

**Autor:**

Rubén Cristóbal Vergara Rodríguez;  
rubenverrod@gmail.com.

**Nombre del proyecto:**

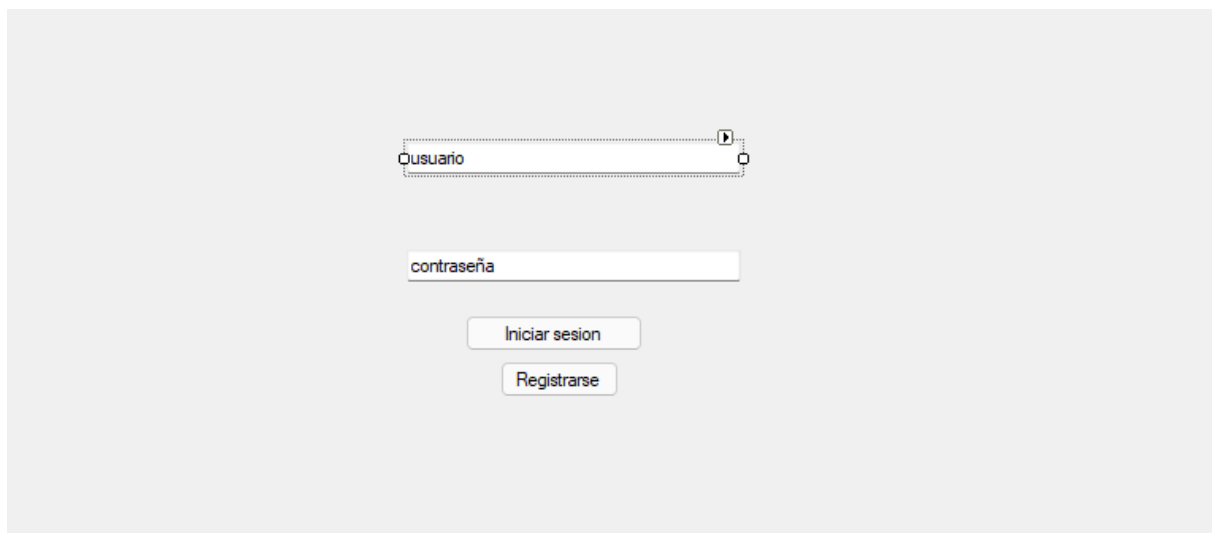
ChatroomWithUserIdentification;  
<https://github.com/FrenzyR/ChatroomWithUserIdentification>.

**Plataforma destino:**

Windows 10 y 11; Usará Servidor de Windows y Windows Forms.

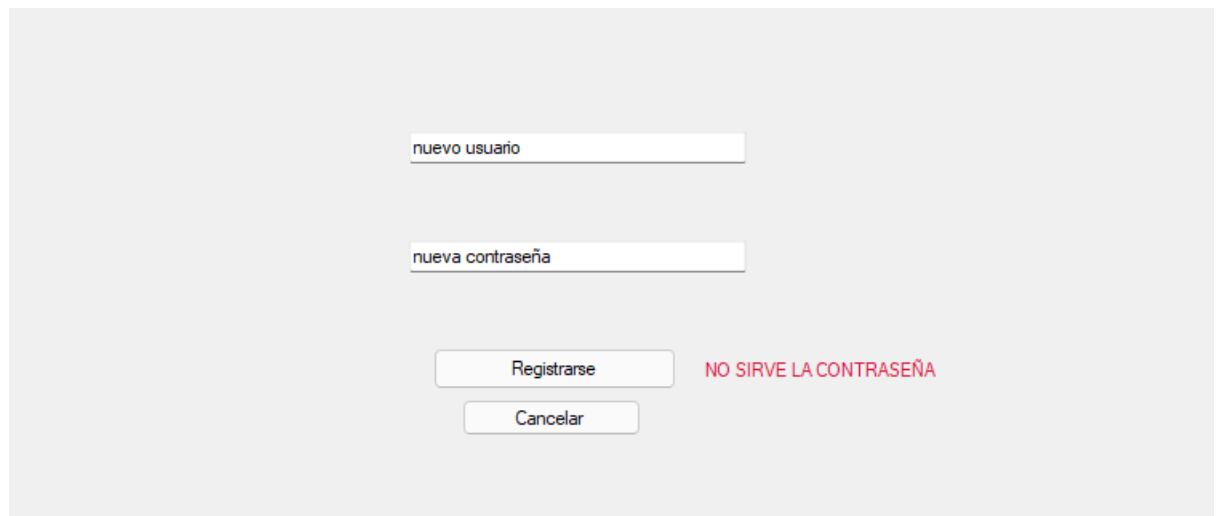
**Especificación:**

El cliente, al abrir el ejecutable, lo primero que verás será un formulario de bienvenida que te preguntará por tu nombre y usuario con dos textboxes y un botón.



The image shows a simple login interface on a light gray background. It features two text input fields: the top one is labeled 'Usuario' and has a small cursor icon at the end; the bottom one is labeled 'contraseña'. Below these fields are two buttons: 'Iniciar sesion' and 'Registrarse', both with rounded rectangular borders.

Si alguno de estos dos no se encuentra en la base de datos, te dará mensaje de error.



A registration form on a light gray background. It features two input fields: the first is labeled 'nuevo usuario' and the second is labeled 'nueva contraseña'. Below these fields are two buttons: 'Registrarse' and 'Cancelar'. To the right of the 'Registrarse' button, there is a red error message that reads 'NO SIRVE LA CONTRASEÑA'.

nuevo usuario

nueva contraseña

Registrarse

Cancelar

NO SIRVE LA CONTRASEÑA

Si todo está correcto entrarás a la sala de chat (donde se pueden conectar varios usuarios). Al mandar un mensaje se enseñará tu mensaje al resto de usuarios con el formato 'nombre@usuario: mensaje', y para ti 'nombre:mensaje' y viceversa. Puedes salir del chatroom con un button que pone 'Salir', esto informará al resto de usuarios que 'nombre@usuario' ha salido.

pep0910: anybody there?

pep0910: hello?

you: nope, just me

pep0910: damn.. sooo how's life been treating you lately?

you: very nice, thank you for asking

*DEEPFANN#1 se ha unido*

pep0910: oh ney nice to see you there, wanna play some deepwoken?

ummmm...

Enviar:

**cooldude101**

Salir

Además te dará la opción con otro button de crear/registrarse, que lanzará un formulario modal con dos textBoxes y dos buttons (Cancelar y aceptar con sus respectivos comportamientos). Si aceptas con un usuario que no exista te creará un usuario.

## Servidor:

La **base de datos** será sencilla (de nombre 'users'), creada en la última versión de SQL y en MariaDB, tendrá una única tabla ('user') con las columnas username (varchar), password (varchar), age (small int), last\_connected(date, nullable), y un token de identificación (int) que se autoincrementa. Los únicos datos que se comprobarán para entrar serán el nombre y la contraseña.

**'Sign in':** Comprueba que los parámetros que se le han pasado están dentro de la base de datos, si comprueba que existe se devolverá true, en el caso contrario false. Está dentro de la clase "BasicDBFuncitonality".

**'Sign up':** Comprobará que no exista anteriormente con el parámetro username, en caso de que no existe, se añadirá con un nuevo id de forma automática (se autoincrementa), y con el resto de parámetros pasados. (password, age...)

Los parámetros que se le pasa son username, age, y password. Last\_connected y id se generan automáticamente, el primero cuando se inicia sesión y el id cuando se añade a la base de datos. Al igual que 'sign in', se encuentra en "BasicDBFuncitonality".

**'Chatroom':** Recibe mensajes del cliente y los devuelve a todos al cliente original con streamreader y streamwriter. Al salir un cliente avisa al resto de clientes lo que ha ocurrido. Solo se cierra desde el panel de Windows.












Tiene su propia clase llamada "ChatServer"

## Cliente:

El botón de iniciar sesión tendrá la funcionalidad ligada de **'sign in'**, comprobará si hay algo escrito en ambos textboxes, y segundo, si se encuentran en la base de datos. Si ambos casos son negativos, te devolverá un texto en un label en color rojo indicando el motivo del error. Si son positivos, creará una thread para el cliente que únicamente se cerrará (sin lanzar una excepción en el servidor) cuando presiones el botón de 'cerrar sesión'.

El botón de registrarse, tendrá la funcionalidad de **'sign up'**, que mandará los datos de los textboxes mediante el botón aceptar a una query que insertará un usuario a la base de datos, si hay algún fallo no le permite hacer la acción y le informará con un label.

En el **'chatroom'** el usuario podrá mandar mensajes en la textbox inferior. Los mensajes del usuario aparecerán a la derecha y los del resto de usuarios a la izquierda. Su tamaño dependerá de lo largo que sea el mensaje mandado.

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	username	varchar(15)	utf8mb4_general_ci		No	Ninguna			 Cambiar  Eliminar Más
<input type="checkbox"/>	2	password	varchar(20)	utf8mb4_general_ci		No	Ninguna			 Cambiar  Eliminar Más
<input type="checkbox"/>	3	age	smallint(6)			Si	NULL			 Cambiar  Eliminar Más
<input type="checkbox"/>	4	last_connected	date			No	Ninguna			 Cambiar  Eliminar Más
<input type="checkbox"/>	5	id 	int(11)			No	Ninguna		AUTO_INCREMENT	 Cambiar  Eliminar Más

El **chatroom**, para el cliente, es el formulario que ven y con el que interactúan.

pep0910: anybody there?

pep0910: hello?

you: nope, just me

pep0910: damn.. sooo how's life been treating you lately?

very nice, thank you for asking

Enviar:

cooldude101

Salir

## Tecnologías usadas:

Será multihilo y usará SQL, Linq, MariaDB, DBeaver(CE 22.3.0) y Project Rider de JetBrains(latest).

## Pasos(Planificación temporal):

TASK NAMES	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7
SIGN IN SERVER							
SIGN UP SERVER							
BACKGROUND SERVER							
CHATROOM SERVER							
SIGN IN FORM							
SIGN UP FORM							
CHATROOM FORM							