93. Recibiendo y manejando archivos en Laravel - imagen de perfil del usuario

config\filesystems.php

```php
<?php

return [

    /*
    |--------------------------------------------------------------------------
    | Default Filesystem Disk
    |--------------------------------------------------------------------------
    |
    | Here you may specify the default filesystem disk that should be used
    | by the framework. The "local" disk, as well as a variety of cloud
    | based disks are available to your application. Just store away!
    |
    */

    'default' => env('FILESYSTEM_DRIVER', 'local'),

    /*
    |--------------------------------------------------------------------------
    | Filesystem Disks
    |--------------------------------------------------------------------------
    |
    | Here you may configure as many filesystem "disks" as you wish, and you
    | may even configure multiple disks of the same driver. Defaults have
    | been setup for each driver as an example of the required options.
    |
    | Supported Drivers: "local", "ftp", "sftp", "s3"
    |
    */

    'disks' => [
        'images' => [
            'driver' => 'local',
            // Guardamos en la carpeta images (debemos duplicar el gitignore para
            // que la carpeta exista siempre)
            'root' => storage_path('app/public/images'),
            'url' => env('APP_URL').'/images',
            'visibility' => 'public',
        ],

        'local' => [
            'driver' => 'local',
            'root' => storage_path('app'),
        ],

        'public' => [
            'driver' => 'local',
            'root' => storage_path('app/public'),
            'url' => env('APP_URL').'/storage',
            'visibility' => 'public',
        ],
```

```
        's3' ⇒ [
            'driver' ⇒ 's3',
            'key' ⇒ env('AWS_ACCESS_KEY_ID'),
            'secret' ⇒ env('AWS_SECRET_ACCESS_KEY'),
            'region' ⇒ env('AWS_DEFAULT_REGION'),
            'bucket' ⇒ env('AWS_BUCKET'),
            'url' ⇒ env('AWS_URL'),
            'endpoint' ⇒ env('AWS_ENDPOINT'),
            'use_path_style_endpoint' ⇒ env('AWS_USE_PATH_STYLE_ENDPOINT', false),
        ],

    ],

    /*
    |--------------------------------------------------------------------------
    | Symbolic Links
    |--------------------------------------------------------------------------
    |
    | Here you may configure the symbolic links that will be created when the
    | `storage:link` Artisan command is executed. The array keys should be
    | the locations of the links and the values should be their targets.
    |
    */
    // Enlaces simbólicos (las imágenes se redirigen a esta ruta)
    'links' ⇒ [
        public_path('storage') ⇒ storage_path('app/public'),
        public_path('images') ⇒ storage_path('app/public/images'),
    ],

];
```

Duplicar .gitignore local

Copiamos el .gitignore de la carpeta public dentro de la nueva carpeta images.

.gitignore del proyecto

Añadir `/public/images`.

En cmd:

`php artisan storage:link` para que Laravel coja los nuevos links simbólicos del archivo filesystems.php.

Commit (asegurarse)

Sobre todo las definiciones importadas son muy importantes.
Commit.

Prueba

```
>>> User::find(21)→image
⇒ App\Models\Image {#4622
    id: "169",
    path: "users/O8btBCcm8ZbEO8ODscBdaxBh4A8ijtLvm2egJCeo.webp",
    created_at: "2023-02-10 10:51:05",
```

```
        updated_at: "2023-02-10 10:51:05",
        imageable_type: "App\Models\User",
        imageable_id: "21",
    }

>>> User::find(21)→image
⇒ App\Models\Image {#4628
        id: "170",
        path: "users/lHB2XiiHcNZiy9zxElSkd7N8ljkmvXljOZINxePu.jpg",
        created_at: "2023-02-10 10:54:10",
        updated_at: "2023-02-10 10:54:10",
        imageable_type: "App\Models\User",
        imageable_id: "21",
    }

>>>
```