

Informe Individual

Ingeniería de Software

Información General

- Proyecto: Cine+
 - Autor: Manuel Antonio Vilas Valiente
 - Equipo: 5
 - Profesor: José L. Castañeda
-

Índice

Informe Individual

Ingeniería de Software

Información General

Informe

Diseño Conceptual de la Aplicación:

Implementación:

Main:

Tablas:

Ejemplo:

Informe

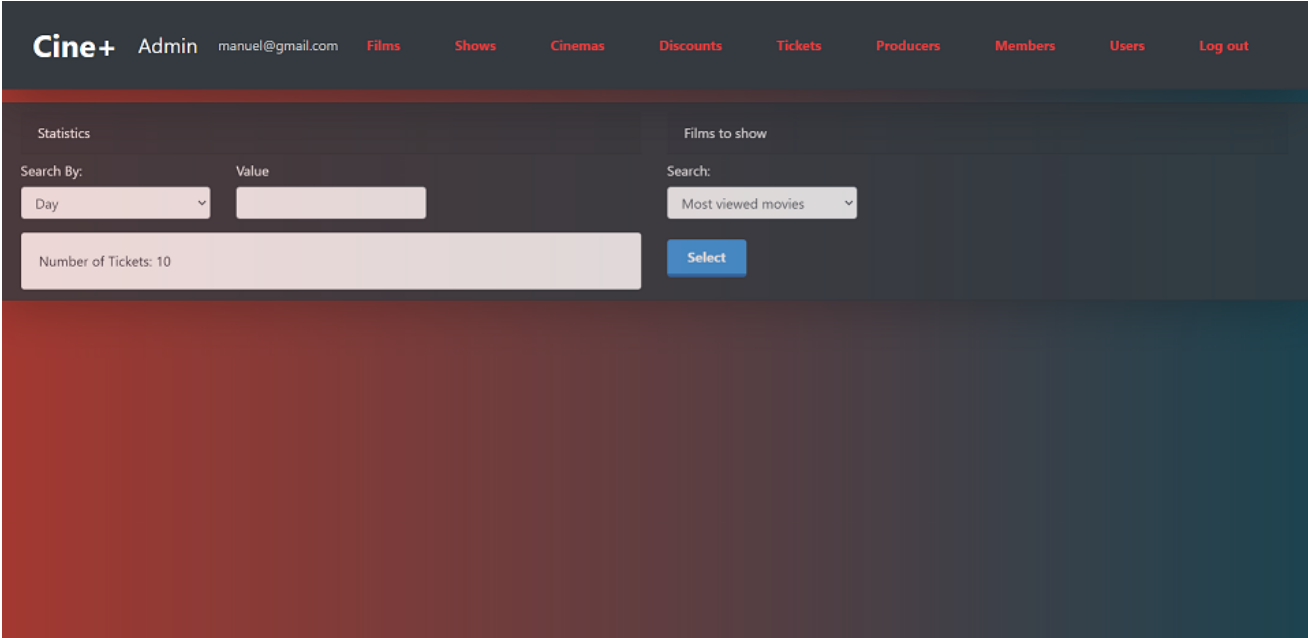
Diseño Conceptual de la Aplicación:

En el diseño de nuestra aplicación se me fue orientado por el jefe del equipo confeccionar el análisis de requerimientos, por lo cual estuve como líder de su creación, aunque como en todo el proyecto, hubo colaboración de todo nuestro equipo.

Implementación:

Main:

En la planificación realizada por el equipo, se me fue otorgada la tarea de implementar el diseño de todas las vistas a través de las cuales el gerente, o administrador, efectuaría el total de las tareas que le corresponden. Además de eso apoyé en la confección de algunos controladores de los cuales dependía para el correcto funcionamiento de dichas vistas. Este admin contará con una página principal, como la que se muestra debajo:



En esta vista, como en todas las otras se presenciara una barra de navegaci3n, la cual posee las siguientes funcionalidades:

Bot3n	Funci3n
Cine+	Este bot3n nos redirige a la p3gina principal de la aplicaci3n, al Main.
Admin	Al hacerle click nos redirige a esta misma p3gina, este est3 para cuando naveguemos por las dem3s vistas poder regresar a la principal.
Films	Este bot3n nos dirige a la vista donde se encuentra la tabla con la informaci3n de todas las pel3culas que hay en la base de datos.
Shows	Este bot3n nos dirige a la vista donde se encuentra la tabla con la informaci3n de todas las funciones que hay en la base de datos.
Cinemas	Este bot3n nos dirige a la vista donde se encuentra la tabla con la informaci3n de todas las salas que hay en la base de datos.
Discounts	Este bot3n nos dirige a la vista donde se encuentra la tabla con la informaci3n de todas las promociones que se encuentran reflejadas en la base de datos.
Tickets	Este bot3n nos dirige a la vista donde se encuentra la tabla con la informaci3n de todas las entradas que existen en la base de datos.
Producers	Este bot3n nos dirige a la vista donde se encuentra la tabla con la informaci3n de todos los realizadores que han sido agregados a la base de datos, tanto de los actores como de los directores.
Members	Este bot3n nos dirige a la vista donde se encuentra la tabla con la informaci3n de todos los usuarios asociados que hay registrados en la base de datos.

Botón	Función
Users	Este botón nos dirige a la vista donde se encuentra la tabla con la información de todos los usuarios que han sido registrados en la base de datos.
Log out	Con este botón nuestro usuario sale de la aplicación y no dirigimos a la página principal.

En el código, lo expuesto anteriormente se expresa mediante el archivo siguiente:

- `./Views/Manager/_Layout.cshtml`

```
[...]
<nav class="flex absolute top-0 items-center justify-between flex-wrap bg-dark shadow-2xl p-8
fixed w-full z-40 top-0">
  <div class="flex items-center flex-shrink-0 text-white mr-6">
    <a class="text-white no-underline hover:text-white hover:no-underline" asp-
controller="Main" asp-action="Index">
      <span class="text-4xl pl-2 font-bold"><i class="em em-grinning"></i>Cine+</span>
    </a>
    <a class="text-white no-underline hover:text-white hover:no-underline" asp-
controller="Manager" asp-action="Main">
      <span class="text-2xl pl-4">Admin</span>
    </a>
  </div>
  <div style="color: lightgray">
    @User.Identity?.Name
  </div>
</nav>

[.]

<div class="w-full flex-grow lg:flex lg:items-center lg:w-auto hidden lg:block pt-6 lg:pt-0"
id="nav-content">
  <ul class="list-reset lg:flex justify-end flex-1 items-center font-bold">
    <li class="mr-3">
      <a class="inline-block text-red-600 no-underline hover:fo hover:text-gray-200
py-2 px-4" asp-controller="Movie" asp-action="MovieList">Films</a>
    </li>
    <li class="mr-3">
      <a class="inline-block text-red-600 no-underline hover:text-gray-200 py-2 px-
4" asp-controller="Show" asp-action="ShowList">Shows</a>
    </li>
    <li class="mr-3">
      <a class="inline-block text-red-600 no-underline hover:text-gray-200 py-2
px-4" asp-controller="Cinema" asp-action="CinemaList">Cinemas</a>
    </li>
  </ul>
</div>

[.]
```

En esta vista, ya en el cuerpo de ella, se ven dos funcionalidades, a la izquierda tenemos la posibilidad de analizar las estadísticas de la venta de entradas por día, mes, año, por película, actores, género, rating, etc. En el dropdown se elige el criterio de búsqueda, en la entrada Value se debe escribir por qué valor del criterio seleccionado se debe regir la búsqueda, y por último en la carta que aparece debajo se mostrará la cantidad de entradas vendidas según los parámetros escogidos. A la derecha tenemos las opciones para seleccionar las películas que se mostrarán en la cartelera, y un botón select que es para implementar la escogida. Esto en código se evidencia en el archivo:

- `./Views/Manager/Main.cshtml`

```
[...]
<div class="form-group row">

  <div class="col-4">

    <label for="example-time-input" class="col-form-label">Search Criteria</label>

    <select id="criterion" class="form-select form-control" aria-label="Default select example">
      <option value="day">By Day</option>

      <option value="month">By Month</option>

      <option value="year">By Year</option>

      <option value="genre">By Genre</option>

      <option value="category">By Category</option>

      <option value="film">By Film</option>

      <option value="rating">By Rating</option>

    </select>
  </div>

  <div class="col-4">

    <label for="example-time-input" class="col-form-label">Value</label>

    <input class="form-control" type="text">

  </div>

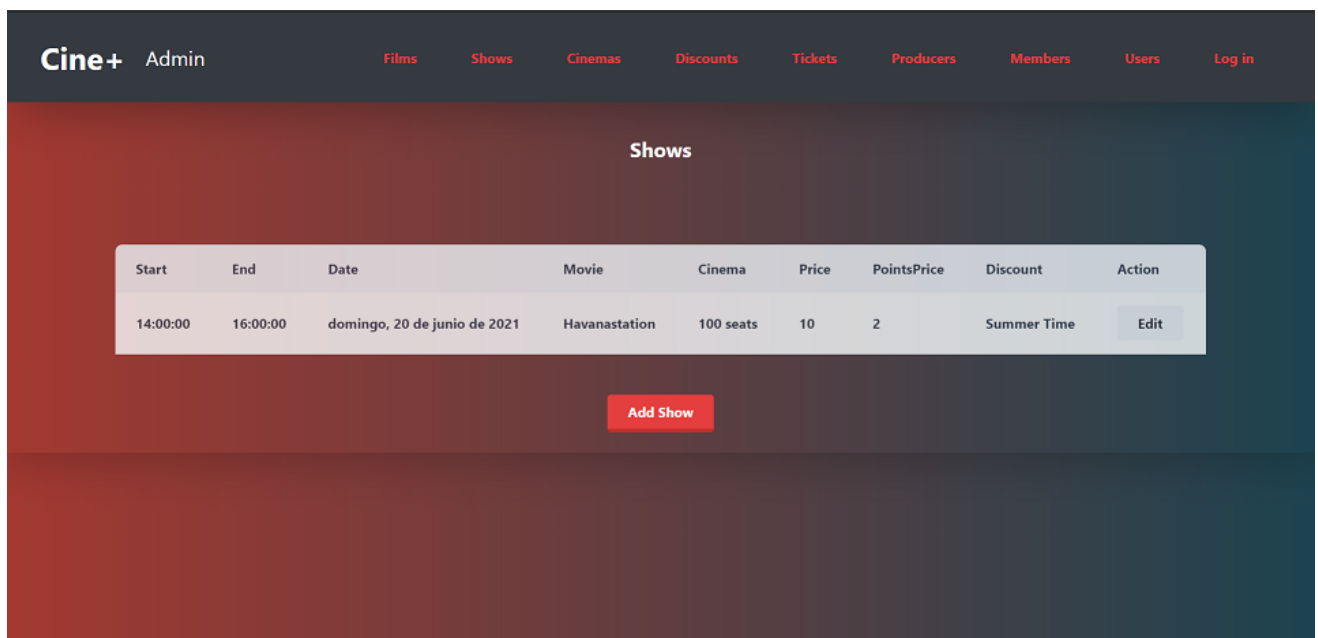
</div>
[...]
```

Tablas:

Como explicaba anteriormente en esta vista tendremos la oportunidad de apreciar toda la información que se tiene acerca de las tablas que existen en la base de datos, además de esto, podremos agregar otros elementos a cada una de ellas, mediante los botones de Add, con la posibilidad también de editar las existentes mediante los botones Edit. El código en los .cshtml de cada una de ellas es similar, cambia principalmente en dependencia de los campos que exhiba cada modelo.

Ejemplo:

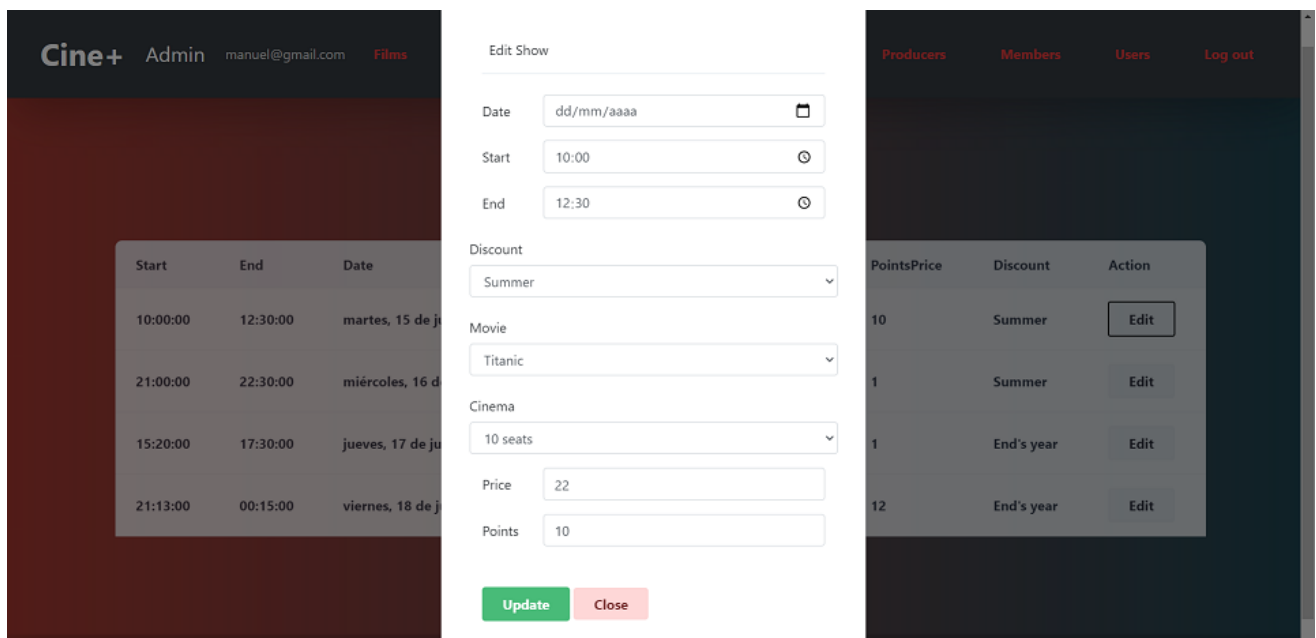
Tabla de Funciones, o Shows:



Un ejemplo en el código, basándonos en el de las funciones (shows) sería:

```
[...]
@{
    if (ViewBag.ShowsCount != 0)
    {
        foreach (var item in ViewBag.Shows)
        {
            <tr class="bg-gray-100 border-b border-gray-200">
                <th class="px-4 py-3">@item.StartTime.ToString("T")</th>
                <th class="px-4 py-3">@item.EndTime.ToString("T")</th>
                <th class="px-4 py-3">@item.Date.ToString("D")</th>
                <th class="px-4 py-3">@item.Movie.Title</th>
                <th class="px-4 py-3">@item.Cinema.NumberOfSeats</th>
                <th class="px-4 py-3">@item.Price</th>
                <th class="px-4 py-3">@item.PointsPrice</th>
                <th class="px-4 py-3">@item.Discount?.Description</th>
                <th class="px-4 py-3">
                    <div class="inline-flex">
                        <button data-modal="@item.ShowId" class="modal-trigger bg-gray-200
hover:bg-green-500 text-black-dark font-bold py-2 px-4 rounded-l">
                            Edit
                        </button>
                    </div>
                </th>
            </tr>
        }
    }
}
```

El objetivo del código de arriba es mostrar en una tabla, la cual se inicializa un poco más arriba, por cada función (show) toda la información que esta contiene, además de agregar un botón Edit el cual servirá para editar cualquiera de los campos del elemento en cuestión.



Parte del código para editar

```
[...]
<div asp-validation-summary="All">

    <form method="post" id="@item.ShowId" class="w-full" asp-controller="Show" asp-
action="Update">
        <input hidden="hidden" value="@item.ShowId" asp-for="ShowId"/>

        <div class="form-group row">

            <label for="example-date-input" class="col-2 col-form-label">Date</label>

            <div class="col-10">

                <input class="form-control" type="date" value="@item.Date.ToString("d")" asp-
for="Date">
                <span asp-validation-for="Date"></span>

            </div>

        </div>

    </div>

[...]
```

Para la edición las entradas las inicializamos con el valor que tenían antes de editar, esto facilita que si no se van a hacer cambios en algunos campos, estos no haya que sobrescribirlos. Importante en este bloque de código la siguiente línea:

```
<input hidden="hidden" value="@item.ShowId" asp-for="ShowId"/>
```

Esta es la que garantiza la edición, ya que teniendo la llave fija y oculta para evitar que se modifique, no se crea un nuevo elemento en la tabla, sino que se actualizan los demás campos únicamente. Esta línea no aparece en el formulario para crear un nuevo Show, claramente, y es la única, además de la no inicialización de las entradas con los valores existentes, que lo diferencia los formularios de agregar y editar.

Todos los archivos donde se encuentran expuestos estos códigos de mi autoría son los siguientes:

- ./Views/Cinema/CinemaList.cshtml
- ./Views/Discount/DiscountList.cshtml
- ./Views/Movie/MovieList.cshtml

- `./Views/Producer/ProducerList.cshtml`
- `./Views/Show/ShowList.cshtml`
- `./Views/TheaterMember/MemberList.cshtml`
- `./Views/TheaterUser/UserList.cshtml`
- `./Views/Ticket/TicketList.cshtml`

Para el diseño de estas vistas usé principalmente Bootstrap y Tailwind.