

Universidad Don Bosco



Estudiantes:

Derek Marcelo Monge Aguilar
Kenneth Gabriel Monge Aguilar
Andres Eduardo Navidad Flores
Abner Ismael Rivera Leiva

Asignatura:

Desarrollo de Aplic. Web con Soft. Interpret. en el Cliente

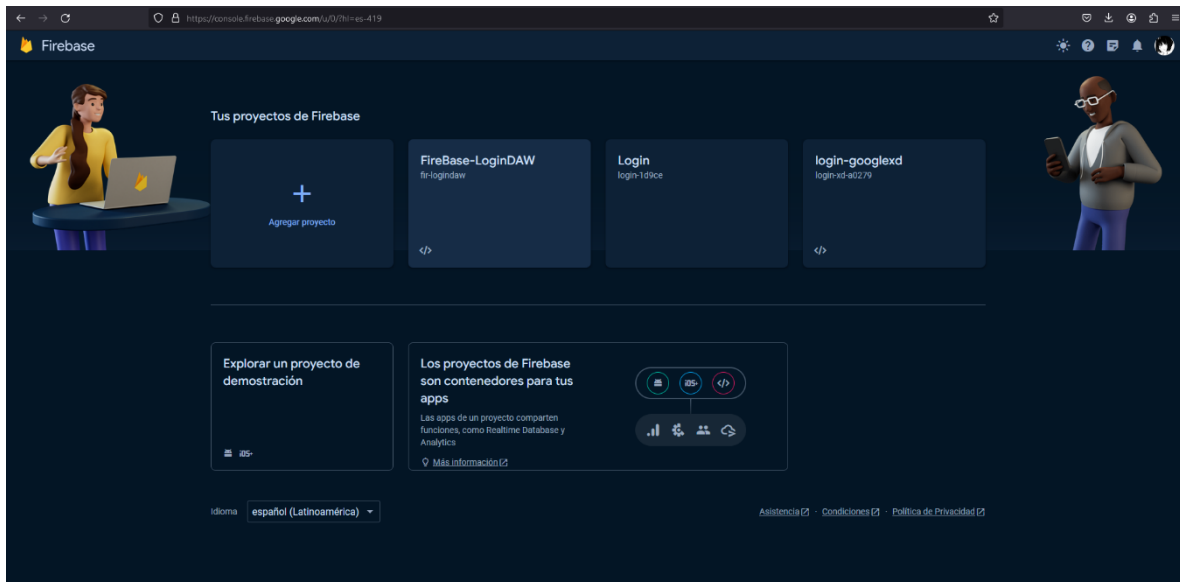
Para realizar nuestro proyecto sobre la Creación de nuestra Aplicación Web ocupamos diferentes tipos de tecnologías tanto para el **Front-end y Back-end**.

Como Back-end utilizamos **Firebase**:

Firebase es una plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google. Ofrece una variedad de servicios y herramientas que ayudan a los desarrolladores a crear, mejorar y hacer crecer sus aplicaciones. Algunas de las características principales de Firebase incluyen:

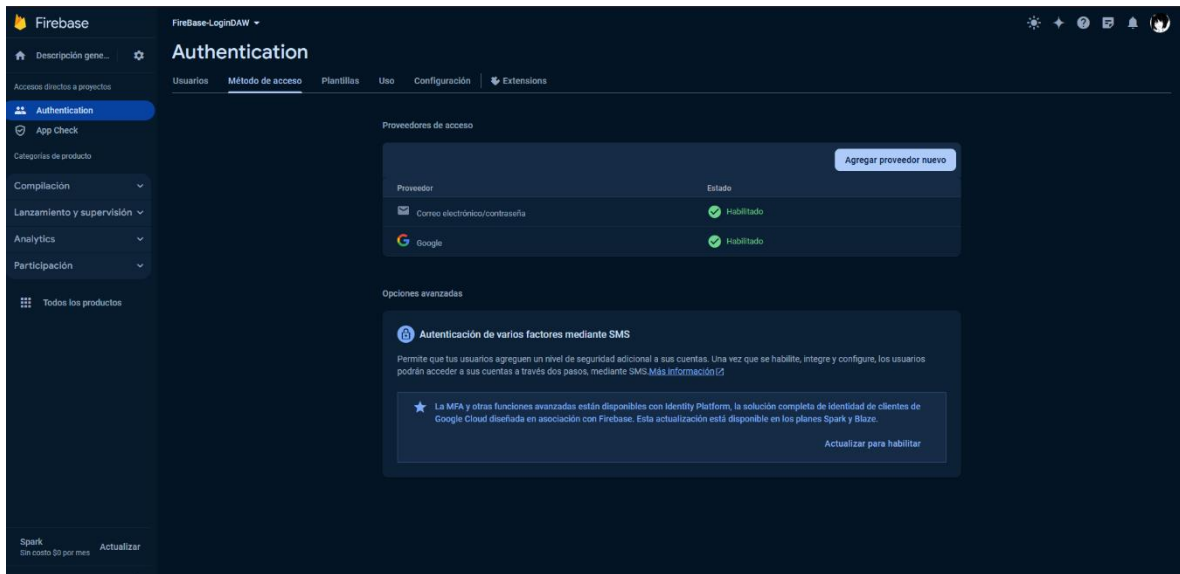
1. **Base de datos en tiempo real:** Firebase proporciona una base de datos en tiempo real que permite sincronizar datos entre usuarios en tiempo real.
2. **Autenticación de usuarios:** Permite a los desarrolladores autenticar usuarios fácilmente utilizando métodos como correo electrónico/password, autenticación con proveedores de identidad social (como Google, Facebook, Twitter) y autenticación anónima.

Firebase jugará un papel importante para la autenticación de usuarios.



“Ejemplo de su interfaz principal”

Agregamos un nuevo proyecto y seleccionamos los diferentes tipos de servicios de autenticación, ya sea por correo, Google, Facebook entre otros.



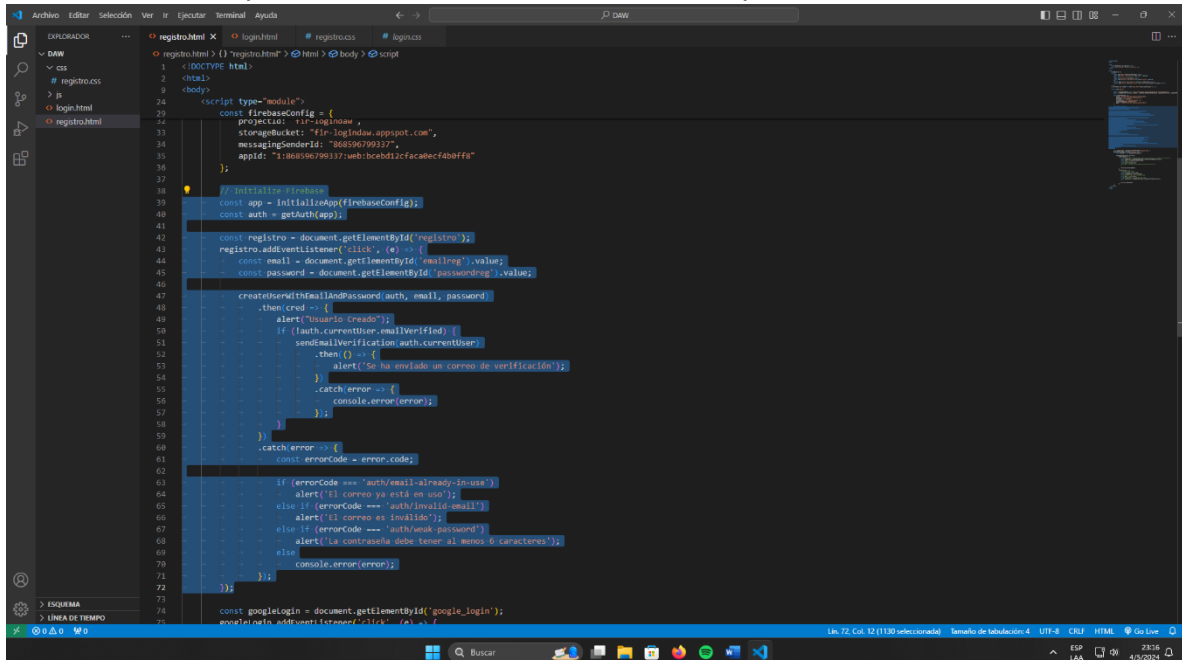
“Ejemplo de su interfaz de autenticación”

Para el funcionamiento de nuestro login y que las peticiones se vean reflejadas desde nuestro Firebase necesitamos importar ciertos módulos a nuestro código del proyecto:

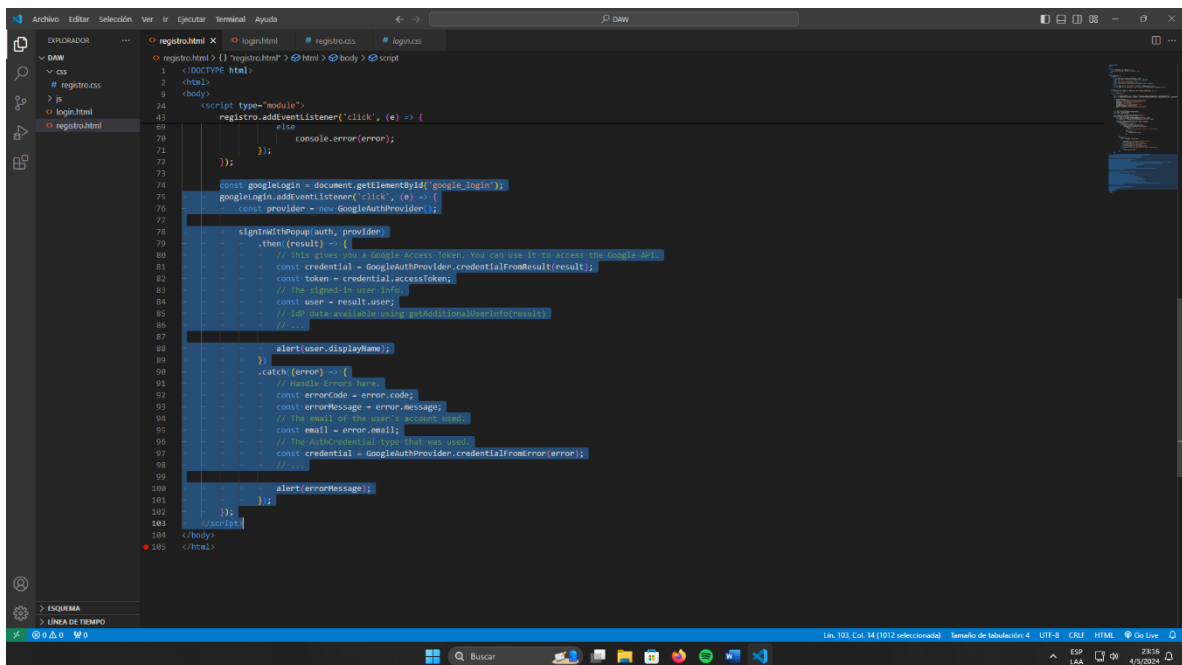
```
<script type="module">
  import { initializeApp } from 'https://www.gstatic.com/firebasejs/10.11.1/firebase-app.js';
  import { sendEmailVerification, getAuth, createUserWithEmailAndPassword, GoogleAuthProvider, signInWithPopup } from 'https://www.gstatic.com/firebasejs/10.11.1/firebase-auth.js';

  const firebaseConfig = {
    apiKey: "AIzaSyCRsLBYDCOE6zwoaEYRZL-MKVK7zmHMS8A",
    authDomain: "fir-loginaw.firebaseio.com",
    projectId: "fir-loginaw",
    storageBucket: "fir-loginaw.appspot.com",
    messagingSenderId: "868596799337",
    appId: "1:868596799337:web:bcebd12cfaca8ecf4b0ff8"
  };
</script>
```

Luego de importar los módulos, Inicializamos firebase y agregamos ciertos eventos para el funcionamiento de nuestra Aplicación Web:



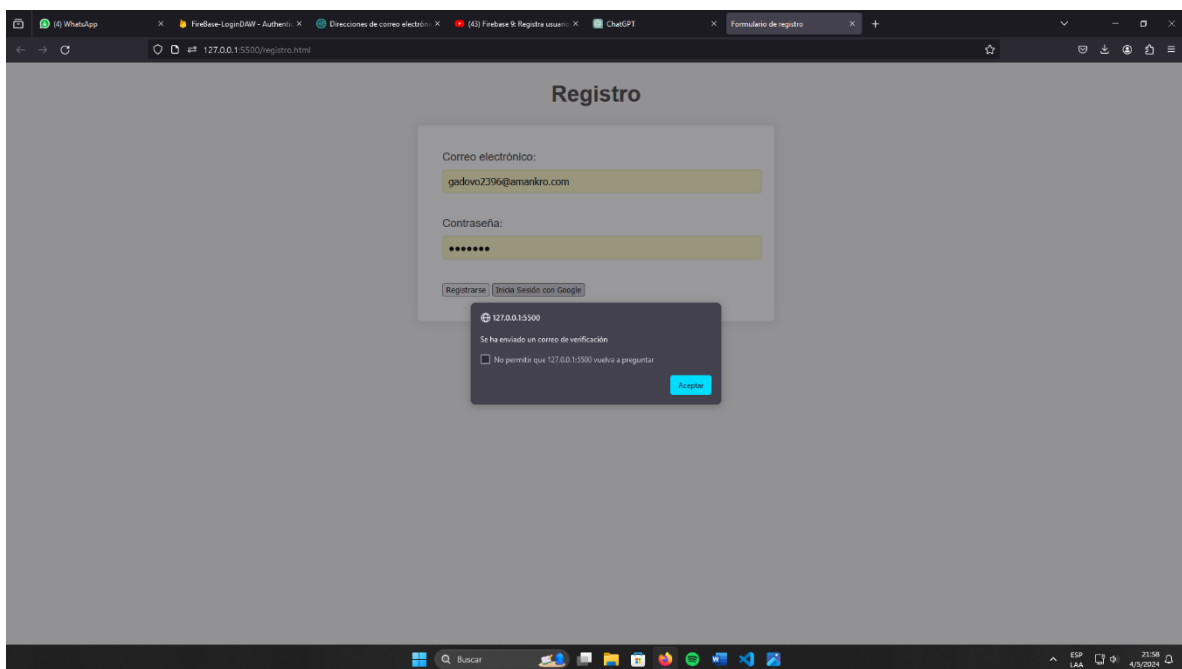
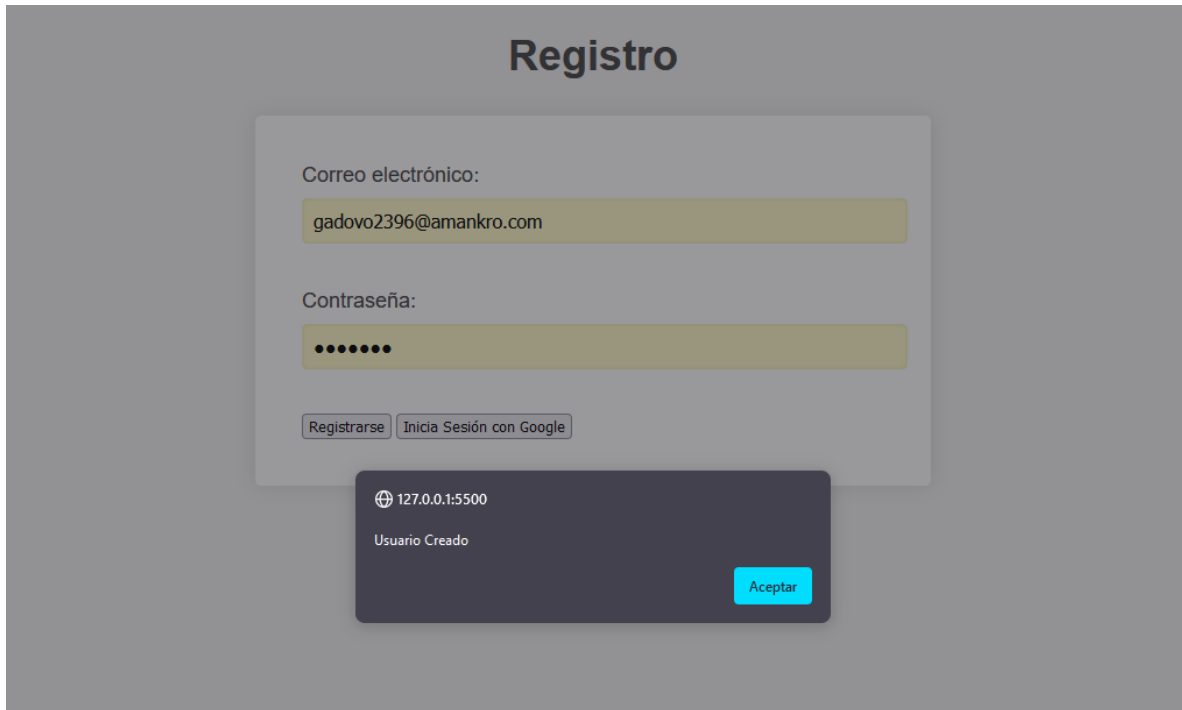
```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <script type="module">
6
7   const firebaseConfig = {
8     projectId: "fir-logindaw",
9     storageBucket: "fir-logindaw.appspot.com",
10    messagingSenderId: "868596799337",
11    appId: "1:868596799337:web:bcbad12cfacaeef4b0ffa"
12  };
13
14  // Initialize Firebase
15  const app = initializeApp(firebaseConfig);
16  const auth = getAuth(app);
17
18  const registro = document.getElementById('registro');
19  registro.addEventListener('click', (e) => {
20    const email = document.getElementById('email-reg').value;
21    const password = document.getElementById('password-reg').value;
22
23    createWithEmailAndPassword(auth, email, password)
24      .then((cred) => {
25        alert('Usuario Creado');
26        if (!auth.currentUser.emailVerified) {
27          sendEmailVerification(auth.currentUser);
28          .then(() => {
29            alert('Se ha enviado un correo de verificación');
30          })
31          .catch(error => {
32            console.error(error);
33          });
34        }
35      }, catch error => {
36        const errorCode = error.code;
37
38        if (errorCode === 'auth/email-already-in-use') {
39          alert('El correo ya está en uso');
40        } else if (errorCode === 'auth/invalid-email') {
41          alert('El correo es inválido');
42        } else if (errorCode === 'auth/weak-password') {
43          alert('La contraseña debe tener al menos 6 caracteres');
44        } else {
45          console.error(error);
46        }
47      });
48  });
49
50  const googlelogin = document.getElementById('google_login');
51  googlelogin.addEventListener('click', (e) => {
```



```
52    // This gives you a Google Access token. You can use it to access the Google API.
53    const credential = GoogleAuthProvider.credentialFromResult(result);
54    const token = credential.accessToken;
55    // the signed-in user info
56    const user = result.user;
57    // See how to retrieve additional user info
58
59    alert(user.displayName);
60  }, catch (error) => {
61    // Handle Errors here
62    const errorCode = error.code;
63    const errorMessage = error.message;
64    // the email of the user's account used
65    const email = error.email;
66    // The AuthCredential type that was used
67    const credential = GoogleAuthProvider.credentialFromError(error);
68
69    alert(errorMessage);
70  });
71  }
72  </script>
73 </body>
74 </html>
```

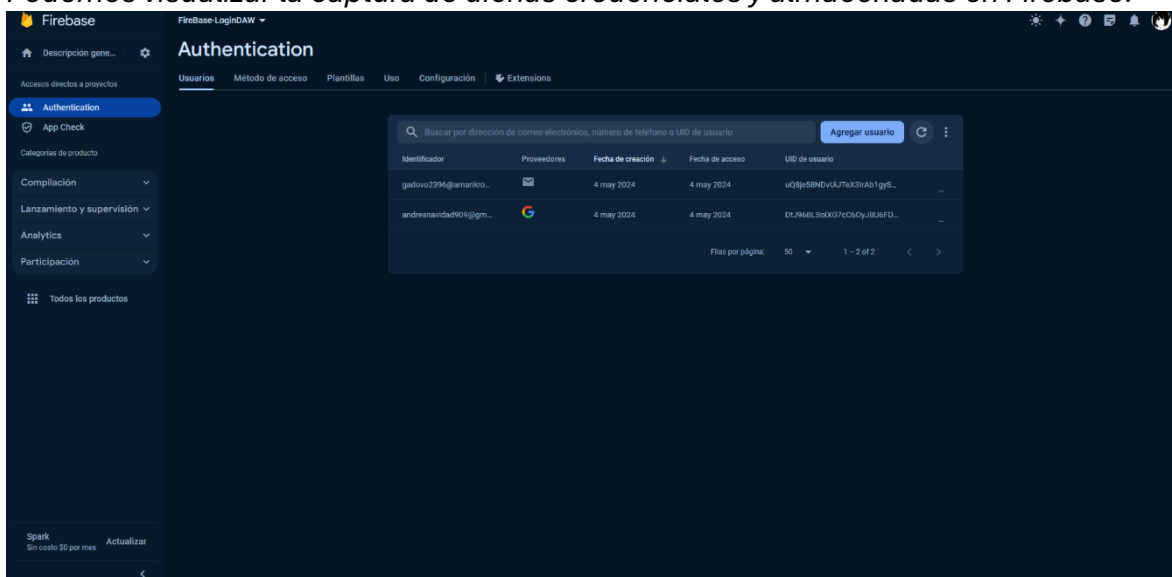
“Ejemplo de eventos creados”

Un ejemplo de su funcionamiento sería lo siguiente:
Al iniciar y guardar información “credenciales de inicio estas se deben de reflejar en nuestro firebase”.



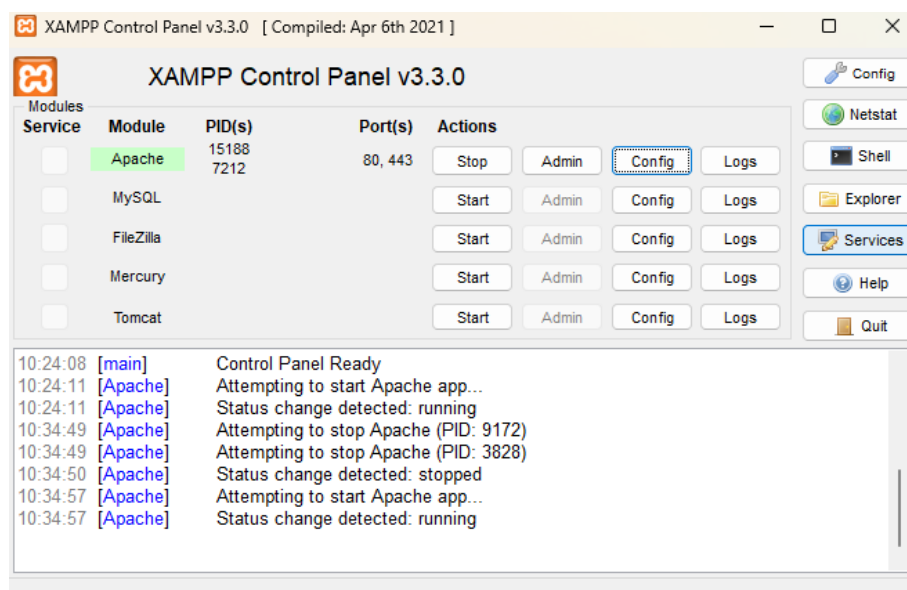
“Ejemplo de proceso de creación de usuario”

Podemos visualizar la captura de dichas credenciales y almacenadas en Firebase:

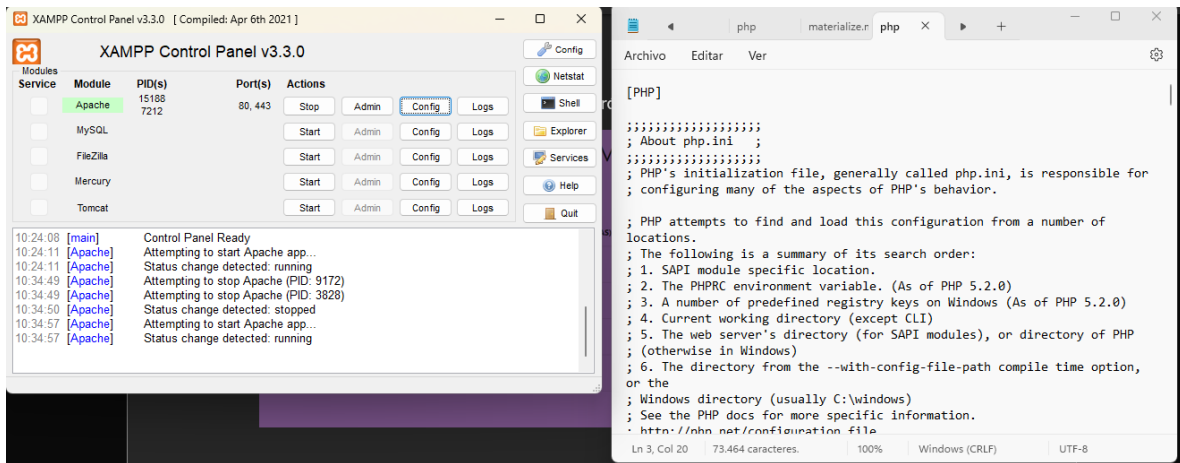


Esto funciona tanto para la validación de correos y contraseñas como con los proveedores Google entre otros.

Usaremos de servidor local **XAMPP** para probar nuestro sitio
Iniciaremos el servicio **Apache** para ver nuestro Localhost



Para que nuestra conexión del proyecto funcione tendremos
editar un ini que se encuentra en el botón config (php.ini)

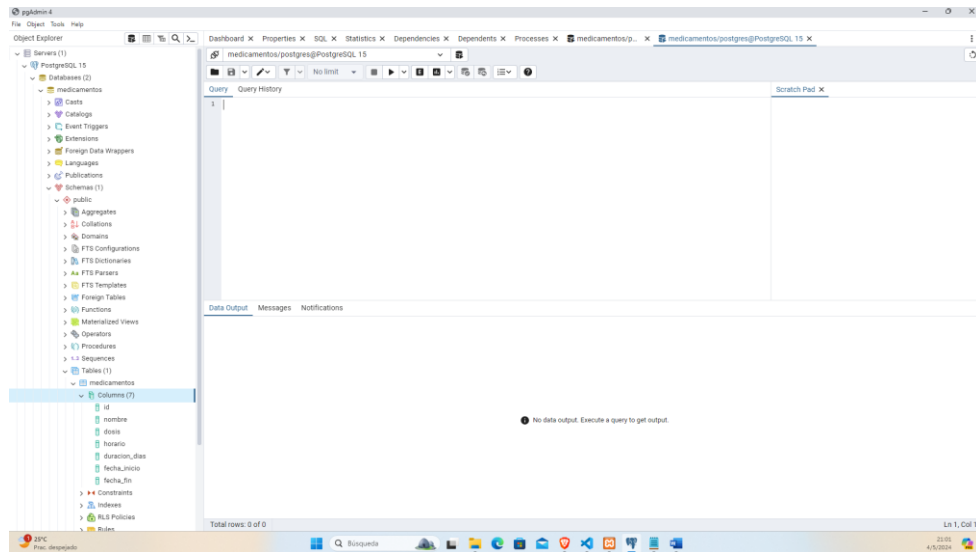


Una vez ahí, le damos a el botón del Notepad llamado **Editar**
seleccionamos la opción de buscar: extension=pgsql
 por defecto viene con un ; el cual tendremos que quitar y guardar el documento.







La base de datos que utilizaremos es **Postgres**
 Este es el script que utilizamos para el registro de datos

```
CREATE TABLE medicamentos (
  id SERIAL PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  dosis VARCHAR(50) NOT NULL,
  horario VARCHAR(50) NOT NULL,
  duracion_dias INTEGER NOT NULL
);
ALTER TABLE medicamentos
ADD fecha_inicio DATE,
ADD fecha_fin DATE;
```

Esta es la interfaz de Postgres (PGADMIN)
 aquí pegaremos la Query obtenida en este documento



Ya creada la base podemos empezar a trabajar en nuestro proyecto para empezar, crearemos la estructura del mismo.

Nombre	Fecha de modificación	Tipo	Tamaño
 api	4/5/2024 15:51	Carpeta de archivos	
 css	4/5/2024 10:49	Carpeta de archivos	
 img	4/5/2024 20:41	Carpeta de archivos	
 js	4/5/2024 15:38	Carpeta de archivos	
 historial	4/5/2024 15:46	Microsoft Edge H...	2 KB
 index	4/5/2024 15:25	Microsoft Edge H...	6 KB

En la carpeta **api** se encontrarán nuestros php que hacen conexión con la base de datos, en el **css** se encuentran nuestros estilos. en la carpeta **img** el logo que utilizamos y **js** todo el JavaScript de nuestro proyecto, afuera de todo van nuestros HTML

Para obtener todo el proyecto ingresar al Github

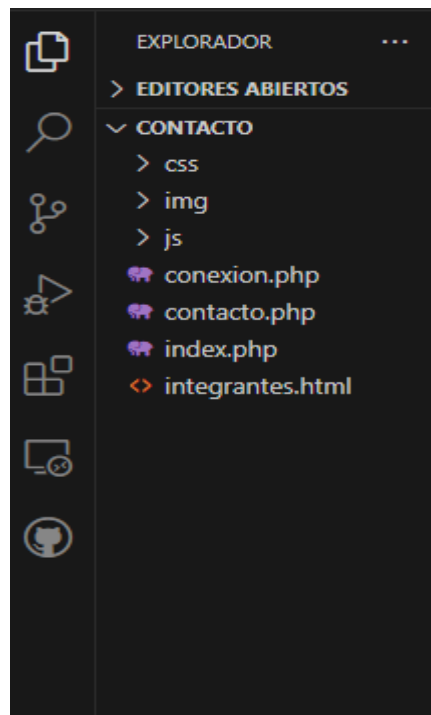
<https://github.com/KennethMonge808/P0royectoCatedraDaw-FaseFinal>

Se crea la tabla datos para la de pagina de contactos en pgadmin

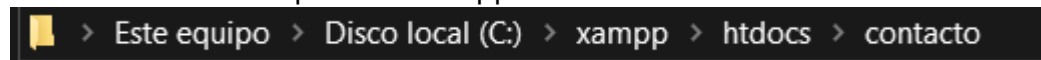
```
CREATE TABLE datos (  
  id SERIAL PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  email VARCHAR(50) NOT NULL,  
  direccion VARCHAR(50) NOT NULL,  
  telefono VARCHAR(20) NOT NULL,  
  mensaje VARCHAR(450) NOT NULL,  
  fecha TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
);
```

En nuestro Visual Studio creamos una Carpeta con nombre contacto y 3 archivos con los siguientes nombres:

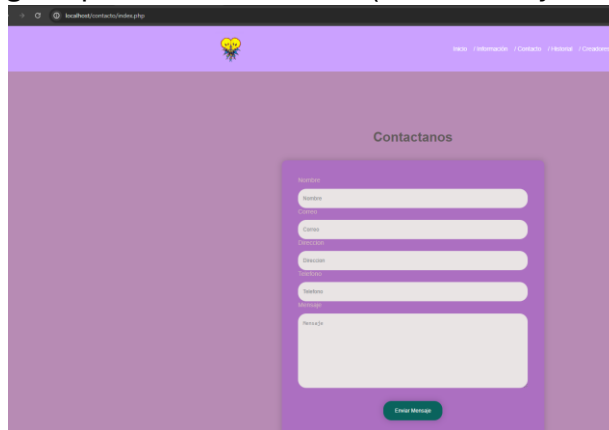
- Conexión.php
- Contacto.php
- Index.php



Para visualizar lo que estamos realizando es importante poder tener nuestro documento en la carpeta “C:\xampp\htdocs” dentro de esta ruta de esta forma:



Y se visualizará la pagina que estamos creando(se muestra ya creada).



En nuestro documento “index.php” llevara nuestro HTML, el cual después de diseñar nuestro modelo “formulario de contacto”, pondremos en la parte inferior el siguiente código de esta forma:

```
<?php
    include("contacto.php")
?>
```

Esto para poder utilizar el archivo de contacto.php que ya habías creado

```

index.php
29 <div class="contenedor1">
30 <div class="form-container">
31 <br> <h2>Contactanos</h2><br><br>
32 <form id="form" action="" method="post" autocomplete="off" >
33 <div class="form-content">
34 <input type="text" id="name" name="name" placeholder="Nombre" />
35 </div>
36 <div class="form-content">
37 <input type="text" id="email" name="email" placeholder="Correo" />
38 </div>
39 <div class="form-content">
40 <input type="text" id="direction" name="direction" placeholder="Direccion" />
41 </div>
42 <div class="form-content">
43 <input type="text" id="phone" name="phone" placeholder="Telefono" />
44 </div>
45 <div class="form-content">
46 <input type="text" id="message" name="message" placeholder="Mensaje" />
47 </div>
48 <div class="form-content">
49 <input type="submit" value="Enviar Mensaje" />
50 </div>
51 </form>
52 </div>
53 </div>
54 <script src="js/contacto.js" ></script>
55 <?php
56 include("conexion.php")
57 ?>
58 </body>
59 </html>

```

Posteriormente configuramos el Archivo conexión.php donde haremos que se conecten los datos de nuestra carpeta “contacto”, esto permitirá que pueda tener conexión la base datos con los datos que hallamos creado, y que los datos que el usuario digite se envíen sin problema.

Lo configuramos de la siguiente forma:

```

conexion.php
1 <?php
2
3 $conex = mysqli_connect("localhost", "root", "", "contacto");
4
5
6 ?>

```

Después configuramos nuestro archivo “contacto.php”, este documento es el que hará que funcione prácticamente la página de contacto, para que se envíe la información al servidor, lo configuramos de la siguiente forma para que el documento tome los datos que usuario escribe y envíe mensajes de “éxito” o “error”

```

contacto.php
1  <?php
2
3  include("conexion.php");
4
5  if (isset($_POST['contact'])) {
6      if (
7          strlen($_POST['name']) >= 1 &&
8          strlen($_POST['email']) >= 1 &&
9          strlen($_POST['direction']) >= 1 &&
10         strlen($_POST['phone']) >= 1 &&
11         strlen($_POST['message']) >= 1
12     ) {
13         $name = trim($_POST['name']);
14         $email = trim($_POST['email']);
15         $direction = trim($_POST['direction']);
16         $phone = trim($_POST['phone']);
17         $message = trim($_POST['message']);
18         $fecha = date("d/m/y");
19
20         $consulta = "INSERT INTO datos(nombre , email, direccion, telefono, mensaje, fecha)
21         VALUES ('$name', '$email', '$direction', '$phone', '$message', '$fecha')";
22         $resultado = mysqli_query($conex, $consulta);
23         if ($resultado) {
24             ?>
25             <h3 class="success">Tu Mensaje se ha enviado correctamente <br> Te contactaremos</h3>
26             <?php
27         } else {
28             ?>
29             <h3 class="error">Ocurrio un error</h3>
30             <?php
31         }
32     } else { ?> <h3 class="error">llena todos los campos</h3> <?php }
33 }
34
35
36
37
38  ?>

```

Luego validamos que al llenar el formulario se envíen de manera correcta la información del usuario.

Llenamos el formulario de la siguiente forma:

Contactanos

Nombre

Juan Carlos Menjivar Ramirez

Correo

juan.ramirez@docente.udb.edu.sv

Direccion

san salvador

Telefono

76894891

Mensaje

Por este medio podemos enviar, consultas, dudas o sugerencias al servidor

Enviar Mensaje