interno | 3 |
| Nómina | Interno | 4 |
| Operario | Externo | 25 |
| Recursos Humanos | Interno | 4 |
| Seguimiento | Interno | 5 |
| Sistemas | Interno | 2 |
| Tesorería | Interno | 4 |

Los usuarios realizan sus actividades utilizando las TIC implementadas en la empresa, con la finalidad de lograr más eficiencia en los procesos al realizar seguimiento de sus actividades, de esta manera, la gerencia controla el manejo de su negocio. No obstante, en algunas ocasiones, se pueden presentar inconvenientes por diferentes motivos que encierra el ambiente tecnológico y humano, lo que ocasiona incidentes en el área de sistemas, asunto que requiere una solución oportuna.

Gracias a la observación realizada en la organización, los incidentes no tenían ningún seguimiento, se notó que algunas veces, se daba solución yendo al lugar donde se encontraba el usuario. Con base en lo anteriormente planteado, para este estudio se tomarán los datos de los

incidentes reportados durante el año 2019, los cuales se consolidan y se les agrega características importantes que servirán como primer análisis de cómo se encuentra la empresa del servicio en la gestión de sus incidentes y qué tan oportuna es su respuesta frente al tiempo de respuesta [5].

En consecuencia, los datos muestran que hay un incremento entre el primer y segundo semestre de más del 40% de la demanda de las solicitudes realizadas por los usuarios con un tiempo promedio de respuesta a la solicitud de 3,7 días aproximadamente, una respuesta muy demorada en comparación de los estándares que manejan la mesa de ayuda *Help Desk*.

Ahora bien, en el segundo semestre se implementó un plan piloto para las PQRSD [6], donde se quería hacer un seguimiento sobre cómo se sentían sus clientes con respecto a la prestación del servicio de la empresa; de este modo, evaluar las posibilidades de mejora y dar el plus de la realimentación que hace el cliente con la compañía. Para ilustrar lo mencionado, se muestra el comportamiento de los incidentes en la Figura 1:

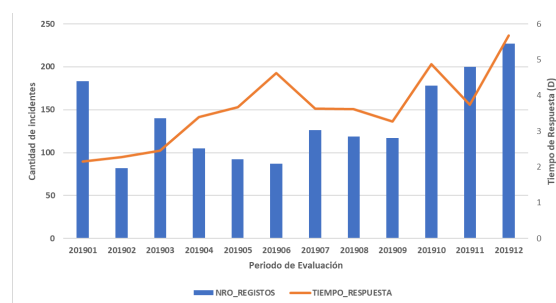


Figura 1. Incidentes reportados con sus tiempos de respuesta para el año 2019.

2. 2. Metodología de selección de buenas prácticas de gestión de servicios

En razón a la necesidad encontrada en la organización, se diagnóstica la implementación de un sistema de gestión de servicio para el trámite de los problemas que se presentan en la compañía, principalmente, aquellos orientados a las TIC. Con base en la historia de la gestión que tuvo el incidente, también se tienen en cuenta elementos como

la solución que se va a aplicar, la respuesta en el menor tiempo posible y la satisfacción de los usuarios frente a las problemáticas expuestas.

En este sentido, implementar un software capaz de gestionar el servicio de la administración de incidentes, permitirá tramitar adecuadamente el problema y llegar a la solución para evitar tiempos muertos para los usuarios que presentan dichos inconvenientes. Para ello, es necesario incorporar el ciclo de vida definido por ITIL enfocado a las necesidades de la compañía, a continuación, se exponen:

2. 2. 1. Estrategia

Para definir la estrategia, es necesario saber lo que espera y requiere la empresa para la implementación del *Help Desk* con la gestión de incidentes de sus usuarios; por esta razón, se identifican las necesidades mediante el levantamiento de requerimientos [7], al establecer un contrato de desarrollo para una aplicación a la medida, como se muestra en la Figura 2:



Figura 2. Fases de procesos de la ingeniería de requerimientos.

Además, se define el rol principal de seguimiento administrado por el área de sistemas; dicha área es la encargada de liderar el estado de las TIC de la empresa y corroborar que estas se encuentren en óptimas condiciones para prestar el servicio, con la creación de un control de gestión para realizar procesos de mantenimiento y evitar tiempos muertos prolongados para los usuarios.

2. 2. 2. Diseño

La creación del diseño de servicio define un catálogo que es el encargado de realizar la oferta a los usuarios internos y externos sobre lo que va a atender el software especializado para la gestión de incidentes; este catálogo se va a crear con base a los problemas reportados durante el año 2019 y de algunas proyecciones que se realizarán

en el modelo analítico. La mesa de ayuda implementará diferentes servicios para cumplir los estándares de calidad, según lo muestra la Figura 3:



Figura 3. Diseño del servicio que ofrece el aplicativo *Help Desk*.

Los valores de entrada son los incidentes que generan los usuarios, quienes deberán clasificar su problema con un catálogo establecido para etiquetar los tipos de servicios que ofrecerá el aplicativo para así mejorar los tiempos de respuesta gracias a la claridad de la solicitud que se va a recibir.

2. 2. 3. Transición

La aplicación *Help Desk* pretende mejorar los tiempos de respuesta un 40% más eficiente, con el seguimiento de inicio a fin cada solicitud, en este sentido, se involucran en él las áreas asociadas para cada incidente para que se guarde el ciclo de vida de estas solicitudes en el sistema y se pueda utilizar posteriormente para mejorar la precisión y personalizar las soluciones para la satisfacción de los usuarios internos y externos.

Estos usuarios deben tener una plataforma donde puedan acceder por cualquier medio y dispositivo dependiendo de su rol y la urgencia que se tiene para las solicitudes; de tal manera que, se mantenga la integridad de la aplicación [8] mediante la autenticación que permita relacionar el rol específico para acceder a las funciones relacionadas en dichos roles.

2. 2. 4. Operación

La mesa de ayuda heredará las solicitudes realizadas durante el 2019, ya previamente analizadas y estandarizadas con la nueva estructura. Con ello, el servicio que prestará el aplicativo se realizará en un host en la nube para que esté disponible para los usuarios las 24 horas.

La necesidad latente, propia de esta aplicación, radica en la disponibilidad [9], para ello el área de sistemas deberá estar monitoreando el sitio en la nube y mantener los servicios en estados óptimos para el trámite de la mesa de ayuda.

2. 2. 5. Mejora continua

El catálogo que entraría a clasificar los incidentes de los usuarios es definido mediante un análisis descriptivo que empieza a categorizar los problemas registrados en el año 2019; con esto, se refinan las etiquetas mediante el modelo de predicción que dejará en evidencia los atributos con más afinidad del catálogo de servicios.

Los incidentes reportados contienen una tendencia de crecimiento, debido a que la compañía ha estado incrementando sus clientes en el último semestre del año 2019 y se requiere con urgencia la gestión óptima de los problemas de los usuarios. Además, es necesario que la aplicación contenga la historia del trámite de la solución al inconveniente; este *back up*, permitirá calcular nuevos servicios del catálogo, de esta manera, se volverá a actualizar con el modelo predictivo que se desarrolló.

2. 3. Modelamiento predictivo de incidentes

El proceso de manejo de incidentes, descrito en el panorama inicial, ilustró que había una pérdida de información del trámite del inconveniente, esto impedía el mejoramiento de la gestión del servicio. Una vez se reconoció el problema, el área de sistemas procedía a dar solución rápidamente y no prestaba atención a la relación de los casos reportados; concurriendo en procesos repetitivos en solicitudes con características similares de solución.

Por lo anterior, se procedió al levantamiento de toda la información de las solicitudes realizadas por los usuarios durante el 2019 y se comenzó a hacer un análisis descriptivo de cómo se encontraba la compañía con respecto a los tramites realizados para la solución de problemas. Esto se refleja en la distribución de solicitudes (ver Figura 4).

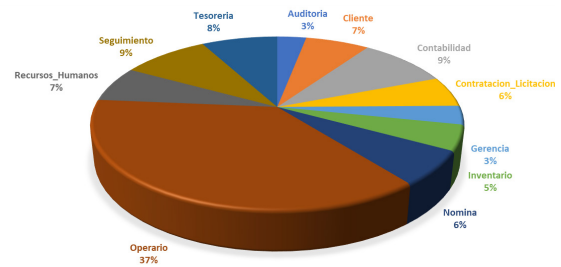


Figura 4. Distribución de solicitudes por áreas de la compañía.

La anterior figura ilustra que el área de Operarios fue la que realizó más solicitudes, debido a que son los empleados encargados de hacer labores de aseo directamente con los clientes. Al realizar esta prestación del servicio, se expone al empleado y este queda sin la atención directa o la solución a los posibles inconvenientes que se les pueda presentar. Con lo anterior, se puede observar que los operarios presentaban un problema en su ciclo de vida, ya que eran solicitudes que escalaban al área de Seguimiento sin que se hiciera control de la solución aplicada.

Otro elemento importante que debe tomarse en cuenta, son los tiempos de respuesta, considerados muy altos, de 3,7 días promedio por solicitud, dichos tiempos se encuentran fuera de los estándares de respuesta que se espera de una mesa de servicio [10]. Esto debido a que no se tiene implementado un atributo que permita medir el impacto que tiene esa solicitud con la continuidad del trabajo. Los tiempos se estipulan, según el estándar y la importancia que se tiene para el trámite de los incidentes, se definen en la Tabla 2.

Tabla 2. Tiempos de impactos para atender incidentes.

| Tiempo atención incidentes | | |
|----------------------------|--------------|------------|
| Impacto | 1° Respuesta | Max Tiempo |
| Alto | 2H | 24H |
| Medio | 4H | 60H |
| Bajo | 4H | 96H |

En primer lugar, el impacto alto es aquel incidente que compromete TIC críticos y hace que las actividades en varias áreas sean interrumpidas por este, con una probabilidad de pérdida de información; En segundo lugar, el impacto medio es aquel que se compromete TIC que hacen que una o varias personas en algún área se encuentren comprometidas en la realización de sus actividades. Por último, el impacto bajo, compete a aquellos incidentes que compromete TIC no críticos hasta máximo un área donde evita la funcionalidad de un proceso interno de información.

Para la creación del catálogo de servicios que va a prestar la mesa de ayuda, se utilizará el Procesamiento de Lenguaje Natural (PNL), la cual hace parte del campo de conocimiento de la inteligencia artificial para predecir el tipo de inconveniente de los incidentes reportados en el año 2019 y así categorizarlos para que los usuarios puedan especificar el problema presentando; en consecuencia, se puede mejorar los pasos a seguir en la solución y mejora de los tiempos de respuesta.

Se utilizó la arquitectura de la librería open-source spaCy, desarrollada en un paquete de Python para el PNL [11], ya que proporciona funciones con modelos pre-entrenados que permite realizar el procesamiento del lenguaje mediante diferentes técnicas a través de la inteligencia artificial, como lo son las redes neuronales.

Para ello, esta librería interpreta un texto dependiendo del lenguaje seleccionado que en este caso es español con la referencia es_core_news_lg [12] que son vectores de palabras con similitud y contiene grandes dimensiones de llaves y vectores únicos.

El primer paso es generar el texto para integrar al tubo de procesamiento de PNL, allí se debe seleccionar la información que se va a ingresar para empezar, en la base de consolidación de incidentes se carga en un DataFrame, utilizando las columnas CONSECUTIVO que permitirá identificar a que incidente hace referencia, con la combinación de campos OBSERVACIONES_ENCARGADO que contiene el dictamen del problema que reportó el

usuario y SOLUCION_APLICADA que hace referencia al proceso ejecutado para solucionar el problema.

El otro texto con el cual se quiere comparar es el borrador del catálogo que se preseleccionó analizando los casos que se encontraban en los consecutivos y se creó 20 categorías con 79 subcategorías, esto para realizar el primer análisis de los incidentes con respecto a este catálogo. Donde se realizó una asociación de palabras propias que caracterizan cada subcategoría; teniendo en cuenta la composición de la funcionalidad que contiene la tubería descrita en la Figura 5.

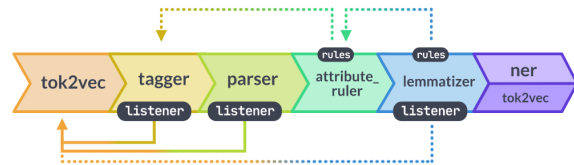


Figura 5. Diseño tubería CNN/CPU [13].

El primer paso que se realiza es tokenizar el texto, es decir, es la función de la tubería tok2vec capaz transformar en vectores cada una de las palabras para crear un etiquetado que contenga el texto y agregue atributos propios de cada palabra para poderlo identificar; este proceso lo realiza la funcionalidad que tiene la tubería de reglas de atributos, el cual describe cada una de las palabras del vector si es un adjetivo, adverbio, pronombre, verbo, sustantivo, conector, etc [14]. Este proceso de tokenizar se deberá aplicar para el texto creado de los incidentes y el texto creado con el catálogo propuesto.

El siguiente paso es uno de los más importantes, luego de realizar una limpieza en los vectores, donde las palabras irrelevantes se van a eliminar como lo son los conectores utilizando la función de attribute_ruler; también se debe contar con los adverbios, términos numerales, eliminar signos de puntuación y listas de exclusión. Además, volver las palabras mayúsculas y aplicar la derivación regresiva, que hace referencia a la lematización de las palabras, es decir, dejándolas sin conjugar.

Después de seleccionar el método para realizar la comparación y evaluación de los textos para ver la

relación entre los incidentes reportados con el catálogo que se propone; el cual es la funcionalidad *similarity* que calcula la afinidad de un texto con el otro, comparando los documentos, intervalos y tokens que resultaron del tubo de procesamiento, ponderándolo de un número del 0 al 1, donde el 1 equivale el 100% de afinidad entre ambos textos.

Además, se utiliza incrustaciones de palabras con significados multidimensionales de una palabra generados por el algoritmo word2vec, el cual representa un grupo de modelos para producir *Word embeddings*, el cual vectoriza una palabra guardando información semántica de esta, este proceso permite realizar la asociación o disociación entre otros vectores dependiendo de sus contextos gramaticales; dichos vectores tienen como característica una longitud y una orientación, ubicados en planos multidimensionales [15].

Con ello, son ingresados para su entrenamiento en diferentes redes neuronales para su aprendizaje con asociación de relaciones complejas entre las palabras por conocer entre ellas su semántica; utilizando para estas asociaciones, un corpus de texto pre-entrenado con aproximaciones de verosimilitud logarítmica condicional [16].

A continuación, el paso a seguir, es unir la información generada en una tabla para poder seleccionar los cinco casos más propensos a asignarse a una subcategoría, para analizarlos y poderlos perfilar a la categoría que es la adecuada [17]. Además, se evidencia si se requiere agregar o eliminar etiquetas en el catálogo para refinar el resultado esperado.

2. 4. Prototipo de aplicativo Help Desk

La importancia que tiene un *Help Desk* para una empresa es que centraliza todo el proceso de gestión de los incidentes, para mantener un control de la calidad de servicio con el fin de satisfacer de manera óptima la solución de los problemas con respuestas ágiles para que los usuarios retomen sus actividades en el menor tiempo posible. Con esto, es importante mantener el desarrollo

con un diseño estructurado para que este no haga perder el servicio de calidad que se espera.

Con lo anterior, se realizó un prototipo de software para la gestión de incidentes y documentación de manuales para soporte. Accesible a través de internet y con diseño para móviles.

2. 4. 1. Requerimientos funcionales y no funcionales del sistema

Los Requerimientos Funcionales (RF) son aquellas descripciones explícitas del comportamiento, las cuales deben tener una solución de software, además, el conocimiento de la información que debe manejar según la necesidad planteada por el cliente [18]. De otro lado, los requerimientos no funcionales (RNF) son aquellas propiedades que va a tener el sistema, como lo son la disponibilidad y el rendimiento; realizando este proceso según las necesidades de la empresa, se genera las siguientes funcionalidades:

- RF01: Disponibilidad de servidor para alojar contenedores Docker. Prioridad- primario.
- RF02: Garantizar el acceso a la página desde cualquier dispositivo (Responsive Design). Prioridad – primario.
- RF03: Permitir la consulta de manuales de usuario en cualquier momento. Prioridad – primario.
- RF04: Asegurar la integridad de los incidentes abiertos, así como su gestión en todas sus etapas. Prioridad – primario.
- RF05: Garantizar la seguridad de las sesiones de los usuarios, así como el cifrado de sus datos sensibles. Prioridad – primario.
- RF06: Garantizar la escalabilidad del programa (tanto horizontal como vertical) en caso de ser requerido. Prioridad – primario.
- RNF01: Elaborar una interfaz amigable al usuario cumpliendo requisitos de UI/UX. Prioridad – secundario.
- RNF02: Mantener la imagen institucional en el diseño del programa. Prioridad – secundario.

2. 4. 2. Diagrama general de caso de uso

De acuerdo con las definiciones y especificaciones de los requerimientos, el paso a seguir es reflejar estas funcionalidades requeridas mediante los casos de uso con la interacción que se va a tener con los usuarios y el sistema en general, como se observa en la Figura 6.

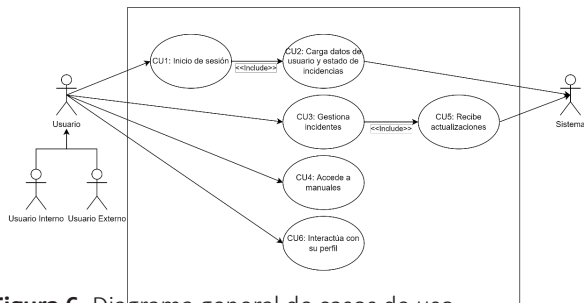


Figura 6. Diagrama general de casos de uso.

2. 4. 3. Modelo de Persistencia

Para la definición del modelo de persistencia se tuvieron en cuenta criterios de *Open Source* [19] y con una documentación de implementación ágil; dicho sistema de gestión de base de datos fue el motor MySQL. En la Figura 7 se puede observar el modelo entidad relación utilizado para la base de datos.

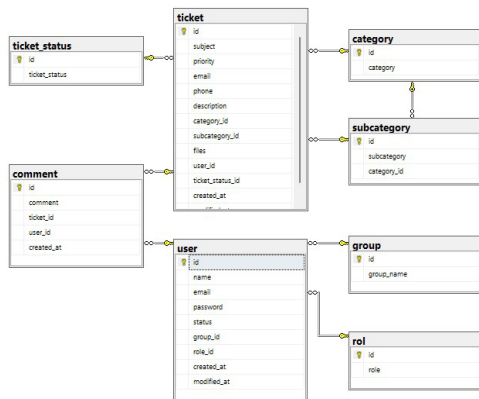


Figura 7. Diagrama entidad relación para el modelo de persistencia.

2. 4. 4. Arquitectura del Help Desk

Como se requiere que la aplicación este accesible a través de internet, se implementó una solución basada en servicios distribuidos en contenedores usando Docker; teniendo como sistema operativo base Ubuntu 20.04

(Focal Fossa) alojado en un servidor virtual Droplet de *Digital Ocean* [20].

Para mejorar su rendimiento, se utiliza un balanceador para optimizar el consumo de red, así como, para controlar las peticiones y configuración de servicio para denegar o permitir accesos. Como gestor de contenedores se implementa *Docker-compose* [21] para su redistribución y escalamiento.

Además, se separan los servicios de la APP (PHP/Apache), del SGBD (MySQL) y del administrador de base de datos (PHPMyAdmin) en los puertos 80, 3306 y 8000 respectivamente.

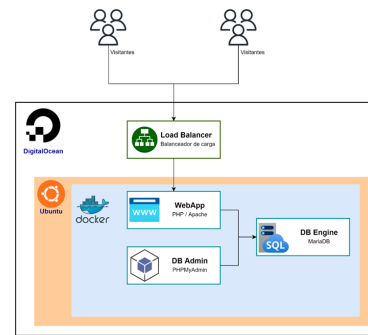


Figura 8. Arquitectura de implementación.

2. 4. 5. Implementación del Software - GUI

La interfaz de usuario (GUI) utilizado en la aplicación está definido con la utilización de estilos definidos en el CSS que nos ofrece el *framework Bootstrap* en su versión 5 como diseño y Javascript VanillaJS basado en ecmaScript en su versión 6 para interacción con el usuario. La plataforma está diseñada bajo el concepto *mobile-first*, optimizando la experiencia en dispositivos móviles.

2. 4. 6. Desarrollo de Software

El lenguaje de programación utilizada para el desarrollo *backend* de la mesa de ayuda fue PHP en su versión 7.4, por su versatilidad, orientado a aplicaciones web y el uso del paradigma de Programación Orientado a Objetos (POO). Además, seleccionar el patrón de arquitectura MVC (Modelo Vista Controlador) y TDD (*Test Driven Development*). Con el fin de facilitar el manejo de pruebas

unitarias, despliegues continuos y escalabilidad para nuevas funciones.

El diagrama UML del MVC de la aplicación, como se muestra en la Figura 9 se compone de las siguientes partes:

- Clase Database: Crea la conexión a la base de datos (indiferente al motor de base de datos a través del parámetro engine).
- Clase Modelo: Recibe los parámetros de consulta a través de la implementación de PDO de PHP en forma nativa, usando los métodos Query y Prepare para ejecución directa y ejecución con sentencias preparadas a fin de evitar problemas de inyección SQL y rendimiento.
- Clase Controlador:
 - Recibe parámetros desde la URL para asignar a cada uno de los controladores del módulo, carga el modelo de cada módulo para su interacción con la vista y el controlador.
 - Valida los datos enviados a través de POST y GET, así como implementación de archivos a través de \$_FILES.
 - Gestiona la sesión del usuario a través de sesiones del sistema con \$_SESSION.
- Clase Vista: Muestra la información proveniente del controlador, así como genera los formularios necesarios para la interacción con el modelo.

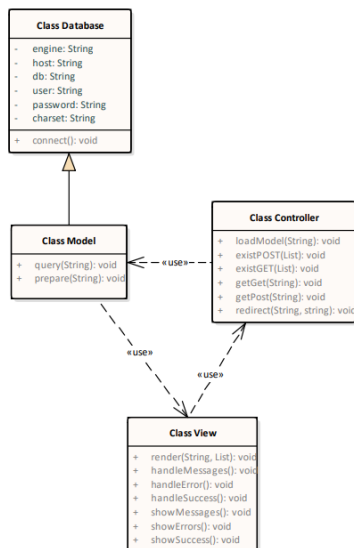


Figura 9. Diagrama UML de APP bajo MVC.

La clase de usuario, la cual se evidencia en la Figura 10, extiende de la clase modelo para usar sus métodos implementando una interface IModel para usar los métodos *Save*, *Get*, *GetAll*, *Delete* con el fin de asegurar la integridad del módulo Usuario. La clase modelo implementa a su vez los métodos de la clase nativa de PHP PDO para lenguaje estructurado en consultas a bases de datos.

```

class UserModel extends Model implements IModel{

    private $id;
    private $employeeId;
    private $name;
    private $email;
    private $password;
    private $status;
    private $groupId;
    private $groupName;
    private $roleId;
    private $roleName;
    private $createdAt;
    private $modifiedAt;

    public function __construct(){
        parent::__construct();
        $this->employeeId=0;
        $this->name='';
        $this->email='';

        $this->password='';
        $this->status=false;
        $this->groupId=0;
        $this->roleId=0;
        $this->createdAt='';
        $this->modifiedAt='';
    }
}
    
```

Figura 10. Clase de Usuarios con implementación Constructor POO PHP.

2. 4. 7. Control de Versiones

Para el Sistema de Control de Versiones (VCS) se implementa GIT bajo la plataforma Gitlab, por su concepto de eficiencia y la confiabilidad, utilizando *gitflow* para el correcto manejo de las ramas dentro del repositorio. Se usó la rama *develop* para salida a master una vez aprobados los PR de los *features* realizados; todo esto basado en entregables asociados por *Sprint*. La funcionalidad de las ramas se representa en la Figura 11 y se encargan de lo siguiente:

- Rama *master*: una de las ramas principales y obligatorias, son las versiones estables del desarrollo, donde mediante comando *commit* realizada en la rama *develop*, se

procederá a subir a producción.

- Rama *hotfix*: rama secundaria o de función que no es obligatoria, basada en la rama *master* que se encarga de reparar o corregir funcionalidades de las versiones publicadas. Cuando cumple su objetivo se fusiona con la rama *master*.

- Rama *releases*: rama secundaria o de función que no es obligatorio, basada en la rama *develop*, donde contiene el código desarrollado que se encuentra estable y está próximo a salir en algún despliegue a producción. Cuando cumple su objetivo se fusiona con la rama *master* y *develop*.

- Rama *develop*: una de las ramas principales y obligatorias, contiene el histórico de los desarrollos realizados. Creada a partir de la rama *master*.

Rama *feature*: rama secundaria o de función que no es obligatoria, basada en la rama *develop*, donde se encuentra nuevas características de desarrollo. Cuando cumple su objetivo se fusiona con la rama *develop*.

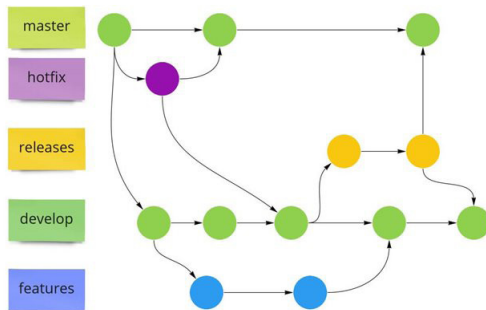


Figura 11. Ramas implementadas durante el desarrollo de un proyecto [22].

3. INTERPRETACIÓN DE RESULTADOS

La importancia de un control en el servicio de la mesa de ayuda, es fundamental para alcanzar objetivos de eficiencia que espera la empresa; con la implementación de un software que se encargue de inicio a fin la gestión realizada por los incidentes reportados de los usuarios, permitiendo mejorar el tiempo y el tipo de solución de estos.

En primera instancia, el modelo predictivo permitió generar el catálogo para la organización de incidentes,

mostrando en sus análisis ver la tendencia de los problemas que presentan los usuarios. Dicho modelo será más efectivo si se tiene la historia de la gestión y muchos más casos guardados en el repositorio de datos. El modelo arrojó la afinidad que se tenía en cada subcategoría, se emuló varias veces y se acortó a 18 categorías y 69 subcategorías, que son las más relevantes.

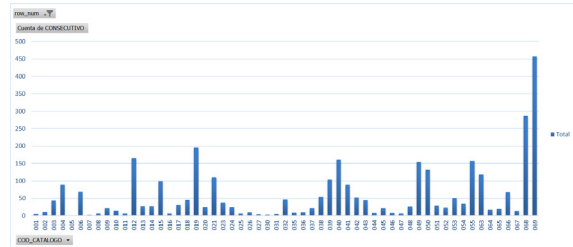


Figura 12. Distribución de similitud de los incidentes con el catálogo refinado.

Además, gracias a la implementación del aplicativo y la asignación de impactos en los tiempos de respuesta, se espera reducir del 3,7 a 2,2 ~ 2 días, teniendo presente que el 21% de los incidentes se demoraron su trámite entre 6 o más días (con base a la tabla impacto). También por la accesibilidad que tiene la aplicación, que independientemente donde se encuentre (usuarios externos) se puede acceder por medio de un dispositivo móvil sin perder tiempo hasta tener a la mano un ordenador.

Con la definición de los servicios que se van a presentar, se procede a realizar una aplicación de mesa de ayuda enfocada a la calidad de servicio con regulación ITIL y control con la ISO 20000 para la estructuración de los módulos con la gestión adecuada para la demanda de la empresa; centralizando la gestión de dichos servicios con el soporte detallado de cada una de las categorías, con ello, establecer unos manuales de usuario capaz de describir algunos servicios para que haya una posibilidad de que el mismo pueda solucionar su inconveniente y descongestionar el área de sistemas.

Con la implementación del control de versiones se evidenció entregas conforme con los tiempos estipulados, los cuales fueron seis (6), manejando en el desarrollo

las ramas de main, develop y releases. En cada entrega se pretende proporcionar cada uno de los módulos desarrollados, según los *sprint*.

A partir de los *sprint* se tuvo interacciones de aproximadamente 2 semanas, basándose en su metodología y demanda de la empresa. En total fueron 4 *sprint*, teniendo en cuenta que para los despliegues continuos no se usó herramientas de automatización CI/CD como Jenkins por temas de tiempo y programadores asignados, todo basado en flujo de despliegue que se muestra en la Figura 13.

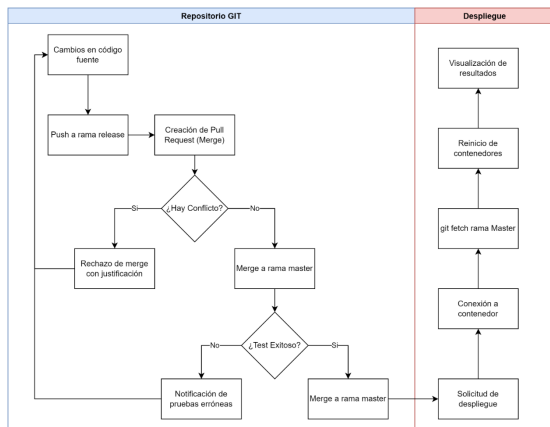


Figura 13. Modelo de despliegue continuo a través de git.

4. CONCLUSIONES

El cambio de las condiciones y control de gestión de servicio, son elementos de gran impacto para las mediciones de calidad que espera el área de sistemas y la empresa en general. En donde se le dio importancia al servicio cuando los tiempos de respuesta empezaron a aumentar por la falta de control, como se evidenció en el análisis de levantamiento y perfilación del catálogo.

Al implementar un aplicativo que contiene en su estructura un control de gestión de los incidentes, permite que se instaure una administración de calidad de los trámites que se tienen pendientes para no sobrepasar los tiempos límite que se establecieron con base en los requerimientos funcionales.

Con la organización de la información que se realizó conforme a las necesidades del modelo, se evidenció falencias en toda la gestión de los incidentes, de lo cual no se tenía un control específico en el ciclo de vida, en alguno de sus casos. Dicha situación ocasionó que se perdiera la gestión en alguna parte del ciclo de vida del incidente. Para ello, el modelo entró a organizar todos estos incidentes con la propuesta de un catálogo de servicios que se encargaría de organizar mediante una predicción del problema planteado con la solución que se espera.

Las métricas levantadas en la ejecución del todo el proceso que contiene el Procesamiento de Lenguaje Natural donde se evidencia que el consumo de recursos informativos (16 GB RAM, 512 GB SSD, Intel i7-9750H 2.60 GHz) es realmente demandante, teniendo presente la cantidad de información procesada se demora en promedio 20 min en su ejecución, donde el 86,8% de este tiempo únicamente se utiliza para la preparación y ejecución de la red neuronal.

La combinación del control de calidad de servicio, implementación de una estructura enfocada al servicio, el modelo predictivo para establecer los servicios prestados y la aplicación que permite la gestión de los incidentes de los usuarios permitirá organizar tanto el área de sistemas como las áreas involucradas para que tengan una solución basada en estándares de calidad, para solventar los problemas y evitar tiempos muertos en la empresa.

5. AGRADECIMIENTOS

Los autores desean agradecer a la Universidad Distrital Francisco José de Caldas y al Grupo de Investigación GESETIC, que hicieron posible este estudio y, consigo, llegar al resultado para mejorar un proceso tan importante para la empresa.

REFERENCIAS

- [1] G. Villegas, "Modelo estadístico pedagógico para la toma de decisiones administrativas y académicas con impacto en el mejoramiento continuo del rendimiento de los estudiantes universitarios, basado en los métodos de selección CUR", Tesis doctoral, Departamento de Estadística, Universidad de Salamanca, Salamanca, España, 2018.
- [2] M. Honnibal (2022, June 10). Library Architecture - Processing pipeline [Online]. Available: <https://spacy.io/api>.
- [3] J. Rubio & M. Arcilla, "How to Optimize the Implementation of ITIL through a Process Ordering Algorithm", MDPI, *Applied Sciences*, 10(1), 34, pp. 1-4, December, 2019. <https://doi.org/10.3390/app10010034>
- [4] G. Garzón, J. Merchán y K. Morea, "Implementación de buenas prácticas basadas en ITIL 4 e ISO 20000 para la gestión de incidentes y reducción de riesgos del service desk de la empresa INGEAL S.A", Tesis pregrado, Facultad de Ingeniería, Universidad Cooperativa de Colombia, Bogotá, Colombia, 2020.
- [5] F. AL-Hawari & H. Barham, "A machine learning based help desk system for IT service management", *Journal of King Saud University - Computer and Information Sciences*, Vol. 33 (6), July 2021. <https://doi.org/10.1016/j.jksuci.2019.04.001>
- [6] A. Vargas y C. González, "Diseño del módulo de PQRS para la recepción de inquietudes del portal orientación de la Empresa Universia Colombia", Tesis pregrado, Facultad de Ingeniería, Universidad Católica de Colombia, Bogotá, Colombia, 2021.
- [7] A. Bonilla, L. Cortés y M. Cortés, "Diseño y elaboración de una Guía Metodológica para el levantamiento de requerimientos en proyectos de implementación de Sistemas de Gestión Financiera para empresas en Colombia", Tesis maestría, Escuela Colombiana de Ingeniería Julio Garavito, Bogotá, Colombia, 2022.
- [8] F. Novello y P. Forcart, "Implementación de Mejores Prácticas ITIL", Tesis Licenciatura, Universidad Nacional del Noroeste de la Provincia de Buenos Aires, Buenos Aires, Argentina, 2020.
- [9] S. Chisco, J. Gutiérrez, N. Guzmán y C. Santiago, "Gestión de Servicios de Tecnología de Información usando ITIL en MYPIME", 16th LACCEI International Multi-Conference for Engineering, Education, and Technology, Lima, Perú, 2018, p. 11.
- [10] S. Becerra, "Implementación del proceso de gestión de incidentes y su influencia en el servicio de atención y soporte de tecnología de información en el área de help desk del proyecto gold fields salares norte, 2018", Universidad Nacional de Cajamarca, Escuela de Posgrado, Perú, 2021.
- [11] N. Colic & F. Rinaldi, "Improving spaCy dependency annotation and PoS tagging web service using independent NER services", *Genomics & Informatics*, 17(2): e21, 2019. <https://doi.org/10.5808/GI.2019.17.2.e21>
- [12] D. Sammy, "Reconocimiento y clasificación de entidades nombradas en textos legales en español", Sociedad Española para el Procesamiento del Lenguaje Natural, España, 2021.
- [13] M. Honnibal (2022, June 09). Industrial- Strength Natural Language Processing [Online]. Available: <https://swatimeena989.medium.com/training-word2vec-using-gensim-14433890e8e4>.
- [14] A. Fardeen (2021, Agosto 13), Tutorial on Spacy Part of Speech (POS) Tagging [Online], Machine Learning Knowledge – MLK, Available: <https://machinelearningknowledge.ai/tutorial-on-spacy-part-of-speech-pos-tagging/>.
- [15] Enzyme advising group (2019, agosto 16), Natural Language Processing: ¿cómo es la técnica Word Embeddings? [Online], Available: <https://blog.enzymeadvisinggroup.com/natural-language-processing>.
- [16] M. Gregoryev, "Generación de textos en ruso mediante técnicas de Aprendizaje Automático para la industria del lenguaje", Tesis pregrado, Universitat Politècnica de València, 2022
- [17] B. Jang, I. Kim and J. W. Kim, "Word2vec convolutional neural networks for classification of news articles and tweets". *PloS one*, 2019.

- [18] IIBA, "A Guide to the Business Analysis Body of Knowledge (BABOK)", Vol. 3, International Institute of Business Analysis, 2015.
- [19] Red Hat (2019, October 24), ¿Qué es el open source?, [Online], Available: <https://www.redhat.com/es/topics/open-source/what-is-open-source>
- [20] DigitalOcean (2022, June 14), Simpler cloud, Happier devs, Better results [Online], Available: <https://www.digitalocean.com/>.
- [21] Docker docs (2022, June 10), Overview of Docker Compose [Online], Available: <https://docs.docker.com/compose/>.
- [22] P. Borja (2021, abril 30), Cinco Git Workflows para mejorar nuestros proyectos, BabelGroup [Online], Available: <https://www.babelgroup.com/es/Media/Blog/Abril-2021/Cinco-Git-Workflows-para-mejores-proyectos>.