

Pizarra Virtual Compartida Websockets; una solución para trabajo remoto colaborativo

(Websockets Shared Virtual Whiteboard; an alternative for remote collaborative work)

Xavier Sotomayor¹, Diego Ordóñez Camacho¹

Resumen:

La Pizarra Virtual Compartida Websockets, PVCW, es un sistema para trabajo compartido que habilita a sus usuarios a colaborar de manera remota, en línea y en tiempo real. El sistema proporciona una pizarra virtual que actúa como un espacio de trabajo en el cual las interacciones de cualquier participante se sincronizan automática e inmediatamente con los demás. En el lado servidor el sistema se implementa mediante la Java Enterprise Edition utilizando la tecnología WebSockets. En el lado cliente no es necesario efectuar ninguno tipo de instalación dado que cualquier browser de nueva generación que soporte WebSockets mediante Javascript, sea en dispositivos de escritorio o móviles, es capaz de integrarse a una sesión de trabajo. Esta configuración permite a los usuarios, dibujar a mano alzada, utilizar figuras geométricas de base y cuadros de texto; pueden guardar una pizarra, cargarla nuevamente en cualquier momento y editar su contenido; adicionalmente se habilita una herramienta de chat que brinda un espacio complementario para añadir explicaciones contextuales a la sesión, todo en un entorno colaborativo.

Palabras clave: pizarra virtual compartida; websockets; trabajo en línea.

Abstract:

The Websockets Shared Virtual Whiteboard, is a system for cooperative work, that allows its users to collaborate remotely, online and in real time. The system provides with a virtual whiteboard acting as a shared workspace where any user's interactions synchronize automatically and immediately with all the other users. Server side, the system is implemented with the Java Enterprise Edition, mainly using WebSockets technology. Client side, no installing process is necessary, given that any recent browser supporting WebSockets through Javascript, whether on desktop or mobile systems, is able to integrate itself into a working session. The chosen configuration allows the users to hand-draw, and use geometric figures and text boxes; it is possible to save any stage of the working environment to file and reload it for a different session; moreover, a chat tool has been incorporated, which provides with a complementary workspace for adding contextual explanations to the session, everything in a collaborative environment.

Keywords: virtual whiteboard; websockets; shared whiteboard; online workspace.

¹ Universidad Tecnológica Equinoccial, Quito – Ecuador ({sboxa87133, dordonez} @ute.edu.ec)

1. Introducción

La Pizarra Virtual Compartida Websockets, PVCW, es un sistema para trabajo colaborativo que habilita a sus usuarios a colaborar de manera remota, en línea y en tiempo real. El sistema proporciona una pizarra virtual que actúa como un espacio de trabajo en el cual las interacciones de cualquier participante se sincronizan automática e inmediatamente con los demás. En el lado servidor el sistema se implementa mediante la Java Enterprise Edition utilizando la tecnología WebSockets. En el lado cliente no es necesario efectuar ninguno tipo de instalación dado que cualquier browser de nueva generación que soporte WebSockets mediante Javascript, sea en dispositivos de escritorio o móviles, es capaz de integrarse a una sesión de trabajo. Esta configuración permite a los usuarios, dibujar a mano alzada, utilizar figuras geométricas de base y cuadros de texto; pueden guardar una pizarra, cargarla nuevamente en cualquier momento y editar su contenido; adicionalmente se habilita una herramienta de chat que brinda un espacio complementario para añadir explicaciones contextuales a la sesión, todo en un entorno colaborativo.

El trabajo colaborativo es imprescindible en múltiples disciplinas y escenarios, por ejemplo en diseño y desarrollo de software, especialmente cuando el problema no está correctamente definido, o cuando se debe identificar las necesidades del usuario o del cliente. La colaboración frente a la pizarra es un instrumento sumamente valioso para mantener el foco en el problema (Kleffmann, Hesenius, & Gruhn, 2015; Rooksby & Ikeya, 2012). Esta colaboración, sin embargo, no siempre es factible debido a la imposibilidad de juntar a los involucrados en el mismo espacio físico al mismo tiempo.

En la actualidad, ventajosamente, el desarrollo de las tecnologías de red amplía la gama de alternativas para poder generar sistemas y aplicaciones que aporten en los procesos colaborativos de comunicación y educación (Hou & Xia, 2009). De la misma manera, varios emprendimientos apuntan al desarrollo de metodologías para mejorar el trabajo colaborativo en grupo así como el diseño del aprendizaje, mediante el uso de herramientas computarizadas gráficas de alto nivel (Mota, Carvalho, & Reis, 2011).

La línea de acción que se perfila va de la mano con las nociones de construcción de equipos de trabajo, las cuales manifiestan que en un proyecto la visión compartida del objetivo es central, y que esta visión se soporta también en el aprovechamiento de la descripción gráfica y su mejora en un ambiente de aporte colaborativo y creativo (Dalsgaard, Halskov, Mackay, Maiden, & Martens, 2015; Russell & Creighton, 2011).

Más aún, no solamente equipos profesionales y de alto nivel se benefician de estos espacios de trabajo virtuales compartidos, sino que en la misma aula de clases el uso de estas tecnologías mejora los niveles de motivación al proporcionar, ente otras cosas, un medio actual y divertido que enriquece la experiencia de aprendizaje (Hamdan, Al-Qirim, & Asmar, 2012).

La influencia y relevancia de los dispositivos móviles dentro de los procesos de trabajo colaborativos y de enseñanza-aprendizaje también debe ser considerada cuidadosamente. Los dispositivos móviles abren la posibilidad de trabajo desde cualquier lugar, sin embargo esta misma circunstancia atenta contra el trabajo en grupo dificultando la posibilidad de coincidir en hora y lugar. Hay entonces que explotar las ventajas de la ubicuidad y tomarla en cuenta al diseñar y desarrollar sistemas para espacios de trabajo compartido (Jaramillo, Nguyen, & Newhook, 2014; Limtrairut, Marshall, & Andreae, 2016).

1.1 Trabajos relacionados

Varias herramientas existen que de una manera u otra intentan brindar una solución a la problemática general tratada en este artículo. Todas ellas con su enfoque particular y su especificidad.

En el campo de las pizarras interactivas educativas tenemos Virtual Whiteboard (Lech, Kostek, & Czyzewski, 2012) e ImmerseBoard (Higuchi, Chen, Chou, Zhang, & Liu, 2015) que son herramientas sobre todo pensadas como reemplazo a las pizarras electrónicas usadas en el aula, con las cuales se puede interactuar completamente mediante gestos con las manos, gracias a las cámaras de captura de video que utilizan estos sistemas para detectar los gestos; esto sin embargo conlleva una configuración de hardware más especializada que limita su uso en espacios de trabajo más sencillos o estándar.

Los dispositivos móviles tienen su propia gama de aplicaciones que en general aprovechan bien el hardware, sobre todo la pantalla táctil, pero que asimismo suelen restringir la colaboración a dispositivos con igual sistema operativo, e incluso igual versión en ciertos casos. Un ejemplo es AegenBoard (Zhang & Yu, 2014) que además de la pizarra compartida incluye un sistema de voice over IP y permite grabar la sesión. Solo funciona en el iPad y la comunicación está limitada a dos personas. Para el sistema Android existe por ejemplo FlexiSketch (Wüest, Seyff, & Glinz, 2015) la cual está orientada sobre todo a la generación de diagramas, para lo cual presenta una amplia gama de opciones. Para Windows exclusivamente hay también alternativas como Easy Whiteboard (Brinkmann, 2015), bien provisto con opciones para dibujo aunque muy limitado para compartir y en alemán, y Shared Whiteboard (Akison, 2014) que explota las posibilidades de colaboración pero se enfoca en dibujo a mano alzada para pantallas táctiles.

Los sistemas que globalmente se muestran más flexibles son aquellos basados en tecnologías web, los cuales en general permiten colaborar en tiempo real entre varios usuarios y proporcionan opciones similares en lo relacionado al dibujo en la pizarra, como mano alzada con variedad de colores y grosores, y pequeñas notas de texto. Asimismo, la gran mayoría de las alternativas entregan funcionalidad limitada de manera gratuita, pero para un uso total es necesaria una suscripción pagada. Como ejemplos representativos tenemos A Web Whiteboard (Good Code,

2016) que aunque interactúa bien con dispositivos móviles, no posee trazado de figuras prediseñadas y no permite editar la información al compartirla. Muy similar, incluso en nombre, Web Whiteboard (Kniberg, 2016) permite desplazar las notas de texto creadas y guardar la pizarra, aunque esto solo como bitmap. Twiddla (Twiddla, 2016) si bien es pagado brinda la posibilidad de un acceso libre, pero totalmente público, lo cual impide un trabajo confidencial. Permite dibujar a mano alzada, a cada participante le brinda un color predeterminado. Scribblar (Scribblar, 2016) se presenta como bastante completo en cuanto a funcionalidad para trabajo colaborativo. Para su uso es imprescindible crear una cuenta, con un máximo de dos semanas de período de prueba gratuita; adicionalmente requiere flash, lo cual en dispositivos Android es un limitante por la falta de soporte oficial para este complemento.

A continuación el artículo se encuentra dividido de la siguiente manera: la sección 2 detalla los métodos, tecnologías y herramientas utilizadas; en la sección 3 se presentan los detalles del sistema resultante, considerando funcionalidad, diseño y pruebas; la sección 4 entabla una discusión que intenta comparar el sistema presentado con aquellos existentes destacando los puntos fuertes y débiles relevantes; finalmente la sección 5 destaca las conclusiones más importantes así como ciertas recomendaciones de trabajo a futuro.

2. Metodología

Los requisitos fundamentales para el sistema se obtuvieron mediante focus-groups y lluvia de ideas con usuarios tipo. Gracias al análisis de los trabajos relacionados, y hasta donde se pudo observar, se estableció que no existía una herramienta ya desarrollada que cumpla con la funcionalidad esperada, y se pudo afinar los requerimientos del sistema a construir tanto en lo funcional cuanto en lo relacionado a tecnologías y arquitectura.

Se decidió desarrollar un sistema basado en web que brinde independencia de arquitectura del lado cliente; se utilizó WebSockets (Fette & Melnikov, 2011) para simplificar el mantenimiento de la sesión y la conexión sobre HTTP, así como el envío de información en la forma de mensajes.

El front-end se desarrolló en HTML5, CSS y JavaScript para brindar la mejor compatibilidad con WebSockets (Wang, Salim, & Moskovits, 2012) y para evitar el uso de plugins.

Para el back-end se decidió trabajar con JEE y Glassfish utilizando la implementación Tyrus para WebSockets, combinación sólida para aplicaciones empresariales de alto rendimiento (Heffelfinger, 2014).

Ya en los específicos de la funcionalidad del sistema, se creó una interfaz gráfica genérica para cualquier browser compatible con WebSockets, y se creó un protocolo de aplicación propio de comunicación de mensajes estructurados en formato Json. Como se muestra en la *Figura 1*, las opciones más relevantes permiten el dibujar con el mouse, dedo o stylus si la pantalla es táctil,

pudiendo hacerlo a mano alzada, o seleccionando gráficos ya establecidos como las figuras geométricas; es posible seleccionar distintos colores y por último se implementó el almacenamiento del estado de las pizarras en archivos, del lado cliente, para su posterior recuperación y edición en una nueva sesión de trabajo.

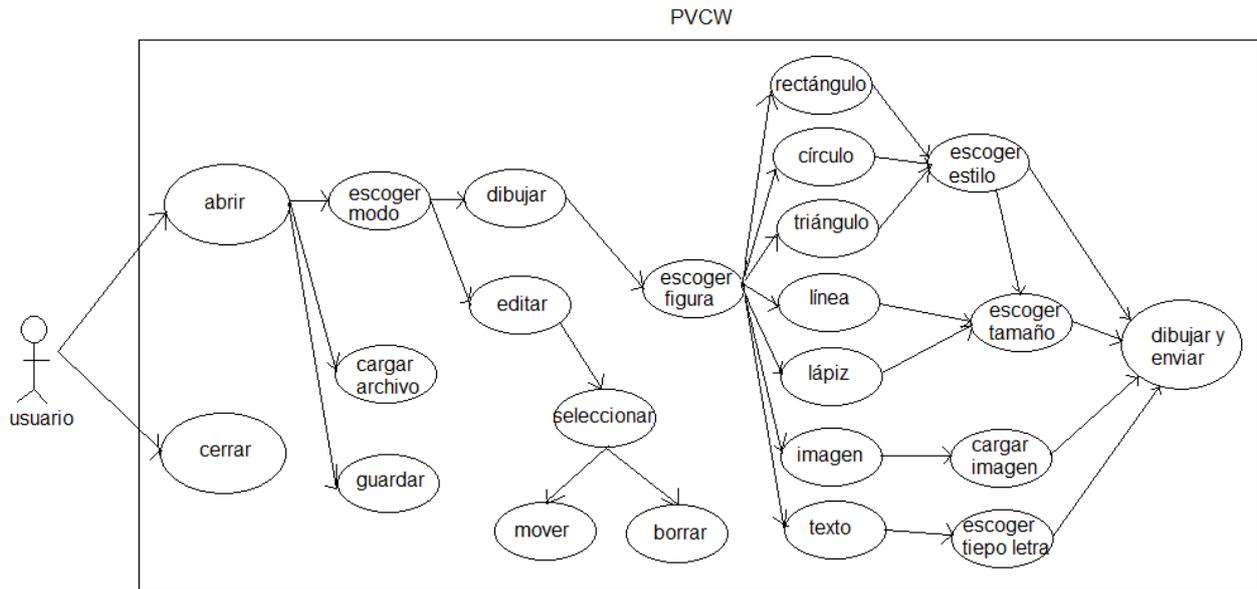


Figura 1. Diseño estructura interna PVCW

Se probó la funcionalidad de la aplicación en Google Chrome 52, Mozilla Firefox 43, Internet Explorer 11, Microsoft Edge 25 y Opera 39 con al menos dos usuarios conectados al mismo tiempo por cada uno; ejecutándose sets de pruebas tanto de caja blanca como de caja negra.

3. Resultados

El sistema se denominó Pizarra Virtual Compartida WebSockets, PVCW. La *Figura 2* presenta la interfaz gráfica de la aplicación con la implementación y uso de controles, donde, agrupado y marcado con letras en rojo, se tiene lo siguiente: A) tres controles para: abrir conexión, cerrar conexión y guardar como archivo el estado de la pizarra. B) figuras geométricas prediseñadas como: rectángulo, triángulo, línea recta, círculo, texto, lápiz y carga de imágenes. C) doce tamaños de dibujo definidos para el lápiz, línea recta, borrador y el borde de una figura geométrica sin relleno. D) doce tipos de letra definidos para el texto, los cuales son: Arial, Arial Black, Comic Sans, Courier, Cursive, Fantasy, Georgia, Helvética, Lucida Sans, Monospace, Times New Roman y Verdana. E) opción para dibujar un cuadrado, círculo o triángulo con relleno o sin relleno. F) campo de texto en donde se insertará el valor de la opción texto a dibujar. G) dos modos, dibujar y editar los cuales ayudan a identificar el momento en que se desea cambiar de posición o eliminar una figura geométrica o texto ya dibujados. H) doce colores definidos para todas las figuras geométricas a dibujar, texto y lápiz, estos son: verde, azul, rojo, amarillo, celeste, verde claro, tomate, negro, rosado, café, morado y gris. J) borrador, el cual solo puede borrar lo dibujado con el lápiz, las figuras y el texto se borran con la tecla “suprimir” en el modo “editar”. K) opción para

poder leer las instrucciones y usar de manera correcta la pizarra. Como parte del diseño de figuras geométricas, texto y lápiz se tiene lo siguiente: L) se pudo efectuar trazos de lápiz de diferente grosor y color. M) se pudo efectuar trazos de línea recta de diferente grosor y color. N) se pudo efectuar trazos de círculo con relleno, sin relleno, de diferente tamaño de borde y color. O) se pudo efectuar trazos de cuadrado con relleno, sin relleno, de diferente tamaño de borde y color. P) se pudo efectuar trazos de triangulo con relleno, sin relleno, de diferente tamaño de borde y color. Q) se pudo efectuar trazos de diferentes tipos de letra de texto a dibujar, color y tamaño. A excepción del lápiz, todas las figuras y el texto se pueden dibujar de diferentes tamaños y esto se lo hace con el mouse. Como parte de la edición de las figuras geométricas y el texto se obtuvo lo siguiente: R) Se puede ver el color con el que se dibuja la figura seleccionada, esto se lo hace al escoger el modo “editar”, una vez seleccionado se puede arrastrar con el mouse a una nueva posición. También se puede señalar el texto dibujado y cambiar el tipo de letra o el contenido escribiendo el nuevo contenido en el campo de texto mencionado en el punto F. Como parte del guardado y recuperado de la información se obtuvo lo siguiente: S) se implementó la carga de archivos de extensión “vwb” el cual permite cargar un estado de pizarra dibujado y guardado anteriormente.

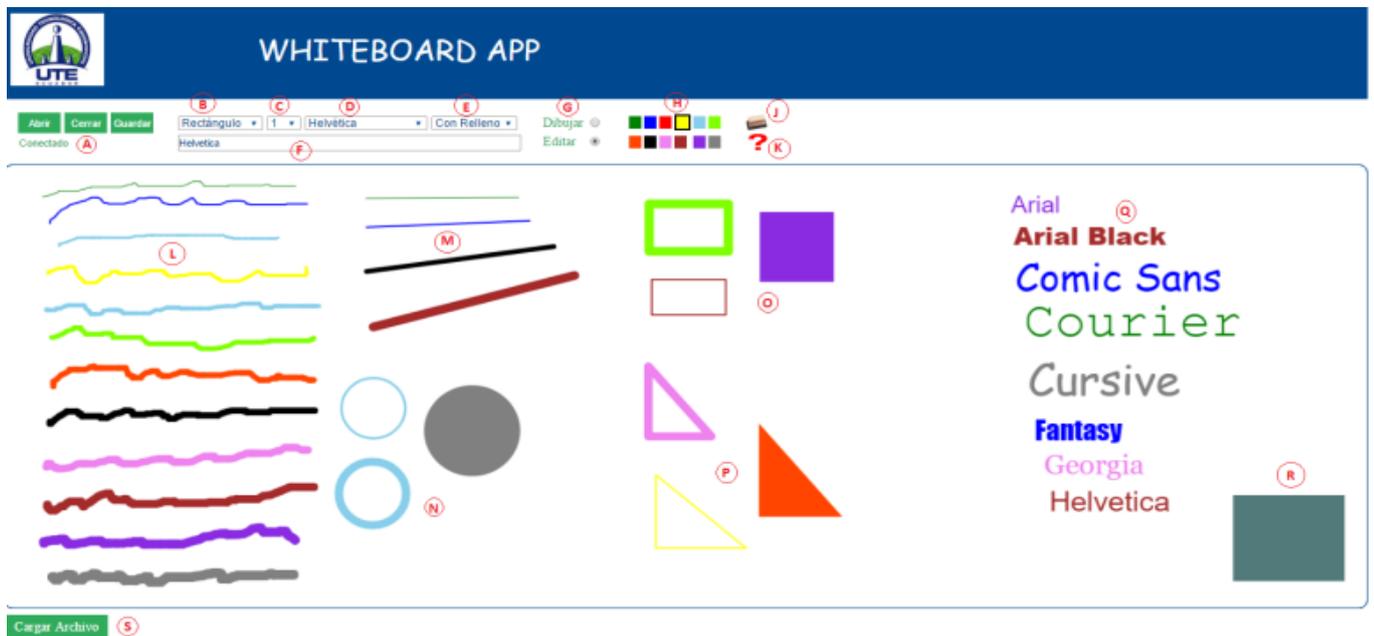


Figure 2. Área de trabajo PVCW

Como parte del intercambio de información se implementó la tecnología websockets y el tipo de formato Json como estructura del mensaje. La estructura del mensaje por cada trazo (Figura 2) es la siguiente:

{figura: nombre_figura, x1: coordenada inicial en x, x2: coordenada final en x, y1: coordenada inicial en y, y2: coordenada final en y, color: color_figura, tamaño: tamaño_figura, estilo: estilo_figura}.

Se probó la funcionalidad de la aplicación con un total de diez usuarios conectados al mismo tiempo en los diferentes navegadores mencionados en la sección anterior. La actualización de la información en las pizarras de todos los usuarios apareció correctamente y sin demoras que destacar, el borrado y cambio de posición de las figuras dibujadas también ocurrió y se percibió en las mismas condiciones. La carga de imágenes fue correcta pero con una leve demora para imágenes mayores de 200 KB, la descarga en archivo de la información de la pizarra y la carga de la misma también se realizó de manera correcta, en definitiva, cualquier acción que se ejecutó en la pizarra de un usuario, se reflejó correctamente y en tiempo real en la pizarra del resto de los usuarios conectados.

3. Discusión

PVCW, en primer lugar, cumple con las definiciones del proyecto; proporciona una amplia variedad de opciones de dibujo incluyendo figuras geométricas editables, con lo que no se cuenta en (Akison, 2014; Good Code, 2016; Kniberg, 2016); al ser basado en web no requiere ser instalado en el cliente como (Brinkmann, 2015), y no tiene las limitaciones de interacción relacionadas al uso de un sistema operativo específico que presentan (Brinkmann, 2015; Wüest et al., 2015; Zhang & Yu, 2014); es cierto que plataformas como (Higuchi et al., 2015; Lech et al., 2012) poseen cualidades multimedia e interactivas superiores, pero (Lech et al., 2012) requiere equipamiento específico como cámaras, y (Higuchi et al., 2015) no permite la colaboración de más de dos participantes; finalmente, la respuesta de PVCW en dispositivos móviles fue adecuada, especialmente si consideramos que no se presentan problemas que incluso sistemas de pago manifiestan a este nivel ya que (Twiddla, 2016) no reaccionó al cambio de orientación del dispositivo, y (Scribblar, 2016) no permitió dibujar en un dispositivo Android incluso usando un browser con soporte para flash.

4. Conclusiones y Recomendaciones

PVCW, a pesar de encontrarse en su primera versión, todavía a nivel de prototipo, proporciona funcionalidades muy competitivas, comparado con sistemas creados tanto para investigación como con fines comerciales. La interfaz gráfica es funcional en diversos sistemas operativos, browsers y dispositivos, sobre todo móviles; el conjunto de opciones para compartir información: dibujos, figuras y texto, cubre una amplia gama de necesidades de los usuarios; la sesión completa puede grabarse por cualquier cliente, localmente, y cargarse incluso fuera de línea conservando toda la distribución de componentes; el número de usuarios concurrentes está limitado más por los recursos del sistema (hardware y red) que por la aplicación; finalmente la

respuesta a los eventos generados por los usuarios es en tiempo real, con demoras no significativas, al menos en las pruebas realizadas.

Para las futuras versiones del sistema se propone incluir chats de texto con secciones gráficas bien definidas (actualmente puede hacerse ad-hoc en un campo de texto), así como estudiar las posibilidades de incluir audio y video, sea a través del mismo canal WebSockets, o mediante tecnologías alternativas como WebRTC (Vashishth, Sinha, & Babu, 2016). Finalmente se espera trabajar en seguridad, a nivel de cifrado, y en espacios de trabajo separados para grupos independientes.

Bibliografía

- Akison, A. (2014, mayo 7). Look Back at Shared Whiteboard. Recuperado el 24 de agosto de 2016, a partir de <http://www.akison.com/look-back-at-shared-whiteboard/>
- Brinkmann. (2015, mayo 27). Computer Whiteboard software Easy Whiteboard. Recuperado el 24 de agosto de 2016, a partir de <http://www.ghacks.net/2009/01/18/computer-whiteboard/>
- Dalsgaard, P., Halskov, K., Mackay, W., Maiden, N., & Martens, J.-B. (2015). Supporting Creative Design Processes in Blended Interaction Spaces. En Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition (pp. 395–396). New York, NY, USA: ACM. <https://doi.org/10.1145/2757226.2767184>
- Fette, I., & Melnikov, A. (2011). The WebSocket Protocol. IETF-RFC. Recuperado a partir de <http://www.rfc-editor.org/info/rfc6455>
- Good Code. (2016). A web whiteboard. Recuperado el 24 de agosto de 2016, a partir de <https://awwapp.com/info/>
- Hamdan, K., Al-Qirim, N., & Asmar, M. (2012). The effect of Smart Board on students behavior and motivation. En 2012 International Conference on Innovations in Information Technology (IIT) (pp. 162–166). <https://doi.org/10.1109/INNOVATIONS.2012.6207723>
- Heffelfinger, D. (2014). Java EE 7 with GlassFish 4 Application Server. PACKT Books. Recuperado a partir de <https://www.packtpub.com/application-development/java-ee-7-glassfish-4-application-server>
- Higuchi, K., Chen, Y., Chou, P. A., Zhang, Z., & Liu, Z. (2015). ImmerseBoard: Immersive Telepresence Experience Using a Digital Whiteboard. En Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (pp. 2383–2392). New York, NY, USA: ACM. <https://doi.org/10.1145/2702123.2702160>

- Hou, D., & Xia, H. (2009). Analyzing and Designing of the Classroom Teaching System Based on the Network. En International Conference on Environmental Science and Information Application Technology, 2009. ESIAT 2009 (Vol. 2, pp. 614–617). <https://doi.org/10.1109/ESIAT.2009.232>
- Jaramillo, D., Nguyen, V. D., & Newhook, R. (2014). Real-time experience techniques for collaborative tools on mobile. En IEEE SOUTHEASTCON 2014 (pp. 1–6). <https://doi.org/10.1109/SECON.2014.6950669>
- Kleffmann, M., Hesenius, M., & Gruhn, V. (2015). Connecting UI and Business Processes in a Collaborative Sketching Environment. En Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (pp. 200–205). New York, NY, USA: ACM. <https://doi.org/10.1145/2774225.2775076>
- Kniberg, H. (2016). Web Whiteboard. Recuperado el 24 de agosto de 2016, a partir de <https://www.webwhiteboard.com/>
- Lech, M., Kostek, B., & Czyzewski, A. (2012). Virtual Whiteboard: A gesture-controlled pen-free tool emulating school whiteboard. Intelligent Decision Technologies, 6(2), 161–169. <https://doi.org/10.3233/IDT-2012-0132>
- Limtrairut, P., Marshall, S., & Andreae, P. (2016). Mobile Learning Application for Computer Science Students: A Transactional Distance Perspective. En Proceedings of the 2016 ACM Conference on International Computing Education Research (pp. 285–286). New York, NY, USA: ACM. <https://doi.org/10.1145/2960310.2960350>
- Mota, D., Carvalho, C. V. de, & Reis, L. P. (2011). Fostering Collaborative Work between educators in higher education. En 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1286–1291). <https://doi.org/10.1109/ICSMC.2011.6083837>
- Rooksby, J., & Ikeya, N. (2012). Collaboration in Formative Design: Working Together at a Whiteboard. IEEE Software, 29(1), 56–60. <https://doi.org/10.1109/MS.2011.123>
- Russell, S., & Creighton, O. (2011). Artwork for requirements definition. En 2011 Fourth International Workshop on Multimedia and Enjoyable Requirements Engineering (MERE'11) (pp. 34–37). <https://doi.org/10.1109/MERE.2011.6043941>
- Scribblar. (2016, agosto 22). Online Whiteboard and Collaboration. Recuperado el 22 de agosto de 2016, a partir de <https://scribblar.com/>
- Twiddla. (2016, agosto 22). Team Collaboration for the Web. Recuperado el 22 de agosto de 2016, a partir de <https://www.twiddla.com/>

- Vashishth, S., Sinha, Y., & Babu, K. H. (2016). Addressing Challenges in Browser Based P2P Content Sharing Framework Using WebRTC. En 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA) (pp. 850–857). <https://doi.org/10.1109/AINA.2016.143>
- Wang, V., Salim, F., & Moskovits, P. (2012). The Definitive Guide to HTML5 WebSocket. Apress.
- Wüest, D., Seyff, N., & Glinz, M. (2015). FlexiSketch Team: Collaborative Sketching and Notation Creation on the Fly. En Proceedings of the 37th International Conference on Software Engineering - Volume 2 (pp. 685–688). Piscataway, NJ, USA: IEEE Press. Recuperado a partir de <http://dl.acm.org/citation.cfm?id=2819009.2819138>
- Zhang, F., & Yu, H. (2014). AegeanBoard: An Interactive Whiteboard Messenger for iPad. En 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud) (pp. 257–258). <https://doi.org/10.1109/MobileCloud.2014.42>