

# **SISTEMA DE MONITOREO PARA LA INFRAESTRUCTURA TECNOLÓGICA DEL TECNOLÓGICO NACIONAL DE MÉXICO EN CELAYA**

## **MONITORING SYSTEM FOR TECHNOLOGICAL INFRASTRUCTURE OF THE TECNOLÓGICO NACIONAL DE MÉXICO EN CELAYA**

### **José Jesús Sánchez Farías**

Tecnológico Nacional de México en Celaya, México  
*jesus.sanchez@itcelaya.edu.mx*

### **Luis Adrián Rodríguez Rodríguez**

Tecnológico Nacional de México en Celaya, México  
*14030714@itcelaya.edu.mx*

### **José Antonio Medina Malagón**

Tecnológico Nacional de México en Celaya, México  
*14030678@itcelaya.edu.mx*

### **Juan Ignacio Cerca Vázquez**

Tecnológico Nacional de México en Celaya, México  
*nacho@itcelaya.edu.mx*

### **Luis Alberto López González**

Tecnológico Nacional de México en Celaya, México  
*luislao@itcelaya.edu.mx*

**Recepción:** 31/octubre/2019

**Aceptación:** 17/diciembre/2019

## **Resumen**

El Tecnológico Nacional de México en Celaya cuenta con una gran infraestructura tecnológica que le permite ofrecer servicios a los estudiantes, profesores, personal administrativo y comunidad en general. Dentro de esta infraestructura se encuentran servidores que hospedan portales de internet, sistemas de información, bases de datos y otros servicios que permiten el correcto funcionamiento de los sistemas, existen problemáticas que se presentan al mantener toda esta infraestructura como son cortes de energía eléctrica; errores en procesos de instalación y configuración; errores humanos y en general fallas por sobrecargas y mal funcionamiento de los servidores provocando el ofrecer un mal servicio a los usuarios. En ocasiones los administradores de sistemas no se enteran de las fallas hasta que no se les es reportado por los mismos usuarios o compañeros de trabajo, permitiendo

prolongados periodos sin ofrecer el servicio. Para esto se decidió desarrollar un sistema de información que permita el monitoreo de la infraestructura tecnológica, abarcando cualquier dispositivo que se pueda interconectar a la red institucional y así mantener en constante vigilancia los servicios que éstos ofrezcan. De esta manera cada vez que se presente una falla en un dispositivo, casi de manera inmediata se reportará la falla a los administradores de sistemas mediante el envío de correos electrónicos y notificaciones a un dispositivo móvil con una aplicación previamente instalada. Dentro de los resultados obtenidos se encuentra el desarrollo de una aplicación web que permita la gestión de dispositivos a monitorear, usuarios con diferentes perfiles y servicios asociados a los dispositivos; así también, se tiene el desarrollo de una aplicación móvil para las plataformas Android y IOS que permite recibir las notificaciones de fallas y permitir a los administradores de sistemas dar seguimiento a las fallas de manera inmediata.

**Palabras Clave:** Desarrollo móvil, Desarrollo web, Infraestructura Tecnológica, Monitoreo, Nagios.

### **Abstract**

*The Tecnológico Nacional de México en Celaya has a large technological infrastructure that allows it to offer services to students, teachers, administrative staff and the community in general. Within this infrastructure are servers that host Internet web sites, information systems, databases and other services that allow the systems to function properly, there are problems that arise when maintaining all this infrastructure such as power outages; errors in installation and configuration processes; human errors and in general failures due to overloads and malfunctions of the servers causing the poor service to the users. Occasionally, system administrators do not know about failures until they are reported by the same users or co-workers, allowing extended periods without offering the service. For this, it was decided to develop an information system that allows the monitoring of the technological infrastructure, covering any device that can be interconnected to the institutional network and thus keep the services offered by them in constant vigilance. In this way every time a failure occurs in a device, the failure will almost*

*immediately be reported to system administrators by sending emails and notifications to a mobile device with a previously installed application. Among the results obtained is the development of a web application that allows the management of devices to be monitored, users with different profiles and services associated with the devices; likewise, there is the development of a mobile application for the Android and IOS platforms that allows to receive fault notifications and allow system administrators to track the faults immediately.*

**Keywords:** *Mobile development, Web development, Technological infrastructure, Monitoring, Nagios.*

## **1. Introducción**

El Tecnológico Nacional de México en Celaya se caracteriza por ofrecer una gran cantidad de servicios digitales a la comunidad estudiantil, personal docente y administrativo, así como la población en general que en algún momento se interesa por un servicio educativo o tecnológico. Dentro de los servicios que se ofrecen se encuentran el portal web oficial de la institución, el sistema integral de información, portales web de cada una de las carreras de licenciatura y posgrado, sistemas para la gestión de la calidad, servicio de correo electrónico, así como servicios de red más específicos como dns, dhcp, web, ftp, smtp, ssh, vnc, vpn, servicios web, etc., para lo cual es necesario tener una gran infraestructura tecnológica que permita hospedar, mantener y gestionar todos estos servicios, así como tener personal altamente capacitado que se encargue de realizar esas labores de gestión.

Durante este proceso de gestión surgen fallas, algunas de ellas ajenas al control interno del departamento y otras que se pueden anticipar y tomar medidas para evitarlas. Algunas de las fallas a las que se ha enfrentado el equipo de trabajo se encuentran los cortes continuos de la energía eléctrica, las malas instalaciones y configuraciones de los equipos de red y servidores, así como vulnerabilidades de software y sobrecargas en los servidores ante la demanda de servicios por parte de los usuarios finales. Estas fallas dan lugar a la oferta de un mal servicio, dejando a los usuarios sin poder realizar sus actividades, trámites, consulta de información y

detener el servicio, incluso de departamentos completos que dependen totalmente de sistemas de información para trabajar.

Un administrador de sistemas es el encargado de mantener en funcionamiento todos los servicios digitales que se ofrecen en la Institución, ante la presencia de las fallas antes mencionadas, los administradores en ocasiones no se enteran del problema hasta que no se les es reportado vía telefónica, correo electrónico o de manera directa, para que esto suceda, llegan a pasar varias horas e incluso días cuando son periodos de fin de semana o vacaciones hasta que el administrador se entera de la problemática y comienza a realizar labores para solucionar este problema. El propósito de la investigación es desarrollar un sistema de información que permita el monitoreo de los dispositivos de red que forman parte de la infraestructura tecnológica de la institución y que permiten ofrecer sus servicios. El sistema será capaz de detectar fallas casi de manera inmediata y reportar a los administradores de sistemas vía notificaciones en los dispositivos móviles y correos electrónicos, así como proporcionar las herramientas necesarias para dar seguimiento a la problemática.

Para el desarrollo del sistema de monitoreo se utilizaron tecnologías de vanguardia para el monitoreo de sistemas como es Nagios, Laravel como *framework* para el desarrollo de aplicaciones web y Flutter para el desarrollo de la aplicación móvil, así como una metodología oficial y considerada ágil para el desarrollo de software.

### **Métodología ágil de programación extrema**

Extreme Programming (XP) surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. Las metodologías de desarrollo de software tradicionales (ciclo de vida en cascada, evolutivo, en espiral, iterativo, etc.) aparecen, comparados con los nuevos métodos propuestos en XP, como pesados y poco eficientes. Como respuesta a esto, se ha visto en los últimos tiempos el surgimiento de “Metodologías Ágiles”. XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan, así también, alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aun en fases tardías del ciclo de

vida del desarrollo. La metodología también enfatiza el trabajo en equipo. Tanto gerentes como clientes y desarrolladores son partes del mismo equipo dedicado a entregar software de calidad [Joskowickz, 2008].

## **GitLab**

GitLab es una plataforma de desarrollo de software totalmente integrada que permite que un equipo sea transparente, rápido, efectivo y coherente desde el debate sobre una nueva idea hasta la producción, todo en la misma plataforma [GitLab Docs, 2019]. GitLab es un administrador de repositorios basado en Git y una potente aplicación completa para el desarrollo de software.

Tiene una interfaz amigable para usuarios expertos y novatos, permite trabajar de manera efectiva, tanto desde la línea de comandos como desde la propia interfaz de usuario. No solo es útil para los desarrolladores, sino que también puede integrarse en todo un equipo para reunir a todos en una plataforma única. La lógica de flujo de trabajo de GitLab es intuitiva y predecible, lo que hace que toda la plataforma sea fácil de usar y más fácil de adoptar para diferentes perfiles de desarrolladores de software [Ramos, 2016].

## **Nagios**

Nagios es una herramienta de software libre de código abierto para monitoreo continuo. Le ayuda a monitorear el sistema, la red y la infraestructura. Se utiliza para el monitoreo continuo de sistemas, aplicaciones, servicios y procesos comerciales en una cultura DevOps. Nagios ejecuta complementos almacenados en el mismo servidor. El complemento se conecta con un host u otro servidor en su red o Internet. Por lo tanto, en caso de falla, el núcleo de Nagios puede alertar al personal técnico sobre los problemas. Para que su equipo técnico realice el proceso de recuperación antes de la interrupción de los procesos comerciales [Guru99, 2019].

## **Laravel**

Creado por Taylor Otwell como un *framework* web PHP gratuito de código abierto, Laravel está destinado a facilitar y acelerar el proceso de desarrollo de aplicaciones

web con un gran gusto por la simplicidad. Sigue el patrón arquitectónico modelo-vista-controlador (MVC), así como la codificación estándar PSR-2 y el estándar de carga automática PSR-4.

Laravel tiene una arquitectura de microservicios, lo que lo hace muy extensible mediante el uso de paquetes existentes y otros creados a la medida por terceros [Stack Overflow contributors, 2018].

## **PostgreSQL**

PostgreSQL es un sistema de gestión de base de datos relacional de objetos (ORDBMS) basado en POSTGRES, versión 4.2, desarrollado en el Departamento de Informática de la Universidad de California en Berkeley. POSTGRES fue pionero en muchos conceptos que solo estuvieron disponibles en algunos sistemas de bases de datos comerciales mucho más tarde.

PostgreSQL es un descendiente de código abierto del código original de Berkeley. Admite una gran parte del estándar SQL y ofrece muchas características modernas como consultas complejas, llaves foráneas, disparadores, vistas actualizables, integridad transaccional y control de concurrencia [PostgreSQL Documentation, 2019].

## **Flutter**

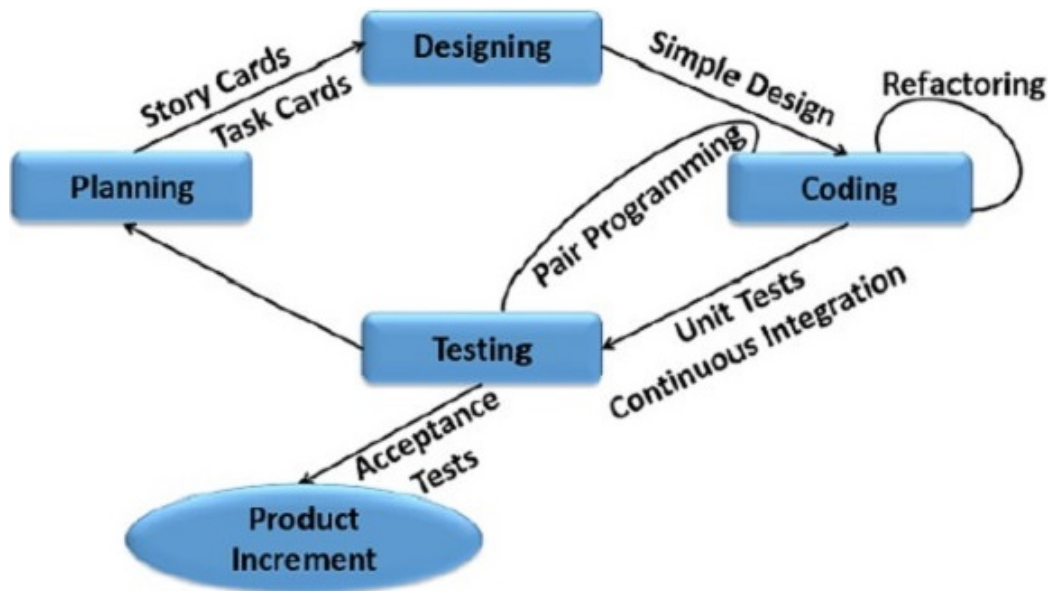
Flutter es el kit de herramientas de UI de Google para realizar hermosas aplicaciones, compiladas nativamente, para móvil, web y escritorio desde una única base de código. Proporciona una manera para desarrollar interfaces gráficas multiplataforma para los sistemas operativos Android y IOS. Flutter se integra con código Java para Android, así como con Objective C y Swift para IOS por lo que no es necesario reescribir completamente aplicaciones existentes para comenzar a utilizarlo [Flutter Documentation, 2019].

El uso de las tecnologías antes descritas permitirá a la Institución tener una plataforma completa bajo el uso de software libre, promoviendo su uso y ahorrando la compra de licencias o la adquisición de soluciones comerciales muy costosas y que no están adaptadas a las necesidades particulares de los usuarios.

## 2. Métodos

Cabe mencionar que el sistema de monitoreo se encuentra aún en proceso de desarrollo durante la escritura de este artículo, siendo un proyecto planeado para seis meses en conjunto con dos residentes de la carrera Ingeniería en Sistemas Computacionales. Los métodos que se están utilizando para el desarrollo de la solución informática conlleva varias fases en las cuales se realizan acciones para lograr metas específicas que permitirán alcanzar el objetivo final. La metodología utilizada está basada en una metodología formal de desarrollo de software llamada eXtreme Programming (XP), la cual permite el desarrollo de un software de manera más simple y ágil enfocada a los resultados.

En la figura 1 se muestra el esquema de trabajo bajo esta metodología y que permite llevar a cabo el proceso de construcción del sistema de información, puntualizando que es un proceso iterativo con el cual se logran entregas periódicas de avances y de las cuales los usuarios están en constante contacto y realizando pruebas de funcionamiento.



Fuente: [Quest10 Meets Answer, 2018]

Figura 1 Diagrama metodología “eXtreme Programming” (XP).

A continuación, se listan y describen las fases establecidas para el desarrollo del software y que forman parte de la metodología ágil XP:

- **Exploración.** Dentro de esta fase se definió el alcance general del proyecto por parte del cliente principal del sistema de información que es el Depto. Centro de Cómputo a través de sus áreas de desarrollo de software, redes y operaciones. A través de la redacción de historias de usuarios, el cliente definió sus necesidades y requerimientos para las aplicaciones, así también, en el equipo de trabajo del proyecto se estimaron los tiempos de desarrollo a través de la información proporcionada, estimaciones que son consideradas primarias y que pudieron cambiar en las fases de iteración durante un análisis más detallado.

En esta fase se definió que era necesario desarrollar dos aplicaciones para dar solución a la problemática; una aplicación web la cual permitiera a los coordinadores de área del Depto. Centro de Cómputo gestionar los dispositivos y servicios a monitorear de los cuales pueda listar, agregar, modificar, eliminar y visualizar pantalla de monitoreo de los mismos, así como consultar bitácoras de movimientos y fallas que hayan surgido; por otra parte, la segunda aplicación consiste en una aplicación móvil que permita también visualizar los dispositivos y servicios que se monitorean y además recibir notificaciones en tiempo real ante las fallas que surjan.

Durante esta etapa se definieron las plataformas para el desarrollo de las aplicaciones. Para el caso de la aplicación web se eligió el *framework* Laravel, debido a su simplicidad, flexibilidad y seguridad, además que es una plataforma que cumple con los más altos estándares de buenas prácticas de programación y con las políticas de desarrollo de software del Tecnológico Nacional de México en Celaya. En el caso de la aplicación móvil, se tomó la decisión para desarrollarla en la plataforma Flutter, la cual permite crear aplicaciones móviles multiplataforma enfocadas principalmente para los sistemas operativos Android y IOS, lo cual permitirá a los usuarios poder instalar la aplicación en cualquier dispositivo que contenga alguno de estos sistemas operativos.

- **Planificación.** En esta fase se realizaron tres reuniones con los coordinadores de área, el jefe del Centro de Cómputo y los desarrolladores



del sistema para organizar las historias de usuario, planificar el orden y tiempos de implementación de estas, así como los entregables. Como resultado de esta fase se obtuvo el “Plan de Entregas”, consistiendo en un cronograma donde se determinó un número fijo de iteraciones y las historias de usuario que comprende cada una de ellas con sus respectivos tiempos de análisis, desarrollo e implementación. Las iteraciones definidas como parte del plan fueron:

- ✓ Instalación y puesta en marcha del servidor.
  - ✓ Instalación y configuración de herramientas.
  - ✓ Desarrollo de interfaces gráficas web para catálogos generales
  - ✓ Desarrollo de interfaz web de monitoreo y gestión de dispositivos
  - ✓ Desarrollo servicios web
  - ✓ Desarrollo de aplicación móvil
- **Iteraciones.** Como se determinó en el punto anterior, fueron seis el número de iteraciones para el desarrollo del sistema de monitoreo:
    - ✓ **Instalación y puesta en marcha del servidor.** En esta iteración se asignó el equipo de cómputo que actuó como servidor para hospedar todas las herramientas necesarias y el sistema de monitoreo. El equipo consiste en una máquina virtual que corre sobre un servidor con las características que se muestran en la tabla 1.

Tabla 1 Características generales del servidor

Marca	Modelo	CPU	Memoria	Disco Duro	Sistema Operativo
HP	ProLiant ML350 G6	8 CPU's Intel Xeon 2.13 GHZ	24 GB	1TB	VMWare

Fuente: Elaboración propia

En la máquina virtual creada se asignaron 8GB de memoria con 200GB de espacio en disco y se instaló el sistema operativo Debian GNU/Linux 10 (64-bit), al cual se aplicaron las últimas actualizaciones de seguridad y paquetería estable. Aunque la institución cuenta con una infraestructura de seguridad para la protección de los servidores,

se decidió instalar un firewall para la protección interna en la red, éste fue el paquete UFW mejor conocido como Uncomplicated Firewall el cual tiene la configuración de bloqueo automático de entradas y salidas de paquetes y se habilitaron sólo los puertos 80 y 5432 para Apache y PostgreSQL respectivamente.

- ✓ **Instalación y configuración de herramientas.** Se instalaron y configuraron las herramientas necesarias para el desarrollo del proyecto, las cuáles son imprescindibles para su correcto funcionamiento. Estas herramientas son paquetes de software libre que tienen una determinada funcionalidad:
  - a) **Sistema de monitoreo Nagios.** Consiste en el corazón principal del sistema de información, el cual ejecuta los servicios y procesos necesarios para llevar a cabo el monitoreo de dispositivos y procesos. Recibe peticiones y envía respuesta con el estatus e información adicional del servicio a monitorear.
  - b) **Motor de base de datos PostgreSQL.** Es el sistema gestor de base de datos que permite almacenar la información de una manera organizada y un propósito en común, contiene el esquema de base de datos definido para el funcionamiento de las aplicaciones web y móvil.
  - c) **Framework de desarrollo Laravel.** Es el *framework* que permite a los desarrolladores construir la aplicación web en el lenguaje de programación PHP bajo ciertos estándares y estructuras bien definidas.
  - d) **Sistema GitLab.** Contiene el repositorio del código fuente del proyecto tanto para la aplicación web como la móvil, así también, la documentación técnica de las aplicaciones y un tablero para el seguimiento de las iteraciones y asignación de tareas a los usuarios involucrados en el proyecto.
- ✓ **Desarrollo de interfaces gráficas web para catálogos generales.** Dentro de las iteraciones que tienen que ver con el desarrollo propio

del sistema de información, se pasa por una serie de etapas que abarca el análisis, diseño, implementación y pruebas. Para este caso particular, se desarrollaron todas las interfaces web que permiten gestionar catálogos generales como son: servicios, dispositivos, categorías, ubicaciones, usuarios y la asociación que existe entre ellos. Las pantallas consisten de manera general en una interfaz que me permite listar la información de cada catálogo, así como realizar operaciones de altas, bajas y modificaciones sobre los mismos. La información que se maneja en estos catálogos generales es fundamental ya que serán los que alimenten a las interfaces clave que permitirán realizar la actividad principal de monitoreo y seguimiento a los problemas. Durante la etapa de implementación de esta iteración, se liberaron los módulos web en el servidor de desarrollo para que pasara a una etapa posterior de pruebas por parte de la persona que fungirá como administrador principal de la aplicación.

- ✓ **Desarrollo de interfaz web para monitoreo de infraestructura.** Ésta es la interfaz principal del sistema y la que llevó más tiempo durante el desarrollo ya que es la que permite realizar la actividad principal del sistema que es monitorear los diferentes dispositivos y servicios que los usuarios hayan definido, éstos últimos pueden dar de alta los propios y programar el tiempo en que el sistema tiene que estar verificando su estatus. Cada coordinador de área puede manejar cualquier cantidad de dispositivos y/o servicios a monitorear, para lo cual debe proporcionar la siguiente información: nombre, descripción, dirección IP, dirección MAC, puerto TCP/IP y periodo; en este último podrá decidir el periodo de tiempo de verificación indicando una hora específica y los días de la semana.
- ✓ **Desarrollo servicios web.** Debido a que el sistema de información abarca también el desarrollo de una aplicación móvil que permitiera la actividad de monitoreo, se tuvo que compartir la misma base de datos con la información de los usuarios, dispositivos y servicios. Para esto,

se desarrolló una API (Application Program Interface) que permitió monitorear el consumo de servicios web para proporcionar la información necesaria a la aplicación móvil para su correcto funcionamiento. El API se desarrolló utilizando la tecnología REST (Representational State Transfer), la cual permitió mediante sus diferentes métodos, consultar la información de la base de datos principal.

A continuación, se describen los distintos servicios web desarrollados:

- a) **Generación token de autenticación.** Genera un token o cadena alfanumérica de autenticación que permitirán iniciar sesión a un usuario desde la aplicación móvil. Para esto, el servicio web recibe como parámetros el nombre de usuario y contraseña.
- b) **Recuperación de información básica del usuario.** Una vez que un usuario ha sido autenticado, la aplicación móvil puede recuperar información básica del mismo como es nombre, estatus, descripción y los diferentes dispositivos que dados de alta desde la aplicación web para su monitoreo.
- c) **Recuperación de información detallada del usuario.** Permite recuperar más información relacionada con el usuario autenticado como es nombres, apellidos, correo electrónico, CURP, teléfono, dirección y rol. Con esta información es posible acceder a una sección de perfil del usuario desde la aplicación móvil para su consulta.
- d) **Notificaciones de usuarios.** A través de este servicio web es posible recuperar todas las notificaciones recibidas de un determinado usuario, enviando como parámetro su token de autenticación.
- e) **Información de dispositivos.** Desde la aplicación móvil es posible mostrar los diferentes dispositivos que puede monitorear un usuario, para esto, el servicio web a través del

token de autenticación, recupera información detalladas de los dispositivos y servicios asociados al usuario. La información recuperada contiene nombre, descripción, comandos, tipos de de servicios y mensajes.

- f) **Guardado de token de dispositivo.** Una característica importante en la aplicación móvil es su capacidad para recibir notificaciones cuando se presente una falla, para esto, es necesario generar un token de dispositivo (*device token*), la cual es una llave única para un dispositivo (*smartphone, tablet*) emitido por Apple o Google para el envío de notificaciones. Este token permitirá enrutar los mensajes por parte de los proveedores y asegurarse que la notificación es entregada exactamente al dispositivo con la aplicación correcta. Para el almacenamiento de este token en la base de datos, se desarrolló un servicio web que actuará al inicio de sesión de cada usuario, así se tendrá actualizado el dispositivo en el cual se recibirán las notificaciones.

✓ **Desarrollo de aplicación móvil.**

Ésta es la última iteración que comprende el proyecto de software y consiste en el desarrollo de la aplicación móvil que permite la actividad de monitoreo para los usuarios, así como recibir notificaciones en caso de fallas. La aplicación se conforma de las siguientes interfaces: pantalla de autenticación, consulta del perfil del usuario, consulta de dispositivos y su estatus, recepción de notificaciones y consulta de datos históricos.

- **Puesta en producción.** Ésta representa la última fase del proyecto, la cual consiste en la puesta en marcha del sistema de información en un entorno productivo. En éste, los usuarios del sistema podrán acceder a él mediante una dirección de internet y dominio institucional. Se llevará a cabo un proceso de migración de la etapa de desarrollo al entorno de producción, en el cual se migrará el código fuente del sistema, la base de datos y se tramitará la

gestión de una dirección IP homologada y su correspondiente dominio para su acceso por internet.

### **3. Resultados**

Como resultados de la investigación se obtuvieron puntualmente dos productos de software que permiten realizar el monitoreo de la infraestructura tecnológica, así como el seguimiento a las fallas presentadas. Aunque el proyecto se encuentra en sus últimas fases de construcción, ya ha sido posible comenzar a realizar pruebas, esto gracias a la metodología ágil que se utilizó. Para esto se implementó un servidor de desarrollo que permite tener en funcionamiento el sistema de información y accesible para los usuarios principales.

#### **Aplicación Web**

Dentro de la aplicación web desarrollada se crearon varios módulos que tienen una función específica:

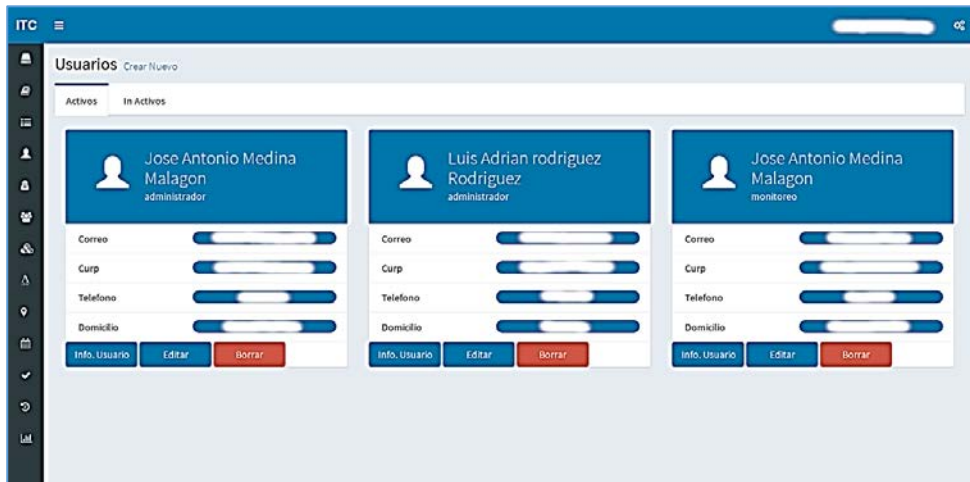
- Autenticación
- Registro de usuarios
- Gestión de usuarios
- Recuperación de contraseña
- Gestión de dispositivos y servicios
- Históricos

En seguida se presentan las figuras que corresponden a algunos de los módulos desarrollados. En la figura 2 se muestra la pantalla de autenticación que permite acceder al sistema principal. En la figura 3 se muestra la interfaz que permite la gestión de los usuarios del sistema. En la figura 4 se muestra una pantalla para la consulta de dispositivos, su estado, información general y la posibilidad de agregar servicios y responsables del mismo. La pantalla del sistema mostrada en la figura 5, es una de las principales, ya que es la que muestra la interfaz de monitoreo. Se puede consultar el estatus de los dispositivos, sus servicios, dirección IP y mensaje de respuesta al momento de la verificación.



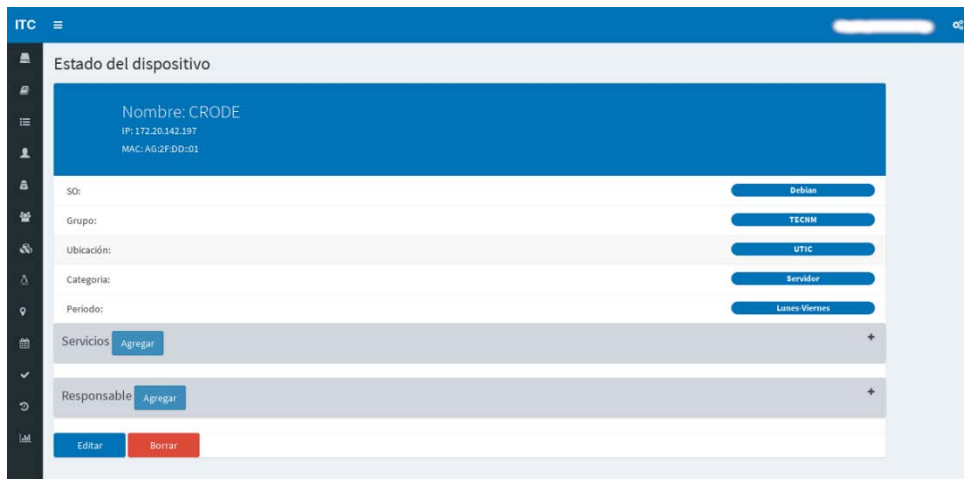
Fuente: [SMITC, 2019]

Figura 2 Pantalla de autenticación.



Fuente: [SMITC, 2019]

Figura 3 Gestión de usuarios.



Fuente: [SMITC, 2019]

Figura 4 Dispositivos verificados.

The screenshot shows the 'Monitoreo ITC' interface. On the left is a navigation menu with options like 'Dispositivo', 'Servicio', 'Tipo Servicio', 'Usuario', 'Rol', 'Grupo', 'Categoria', 'Sistema', 'Ubicación', 'Período', 'Estado', and 'Historico'. The main area displays a table with columns: 'Dispositivo', 'Servicio', 'IP', 'Fecha', 'Estatus', and 'Mensaje'. The table contains several rows of data, including entries for 'CRODEcito' with various services (HTTP, SSH, POP3) and their corresponding status (Correcto or Crítico) and messages.

Dispositivo	Servicio	IP	Fecha	Estatus	Mensaje
CRODEcito	HTTP	[Redacted]	2019-10-30 09:10	Correcto	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time
CRODEcito	HTTP	[Redacted]	2019-10-16 13:10	Crítico	CHECK_NRPE: Error - Could not connect to 172.20.142.197. Check system logs on 172.20.142.197
CRODEcito	HTTP	[Redacted]	2019-10-23 09:10	Crítico	HTTP CRITICAL - Unable to open TCP socket
CRODEcito	SSH	[Redacted]	2019-10-29 10:10	Crítico	CRITICAL - Socket timeout
CRODEcito	HTTP	[Redacted]	2019-10-30 09:10	Crítico	HTTP CRITICAL - Unable to open TCP socket
CRODEcito	HTTP	[Redacted]	2019-10-16 13:10	Correcto	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time
CRODEcito	HTTP	[Redacted]	2019-10-23 09:10	Correcto	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time
CRODEcito	POP3	[Redacted]	2019-10-29 10:10	Crítico	CRITICAL - Socket timeout
CRODEcito	HTTP	[Redacted]	2019-10-30 09:10	Correcto	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.002 second response time

Fuente: [SMITC, 2019]

Figura 5 Monitoreo de dispositivos.

En la figura 6 se muestra el detalle del dispositivo monitoreado, se consulta información detallada del mismo, así como los servicios de red y hardware asociados.

The screenshot shows the 'Detalle del dispositivo' page for 'CRODE'. It includes a 'Actualizar' button and a 'Detalles' section with the following information: Responsable: Enviar Correo; Luis Adrian rodriguez Rodriguez; Jose Antonio Medina Malagon; IP Address: [Redacted]; Mac Address: A6:2F:DD:01; Sistema Operativo: Debian (10); Grupo: TECHM; Ubicación: UTIC - Laboratorio utics campus 2. Below this are two sections: 'Servicios de Red' showing 'FTP' with a message 'The service 'FTP' on host 'CRODE' could not be found.' and 'Servicios de Hardware' showing 'Uso Disco Duro' with a message 'The service 'Uso' on host 'CRODEcito' could not be found.' There are buttons for 'Iniciar', 'Reiniciar', and 'Parar' at the bottom.

Fuente: [SMITC, 2019]

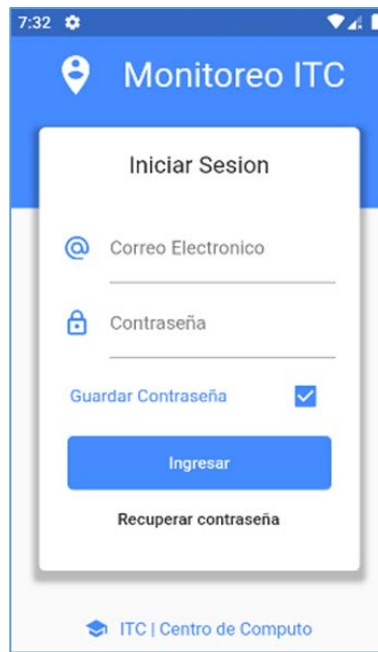
Figura 6 Detalle del dispositivo.

## Aplicación Móvil

Como resultados de la aplicación móvil, se obtuvieron los módulos: autenticación de usuarios, consulta de perfil para el usuario autenticado, acceso a los dispositivos

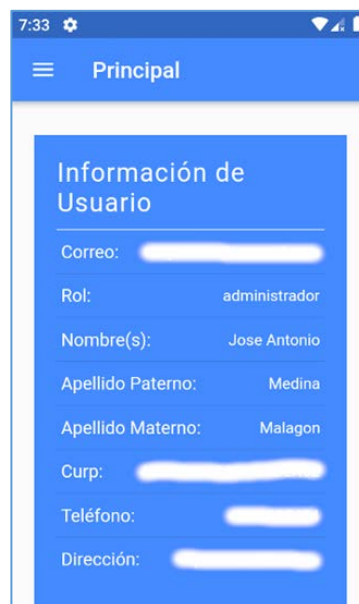


monitoreados, detalles de éstos y recepción de notificaciones. En las figuras 7, 8, 9 y 10 se muestran las diferentes pantallas que representan algunas de las funcionalidades de estos módulos.



Fuente: [SMITC, 2019]

Figura 7 Pantalla de autenticación.



Fuente: [SMITC, 2019]

Figura 8 Información y perfil del usuario.

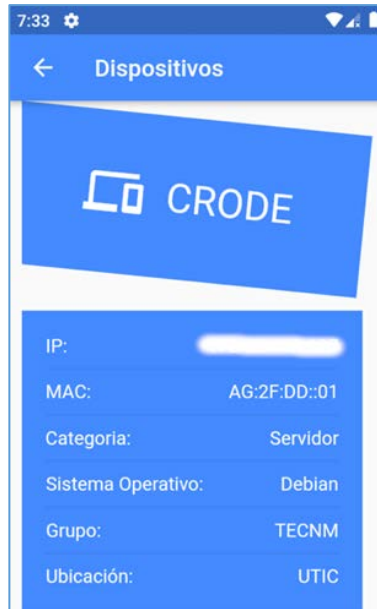
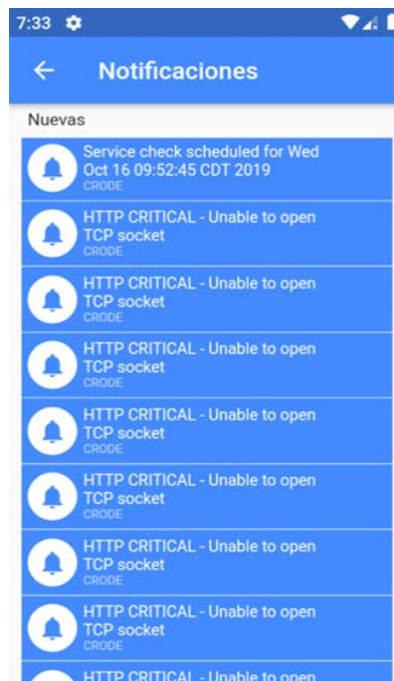


Figura 9 Consulta de dispositivos.



Fuente: [SMITC, 2019]

Figura 10 Consulta de notificaciones

En seguida se listan los beneficios que se obtendrán como consecuencia de la puesta en marcha de las dos aplicaciones:

- Notificaciones en tiempo real ante fallas.

- Mejorar la disponibilidad de los servicios que ofrece la institución.
- Reducción en tiempos de respuesta ante fallas en los servicios.
- Reducción de problemas de hardware en los dispositivos.
- Mejora en la organización y manejo de la infraestructura tecnológica.

Se proyecta que como resultado de la operación de la herramienta se tendrán una disminución de los problemas actuales de hasta un 80%, las aplicaciones permitirán tener una infraestructura tecnológica monitoreada y predecir posibles fallas de hardware y lógicas en los dispositivos. La hipótesis se evaluará al momento de poner en un entorno productivo el sistema de información, lo cual se realizará a inicios del mes de diciembre de año 2019.

#### **4. Discusión**

El monitoreo de toda la infraestructura de red de una empresa o institución no es una tarea sencilla, es necesario contar con las herramientas adecuadas para que los administradores de sistemas puedan llevar un control y seguimiento de todos los dispositivos de red, estas herramientas permitirán el monitoreo tanto de hardware como software que son la base fundamental para el correcto funcionamiento de los productos o servicios digitales que se ofrezcan. Existen instituciones como el Tecnológico Nacional de México en Celaya que apuesta al uso de software libre para el desarrollo e implementación de sus sistemas de información, no es diferente ahora para el desarrollo de la solución de monitoreo. Con el uso del Nagios, Laravel, Git, Flutter entre otros, se ha logrado el desarrollo de una solución informática que tiene como base una infraestructura de software libre que le permitirá ser un sistema sólido, estable, seguro y escalable.

Será un reto importante el mantenimiento de la solución a través del tiempo, ya que como lo dicta la evolución de la tecnología, las necesidades y requerimientos cambian, los paquetes de software se actualizan, por lo que será necesario establecer un programa de mantenimiento continuo que permita la operabilidad del sistema.

## **5. Bibliografía y Referencias**

- [1] Flutter Documentation (2019). Getting Started with Flutter. Recuperado de: <https://flutter-es.io/docs>
- [2] GitLab Docs (2019). Docs para usuarios. Documentación oficial de GitLab. Recuperado de: <https://docs.gitlab.com/ee/user/index.html>
- [3] Guru99 (2019). Nagios Tutorial for beginners: <https://www.guru99.com/nagios-tutorial.html>.
- [4] Joskowickz José (2008). Reglas y Prácticas en eXtreme Programming. pág. 4, 5.
- [5] PostgreSQL 12 Documentation (2019). What is PostgreSQL: <https://www.postgresql.org/docs/12/intro-what-is.html>.
- [6] Quest10 Meets Answer (2018). Explain in diagram Extreme Programming [XP]: <https://bit.ly/2PMyiF6>
- [7] Ramos Marcia (2016). GitLab Workflow: An Overview, The software development lifecycle in one single interface: <https://about.gitlab.com/blog/2016/10/25/gitlab-workflow-an-overview/>.
- [8] SMITC (2019). Sistema de Monitoreo para la Infraestructura Tecnológica del TecNM Celaya.
- [9] Stack Overflow contributors (2018). Learning Laravel. FREE eBook: <https://riptutorial.com/>.