

Desarrollo de una herramienta informática para la gestión administrativa de servidores virtuales privados

RESUMEN: En este artículo, se plasma el trabajo interdisciplinario de la Maestría en Sistemas Computacionales con apoyo del CONACYT, impartida en el Instituto Tecnológico de Acapulco. El artículo tiene por meta presentar el desarrollo utilizando las diferentes tecnologías de programación web como lo es el framework ASP.NET Core y en framework Vuetify para el diseño de la aplicación web, que permita la gestión administrativa de VPS (Servidores Virtuales Privados). Se hace mención a que este artículo forma parte de una serie de trabajos siendo el tercero de estos, el cual, da seguimiento a un artículo previo titulado: Diseño de una herramienta para la gestión administrativa de servidores virtuales privados [1].

PALABRAS CLAVE: ASP .NET, Framework, MVC, SPA, UML, VPS, Vuetify

ABSTRACT: In this article, the interdisciplinary work of the Master in Computer Systems with the support of CONACYT, taught at the Technological Institute of Acapulco, is captured. The aim of the article is to present the development using the different web programming technologies such as the ASP.NET Core framework and the Vuetify framework for the design of the web application, which allows the administrative management of VPS (Virtual Private Servers). It is mentioned that this article is part of a series of works being the third of these, which follows up on a previous article entitled: Design of a tool for the administrative management of private virtual servers.

KEYWORDS: ASP .NET, Framework, MVC, SPA, UML, VPS, Vuetify



Colaboración

Ángel Rodríguez Rayo; Alma Delia de Jesús Islao; Eduardo de la Cruz Gámez; Rafael Hernández Reyna, Tecnológico Nacional de México / Instituto Tecnológico de Acapulco

INTRODUCCIÓN

En el presente artículo se aborda principalmente el desarrollo de una herramienta para la gestión administrativa de servidores virtuales (VPS, del inglés Virtual Private Server), que es una aplicación web para gestionar la información relacionada con servidores virtuales privados y los clientes responsables de dicho servicio, herramienta que notificará vía correo electrónico de forma automática cuando su periodo de pago esté por finalizar y así evitar la suspensión de su servicio. El proyecto se propone a una empresa dedicada a proveer soluciones de comunicación en el municipio de Acapulco de Juárez del estado de Guerrero, encargada de la virtualización de servidores de cómputo para el almacenamiento de datos y/u operación de aplicaciones para una gran variedad de empresas, debido a que en la actualidad es mucho más rentable que poseer un servidor físico y sus instalaciones correspondientes, con esta herramienta se busca evitar el atraso de los pagos correspondientes en los servidores virtuales privados.

Antecedentes

La organización donde propondremos nuestra herramienta es BTU, una empresa dedicada a proveer soluciones de comunicación, tanto al ramo empresarial como usuarios particulares. Estas soluciones de comunicación abarcan desde Servicios de Acceso a Internet, como Instalación y configuración de redes LAN y WAN cableadas e inalámbricas, redes privadas virtuales (VPN's), cableado estructurado (UTP y Fibra óptica), soluciones de seguridad (Firewalls, Cámaras de Vigilancia IP, Controles de Acceso) y Servidores Virtuales Privados (VPS).

La compañía tiene el reconocimiento de proveer servicios de muy buena calidad en los diferentes ámbitos que maneja, cuenta con una disponibilidad del 99.73% ya que proporciona un soporte técnico de veinticuatro horas los siete días de la semana. Sin embargo, a lo largo del tiempo que la empresa ha proporcionado estos servicios se ha presentado una problemática, muchos de sus clientes no realizan el pago de su factura en tiempo y forma.

Para la empresa esto genera problemas, principalmente de forma administrativa, y a su vez problemas de credibilidad. Genera problemas de forma económica debido a todos los gastos con los que tiene que cumplir para seguir subsistiendo.

Por estas razones la empresa al detectar que cierto número de clientes no ha realizado el pago de sus servicios notifica por correo electrónico uno por uno de los clientes de cualquier servicio que está por terminarse el plazo para realizar el pago de su factura, en la gran mayoría de las personas que son notificadas no dan respuesta alguna por lo cual se decide suspender el servicio, ya sea Servidor Virtual Privado, Cuentas de correo, Pago de Hosting Web, Pago de dominio Web, Enlaces de internet dedicados y enlaces domésticos. Para esto hay que identificar qué servicio es el que se le provee a la persona y proceder a suspenderlo.

Esta acción suele generar un impacto negativo en muchos clientes, se comunican al área de soporte para reportar el servicio y argumentando mala calidad, o mencionando por qué no se les avisó antes si se iba a realizar algún mantenimiento, diciendo que su servicio no puede estar fuera de línea que es lo que utilizan para trabajar día a día, molestos comentan que no era la primera vez que sucede y se genera un problema mucho más grande de lo que en realidad había sucedido.

A estos clientes se les notifica que han sido suspendidos debido a que no se recibió el pago de su factura, se les comenta que fueron notificados mediante correo electrónico en al menos dos ocasiones en diferentes días al área o persona encargada de realizar los pagos, se les notifica que el periodo para realizar su depósito está por finalizar y de no subsanarse el servicio se verá suspendido.

El área donde ocurre principalmente esta problemática es el la de Servidores Virtuales Privados, razón por la cual la herramienta se enfoca en ellos.

Planteamiento del problema

En BTU actualmente no se cuenta con un sistema que controle la disposición de sus servicios mediante la realización de un pago en tiempo y forma, el no contar con esta plataforma ha concedido la oportunidad para que se generen malos entendidos entre los clientes y la compañía, así como problemas administrativos dentro de la misma, viéndose obligada a utilizar recursos de otras áreas para subsanar otros gastos.

Es importante atender este problema ya que es de gran importancia para la empresa, porque la generación de este tipo de conflictos puede repercutir para el futuro, ya sea que el cliente busque otro proveedor, no contrate más servicios con la empresa o simplemente siga prolongando sus pagos. Para la empresa

es un tema de alta relevancia el cuidar su imagen y no perder clientes, ya que esto no atrae a nuevos posibles clientes.

El problema en esta empresa es que no se incorporó desde un inicio este software y no se generó esta norma desde un comienzo, probablemente por el hecho de que no es una empresa muy grande o con renombre como otras que existen en la actualidad y brindan este servicio, que cortan de inmediato el servicio al no recibir el pago acordado, el principal objetivo fue adquirir clientes y hacer crecer esta infraestructura, aún no es una infraestructura tan grande como la de otras compañías y se empiezan a generar este tipo de problemas, la razón por la cual se busca dar solución a este problema.

El problema identificado comienza desde el envío por operador de notificaciones por medio de correo electrónico, las cuales se envían faltando cinco y dos días para concluir con el límite del periodo para realizar su pago, suspender el servicio de forma manual, en donde el personal administrativo se comunica con el área de soporte técnico cada vez que desea suspender un servicio, notificación de que el equipo ha sido suspendido por falta de pago, y volver a notificar cuando el servicio ha sido reactivado.

Propuesta de solución

Por lo tanto, en el presente artículo se propone desarrollar una herramienta que ayude a notificar vía correo electrónico de forma automática a los clientes que cuenten con el servicio de servidor virtual privado, la intención fundamental de esta aplicación web es evitar llegar al nivel de suspender un servidor virtual por falta de pago y evitar malos entendidos entre la compañía y el cliente.

La herramienta propuesta es una solución que consta de tres proyectos relacionados entre sí, esto para generar capas de abstracción entre los datos, entidades y las reglas de negocio del software.

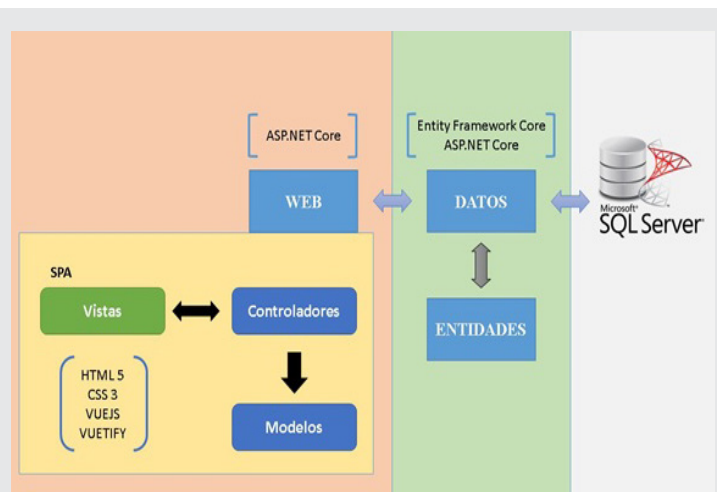


Figura 1. Arquitectura de la herramienta propuesta.

El gestor de base de datos a emplear en el desarrollo de este proyecto será SQL Server, se puede utilizar SQL Server 2008 en adelante hasta la versión más reciente, la versión que utilizaremos es la de SQL Server 2017.

En el proyecto Datos, esta será la capa que interactúe con la base de datos, aquí estarán las referencias a Entity Framework y la referencia al gestor de base de datos con el que se va a trabajar. Para nuestro proyecto utilizaremos la referencia a SQL Server.

Entidades, en esta capa se encuentran las clases que definen el modelo de datos, aquí tendremos una clase por cada tabla de la base de datos.

Por último nuestro proyecto Web, en esta capa tendremos todo nuestro proyecto web utilizando el patrón MVC. Este patrón arquitectónico separa una aplicación en tres grupos principales de componentes, que son los modelos, las vistas y los controladores, permitiendo la separación de las preocupaciones. Con este patrón las solicitudes de los usuarios se enrutan en un controlador que es responsable de trabajar con el modelo para realizar acciones de usuario y/o recuperar resultados de consultas. El controlador elige la vista que le mostrará al usuario y le proporciona los datos del modelo que requiere.

Las vistas trabajarán con SPA (Aplicación de una Sola Página, en inglés Single Page Application). La vista será una sola página web con el propósito de dar una experiencia más fluida a los usuarios, Para esto utilizaremos VueJS y Vuetify que es un framework progresivo de componentes para VueJS sobre Material Design.

Se desarrollará una herramienta web con una arquitectura robusta, crearemos nuestra solución dividida en tres proyectos. Para el backend de nuestra solución utilizaremos ASP.NET Core MVC y Entity Framework Core.

En nuestro Frontend se trabajará un SPA utilizando VueJS y el framework Vuetify, trabajaremos con el gestor de base de datos SQL Server, para la gestión de accesos y autorización utilizaremos Jason web tokens. Diseñaremos nuestra plantilla con Vuetify.

Nuestro Backend lo vamos a manejar con nuestro IDE Microsoft Visual Studio, dentro de nuestra solución tendremos el proyecto datos donde realizaremos la manipulación de datos de nuestra base de datos. Tendremos también nuestro proyecto entidades con las clases que van a representar a cada una de las tablas de la base de datos y tendremos nuestro proyecto web, donde aplicaremos el patrón MVC. Tenemos aquí los controladores, los modelos y también tenemos en este caso las vistas de nuestro proyecto. Pero las vistas las trabajaremos en este caso utilizando VueJS.

Para el Frontend implementaremos SPA utilizando VueJS, aquí tendremos todos los componentes que hacen peticiones Ajax mediante axios a nuestro backend (Todo esto para tener un código mejor ordenado).

MATERIAL Y MÉTODOS

Para desarrollar la herramienta web fue necesario el uso de las siguientes herramientas:

Enterprise Architect 12.0.1210 Version

Fiddler 4

Visual Studio Code

Visual Studio 2019

Para empezar el desarrollo se contemplan primeramente los requerimientos de sistema, y estas se anotan en el documento de requerimientos como dice [2], con base en las necesidades de la empresa y haciendo un análisis de la problemática que se requiere resolver; en este caso, para que posteriormente se modelen mediante una herramienta de lenguaje de presentación gráfica descriptiva, denominado Unified Modeling Language (UML) traducido al español como Lenguaje Unificado de Modelado con la finalidad de describir gráficamente una serie de comportamientos del sistema que puede interpretar un programador para desarrollar el sistema, estos modelos son como “los planos arquitectónicos en lo que se desarrollará el software” [2], los modelos que se desarrollaron en el presente estudio y que atienden a las necesidades de desarrollo de sistemas web [3] son: modelado de negocios, modelos de casos de uso, diagrama de clases, diagrama de entidad asociación y diagrama de paquetes desarrollados con la aplicación Enterprise Architect 12.0.1210.

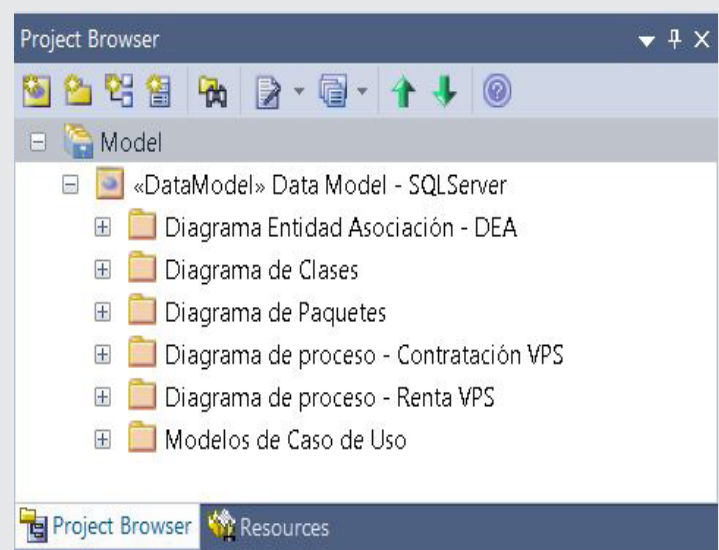


Figura 2. Modelos del sistema desarrollados en la herramienta de Enterprise Architect 12.0.1210.

RESULTADOS

Una vez considerada como concluida la etapa de modelado del sistema y herramientas elegidas, se pro-

cede a desarrollar la herramienta, una aplicación web consta de dos partes principales, el front-end y el back-end.



Figura 3. Pila de tecnologías utilizadas en el desarrollo de la aplicación web del sistema.

Para la parte de la aplicación web encargada de realizar las notificaciones vía correo electrónico a los clientes con Servidores Virtuales activos, se desarrolló la aplicación utilizando el .NET Framework con la tecnología de ASP.NET Core, utilizando el concepto de desarrollo de las SPA (Single Page Application) o aplicaciones de una sola página la cual ofrece un mayor rendimiento y su carga en el navegador es mucho más rápida, dado que ya no se hacen peticiones directamente al servidor para cargar todo el DOM Document Object Model de HTML, es decir que este concepto utiliza llamadas asíncronas mediante AJAX (Asynchronous JavaScript And XML) o basada en WebSockets para actualizar únicamente las partes necesarias en el DOM de HTML, por lo que, para su uso en aplicaciones de nueva generación que emiten información de forma automática es recomendado [4].

Para la parte de la arquitectura y diseño de la aplicación en la figura 1 se puede observar la segmentación física del proyecto, aplicando el patrón MVC (Model View Controller), la librería de Vue JS y la arquitectura de una aplicación de Single Page Application [4].

Como resultado del desarrollo de la aplicación web se tiene las siguientes interfaces gráficas del sistema.

- Página de inicio de sesión para ingresar al sistema la funcionalidad es introducir las credenciales de usuario autorizado dado de alta en la base de datos del sistema, para acceder y controlar el sistema, ver figura 4.
- Si el usuario está autenticado y autorizado, la página de inicio mostrará la vista principal del sistema, esto

es, que nos muestra el logo de la empresa BTU y una breve descripción de los servicios que proporciona, del lado izquierdo se encuentra el menú principal que consta de 5 submenús, el principal es donde se contendrá un listado de los clientes que han adquirido un servidor virtual, el siguiente submenú contendrá los listados de los servidores virtuales y los servidores físicos que son utilizados para la virtualización, seguido de este submenú se encuentra el de consumibles, estos hacen referencia a todos los complementos que pueden integrarse al servicio de servidor virtual, después procederemos a encontrarnos con el submenú accesos, en este apartado encontraremos un listado de usuarios en donde podremos registrar nuevos y modificar datos o contraseñas de los existentes, además de un listado de los roles existentes para interactuar con el sistema, por último, encontraremos el submenú notificaciones, dentro encontraremos dos apartados, el principal será donde podremos realizar las modificaciones al periodo de notificación, y en la siguiente vista podremos ver un listado de las notificaciones vía correo electrónico ya emitidas.

CONCLUSIONES

En la presentación de la funcionalidad y operación del sistema de gestión administrativa para la notificación de clientes con servidores virtuales activos al responsable del área de sistemas computacionales de la empresa BTU del municipio de Acapulco se demostró que utilizando herramientas de nueva generación para desarrollar software y utilizando patrones de diseño de software, se pueden crear herramientas a la medida de la organización de tal manera que puedan agilizar el periodo de tiempo para realizar actividades que son vitales para ellos, en este caso la notificación a los clientes por motivo de pago se redujo en un 80% y gradualmente se espera disminuir el número de servidores virtuales suspendidos.

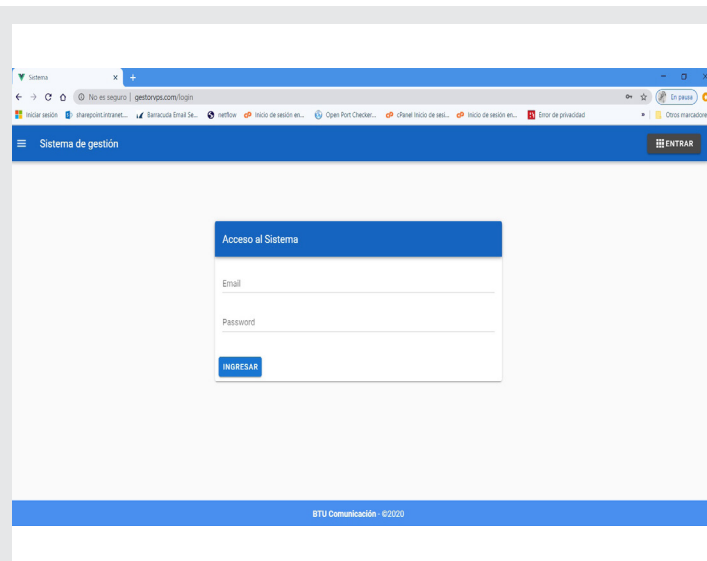


Figura 4. Página de inicio de sesión para ingresar al sistema.

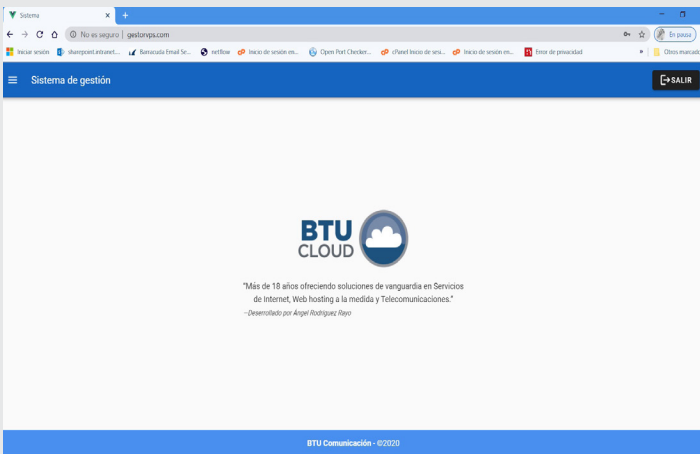


Figura 5. Página principal del sistema.

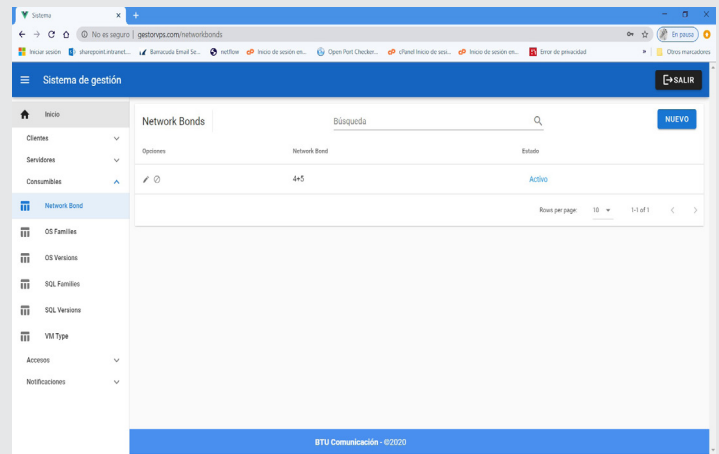


Figura 8. Submenú Consumibles, dentro de ellos se obtiene el acceso a los catálogos de network bond, sistemas operativos, versiones de sistemas operativos, gestores de base de datos disponibles, versiones de los gestores y tipo de servidores virtuales.

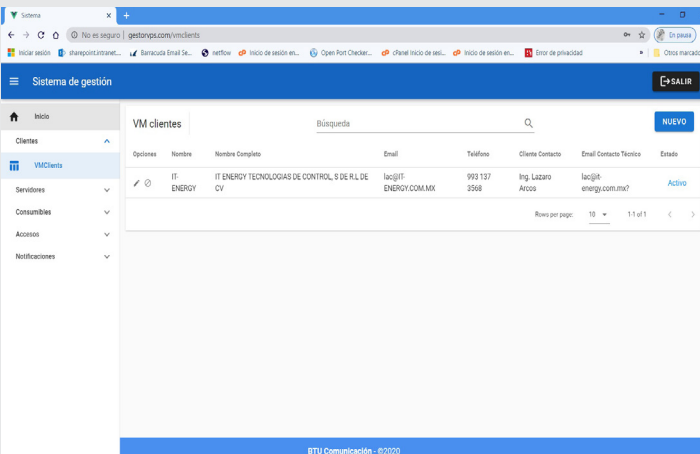


Figura 6. Submenú Clientes, vista VMList, donde se muestra el listado de clientes que adquirieron un servidor virtual privado.

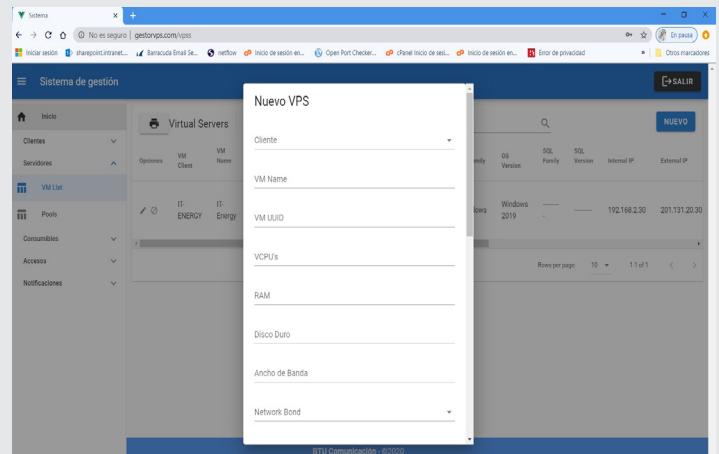


Figura 9. Página de registro de servidores virtuales, VPS.

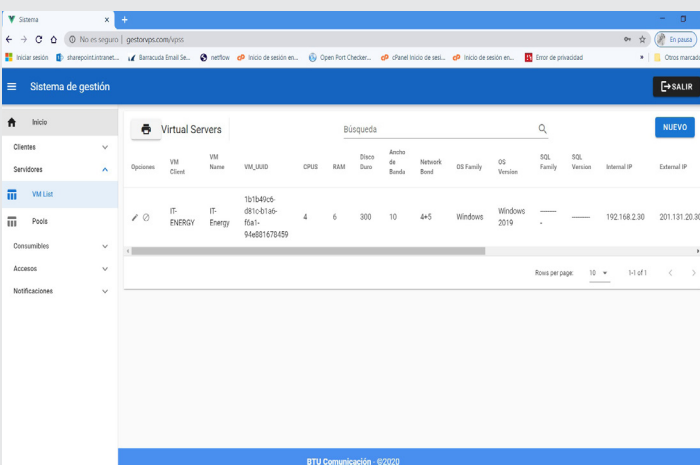


Figura 7. Submenú Servidores, vista VPS List, donde se muestra el listado de servidores virtuales privados.

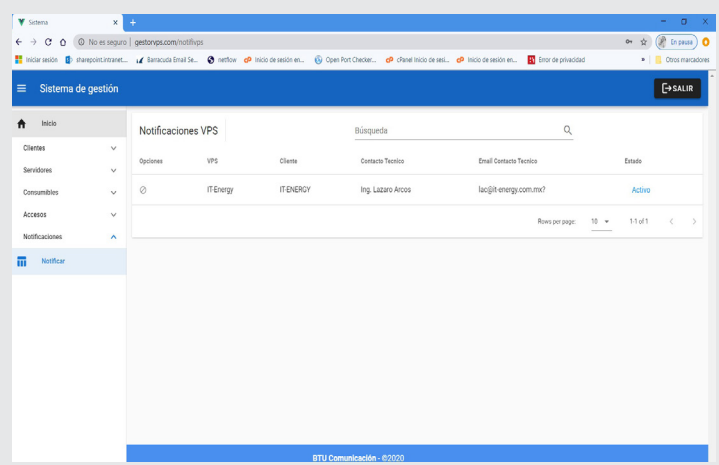


Figura 10. Listado de los servidores que serán notificados automáticamente, tras agregar un registro de VPS se agregara dicho servidor a este listado.



Figura 11. Notificación vía correo electrónico.

BIBLIOGRAFÍA

- [1] *Rodríguez Rayo, A., Cadena Mendoza, E., de la Cruz Gámez, E., & Hernández Reyna, R. (15 de agosto de 2019). Propuesta de una herramienta para la gestión administrativa de servidores virtuales privados. Academia Journals, 14, 2263-2268.*
- [2] *H. Deitel and P. Deitel, Cómo programar en C#, 2da edición. México: Pearson Educación, 2007, pp. 9-15.*
- [3] *Braun, R., & Schlieter, H. (2014). Requirements-based development of BPMN extensions: The case of clinical pathways. IEEE Xplore, 39-44.*
- [4] *E. Scott, SPA Design and Architecture Understanding single-page web applications, 1st ed. New York, EU.: Manning Publications Co., 2016, pp. 3-184.*
- [5] *Sommerville, I. (2016). Ingeniería de software. México: Pearson.*
- [6] *Tahuiton Mora, J. (2011). Arquitectura de software para aplicaciones web. Unidad Zacatenco.*