



Universidad Tecnológica Centroamericana
Facultad de Ingeniería

Examen #1

Integrantes de Grupo:

Andrea J. Orteiz	12241140
Tatiana Z. Garcia	12241079

Clase y Sección:

Experiencia de Usuario 1945

Docente:

Ing. Claudia Cortés

Tegucigalpa M.D.C

11 de noviembre de 2024

Índice

EJECUCIÓN DEL PROYECTO	3
LIBRERÍAS UTILIZADAS.....	4
EXPLICACIÓN DE CÓDIGO	5
APPBAR	6
CARRUSEL	7
FOOTER	8
INDEX	9
SECCIONES	9

Ejecución del Proyecto

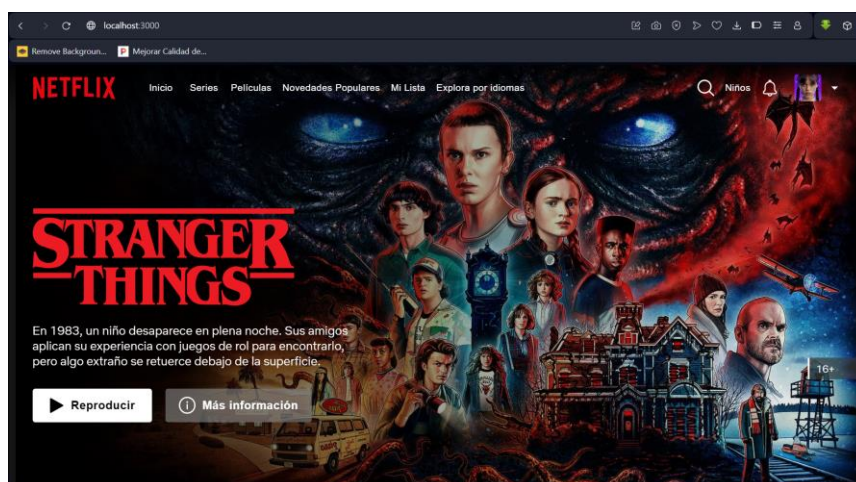
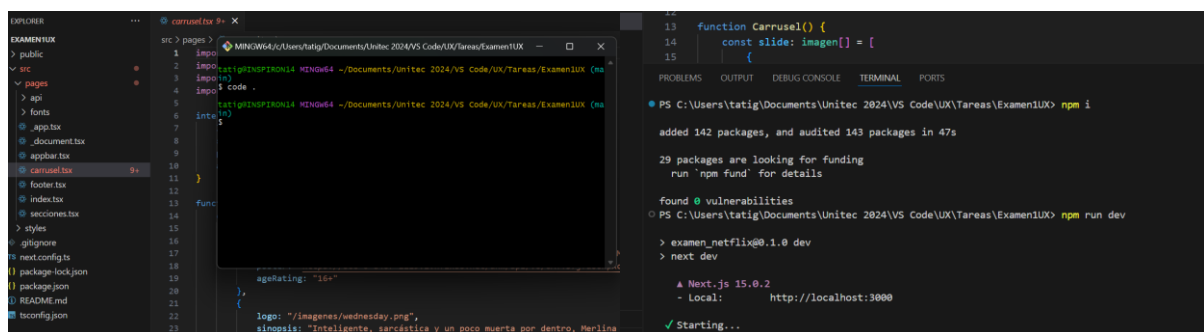
Para la ejecución del proyecto, se requiere clonar en nuestro [repositorio](#) y la instalación de [Node.js](#) para ejecutar los siguientes comandos, en nuestro caso en la terminal de VS Code:

- `npm i` (npm install) – para instalar todas las librerías
- `npm run dev` – para la ejecución del proyecto, y abrir el link del server que provee el comando.

En caso de errores al copilar ejecutar los siguientes comandos:

- `npm install react@^18.0.0 react-dom@^18.0.0`
- `npm install @mui/material @emotion/react @emotion/styled`
- `npm install react-material-ui-carousel --save`

Con esto ya se podrá abrir nuestra página principal de Netflix.



Nota: se puede usar 'code .' para abrir la dirección ubicada en VS Code

Librerías Utilizadas

Principalmente usamos la versión 18 de React, para el uso ciertos elementos que nos dio la posibilidad de crear la página principal de Netflix, así como el uso de Material UI (MUI), que nos guiamos por probar este recurso, por los elementos que proveía y porque Netflix.

Elementos usados:

- [Carousel](#): Esta librería fue utilizada para listar cada una de las series y películas (Cards), así como el cambio de imágenes del banner del contenido destacado que esta en el principio.
- [Stack](#): Ayuda listando y distribuyendo las Cards, para que aparezcan en una fila.
- [Card](#): Este elemento fue utilizado para las películas y series que se muestran en cada sección, con el fin de que al pasar el cursor en la imagen pueda mostrarse detalle y un tráiler o vista previa.
- [Box](#): Utilizado para una mejor manipulación de los otros elementos.
- [AppBar](#): Esto ayudo en la realización de la barra de navegación en la parte superior.

Estos son los elementos principales que se utilizaron para realizar el Netflix, pero también hubo otros que se explicarían en breve con el código en la siguiente sección. Cabe a mencionar, que se hizo uso de elementos de React como `useEffect()` y `useState()` para detección y definición de ciertos comportamientos de la página.

Explicación de Código

El código de Netflix de React se divide de la siguiente manera:

- Appbar
- Carrusel
- Footer
- Index
- Secciones

También se usó unos cuantos archivos de CSS, que no entraremos tan a detalle, pero cabe mencionar:

- Styles.css
- Globals.css

Home.module.css fue importado directamente por react pero no fue editado.

Appbar

El **Appbar** se enfoca en el diseño y estructura de la barra de navegación en la parte superior, como Logos, barra de tareas, e iconos representativos como tiene Netflix. También se hizo uso de eventos para

definir cuando se abre el menú de usuario. Aquí también se definieron algunos detalles como degradado, cambio de color de la barra de navegación al desplazarse hacia abajo y demás aspectos de CSS, para que se parezca lo más posible al de la página oficial.

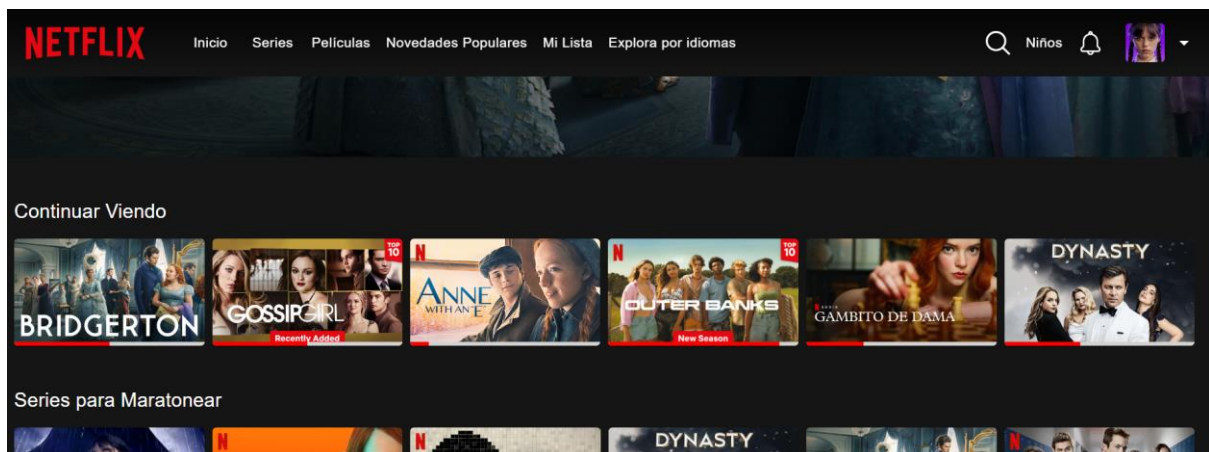
```

1 import * as React from 'react';
2 import AppBar from '@mui/material/AppBar';
3 import Box from '@mui/material/Box';
4 import Toolbar from '@mui/material/Toolbar';
5 import IconButton from '@mui/material/IconButton';
6 import Typography from '@mui/material/Typography';
7 import Menu from '@mui/material/Menu';
8 import MenuItem from '@mui/material/MenuItem';
9 import Container from '@mui/material/Container';
10 import Avatar from '@mui/material/Avatar';
11 import Button from '@mui/material/Button';
12 import Tooltip from '@mui/material/Tooltip';
13 import ListItem from '@mui/material/ListItem';
14 import ArrowDropDownIcon from '@mui/icons-material/ArrowDropDown';
15 import ArrowDropDownIcon from '@mui/icons-material/ArrowDropDown';

```

```

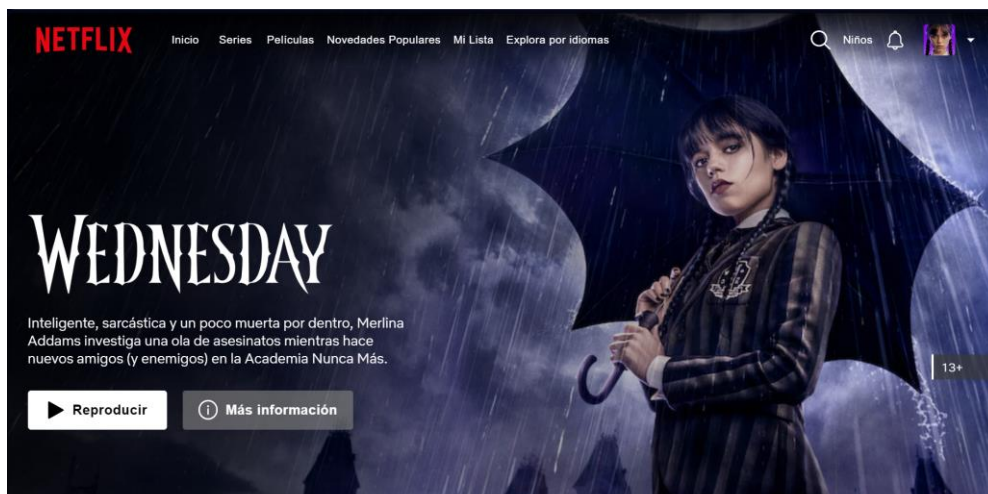
17 const pages = ['Inicio', 'Series', 'Películas', 'Novedades Populares', 'Mi Lista', 'Explora por idiomas'];
18 const settings = ['Administrar perfiles', 'Transferir perfil', 'Cuenta', 'Cerrar sesión'];
19
20 function ResponsiveAppBar() {
21   const [anchorElNav, setAnchorElNav] = React.useState(null | HTMLElement)(null);
22   const [anchorElUser, setAnchorElUser] = React.useState(null | HTMLElement)(null);
23   const [isMenuOpen, setIsMenuOpen] = React.useState(false);
24   const [bgColor, setBgColor] = React.useState('transparent');
25
26   const handleScroll = () => { ...
27     };
28
29   React.useEffect(() => { ...
30     }, []);
31
32   const handleOpenNavMenu = (event: React.MouseEvent<HTMLElement>) => { ...
33     };
34   const handleOpenUserMenu = (event: React.MouseEvent<HTMLElement>) => { ...
35     };
36   const handleCloseNavMenu = () => { ...
37     };
38   const handleCloseUserMenu = () => { ...
39     };
40
41   return (
42     <AppBar position="fixed" sx={{
43       backgroundColor: bgColor,
44       boxShadow: 'none',
45       textTransform: 'none',
46       transition: 'background-color 0.3s ease',
47     }}>
48
49       <div style={{ ...
50         }}>
51         </div>
52         <Container maxWidth="xl">
53           <Toolbar disableGutters>
54             <IconButton sx={{ display: { xs: 'none', md: 'flex' }, mr: 1 }}>
55             </IconButton>
56             <Typography ...
57             </Typography>
58             < /* Menú de usuario */
59             <Box sx={{ flexGrow: 1, display: { xs: 'flex', md: 'none' } }}>
60             </Box>
61             <IconButton sx={{ display: { xs: 'flex', md: 'none' }, mr: 1 }}>
62             </IconButton>
63             <Typography ...
64             </Typography>
65             <Box sx={{ flexGrow: 1, display: { xs: 'none', md: 'flex' } }}>
66             </Box>
67             <Box sx={{ ...
68             </Box>
69
70             < /* avatar */
71             <Box sx={{ flexGrow: 0, position: 'relative' }}>
72             >
73               <Tooltip title="Open settings"> ...
74               </Tooltip>
75               <Menu ...
76               </Menu>
77             </Box>
78
79           </Toolbar>
80
81         </Container>
82       </div>
83     </AppBar>
84   );
85 }
86
87 export default ResponsiveAppBar;
88
89 
```



Carrusel

En carrusel.tsx están los elementos para hacer el banner de la página principal, aquí se encuentra la información de las películas y series mostradas en el mismo, para que sean aplicados dentro del elemento *Carousel* y se muestren de manera automática el paso de imágenes con su respectiva información. Dentro de *Carousel*, se hace el mapeo a **Item** para que muestre cada una de estas imágenes, con su respectivo detalle, como la capa oscura en degradado como esta en Netflix y demás detalles de CSS.

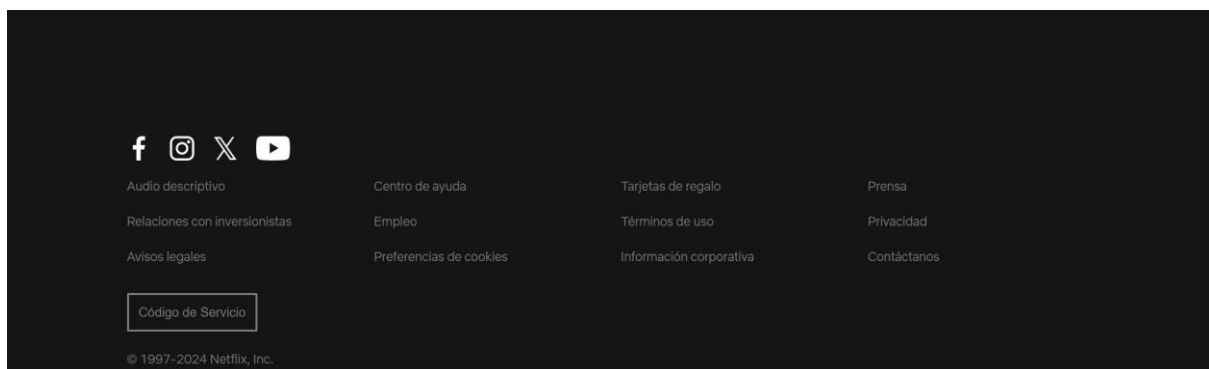
```
1 import * as React from 'react';
2 import Carousel from 'react-material-ui-carousel';
3 import { Paper, Button } from '@mui/material';
4 import Box from '@mui/material/Box';
5
6 > interface imagen { ...
11 }
12
13 function Carrusel() {
14 >   const slide: imagen[] = [ ...
45 ]
46
47   return (
48 >     <Carousel indicators={false} sx={{ width: '100vw' }}> ...
52     </Carousel>
53   )
54 }
55
56 function Item(props: any) {
57   return (
58     <Paper
59 >       style={{ ...
68 }}
69 >
70     < /* Capa oscura en degradado */ >
71 >     <div ...
81 >     ></div>
82
83     < /* Contenido del carrusel */ >
84 >     <div style={{ ...
116 >     </div>
117 >     <Box id="ageStyle" sx={{ ...
128 >     </Box>
129 >     </Paper>
```



Footer

El `footer.tsx` ya se enfocaría en la estructura y diseño del pie de página, donde igual que Netflix muestra todos los detalles que la empresa muestra en la oficial, como las redes y links para información, como el de “Avisos Legales”, “Audio descriptivo”, etc.

```
1 import React from "react";
2
3 const Footer = () => {
4   return (
5     <footer>
6       <div className="footer">
7         <div id="redes">
8           <a href="https://www.facebook.com/Netflixxmx">
9             
10          </a>
11          <a href="https://www.instagram.com/NetflixLAT">
12            
13          </a>
14          <a href="https://x.com/netflixLAT">
15            
16          </a>
17          <a href="https://www.youtube.com/channel/UC5ZiUaIJ2b5dYBGf5iEUrA">
18            
19          </a>
20        </div>
21        <div className="footerTextStyle">
22          <div><p>Audio descriptivo</p></div>
23          <div><p>Centro de ayuda</p></div>
24          <div><p>Tarjetas de regalo</p></div>
25          <div><p>Prensa</p></div>
26          <div><p>Relaciones con inversionistas</p></div>
27          <div><p>Empleo</p></div>
28          <div><p>Términos de uso</p></div>
29          <div><p>Privacidad</p></div>
30          <div><p>Avisos legales</p></div>
31          <div><p>Preferencias de cookies</p></div>
32          <div><p>Información corporativa</p></div>
33          <div><p>Contáctanos</p></div>
34        </div>
35        <button id="FooterButton">Código de Servicio</button>
36        <p id="copyright">© 1997-2024 Netflix, Inc.</p>
37      </div>
38    </footer>
39  );
40 };
41
42 export default Footer;
```



Index

Enfocado mas que todo en llamar las otras clases para la ejecución de estas y se puedan ver en el sitio web o dispositivo. También, aquí estaría la información de las Cards, que se pide en Secciones.tsx

```
1  export default function Home() {
2    return (
3      <>
4        <Head>
5          <link rel="icon" href="/imagenes/netflixLogo.png" />
6          <title>Netflix</title>
7        </Head>
8        <AppBar />
9        <Carrusel />
10       <Secciones titulo="Continuar Viendo" cards={cv_cards} />
11       <Secciones titulo="Series para Maratonear" cards={r_cards} />
12       <Secciones titulo="Mi Lista" cards={sp_cards} />
13       <Secciones titulo="Favoritos del público" cards={fp_cards} />
14       <Secciones titulo="Series Dramáticas" cards={N_cards} />
15       <Secciones titulo="Volver a verlo" cards={v_cards} />
16       <Footer />
17     </>
18   );
19 }
```

Secciones

El archivo *secciones.tsx* está enfocado en la estructura y funcionamiento de lo que es la sección de series/películas listadas.

Librerías importadas:

```
1  import React, { useEffect, useState } from 'react';
2  import Carousel from 'react-material-ui-carousel';
3  import Stack from '@mui/material/Stack';
4  import { Card, CardActions, IconButton, CardActionArea, CardMedia, CardContent, Typography, useMediaQuery, useTheme, Button } from '@mui/material';
5  import AddIcon from '@mui/icons-material/Add';
6  import PlayArrowIcon from '@mui/icons-material/PlayArrow';
7  import ExpandMoreIcon from '@mui/icons-material/ExpandMore';
8  import LinearProgress from '@mui/material/LinearProgress';
```

Estas propiedades provee elementos pedidos en el index.tsx para la información de cada sección y sus cartas (películas y series).

```
1  interface Propiedades {
2    titulo: string;
3    cards: {
4      imagen: string;
5      titulo: string;
6      edad: string;
7      detalle: string; //Si son series N° de temporadas, si son películas duración
8      info: string;
9      barra?: number;
10     url: string;
11   };
12 }
```

En la función Secciones pedimos título de la Sección (como Continuar Viendo, Mi lista, etc), así como la información de las Cards. *useMediaQuery* nos ayuda a identificar el tamaño de dispositivo con el uso de *useTheme()*, con esto podemos definir cuantas de las Cards mostrar en el listado, con el uso de *useEffect()* y *useState()*. En **Secciones**, hace que se muestre el título de la sección, y el Carusel con las cartas, este llama a row, que divide las cartas en grupos para ser mostradas en el carusel. Se puede visitar la [página](#) utilizada como referencia.

```

35 function Secciones({ titulo, cards } : Propiedades) {
36   // Variables
37   const [showCards, setShowCards] = useState(6);
38   const smallScreen = useMediaQuery(useTheme()).breakpoints.down('sm');
39   const mediumScreen = useMediaQuery(useTheme()).breakpoints.down('md');
40   const largeScreen = useMediaQuery(useTheme()).breakpoints.down('lg');
41   const xlargeScreen = useMediaQuery(useTheme()).breakpoints.down('xl');
42
43   > useEffect(() => { ...
44     }, [smallScreen, mediumScreen, largeScreen, xlargeScreen]);
45
46   const row = () => {
47     const n_stack = Math.ceil(cards.length / showCards);
48     return Array.from({ length: n_stack }).map((_, i) => {
49       <Stack key={i} direction="row" spacing={1} sx={{ justifyContent: "center" }}>
50         {cards.slice(i * showCards, i * showCards + showCards).map((card, index) => {
51           <Item key={index} imagen={card.imagen} titulo={card.titulo}
52             edad={card.edad} detalle={card.detalle} info={card.info} url={card.url} barra={card.barra}
53           >
54             </Stack>
55         )}
56       </Stack>
57     )};
58   };
59
60   return (
61     <div>
62       <Typography ...
63       </Typography>
64
65       <Carousel ...
66       </Carousel>
67     </div>
68   );
69 }

```

En el método **Item** se piden todos los elementos de la Card. Aquí es donde se define el estilo que tendrá cada Card y su comportamiento cuando se pasa el cursor (*Hover*). Donde se valida de que, si esto pasa, se muestra un video y sino una imagen. Luego, se valida que, si se pasa el cursor del ratón, muestre los detalles de la serie o película.

```

104 const Item = ({ imagen, titulo, edad, detalle, info, barra, url }: CardProp) => {
105   const [isHovered, setIsHovered] = useState(false);
106
107   return (
108     <div
109       style={{ ...
110     }}
111   >
112     <Card
113       raised={isHovered}
114       sx={{
115         transform: isHovered ? "scale(1.25)" : "scale(1)",
116         zIndex: isHovered ? 10 : 1,
117         transition: "transform 0.15s ease-in-out",
118         position: 'relative',
119         overflow: 'hidden',
120         backgroundColor: '#151515',
121         color: 'white',
122       }}
123     >
124       <CardActionArea
125         onMouseEnter={() => setIsHovered(true)}
126         onMouseLeave={() => setIsHovered(false)}
127       >
128         {isHovered ? (...
129         ) : (...
130         )}
131         {isHovered && (...
132         )}
133       </CardActionArea>
134     </Card>
135   </div>
136 );
137 }

```

