

13-2-2023

Manual Técnico

Gestión de cines AstroCines



DAM202@ACARBALLEIRA.LOCAL
NOEMÍ ÁLVAREZ BOUZÁN

Índice

- 1 Introducción.....2
 - 1.1 Descripción del proyecto.....2
 - 1.2 Justificación.....2
- 2 Planificación.....2
 - 2.1 Requisitos.....2
 - 2.2 Recursos.....2
 - 2.3 Planificación temporal.....3
 - 2.4 Planificación económica.....3
- 3 Tecnologías.....3
 - 3.1 Tecnologías empleadas.....3
 - 3.2 Explicación.....3
- 4 Diseño y desarrollo.....6
 - 4.1 Diseño de la aplicación.....6
 - 4.2 Pruebas de la aplicación.....12
- 5 Organización del proyecto.....12
 - 5.1 Carpetas.....13
- 6 Código: clases y código de la aplicación.....13
- 7 Conclusiones.....17
 - 7.1 Propuestas de mejora.....17

1 Introducción

1.1 Descripción del proyecto

AstroCines es una aplicación de escritorio cuya finalidad es la administración de Cines en Centros Comerciales. Gestiona el personal de los diferentes cines y administra las ventas de entradas. Su interfaz es práctica y el diseño es fresco.

1.2 Justificación

Me interesó la propuesta de proyecto gestión y control de salas de cine porque siento pasión por el mundo cinematográfico. Además he trabajado muchos años en un cine, en las diferentes áreas: taquilla, tiendas, salas...

2 Planificación

2.1 Requisitos

2.1.1 Requisitos funcionales

Aquí se van a plantear todas las funciones que la aplicación va a poder desarrollar:

- Login
- Administrar cines, emplead@s.
- Navegación entre páginas.
- Exportar informes a un documento pdf.

2.1.2 Requisitos no funcionales

- Intuitivo y rápido
- Sencillo y visual (Control de errores)
- Adaptable al tamaño de la pantalla

2.2 Recursos

Aplicación basada en WPF (Presentation Foundation), compatible con Windows 10 y versiones superiores.

El programa no necesita ningún programa externo pre-instalado, todo está incluido en el instalador. La base de datos se genera automáticamente en un servidor SQL Server (Es necesario tener SQL Server Management) el usuario y contraseña se vinculan con el usuario utilizado en Windows (WindowsAuthentication).

2.3 Planificación temporal

• Bocetos	10h
• Desarrollo tablas	30h
• Desarrollo de la interfaz	40h
• Programación frontend	42h
• Programación backend	42h
• Integración front-back	18h
• Documentación del proyecto	20h
	Total 202h
	<small>*Tiempo aproximado</small>

2.4 Planificación económica

En esta tabla se describen todos los gastos que van a surgir con el desarrollo de la aplicación.

- En este caso el gasto de material es nulo porque trabajé con el ordenador de clase.

3 Tecnologías

3.1 Tecnologías empleadas

Frontend

- Xaml
- Material Design Themes

Backend

- C#
- Sql Server (SQLEXPRESS)
- Microsoft.EntityFrameworkCore
- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools
- iTextSharp

3.2 Explicación

3.2.1 Xaml



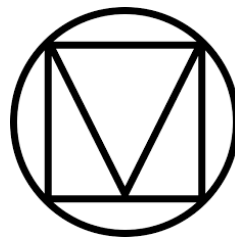
- XAML (eXtensible Application Markup Language) es el lenguaje de formato para la interfaz de usuario para la Base de Presentacion de Windows (WPF por sus siglas en ingles) y Silverlight(wpf/e), el cual es uno de los "pilares" de la interfaz de programación de aplicaciones .NET en su version 3.0 (conocida con anterioridad con el nombre clave WinFX).

- XAML es un lenguaje declarativo basado en XML, optimizado para describir gráficamente interfaces de usuario visuales ricas desde el punto de vista gráfico, tales como las creadas por medio de Adobe Flash.

XUL y UIML son otros ejemplos de lenguajes de interfaz basados en XML.

- SVG es un estándar de la organización W3C, el cual soporta gráficos, animaciones, audio y video integrados, eventos y comportamiento descrito por medio de escritura y puede ser utilizado como lenguaje de interfaz basado en XML.

3.2.2 Material Design



- Es una herramienta de código abierto y una de las bibliotecas GUI más populares para WPF, la biblioteca también es compatible con MahApps y Dragablz.

- Se ha utilizado para mejorar la interfaz, así como para utilizar iconos y demás complementos como las notificaciones

3.2.3 C#



- C#" es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.
- Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.
- Aunque C# forma parte de la plataforma .NET, esta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows Microsoft, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

3.2.4 iTextsharp



- iText es una biblioteca PDF que permite a los desarrolladores crear, adaptar, inspeccionar y mantener documentos en el formato de documento portátil (PDF). Al utilizar iText, los desarrolladores pueden generar documentos e informes basados en datos de un archivo XML o una base de datos.

3.2.5 SQL Server Management



- SQL Server Management Studio (SSMS para abreviar) es un entorno de desarrollo integrado para administrar cualquier infraestructura SQL. Se utiliza para acceder, administrar, configurar y desarrollar todos los componentes de SQL Server y SQL Database. Microsoft lo ha optimizado a lo largo de los años y es un programa de administración de servidores y bases de datos muy popular.

3.2.6 SQL Server (SqlExpress)



- Es una versión "ligera" de Microsoft SQL server. Ésta sirve para uso libre y distribuible a los desarrolladores de software.
- Es el motor de la base de datos utilizada para la aplicación.
- Nos logueamos con un login con autenticación de windows, para entrar en la BD.

4 Diseño y desarrollo

4.1 Diseño de la aplicación

4.1.1 Diseño de interfaces

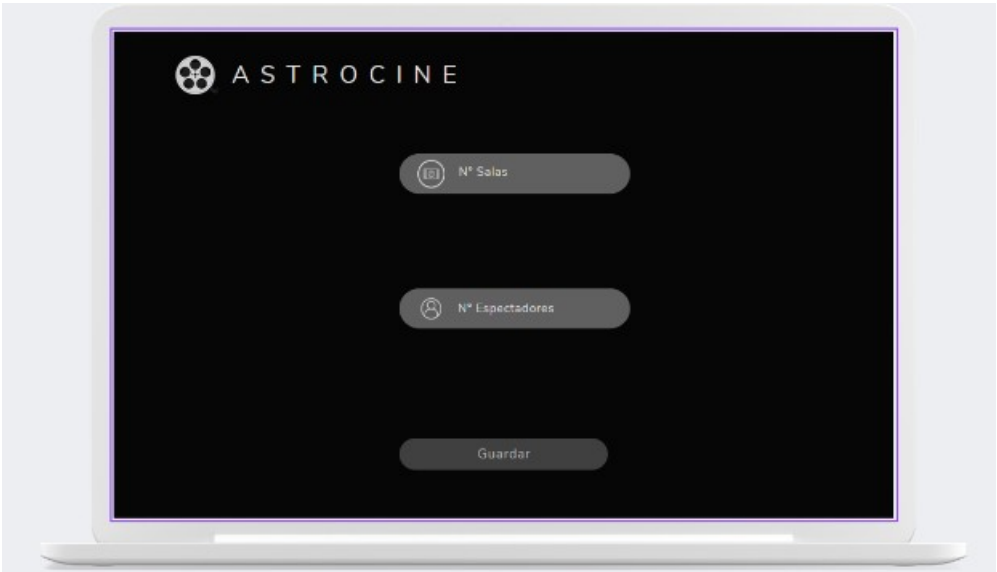
Al diseñar las interfaces he contemplado que fuera lo más intuitivo posible, adaptable y legible.

Mi primer boceto no tiene mucho que ver con el resultado final. Para realizar los bocetos utilicé Canva muy intuitivo.

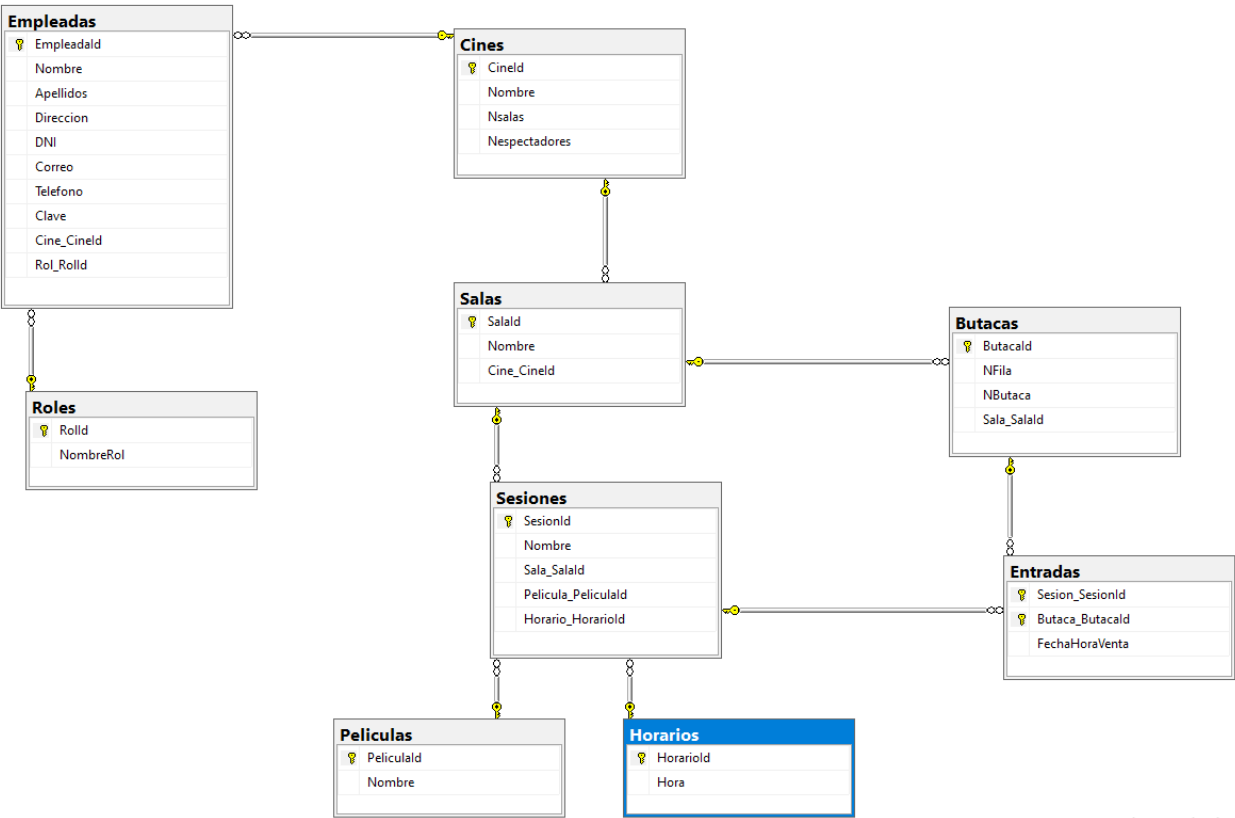
https://www.canva.com/es_es/

- **Bocetos**



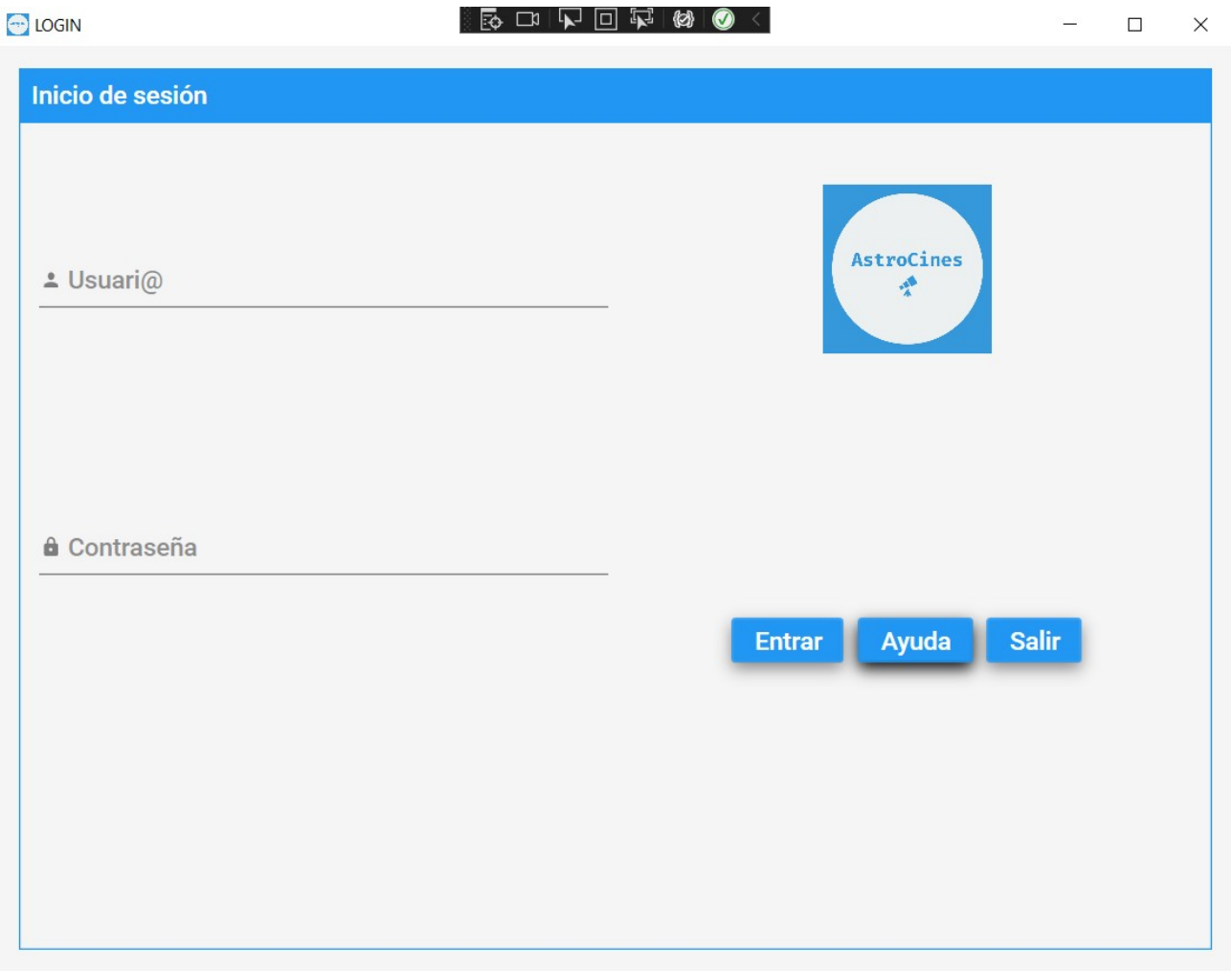


• Diagrama de ER.

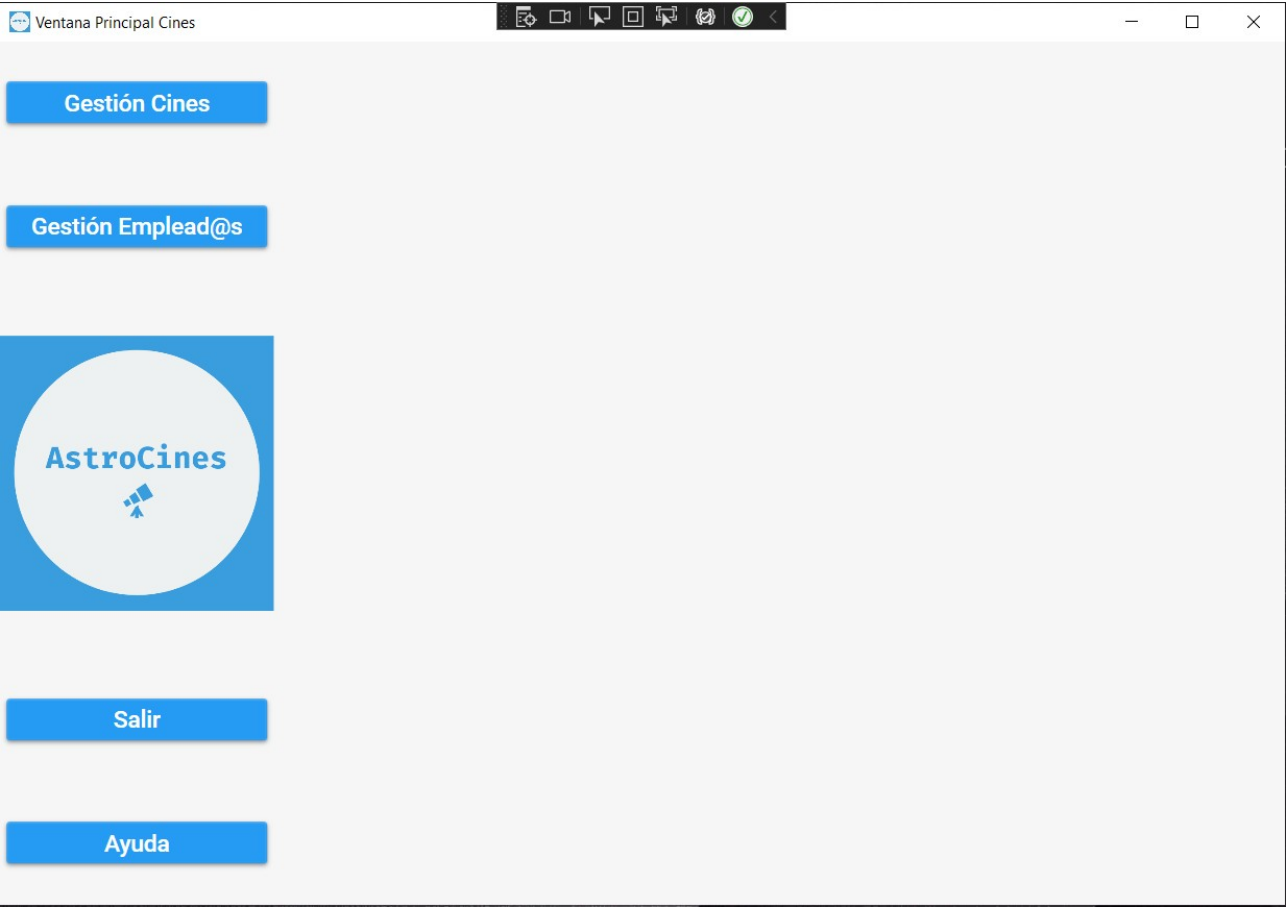


- Login para personal de taquilla

Usuari@: noemi@gmail.com y password: abc123.



- **Ventana Principal con marco para navegar por las diferentes páginas**



- **Ventana Principal al pulsar en el botón Gestión Cines**

Gestión Cines

Gestión Empleado@s

AstroCines

Salir

Ayuda

Datos de cine

Nombre

Nº Espectadores

0

Nº Salas

0

Generar

ID	Nombre	Nº Salas	Nº Espectadores
1	AstroCines	8	800
2	TeleCines	5	500

Eliminar

Guardar

Cancelar

- **Ventana Principal al pulsar en el botón Gestión Emplead@s**

Gestión Cines

Gestión Emplead@s

AstroCines

Salir

Ayuda

Datos de emplead@

Nombre

Noemí

DNI

44458565W

Apellidos

Álvarez Bouzán

Correo electrónico

noemi@gmail.com

Dirección

Teléfono

Rol

Encargada

Cine

AstroCines

Contraseña

●●●●●●

Repite la contraseña

●●●●●●

✓

✖

🔄

📄

ID	Nombre	Apellidos	Dirección	DNI	Correo	Clave	Teléfono	Rol	Cine
1	Noemí	Álvarez Bouzán		44458565W	noemi@gmail.com	abc123.		3	1
2	María	Pérez Sánchez		11111111A	maria@gmail.com	abc123.		3	2

4.1.2 Colores

He utilizado azul porque se puede vincular a productos de alta precisión, o bien con tecnología y ciencia. También he utilizado colores muy claros o casi blancos para dar contraste a la aplicación.

Fondo

- WhiteSmoke

Tono azul

- #0277BD

Botones

- Letra: #FFF
- Fondo: #0277BD

4.1.3 Fuente

- Como fuente he utilizado una que viene con material design, llamada Roboto
- Los títulos utilizan un FontSize de 18
- El que pongo a todas las fuentes utilizan un FontSize de 18 por defecto.
- El tipo de fuente de los informes es Calibri, su FontSize es de 12.

4.1.4 Imágenes e iconos

Logo de AstroCines e icono creado en shopify

<https://www.shopify.com/es/herramientas/generador-de-logos/onboarding/customize-logo>



4.2 Pruebas de la aplicación

La aplicación ha sido sometida a un exhausto examen de pruebas, aun así, es posible que el usuario pueda encontrar algún tipo de fallo y es importante realizar una buena gestión de errores, dando feedback directo al usuario respecto a que ha sucedido y por qué, si es información de la que debería disponer.

Teniendo en cuenta lo mencionado anteriormente nos centraremos en las partes de la aplicación en las que hay algún tipo de procesamiento de datos o algo similar, ya que sin duda es la parte más importante del programa para que funcione correctamente y muestre los resultados que debe.

5 Organización del proyecto

5.1 Carpetas

PRACTICA_FINAL > CINEMA_ORM > CINEMA_ORM					Buscar en CINEMA_ORM	
	Nombre	Fecha de modificación	Tipo	Tamaño		
	bin	04/03/2023 15:57	Carpeta de archivos			
	DAL	04/03/2023 17:55	Carpeta de archivos			
	Imagenes	04/03/2023 15:57	Carpeta de archivos			
	Modelo	04/03/2023 15:57	Carpeta de archivos			
	obj	04/03/2023 17:55	Carpeta de archivos			
	App	07/02/2023 10:48	Archivo de marca...	2 KB		
	App.xaml.cs	03/02/2023 13:44	C# Source File	1 KB		
	AssemblyInfo.cs	03/02/2023 13:44	C# Source File	1 KB		
	CINEMA_ORM	01/03/2023 10:51	C# Project file	2 KB		
	CINEMA_ORM.csproj.user	13/02/2023 10:37	Per-User Project O...	2 KB		
	GestionCines	03/03/2023 13:16	Archivo de marca...	7 KB		
	GestionCines.xaml.cs	03/03/2023 13:27	C# Source File	9 KB		
	GestionEmpleadas	03/03/2023 14:04	Archivo de marca...	10 KB		
	GestionEmpleadas.xaml.cs	03/03/2023 13:47	C# Source File	6 KB		
	LogoAstrocines3	25/02/2023 17:29	Archivo PNG	17 KB		
	MainWindow	26/02/2023 0:19	Archivo de marca...	7 KB		
	MainWindow.xaml.cs	01/03/2023 9:00	C# Source File	3 KB		
	pdf	10/02/2023 13:39	Archivo PNG	11 KB		
	Registrar	25/02/2023 17:58	Archivo de marca...	5 KB		
	Registrar.xaml.cs	25/02/2023 15:47	C# Source File	2 KB		
	VentanaPrincipal	03/03/2023 13:55	Archivo de marca...	5 KB		
	VentanaPrincipal.xaml.cs	03/03/2023 13:02	C# Source File	2 KB		
	Wpdf	25/02/2023 17:58	Archivo de marca...	1 KB		
	Wpdf.xaml.cs	10/02/2023 13:42	C# Source File	1 KB		

6 Código: clases y código de la aplicación

- Login para personal de taquilla:

Cuando pulsamos en entrar, validamos que los campos no estén vacíos y que el correo y la contraseña pertenezcan a una empleada.

```

1 referencia
private void BT_ENTRAR_MouseDown(object sender, RoutedEventArgs e)
{
    //Validamos que los campos no estén vacíos
    if (TB_USER.Text.Length == 0)
    {
        TB_ERROR.Text = "Introduce un/a usuari@. Por favor.";
        TB_USER.Focus();
    }
    else if (TB_USER.Text.Length < 2)
    {
        TB_ERROR.Text = "Introduce un/a usuari@ válid@";
        TB_USER.Select(0, TB_USER.Text.Length);
        TB_USER.Focus();
    }
    else if (PB_PASS.Password.Length == 0)
    {
        TB_ERROR.Text = "Introduce una contraseña. Por favor.";
        PB_PASS.Focus();
    }
    else if (PB_PASS.Password.Length < 5)
    {
        TB_ERROR.Text = "Por favor introduce un password válido.";
        PB_PASS.Focus();
    }
    //Entramos en la ventana principal de la app si la empleada es distinta de null
    else
    {
        empleadx = bd.EmpleadaRepository.EmpleadaCompleta(TB_USER.Text, PB_PASS.Password.ToString());
        if(empleadx != null)
        {
            VentanaPrincipal principal = new VentanaPrincipal(empleadx);
            principal.Show();
            this.Close();
        }
        TB_ERROR.Text = "Por favor debes introducir un correo y pass válidos";
    }
}

```

- Gestión de cines:

-Cuando pulsamos el botón generar llama a este método dónde se crea un pdf con los datos introducidos en el formulario.

```

//Método que genera informe al hacer click en el botón generar
1 referencia
private void BT_GENERAR_INFORME_Click(object sender, RoutedEventArgs e)
{
    Document doc = new Document();
    if (doc != null)
    {
        PdfWriter writer = PdfWriter.GetInstance(doc, new FileStream("InformeCines.pdf",
            FileMode.Create));
        //Agregamos metadatos
        doc.AddAuthor("Noemí Álvarez Bouzán");
        doc.AddCreator("Creación de PDF usando iTextSharp");
        doc.AddKeywords("PDF tutorial");
        doc.AddTitle("Informe Cines - Creación de PDF usando iTextSharp");
        // Abrimos el documento
        doc.Open();
        // Agregamos un párrafo al documento
        doc.Add(new Paragraph("Creación de PDF usando iTextSharp " +
            " " +
            " "));
        doc.Add(new Paragraph(""));

        var tab = new PdfPTable(3) { WidthPercentage = 100 };
        // Esta es la primera fila
        tab.AddCell("Nombre");
        tab.AddCell("Nsalas");
        tab.AddCell("Nespectadores");

        // Segunda fila
        tab.AddCell(TB_NOMBRE.Text);
        tab.AddCell(TB_SALAS.Text);
        tab.AddCell(TB_ESPECTADORES.Text);

        doc.Add(tab);
        // Cerramos documento
        doc.Close();
        //Cerramos la instancia del escritor de pdf
        writer.Close();
        //Mostramos el informe con el visualizador
        new Wpdf(new Uri(Environment.CurrentDirectory + "\\InformeCines.pdf")).Show();
    }
}

```

-Al pulsar en guardar llama a este método dónde dependiendo de una booleana nuevo añade o modifica.

```

1 referencia
private void BT_REGISTRAR_Click(object sender, RoutedEventArgs e)
{
    string errores = Validacion.errores(cine);

    if (errores.Equals(""))
    {
        if (nuevo)//Añadir
        {
            VentanaPrincipal.bd.CineRepository.Añadir(cine);
            VentanaPrincipal.bd.Save();
            MessageBox.Show("Cine insertado correctamente",
                "Gestión de cines", MessageBoxButton.OK, MessageBoxImage.Information);
            DG_CINES.ItemsSource = VentanaPrincipal.bd.CineRepository.GetAll();
        }
        else//Modificar
        {
            VentanaPrincipal.bd.CineRepository.Update(cine);
            VentanaPrincipal.bd.Save();
            MessageBox.Show("Cine modificado correctamente",
                "Gestión de cines", MessageBoxButton.OK, MessageBoxImage.Information);
            DG_CINES.ItemsSource = VentanaPrincipal.bd.CineRepository.GetAll();
        }
        Limpiar();
    }
    else MessageBox.Show(errores);
}

```


• Gestión de empleadas:

-Al pulsar en el botón guardar emplead@ llama a este método:

```
//Método que guarda empleada dependiendo de la booleana
1 referencia
private void BT_GUARDAR_Click(object sender, RoutedEventArgs e)
{
    string errores = Validacion.errores(empleadx);

    if (errores.Equals(""))
    {
        if (nuevo) // Añadir
        {
            //Si los dos campos coinciden engadimos
            if (PB_PASS.Password.ToString() == PB_PASS_REP.Password.ToString())
            {
                empleadx.Clave = PB_PASS.Password;
                VentanaPrincipal.bd.EmpleadaRepository.Añadir(empleadx);
                VentanaPrincipal.bd.Save();
                MessageBox.Show("Registrad@ correctamente",
                    "Gestión de empleadas", MessageBoxButton.OK, MessageBoxImage.Information);
                DG_EMPLEADX.ItemsSource = VentanaPrincipal.bd.EmpleadaRepository.GetAll();
            }
            // Si no mensaje de error
            else
            {
                MessageBox.Show("La contraseña debe ser la misma",
                    "Gestión de empleadas", MessageBoxButton.OK, MessageBoxImage.Error);
                PB_PASS_REP.Focus();
            }
        }
        else // Modificar
        {
            //Si los dos campos coinciden modificamos
            if (PB_PASS.Password.ToString() == PB_PASS_REP.Password.ToString())
            {
                empleadx.Clave = PB_PASS.Password;
                VentanaPrincipal.bd.EmpleadaRepository.Update(empleadx);
                VentanaPrincipal.bd.Save();
                MessageBox.Show("Modificad@ correctamente",
                    "Gestión de empleadas", MessageBoxButton.OK, MessageBoxImage.Information);
                DG_EMPLEADX.ItemsSource = VentanaPrincipal.bd.EmpleadaRepository.GetAll();
            }
            // Si no mensaje de error
            else
            {
                MessageBox.Show("La contraseña debe ser la misma",
                    "Gestión de empleadas", MessageBoxButton.OK, MessageBoxImage.Error);
                PB_PASS_REP.Focus();
            }
        }
        //BT_LIMPIAR_Click(sender, e);
    }
    else MessageBox.Show(errores);
}
```

7 Conclusiones

La aplicación está pensada para el personal del cine, sobre todo para taquilla y personal encargado de administrar dichos cines.

7.1 Propuestas de mejora

En las siguientes versiones de la aplicación vamos a implementar:

- La gestión de las salas, su distribución en butacas para vender dicha entrada.
- Asignación de las películas a las diferentes sesiones.
- Hacer informe de cada entrada vendida que indica la sala, película y butaca asignada .
- Dependiendo de la sesión mostraremos una distribución o otra.

Si me sobra tiempo :

- Diferentes informes como ventas por sala, por película, por horario, recaudación,