

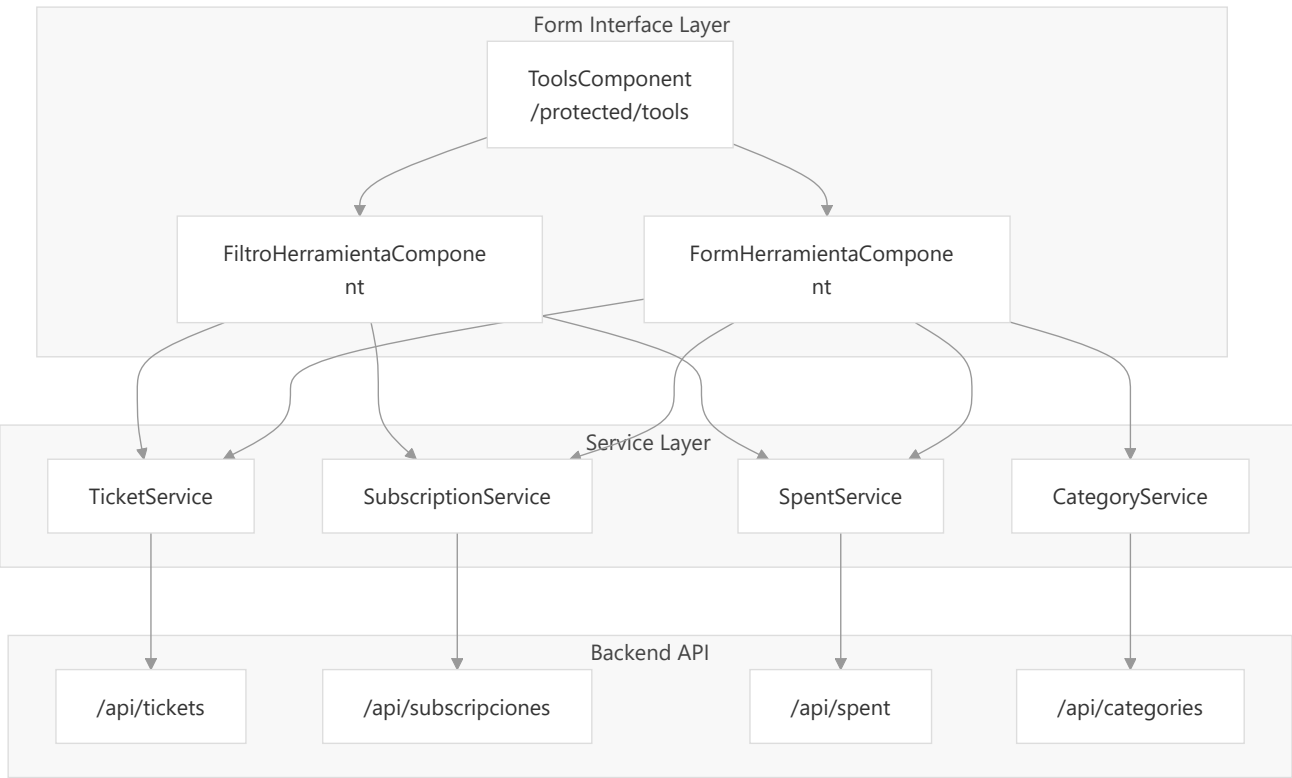
# Interfaz de gestión de gastos

Este documento describe los componentes e interfaces frontend de Angular responsables de la gestión de registros financieros en el sistema de seguimiento de gastos. La interfaz de gestión de gastos proporciona funcionalidad CRUD (Crear, Leer, Actualizar, Eliminar) para tickets, suscripciones y gastos generales mediante componentes de formulario dedicados e interfaces de filtrado.

Para la autenticación de usuarios y la funcionalidad del panel, consulte [Panel de usuario y autenticación](#) . Para la supervisión del sistema administrativo, consulte [Panel de administración](#) . Para los componentes de interfaz de usuario reutilizables que se utilizan en estas interfaces, consulte [Componentes de interfaz de usuario y navegación](#) .

## Arquitectura de la interfaz de formulario

La interfaz de gestión de gastos está construida alrededor de un sistema de formulario unificado que maneja tres tipos de registros financieros: tickets (recibos), suscripciones (gastos recurrentes) y gastos generales.



# Componente del formulario de gastos

Sirve `FormHerramientaComponent` como interfaz principal para crear y editar registros financieros. Utiliza un formulario dinámico que se adapta según el tipo de gasto.

## Estructura del componente

Propiedad	Tipo	Objetivo
<code>tipo</code>	<code>'ticket'   'suscripcion'   'gasto'</code>	Determina los campos del formulario y la validación.
<code>id</code>	<code>number</code>	Identificador de registro (0 para registros nuevos)
<code>form</code>	<code>FormGroup</code>	Instancia de formulario reactivo
<code>categories</code>	<code>CategoryDto[]</code>	Categorías de gastos disponibles
<code>productosLocal</code>	<code>Producto[]</code>	Productos para registros de tickets


## Configuración del formulario

La configuración del formulario se construye dinámicamente en función del tipo de gasto:


## Validación de formularios y lógica de negocios

El componente implementa varias reglas de negocio:


- Lógica de suscripción** : El `start` campo está sincronizado con `fechaCompra` y `accumulate` se calcula en función de la duración y los intervalos.

 `ServidorWebFrontend/gesthorAngular/src/app/pages/protected/form-herramienta/form-herramienta.component.ts`

231-258
- Gestión de productos** : para los tickets, los productos se gestionan a través de un sistema dedicado. `FormProductosComponent`

 `ServidorWebFrontend/gesthorAngular/src/app/pages/protected/form-herramienta/form-herramienta.component.html`

117
- Validación dinámica** : los campos obligatorios y las reglas de validación cambian según el tipo de gasto

 `ServidorWebFrontend/gesthorAngular/src/app/pages/protected/form-herramienta/form-herramienta.component.ts`

116-124

## Estructura de la plantilla de formulario

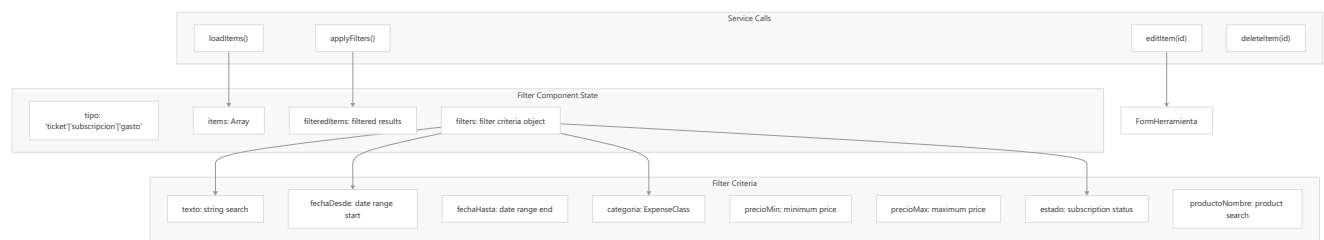
La plantilla de formulario utiliza formularios reactivos de Angular con representación de campos condicionales:



## Interfaz de filtrado y listado

Proporciona `FiltroHerramientaComponent` capacidades integrales de filtrado y listado de registros de gastos.

## Arquitectura de componentes



## Implementación de filtros

El sistema de filtrado aplica múltiples criterios simultáneamente:

Tipo de filtro	Se aplica a	Implementación
Búsqueda de texto	name campo	Coincidencia de subcadenas sin distinción entre mayúsculas y minúsculas
Rango de fechas	fechaCompra	Entre fechaDesde y fechaHasta
Gama de precios	total campo	Entre precioMin y precioMax
Categoría	typeExpense	Coincidencia exacta con ExpenseClass
Búsqueda de productos	productsJSON	Busca entradas dentro de los nombres de productos
Estado de suscripción	activa campo	Filtro activo/cancelado para suscripciones

# Interfaz de navegación de herramientas

Proporciona `ToolsComponent` acceso organizado a la funcionalidad de gestión de gastos a través de tres categorías de herramientas.

## Categorías de herramientas



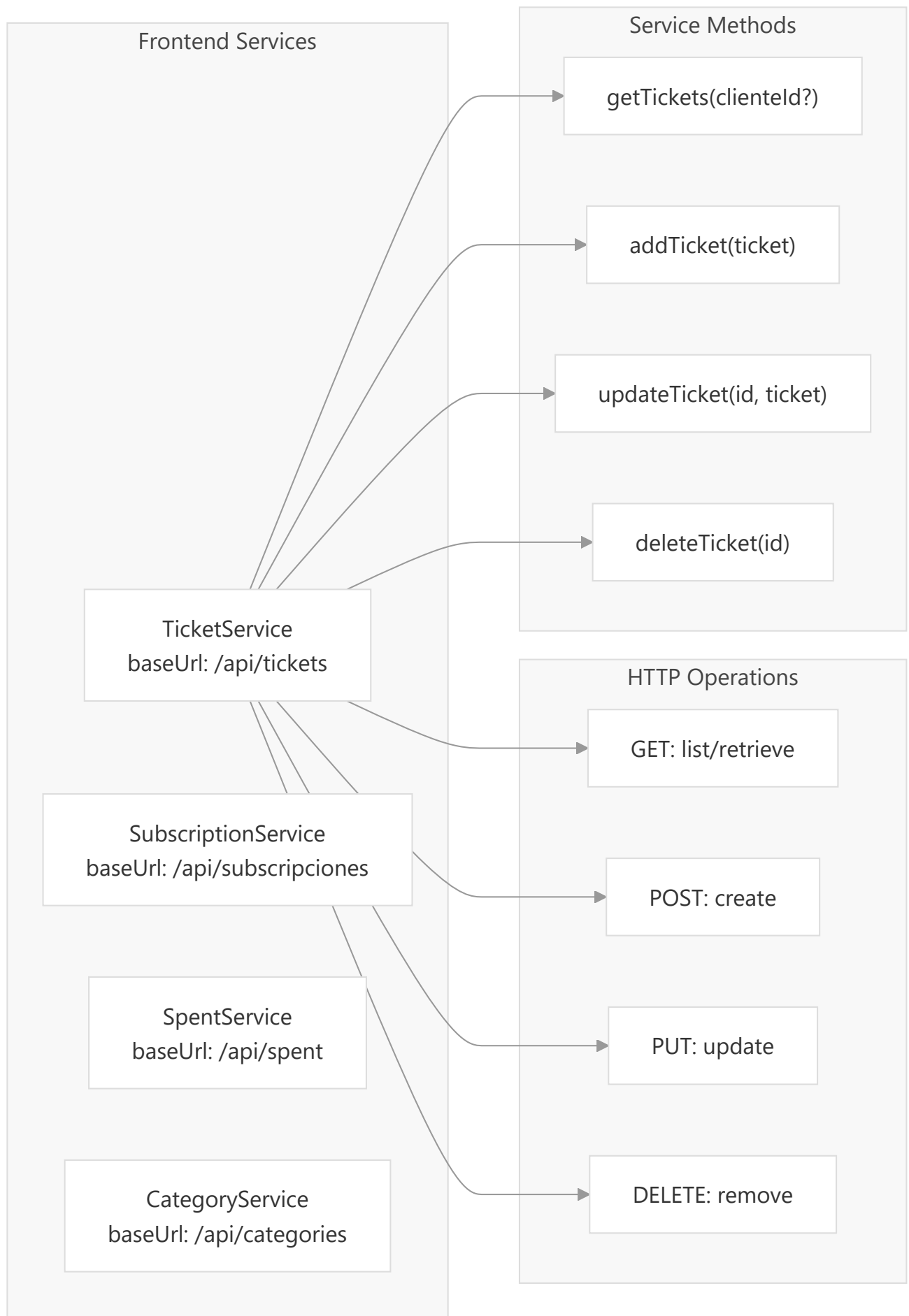
Cada categoría de herramientas tiene un propósito específico:

- **Herramientas de carga** : procese recibos e imágenes digitales mediante OCR
- **Herramientas de creación** : Entrada manual de nuevos registros financieros
- **Herramientas de administración** : explorar, editar y eliminar registros existentes

## Integración de servicios

La interfaz de gestión de gastos se integra con los servicios backend a través de servicios Angular dedicados.

## Arquitectura de la capa de servicio



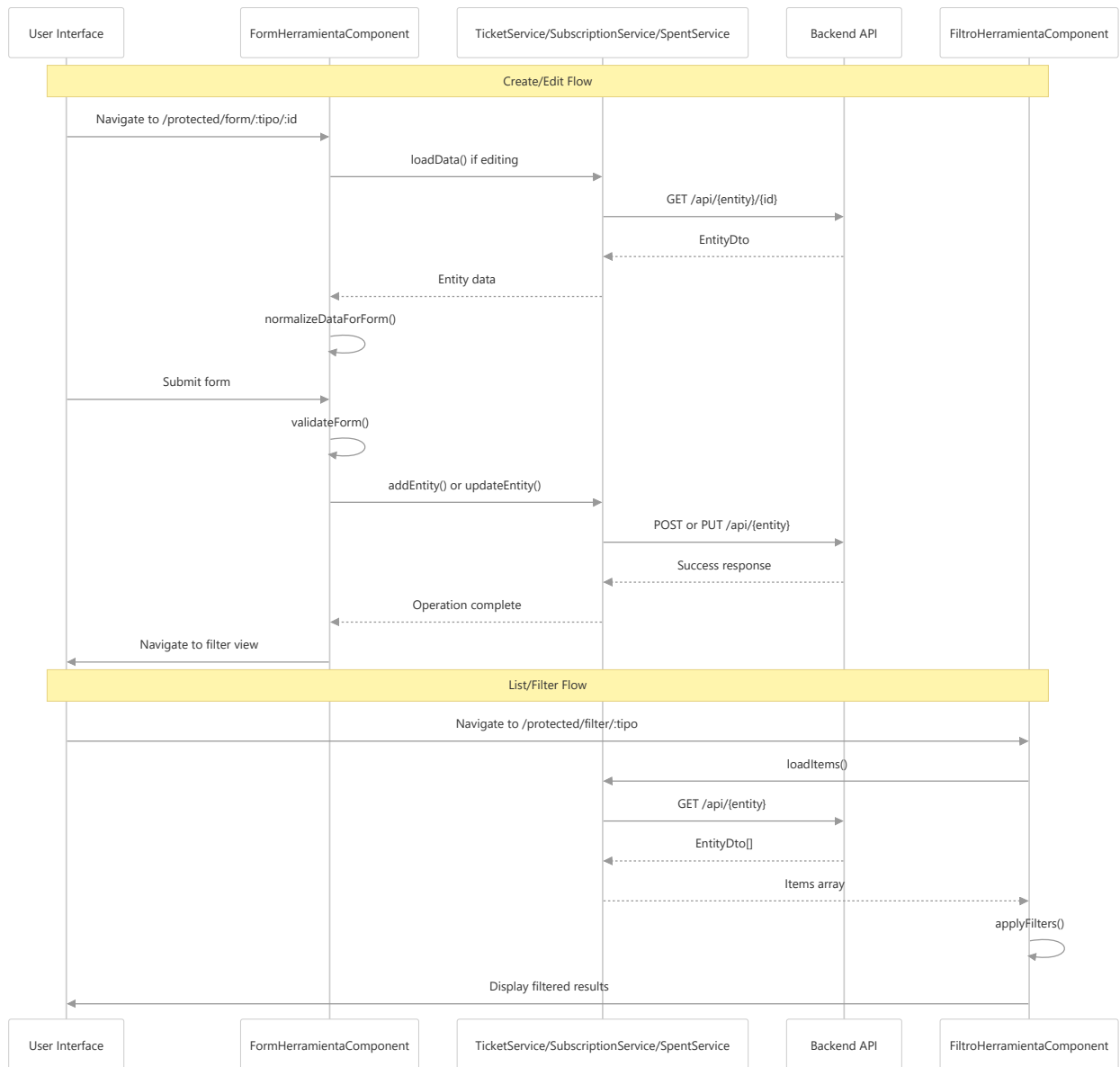
## Patrones de métodos de servicio

Todos los servicios siguen patrones consistentes para las operaciones CRUD:

Patrón de método	Objetivo	Parámetros	Tipo de retorno
<code>get{Entity}s(clienteId?)</code>	Listar registros	Filtro de cliente opcional	<code>Observable&lt;EntityDto[]&gt;</code>
<code>get{Entity}ById(id)</code>	Recuperar un solo registro	ID de entidad	<code>Observable&lt;EntityDto&gt;</code>
<code>add{Entity}(entity)</code>	Crear nuevo registro	Datos de la entidad	<code>Observable&lt;EntityDto&gt;</code>
<code>update{Entity}(id, entity)</code>	Actualizar registro existente	Datos de identificación y entidad	<code>Observable&lt;EntityDto&gt;</code>
<code>delete{Entity}(id)</code>	Eliminar registro	ID de entidad	<code>Observable&lt;void&gt;</code>

## Flujo de datos

El flujo de datos completo para las operaciones de gestión de gastos sigue este patrón:



La interfaz de gestión de gastos proporciona un sistema integral para la gestión de registros financieros, integrando validación de formularios, representación dinámica de campos, filtrado avanzado y patrones de servicio consistentes para brindar una experiencia CRUD completa para los usuarios que administran sus gastos, tickets y suscripciones.