

# Satellite

**Integrantes del grupo** sólo Mateo Crimella

**Fecha** 2025-09-17

**Versión** 1

**Repositorio** <https://github.com/Bowuigi/Satellite>

## Descripción

Una red social centrada en filtros (como los de un motor de búsqueda) e hilos (como los comentarios de Reddit o las publicaciones de X), producto de un rediseño de <https://codeberg.org/Bowuigi/Satellite>.

Utiliza un login con nombre de usuario y una contraseña.

La página inicial de la red social muestra una lista de filtros, que el usuario puede elegir para ver las publicaciones asociadas a estos.

Un filtro es un query sencillo cuyo fin es limitar la cantidad de publicaciones mostradas. Permite búsquedas en el texto, filtrar por usuarios particulares, filtrar por cantidad o relación de votos y filtrar por fecha de publicación.

A futuro (ya no es a futuro desde la separación frontend-backend), permitir a los usuarios ver los resultados de aplicar un filtro en formato JSON sería útil para integrarlo con otros sistemas.

Incluye dos roles: Administrador de bases de datos, con acceso a todo pero solo desde la base de datos, y usuario, que no puede borrar nada, pero puede usar el servicio normalmente.

## Objetivos

Por un lado, conectar personas de forma que se respeten sus intereses, teniendo control total sobre su “algoritmo”; por el otro, reducir el concepto de “red social” al mínimo posible sin remover funcionalidades críticas.

Convenientemente, su diseño actual generaliza el concepto de “red social” lo suficiente como para que sea útil fuera de ese contexto, por lo que podría utilizarse para integrar distintas fuentes de información (canales de noticias, actualizaciones, avisos generales, reportes sobre servicios) de manera que se puedan filtrar y ordenar a gusto. En este aspecto, es similar a <https://ntfy.sh/>.

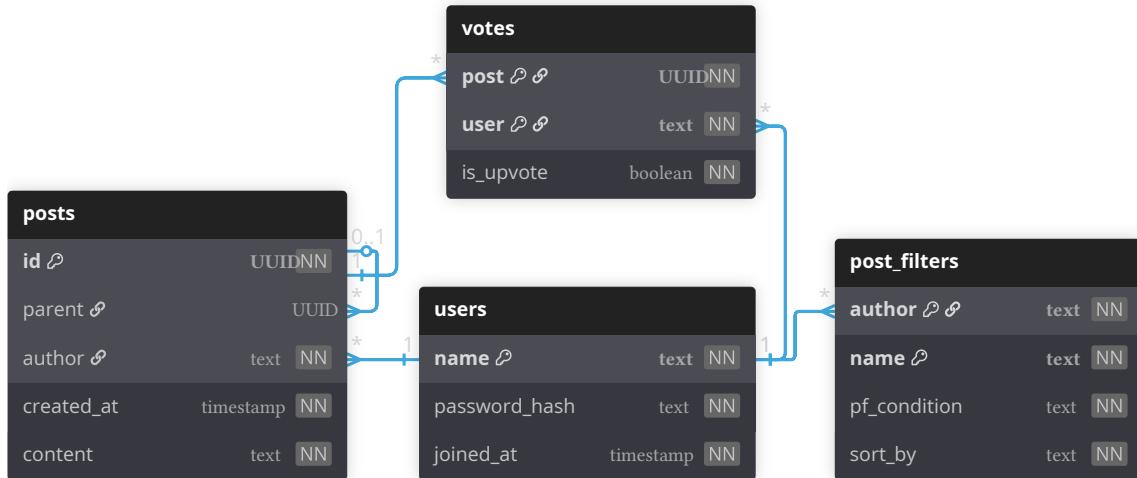
## Requisitos funcionales

- Autenticación + autorización
  - Registro con un nombre de usuario y una contraseña (mediante un login)
  - Inicio de sesión (mismos datos)
- Publicaciones e hilos
  - Lista de publicaciones en el mismo nivel (en formato HTML y opcionalmente JSON)
  - Creación de publicaciones por parte de cualquier usuario en cualquier nivel
  - Posibilidad de votar positiva o negativamente a una publicación
- Filtros aplicados
  - Lista de filtros
  - Aplicación de filtros para limitar las publicaciones visualizadas
- Lenguaje de filtrado
  - Filtrado por texto
  - Filtrado por nombre de usuario
  - Filtrado por reputación
  - Filtrado por fecha de publicación
  - Combinación de filtros utilizando conjunción
  - Combinación de filtros utilizando disyunción

## Stack elegido

- Podman + docker-compose para armar el entorno (es compatible con docker + docker-compose, opcional)
- MariaDB como base de datos, sin gestor adicional
- Caddy como reverse proxy (se usa php -S localhost:8080 como servidor del backend)
- PHP como lenguaje para el backend
- HTML + CSS + JS para el frontend, usando funcionalidades con 90%+ en [Can I use](#)

# Diagrama Entidad-Relación



## SQL

```
create database satellite;
use satellite;
```

```
create table users (
    name          text not null primary key,
    password_hash text not null,
    joined_at     timestamp not null
);

create table posts (
    id            UUID not null primary key,
    parent        UUID,
    author        text not null,
    created_at    timestamp not null,
    content       text not null,
    foreign key (parent) references posts (id),
    foreign key (author) references users (name)
);
```

```
create table votes (
    post          UUID not null,
    user          text not null,
    is_upvote    boolean not null,
    foreign key (post) references posts(id),
    foreign key (user) references users(name),
    primary key (post, user)
);

create table post_filters (
    name          text not null,
    author        text not null,
    pf_condition  text not null,
    sort_by       text not null,
    foreign key (author) references users(name),
    primary key (name, author)
);
```