**Plan de Acción Detallado**

**Paso 1: Base de Datos y Modelos (Los Cimientos)**

Antes de poder guardar o mostrar cualquier dato nuevo, necesitamos que nuestra base de datos y nuestros modelos de Laravel sepan que existen.

1. **Ampliar la tabla users:**
   * **Acción:** Crear una nueva migración para añadir las columnas necesarias a la tabla users.
   * **Columnas a añadir:**
     + username (string, único, para URLs de perfil amigables como /perfil/nombredeusuario).
     + avatar\_path (string, nullable, para guardar la ruta a la foto de perfil).
     + bio (text, nullable, para la biografía del usuario).
     + social\_links (json, nullable, para guardar los enlaces a redes sociales de forma flexible).
   * **Archivo a crear:** database/migrations/xxxx\_add\_profile\_fields\_to\_users\_table.php
2. **Crear el sistema de "Seguimiento de Usuarios" (Usuarios Afines):**
   * **Acción:** Crear una tabla pivote para gestionar la relación "muchos a muchos" entre usuarios. Un usuario puede seguir a muchos otros, y ser seguido por muchos otros.
   * **Tabla a crear:** followers (o user\_follows).
   * **Columnas:** user\_id (quien sigue), following\_user\_id (a quien se sigue).
   * **Archivo a crear:** database/migrations/xxxx\_create\_followers\_table.php
3. **Crear el sistema de "Posts Favoritos":**
   * **Acción:** Crear otra tabla pivote para la relación "muchos a muchos" entre usuarios y posts.
   * **Tabla a crear:** favorite\_posts (o post\_user).
   * **Columnas:** user\_id, post\_id.
   * **Archivo a crear:** database/migrations/xxxx\_create\_favorite\_posts\_table.php
4. **Actualizar el Modelo User:**
   * **Acción:** Definir las nuevas relaciones y propiedades en el modelo User.
   * **Cambios a realizar:**
     + Añadir las nuevas columnas (username, avatar\_path, bio, social\_links) al array $fillable para permitir la asignación masiva.
     + Crear la relación following() para obtener los usuarios que sigue.
     + Crear la relación followers() para obtener los usuarios que le siguen.
     + Crear la relación favoritePosts() para obtener sus posts favoritos.
   * **Archivo a modificar:** app/Models/User.php
5. **Actualizar el Modelo Post:**
   * **Acción:** Definir la relación inversa para saber qué usuarios han marcado un post como favorito.
   * **Cambios a realizar:**
     + Crear la relación favoritedBy() para obtener los usuarios que han añadido el post a favoritos.
   * **Archivo a modificar:** app/Models/Post.php

**Paso 2: Lógica del Backend (El Motor)**

Con la base de datos lista, ahora programamos la lógica para manejar toda esta nueva información.

1. **Actualizar el ProfileController:**
   * **Acción:** Modificar el método update para que guarde la nueva información del perfil.
   * **Lógica a añadir:**
     + Validar y guardar username, bio y social\_links.
     + Procesar la subida de la foto de perfil: si el usuario sube una, validarla, guardarla en storage/app/public/avatars, borrar la antigua (si existe) y actualizar la columna avatar\_path.
   * **Archivo a modificar:** app/Http/Controllers/ProfileController.php
2. **Actualizar el ProfileUpdateRequest:**
   * **Acción:** Añadir las reglas de validación para los nuevos campos del formulario.
   * **Reglas a añadir:**
     + Validación para username (requerido, único, alfanumérico).
     + Validación para avatar (opcional, tipo imagen, tamaño máximo).
     + Validación para bio y social\_links.
   * **Archivo a modificar:** app/Http/Requests/ProfileUpdateRequest.php
3. **Crear la lógica para Seguir/Dejar de Seguir:**
   * **Acción:** Crear un nuevo controlador y rutas para esta funcionalidad.
   * **Controlador a crear:** FollowController con métodos store (seguir) y destroy (dejar de seguir).
   * **Rutas a añadir:** POST /users/{user}/follow y DELETE /users/{user}/follow.
   * **Archivos a crear/modificar:** app/Http/Controllers/FollowController.php y routes/web.php.
4. **Crear la lógica para Favorito/Quitar Favorito:**
   * **Acción:** Similar al sistema de seguimiento, pero para posts.
   * **Controlador a crear:** FavoritePostController con métodos store y destroy.
   * **Rutas a añadir:** POST /posts/{post}/favorite y DELETE /posts/{post}/favorite.
   * **Archivos a crear/modificar:** app/Http/Controllers/FavoritePostController.php y routes/web.php.
5. **Configurar el Almacenamiento de Archivos:**
   * **Acción:** Ejecutar un comando de Artisan para crear el enlace simbólico que hará públicos los archivos subidos.
   * **Comando:** php artisan storage:link

**Paso 3: Interfaz de Usuario (Vistas)**

Finalmente, creamos y modificamos las vistas para que el usuario pueda ver e interactuar con todo lo que hemos construido.

1. **Modificar la Vista de Edición de Perfil:**
   * **Acción:** Añadir los nuevos campos al formulario de edición de perfil que ya existe.
   * **Campos a añadir:**
     + Input para username.
     + Textarea para bio.
     + Input de tipo file para la foto de perfil (avatar).
     + Inputs para los enlaces de redes sociales.
     + Mostrar la foto de perfil actual.
   * **Importante:** Añadir enctype="multipart/form-data" a la etiqueta <form> para permitir la subida de archivos.
   * **Archivo a modificar:** resources/views/profile/edit.blade.php (y sus parciales en resources/views/profile/partials/).
2. **Crear la Página de Perfil Pública:**
   * **Acción:** Crear una nueva ruta y vista para que se pueda ver el perfil de cualquier usuario (p. ej., tusitio.com/perfil/nombredeusuario).
   * **Ruta a añadir:** GET /perfil/{user:username}.
   * **Controlador a modificar:** Añadir un método show al ProfileController (o al UserController) para manejar esta vista.
   * **Vista a crear:** resources/views/profile/show.blade.php.
   * **Contenido de la vista:**
     + Información del usuario (avatar, nombre, username, bio, redes sociales).
     + Botón para "Seguir" / "Dejar de seguir" (visible solo para usuarios autenticados que no sean el propio perfil).
     + Pestañas o secciones para mostrar:
       - Últimos comentarios del usuario.
       - Posts favoritos del usuario.
       - Lista de usuarios a los que sigue y que le siguen.

Este es el mapa completo. Como ves, es un trabajo considerable, pero si lo hacemos paso a paso, será muy manejable y el resultado será excelente.

Cuando quieras, dime y **empezamos con el "Paso 1.1: Ampliar la tabla users"**