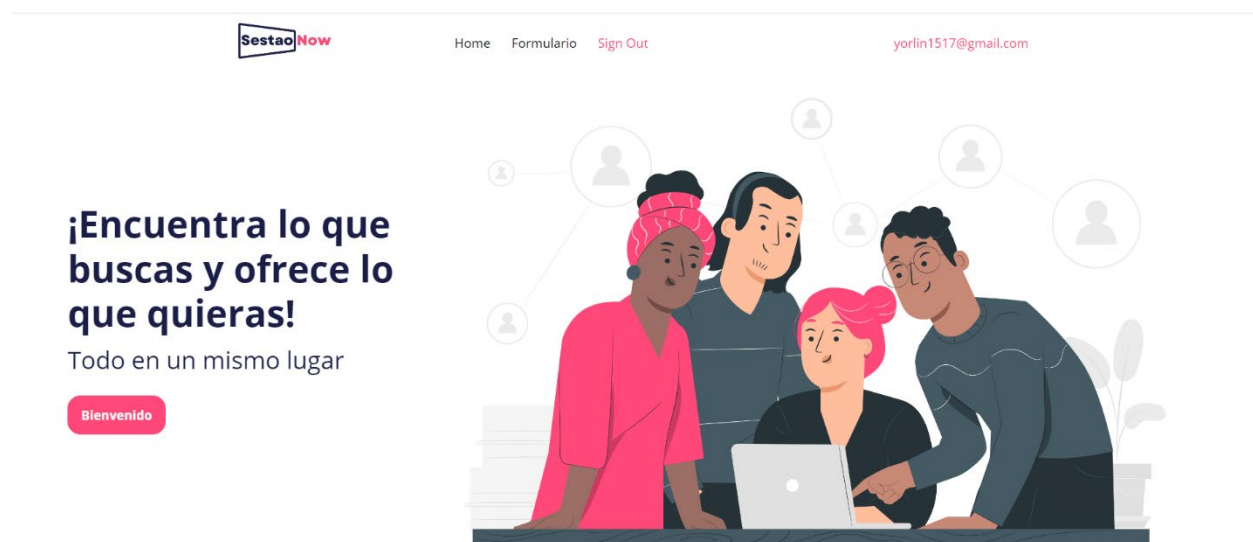


Proyecto Personal Sestao Now

Descubre Sestao Now: Tu portal de anuncios local. Nuestra plataforma minimalista nació con la visión de conectar a la comunidad de **Sestao**, permitiendo la publicación de diversos anuncios para satisfacer las necesidades locales. En este documento, encontrarás una detallada guía paso a paso para aprovechar al máximo nuestra web. Además, te invitamos a ver nuestro video demostrativo en **YouTube**, donde podrás experimentar el funcionamiento de **Sestao Now** en diferentes dispositivos. Únete a nosotros y explora la conveniencia y alcance que ofrecemos para anunciar y encontrar ofertas en Sestao. ¡Bienvenido a **Sestao Now**!

A medida que te voy mostrando la web con algunas capturas, te iré contando como se realizo y todas sus características.



Este proyecto lo podemos dividir en 2, la parte del **Front** y la del **Back**.

Front: En esta parte utilizamos diferentes tecnologías, tales como: HTML, CSS3, JAVASCRIPT, BOOTSTRAP, VUE3 y una librería llamada SWEETALERT. Y podemos añadir PHOTOSHOP y AFTEREFFECTS para su edición de la presentación.

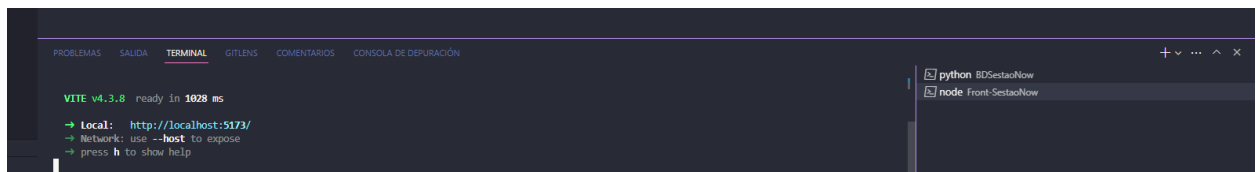
Back: En esta sección se implementó: PYTHON, SQL Y SQLITE.

¿COMO HACEMOS PARA PODERLO EN MARCHA?

- 1) Abrimos el proyecto SestaoNow en Visual Studio Code donde aparecerán 2 carpetas. **BDSestaoNow** (Carpeta del Back) y **Front-SestaoNow** (Carpeta del Front).

- 2) Aquí mismo podemos abrir 2 terminales, la primera para el back (asegúrenos que estemos en **BDSestaoNow**) donde primero haremos este comando: **npm install** , donde instalaremos los paquetes necesarios. Luego de esto lo ponemos en marcha con: **python3 run.py** o **python run.py**. Al hacer esto, como resultado tendremos el servidor funcionando con esta ruta: * **Running on http://127.0.0.1:5000**
- 3) El siguiente paso es abrir otra terminal, **PERO SIN CERRAR LA DE ARRIBA**, asegurémonos que estemos en (**Front-SestaoNow**) y ejecutamos el mismo comando: **npm install** y luego de instalar paquetes continuamos con el otro comando: **npm run dev**. Aquí abrimos esta dirección con nuestro navegador y si esta todo bien, se vera la web sin errores.

Aquí puedes ver el resultado para abrirlo en el navegador y los 2 terminales abiertos en un solo visual code.



Empecemos, En su **Home** podrás ver la barra de menú donde podrás ver **el email del usuario** que se ha registrado y la opción de cerrar sesión.

Muy importante, cualquier usuario podrá ser registrado y podrá ver los detalles de los anuncios, pero **NO** podrá entrar a la pagina de **FORMULARIO**, esta parte solo el **ADMINISTRADOR** tendrá acceso, ya que nuestra página tiene **protección de rutas** y aquí es donde se crearán los anuncios.

[Home](#) [Formulario](#) [Sign Out](#)

yorlin1517@gmail.com

Mas abajo podrás ver la sección de categorías, donde aquí el usuario podrá interactuar con la web y seleccionar que categoría quiere ver, hay 3 opciones.



Continuando, llegamos a la **parte principal** donde se **mostrarán todos los anuncios disponibles**, pero **NO** se podrá ver el contacto y el nombre de la persona hasta que este usuario este registrado.



Comida Asiatica
comida - Sestao
Deliciosa comida ...
Yorlin
2023-06-14

[Detalles](#)



camarer@
empleo - Barakaldo
Busco camarer@ ...
Fabian
2023-06-14

[Detalles](#)



Comida latina
comida - sestao
Deliciosa comida ...
Carlos
2023-06-14

[Detalles](#)



Paseador canino
empleo - Barakaldo
Paseador amable...
Juan
2023-06-14

[Detalles](#)

Y finalizando este home, tendremos un footer.

Ubicacion

Sestao - Vizcaya




Contactanos

yorlin1517@gmail.com

Supongamos que nuestro usuario quiera ver los detalles del anuncio **sin haberse registrado** primero, esto sería lo que vería:

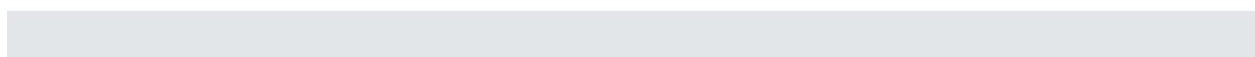


Upss!!



"Para ver más detalles del anuncio, por favor inicia sesión. Es rápido y sencillo.
¡Descubre lo que te estamos preparando!"

Sign In



Ahora este mismo usuario ha entendido y se ha registrado, supongamos que ha seleccionado el anuncio de Comida Asiática, esto sería lo que vería:

Comida Asiatica

comida

Yorlin



Deliciosa comida ...

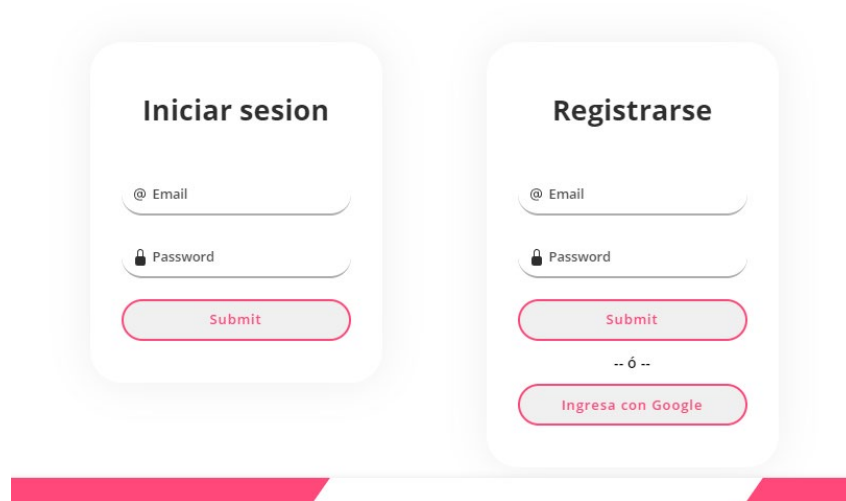
Contacto!!	Sestao
6873452	2023-06-14

Ahora podrá ver el dato del contacto para poder comunicarse con el dueño del anuncio.

¿Como seria la sección de registro?

Al ingresar, notarás que ofrecemos un práctico sistema de inicio de sesión. Si ya eres miembro, simplemente inicia sesión con tus credenciales. Para nuevos usuarios, te brindamos la opción de crear una nueva cuenta. Además, si prefieres utilizar tus credenciales de **Google**, también te ofrecemos la posibilidad de iniciar sesión de manera rápida y segura con esta opción. En Sestao

Now, facilitamos el acceso a nuestra comunidad para que puedas comenzar a disfrutar de nuestros servicios sin complicaciones.



The image shows two side-by-side white rounded rectangular forms on a light gray background. The left form is titled 'Iniciar sesion' and contains two input fields: '@ Email' and 'Password', each with a corresponding icon (an '@' symbol and a lock icon). Below these fields is a pink 'Submit' button. The right form is titled 'Registrarse' and contains the same '@ Email' and 'Password' input fields. Below these is a pink 'Submit' button, followed by a link that says '-- ó --', and then a pink button that says 'Ingresa con Google'.

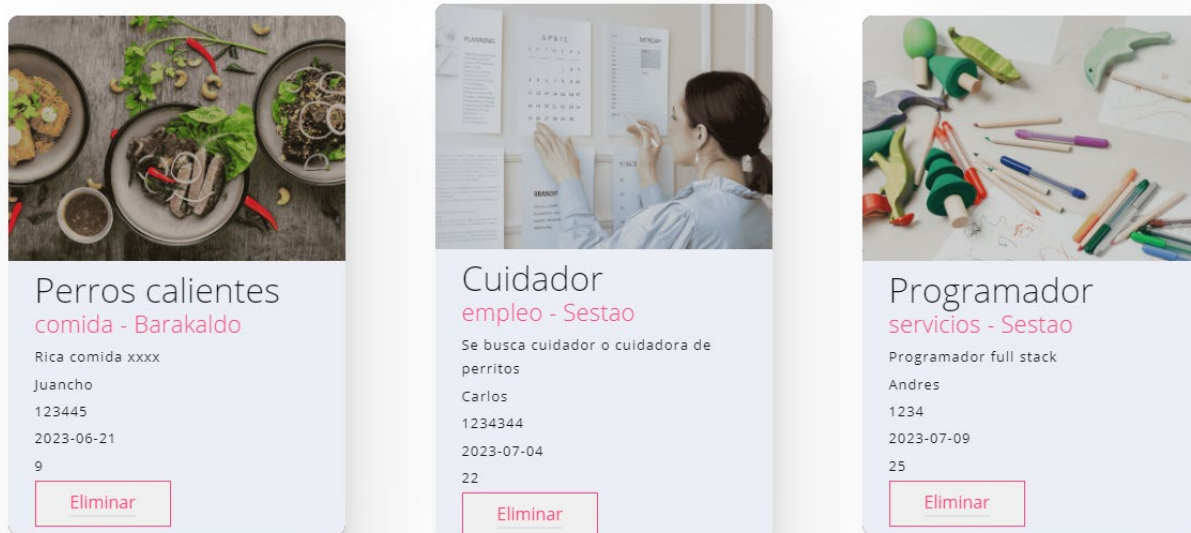
Si ingresamos con la credencial del **ADMINISTRADOR**, en la pagina del **FORMULARIO** se podrá observar lo siguiente:



The image shows a registration form for an advertisement. At the top, there is a pink circular icon followed by the text 'Registra tu anuncio'. Below this, a message says 'Muchas personas lo estaran esperando!!'. The form consists of several input fields: 'Titulo' and 'Nombre' (two separate boxes), 'Categoria:' followed by a dropdown menu with 'Elige...' and a downward arrow, 'Ubicacion:' followed by a dropdown menu with 'Elige...' and a downward arrow, 'Descripcion' (a larger text area), and 'Telefono'. At the bottom, there are three pink buttons: 'Postear anuncio', 'Eliminar anuncio', and 'Actualizar anuncio'.

Aquí el administrador podrá hacer el **CRUD**, Crear, Modificar y Eliminar los anuncios. Se tiene varios campos donde se ingresarán los datos y otros con opciones ya establecidas.

En esta misma página se podrán ver los anuncios, pero con una diferencia, **se podrá ver toda la información** y un botón de **eliminar el post**.



Hasta ahora, te he presentado una vista previa del diseño frontal de nuestra plataforma web. **Pronto**, proporcionaremos un **enlace** para que puedas experimentarla en acción y explorar todos sus aspectos.

¡¡AHORA EL BACK!!

Es el momento de explorar el "back-end", la parte central que gestiona toda la comunicación entre el "front-end" y nuestra base de datos. Aquí reside el núcleo de nuestra plataforma, asegurando una interacción fluida y eficiente para brindarte la mejor experiencia posible. Descubre cómo esta parte esencial de Sestao Now garantiza un funcionamiento cohesivo y confiable en cada aspecto de nuestro portal web.

En el archivo **app.py**, creamos automáticamente la tabla si aún no existe, y encontrarás diversas funciones que según su ruta realizarán diferentes acciones que forman el **CRUD (Crear, Leer, Actualizar, Eliminar)**. Estas funciones, a su vez, llaman a otras responsables de gestionar la comunicación con la base de datos. Es en este componente esencial donde se encuentra la lógica que garantiza la operatividad completa de nuestro portal web.

```
1 from flask import Flask, jsonify, request
2 import database
3 # importante agregar las 2 lineas de codigo para las CORS
4 from flask_cors import CORS
5
6 app = Flask(__name__)
7 CORS(app)
8
9 # Crear la tabla si no existe
10 database.create_table()
11
12 # POST nuevo articulo
13 @app.route('/add', methods=['POST'])
14 def add_post():
15     data = request.get_json()
16     categoria = data['categoria']
17     titulo = data['titulo']
18     ubicacion = data['ubicacion']
19     descripcion = data['descripcion']
20     nombre = data['nombre']
21     telefono = data['telefono']
22     database.add_anuncio(categoria,titulo,ubicacion,descripcion,nombre,telefono)
23     return "Anuncio agregada exitosamente"
24
25 # GET ALL
26 @app.route('/categoria', methods=['GET'])
27 def list_anuncios():
28     anuncios = database.get_anuncios()
29     return jsonify(anuncios)
30
31 # Get one by ID
32 @app.route('/anuncio/<id>', methods=['GET'])
33 def get_one_by_id(id):
34     anuncio = database.get_anuncio_by_id(id)
35     return jsonify(anuncio)
36
37 # GET ONE CATEGORY COMIDA / SERVICIOS / EMPLEO
38 @app.route('/categoria/<categoria>', methods=['GET'])
39 def get_anuncio_comida(categoria):
40     anuncio = database.get_anuncio_by_category(categoria)
41     return jsonify(anuncio)
42
43
44 # DELETE
45 @app.route('/delete/<id>', methods=['DELETE'])
46 def delete_pelicula(id):
47     database.delete_anuncio(id)
48     return "Anuncio eliminado exitosamente"
49
50 # UPDATE
51 @app.route('/anuncio/<id>', methods=['PUT'])
52 def update_pelicula(id):
53     data = request.get_json()
54     new_categoria = data['categoria']
55     new_titulo = data['titulo']
56     new_ubicacion = data['ubicacion']
57     new_descripcion = data['descripcion']
58     new_nombre = data['nombre']
59     new_telefono = data['telefono']
60
61     database.update_anuncio(id, new_categoria, new_titulo, new_ubicacion, new_descripcion, new_nombre, new_telefono)
62     return "Anuncio actualizado exitosamente"
63
64
```


En el archivo database.py, nuestra plataforma establece una comunicación directa y eficiente con la base de datos. Aquí se llevan a cabo diversas consultas de SQL, permitiendo un acceso fluido y preciso a los datos almacenados. Este componente es clave para asegurar la integridad y confiabilidad de la información en Sestao Now.

```
1 import sqlite3
2
3 def create_table():
4     conn = sqlite3.connect('anunciosSestaoNow.db')
5     cur = conn.cursor()
6     cur.execute("""CREATE TABLE IF NOT EXISTS anuncios (id INTEGER PRIMARY KEY AUTOINCREMENT,
7     categoria TEXT,
8     titulo TEXT,
9     ubicacion TEXT,
10    descripcion TEXT,
11    nombre TEXT,
12    telefono Text,
13    fecha date)""")
14    conn.commit()
15    conn.close()
16
17 def add_anuncio(categoria, titulo,ubicacion,descripcion,nombre,telefono):
18     conn = sqlite3.connect('anunciosSestaoNow.db')
19     cur = conn.cursor()
20     cur.execute("INSERT INTO anuncios (categoria, titulo, ubicacion, descripcion, nombre,telefono, fecha) VALUES (?, ?, ?, ?, ?, ?, date('now'))",
21     (categoria, titulo, ubicacion, descripcion, nombre, telefono))
22     conn.commit()
23     conn.close()
24
25 def get_anuncios():
26     conn = sqlite3.connect('anunciosSestaoNow.db')
27     cur = conn.cursor()
28     cur.execute("SELECT * FROM anuncios")
29     anuncios = [{'id': row[0],
30     'categoria': row[1],
31     'titulo': row[2],
32     'ubicacion': row[3],
33     'descripcion': row[4],
34     'nombre': row[5],
35     'telefono': row[6],
36     'fecha': row[7]} for row in cur.fetchall()]
37     conn.close()
38     return anuncios
39
40 def get_anuncio_by_category(categoria):
41     conn = sqlite3.connect('anunciosSestaoNow.db')
42     cur = conn.cursor()
43     #To create a tuple with only one item, you have to add a comma after the item, otherwise Python will not recognize it as a tuple.
44     cur.execute("SELECT * FROM anuncios WHERE categoria = ?", (categoria,))
45     # anuncio = cur.fetchall()
46     anuncio = [{'id': row[0],
47     'categoria': row[1],
48     'titulo': row[2],
49     'ubicacion': row[3],
50     'descripcion': row[4],
51     'nombre': row[5],
52     'telefono': row[6],
53     'fecha': row[7]} for row in cur.fetchall()]
54     conn.close()
55     return anuncio
```

```
1 def get_anuncio_by_id(id):
2     conn = sqlite3.connect('anunciosSestaoNow.db')
3     cur = conn.cursor()
4     #To create a tuple with only one item, you have to add a comma after the item, otherwise Python will not recognize it as a tuple.
5     cur.execute("SELECT * FROM anuncios WHERE id = ?", (id,))
6     anuncio = cur.fetchone()
7     if anuncio:
8         anuncioDatos = {'id': anuncio[0],
9                          'categoria': anuncio[1],
10                         'titulo': anuncio[2],
11                         'ubicacion': anuncio[3],
12                         'descripcion': anuncio[4],
13                         'nombre': anuncio[5],
14                         'telefono': anuncio[6],
15                         'fecha': anuncio[7]}
16     conn.close()
17     return anuncioDatos
18 else:
19     return 'Post no encontrado'
20
21
22 def delete_anuncio(id):
23     conn = sqlite3.connect('anunciosSestaoNow.db')
24     cur = conn.cursor()
25     cur.execute("DELETE FROM anuncios WHERE id = ?", (id,))
26     conn.commit()
27     conn.close()
28
29 def update_anuncio(id, new_categoria, new_titulo, new_ubicacion, new_descripcion, new_nombre, new_telefono):
30     conn = sqlite3.connect('anunciosSestaoNow.db')
31     cur = conn.cursor()
32     cur.execute("""UPDATE anuncios SET
33                 categoria = ?,
34                 titulo = ?,
35                 ubicacion = ?,
36                 descripcion = ?,
37                 nombre = ?,
38                 telefono = ?
39                 WHERE id = ?""", (new_categoria,
40                                   new_titulo,
41                                   new_ubicacion,
42                                   new_descripcion,
43                                   new_nombre,
44                                   new_telefono,
45                                   id))
46     conn.commit()
47     conn.close()
48
```

Al llegar al final de este código, es el momento de presentarte la **representación de nuestra base de datos en forma de tabla**. Aquí se almacenan todos los datos relevantes que dan vida a Sestao Now. Esta estructura tabular garantiza un almacenamiento organizado y accesible, asegurando que la información esté siempre disponible para las operaciones que se realizan en la plataforma.

Gracias a esta representación en forma de tabla, podemos gestionar eficientemente los anuncios, usuarios y otros elementos clave en nuestra comunidad local.

SQL ▾ < 1 / 1 > 1 - 8 of 8							
id	categoria	titulo	ubicacion	descripcion	nombre	telefono	fecha
2	comida	Comida Asiatica	Sestao	Deliciosa comida ...	Vorlin	6873452	2023-06-14
3	empleo	camarer@	Barakaldo	Busco camarer@ ...	Fabian	6873888	2023-06-14
5	comida	Comida latina	sestao	Deliciosa comida ...	Carlos	6873855	2023-06-14
6	empleo	Paseador canino	Barakaldo	Paseador amable...	Juan	6873811	2023-06-14
7	servicios	Instructora	Sestao	Instructor de yoga...	Isabel	68738110	2023-06-14
9	comida	Perros calientes	Barakaldo	Rica comida xxxx	Juancho	123445	2023-06-21
22	empleo	Cuidador	Sestao	Se busca cuidador o cuidadora de perritos	Carlos	1234344	2023-07-04
25	servicios	Programador	Sestao	Programador full stack	Andres	1234	2023-07-09

Algunos comentarios finales:

Como mejoras futuras, me gustaria añadir mas interacción del usuario con la web, como por ejemplo:

- Que pueda dar me gusta a los diferentes anuncios y guardarlos en su cuenta.
- Que el usuario pueda agregar su propio anuncio y esperar el proceso de verificación para ser publicado en la web.

AHORA SI!!

¡La espera ha terminado!

Cumplimos nuestra promesa y ahora es el momento de ver en acción todo el esfuerzo y dedicación invertidos en Sestao Now. Te emocionará ver cómo funciona nuestro portal en esta demostración exclusiva que hemos preparado para ti.

¡Haz clic en el siguiente enlace y sumérgete en la experiencia completa de Sestao Now!

Mejor experiencia en HD 1080 Y Horizontal [SestaoNow](#)

¡Disfruta de la demostración y bienvenido a Sestao Now!