

SPRINT SEMANA II

BLAZOR DEL LADO DEL CLIENTE

ANTES DE INICIAR TEN EN CUENTA LA SIGUIENTE INFORMACIÓN:

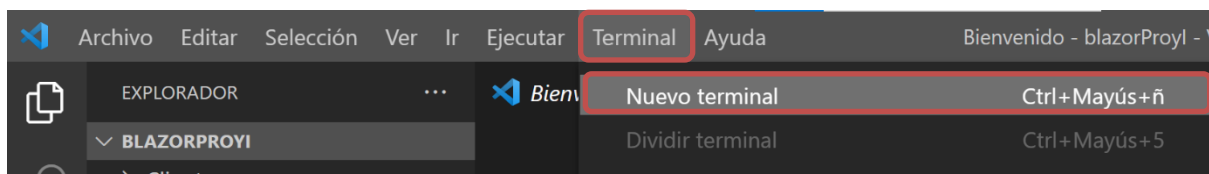
- Blazor es un framework que nos permite desarrollar aplicaciones web interactivas con .NET
- Existen dos modelos de alojamiento con Blazor:
 - o Del lado del cliente (dotnet new blazorwasm – Blazor WebAssembly App)
 - o Del lado del servidor (dotnet new blazorserver – Blazor Server App)
- Blazor del lado del cliente nos permite correr nuestra aplicación de Blazor en el navegador del usuario
- Blazor del lado del servidor significa que nuestra aplicación va a ejecutarse desde el servidor
- Blazor se puede utilizar en cualquier navegador moderno
- Podemos utilizar Visual Studio Community, Visual Studio Code o cualquier otro editor de código, como Sublime, Atom, Vim, Brackets, CodeShare, entre otros.

TEMA I: BLAZOR

1. Crear proyecto Blazor

```
C:\>cd ProysCicloIII
C:\ProysCicloIII>dotnet new blazorwasm -o blazorProyI --hosted
La plantilla "Blazor WebAssembly App" se creó correctamente.
Esta plantilla contiene tecnologías de terceros ajenos a Microsoft, consulte https://aka.ms/aspnetcore/5.
Procesando acciones posteriores a la creación...
Ejecutando "dotnet restore" en blazorProyI\blazorProyI.sln...
Determinando los proyectos que se van a restaurar...
Se ha restaurado C:\ProysCicloIII\blazorProyI\Shared\blazorProyI.Shared.csproj (en 71 ms).
Se ha restaurado C:\ProysCicloIII\blazorProyI\Server\blazorProyI.Server.csproj (en 5,6 sec).
Se ha restaurado C:\ProysCicloIII\blazorProyI\Client\blazorProyI.Client.csproj (en 16,37 sec).
Restauración realizada correctamente.
C:\ProysCicloIII>cd blazorProyI
C:\ProysCicloIII\blazorProyI>code .
```

2. Abrimos un nuevo terminal desde el Visual Studio Code.



3. Ejecutamos el proyecto de server

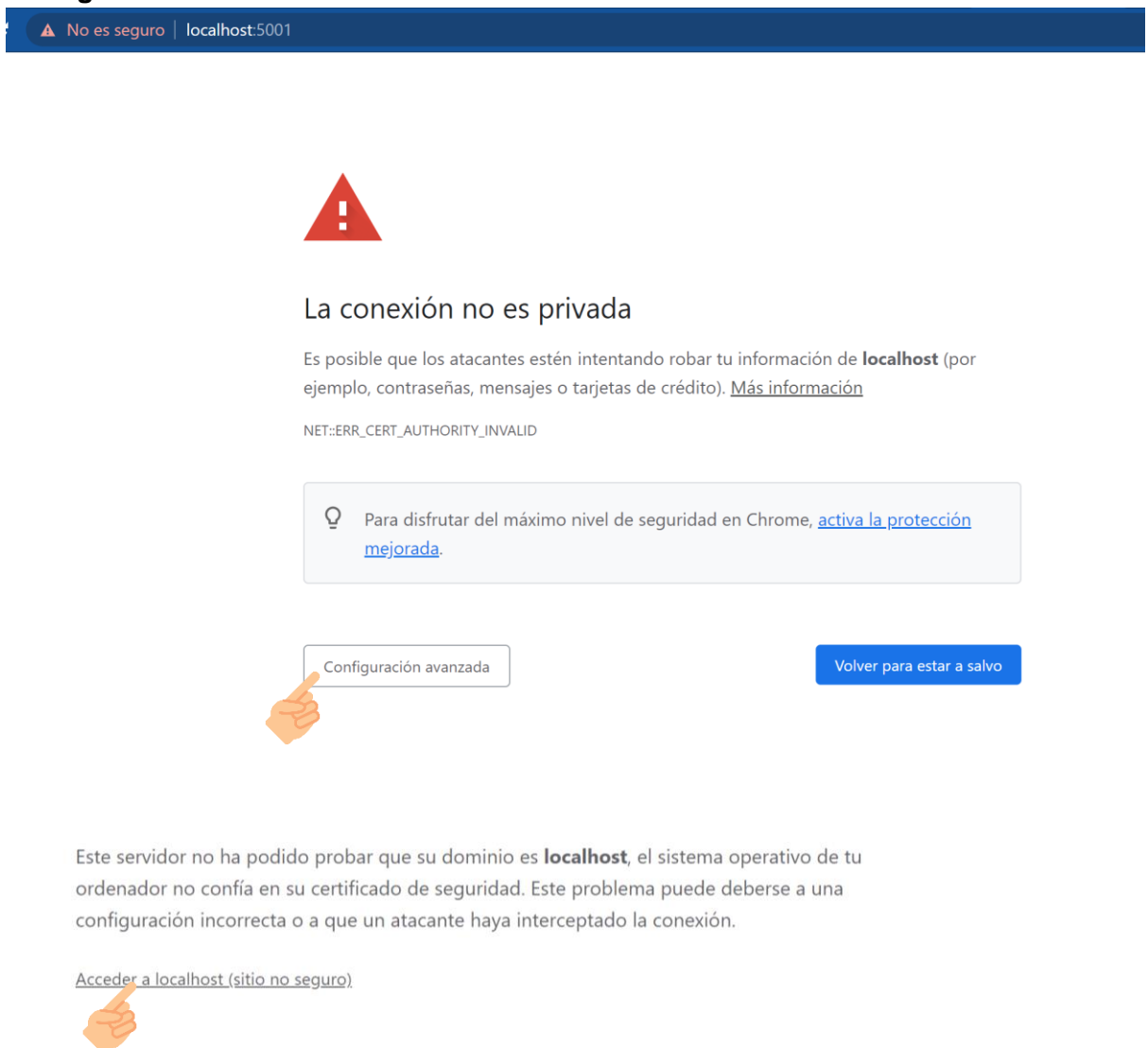
IMPORTANTE: Para que se muestren los cambios en el navegador en tiempo real, ejecutar: dotnet watch run

```
PS C:\ProysCicloIII\Sprints_G02\ProyPeliculas>cd .\Server\
PS C:\ProysCicloIII\Sprints_G02\ProyPeliculas\Server>dotnet watch run
```

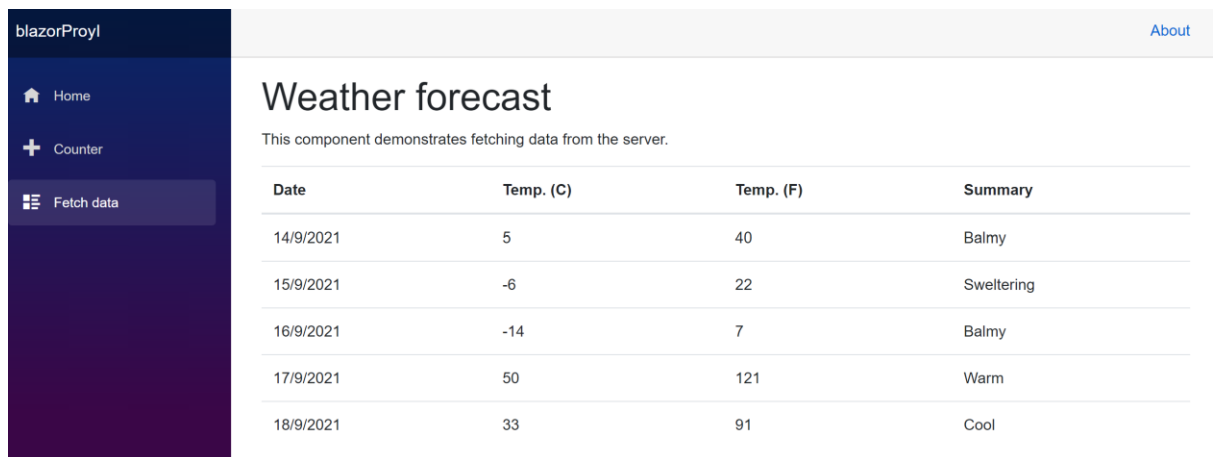
4. Seleccionamos con Ctrl oprimido el localhost

```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN  powershell
localhost:5001
info: Microsoft.Hosting.Lifetime[0] host:5000
      Now listening on: http://localhost:5000
host:5000 1+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
1+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\ProysCicloIII\blazorProyI\Server
```

5. Al dar clic se abre una ventana en el navegador principal de tu computador. Si muestra un mensaje de **Conexión no es segura**, puedes seleccionar la opción de **Configuración avanzada**.



6. Salida esperada:

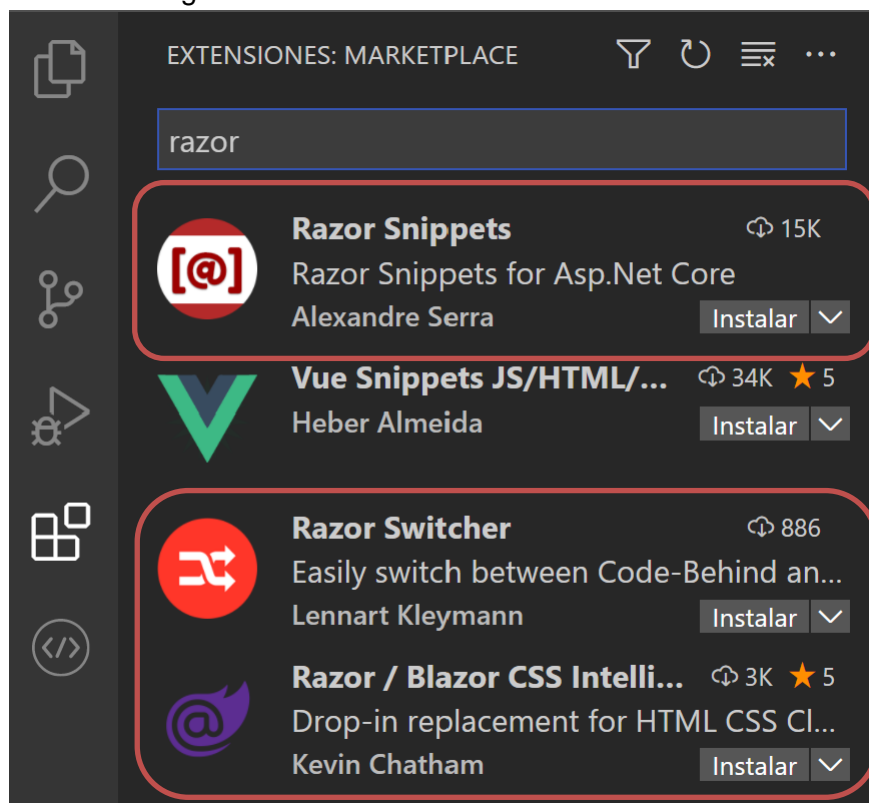


TEMA II: RAZOR

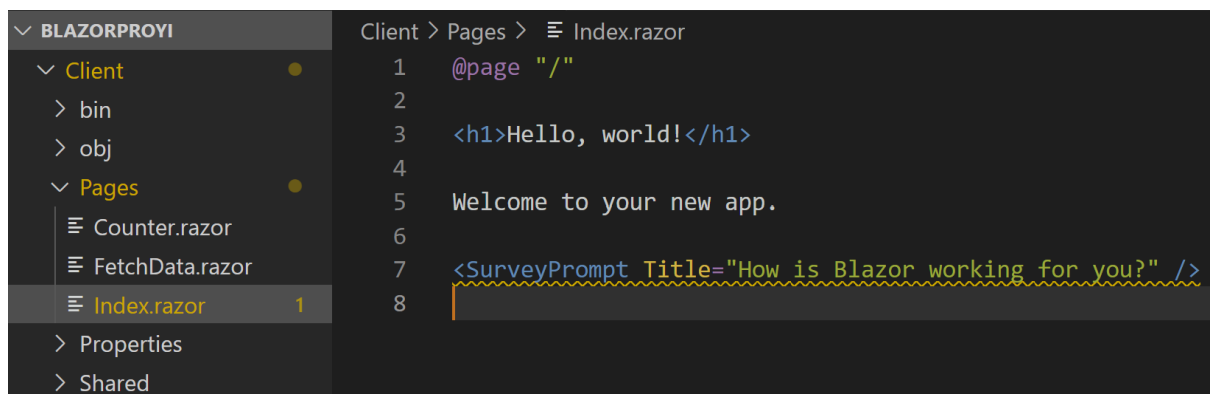
IMPORTANTE PARA QUIENES ESTAN TRABAJANDO EN EL IDE DE VISUAL STUDIO:

Necesitas la versión de Visual Studio 2019, debido a que RAZOR es una herramienta reciente.

1. Instalar las siguientes extensiones en el VSc:



2. Acceder a:
Proyecto: Client
Subcarpeta: Pages
Archivo: Index.razor



```
Client > Pages > Index.razor
1  @page "/"
2
3  <h1>Hello, world!</h1>
4
5  Welcome to your new app.
6
7  <SurveyPrompt Title="How is Blazor working for you?" />
8
```

3. Empieza a modificar el código y a visualizar los cambios en el navegador

Ten en cuenta la siguiente información:

- Razor, utiliza directivas que inician con el @
- **@page** "URI" indica la URI de la ruta de la vista que se esta creando.
- Algunas etiquetas HTML que se asocian en el ejemplo son:
 - **h5**: Para escribir texto, inicia en h1 y finaliza en h6.
 - **br**: Se utiliza para salto de línea
 - **p**: Para la creación de párrafos.
 - **form**: Para la creación de formularios
 - **input:text**: Para la creación de campos que permiten el ingreso de texto por parte del usuario
 - **input:number**: Para la creación de campos que permiten el ingreso de números por parte del usuario
 - **input:email**: Para la creación de campos que permiten el ingreso de correo electrónico por parte del usuario

Etiqueta EMMET: Es un helper para generar estructuras de código de forma rápida, mediante nombres de etiquetas.

- *En este caso el comando `input:text` genera código de forma rápida. Escribes la instrucción y oprimas la tecla Enter.*

```

Client > Pages > Index.razor
1  @page "/"
2  <div>
3  |    <h1>Películas</h1>
4  </div>
5  <div class="row row-cols-1 row-cols-md-3 g-4 mb-3">
6  |    <div class="col">
7  |        <div class="card mb-3 h-100" style="width: 18rem;">
8  |            
9  |            <div class="card-body">
10 |                <h5 class="card-title">@movie1</h5>
11 |                <p class="card-text">Sinopsis</p>
12 |                <a href="#" class="btn btn-primary">Ver película</a>
13 |            </div>
14 |        </div>
15 |    </div>
16 |    <div class="col">
17 |        <div class="card mb-3 h-100" style="width: 18rem;">
18 |            
19 |            <div class="card-body">
20 |                <h5 class="card-title">@movie2</h5>
21 |                <p class="card-text">Sinopsis</p>
22 |                <a href="#" class="btn btn-primary">Ver película</a>
23 |            </div>
24 |        </div>
25 |    </div>
26 </div>
27 <div class="mb-3">
28 |    <input class="form-control form-control-sm" type="text" placeholder="Película">
29 </div>
30 <div class="mb-3">
31 |    <input class="form-control form-control-sm" type="text" placeholder="Genero">
32 </div>
33 <div class="mb-3">
34 |    <input type="file" name="" id="" class="form-control">
35 </div>

```

4. Crea el primer método que se active con un evento clic desde un botón. Después del cierre de la etiqueta form, escribe el siguiente código:

```

46
47 <div>
48     <label class="form-label">Película 1 @function_convert_mayusc(movie1)</label>
49     <br>
50     <label class="form-label">Película 2 @function_convert_minusc(movie2)</label>
51 </div>
52 <div>
53     <button class="btn btn-info" @onclick="@clik_event_button">Da clic</button>
54     <button class="btn btn-danger" @onclick="@clik_event_cancel">Cancelar</button>
55 </div>
56 @code {
57     public string movie1 = "Star wars";
58     public string movie2 = "Avengers";
59     public string movie3 = "Spiderman";
60     /* Funciones flecha estructura: variable(parametros) => instrucciones; */
61     public string function_convert_mayusc(string value) => value.ToUpper();
62     public string function_convert_minusc(string value) => value.ToLower();
63
64     /* Crear métodos */
65     public void clic_event_button()
66     {
67         Console.WriteLine("Has dado clic en el botón");
68     }
69     public void clic_event_cancel()
70     {
71         Console.WriteLine("Cancelaste el proceso");
72     }
73 }

```

5. Verifica la salida en el navegador.

Puedes **detener el anterior proceso** en el terminal de Visual Studio Code con Ctrl + C

Volver a ejecutar el comando ***dotnet watch run***

6. ¿Qué sucede si modificamos el orden de las etiquetas label e input?

Recomendación: Manejar toda la distribución de componentes input y label en filas segmentadas por columnas

7. Salida generada:

```

28     <div class="button-group">
29         <button class="btn btn-info" @onclick="@Confirmar">Aceptar</button>
30     </div>
31 </div>
32 @code {
33     string ciclo = "Desarrollo de Software";
34     private void Confirmar(){
35         Console.WriteLine("Acabas de dar clic en el boton, evento capturado");
36     }
37 }

```


IMPORTANTE: Console.WriteLine, es un método que escribe una salida por consola del navegador.

- Ingresa con F12.
- Selecciona la opción Console
- Diligencia los 3 campos del formulario y da clic en el botón
- Verifica que por consola se muestre el mensaje: Acabas de dar clic en el botón...


ProyPelículas_G63
About

Home
Counter
Fetch data
Movie


Películas



Star wars
Some quick example text to build



Avengers
Some quick example text to build



Spiderman
Some quick example text to build

Seleccionar archivo
Ningún archivo seleccionado

Película 1 STAR WARS
Película 2 avengers

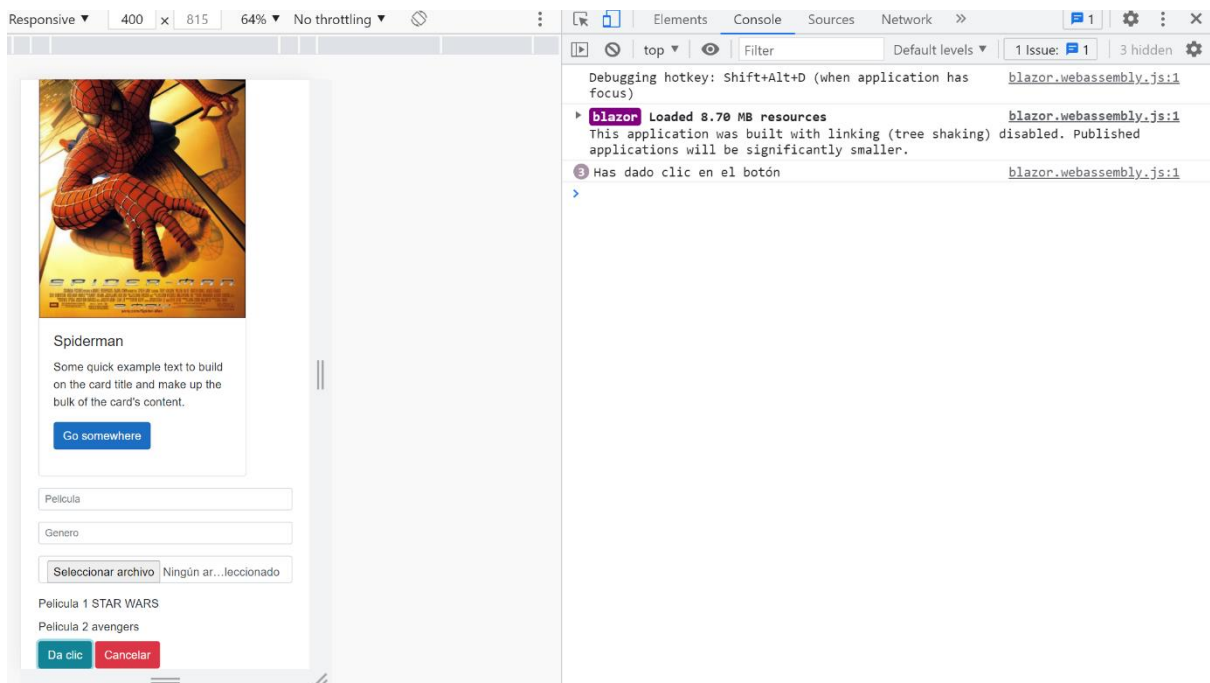
Da clic
Cancelar

- Modifica el formulario.
 - Crea un textarea para comentarios.

b. Inserta clases para dar estilo al input y mejore la visualización para el usuario

```
Client > Pages > Index.razor
1 @page "/"
2 <h5>Nombre del ciclo, @ciclo.ToUpper()</h5>
3 <br>
4 <div class="container section">
5     <form>
6         /* Primera fila, contiene 1 input de texto */
7         <div class="form-group row">
8             <label for="name" class="col-md-2 col-form-label">Nombre</label>
9             <input type="text" name="name" id="name" class="form-control">
10        </div>
11        /* Segunda fila contiene 2 input, uno recibe texto y otro números */
12        <div class="form-group row">
13            <div class="col">
14                <label for="ciclo" class="col-md-2 col-form-label">Ciclo</label>
15                <input type="text" name="ciclo" id="ciclo" class="form-control">
16            </div>
17            <div class="col">
18                <label for="num_ciclo" class="col-md-2 col-form-label">Número</label>
19                <input type="number" name="num_ciclo" id="num_ciclo" class="form-control">
20            </div>
21        </div>
22        /* Tercer fila contiene un textarea */
23        <div class="form-group row">
24            <label for="message" class="col-md-2 col-form-label">Comentarios</label>
25            <div class="col-md-10">
26                <textarea name="message" id="message" cols="30" rows="10" placeholder="Aqui van tus comentarios" class="form-control">
27            </div>
28        </div>
29    </div>
30 </form>
```

c. Visualiza la salida en el navegador:



HA LLEGADO EL MOMENTO DE DIVIDIR ACTIVIDADES PARA EL SPRINT II:

Enlace drive:

https://drive.google.com/drive/u/0/folders/1aVunIz1GrgN1o7EeNnW7k3n0E_aiNvX3

1. Administrador de la configuración:

- a. Crea en el GitHub la siguiente distribución de carpetas:

YanethM Create readme 089406e 26 seconds ago 16 commits		
📁 Sprint I	Rename readme to readme.md	3 days ago
📁 Sprint II	Create readme	3 days ago
📁 Sprint III	Create readme	1 minute ago
📁 Sprint IV	Create readme	1 minute ago
📁 Sprint V	Create readme	26 seconds ago
📄 README.md	Initial commit	3 days ago

- d. Dentro de la carpeta Sprint II, crea la siguiente distribución:

main MinTIC_g63 / Sprint II / Go to file Add file ...		
YanethM Create read d5d90fd now History		
..		
📁 Código fuente	Create read	32 seconds ago
📁 Mapeo de datos	Create read	now
📁 Tablero Kanban	Create read	1 minute ago
📁 Wireframe	Create read	1 minute ago
📄 DailyMeeting	Create DailyMeeting	3 minutes ago
📄 readme	Create readme	3 days ago

2. Líder del proyecto:

- a. Garantiza que todos los miembros del equipo realizaron el paso a paso propuesto, solicitando a cada uno de ellos una captura de pantalla al final del formulario.
- b. En la carpeta Sprint I del GitHub, validar que en cada reunión se actualiza el archivo Daily Meeting, donde cada desarrollador del equipo debe responder las siguientes 3 preguntas:
 - i. ¿Qué hizo ayer?
 - ii. ¿Qué hará hoy?
 - iii. ¿Qué dificultades ha tenido?
 - iv. Genera la siguiente estructura para facilitar a los demás integrantes el diligenciamiento de dicho seguimiento diario:
 - v. **NO OLVIDES GRABAR LA REUNIÓN.**

<> Edit new file

Preview

Spaces

2

No wrap

```
1 *****
2 Enlace grabación reunión:
3 Fecha:
4 *****
5 Desarrollador:
6 i. ¿Qué hizo ayer?
7 ii. ¿Qué hará hoy?
8 iii. ¿Qué dificultades ha tenido?
9 *****
10 Desarrollador:
11 i. ¿Qué hizo ayer?
12 ii. ¿Qué hará hoy?
13 iii. ¿Qué dificultades ha tenido?
14 *****
15 Desarrollador:|
16 i. ¿Qué hizo ayer?
17 ii. ¿Qué hará hoy?
18 iii. ¿Qué dificultades ha tenido?
19 *****
20 Desarrollador:
21 i. ¿Qué hizo ayer?
22 ii. ¿Qué hará hoy?
23 iii. ¿Qué dificultades ha tenido?
24 *****
```

vi. Genera el tablero Kanban, en cualquiera de las siguientes plataformas:

1. Jira
2. Trello
3. GitHub

vii. Sube una imagen en el GitHub del tablero generado, en la carpeta Tablero Kanban.

3. Diseñador de UI:

- a. Genera un formulario que permita la administración de películas que hay disponibles en cartelera para ver en cine. ¿Qué información es relevante? Antes de ponerlo en consideración con el grupo de trabajo, genera tu propia propuesta de interfaz. Luego, puedes generar una nueva versión del diseño.
- b. Todas las versiones que realices, debes de subirlas en la carpeta de Wireframe.
- c. El wireframe lo debes de realizar utilizando la herramienta Figma, puedes ingresar al sitio: <https://www.figma.com/>. **Recuerda:** Incluye la capacitación en el manejo de la herramienta, como tiempo que utilizas en desarrollo de tu actividad en el proyecto.
- d. Cuando se tenga la primera versión del formulario generado en Razor, valida que tu diseño si hubiese sido implementado. Todo cambio o variación que percibas lo debes de comentar.

4. Diseñador de software:

- a. Debes reunirte previamente con el diseñador de UI y revisar los campos del formulario, validar como generarías el Modelo Entidad Relación, independiente de que sólo sea un formulario, pueden existir campos que contengan información de otras interfaces, aún desconocidas para el equipo de desarrollo, pero que puedes plantear en tu diseño del mapeo de la Bases de datos.
- b. Recuerda, tu te encargas de definir estándares y la mnemotecnica para el nombramiento de atributos, clases y entidades.

5. Tester:

- a. Plantea los tipos de validaciones que deberían de incluirse en cada campo del formulario, utilizando las pruebas de partición equivalente en los campos Input. Ejemplo:
 - i. Es un campo requerido u opcional
 - ii. Tiene un límite mínimo y máximo de caracteres
 - iii. Tiene un manejo de rangos, o una cantidad específica de dígitos
 - iv. Debe tener un valor por default
 - v. Es una lista desplegable dinámica o estática (Pertenece a un conjunto de datos predefinido)
 - vi. Pon en consideración que campos input, sería preferible limitar a ser una lista desplegable o algún tipo de componente de selección para evitar entradas y errores ortográficos por parte del usuario
- b. Recuerda: Incluye la capacitación en la generación de pruebas de partición equivalente, como tiempo que utilizas en desarrollo de tu actividad en el proyecto.
- c. Valida el funcionamiento y ejecución del proyecto.