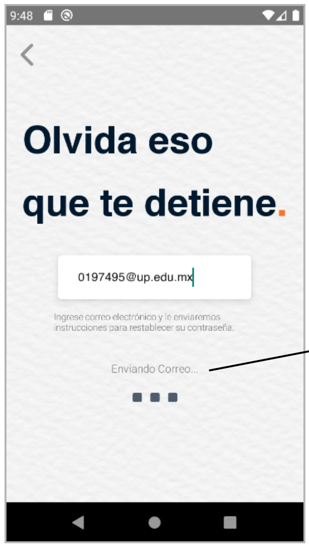
ForgotPasswordScreen



9:48

Olvida eso

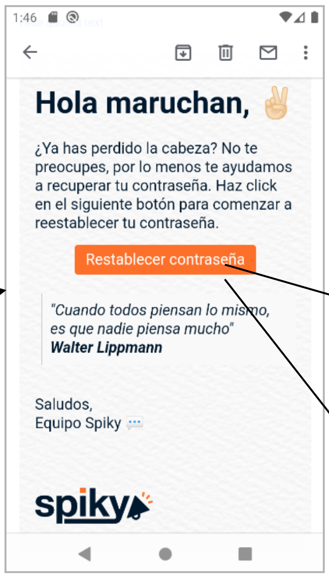
que te detiene.

0197495@up.edu.mx

Ingrese correo electrónico y le enviaremos instrucciones para restablecer su contraseña.

Enviando Correo...

user goes to email



1:46

Hola maruchan, ✌

¿Ya has perdido la cabeza? No te preocupes, por lo menos te ayudamos a recuperar tu contraseña. Haz click en el siguiente botón para comenzar a reestablecer tu contraseña.

Restablecer contraseña

"Cuando todos piensan lo mismo, es que nadie piensa mucho"
Walter Lippmann

Saludos,
Equipo Spiky

spiky

```
this.instance.get
<ForgotPasswordResponse>
('auth/forgot-password?
correo=' + email);
```

endpoint

gmail, outlook, hotmail deletes
any link that does not start with
'http' or 'https' in <a/>

option 1

<a/>

```
{http://192.168.1.70:4000}
/api/auth/redirect
?token={token}
&correoValid={0197495@up.edu.mx}
```

option 2

WEB_LINK

jwt.sign(correo, 'FG', '1h')

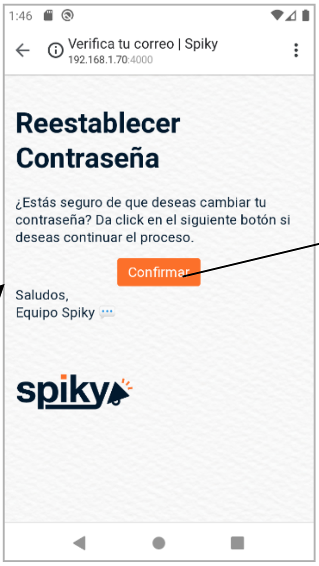
req.query.correo

any secret string

```
http://192.168.1.70:4000/api/auth/redirect?
token={token}&correoValid={correo}
```

endpoint

```
route.get('/redirect',
handleRedirect);
```



Reestablecer Contraseña

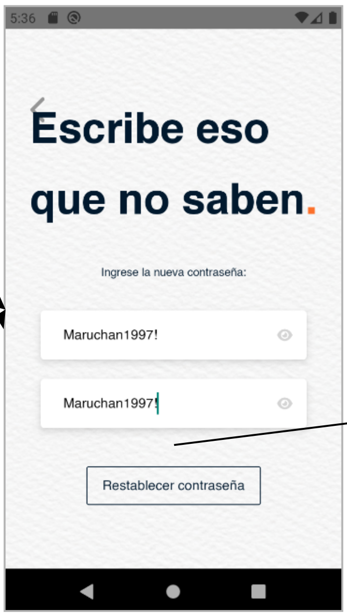
¿Estás seguro de que deseas cambiar tu contraseña? Da click en el siguiente botón si deseas continuar el proceso.

Saludos,
Equipo Spiky

```
{spikyapp://changeforgotpassword?
token={token}&correoValid=
{correo}}
```

this is an html file rewritten every time the server handles it, rewritten with <a/> now. file is in public/

ChangePasswordScreen



```
route.params.correoValid
```

```
return this.instance.put
<UpdatePasswordUri>
('auth/change-password-uri',
{
        validCorreo: correoValid,
        nuevaContrasena: newPassword,
        keyword:'FG'
}, config);
```

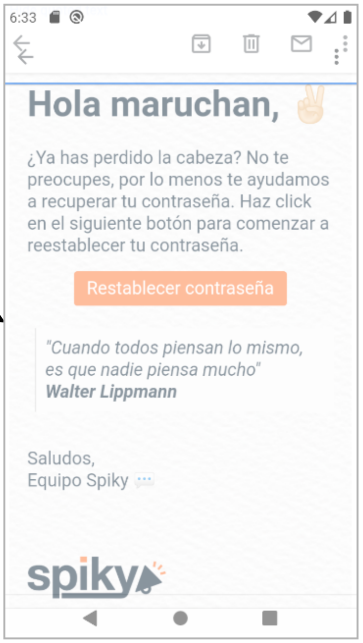
```
let config = {headers: {
              'x-token': tokenEmail,
              },
              };
```

endpoint

```
router.put('change-password-uri',
[...,validarVerifJWT], myFunc);
```

```
const token = req.header('x-token');
const {validCorreo} = req.body;
const {correo} = jwt.verify(token, seed);
if (validCorreo === correo) next();
```

continues with myFunc that updates password



Hola maruchan,

¿Ya has perdido la cabeza? No te preocupes, por lo menos te ayudamos a recuperar tu contraseña. Haz click en el siguiente botón para comenzar a reestablecer tu contraseña.

"Cuando todos piensan lo mismo, es que nadie piensa mucho"
Walter Lippmann

Saludos,
Equipo Spiky

this is an html file rewritten everytime the server handles it.

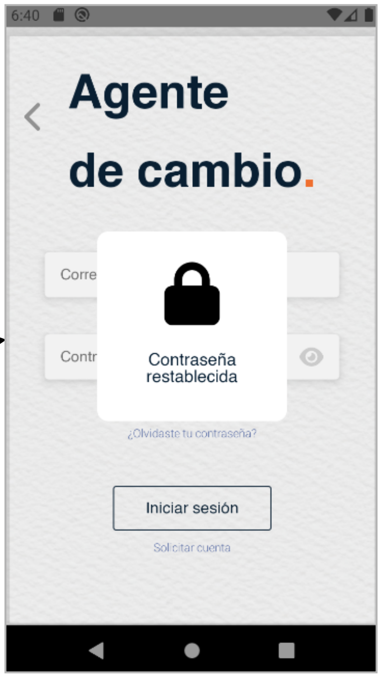
```
window.onload = function() {
window.location.href =
"spikyapp://changeforgotpassword?token=
{token}&correoValid={0197495@up.edu.mx}";
}
```

```
let url = process.env.DEEP_LINK;
let email = req.query.correoValid;
let token = req.query.token
let endpoint = url + '?token=' + token +
'&correoValid=' + email;
let html = templateRedirectHTML(endpoint);
fs.writeFileSync("./public/mail.html", html);
res.sendFile('mail.html', {...})
```

endpoint

```
route.get('/redirect',
handleRedirect);
```

LoginScreen



```
await updatePasswordUri(...);
navigation.navigate('LoginScreen');
```

must be in same stack

1. Create linking object for props in <NavigationContainer/>

```
    const linking = {
            prefixes: ['spikyapp://'],
            config: {
                initialRouteName: 'HomeScreen',
                screens: {
                    HomeScreen: {
                        path: 'homescreen',
                    },
                    ChangeForgotPasswordScreen: {
                        path: 'changeforgotpassword',
                    },
                },
            },
        };
```

2. Using URI-scheme package to configure URI schemes:

```
$ npx uri-scheme add spikyapp --ios
$ npx uri-scheme add spikyapp --android
```

3. Test the link from terminal:

```
$ npx uri-scheme open spikyapp://changeforgotpassword?token=asdf&correoValid=0197495@up.edu.mx --android
```

4. To receive the correoValid and token params in ChangeForgotPasswordScreen:

```
export const ChangeForgotPasswordScreen = ({ route }) => { route.params }
```

5. Configure all endpoints and screens following the previous diagram.