

Matías Bidarte – 325937

Grupo N2B

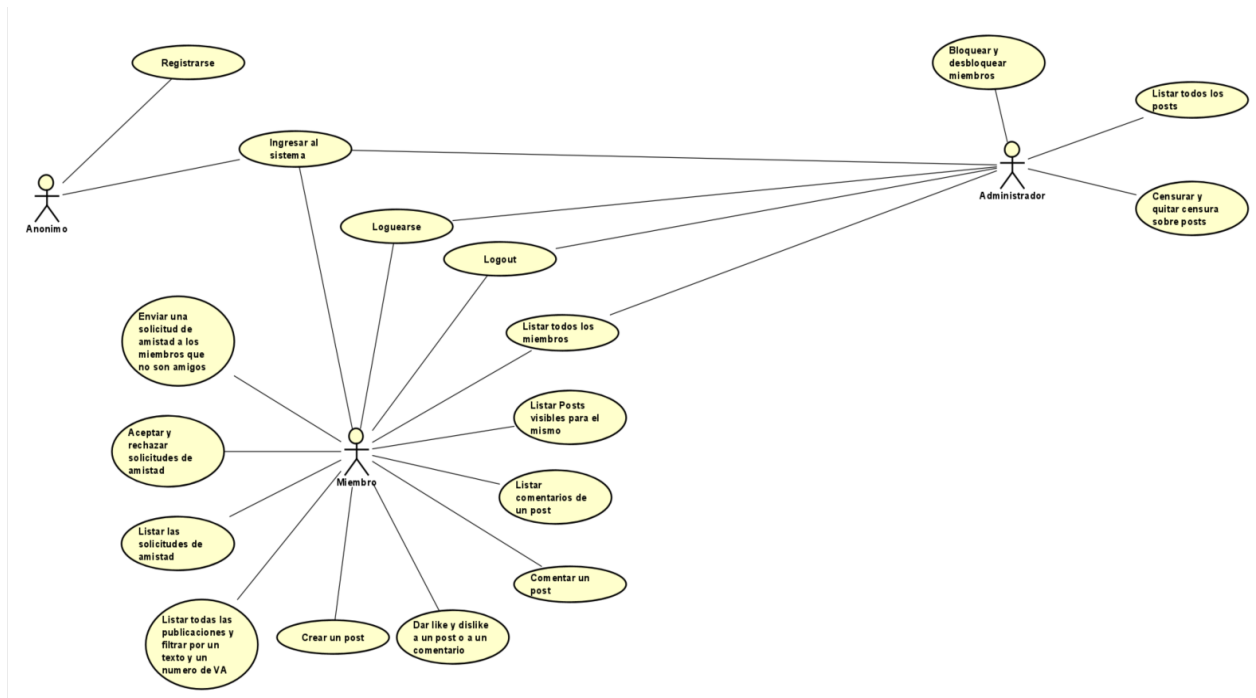
# Obligatorio 2

Docente – Santiago Baillo



[illegible]

# Diagrama de casos de uso



Url del proyecyo en somee: <http://www.P2Obli2Ort.somee.com>

# Testing

- **Funcionalidad: Login**

- Caso de prueba: Login exitoso de un usuario administrador
  - Pasos: 1) Navegar a la pantalla de Login
    - 2) Ingresar con email y contraseña validos
    - 3) Clickear el boton "Login"
  - Datos: - usuario: [maty@gmail.com](mailto:maty@gmail.com)
    - contraseña: 123
  - Resultado esperado: El administrador logra ingresar al sistema y puede ver las siguientes opciones: Home, Privacy, ListadoMiembros, Listado Posts, Logout.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

[MVC](#) [Home](#) [Privacy](#) [Listado Miembros](#) [Listado Posts](#) [Logout](#)

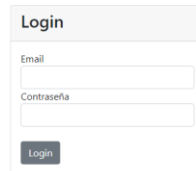
Welcome

Learn about [building Web apps with ASP.NET Core](#).

© 2023 - MVC - [Privacy](#)

- Caso de prueba: Login fallido de un administrador
  - Pasos: 1) Navegar a la pantalla de Login
    - 2) Ingresar con email y contraseña validos
    - 3) Clickear el botón "Login"
  - Datos: - usuario: [maty@gmail.com](mailto:maty@gmail.com)
    - contraseña: algo123

- Resultado esperado: Se redirige a la misma pantalla de login, y se despliega en rojo un error con el texto “Usuario o contraseña incorrectos”.
- Resultado obtenido: LOGRADO.
- Evidencia:



email o contraseña incorrectos

- **Funcionalidad: Ver todos los miembros del sistema**
  - Caso de prueba: Ver todos los usuarios del sistema.
    - Pasos: 1) Ingresar al sistema con un usuario existente
      - 2) Navegar hacia la pantalla “Listado Miembros”
    - Datos: - usuario: [maty@gmail.com](mailto:maty@gmail.com)
      - contraseña: 123
    - Resultado esperado: Se ve una tabla con todos los datos de los miembros del sistema, ordenados por apellido y nombre, y una columna “Acciones” para realizar sobre los mismos.
    - Resultado obtenido: LOGRADO.
    - Evidencia:

### Lista de miembros

Email	Nombre	Apellido	Fecha de nacimiento	Acciones
nicolas.buffa@gmail.com	nicolas	buffa	01/10/2002	<a href="#">Bloquear</a>
emi@gmail.com	emiliano	ferreira	04/01/2002	<a href="#">Bloquear</a>
block@gmail.com	block	gomez	24/06/1998	<a href="#">Desbloquear</a>
jose@gmail.com	jose	gomez	24/06/1998	<a href="#">Bloquear</a>
pepe@gmail.com	pepe	Gomez	12/12/1980	<a href="#">Bloquear</a>
lola.manyaxiempregmail.com	lola	gonzalez	23/04/2002	<a href="#">Bloquear</a>
lolo@gmail.com	lolo	jimenez	14/07/1999	<a href="#">Bloquear</a>
roberto.lol@gmail.com	roberto	llores	10/10/2010	<a href="#">Bloquear</a>
lucracia.robot@gmail.com	lucracia	lorenza	19/06/2007	<a href="#">Bloquear</a>
luisa.robb@gmail.com	luisa	robb	14/08/2000	<a href="#">Bloquear</a>
lore.progamer@gmail.com	lore	rodriguez	16/05/2005	<a href="#">Bloquear</a>

- **Funcionalidad: Bloquear y desbloquear miembros (administrador)**

- Caso de prueba: Bloquear y desbloquear miembros
  - Pasos: 1) Ingresar al sistema como usuario administrador
    - 2) Navegar hacia la pantalla “Listado Miembros”
    - 3) Clickear el botón “Bloquear” o “Desbloquear”
  - Datos: - usuario: [maty@gmail.com](mailto:maty@gmail.com)
    - contraseña: 123
  - Resultado esperado: Si el usuario estaba inicialmente desbloqueado, se verá cómo cambia de color el botón de rojo a verde, y cambia el texto de “Bloquear” a “Desbloquear”. Si el usuario estaba inicialmente bloqueado, se verá cómo cambia de color el botón de verde a rojo, y cambia el texto de “Desbloquear” a “Bloquear”.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

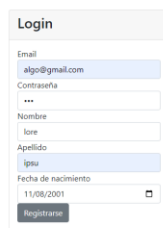
### Acciones

- Bloquear
- Bloquear
- Desbloquear
- Bloquear
- Bloquear
- Bloquear
- Bloquear
- Bloquear
- Bloquear

- **Funcionalidad: Registrarse**

- Caso de prueba: Registrarse en el sistema
  - Pasos: 1) Navegar a la pantalla de Registrarse
    - 2) Rellenar el formulario con los datos correspondientes
    - 3) Clickear el botón “Registrarse”
  - Datos: - email: [algo@gmail.com](mailto:algo@gmail.com)
    - contraseña: 123
    - nombre: lore
    - apellido: ipsu
    - fecha de nacimiento: 11/08/2001
  - Resultado esperado: Se redirige al usuario a la pagina principal, donde puede ver las opciones: Posts, Listado de miembros, Listado de solicitudes, Crear post, Publis por texto y VA, Logout.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

MVC Home Privacy Registrarse Login



Login

Email  
algo@gmail.com

Contraseña  
123

Nombre  
lore

Apellido  
ipsu

Fecha de nacimiento  
11/08/2001

Registrarse

© 2023 - MVC - Privacy

- Caso de prueba: Registrarse en el sistema con un mail ya existente
  - Pasos: 1) Navegar a la pantalla de Registrarse
    - 2) Rellenar el formulario con los datos correspondientes
    - 3) Clickear el botón “Registrarse”
  - Datos: - email: lolo@gmail.com
    - contraseña: 123
    - nombre: lore
    - apellido: ipsu
    - fecha de nacimiento: 11/08/2001



- Resultado esperado: Se redirige al usuario a la misma pantalla, y se le muestra un mensaje de error en color rojo que dice “El usuario ya está ingresado”
- Resultado obtenido: LOGRADO.
- Evidencia:

Login

Email

lolo@gmail.com

Contraseña

\*\*\*

Nombre

lone

Apellido

lpsu

Fecha de nacimiento

dd/mm/aaaa

Registrarse

El usuario ya está ingresado

- Caso de prueba: Registrarse en el sistema con datos vacíos
  - Pasos: 1) Navegar a la pantalla de Registrarse
    - 2) Clickear el botón “Registrarse”
  - Datos: Datos vacíos (o algún campo vacío)
  - Resultado esperado: Se redirige al usuario a la misma pantalla, y se le muestra un mensaje de error en color rojo que marca que algún campo no puede ser nulo.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

Login

Email

lolo@gmail.com

Contraseña

Nombre

lone

Apellido

lpsu

Fecha de nacimiento

dd/mm/aaaa

Registrarse

La contraseña no puede ser nula

- **Funcionalidad: Logout**

- Caso de prueba: Salir de la aplicación mediante logout
  - Pasos: 1) Ingresar al sistema con un usuario existente
  - 2) Clickear la pantalla "Logout"
  - Datos: - usuario: [maty@gmail.com](mailto:maty@gmail.com)
  - contraseña: 123
  - Resultado esperado: Se redirecciona a la pantalla inicial, viendo las opciones: Registrarse, Login.
  - Resultado obtenido: LOGRADO.
  - Evidencia:



- **Funcionalidad: Listado de posts (administrador)**

- Caso de prueba: Listar todos los posts exitosamente
  - Pasos: 1) Ingresar al sistema con un usuario administrador
  - 2) Navegar a la pantalla “Listado Posts”
  - Datos: - usuario: [maty@gmail.com](mailto:maty@gmail.com)
  - contraseña: 123
  - Resultado esperado: Se muestra en la pantalla una tabla con todos los datos de los posts en el sistema, y una columna “Acciones” para realizar sobre los mismos
  - Resultado obtenido: LOGRADO.
  - Evidencia:



- **Funcionalidad: Censurar y quitar censura de posts (administrador)**

- Caso de prueba: Listar todos los posts de manera exitosa
  - Pasos: 1) Ingresar al sistema con un usuario administrador  
2) Navegar a la pantalla "Listado Posts"
  - Datos: - usuario: [maty@gmail.com](mailto:maty@gmail.com)  
- contraseña: 123
  - Resultado esperado: Se muestra en la pantalla una tabla con todos los datos de los posts en el sistema, y una columna "Acciones" para realizar sobre los mismos
  - Resultado obtenido: LOGRADO.
  - Evidencia:

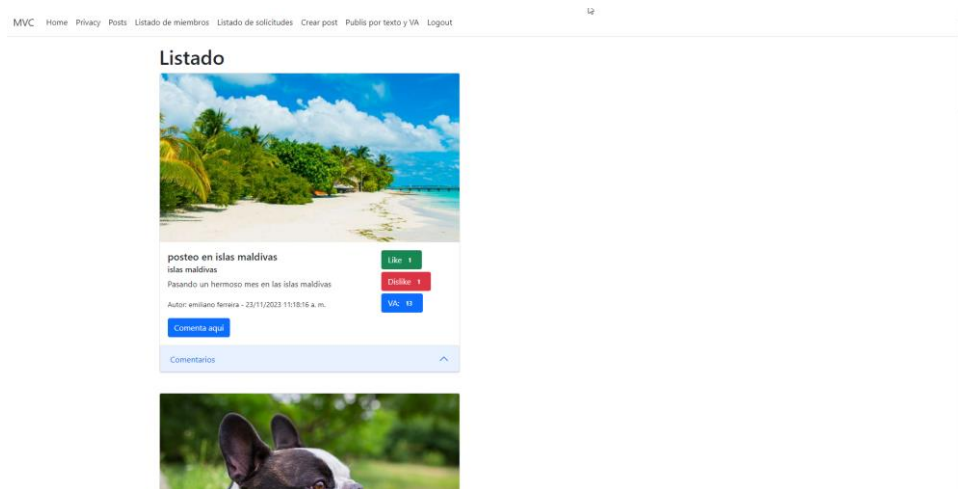
Acciones	
Censurar	
Censurar	
Censurar	
Censurar	
Censurar	
Censurar	
Censurar	
Quitar Censura	

- **Funcionalidad: Listar posts (miembro)**

- Caso de prueba: Listar los posts visibles para un miembro exitosamente
  - Pasos: 1) Ingresar al sistema con un usuario miembro  
2) Navegar a la pantalla "Posts"
  - Datos: - usuario: lolo@gmail.com  
- contraseña: 123
  - Resultado esperado: Se muestra en la pantalla todos los posts visibles para dicho miembro, mostrando todos los datos del post, botones para dar Like

y Dislike al post, un botón para comentar el post, y un acordeon que se abre para ver los comentarios del post.

- Resultado obtenido: LOGRADO.
- Evidencia:



- **Funcionalidad: Listar comentarios (miembro)**

- Caso de prueba: Listar los comentarios de un post dado de manera exitosa

- Pasos: 1) Ingresar al sistema con un usuario miembro

2) Navegar a la pantalla "Posts"

3) Clickear la sección "Comentarios" de un post específico

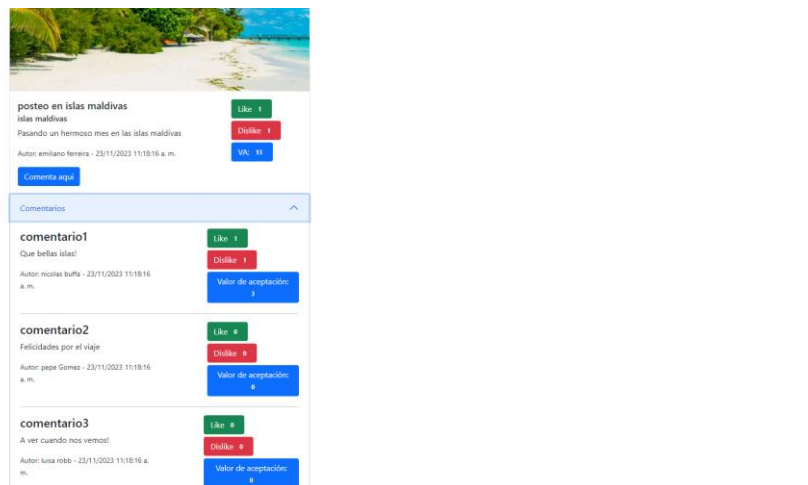
- Datos: - usuario: lolo@gmail.com

- contraseña: 123

- Resultado esperado: Se muestra en la pantalla todos los comentarios de un post, mostrando todos los datos del comentario y botones para dar Like y dislike al comentario

- Resultado obtenido: LOGRADO.

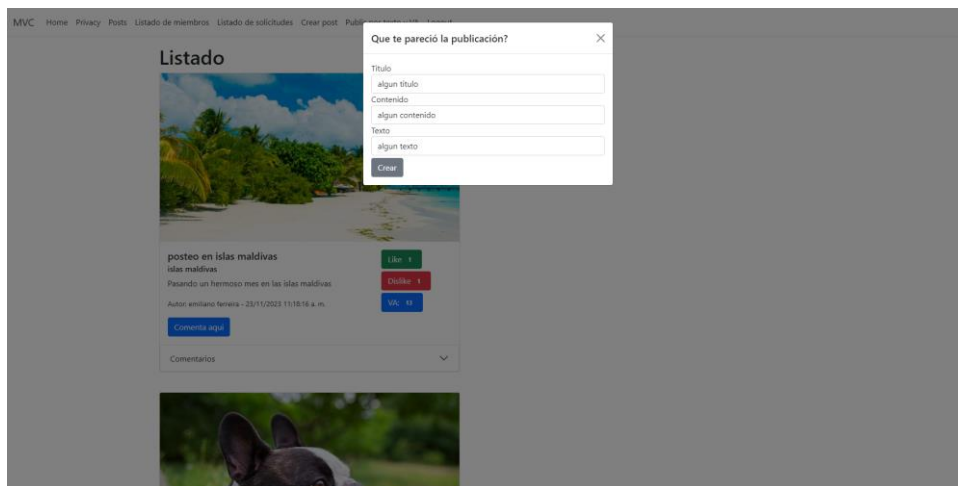
- Evidencia:





- **Funcionalidad: Comentar un post (miembro)**

- Caso de prueba: Comentar un post específico de manera exitosa
  - Pasos: 1) Ingresar al sistema con un usuario miembro
    - 2) Navegar a la pantalla “Posts”
    - 3) Clickear el botón “Comenta aquí” de un post específico
    - 4) Completar el formulario con los datos requeridos
    - 5) Clickear el botón “Crear”
  - Datos: - usuario: lolo@gmail.com
    - contraseña: 123
    - título del comentario: Algún titulo
    - contenido del comentario: Algún contenido
    - texto del comentario: Algún texto
  - Resultado esperado: Se redirige al usuario a la pantalla donde se listan todos los posts, y en la parte superior de la pantalla aparece un mensaje en color verde, con el texto “Comentario creado con éxito” y el comentario se encontraría al final de la lista de comentarios del post. Si el post no tenía comentarios anteriormente, se crea el acordeón de comentarios, y estará el mismo allí.
  - Resultado obtenido: LOGRADO.
  - Evidencia:



Comentario creado con éxito

## Listado



posteo en islas maldivas

islas maldivas

Pasando un hermoso mes en las islas maldivas

Autor: emiliano fernera - 23/11/2023 11:18:16 a. m.

[Comenta aquí](#)

Comentarios



islas maldivas

Pasando un hermoso mes en las islas maldivas

Autor: emiliano fernera - 23/11/2023 11:18:16 a. m.

[Comenta aquí](#)

Comentarios

comentario1

Que bellas islas!

Autor: nicolas buffa - 23/11/2023 11:18:16 a. m.

comentario2

Felicidades por el viaje

Autor: pepe Gomez - 23/11/2023 11:18:16 a. m.

comentario3

A ver cuando nos vemos!

Autor: luis robb - 23/11/2023 11:18:16 a. m.

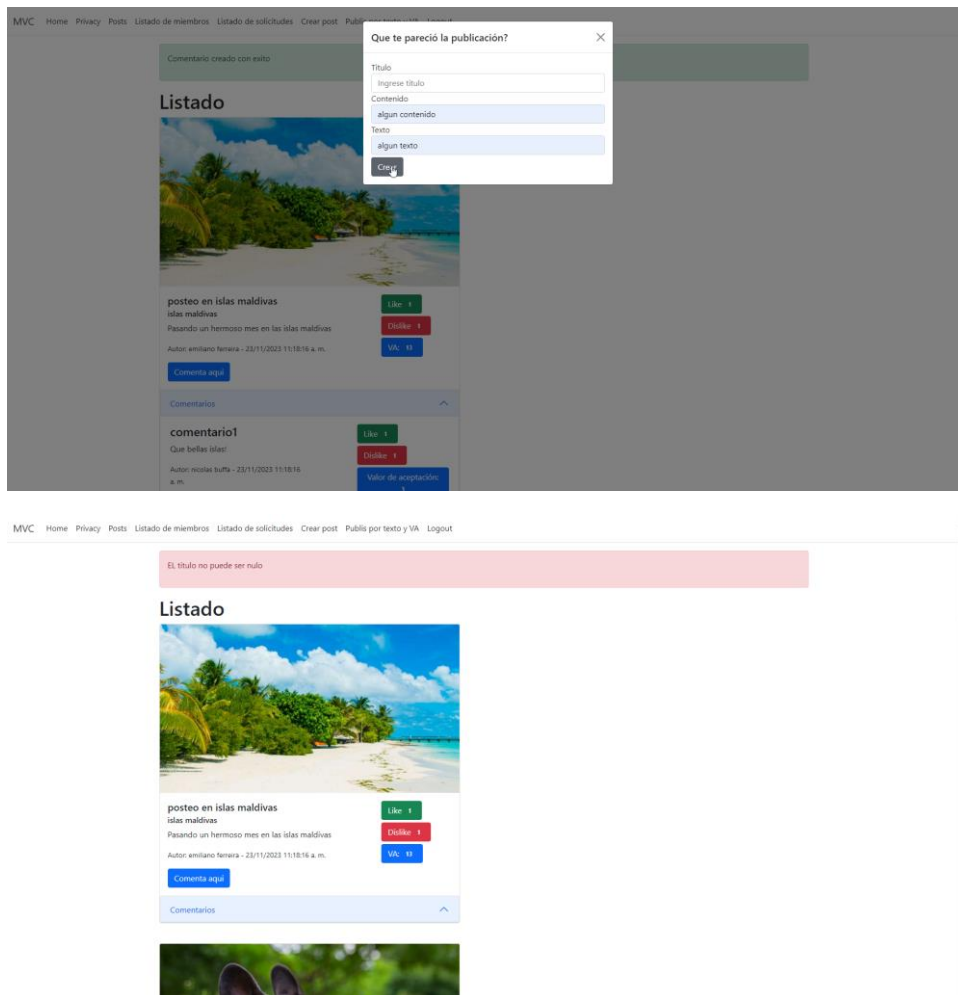
algun titulo

algun contenido

Autor: lolo jimeez - 23/11/2023 11:34:10 a. m.

- Caso de prueba: Comentar un post con datos vacíos
  - Pasos: 1) Ingresar al sistema con un usuario miembro
    - 2) Navegar a la pantalla “Posts”
    - 3) Clickear el botón “Comenta aquí” de un post específico
    - 4) Completar el formulario con algún campo vacío
    - 5) Clickear el botón “Crear”
  - Datos: - usuario: lolo@gmail.com
    - contraseña: 123
    - título del comentario: \*vacío\*
    - contenido del comentario: Algún contenido
    - texto del comentario: Algún texto
  - Resultado esperado: Se redirige al usuario a la pantalla donde se listan todos los posts, y en la parte superior de la pantalla aparece un mensaje en color rojo que muestra el texto “El titulo no puede ser nulo”.
  - Resultado obtenido: LOGRADO.

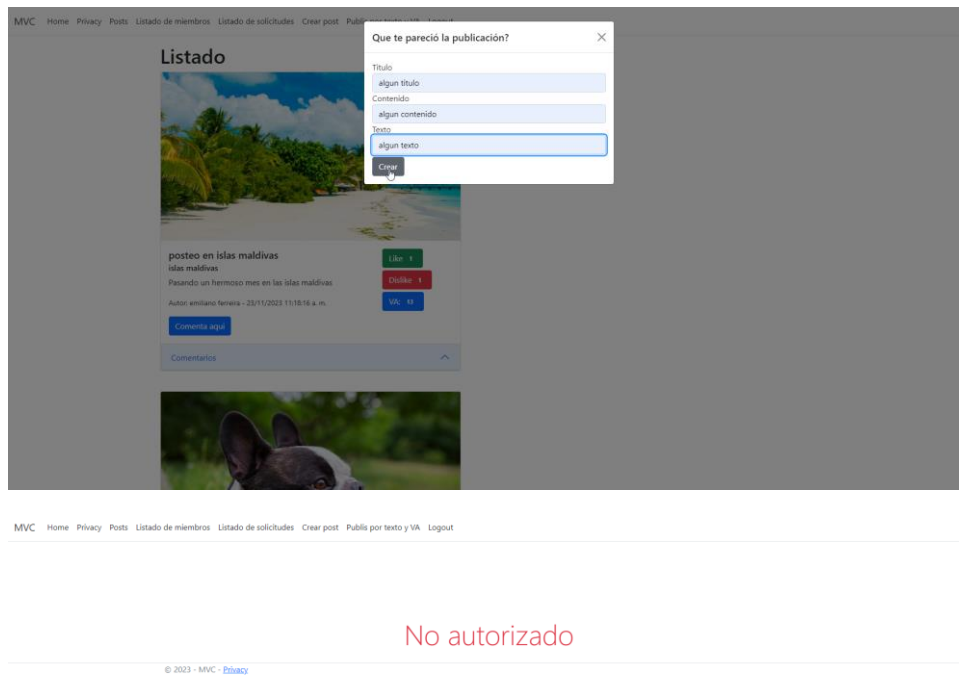
- Evidencia:



- Caso de prueba: Comentar un post estando el usuario bloqueado
  - Pasos: 1) Ingresar al sistema con un usuario miembro bloqueado
    - 2) Navegar a la pantalla “Posts”
    - 3) Clickear la sección “Comenta aquí” de un post específico
    - 4) Completar el formulario con datos
    - 5) Clickear el botón “Crear”
  - Datos: - usuario: block@gmail.com
    - contraseña: 123
    - título del comentario: Algún titulo
    - contenido del comentario: Algún contenido
    - texto del comentario: Algún texto
  - Resultado esperado: Se redirige al usuario a una pantalla genérica de error, que muestra el texto “No autorizado”.



- Resultado obtenido: LOGRADO.
- Evidencia:



- **Funcionalidad: Reaccionar un post o comentario (miembro)**

- Caso de prueba: Reaccionar a un post o un comentario exitosamente

- Pasos: 1) Ingresar al sistema con un usuario miembro

- 2) Navegar a la pantalla "Posts"

- 3) Clickear el botón "Like" o "Dislike" de un comentario o un post

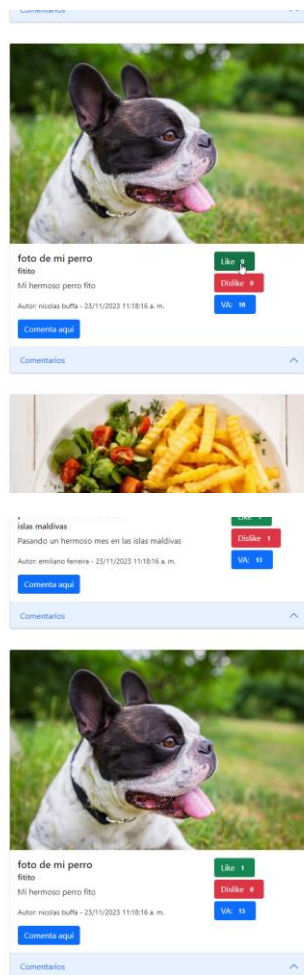
- Datos: - usuario: lolo@gmail.com

- contraseña: 123

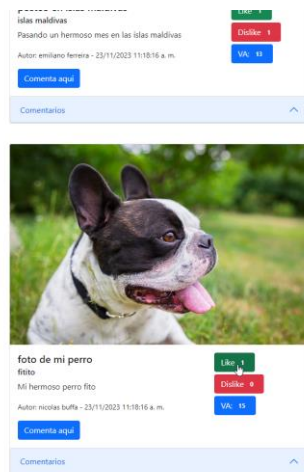
- Resultado esperado: Se redirige al usuario a la parte superior de la pantalla de posts, y dentro de la publicación reaccionada, tendría una reacción más de las que ya tenía anteriormente.

- Resultado obtenido: LOGRADO.

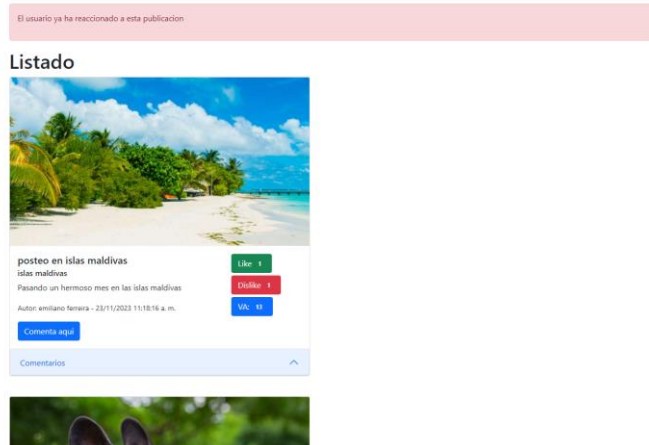
- Evidencia:



- Caso de prueba: Reaccionar a un post o comentario ya reaccionado.
  - Pasos: 1) Ingresar al sistema con un usuario miembro
  - 2) Navegar a la pantalla “Posts”
  - 3) Clickear el botón “Like” o “Dislike” de un comentario o un post, cuando este ya había sido likeado o dislikeado.
- Datos: - usuario: lolo@gmail.com
- contraseña: 123
- Resultado esperado: Se redirige al usuario a la parte superior de la pantalla de posts y se muestra un mensaje de color rojo con el texto “El usuario ya ha reaccionado a esta publicación”.
- Resultado obtenido: LOGRADO.
- Evidencia:



MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout



- **Funcionalidad: Crear un post (miembro)**

- Caso de prueba: Crear un post exitosamente
  - Pasos: 1) Ingresar al sistema con un usuario miembro
    - 2) Navegar a la pantalla “Crear Post”
    - 3) Rellenar el formulario con los datos correspondientes
    - 4) Clickear el botón “Crear”
  - Datos: - usuario: lolo@gmail.com
    - contraseña: 123
    - título del post: Algún titulo
    - contenido del post: Algún contenido
    - texto del post: Algún texto
    - privacidad: publico/privado
    - imagen: \*alguna imagen\*
  - Resultado esperado: Desaparecen todos los datos ingresados en el formulario, y en la parte superior de la pantalla aparece un mensaje de color verde que muestra “Post creado con éxito”
  - Resultado obtenido: LOGRADO.
  - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

### Crear nuevo post

Título	<input type="text" value="algún título"/>
Contenido	<input type="text" value="Algún contenido"/>
Texto	<input type="text" value="Algún texto"/>
Privacidad	<input type="text" value="Publico"/>
Imagen	<div>Seleccionar archivo WhatsApp Image 2023-10-31 at 22.14.54.jpeg</div>
<div>Crear</div>	

© 2023 - MVC - Privacy

## Crear nuevo post

Post creado con éxito

Título  
Ingresar título

Contenido  
Ingresar contenido

Texto  
Ingresar texto

Privacidad  
Publico

Imagen  
Seleccionar archivo Sin archivos seleccionados

Crear

© 2023 - MVC - Privacy

- Caso de prueba: Intentar crear un post con datos vacíos
  - Pasos: 1) Ingresar al sistema con un usuario miembro
    - 2) Navegar a la pantalla “Crear Post”
    - 3) Rellenar el formulario con algún dato vacío
    - 4) Clickear el botón “Crear”
  - Datos: - usuario: lolo@gmail.com
    - contraseña: 123
    - título del post: \*vacío\*
    - contenido del post: Algún contenido
    - texto del post: Algún texto
    - privacidad: publico/privado
    - imagen: \*alguna imagen\*
  - Resultado esperado: Desaparecen todos los datos ingresados en el formulario, y en la parte superior de la pantalla aparece un mensaje de color verde que muestra “Post creado con éxito”
  - Resultado obtenido: LOGRADO.
  - Evidencia:

## Crear nuevo post

El título no puede ser nulo

Título  
Ingresar título

Contenido  
Algún contenido

Texto  
Algún texto

Privacidad  
Publico

Imagen  
Seleccionar archivo Sin archivos seleccionados

Crear

© 2023 - MVC - Privacy



- Caso de prueba: Crear un post estando el usuario bloqueado
  - Pasos: 1) Ingresar al sistema con un usuario miembro bloqueado
    - 2) Navegar a la pantalla “Crear post”
    - 3) Completar el formulario con datos
    - 4) Clickear el botón “Crear”
  - Datos: - usuario: block@gmail.com
    - contraseña: 123
    - título del post: Algún titulo
    - contenido del post: Algún contenido
    - texto del post: Algún texto
    - privacidad del post: publico/privado
    - imagen del post: \*alguna imagen\*
  - Resultado esperado: Se redirige al usuario a una pantalla genérica de error, que muestra el texto “No autorizado”.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y V/A Logout

---

### Crear nuevo post

Título  
algún titulo

Contenido  
Algún contenido

Texto  
Algún texto

Privacidad  
Publico

Imagen  
Seleccionar archivo WhatsApp image 2023-10-31 at 22:14:54.jpeg

© 2023 - MVC - [Privacy](#)

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y V/A Logout

---

No autorizado

---

© 2023 - MVC - [Privacy](#)

- **Funcionalidad: Listar publicaciones filtradas por texto y valor de VA (miembro)**
  - Caso de prueba: Listar las publicaciones que cumplan con los filtros de manera exitosa
    - Pasos: 1) Ingresar al sistema con un usuario miembro
      - 2) Navegar a la pantalla “Publis por texto y VA”
      - 3) Ingresar un texto en el campo de texto
      - 4) Ingresar un numero en el campo de números
      - 5) Clickear el botón “Filtrar”
    - Datos: - usuario: lolo@gmail.com
      - contraseña: 123
      - texto: perro
      - numero: 5
    - Resultado esperado: Se muestra en la pantalla una tabla que muestra los datos de las publicaciones que cumplen con los requerimientos del filtro
    - Resultado obtenido: LOGRADO.
    - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publicar por texto y VA Logout

### Publicaciones por texto y VA

perro 0 Filtrar Reiniciar lista

Título	Creador	Contenido	Valor de aceptación	Tipo
comentario1	nicolas.buffa@gmail.com	Que bellas islas!	3	Comentario
comentario2	pepe@gmail.com	Felicidades por el viaje	0	Comentario
comentario3	luisa.rob@gmail.com	A ver cuando nos vemos!	0	Comentario
comentario1	pepe@gmail.com	el mejor pan del uruguay	0	Comentario
comentario2	roberto.lol@gmail.com	No me gusta el pan flauta, es feo	0	Comentario
comentario3	jose@gmail.com	Me gusta la flauta, pero prefiero el pan de molde	0	Comentario
comentario1	emi@gmail.com	Me encanta tu perro, es muy lindo	0	Comentario
comentario2	lolo@gmail.com	Felicidades por el perro!	0	Comentario
comentario3	lola.manyasiempre@gmail.com	Que feo perro, cambialo por otro	0	Comentario
comentario1	lolo@gmail.com	Salud!	0	Comentario
comentario2	lola.manyasiempre@gmail.com	Me diste hambre	0	Comentario
comentario3	nicolas.buffa@gmail.com	A ver cuando invitas	0	Comentario
comentario1	emi@gmail.com	Le tengo alergia a los gatos	0	Comentario
comentario2	nicolas.buffa@gmail.com	Muy lindo gato	0	Comentario
comentario3	roberto.lol@gmail.com	Que gato tan grande	0	Comentario
comentario perdido	lore.progamer@gmail.com	Este comentario es solo visible para mi	0	Comentario
posteo en las maldivas	emi@gmail.com	Pasando un hermoso mes en las islas maldivas	13	Post
imagen comiendo pan	jose@gmail.com	Un excelente pan flauta	0	Post

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publicar por texto y VA Logout

### Publicaciones por texto y VA

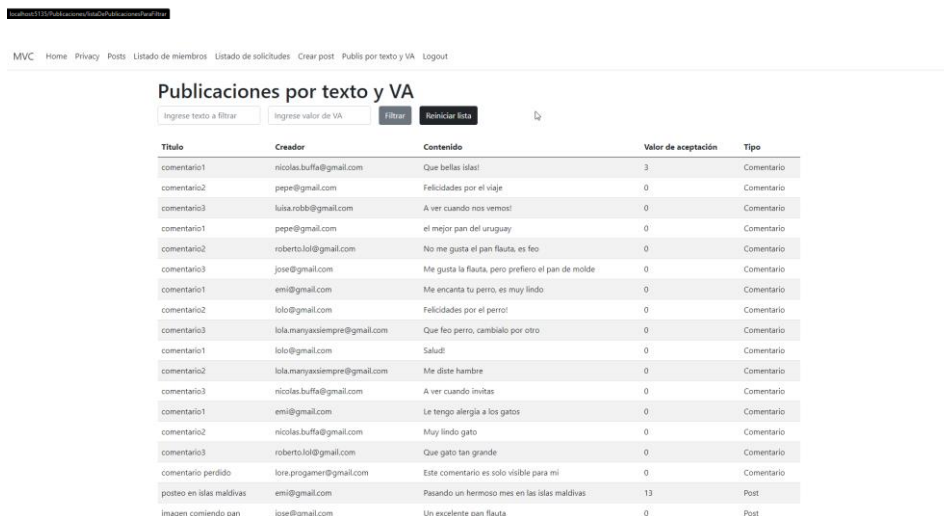
Ingresar texto a filtrar Ingresar valor de VA Filtrar Reiniciar lista

Título	Creador	Contenido	Valor de aceptación	Tipo
foto de mi perro	nicolas.buffa@gmail.com	Mi hermoso perro flto	15	Post

© 2023 - MVC - Privacy



- Caso de prueba: Reiniciar la lista de publicaciones filtrada
  - Pasos: 1) Ingresar al sistema con un usuario miembro
    - 2) Navegar a la pantalla “Publis por texto y VA”
    - 3) Ingresar un texto en el campo de texto
    - 4) Ingresar un numero en el campo de números
    - 5) Clickear el botón “Filtrar”
    - 6) Clickear el botón “Reiniciar lista”
  - Datos: - usuario: lolo@gmail.com
    - contraseña: 123
    - texto: perro
    - numero: 5
  - Resultado esperado: Se muestra en la pantalla la tabla de todas las publicaciones en el sistema, mostrando sus datos.
  - Resultado obtenido: LOGRADO.
  - Evidencia:



- Caso de prueba: Intentar filtrar sin completar algún campo
  - Pasos: 1) Ingresar al sistema con un usuario miembro
    - 2) Navegar a la pantalla “Publis por texto y VA”
    - 3) Clickear el botón “Filtrar”
    - 4) Clickear el botón “Reiniciar lista”
  - Datos: - usuario: lolo@gmail.com
    - contraseña: 123
    - texto: \*vacío\*
    - numero: \*vacío\*
  - Resultado esperado: Se muestra en la pantalla un mensaje de error en color rojo, que dice “el texto no puede ser nulo”.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

### Publicaciones por texto y VA

Ingrese texto a filtrar Ingrese valor de VA **Filtrar** **Reiniciar lista**

Título	Creador	Contenido	Valor de aceptación	Tipo
comentario1	nicolas.buffa@gmail.com	Que bellas islas!	3	Comentario
comentario2	pepe@gmail.com	Felicidades por el viaje	0	Comentario
comentario3	luisa.rob@gmail.com	A ver cuando nos vemos!	0	Comentario
comentario1	pepe@gmail.com	el mejor pan del uruguay	0	Comentario
comentario2	roberto.lol@gmail.com	No me gusta el pan flauta, es feo	0	Comentario
comentario3	jose@gmail.com	Me gusta la flauta, pero prefiero el pan de molde	0	Comentario
comentario1	emi@gmail.com	Me encanta tu perro, es muy lindo	0	Comentario
comentario2	lolo@gmail.com	Felicidades por el perro!	0	Comentario
comentario3	lola.manyaxiemp@gmail.com	Que feo perro, cambialo por otro	0	Comentario
comentario1	lolo@gmail.com	Salud!	0	Comentario
comentario2	lola.manyaxiemp@gmail.com	Me diste hambre	0	Comentario
comentario3	nicolas.buffa@gmail.com	A ver cuando invitas	0	Comentario
comentario1	emi@gmail.com	Le tengo alergia a los gatos	0	Comentario
comentario2	nicolas.buffa@gmail.com	Muy lindo gato	0	Comentario
comentario3	roberto.lol@gmail.com	Que gato tan grande	0	Comentario
comentario perdido	lore.progamer@gmail.com	Este comentario es solo visible para mi	0	Comentario
posteo en islas maldivas	emi@gmail.com	Pasando un hermoso mes en las islas maldivas	13	Post
imagen comiendo pan	jose@gmail.com	Un excelente pan flauta	0	Post

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

### Publicaciones por texto y VA

Ingrese texto a filtrar Ingrese valor de VA **Filtrar** **Reiniciar lista**

El texto no puede ser nulo

Título	Creador	Contenido	Valor de aceptación	Tipo
--------	---------	-----------	---------------------	------

© 2023 - MVC - Privacy

- Caso de prueba: No existen publicaciones para dichos filtros
  - Pasos: 1) Ingresar al sistema con un usuario miembro
  - 2) Navegar a la pantalla “Publis por texto y VA”
  - 3) Ingresar un texto en el campo de texto
  - 4) Ingresar un numero en el campo de números
  - 5) Clickear el botón “Filtrar”
  - 6) Clickear el botón “Reiniciar lista”
  - Datos: - usuario: lolo@gmail.com
  - contraseña: 123
  - texto: perro
  - numero: 50
  - Resultado esperado: Se muestra en la pantalla un mensaje informativo, en color celeste, que dice “No se encontraron publicaciones”.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publis por texto y VA Logout

### Publicaciones por texto y VA

perro 15

Título	Creador	Contenido	Valor de aceptación	Tipo
comentario1	nicolas.buffa@gmail.com	Que bellas islas!	3	Comentario
comentario2	pepe@gmail.com	Felicidades por el viaje	0	Comentario
comentario3	luisa.rob@gmail.com	A ver cuando nos vemos!	0	Comentario
comentario1	pepe@gmail.com	el mejor pan del uruguay	0	Comentario
comentario2	roberto.lol@gmail.com	No me gusta el pan flauta, es feo	0	Comentario
comentario3	jose@gmail.com	Me gusta la flauta, pero prefiero el pan de molde	0	Comentario
comentario1	emi@gmail.com	Me encanta tu perro, es muy lindo	0	Comentario
comentario2	lolo@gmail.com	Felicidades por el perro!	0	Comentario
comentario3	lola.manyasiempre@gmail.com	Que feo perro, cambialo por otro	0	Comentario
comentario1	lolo@gmail.com	Salud!	0	Comentario
comentario2	lola.manyasiempre@gmail.com	Me diste hambre	0	Comentario
comentario3	nicolas.buffa@gmail.com	A ver cuando invitas	0	Comentario
comentario1	emi@gmail.com	Le tengo alergia a los gatos	0	Comentario
comentario2	nicolas.buffa@gmail.com	Muy lindo gato	0	Comentario
comentario3	roberto.lol@gmail.com	Que gato tan grande	0	Comentario
comentario perdido	lore.progamer@gmail.com	Este comentario es solo visible para mi	0	Comentario
posteo en islas maldivas	emi@gmail.com	Pasando un hermoso mes en las islas maldivas	13	Post
imagen comiendo pan	jose@gmail.com	Un excelente pan flauta	0	Post

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publis por texto y VA Logout

### Publicaciones por texto y VA

Ingrese texto a filtrar Ingrese valor de VA

No se encontraron publicaciones

Título	Creador	Contenido	Valor de aceptación	Tipo
--------	---------	-----------	---------------------	------

© 2023 - MVC - [Privacy](#)

- **Funcionalidad: Listar las solicitudes de amistad (miembro)**

- Caso de prueba: Listar las solicitudes de amistad de un miembro exitosamente
  - Pasos: 1) Ingresar al sistema con un usuario miembro  
2) Navegar a la pantalla “Listado de solicitudes”
  - Datos: - usuario: lolo@gmail.com  
- contraseña: 123
  - Resultado esperado: Se redirige al usuario a una pantalla con sus solicitudes de amistad donde él es el solicitado, que se ven en una tabla, mostrándose todos sus datos, y una columna “Acciones” donde podrá aceptar o rechazar las solicitudes en estado “Pendiente Aprobación”
  - Resultado obtenido: LOGRADO.
  - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

### Listado de Solicitudes

Miembro Solicitante	Miembro Solicitado	Estado	Fecha de creación	Acciones
jose gomez	lolo jimenez	PENDIENTE_APROBACION	23/11/2023	<a href="#">Aceptar</a> <a href="#">Rechazar</a>
emiliano ferreira	lolo jimenez	APROBADA	23/11/2023	
nicolas buffa	lolo jimenez	APROBADA	23/11/2023	

© 2023 - MVC - Privacy

- **Funcionalidad: Aceptar una solicitud de amistad (miembro)**
  - Caso de prueba: Se acepta una solicitud de amistad exitosamente
    - Pasos: 1) Ingresar al sistema con un usuario miembro
      - 2) Navegar a la pantalla “Listado de solicitudes”
      - 3) Clickear el botón “Aceptar” para una solicitud
    - Datos: - usuario: lolo@gmail.com
      - contraseña: 123
    - Resultado esperado: Se redirige al usuario a la misma pantalla de listado de solicitudes, y se muestra un mensaje de éxito, en color verde, que dice “Solicitud aceptada!”, y el estado de dicha solicitud se cambia a “APROBADA”.
    - Resultado obtenido: LOGRADO.
    - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

### Lista de Solicitudes

Miembro Solicitante	Miembro Solicitado	Estado	Fecha de creación	Acciones
jose gomez	lolo jimenez	PENDIENTE_APROBACION	23/11/2023	<div>Aceptar</div> <div>Rechazar</div>
emiliano ferreira	lolo jimenez	APROBADA	23/11/2023	
nicolas buffa	lolo jimenez	APROBADA	23/11/2023	

© 2023 - MVC - [Privacy](#)

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

Solicitud aceptada!

### Lista de Solicitudes

Miembro Solicitante	Miembro Solicitado	Estado	Fecha de creación	Acciones
jose gomez	lolo jimenez	APROBADA	23/11/2023	
emiliano ferreira	lolo jimenez	APROBADA	23/11/2023	
nicolas buffa	lolo jimenez	APROBADA	23/11/2023	

© 2023 - MVC - [Privacy](#)

- **Funcionalidad: Rechazar una solicitud de amistad (miembro)**
  - Caso de prueba: Se rechaza una solicitud de amistad exitosamente
    - Pasos: 1) Ingresar al sistema con un usuario miembro
      - 2) Navegar a la pantalla “Listado de solicitudes”
      - 3) Clickear el botón “Rechazar” para una solicitud
    - Datos: - usuario: lolo@gmail.com
      - contraseña: 123
    - Resultado esperado: Se redirige al usuario a la misma pantalla de listado de solicitudes, y se muestra un mensaje de éxito, en color verde, que dice “Solicitud rechazada!”, y el estado de dicha solicitud se cambia a “RECHAZADA”.
    - Resultado obtenido: LOGRADO.
    - Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

### Listado de Solicitudes

Miembro Solicitante	Miembro Solicitado	Estado	Fecha de creación	Acciones
jose gomez	lolo jimenez	PENDIENTE_APROBACION	23/11/2023	<div><div>Aceptar</div><div>Rechazar</div></div>
emiliano ferreira	lolo jimenez	APROBADA	23/11/2023	
nicolas buffa	lolo jimenez	APROBADA	23/11/2023	

© 2023 - MVC - [Privacy](#)

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

Solicitud rechazada!

### Listado de Solicitudes

Miembro Solicitante	Miembro Solicitado	Estado	Fecha de creación	Acciones
jose gomez	lolo jimenez	RECHAZADA	23/11/2023	
emiliano ferreira	lolo jimenez	APROBADA	23/11/2023	
nicolas buffa	lolo jimenez	APROBADA	23/11/2023	

© 2023 - MVC - [Privacy](#)

- **Funcionalidad: Enviar una solicitud de amistad (miembro)**
  - Caso de prueba: Se envía una solicitud de amistad a un miembro que no es amigo
    - Pasos: 1) Ingresar al sistema con un usuario miembro
    - 2) Navegar a la pantalla “Listado de miembros”
    - 3) Clickear el botón “Enviar solicitud de amistad” para algún miembro
  - Datos: - usuario: lolo@gmail.com
  - contraseña: 123
  - Resultado esperado: Se redirige al usuario a la misma pantalla de listado de miembros, se crea la solicitud de amistad en estado “Pendiente Aprobación”, y se muestra un mensaje de éxito, en color verde, que dice “Solicitud enviada con éxito”.
  - Resultado obtenido: LOGRADO.
  - Evidencia:

### Listado de miembros

Email	Nombre	Apellido	Fecha de nacimiento	Es amigo	Acciones
nicolas.buffa@gmail.com	nicolas	buffa	01/10/2002	si	
emi@gmail.com	emiliano	ferreira	04/01/2002	si	
block@gmail.com	block	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
jose@gmail.com	jose	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
pepe@gmail.com	pepe	Gomez	12/12/1980	no	Enviar Solicitud de amistad
lola.manyasiempre@gmail.com	lola	gonzalez	23/04/2002	no	Enviar Solicitud de amistad
roberto.lol@gmail.com	roberto	llores	10/10/2010	no	Enviar Solicitud de amistad
lucracia.robot@gmail.com	lucracia	lorenza	19/06/2007	no	Enviar Solicitud de amistad
luisa.rob@gmail.com	luisa	robb	14/08/2000	no	Enviar Solicitud de amistad
lore.progamer@gmail.com	lore	rodriguez	16/05/2005	no	Enviar Solicitud de amistad

Solicitud enviada con éxito

### Listado de miembros

Email	Nombre	Apellido	Fecha de nacimiento	Es amigo	Acciones
nicolas.buffa@gmail.com	nicolas	buffa	01/10/2002	si	
emi@gmail.com	emiliano	ferreira	04/01/2002	si	
block@gmail.com	block	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
jose@gmail.com	jose	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
pepe@gmail.com	pepe	Gomez	12/12/1980	no	Enviar Solicitud de amistad
lola.manyasiempre@gmail.com	lola	gonzalez	23/04/2002	no	Enviar Solicitud de amistad
roberto.lol@gmail.com	roberto	llores	10/10/2010	no	Enviar Solicitud de amistad
lucracia.robot@gmail.com	lucracia	lorenza	19/06/2007	no	Enviar Solicitud de amistad
luisa.rob@gmail.com	luisa	robb	14/08/2000	no	Enviar Solicitud de amistad
lore.progamer@gmail.com	lore	rodriguez	16/05/2005	no	Enviar Solicitud de amistad

- Caso de prueba: Se envía una solicitud de amistad a un miembro que ya tiene una solicitud de amistad creada
  - Pasos: 1) Ingresar al sistema con un usuario miembro
  - 2) Navegar a la pantalla “Listado de miembros”
  - 3) Clickear el botón “Enviar solicitud de amistad” para un miembro que ya tenga una solicitud de amistad
- Datos: - usuario: lolo@gmail.com
- contraseña: 123
- Resultado esperado: Se redirige al usuario a la misma pantalla de listado de miembros y se muestra un mensaje de error, en color rojo, que dice “Ya hay una solicitud ingresada en el sistema.
- Resultado obtenido: LOGRADO.
- Evidencia:

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

### Listado de miembros

Email	Nombre	Apellido	Fecha de nacimiento	Es amigo	Acciones
nicolas.buffa@gmail.com	nicolas	buffa	01/10/2002	si	
emi@gmail.com	emiliano	ferreira	04/01/2002	si	
block@gmail.com	block	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
jose@gmail.com	jose	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
pepe@gmail.com	pepe	Gomez	12/12/1980	no	Enviar Solicitud de amistad
lola.manyaxiempres@gmail.com	lola	gonzalez	23/04/2002	no	Enviar Solicitud de amistad
roberto.lol@gmail.com	roberto	llores	10/10/2010	no	Enviar Solicitud de amistad
lucracia.robot@gmail.com	lucracia	lorenza	19/06/2007	no	Enviar Solicitud de amistad
luisa.robb@gmail.com	luisa	robb	14/08/2000	no	Enviar Solicitud de amistad
lore.progamer@gmail.com	lore	rodriguez	16/05/2005	no	Enviar Solicitud de amistad

© 2023 - MVC - Privacy

MVC Home Privacy Posts Listado de miembros Listado de solicitudes Crear post Publica por texto y VA Logout

Ya hay una solicitud ingresada en el sistema

### Listado de miembros

Email	Nombre	Apellido	Fecha de nacimiento	Es amigo	Acciones
nicolas.buffa@gmail.com	nicolas	buffa	01/10/2002	si	
emi@gmail.com	emiliano	ferreira	04/01/2002	si	
block@gmail.com	block	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
jose@gmail.com	jose	gonzalez	24/06/1998	no	Enviar Solicitud de amistad
pepe@gmail.com	pepe	Gomez	12/12/1980	no	Enviar Solicitud de amistad
lola.manyaxiempres@gmail.com	lola	gonzalez	23/04/2002	no	Enviar Solicitud de amistad
roberto.lol@gmail.com	roberto	llores	10/10/2010	no	Enviar Solicitud de amistad
lucracia.robot@gmail.com	lucracia	lorenza	19/06/2007	no	Enviar Solicitud de amistad
luisa.robb@gmail.com	luisa	robb	14/08/2000	no	Enviar Solicitud de amistad
lore.progamer@gmail.com	lore	rodriguez	16/05/2005	no	Enviar Solicitud de amistad

© 2023 - MVC - Privacy



# Código fuente de la aplicación

## Usuario:

```
using Dominio.Interfaces;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dominio
{
    public abstract class Usuario : IValidable
    {
        protected string _email;
        private string _contrasenia;

        public Usuario(string email, string contrasenia)
        {
            _email = email;
            _contrasenia = contrasenia;
        }

        public Usuario() {}

        public string Email
        {
            get { return _email; }
            set { _email = value; }
        }
    }
}
```

```

public string Contraseña
{
    get { return _contraseña; }
    set { _contraseña = value; }
}

private void ValidarEmail()
{
    if (string.IsNullOrEmpty(_email)) throw new Exception("El email no puede ser vacío");
}

private void ValidarContraseña()
{
    if (string.IsNullOrEmpty(_contraseña)) throw new Exception("La contraseña no puede ser vacía");
}

public virtual void Validar()
{
    ValidarEmail();
    ValidarContraseña();
}

public abstract string Tipo();

public override bool Equals(object? obj)
{
    Usuario user = obj as Usuario;
    return user != null && _email.Equals(user._email);
}
}

```

## Administrador:

using System;

namespace Dominio

{

public class Administrador : Usuario

{

public Administrador(string email, string contrasenia) : base(email, contrasenia)

{

}

public override void Validar()

{

base.Validar();

}

public override string Tipo()

{

return "Admin";

}

public override string ToString()

{

return \$"Admin: {\_email}";

}

}

}

## Miembro:

```
using System;
```

```
using System.Security.Cryptography;
```

```
using System.Xml.Linq;
```

```
namespace Dominio
```

```
{
```

```
    public class Miembro : Usuario, IComparable<Miembro>
```

```
    {
```

```
        private string _nombre;
```

```
        private string _apellido;
```

```
        private DateTime _fechaNacimiento;
```

```
        private List<Miembro> _amigos;
```

```
        private bool _estaBloqueado;
```

```
        public Miembro(string email, string contrasenia, string nombre, string apellido, DateTime fechaNacimiento) : base(email, contrasenia)
```

```
        {
```

```
            _nombre = nombre;
```

```
            _apellido = apellido;
```

```
            _fechaNacimiento = fechaNacimiento;
```

```
            _amigos = new List<Miembro>();
```

```
            _estaBloqueado = false;
```

```
        }
```

```
        public Miembro(string email, string contrasenia, string nombre, string apellido, DateTime fechaNacimiento, bool estaBloqueado) : base(email, contrasenia)
```

```
        {
```

```
            _nombre = nombre;
```

```
            _apellido = apellido;
```

```
    _fechaNacimiento = fechaNacimiento;
    _amigos = new List<Miembro>();
    _estaBloqueado = estaBloqueado;
}
```

```
public Miembro() {}
```

```
public bool EstaBloqueado
{
    get { return _estaBloqueado; }
}
```

```
public string Nombre
{
    get { return _nombre; }
    set { _nombre = value; }
}
```

```
public string Apellido
{
    get { return _apellido; }
    set { _apellido = value; }
}
```

```
public DateTime FechaNacimiento
{
    get { return _fechaNacimiento; }
    set { _fechaNacimiento = value; }
}
```

```
public void ValidarMiembroNoEsAmigo(Miembro m)
{
    if (_amigos.Contains(m)) throw new Exception("El usuario solicitante ya esta en la lista de amigos del solicitado");
}

public bool EsAmigo(Miembro m)
{
    bool retorno = false;
    if (_amigos.Contains(m)) retorno = true;
    return retorno;
}

public void AgregarAmigo(Miembro m)
{
    _amigos.Add(m);
}

private void ValidarNombre()
{
    if (string.IsNullOrEmpty(_nombre)) throw new Exception("El nombre no puede ser nulo");
}

private void ValidarApellido()
{
    if (string.IsNullOrEmpty(_apellido)) throw new Exception("El apellido no puede ser nulo");
}
```

```
private void ValidarFechaNacimiento()
{
    if (_fechaNacimiento == null) throw new Exception("La fecha de nacimiento no puede ser null");
}

public override void Validar()
{
    base.Validar();
    ValidarNombre();
    ValidarApellido();
    ValidarFechaNacimiento();
}

public void Bloquear()
{
    _estaBloqueado = true;
}

public void Desbloquear()
{
    _estaBloqueado = false;
}

public override string Tipo()
{
    return "Miembro";
}
```

```

public override string ToString()
{
    return $"Usuario: {_nombre} {_apellido}";
}

public int CompareTo(Miembro? other)
{
    int orden = _apellido.CompareTo(other._apellido);
    if (orden == 0) orden = _nombre.CompareTo(other._nombre);
    return orden;
}
}
}

```

## Publicacion:

```

using Dominio.Interfaces;
using System;
namespace Dominio
{
    public abstract class Publicacion : IValidable
    {
        private static int s_ultID = 1;
        protected int _id;
        protected string _titulo;
        protected Miembro _creador;
        protected DateTime _fechaCreacion;
        protected string _contenido;
        protected string _texto;
        protected Privacidad _privacidad;
        protected List<Reaccion> _reacciones;
    }
}

```



/\*Se pasa la fecha de creación por el constructor para así poder hacer mejores pruebas para el punto 4 del obligatorio\*/

```
public Publicacion(string titulo, DateTime fechaCreacion, Miembro creador, Privacidad
privacidad ,string texto, string contenido)
```

```
{
    _id = s_ultID;
    s_ultID++;
    _texto = texto;
    _fechaCreacion = fechaCreacion;
    _creador = creador;
    _privacidad = privacidad;
    _titulo = titulo;
    _contenido = contenido;
    _reacciones = new List<Reaccion>();
}
```

```
public Publicacion()
{
    _id = s_ultID;
    s_ultID++;
    _reacciones = new List<Reaccion>();
}
```

```
public string Contenido
{
    get { return _contenido; }
    set { _contenido = value; }
}
```

```
public string Texto
{
    get { return _texto; }
    set { _texto = value; }
```

```
}

public string Titulo
{
    get { return _titulo; }
    set { _titulo = value; }
}

public Privacidad Privacidad
{
    get { return _privacidad; }
    set { _privacidad = value;}
}

public DateTime FechaCreacion
{
    get { return _fechaCreacion; }
    set { _fechaCreacion= value; }
}

public List<Reaccion> Reacciones
{
    get { return _reacciones; }
    set { _reacciones = value;}
}

public Miembro Creador
{
    get { return _creador; }
    set { _creador = value; }
}
```

```
public int Id
{
    get { return _id; }
    set { _id = value; }
}

private void ValidarTitulo()
{
    if (string.IsNullOrEmpty(_texto) || _texto.Length < 3) throw new Exception("el texto no
puede ser vacio ni menor a 3 caracteres");
}

private void ValidarContenido()
{
    if (string.IsNullOrEmpty(_contenido)) throw new Exception("el texto no puede ser
vacio");
}

private void ValidarTexto()
{
    if (string.IsNullOrEmpty(_texto)) throw new Exception("el texto no puede ser vacio");
}

private void ValidarMiembro()
{
    if (_creador == null) throw new Exception("el creador no puede ser vacio");
}
```

```
public virtual void Validar()
```

```
{
```

```
    ValidarMiembro();
```

```
    ValidarTexto();
```

```
    ValidarContenido();
```

```
    ValidarTitulo();
```

```
}
```

```
public int CantidadReaccion(TipoReaccion reaccion)
```

```
{
```

```
    int contador = 0;
```

```
    foreach (Reaccion r in _reacciones)
```

```
    {
```

```
        if (r.TipoReaccion == reaccion) contador++;
```

```
    }
```

```
    return contador;
```

```
}
```

```
public virtual int CalcularVA()
```

```
{
```

```
    int cantLikes = CantidadReaccion(TipoReaccion.LIKE);
```

```
    int cantDislikes = CantidadReaccion(TipoReaccion.DISLIKE);
```

```
    return (cantLikes * 5) + (cantDislikes * -2);
```

```
}
```

```
public abstract string Tipo();
```

```

    public override bool Equals(object? obj)
    {
        Publicacion p = obj as Publicacion;
        return p != null && this._id.Equals(p._id);
    }
}

```

## Post:

```

using System;

using System.Security.Cryptography;

namespace Dominio
{
    public class Post : Publicacion, IComparable<Post>
    {
        private string _imagen;

        private List<Comentario> _comentarios;

        private bool _estaCensurado;

        private bool _textoMenorA50; /*Propiedad hecha para el punto 4 del obligatorio*/

        public Post(string titulo, DateTime fechaCreacion, Miembro creador, string imagen,
Privacidad privacidad , bool estaCensurado, string texto, string contenido) : base(titulo,
fechaCreacion, creador, privacidad ,texto, contenido)
        {
            _imagen = imagen;

            _estaCensurado = estaCensurado;

            _comentarios = new List<Comentario>();

            _textoMenorA50 = false;
        }
    }
}

```

```
public Post()
{
    _comentarios = new List<Comentario>();
    _textoMenorA50 = false;
}
```

```
public string Imagen
{
    get { return _imagen; }
    set { _imagen = value; }
}
```

```
public bool TextoMenorA50
{
    get { return _textoMenorA50; }
}
```

```
public List<Comentario> Comentarios
{
    get { return _comentarios; }
    set { _comentarios = value; }
}
```

```
public bool EstaCensurado
{
    get { return _estaCensurado; }
    set { _estaCensurado = value; }
}
```

```

private void ValidarImagen() /*Valida si la extension es jpg o png o jpeg*/
{
    if (string.IsNullOrEmpty(_imagen)) throw new Exception("La imagen no puede ser nula");

    string[] words = _imagen.Split('.');

    if (words.Length != 2) throw new Exception("El nombre del archivo es erroneo");

    if (words[1] != "jpg" && words[1] != "png" && words[1] != "jpeg") throw new
Exception("La extension de la imagen es invalida");
}

public override void Validar()
{
    base.Validar();

    ValidarImagen();
}

public void AgregarComentario(Comentario c)
{
    _comentarios.Add(c);
}

/*Metodo hecho para el punto 4, asi poder devolver solo los primeros 50 caracteres de la
propiedad texto*/
private string RetornarTexto()
{
    if (_textoMenorA50)
    {
        if (_texto.Length > 50)
        {
            string textoCorto = _texto.Substring(0, 50);

            return textoCorto;
        }
    }
}

```

```

        }

        else return _texto;

    }

    else return _texto;

}

/*Metodo usado solo en el punto 4 para indicar que el texto tiene que acortarse a 50
caracteres*/

public void ModificarLongitudDeTexto()
{
    _textoMenorA50 = !_textoMenorA50;
}

public override string ToString()
{
    return $"Tipo: Post - Id: {_id} - Fecha de cracion: {_fechaCreacion} - Titulo: {_titulo} -
Texto: {RetornarTexto()}";
}

public void Censurar()
{
    _estaCensurado = true;
}

public void QuitarCensura()
{
    _estaCensurado = false;
}

public override int CalcularVA()
{
    return _privacidad == Privacidad.Publico ? base.CalcularVA() + 10 : base.CalcularVA();
}

```



```

    }

    public override string Tipo()
    {
        return "Post";
    }

    /*Metodo CompareTo hecho para que se compare por titulo*/
    public int CompareTo(Post? other)
    {
        return _titulo.CompareTo(other._titulo) * -1;
    }
}
}

```

## Comentario:

```

using System;

namespace Dominio
{
    public class Comentario : Publicacion
    {

        public Comentario(string titulo, DateTime fechaCreacion, Miembro creador, Privacidad
privacidad, string texto, string contenido) : base(titulo, fechaCreacion, creador, privacidad,
texto, contenido)
        {
        }

        public override string Tipo()
        {
            return "Comentario";
        }
    }
}

```



```
/*Cuando se quiera mostrar un comentario, tambien se mostraran las reacciones en el*/  
public override string ToString()  
{  
    string retorno = $"Tipo: Comentario - Id: {_id} - Titulo: {_titulo} - Creador: {_creador} -  
Privacidad: {_privacidad}";  
    if (_reacciones.Count != 0)  
    {  
        retorno += "\n Reacciones en el comentario:";  
        foreach (Reaccion r in _reacciones)  
        {  
            retorno += $" \n  {r}";  
        }  
    }  
    return retorno;  
}  
}  
}
```

## Invitacion:

```
using Dominio.Interfaces;

using System;

namespace Dominio
{
    public class Invitacion : IValidable
    {
        private static int s_ultId = 1;
        private int _id;
        private Miembro _miembroSolicitante;
        private Miembro _miembroSolicitado;
        private Estado _estado;
        private DateTime _fecha;

        /* Se inicializa el estado en PENDIENTE_APROBACION desde el constructor */
        public Invitacion(Miembro miembroSolicitante, Miembro miembroSolicitado)
        {
            _id = s_ultId;
            s_ultId++;

            _miembroSolicitante = miembroSolicitante;
            _miembroSolicitado = miembroSolicitado;
            _estado = Estado.PENDIENTE_APROBACION;
            _fecha = DateTime.Now;
        }

        public DateTime Fecha
        {
            get { return _fecha; }
        }
    }
}
```

```
public Miembro MiembroSolicitante
{
    get { return _miembroSolicitante; }
}

public Miembro MiembroSolicitado
{
    get { return _miembroSolicitado; }
}

public int ID
{
    get { return _id; }
}

public Estado Estado { get { return _estado; } }

private void ValidarMiembroSolicitado()
{
    if (_miembroSolicitado == null) throw new Exception("El miembro solicitado no puede
ser nulo");
}

private void ValidarMiembroSolicitante()
{
    if (_miembroSolicitante == null) throw new Exception("El miembro solicitante no puede
ser nulo");
}
```

```

/*Metodo que valida una invitacion, y todo lo que requiere para que sea valida*/
private void ValidarInvitacion()
{
    if (_miembroSolicitante.Equals(_miembroSolicitado)) throw new Exception("El miembro
solicitado es igual al solicitante");

    _miembroSolicitante.ValidarMiembroNoEsAmigo(_miembroSolicitado);

    if (_estado != Estado.PENDIENTE_APROBACION) throw new Exception("El estado de la
solicitud debe de estar en pendiente aprobacion");
}

public void Validar()
{
    ValidarMiembroSolicitante();
    ValidarMiembroSolicitado();
    ValidarInvitacion();
}

public void AceptarInvitacion(Miembro solicitado)
{
    ValidarInvitacion();

    _estado = Estado.APROBADA;
    _miembroSolicitante.AgregarAmigo(solicitado);
    solicitado.AgregarAmigo(_miembroSolicitante);
}

public void RechazarInvitacion()
{
    ValidarInvitacion();

    _estado = Estado.RECHAZADA;
}

```

```
        public override string ToString()
        {
            return $"Id: {_id} - Miembro solicitante: {_miembroSolicitante} - Miembro solicitado: {_miembroSolicitado}";
        }
    }
}
```

## Estado:

```
using System;

namespace Dominio
{
    public enum Estado
    {
        PENDIENTE_APROBACION,
        APROBADA,
        RECHAZADA
    }
}
```

## Privacidad:

```
using System;

namespace Dominio
{
    public enum Privacidad
    {
        Publico,
        Privado,
    }
}
```

## TipoReaccion:

```
using System;

namespace Dominio
{
    public enum TipoReaccion
    {
        LIKE,

        DISLIKE
    }
}
```

## Reaccion:

```
using Dominio.Interfaces;

using System;

namespace Dominio
{
    public class Reaccion : IValidable
    {
        private TipoReaccion _tipoReaccion;
        private Miembro _creador;

        public Reaccion(TipoReaccion tipoReaccion, Miembro creador)
        {
            _tipoReaccion = tipoReaccion;
            _creador = creador;
        }

        public Miembro Creador
        {
            get { return _creador; }
        }
    }
}
```



```
}

public TipoReaccion TipoReaccion
{
    get { return _tipoReaccion; }
}

private void ValidarCreador()
{
    if (_creador == null) throw new Exception("El creador no puede ser nulo");
}

public void Validar()
{
    ValidarCreador();
}

public override string ToString()
{
    return $"Creador: {_creador} - Reaccion: {_tipoReaccion}";
}
}
}
```