

*Manual del usuario.*  
**Entrega TFG 15-06-2022**

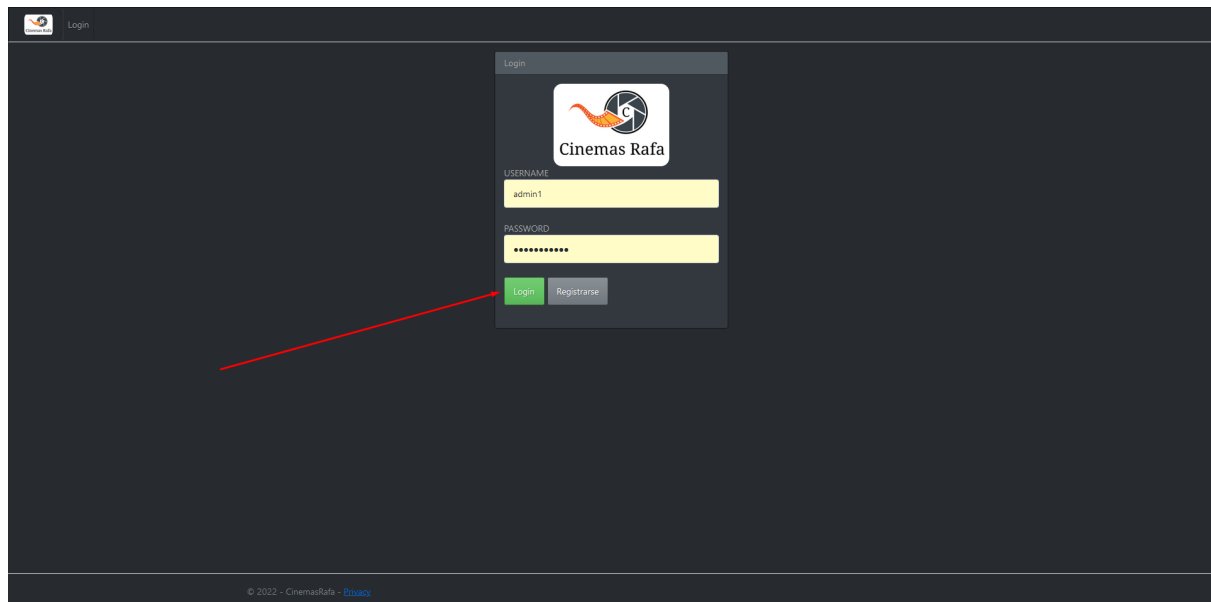
CinemasRafa



*Rafael Prieto Alvarez*

## Login

Al entrar en la aplicación lo primero que vemos es esta pantalla.



La aplicación se inicia, hace la conexión con la base de datos, añade la sesión para así controlar no poder acceder desde distintos navegadores sin relogearse y controla los posibles filtros que se pueden usar.

```
- referencias
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();

    services.AddDbContext<ControlContext>(options =>
    {
        options.UseMySQL(Configuration.GetConnectionString("miConexion"),
            ServerVersion.AutoDetect(Configuration.GetConnectionString("miConexion")),
            builder =>
            {
                builder.EnableRetryOnFailure(5, TimeSpan.FromSeconds(10), null);
            });
    });

    services.AddSession(options =>
    {
        options.Cookie.HttpOnly = true;
        options.Cookie.IsEssential = true;
    });

    services.AddScoped<AdminFilter>();
    services.AddScoped<WorkerFilter>();
    services.AddScoped<SessionFilter>();
}
```

LoginController es el controlador encargado de controlar que es las posibles cosas que pasan en la vista login y el logout.

En el login se comprueban los datos introducidos para entrar se controlan las posibles excepciones y si es correcto se guarda la sesión y se redirige a la vista home.

```
0 referencias
public IActionResult Login(string username, string password)
{
    if (password != null)
    {
        string encryptedPass = Utils.Encrypt(password);
        var usuario = _context.CuentaUsuario.Where(u => u.Username == username && u.Password == encryptedPass).FirstOrDefault();

        if (usuario != null)
        {
            if(usuario.Active == false)
            {
                TempData["ErrorMessage"] = "Su cuenta ha sido suspendida.";
                return RedirectToAction(nameof(Index));
            }
            HttpContext.Session.SetString("usuario", usuario.ID.ToString());
            HttpContext.Session.SetString("username", usuario.Username.ToString());

            var rol = _context.Role.Where(r => r.ID.Equals(usuario.RoleID)).FirstOrDefault();
            HttpContext.Session.SetString("rol", rol.RoleName.ToString());

            Utils.Menus = _context.RoleHasMenu.Include(m => m.Menu).Where(m => m.RoleID == usuario.RoleID).Select(m => m.Menu).ToList();
            string labels = string.Empty;
            string controllers = string.Empty;
            string actions = string.Empty;
            foreach(Menu menu in Utils.Menus)
            {
                labels += menu.Label + ",";
                controllers += menu.Controller + ",";
                actions += menu.Action + ",";
            }
            HttpContext.Session.SetString("labels", labels);
            HttpContext.Session.SetString("controllers", controllers);
            HttpContext.Session.SetString("actions", actions);
        }
        else
        {
            TempData["ErrorMessage"] = "Error en login. Error en usuario o contraseña.";
            return RedirectToAction(nameof(Index));
        }
    }
    else
    {
        TempData["ErrorMessage"] = "Debe rellenar los campos para continuar.";
        return RedirectToAction(nameof(Index));
    }

    return RedirectToAction("Index", "Home");
}
```

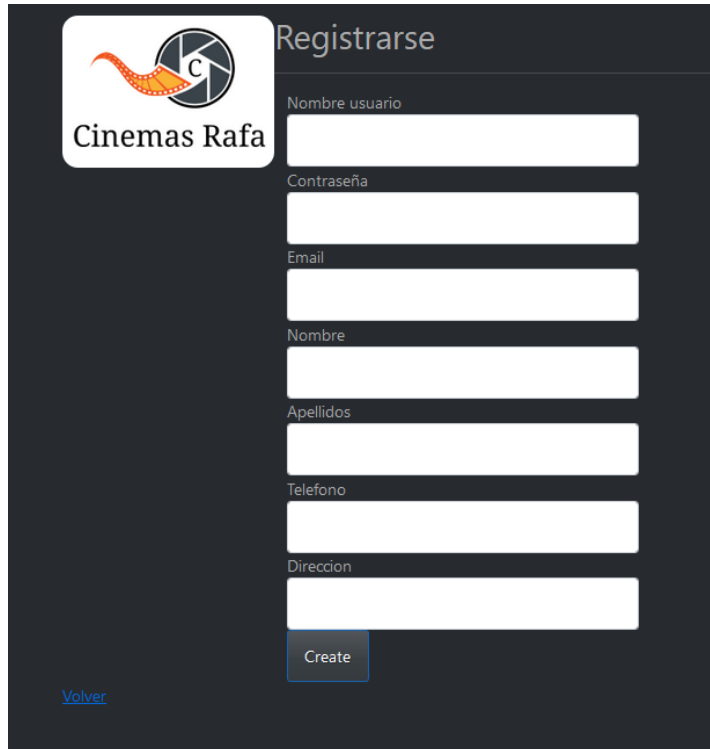
Logout - Elimina la sesión en la que se está logueado.

```
0 referencias
public IActionResult Logout()
{
    HttpContext.Session.Clear();

    //Session.Clear

    Utils.Menus = new List<Menu>();
    return RedirectToAction("Index", "Login");
}
```

## Register



The screenshot shows a registration form titled 'Registrar' for 'Cinemas Rafa'. The form is set against a dark background. It includes input fields for 'Nombre usuario', 'Contraseña', 'Email', 'Nombre', 'Apellidos', 'Telefono', and 'Direccion'. A 'Create' button is located at the bottom right of the form. A link labeled 'Volver' is positioned at the bottom left. The 'Cinemas Rafa' logo, featuring a stylized camera lens and film strip, is in the top left corner.

En el registro a través del controlador `customerDtoController` es capaz de crear un nuevo customer con todos los datos necesarios.

```
[HttpPost]
[ValidateAntiForgeryToken]
0 referencias
public async Task<IActionResult> Create([Bind("ID,Username,Password,Email,Active,Name,Surname,Phone,Address")] CustomerDto CustomerDto)
{
    Role role = _context.Role.Where(r => r.RoleName == "Customer").FirstOrDefault();

    if (ModelState.IsValid)
    {
        Customer Customer = new Customer();
        CuentaUsuario cuenta = new CuentaUsuario();

        cuenta.Username = CustomerDto.Username;
        cuenta.Password = Utils.Encrypt(CustomerDto.Password);
        cuenta.Email = CustomerDto.Email;
        cuenta.RoleID = role.ID;
        cuenta.Active = CustomerDto.Active;

        _context.Add(cuenta);
        await _context.SaveChangesAsync();

        Customer.Name = CustomerDto.Name;
        Customer.Surname = CustomerDto.Surname;
        Customer.Phone = CustomerDto.Phone;
        Customer.Address = CustomerDto.Address;
        Customer.CuentaUsuarioID = cuenta.ID;
        _context.Add(Customer);

        await _context.SaveChangesAsync();
        return RedirectToAction("Index", "Login");
    }

    return View(CustomerDto);
}
```

## Vista Home



Cuando se inicia la sesión en función del rol que tenga la cuenta con la que entramos encontraremos un menú u otro.

Esto es posible gracias a la vista `_layout` la cual crea el menú dinámicamente en función del rol.

```
<header>
  <nav class="navbar-expand-sm navbar-togglerable-sm navbar-dark border-bottom box-shadow mb-3">
    <div class="container-fluid">
      </img>
      <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
        <ul class="navbar-nav flex-grow-1">
          @if (Context.Session.GetString("Labels") != null && Context.Session.GetString("controllers") != null && Context.Session.GetString("actions") != null)
          {
            string[] controllers = Context.Session.GetString("controllers").Split(',');
            string[] labels = Context.Session.GetString("Labels").Split(',');
            string[] actions = Context.Session.GetString("actions").Split(',');
            @for (int i = 0; i < controllers.Count(); i++)
            {
              if( i < controllers.Count() - 1) {
                <li class="nav-item">
                  <a class="nav-link" " asp-area="" asp-controller="@controllers[i]" asp-action="@actions[i]">@labels[i]</a>
                </li>
              }
            }

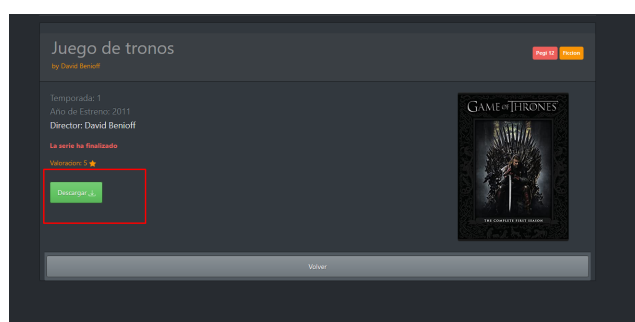
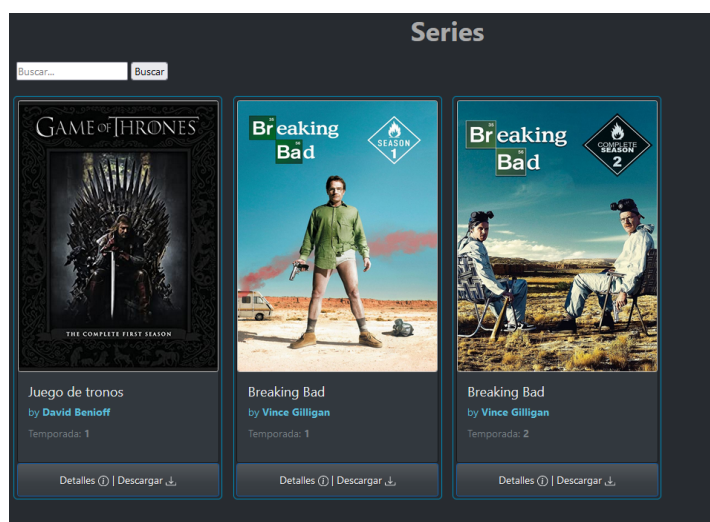
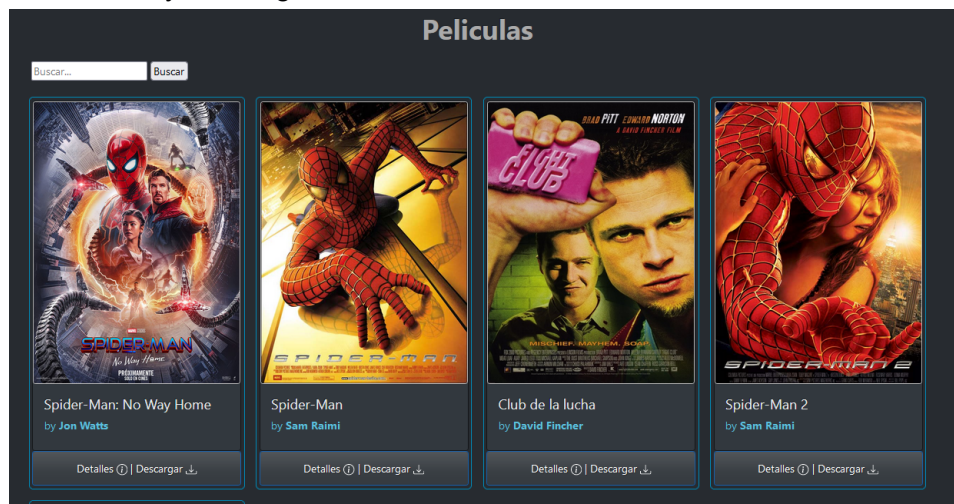
            <li class="nav-item ms-auto d-flex flex-row">
              <a class="text-light btn btn-danger" asp-area="" asp-controller="Login" asp-action="Logout">Logout</a>
            </li>
          }
          else
          {
            <li class="nav-item">
              <a class="nav-link" " asp-area="" asp-controller="Login" asp-action="Index">Login</a>
            </li>
          }
        </ul>
      </div>
    </div>
  </nav>
</header>
<div class="container">
```

## Customers


Solo podrán ver en el menu las opciones Películas y Series.

En las cuales verán unas vistas las cuales encontrarán información de las películas y tendrán la opción de buscar también.

Una vez una de las películas o series sea escogida por el customer podra ver mas información y descargarla a través de un link.



## Workers

	Customer	Pegis	Peliculas	Series	Categorias	Control Peliculas	Control Series
-----------------------------------------------------------------------------------	----------	-------	-----------	--------	------------	-------------------	----------------

Encontrarán en el menu las mismas y más opciones que los customers.

Serán capaces de ver, crear, modificar y eliminar datos de los customers, pegis, categorías y películas y series. Estas dos últimas a través de las opciones Control Peliculas y Control Series. Una de las funcionalidades interesantes de esto es que un trabajador puede ser capaz de banear-desactivar la cuenta de un usuario que haga uso inadecuado.


Customer						
Crear Nuevo						
Direccion	Nombre	Apellidos	Telefono	CuentaUsuario		
Avenida M	customer	custo	642443234	<a href="mailto:customer1@hotmail.com">customer1@hotmail.com</a>	Editar	Info Eliminar
Avenida M	customer	custo	642443265	<a href="mailto:customer2@hotmail.com">customer2@hotmail.com</a>	Editar	Info Eliminar
C/cc	Alvaro	Alvaro	698745123	<a href="mailto:alvaro@altair.com">alvaro@altair.com</a>	Editar	Info Eliminar

Control Series						
Crear Nuevo						
Nombre	Director	Categoria	Pegi	Temporada		
Juego de tronos	David Benioff	Ficcion	Pegi 12	Temporada	Editar	Info Eliminar
Breaking Bad	Vince Gilligan	Ficcion	Pegi 12	Temporada	Editar	Info Eliminar
Breaking Bad	Vince Gilligan	Ficcion	Pegi 12	Temporada	Editar	Info Eliminar

Pegis		
Crear Nuevo		
EdadMin	PegiEdad	
Pegi 3	3	Editar Details

Categorias		
Crear Categoria		
Name		
Ficcion	Editar	Details

## Admin

	Cuentas	Administradores	Workers	Customer	Menus	RoleMenus	Roles	Pegis	Peliculas	Series	Categorias	Control Peliculas	Control Series
-----------------------------------------------------------------------------------	---------	-----------------	---------	----------	-------	-----------	-------	-------	-----------	--------	------------	-------------------	----------------

Ese es el menu al entrar un administrador, este como el anterior tendrá acceso a todos los menus anteriormente nombrados y además tendrá acceso a:

Workers, Administradores así pudiendo registrar, eliminar o editar nuevas cuentas para trabajadores o administradores.

Worker				
<a href="#">Crear Nuevo</a>				
Nombre	Apellidos	Telefono	CuentaUsuario	
worker	wer	642443214	<a href="#">worker1@hotmail.com</a>	<a href="#">Editar</a> <a href="#">Info</a> <a href="#">Eliminar</a>
worker	worke	642443217	<a href="#">worker2@hotmail.com</a>	<a href="#">Editar</a> <a href="#">Info</a> <a href="#">Eliminar</a>

Admins				
<a href="#">Crear Nuevos</a>				
Nombre	Apellidos	Telefono	CuentaUsuario	
admin	min	642443212	<a href="#">admin1@hotmail.com</a>	<a href="#">Editar</a> <a href="#">Info</a> <a href="#">Eliminar</a>

Menus: Crear nuevos menus para nuevas funcionalidades dentro de la aplicación y RoleMenus con la cual podrá adjuntar a que tipo de rol se les mostrara dicha info.

Role Has Menu				
<a href="#">Crear Nuevo</a>				
Menu	Role			
Cuentas	Admin rol			<a href="#">Editar</a> <a href="#">Info</a> <a href="#">Eliminar</a>
Administradores	Admin rol			<a href="#">Editar</a> <a href="#">Info</a> <a href="#">Eliminar</a>

Menus				
<a href="#">Crear Nuevo</a>				
Controller	Action	Label		
CuentaUsuarios	Index	Cuentas	<a href="#">Editar</a>	<a href="#">Info</a> <a href="#">Eliminar</a>
Admins	Index	Administradores	<a href="#">Editar</a>	<a href="#">Info</a> <a href="#">Eliminar</a>
Worker	Index	Workers	<a href="#">Editar</a>	<a href="#">Info</a> <a href="#">Eliminar</a>



Cuentas: Con lo que podra modificar las cuentas que hayan creadas.

Cuentas de Usuario

Crear Nuevo

Nombre usuario	Contraseña	Email	Active	Role			
admin1	QQBsAHQAYQBpAHIAMQyADMAJAAIAA==	<a href="#">admin1@hotmail.com</a>	<input checked="" type="checkbox"/>	Admin rol	Editar	Detalles	Eliminar
admin2	QQBsAHQAYQBpAHIAMQyADMAJAAIAA==	<a href="#">admin2@hotmail.com</a>	<input type="checkbox"/>	Admin rol	Editar	Detalles	Eliminar
customer1	QQBsAHQAYQBpAHIAMQyADMAJAAIAA==	<a href="#">customer1@hotmail.com</a>	<input checked="" type="checkbox"/>	Customer rol	Editar	Detalles	Eliminar