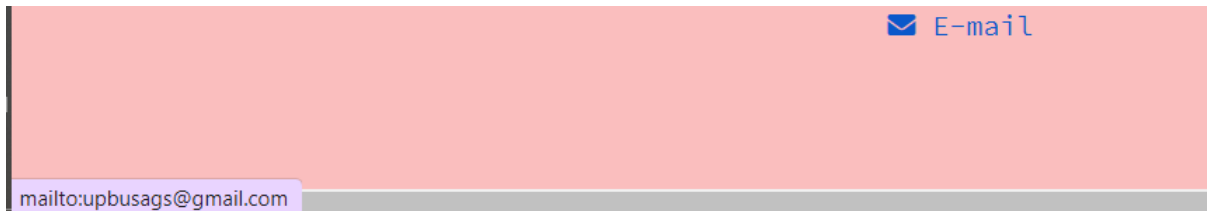


## UPBus

->READ.me (redacción de las descripciones y uso del proyecto)

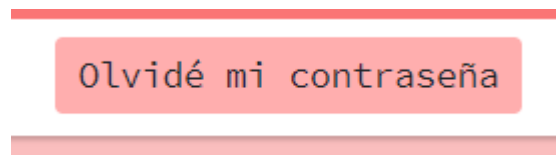
->Corrección en "iniciar sesión" y "registrarse" //pedía correo en el espacio de usuario en lugar de texto en general

->Investigación del funcionamiento de mailto, añadido en cada footer "E-mail"

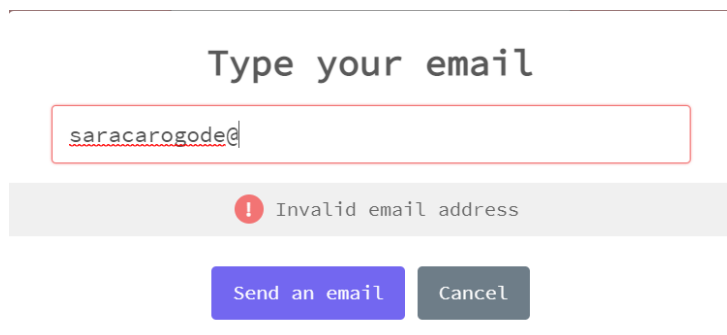


->Investigación de ejemplo específico de sweet alert anidados para lograr la siguiente idea: (la cual también colaboré en definir)

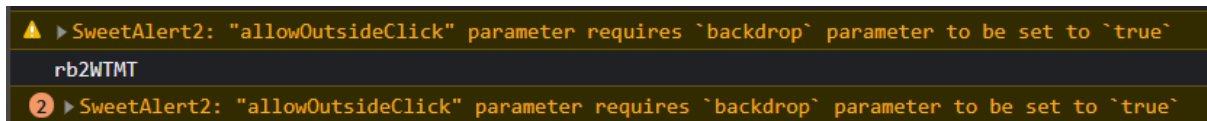
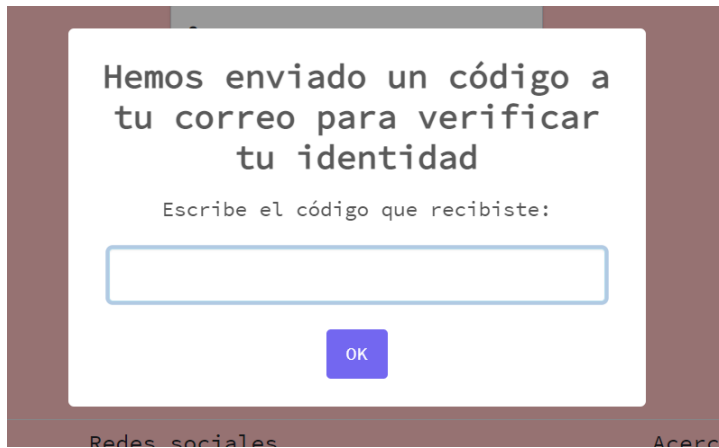
1. Usuario presiona botón de "olvidé mi contraseña"



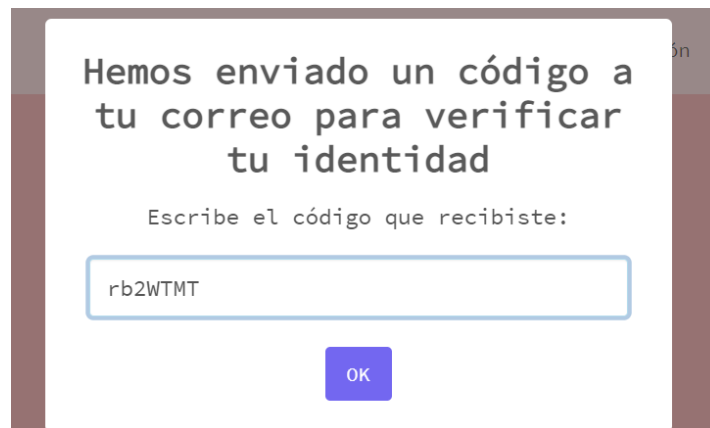
2. Se abre un sweet alert donde utilizo html para pedir como input un email (si el texto no cumple los requisitos de email, no continua y aparece error 'invalid email address')



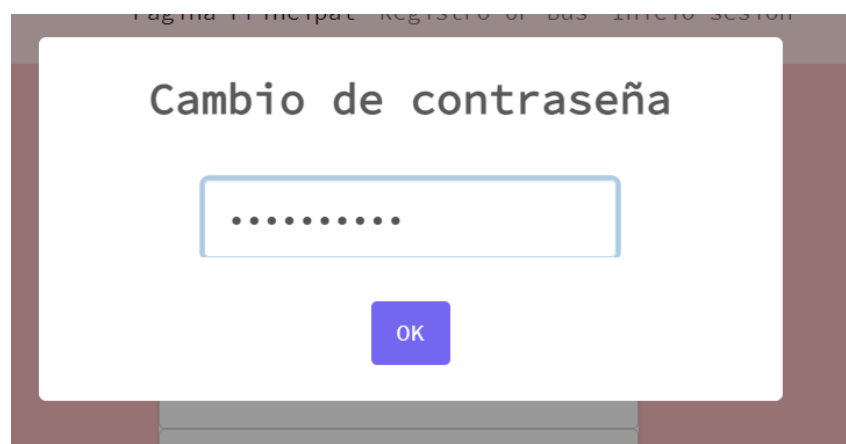
3. Se presiona el botón 'Send an email' y aparece el segundo sweet alert pidiendo el código que se generó.



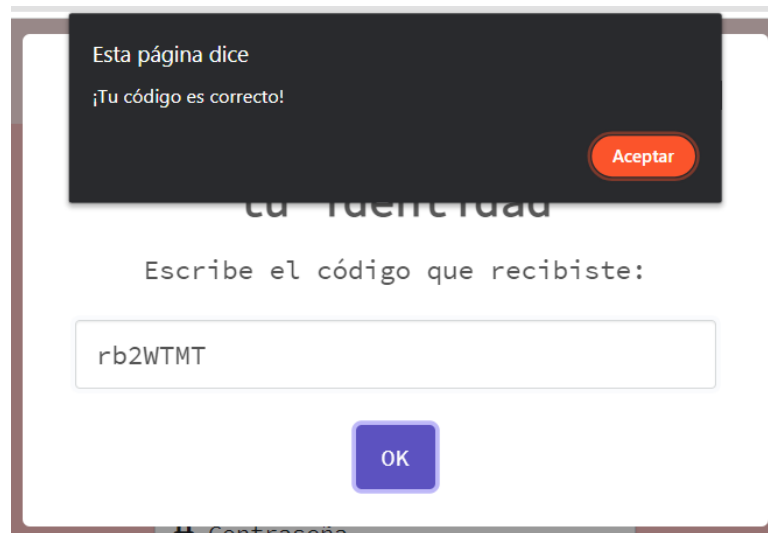
4. Usuario ingresa el código que recibió, se presiona OK



Si es correcto aparece una nueva alerta indicando que está bien, y solicita la nueva contraseña (aquí di la idea de que pidiera dos contraseñas y las comparara, pero al final no se quedó)



la contraseña se encripta con puntitos (colaboré corrigiendo eso también, ya que era visible la contraseña)



Esta página dice  
¡Tu código es correcto!

Aceptar

tu identidad

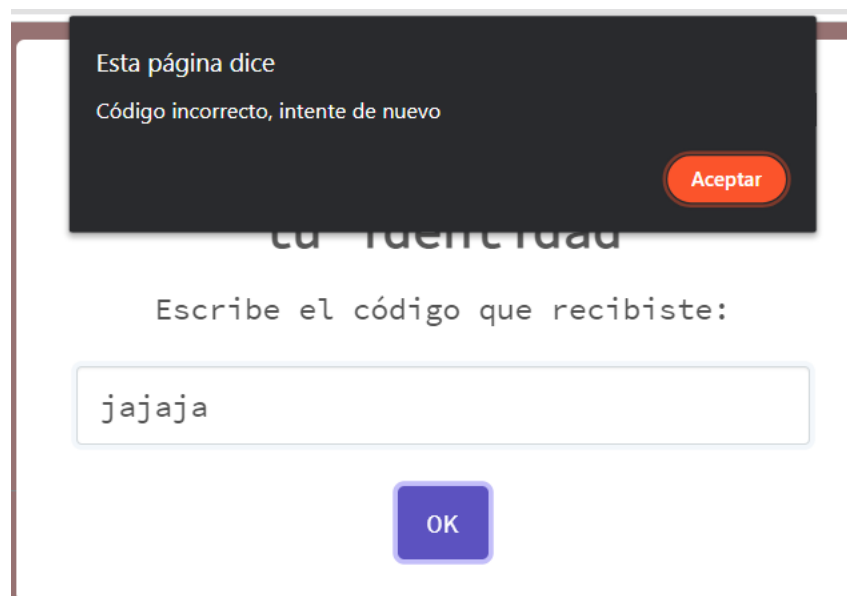
Escribe el código que recibiste:

rb2WTMT

OK

Contraseña

De lo contrario, aparece otra alerta que indica que está mal



Esta página dice  
Código incorrecto, intente de nuevo

Aceptar

tu identidad

Escribe el código que recibiste:

jajaja

OK

->Bastante tiempo investigando e intento satisfactorio de mandar un correo a través de nodemailer (sólo que no pude conectar el js con html)

Pasos para ver el funcionamiento:

1. abrir el archivo forgotpassword.js ubicado en \UPBus\Front-End
2. Modificar solamente el correo deseado en "to: tucorreo@gmail.com"

```
97 < async function run() {
98 <   let enviarMensaje = await upbusTransport.sendMail({
99 <     from: "UpBus <upbusags@gmail.com>",
100 <     to: "0226594@up.edu.mx", //sacar ese email de la base de datos!!
101 <     //to: `${user_mail}`,
102 <     subject: "¿Olvidaste tu contraseña de UpBus?",
103 <     text: "Hola, este es un correo de UpBus, tu código para recuperar tu contraseña es: "+ codigo,
104 <     html: "<body><h1>Hola!, esto es un correo de UpBus</h1></body>" + codigo,
105 <   })
106 <   console.log(enviarMensaje);
107 < };
108 <
109 < run().catch(err => console.error(err));|
110 <
```

3. *terminal visual*: npm i dotnev

4. *terminal visual*: node forgotpassword.js. //Esto generará un código aleatorio (la generación aleatoria la hizo aline), además de mostrar en consola cómo se vería el correo

Se  
recibirá  
correo  
siguiente mensaje (quizá esté en spam)

```
TERMINAL OUTPUT
PS C:\Users\scago\Documents\GitHub\UPBus\Front-End> node .\forgotpassword.js
XGMNIWq
{
  accepted: [ '0226594@up.edu.mx' ],
  rejected: [],
  envelopeTime: 390,
  messageTime: 512,
  messageSize: 780,
  response: '250 2.0.0 OK 1635629592 n26sm2744645oie.51 - smtp',
  envelope: { from: 'upbusags@gmail.com', to: [ '0226594@up.edu.mx' ] },
  messageId: '<4f26431b-ac5b-f1e8-f958-86dee53d41bf@gmail.com>'
}
PS C:\Users\scago\Documents\GitHub\UPBus\Front-End> |
```

un  
con el

¿Olvidaste tu contraseña de UpBus?

Externo

Spam x



UpBus <upbusags@gmail.com>

para mí ▾



inglés ▾



español ▾

[Traducir mensaje](#)

**Hola!, esto es un correo de UpBus**

XGMNIWq

← Responder

→ Reenviar



- >Colaboración con el equipo, intentando coordinar el trabajo en equipo.
- >Scrum master las 3 semanas intentando estar al pendiente de cómo iba el proyecto
- >Actualización de Trello
- >Creación del repositorio general //Aquí se nos complicó para evitar deshacer el trabajo de los demás