

# Caso de uso:

Login por consola, el usuario proporciona sus credenciales (nombre de usuario y contraseña) y si son correctos accede al sistema.

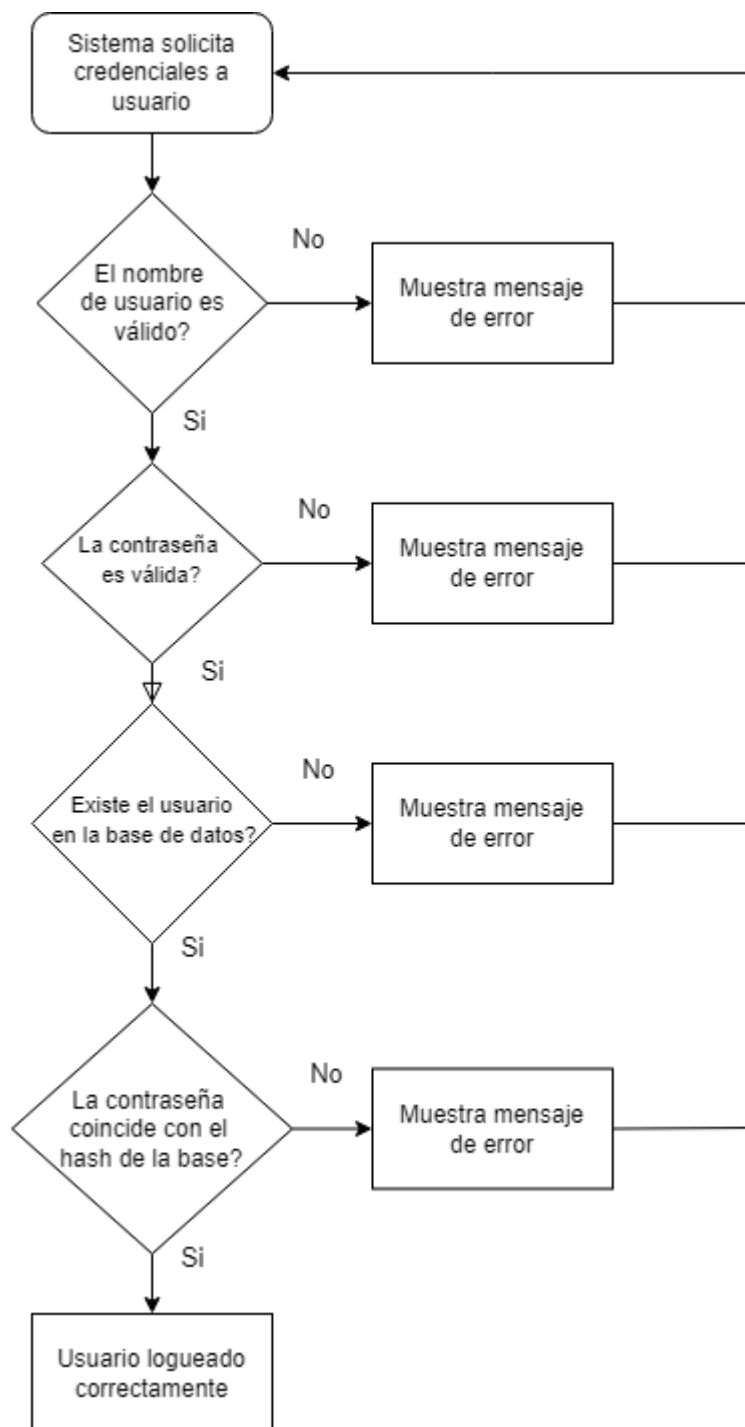
Restricciones:

- El nombre de usuario debe ser un email válido
- La contraseña debe cumplir las siguientes condiciones:
  - Ser alfanumérica
  - Ser mayor a 8 caracteres
  - Ser diferente del nombre de usuario

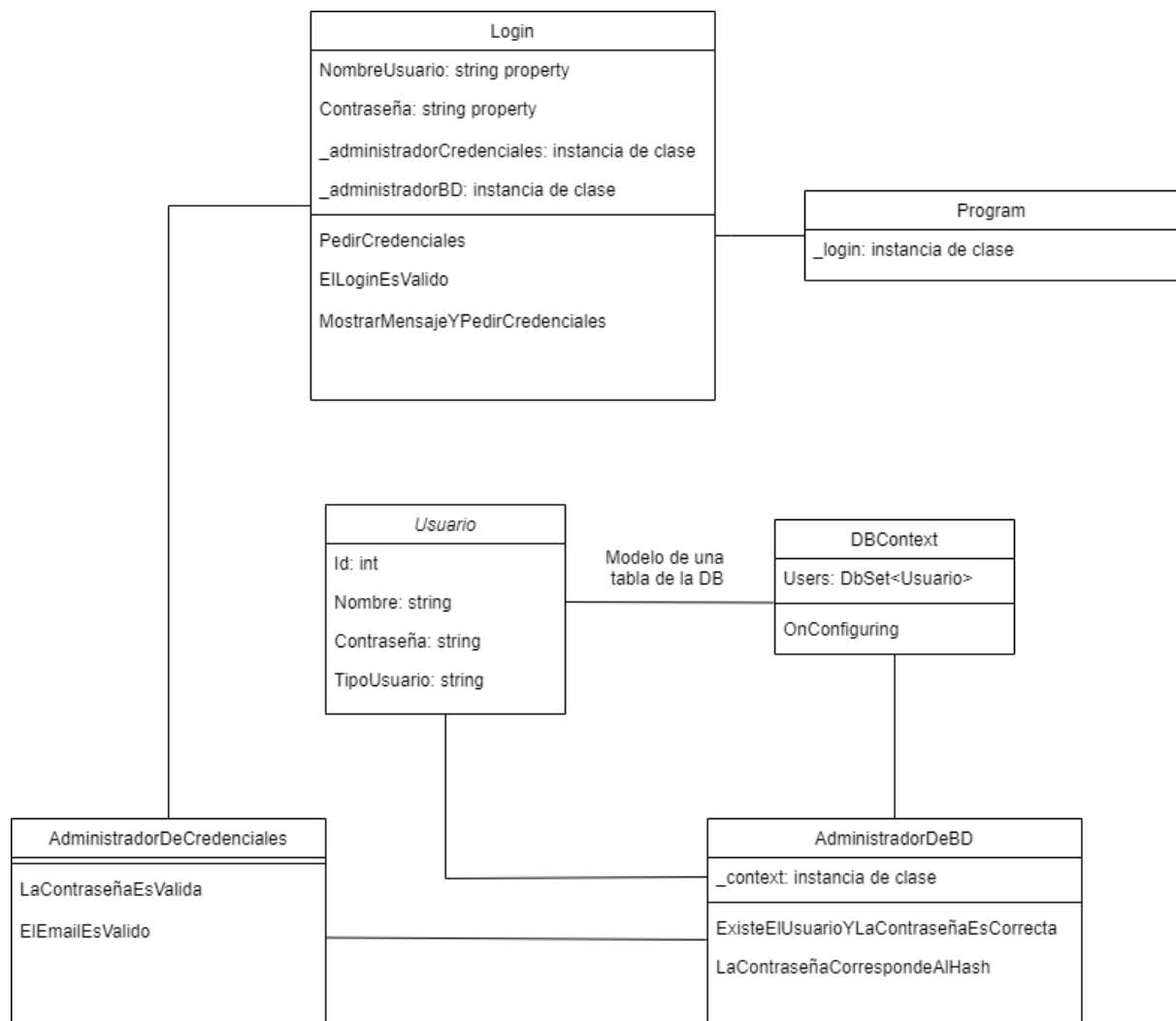
Nombre del caso de uso	LOGIN
Descripción	Se solicitan 2 datos, nombre de usuario (en este caso debe ser un email) y la contraseña del mismo
Flujo principal	<ol style="list-style-type: none"><li>1.Sistema solicita credenciales</li><li>2. Valida que el email a sea válido</li><li>3. Valida que la contraseña sea válida</li><li>3. Valida que el usuario exista en la base de datos</li><li>4. Valida que la contraseña proporcionada por el usuario coincida con el hash de la misma en la BD.</li><li>5. Se concede el acceso al sistema</li><li>6. Fin del caso de uso</li></ol>
Flujo alternativo A1	<ol style="list-style-type: none"><li>2.1. El email no es válido</li><li>2.2. Se despliega un mensaje de error</li><li>2.3. Se vuelven a solicitar las credenciales</li></ol>
Flujo alternativo A2	<ol style="list-style-type: none"><li>3.1. La contraseña no es válida</li><li>3.2. Se despliega un mensaje de error</li></ol>

	3.3. Se vuelven a solicitar las credenciales
Flujo alternativo A3	4.1. El usuario no existe 4.2. Se despliega un mensaje de error 4.3. Se vuelven a solicitar las credenciales
Flujo alternativo A4	5.1. La contraseña es incorrecta 5.2. Se despliega un mensaje de error 5.3. Se vuelven a solicitar las credenciales

Diagrama de flujos del caso de uso:



## Diagrama de clases:



La clase **Program** es la clase main del programa, su única función es llamar un método de la clase **Login** que inicia el proceso de logueo

La clase **Login** solicita las credenciales de acceso al usuario, valida que los datos ingresados por el usuario cumplan ciertos requisitos y si los cumplen valida los mismos en la base de datos.

La clase **AdministradorDeCredenciales** valida que la contraseña cumpla varios requisitos (cantidad de caracteres mayor a 8, que sea alfanumérica, etc) y lo mismo para el email del usuario, valida que sea un email válido.

La clase **AdministradorDeBD** luego de validado que los datos ingresados por el usuario cumple los requisitos se consultan las credenciales en la base.

La clase **DBContext** es el contexto de base de datos, es la comunicación del programa con la base de datos.

La clase **Usuario** es el modelo de la tabla que almacena la información de cada usuario.

# Implementación

Lenguaje: C#

Framework: .net 8.0

Gestor de base de datos: SQLite

Proyecto subido a github, una vez clonado el repositorio el programa ya está listo para ser usado.