

# Sistema informático para el seguimiento y control del mantenimiento de infraestructura y equipo



## Colaboración

Rogelio Ramírez Silva; Francisco Javier Gutiérrez Mata; Alma Delia de Jesús Islao; Eloy Cadena Mendoza, Tecnológico Nacional de México / IT de Acapulco

Fecha de recepción: 15 de junio de 2020

Fecha de aceptación: 28 de abril del 2021

**RESUMEN:** El objeto del presente artículo es desarrollar la herramienta informática SISCMIE, la cual es un sistema informático web destinado al control y seguimiento del mantenimiento de infraestructura y equipo en el Instituto Tecnológico de San Marcos (ITSM), para su construcción se hizo uso del marco de trabajo Scrum, así como el uso de una metodología propia del desarrollo de software llamada OOHDM la cual dicta los tipos de diagramas utilizados para el diseño y desarrollo del software. La implementación de este sistema permitirá centralizar la información, así también, un mayor control de la información al que actualmente tienen al realizar dicho proceso en documentos Word y hojas de Excel.

Se hace mención a que este artículo forma parte de una serie de trabajos siendo el cuarto y último de estos, el cual da seguimiento a un artículo previo titulado: Implementación del backend de una aplicación web para el mantenimiento de infraestructura y equipo en el ITSM, escrito por los mismos autores y que fue aceptado en el séptimo congreso internacional de robótica y computación del Instituto Tecnológico de La Paz.

**PALABRAS CLAVE:** Front-End, NodeJs, Metodología Scrum, Back-End, Angular.

**ABSTRACT:** The purpose of this article is to develop the SISCMIE computer tool, which is a web computer system designed to control and monitor the maintenance of infrastructure and equipment at the Technological Institute of San Marcos (ITSM), for its construction the framework of Scrum work, as well as the use of a methodology of software development called OOHDM which dictates the types of diagrams used for the design and development of the software. The implementation of this system will allow to centralize the information, as well as a greater control of the information than they currently have when carrying out this process in Word documents and Excel sheets.

It is mentioned that this article is part of a series of works being the fourth and last of these, which follows a previous article entitled: Implementation of the backend of a web application for the maintenance of infrastructure and equipment in the ITSM, written by the same authors and which was accepted at the seventh international robotics and computing congress of the Instituto Tecnológico de La Paz.

**KEYWORDS:** Front-end, NodeJs, Scrum Methodology, Back-End, Angular.

## INTRODUCCIÓN

En el Instituto Tecnológico de San Marcos (ITSM) existen dos departamentos los cuales son Centro de Cómputo y Recursos Materiales y Servicios, dichos departamentos se encargan de dar mantenimiento y soporte técnico a los equipos de aire acondicionado, equipos de cómputo, impresoras, redes eléctricas, instalaciones de equipos multimedia, también se encargan de hacer trabajos de herrería, carpintería, plomería, manteni-

miento de aulas y muchos otros trabajos que son necesarios realizar para que las áreas del tecnológico lleven a cabo sus actividades de una manera eficiente.

En el ITSM existe un procedimiento que se lleva a cabo cuando se solicita el mantenimiento de una infraestructura o equipo en la institución, dicho procedimiento consta de una serie de pasos que llevan a cabo los jefes de departamento que brindan el servicio. Dependiendo de la actividad a realizar son los pasos que se realizan, las actividades que dicta dicho procedimiento son, la verificación de infraestructura y equipo la cual consta de revisar las anomalías (daños) de todos los edificios (infraestructuras) y equipos de cómputo, equipos eléctricos, aires acondicionados y demás equipos con los que cuenta la institución y que requieran mantenimiento. Otra de las tareas que se realizan una vez que se hizo la verificación, es programar (plan de trabajo) los días en que se resolverá la anomalía encontrada si la hubiera y asignársela a un trabajador para que realice dicho trabajo en la fecha y departamento programado.

Otra actividad que dicta el procedimiento es la atención de solicitudes de mantenimiento correctivo, para la cual son los mismos pasos que se realizan para una verificación, los cuales son, reportar la anomalía al departamento correspondiente y el departamento que atenderá la anomalía programará la fecha y asignará el trabajo a la persona que resolverá el problema reportado. Es importante mencionar que el ITSM es una institución que consta tan solo de dos edificios y, que por tal motivo solo realizan la verificación de infraestructuras y realización de plan de trabajo.

## OBJETIVO GENERAL

El objetivo principal es desarrollar un sistema informático web para control y seguimiento del mantenimiento de infraestructuras y equipos en el Instituto Tecnológico de San Marcos, haciendo uso del marco de trabajo Scrum, utilizando herramientas tecnológicas como NodeJS y Angular.

## MATERIAL Y MÉTODOS

En esta sección se describe la arquitectura del sistema del cual trata este artículo, las herramientas utilizadas en el desarrollo, así como la metodología de desarrollo del proyecto.

Se decidió utilizar para el desarrollo del sistema el marco de trabajo Scrum para marcar las pautas de las entregas de los módulos del sistema con el cliente, debido a que las metodologías tradicionales no pueden entregar un producto en el marco del tiempo dado [1], así también, se hace uso de una metodología propia para el desarrollo del software, la cual ayuda a dicta los modelos y diagramas que se deben de diseñar para el desarrollo del producto software, dicha metodología es OOHDM (Object Oriented Hypermedia Design Methodology), esta metodología se utiliza debido a que

es orientada a objetos y porque el desarrollador del software la conoce y la ha utilizado en otros proyectos similares, también por ser una de las más utilizadas ya que permite desarrollar aplicaciones web a partir de la utilización de modelos especializados como: conceptual, navegacional e interfaz de usuario teniendo como objetivo simplificar y hacer más eficaz el diseño de aplicaciones [2].

En la figura 1 se puede apreciar el flujo de estas etapas de la metodología implementada.

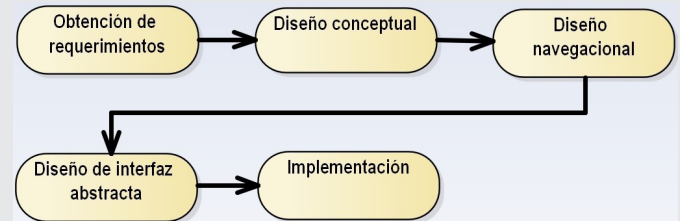


Figura 1. Elementos de un Panel Fotovoltaico.

En la etapa inicial se definen los requerimientos funcionales. Los cuales nos indican las restricciones y las metas del sistema y se obtienen mediante la consulta a los usuarios que harán uso del sistema. Una vez obtenidos se definen a detalle y cada uno servirá como una especificación del sistema. A continuación, se listarán los requerimientos funcionales más relevantes.

### Requerimientos funcionales

Autenticación de usuario.

Registrar usuarios.

Gestionar verificaciones de infraestructura y equipo.

Gestionar el mantenimiento de infraestructura y equipo.

Generar reportes de las verificaciones y mantenimientos de las infraestructuras.

El diagrama de la figura 2 muestra los componentes que conforman cada una de las capas de la arquitectura del sistema y la manera en que interactúan. En la capa de acceso a datos se puede observar el componente base de datos la cual se utiliza mongodb y la cual fue expuesto su implementación en un artículo anterior [3].

A continuación, se describe brevemente el flujo que sigue una petición de información realizada por un usuario, ya sea para insertar información o consultarla. El usuario realiza una petición internamente al controlador a través de una petición http al controlador el cual contiene la lógica de negocio, este crea un objeto del modelo haciendo uso de alguna de las clases (verificación, usuario, plan, solicitud) que se encuentran internamente en el componente modelo, permitiendo crear una capa abstracta de las colecciones que almacenan los datos en la base de datos y que se almacenan temporalmente en memoria al tiempo de mostrarlos en el

componente vistas (formularios y tablas) que muestran la información al usuario final. El controlador es un componente que permite insertar información en la base de datos, así también, la base de datos devuelve la información al controlador, el cual lo retorna dicha información a las vistas para que el usuario pueda ver dicha información. En la figura 2 se puede ver una división de componentes, esto se debe a que las tecnologías que se ejecutan del lado del servidor (backend) se desarrollaron haciendo uso de un api-rest, tal como se planteó en un artículo anterior (citar), esto con la finalidad de que en futuros trabajos se pueda desarrollar una aplicación móvil y, dicha api se pueda consumir de una manera fácil.

el propósito de haber usado cada tecnología tanto del backend, como del frontend en la implementación de los componentes para el desarrollo del sistema.

El equipo de cómputo donde se desarrolló la aplicación tiene las siguientes características:

Laptop marca Dell

Procesador i5

Memoria Ram de 6 Gigas

Sistema operativo Windows 10.

Las herramientas para el desarrollo del sistema propuesto son las siguientes:

Visual Studio Code 2019.

Robo 3T.

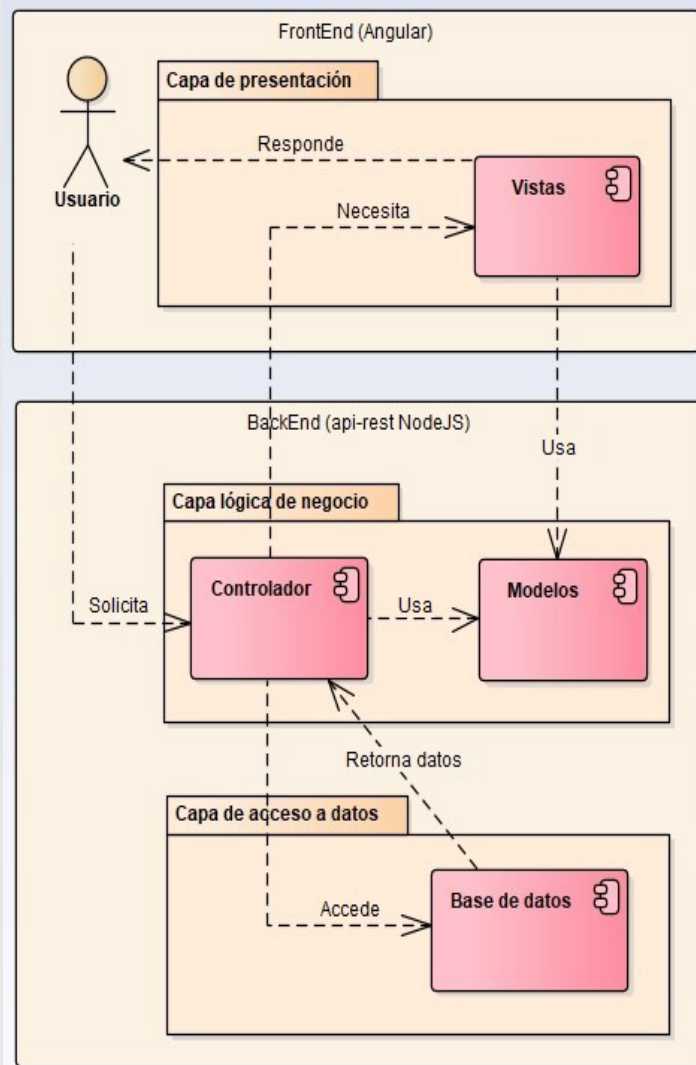


Figura 2 Arquitectura del sistema implementando el patrón arquitectónico MVC.

Para el desarrollo del sistema propuesto, como se puede observar se dividió en dos intereses backend el cual como ya se mencionó existe un artículo en el cual se describe el desarrollo de este, el segundo interés es el frontend, para el cual se hicieron uso de tecnologías de las cuales describiremos, en el ANEXO 1 se describe

## RESULTADOS

Con base en el diseño del sistema presentado en el primer artículo [4], donde se mostraron los diagramas de casos de uso, y en el segundo artículo, el diagrama de clases haciendo uso del UML, estos diagramas dan soporte al desarrollo del sistema. En esta sección se presentan los resultados del uso las tecnologías explicadas anteriormente.

En la figura 3 se presenta la instancia (solución) creada en el IDE Visual Code, que contiene el sistema (mean-siscmie-api).

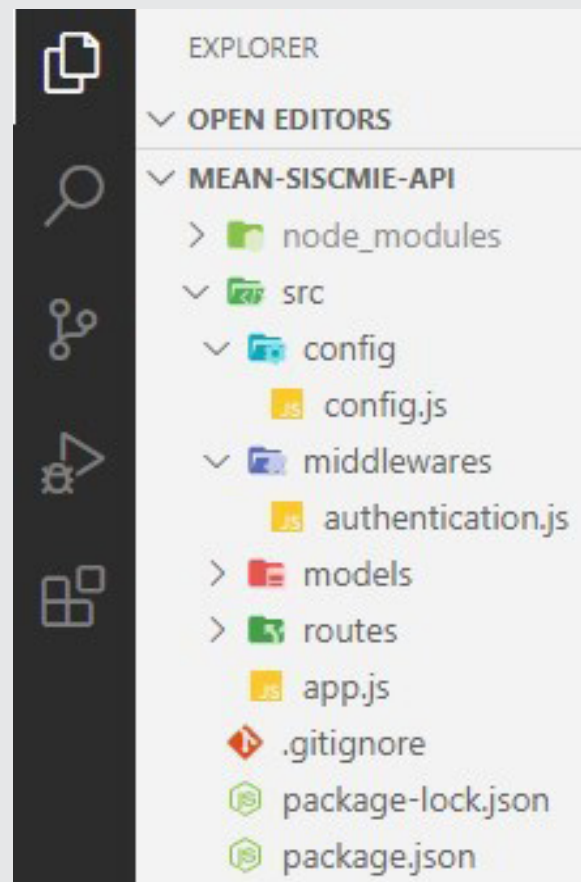


Figura 3 Instancia del sistema en Visual Studio Code.

En la figura 3 se puede observar un proyecto llamado MEAN-SISCMIE-API, el cual es el API REST que proporciona servicios al proyecto cliente programado en angular que muestra los datos al cliente y del cual se hablará más adelante en esta sección. Como se puede observar en la figura 3, existe un conjunto de carpetas las cuales conforman el proyecto del lado del backend, la carpeta node\_modules contiene todas las librerías descargadas del gestor de paquetes NPM, la carpeta config (archivo config.js) contiene las configuraciones de conexión al servidor de base de datos, así como puerto al que se debe conectar y una semilla de encriptación de contraseñas de usuario. La carpeta middlewares (archivo authentication.js) contiene un método que ayuda a verificar las credenciales de los usuarios del sistema antes de iniciar sesión en el sistema y consumir información de la base de datos, la carpeta models contiene las clases que se utilizan para mapear la información de la base de datos, las cuales son similares a los nombres que tiene en la base de datos las colecciones, como se observa en la figura 4 (plans, services, users, requests, providers, groups, depts, articles). La carpeta routes contiene un conjunto de archivos que son los controladores que acceden y responden a las peticiones de los usuarios una vez que iniciaron sesión correctamente en el sistema. El archivo app.js contiene la configuración de las rutas y accesos a la base de datos e iniciador del servidor una vez que se despliega la aplicación en producción.

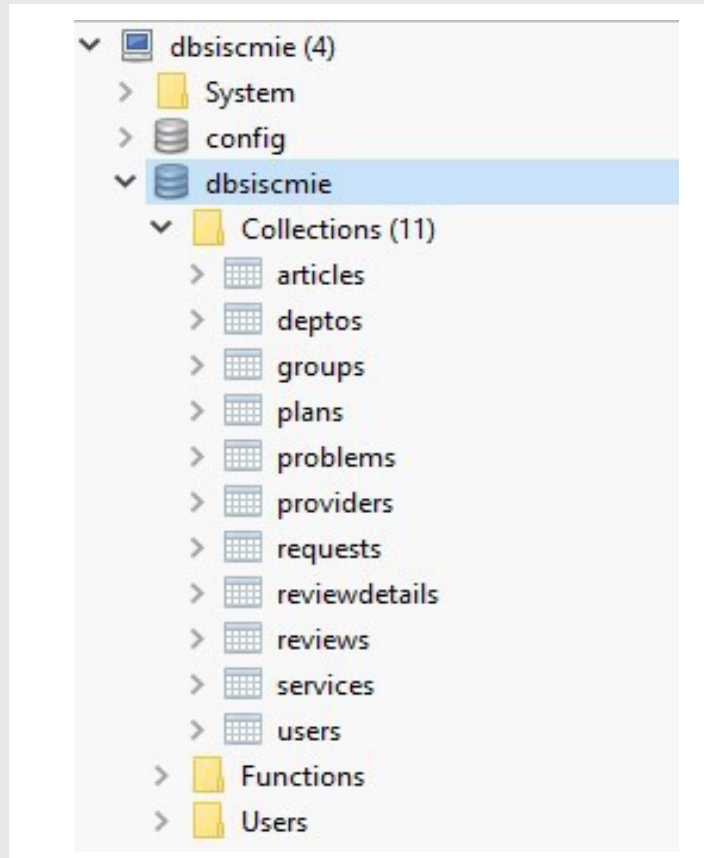


Figura 4 Base de datos del sistema propuesto en Robo 3T.

En la figura 4, se ilustra Robo 3T, el cual nos permite administrar la base de datos que se crea una vez que se ejecuta la aplicación en el servidor. Como se puede observar en la figura 4, la base de datos llamada dbisicmie contiene las colecciones generadas a través de los modelos creados en el proyecto MEAN-SISCMIE-API, la librería que hace posible la migración es moongoose, la cual fue mencionada en la sección anterior, esta librería aparte de que ayuda a mapear, también permite una gran cantidad de funcionalidades, es decir, permite convertir mayúsculas a minúsculas en las validaciones, así como una configuración rápida en el ODM [5].

Es importante mencionar que la estructura de carpetas del proyecto MEAN-SISCMIE-API, la hace el programador del proyecto, ya que el framework NodeJS no te exige una estructura en específico [6].

Como se observa en la figura 2 el proyecto se dividió en dos intereses backend que se desarrolló en NodeJS y frontend que se programó en Angular. Es por eso que la solución final consta de dos proyectos, como se observa en la figura 5, se muestra la solución de Angular.

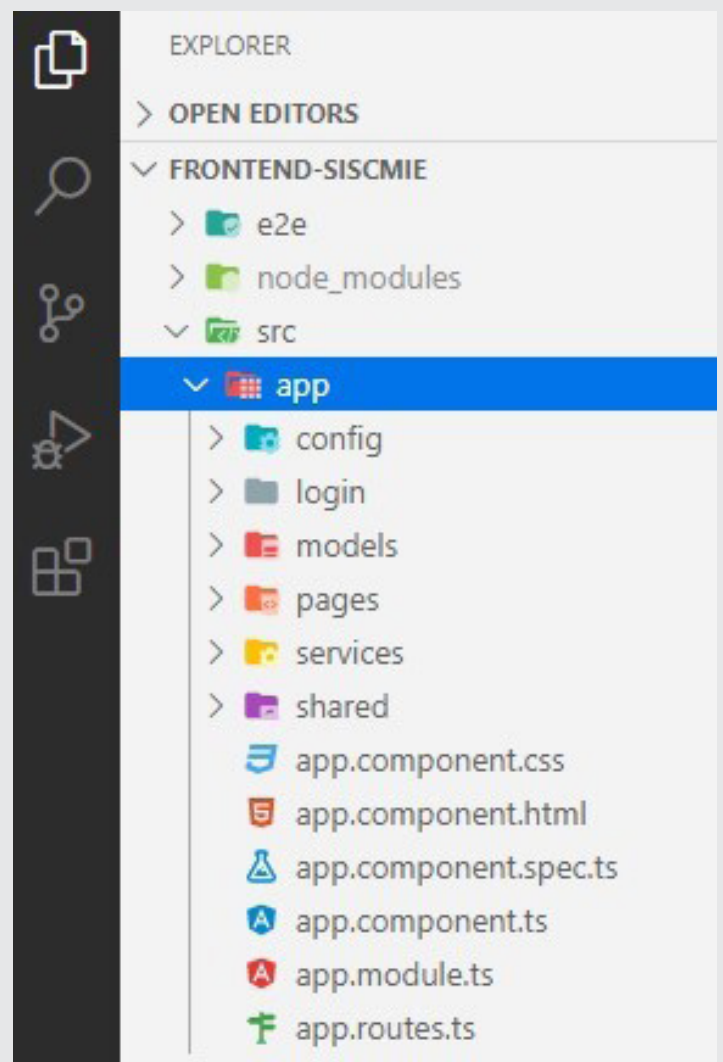


Figura 5 Proyecto frontend en Angular.

En la figura 5 se puede observar que el proyecto cliente está conformado por un conjunto de carpetas, es importante mencionar que esta solución es con la que el usuario final interactuará directamente. A continuación, se describirán las carpetas más relevantes del proyecto. La carpeta config contiene la ruta del api de la cual consumirá los datos a mostrar en las vistas, la carpeta models contiene todas las clases que representan la información que se carga en tiempo de ejecución y que es extraída de la base de datos en formato JSON, el cual es una forma de intercambiar datos entre el backend y frontend [7] de una manera fácil que no necesita de algún programa que interprete la información que llega desde la base de datos a las vistas que se le muestran al usuario final de la aplicación. Los archivos que hacen posible obtener la información de la base de datos son los servicios que se encuentran en la carpeta services, las vistas que hacen posible la interacción con el usuario se encuentran en la carpeta pages, es importante mencionar que Angular es un framework orientado a componentes los cuales pueden ser reutilizables, esto permite que se puedan reutilizar páginas en otros módulos del sistema, dichas vistas (páginas HTML) compartidas se encuentran en la carpeta shared, la carpeta login contiene la página de inicio de sesión, la cual es la primera con la que interactúa el usuario que desea hacer uso del sistema.

**Como resultado del desarrollo de la aplicación web se tiene las siguientes vistas del sistema.**

En la figura 6 se puede observar la página principal para la administración de los planes de mantenimiento, así como la opción para la gestión de usuarios, configuraciones, catálogo del sistema donde se almacenan los departamentos a los cuales se les hacen verificaciones y mantenimientos. La opción llamada principal permite mostrar indicadores de la cantidad de mantenimientos registrados en la plataforma, así como, cuantos están en proceso, cerrados o cancelados.



Figura 6 Menú principal del sistema.

En la figura 7, se observa una tabla con la información de las verificaciones registradas en el departamento de Centro de Cómputo, en cada una de las listas de verificación registradas se pueden ejecutar operaciones como editar, registrar fallos e imprimir la lista con

los fallos registrados, no se implementa la opción de imprimir debido a que, por requerimiento de los usuarios finales, la información no debe de ser borrada una vez que se registra en la base de datos.

Figura 7 Vista que permite visualizar las listas de verificación de infraestructura y equipo registradas.

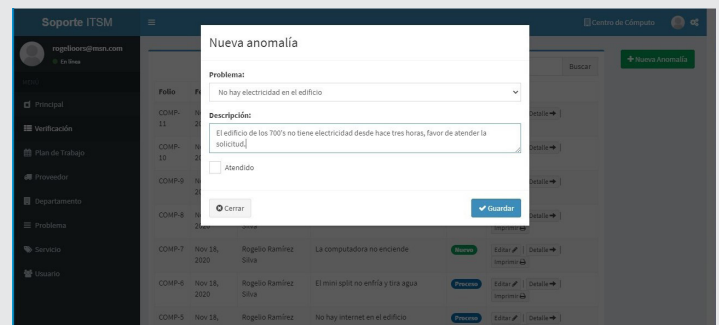


Figura 7 Vista que permite visualizar las listas de verificación de infraestructura y equipo registradas.

En la figura 8 se muestra el formulario que permite registrar una anomalía en una lista de verificación de infraestructura y equipo.

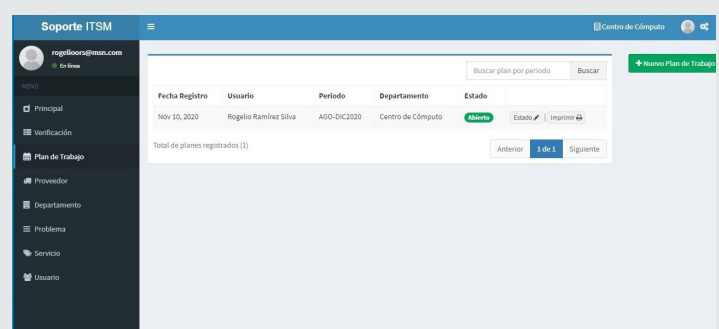


Figura 8 Vista del formulario para el registro de una anomalía en una lista de verificación.

En la figura 9, se muestra el panel en el cual se listan los mantenimientos a realizar por un departamento, dicho panel muestra las opciones de estado e imprimir, esta última opción se utiliza para descargar un documento con formato pdf que contiene todas las anomalías que se programan para realizarse en la fecha indicada por el usuario encargado de dicha actividad.

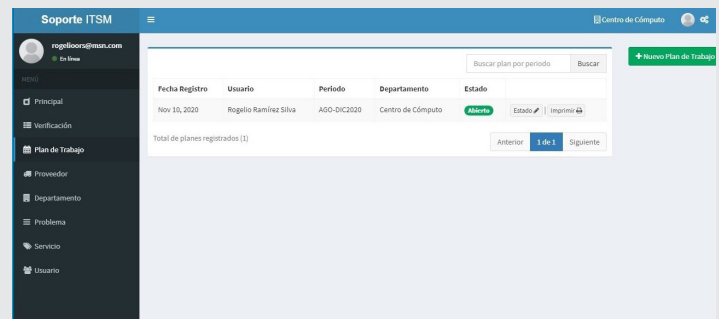


Figura 8 Vista del formulario para el registro de una anomalía en una lista de verificación.

En la figura 10, se observa el formulario que se muestra después de presionar el botón que dice Nuevo plan de trabajo de la figura 9.

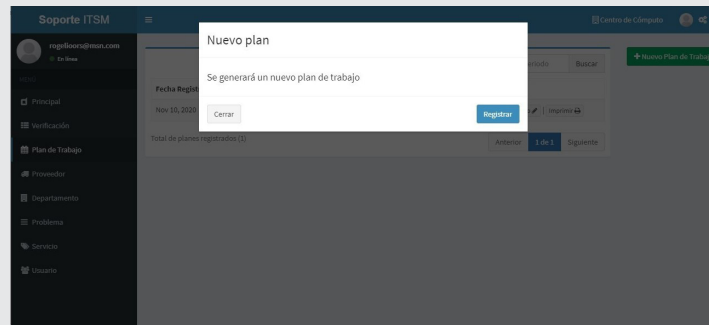


Figura 10 Vista de la creación de un plan de trabajo.

En base al procedimiento que se lleva a cabo en el Instituto Tecnológico de San Marcos, el primer paso del procedimiento es crear una lista de verificación (ver figura 7), el segundo paso es agregarle fallos de equipos o daños de infraestructuras en las listas (ver figura 8), el tercer paso dicta que una vez encontrados todos los fallos de un departamento en el equipo que contiene o infraestructura del mismo edificio se debe de hacer un plan de trabajo para programar en que fechas se deben de resolver dichas anomalías (ver figura 8 y 10), una vez que es creado el plan de trabajo se comienza a programar las actividades seleccionando cada una de ellas, de la lista de verificación para asignarlas al plan deseado como se muestra en la figura 11.

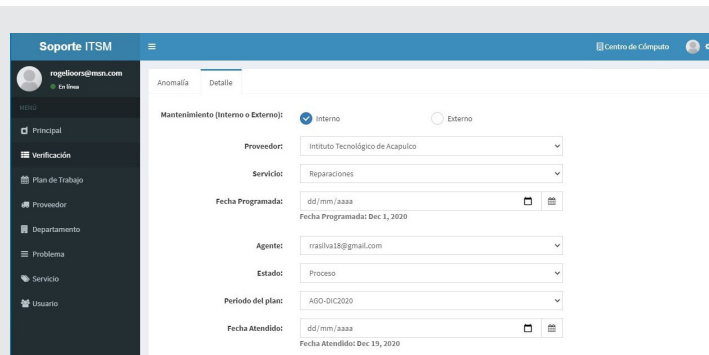


Figura 11 Vista para la programación de un fallo.

Como se observa en la figura 11, una vez que es creado el plan de trabajo aparecerá en el campo periodo del plan de la figura anterior el periodo del plan al cual será asignado el fallo.

El cuarto paso que dicta el procedimiento para el mantenimiento de infraestructura de equipo es que cuando se llegue la fecha de resolver el fallo programado en el plan de trabajo se debe generar una orden de trabajo como la que se muestra en la figura 12.

La orden de trabajo que se genera mostrada en la figura 12, debe ser firmada por el jefe del área donde se

reportó el fallo y también por el jefe del departamento que repara el fallo.



Figura 12 Vista del formato pdf de la orden de trabajo.

## CONCLUSIONES

Como producto de esta investigación se logró desarrollar una plataforma web, la cual cumple con el objetivo principal, que es la implementación de una herramienta informática que permita centralizar, controlar y dar seguimiento a toda la información generada en el proceso del mantenimiento de la infraestructura y el equipo con el que cuenta el Instituto Tecnológico de San Marcos, Guerrero.

El uso de este sistema no garantiza que todas las actividades que se llevan a cabo actualmente en los departamentos que utilicen este sistema dejarán de usar papel al cien por ciento, debido a que actualmente en el estado de Guerrero existen organizaciones e instituciones que se resisten al uso de las tecnologías, así como a la automatización de ciertos procesos.

Los trabajos a futuro del proyecto, son la implementación de nuevos módulos a la plataforma, como notificaciones en tiempo real de los fallos de un equipo, un ejemplo claro puede ser, recibir información de los aires acondicionados haciendo uso de sensores de temperatura, que utilizando el internet de las cosas se puedan mandar mensajes a la plataforma en tiempo real cuando un aire no está enfriando o cuando se apague repentinamente.

## BIBLIOGRAFÍA

[1] Z. Ereiz, «Scrum Without a Scrum Master,» 9 IEEE International Conference on Computer Science and Educational Informatization (CSEI), pp. 325-328, 2019.

[2] Jimmy R. Molina, Mariuxi P. Zea, «Comparación de metodologías en aplicaciones web,» *3Ciencias*, pp. 1-19, 2018.

[3] F. J. G. M. A. D. D. J. I. E. C. M. Rogelio Ramírez Silva, «Implementación del backend de una aplicación web para el mantenimiento de infraestructura y equipo en el ITSM,» *Robotica y Computación. Nuevos Avances*, pp. 286-290, 2020.

[4] F. J. G. M. E. d. I. C. G. E. C. M. Rogelio Ramírez Silva, «Propuesta de diseño de un sistema de información web para el seguimiento y control del mantenimiento de infraestructura y equipo.,» *Academia Journals*, pp. 940-946, 2019.

[5] S. Holmes, *Mongoose for Application Development*, Packt, 2013.

[6] J. Munro, «envatotuts,» 09 Octubre 2017. [En línea]. Available: <https://code.tutsplus.com/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527>.

[7] Alejandro Corbellini, Cristian Mateos, «Persisting big-data: The NoSQL landscape,» *ScienceDirect. Information Systems*, pp. 1-23, 2017.

[8] «Compliance monitoring in business processes: Functionalities, application, and tool-support,» *ELSEVIER. Information Systems*, n.º 54, pp. 209-234, 2015.

[9] A. Sterling, «NodeJS and Angular Tools for JSON-LD,» *IEEE Xplore 13th International Conference on Semantic Computing*, pp. 392-394, 2019.

ANEXO 1 Descripción de uso de cada tecnología para el desarrollo del sistema.

Back-End	
Tecnología	Uso
NodeJS	Proporciona un marco de trabajo para desarrollar una aplicación web basado en el patrón <b>API-REST</b> .
JavaScript	El lenguaje de programación JavaScript que permite la creación de la <b>lógica de negocio</b> en los <b>controladores</b> y de los <b>modelos</b> .
Express	Esta tecnología facilitó la creación de un servidor para recibir peticiones de los clientes (navegadores web) del lado de la tecnología Angular. Es una librería de uso libre que fue descargada del gestor de paquetes <b>NPM</b> .
Mongoose	La librería <b>mongoose</b> permitió crear una conexión en el servidor <b>Express</b> a la base de datos <b>NoSQL Mongodb</b> para insertar y consultar información. Así como mapear los modelos creados y crear a partir de esos modelos las clases de la aplicación.
Mongo-Validator	Esta librería facilita la creación de validaciones del <b>API-REST</b> , es decir, utilizando el lenguaje <b>JavaScript</b> se validan los datos que son insertados.
MongoDB	Para persistir la información se hizo uso del gestor de base de datos <b>NoSQL Mongodb</b> .

Front-End	
Tecnología	Uso
HTML	Como tecnología principal del lado del cliente se hizo uso de HTML la cual proporciona una estructura <b>web</b> general para cada <b>vista</b> del sistema.
CSS	El uso de CSS es indispensable en el desarrollo de aplicaciones web, debido a que se usa con la finalidad de hacer que el sistema adquiera un aspecto agradable al usuario. Debido a que <b>CSS</b> permite manipular el aspecto estructural de las imágenes en cuanto a color, orden, tamaño, así como también el de los elementos HTML que se utilizaron.
JavaScript	Aunque indirectamente, también se utilizó para la manipulación del contenido en el <b>Front-End</b> realizando validaciones.
TypeScript	Todo lo que se programa en Angular se transpila a JavaScript, ya que es el lenguaje nativo que entiende <b>Angular</b> .
Angular	Angular es el framework utilizado para desarrollar la aplicación web cliente, es decir, todas las tecnologías mencionadas anteriormente fueron implementadas dentro de estas tecnologías. La cual nos dicta los componentes que se deben programar y las estructuras que se deben desarrollar para desarrollar un software funcional al cien por ciento.
Bootstrap	Para crear <b>vistas</b> con <b>diseño</b> que alguien más ya desarrollo y que están disponibles gratis, se implementó el framework <b>Bootstrap</b> , esto se hizo con la finalidad de no invertir tiempo en algo que ya alguien más se tomó la molestia de diseñar.
AdminLTE2	Es una plantilla diseñada por un tercero que está disponible sin ningún costo y con demasiado soporte y que internamente utiliza <b>Bootstrap</b> , esta plantilla te proporciona los formularios, tablas, labels, inputs, modal-popups con diseños agradables, con colores, bordes, tipado de letras y estilos en general.
ChartJS	Para la graficación de los reportes se utilizó la librería <b>ChartJS</b> .