

## Metodología de desarrollo de software para videojuegos con tecnología Google Cast®

QUIJANO-ABAN, Víctor Manuel \*†, CEH-VARELA, Edgar Eduardo, CHABLE-MUKUL, Rodrigo

*Universidad Tecnológica Metropolitana*

Recibido Junio 15, 2017; Aceptado Noviembre 21, 2017

### Resumen

El presente artículo es una propuesta para desarrollar videojuegos educativos con tecnología Cast. En este proyecto se utilizó la tecnología Cast para crear videojuegos para los dispositivos móviles como controles de juego y la televisión como escenario del tablero con uno o más jugadores a la vez. Se utilizaron herramientas open source y librerías Google Cast SDK para desarrollo de aplicaciones nativas Android y aplicaciones web HTML5. Con las librerías adecuadas y herramientas de desarrollo es posible crear videojuegos con: un dispositivo *streaming* multimedia, los dispositivos móviles y una la televisión. El SDK de Google Cast amplía las aplicaciones: Android, Chrome o iOS para controlar un televisor o un sistema de sonido. Soporta muchos formatos de contenidos multimedios, protocolos y codecs para facilitar la integración. El resultado fue el diseño y desarrollo del videojuego “Tres en línea” que se proyecta el tablero en un televisor y pueden jugar uno o dos jugadores. El videojuego cuenta con niveles de dificultad e instrucciones para jugar. Además se tiene la documentación del videojuego para futuros desarrollos, ya que el proceso de desarrollo de un videojuego con tecnología Cast requiere trabajo disciplinado y una técnica de programación para el manejo de estados y eventos en tiempo de ejecución.

### Tecnología Cast, Videojuegos para TV, Juegos de tablero

### Abstract

This article is a proposal to develop educational video games with Cast technology. In this project, Cast technology was used to create videogames for mobile devices such as a game controls and TV as the stage of the board with one or more players at a same time. Open source tools and Google Cast SDK libraries were used to develop native Android applications and HTML5 web applications. With the right libraries and development tools it is possible to create video games with: a multimedia streaming device, mobile devices and a television. The Google Cast SDK extends apps: Android, Chrome, or iOS to control a TV or sound system. It supports many formats of multimedia contents, protocols and codecs to facilitate the integration. The result was the design and development of the video game "Tic-Tac-Toe" the board is deployed on a television and can play one or two players. The game has levels of difficulty and instructions to play. In addition, we have the documentation of the video game for future developments, since the process of developing a video game with Cast technology requires disciplined work and a programming technique for the management of states and events at run time.

### Cast technology, TV game apps, Strategy board games

**Citación:** QUIJANO-ABAN, Víctor Manuel, CEH-VARELA, Edgar Eduardo, CHABLE-MUKUL, Rodrigo. Metodología de desarrollo de software para videojuegos con tecnología Google Cast®. Revista de Tecnología e Innovación 2017, 4-13: 32-39.

\* Correspondencia al Autor (Correo electrónico: victor.quijano@utmetropolitana.edu.mx)

† Investigador contribuyendo como primer autor.

## Introducción

En la actualidad los videojuegos han extendido la experiencia de juego de la consola a los dispositivos móviles. Los videojuegos, son programas electrónicos que involucran la interacción de una o más personas con una máquina o *interfase* que genera gráficos (Prensky, 2001).

Los videojuegos han ido adquiriendo mayor presencia pública en las últimas décadas. Los videojuegos han trascendido su papel inicial de fuente de entretenimiento para niños y se han convertido en un producto cultural (Albarracín, *et al.*, 2017). El público objetivo ha ampliado su rango de edad al aumentar el número y tipo de dispositivos en los que es posible jugar desde las computadoras portátiles a los dispositivos móviles pasando por las consolas.

Los videojuegos no hubieran existido nunca sin la electrónica y las computadoras, ejemplos varios: *Pong*, *Pacman*, *Space Invaders*, etc. Ninguno de ellos existiría sin el uso masivo de la computadora personal y la televisión. La historia inicia en el año 1972, cuando aparece una extraña cabina de juego instalada en el bar *Andy Capp's* en EE.UU. Está se parecía un poco a las máquinas de *pinball*, pero en lugar de una mesa de juego tenía un monitor de televisión (Pérez-Latorre, 2010). El juego se llamaba *Pong*. Éste fue el último prototipo de Nolan Bushnell y Ted Dabney, unos jóvenes ingenieros fundadores de la compañía llamada Atari (Kent, 2001).

Los videojuegos, pueden considerarse herramientas para el aprendizaje. Actualmente los videojuegos poseen características que los hacen atractivos para los jugadores y que los dotan de una gran potencialidad como instrumentos para aprender o desarrollar estrategias específicas y para la adquisición de conocimientos (Gros, 2007).

Otras características esenciales de los videojuegos los habilitan como herramientas en entornos educativos y son: la competición, la presencia de objetivos, la existencia de reglas bien definidas y la necesidad de tomar decisiones al momento (Charsky, 2010).

A continuación se presenta una propuesta para desarrollar juegos de tablero en videojuegos educativos con la tecnología *Cast*. En este proyecto se utilizó la tecnología *Cast* para crear videojuegos con los dispositivos móviles como controles de juego y la televisión como escenario del tablero para uno o más jugadores a la vez.

La tecnología *Cast* de Google® “permite la presentación de contenidos *multi-screen* con un ChromeCast™ entre dispositivos móviles (teléfonos inteligentes, tabletas, computadoras portátiles) y pantallas *large-screen*”. (Google Cast, 2017).

El ChromeCast™, es un *dongle* o pequeño dispositivo que se conecta a una entrada *HDMI* (High-Definition Multimedia Interface por sus siglas en inglés) en la televisión para reproducir contenidos multimedios. Después de encender la televisión y cambiar al puerto *HDMI* correcto del ChromeCast™, se descarga una aplicación de configuración. La aplicación configura el ChromeCast™y la conecta a la red Wi-Fi del usuario. Una vez configurado, los usuarios pueden descargar cualquier aplicación compatible; videojuegos en nuestro caso.

A demás del Chromecast se utilizó Google Cast SDK o librerías de software necesarias para desarrollar y publicar una aplicación (*app*). Las librerías SDK se pueden obtener descargando el instalador sin costo y resulta simple de integrar, ya que no es necesario desarrollar una nueva *app*. Se debe incorporar la librería a una aplicación móvil para enviar el contenido al TV. (Noor, 2007).

El dispositivo Chromecast™ utilizado para el desarrollo del videojuego tiene la forma de disco circular y pesa 39 gramos, como se muestra en la Figura 1. Éste viene con conector HDMI que sale por un cable que va a la TV. En su interior tiene un procesador dual core a 1.3GHz y 512MB de RAM. Sus estándares de conexión son 802.11 b/g/n/ac Wi-Fi (2,4 GHz/5 GHz). Admite seguridad WEP, WPA/WPA2. Actualmente está disponible la tercera generación desde 2017.



**Figura 1** Dispositivo ChromeCast™  
Fuente: Google ChromeCast™

## Metodología

La problemática que aborda este trabajo, es realizar una propuesta para desarrollar los juegos de tablero en videojuegos educativos con la tecnología Cast y que incluyan características de: funcionamiento, funcionalidad y usabilidad.

La metodología que se siguió es la sugerida por Google (Google Developer, 2017). Consiste en un proceso de desarrollo de software iterativo y con cinco actividades principales:

1. Diseño de la aplicación.
2. Codificar la aplicación.
3. Registrar la aplicación.
4. Probar la aplicación.
5. Publicar la aplicación.

1. Diseño de la aplicación. Para hacer un videojuego se requieren actividades de diseño gráfico, color, etc. (Abril-García, *et al.*, 2017). Además del diseño se cuidó en seguir las reglas de buenas prácticas para la interactividad y usabilidad según la guía técnica para la interface de usuario UX propuesta por Google (Google Cast Game, 2017).

2. Codificar la aplicación. Para desarrollar una aplicación Chromecast™ se definió: la plataforma de desarrollo, el tipo de *receiver app* y no reproducir multimedios. Con respecto a la plataforma de desarrollo puede ser: Android, iOS o Chrome. Una aplicación Chromecast™ está formada por dos aplicaciones que se ejecutan al mismo tiempo: *sender* y *receiver*. La aplicación *sender* se instala en los dispositivos móviles y las aplicaciones *receiver* en un servidor web que se muestra en la TV.

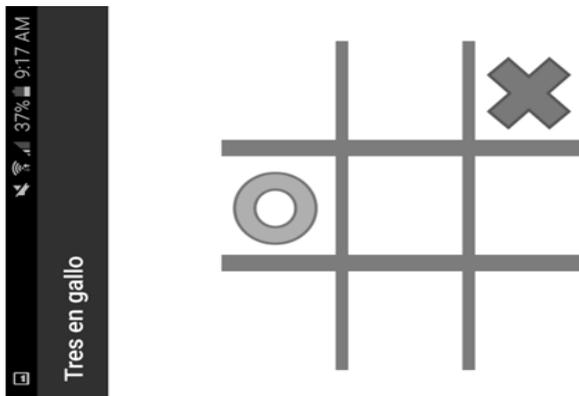
La *receiver app*, es una aplicación web moderna que usa lenguaje de marcas de hipertexto (HTML5/js, por sus siglas en inglés) para ser ejecutado en el aparato ChromeCast™. La aplicación se debe “hospedar” en un servidor web y ejecutarse en el Chromecast™. En la Figura 2, se muestra el videojuego en un televisor LCD de pantalla plana.

Se utilizaron herramientas open source: Chrome Dev Editor, las librerías mencionadas antes y el editor IDE Android Studio para el desarrollo de aplicaciones nativas Android propuesto por (Baez-Ibarra. *et al*, 2016).



**Figura 2 Aplicación receiver app**  
Fuente: Elaboración propia

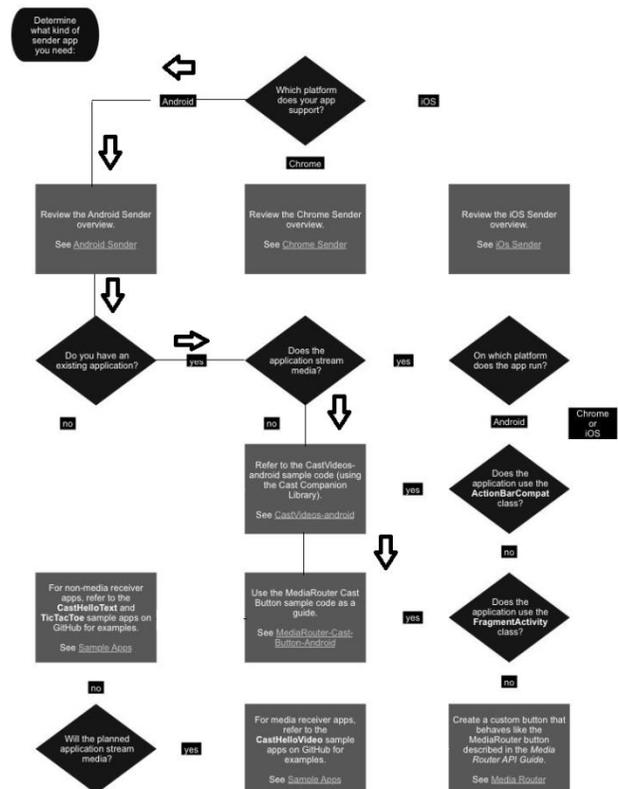
La *sender app*, se encarga de conectar y sincronizar con la *receiver app* en el aparato ChromeCast™, con esto se asegura y mantiene la comunicación por paso de mensajes. Para este proyecto se desarrolló una aplicación Android en un teléfono inteligente con sistema operativo Android Lollipop 5.1. En la Figura 3 se tiene una *app* que funciona como un control remoto para el jugador corriendo en el teléfono.



**Figura 3 Aplicación sender app**  
Fuente: Elaboración propia

En la Figura 4, Se muestra un diagrama de flujo y se explica siguiendo las flechas de arriba-abajo. Para este proyecto (en la primera flecha) se eligió Android como plataforma.

Luego *Custom receiver app* para el tipo de *app*, esto quiere decir que las funciones de interactividad y los elementos gráficos de la *interface* deben ser codificados completamente por el desarrollador. Y siguiendo con la última flecha, muestra que para este videojuego no se utilizó la librería que soporta multimedia (*media types*, en inglés).



**Figura 4 Desarrollo de un Sender app**  
Fuente: Google Cast Developers

3. Registrar la aplicación. Para esta actividad se requirió previo pago de licencia de desarrollador Google. Esto se hace en un sitio web llamado la Consola de Desarrollador para alta a la aplicación y el dispositivo Chromecast™ utilizado. Ver Figura 5. Lo anterior generará una clave ID (Google Developer Console, 2017). La clave ID sirve de código de seguridad para sincronizar el *receiver* a solicitud de la aplicación *sender*.

Luego del registro se debe esperar un tiempo de horas para utilizar la *app*. Depende del contenido y estará sujeto a aprobación por parte de las políticas de Google. Esta es una medida para evitar correr código mal intencionado de terceros.

4. Probar la aplicación. Se realizaron pruebas de funcionamiento con la aplicación móvil, el aparato ChromeCast™ y el televisor. Las aplicaciones *sender* y *receiver* deben correr en la misma red para sincronizar e intercambiar contenido, mensajes y no generar errores de conexión. Además se realizaron pruebas de unidad (*Unit test*) con la herramienta Android Studio. Una prueba de unidad se refiere a la prueba individual de métodos y clases del proyecto de modo que el desarrollador pueda tener confianza de que los cambios recientes en el código no afectaran en el futuro su funcionamiento.

### Welcome to the Google Cast SDK Developer Console

The Google Cast Developer Console enables developers to register applications and authorize devices for testing.

#### Applications

Application ID	Application Name	Status	Statistics
FF01D6A4	Tres en gallo	Published	View

ADD NEW APPLICATION

**Figura 5** Developer console

Fuente: Elaboración propia

5. Publicar la aplicación. Finalmente la *receiver app* es una aplicación web que se debe alojar en un servidor web comercial (Mendoza-Manriquez. *et al*, 2016), además el servicio debe ser 24 hrs. por 365 días y para este proyecto fue necesario un certificado SSL vigente. SSL (Secure Socket Layer, en inglés) es un protocolo de seguridad que hace que los datos viajen de manera íntegra y segura (Ristic, 2014).

## Resultados

A continuación se muestran los resultados obtenidos del proceso de desarrollo. En primer lugar se desarrolló el juego “tres en línea” que proyecta en la televisión el tablero y con un dispositivo móvil Android como un control para jugar. Ver Figura 6. El videojuego tiene niveles: fácil, y difícil para un jugador contra la máquina. Para dos jugadores no hay niveles ya que cada jugador desarrolla su estrategia de anticipación para ganar.



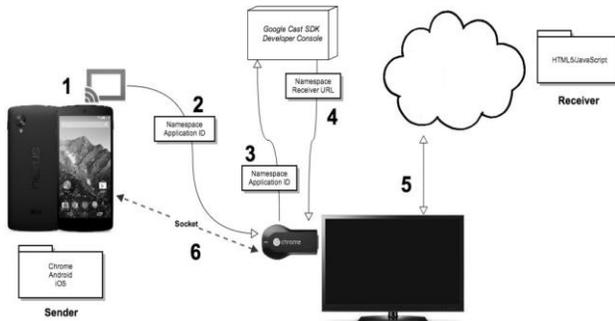
**Figura 6** Videojuego Tres en línea

Fuente: Elaboración propia

El juego tres en línea, también conocido tres en raya (en España, Ecuador y Bolivia), juego del gato (en Chile y México), Tic-Tac-Toe (en Estados Unidos y UK), Tres en gallo (en Venezuela y partes de México). Es un juego de lápiz y papel entre dos jugadores identificados por: O y X, que marcan los espacios de un tablero de 3x3 alternadamente. Un jugador gana si consigue tener una línea de tres de sus símbolos: la línea puede ser horizontal, vertical o diagonal (Hartnell, 1995).

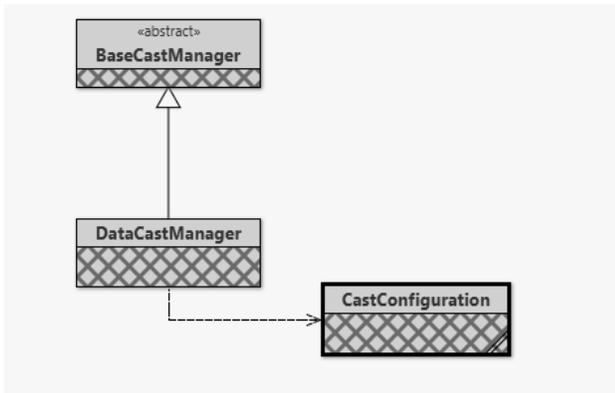
En segundo lugar se tiene la documentación del videojuego para futuros desarrollos. En la Figura 7, inicia la *sender app* y “descubre” (1) que tiene a su alcance un dispositivo Chromecast™ para jugar.

Luego se “dirige” a (2) la dirección URL (*namespace receiver url*) del *host* y la app ID mencionada antes. La aplicación envía la app ID (3) al aparato Chromecast™ y este se comunica al sitio Developer Console que valida la app (4), luego toma la URL para ejecutar la *receiver app* en el CromeCast (5) y finalmente muestra las instrucciones para jugar en el televisor (6).



**Figura 7** Ejecución del videojuego en TV  
Fuente: *Chromecast Technical blog*

En la Figura 8, se muestra la jerarquía de clases para desarrollar la *sender app* con Android, donde todo deriva de la clase abstracta *BaseCastManager*. Pero es responsabilidad de la clase *DataCastManager* implementar el botón “Cast” para descubrir un dispositivo Chromecast™ (*device discovery*).



**Figura 8** Jerarquía de clases Android  
Fuente: Elaboración propia

Finalmente se aplicó una encuesta al azar Ver Figura 9, a treinta y un estudiantes asistentes de la “Expo Talento UTM” (Yucatán, 2017) para evaluar la percepción de los usuarios y como expresan la experiencia de juego, durante la presentación de proyectos de la Universidad Tecnológica Metropolitana. Los resultados de la encuesta, el 50% de los que respondieron, consideró que debe mejorar la presentación del videojuego. El 60% que es adecuada la funcionalidad (fácil jugar). Y el 90% no reportó problemas para descargarlo e instalarlo (funcionamiento). Ver Anexos.



**Figura 9** Expo Talento UTM  
Fuente: Elaboración propia

**Agradecimientos**

A las autoridades de la Universidad Tecnológica Metropolitana por las facilidades prestadas para la realización de este proyecto.

A los alumnos Rodrigo Chable Mukul y José G. Moo Pech por su participación en este proyecto en la codificación y pruebas de la primera versión del videojuego. A todos los estudiantes que jugaron y opinaron sobre el videojuego, las sugerencias serán tomadas en cuenta para futuras versiones.

## Conclusiones

Con el dispositivo ChromeCast™ se abre un abanico de posibilidades para desarrollar videojuegos multijugador y multiplataforma con *streaming* multimedia. Los videojuegos en dispositivos de consumo y uso común en la vida diaria permiten extender la experiencia de juego como pudimos comprobar en este trabajo.

La codificación de videojuegos en más de un dispositivo permite llegar a un mayor número de público que demanda siempre jugar el último juego. Sin embargo el proceso de desarrollo de un videojuego con tecnología Cast requiere trabajo disciplinado y una técnica de programación para el manejo de estados y eventos: *device discovery*, sesiones y mensajes entre *sender* y *receiver*.

Con respecto a las pruebas de unidad al código, éstas le agregan valor al proceso de desarrollo propuesto, ya que permiten automatizar las pruebas de funcionalidad cada vez que se realice un cambio o mejora al juego.

Como trabajo futuro ya se iniciaron otros juegos de tablero tales como: “el zorro y las gallinas”, “cuatro en línea” y “Nim”. Además se incluirán algoritmos de Inteligencia Artificial (IA) para búsqueda, anticipación y estrategia ganadora para que exija un esfuerzo extra por parte del jugador en la resolución de problemas con videojuegos educativos.

## Anexos

Encuesta de salida para el jugador.

1. La aplicación se descargó con éxito en el dispositivo.

Si No.

2. La instalación de la aplicación fue exitosa en el dispositivo.

Si No.

3. Las instrucciones incluidas en el videojuego son claras.

Si No.

4. Jugar con la aplicación resultó fácil.

Si No.

5. La maniobra con las diferentes pantallas entre la TV y el móvil fue exitosa.

Si No.

6. Los colores, textos, gráficos y audio son de su agrado.

Si No.

## Referencias

Abril-García, J. H., Meza-Ibarra, I. D. y Chenoweth-Chenoweth I. R. (2017). Interfaz gráfica en WPF XNA para monitoreo de control de Xbox 360. Revista de Sistemas Computacionales y TIC'S. ECORFAN. 3-7:1-6.

Albarracín, L., Hernández-Sabate, A. y Gorgorió, N. (2017). Los videojuegos como objeto de investigación en Educación Matemática. Modelling in Science Education and Learning. Volumen 10(1). pp.53-69.

Baez-Ibarra, A., Arellanes-Cancino, N. y Sosa-Perdomo, A. (2016). Efectividad de la aplicación de metodologías ágiles para el desarrollo de apps móviles. Un caso de estudio. Revista de Sistemas Computacionales y TIC'S. ECORFAN.

Charsky, D. (2010). From edutainment to serious games: A change in the use of game characteristics. *Games and culture*, Vol. 5, N. 2, pp.177–198.

Gros, B. (2007). Digital games in education: The design of games-based learning environments. *Journal of Research on Technology in Education*. Vol. 40, No. 1, pp.23–38.

Google Cast. (2017). Sitio Google Cast. Recuperado el 10 de junio de 2017, desde <https://developers.google.com/cast/>

Google Developer. (2017). Comunidad Google Cast Developers. Recuperado el 20 de junio de 2017, desde <https://plus.google.com/communities/115742157569103585450>

Google Developer Console. (2017). Sitio Google Cast Developers Console. Recuperado el 26 de junio de 2017, desde <https://cast.google.com/publish>

Google Cast Game. (2017). Google Cast Gaming UX Mockup Guidelines for Chromecast. Recuperado el 26 de junio de 2017, desde <https://developers.google.com/cast/downloads/GoogleCastGameUXguidelinesv20151203.pdf>

Hartnell, T. (1995). El libro gigante de los juegos para computadora, 2da ed., Anaya Multimedia. pp.15–18.

Kent, S. L. (2001). The ultimate history of videogames. Three Rivers Press. Nueva York.

Mendoza-Manriquez, R. F., Tapia-Tinoco, G., Ortega-Herrera, F. J., y García-Guzmán, J. M. (2016). Aplicación web para el control remoto de mecanismos mediante Arduino. *Revista de Tecnología e Innovación*. ECORFAN. 3-6: 1-7.

Noor, A. (2014). Designing the Chromecast Out-of-Box Experience. Recuperado el 27 de julio de 2017, desde Research Google <https://research.google.com/pubs/archive/42549.pdf>

Pérez-Latorre, O. (2010). Análisis de la significación del videojuego. Fundamentos teóricos del juego, el mundo narrativo y la enunciación interactiva como perspectivas de estudio del discurso. Tesis doctoral. Universidad Ponpeu-Fabra. Barcelona. España.

Prensky, M. (2001). Digital game-based learning. *Games for entertainment and learning*. ACM Computers in Entertainment. Volumen 1(1), pp.1–4.

Ristic, I. (2014). *Bulletproof SSL and TLS*. Feisty Duck Limited. pp.497.

Yucatán. (2017). Talento yucateco, altamente competitivo. Sala de prensa del gobierno del estado de Yucatán. Recuperado el 18 de agosto de 2017, desde [http://www.yucatan.gob.mx/saladeprensa/ver\\_noticia.php?id=202827](http://www.yucatan.gob.mx/saladeprensa/ver_noticia.php?id=202827)