

Propuesta de una herramienta para la gestión administrativa de servidores virtuales privados

Ing. Ángel Rodríguez Rayo¹, M.T.I. Eloy Cadena Mendoza²,
Dr. Eduardo de la Cruz Gámez³ y M.T.I. Rafael Hernández Reyna⁴

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 alternativas de arquitecturas de software para el desarrollo de una aplicación que permita el control de acceso a los servidores virtuales privados y notificación vía correo electrónico del estado actual de su pago en la empresa BTU Comunicación.

Palabras clave— Arquitectura de software, Modelo-Vista-Controlador (MVC), VPS.

Introducción

BTU Comunicación es 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, renta del inmueble en donde están las oficinas y el pago de sus servicios como lo son luz y agua, enlaces empresariales de Internet, mantenimiento a los aires acondicionados del centro de datos y sistema anti incendios, licencias de Microsoft, Netflow, Monitis, Seguro médico para empleados, de la misma forma realizar los pagos de nómina para los empleados, así como material de trabajo como bobinas de cable UTP de diferentes categorías, conectores, antenas que trabajan por microondas, Routers, Switches, discos duros, material de papelería y consumibles de oficina.

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 porque no se les aviso 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.

Esta información es proporcionada principalmente a la persona que opera el servicio que no siempre es la

¹ Ing. Ángel Rodríguez Rayo es estudiante de Maestría en Sistemas Computacionales en un programa PNPC en el Instituto Tecnológico de Acapulco, ardzrayo18@gmail.com (autor correspondiente).

² M.T.I. Eloy Cadena Mendoza es docente de Maestría del Instituto Tecnológico de Acapulco, eloy_cadena@yahoo.com

³ Dr. Eduardo de la Cruz Gámez es docente de Maestría del Instituto Tecnológico de Acapulco, gamezeduardo@yahoo.com

⁴ M.T.I. Rafael Hernández Reyna es docente de Maestría del Instituto Tecnológico de Acapulco, rhernan7@yahoo.com.mx

persona que se encarga de realizar el pago, siempre de forma muy respetuosa, e inclusive el cliente sigue molesto afirmando que su departamento, tratándose de una empresa, no es el encargado de realizar los pagos, por lo general quien realiza los pagos es el área de recursos humanos o gestión de pagos. En su gran mayoría solicitan que se les reestablezca el servicio y se comunicarán con el área encargada y realizarán el pago, para esto se solicita primeramente que ellos se comuniquen con su departamento, ya que fue imposible para la empresa contactarlos, también se solicita proporcione fecha y hora que a más tardar se realizará el pago. En algunas ocasiones el cliente vuelve a contactarnos minutos después afirmando que se ha puesto en contacto con el encargado o área que realiza el pago y nos proporcionan la fecha solicitada, se confía en su palabra y se procede a reactivar el servicio para evitar prolongar el conflicto y se espera el pago de acuerdo a lo acordado, en la mayoría de las ocasiones pasa la fecha acordada y no se ha realizado el pago. Por lo cual la empresa se ve en la necesidad de suspender el servicio, regresando a la situación anterior en la que el cliente llama para reportar su servicio, ignorando el hecho de que no se ha subsanado su problema, se vuelve a notificar que no se ha recibido el pago, solicitan que el servicio está pronto a ser pagado y que se les reestablezca nuevamente el servicio. En esta ocasión ya no se reactiva, se espera se vea reflejado el pago y se procede con la reactivación y notificando mediante correo electrónico.

Este caso suele presentarse en servicios como Servidores Virtuales Privados, Cuentas de correo electrónico, y enlaces de Internet dedicados. Principalmente es una falla de comunicación entre los diferentes departamentos de la compañía cliente los cuales no notifican a otros departamentos que por tal motivo no se ha realizado el pago de la factura a tiempo, perjudicando a BTU Comunicación.

El problema es más pertinente en el área de Servidores Virtuales Privados y una de las causas que generó este problema es que este ámbito no es el principal servicio que se proporciona, es un área relativamente nueva, misma razón por la cual se les concedieron muchas facilidades a los nuevos clientes, esto con propósito de hacer crecer este sector de productividad, esto provocó que un gran número de clientes cerca del veinte por ciento se despreocuparán en el sentido de realizar los pagos con esta empresa, dejándolos quizá para el final o quitándole prioridad, otros más por el hecho de tener varios servidores rentados con la empresa toman esta actitud, esto no quiere decir que todas las empresas lo hagan de esta forma, hay clientes que tienen más de un servidor y realizan sus pagos en tiempo y forma, me parece que es la razón por la cual la empresa no ha decidido la implementación de un sistema que le permita ser más estricto con sus cobros, sin embargo la cantidad de problemáticas generadas por esta razón son muchas y desgastantes como para los empleados y directivos, además que el recibir de forma atrasada el pago de los servicios genera problemas administrativos para la empresa, esta inconsistencia se veía reflejadas en cada corte de fin de mes en un periodo dos años.

Debido a esto y a que la empresa desea crecer en este sector productivo se ve en la necesidad de adquirir un sistema que administre la disponibilidad de los servicios en relación a su pago para futuros clientes y no dependa de los empleados, se considera que si este problema se repite en varias ocasiones y se toma en cuenta que el número de clientes va en aumento este problema se generaría en una escala mayor y si a los clientes se les notifica de este sistema desde el inicio de la contratación, con el paso del tiempo los resultados se verían reflejados, la idea es recibir el pago en tiempo y forma y evitar confrontaciones por el funcionamiento del servicio y pérdida de ingresos.

De esta forma los clientes no prolongarían más la fecha para realizar su pago y sabrían que no existe otra forma para restablecer su servicio que liquidando su factura, ya no tratarían de llamar a la línea de soporte para restablecer su servicio o con los administrativos para que se los restablezca, debido a que ninguno de ellos tiene un control de eso y de esta forma evitar la prolongación de pagos actual.

Objetivo general

Desarrollar un sistema que suspenda de forma automática el servicio de Servidores Virtuales Privados en BTU Comunicación por retraso en su pago.

Objetivos específicos

- Desarrollar una herramienta dentro del sistema que automatice el envío de notificaciones a los clientes que no han realizado su respectivo pago faltando cinco y dos días antes de terminar su periodo.
- Desarrollar una herramienta dentro del sistema que permita el control (Registro, Modificación, Búsqueda y Eliminación) de clientes, servidores y contratos de servidores virtuales privados dentro de la organización.
- Suspender de forma automática los Servidores Virtuales Privados que han llegado a su fecha límite para realizar su pago.

- Llevar un control de las notificaciones relacionadas con los Servidores.

Arquitecturas de software

A partir de esta sección se describen algunas de las arquitecturas como alternativa para el diseño de software y a su vez, ayuden al desarrollo de la herramienta web propuesta.

Arquitectura cliente-servidor

El modelo Cliente/Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas. Esta arquitectura puede ser aplicada a programas que se ejecutan en una sola computadora.

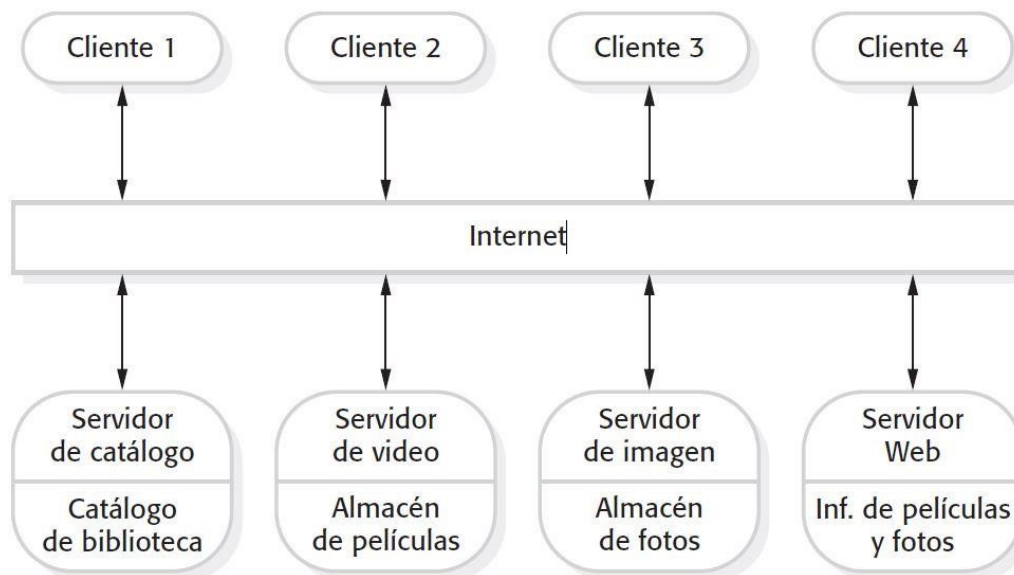


Figura 1. Arquitectura cliente-servidor (Sommerville, 2016).

Arquitectura de tres capas

Separa una aplicación en tres capas:

- **Presentación:** Su responsabilidad es la representación de datos al usuario mediante los componentes que residen en esta capa y su interacción con esta. Estos componentes permiten al usuario interactuar con los procesos de la capa de negocio de una forma segura e intuitiva.
- **Negocio:** Gestiona la lógica empresarial de la aplicación y puede acceder a los servicios de la tercera capa. En esta capa se produce la mayor parte del trabajo de los procesos de una aplicación, varios componentes de la capa de presentación pueden acceder simultáneamente a los procesos de la capa de negocio y esta debe gestionar sus propias transacciones.
- **Datos:** Esta capa solamente puede comunicarse con la capa de negocio para proveer acceso a los datos y que pueda procesarlos para mostrarlos en la capa de presentación.

A continuación, se muestra en la figura 2 un ejemplo de una arquitectura en tres capas.

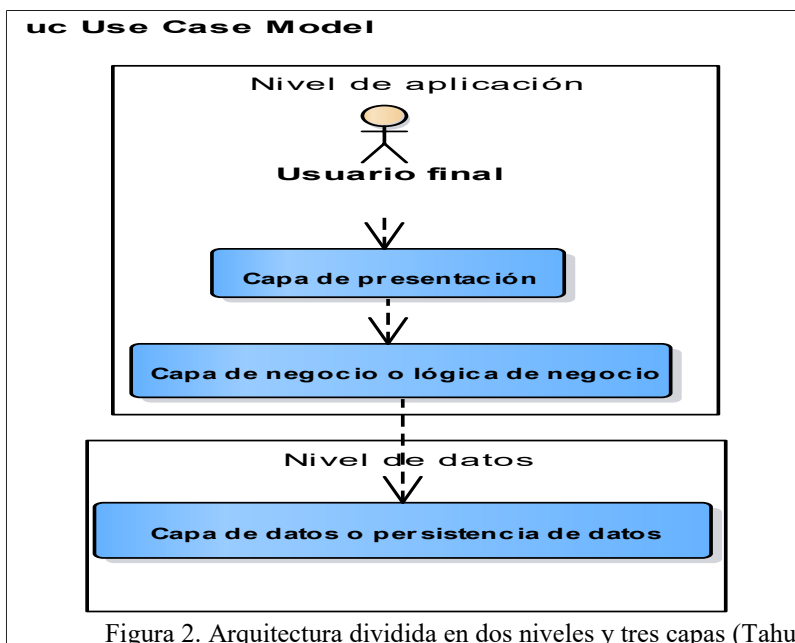


Figura 2. Arquitectura dividida en dos niveles y tres capas (Tahujon Mora, 2011).

Modelo-Vista-Controlador o MVC (Model-View-Controller)

Es un patrón de diseño de software el cual hace una separación clara de los componentes de un sistema, de modo que cada uno de estos ejecute una clase de instrucciones y que al compilarse se logren unir en la ejecución.

El patrón MVC se divide en tres capas que son importantes en una aplicación las cuales son:

- Modelo: Es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. Este no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- Vista: Es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.
- Controlador: Es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

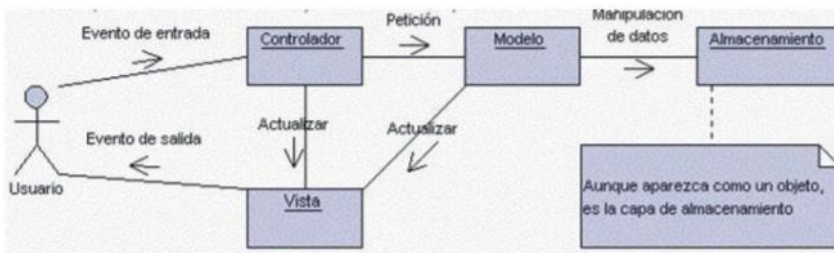


Figura 3. Interacción entre las capas de la arquitectura MVC (Fernández & Díaz, 2012)

Con las arquitecturas de software presentadas anteriormente, se puede decir que cada una de ellas está enfocada para un tipo de aplicación en específico, nuestra herramienta podría ser desarrollada en alguna de estas, ya que en la actualidad, gran parte de las herramientas digitales se componen de estos tipos de arquitecturas debido a la importancia de seleccionar los niveles en los cuales se va a trabajar. La arquitectura de software seleccionada para el desarrollo de la herramienta propuesta es Modelo-Vista-Controlador (MVC) debido a que es adaptable para el desarrollo de aplicaciones web, por tal motivo fue la arquitectura seleccionada; además, la mayoría de los frameworks existentes para el desarrollo de aplicaciones web de lado del cliente y de lado del servidor implementan

este patrón arquitectónico, otra de las grandes ventajas que ofrece es que al dar mantenimiento al código fuente se actualiza el nivel deseado, optimizando el tiempo de búsqueda y mejorando la seguridad de los métodos.

Metodología de desarrollo del software

Para el desarrollo de la herramienta propuesta la metodología a seguir será la metodología Scrum, debido a que la herramienta que se pretende desarrollar será dividida en una serie de módulos, los cuales deberán analizarse y diseñarse para poder llevar un correcto seguimiento de tiempo y a así dar paso a las pruebas unitarias, tanto en el lugar de desarrollo como también en un ambiente real, lo cual da una ventaja en el aspecto de presentar las diferentes vistas de las que estará compuesta la aplicación web. Esto sumando a la retroalimentación que existirá durante en funcionamiento de cada una de las entregas o en caso de que se presenten nuevos requerimientos, lo cual es una de las características principales de esta metodología.

Metodología ágil Scrum

Es una metodología de desarrollo ágil, que se basa en sprints: estos son iteraciones o ciclos de desarrollo cortos en los que se diseña y desarrolla un incremento del sistema. SCRUM puede usarse como base para gestión de proyectos ágiles, y puede trabajar en conjunto con otras metodologías. Esta metodología se secciona en tres fases de trabajo (Sommerville, 2016). En la primera se realiza la planeación del bosquejo y diseño de la arquitectura, se establecen los objetivos del sistema y el alcance de este. En la segunda fase se realiza una serie de iteraciones o ciclos sprint en los que se analiza, diseña y desarrolla cada uno de los módulos de los que estará conformado el sistema. Cabe mencionar que el cliente o usuario final al que pertenecerá el sistema interviene en esta fase con la finalidad de añadir nuevos requerimientos o tareas y revisar los ya desarrollados de forma que exista una retroalimentación entre desarrolladores y usuarios finales y pulir el cierre del sprint.

En la última fase se concluye el proyecto, se compacta toda la documentación generada, esto incluye manuales de usuario del sistema, manuales técnicos del sistema, documentación de apoyo, y una vez completado se entrega el sistema y sus componentes a los usuarios finales o clientes.

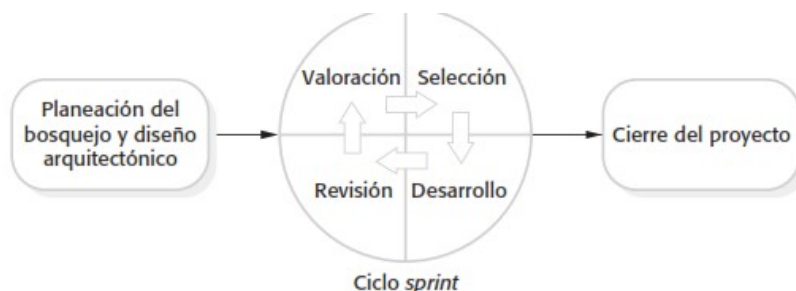


Figura 4. Proceso de un ciclo aplicando la metodología SCRUM.

Comentarios Finales

Conclusiones

En este artículo, se describe la problemática del área administrativa de la empresa BTU Comunicación. Así como también se realizó un análisis de algunas arquitecturas como alternativas para el desarrollo de la herramienta, así como la descripción de la metodología de desarrollo ágil Scrum, debido a que es una metodología adaptable a cambios en los requerimientos, ya que en el desarrollo de software los cambios en los requerimientos es algo común. Con los argumentos expuestos y posterior a su análisis, se determinó que la arquitectura de software a implementar en el desarrollo de la aplicación será MVC, debido a que la aplicación a desarrollar se trata de una aplicación web, debido a que MVC en comparación a las otras arquitecturas está enfocada en el desarrollo de aplicaciones web.

Trabajos a futuro

Este artículo se analizan tres arquitecturas de software para el desarrollo de la aplicación web, arquitectura cliente-servidor, tres capas y MVC, donde se propone utilizar la arquitectura MVC debido a que esta mayormente orientada al desarrollo web, por lo cual a trabajos a futuro se procederá a la siguiente etapa del desarrollo, el diseño de los requerimientos de la herramienta propuesta aplicando algunos diagramas UML aun cuando la metodología

Scrum no exija diagramas, estos permitirán obtener una mejor comprensión de las entidades o procesos y poder realizar el desarrollo de la herramienta con ayuda del análisis de estos diagramas.

Referencias

- Fernández, Y., & Díaz, Y. (2012). Patrón Modelo-Vista-Controlador. *Telem@tica, Vol. 11*.
- Vargas del Valle, R. J., & Maltes Granados, J. P. (s.f.). Programación en Capas. San José, Costa Rica.
- Sommerville, I. (2016). *Ingeniería de software*. México: Pearson.
- Tahuiton Mora, J. (2011). Arquitectura de software para aplicaciones web. Unidad Zacatenco.