

Informe de Diseño

Proyecto de Ingeniería del Software 1



Grupo: Innovando Mentes

Estudiantes:

Aracellys Mercado Cano

Ana Lorena González Espinoza

Byron Abarca Chacón

Profesor:

Álvaro Cordero Díaz.

Periodo lectivo II cuatrimestre 2021

Documento: Proyecto 1 Plantilla –Informe de diseño de software	Código: BISOFT_PL_IAS
Equipo:	Última modificación:
Revisado por:	Aprobado por:
Versión: 1.0	Página 1 de 4

Tabla de contenido

I. Introducción	3
II. Referencias	4
III. Especificación de diseño	5
1. Diseño gráfico	5
2. Estructura de la aplicación	6
3. Mapa de navegación	8
4. Wireframes	10
5. Estándares	20

I. Introducción

Por medio de la información presentada en este documento se desea dar a conocer todos los detalles tomados en cuenta para la creación y desarrollo de la página web de Cine Rex.

El grupo encargado de diseñar esta página web, llamado Innovando Mentes los cuales se han enfocado principalmente en las necesidades del cliente, para así poder lograr un buen trabajo y poder concluir con éxito su propósito.

Para así lograr tener una buena aceptación de parte de la audiencia y culminar con éxito.

Teniendo así como visión principal realizar un buen trabajo grupal, siempre tomando en cuenta las necesidades y peticiones del cliente para así poder dar la talla y entregar un buen trabajo.

II. Referencias

Abarca, B. González, A y Mercado, A. (2021). Lista de requerimientos funcionales.

https://moodle.ucenfotec.ac.cr/inicio/pluginfile.php/309265/mod_resource/content/2/Documento%20de%20dise%C3%B1o.pdf

<https://moodle.ucenfotec.ac.cr/inicio/mod/resource/view.php?id=242572>

III. Especificación de diseño

La especificación del diseño se encuentra conformado por cinco partes que son el diseño gráfico; en esta parte se podrá visualizar la tipografía que se utilizara en la página web como el tamaño de letra entre otras características, estructura de la aplicación; la estructura de la aplicación están las carpetas y archivos donde se almacenará toda la información de entretenimiento que se refleja en el navegador, mapa de navegación; este es un mapa tipo esquema donde estarán todos los títulos de los wireframes que permitirá saber cuales son los html que se deben crear en el aplicativo, wireframes; en esta sección solo se refleja las imágenes de las acciones y la información que se requiere para poder completar los html, etc y para finalizar están los estándares que requieren los HTML, CSS y JS.

1. Diseño gráfico

Es el encargado de la comunicación visual a través de

texto, imágenes, símbolos o ilustraciones. Cada persona implementa y desarrolla su diseño acorde a su personalidad, ideas y objetivos. Siempre es importante tener en cuenta los aspectos básicos que nos van a facilitar la visión principal de dicha creación.

Los pilares del diseño gráfico se pueden representar de esta manera:

Colores: Determinará el impacto y personalidad de cada diseñador



Figura 1 Representación de la paleta de colores

Tipografía: Determina el proceso de diseñar, arreglar o acomodar el texto.

- Cookie-Regular
- Raleway-Regular

Arte Visual: fotografía e ilustración



Figura 2 Arte visual

2. Estructura de la aplicación

Entre los estándares que se van a utilizar en este proyecto, está la creación de carpetas y archivos que permitirán aplicar las diferentes funcionalidades que

tendrá la página web.

Se dará inicio con la creación de una carpeta principal que se creará en el escritorio de la herramienta de trabajo, luego se integrarán los archivos que formarán parte del diseño de las páginas HTML que se podrán visualizar en la web.

La carpeta principal contiene tres subcarpetas conformadas por CSS, JS y IMG como también los archivos principales del HTML, en la Subcarpeta de CSS se encontrarán los archivos de cascada es decir los estilos que se aplicará a los archivos del HTML, en la subcarpeta de IMG estarán todas las imágenes que se usarán en el archivo HTML y para finalizar en la subcarpeta de JS que es el lenguaje de programación que se usará para mostrar la página en el navegador se encontrará la programación que necesitan todos los archivos HTML.

Ejemplo:

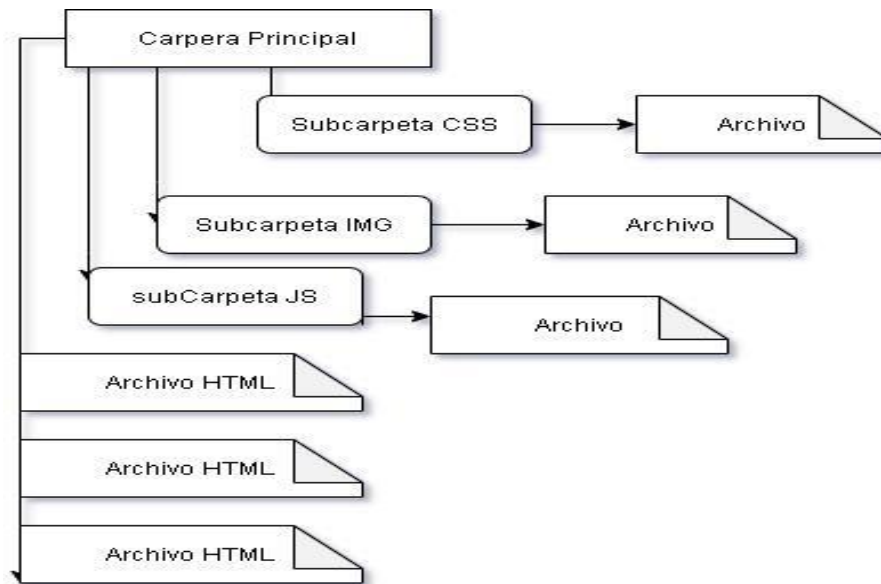


Figura 3 Estructura del proyecto 1

3. Mapa de navegación

En este apartado se representan todos los elementos que están conformados por los wireframe de cada apartado de la página.

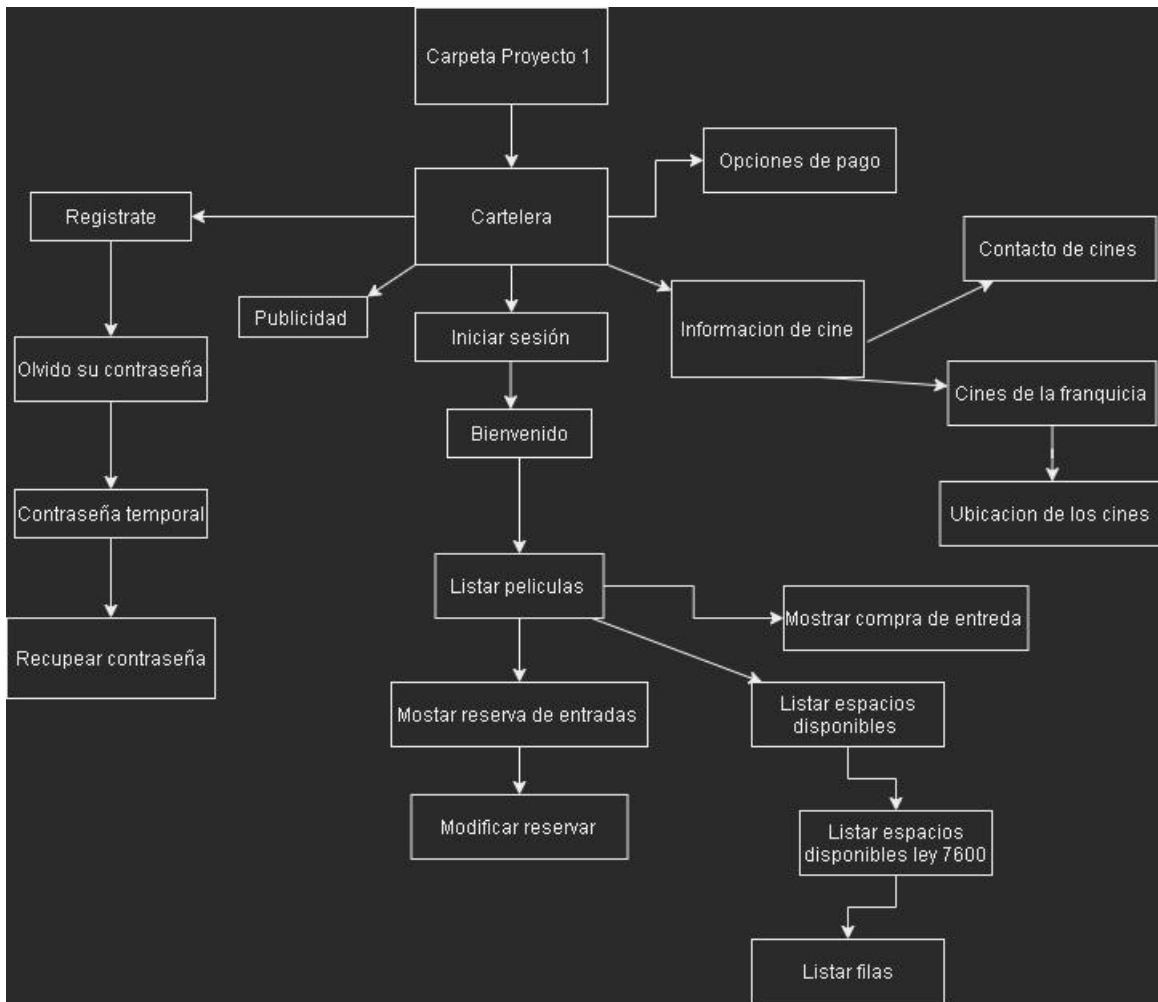


Figura 4 Mapa de navegación

Documento: Proyecto 1 Plantilla –Informe de diseño de software	Código: BISOFT_PL_IAS
Equipo:	Última modificación:
Revisado por:	Aprobado por:
Versión: 1.0	Página 2 de 4

4. Wireframes

Esta sección consta de los diferentes wireframes que se utilizaran para la elaboración de los diferentes sitios web y así mismo permite visualizar la funcionalidad que la página presenta al público. Por tanto, la funcionalidad incluye las acciones presentes en los requerimientos las cuales serán reflejadas a los usuarios y como estos podrán usar dicha página o aplicación.

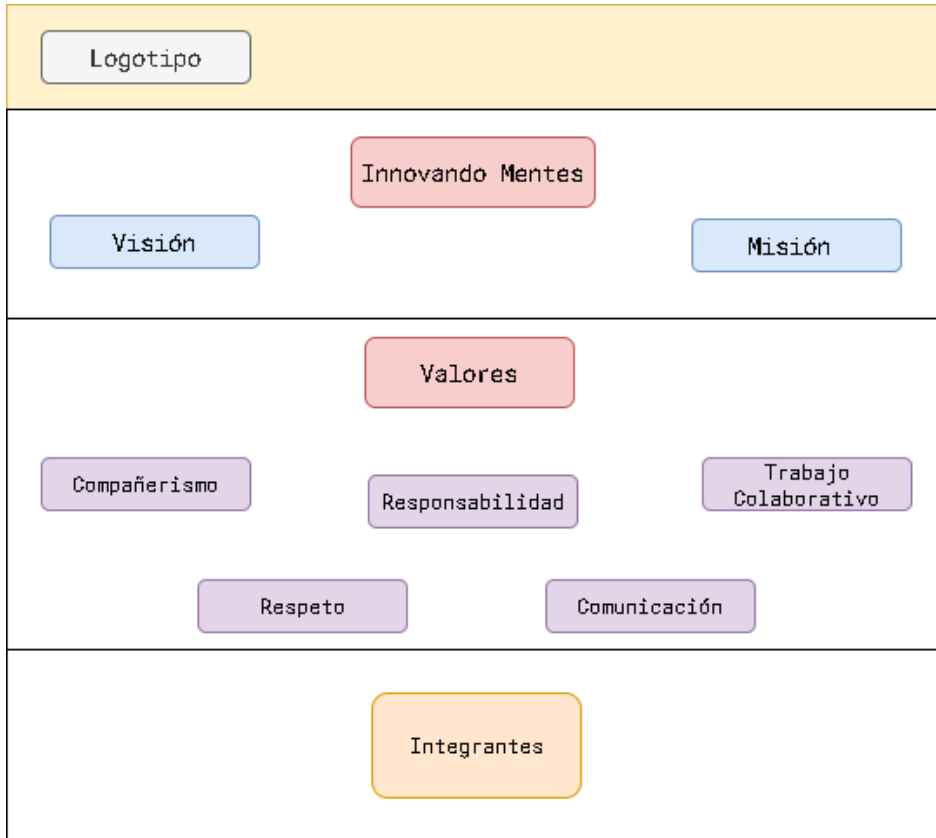


Figura 5 Esta figura representa el landing space del equipo

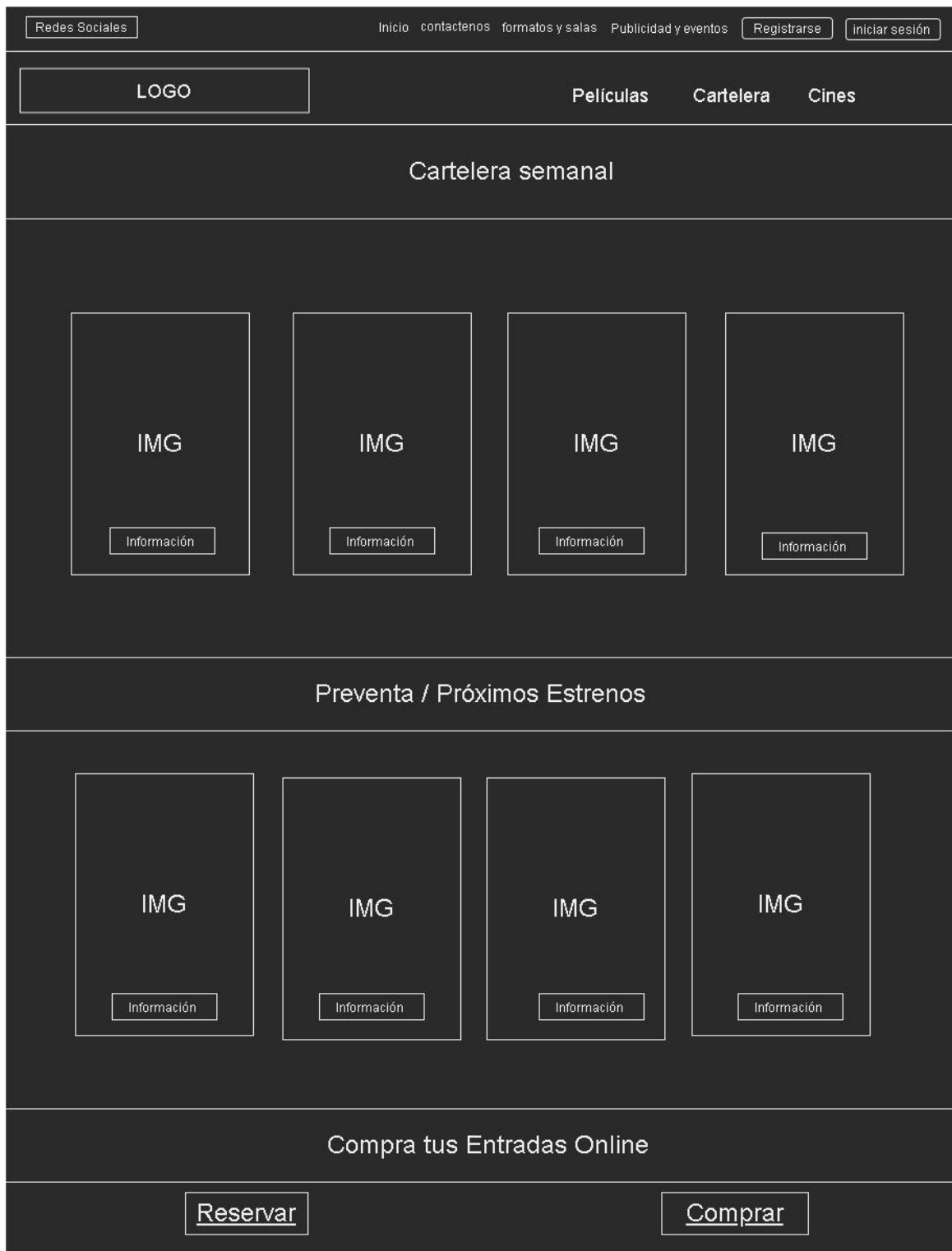


Figura 6 Diseño de la página

Bienvenido

Usuario o email

Contraseña

recuérdame [recuperar contraseña](#)

[Registrarme](#)

Figura 7 Inicio de sesión

Olvidaste tu contraseña

¿Cómo te gustaría recuperarla?

Email

Te enviaremos un email con las instrucciones sobre como restablecer la contraseña.

Figura 8 Olvidó contraseña

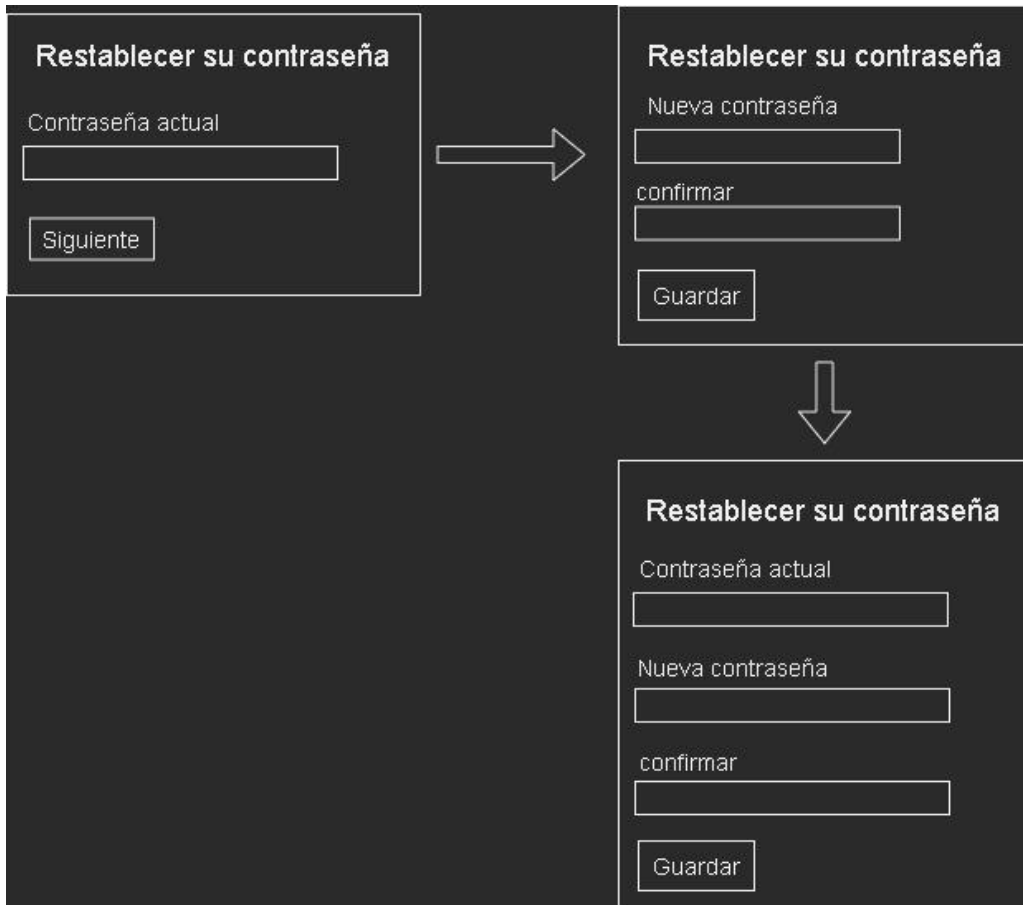


Figura 9 Restablecimiento de contraseña temporal y restablecimiento de olvido de contraseña.

opciones de pago

Elija su opción de pago

Tarjeta de crédito Tarjeta de debito

Pago con crédito

Ingrese la Información de su tarjeta de crédito

Titular de tarjeta

Numero de tarjeta

Fecha de expiración

Código de seguridad

Figura 10 Metodología de pago

Listar Películas

Categorías

▾

▾

▾

▾

Figura 11 Lista la categoría de las películas

Listar espacios disponibles de salas

Tipos de sala ▾

Butacas disponibles ▾

butacas seleccionadas ▾

Figura 12 Lista de espacios

disponibles

Listar compra de entradas

Tipos de entrada		Horario de películas	
Valor de la entrada	▾	Película Disponible	▾
Público	▾	Hora de inicio y finalización	▾

Figura 13 compra

de entradas

Modificar reserva

Modificar Sala ▾	Modificar Butaca ▾
Modificar categoría de película ▾	Modificar fila de butaca ▾
Modificar fecha y hora de la película ▾	Modificar tipo de butaca ▾

Esta seguro de realizar la reserva ▾

Factura de reserva

Esta seguro de realizar la reserva ▾

Figura 14 Modificación de la reserva

Mostrar reserva de las entradas

Tipo de sala ▾	Cantidad de butacas ▾
Tipo de butaca ▾	Número de filas ▾
Tipo de película ▾	Día ▾
Películas disponibles ▾	Hora ▾

Confirmar reserva ▾

Figura 15 Reserva de entrada

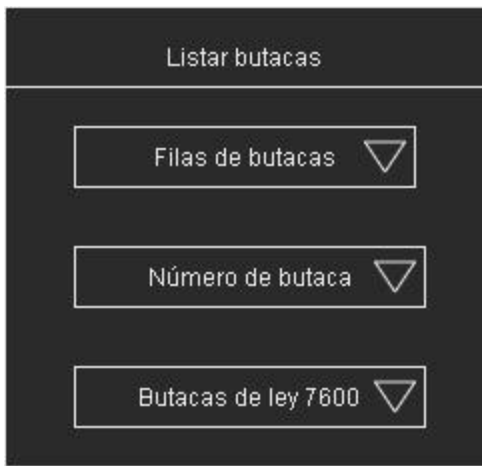


Figura 16 Lista las butacas

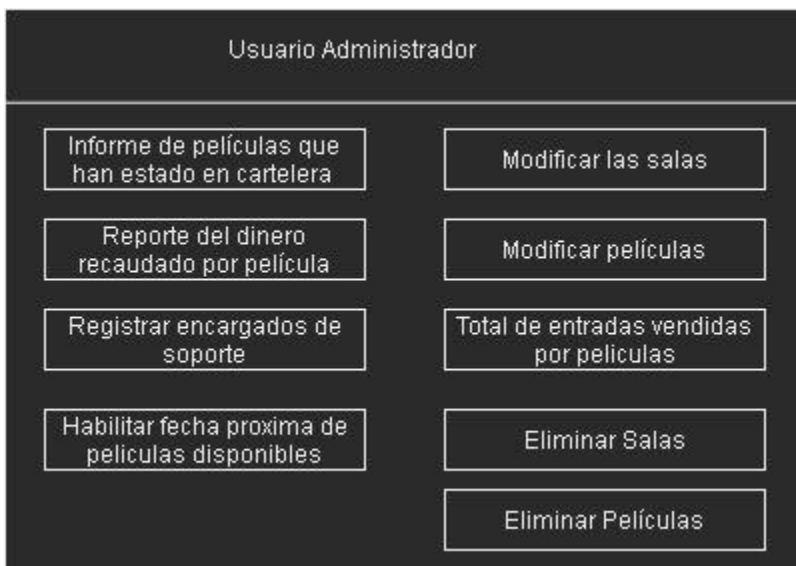


Figura 17 Opciones de administrador



Figura 18 Formas de contacto

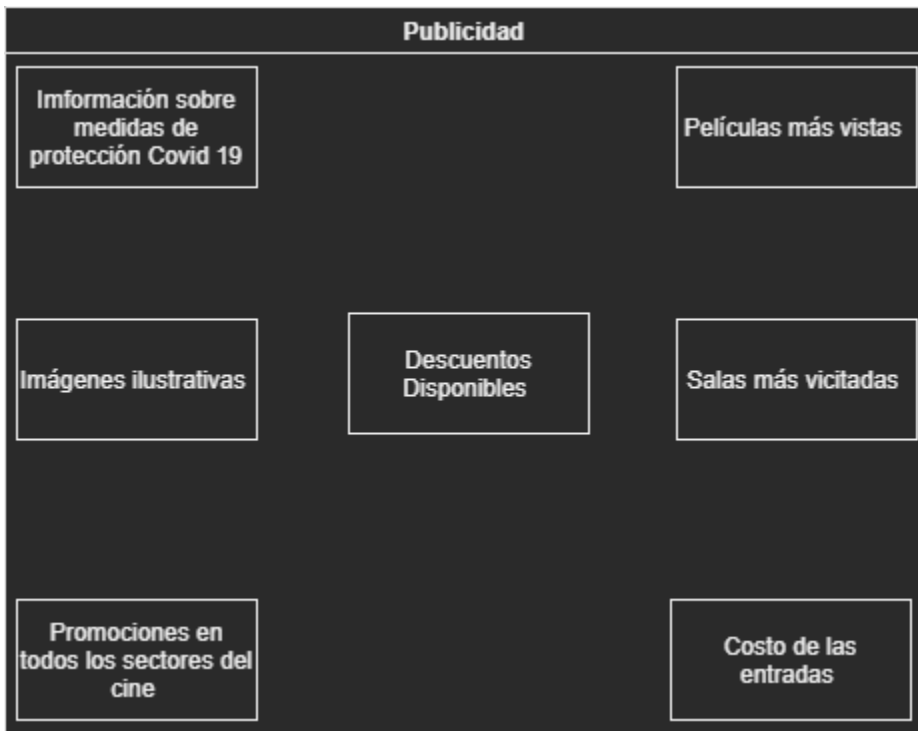



Figura 19 Publicidad del cine

Informacion de Cinex

Nombre de cada cine que sea parte de la cadena 


Ubicacion de cada uno de los cines 

Figura 20 Información del cine

5. Estándares

En esta sección se incluyen los estándares que se utilizan en las siguientes tecnologías:HTML,CSS Y JS.

General

Para nombrar los archivos vamos a utilizar la convención nomenclatura “camelcase“ y el nombre del archivo debe siempre hacer contexto a lo que se está trabajando dejando de lado las proposiciones contextuales.

- a. Bueno (vista Ejemplo)

Vista Ejemplos.html.

- b. Malo (vista de ejemplo)

vista de ejemplo.html

vista ejemplo.html

vista Ejemplo.html

Html

1. Utilizar siempre el mismo espaciado para indentación en todos los archivos, en el caso del HTML vamos a utilizar 4 espacios (se pueden utilizar smart tabs).

2. Utilizar siempre comillas dobles en los atributos en lugar de simples:

- a. Bueno

```
<div class="Tejemplo"></div>
```

- b. Malo:

```
<div class=fjemplo'></div>
```

3. El archivo que de entrada (landing page) por estándares siempre los vamos a llamar index.html

- a. Bueno

index.html

b. Malo:

home.html

II. dashboard.html

III. Entrada.html

En los archivos HTML no deberá haber CSS o JavaScript ya sea en el encabezado de la página o en línea, esto por cuestión de orden calidad y escalabilidad se deberá siempre manejar en archivos externos con su respectiva referencia en la cabecera de la página.

III.CSS

1. Los nombres de las clases siempre van a ir separadas por un guión usando un nombre explicativo de lo que hace dicha clase por ejemplo:

```
button {  
width: 100%;  
color: black;  
button error  
width: 100%;  
color: red;
```

2. El uso de id (identificadores), se van a utilizar solamente en casos en donde el componente por utilizar, no vaya a ser representado en alguna otra página, esto con el fin de lograr un grado de reutilización alto.

3. Para interpolación de strings o concatenación siempre vamos a usar interpolación:

a. Bueno:

```
const helloWorld = 'hello world Shello World);
```

b. Malo:

```
const helloWorld = "hello world" + helloWorld;
```

4. Para callbacks con funciones anónimas vamos a usar arrow

functions:

a. Bueno:

```
[1,2,3].map(x => x* 2);
```

b. Malo:

```
[1,2,3].map(function () {
```

```
return x2:
```

```
})
```

5. Mantener las funciones pequeñas y que hagan una sola tarea a la vez, por ejemplo, se ocupó hacer una suma y luego multiplicar el resultado, vamos a crear una función que haga la suma y retorne el resultado, y otra que haga la multiplicación que puede tomar ese resultado y multiplicarlo después, NUNCA una función que haga las dos operaciones al mismo tiempo.

6. Tratar de reutilizar el código lo más que se pueda usando módulos ES6.

III. JS

1. Utilizar los estándares de codificación

Mantiene el código consistente

Más fácil de leer y comprender

Más fácil de mantener

Más fácil de refactorizar

2. Evitar el uso de variables globales

Minimizar el uso de variables globales. Las variables globales son una idea terriblemente mala. El problema con las variables y funciones globales son las secuencias de comandos las que pueden sobrescribir.

3. Nombrar las variables

Intentar encontrar nombres que tengan sentido, que no sean demasiado largos. ¡Nombrar las variables en la codificación es en realidad una de las cosas más difíciles de hacer!

4. Use async-await sobre promesas

async-await porque nos resultará más fácil de leer las promesas.

5. Elegir un método y apegarnos a él

XMLHttpRequest

ir a buscar

Axios

jQuery

6. Utilizar importaciones con nombre sobre las importaciones predeterminadas cuando sea posible.

7. Utilizar las funciones de flecha ES6 siempre que sea posible

Las funciones de flecha son una sintaxis más breve para una expresión de función de escritura. Son anónimos y cambian la forma en que se enlazan las funciones.

Falla (s):

```
var multiply = function(a, b) {  
  return a* b;  
};  
  
const multiply = (a, b) => a * b;
```

functions debería ser camelCase.myFunction

Falla (s):

```
const Multiply = (a, b) => { return a * b};  
  
const MULTIPLY = (a, b) => { return a * b};  
  
const multiply = (a, b) => { return a * b};
```

Los estándares de codificación en cualquier idioma realmente pueden ayudar con la legibilidad y el mantenimiento de una aplicación. Muchos de ellos se reducen a preferencias y para repetir lo anterior, son nuestras propias opiniones. El punto principal es ser coherentes, ya que realmente

ayudan a escalar una aplicación de forma ordenada.

Una gran cosa que creo que falta en la lista anterior es cómo está estructurada la aplicación. No agregamos esto, ya que puede depender de la aplicación y si está utilizando un marco.

Otra cosa que puede resultar complicada cuando se trabaja en equipo es hacer cumplir los estándares de codificación. Aquí hay algunas ideas nuestras para ayudarnos:

- Revisiones de escritorio que revisan el código línea por línea.

- Cree una aplicación coherente en la que cada miembro sepa qué crear / actualizar.

Documento: Proyecto 1 Plantilla –Informe de diseño de software	Código: BISOFT_PL_IAS
Equipo:	Última modificación:
Revisado por:	Aprobado por:
Versión: 1.0	Página 3 de 4