Monitoreo de aplicaciones web

Adolfo Bravo Hernández [1], Fernando Vázquez Torres[1]

[1] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas, Sección de Estudios de posgrado, Av. Té 950, Granjas México, 08400 Ciudad de México, CDMX
ofloda@gmail.com, fvazquezt@gmail.com

Resumen. Muchas de nuestras actividades cotidianas dependen de soluciones informáticas, una buena cantidad de estas soluciones son aplicaciones web que proporcionan microempresas o bien empresas pequeñas y medianas. El trabajo realizado busca proveer a las empresas micro, pequeñas y medianas, un análisis que les permita tener un monitoreo adecuado de una aplicación web, además de la implementación de un prototipo (en progreso) con un marco ligero de los aspectos que debe contemplar un sistema de monitoreo de una aplicación web. Dando fundamentos a los equipos que se encargan de la operación en las empresas, al conocer las circunstancias o comportamientos que anteceden a una caída en el sistema, y pudiendo reaccionar de manera ágil en caso de presentarse un problema. El proceso de monitoreo debe fortalecer la memoria sobre los incidentes y estadísticas de disponibilidad que faciliten el control y las mejoras de las aplicaciones web como parte de sus sistemas.

Palabras clave: Monitoreo, aplicación, web.

Abstract. Many of our daily activities depend on computer solutions, a lot of these solutions are web applications, provided by micro-companies or small and medium-sized companies. This work out seeks to provide micro, small and medium companies an analysis that allows them to have an adequate monitoring of a web application, as well as the implementation of a prototype (in progress) with a light framework of the aspects that a monitoring system should contemplate to monitor a web application. Giving fundamentals to the teams that are in charge of the operation in the companies, when knowing what are the circumstances or behaviors that precedes fail in the system, and being able to react in an agile manner in case of presenting a problem. The monitoring process should strengthen the memory of incidents and availability of those systems that facilitate control and improvements of web applications.

Keywords: Monitor, application, web.

1 Introducción

Los sistemas web tienen cada vez mayores exigencias con lo cual los procesos de monitoreo han ido tomando relevancia, en primera instancia para saber si dichos servicios están disponibles y como siguiente paso para saber si están teniendo un buen rendimiento.

Por lo que el presente trabajo está dirigido a:

- Contar con un sistema que permita a una empresa micro o pequeña que preste alguna funcionalidad web tener cierta certidumbre que todo está disponible y, ser alertado en caso de que algo marche mal puede contribuir a que presten servicios más robustos.
- Saber cuándo las aplicaciones o sus servicios no estén disponibles o si se presenta un mal rendimiento, esto le dará la posibilidad de actuar con mayor oportunidad.
- Como parte de los procesos de las empresas además se necesita tener la relación histórica de los incidentes en el aplicativo que impacta su disponibilidad. Además de tener presente las fechas de los procesos naturales de despliegue, como actualización de contratos o de certificados.

Esto tendrá en cuenta las siguientes limitaciones:

- Se estará censando la capa más expuesta de las aplicaciones.

- No se validará la funcionalidad de las aplicaciones.
- Se considera la capa de transporte (red) como estable.

2 Estado del arte

Existen tendencias en la creación de aplicaciones que consideran la supervisión de la disponibilidad de los servicios que la soportan. Como parte de la estructura general, tal es el caso de algunas implementaciones de microservicios que utilizan herramientas como *hystrix* que proporciona el manejo de los errores que proporciona los datos a herramientas especializadas en mostrar esa información, como lo es *turbine*.
En el caso de aplicaciones pequeñas, el monitoreo normalmente se remite a la supervisión de la infraestructura que las soporta.

3 Metodología usada

Se utiliza una metodología hibrida, que consiste en los siguientes pasos:

Análisis del marco teórico general de aplicaciones web

Se dan fundamentos teóricos de las aplicaciones *web*, dando énfasis a los siguientes temas:

- Definición de es una aplicación y su arquitectura. Se presentan lo que son los patrones de diseño como partes de software que acumulan aprendizaje empírico de la construcción de aplicaciones.
- Se propone y se da una introducción del lenguaje unificado de modelado. UML (Unified Modeling Language) como código común para plasmar de manera adecuada la arquitectura de las aplicaciones.
- Panorama del uso de las 4 + 1 Vistas de Kruchten.
- Se realza la diferencia entre las capas y niveles de una aplicación de web.
- Ejemplos de arquitecturas de aplicaciones web, populares y vigentes.

Análisis de patrones de diseño principales como candidatos en la creación de la arquitectura

El trabajo realizado se dirige principalmente a micro y pequeñas empresas, por lo que se ahonda en dos patrones candidatos para la creación de un prototipo de un monitor de aplicaciones. Estos son el MVC y MVVM.

Requerimientos de usuario

Son listados y especificados los requerimientos que debe cubrir el prototipo de monitor de aplicaciones *web*, como ejemplo de lo que debe cumplir un monitor de aplicaciones dirigido a la aplicación per se.

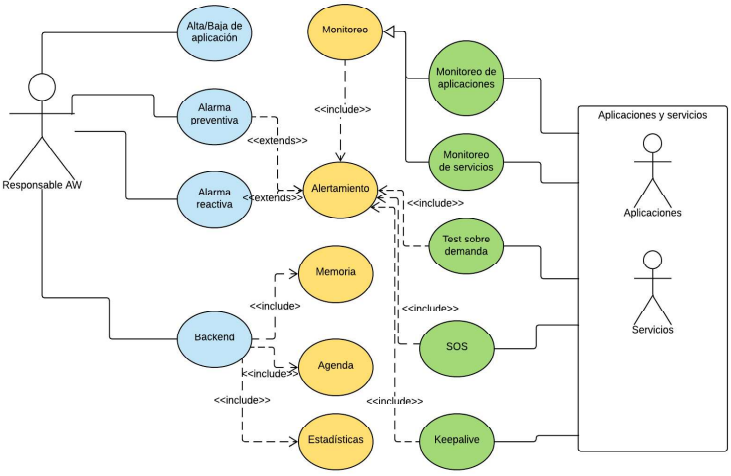
Requerimientos de sistema

Se desarrollan los requerimientos del sistema propuesto, con base en los requerimientos de usuario.

Creación de arquitectura, con base en 4 + 1 vistas de Philippe Kruchten

Son mostrados las vistas que en conjunto plasman la arquitectura propuesta para el prototipo a implementar.

El prototipo se desarrollará a la medida para poder acotar de manera precisa las funcionalidades a proporcionar.



Las tecnologías utilizadas, lo que recibe el nombre de pila tecnológica, fueron elegidas respetando el requerimiento de portabilidad propuesta en el presente trabajo

Los criterios para su elección fueron su popularidad, pues es conveniente que sea común entre los profesionales del desarrollo de software, para no generar dependencia con gente demasiado especializada. Otro criterio es que sean herramientas o lenguajes con un proceso de madurez amplio, pues suelen generar versiones estables y confiables; además claras de que cuenten con las características para cubrir lo que se requiere construir y que pudiera extender sus funcionalidades. Como parte de la descripción del prototipo se listan dichas tecnologías a continuación:

Lenguaje para la capa de presentación:

• Angular (versión 4). Marco de trabajo utilizado en la construcción de aplicaciones web, desde una página básica a una SPA, utiliza un lenguaje llamado TypeScript, es muy usando bajo el patrón MVC, pero puede usarse con cualquier patrón Modelo Vista, como el MVVM. Es una herramienta de desarrollo, robusta, confiable y de cada vez mayor difusión entre los profesionales del desarrollo de software.

Lenguajes para las capas de lógica de negocio e integración:

• NodeJS (versión 9.11.1). Basado en el lenguaje javascript (ECMAScript), es multiplataforma, y realiza ejecuciones del lado del servidor, lo cual rompió en su momento paradigmas de mantener a javascript como un lenguaje que se pudiera ocupar sólo del lado del cliente. Utiliza programación orientada a eventos.

• Java (versión 1.8.x). Lenguaje multiparadigma, mayormente usado bajo el paradigma orientado a objetos, es uno de los lenguajes más utilizado por los desarrolladores de software. Proporciona una buena capacidad de portabilidad, al poder ser ejecutado en distintas plataformas. Y cuenta con una gran cantidad de *API's*.

Base de datos:

• MySQL Community Edition (Versión 5.7.21). Una de las bases de datos con mayor difusión, según DB-Engines Ranking es la segunda más popular sólo después de Oracle. Tiene buen rendimiento. Es un producto ampliamente documentado, se puede integrar con muchas otras herramientas del mercado. Utiliza un gestor de bases de datos relacionales.

Herramientas adicionales:

• Weka (versión 3.8). Es un programa utilizado principalmente en minería de datos, fue construido en Java, cuenta con una gran cantidad de algoritmos para análisis de datos, agrupamiento y clasificación. Es un software libre con licencia GNU-GPL. Además de ocupa para el análisis de datos.

Servidor de aplicaciones:

• Apache TomEE (versión 7). Es un servidor de aplicaciones que cumple con los estándares de los servidores de java empresarial (J2EE), se basó de manera inicial en el contenedor de servlets Tomcat (Tomcat + Java EE = TomEE).- Realizar un comparativo entre las aplicaciones similares que ya existen en el mercado contra la que van a implementar, mencionando ventajas y desventajas

Costos

Si este proyecto se fuera a desarrollar por un equipo de desarrollo el costo aproximado sería de $176,000 pesos, pero si se toma como base el prototipo (el cual esta se desarrollará en un trabajo fututo), el costo sería del tiempo de configuración y despliegue, que lo puede hacer un ingeniero en sistemas en alrededor de un par de semanas. Aproximadamente $20,000 pesos, más el uso o renta recurrente de un servidor, el cual podría ser tomado como la prestación de un servicio por una microempresa externa con una renta mensual, o con los recursos que cuente la misma empresa.

Detalle de los costos, en caso de construirse por un equipo de desarrollo:

ROLES	POR DIA	DIAS	COSTO
Director de Proyecto	$3,000.00	13	$39,000.00
Analista/consultor en Datawarehouse	$1,500.00	10	$15,000.00
Consultor en Java	$1,500.00	8	$12,000.00
Desarrollador Business Objects y de ETL.	$2,000.00	30	$60,000.00
COSTO POR USO DE HARDWARE laps de consultores + 2 Micro para la oficina se rifan al final si se termina en tiempo y forma. Micro : Intel Core i5 3ra G. Pantalla : 14 pulgadas Stock : 5 Fuente: http://www.alquilerdepc.com/alquiler_de_pc_precios.php	PC		$1,800.00
COSTO OFICINA Wework, oficina por mes con todos los servicios https://www.wework.com/es-LA/l/mexico-city--DIF	2 MES 4 personas		$38,800.00

COMPRA CAÑON EPSON. SE RIFA AL FINAL DEL PROYECTO SI SE ENTREGA EN TIEMPO Y FORMA Fuente: Walmart			$9,299.00
COSTO DEL PROYECTO			$175,899.00

4 Resultados experimentales

El trabajo actualmente está en su etapa de diseño, por lo que aún no se cuentan con resultados experimentales.

5 Conclusiones y futuras líneas de investigación

La supervisión de las aplicaciones en micro y pequeñas empresas normalmente es reactiva, es decir se realiza hasta que hay un problema. Cuando llega a haber alguna supervisión proactiva de las aplicaciones, esta se remite a revisiones de la infraestructura que las soportan.

Existen productos y servicios que ayudan a monitorear aplicaciones *web*, pero normalmente dejan separados el proceso de tecnológico, de los procesos de negocio de la empresa.

Los patrones de diseño son fundamentales como elementos que acumulan experiencia y buenas prácticas, en las soluciones tecnológicas.

Se pueden tomar en cuenta módulos en el monitoreo de aplicaciones que den valor a la empresa, sin requerir de los recursos que tienen las grandes corporaciones.

Actualmente el prototipo de monitor está en etapa de diseño, por lo que los próximos pasos serán construirlo y obtener resultados experimentales los cuales serán analizados con la metodología CRISP-DM.

Bibliografía

[1] George Beekman: Introducción a la computación (Beekman, George, ed.). PEARSON PRENTICE HALL, 2004.

[2] Humberto Cervantes: Arquitectura de Software. 2010. URL https://sg.com.mx/revista/27/arquitectura-software#.WaGv4bpFzIU.

[3] Gamma.; Helm.; Johnson.; Vlissides.: Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J., ed.). PEARSON, 1994. URL http://www.ebook.de/de/product/3236753/design_patterns.html.

[4] Inc. Gartner: Hype Cycle for Application Architecture, 2017. 2017. URL https://www.gartner.com/doc/3763463/hype-cycle-application-architecture-.

[5] Grady Booch: Object-Oriented Analysis and Design with Applications. Banjamin / Cummings, 1994.

[6] James Turnbull: The Art of Monitoring. 2016.

[7] John Gossman: Introduction to Model/View/ViewModel pattern for building WPF apps. 2005. URL https://blogs.msdn.microsoft.com /johngossman /2005/10/08 /introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps/.

[8] Ludwig von Bertalanffy: Teoría general de los sistemas: fundamentos, desarrollo, aplicaciones (Ciencia y tecnologia) (Spanish Edition). Fondo de Cultura Económica, 1976.
[9] Michael Mikowski: Single Web Applications. Manning, 2013. URL http://www.ebook.de /de /product/19471259 /michael_mikowski_single_web_applications.html.
[10] Philippe Kruchten: "El Modelo de 4+1 Vistas de la Arquitectura de Software", Rational Software Corp., 1995.
[11] J. C. Tello: Patrones de diseño. 2009. URL http://www.um.es/ead/red/M10/.
[12] Yanette Díaz González, Yenisleidy Fernández Romero: "Patrón Modelo-Vista-Controlador.", Revista Telemática, pp. 47—57, 2012. URL http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15.