

# CUARTA ENTREGA

SISTEMA DE GESTIÓN DEL  
CINEMA BLACK WOLF

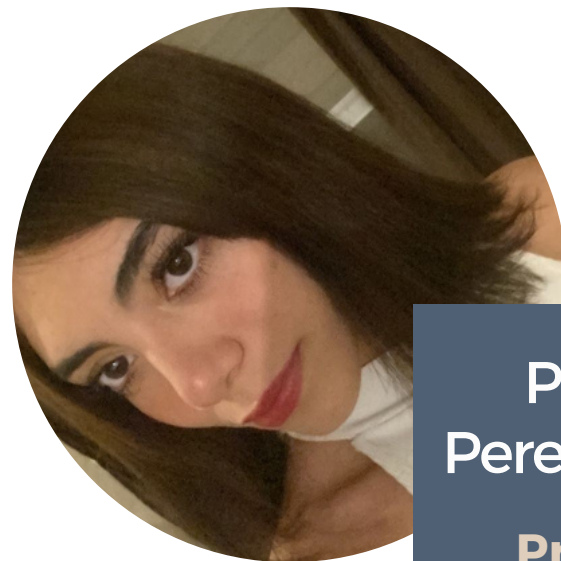


Octubre- 2023

# Nuestro Equipo



Ruth Betzabe  
Castro Acosta  
Scrum Master



Paulina María  
Perera Bastarrachea  
Project Manager



Eyli Burguete  
Pech  
Developer



Álvaro de Jesús  
Xool Canul  
Developer



Héctor Joaquín  
Pérez Can  
Developer



Esteban Vargas  
Alonso  
Developer

# Calendario de Actividades






# SPRINT 1: 23 AGOSTO - 31 AGOSTO

# AGOSTO

notas

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	DOMINGO	
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

24 DE AGOSTO: PROPUESTA DE						
IDEA DE PROYECTO ENTRE						
INTEGRANTES DEL EQUIPO,						
DEFINICION A GRANDES						
RASGOS DEL PROYECTO A						
REALIZAR						
30 DE AGOSTO:						
DEMOSTRACIÓN RÁPIDA DE LA						
IDEA GENERAL DE PROYECTO						
AL PROFESOR, LA IDEA FUE						
ACEPTADA						
31 DE AGOSTO: PRESENTACIÓN						
TENTATIVA ANTE LA CLASE DE						
LA PROBLEMÁTICA, SOLUCIÓN						
Y ORGANIZACIÓN DEL EQUIPO.						
ENTREGA DE PDF CON EL						
PRIMER AVANCE DEL						
PROYECTO						

-  Sesiones de equipo
-  Fase de revisión de requerimientos
-  Entrega de primer avance

# SPRINT 2: 1 SEPTIEMBRE - 20 SEPTIEMBRE

## SEPTIEMBRE

LUNES MARTES MIÉRCOLES JUEVES VIERNES SÁBADO DOMINGO

				● 1	2	3
4	● 5	6	7	8	● 9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	● 30	

### notas

1 DE SEPTIEMBRE: SESIÓN VIRTUAL DE EQUIPO PARA DISCUTIR SOBRE LOS COMENTARIOS RECIBIDOS EN LA PRESENTACIÓN. POSIBILIDAD DE CAMBIOS EN DETALLES DE LA ORGANIZACIÓN DE ACUERDO A LAS SUGERENCIAS GRUPALES.
PRIMERA SEMANA DE SEPTIEMBRE: ANÁLISIS SOBRE EL DISEÑO DE PROGRAMA
11 AL 18 DE SEPTIEMBRE: COMIENZO DEL DESARROLLO EN CÓDIGO
18 DE SEPTIEMBRE: REVISIÓN GENERAL ANTES DE SEGUNDA ENTREGA DE AVANCES

- Sesiones de equipo
- Fase de revisión de requerimientos
- Diseño de programa
- Fase de desarrollo
- Revisión general antes de entrega
- Entrega de segundo avance
- Refactorización de acuerdo a observaciones



# SEPTIEMBRE

LUNES MARTES MIÉRCOLES JUEVES VIERNES SÁBADO DOMINGO

				● 1	2	3
4	● 5	6	7	8	● 9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

notas

1 DE SEPTIEMBRE: SESIÓN VIRTUAL DE EQUIPO PARA DISCUTIR SOBRE LOS COMENTARIOS RECIBIDOS EN LA PRESENTACIÓN.
POSIBILIDAD DE CAMBIOS EN DETALLES DE LA ORGANIZACIÓN DE ACUERDO A LAS SUGERENCIAS GRUPALES.
PRIMERA SEMANA DE SEPTIEMBRE: ANÁLISIS SOBRE EL DISEÑO DE PROGRAMA
11 AL 18 DE SEPTIEMBRE: COMIENZO DEL DESARROLLO EN CÓDIGO
18 DE SEPTIEMBRE: REVISIÓN GENERAL ANTES DE SEGUNDA ENTREGA DE AVANCES

- Sesiones de equipo
- Fase de revisión de requerimientos
- Diseño de programa
- Revisión general antes de entrega
- Entrega de segundo avance
- Refactorización de acuerdo a observaciones

SPRINT 3: 21 SEPTIEMBRE - 26 OCTUBRE

OCTUBRE

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

notas									
TENTATIVAMENTE, EN EL MES SE TENDRÁN DIEZ SESIONES INFORMATIVAS									
DESPUÉS DE LAS SESIONES DE CLASES, SE ANALIZARÁ EL DISEÑO A FIN DE VERIFICAR QUE CUMPLA CON CIERTOS LINEAMIENTOS									
SE REALIZAN REVISIONES GENERALES ANTES DEL DÍA DE ENTREGA									

- Sesiones de equipo
- Refactorización de acuerdo a observaciones
- Desarrollo del programa
- Revisión general antes de entrega
- Entrega de tercer y cuarto avance
- Diseño de programa

notas

TENTATIVAMENTE, EN EL MES
SE TENDRÁN DIEZ SESIONES
INFORMATIVAS

DESPUÉS DE LAS SESIONES DE CLASES, SE ANALIZARÁ EL DISEÑO A FIN DE VERIFICAR QUE CUMPLA CON CIERTOS LINEAMIENTOS

[illegible]

- Sesiones de equipo
- Refactorización de acuerdo a observaciones
- Desarrollo del programa
- Revisión general antes de entrega
- Entrega de tercer y cuarto avance
- Diseño de programa



SPRINT 4: 27 OCTUBRE - 29 NOVIEMBRE

NOVIEMBRE

LUNE

		<div><div></div></div> 1	<div><div></div></div> 2	3	4	5
		<div><div></div><div></div></div>				
6	7	8	9	<div><div></div></div> 10	11	12
<div><div></div></div>				<div><div></div></div>		
<div><div></div></div> 13	14	15	16	<div><div></div></div> 17	18	19
<div><div></div></div>				<div><div></div></div>		
<div><div></div></div> 20	21	22	23	<div><div></div></div> 24	25	26
<div><div></div><div></div><div></div><div></div></div>	<div><div></div></div>	<div><div></div></div>				
<div><div></div></div> 27	28	29	30			
	<div><div></div></div>					

notas

TENTATIVAMENTE, EN EL MES SE TENDRÁN NUEVE SESIONES INFORMATIVAS											
DESPUÉS DE LAS SESIONES DE LA SEMANA, LOS VIERNES SE REALIZARÁN REUNIONES ENFOCADAS EN LA REFACTORIZACIÓN											
20 Y 21 DE NOVIEMBRE: FECHAS TENTATIVAS DE FINALIZACIÓN DE PROYECTO											
22,23,28 Y 29 DE NOVIEMBRE: DIAS DE ENTREGA FINAL DE PROYECTO											

- Sesiones de equipo
- Finalización de proyecto
- Refactorización de acuerdo a observaciones
- Desarrollo del programa
- ENTREGA FINAL DE PROYECTO

# Actividades realizadas



Borrador de diagrama de la base de datos y estándares a seguir	Alvaro
Planteamiento ADT	Alvaro, Paulina
Mockups Front end	Hector, Ruth
Borrador diagrama UML de clases principales	Paulina
Checklist variables	Alvaro, Paulina
Checklist clases	Esteban, Ruth
Checklist funciones	Eyli,Hector
Casos de uso	Esteban

Septiembre -2023

---

# Requerimientos del Proyecto

- 1 Gestión de usuarios**  
El sistema deber ser capaz de registrar, modificar y eliminar usuarios además de autenticarlos por medio de usuario y contraseña
- 3 Gestión de las funciones**  
El sistema deber ser capaz de registrar, modificar y eliminar las funciones de cada obra además no debe admitir más de una función el mismo día a la misma hora

- 2 Gestión de peliculas**  
El sistema deber ser capaz de registrar, modificar y eliminar las peliculas
- 4 Venta de boletos**  
El sistema debe ser capaz de seleccionar/deseleccionar asientos, vender los asientos seleccionados y generar un ticket con informarción detallada
- 5 Generación de reportes**  
El sistema deber ser capaz de generar un reporte de ventas diarias y mensuales

# **ADTS y abstracción**



"El sistema permite el registro de usuarios, para registrar un usuario es necesaria su información como primer nombre y apellido, un identificador para diferenciar a los usuarios, una contraseña y un correo electrónico"

Usuario

- nombre
- apellido
- Identificador de usuario
- Contraseña
- Correo electrónico

"Un administrador es un tipo de usuario que tiene más funciones, es un empleado. El administrador tiene la misma información que un usuario, excepto que se agrega su CURP pues es requisito de la empresa que el empleado se registre con esa información, además, el administrador puede modificar la información de las películas y las proyecciones"

Administrador

- nombre
- apellido
- Identificador de administrador
- Contraseña
- Correo electrónico



"Un ticket de venta es generado cuando se compran asientos para una proyección. Contiene información como su precio, el número y para que proyección se compró el ticket"

Ticket  
-Precio  
-Numero de ticket  
-Proyeccion

"En los cines existen salas, en estas, se encuentran los asientos, además, las salas deben de tener un numero e identificador para poder diferenciarlas"

Sala  
-Asientos  
-Numero de sala  
-Identificador de sala

"En la proyección se mostrarán las películas, las proyecciones deben de tener un identificador para diferenciarlas, además, las proyecciones se realizan en una fecha y hora determinada"

#### Proyeccion

- Identificador de proyeccion
- Fecha
- Hora
- Película que se proyectará

"Una película tiene un nombre, actores que participan en ella, género, a las personas les interesa de que trata la película, por lo tanto, es necesario un resumen, además, las películas tienen una duración"

#### Película

- Nombre de película
- Actores
- Género
- Resumen

-Duración

"Los asientos tienen un número, y un lugar donde estarán ubicados, los asientos se encuentran en las salas de cine"

Asientos

-numero de asiento

-fila de asiento

# Casos de uso



Diagrama de casos de usos de cine



# ADTS

operaciones

relacionadas





## User

- Log In
- Log out
- Create account
- View account information
- Modify account

## Movie

- Add movie
- Delete movie
- Search movie
- View movie information
- Modify movie

## Projection

- Create projection
- Modify projection
- Cancel projection
- View projection information



```
1
2 public class User {
3
4     private String firtsName;
5     private String lastName;
6     private String userName;
7     private String password;
8     private String email;
9
10    public User(String firtsName, String lastName, String userName, String password) {
11        this.firtsName = firtsName;
12        this.lastName = lastName;
13        this.userName = userName;
14        this.password = password;
15    }
16
17    public String getFirtsName() {
18        return firtsName;
19    }
20
21    @Override
22    public String toString() {
23        return firtsName + " " + lastName + " " + userName + " " + password + " " + email + " ";
24    }
25
```

```
33
34     public void setLastName(String lastName) {
35         this.lastName = lastName;
36     }
37
38     public String getUsername() {
39         return userName;
40     }
41
42     public void setUsername(String userName) {
43         this.userName = userName;
44     }
45
46     public String getPassword() {
47         return password;
48     }
49
50     public void setPassword(String password) {
51         this.password = password;
52     }
53
54 }
55
```

```

1 public class Admin {
2     private ArrayList<User> listaUsuarios = new ArrayList<User>();
3
4     public void userList(JTable table ){
5         File file=new File("Usuario.txt");
6         try{
7             BufferedReader br=new BufferedReader(new FileReader(file));
8             DefaultTableModel model=new DefaultTableModel(){
9                 @Override
10                 public boolean isCellEditable(int row, int column) {
11                     return false;
12                 }
13             };
14             model.addColumn("Nombre");
15             model.addColumn("Apellido");
16             model.addColumn("CURP");
17             model.addColumn("Contraseña");
18             table.setModel(model);
19             Object[] tableContent=br.lines().toArray();
20
21             for (int i = 0; i < tableContent.length; i++) {
22                 String line=tableContent[i].toString().trim();
23                 String[] datarow=line.split(" ");
24                 model.addRow(datarow);
25                 table.setModel(model);
26             }
27         }catch(Exception userListException){
28             JOptionPane.showMessageDialog(null, "Error");
29         }
30     }
31

```

```

32     public void deleteUser(JTable userTable, JTextField userName){
33         DefaultTableModel model=(DefaultTableModel)userTable.getModel();
34         for (int i = 0; i <model.getRowCount(); i++) {
35             if(((String)model.getValueAt(i, 0)).equals(userName.getText())){
36                 model.removeRow(i);
37                 break;
38             }
39         }
40         try{
41             PrintWriter escribir=new PrintWriter("Usuario.txt");
42             escribir.print("");
43             escribir.close();
44         }catch(Exception newUserioException){
45             JOptionPane.showMessageDialog(null, "ERROR");
46         }
47
48         try(BufferedWriter bufferedwriter=new BufferedWriter(new FileWriter(new File("Usuario.txt")))){
49             for (int row = 0; row <userTable.getRowCount(); row++) {
50                 StringJoiner joiner=new StringJoiner(" ");
51                 for (int column = 0; column < userTable.getColumnCount(); column++) {
52                     Object obj=userTable.getValueAt(row, column);
53                     String valor = obj ==null ? "null" : obj.toString();
54                     joiner.add(valor);
55                 }
56                 bufferedwriter.write(joiner.toString());
57                 bufferedwriter.newLine();
58             }
59         }catch(Exception writeToFileException){
60             JOptionPane.showMessageDialog(null, "ERROR");
61         }
62     }
63
64     public ArrayList<User> getUserList() {
65         return userList;
66     }
67
68     public void setUserList(ArrayList<User> listaUsuarios) {
69         this.userList = userList;
70     }
71 }
72

```

```
@Getter
@Setter
@EqualsAndHashCode
@AllArgsConstructor
public class User {

    @Id
    @Column(name="userName")
    private String userName;
    @Column(name="name")
    private String name;
    @Column(name="last_Name")
    private String lastName;
    @Column(name="email")
    private String email;
    @Column(name="password")
    private String password;
    @Column(name="role")
    private String role;
```

```
}
```

```
public class UserController {
```

```
    1 usage
```

```
    @Autowired
```

```
    private UserDao userDao;
```

```
no usages  Alvaro Xool *
```

```
    @PostMapping
```

```
    public void registerUser(@RequestBody User user) {
```

```
        Argon2 argon2 = Argon2Factory.create(Argon2Factory.Argon2Types.ARGON2id);
```

```
        String hashPassword = argon2.hash(i: 1, i1: 1024, i2: 1, user.getPassword());
```

```
        user.setPassword(hashPassword);
```

```
        userDao.register(user);
```

```
    }
```



@RestController

public class AuthenticateController {

1 usage

@Autowired

private UserDao userDao;

no usages  Alvaro Xool \*

@RequestMapping(value= "api/login", method = RequestMethod.POST)

public String login(@RequestBody User user){

String response="Fail";

if(userDao.verifyEmailAndPassword(user)){

response= "Ok";

}

return response;

}

}

```
public class UserDaoImp implements UserDao{
```

2 usages

```
@PersistenceContext
```

```
private EntityManager entityManager;
```

1 usage  Alvaro Xool

```
@Override
```

> 

```
public void register(User user) { entityManager.merge(user); }
```

1 usage  Alvaro Xool

```
@Override
```

```
public boolean verifyEmailAndPassword(User user){
```

```
    String query = "FROM User WHERE email = :email";
```

```
    List<User> lista= entityManager.createQuery(query)
```

```
        .setParameter( name: "email",user.getEmail())
```

```
        .getResultList();
```

```
    if(lista.isEmpty()){
```

```
        return false;
```

```
    }
```

```
    String passwordHasher= lista.get(0).getPassword();
```

```
    Argon2 argon2 = Argon2Factory.create(Argon2Factory.Argon2Types.ARGON2id);
```

```
    return argon2.verify(passwordHasher, user.getPassword());
```

```
}
```

Septiembre -2023

# CheckList

## Clases

No.	Consideraciones para el Nombramiento de Clases	Cumplido (Sí/No)
1	Sigue convenciones de nomenclatura consistentes	SI
2	Utiliza nombres descriptivos y sustantivos	SI
3	Utiliza PascalCase para nombres de clases	SI
4	Evita nombres demasiado largos	SI
5	Evita nombres confusos o similares	SI
6	Utiliza nombres que reflejen el propósito de la clase	SI
7	Utiliza nombres singulares y no plurales, y no verbos	SI
8	Evita acrónimos y abreviaturas confusas	SI
9	Sé consistente en la traducción de nombres si es necesario	SI
10	Evita prefijos o sufijos innecesarios	SI
11	Utiliza nombres que sean fácilmente comprensibles y memorables	SI

Septiembre -2023

# CheckList

## Clases

No.	Consideraciones para la creación de interfaces de clase	Cumplido (Sí/No)
12	La clase implementa un sólo TDA	SI
13	Prestar servicios por parejas con sus opuestos	SI
14	Sólo hay información de la clase	SI
15	Las interfaces deben ser programáticas en lugar de semánticas	SI
16	No agregar métodos públicos que sean inconsistentes con la abstracción de la interfaz	SI
17	La clase es fácil de entender y utilizar desde el exterior (abstracción)	SI
18	Mantiene una estructura interna organizada y relacionada (cohesión).	SÍ

Septiembre -2023

# CheckList

## Variables

No.	Consideración para el Nombramiento de Variables	Cumplido (Sí/No)
1	Sigue convenciones de nomenclatura consistentes	--
2	Utiliza nombres descriptivos	--
3	Evita nombres demasiado largos	Si
4	Elige un estilo de nomenclatura coherente	Si
5	Evita nombres confusos o similares	Si
6	Utiliza nombres plurales para colecciones	No
7	Prefija variables globales o de clase si es necesario	No
8	Evita acrónimos y abreviaturas confusas	Si
9	Revisa y actualiza nombres regularmente	No
10	Sé consistente en la traducción de nombres	Si
11	Utiliza nombres que eviten comentarios innecesarios	Si

Septiembre -2023

# CheckList

## Funciones

Reglas para un buen  
nombramiento de las  
funciones

No.	Consideración para el Nombramiento de	Cumplido (Sí/No)
1	Describe todo lo que la función realiza	Sí
2	No se basa en números para diferenciar las rutinas	Sí
3	Evita nombres demasiado largos	Sí
4	El nombre de la función usa en él el valor a devolver	Sí
5	Para nombrar un procedimiento, menciona un verbo descriptivo junto a un objeto.	Sí
6	Usa los opuestos de manera correcta	Sí



Septiembre -2023

# MockUps

## Inicio

BLACK WOLF CINEMA

CARTELERA

PREVENTAS

MI SESIÓN

Iniciar Sesión

Email

Contraseña

Ingresar

¿Aún no tienes cuenta?

Crear cuenta

Olvidé mi contraseña

Contenido del sitio 2023© Derechos reservados BLACK WOLF CINEMA S.S.A. de C.V.

Septiembre -2023

# MockUps

## Inciar Sesion

BLACK WOLF CINEMA

CARTELERA

PREVENTAS

MI SESIÓN

Iniciar Sesión

Email

Contraseña

Ingresar

¿Aún no tienes cuenta?

Crear cuenta

Olvidé mi contraseña

Contenido del sitio web © Derechos reservados BLACK WOLF CINEMA S.S.A. de C.V.

Septiembre -2023

# MockUps

## Inicio



CARTELERA

PREVENTAS

EN CARTELERA



LA MONJA



SONIDO DE LIBERTAD


ENTRADAS  
2X1

MIÉRCOLES Y JUEVES



10% OFF

EN DULCERIA CON  
TUS BOLETOS\*



BOLETO  
VIP 0 3D

A PRECIO NORMAL  
TODOS LOS LUNES\*



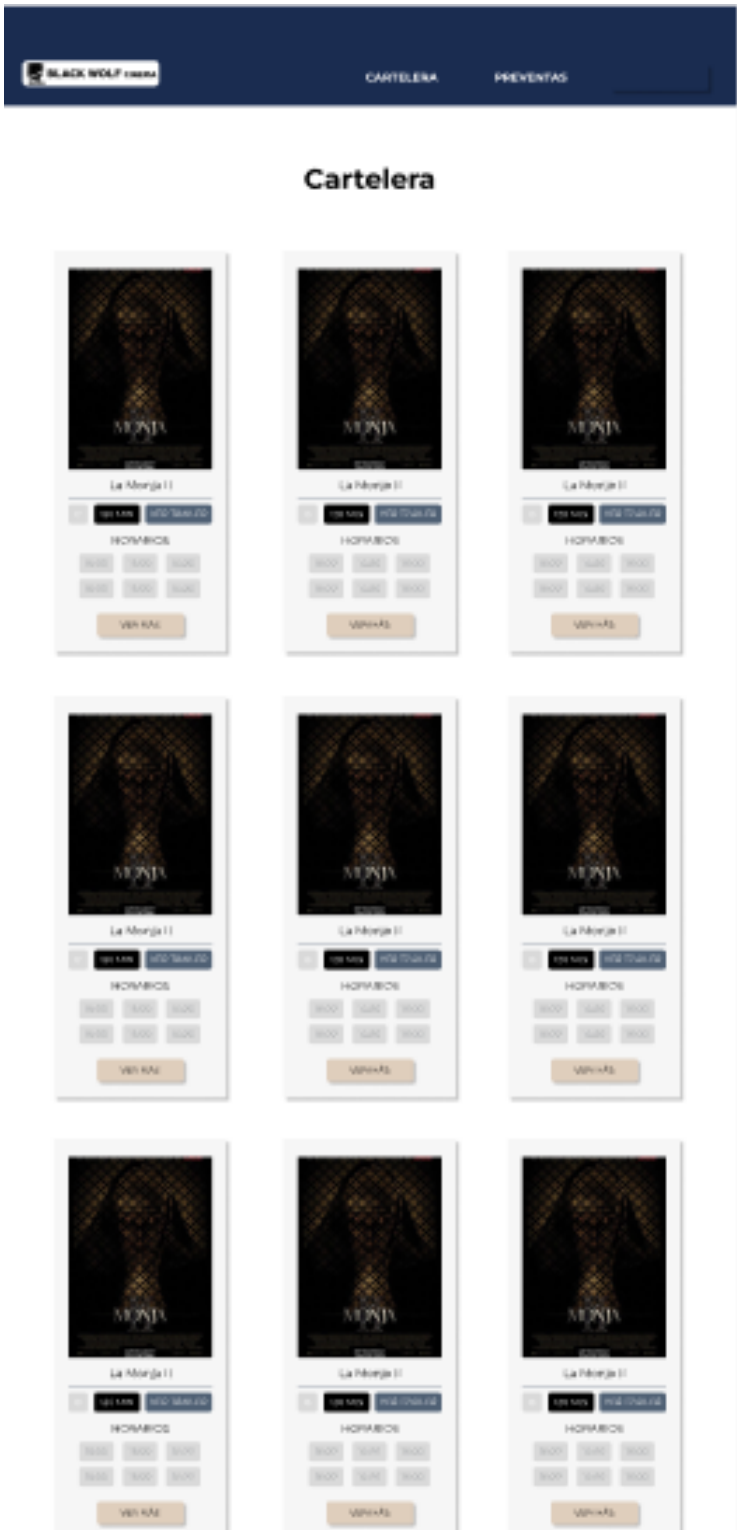
Copyright del año 2023. Dirección reservada: CINEMA AMPARADO MANDANERO S.A. de C.V.

Septiembre -2023

# MockUps

## cartelera

## usuario



Septiembre -2023

# MockUps seleccion

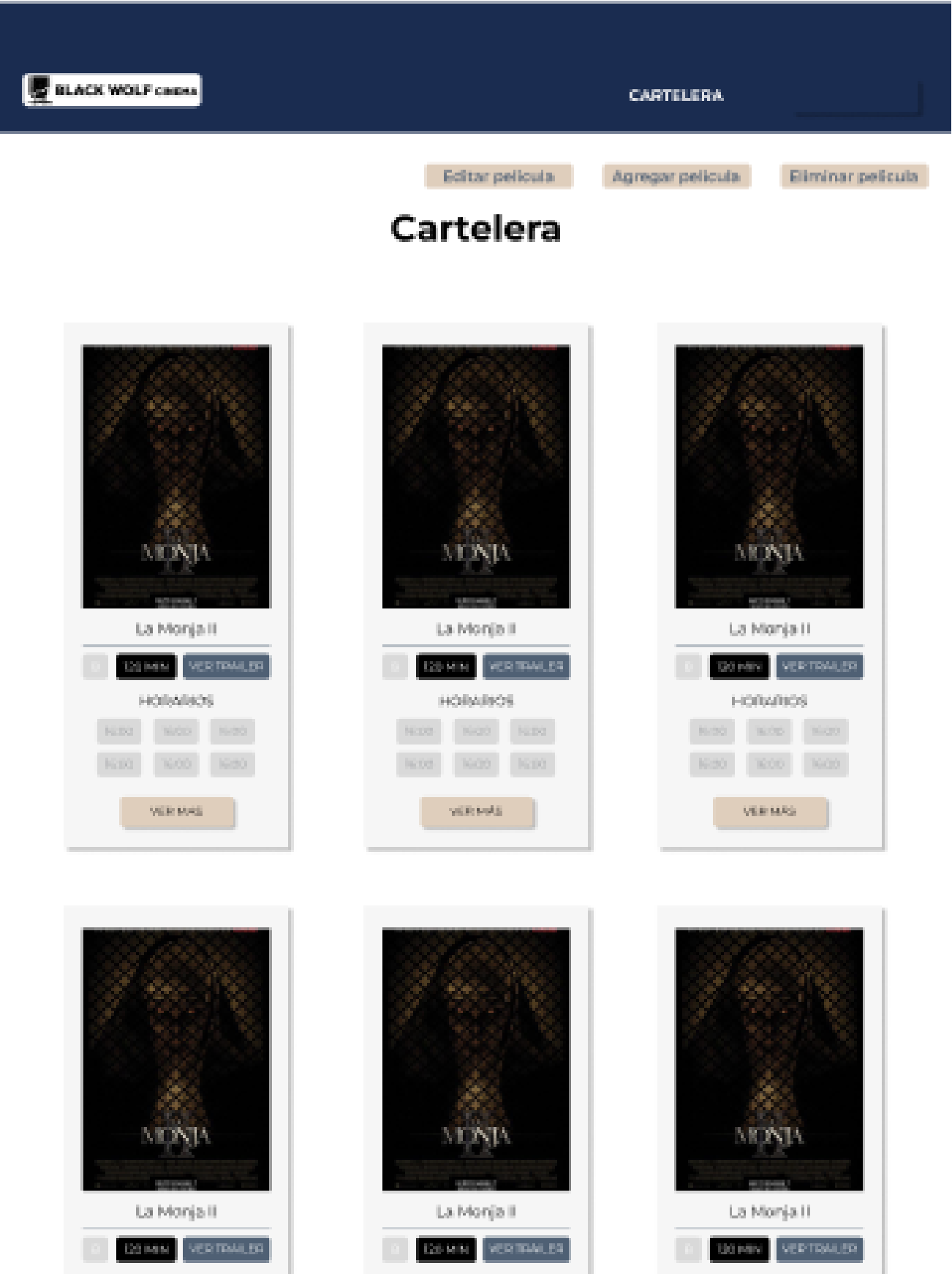


Septiembre -2023

# MockUps

## cartelera

## admin



Septiembre -2023

# MockUps

agregar, editar y  
eliminar en admin

BLACK WOLF CINEHA

Agregar nueva película a la Cartelera

Título

Clasificación

Duración

Sínpsis

Horarios

Agregar

Copyright del año 2023. Todos los derechos reservados. BLACK WOLF CINEHA S.S.A. de C.V.

BLACK WOLF CINEHA

Eliminar película a la Cartelera

Título

Eliminar

Copyright del año 2023. Todos los derechos reservados. BLACK WOLF CINEHA S.S.A. de C.V.

BLACK WOLF CINEHA

Editar nueva película a la Cartelera

Título

Clasificación

Duración

Sínpsis

Horarios

Agregar

Copyright del año 2023. Todos los derechos reservados. BLACK WOLF CINEHA S.S.A. de C.V.



**¡MUCHAS**  
**GRACIAS!**

