

Universidad Luterana Salvadoreña
Facultad de Ciencias del Hombre y la naturaleza
Licenciatura en Ciencias de la Computación
Cátedra Nuevas Tendencias De programación



TEMA:

“Proyecto App de Película”

Docente: Lic. Rafael Díaz

Ciclo/Año: II/2022

Integrantes:

Elsi María Quesada Arenivar	QA01135709
Diana Carolina Barillas Vásquez	BV01135639
Nathaly Rosmery Zelaya Flores	ZF01135967
Ángel Noe Rivera Rivas	RR01135891
Alberto Antonio Martínez Linares	ML01135708

San Salvador, 17 de noviembre del

2022

índice

CAPÍTULO I MARCO REFERENCIAL

1.1 Antecedentes

Flix olé la plataforma streaming de películas, se le brindará al usuario una experiencia excelente con la búsqueda de sus películas favoritas, dicho sitio será de entretenimiento y el tráfico de las personas que navegan generará un interés por lo cual habrá quienes estén interesados en poner pequeños anuncios publicitarios.

Misión y visión de la Empresa.

Misión

Ofrecer a nuestros usuarios un amplio catálogo de entretenimiento generando popularidad y brindando respuestas rápidas de petición de búsqueda de los usuarios.

Visión.

Como sitio de entretenimiento flix olé se ofrecerá las búsquedas más acertadas de y variedad de catálogo de entretenimiento disponible.

1.2. Planteamiento del problema

La dificultad en los momentos que se busca entretenimiento, para eso, la plataforma de flix olé tiene la oportunidad para dar servicios en línea, entretenimiento y para no salirse de la ley sobre los derechos de materiales protegidos, se utilizará un servidor que usa links externos para redireccionar a sus usuarios al contenido que buscan.

Descripción del proyecto.

Es un proyecto basado en la app de Netflix donde hacemos el consumo de la misma aplicación con react ya que nos basamos en una versión igual pero que contenga una conexión con mongo donde nos muestre nuestro usuario y las características principales de cada película que posee.

1.3. Justificación

En la actualidad el streaming se utiliza en varias aplicaciones para transmisiones por lo que resulta complicado ya que se debe descargar e instalar y pagar suscripción a veces no hay el espacio o la conexión no es suficiente para el acceso a esta segunda aplicación. La aplicación se desarrolló en el con las siguientes herramientas MongoDB, Express, AngularJS y Nodejs, un conjunto software utilizado para el desarrollo web dinámicas basadas en JavaScript. Y para el apartado de front-end utilizaremos HTML y CSS. Basado en el estudio de estándares de audio/vídeo. El impacto tecnológico que las personas van a acceder es la nueva forma de mirar diferentes entretenimientos audio visual, en sus dispositivos sin necesidad de adquirir otros equipos más avanzados sino reutilizar los equipos existentes gracias a las nuevas arquitecturas que son adaptables por medio del desarrollo en tecnologías del momento. El impacto en la sociedad será estar más comunicado con los medios ahora por medio de streaming, adquirir conocimientos, mensajes, capacitaciones será un enlace más cercano con los usuarios. Obteniendo la fidelización de usuarios que se encuentran en entornos digitales. En cuanto al ambiente se ahorró en la adquisición de nuevos equipos si no también utilizaremos los equipos existentes, sin la necesidad que se tengan que adquirir equipos de alta gama ya que la aplicación es adaptable a cualquier PC.

1.4. Objetivos

a. General

Desarrollar un sistema de streaming que permite ver contenidos multimedia (películas, series, video musicales, etc.) de nuestra preferencia en cualquier momento que lo desee. puedes acceder en una PC o aparatos móviles a través de internet.

b. Específicos

- Registrar el usuario a la plataforma.
- Ingresar correctamente a la plataforma.

- Buscar el contenido multimedia disponible sin ningún problema.
- Descargar el video, velocidad de reproducción e imagen completa.

1.5. Alcances

En la actualidad la tecnología de transmisión multimedia se ha convertido en algo cotidiano. Esta tecnología está muy relacionada con los dispositivos móviles, que transmiten y reproducen audio y vídeo en tiempo real. Cada día aparecen nuevos dispositivos la cual hace posible administrar una variada gama de aplicaciones con mejores capacidades. El sistema tiene como objetivo optimizar la transmisión de audio y vídeo, el desarrollo del sistema estandarizada y facilitará la reproducción de audio y vídeo en vivo sin necesidad de otra aplicación permitirá escalabilidad y adaptabilidad, dando mayor facilidad a los usuarios que se beneficiaran del manejo óptimo de la aplicación.

1.6. Limitaciones

- Inicialmente no se cuenta con un hosting para ir visualizando cómo está quedando el sistema en modo producción. Sino que se utilizará un servidor local.
- La plataforma de streaming aún está en desarrollo por parte de la integración de más móviles a gran escala como las hay en el mercado.
- Mayor parte son tráileres de las películas próximamente disponibles en la plataforma.
- No se cuenta con clasificación de edad.

1.7. Metodología de la investigación

Para este proyecto se utilizó el método: grupos de enfoque. en equipos de trabajo llevamos a mesa un proyecto viable que integre las tecnologías para aprender y practicar en la cátedra de Nuevas tendencias de la programación.

a. Técnicas científicas para encontrar hechos

Debido al avance tecnológico en los últimos años, el internet es muy usado como un medio de comunicación y entretenimiento; por lo tanto, como enlace de relaciones sociales, por ejemplo, en el caso de los migrantes que necesitan saber las noticias de su país. (Vinuesa, 2011, pág. 1) La característica más novedosa que ha aportado Internet a la radio y que incluso puede resultar una utopía es la de convertir un medio

ciego desde sus orígenes en visual. La tecnología ha hecho posible la transmisión de sonido o audio por Internet. Hoy en día todos los hogares disponen al menos de un ordenador. Esta generación en el uso del PC con su correspondiente conexión a internet, ha influido notablemente en el precio de los equipos informáticos, los cuales han abaratado su precio 5 considerablemente, siendo necesarios para poder acceder a los servicios que nos ofrece. Pero no solo los equipos informáticos se han visto influidos por una generalización en su uso, también las líneas RDSI han abaratado costes y, sin embargo, han aumentado la calidad del sonido en cuanto a recepción y la velocidad en el proceso de transmisión. (Espino Narváez & Pena, 2012, pág. 91) La ventaja de emplear radio online como herramienta de comunicación en la actualidad sirve para poder escuchar mientras se realizan otras tareas. También los otros servicios como, por ejemplo, la televisión ha tenido rechazo entre los usuarios por la dedicación y atención que necesita, es decir tenía que dejar de hacer lo que estaba haciendo para prestar atención a este servicio.

1.8. Marco teórico

¿Qué es la tecnología Streaming?

Es una tecnología (también conocida como media-flujo) que permite la distribución de archivos multimedia (principalmente audio y video) a través de la red de computadores en tiempo real. Estos son cargados en un buffer de datos donde se puedan reproducir al mismo tiempo que se descargan y luego se descartan, sin quedar almacenados en el disco duro del cliente. Para hacer uso de esta tecnología, se necesita por lo menos tener un ancho de banda mínimo igual a la tasa de transferencia del servidor, de lo contrario se producirían cortes y la calidad de reproducción se verá afectada, que es justamente lo que esta tecnología no quiere, ya que busca un flujo continuo de archivos. Esta tecnología nació en el año 1995 junto con la aparición de RealAudio 1.0, un formato que permite justamente reproducir el audio en tiempo real; pero no se hizo popular sino hasta el año 2000 cuando el ancho de banda se hizo lo suficientemente barato y rápido para gran parte de la población. Antes de la aparición de esta tecnología, la reproducción de contenido multimedia a través de Internet implicaba descargar completamente el archivo al disco duro local. Como estos archivos multimedia son de gran tamaño, su descarga y acceso como paquetes completos resultaba un mecanismo lento.

Componentes de la Tecnología Streaming

Para saber cómo funciona Streaming, primero debemos comprender un poco los componentes básicos de un sistema Streaming :

Códecs: son archivos residentes en el Host cliente que interpretan el contenido multimedia y hacen posible su reproducción.

Protocolos: HTTP, RTSP, RTP, RTCP, TCP, UDP. Siendo los protocolos UDP, RTSP (Real Time Transfer Protocol) los más importantes, ya que 3 hacen que la entrega de paquetes de datos desde el servidor al cliente sea con una velocidad mucho mayor que con los protocolos HTTP y TCP (aplicaciones tolerantes a pérdidas).

Precarga: el cliente precarga o almacena el archivo multimedia en un buffer de datos antes de reproducirlo, con el fin de evitar interrupciones y mala calidad de reproducción debido a la congestión de la red (Lag).

Red de datos: Si un determinado contenido comienza a atraer una cantidad de usuarios mayor que su capacidad de Ancho de Banda estos usuarios sufrirán interrupciones (también conocido como Lag). Finalmente, se llega a un punto en que la calidad del Stream es pésima.

Segmentación: la información multimedia es segmentada, para luego ser enviada como paquetes a la red los cuales el cliente recibe y reproduce inmediatamente.

Servicios que ofrece Streaming

La tecnología Streaming permite ofrecer múltiples servicios:

En directo (Live): Similar a un canal de televisión o una estación de radio.

- Está orientado a la multidifusión ip (unicast o multicast):
 - Unicast: se envía un flujo de información a cada usuario.
 - Multicast: se envía un flujo único de información.
 - El servidor comienza a transmitir en un instante dado
 - Los usuarios ven la información que se está emitiendo
 - En este tipo de servicio no existe interacción.
 - Únicamente se le permite realizar pausas (cuando el usuario recupere la reproducción podrá ver la información que se está transmitiendo en ese instante)
- Bajo demanda (On-Demand):
 - Similar a un reproductor de video:
 - Los usuarios solicitan el envío del archivo en cualquier instante, no necesitan esperar a que el servidor lo transmita.
 - Es Punto a Punto (Unicast: envío de flujo a cada usuario)
 - El usuario puede interactuar (pausar, avanzar, retroceder, etc.)
- Casi bajo demanda: simula el funcionamiento de un servicio bajo demanda con flujos de vídeo en directo.

Arquitectura de la Tecnología Streaming

Entre las arquitecturas que emplea Streaming podemos encontrar:

- Arquitectura Típica: usa la arquitectura típica Cliente-Servidor. Acá los protocolos más usados son:
 - Sin control sobre la transmisión: HTTP.
 - Con control sobre la transmisión:
 - Capa de Aplicación: RTSP(Real Time Streaming Protocol) es el protocolo responsable de la entrega de datos y audio en tiempo real, no orientado a conexión, donde el control y reenvío de datos corre a cargo de TCP; MMS(Microsoft Media Server), RTMP(Real Time Messaging Protocol) y RTMFP(Real Time Media Flow Protocol).
 - Capa de Transporte: RTP(Real Time Transport Protocol), UDP y TCP.
- Arquitectura Sin Servidor(Server-Less): Acá no existe un servidor de Audio-Video, el archivo se le proporciona al cliente mediante un servidor Web (pseudo-streaming o Fast-Start). Usa los Protocolos TCP y HTTP.

a. Referencia histórica.

La primera patente que utilizó la palabra streaming fue en los años 20, cuando la empresa Muzak desarrolló una plataforma de música continua para negocios, este dato es interesante ya que ni siquiera existían computadoras en ese entonces.

Parte importante en la historia del streaming ha sido la evolución del internet, así como de las conexiones de banda ancha, recordemos que los primeros accesos a la red eran hechos a través de líneas de teléfono convencionales, lo máximo que podíamos conseguir de velocidad eran 27kbps (cuando bien nos iba) era prácticamente imposible hacer transmisiones en tiempo real, pero gracias al crecimiento en infraestructura como DSL o fibra óptica, y la comercialización del Internet con costos cada vez más accesibles para las personas, la velocidad del internet dejó de ser un problema y es ahí cuando el streaming vio su oportunidad real de ser un estándar como lo es hoy en día.

Inicios del Streaming

Sin duda las estaciones de radio por internet fueron el primer boom del streaming, ya que no se necesitaba tanta velocidad para sintonizar el audio de manera fluida, era lógico pensar que su metamorfosis iba encaminada al video, cuando las primeras

transmisiones en tiempo real salieron a flote, sólo eran vistas por 2 ó 3 personas, ya que no había la capacidad de hacerlo escalable, y eran eventos de máximo 4 horas. En 1997 la banda Severe Tire Damage hizo historia al transmitir su concierto en vivo a todo el mundo, esto fue en el Xerox PARC, más tarde en 1995 la empresa Real Networks transmitió por primera vez en streaming un juego de béisbol de los playoffs, ese mismo año en Seattle se realizó la transmisión de un concierto de la sinfónica en el Paramount Theater.

Todos estos hechos representan esfuerzos individuales del gran potencial que tiene el streaming al día de hoy, eran una pequeña probadita de lo varios años después sin duda es el servicio más popular a través de internet a nivel mundial.

El streaming al alcance de todos.

La comercialización del streaming tuvo que esperar varios años después, como se mencionó anteriormente, va de la mano con los avances tecnológicos y el crecimiento exponencial del uso del internet y de las capacidades en hardware de las computadoras, por ejemplo, en 1995 una computadora Dell tenía un procesador de 66MHz, 1Gb de espacio en disco duro y 8Mb de memoria RAM, con un costo de \$4,000 dólares, ¡una fortuna!

Con todas estas limitantes la habilidad para procesar y recibir una señal en streaming era prácticamente imposible, posteriormente computadoras más veloces y un internet más rápido resolvieron la ecuación.

Sin lugar a dudas el año 2005 fue muy importante para la proliferación de los videos gracias al lanzamiento del portal de YouTube (cuando no le pertenecía a Google) además de la creación de los reproductores de videos basados en flash, quien iba a decir que apenas 11 años después la tecnología flash estaba destinada a morir.

El streaming hoy y su futuro.

Es indudable que el streaming es parte fundamental de nuestras vidas, ya que gracias a él existen plataformas como Netflix, Spotify, YouTube, CWS entre otras, cualquier puede poner en su sitio web un reproductor de video en vivo en sólo unos segundos. Se está desarrollando un códec llamado h.265 o HEVC (high efficiency video coding) con el cual se podrá transmitir la misma calidad de video actual, pero con un 50% menos de ancho de banda, lo cual significa que se podrán enviar videos en 1080p o 4k y podrán ser vistos con conexiones caseras de internet sin temor a cortes.

Lo único certero con el streaming es que no se va a ir a ninguna parte, ya que incluso cambió la manera en cómo consumimos los contenidos de video como la TV tradicional, nunca pensaron que hubiera algo que lo reemplazaría, pues ya vemos que estaban muy equivocados.

b. Marco conceptual

Internet: Es una red mundial de ordenadores formada por la unión de redes más pequeñas. A esta red se puede acceder desde cualquier ordenador y permite el intercambio de información entre los usuarios. También se le denomina Word Wide Web (WWW).

Navegación: Navegar por la Red es el paso de una página a otra mediante los enlaces que contienen. Desde la primera página web, podemos acceder a otras páginas pulsando distintas partes de la misma.

Sitio Web/Página Web: es el nombre de un documento o información electrónica adaptada para la Word Wide Web y que puede ser accedida mediante un navegador para mostrarse en un monitor de computadora o cualquier dispositivo con acceso a la red.

Enlace: referencia de un documento de hipertexto o recurso en línea a otro documento o recurso.

Online: En telecomunicaciones, on-line (en línea) se utiliza para designar a una computadora que está conectada al sistema, está operativa, está encendida o accede a internet. También se define este término como estado de un usuario cuando está conectado a Internet.

HTTP: El Protocolo de transferencia de hipertexto es el protocolo de comunicación que permite las transferencias de información a través de archivos en la Word Wide Web.

RTSP: El protocolo de transmisión en tiempo real establece y controla uno o muchos flujos sincronizados de datos, ya sean de audio o de video. El RTSP actúa como un mando a distancia mediante la red para servidores multimedia.

RTP: El protocolo de transporte en tiempo real o RTP, es un protocolo de nivel de aplicación utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en una videoconferencia.

RTCP: El protocolo de control en tiempo real es un protocolo de comunicación que proporciona información de control que está asociado con un flujo de datos para una aplicación multimedia. Trabaja junto con RTP en el transporte y empaquetado de datos multimedia, pero no transporta ningún dato por sí mismo.

TCP: La familia de protocolos de internet es un conjunto de protocolos de red en los que se basa internet y que permiten la transmisión de datos entre computadoras.

UDP: El protocolo de datagramas de usuario es un protocolo del nivel de transporte basado en la transmisión sin conexión de datagramas y representa una alternativa al protocolo TCP.

CSS, en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

CAPÍTULO IV. MODELADO Y DESARROLLO DEL SISTEMA

4.1 Propuesta de solución

Como propuesta de Solución se realizó un prototipado de streaming de api, de películas como Netflix con el fin de que los usuarios puedan ver y descargar películas, sin necesidad de realizar cargos de dinero, de cuotas grandes, como otras plataformas lo hacen, si no que el usuario tenga contenido gratis con el uso del sistema.

4.2 Modelado del sistema según metodología utilizada

Tecnología utilizada Las tecnologías a utilizar son, el cuadro de herramientas de trabajo, acrónimo inglés para MongoDB, Express, AngularJS y Nodejs, un conjunto software utilizado para el desarrollo web dinámicas basadas en JavaScript. Y para el apartado de front-end utilizaremos HTML y CSS.



HTML El lenguaje más importante y conocido para la creación de páginas web, dándoles una estructura y un contenido. Basado en etiquetas, breves instrucciones de apertura y cierre. HTML5 es la última versión de HTML, la cual incluye una nueva estructura que facilita la forma de generar contenido, etiquetado como `<div>`, ``, `<h1>`, y `<h2>` facilitan mucho la estructura de la información en la aplicación.

También trae mejoras sobre la etiqueta input, pudiendo indicar si es un url, un email, un entero, etc. También destacar que al ser la última versión incluirá la corrección de errores en etiquetas y comportamientos que podrían afectar al correcto funcionamiento de la aplicación



Al igual que HTML no es considerado un lenguaje de programación sino un lenguaje de estilo, separa el contenido de la forma y permite un mayor control sobre la apariencia. CSS3 es la última versión de CSS, incluye nuevos efectos, con una mayor facilidad de aplicación y conseguir que la parte visual de nuestros documentos sean más llamativos como son los bordes o el sombreado. Y la corrección de errores de antiguas versiones.



JavaScript es un lenguaje de programación orientado a objetos utilizado principalmente para la creación de páginas web dinámicas del lado del cliente y en ocasiones del lado del servidor. El aumento de las APIs lo ha convertido en uno de los lenguajes de programación más utilizados en el mundo y el adecuado para desarrollar la aplicación.

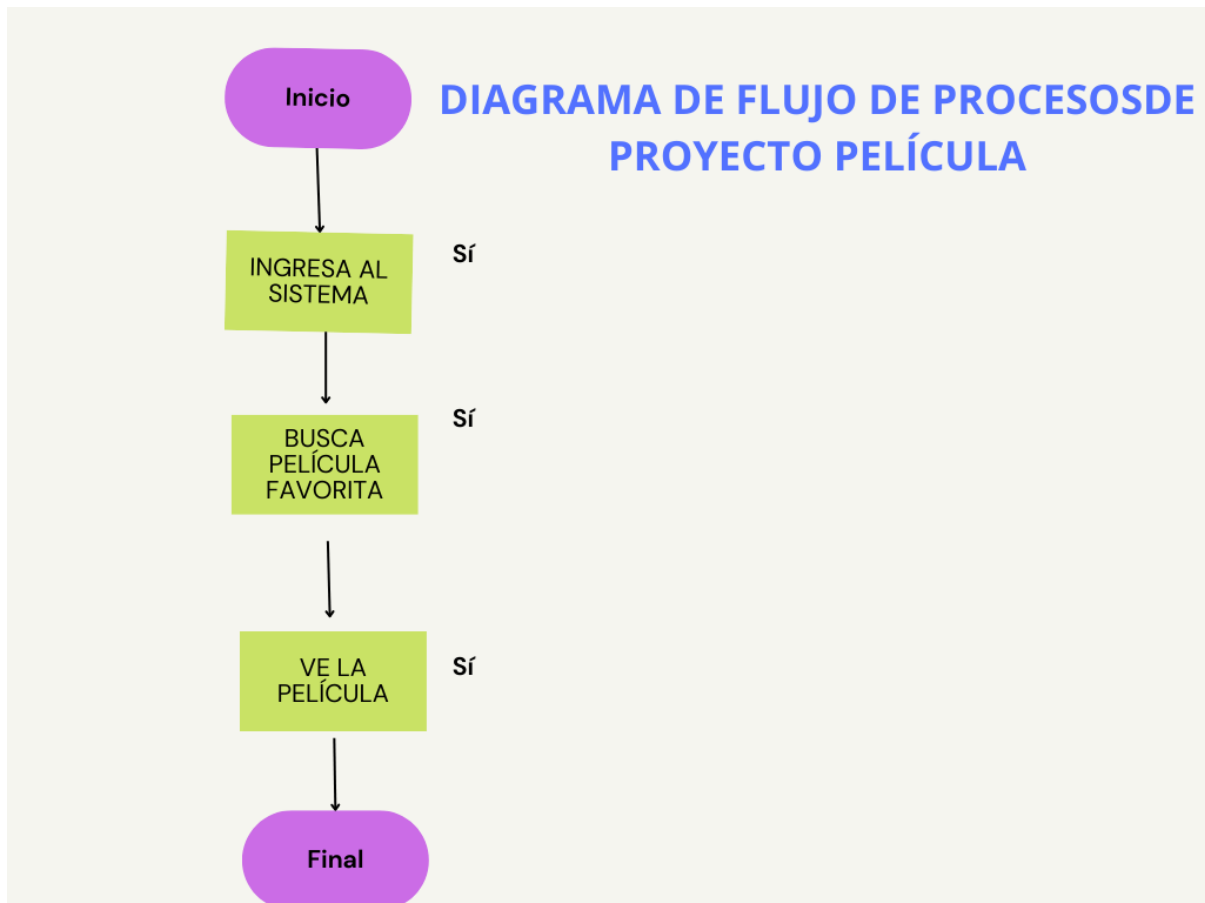


MongoDB Base de datos NoSQL, almacena los datos en una estructura llave-valor. Básicamente los valores de estas llaves, llamadas colecciones, son estructuras tipo JSON también llamados documentos. Creado para ofrecer escalabilidad, rendimiento y gran disponibilidad.

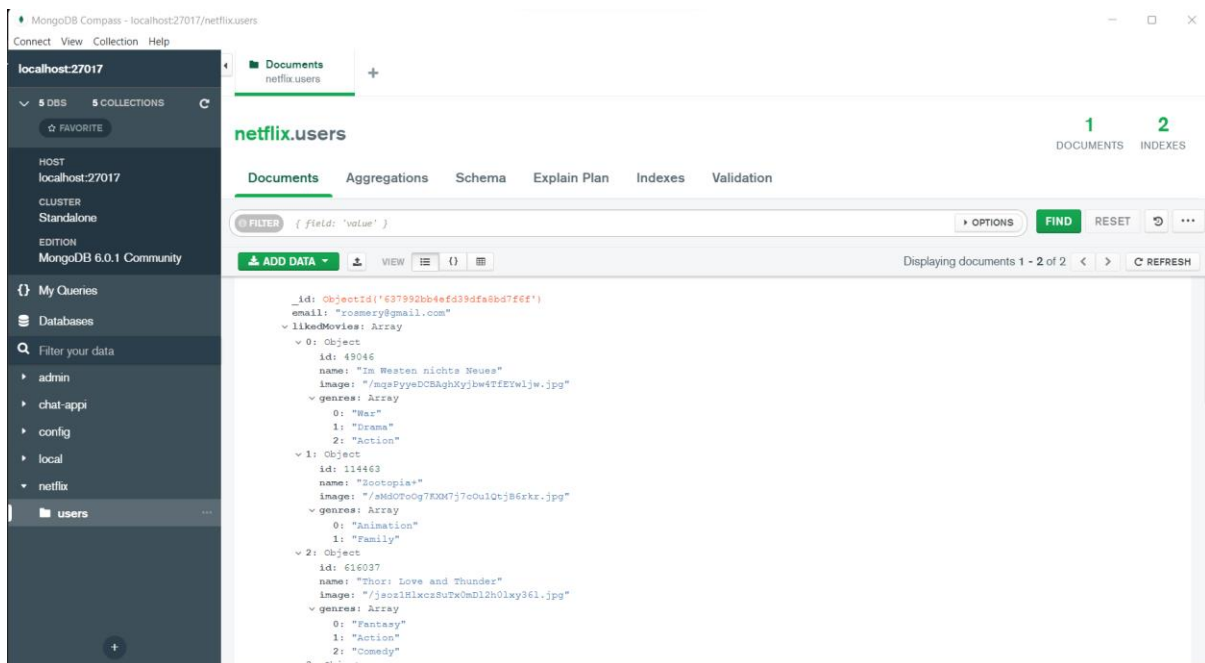


Entorno en tiempo de ejecución multiplataforma encargado del funcionamiento de la capa del servidor, basado en el lenguaje de programación JavaScript y orientado a eventos asíncronos.

4.2.3 diagrama de flujo de proceso de datos



4.3. Base de datos Diccionario de datos



MongoDB Compass - localhost:27017/netflix/users

Connect View Collection Help

localhost:27017

5 DBS 5 COLLECTIONS

FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 6.0.1 Community

My Queries

Databases

Filter your data

- admin
- chat-app
- config
- local
- netflix
 - users

Documents netflix.users

netflix.users

1 DOCUMENTS 2 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 2 of 2 REFRESH

```

{
  "_id": ObjectId("637992bb4e6d39dfe8bd766f"),
  "email": "rosamery@gmail.com",
  "likedMovies": Array,
  "__v": 0
}

{
  "_id": ObjectId("637d93d3167f7beb970dba96"),
  "email": "elay@gmail.com",
  "likedMovies": Array
    > 0: Object
      id: 436270
      name: "Black Adam"
      image: "/bGKAq8k2Fgc46uCVWgoFz3L5Dtr.jpg"
      genres: Array
        > 0: "Action"
        > 1: "Fantasy"
        > 2: "Science Fiction"
    > 1: Object
      id: 505642
      name: "Black Panther: Wakanda Forever"
      image: "/ytrvN5WFe27JnRzbtY0CKuo4Iaw.jpg"
      genres: Array
        > 0: "Action"
        > 1: "Adventure"
        > 2: "Science Fiction"
    > 2: Object
  }

```

MongoDB Compass - localhost:27017/netflix/users

Connect View Collection Help

localhost:27017

5 DBS 5 COLLECTIONS

FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 6.0.1 Community

My Queries

Databases

Filter your data

- admin
- chat-app
- config
- local
- netflix
 - users

Documents netflix.users

netflix.users

1 DOCUMENTS 2 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 2 of 2 REFRESH

```

{
  "_id": ObjectId("637992bb4e6d39dfe8bd766f"),
  "email": "rosamery@gmail.com",
  "likedMovies": Array,
  "__v": 0
}

{
  "_id": ObjectId("637d93d3167f7beb970dba96"),
  "email": "elay@gmail.com",
  "likedMovies": Array,
  "__v": 0
}

```

> _MONGOSH

4.5. Interfaz del usuario

CASO DE USO APP DE PELICULAS

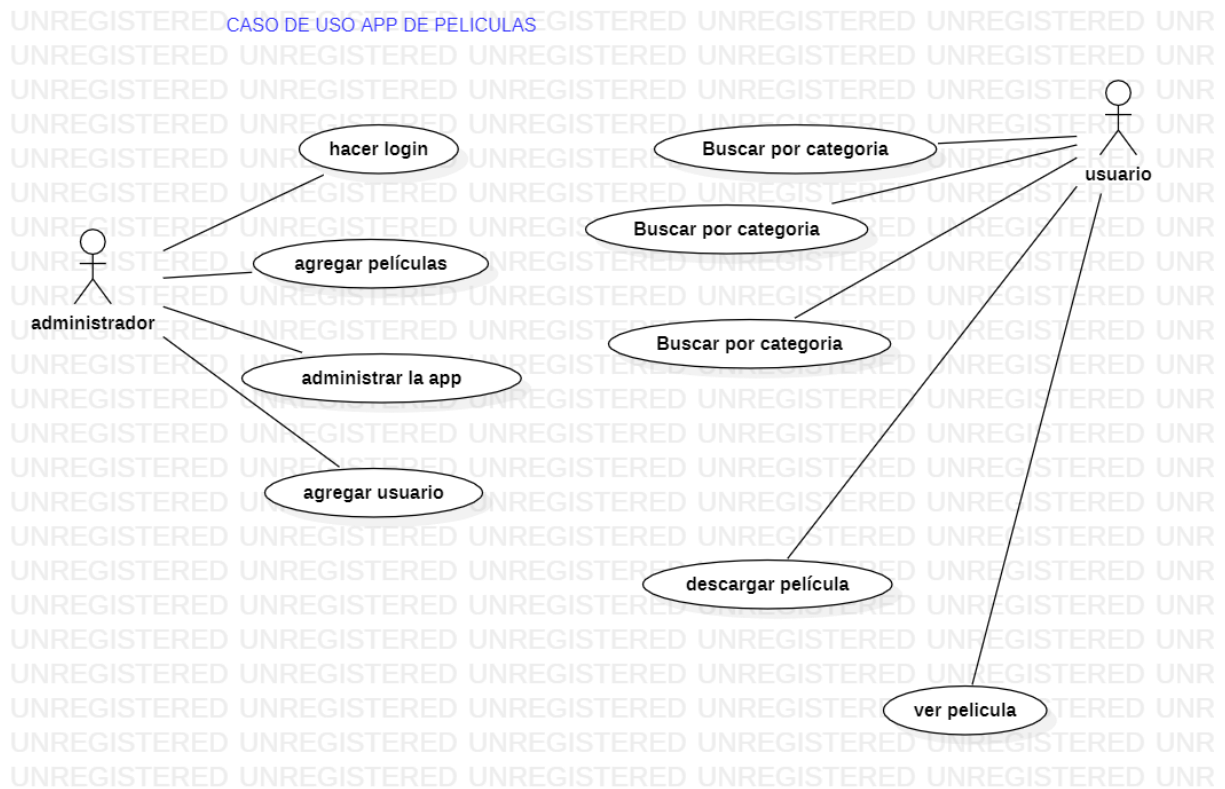
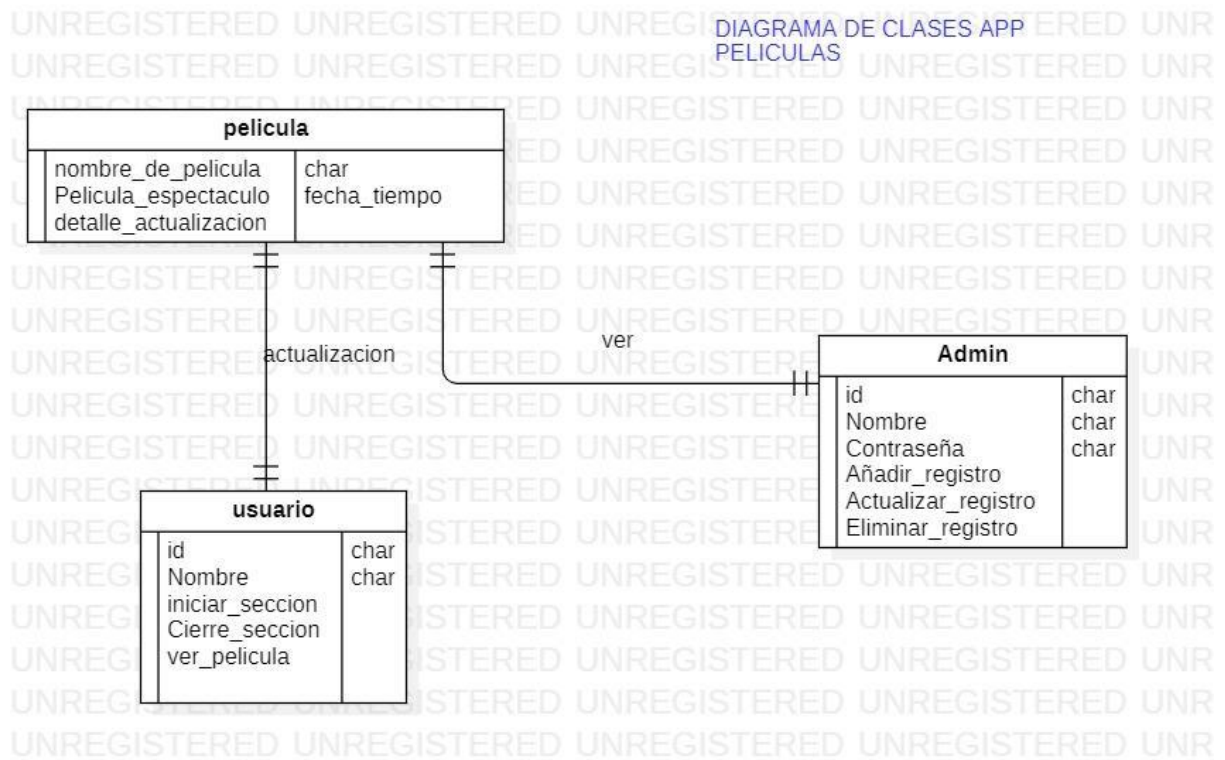


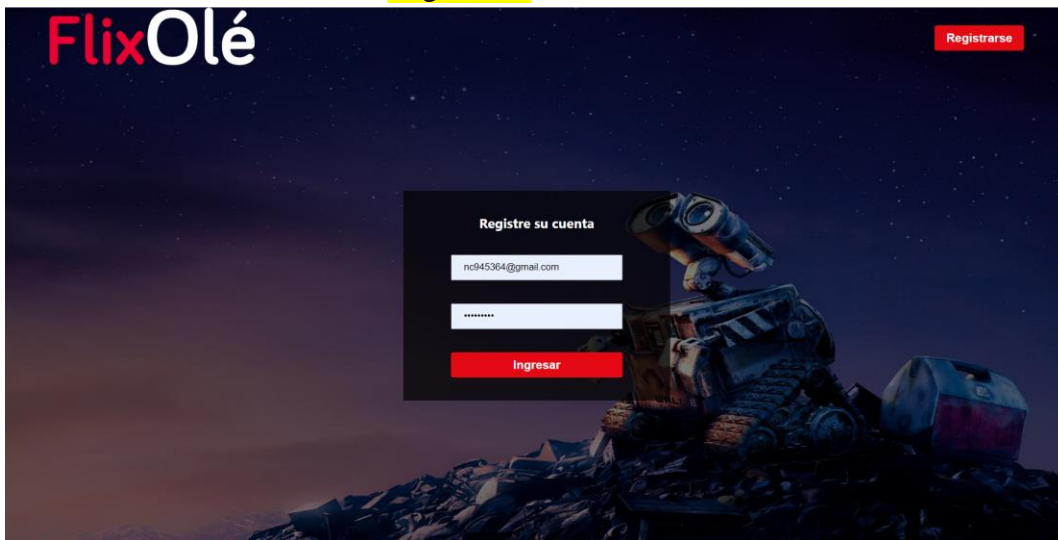
DIAGRAMA DE CLASES APP PELICULAS



CAPÍTULO VI (FASE IV) PUESTA EN PRODUCCIÓN

6.1.2 Manual de usuario

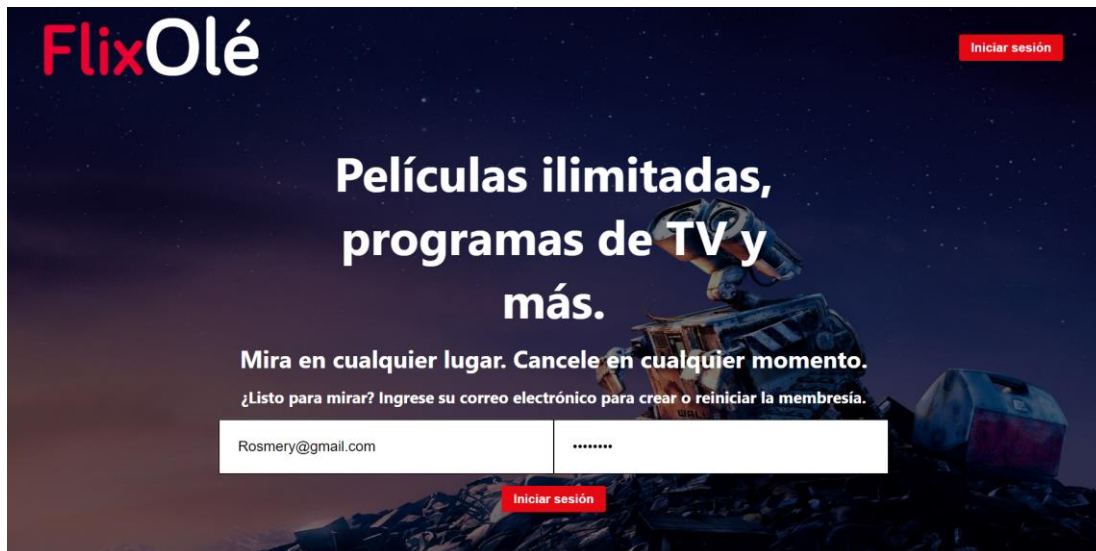
Al iniciar en pantalla principal, se pide que ingrese su cuenta de correo y contraseña, para crear una nos vamos al botón **Registrarse**



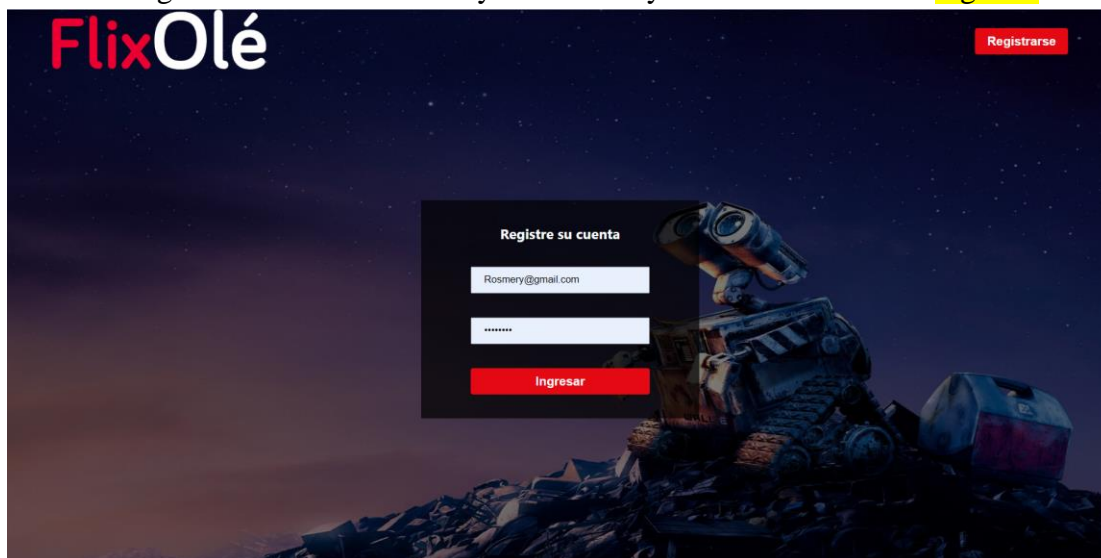
Estando en registros nos pide que coloquemos nuestro correo electrónico que se desee y damos clic en el **botón empezar.**



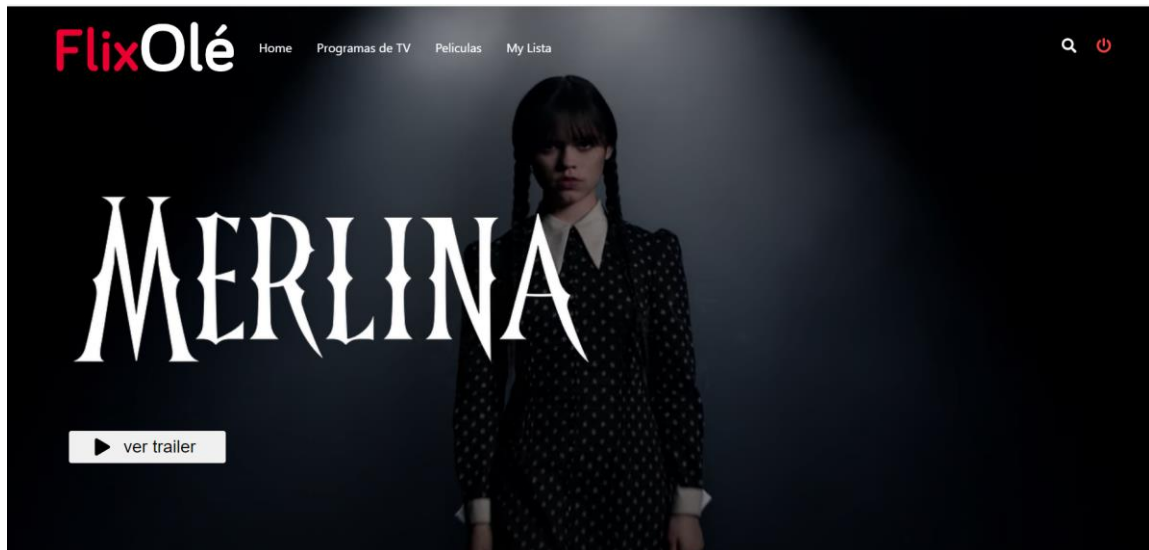
Cuando damos clic en empezar nos muestra una ventana donde nos pide colocar nuevamente el correo y contraseña que contenga 8 caracteres. Damos clic **botón iniciar sesión.**



Entonces ingresamos nuestro correo y contraseña y damos clic en botón **ingresar**

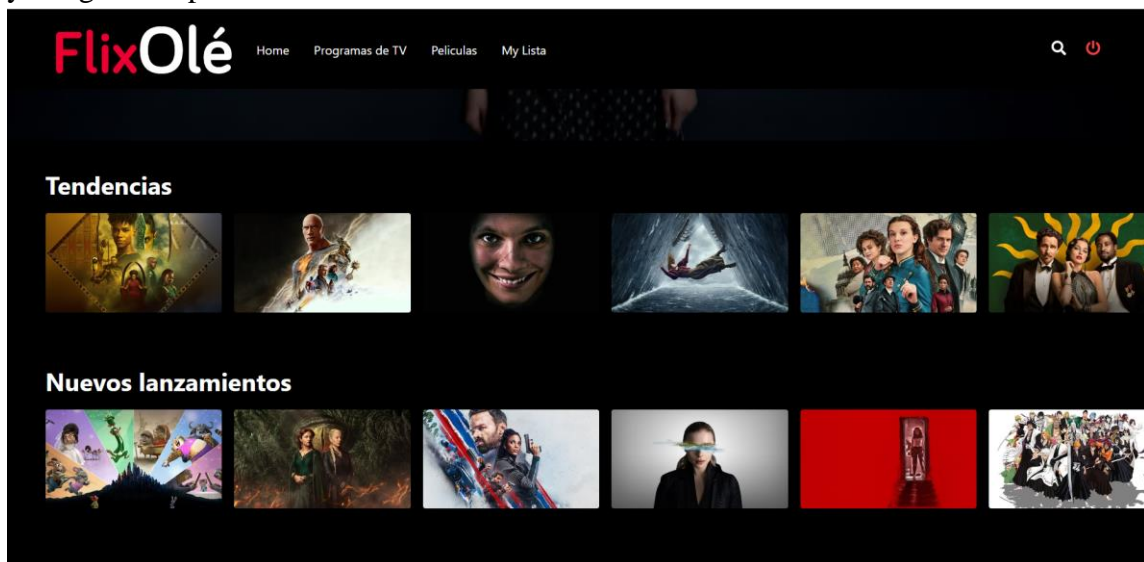


Podemos observar nuestra ventana principal donde encontramos nuestro inicio, programas de tv, y mi lista de películas, donde nos permite buscar las películas que se tienen alojadas.

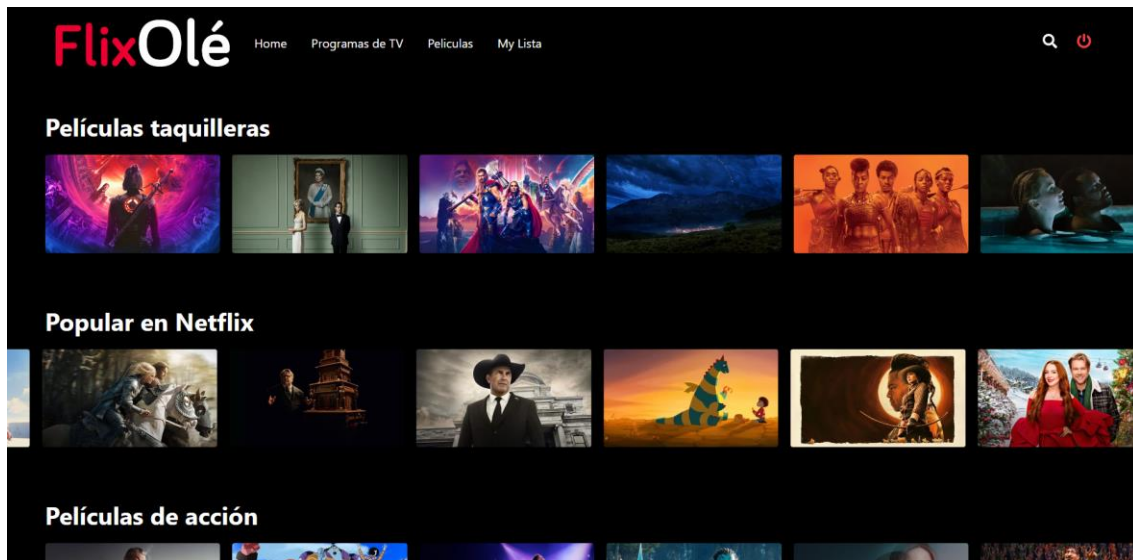


En nuestra página principal de la película se encuentra un botón **ver tráiler**

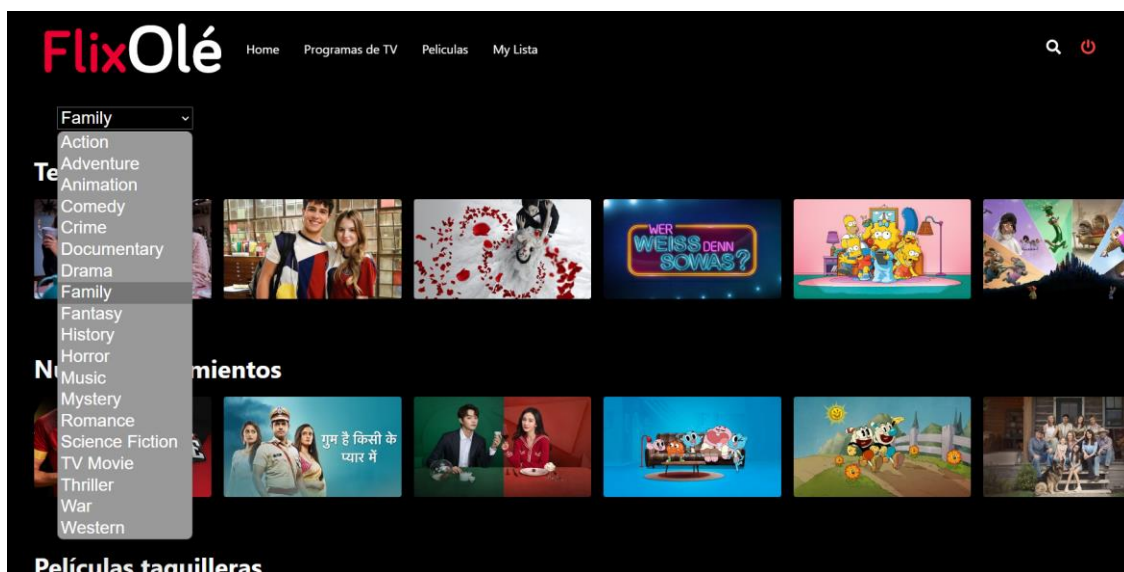
Estando dentro del video se tiene la opción de descargar el video, velocidad de reproducción y imagen completa.



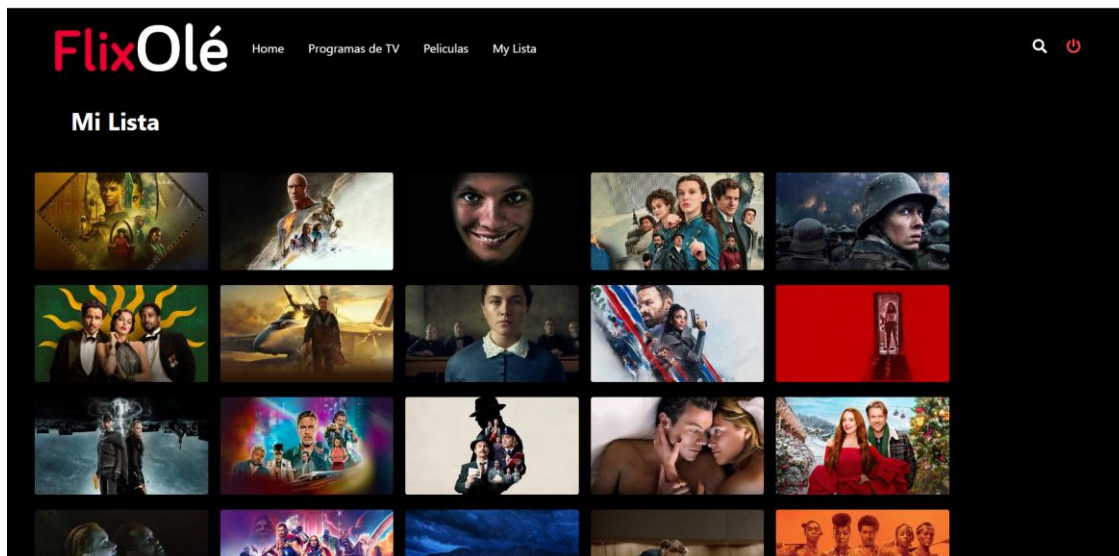
Encontramos lo más en tendencia, nuevos lanzamientos,



Encontramos películas taquilleras, lo más popular en Netflix, películas de acción

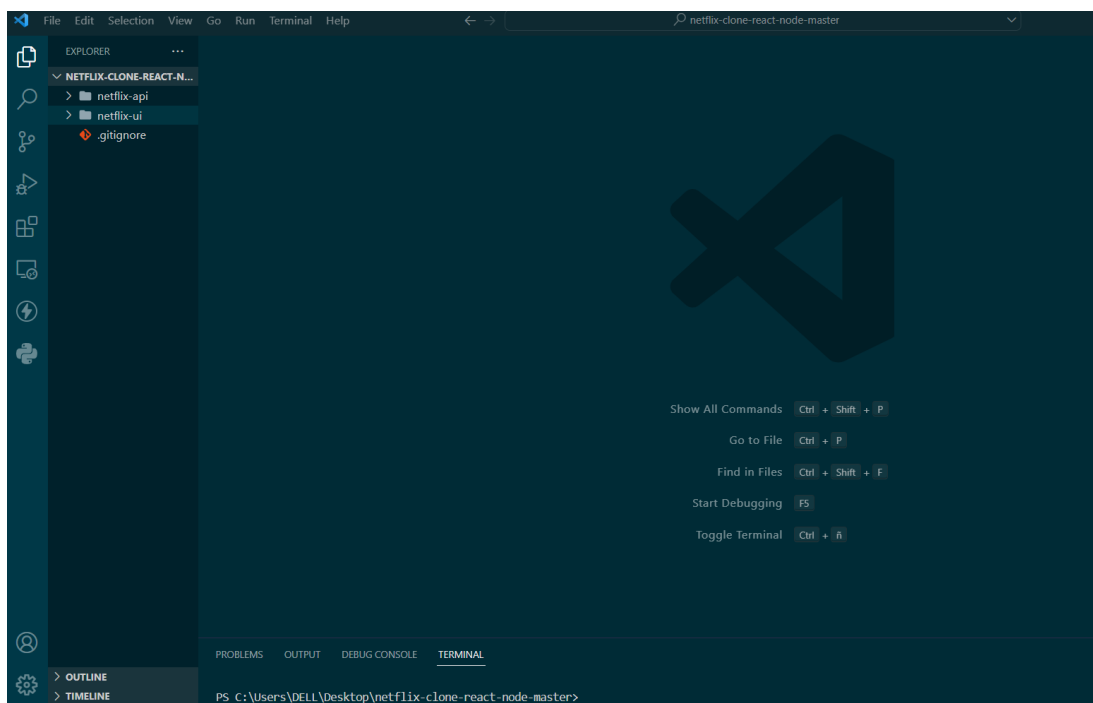


Encontramos géneros de acción, aventura, crimen y cuando se escoge un género que no existe nos muestra un mensaje diciendo que no hay programas de TV disponibles para el género seleccionado. Seleccione un género diferente.

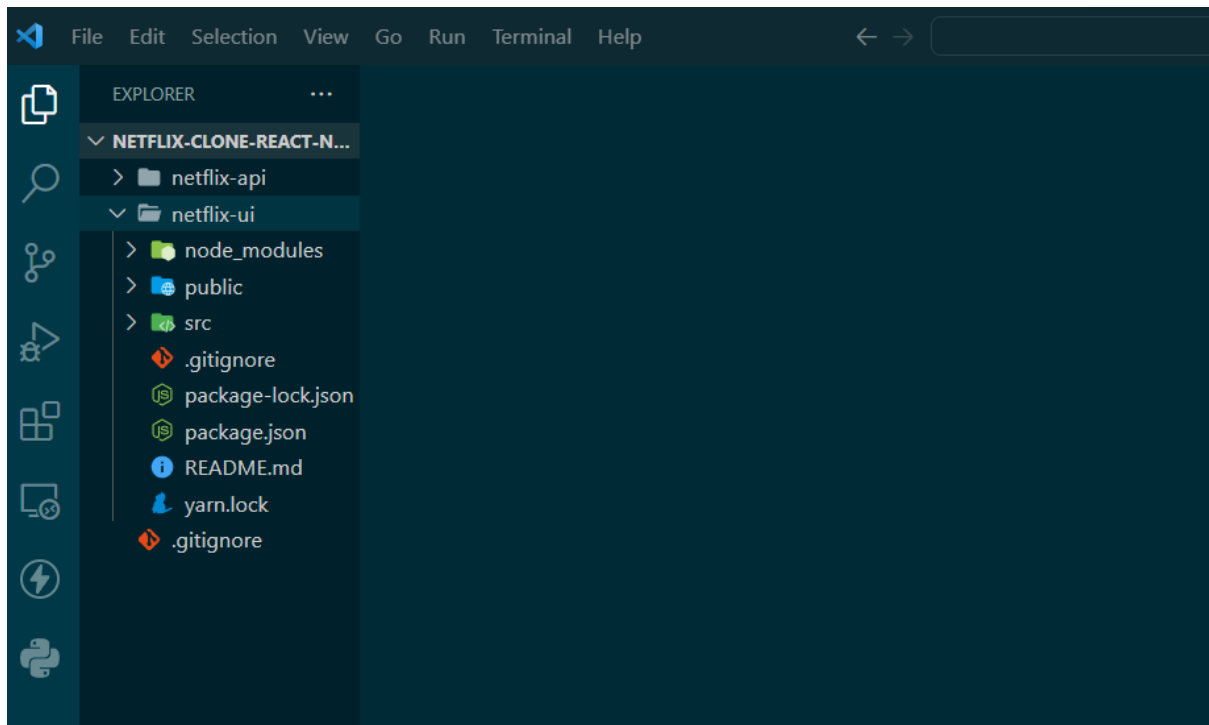


Cuando damos clic en mi lista encontramos las películas que hemos seleccionado a guardar a nuestra lista de ver más tarde.

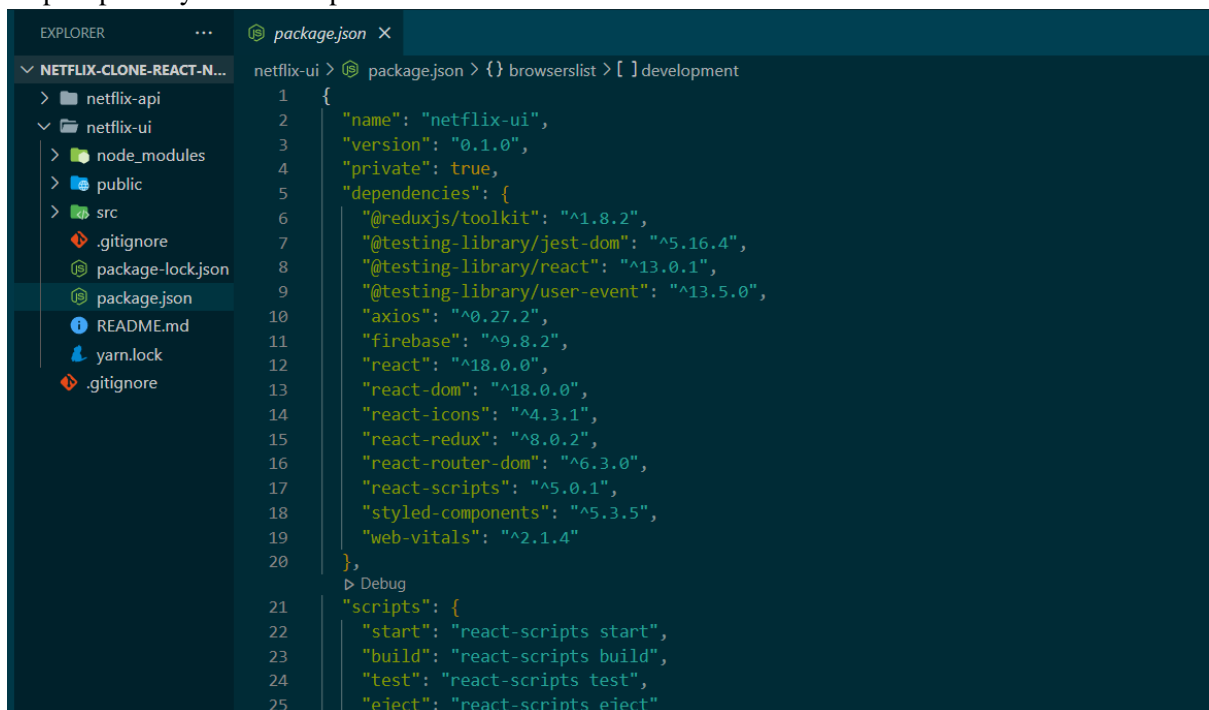
6.1.3 Manual de Administrador



Para separar de nuestra Frontend de nuestros backend nos definimos como la carpeta netflix-ui como Frontend y Netflix-api como nuestro backend sería nuestra base de datos mongo.

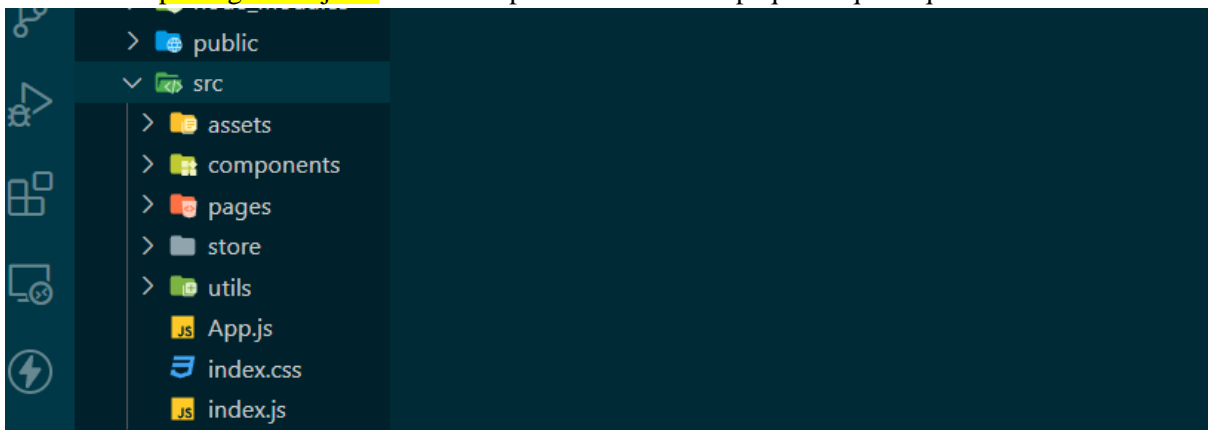


En la carpeta netflix-ui, encontramos la paquetería de package Json, nuestra carpeta src, nuestra carpeta public y nuestra carpeta de node modules

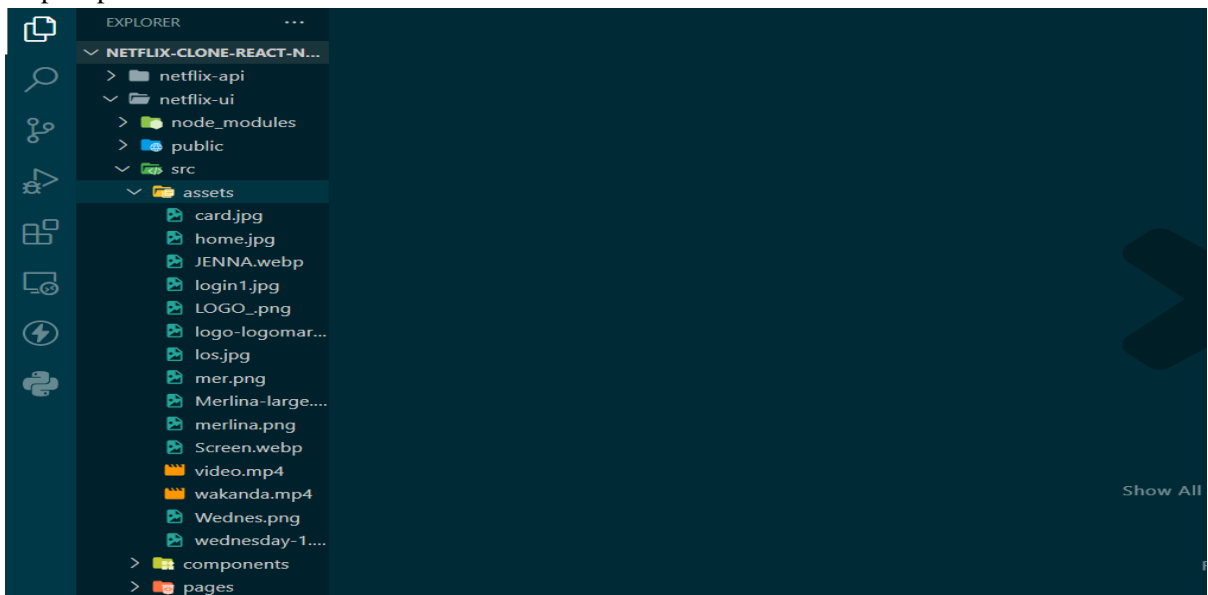


```
1 {
2   "name": "netflix-ui",
3   "version": "0.1.0",
4   "lockfileVersion": 2,
5   "requires": true,
6   "packages": {
7     "": {
8       "name": "netflix-ui",
9       "version": "0.1.0",
10      "dependencies": {
11        "@reduxjs/toolkit": "^1.8.2",
12        "@testing-library/jest-dom": "^5.16.4",
13        "@testing-library/react": "^13.0.1",
14        "@testing-library/user-event": "^13.5.0",
15        "axios": "^0.27.2",
16        "firebase": "^9.8.2",
17        "react": "^18.0.0",
18        "react-dom": "^18.0.0",
19        "react-icons": "^4.3.1",
20        "react-redux": "^8.0.2",
21        "react-router-dom": "^6.3.0",
22        "react-scripts": "^5.0.1",
23        "styled-components": "^5.3.5",
24        "web-vitals": "^2.1.4"
25      }
26    }
27  }
28 }
```

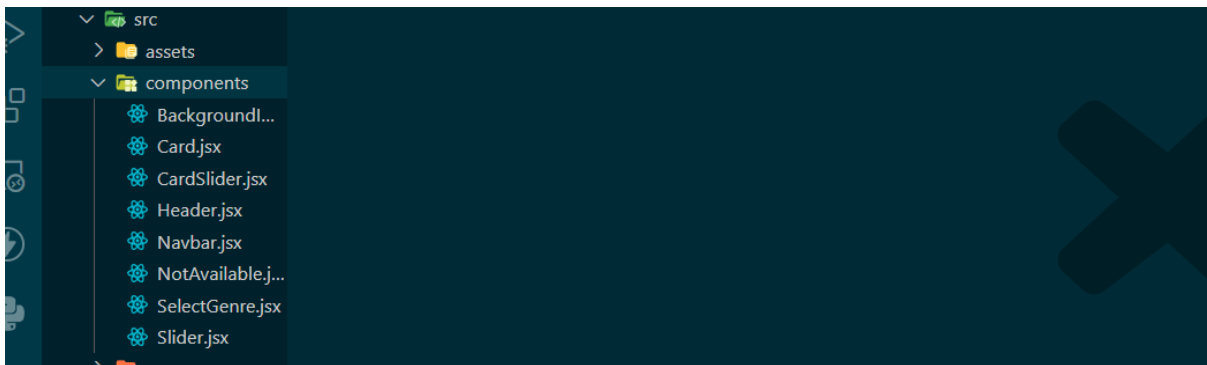
Observamos **package-lock.json** si está completa con todos los paquetes que requerimos



En la carpeta **src** encontramos, la carpeta, assets, components, pages, store, utils son parte de esta carpeta para el uso necesario.



Aquí en carpeta **assets** encontramos las imágenes y video que utilizamos de foto de portada



En la carpeta components, encontramos, archivos donde utilizamos react,

```

1  import React from "react";
2  import styled from "styled-components";
3  import background from "../assets/login1.jpg";
4
5  export default function BackgroundImage() {
6    return (
7      <Container>
8        | <img src={background} alt="background" />
9      </Container>
10   );
11 }
12
13 const Container = styled.div`
14   height: 100vh;
15   width: 100vw;
16   img {
17     height: 100vh;
18     width: 100vw;
19   }
20 `;

```

Nos dirigimos al archivo BackgroundImage.jsx es donde importamos la imagen de portada de nuestro login, además hacemos importación de los componentes. y creamos en si un estilo de la posición de la imagen.

```

1  import React, { useState } from "react";
2  import { useNavigate } from "react-router-dom";
3  import styled from "styled-components";
4  import { IoPlayCircleSharp } from "react-icons/io5";
5  import { AiOutlinePlus } from "react-icons/ai";
6  import { RiThumbUpFill, RiThumbDownFill } from "react-icons/ri";
7  import { BiChevronDown } from "react-icons/bi";
8  import { BsCheck } from "react-icons/bs";
9  import axios from "axios";
10 import { onAuthStateChanged } from "firebase/auth";
11 import { firebaseAuth } from "../utils/firebase-config";
12 import { useDispatch } from "react-redux";
13 import { removeMovieFromLiked } from "../store";
14 import video from "../assets/video.mp4";
15
16 export default React.memo(function Card({ index, movieData, isLiked = false }) {
17   const navigate = useNavigate();
18   const dispatch = useDispatch();
19   const [isHovered, setIsHovered] = useState(false);
20   const [email, setEmail] = useState(undefined);
21
22   onAuthStateChanged(firebaseAuth, (currentUser) => {
23     if (currentUser) {

```

Encontramos las importaciones de nuestros módulos donde están conectadas y además importamos nuestro video de la portada del tráiler y donde se crea la parte de agregar a la lista de películas o información de la película.

```

netflix-ui > src > components > CardSlider > default > CardSlider
1 import React, { useRef, useState } from "react";
2 import styled from "styled-components";
3 import { AiOutlineLeft, AiOutlineRight } from "react-icons/ai";
4 import Card from "../Card";
5 export default React.memo(function CardSlider({ data, title }) {
6   const listRef = useRef();
7   const [sliderPosition, setSliderPosition] = useState(0);
8   const [showControls, setShowControls] = useState(false);
9   const handleDirection = (direction) => {
10     let distance = listRef.current.getBoundingClientRect().x - 70;
11     if (direction === "left" && sliderPosition > 0) {
12       listRef.current.style.transform = `translateX(${230 + distance}px)`;
13       setSliderPosition(sliderPosition - 1);
14     }
15     if (direction === "right" && sliderPosition < 4) {
16       listRef.current.style.transform = `translateX(${ -230 + distance}px)`;
17       setSliderPosition(sliderPosition + 1);
18     }
19   };
20 }

```

En CardSlider hace la función de los datos de la dirección, donde se encuentra el usuario, y hace la posición el estilo de la película donde nos muestra su diseño de tarjeta.

```

src
├── assets
├── components
│   ├── Backgroundl...
│   ├── Card.jsx
│   ├── CardSlider.jsx
│   ├── Header.jsx
│   ├── NavBar.jsx
│   ├── NotAvailable.j...
│   ├── SelectGenre.jsx
│   └── Slider.jsx
├── pages
├── store
├── utils
├── App.js
├── index.css
└── ...

```

```

6 export default function Header(props) {
7   const navigate = useNavigate();
8   return (
9     <StyledHeader className="flex a-center j-between">
10       <div className="logo">
11         <img src={logo} alt="logo" />
12       </div>
13       <button onClick={() => navigate(props.login ? "/login" : "/signup")}>
14         {props.login ? "Iniciar sesión" : "Registrarse"}
15       </button>
16     </StyledHeader>
17   );
18 }
19 const StyledHeader = styled.header`
20   padding: 0 4rem;
21   .logo {
22     img {
23       height: 5rem;
24     }

```

En el Header es donde encontramos nuestro login de registro donde iniciamos sección y registramos a el usuario

```

EXPLORER
├── NETFLIX-CLONE-REACT-N...
│   ├── netflix-api
│   ├── netflix-ui
│   ├── node_modules
│   ├── public
│   └── src
│       ├── assets
│       ├── components
│       │   ├── Backgroundl...
│       │   ├── Card.jsx
│       │   ├── CardSlider.jsx
│       │   ├── Header.jsx
│       │   ├── NavBar.jsx
│       │   ├── NotAvailable.j...
│       │   ├── SelectGenre.jsx
│       │   └── Slider.jsx
│       ├── pages
│       ├── store
│       ├── utils
│       ├── App.js
│       └── index.css

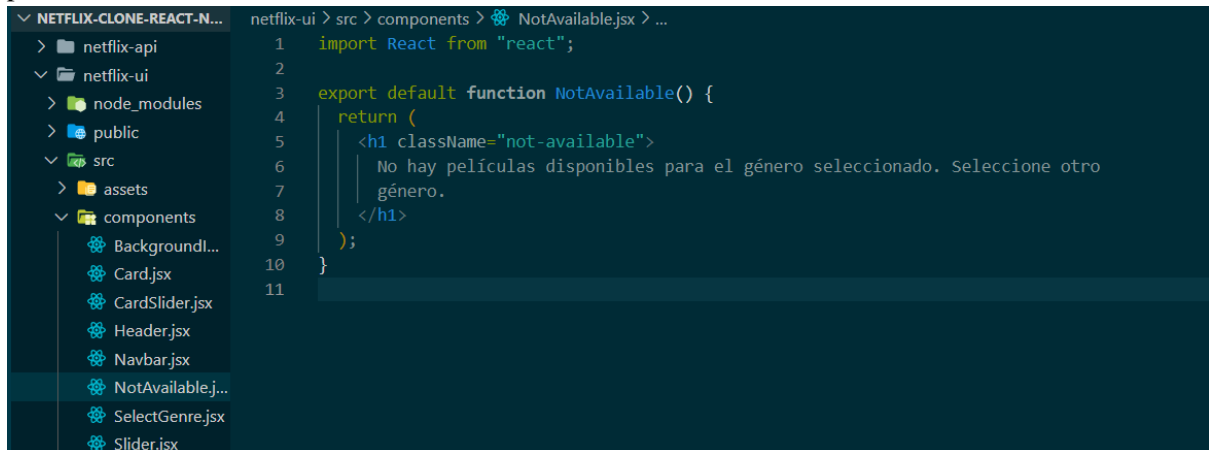
```

```

netflix-ui > src > components > NavBar.jsx > NavBar > links > name
1 import { signOut } from "firebase/auth";
2 import React, { useState } from "react";
3 import { Link } from "react-router-dom";
4 import styled from "styled-components";
5 import logo from "../assets/LOGO.png";
6 import { firebaseAuth } from "../utils/firebase-config";
7 import { FaPowerOff, FaSearch } from "react-icons/fa";
8 export default function NavBar({ isScrolled }) {
9   const [showSearch, setShowSearch] = useState(false);
10  const [inputHover, setInputHover] = useState(false);
11  const links = [
12    { name: "Home", link: "/" },
13    { name: "Programas de TV", link: "/tv" },
14    { name: "Películas", link: "/movies" },
15    { name: "My Lista", link: "/mylist" },
16  ];
17
18  return (
19    <Container>
20      <nav className=${isScrolled ? "scrolled" : ""} flex`>
21        <div className="left flex a-center">
22          <div className="brand flex a-center j-center">
23            <img src={logo} alt="Logo" />

```


Navbar encontramos los links de home que es el inicio, Programas de tv donde encontramos los diferentes contenidos, películas, y nuestra lista de películas, y el estilo de la misma donde ira posicionada.



```
netflix-ui > src > components > NotAvailable.jsx > ...
1 import React from "react";
2
3 export default function NotAvailable() {
4   return (
5     <h1 className="not-available">
6       No hay películas disponibles para el género seleccionado. Seleccione otro
7       género.
8     </h1>
9   );
10 }
11
```

En esta parte se conecta con los géneros, es como un alert donde nos dice que género no se encuentra registrado y nos manda un mensaje.



```
netflix-ui > src > components > SelectGenre.jsx > ...
1 import React from "react";
2 import { useDispatch } from "react-redux";
3 import styled from "styled-components";
4 import { fetchDataByGenre } from "../store";
5 export default function SelectGenre({ genres, type }) {
6   const dispatch = useDispatch();
7   return (
8     <Select
9       className="flex"
10      onChange={(e) => {
11        dispatch(
12          fetchDataByGenre({
13            genres,
14            genre: e.target.value,
15            type,
16          })
17        );
18      }}
19    />
20 )
21 }
22
```

SelectGenre se encarga de hacer una función de los datos que se quiere que lleven de los géneros que se encuentra de las películas ejemplo acción, drama.



```
netflix-ui > src > components > Slider.jsx > Slider
1 import React from "react";
2 import styled from "styled-components";
3 import CardSlider from "../CardSlider";
4 export default function Slider({ movies }) {
5   const getMoviesFromRange = (from, to) => {
6     return movies.slice(from, to);
7   };
8   return (
9     <Container>
10      <CardSlider data={getMoviesFromRange(0, 10)} title="Tendencias" />
11      <CardSlider data={getMoviesFromRange(10, 20)} title="Nuevos lanzamientos" />
12      <CardSlider
13        data={getMoviesFromRange(20, 30)}
14        title="Películas taquilleras"
15      />
16      <CardSlider
17        data={getMoviesFromRange(30, 40)}
18        title="Popular en Netflix"
19      />
20      <CardSlider data={getMoviesFromRange(40, 50)} title="Películas de acción" />
21      <CardSlider data={getMoviesFromRange(50, 60)} title="Epicas" />
22    </Container>
23  )
24 }
25
```

Aquí se encuentra con un card list de los datos de las películas, donde nos hace la referencia del dato de las películas en tendencia, nuevos lanzamientos entonces se conectan con el dato de películas que existe y hacen la función de los datos de las películas que se conecta con los géneros.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > pages > Login.jsx > Login > onAuthStateChanged() callback
1 import React, { useState } from "react";
2 import styled from "styled-components";
3 import logo from "../assets/LOGO.png";
4 import background from "../assets/login1.jpg";
5 import { useNavigate } from "react-router-dom";
6 import BackgroundImage from "../components/BackgroundImage";
7 import Header from "../components/Header";
8 import { onAuthStateChanged, signInWithEmailAndPassword } from "firebase/auth";
9 import { firebaseAuth } from "../utils/firebase-config";
10
11 function Login() {
12   const [email, setEmail] = useState("");
13   const [password, setPassword] = useState("");
14   const navigate = useNavigate();
15
16   const handleLogin = async () => {
17     try {
18       await signInWithEmailAndPassword(firebaseAuth, email, password);
19     } catch (error) {
20       console.log(error.code);
21     }
22   };

```

Hace uso de conectar y guardar los datos nuestro email, y contraseña es donde se hace la función de la base, donde hace referencia y se conecta a la navegación.

```
netflix-api netflix-ui > src > pages > Movies.jsx
1 import React, { useEffect, useState } from "react";
2 import styled from "styled-components";
3 import Navbar from "../components/Navbar";
4 import CardSlider from "../components/CardSlider";
5 import { onAuthStateChanged } from "firebase/auth";
6 import { firebaseAuth } from "../utils/firebase-config";
7 import { useNavigate } from "react-router-dom";
8 import { useSelector, useDispatch } from "react-redux";
9 import { fetchMovies, getGenres } from "../store";
10 import SelectGenre from "../components/SelectGenre";
11 import Slider from "../components/Slider";
12 import NotAvailable from "../components/NotAvailable";
13
14 function MoviePage() {
15   const [isScrolled, setIsScrolled] = useState(false);
16   const movies = useSelector((state) => state.netflix.movies);
17   const genres = useSelector((state) => state.netflix.genres);
18   const genresLoaded = useSelector((state) => state.netflix.genresLoaded);
19
20   const navigate = useNavigate();
21   const dispatch = useDispatch();
22
23   useEffect(() => {
24     dispatch(getGenres());
25   }, []);

```

En donde se encuentra los componentes de la página de las películas, donde conecta los géneros, las películas y hace un efecto en el diseño de la búsqueda de películas.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > pages > Netflix.jsx > Netflix
1 import React, { useEffect, useState } from "react";
2 import styled from "styled-components";
3 import Navbar from "../components/Navbar";
4 import backgroundImage from "../assets/Screen.webp";
5 import MovieLogo from "../assets/mer.png";
6
7 import { onAuthStateChanged } from "firebase/auth";
8 import { firebaseAuth } from "../utils/firebase-config";
9 import { useNavigate } from "react-router-dom";
10 import { useSelector, useDispatch } from "react-redux";
11 import { fetchMovies, getGenres } from "../store";
12 import { FaPlay } from "react-icons/fa";
13 import { AiOutlineInfoCircle } from "react-icons/ai";
14 import Slider from "../components/Slider";
15 function Netflix() {
16   const [isScrolled, setIsScrolled] = useState(false);
17   const movies = useSelector((state) => state.netflix.movies);
18   const genres = useSelector((state) => state.netflix.genres);
19   const genresLoaded = useSelector((state) => state.netflix.genresLoaded);
20
21   const navigate = useNavigate();
22   const dispatch = useDispatch();
```

en donde hacemos la importación de muchos componentes y donde importa nuestra imagen de portada, hacemos las importaciones que conectan con nuestro home.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > pages > Player.jsx > Player
1 import React from "react";
2 import styled from "styled-components";
3 import { BsArrowLeft } from "react-icons/bs";
4 import { useNavigate } from "react-router-dom";
5 import video from "../assets/video.mp4";
6 export default function Player() {
7   const navigate = useNavigate();
8
9   return (
10     <Container>
11       <div className="player">
12         <div className="back">
13           <BsArrowLeft onClick={() => navigate(-1)} />
14         </div>
15         <video src={video} autoPlay loop controls muted />
16       </div>
17     </Container>
18   );
19 }
20
```

Esta función es de video donde indica el icono player que haga una función que al hacer clic muestre el video que se seleccionó y que nos muestra su diseño dependiendo.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > pages > Signup.jsx > Signup
1 import {
2   createUserWithEmailAndPassword,
3   onAuthStateChanged,
4 } from "firebase/auth";
5 import React, { useState } from "react";
6 import { useNavigate } from "react-router-dom";
7 import styled from "styled-components";
8 import BackgroundImage from "../components/BackgroundImage";
9 import Header from "../components/Header";
10 import { firebaseAuth } from "../utils/firebase-config";
11 function Signup() {
12   const [showPassword, setShowPassword] = useState(false);
13   const [formValues, setFormValues] = useState({
14     email: "",
15     password: "",
16   });
17   const navigate = useNavigate();
18
19   const handleSignIn = async () => {
20     try {
21       const { email, password } = formValues;
22       await createUserWithEmailAndPassword(firebaseAuth, email, password);
23     } catch (error) {
24       console.log(error);
25     }
26   };

```

Aquí hacemos uso de igual manera con nuestro login donde se hace una validación de nuestro correo y contraseña. y el error en la base de datos.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > pages > TVShows.jsx > ...
1 import React, { useEffect, useState } from "react";
2 import styled from "styled-components";
3 import Navbar from "../components/Navbar";
4 import CardSlider from "../components/CardSlider";
5 import { onAuthStateChanged } from "firebase/auth";
6 import { firebaseAuth } from "../utils/firebase-config";
7 import { useNavigate } from "react-router-dom";
8 import { useSelector, useDispatch } from "react-redux";
9 import { fetchMovies, getGenres } from "../store";
10 import SelectGenre from "../components/SelectGenre";
11 import Slider from "../components/Slider";
12
13 function TVShows() {
14   const [isScrolled, setIsScrolled] = useState(false);
15   const movies = useSelector((state) => state.netflix.movies);
16   const genres = useSelector((state) => state.netflix.genres);
17   const genresLoaded = useSelector((state) => state.netflix.genresLoaded);
18   const dataLoading = useSelector((state) => state.netflix.dataLoading);
19
20   const navigate = useNavigate();
21   const dispatch = useDispatch();

```

Este conecta con la función de los shows de tv donde selecciona la película y el género de la misma.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > pages > UserListedMovies.jsx > UserListedMovies
1 import axios from "axios";
2 import { onAuthStateChanged } from "firebase/auth";
3 import React, { useEffect, useState } from "react";
4 import { useNavigate } from "react-router-dom";
5 import { firebaseAuth } from "../utils/firebase-config";
6 import Card from "../components/Card";
7 import styled from "styled-components";
8 import Navbar from "../components/Navbar";
9 import { getUsersLikedMovies } from "../store";
10 import { useDispatch, useSelector } from "react-redux";
11
12 export default function UserListedMovies() {
13   const movies = useSelector((state) => state.netflix.movies);
14   const dispatch = useDispatch();
15   const navigate = useNavigate();
16   const [isScrolled, setIsScrolled] = useState(false);
17   const [email, setEmail] = useState(undefined);
18
19   onAuthStateChanged(firebaseAuth, (currentUser) => {
20     if (currentUser) setEmail(currentUser.email);
21     else navigate("/login");
22   });
```

Hacemos uso de la lista de películas donde se hace una función de la lista y define el login de registro de nuestro correo.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > store > index.js > removeMovieFromLiked > createAsyncThunk("netflix/deleteLiked") callback
1 import {
2   configureStore,
3   createAsyncThunk,
4   createSlice,
5 } from "@reduxjs/toolkit";
6 import axios from "axios";
7 import { API_KEY, TMDB_BASE_URL } from "../utils/constants";
8
9 const initialState = {
10   movies: [],
11   genresLoaded: false,
12   genres: [],
13 };
14
15 export const getGenres = createAsyncThunk("netflix/genres", async () => {
16   const {
17     data: { genres },
18   } = await axios.get(
19     "https://api.themoviedb.org/3/genre/movie/list?api_key=3d39d6bfe362592e6aa293f01fbcf9b9"
20   );
21   return genres;
22 });
```

se importa la paquetería de axios y se hace una constante de inicialización de trabajo

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > utils > constants.js > TMDB_BASE_URL
1 import const API_KEY = "3d39d6bfe362592e6aa293f01fbcf9b9";
2 export const TMDB_BASE_URL = "https://api.themoviedb.org/3";
3
```

en nuestro script exportamos la url de la base de datos y nuestra llave foránea

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > utils > JS firebase-config.js > ...
1 import { getAuth } from "firebase/auth";
2 import { initializeApp } from "firebase/app";
3
4 const firebaseConfig = {
5   apiKey: "AIzaSyC1Hf0_rdWLBzDPJPcO9CJN4y6M6-EgKH4",
6   authDomain: "react-auth-6788d.firebaseio.com",
7   projectId: "react-auth-6788d",
8   storageBucket: "react-auth-6788d.appspot.com",
9   messagingSenderId: "131797845021",
10  appId: "1:131797845021:web:3f7ff4766e2b89ca5d32f4",
11  measurementId: "G-VWPBR1NSLL",
12 };
13
14 const app = initializeApp(firebaseConfig);
15 export const firebaseAuth = getAuth(app);
16
```

Aquí hacemos uso de firebase que se encarga de nuestro usuario donde esta aplicación permite una restricción de nuestros usuarios donde nos da nuestra llave foránea.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > JS App.js > ...
1 import React from "react";
2 import { BrowserRouter, Route, Routes } from "react-router-dom";
3 import Login from "./pages/Login";
4 import MoviePage from "./pages/Movies";
5 import Netflix from "./pages/Netflix";
6 import Player from "./pages/Player";
7 import Signup from "./pages/Signup";
8 import TVShows from "./pages/TVShows";
9 import UserListedMovies from "./pages/UserListedMovies";
10
11 export default function App() {
12   return (
13     <BrowserRouter>
14       <Routes>
15         <Route exact path="/login" element={<Login />} />
16         <Route exact path="/signup" element={<Signup />} />
17         <Route exact path="/player" element={<Player />} />
18         <Route exact path="/tv" element={<TVShows />} />
19         <Route exact path="/movies" element={<MoviePage />} />
20         <Route exact path="/new" element={<Player />} />
21         <Route exact path="/mylist" element={<UserListedMovies />} />
22         <Route exact path="/" element={<Netflix />} />
23       </Routes>
24     </BrowserRouter>
25   );
26 }
```

Aquí hace la función de referencia de nuestro elemento de el navbar donde nos conecta con cada uno de nuestros links.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > index.css > body
1 body {
2   margin: 0;
3   font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",
4     "Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",
5     sans-serif;
6   -webkit-font-smoothing: antialiased;
7   -moz-osx-font-smoothing: grayscale;
8   background-color: black;
9   color: white;
10  }
11
12  * {
13    margin: 0;
14    padding: 0;
15    box-sizing: border-box;
16  }
17
18  html,
19  body {
```

Es una parte de nuestro diseño de nuestro texto, tipo de escritura, color y el borde.

```
NETFLIX-CLONE-REACT-N... netflix-ui > src > index.js > ...
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import { Provider } from "react-redux";
4 import "./index.css";
5 import App from "./App";
6 import { store } from "./store";
7
8 const root = ReactDOM.createRoot(document.getElementById("root"));
9 root.render(
10   <React.StrictMode>
11     <Provider store={store}>
12       <App />
13     </Provider>
14   </React.StrictMode>
15 );
16
```

hacemos referencia a nuestro css para hacer efectivo el cambio de texto.

Carpeta netflix-api

```
netflix-api > controllers > JS UserController.js > ...
1  const User = require("../models/UserModel");
2
3  module.exports.getLikedMovies = async (req, res) => {
4    try {
5      const { email } = req.params;
6      const user = await await User.findOne({ email });
7      if (user) {
8        return res.json({ msg: "success", movies: user.likedMovies });
9      } else return res.json({ msg: "User with given email not found." });
10   } catch (error) {
11     return res.json({ msg: "Error fetching movies." });
12   }
13 };
14
15 module.exports.addToLikedMovies = async (req, res) => {
16   try {
17     const { email, data } = req.body;
18     const user = await await User.findOne({ email });
19     if (user) {
20       const { likedMovies } = user;
21       const movieAlreadyLiked = likedMovies.find(({ id }) => id === data.id);
22       if (!movieAlreadyLiked) {
23         await User.findByIdAndUpdate(
24           user.id,
```

Se crea una constante donde conecta nuestro Gmail, prácticamente donde se ira la información de registro.

```
netflix-api > models > JS UserModel.js > ...
1  const mongoose = require("mongoose");
2
3  const userSchema = new mongoose.Schema({
4    email: {
5      type: String,
6      required: true,
7      unique: true,
8      max: 50,
9    },
10   likedMovies: Array,
11 });
12
13 module.exports = mongoose.model("users", userSchema);
14
```

Creamos un modal donde hacemos referencia a nuestra mongoose que requiere el Gmail de tipo string


```
netflix-api > routes > UserRoutes.js > ...
1  const {
2    addToLikedMovies,
3    getLikedMovies,
4    removeFromLikedMovies,
5  } = require("../controllers/UserController");
6
7  const router = require("express").Router();
8
9  router.get("/liked/:email", getLikedMovies);
10 router.post("/add", addToLikedMovies);
11 router.put("/remove", removeFromLikedMovies);
12
13 module.exports = router;
14
```

Aquí seleccionamos las rutas donde llamamos los controles y donde requerimos express

```
netflix-api > .env
1  MONGO_URL="mongodb://localhost:27017/netflix"
```

Llamado de nuestra base de datos con un link para conectar a mongo

```
netflix-api > package.json > {} scripts > start
1  {
2    "name": "netflix-api",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start": "nodemon Server.js"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "cors": "^2.8.5",
14     "express": "^4.18.1",
15     "mongoose": "^6.3.6",
16     "nodemon": "^2.0.16"
17   }
18 }
19
20
```

Encontramos las dependencias utilizadas y llamamos con start a que nos dirija a nuestro script podemos a nuestro server

```

1  const express = require("express");
2  const cors = require("cors");
3  const userRoutes = require("../routes/UserRoutes");
4  const mongoose = require("mongoose");
5
6  const app = express();
7
8  app.use(cors());
9  app.use(express.json());
10
11  mongoose
12  .connect("mongodb://localhost:27017/netflix", {
13    useNewUrlParser: true,
14    useUnifiedTopology: true,
15  })
16  .then(() => {
17    console.log("DB Connection Successfull");
18  })
19  .catch((err) => {
20    console.log(err.message);
21  });
22
23  app.use("/api/user", userRoutes);
24
25  app.listen(5000, () => {
26    console.log("server started on port 5000");
27  });

```

Aquí colocamos el llamado a nuestra base de datos de referencia con su respectivo nombre, donde le decimos que en el puerto 5000 se encontrará alojado y llamar a nuestro api user

6.3 PRESUPUESTO DE IMPLEMENTACIÓN

sistema de streaming

En esta parte se definirán todos los recursos técnicos necesarios para la realización del proyecto identificando las necesidades para la construcción o instalación del proyecto, su implementación y operación, para ello se cuenta con información especializada proporcionada por expertos en la técnica de transmisión de audio y video por redes. El objetivo de esta etapa es el de estudiar y definir sobre la selección de los insumos, equipos técnicos y todos los recursos necesarios para llevar a cabo el proyecto que permita tener una evaluación económica del proyecto y establecer los elementos técnicos necesarios para el mismo.

Tecnología. Un factor diferenciador con respecto a otras empresas que ofrecen el servicio de streaming es la tecnología utilizada para ofrecer el servicio. Se contará con el software que proporciona las herramientas de transmisión a través de una plataforma desarrollada en software libre que proporcione el servicio de streaming en vivo y en diferido.

Equipos y software. Servidor Streaming: Un servidor de streaming es un equipo de cómputo dedicado a la transmisión de audio y video, en el servidor se ejecuta una herramienta que permite ejecutar el servicio. El servidor cuenta con elementos hardware como memoria, discos duros, procesadores (varios núcleos), etc. que proporcionan un número determinado de conexiones dependiendo de las características de su hardware.

Inversión en software y equipo.

Artículo	Cant.	Valor unitario	Valor total	Vida útil
----------	-------	----------------	-------------	-----------

Computador portátil	2	\$800	1,600	8 años
licencia player	2	\$15	30	10 años
licencia servidor streaming	1	\$350	350	10 años
Conexión a internet 15GB 4G por mes	1	\$25	\$300	1 año
total			\$2,280	

Servicios de streaming

Producto. El proyecto plantea diferentes productos como; streaming, video bajo demanda y almacenamiento. Los cuales se van a ofrecer mediante una plataforma a la cual una vez el cliente haya realizado el pago, podrá recibir el servicio de streaming.

Demanda. En el caso del presente proyecto de streaming, está dirigido a la prestación de un servicio a través de una plataforma que permitirá conectar a un público o auditorio con un evento en cualquier parte del mundo, sumado a factores como el económico, facilidad de acceso, cobertura, entre otros. Se están inclinando diferentes empresas de todos los sectores a emplear transmisiones por esta herramienta multimedia.

Oferta. De esta manera, el servicio de streaming tiene su oferta en empresas del orden regional. La idea a mediano plazo es establecer contactos con empresas de otros sectores económicos en las cuales se identifique previamente mediante investigación de mercados la necesidad de emplear el servicio.

Precio o Tarifa. Se denomina precio al pago o recompensa asignado a la obtención de un bien o servicio, en este caso el precio es el valor expresado en dinero del servicio ofrecido en el mercado. Es un elemento fundamental en el proyecto, pues es uno de los determinantes que participa en la definición de ingresos.

Servicio	Costo Unitario
Plan Streaming 1 (2 ventanas)	\$10.00
Plan Streaming 2 (4 ventanas)	\$15.00

A. Glosario

- *API (Application Programming Interface)*: Es el conjunto de funciones y subrutinas que nos ofrece la web para obtener toda la información sobre películas, estrenos y listas.
- *Sistema*: Referente a la aplicación web desarrollada.
- *Modelos*: Es la estructura creada para el usuario que será almacenada en la base de datos y contendrá toda su información.

B. Fuentes de información

BIBLIOGRAFÍAS

-JOHN WILEY & SONS INC. Pedro Teixeira. Professional Node.js: Building Javascript Based Scalable Software. 2013.