Documento Técnico
HispanoTech-UGC

Por: Sento Marcos Ibarra

Diseño base de dato

Descripción de las tablas

1. usuarios

- **`num_placa`** (varchar) – Clave primaria, identifica al usuario.
- **`password`** (varchar) – Contraseña del usuario.
- **`ultima_sesion`** (timestamp) – Última vez que inició sesión.
- **`rol`** (int4) – Clave foránea a la tabla *`roles`*.
- **`cuerpo`** (int8) – Clave foránea a la tabla *`cuerpos`*.

---

2. roles

- **`id`** (int4) – Clave primaria.
- **`nombre`** (varchar) – Nombre del rol (por ejemplo: admin, operador…).

---

3. cuerpos

- **`id`** (int8) – Clave primaria.
- **`cuerpo`** (varchar) – Nombre del cuerpo (por ejemplo: policía local, guardia civil…).

---

4. informes

- **`informe_id`** (int4) – Clave primaria.
  `fecha_ini` (timestamp) – Fecha de inicio del informe.
- **`fecha_fin`** (timestamp) – Fecha de fin del informe.
- **`num_placa`** (varchar) – Clave foránea a *`usuarios.num_placa`*, indica quién generó el informe.
- **`robot_id`** (int4) – Clave foránea a *`robots.robot_id`*.
- **`titulo_informe`** (varchar) – Título del informe.

---

5. imagen

- **`imagen_id`** (int4) – Clave primaria.
- **`metadatos`** (text) – Información adicional sobre la imagen (por ejemplo, JSON con geolocalización, etiquetas...).

---

6. informeimagen

- **`informe_id`** (int4) – Clave foránea a *`informes`*.
- **`imagen_id`** (int4) – Clave foránea a *`imagen`*.

Tabla intermedia para asociar múltiples imágenes a un informe.

---

7. robots

- **`robot_id`** (int4) – Clave primaria.
- **`modelo`** (varchar) – Modelo del robot.
- **`url`** (varchar) – Posible enlace a la cámara o al panel de control del robot.

---

8. eventos

- **`evento_id`** (int4) – Clave primaria.
- **`robot_id`** (int4) – Clave foránea a *`robots.robot_id`*.

Tipos de usuarios

Roles

- En cuanto a roles se refiere tenemos dos

    - **Admin(id:1)**: que sería el encargado de gestionar el sistema los usuarios de su cuerpo en especifico (ej: un admin de la Guardia Civil unicamente podra gestionar los perfiles de la Guardia civil)

    - **Agente(Id:2)**: Sería referente a todos los usuarios de un cuerpo que deban tener acceso a los drones negociadores

Cuerpo

- Aquí se almacenan los cuerpos de seguridad del estado a los que damos servicio

    - *Cuerpo Nacional De Policia(id:1)*

    - *Guardia Civil(id:2)*

Ejemplo

num_placa	password	ultima_sesion	rol	cuerpo
CNP001	***	2025-04-10 10:00:00	1	1
CNP002	***	2025-04-11 09:30:00	2	1
GCC001	***	2025-04-10 18:45:00	1	2
GCC002	***	2025-04-11 08:15:00	2	2

Leyenda:

- *CNP* = Cuerpo Nacional de Policía (cuerpo ID 1)
- *GCC* = Guardia Civil (cuerpo ID 2)
- Rol 1 = Admin, Rol 2 = Agente

Tecnologías Utilizadas

Hosting web - GitHub Pages

Explicación

La web de **HispanoTechUGC** está alojada mediante **GitHub Pages**, una plataforma gratuita ofrecida por GitHub que permite desplegar sitios web directamente desde un El proceso de despliegue es el siguiente:

1. **Repositorio público o privado** en GitHub que contiene todo el código fuente de la web (HTML, CSS, JS o framework como React/Vite).

2. Se utiliza una **rama específica** (por defecto `main` o `gh-pages`) para servir el contenido estático.

3. En la configuración del repositorio, se habilita **GitHub Pages**, indicando la rama y carpeta raíz (por ejemplo, `/docs` o `/build`).

4. GitHub genera automáticamente una URL del tipo:
   `https://<usuario>.github.io/<nombre-repositorio>/`
   En este caso, por ejemplo:
   `https://hispanotech-ugc.github.io/HispanoTech-UGC-WEB/`

Ventajas de usar GitHub Pages

- **Hosting 100% gratuito** sin necesidad de servidores externos.
- **Despliegue automático** desde Git: solo hace falta hacer `push`.
- **Integración con GitHub Actions** para automatizar tareas como tests o builds.
- **Alta disponibilidad y rendimiento** gracias a la infraestructura de GitHub.
- **Facilidad de mantenimiento** y control de versiones.
- **Personalización del dominio** con soporte HTTPS incluido.
- **Alta tolerancia a caídas de servicios**

Por esta razón se a optado por el uso de esta herramienta en vez de otras como plesk

Hosting BBDD - SUPABASE

La base de datos de **HispanoTechUGC** está alojada en **Supabase**, una plataforma backend de código abierto que proporciona una alternativa moderna y potente a Firebase, basada en **PostgreSQL**.

¿Cómo se está utilizando Supabase?

- Se ha creado un **proyecto Supabase** que contiene toda la estructura de la base de datos relacional.
- Supabase expone una **API RESTful y una API GraphQL** automáticamente a partir del esquema de la base de datos.

Funcionalidades principales que aporta Supabase

- **Base de datos PostgreSQL gestionada**, ideal para relaciones complejas como las de informes, imágenes y usuarios.
- **Realtime**: permite escuchar cambios en tiempo real sobre tablas específicas (ideal para informes activos).
- **Storage**: servicio para almacenar archivos (como imágenes capturadas por drones).
- **Editor visual de base de datos**: para modificar tablas, relaciones y datos fácilmente.
- **Panel de administración** con métricas, logs y editor de SQL.
- **API auto-generada** con documentación automática (PostgREST).

Ventajas de usar Supabase

- **Totalmente gestionado**: no necesitas preocuparte por la infraestructura del servidor de base de datos.
- **Basado en PostgreSQL**, una base de datos relacional robusta, escalable y ampliamente adoptada.
- **Código abierto**: puedes autohostear Supabase si lo deseas en el futuro.
- **Fácil integración con frontend moderno** (Flutter, React, etc.) gracias a sus SDKs.
- **Control de acceso fino**: mediante políticas RLS y autenticación de usuarios.
- **Escalable** y preparado para producción desde el primer día.
- **Soporte para funciones, triggers y extensiones de PostgreSQL.**

Ejemplo de Peticiones a BBDD SupaBase

Login (verificar placa y contraseña)

```
const { data, error } = await supabase
  .from('usuarios')
  .select('*')
  .eq('num_placa', 'CNP001')
  .single();

if (data && data.password === inputPassword) {
  // Login correcto
} else {
  // Placa no existe o contraseña incorrecta
}
```

Actualizar la última sesión tras login

```
await supabase
  .from('usuarios')
  .update({ ultima_sesion: new Date().toISOString() })
  .eq('num_placa', 'CNP001');
```

Obtener todos los informes de un usuario (por placa)

```
const { data, error } = await supabase
  .from('informes')
  .select('*')
  .eq('num_placa', 'CNP001');
```

Crear un nuevo informe

```
const { data, error } = await supabase
  .from('informes')
  .insert([
    {
      fecha_ini: '2025-04-11T09:00:00Z',
      fecha_fin: '2025-04-11T10:00:00Z',
      num_placa: 'GCC002',
      robot_id: 1,
      titulo_informe: 'Supervisión de perímetro rural'
    }
  ]);
```

Asociar imágenes a un informe

```
await supabase
  .from('informeimagen')
  .insert([
    { informe_id: 6, imagen_id: 101 },
    { informe_id: 6, imagen_id: 102 }
  ]);
```

Obtener todas las imágenes de un informe

```
const { data, error } = await supabase
  .from('informeimagen')
  .select('imagen_id, imagen(metadatos)')
  .eq('informe_id', 6);
```

Obtener robots disponibles

```
const { data, error } = await supabase
  .from('robots')
  .select('*');
```

Admin obtiene todos los usuarios de su cuerpo

```
const { data, error } = await supabase
  .from('usuarios')
  .select('num_placa, rol, ultima_sesion')
  .eq('cuerpo', 2);
```

Obtener eventos registrados por un robot

```
const { data, error } = await supabase
  .from('eventos')
  .select('*')
  .eq('robot_id', 1);
```

Comprobar si un usuario es admin

```
const { data, error } = await supabase
  .from('usuarios')
  .select('rol')
  .eq('num_placa', 'GCC001')
  .single();

const esAdmin = data?.rol === 1;
```

Bibliografía

GitHub Pages

1. Documentación oficial de GitHub Pages
   https://docs.github.com/en/pages
2. Configurar GitHub Pages en un repositorio
   https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site
3. Personalizar dominio con GitHub Pages
   https://docs.github.com/en/pages/configuring-a-custom-domain-for-your-github-pages-site

Supabase

4. Documentación oficial de Supabase
   https://supabase.com/docs
5. Supabase Database (PostgreSQL gestionado)
   https://supabase.com/docs/guides/database
6. Supabase JavaScript Client (SDK)
   https://supabase.com/docs/reference/javascript/introduction
7. Autenticación personalizada
   https://supabase.com/docs/guides/auth/auth-helpers
8. Realtime Database en Supabase
   https://supabase.com/docs/guides/realtime
9. Storage: almacenamiento de archivos en Supabase
   https://supabase.com/docs/guides/storage
10. Row Level Security (RLS)
    https://supabase.com/docs/guides/auth/row-level-security