

Aplicación frontend

Este documento describe la aplicación frontend de Angular que sirve como interfaz de usuario para el sistema de gestión de gastos ProyectoGestorGastos. La interfaz proporciona paneles de control, interfaces de gestión de gastos, flujos de autenticación y funciones de visualización de datos.

Para obtener más información sobre la integración de la API de backend, consulte [Servicios de backend](#) . Para la implementación del sistema de autenticación, consulte [Seguridad y autenticación](#) . Para obtener información sobre la funcionalidad del frontend específica para el administrador, consulte [Panel de administración](#) .

Arquitectura de la aplicación

El frontend de Angular sigue una arquitectura modular con una clara separación entre rutas públicas, protegidas y privadas. La aplicación está desarrollada con Angular 18 y TypeScript y se integra con el backend de Spring Boot mediante llamadas a la API HTTP.

Estructura del módulo principal

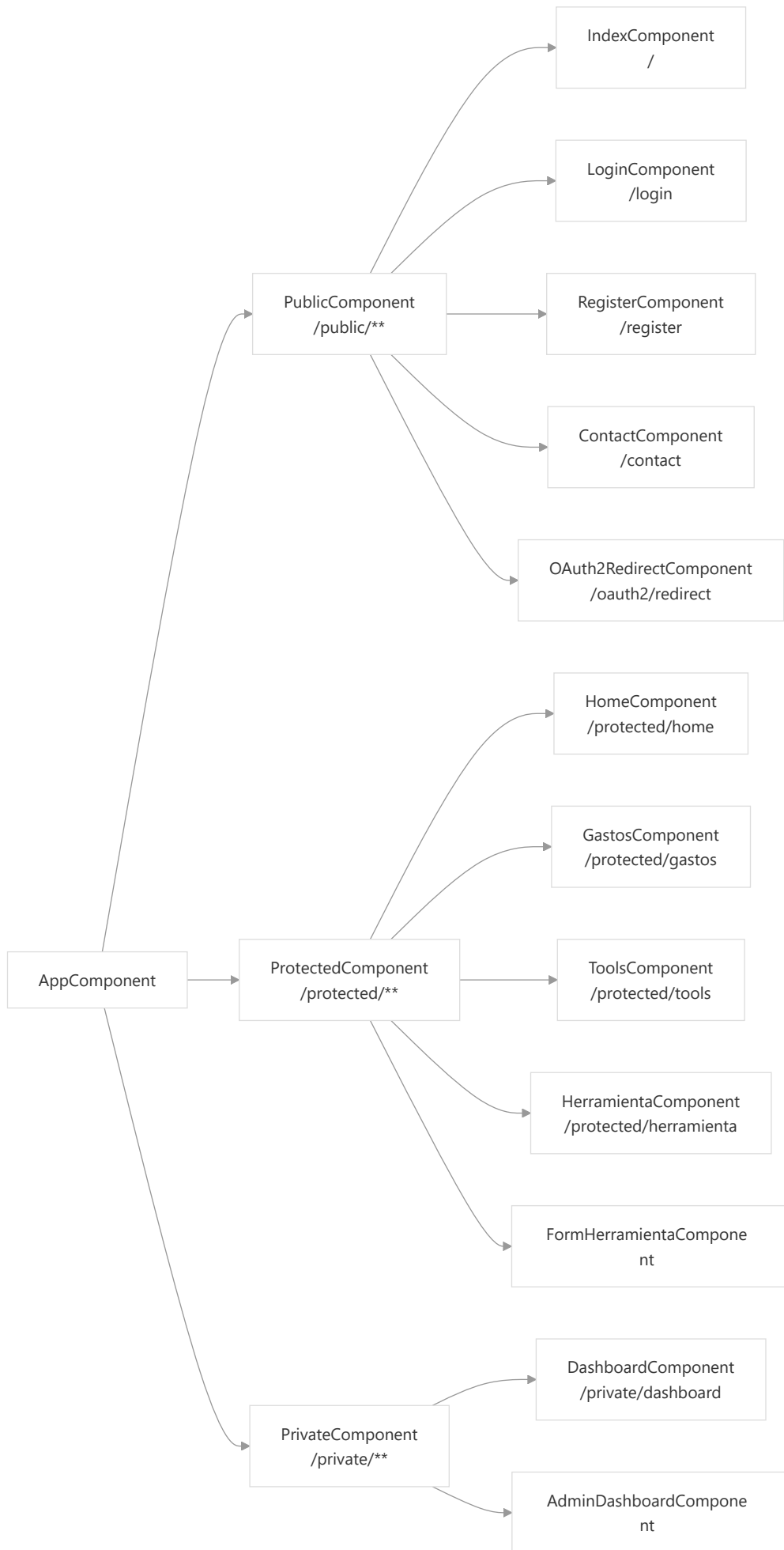


Configuración del módulo de aplicación

El módulo principal de la aplicación configura todos los componentes, servicios e interceptores.

TokenInterceptor Agrega automáticamente tokens JWT a las solicitudes HTTP para los endpoints autenticados.

Arquitectura de enrutamiento de componentes



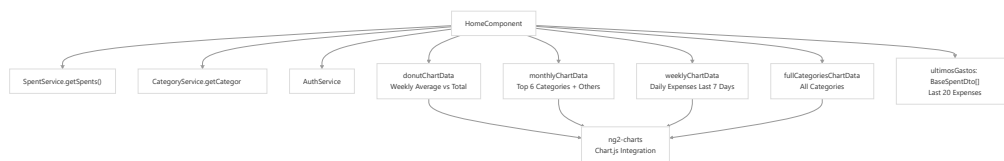
Niveles de protección de ruta

- **Rutas públicas** : Accesibles sin autenticación (`IndexComponent` , `LoginComponent` , `RegisterComponent`)
- **Rutas protegidas** : Requieren autenticación de usuario (`HomeComponent` , `GastosComponent`)
- **Rutas privadas** : Requiere rol de administrador (`AdminDashboardComponent`)

Componentes y características clave

Componente del panel de inicio

Sirve `HomeComponent` como panel principal que muestra análisis de gastos a través de múltiples tipos de gráficos y listas de gastos recientes.



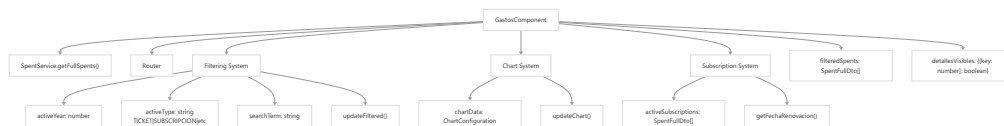
Procesamiento de datos de gráficos

El componente procesa datos de gastos en múltiples formatos de visualización:

- `actualizarDonutChart()` : Calcula el gasto promedio semanal vs. el gasto total
- `actualizarMonthlyChart()` : Agrupa los gastos por categoría, muestra los 6 principales más "Otros".
- `actualizarWeeklyChart()` : Mapas de gastos diarios de los últimos 7 días
- `actualizarFullCategoriesChart()` : Muestra todas las proporciones de categorías

Componente de gestión de gastos

Proporciona `GastosComponent` capacidades integrales de visualización, filtrado y análisis de gastos.

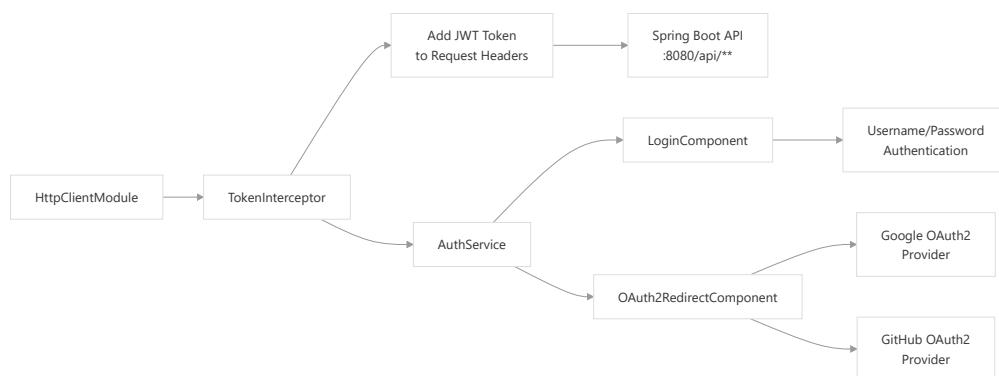


Funciones de filtrado avanzadas

- **Navegación anual** : `changeYear()` y `canNavigateYear()` los métodos permiten navegar los gastos por año
- **Filtrado de tipos** : filtrar por `ExpenseClass` valores de enumeración (`TICKET` , `SUBSCRIPCION` , `TRANSFERENCIA` , `GASTO_GENERICO`)
- **Búsqueda de texto** : busca en los campos `name` , `description` , y analizados `productsJSON`
- **Gestión de suscripciones** : realiza un seguimiento de las suscripciones activas con cálculos de fecha de renovación

Integración de autenticación

Gestión de tokens e interceptores



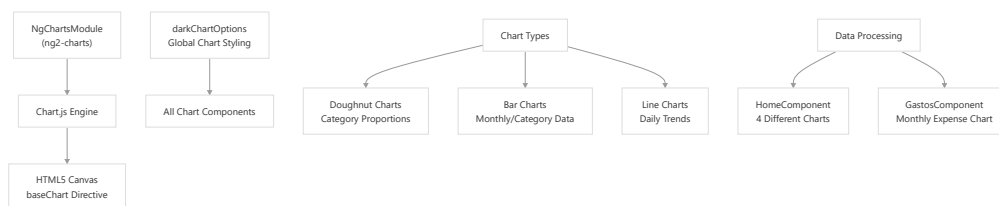
Componentes del flujo de autenticación

- **TokenInterceptor** : Adjunta automáticamente tokens JWT a solicitudes HTTP
- **LoginComponent** : Maneja la autenticación de nombre de usuario/contraseña
- **OAuth2RedirectComponent** : Procesa devoluciones de llamadas OAuth2 de proveedores externos
- **RegisterComponent** : Registro de usuario con validación
- **ForgotPasswordComponent** : Funcionalidad de recuperación de contraseña

Sistema de visualización de datos

Integración con Chart.js

La aplicación utiliza **ng2-charts** una biblioteca para la visualización de datos con el backend Chart.js. Todos los gráficos tienen un estilo de tema oscuro para mayor consistencia.



Estándares de configuración de gráficos

Todos los gráficos utilizan **darkChartOptions** una configuración estandarizada:

- Etiquetas y leyendas de texto en blanco
- Líneas de cuadrícula oscuras (**#444**)
- Diseño responsivo con **maintainAspectRatio: false**
- Paleta de colores consistente mediante **getChartColor()** el método

Componentes de interfaz de usuario reutilizables

Componentes de formulario y entrada

Tipo de componente	Ubicación	Objetivo
Seguridad	<code>/security/</code>	Componentes de autenticación (<code>LoginComponent</code> , <code>RegisterComponent</code> , <code>OAuth2RedirectComponent</code>)
Páginas públicas	<code>/pages/public/</code>	Páginas de acceso público (<code>IndexComponent</code> , <code>ContactComponent</code>)
Páginas protegidas	<code>/pages/protected/</code>	Páginas autenticadas por el usuario (<code>HomeController</code> , <code>GastosComponent</code> , <code>ToolsComponent</code>)
Páginas privadas	<code>/pages/private/</code>	Páginas solo para administradores (<code>AdminDashboardComponent</code> , <code>DashboardComponent</code>)
Componentes	<code>/components/</code>	Componentes de interfaz de usuario compartidos (<code>NavbarComponent</code> , <code>FooterComponent</code>)