



Tecnológico Nacional de
México

Instituto Tecnológico de
Reynosa

Materia: Negocios electrónicos

Maestro: Miriam Puente Jiménez

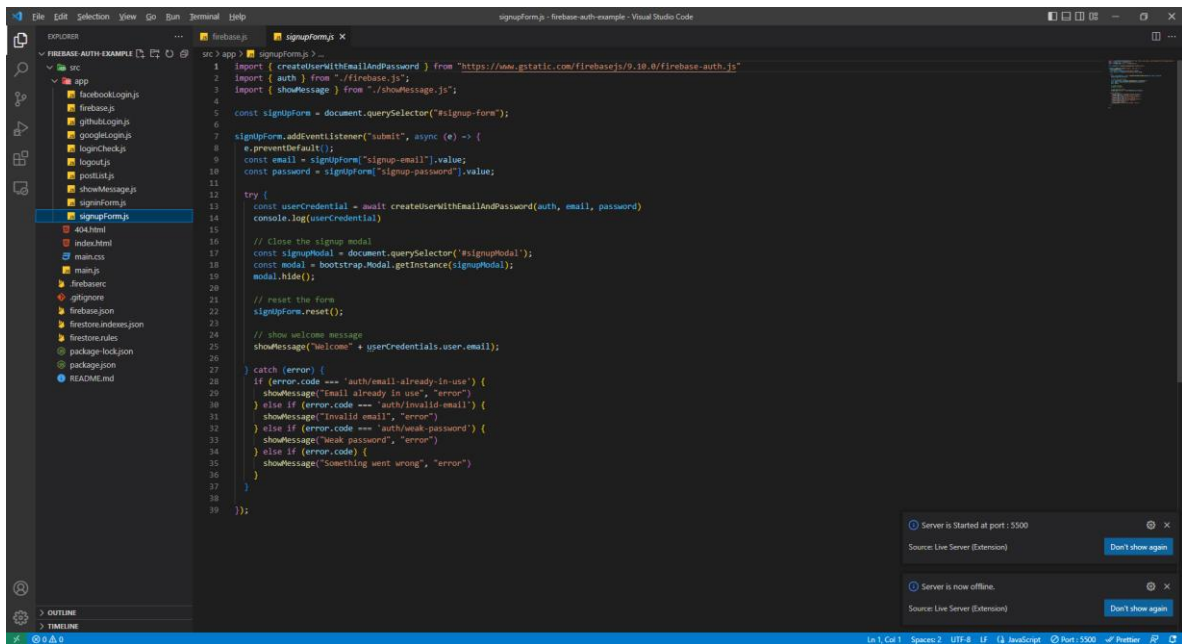
Título: Unidad 3 Tarea 2 Negocios
Electrónicos

Fecha de entrega: 24/10/22

Alumno: Castillo Jr. Gregorio

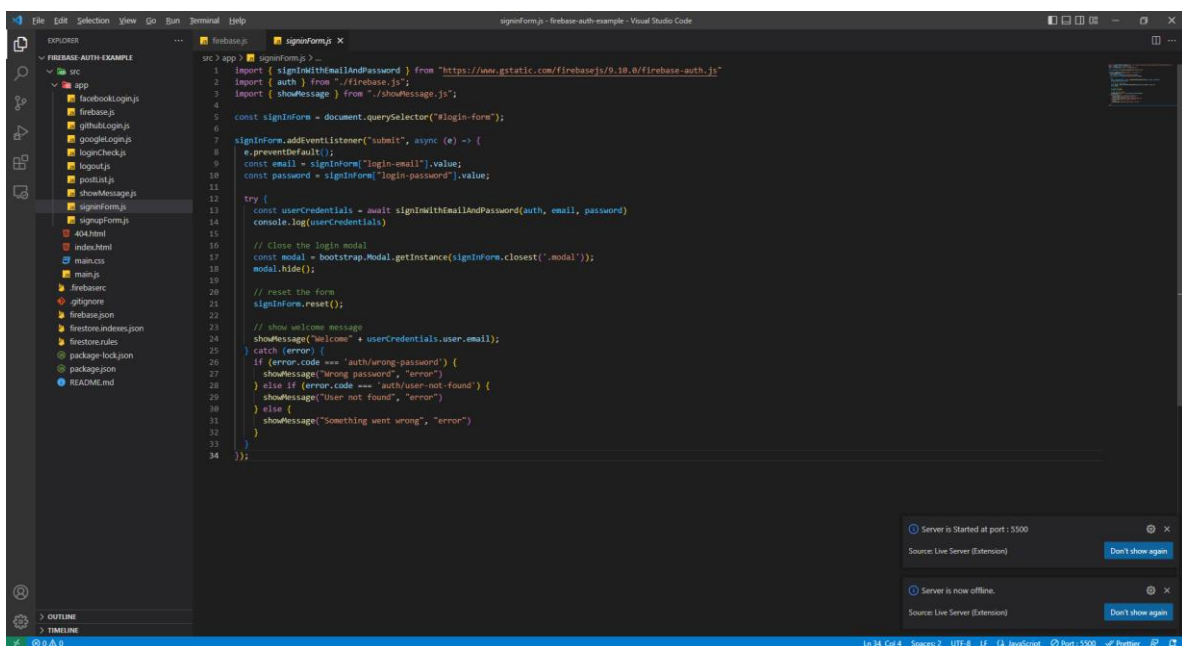
Numero de control: 19580589

Creación de sign in y sign up



The screenshot shows the Visual Studio Code editor with a file explorer on the left displaying a project structure for 'FIREBASE AUTH EXAMPLE'. The main editor window shows the 'signUpForm.js' file. The code imports Firebase functions and initializes a sign up form. It includes an event listener for the form's submit button, which calls 'createUserWithEmailAndPassword' and handles various error states like 'email-already-in-use', 'auth/invalid-email', 'auth/weak-password', and 'auth/user-not-found'. A Bootstrap modal is used to display messages and manage the form's visibility.

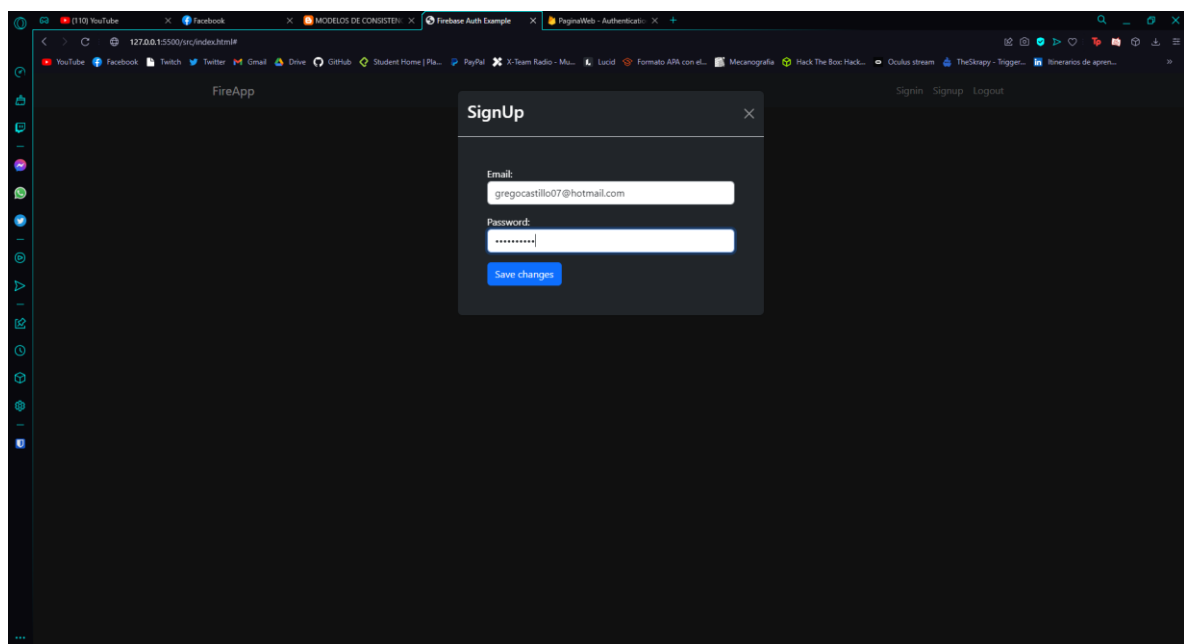
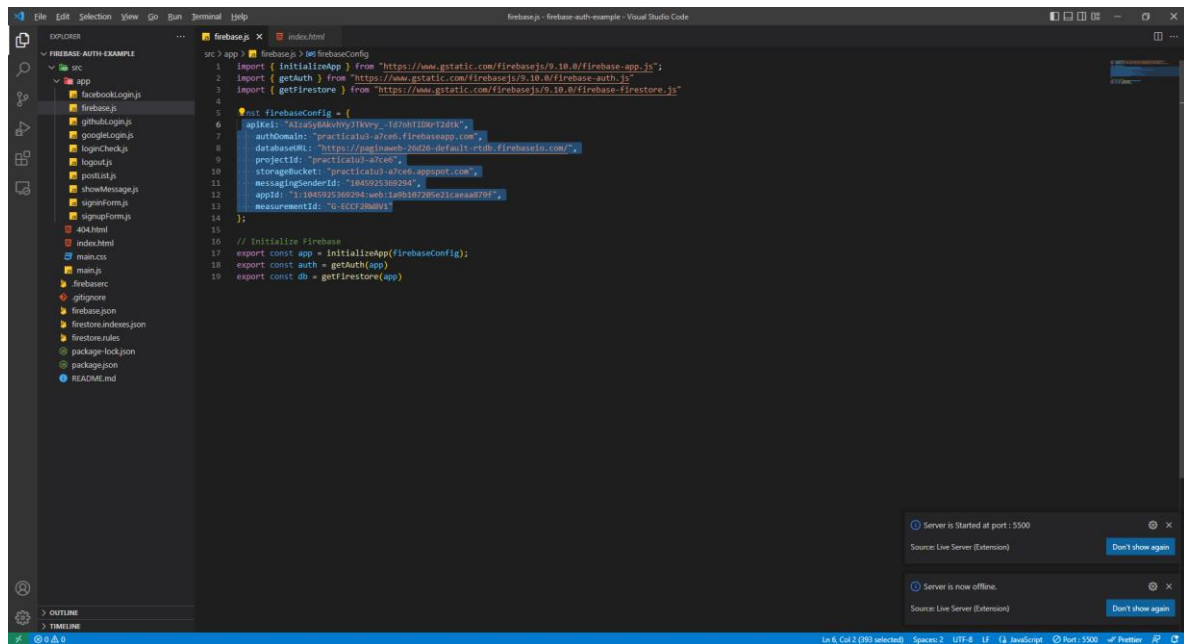
```
1 import { createUserWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/9.10.0/firebase-auth.js"
2 import { auth } from "../firebase.js";
3 import { showMessage } from "../showMessage.js";
4
5 const signUpForm = document.querySelector("#signup-form");
6
7 signUpForm.addEventListener("submit", async (e) => {
8   e.preventDefault();
9   const email = signUpForm["signup-email"].value;
10  const password = signUpForm["signup-password"].value;
11
12  try {
13    const userCredential = await createUserWithEmailAndPassword(auth, email, password)
14    console.log(userCredential)
15
16    // Close the signup modal
17    const signUpModal = document.querySelector("#signupModal");
18    const modal = bootstrap.Modal.getInstance(signUpModal);
19    modal.hide();
20
21    // reset the form
22    signUpForm.reset();
23
24    // show welcome message
25    showMessage("welcome" + userCredential.user.email);
26
27  } catch (error) {
28    if (error.code === 'auth/email-already-in-use') {
29      showMessage("Email already in use", "error");
30    } else if (error.code === 'auth/invalid-email') {
31      showMessage("Invalid email", "error");
32    } else if (error.code === 'auth/weak-password') {
33      showMessage("Weak password", "error");
34    } else if (error.code) {
35      showMessage("Something went wrong", "error");
36    }
37  }
38
39 });
```



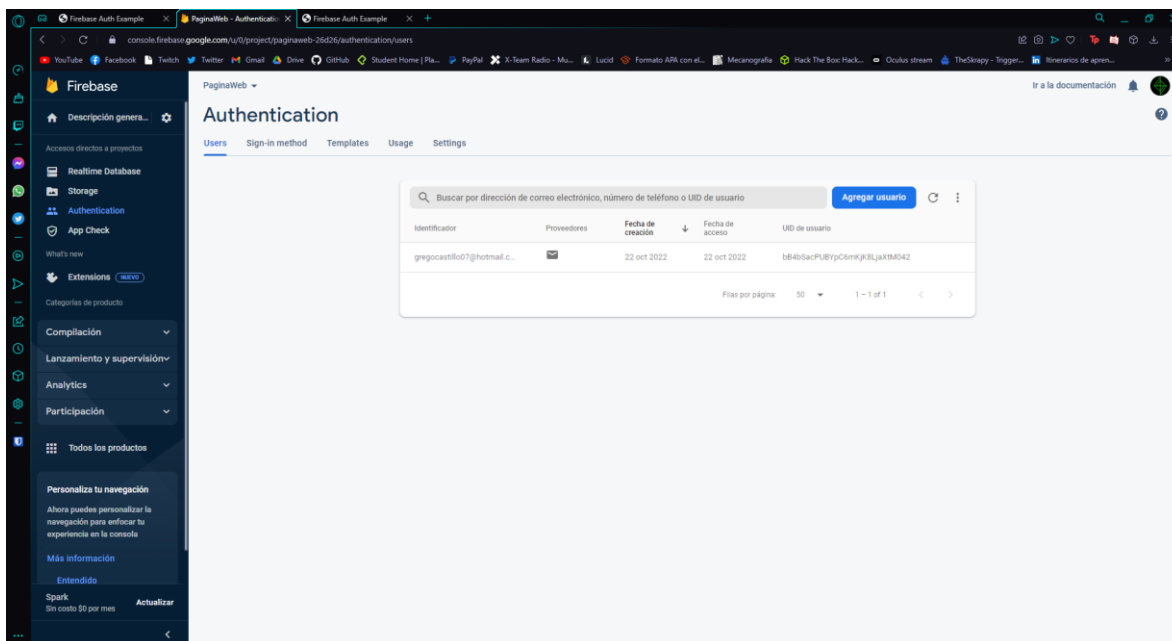
The screenshot shows the Visual Studio Code editor with the same project structure. The main editor window shows the 'signInForm.js' file. The code imports Firebase functions and initializes a sign in form. It includes an event listener for the form's submit button, which calls 'signInWithEmailAndPassword' and handles error states like 'auth/wrong-password', 'auth/user-not-found', and 'auth/invalid-email'. A Bootstrap modal is used to display messages and manage the form's visibility.

```
1 import { signInWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/9.10.0/firebase-auth.js"
2 import { auth } from "../firebase.js";
3 import { showMessage } from "../showMessage.js";
4
5 const signInForm = document.querySelector("#login-form");
6
7 signInForm.addEventListener("submit", async (e) => {
8   e.preventDefault();
9   const email = signInForm["login-email"].value;
10  const password = signInForm["login-password"].value;
11
12  try {
13    const userCredential = await signInWithEmailAndPassword(auth, email, password)
14    console.log(userCredential)
15
16    // Close the login modal
17    const modal = bootstrap.Modal.getInstance(signInForm.closest('.modal'));
18    modal.hide();
19
20    // reset the form
21    signInForm.reset();
22
23    // show welcome message
24    showMessage("welcome" + userCredential.user.email);
25
26  } catch (error) {
27    if (error.code === 'auth/wrong-password') {
28      showMessage("wrong password", "error");
29    } else if (error.code === 'auth/user-not-found') {
30      showMessage("User not found", "error");
31    } else {
32      showMessage("Something went wrong", "error");
33    }
34  }
35
36 });
```

Uso de firebase



Resultados de firebase



Conclusiones

Se trata de una actividad bastante didáctica y con la cual podemos generar modelos de autenticación con firebase, estos pueden ser de distintos tipos, como lo son vía email, Facebook, GitHub, Google, Apple, Microsoft, y muchos otros más, esto permitiendo una versatilidad casi infinita, ya que entendemos que no todos usan el mismo tipo de cuentas o redes sociales, de tal modo tener una amplia gamma implica en que podamos trabajar con múltiples recursos y personas, sin limitarnos a un solo nicho, atrayendo más clientes y trabajadores