UNIVERSIDAD NACIONAL DEL ALTIPLANO
REVISTA DE INVESTIGACIONES DE LA ESCUELA DE POSGRADO
Versión electrónica ISSN 2077–8686
Versión Impresa ISSN 1997-4035
ARTÍCULO ORIGINAL

Vol. 9 No. 4, pp. 283-292, octubre/diciembre 2020
Recibido 10/10/2019
Aceptado 30/10/2020
Publicado 31/10/2020
DOI: https://doi.org/10.26788/riepg.v9i4.1632

ARQUITECTURA PERVASIVA CON TECNOLOGÍAS WebRTC HÍBRIDAS PARA EL DESARROLLO DE UN FRAMEWORK MODELO VISTA CONTROLADOR DE TIEMPO REAL

PERVASIVE ARCHITECTURE WITH HYBRID WebRTC TECHNOLOGIES TO DEVELOP A REAL TIME MODEL VIEW CONTROLLER FRAMEWORK

Ramiro Pedro Laura Murillo[1]
Romel P. Melgarejo Bolivar[1]

[1]Universidad Nacional del Altiplano, Instituto de Investigación en Ciencias de la Computación, Av. Sesquicentenario Nº 1153, Puno, Perú, rlaura@unap.edu.pe

RESUMEN

Las actuales arquitecturas de desarrollo potencian el desarrollo de aplicaciones web y el establecimiento de estándares parciales de desarrollo, un Framework tiene características estrictas en la nomenclatura de invocación de funciones, en lenguajes de programación como PHP ó ASP.NET, en el caso de los Frameworks más populares las llamadas internas de comprobación, etiquetado, etc. Estas acciones generan un costo de procesado sobre el navegador y teniendo en cuenta que la aplicación será accesada por un alto número de usuarios esto generaría un lapso de tiempo duplicado debido a retardos en las peticiones sobre la ejecución de funciones o métodos. Proponemos un esquema Pervasivo con tecnologías WebRTC para la mejora de la arquitectura MVC. Así establecer una variante mejorada denominada P.M.V.C. (Pervasive Model View Controller), como otros Frameworks y las mejoras en la inyección de dependencias D.I. (Dependency Injection) para flexibilizar el uso de objetos de conexión a bases de datos, modelos y vistas finales para la renderización de resultados en HTML, todo esto dentro de un controlador balanceado y organizado en sus dependencias. Se obtuvo resultados de las comparativas con el desempeño en una cantidad representativa de Frameworks MVC, finalmente las herramientas DOM para la interoperabilidad entre vista y controlador del Framework aquí propuesto.

Palabras clave: Base de datos, framework, pervasivo, reducción y tiempo-real.

ABSTRACT

The current development architectures are enhanced development of web applications and the development of partial development standards for developers, this is because a framework has strict features in the nomenclature of functional invocation, as well as the processing of programming languages such as PHP or ASP. NET, in the case of the popular frameworks, the internal check - in calls, labelling, etc. These actions generate a procedural cost of the browser and taking into account that the application will be accessed by a large number of users that would generate a duplicate time span due to delays in requests for execution of functions or methods. This is why we propose a pervasive scheme with WEBRTC technologies for improving the model development architecture viewed. Thus establish an improved variant that we will call architecture P. M. V. C. (pervasive model view controller), with a clean hierarchy in the abstraction of class obtained by comparative with other frameworks and improvements in the injection of D. I. (dependency injection) to flexibilize the use of connection objects to databases, models and final views for HTML results rendering, the support medium of real time events, all of this within a balanced and organized controller in their dependencies. Results were obtained from comparatives with performance in a representative quantity of frameworks MVC, finally increasing DOM tools of the proposed framework.

Keywords: Database, framework, pervasive, reduction and time – real.

*Autor para correspondencia: rlaura@unap.edu.pe

INTRODUCCIÓN

La aplicación de la arquitectura MVC (Modelo-vista-controlado), incluso para sistema adaptativos como propuesta de solución en la computación (Pan & Lin 2018), teniendo en cuenta propuestas de implementación con suma sencillez en un esquema balanceado (Dissanayake 2018), contribuyen en la robustez de esta arquitectura de desarrollo de software web. Teniendo al lenguaje PHP como el más difundido para el desarrollo de las arquitecturas modelo vista controlador (Wei 2009). Llegando la influencia a ser portada al lenguaje Java sobre el Framework Spring la que se ha establecido como estándar sobre este lenguaje (Singh *et al.* 2018).

El patrón MVC (Modelo-vista-controlado), se ha adoptado como una arquitectura para aplicaciones Web en los principales lenguajes de programación. Si bien, muchos marcos web comerciales y no comerciales son muy populares y se aplican ampliamente MVC (Li *et al.* 2016). Por otro lado, como una opción ligera y veloz por lo compacto de su implementación. Teniendo en suma además que las aplicaciones de las nuevas herramientas desarrolladas estarán presentes en la nube como la plataforma iECU (Lu *et al.* 2017), por su flexibilidad de ejecución y visualización sobre cualquier sistema operativo (Lu *et al.* 2017), también el trabajo sobre el streaming de audio y video enfocado en una arquitectura modelo vista controlador (Kordelas 2016).

Todo tipo de implementación se ha realizado también a la enriquecida documentación que han dejado trabajos que describen el comportamiento y comparativa entre (Jailia *et al.* 2016). El núcleo de Spring, en su "core", realiza una inyección de dependencias. Básicamente lo que significa esto es que la creación de nuestros objetos las lleva a cabo un contenedor externo inyectándolos a otros objetos que dependan de los primeros (Dandan 2013). La aplicación sobre distintas disciplinas como los deportes (Yue 2016) y la educación (Xu 2011) realizado en el trabajo presentado por (Yue 2016), permite incrementar el valor del trabajo sobre este campo, (Lu *et al.* 2017); también nos muestra un futuro alterno con aplicación MSA - aplicaciones de micro servicio el cual consiste en distribuir módulos como si fueran servidores para acciones y una vez consolidados sincronizan con el servidor principal, (Selfa 2006), todo ellos testeados sobre su robustez para administrar el tráfico de peticiones denominadas HTTP Requests (Zhang 2017).

A mediados de la década de 1990, IBM comenzó una línea de investigación que se denomina Computación pervasiva (IBM Mobile and Pervasive Computing) (Jurado 2014), el uso de esta tecnología proporciona una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos (Bormann *et al.* 2018). Sin embargo, la comunicación real (el intercambio de datos) no utiliza el sistema de ficheros, sino buffers de memoria del núcleo

(Powers *et al.* 2017). Por ejemplo, esto le permite al analizador del antivirus ClamAV ejecutar un demonio sin privilegios en Linux o BSD, aunque puede enviar cualquier archivo al socket del Daemon (Shao *et al.* 2016). Finalmente es una solución tecnológica que resultó de un esfuerzo conjunto entre la World Wide Web Consortium (W3C) y el Internet Engineering Task Force (IETF) por proporcionar comunicación en tiempo real punto a punto, a través del navegador (Johnston & Burnett 2012).

El Objetivo general de esta investigación fue: Analizar, desarrollar e implementar una arquitectura Modelo Vista Controlador Pervasivo basado en tecnologías WebRTC híbridas, como alternativa y mejora de desarrollo de aplicaciones web.

MATERIALES Y MÉTODOS

Ámbito o lugar de estudio

La investigación se llevó a cabo durante el periodo comprendido entre los meses abril a agosto del año 2019, en el área de plataforma y desarrollo del Vicerrectorado de investigación, está ubicado geográficamente en la parte sureste del Perú, entre los 13° 00´ 00" y 17° 17´ 30" de latitud sur y los 71° 06´ 57" y 68° 48´ 46" de longitud oeste del meridiano de Greenwich, en la zona norte de la ciudad de Puno, en la Av. Floral Nº 1153 - Puno, en las instalaciones del centro de cómputo del Vicerrectorado de investigación.

Descripción de métodos

Para el desarrollo del Framework Modelo Vista Controlador Pervasivo como arquitectura de desarrollo de aplicaciones web, se utilizó el tipo de investigación descriptivo que tiene un enfoque analítico, esto debido a la confrontación de la teoría con la realidad, mientras que el enfoque comparativo de las características de los otros productos, se analiza la relación que existe entre las características en el desarrollo de aplicaciones web y percepción de los usuarios finales; en Ciencias de la Computación ('Computer Science') el término "tecnologia de información" denota un objeto que se estudia son todas aquellas Tecnologías de la Información y Comunicación. Incluyen todos los equipos y aparatos desde pc hasta celulares, equipos de seguridad perimétrica como en el caso del Internet de las Cosas, así como los sistemas de vigilancia.

Arquitectura pervasiva

También se desarrollo de una arquitectura pervasiva para el desarrollo de aplicaciones web, se utilizó como diseño de investigación el cuasi experimental, debido a que los diseños cuasi experimentales manipulan al menos una variable independiente, esto con la finalidad de verificar su efecto y relación con una o más variables dependientes. Con este diseño cuasi experimental se seleccionará un grupo experimental y se someterá a un análisis comparativo entre los frameworks más destacados en el área, La recolección de datos

para el presente trabajo de investigación, se realizó mediante encuestas, porque el método consiste en obtener información de los usuarios en estudio, para nuestro caso el proyectista, el supervisor de obra y el residente de obra, que son considerados como los usuarios finales, Asu vez se realizó un análisis comparativo es un estudio profesional en laboratorio realizado por un conjunto numeroso de asociaciones de consumidores a escala sobre productos de consumo (sistemas informáticos, vehículos, alimentos, etc.) al objeto de publicar sus resultados en sus respectivas revistas de análisis comparativos, reduciendo el coste de su realización separadamente y aprovechando la gran similitud de productos, marcas y modelos en los mercados de estos países.

RESULTADOS Y DISCUSIÓN

Para comenzar en detalle con el estudio de los Frameworks web; es necesario identificar los más usados por desarrolladores de aplicaciones, así elegir los más representativos por sus características, para seleccionarlos según los criterios establecidos.

Framework Web y basado en el lenguaje PHP

recurriremos al archivo de configuración host el archivo ***.htaccess*** cuyos privilegios de ejecución están por encima de los archivos ***index.**** la modificación será (Framework web):

```
Options All -Indexes
RewriteEngine on
Options -Indexes
RewriteCond $1 !^(index\.php|robots\.txt|sitemap\.xml|wroot)
RewriteRule ^(.*)$ index.php?url=$1 [L]
```

Seguidamente se pasa el control al archivo index de forma manual pudiéndose cambiar si se requiere, solo por convención mantenemos el archivo (Tabla 1).

Tabla 1. Indicadores de tiempo y versiones en el mercado.

AngularJS	Se encuentra en el mercado desde el 2009, mantenido por Google, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC). Teniendo las siguientes versiones en el mercado: • ANGULARJS 1.0; 1.1; 1.2; Su última versión ANGULARJS 1.3.15 a partir de 17 de Marzo 2015.

Laravel	Se encuentra en el mercado desde el 2011, desarrollado por Taylor Owell, requiere PHP 5. Para desarrollo de aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC). Teniendo las siguientes versiones en el mercado: • Laravel 5.7; 5.6; 1.2; con la versión 5.9 LTS pronta a salir a fines de 2019.
Symfony	Symfony es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Teniendo las siguientes versiones en el mercado: • 2003 el inicio, 2007 la v 1.0, en 2011 la v 2.0, el 2017 la v. 4.0 y en 2019 la v. 4.4.
ASP.NET Core	ASP.NET Core es un nuevo framework de código abierto y multiplataforma para la creación de aplicaciones modernas conectadas a Internet, como aplicaciones web y APIs Web. • .Net Core 1.0 lanzado en 2013, .NET Core 2.0 lanzado 2015 y actualmente en la versión 2.2, existe un preview .NET Core 3.0, lanzamiento para el 2020 con SignalR

Las direcciones físicas o de locación URL serán reemplazadas dentro del nucleo del framework para mantener el enrutamiento de forma limpia, además de asegurar una limpieza contra ataques XSS basadas en la inyección de código malicioso, una vez interceptado la línea de pedido URL este podrá ser reemplazado internamente así poder operar dentro de los parámetros del Framework. Todo tipo de implementación se ha realizado también a la enriquecida documentación que han dejado trabajos que describen el comportamiento y comparativa entre (Jailia *et al.* 2016). El núcleo de Spring, en su "core", realiza una inyección de dependencias. Básicamente lo que significa esto es que la creación de nuestros objetos las lleva a cabo un contenedor externo inyectándolos a otros objetos que dependan de los primeros (Dandan 2013). La aplicación sobre distintas disciplinas como los deportes

(Yue 2016) y la educación (Xu 2011) realizado en el trabajo presentado por (Yue 2016), permite incrementar el valor del trabajo sobre este campo (Lu *et al.* 2017).

Sockets UNIX nativos

Nuestra implementación tenemos el esquema igual a un programa en lenguaje C, para la creación de una instancia PIPE y Socket en modo flujo STREAM bajo el protocolo TCP (Transmision Control Protocol), seguidamente preparación del puerto escucha y para la optimización de los clientes conectados se implementa una lista de enlace simple, para recorrer con una iteración ***foreach*** en lugar de un iterador ***for*** (Figura 1).

No se está disponiendo del modo SOCK_DGRAM que está bajo el protocolo UDP (User Datagram Protocol) generalmente usado para video streaming, por ello no fue implementado en nuestro servidor (Figura 1).

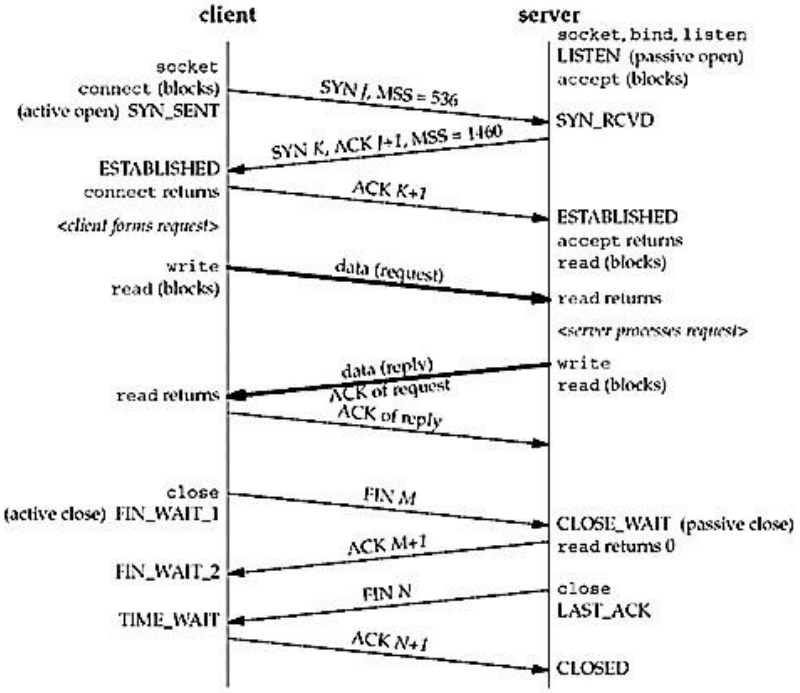

Figura 1. Intercomunicación entre Sockets una vez establecido la negociación.

Fuente: obtenido de notes.sichao.io/unp/ch05

La Intercomunicación entre Sockets resultado de la figura afirma lo indicado en la línea de investigación que se denomina Computación pervasiva (IBM Mobile and Pervasive Computing) (Jurado 2014), el uso de esta tecnología proporciona una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos (Bormann *et al.* 2018). Sin embargo, la comunicación real (el intercambio de datos) no utiliza el sistema de ficheros, sino buffers de memoria del núcleo (Powers *et al.* 2017), por ejemplo, esto le permite al analizador del antivirus ClamAV ejecutar un demonio sin

privilegios en Linux o BSD, aunque puede enviar cualquier archivo al socket del Daemon (Shao *et al.* 2016). Finalmente es una solución tecnológica que resultó de un esfuerzo conjunto entre la World Wide Web Consortium (W3C) y el Internet Engineering Task Force (IETF) por proporcionar comunicación en tiempo real punto a punto, a través del navegador (Johnston & Burnett 2012).

CONCLUSIONES

Las arquitecturas Modelo Vista Controlador MVC clásicas requieren en el caso de Laravel y CakePHP la compilación y modificación de la configuración del servidor apache para la completa integración de sus componentes, asu vez se cumple con implementar manualmente las rutinas de configuración e inicio de sockets UNIX nativos para una aceleración en el procesamiento de las peticiones o HTPP Requests así procesar inmediamente y de ser necesario realizar la transmisión dirigida a las aplicaciones clientes (Broadcasting), esto debido que componentes existentes como WebRTC y SocketJS duplicaría el encaminado de las peticiones. Se hace hincapié en el uso de socket nativos como base de la comunicación por la sencillez de implementación final después de una depuración realizada durante varias iteraciones y depuración luego agregar el protocolo de negociación (handshake) para la conversión y uniformización de datos y la interoperabilidad entre sockets UNIX y los WebSockets de los navegadores web al tener en cuenta que el protocolo HTTP no procesado datos binarios en ráfaga o RAWs sino su contenido se desplaza en un contenido de texto y para los datos que requieren datos binarios estos son codificados en base 64 para la trasmisión.

CONFLICTO DE INTERÉS

Los autores, no tiene conflictos de ninguna índole.

REFERENCIAS

Bormann C., Lemay S., Tschofenig H., Hartke K., Silverajan B., Raymor B. 2018. CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets (No. RFC 8323). Universitaet Bremen TZI, Standards Track Zebra Technologies. https://tools.ietf.org/html/rfc8323

Coulouris G., Dollimore J., Kindberg T. Blair G. 2013. Sistemas Distribuídos - Conceitos e Projeto. Bookman Editora. http://www.inf.ufsc.br/~bosco.sobral/ensino/ine5645/coulouris.pdf

Dandan Y. 2013. Research on Lightweight MVC Framework Based on Spring MVC and Mybatis. *Sixth International Symposium on Computational Intelligence and Design* 2013. 350-353. IEEE. doi: 10.1109/iscid.2013.94

Dissanayake R., Dias K. 2018. Balanced Abstract Web-MVC Style: An Abstract MVC Implementation for Web-based Applications. GSTF Journal on Computing (JoC), 5(3). https://www.researchgate.net/publication/317063763_Balanced_Abstract_Web-MVC_Style_An_Abstract_MVC_Implementation_for_Web-based_Applications

Galloway J., Wilson B., Allen S., Haack P. 2012. Professional ASP.NET MVC 4. *Professional*. doi: 10.1017/CBO9781107415324.004

Grozev B., Politis G., Ivov E., Noel T., Singh V. 2017. Experimental evaluation of simulcast for WebRTC. *IEEE Communications Standards Magazine*, 1(2), 52-59. https://ieeexplore.ieee.org/document/7992929

González D., Romero F. 2012. Patrón Modelo-Vista-Controlador. *Revista Telemática*, 11(1), 47-57.

Hernández T. 2015. Symfony Framework. Desarrollo Rápido de Aplicaciones Web. IT Campus Academy. https://www.amazon.co.uk/Symfony-Framework-Desarrollo-Aplicaciones-Edici%C3%B3n/dp/1540420027

Hernández R., Fernández C., Baptista P. 2010. Metodología de la investigación (3ra ed). México: McGraw-Hill. https://www.casadellibro.com/libro-metodologia-de-la-investigacion-3-ed-incluye-cd-rom/9789701036327/866740

Jailia M., Kumar A., Agarwal M., Sinha I. 2016. Behavior of MVC (Model View Controller) based Web Application developed in PHP and. NET framework. In 2016 International Conference on ICT in Business Industry & Government (ICTBIG) (pp. 1-5). *IEEE*. doi: 10.1109/ICTBIG.2016.7892651

Johnston B., Burnett C. 2012. WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web. Digital Codex LLC. https://webrtcbook.com/excerpt/excerpt.pdf

Jurado A. 2014. Arquitecturas de Computación Pervasiva Basadas en Servicios REST. Universidad Politécnica de Madrid, Ingeniería de Redes y Servicios Telemáticos. España. http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2013-2014/TFM_Luis_Alberto_Jurado_2014.pdf

Kordelas A., Politis I., Dagiuklas T. 2015. Transport analysis and quality evaluation of MVC video streaming. Multimedia Tools and Applications. *ResearchGate*. doi: 10.1007/s11042-015-2530-8

Krasner E., Stephen P. 1988. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. The JOT (SIGS Publications). Also published as "A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System" (Report), ParcPlace Systems; Retrieved 2012-06-05. https://edisciplinas.usp.br/mod/resource/view.php?id=1727451

Krumm J., Bardram J., Bernheim B. 2010. Ubiquitous Computing Fundamentals, C R C Press LLC, ISBN 978-1-4200-9360-5. http://sociotech.pbworks.com/f/UbiquitousComputFundamen.pdf

Li X., Liu N. 2016. Research on L-MVC Framework. In 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (P.D.C.A.T.), (pp. 151-154). *IEEE*. doi: 10.1109/PDCAT.2016.043

Lu Y., Liu W., Cui H. 2018. MSA vs. MVC: Future Trends for Big Data Processing Platforms. In: Qiu M. (eds) Smart Computing and Communication. SmartCom 2017. Lecture Notes in Computer Science, vol 10699. Springer, Cham https://www.hpe.com/us/en/resources/solutions/mapr-data-platform.html?chatsrc=ot-en&jumpid=ps_hb5pv83s76_aid-520042861&gclid=Cj0KCQiAhZT9BRDmARIsAN2E-J1AuOtb0PyLCZ-Qq0wlOoeuaKq-iGEOTBO80bSfO33FvX_KhlpgKwEaAvpZEALw_wcB&gclsrc=aw.ds

Lu C., Zeng, J., Zeng Y., Shi T., Zhang, Y., Guo Y. 2017. Research and design of APP for new energy vehicles electronic control system based on cloud platform. In 2017 6th International Conference on Computer Science and Network Technology (ICCSNT) (pp. 179-183). *IEEE*. DOI: 10.1007/978-3-319-73830-7_31

Microsoft. 2014. ASP.NET MVC Overview. Microsoft Developer Network. https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started

Pan C., Lin C. 2018. Designing and implementing a computerized adaptive testing system with an MVC framework: A case study of the IEEE floating-point standard. In 2018 IEEE International Conference on Applied System Invention (ICASI) (pp. 609-612). *IEEE*. doi: 10.1109/ICASI.2018.8394328

Powers B., Vilk J., Berger D. 2017. Browsix: Bridging the Gap Between Unix and the Browser. ACM SIGOPS Operating Systems Review, 51(2), 253-266. https://www.researchgate.net/publication/316898647_Browsix_Bridging_the_Gap_Between_Unix_and_the_Browser

Reenskaug T. 1978 MVC Pattern XEROX PARC 1978-79. http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html

Sanchez L., Althmann, F. 2014. Desenvolvimento web com ASP.NET MVC. Casa Do Código. https://www.casadocodigo.com.br/products/livro-aspnet-mvc

Selfa D. M., Carrillo M., Boone D. R. 2006. A database and web application based on MVC architecture. In 16th International Conference on Electronics, Communications and Computers (CONIELECOMP'06). *IEEE*. doi: 10.1109/CONIELECOMP.2006.6

Sinha S. 2017. A CRUD Application. In Beginning Laravel. Apress, Berkeley, CA. 67-79. https://www.apress.com/gp/book/9781484225387

Singh A., Chawla P., Singh K., Singh A. K. 2018. Formulating an MVC Framework for Web Development in JAVA. In 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI) 926-929. IEEE. https://ieeexplore.ieee.org/abstract/document/8553746/

Shao Y., Ott J., Jia Y. J., Qian Z., Mao Z. M. 2016. The misuse of android unix domain sockets and security implications. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 80-91). ACM. https://dl.acm.org/doi/10.1145/2976749.2978297

Stroustrup B. 2013. C++ Programming Language. Addison-Wesley. ISBN 0-321-56384-0 https://www.stroustrup.com/4th.html

Wei C., Lin H., LiJing L., Jing L. 2009. The Research of PHP Development Framework Based on MVC Pattern. 2009, Fourth International Conference on Computer Sciences and Convergence Information Technology. ICCIT 2009. https://ieeexplore.ieee.org/document/5369976

Xu S., Yang T. 2011. Application of Struts framework based on MVC in Online Countryside Teachers' Training System in China. In 2011 International Conference on Multimedia Technology (pp. 6252-6255). IEEE. https://ieeexplore.ieee.org/document/6001821/

Yue J., Ye Y., Wei Z., Li Z. 2016. The design and implementation of national traditional sports professional teaching resources platform based on MVC. In 2016 International Conference on

Intelligent Transportation, Big Data & Smart City (ICITBS) (pp. 277-280). *IEEE*. doi: 10.1109/ICITBS.2016.111

Zhang S., Liu Z. 2017. Research on the construction and robustness testing of SaaS cloud computing data center based on the MVC design pattern. In 2017 International Conference on Inventive Systems and Control (ICISC) (pp. 1-4). *IEEE*. doi: 10.1109/ICISC.2017.8068723