Novoclick Teatro

## ¿Qué vamos a utilizar y por qué?

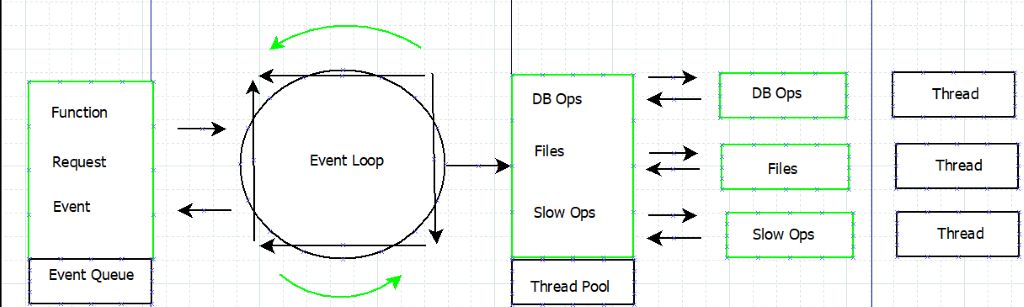
en primer lugar hablaremos de las principales tecnologías que usaremos para el desarrollo y buen desempeño del sistema

M.E.R.N stack ¿en qué consiste esta tecnología?



Empezaremos diciendo que MERN es uno de los stack tecnológicos más usados en la actualidad, utiliza las tecnologías más ágiles y rápidas, del lado del backend estaremos utilizando express.js el cual es un marco de aplicación de node.js, está basado casi en su totalidad en middlewares, las cuales son pequeñas piezas de código que facilitan el intercambio y la interacción con otros paquetes o programas, gracias a esto implementaremos de una manera fácil las rutas y protocolos https, esto último para que las rutas de nuestro sistema sean rápidas y muy seguras a su vez.

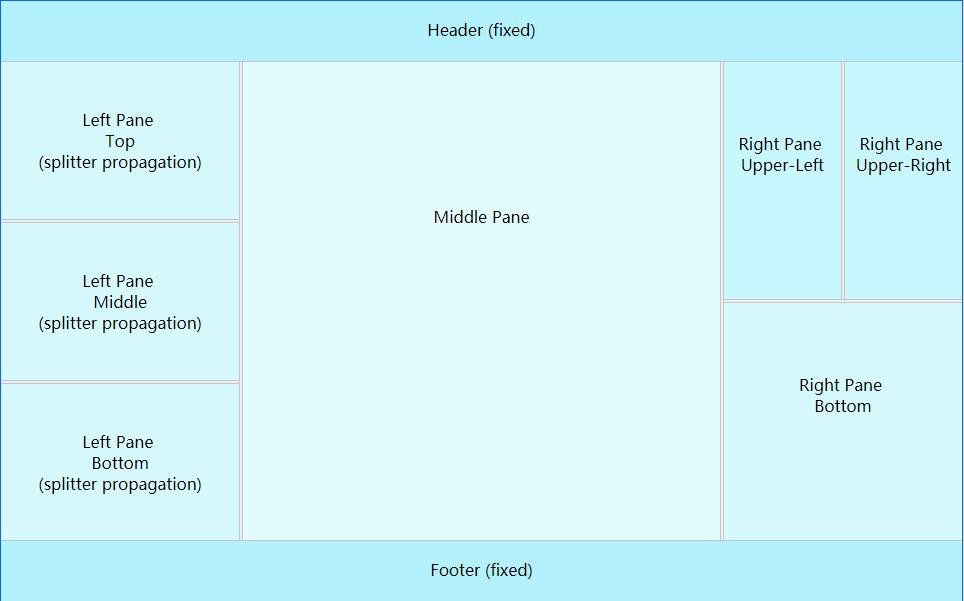
así mismo, como motor principal del servidor utilizaremos node.js y toda su asincronía, la cual nos permitirá manejar los eventos de la aplicación de manera asíncrona, evitando así problemas de bloqueos y sobrecarga de tareas al servidor, acá mostraremos gráficamente cómo funciona el bucle de eventos asíncronos de node.js.



por otra parte en cuanto al manejo de los datos, implementaremos otra gran tecnología que forma parte del MERN stack la cual es mongoDB, nos ayudara en toda la implementación y estructuración de la base de datos, mediante una estructura sencilla y fácil de usar podremos gestionar, guardar eliminar y consultar datos, casi de manera automática, cabe destacar que al ser una base de datos no relacional es muy fácil la escalabilidad y reestructuración de los datos, lo cual es una gran ventaja pensado a corto y largo plazo.

A su vez, del lado del front MERN stack utiliza react.js para desplegar toda la parte gráfica y de interactividad con el sistema, react.js es una biblioteca de javascript potente, la cual nos permite el desarrollo de single page application (SPA) de forma sencilla. A su vez react nos ofrece la posibilidad de diseñar nuestro sistema basado en componentes de fácil realización, los cuales pueden ser compartidos o reutilizados en diferentes áreas o vistas de la aplicación, evitando así complejizar el código, basándonos en lo verdaderamente importante, generar interacción cliente-aplicación eficaz rápida y fácil de usar para cualquier usuario.

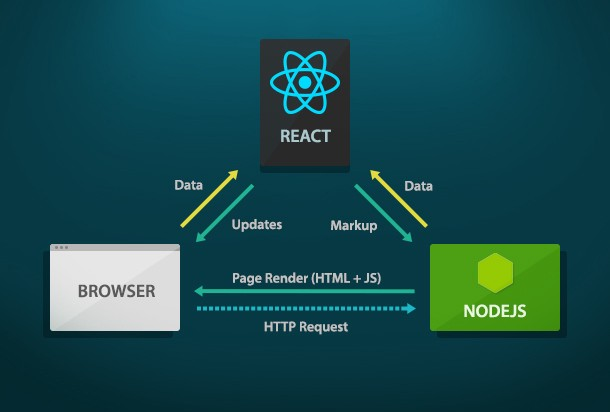
ejemplo de cómo se reutilizan los componentes en react.js



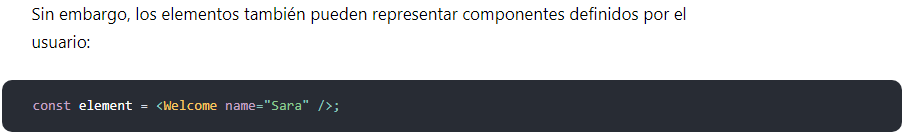
todos los recuadros azules son un componente de la página que pueden ser estructurados y reestructurados de manera que si la página tiene varias vistas(“páginas”) va a poder ser importada fácilmente evitando así la complejidad del código agilizando la arquitectura,

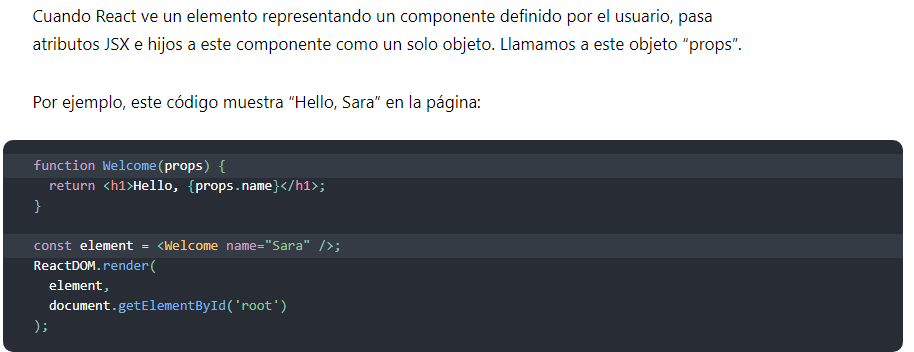
cabe destacar que al ser una single page application, el servidor no carga una página como tal, solo carga los componentes necesarios de manera asíncrona agilizando la carga y servicios a consumir, cabe destacar que la carga de estos componentes se hacen del lado del servidor(“server-side-render”)evitando así la recarga de la página en el navegador (“client-side-render”), esta característica de react sirve de manera más rápida la aplicación al usuario disminuyendo casi en su totalidad el tiempo de espera para usar la aplicación , o mostrar algún elemento, es una gran ventaja para todo el SEO y posicionamiento en buscadores.

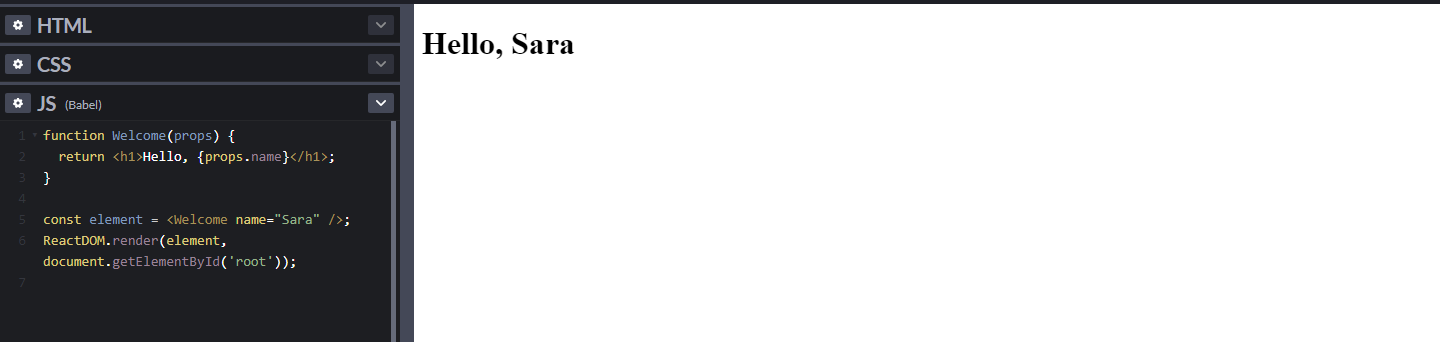
A continuación, vamos a dar un ejemplo más descriptivo técnicamente y gráfico











En el lado derecho se evidencia lo que vería el usuario final en su pantalla

En general su función implica

1. Llamamos a ReactDOM.render() con el elemento Welcome name="Sara" />.
2. React llama al componente Welcome con {name: 'Sara'} como “props”.
3. Nuestro componente Welcome devuelve un elemento <h1>Hello, Sara</h1> como resultado.
4. React DOM actualiza eficientemente el DOM para que coincida con <h1>Hello, Sara</h1>.

React native / Móvil

Esta tecnología será la utilizada para la creación de la aplicación móvil, al estar utilizando MERN stack del lado web podemos reutilizar todos los componentes y parte del backend, esto nos permite tener una misma arquitectura en ambos sistemas.

React native nos ofrece la facilidad de escribir aplicaciones híbridas sin realizar un código para cada sistema operativo ya sea Android o IOS , con tan solo utilizar la propiedad platform react native prácticamente se adapta al sistema operativo que se está ejecutando manteniendo todas y cada una de las características híbridas de cada sistema operativo

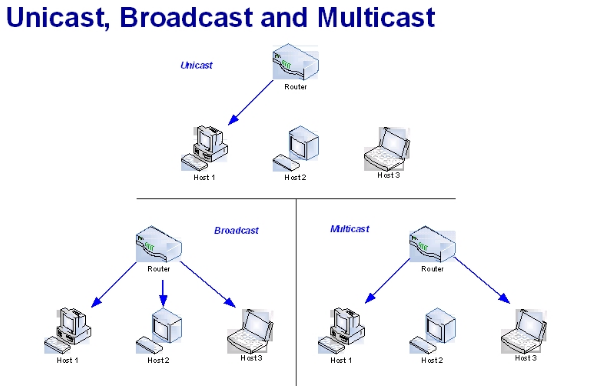
Otras tecnologías importantes que se van a utilizar PWA

Las PWA nos ofrecen la capacidad de migrar una aplicación web a híbrida, tomando a su vez características nativas del sistema operativo del móvil en el que se esté instalando. Así mismo ofrece mejoras en la carga de contenido no perdiendo muchas de las cualidades o características propias nativas de los sistemas

Websocket

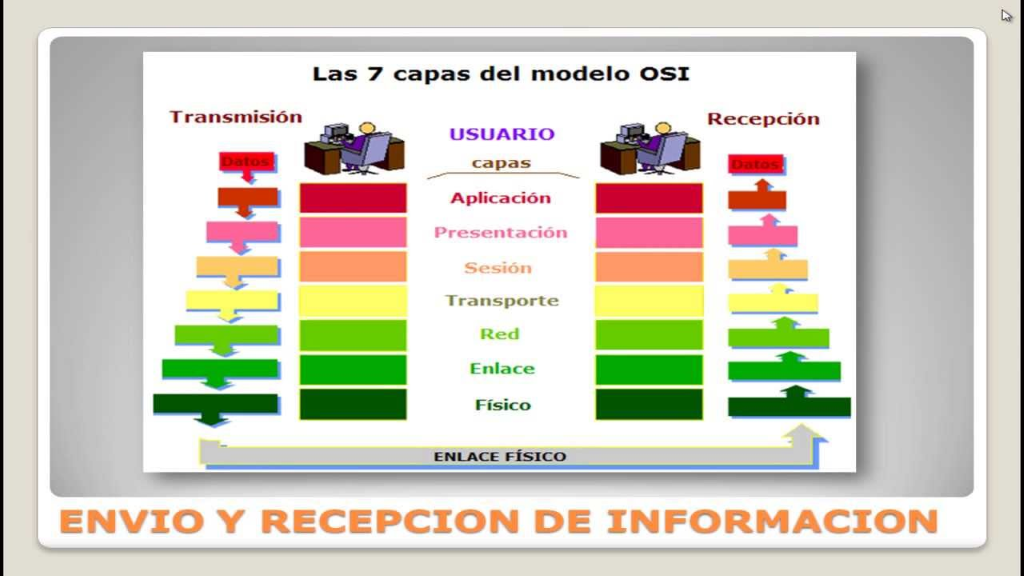
Utilizaremos esta tecnología para toda la parte de comunicación ofreciéndole a los usuarios un canal único seguro y rápido para que puedan comunicarse entre sí ya sea entre clientes y administradores o pudiendo dejar un mensaje durante la trasmisión en vivo, así mismo poder interactuar con otros usuarios, todo esto de manera segura.

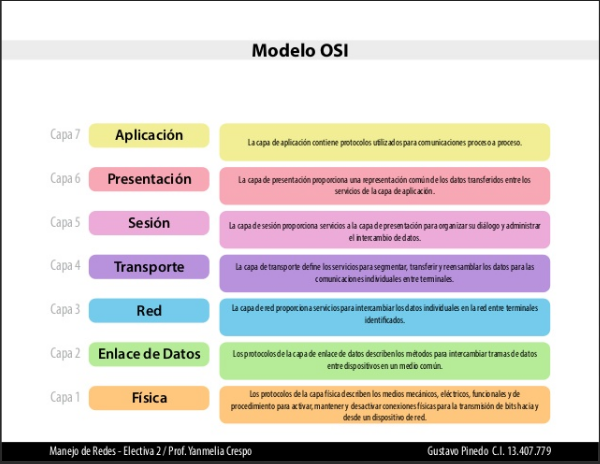
Con la implementación de websocket además de todas estas grandes ventajas de seguridad estamos evitando consumir recursos, hacerle muchas peticiones al servidor, evitando fallos a la hora de enviar y recibir paquetes ya que usa varios protocolos para administrar y renderizar sus recursos físicos y virtuales



Modelo OSI

El modelo OSI está conformado por 7 capas o niveles de abstracción. Cada uno de estos niveles tendrá sus propias funciones para que en conjunto sean capaces de poder alcanzar su objetivo final. Precisamente esta separación en niveles hace posible la intercomunicación de protocolos distintos al concentrar funciones específicas en cada nivel de operación.



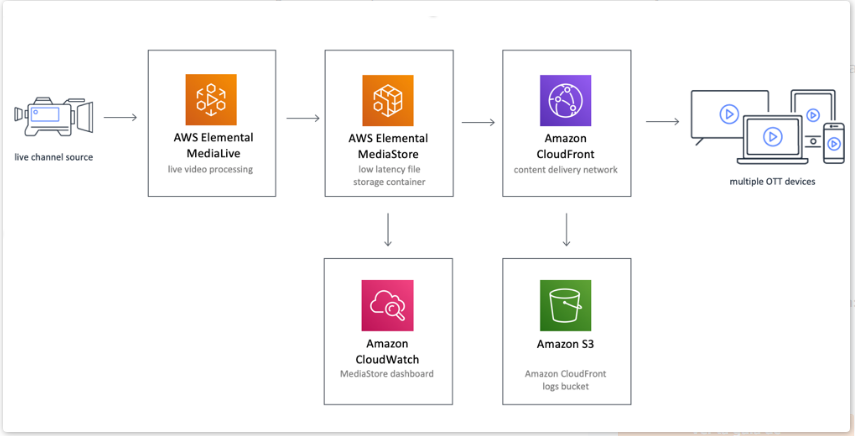


JWT

Utilizaremos JWT tokens para brindar aún más seguridad y gestionar los diferentes roles de usuarios, gracias a este servicio además de cada usuario tener un id único, nuestro sistema también responderá al envío y recepción de tokens a la hora de registro y loggeo , paralelamente también usaremos JWT para gestionar el acceso a las salas el usuario necesitará este ticket único para poder acceder a la sala,este ticket tendrá una fecha de vencimiento generada por el sistema.

amazon web service

AWS ofrece una solución de streaming en directo que combina AWS Elemental MediaLive y AWS Elemental MediaStore con Amazon CloudFront para crear una arquitectura con alto nivel de resiliencia y escalabilidad que distribuye su contenido en directo a todo el mundo. El siguiente esquema muestra la arquitectura de streaming de video en directo que puede implementar automáticamente con la guía de implementación de la solución y la plantilla de AWS CloudFormation correspondiente. Es importante resaltar que gracias a la adquisición de este servicio podemos tener un panel de administración como el que se puede gestionar todo el flujo multimedia desplegado en el cloud.



Arquitectura de la aplicación live streaming en AWS

La solución implementa [AWS Elemental MediaLive](https://aws.amazon.com/es/medialive/), que incorpora dos fuentes de entrada y transcodifica el contenido en dos transmisiones de HTTP Live Streaming (HLS) de tasa de bits adaptativa como salida.

[AWS Elemental Media Package](https://aws.amazon.com/es/mediapackage/) incorpora la salida de MediaLive y empaqueta la transmisión en directo en formatos de HLS, streaming dinámico adaptativo a través de HTTP (DASH), Microsoft Smooth Streaming (MSS) y Common Media Application Format (CMAF) que se entregan a cuatro puntos de enlace personalizados de Media Package.

Se configura una distribución de [Amazon CloudFront](https://aws.amazon.com/es/cloudfront/) para usar los puntos de enlace personalizados de Media Package como origen. La distribución de CloudFront entrega la transmisión en directo a los espectadores con baja latencia y velocidades de transferencia altas.

La solución también incluye la demostración de un reproductor de vista previa de HTML que puede usar para probar la solución. El reproductor es un sitio web estático alojado en un bucket de [Amazon Simple Storage Service](https://aws.amazon.com/es/s3/) (Amazon S3). Amazon CloudFront se utiliza para restringir el acceso al contenido del bucket del sitio web de la solución.

Características de la aplicación live streaming en AWS

Live Streaming en AWS proporciona configuraciones automáticas

-Configure AWS Elemental MediaLive y AWS Elemental Media Package de forma automática para codificar y empaquetar el contenido para el streaming de tasa de bits adaptable en varias pantallas.

Plantilla completa de AWS CloudFormation

La plantilla de AWS CloudFormation de la solución lanza los servicios y los productos de AWS necesarios para incorporar, transcodificar y entregar streaming de vídeo en directo.

soluciones flexibles

-Configure la solución para la incorporación de transmisiones de protocolo de transporte en tiempo real (RTP), protocolo de mensajería en tiempo real (RTMP) y HLS, o de flujos de MediaConnect como entrada. Hay tres perfiles de codificación disponibles durante la configuración inicial.

consumo de tecnología Api rest

En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios profesionales a partir de ese software. Twitter, YouTube, los sistemas de identificación con Facebook… hay cientos de empresas que generan negocio gracias a REST y las APIs REST. Sin ellas, todo el crecimiento en horizontal sería prácticamente imposible. Esto es así porque REST es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet.

