



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PRÁCTICA PROFESIONAL

ICOMS

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR

11511308 OSCAR DAVID DÍAZ HERNÁNDEZ

ASESOR: LIC. TANIA LUCILA MEZA

CAMPUS TEGUCIGALPA; MARZO, 2020

LISTA DE SIGLAS

API	Application Programming Interface (Interfaz de Programación de Aplicaciones)
AR	Realidad Aumentada
Backend	server-side (lado-servidor)
CEO	Chief Executive Officer (Director Ejecutivo)
CSS	Cascade Spread Sheet
CTO	Chief Technology Officer (Director de Tecnología)
DASH	Dynamic Adaptive Streaming HTTP (Retransmisión Adaptable Dinámica HTTP)
DRM	Digital Right Management
Frontend	client-side (lado-cliente)
HLS	HTTP Live Streaming (Retransmisión en vivo HTTP)
HTTP	Protocolo de Transferencia de Hipertexto
I18N	Estandarización para multilinguaje
IDE	Entorno de Desarrollo Integrado
JS	Java Script
JSON	JavaScript Object Notation
OTT	Over-the-top
PaaS	Platform as a Service (Plataforma como servicio)
PP	Práctica Profesional
QA	Quality assurance (Control de Calidad)
SDK	Kit de Desarrollo de Software
SSR	Server Side Rendering (Renderizado en lado servidor)
Streaming	Retransmisión en directo

VR

Realidad Virtual

RESUMEN EJECUTIVO

El presente informe se entrega como requisito para optar al título profesional de Ingeniero en Sistemas Computacionales, describe las actividades realizadas durante el desarrollo de la práctica profesional desarrollada en el departamento Web de la empresa ICOMS durante el periodo de Octubre de 2019 a Marzo de 2020.

Al iniciar se tomó un curso sobre React JS y Redux, debido a que todo el desarrollo Web de la empresa es en base a estas tecnologías. Luego se realizó un proyecto de prueba para televisores LG y Samsung con el fin de demostrar el conocimiento en el área web y como una introducción así mismo, luego se asignaron actividades a realizar en dos proyectos; uno consiste en un juego de trivia en dispositivos móviles y el otro un reproductor para televisores LG y Samsung. Cuando la carga de trabajo de estos dos proyectos disminuyó se asignaron actividades de dos sitios webs que utilizan Renderizado del Lado del Servidor; uno de ellos fue sobre video de noticias de política estadounidense y el otro fue la creación de páginas estáticas.

Simultáneamente se daba apoyo al departamento de QA en dos proyectos, uno consiste en la transferencia de conocimiento de un validador de datos de analíticos y el otro en el probador de carga de un servidor de sockets.

ÍNDICE DE CONTENIDO

I.	Introducción	1
II.	Generalidades De La Empresa	2
2.1.	Descripción de la empresa	2
2.2.	Descripción del departamento	2
2.3.	Objetivos del puesto	2
2.3.1.	Objetivo General	2
2.3.2.	Objetivos Específicos	2
III.	Marco Teórico	3
IV.	Desarrollo	15
4.1.	Descripción del trabajo desarrollado	15
4.1.1.	Capacitación Web	15
4.1.2.	Juego Trivia Web	15
4.1.3.	Transferencia de conocimiento QA	17
4.1.4.	Proyecto de prueba	18
4.1.5.	Reproductor para plataformas OTT	19
4.1.6.	Desarrollo Sistema Web con SSR	21
4.1.7.	Sitio Web de Noticias	22
4.1.8.	Probador de Carga de Server Socket IO	24
4.2.	Cronograma de actividades	24
V.	Conclusiones	26
VI.	Recomendaciones	27
	Bibliografía	28

ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Curso React y Redux en Udemey	4
Ilustración 2 - Ejemplo de historias en Jira	5
Ilustración 3 - Arquitectura Frontend y Backend	6
Ilustración 4- Flujo de Datos Redux	7
Ilustración 5 - Tizen Studio	8
Ilustración 6 - Tabla de soporte del reproductor Shaka por plataforma	9
Ilustración 7 - Árbol Git	10
Ilustración 8 - Ejemplo de Análisis de datos en Amplitude	11
Ilustración 9 - Diferencia tipo de renderizados	12
Ilustración 10 - Ejemplo de Lanzamiento de un producto	13
Ilustración 11 - Captura de pantalla Charles Server	14
Ilustración 12 - Ejemplo Arrastrar y Soltar	16
Ilustración 13 - Pantalla Principal de Sistema de Automatización Analíticos	17
Ilustración 14 - Ejemplo Proyecto de Prueba	18
Ilustración 15 - Reproductor OTT	19
Ilustración 16 - Menú multilinguaje reproductor OTT	21
Ilustración 17 - The Recount versión de escritorio	23
Ilustración 18 - The Recount versión móvil	23
Ilustración 19 - Ejemplo Artillery	24
Ilustración 20 - Diagrama de Gantt	25

I. INTRODUCCIÓN

La importancia de la programación ha estado en aumento desde el momento de su creación y ha ido evolucionando a lo largo del tiempo. En la década de los ochentas comenzó el auge de las computadoras personales, dichas computadoras cumplían con las necesidades básicas, pero luego surgió la necesidad de poder compartir información entre ellas, es por eso a inicios de la década de los noventas empieza el auge de la Web, esta tecnología permite dar un servicio para el intercambio de información entre computadoras, desde entonces la web ha ido evolucionando hasta el día de hoy conocida como Web 2.0.

Actualmente nos encontramos en la época donde la programación web es la más importante. Debido a su forma de ejecución, el lenguaje de más uso es JavaScript, el cual es interpretado, esto significa que el navegador es el encargado de su interpretación y por lo tanto no necesita ser compilado, esto representa una ventaja sobre aquellos lenguajes compilados porque puede ser ejecutado en cualquier dispositivo con un navegador Web. Así mismo estamos en una transición tecnológica de la televisión común a los servicios de retransmisión directa (streaming). Estos servicios tienen la finalidad de la transmisión de contenido multimedia mediante servicios web.

ICOMS Technologies tiene su enfoque principal en servicios streaming; se enfoca en el desarrollo de aplicaciones de entretenimiento para diferentes plataformas móviles, over-the-top (OTT) y web. El departamento web es encargado del desarrollo de las aplicaciones OTT, Chromecast y de sistemas web.

Con la práctica profesional se busca el aprendizaje, desarrollo y mantenimiento de aplicaciones OTT y sistemas web basándose en la metodología Agile, así mismo brindar soporte al departamento de calidad (QA) sobre el sistema de automatización de analíticos y el probador de carga de un servidor de socket.

En el informe se detallan las generalidades de la empresa, descripción del departamento, objetivo general y específicos, el marco teórico donde se describen las tecnologías y herramientas utilizadas, y se describe el desarrollo de las actividades realizadas, así como las conclusiones y recomendaciones de la práctica profesional.

II. GENERALIDADES DE LA EMPRESA

2.1. DESCRIPCIÓN DE LA EMPRESA

ICOMS fue fundado en el año 2010 por Jorge García y Alejandro Corpeño, Jorge actualmente es el Director de Tecnología (CTO) y Alejandro el Director Ejecutivo (CEO). La sede está ubicada en Pasadena, California y también cuenta con oficinas en Tegucigalpa, Honduras. La empresa desarrolla aplicaciones para las distintas plataformas móviles, plataformas de televisión y plataformas web. Los productos desarrollados varían el rubro y van desde aplicaciones de streaming hasta videojuegos de Realidad Aumentada (AR) y Realidad Virtual (VR) (ICOMS, s.f.).

2.2. DESCRIPCIÓN DEL DEPARTAMENTO

El departamento web es el encargado del desarrollo de los productos OTT y también sistemas web en general. Las plataformas OTT manejadas por el departamento Web son: Tizen, Roku, y Chromecast.

2.3. OBJETIVOS DEL PUESTO

2.3.1. OBJETIVO GENERAL

- Desarrollar sistemas web de entretenimiento multimedia de streaming para plataformas OTT como televisores Samsung y LG durante un periodo de 6 meses.

2.3.2. OBJETIVOS ESPECÍFICOS

- Aprender tecnologías y herramientas utilizadas por la empresa para el desarrollo de aplicaciones web.
- Asistir a las ceremonias planteadas por la metodología Agile con el fin de pertenecer al equipo de desarrollo en el rol de desarrollador.
- Colaborar con el departamento de calidad en el soporte del sistema de automatización de analíticos.

III. MARCO TEÓRICO

Hoy en día la programación forma una parte muy importante en el mundo sin importar el rubro que sea, la programación se podría dividir de distintas maneras, una de estas divisiones puede ser entre programación web o programación de escritorio, dicha comparación puede generar controversia de cuál sería la mejor opción para especializarse. En un artículo del diario reforma se menciona que los estudiantes egresados de la carrera de sistemas que pretenden especializarse en el área web tienen una gran oferta laboral. (Reforma). Un rubro que ha estado en constante crecimiento esta década es el entrenamiento virtual, específicamente las aplicaciones orientadas al streaming (Cámara de Comercio Bogotá, 2018). La palabra streaming es una de las palabras más usadas en Web 2.0, se refiere a la tecnología de poder acceder a un archivo sin necesidad de ser descargado completamente y puede ser transmitido en tiempo real (Delgado, 2008).

Una característica de un buen desarrollador de software es ser autodidacta, para esto existen varios sitios web que son de mucho beneficio, desde algunos gratuitos como YouTube hasta otros pagados como Udemy. La cual es una plataforma que permite a expertos sobre un tema ser instructores y conectarlos con personas que desean aprender sobre ello (Udemy, 2020). Debido a la formalidad empresarial, ICOMS prefiere Udemy ya que los instructores son certificados. En la Ilustración 1 se muestra el curso que se asigna a cualquier miembro que ingresa al departamento de desarrollo Web.

Modern React with Redux [2020 Update]
 Master React v16.6.3 and Redux with React Router, Webpack, and Create-React-App. Includes Hooks!
 BESTSELLER ★★★★★ 4.6 (52,772 ratings) 174,076 students enrolled
 Created by Stephen Grider Last updated 1/2020
 English English [Auto-generated], French [Auto-generated], 2 more

What you'll learn

- Build amazing single page applications with React JS and Redux
- Realize the power of building composable components
- Become fluent in the toolchain supporting React, including NPM, Webpack, Babel, and ES6/ES2015 Javascript syntax
- Master fundamental concepts behind structuring Redux applications
- Be the engineer who explains how Redux works to everyone else, because you know the fundamentals so well!

Course content Expand all 487 lectures 47:36:19

Let's Dive In!		57:08
How to Get Help	Preview	01:01
Important Codepen Info		00:46
Our First App	Preview	10:18
CodePen Solution and Blank White Screen Help		01:37

\$13.99 \$179.99 92% off
 5 hours left at this price!
 Add to cart
 Buy now
 30-Day Money-Back Guarantee
 This course includes
 47.5 hours on-demand video
 19 articles
 1 downloadable resource
 6 coding exercises
 Full lifetime access
 Access on mobile and TV
 Certificate of Completion
 Apply Coupon
 Share
 Training 5 or more people?
 Get your team access to 4,000+ top Udemy courses anytime, anywhere.

Ilustración 1 - Curso React y Redux en Udemy

Fuente: (Udemy, 2020)

Las empresas de desarrollo de software utilizan la metodología Agile, Mathis & Vega, 2018 define Agile como un método que facilita el desarrollo basándose en distintos principios, como mantener equipos pequeños, ser flexibles al cambio, solución rápida de problemas, tiempo de desarrollo rápido y trabajo en ciclos de tiempo corto. Como apoyo a esta metodología existe Scrum que es un marco de trabajo para la gestión de productos, proyectos y servicios complejos que facilita un desarrollo mantenido e incremental. Scrum cuenta con tres elementos importantes: artefactos, roles y ceremonias (Galiano, 2016).

Scrum define tres roles básicos para liderar un proyecto (OBS Business School, s.f.):

- **Dueño del Producto:** Es la representación del cliente dentro del equipo, revisa el resultado de cada iteración, transmite las necesidades al equipo y protege los intereses del negocio.
- **Scrum Máster:** Es el encargado de la comunicación entre el cliente y el equipo.
- **Equipo Scrum:** Son los encargados del desarrollo del producto, pueden ser desarrolladores, equipo QA, diseñadores, arquitectos de software, entre otros.

Para el manejo de la metodología Agile, ICOMS utiliza la herramienta Jira, la cual proporciona todos los elementos necesarios para el manejo de esta, sirve para la planeación, desarrollo y

lanzamiento de cualquier software, ayuda a los miembros del equipo y permite la integración con otras herramientas (Atlassian, 2020). En la Ilustración 2 se muestra un ejemplo de una pizarra de Jira con sus respectivas columnas de estado y las actividades de cada una de ellas.

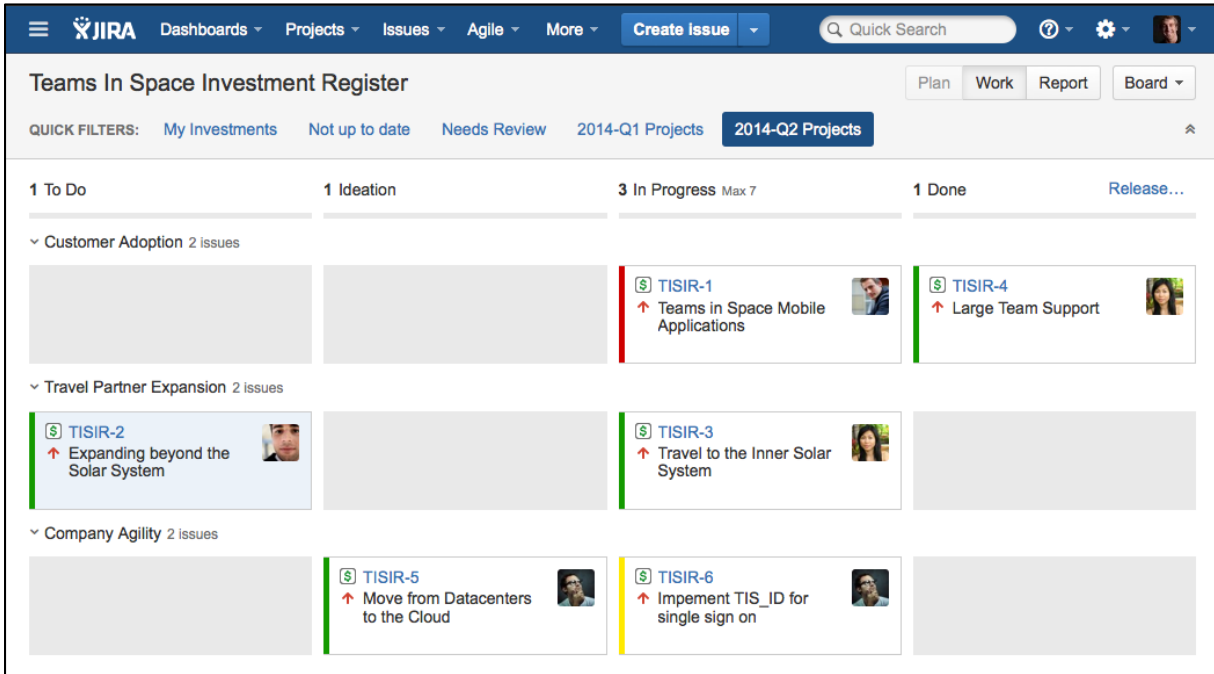


Ilustración 2 - Ejemplo de historias en Jira

Fuente: (Atlassian, 2020)

En el área de desarrollo de software existen dos partes fundamentales Frontend (lado-cliente) y Backend (lado-servidor). El Frontend de un sitio web es todo aquello que interactúa con el navegador como texto, colores, botones, entre otros. En cambio, el Backend es todo aquello que el usuario no ve, es la parte responsable de organizar y almacenar los datos y también encargado de proveer cualquier servicio que el Frontend necesite (Ferguson, s.f.). En la Ilustración 3 se observa cómo el Frontend se encarga sólo de lo que el cliente ve y también cómo el Backend se encarga del manejo de los datos y la comunicación que tiene con el cliente.

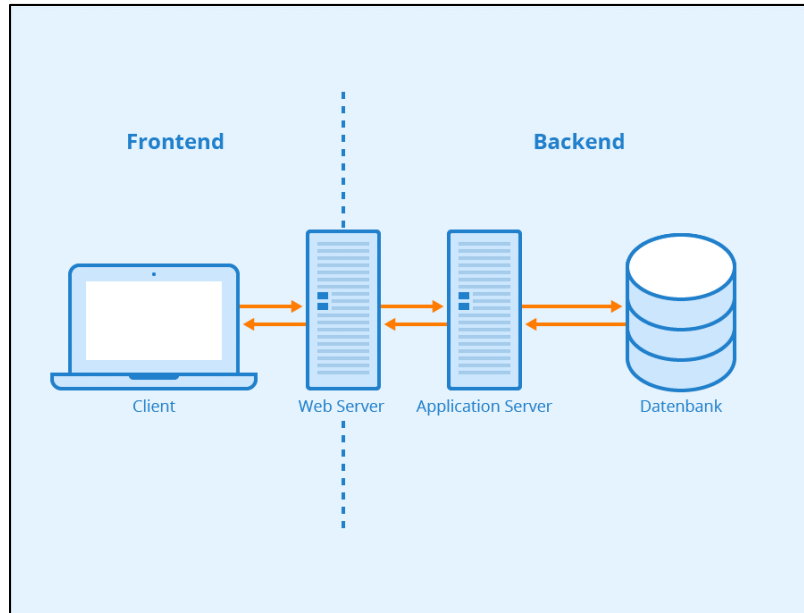


Ilustración 3 - Arquitectura Frontend y Backend

Fuente: (Seobility, s.f.)

Actualmente la tecnología web más utilizada es Node JS, el cual es un entorno de ejecución JavaScript orientado a eventos asíncronos para construir aplicaciones en redes escalables, el Protocolo de Transferencia de Hipertexto (HTTP) es soportado por Node JS y aparte de ello brinda operaciones de streaming y de baja latencia (Node.js Foundation, s.f.).

Así como la programación ha estado en constante evolución, el desarrollo web también ha evolucionado, actualmente existen dos tecnologías que están en auge, React JS y Redux, ambas se complementan y las dos están basadas en Node JS, React JS es una librería de JS que permite la construcción de interfaces de usuario y también sirve para la creación de aplicaciones de una sola página y permite la reutilización de componentes (Alvarez, 2019). Al tener múltiples componentes se genera conflicto con el manejo de los datos por eso se utiliza Redux como solución de este problema, la página oficial de Redux explica ¿por qué la creación de esta librería?, a medida avanza el tiempo las páginas web tienen como requerimiento que sean de una sola página, por lo cual los datos se deben almacenar en el estado de la aplicación y es por ello que se utiliza Redux, el cual es un contenedor predecible de estados de JS que Ayuda a escribir aplicaciones (Redux Org, s.f.). En conclusión, React JS sirve como marco de trabajo de la aplicación

y Redux sirve para el manejo y control de los datos. En la Ilustración 4 se puede observar el flujo de los datos manejados por Redux.

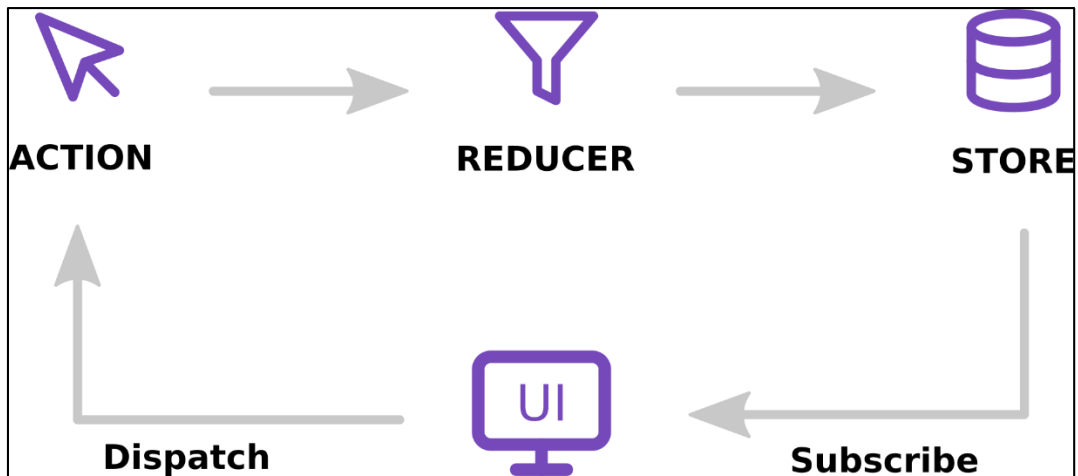


Ilustración 4- Flujo de Datos Redux

Fuente: (Stevanoski, s.f.)

Una vez estructurado el proyecto se necesita resolver otro problema, el paso de mensajes, durante años las redes han desarrollado diferentes maneras del paso de mensajes, actualmente el más fuerte y robusto es el Protocolo de Transferencia de Hipertexto (HTTP), es un protocolo en la capa de aplicación para la transmisión de documentos, sigue el modelo clásico de cliente-servidor y puede funcionar sobre cualquier capa de transporte (Mozilla Developers, 2019). Para generar llamados HTTP se utiliza una librería Axios, es una librería que permite realizar operaciones HTTP, funciona tanto en navegadores como en servidores Node JS y pueden ser síncronos o asíncronos (Desarrollo Web, 2018).

Otra forma de generar llamados es utilizando Web Sockets, es una manera de paso de mensajes bidireccional que funciona con la capa de transporte TCP y utiliza el mismo modelo cliente-servidor, para el uso y manejo de Web Sockets existe la librería Socket.IO que permite el paso de mensajes en tiempo real y funciona con Node JS (Educba, 2019). Para la programación web existe un tipo de objeto llamado JSON, es un objeto simple de leer y escribir para humanos y fácil de convertir y generar para computadoras. Su estructura principal consta de un mapa clave-valor (JSON, s.f.).

Actualmente la versión de JavaScript usada es ES6 o JS 6, pero algunos lenguajes no soportan esta versión, para eso existe Webpack, es un módulo encargado de empaquetar proyectos y también se encarga de la traducción de ES6 a una versión soportada por los dispositivos, también puede traducir CSS (Webpack, s.f.).

Una plataforma OTT son los televisores Samsung, dichos dispositivos utilizan el sistema operativo Tizen. Este está basado en Linux y es utilizado por televisores, celulares y consolas de videojuego. Para el desarrollo de aplicaciones Tizen se utiliza el Entorno de Desarrollo Integrado (IDE) Tizen Studio, el cual contiene un conjunto de herramientas, emuladores, simuladores y manejador de certificados Tizen y Samsung (Tizen, 2012). Un problema con los dispositivos que utilizan Tizen es el soporte de JavaScript ES5, por eso para el desarrollo de aplicaciones web se deben empaquetar los proyectos utilizando Webpack. En la Ilustración 5 se puede observar una captura de pantalla de Tizen Studio.

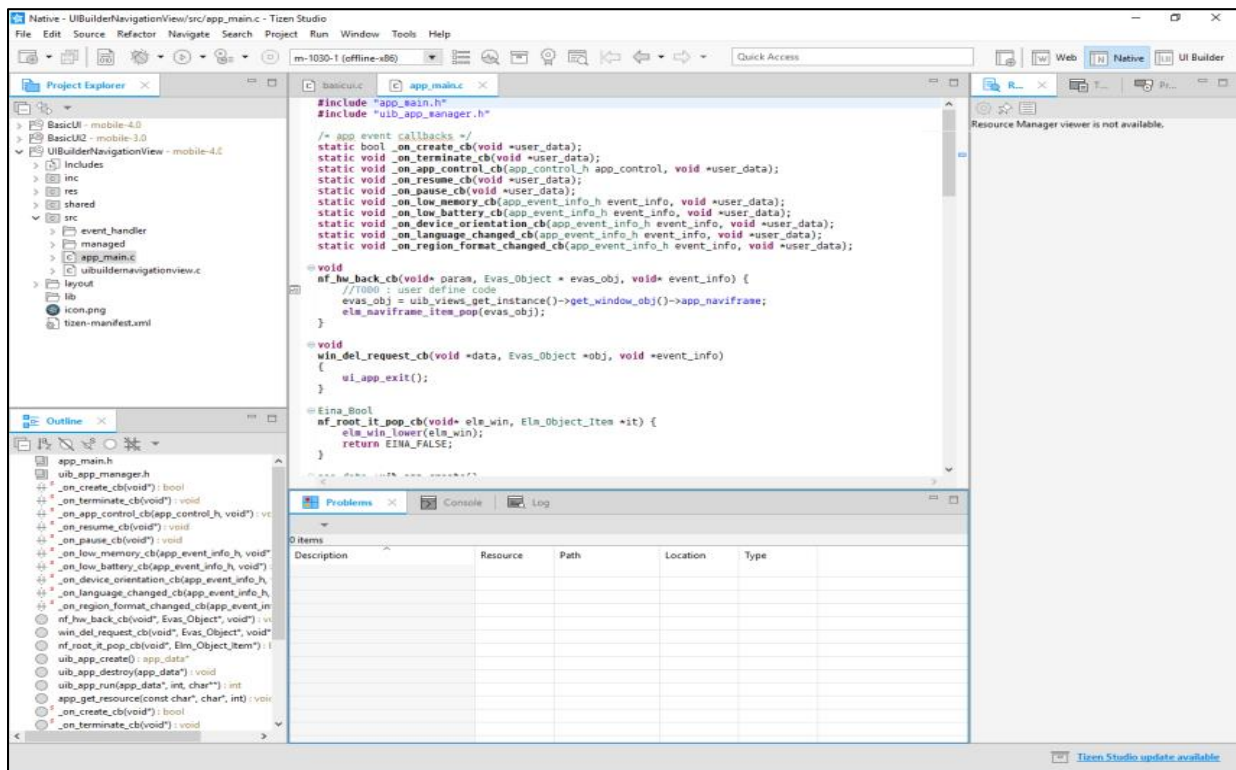


Ilustración 5 - Tizen Studio

Fuente: (Tizen, 2012)

Los servicios de streaming cada día aumentan su auge, así mismo crece la piratería, es por eso que existen los Gestores de Derechos Digitales (DRM), este un tipo de seguridad que encripta los contenidos digitales basándose en licenciamiento, de esta manera se puede controlar el correcto uso y control de niveles de acceso o periodos de tiempo, los más comunes son Widevine, HLS y Playready (Adobe, 2019). Para la reproducción de videos y contenido en vivo en dispositivos Tizen se utiliza el reproductor Avplay, es una librería de JavaScript que permite el manejo de cualquier multimedia (Samsung Developers, 2019). Otro reproductor con mucho auge es Shaka, es de código abierto y desarrollado por Google, soporta DASH y HLS sin la necesidad de ningún plugin extra, una ventaja es que tiene soporte para dispositivos Tizen y también LG (Google, 2020). En la Ilustración 6 se muestra la tabla de soporte del reproductor Shaka en las plataformas y navegadores.

Browser	Windows	Mac	Linux	Android	iOS >= 12	ChromeOS	Other
Chrome ¹	Y	Y	Y	Y	Native	Y	-
Firefox ¹	Y	Y	Y	untested ⁵	Native	-	-
Edge ¹	Y	-	-	-	-	-	-
IE ≤ 10	N	-	-	-	-	-	-
IE 11	Y ⁴	-	-	-	-	-	-
Safari ¹	-	Y	-	-	iPadOS 13 Native	-	-
Opera ¹	untested ⁵	untested ⁵	untested ⁵	untested ⁵	Native	-	-
Chromecast ²	-	-	-	-	-	-	Y
Tizen TV ³	-	-	-	-	-	-	Y

Ilustración 6 - Tabla de soporte del reproductor Shaka por plataforma

Fuente: (Google, 2020)

Para el control de versiones, las empresas de desarrollo utilizan Git, es un sistema encargado del control de versiones, fue desarrollado por Linus Torvalds, fue diseñado para funcionar desde pequeños hasta grandes proyectos con rapidez y eficiencia. Una de las características más importantes de Git es como maneja todos los cambios como un árbol (Git, s.f.). En la Ilustración 7 se muestra un árbol básico de Git.



Ilustración 7 - Árbol Git

Fuente: (Git, s.f.)

Con los avances de la inteligencia artificial y el manejo de datos grandes, las empresas han tenido que adaptar sus sistemas y aplicaciones a estas tecnologías. Para solucionar esto existen plataformas de analíticos, son sistemas encargados del manejo de grandes cantidades de datos y las procesan con el fin de generar información inteligente que ayudan a las empresas para la toma de decisiones (TechoPedia, s.f.). Una plataforma utilizada por ICOMS es Amplitude, se encarga del análisis de los datos del comportamiento de un usuario en una aplicación y permite analizar si una característica es usada u omitida por los usuarios, también categoriza a los usuarios según sus acciones dentro de las aplicaciones (Amplitude, s.f.). Omniture es otra plataforma de analíticos utilizada en ICOMS, provee un conjunto de servicios de análisis avanzado de sistemas web y segmentación, buscador de mercados y una plataforma de integración al mercado (Crunch Base, s.f.). En la Ilustración 8 se muestra un ejemplo de los datos de iTunes en Amplitude.

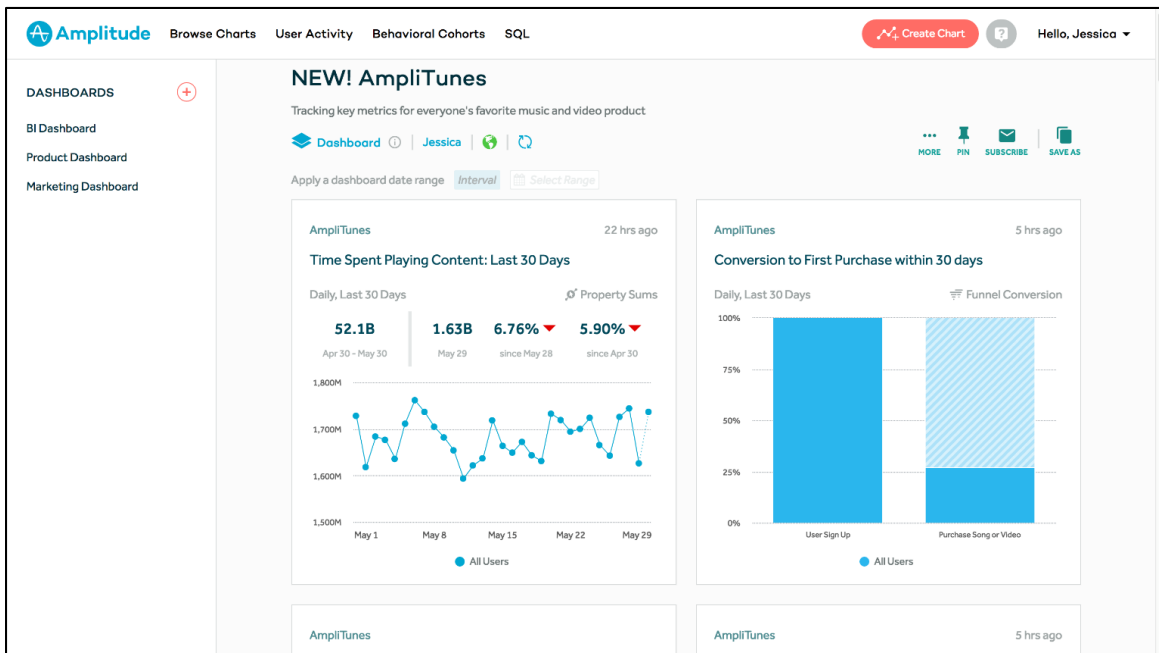


Ilustración 8 - Ejemplo de Análisis de datos en Amplitude

Fuente: (Amplitude, s.f.)

Para el análisis de audio y video se utiliza un Kit de Desarrollo de Software (SDK) que proporciona la compañía Adobe, su nombre es Heartbeat. Este analítico permite analizar toda la interacción que tiene un usuario durante la reproducción multimedia, captura cuando inicia, pausa, reanuda, carga o finaliza el contenido, aparte de ello también se encarga de recolecta información sobre los anuncios. Luego permite al equipo de científicos de datos realizar mejores recomendaciones (Adobe Corp., 2019).

Las aplicaciones se utilizan a nivel global por lo cual deben tener soporte para múltiples lenguajes, para esto se creó el estándar i18n, el cual es un proceso para para que pueda soportar lenguajes locales y configuraciones culturales según la región donde el usuario esté ubicado (Lingport, s.f.).

Hoy en día existen muchos tipos de dispositivos en los que se puede navegar por la web, un problema común es la capacidad de procesamiento, lo que ocasiona que al cargar la página inicial de un sitio, esta sea muy lenta lo cual degrada la experiencia del usuario; la solución a esto es el Renderizado desde el Servidor (SSR). ¿Cómo funciona? El servidor responde al cliente una página web completa ya con los datos en la vista. Esto genera varias ventajas, se realizan menos solicitudes al servidor, también permite a los motores de búsqueda indexar el contenido de la

página web y es escalable. (Next, 2019). En la Ilustración 9 se muestra un gráfico comparativo entre la renderización de la primera página desde el servidor y desde el cliente, se observa que en SSR el tiempo es inferior.

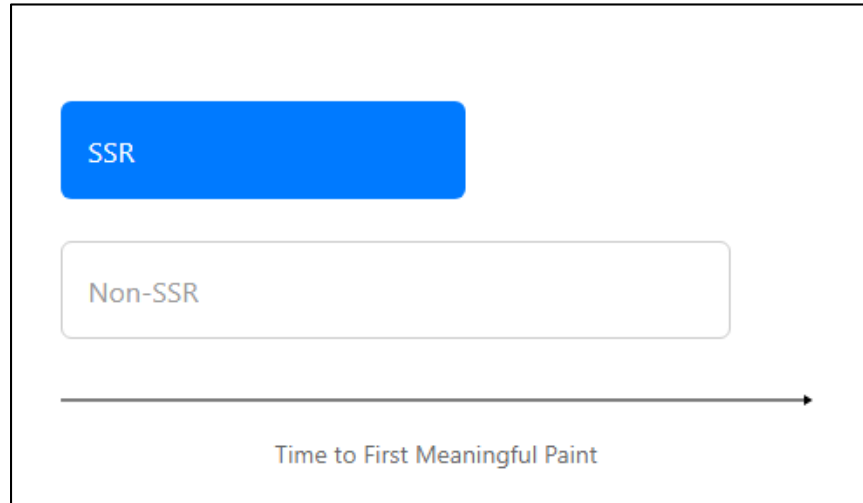


Ilustración 9 - Diferencia tipo de renderizados

Fuente: (Next, 2019)

Una librería que permite el SSR es Next JS, es un marco de trabajo basado en Node JS, este se puede utilizar con React JS proporcionando métodos adicionales para habilitar el SSR.

Luego de tener el software desarrollado se necesita salir al mercado, hoy en día es poco común que las empresas compren un servidor físico, son más utilizadas las Plataformas como Servicio (PaaS). Una PaaS común es Heroku, el cual permite al desarrollador desentenderse de la instalación y mantenimiento de los servidores y que su enfoque sea el desarrollo del producto en sí (Heroku, 2020). En la Ilustración 10 se muestra cómo se realiza el lanzamiento del producto, el único trabajo por parte del programador es subir el código a Heroku y luego este genera el acceso para los usuarios.

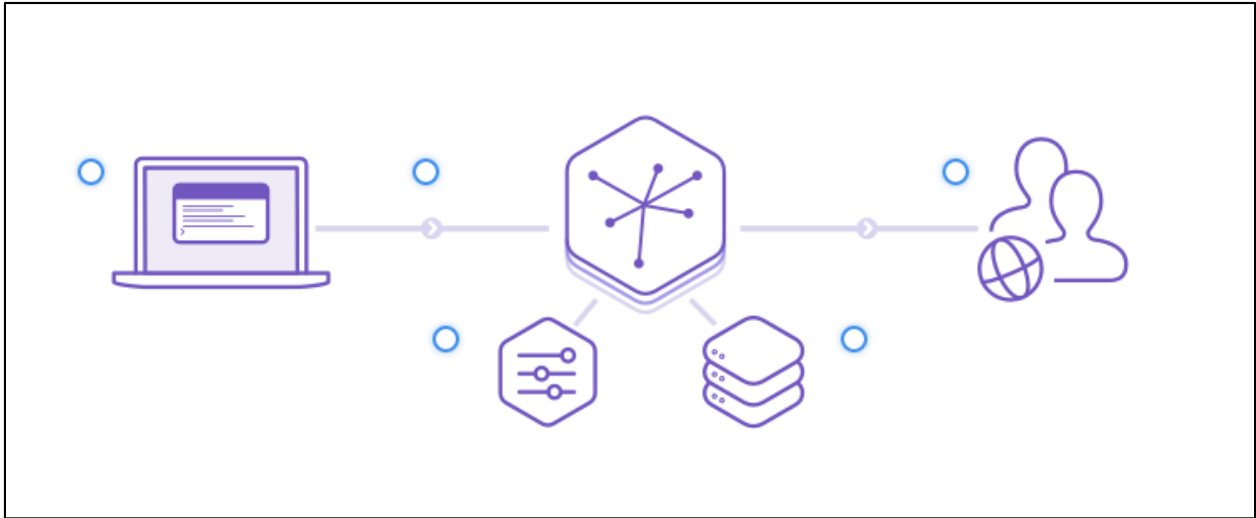


Ilustración 10 - Ejemplo de Lanzamiento de un producto

Fuente: (Heroku, 2020)

La importancia del departamento de Control de Calidad (QA) va en aumento cada día, ha pasado de ser un departamento opcional a uno obligatorio. Las empresas que ofrecen un departamento de QA tienen ventaja sobre aquellas que no, el mayor beneficio es que se genera un producto más confiado, seguro y libre de errores (G&L Group, s.f.). Un problema fuerte en el departamento de QA es probar los analíticos, cada llamado debe cumplir con ciertas normas, como el nombrado de las propiedades o el formato de los valores según su tipo.

Para las pruebas de analíticos se utiliza un software llamado Charles, su función básica es la herramienta de un servidor proxy, en cada dispositivo se configura el proxy hacia la computadora donde esté corriendo el servidor Charles y luego se observan todos los llamados HTTP, en caso de tener un certificado se podrían ver los llamados HTTPS (Charles Proxy, 2019). En la Ilustración 11 se muestra una sesión del servidor proxy Charles.

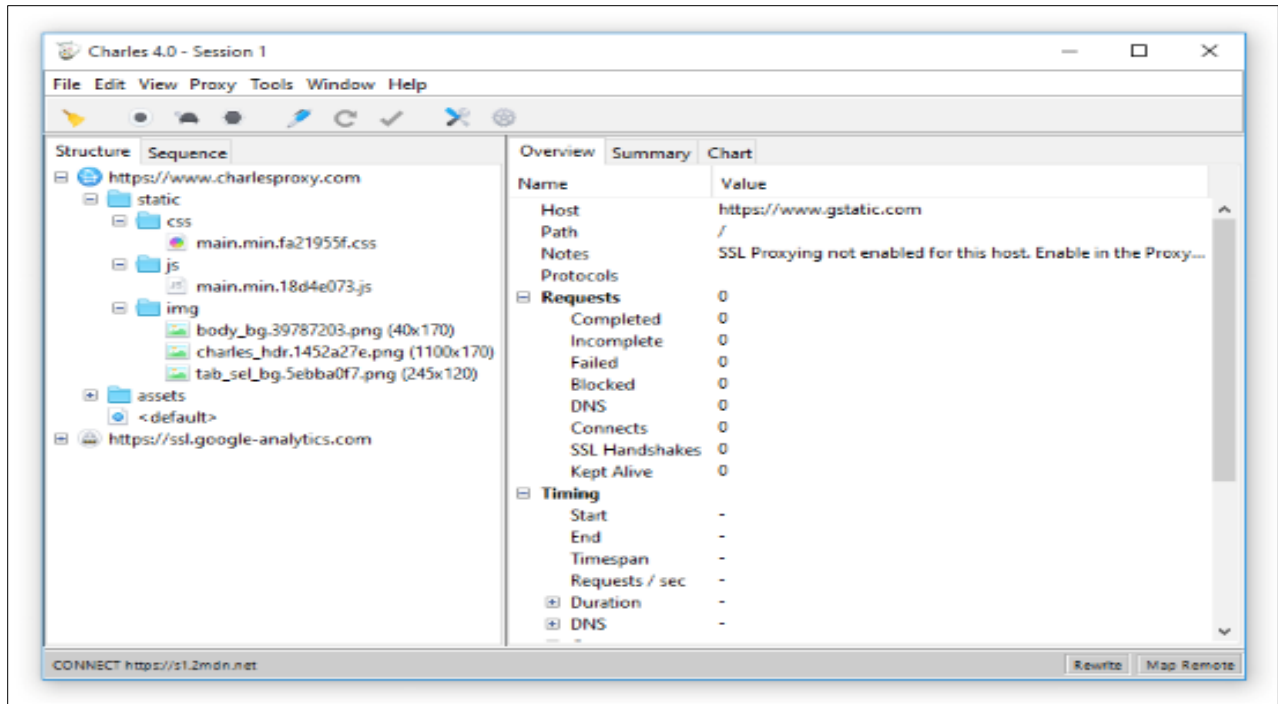


Ilustración 11 - Captura de pantalla Charles Server

Fuente: (Charles Proxy, 2019)

Una prueba muy importante en el departamento de QA son las de carga, consisten en generar múltiples llamados a los servidores que almacenan los datos o se encargan de la comunicación. Para realizar pruebas de cargas a servidores de Socket IO, se utiliza una herramienta llamada Artillery IO, permite la creación de pruebas de cargas mediante archivos de scripts fáciles de escribir y permite monitorear el estado de los llamados, también puede generar pruebas a servidores HTTP (Artillery IO, s.f.).

IV. DESARROLLO

4.1. DESCRIPCIÓN DEL TRABAJO DESARROLLADO

4.1.1. CAPACITACIÓN WEB

Los primeros tres días de práctica consistieron en tomar un curso sobre React y Redux en la plataforma de Udemy, el cual consistió en aprender cómo desarrollar aplicaciones web utilizando este marco de trabajo y con apoyo de Redux. Se realizaron varios proyectos de prueba con la intención de aprender las características importantes del marco de trabajo.

Así mismo en estos primeros días hubo una transferencia de conocimiento de un proyecto ya existente, la transferencia fue por parte de uno de los desarrolladores senior. Dicho proyecto consiste en un pequeño juego estilo trivia que utiliza React y Redux como marco de trabajo principal y tiene una conexión con un dispositivo tvOS utilizando Socket IO.

4.1.2. JUEGO TRIVIA WEB

El juego estaba desarrollado en un 95% pero con pequeños errores, durante el Sprint Planning se definieron cuáles eran los errores que necesitaban ser corregidos. Estos errores consisten en problemas de estilos CSS respecto a posiciones de imágenes y texto, así como problemas de conexión del socket. Esta fase también sirvió para poder conocer la estructura del código y también conocer cómo se lleva el flujo del desarrollo en la empresa.

La estructura general de todo el proyecto consiste en 4 partes, un servidor web, un cliente tvOS, un servidor que administra los datos y servidor de Socket IO, la mayor carga del proyecto recae en el servidor del Socket, por eso se decidió realizar pruebas de carga. El objetivo por parte del equipo de desarrollo web es configurar la fase inicial de un sistema capaz de realizar las pruebas, la asignación consistió en iniciar una prueba básica utilizando Artillery y definir cómo debe ser el flujo para futuros escenarios.

Una actualización solicitada por parte del cliente es un nuevo tipo de pregunta que consiste en ordenar cuatro imágenes, las cuales forman la secuencia de una historia animada, el reto para esta asignación consistió en no utilizar ninguna librería externa para la parte de arrastrar y soltar.

La solución se realizó utilizando solamente Javascript, se utilizaron tres eventos nativos de dispositivos móviles: touchStart (primer toque en la pantalla), touchMove (durante el movimiento del dedo en la pantalla) y touchEnd (al levantar el dedo de la pantalla). Se crearon dos tipos de objetos, los que se pueden arrastrar y las zonas donde se puede soltar. Un problema que se presentó fue el rendimiento en dispositivos con poca capacidad de procesamiento, pero se logró resolver disminuyendo las animaciones realizadas. En la Ilustración 12 se observa el prototipo de la nueva pregunta.

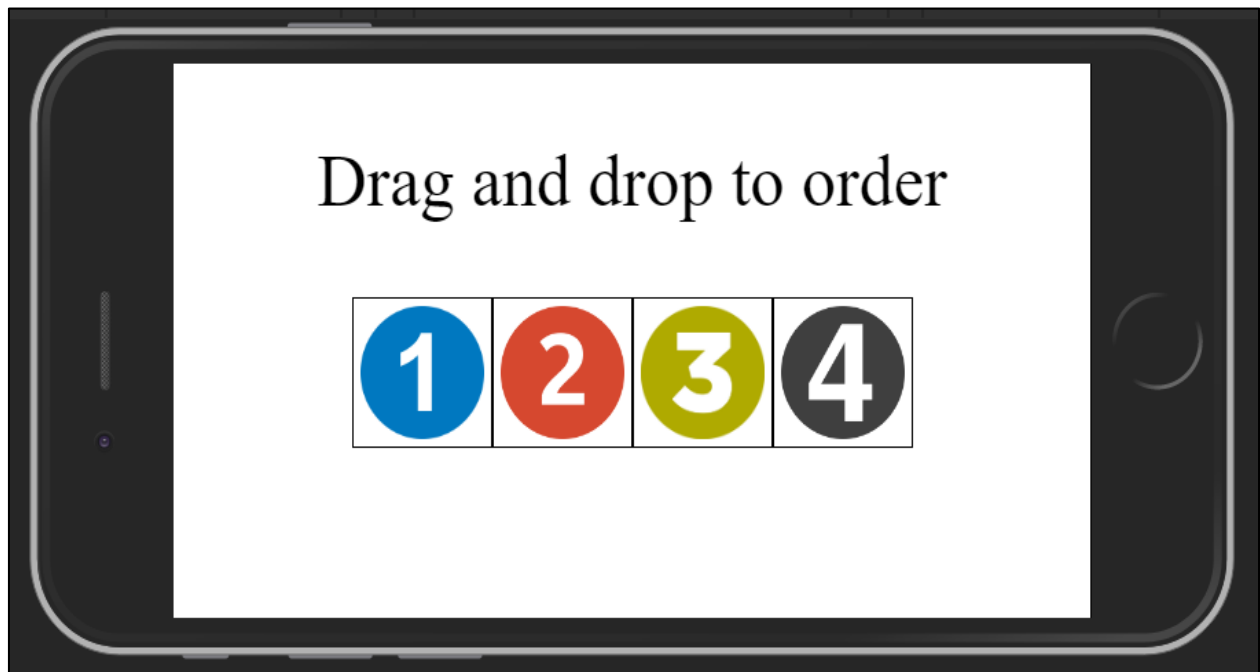


Ilustración 12 - Ejemplo Arrastrar y Soltar

Fuente: Elaboración propia

El dueño del producto sugirió que al momento de mostrar la pregunta donde se necesita ordenar las imágenes, el celular debería girarse automáticamente a modo horizontal y colocar el juego en pantalla completa. Para el desarrollo de este requerimiento los navegadores tienen una Interfaz de Programación de Aplicaciones (API) que permite ambas cosas, pero surgió un problema, el navegador Safari en los dispositivos móviles bloquea esta API por cuestión de seguridad. Se discutió entre todo el equipo cuál sería una solución, se concluyó que se mostraría un mensaje al usuario para que gire el teléfono y luego detectar cuando lo haya girado para quitar el mensaje y mostrar las imágenes a ordenar.

4.1.3. TRANSFERENCIA DE CONOCIMIENTO QA

Anteriormente se realizó una pasantía de un año y ocho meses en el área de QA en la misma empresa, durante ese tiempo el alumno desarrolló un sistema utilizando React, Redux y Charles para la automatización de las plataformas de analíticos Amplitude y Omniture. Debido al cambio de departamento el alumno tuvo que transferir el conocimiento sobre el sistema, el cual está desarrollado en un 90%, se acordó tener varias reuniones para transferir el conocimiento a otro miembro del equipo de QA. Debido a la implementación del analítico Heartbeat en el reproductor OTT, se realizó una reunión para agregar las nuevas propiedades y eventos. En la Ilustración 13 se muestra la pantalla principal del sistema de automatización de analíticos.

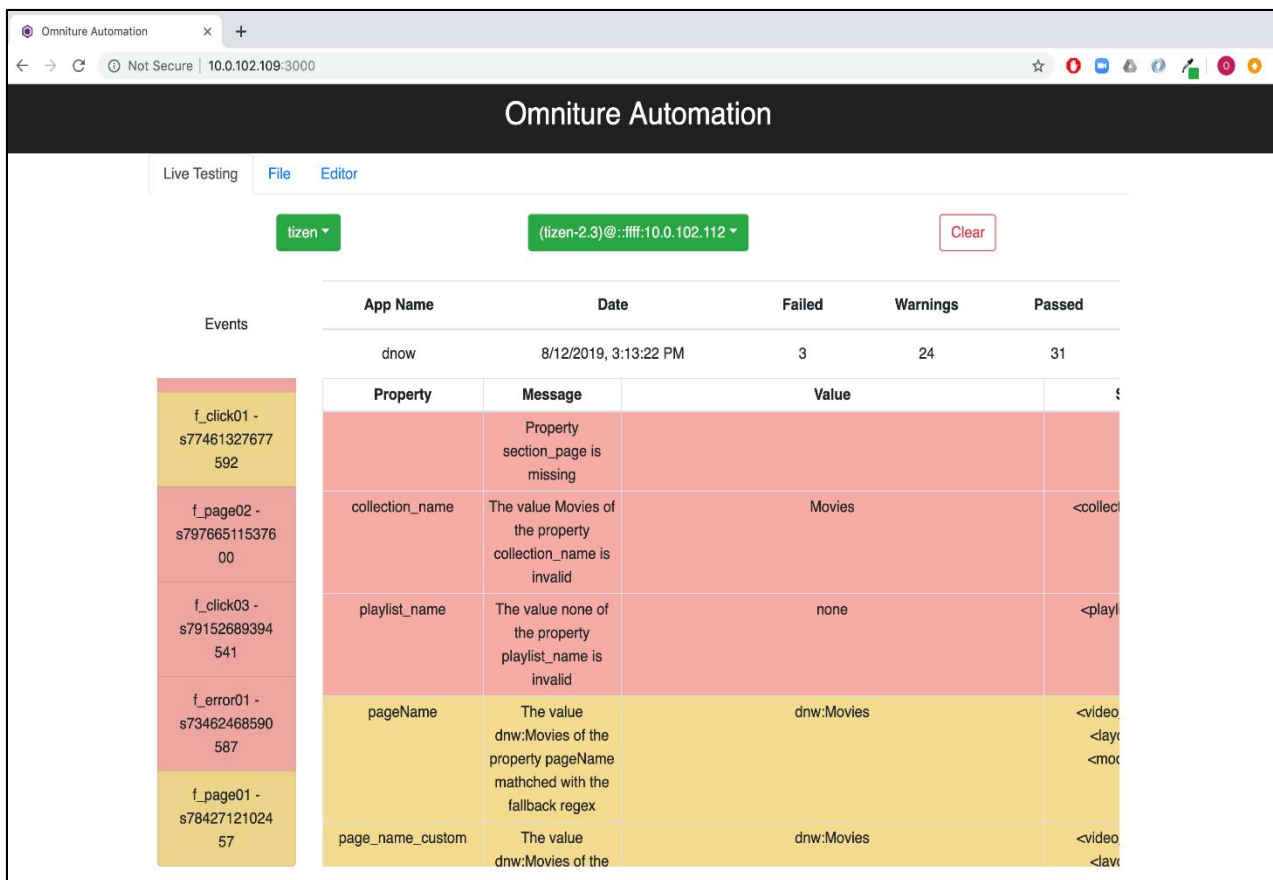


Ilustración 13 - Pantalla Principal de Sistema de Automatización Analíticos

Fuente: Elaboración propia

4.1.4. PROYECTO DE PRUEBA

Debido al prestigio de la empresa se acuerda que todo practicante antes de entrar a un proyecto de manera completa, debe desarrollar un software de prueba para mostrar sus habilidades. Dicho proyecto fue asignado por el líder del equipo. Consiste en generar un buscador de series para dispositivos Tizen. El proyecto debe ser desarrollado utilizando React y Redux y utilizando Tizen Studio como IDE. Los componentes del proyecto son: un teclado desarrollado desde cero, una lista de resultados y un reproductor. Uno de los retos planteados en este proyecto fue utilizar de manera óptima la memoria debido a que los dispositivos Tizen cuentan con una memoria limitada. Debido a la carga presentada por otros proyectos en el departamento web se decidió que no se terminará el proyecto de prueba. En la Ilustración 14 se muestra una captura del buscador de contenidos.

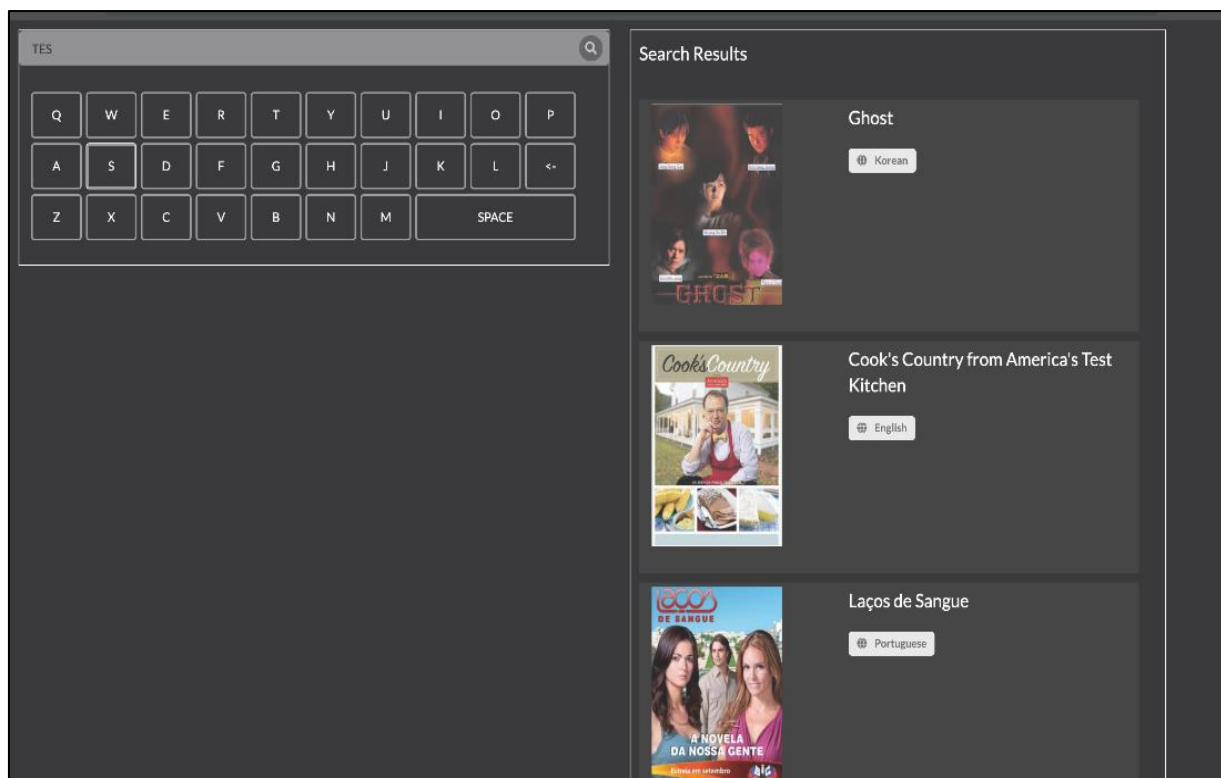


Ilustración 14 - Ejemplo Proyecto de Prueba

Fuente: Elaboración propia

4.1.5. REPRODUCTOR PARA PLATAFORMAS OTT

Un proyecto nuevo para la empresa es un reproductor para televisores de las marcas Samsung y LG. El proyecto consiste en generar una librería de un reproductor para luego solo importar e instanciar el mismo. Los requerimientos del proyecto consisten en:

- Utilizar Avplay como librería principal para la reproducción de multimedia
- Controles del reproductor y navegación mediante teclas
- Subtítulos y diferentes idiomas de audio
- Contenido en vivo y videos en demanda
- Manejo de autenticación y uso de DRM
- Analíticos
- Múltiple lenguaje de la interfaz

En la Ilustración 15 se muestra el reproductor OTT en un televisor.

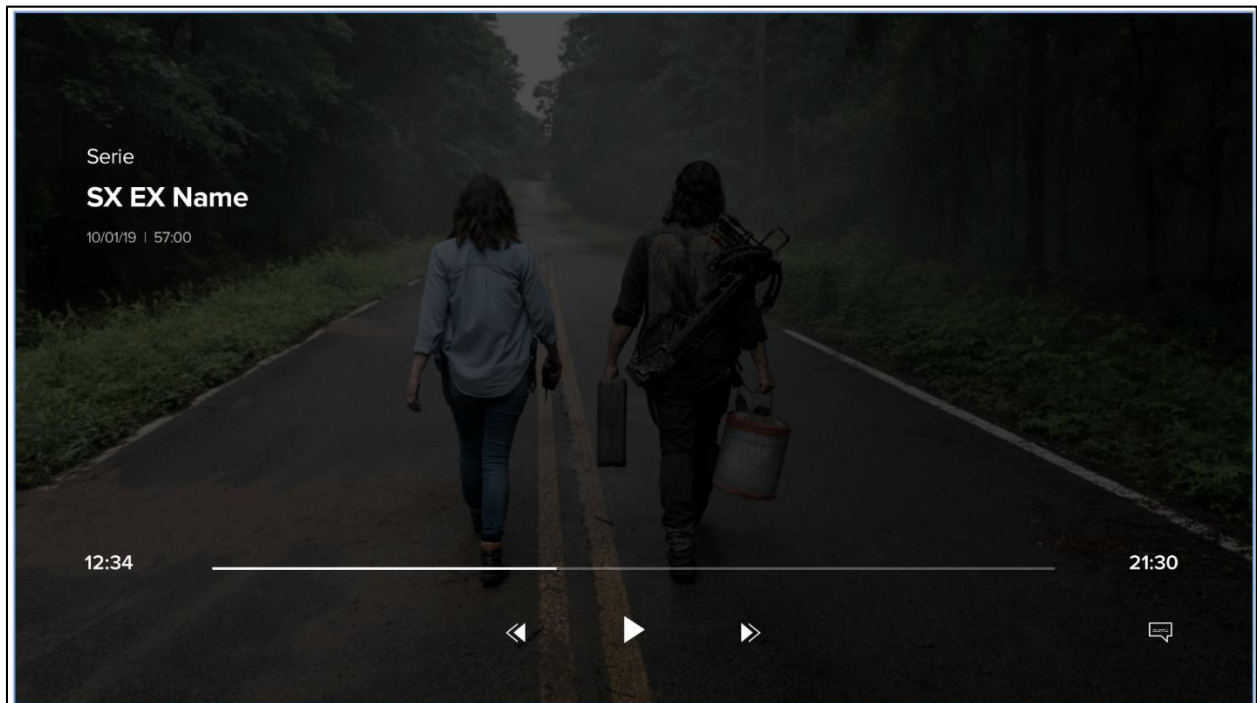


Ilustración 15 - Reproductor OTT

Fuente: Elaboración propia

Hubo una semana de investigación y experimentos para poder reproducir videos y contenido en vivo encriptados con uso de DRM utilizando el reproductor Avplay, las pruebas fueron fallidas utilizando cualquier combinación de las configuraciones de Avplay, pero luego de distintas investigaciones se concluyó que Avplay no era compatible con un tipo de encriptación AES128 y utilizando un DRM de tipo Widevine o Playready. Mientras otro miembro del equipo se encargó de esta situación, se comenzó a trabajar en la interfaz gráfica que consiste en los controles del reproductor, información del video y el manejo de diferentes subtítulos y audios de los videos. Dicho componente se desarrolló utilizando React JS.

Uno de los requerimientos principales del reproductor son los analíticos, los cuales son Omniture y Heartbeat, ambos son desarrollados por Adobe, un analítico consta de dos objetos, los eventos y propiedades, cada evento contiene múltiples propiedades. La lógica entre ellos es muy similar sólo se diferencian por sus valores, entonces se definió crear un componente padre y luego otros dos que heredan de él, de esta manera se simplificaría el código y el uso de ellos. Un problema sobre estos analíticos en particular es la poca documentación que tiene el SDK, muchas veces se tenía que analizar el código del SDK en lugar de leer la documentación proporcionada por Adobe. Los eventos son acciones realizadas por el usuario como iniciar, pausar, reanudar o terminar un video, pero también hay otros automáticos, por ejemplo, uno que se envía cada 10 segundos y otro que se envía en cada cuartil.

Debido a que el reproductor será utilizado en todo América, se tiene que tener soporte para los lenguajes de español, inglés y portugués. Para soportar varios idiomas se decidió usar la Estandarización para multilinguaje (I18N), básicamente se creó un componente capaz de soportar múltiples diccionarios donde cada uno de ellos representa un idioma y que cada uno está formado por un mapa, la llave es una cadena de texto único que representa el texto a mostrar y el valor es el texto traducido según el idioma seleccionado por el usuario. En la Ilustración 16 se muestra el menú de selección de subtítulos y audios.

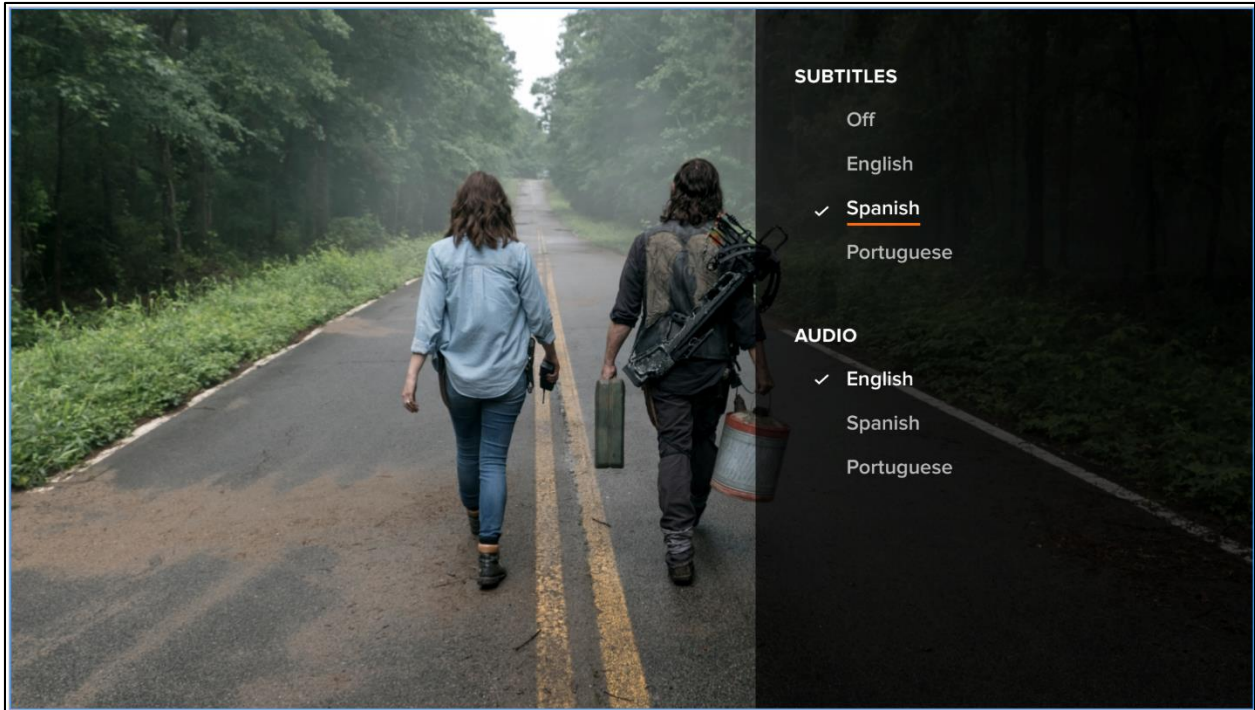


Ilustración 16 - Menú multilinguaje reproductor OTT

Fuente: Elaboración propia

4.1.6. DESARROLLO SISTEMA WEB CON SSR

A medida que la carga de trabajo del reproductor disminuía se solicitó colaboración para el desarrollo de un sistema web con SSR, dicho proyecto consistía en formularios y páginas de información, la complejidad era la parte del SSR, por lo cual se asignó un día exclusivamente para el estudio de esta tecnología y también transferencia de conocimiento de una persona que había estado trabajando en ello. Al momento de crear el proyecto se había decidido tener dos servidores, uno encargado para el Backend y otro para el Frontend, pero se discutió migrar todo a solo un servidor. Luego de realizar esta migración se continuó con el desarrollo de las nuevas funcionalidades solicitadas por el cliente, una de ellas es la implementación de un código para poder ingresar al sitio días antes de su lanzamiento. La manera cómo se desarrolló fue:

- Se creó un modelo en la base de datos para almacenar los códigos encriptados
- En el Frontend se hizo un formulario que se muestra la primera vez que se ingresa al sitio, el cual contiene una caja de texto y botón donde el usuario debe ingresar el código de acceso, luego al presionar el botón se realiza un llamado al Backend donde encripta el

código ingresado y lo compara con el almacenado en la base de datos, en caso de coincidir se redirige al usuario a la página principal

- Se revisa si el código es válido en cada llamado desde el Frontend hasta el Backend.

Los demás cambios consistían en corrección de errores de estilo y agregar nuevo contenido estático al sitio. Por cuestión de confidencialidad con el cliente no se puede adjuntar capturas del sitio o brindar el nombre de la aplicación.

4.1.7. SITIO WEB DE NOTICIAS

The Recount, es un sitio estadounidense gratuito exclusivo para videos cortos de noticias políticas, ICOMS está encargado del desarrollo del Frontend, utiliza React JS y Next JS. El alcance del proyecto consta de:

- Una página principal con una lista de videos
- Un reproductor de video
- Un menú lateral
- 5 páginas estáticas.

La dificultad proviene por el uso de SSR y que el sitio debe ser compatible con cualquier navegador en cualquier plataforma Mac, Windows, iOS y Android. En la Ilustración 17 se muestra el sitio en una computadora.

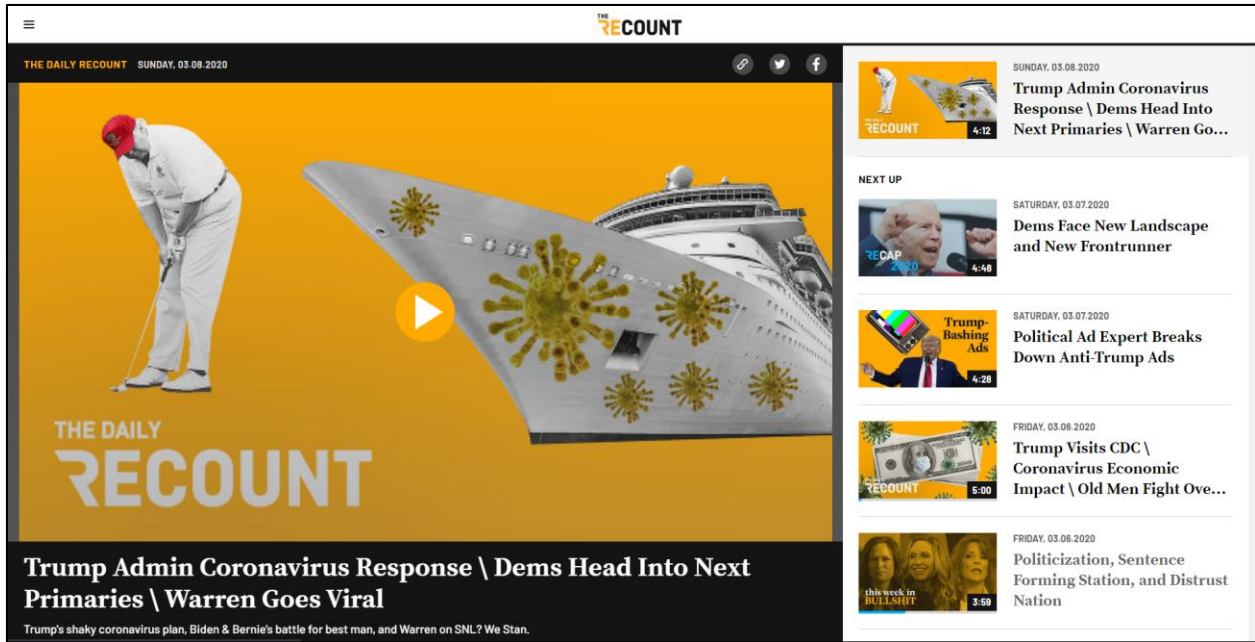


Ilustración 17 - The Recount versión de escritorio

Fuente: Elaboración propia

En la Ilustración 18 se muestra el sitio en un dispositivo móvil.

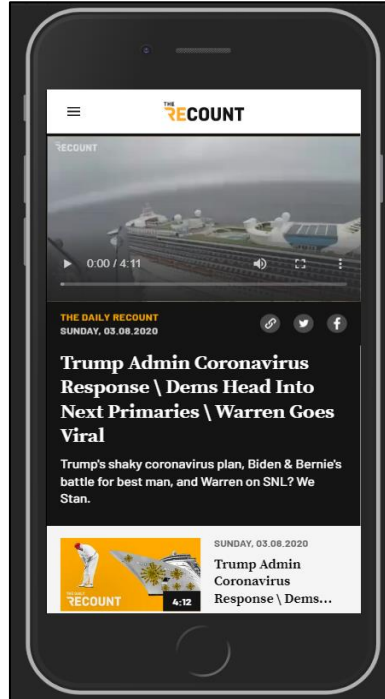
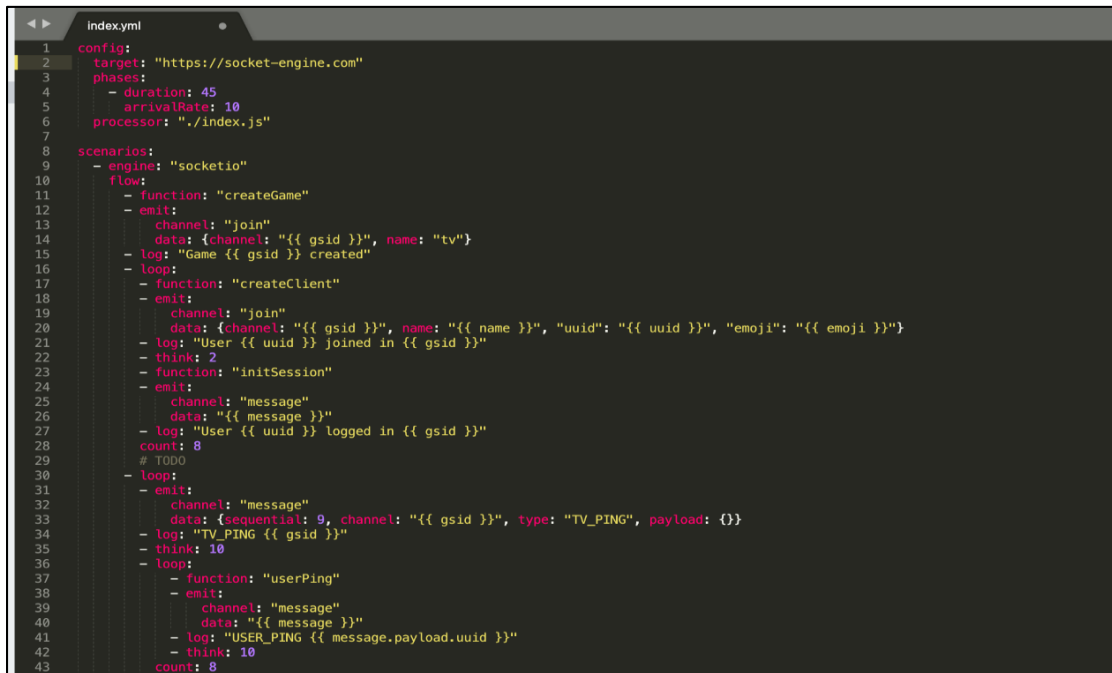


Ilustración 18 - The Recount versión móvil

Fuente: Elaboración propia

4.1.8. PROBADOR DE CARGA DE SERVER SOCKET IO

El juego de trivia contiene un servidor de Socket IO del cual se necesita conocer cuánto es su capacidad en cuanto a cantidad de clientes conectados y cantidad de solicitudes, por eso se tuvo que crear un probador de carga utilizando Artillery. En la Ilustración 19 se muestra el ejemplo de un script de Artillery IO.



```
1 config:
2   target: "https://socket-engine.com"
3   phases:
4     - duration: 45
5       arrivalRate: 10
6     processor: "./index.js"
7
8   escenarios:
9     - engine: "socketio"
10      flow:
11        - function: "createGame"
12          emit:
13            channel: "join"
14            data: {channel: "{{ gsid }}", name: "tv"}
15          log: "Game {{ gsid }} created"
16        - loop:
17          - function: "createClient"
18            emit:
19              channel: "join"
20              data: {channel: "{{ gsid }}", name: "{{ name }}", "uuid": "{{ uuid }}", "emoji": "{{ emoji }}"
21            log: "User {{ uuid }} joined in {{ gsid }}"
22            think: 2
23          - function: "initSession"
24            emit:
25              channel: "message"
26              data: "{{ message }}"
27            log: "User {{ uuid }} logged in {{ gsid }}"
28          count: 8
29          # TODO
30        - loop:
31          - emit:
32            channel: "message"
33            data: {sequential: 9, channel: "{{ gsid }}", type: "TV_PING", payload: {}}
34            log: "TV_PING {{ gsid }}"
35            think: 10
36          - loop:
37            - function: "userPing"
38              emit:
39                channel: "message"
40                data: "{{ message }}"
41              log: "USER_PING {{ message.payload.uuid }}"
42              think: 10
43            count: 8
```

Ilustración 19 - Ejemplo Artillery

Fuente: Elaboración propia

4.2. CRONOGRAMA DE ACTIVIDADES

En la Ilustración 20 se puede observar el cronograma de trabajo utilizado en el desarrollo de las actividades realizadas durante la práctica profesional

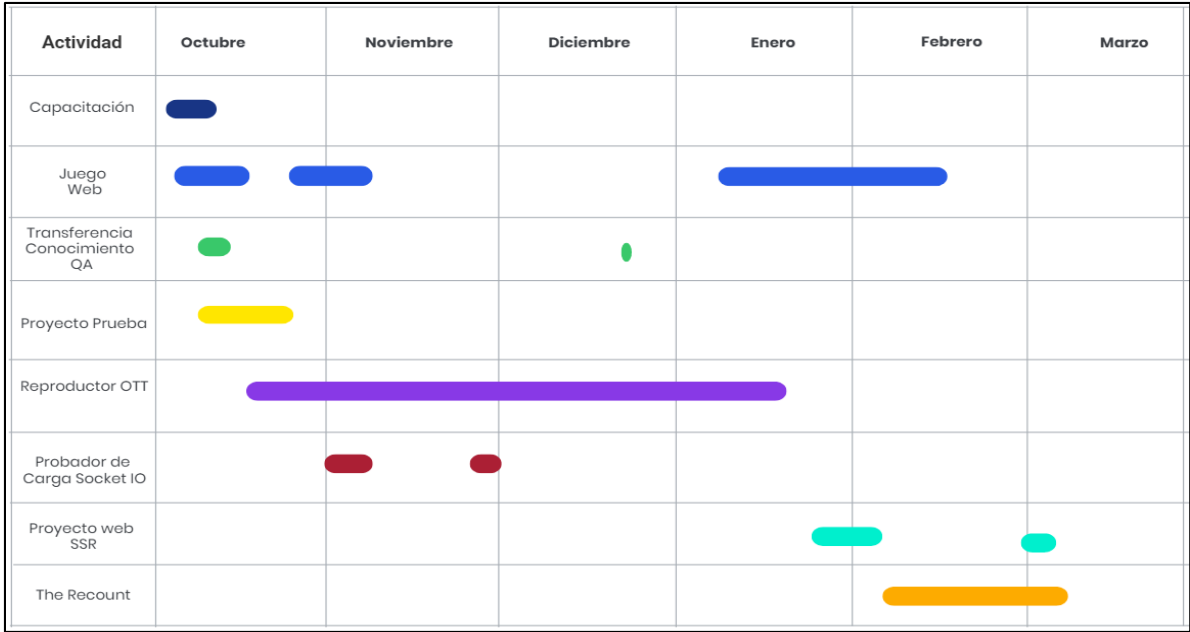


Ilustración 20 - Diagrama de Gantt

Fuente: Elaboración propia

V. CONCLUSIONES

- Se desarrollaron módulos como una interfaz gráfica de controles, soporte para múltiples subtítulos y audios de videos, implementación de i18n y manejo de analíticos para un reproductor web para las plataformas Samsung y LG. Para el desarrollo se utilizó React Redux y el reproductor Shaka.
- Se tomó un curso sobre React y Redux, aparte de ello se logró realizar un proyecto de prueba para la plataforma Samsung utilizando Tizen Studio.
- Se participó todos los días en una reunión de 15 minutos donde se discutió que se hizo el día anterior, qué hará el día de hoy y cuáles son los problemas actuales con la actividad asignada. También se participó en dos reuniones quincenales para presentar a los clientes el trabajo realizado en dos semanas y la otra reunión es para discutir qué se podría mejorar y qué se hizo bien.
- Se entregó la base del proyecto de automatización de analíticos que consiste en la validación automática del formato de las propiedades de los eventos utilizados para el análisis de datos y también se definieron las fases futuras que podría tener el proyecto, el cual está desarrollado utilizando Node JS y React; para poder realizar el proxy se utiliza la herramienta Charles.

VI. RECOMENDACIONES

- En caso de que un miembro del equipo esté en dos proyectos simultáneamente, brindar la distribución del tiempo para cada uno de ellos.
- Colaborar al departamento de QA brindando cursos y soporte sobre desarrollo para que puedan crear herramientas que faciliten y automaticen las pruebas.

BIBLIOGRAFÍA

1. Adobe. (s.f. de s.f. de 2019). *Adobe*. Recuperado el 10 de Noviembre de 2019, de Adobe: https://www.adobe.com/marketing/experience-manager-assets/digital-rights-management.html?gclid=CjwKCAiAh5_uBRA5EiwASW3IakNa9RezvLcgVtT3B0S6-MnHCxE4cO_HrAspHdIDRX2rjeZsY00U4RoCpgQQAuD_BwE&s_cid=7011O0000028JKJQA2&sdid=322TBYTC&mv=search&edtamo=true&ef_id=C
2. Adobe Corp. (4 de Diciembre de 2019). *Adobe*. Recuperado el 21 de Diciembre de 2019, de Adobe Analytics for Audio and Video: <https://docs.adobe.com/content/help/en/media-analytics/using/media-overview.html#>
3. Alvarez, M. A. (25 de Febrero de 2019). *Artículos: Desarrollo Web*. Recuperado el 20 de Octubre de 2019, de Desarrollo Web: <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>
4. Amplitude. (s.f. de s.f. de s.f.). *Amplitude*. Recuperado el 10 de Noviembre de 2019, de Amplitude: <https://amplitude.com/>
5. Artillery IO. (s.f. de s.f. de s.f.). *Artillery IO*. Recuperado el 9 de Noviembre de 2019, de Artillery IO: <https://artillery.io/>
6. Atlassian. (s.f. de s.f. de 2020). *Atlassian Jira Software*. Recuperado el 03 de Febrero de 2019, de Atlassian: https://www.atlassian.com/software/jira/?aceid=&adposition=1t1&adgroup=63127837864&campaign=1439934854&creative=340023072955&device=c&keyword=jira&matchtype=e&network=g&placement=&ds_kids=p34164036290&ds_e=GOOGLE&ds_eid=70000001558501&ds_e1=GOOGLE&gclid=
7. Cámara de Comercio Bogotá. (s.f. de Julio de 2018). *Cámara de Comercio Bogotá*. Recuperado el 9 de Noviembre de 2019, de Cámara de Comercio Bogotá: <https://www.ccb.org.co/Clusters/Cluster-de-Musica/Noticias/2018/Julio-2018/El-crecimiento-del-Streaming-en-los-ultimos-10-anos>

8. Charles Proxy. (s.f. de s.f. de 2019). *Charles Proxy*. Recuperado el 10 de Noviembre de 2019, de Charles Proxy: <https://www.charlesproxy.com/>
9. Crunch Base. (s.f. de s.f. de s.f.). *Crunch Base*. Recuperado el 10 de Noviembre de 2019, de Crunch Base: <https://www.crunchbase.com/organization/omniture#section-overview>
10. Delgado, A. (18 de Septiembre de 2008). *Consumer*. Recuperado el 9 de Noviembre de 2019, de Consumer: <https://www.consumer.es/tecnologia/internet/que-es-el-streaming.html>
11. Desarrollo Web. (14 de Septiembre de 2018). *Desarrollo Web*. Recuperado el 10 de Noviembre de 2019, de Desarrollo Web: <https://desarrolloweb.com/articulos/axios-ajax-cliente-http-javascript.html>
12. Educba. (s.f. de s.f. de 2019). *Educba*. Recuperado el 10 de Noviembre de 2019, de Educba: <https://www.educba.com/websocket-vs-socket-io/>
13. Ferguson, N. (s.f. de s.f. de s.f.). *Career Foundry*. Recuperado el 29 de Febrero de 2020, de Career Foundry: <https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/>
14. G&L Group. (s.f. de s.f. de s.f.). *G&L Group*. Recuperado el 10 de Noviembre de 2019, de G&L Group: <https://www.gylgroup.com/novedades/importancia-quality-assurance>
15. Galiano, J. M. (2016). *Implantar scrum con éxito*. Editorial UOC. Recuperado el 20 de Octubre de 2019
16. Git. (s.f. de s.f. de s.f.). *Git*. Recuperado el 28 de Octubre de 2019, de Git: <https://git-scm.com/about>
17. Google. (4 de Febrero de 2020). *Shaka*. Recuperado el 9 de Febrero de 2020, de Shaka: <https://shaka-player-demo.appspot.com/docs/api/index.html>
18. Heroku. (s.f. de s.f. de 2020). *Platform: Heroku*. Recuperado el 9 de Febrero de 2019, de Heroku: <https://www.heroku.com/platform>

19. ICOMS. (s.f. de s.f. de s.f.). *About US, Hello Iconic*. Recuperado el 20 de Octubre de 2019, de Hello Iconic: <https://helloiconic.com/about-us/>
20. JSON. (s.f. de s.f. de s.f.). *JSON Org*. Recuperado el 28 de Octubre de 2019, de JSON Org: <http://www.json.org/>
21. Lingport. (s.f. de s.f. de s.f.). *Lingport*. Recuperado el 3 de Enero de 2019, de Lingport: <https://lingoport.com/what-is-i18n/>
22. Mathis, B., & Vega, I. (2018). *Agile Project Management para Principiantes: Dominar los Conocimientos Básicos con Scrum*. AP Publishing. Recuperado el 20 de Octubre de 2019
23. Mozilla Developers. (18 de Marzo de 2019). *MDN Web Docs*. Recuperado el 9 de Noviembre de 2019, de MDN Web Docs: <https://developer.mozilla.org/es/docs/Web/HTTP>
24. Next. (s.f. de s.f. de 2019). *Next*. Recuperado el 02 de Febrero de 2020, de Next: <https://nextjs.org/features/server-side-rendering>
25. Node.js Foundation. (s.f. de s.f. de s.f.). *About: Node JS*. Recuperado el 10 de Noviembre de 2019, de Node JS: <https://nodejs.org/es/about/>
26. OBS Business School. (s.f. de s.f. de s.f.). *OBS Business School*. Recuperado el 1 de Marzo de 2020, de OBS Business School: <https://obsbusiness.school/es/blog-project-management/scrum/principales-roles-de-la-metodologia-agil-scrum>
27. Redux Org. (s.f. de s.f. de s.f.). *Introduction: Redux*. Recuperado el 10 de Noviembre de 2019, de Redux Org: <https://es.redux.js.org>
28. Reforma. (s.f.). Tienen más oportunidades en el desarrollo web. *Reforma*, pág. 28. Recuperado el 20 de Octubre de 2019
29. Samsung Developers. (s.f. de s.f. de 2019). *Samsung Developers*. Recuperado el 10 de Noviembre de 2019, de Samsung Developers: <https://developer.samsung.com/tv/develop/api-references/samsung-product-api-references/avplay-api>

30. Seobility. (s.f. de s.f. de s.f.). *Seobility*. Recuperado el 29 de Febrero de 2020, de Seobility: <https://www.seobility.net/de/wiki/Frontend>
31. Stevanoski, H. (30 de Agosto de s.f.). *Medium*. Recuperado el 9 de Noviembre de 2019, de Medium: <https://medium.com/javascript-in-plain-english/the-only-introduction-to-redux-and-react-redux-youll-ever-need-8ce5da9e53c6>
32. TechoPedia. (s.f. de s.f. de s.f.). *TechoPedia*. Recuperado el 10 de Noviembre de 2019, de TechoPedia: <https://www.techopedia.com/definition/29493/analytics-platform>
33. Tizen. (s.f. de s.f. de 2012). *Tizen Org*. Recuperado el 10 de Noviembre de 2019, de Tizen Org.: <https://www.tizen.org/about>
34. Udemy. (s.f. de s.f. de 2020). *Udemy*. Recuperado el 8 de Febrero de 2019, de Udemy: <https://www.udemy.com/>
35. Webpack. (s.f. de s.f. de s.f.). *Webpack*. Recuperado el 10 de Noviembre de 2019, de Webpack: <https://webpack.js.org/>