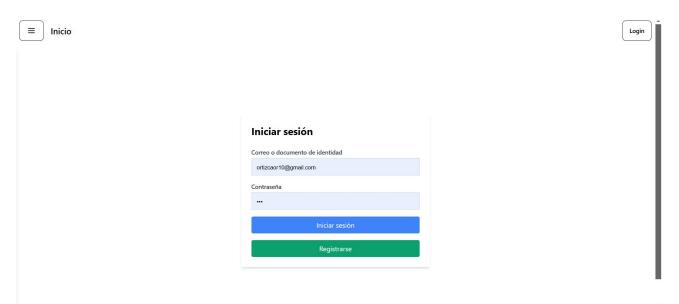
## Anexo 1. Documento con capturas de pantalla del código de la interfaz y su código de programación

1. Formulario de inicio de sesion.



2. Codigo de programacion del componente

```
Archivo Editar Selección Ver Ir Ejecutar
                                                                                                                                                                                                                                                                                                                          EXPLORADOR

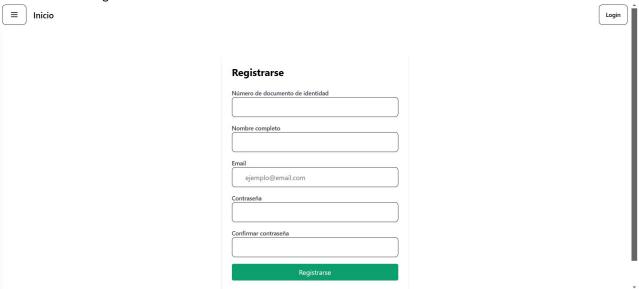
    ⊜ login.tsx ×

c
                                                                        src>pages > auth > @ logintsx > ...
1 import { signIn, useSession } from 'next-auth/react';
2 import Link from 'next/link';
3 import { useRouter } from 'next/router';
4 import { FormEvent, useState, useEffect } from 'react';

✓ REPORTS

          ✓ prisma> migrations♠ schema.prisma
                                                                                      const Login = () => {
  const [errorMessage, setErrorMessage] = useState<string | null>(null);
  const router = useRouter();
  const { data: session, status } = useSession(); // Usa el hook useSession para verificar la sesión
             > components
                                                                                        useEffect(() => {
    if (status === 'authenticated') {
        router.push('/');
    }
                                                                                          const handletogin = async (e: FormEvent<HTMLFormElement>) => {
    e.preventDefault();
               const form = e.currentTarget;
const identifier = (form.elements.namedItem('identifier') as HTMLInputElement).value;
const password = (form.elements.namedItem('password') as HTMLInputElement).value;
                                                                                             const result = await signIn('credentials', {
  redirect: false,
  identifier,
             TS middleware.ts
            .eslintrc.json
                                                                                              if (result && result.error) {
   console.error('Error en el inicio de sesión:', result.error); // Log del error
   setErrorMessage(result.error);
```

3. Formulario de registro de usuarios



4. Codigo de programacion del componente

```
刘 Archivo Editar Selección Ver Ir Ejecutar …
                                                                                                                                                                                                                                                              а ш ...
Ф
         EXPLORADOR
                                     C; P; O ⊕ src > pages > auth > ⊕ register.tsc > [00] Register > [00] onSubmit > ⊕ handleSubmit() callback

1 import { useState, FC } from 'react';

2 import { useFourter } from 'next/router';

M import { userom } from 'react-hook-form';

4 import { user } from '@/model/user.props';
        ∨ REPORTS
         ✓ prisma
> migrations

    schema.prisma
90
42
                                            interface UserFormProps {
  handleCreate: () => void;
  handleClose: () => void;
  initialData?: User;
           > components
> model
           حلك
            document.tsx
         .env eslintrc.json
         gitignorenext-env.d.ts
                                                                         const onSubmit = handleSubmit(async (data: User) => {
    if (data.password !== passwordConfirm) {
        setError("password", { type: "manual", message: "Las contraseñas no coinciden." });
    return;
}
          JS next.config.mjs
         {} package.json
                                                                            const userData = {
  email: data.email,
```