

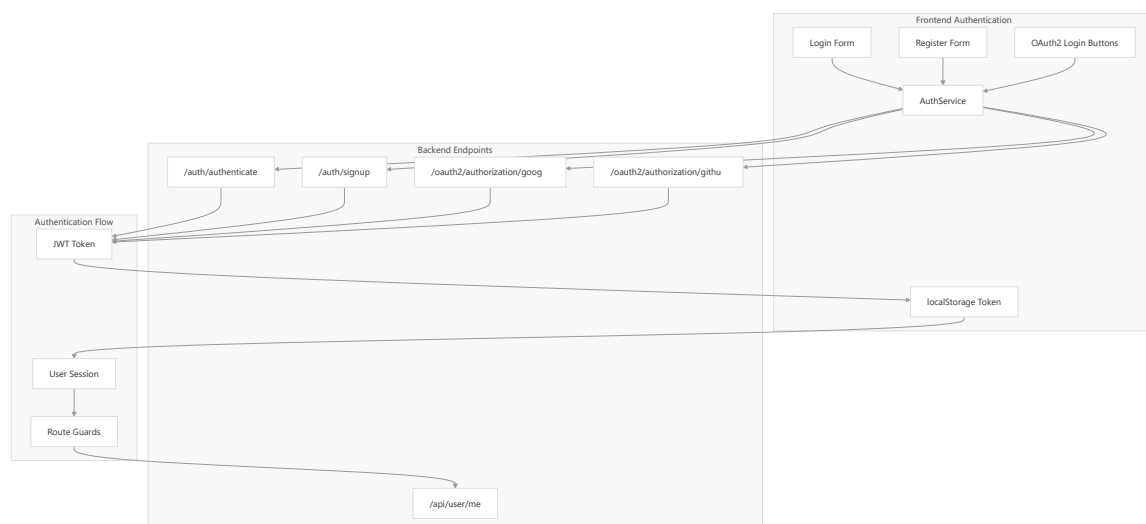
Panel de usuario y autenticación

Propósito y alcance

Este documento describe la interfaz del panel de usuario y el sistema de autenticación de la aplicación frontend de Angular. Detalla cómo los usuarios se autentican, gestionan sus perfiles e interactúan con la interfaz principal del panel. Para obtener información sobre la autenticación y la configuración de seguridad del backend, consulte [Seguridad y autenticación](#) . Para obtener más información sobre las funciones específicas del panel para administradores, consulte [Panel de administración](#) .

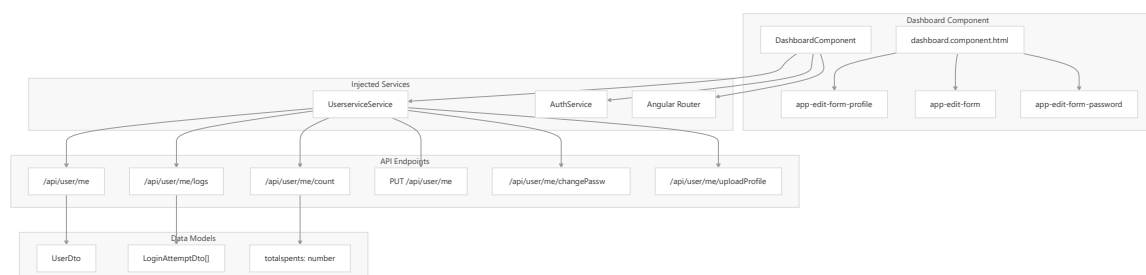
Arquitectura del sistema de autenticación

El sistema de autenticación admite múltiples métodos de inicio de sesión y mantiene las sesiones de usuario a través de tokens JWT.



Arquitectura de componentes del tablero

El componente del panel integra múltiples servicios para proporcionar una interfaz de usuario integral.




Flujo de autenticación

Proceso de inicio de sesión

Proporciona `AuthService` múltiples métodos de autenticación:

Método	Punto final	Descripción
<code>login()</code>	<code>/auth/authenticate</code>	Autenticación de nombre de usuario/contraseña
<code>register()</code>	<code>/auth/signup</code>	Registro de nuevo usuario
<code>loginWithOAuth2()</code>	<code>/oauth2/authorization/{provider}</code>	Autenticación OAuth2
<code>forgotPassword()</code>	<code>/public/forgot-password</code>	Recuperación de contraseña

El flujo de autenticación sigue este patrón:

1. El usuario envía sus credenciales a través del formulario de inicio de sesión
2. `AuthService.login()` envía una solicitud POST a `/auth/authenticate`
 `ServidorWebFrontend/gesthorAngular/src/app/services/auth.service.ts` | 64-67
3. El backend devuelve el token JWT tras una autenticación exitosa
4. Token almacenado en el método `localStorage` vía `getToken()`
 `ServidorWebFrontend/gesthorAngular/src/app/services/auth.service.ts` | 95-97
5. `isAuthenticated()` Comprueba la presencia del token para la protección de la ruta
 `ServidorWebFrontend/gesthorAngular/src/app/services/auth.service.ts` | 136-139



Integración OAuth2


La autenticación OAuth2 es compatible con los proveedores de Google y GitHub:

```
loginWithOAuth2(provider: 'google' | 'github') {  
  window.location.href = `${environment.apiUrl}/oauth2/authorization/${provider}`;  
}
```

Gestión de sesiones

La `AuthService` gestión de las sesiones de usuario se realiza a través de:

- `logout()` :Elimina el token del almacenamiento local y borra los datos del usuario
 `ServidorWebFrontend/gesthorAngular/src/app/services/auth.service.ts` | 111-114
- `isAuthenticated()` : Valida la presencia del token
 `ServidorWebFrontend/gesthorAngular/src/app/services/auth.service.ts` | 136-139

- **isAdmin()** : Comprueba el rol del usuario para el acceso de administrador
 ServidorWebFrontend/gesthorAngular/src/app/services/auth.service.ts | 120-122

Gestión de perfiles de usuario


Estructura de datos del perfil

El panel de control muestra y administra los datos del perfil del usuario a través del **UserDto** modelo:

Campo	Tipo	Descripción
id	número	Identificador de usuario
username	cadena	Nombre de usuario único
name	cadena	Nombre para mostrar
email	cadena	Dirección de correo electrónico
phone	cadena	Número de teléfono
address	cadena	Dirección física
imageUrl	cadena	URL de la imagen de perfil
role	cadena	Rol de usuario (USUARIO/ADMINISTRADOR)
active	booleano	Estado de la cuenta
createdAt	cadena	Fecha de creación de la cuenta
updatedAt	cadena	Fecha de la última modificación

Componentes de edición de perfiles

El panel de control utiliza componentes especializados para diferentes campos de perfil:

- **app-edit-form-profile** :Componente de carga de imágenes
 ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html | 16-22
- **app-edit-form** :Editor de campos de texto
 ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html | 25-62
- **app-edit-form-password** :Componente de cambio de contraseña
 ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html | 78-80

Proceso de actualización de perfil





Las actualizaciones de perfil se gestionan a través del **guardarCampo()** método:

1. El componente recibe un evento de actualización de campo
2. Actualiza `user` el objeto local con un nuevo valor
3. Llamadas `UserserviceService.actualizarUsuario()`
4. Actualiza el campo del servidor desde `environment.apiName`
5. Muestra alertas de éxito/error y recarga datos en caso de error

Características del panel de control

Visualización de estadísticas de usuario

El panel muestra métricas clave del usuario:

- **Gastos totales** : recuperados a través de `getTotalSpents()`
 `ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.ts` | 156-158
- **Rol de la cuenta** : muestra el rol de USUARIO o ADMINISTRADOR
 `ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html` | 99-100
- **Estado de la cuenta** : muestra el estado activo/inactivo
 `ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html` | 111-112
- **Miembro desde** : Fecha de creación de la cuenta
 `ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html` | 105-106

Visualización del historial de inicio de sesión

El panel muestra los intentos de inicio de sesión recientes a través de la `LoginAttemptDto` matriz:


```
logs: LoginAttemptDto[] = [];
```



El historial de inicio de sesión muestra:

- Marca de tiempo del intento formateada con `date: 'medium'` barra vertical
- Estado de éxito/fracaso con clases CSS condicionales
- Limitado a los 10 intentos más recientes

Navegación y acciones

El panel de control proporciona botones de acción para:

- **Cerrar sesión** : `borrarToken()` el método borra la sesión y redirecciona al inicio de sesión
 `ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.ts` | 164-167

- **Configuración** : Botón deshabilitado para la página de configuración futura
 ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html 67
- **Panel de administración** : se muestra condicionalmente para usuarios con rol de ADMINISTRADOR
 ServidorWebFrontend/gesthorAngular/src/app/pages/private/dashboard/dashboard.component.html 68

Características de seguridad

Gestión de contraseñas

Los cambios de contraseña se gestionan a través de un flujo de trabajo dedicado:

1. **app-edit-form-password** El componente captura contraseñas actuales y nuevas
2. **cambiarPassword()** El método procesa la solicitud de cambio
3. **UserServiceService.cambiarPassword()** Envía una actualización cifrada al backend
4. Confirmación de éxito mostrada al usuario

Pantalla de autenticación de dos factores

El panel muestra el estado de 2FA a través del `user.fv2` campo:

```
<span>Verificación en 2 pasos: {{user.fv2 ? 'Activa' : 'Desactivada'}}</span>
```

El botón de alternancia está actualmente deshabilitado, lo que indica que la administración de 2FA aún no está implementada.

Control de acceso basado en roles

El acceso de administrador se controla mediante la verificación de roles:

```
<button *ngIf="user.role == 'ADMIN'" ... (click)="redirectTo('/private/admindashboard')">Admin</button>
```

El `AuthService.isAdmin()` método proporciona verificación de roles centralizada para la aplicación.