

Librevento **Armado**

Materiales

Palas diseño Savonius

Esta parte del proyecto fue hecha con una impresora 3D con PETG (material de impresión 3D), elegimos este diseño vertical ya que tiene una forma que consiguió resultados de eficiencia mejores a muchos otros diseños, como también porque su construcción es sencilla y de bajo costo



Motores de corriente continua 12v

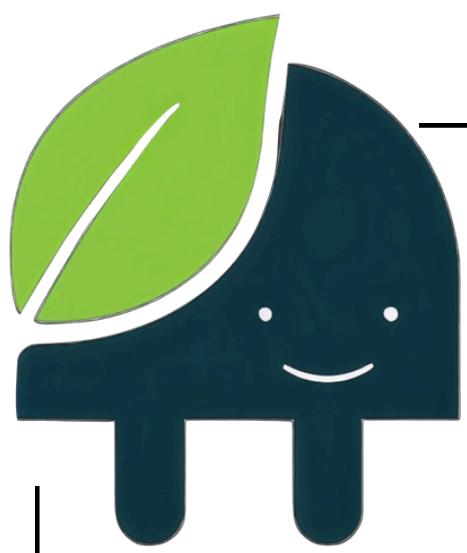
A estos componentes los reciclamos de una impresora vieja la cual no servía. Estos son los que generan la energía al momento de que las palas sean empujadas por el viento.



Engranajes

Estas piezas son clave ya que gracias a ellas el movimiento de las palas se transmite a los motores. Constan de 3 (tres) engranajes hermanados impresos en 3D con el mismo material que las palas, siendo uno más grande, el cual está unido a las palas y otros dos menores que van unidos a los ejes de los motores.





Librevento **Armado**

Materiales

Palas diseño Savonius

Esta parte del proyecto fue hecha con una impresora 3D con PETG (material de impresión 3D), elegimos este diseño vertical ya que tiene una forma que consiguió resultados de eficiencia mejores a muchos otros diseños, como también porque su construcción es sencilla y de bajo costo



Motores de corriente continua 12v

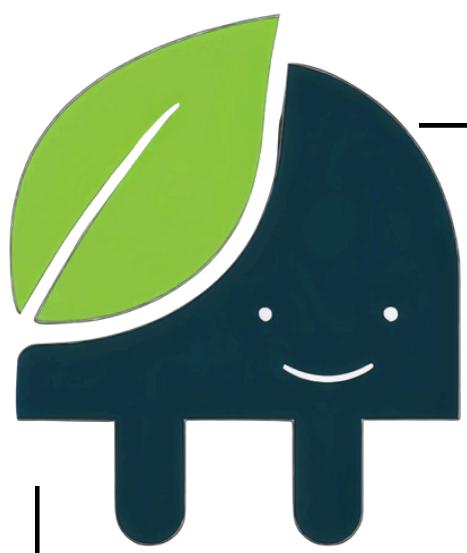
A estos componentes los reciclamos de una impresora vieja la cual no servía. Estos son los que generan la energía al momento de que las palas sean empujadas por el viento.



Engranajes

Estas piezas son clave ya que gracias a ellas el movimiento de las palas se transmite a los motores. Constan de 3 (tres) engranajes hermanados impresos en 3D con el mismo material que las palas, siendo uno más grande, el cual está unido a las palas y otros dos menores que van unidos a los ejes de los motores.

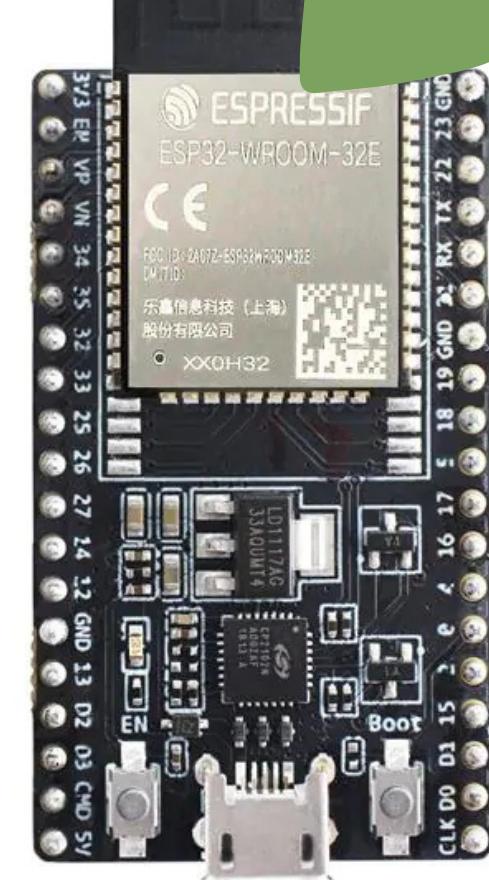




Librevento **Armado**

Materiales

Placa de desarrollo
ESP32 con módulo WiFi
integrado

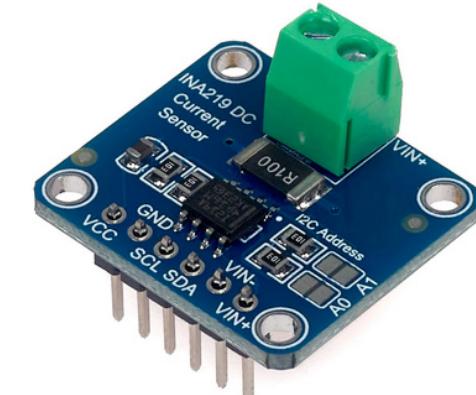


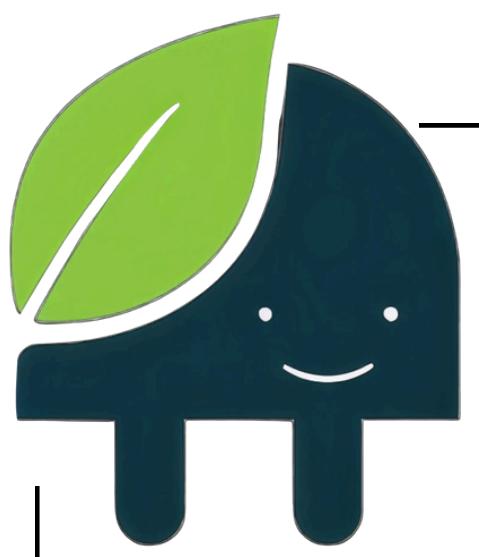
Cables



Sensor de corriente
LNA219

Para monitorear tanto el voltaje generado por los motores como los wats.



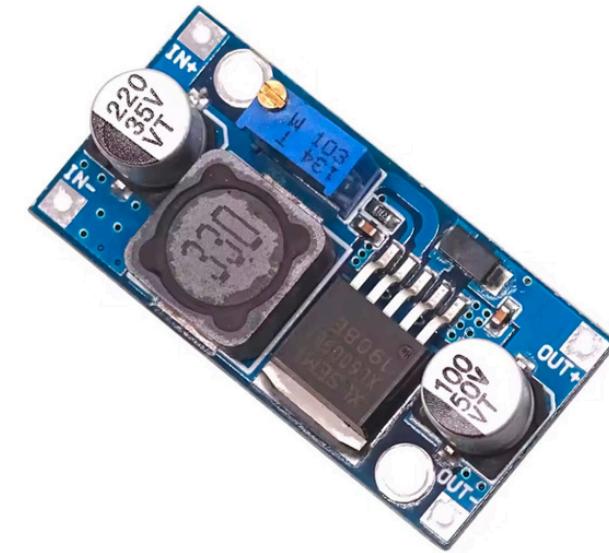


Librevento **Armado**

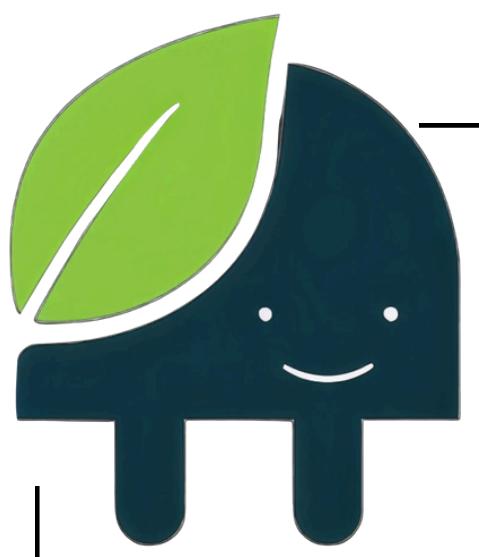
Materiales

Módulo Step-up

Para elevar el voltaje de la energía generada por los motores a un nivel apto para el uso.



Regulador de voltaje casero



Librevento Software

1. Código en placa ESP32

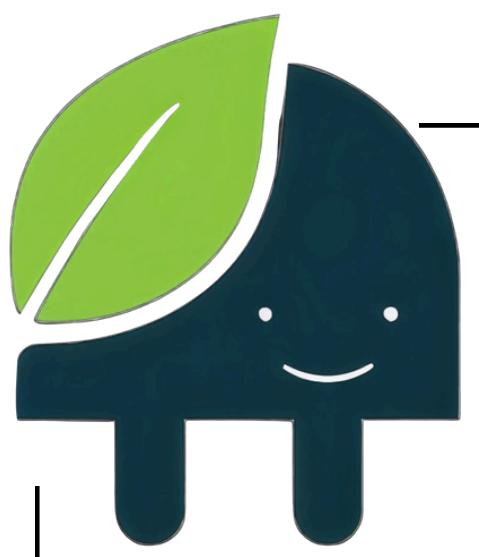
El primer bloque corresponde al software de control de la placa de desarrollo ESP32 previamente mencionada. Al iniciarse el sistema, durante los primeros minutos de operación, este programa establece la conexión inicial entre el dispositivo físico y la base de datos donde se guarda las mediciones de la energía generada.

Esta vinculación se realiza mediante una API desarrollada específicamente para el proyecto, la cual se detalla más adelante en el documento. En cuanto a la programación de la placa, se utilizó el lenguaje C++ dentro del entorno de desarrollo PlatformIO, que nos facilitó la integración de dependencias, compilación de código y pruebas de este.



The screenshot shows the PlatformIO IDE interface. On the left, there's a sidebar with 'PROJECT TASKS' and 'QUICK ACCESS' sections. The 'PROJECT TASKS' section includes options like Default, esp32dev (General: Build, Upload, Monitor, Upload and Monitor, Clean, Full Clean, Devices; Platform: Build Filesystem Image, Program Size). The 'QUICK ACCESS' section includes PIO Home, Open, PIO Account, Inspect, Projects & Configuration, and Libraries. On the right, a code editor window displays a C++ file named 'main.cpp'. The code starts with a void main() function, followed by several lines of code including 'enum', 'Stat', 'int', and 'Wind'. Lines 40 through 51 are visible, showing code like '};', 'Stat', 'int', 'Wind', 'Stat', and '{'.

```
src > C main
19 void
30
31
32
33
34
35
36
37
38
39
40 }
41 enum
42
43
44
45 };
46 Stat
47 int
48
49 Wind
50 Stat
51 {
```



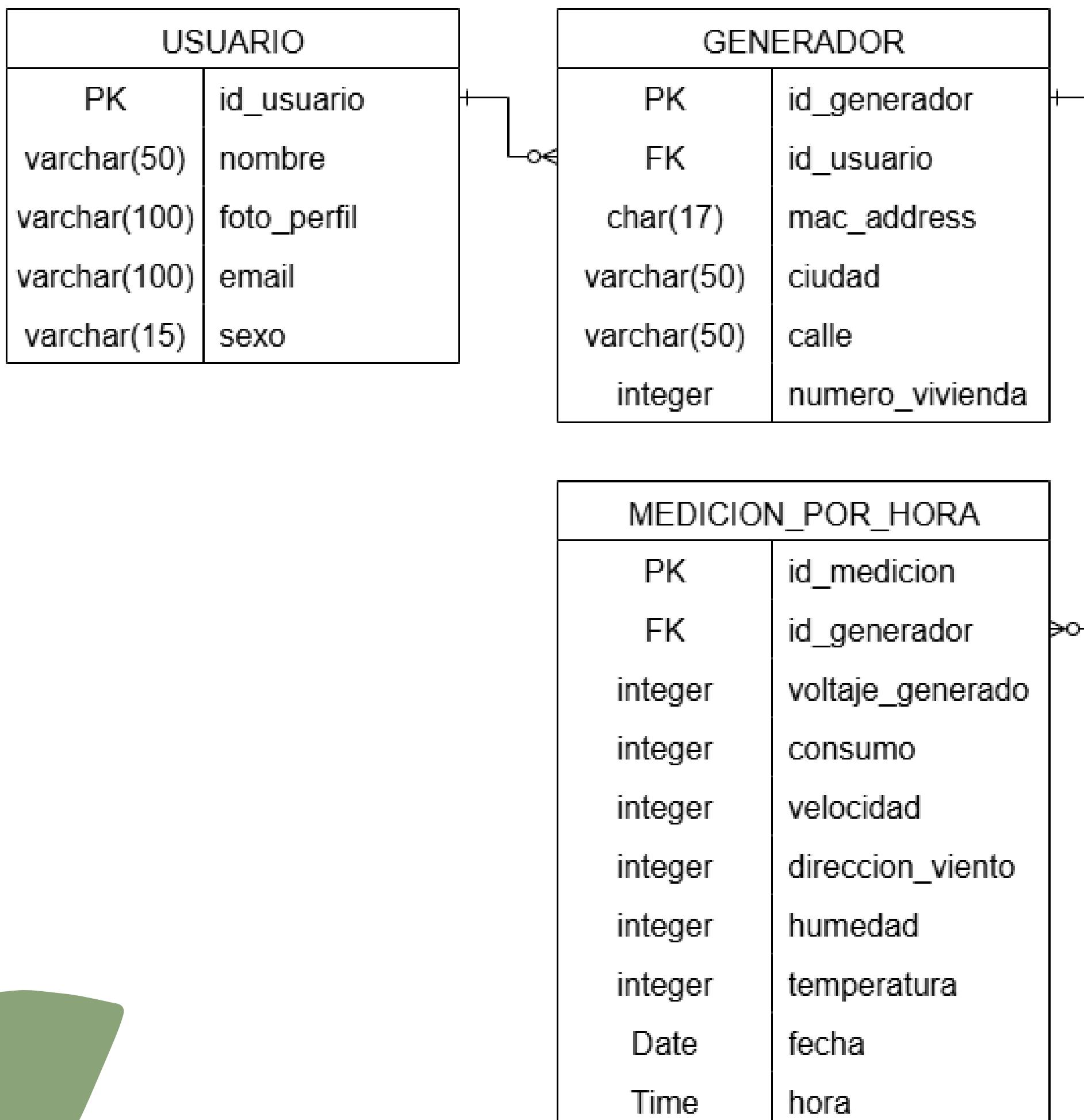
Librevento Software

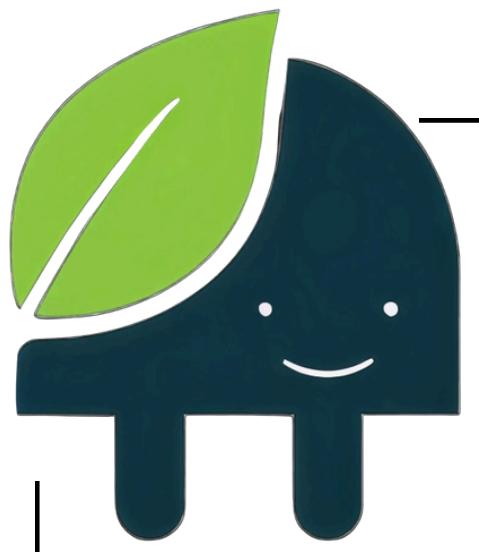
2. Centro de Control



El segundo de estos bloques es el relacionado con la página web para visualizar los datos del generador. Esta etapa inició con el diseño de la interfaz, planteado en Canva, teniendo en cuenta el conseguir una buena navegación y una buena presentación del producto.

Empezamos a diseñar la base de datos que necesitaríamos para almacenar la información, definiendo cómo accederíamos a ella y considerando las conexiones necesarias.





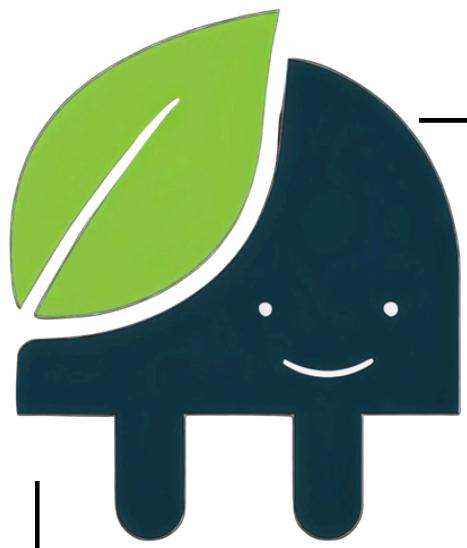
Librevento Software

2. Centro de Control

* Frontend

Esta es la parte visual de la página, en la que utilizamos principalmente la tecnología Svelte junto con SvelteKit para su desarrollo. Elegimos este framework por su sintaxis sencilla y funcionamiento fácil de entender, además de ser ampliamente utilizado en el desarrollo web actual. A lo largo del proceso, incorporamos diferentes librerías que facilitaron la implementación de componentes gráficos para la visualización de datos y recursos decorativos.

The screenshot shows a web application interface. On the left, there's a large orange speech bubble containing the white Svelte logo. Below it is a white sidebar with a blue header bar. The header bar contains the word "Librevento" and two buttons: "Crear Cuenta" and "Iniciar sesión". The sidebar menu includes "Indice", "Armado", "Software", "Centro de Control", "Repositorios", and "¿Quienes somos?". At the bottom of the sidebar, there's a small circular icon with a curved arrow. The main content area has a blue background image of a body of water. In the bottom-left corner of the content area, there's some text: "La necesidad de nuevas formas de generar energía que no contaminen nuestro planeta nos llevó a construir sistemas basados en recursos renovables, que no generen nada -o casi nada- de desechos." and "Librevento es una forma facil y economica de implementar este estilo de vida, con un sencillo armado y control total de lo generado." At the very bottom left, the number "1. Armado" is visible.

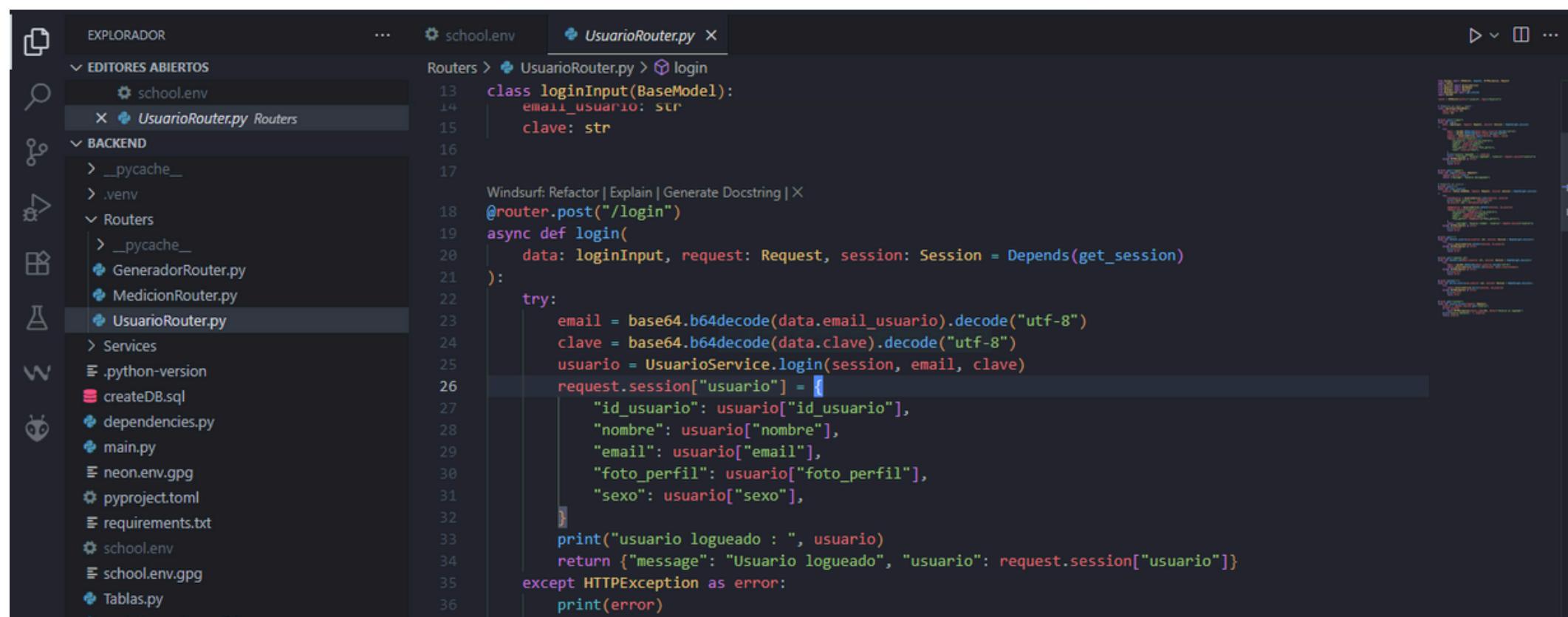


Librevento Software

2. Centro de Control

* Backend

En cuanto al backend, comenzamos con el diseño de la base de datos, la cual está implementada en PostgreSQL, un motor de bases de datos relacional muy utilizado en la actualidad. Además, la adaptamos a MariaDB, ya que este es el motor utilizado para el servidor de la institución. Para conectar la base de datos con el resto del software, creamos una API utilizando Python, junto con librerías especializadas en la creación de APIs, la conexión a bases de datos y algunas adicionales que nos ayudaron a mantener un alto nivel de seguridad tanto durante el desarrollo de la aplicación como en el manejo de datos sensibles: las contraseñas de las cuentas de los usuarios.



```
class loginInput(BaseModel):
    email_usuario: str
    clave: str

@router.post("/login")
async def login(
    data: loginInput, request: Request, session: Session = Depends(get_session)
):
    try:
        email = base64.b64decode(data.email_usuario).decode("utf-8")
        clave = base64.b64decode(data.clave).decode("utf-8")
        usuario = UsuarioService.login(session, email, clave)
        request.session["usuario"] = {
            "id_usuario": usuario["id_usuario"],
            "nombre": usuario["nombre"],
            "email": usuario["email"],
            "foto_perfil": usuario["foto_perfil"],
            "sexo": usuario["sexo"],
        }
        print("usuario logueado : ", usuario)
        return {"message": "Usuario logueado", "usuario": request.session["usuario"]}
    except HTTPException as error:
        print(error)
```

Cabe señalar que, además de las tecnologías mencionadas para la creación del código, utilizamos otras herramientas que facilitaron diversas tareas y nos ayudaron a mantener un orden durante el proceso de desarrollo. Entre ellas se encuentran: Git para gestionar el versionado del proyecto de forma ordenada, GitHub para almacenar el código y poder modificarlo tanto en casa como en el aula sin perder ningún cambio, Vercel para hospedar la página web, Render para desplegar nuestra API de Python en la nube y Neon como servicio de hosting para almacenar nuestra base de datos y acceder a ella a través de la nube.





Por último, este proyecto está orientado a ser un prototipo de una fuente de energía limpia y accesible. Sin embargo, actualmente no genera suficiente electricidad para alimentar un hogar o una habitación.

En el futuro el enfoque del proyecto debería centrarse en aumentar la capacidad de generación de energía, mejorando su eficiencia en la entrega de la misma.

 Librevento