Análisis del Sistema de Información

Versión 1.4

Enero de 2026

EraiKIT – Compositor de Formularios

Este documento es propiedad de Eusko Jaurlaritzen Informatika Elkartea – Sociedad Informática del Gobierno Vasco, S.A. (EJIE) y su contenido es CONFIDENCIAL. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

La plantilla utilizada para generar este documento está licenciada bajo licencia [CC](http://creativecommons.org/licenses/by-nc-sa/3.0/)

[ARINbide](http://www.ejie.eus/y79-04/es/contenidos/informacion/metodologia_desarrollo_ejie/es_0212/metodologia_desarrollo.html) by [EJIE](http://www.ejie.eus/) is licensed under a [Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported License](http://creativecommons.org/licenses/by-nc-sa/3.0/).

| Versión | Fecha | Resumen de cambios | Elaborado por: | Aprobado por: |
| --- | --- | --- | --- | --- |
| 1.0 | Feb-2024 | Versión Inicial | Javier López-Tello Ruiz | EJIE |
| 1.1 | Enero-2025 | Versión 2:   * Revisión apartado DEMO * Componente Imagen * Texto enriquecido * Acciones condicionales * Funciones * Fragmentos * Listados desde Toolkit * Servicio de Consulta de formularios para CCP * Capítulos derivados de la reunificación de documentos:Catálogo de Formularios, Catálogo de SVT’s. | Javier López-Tello Ruiz |  |
| 1.3 | Octubre-2025 | * Reestructuración de documento * Actualización de pantallas del compositor * Ajuste en la definición de los componente, acciones condicionales y funciones de cálculo | Antonio Gallego |  |
| 1.4 | Noviembre-2025 | * Unificación de revisiones EJIE-hiberus | Antonio Gallego |  |
| 1.5 | Diciembre-2025 | * Incluida nueva especificación y uso de fragmentos comunes y específicos | Antonio Gallego |  |
| 1.6 | Enero-2026 | * Catálogo de SVTs * Gestión de versiones de formularios, configuración del CCP y traspaso de formularios | Antonio Gallego |  |

Contenido

[1 Introducción 6](#_Toc219753504)

[1.1 Objetivo 6](#_Toc219753505)

[1.2 Alcance 7](#_Toc219753506)

[2 Glosario 9](#_Toc219753507)

[2.1 Glosario de términos 9](#_Toc219753508)

[3 Compositor de formularios 10](#_Toc219753509)

[3.1 Descripción 10](#_Toc219753510)

[3.1.1 Pantalla principal 11](#_Toc219753511)

[3.1.1.1 Acceso directo a Formularios 12](#_Toc219753512)

[3.1.1.2 SVT 13](#_Toc219753513)

[3.1.1.3 Listado de formularios 14](#_Toc219753514)

[3.1.1.4 Fragmentos 15](#_Toc219753515)

[3.1.2 Creación y edición de un formulario 15](#_Toc219753516)

[3.1.2.1 Aspectos generales 17](#_Toc219753517)

[3.1.2.1 Versiones de formularios 18](#_Toc219753518)

[3.1.2.2 Compositor 19](#_Toc219753519)

[3.1.2.3 Traducción 23](#_Toc219753520)

[3.1.2.4 Revisión y publicación 25](#_Toc219753521)

[3.1.2.5 Excepciones en la edición de formularios 29](#_Toc219753522)

[3.2 Composición del Formulario 30](#_Toc219753523)

[3.2.1 Formulario 32](#_Toc219753524)

[3.2.2 Componentes 33](#_Toc219753525)

[3.2.2.1 Componentes jerárquicos 34](#_Toc219753526)

[3.2.2.1.1 Componente apartado 34](#_Toc219753527)

[3.2.2.1.2 Componente sección 35](#_Toc219753528)

[3.2.2.1.3 Componente bloque 36](#_Toc219753529)

[3.2.2.1.4 Componente módulo: mínimo nivel de agrupación de componentes. 37](#_Toc219753530)

[3.2.2.1.5 Componente GRID: permitirá añadir dos componentes en la misma línea. 38](#_Toc219753531)

[3.2.2.2 Componentes Informativos 40](#_Toc219753532)

[3.2.2.2.1 Componente Imagen 40](#_Toc219753533)

[3.2.2.2.2 Texto enriquecido 40](#_Toc219753534)

[3.2.2.2.3 Componente Alertas 41](#_Toc219753535)

[3.2.2.3 Componentes de captura de datos 42](#_Toc219753536)

[3.2.2.3.1 Componente Input 42](#_Toc219753537)

[3.2.2.3.2 Componente Text Area 45](#_Toc219753538)

[3.2.2.3.3 Componente Date Picker 46](#_Toc219753539)

[3.2.2.3.4 Componente Select 48](#_Toc219753540)

[3.2.2.3.5 Componente Checkbox 52](#_Toc219753541)

[3.2.2.3.6 Componente Radio Button 53](#_Toc219753542)

[3.2.2.3.7 Componente Dirección Postal 54](#_Toc219753543)

[3.2.2.3.8 Componente Tabla 58](#_Toc219753544)

[3.2.2.3.9 Componente Grupo Repetición 61](#_Toc219753545)

[3.2.3 Formatos específicos 63](#_Toc219753546)

[3.2.4 Fragmentos 63](#_Toc219753547)

[3.2.4.1 Nuevo Fragmento 67](#_Toc219753548)

[3.2.4.2 Editar un fragmento 71](#_Toc219753549)

[3.2.4.3 Vista previa de un fragmento 72](#_Toc219753550)

[3.2.4.4 Gestión de versiones 73](#_Toc219753551)

[3.2.4.5 Uso de fragmentos 73](#_Toc219753552)

[3.2.5 Validaciones de Componente 75](#_Toc219753553)

[3.2.6 Acciones Condicionales 77](#_Toc219753554)

[3.2.7 Funciones de Cálculo 84](#_Toc219753555)

[3.2.8 Carga de valores en componentes 87](#_Toc219753556)

[3.2.8.1 Listado manual 87](#_Toc219753557)

[3.2.8.2 Listado desde SVT 88](#_Toc219753558)

[3.2.8.3 Listados desde Toolkit 89](#_Toc219753559)

[4 Catálogo de formularios 91](#_Toc219753560)

[4.1 Alcance 91](#_Toc219753561)

[4.2 Descripción 91](#_Toc219753562)

[4.3 Traspaso de formularios 94](#_Toc219753563)

[4.4 Servicio de consulta de formularios para CCP 96](#_Toc219753564)

[5 Catálogo de SVT’s 97](#_Toc219753565)

[5.1 Alcance 97](#_Toc219753566)

[5.2 Descripción 97](#_Toc219753567)

[5.3 Funciones principales del catálogo de SVTs 97](#_Toc219753568)

[5.4 Configuración de traducción del esquema de retorno al esquema utilizado por la aplicación 98](#_Toc219753569)

[5.5 Listado de SVTs 99](#_Toc219753570)

[5.6 Consulta de un SVT 100](#_Toc219753571)

[5.7 Creación de un SVT 100](#_Toc219753572)

[5.8 Modificación de un SVT 101](#_Toc219753573)

[5.9 Eliminación de un SVT 102](#_Toc219753574)

[5.10 Prueba/Test de un SVT 102](#_Toc219753575)

[5.10.1 Creación de un nuevo SVT 104](#_Toc219753576)

[5.10.2 Validación de un SVT 106](#_Toc219753577)

[5.10.3 Modificación de un SVT 107](#_Toc219753578)

[5.10.4 Consulta de SVT’s 107](#_Toc219753579)

[5.10.5 Baja de un SVT 107](#_Toc219753580)

[5.10.6 Ayuda a la utilización de un SVT en la composición 108](#_Toc219753581)

[5.10.7 Definición de esquemas para las respuestas de SVTs 108](#_Toc219753582)

[6 Ayudas en la composición de formularios 109](#_Toc219753583)

[6.1 Consejos y Normas de Configuración 109](#_Toc219753584)

[7 Configuración del CCP y Traspaso de formularios 110](#_Toc219753585)

[7.1 Configuración en el CCP 110](#_Toc219753586)

[7.2 Proceso de traspaso de formularios entre entornos 110](#_Toc219753587)

# Introducción

## Objetivo

Definir la herramienta que facilitará la creación y edición de formularios, denominada **Compositor de formularios,** así como los aplicativos relacionados, llamados **Catálogo de Formularios y Catálogo de SVT’s (servicios de terceros)**.

Los aplicativos serán consumidos en la red privada (intranet) del Gobierno Vasco, identificados como back-office y cuya seguridad será administrada mediante XLNETS.

Interfaz de usuario gráfica, Diagrama

El contenido generado por IA puede ser incorrecto.

* **Compositor de formularios:** en esta herramienta se crearán, probarán y publicarán los formularios que conformarán el paso 2 de un procedimiento/acción telemática existente en el Catálogo de Procedimientos (CCP).

El Compositor sólo estará implantado en el entorno de Desarrollo Productivo.

Además del Compositor, se desarrollarán otros dos aplicativos adicionales que servirán como soporte a la gestión de formularios:

* [**Catálogo de formularios**](#_Catálogo_de_Formularios)**:** registra los formularios y ofrece servicios que serán consumidos tanto por el Compositor (guardado, consulta, publicación), como por el CCP (consulta de formularios).

El Catálogo se mantendrá desde el Compositor de formularios en el entorno de desarrollo productivo y tendrá una réplica en los entornos de PRE y PRO. Los formularios del catálogo se traspasarán a los entornos integrándose en los procesos de traspaso de los procedimientos/acciones telemáticas.

* [**Catálogo de SVT’s**](#_Catálogo_de_SVT’s)**:** contiene el registro de losservicios de terceros requeridos por los formularios para recuperar o validar datos.

El catálogo de SVT’s se mantendrá desde pantallas añadidas en el Compositor de Formularios dentro del entorno de desarrollo productivo y tendrá una réplica en los entornos de PRE y PRO. El registro de los SVT’s en el catálogo se traspasará a los entornos integrándose en los procesos de traspaso de los SVT’s de los departamentos.

Todas las interacciones entre las aplicaciones propias, así como con aplicaciones externas, deberán realizarse mediante llamadas a servicios o API’s.

Mapa general de relaciones:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## Alcance

En el ámbito de procesamiento de una solicitud, los formularios que podrán editarse y diseñarse en la herramienta serán los correspondientes al Paso 2: datos Negocio.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

El resto de los pasos se obtendrán mediante componentes diseñados para uso común de todas las solicitudes (Componentes transversales definidos en el documento del Orquestador de Solicitudes).

El flujo de creación de formularios cubrirá las siguientes funcionalidades principales:

* Creación, edición y composición de formularios.
* Prueba integral del formulario.
* Publicación de formularios en el Catálogo de formularios para ser consumidos por las aplicaciones.
* Gestión de cambios de formularios mediante versionado..
* Integración con terceros mediante SVT’s para realizar validaciones contra la información existente en las aplicaciones departamentales, así como para recuperar detalles que permitan autocompletar la información.

# Glosario

## Glosario de términos

|  |  |
| --- | --- |
| **Término** | **Definición** |
| Formulario | Formato de recogida de información para un determinado propósito. |
| Solicitud | Captura de información mediante un formulario en formato dato y archivo para su almacenamiento y envío para ser tramitado por las aplicaciones departamentales. Las solicitudes podrán ser de diferentes tipologías: solicitud, subsanación, recurso, alegación… |
| SVT | Servicios de Validación de Terceros. |
| NORA | Sistema de consulta de callejero. |
| CCP | Catálogo de Configuración de Procedimientos. |
| PPS | Proceso Presentación de Solicitud. |
| Grupo funcional | Grupo al que pertenece un funcionario. Está configurado en el CCP. |
| XLNets | Sistema de autenticación de Gobierno Vasco |

# Compositor de formularios

## Descripción

En esta herramienta se crearán, probarán y publicarán los formularios que conformarán el paso 2 de un procedimiento/acción telemática existente en el Catálogo de Procedimientos (CCP).

El flujo de gestión de formularios comprende las siguientes fases:

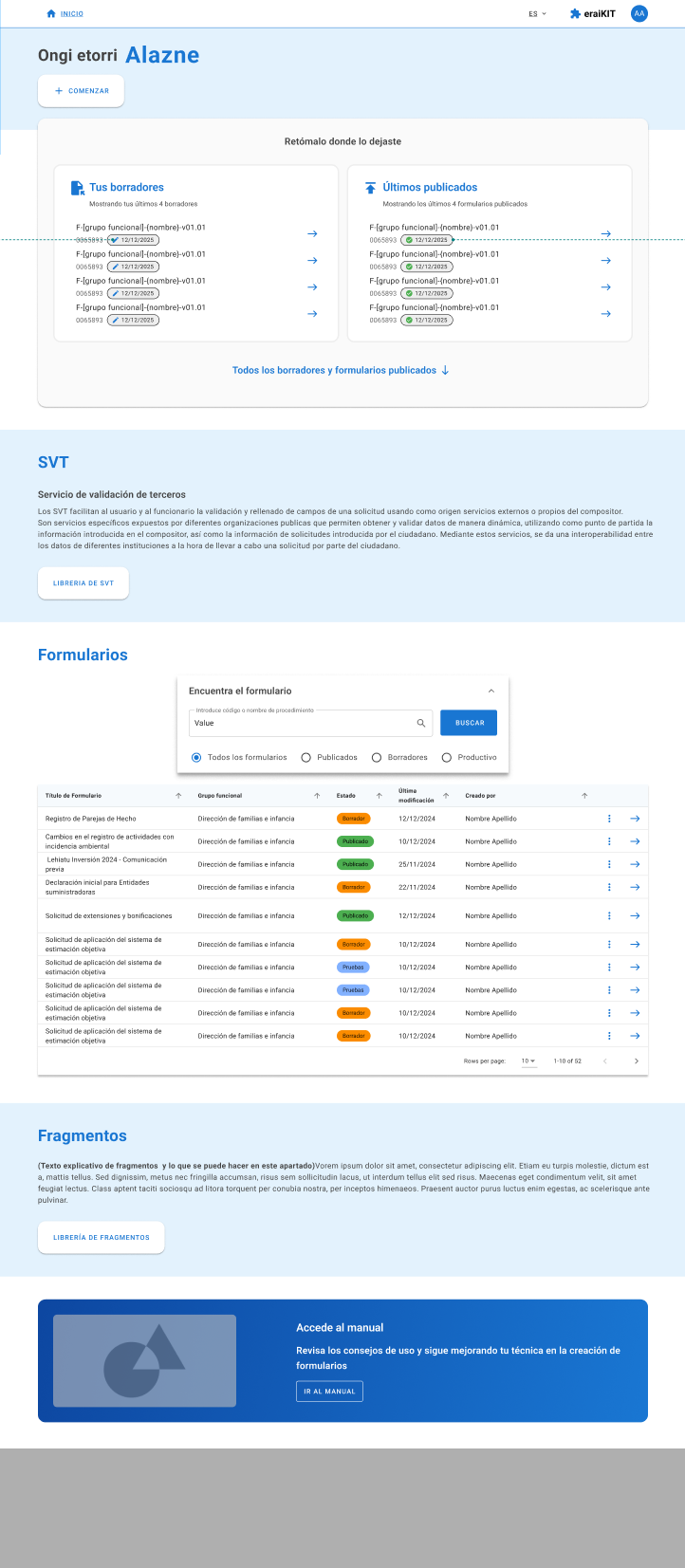


Estado: ------------------ EDICION--------------------- ------PUBLICADO------

Los estados por los que pasará el formulario son:

* **Creación**: se compone el formulario con el fin deseado.
* **Edición**: desde el momento de su creación durante la fase de diseño hasta la publicación (incluye la traducción y previsualización del formulario).
* **Publicación:** el formulario está publicado y puede asociarse a un procedimiento en CCP.

### Pantalla principal



La pantalla principal de EraiKIT está dividido en los siguientes bloques:

* **Cabecera:** muestra el acceso al inicio del compositor, el idioma para acceder al compositor, el nombre de la herramienta (eraiKIT) y el identificador de la persona conectada.
* **Acceso directo a Formularios**: Ofrecerá una ruta guiada para la creación de formularios nuevos, así como el acceso directo a los borradores recientes o a los formularios publicados.
* **SVT:** acceso al dashboard específico de gestión de SVTs desde donde podrá gestionar el catálogo de servicios dados de alta.
* **Listado completo de formularios:** mostrará la lista de todos los formularios existentes, con la información de estado y permitiendo realizar un filtrado a través de un componente de búsqueda.
* **Fragmentos**: con acceso al módulo de gestión y edición de los fragmentos, comunes y/o específicos, según el grupo funcional al que pertenezca el usuario conectado.

Los usuarios técnicos con permisos de acceso a la aplicación (autenticados en XLNETS) podrán acceder al EraiKIT. La URL de acceso formará parte del menú de herramientas del departamento técnico encargado del mantenimiento de formularios.

Una vez logeado en XLNets, se comprobará el grupo o grupos funcionales a los que pertenece el usuario, consultando el servicio expuesto desde el CCP, de manera que se pueda gestionar la seguridad y permitir el acceso a la información correspondiente en base a estos grupos funcionales.

Los usuarios del grupo funcional PLATEA-TRAMITACIÓN podrán ver y crear formularios de cualquier grupo funcional, dado que este grupo es el administrador general de todos ellos.

Los usuarios del resto de grupos funcionales tan sólo podrán acceder a ver y crear los formularios de los grupos funcionales a los que tengan acceso.

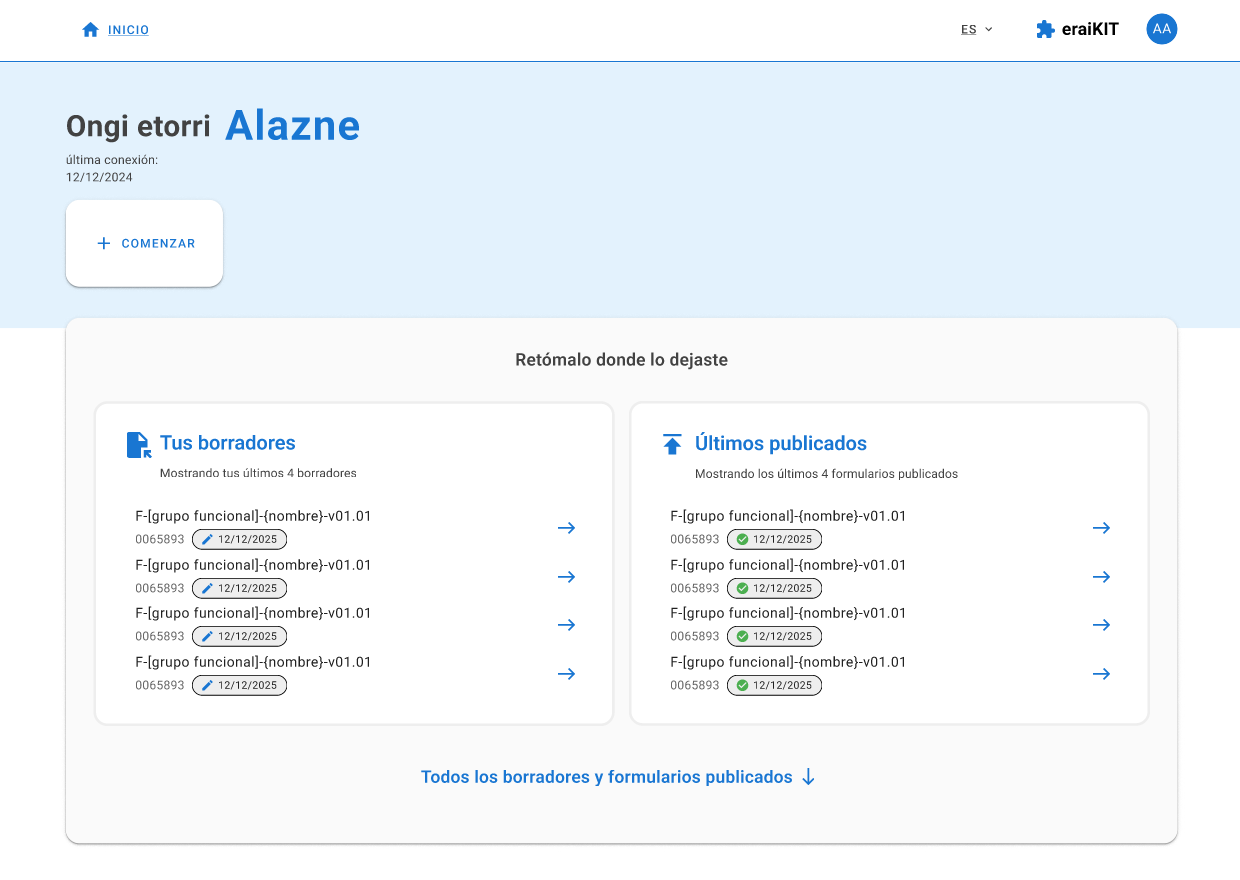
Adicionalmente, si un usuario no tuviera acceso a ningún grupo funcional, EraiKIT le informará de este hecho y no le permitirá realizar ninguna acción ni le mostrará formularios. En ese caso, deberá ponerse en contacto con otros miembros del grupo funcional al que perteneciera para que le incluyan en el grupo (configuración en el CCP) y así poder acceder a editar formularios.

Acceso directo a Formularios

Desde la pantalla principal de EraiKIT se tendrá acceso tanto a la creación de un nuevo formulario como a la actualización y consulta de formularios en estado borrador o publicados.

El formulario podrá crearse desde cero o bien mediante la copia de un formulario existente, ya sea un formulario en edición o un formulario publicado.

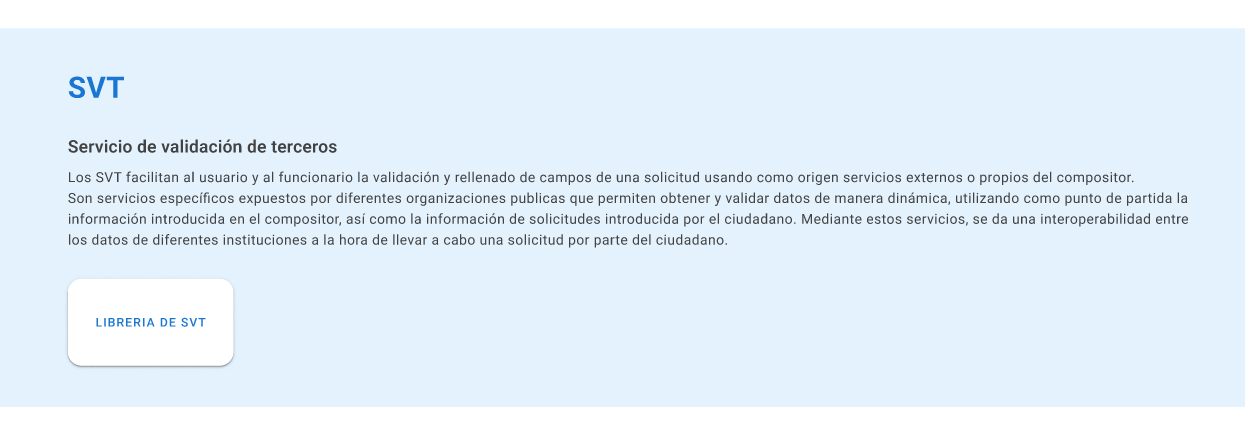
* **Cabecera general**: incluye el enlace a la página de inicio, el idioma escogido, eraKIT y el usuario que está operando en ese momento.
* **Cabecera del compositor**: identifica a la persona usuaria, detalla la fecha de la última conexión y facilita las opciones para comenzar, creando un formulario de cero, haciéndolo a partir de un borrador o utilizando un formulario publicado anteriormente (opción equivalente a copiar ese formulario).
* **Contenido central del primer bloque**: facilita, si las hay, opciones anteriores de borrador y últimos formularios publicados.

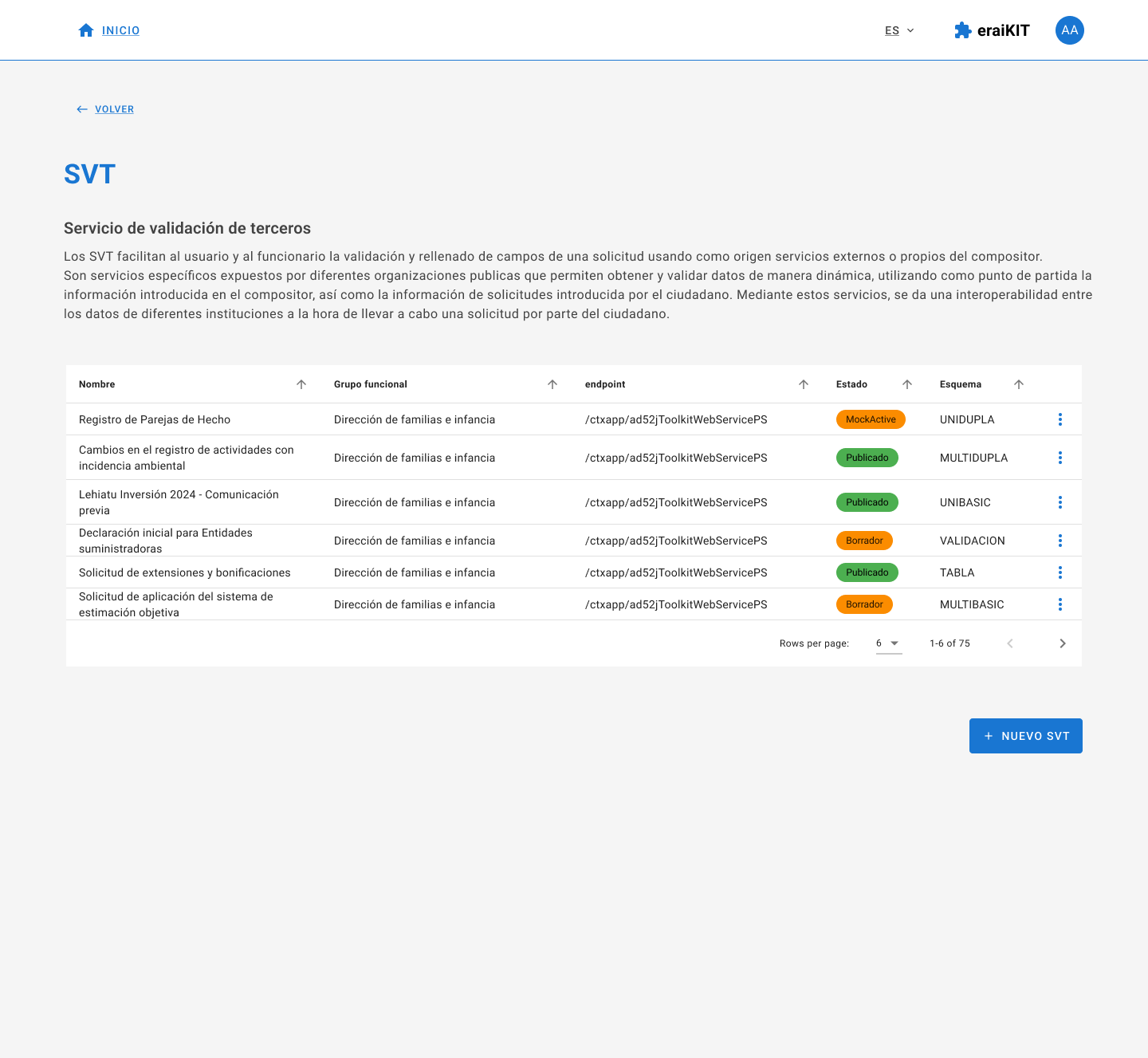


SVT

En la misma pantalla principal de EraiKIT, se visualizará el apartado de configuración de SVTs, desde el que se de acceso al Dashboard de SVTs, donde se puede consultar y catalogar nuevos servicios que serán consumidos posteriormente por los formularios.

Las personas usuarias sólo podrán realizar el mantenimiento de los SVT’s pertenecientes a los grupos funcionales para los que tengan autorización.





Listado de formularios

Se compondrá de la lista de todos los formularios en sus diferentes estados, mostrando aquellos pertenecientes a los grupos funcionales (Justicia, Educación, Hacienda, Cultura, etcétera) a los que tenga acceso la persona usuaria. Podrá aplicar filtros y búsquedas para facilitar la localización de los formularios.

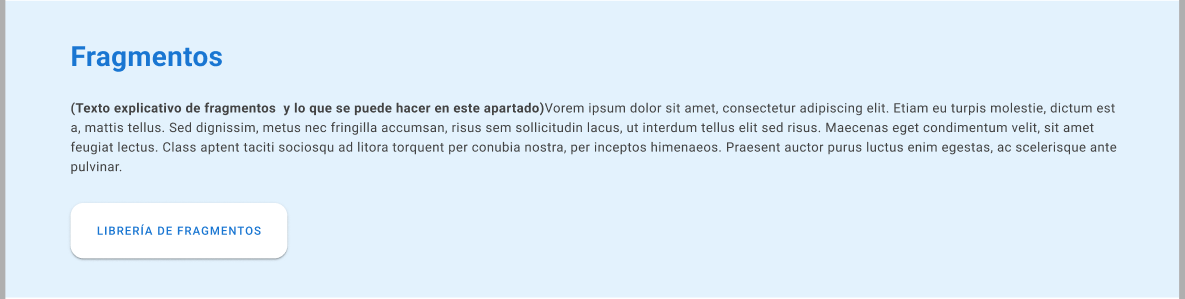


Sobre el listado de formularios se podrán realizar las siguientes acciones:

* Editar: acceder a cada uno de ellos para editarlo si el formulario está en estado borrador, o consultarlo, si el formulario se encuentra en estado publicado.
* Eliminar: eliminar un formulario, siempre que no esté relacionado con ningún procedimiento. Crear una nueva **versión** de un formulario añadiendo o eliminando campos en el formulario y, por tanto, variando la información compartida con la aplicación departamental.

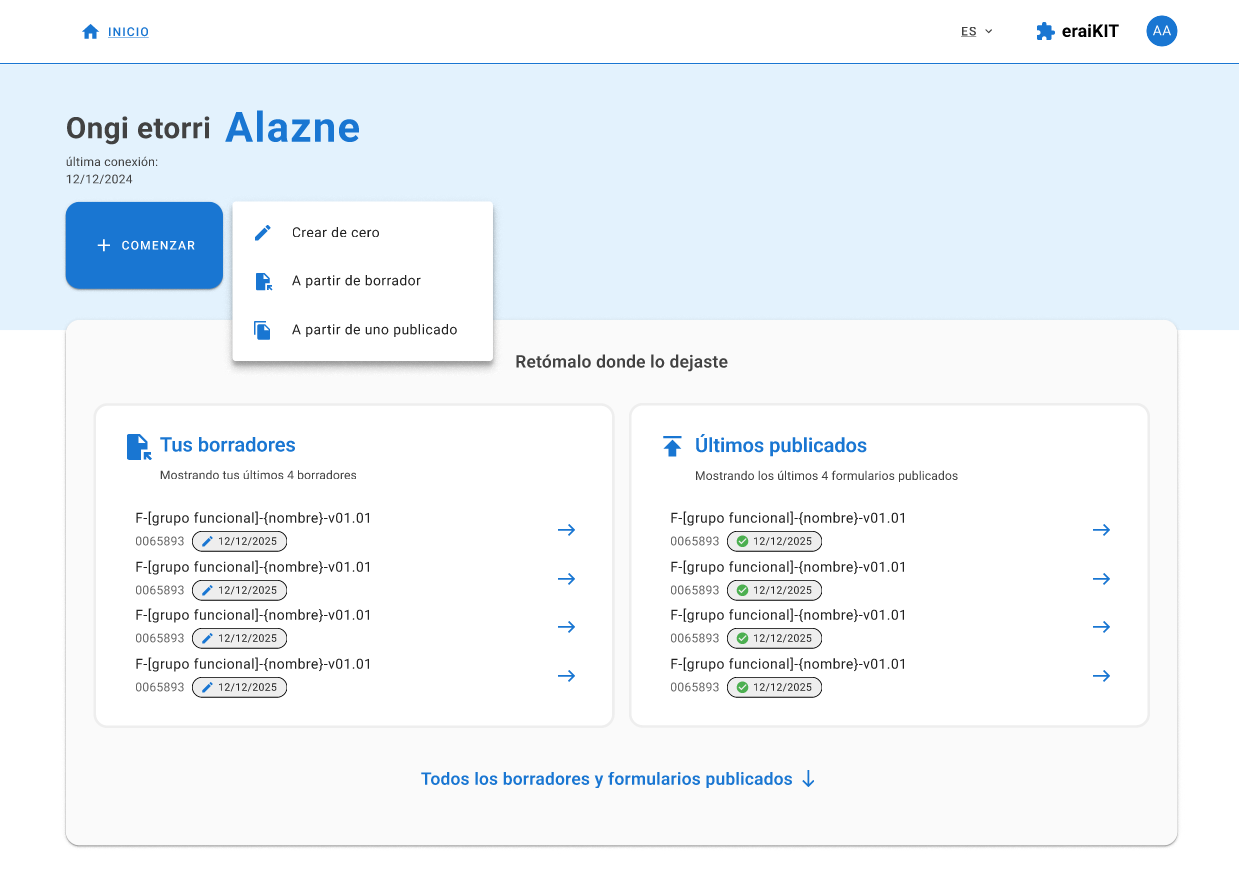
Fragmentos

A partir de aquí se da aceso a la gestión completa de la librería de fragmentos para el grupo o grupos funcionales a los que tiene acceso el usuario conectado.



### Creación y edición de un formulario

EraiKIT permitirá iniciar un nuevo formulario desde cero, continuar con un formulario en estado borrador o realizar una copia de un formulario ya publicado.



La creación de un formulario constará de los siguientes pasos:

* **Aspectos generales**: muestra la información identificativa del formulario.
* **Compositor**: permite la edición y composición del formulario mediante componentes.
* **Traducción**: permite la traducción a diferentes idiomas de los textos del formulario.
* **Revisión y publicación:** muestra la visión en modo ejecución del formulario en proceso de edición, y permite crear una DEMO para visualizar el formulario creado. Una vez el formulario se pase a estado “publicado”, no se podrá seguir editando y será seleccionable para su asociación con la acción telemática de un procedimiento.

3.1.2.1 Aspectos generales

.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Para crear un formulario, será necesaria la información requerida por el Catálogo de formularios:

* Idioma de composición: castellano o euskera.

No podrá ser modificado. Por defecto mostrará el idioma en el que se encuentre la herramienta, visible en la esquina superior derecha de la página.

* Idiomas del formulario: como mínimo obligatorio, deberá estar en castellano y euskera.

Se indicarán los valores en función de los idiomas en los que se pueda presentar el formulario. Esto tendrá reflejo en el paso de traducción.

* Nombre del formulario: se recomienda que tenga una longitud máxima de 50 caracteres.
* Versión. Se completará de forma automática comenzando por la versión 01 (v01).
* Descripción del formulario.
* Grupo funcional: Salud, Justicia, Educación, etcétera.

A seleccionar en función de los grupos funcionales a los que pertenezca la persona usuaria.

Una vez se guarde la información requerida, se generará el ID de formulario con la siguiente nomenclatura:

F-**OOOO-NNNNNNNNN-VNN**

OOOO - Grupo Funcional

**NNNNNNNNN** – Nombre corto descriptivo

VNN – indicando la versión del formulario

Versiones de formularios

Si la creación comienza a partir de un formulario ya publicado, se creará una nueva versión del mismo. El número de versión será gestionado por el propio EraiKIT especificando un nuevo número a partir del último existente. Es decir, si tenemos la versión “v01”, creará la “v02”, si tenemos la versión “v04”, creará la “v05”, etcétera. El número de versión estará formado por la letra “v” y un correlativo de dos cifras comenzando siempre por el “01”. La creación de una versión nueva no permitirá modificar ningún elemento de la pantalla de “Aspectos Generales”, sin embargo, en la pantalla “Compositor” permitirá realizar cualquier cambio desde cambios sencillos sobre textos u orden de los componentes hasta modificar el contenido de los componentes, configuración, validaciones, eliminación, ... Es responsabilidad del usuario mantener la coherencia del formulario dado que su modificación implica, en muchas ocasiones, adecuar la manera en la que los aplicativos departamentales van a recibir y procesar los datos introducidos en él.

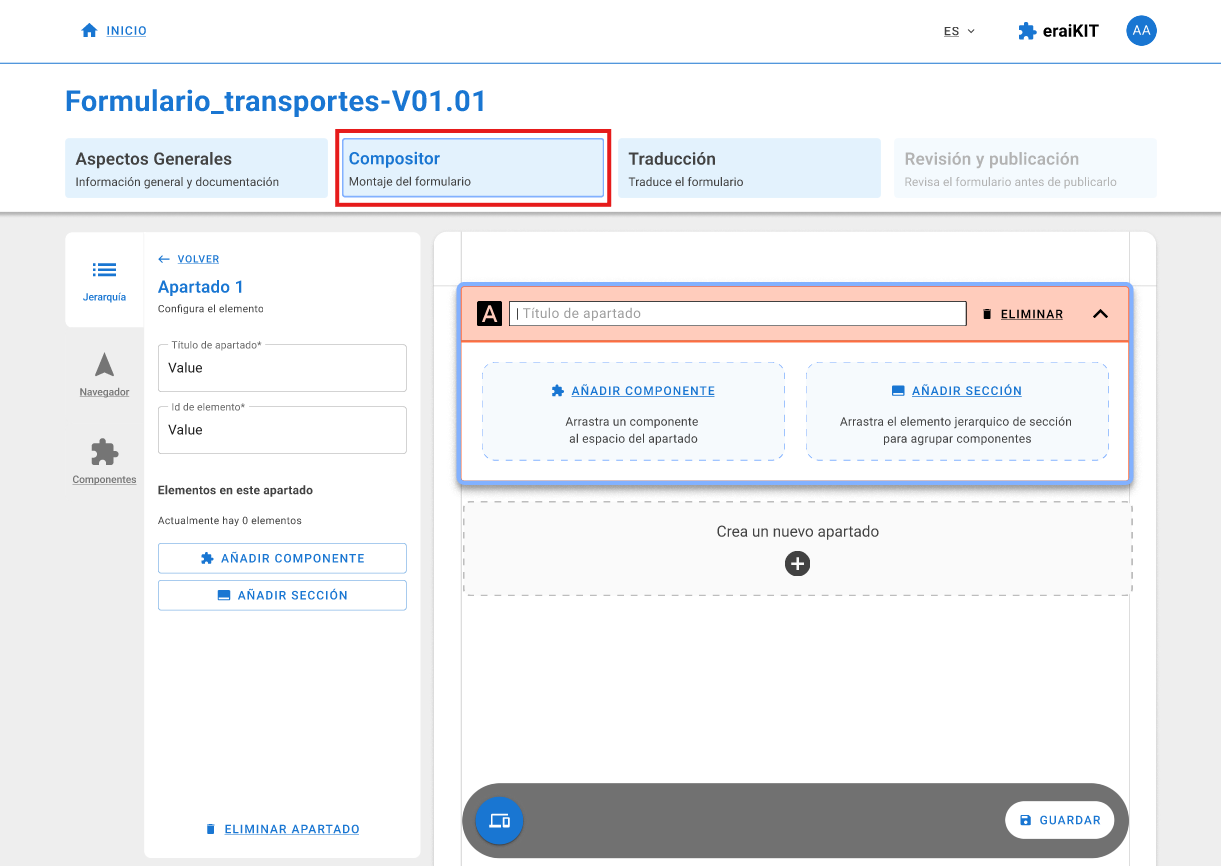
3.1.2.2 Compositor

En el apartado ‘Compositor’, se podrán realizar las siguientes acciones:

* Diseñar y componer la estructura y la presentación del contenido a mostrar, según los niveles organizativos disponibles.
* Configurar cada uno de los componentes tanto la información básica como la información particular en función del tipo de componente, así como el uso de los SVTs y/o las acciones condicionales que afecten a su comportamiento.

El Compositor mostrará el área de edición y los diferentes componentes que pueden establecerse y configurarse.

Inicialmente se comenzará por los componentes de estructura y posteriormente se podrán incluir componentes y fragmentos (se encuentra especificado en los capítulos 3.2.2 Fragmentos y 3.2.3 Componentes).



* Sección ‘Jerarquía’: permite añadir al formulario los distintos apartados con sus elementos, organizar el contenido en su respectivo lugar y estructurarlo de manera jerárquica por apartado, sección, bloque o módulo.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

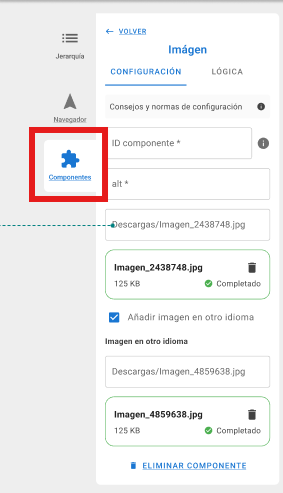
* Sección ‘Navegador’: esta sección muestra la estructura del formulario y permite reordenar los elementos en función de las de anidación y ordenación de los componentes de estructura:
* Un apartado podrá cambiarse de orden en el nivel raíz de apartados.
* Una sección sólo se podrá mover dentro del mismo apartado o a otro apartado.
* Un bloque sólo se podrá mover dentro de la misma sección o a otra sección.
* Un módulo sólo se podrá mover dentro del mismo bloque o a otro bloque.

Asimismo, esta sección ofrece la posibilidad de exportar el mapa del formulario creando un CSV con la estructura completa de componentes incluyendo los componentes organizativos.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

* Sección “Componentes”: en este apartado se muestran los componentes disponibles para usar en los formularios junto con su configuración, lógica y comportamiento. Asimismo, desde esta sección puede eliminarse el uso de cada componente.

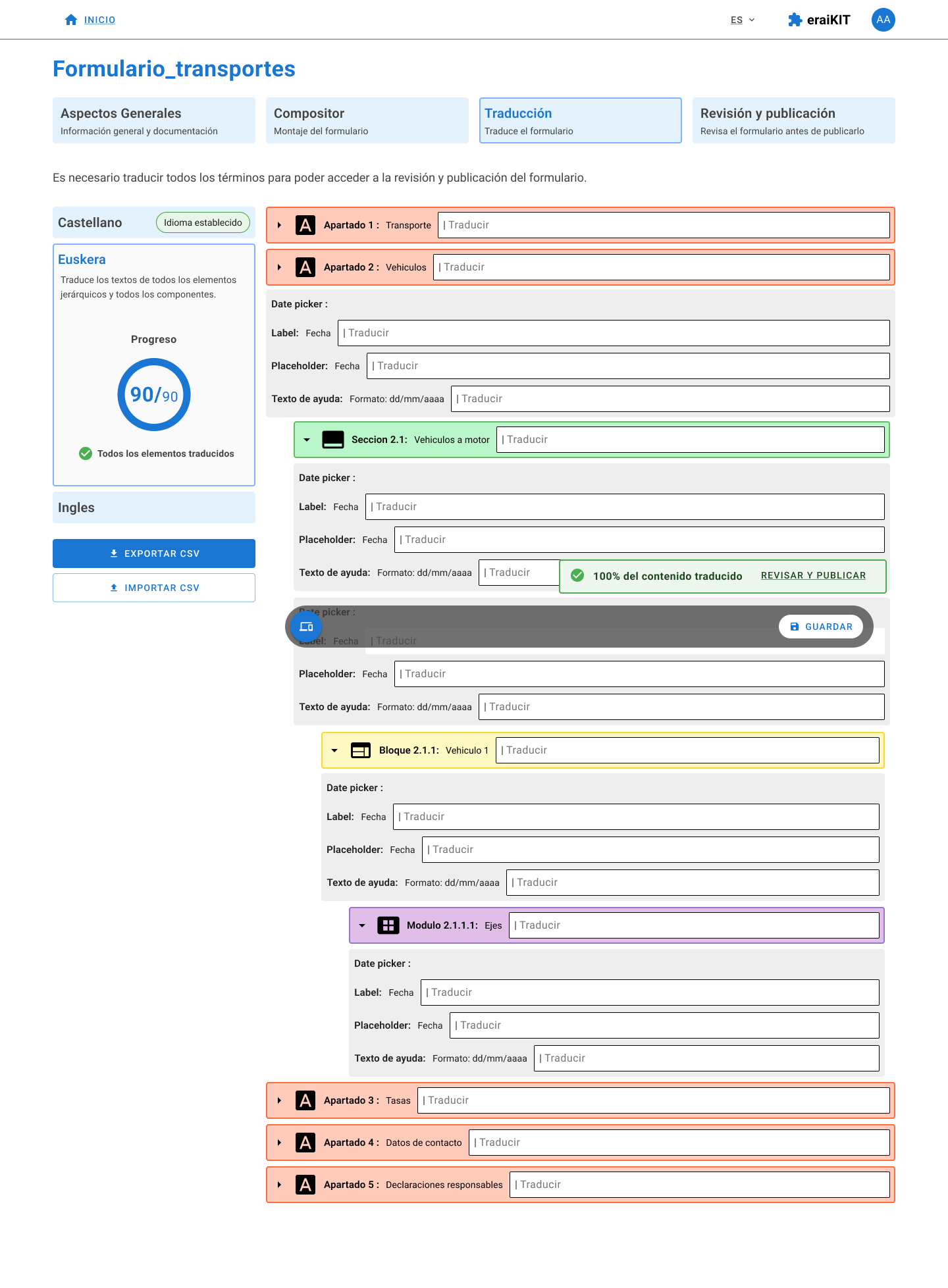


Los componentes de jerarquía sólo se podrán reordenar respetando el orden de anidación. Es decir:

* Un apartado podrá cambiarse de orden en el nivel raíz de apartados.
* Una sección sólo se podrá mover dentro del mismo apartado o a otro apartado.
* Un bloque sólo se podrá mover dentro de la misma sección o a otra sección.
* Un módulo sólo se podrá mover dentro del mismo bloque o a otro bloque.

3.1.2.3 Traducción

En este apartado el usuario tendrá la opción de traducir los literales a los diferentes idiomas en los que se presentará el formulario una vez publicado a partir del idioma de composición:



Existen dos maneras de traducir los campos y contenidos del formulario a otro idioma:

1. Exportar un CSV, facilitando su traducción por un servicio externo y la posibilidad de importarlo una vez traducido.
2. Traducir manualmente los campos y contenido de los mismos, para lo cual bastará con pulsar y editar directamente el texto.

Dada la obligatoriedad de que el contenido esté, al menos, en castellano y euskera, no se activará en botón “continuar“ hasta que la traducción esté completa.

Una vez se pulsa el botón “continuar” se realizan las siguientes validaciones:

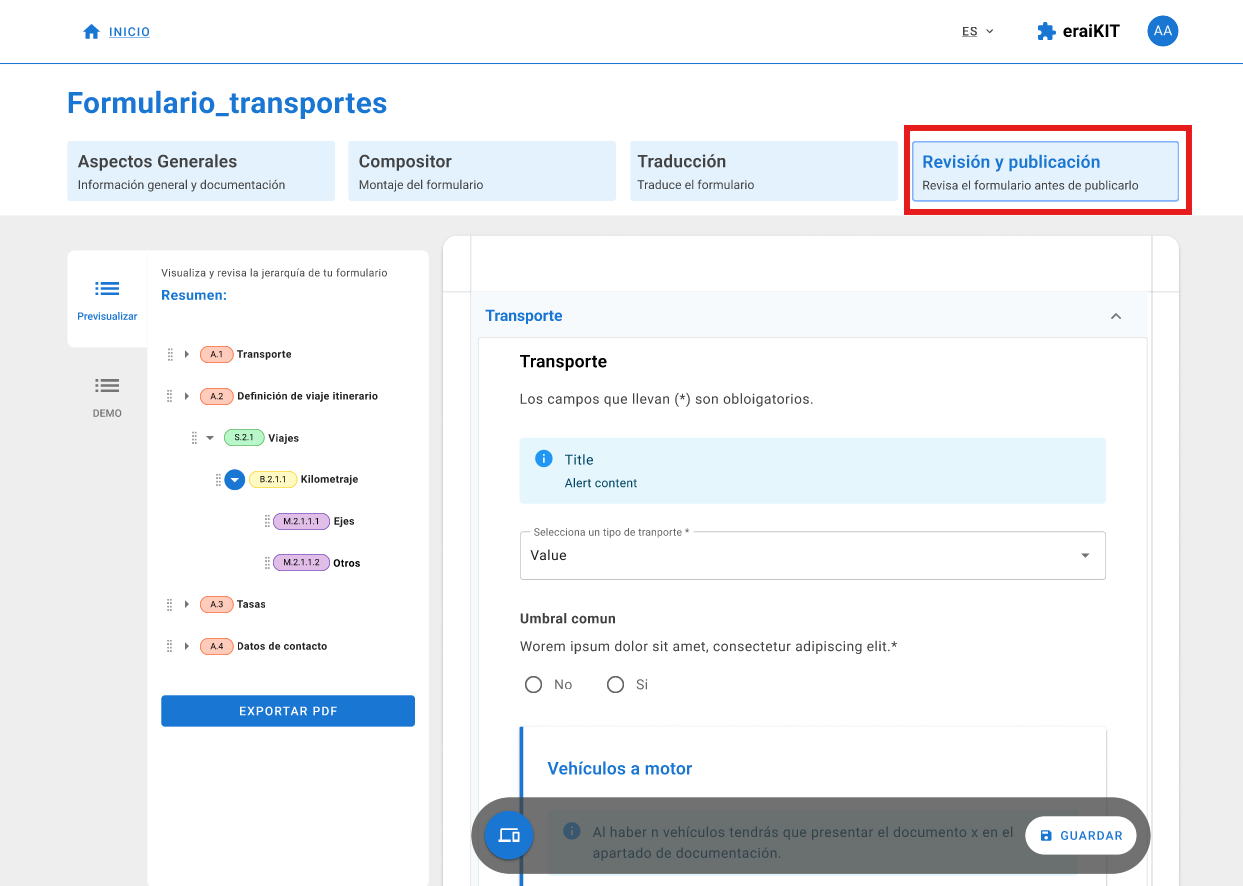
* Comprobaciones básicas de estructura:
  + Validar que no hay componentes organizativos vacíos.
  + Validar que no hay *name* repetidos entre todos los componentes.
  + Validar que no se usen "*name* reservados".
  + Validar que los *name* utilizados en las validaciones y condiciones existen en el formulario.
  + Validar que los *name* utilizados en las llamadas a los SVT existen en el formulario.
* Las traducciones de textos a otros idiomas deben estar completadas.
* Los SVTs usados por el formulario no deben estar en estado mock.

Una vez superadas las validaciones, se activará la pestaña de “revisión y publicación”.

#### 3.1.2.4 Revisión y publicación

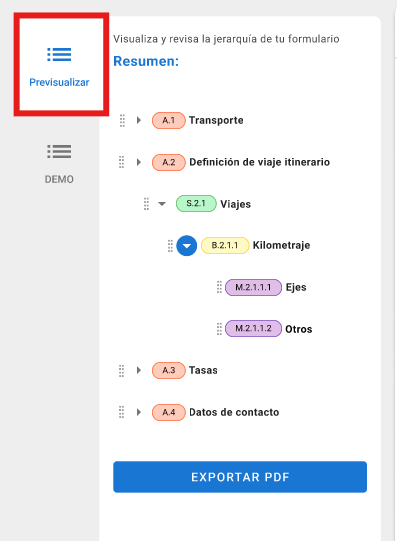
En esta pestaña podrán realizarse 3 actividades:

Previsualizar, demo y publicar.



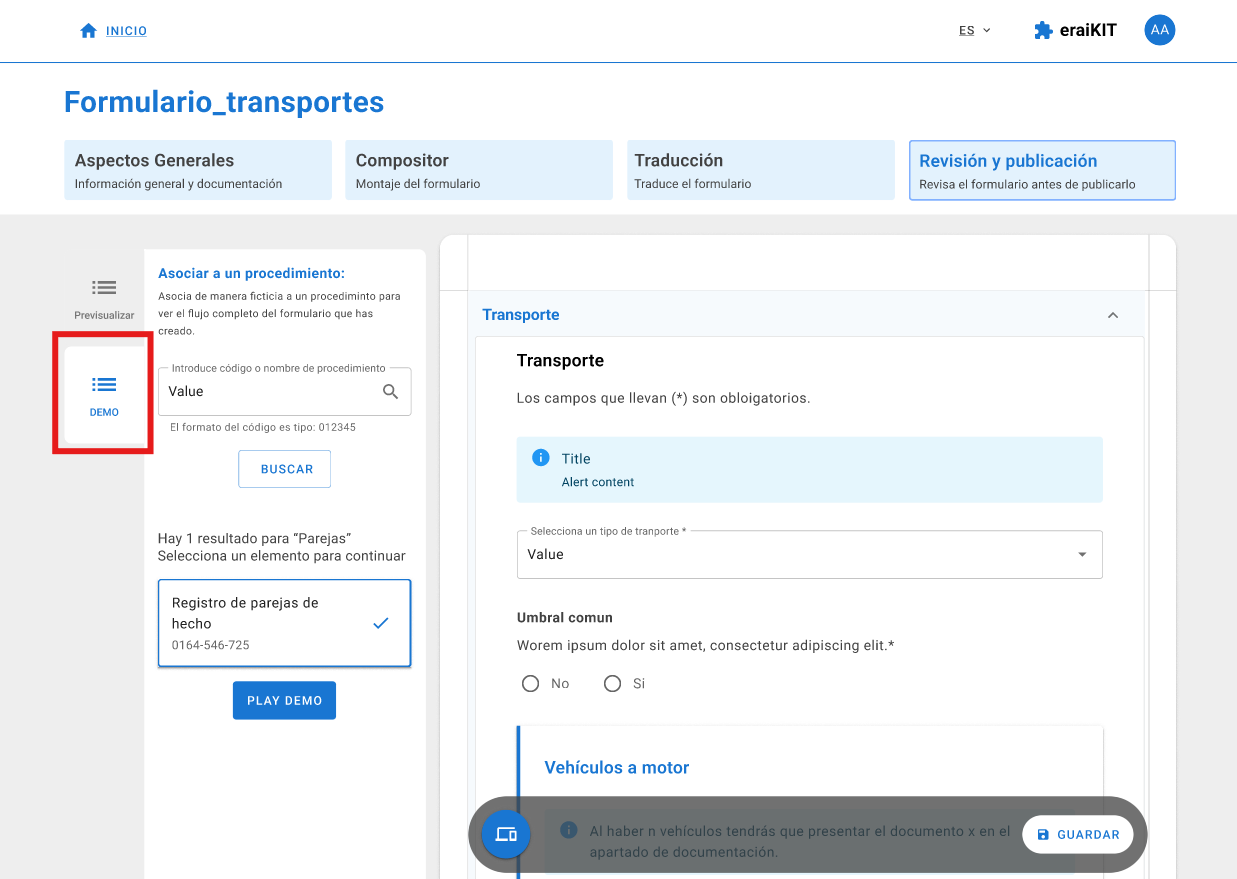
**Previsualizar**

La pestaña de previsualizar permite al usuario ver el resultado final del formulario antes de su publicación. Además, ofrece la opción de descargar un PDF que muestra la estructura del formulario en forma de árbol, proporcionando un resumen completo de los contenidos.



**Demo**

La opción "demo” permitirá la ejecución simulada de una solicitud utilizando el formulario que se está editando antes de publicarlo. . Para poder ejecutar dicha funcionalidad, será necesario especificar un procedimiento y una acción telemática de manera que se puedan recuperar todos los datos configurados en el CCP y contextualizar cómo se va a comportar TollkitBerria en la cumplimentación del formulario y petición de documentos, autorizaciones y declaraciones responsables.



Desde la opción “demo” disponible en el menú izquierdo se podrá seleccionar el procedimiento y la acción telemática. La selección del procedimiento estará condicionada a los grupos funcionales del usuario conectado, es decir, sólo se podrán seleccionar procedimentos pertenecientes a los grupos funcionales del usuario, En el caso de que el usuario pertenezca al grupo funcional “Platea Tramitación” se podrá seleccionar cualquier procedimiento existente en el CCP.

La ejecución de la “demo” abrirá el toolkit Berria ubicando al usuario en el paso de datos específicos y permitiéndole navegar por los paso de autorizaciones, documentos y declaraciones responsables.

Salvo en el paso de datos específicos donde la ejecución del formulario será completa pudiendo ejecutarse condiciones, validaciones, llamadas a los SVTs configurados y demás, los pasos de autorizaciones, documentos y declaraciones responsables verán mermada su funcionalidad no permitiendo la interacción del usuario, pero si mostrando los elementos necesarios en base al a configuración y ejecución de la condiciones que pudieran tener a nivel CCP.

Obviamente, la solicitud no se almacenará en la base de datos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

**Publicación**

Para que un formulario pase a estado publicado se volverá a validar:

* Comprobaciones básicas de estructura:
  + Validar que no hay componentes organizativos vacíos.
  + Validar que no hay *name* repetidos entre todos los componentes.
  + Validar que no se usen "*name* reservados".
  + Validar que los *name* utilizados en las validaciones y condiciones existen en el formulario.
  + Validar que los *name* utilizados en las llamadas a los SVT existen en el formulario.
* Las traducciones de textos a otros idiomas deben estar completadas.
* Los SVTs usados por el formulario no están en estado mock.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

#### 3.1.2.5 Excepciones en la edición de formularios

**Formulario en estado “Publicado”**

Cuando el formulario se encuentra en estado “publicado”, la aplicación mostrará un formulario completo en modo consulta, no pudiendo actualizar ni componentes ni configuraciones ni traducciones.

La pestaña previsualizar y publicar estará activa y se permitirá previsualizar con el botón publicar desactivado.

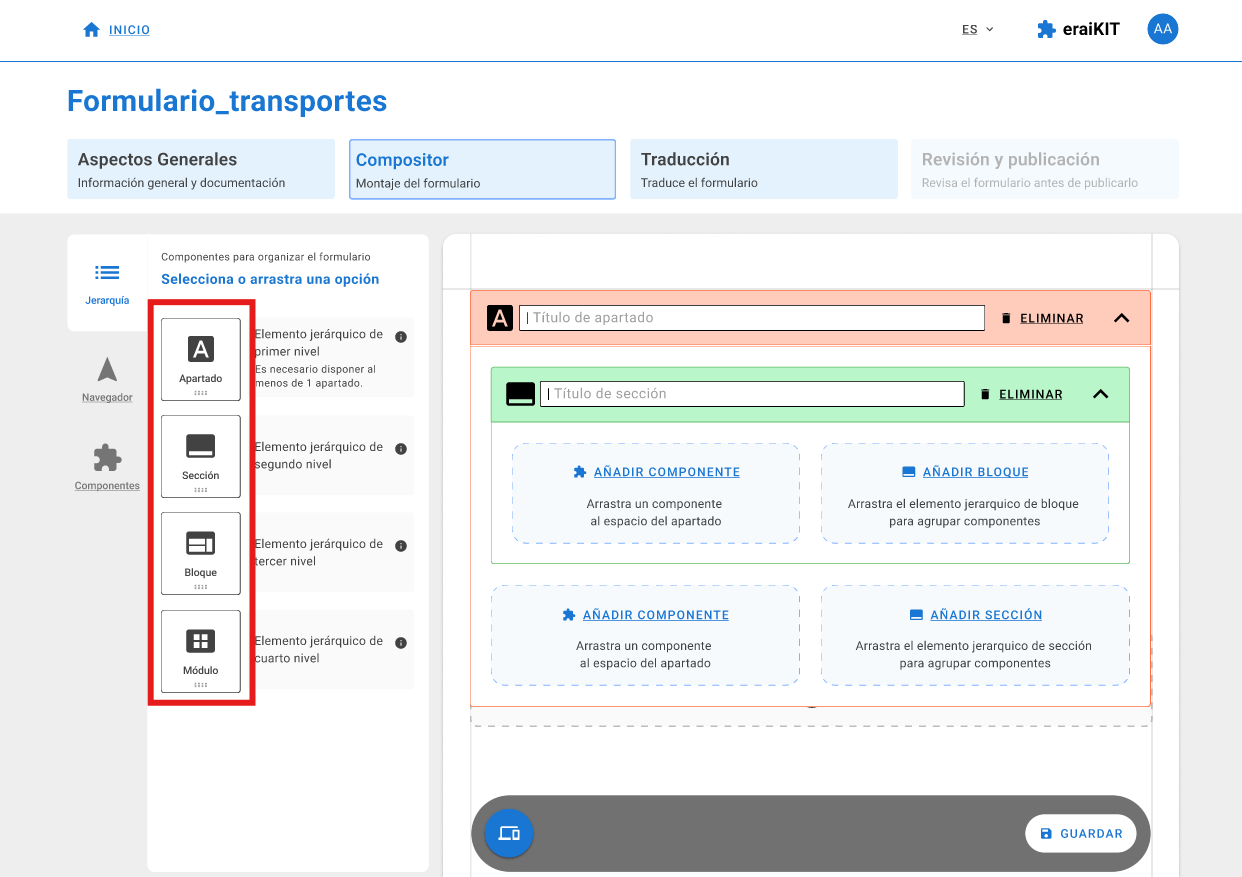
## Composición del Formulario

Un formulario es un formato de recogida de información para un determinado propósito.

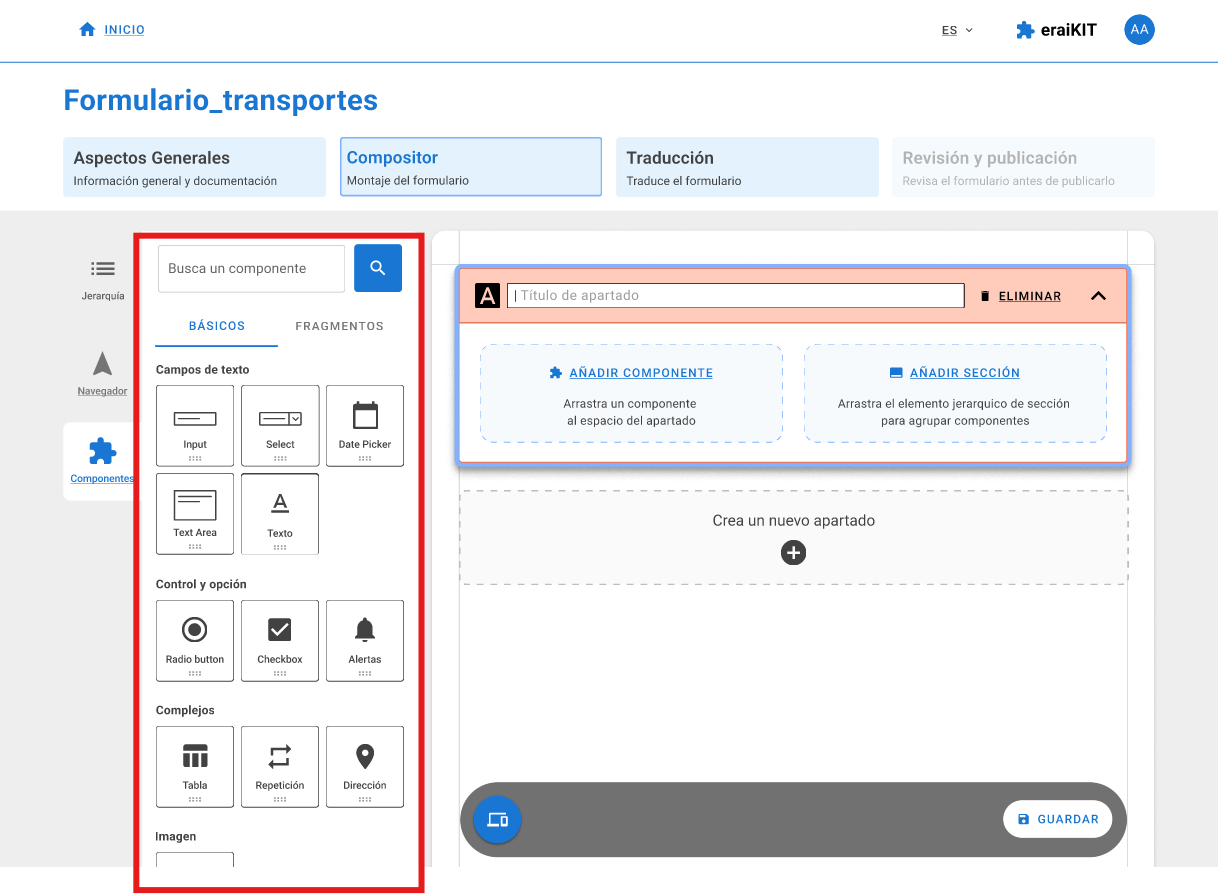
El formulario se diseñará mediante el uso de componentes de 3 tipos: organizativos, informativos y de captura de datos.

En el momento de la composición, a la izquierda se mostrarán los componentes organizativos o jerárquicos que permitirán la anidación a 4 niveles:

**Apartado – Sección – Bloque – Módulo**



Además, mostrará los componentes que podrá utilizar el usuario para diseñar y conformar el formulario, así como el menú de Navegador para facilitar la comprensión de la estructura anidada y realizar cambios organizativos.



### Formulario

La gestión de la identificación del formulario, sus datos y sus cambios de estado, se realizarán mediante el acceso a los servicios del Catálogo de formularios.

*Modelo de datos de formularios*

El formulario llevará asociados todos los elementos que lo componen:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

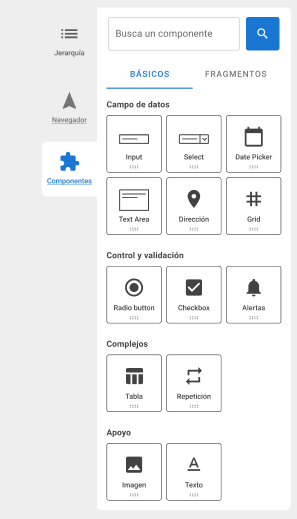
El Compositor contará adicionalmente con un modelo donde almacenará los catálogos-diccionario de componentes y fragmentos, así como el acceso a los servicios de consulta de los servicios consumidos expuestos en el Catálogo de SVT’s.

### Componentes

La herramienta mostrará el catálogo de componentes que permitirá al usuario el diseño del formulario.

Los componentes son las diferentes estructuras que facilitan o simplifican de manera visual el guiado, la comprensión y la captura de datos, cuyo contenido ingresado o seleccionado se recogerá como dato de la solicitud.

Hemos definido 4 tipos de componentes: Campos de datos, Control y validación, Complejos y de Apoyo.



**Campos de dato:** Permiten al usuario ingresar o seleccionar información dentro del formulario.

* **Input**: campo de texto simple para ingresar datos. Tendrá, además, la posibilidad de configurarse como oculto (hidden) o como fijo (fixed).
* **Select**: lista desplegable para elegir una opción. En este caso se podrá configurar como de selección simple o múltiple.
* **Date Picker**: selector de fechas.
* **Text Area**: área de texto para entradas más extensas.
* **Dirección**: campo específico para capturar datos de ubicación. Este componente será ampliamente configurable como se describirá en su propio apartado.
* **Grid**: permite organizar elementos en la misma línea, es decir, permite la posibilidad de incluir dos componentes dentro de la misma línea en lugar de uno.

**Control y validación:** Componentes que facilitan la selección, validación y control de la información ingresada.

* **Radio Butto**n: opción única dentro de un conjunto de alternativas.
* **Checkbox**: selección múltiple de opciones.
* **Alertas**: mensajes de aviso o error para guiar al usuario.

**Complejos:** Componentes que combinan varios elementos o permiten estructuras más avanzadas.

* **Tabla**: presenta información organizada en filas y columna, con un tamaño máximo de 6 columnas por tabla.
* **Grupo de repetición:** cuando una tabla tiene que tener más de seis columnas, lo que quiere decir que existe la posibilidad de incluir más de 6 datos para cada registro, el grupo de repetición permite la inclusión de estos datos en base a componentes específicos.
* .

**Apoyo**: Componentes que enriquecen visualmente el formulario o proporcionan información adicional.

* **Imágenes:** para añadir elementos visuales.
* **Texto**: para incluir información, instrucciones o explicaciones adicionales.

**Los componentes se diseñarán con los estilos marcados en la biblioteca MUI de componentes React que cumplen con los principios de Material Design:**

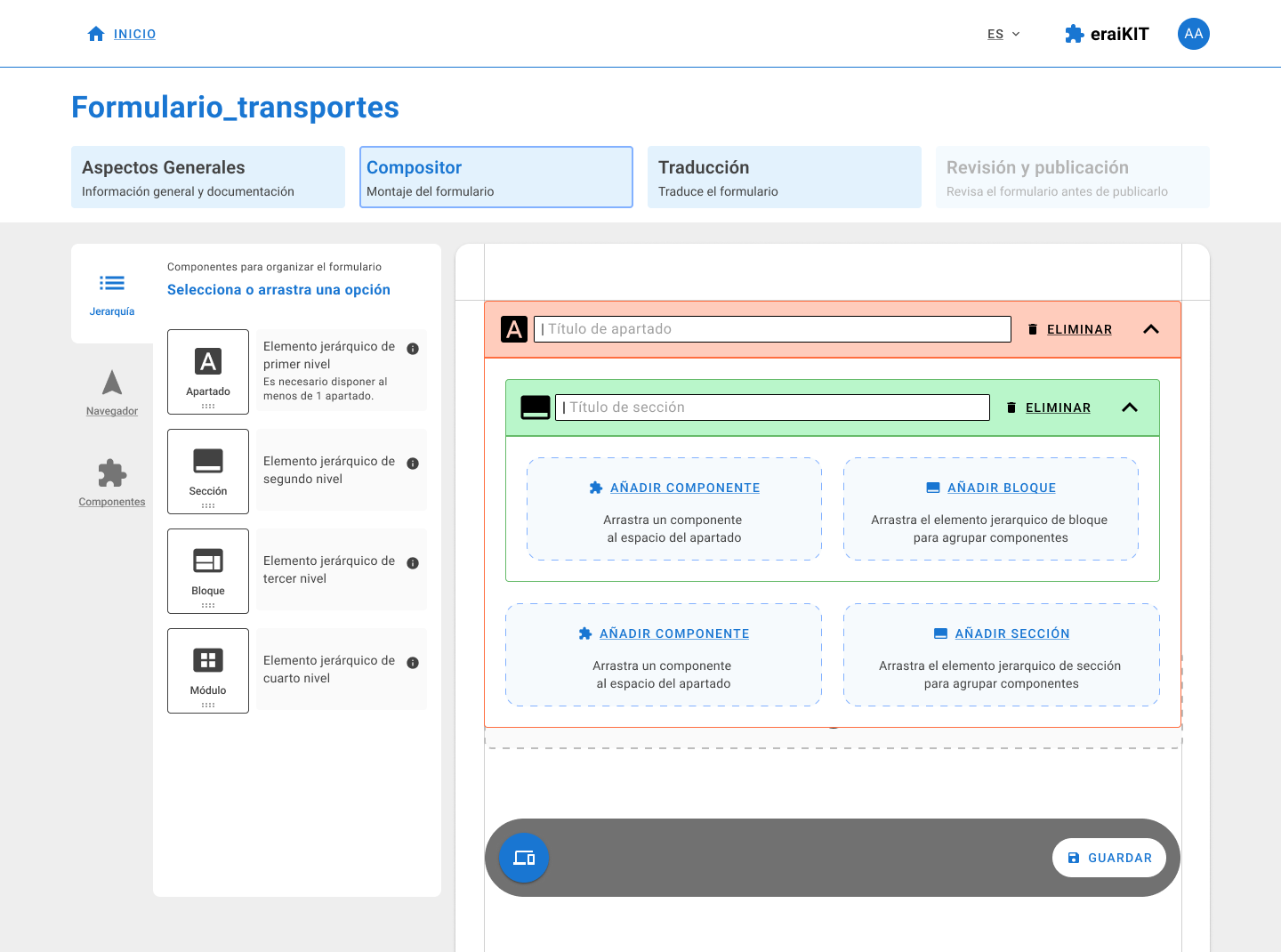
<https://mui.com/material-ui/all-components/>

Todos los componentes tendrán un nombre que los identifique de manera única dentro del formulario. Este nombre tiene que seguir las siguientes especificaciones:

* Puede contener letras, mayúsculas o minúsculas
* Puede contener números (del 0 al 9)
* Puede contener los siguientes símbolos: \_ - : .
* Siempre tiene que comenzar por una letra (mayúscula o mnúscula)

Componentes jerárquicos

Los componentes jerárquicos permitirán la ordenación del formulario en 4 niveles de anidación:



##### Componente apartado

Índice principal del formulario que mostrará los principales grupos de información a capturar. En todos los formularios deberá existir al menos un apartado y no pueden existir componentes fuera de un apartado.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

##### Componente sección

Agrupaciones visuales funcionales de la información dentro de un apartado. Las secciones irán encabezadas por un título de segundo nivel que las especifique.

**Configuración**:

* Podrá ser colapsable, mostrándose abierto por defecto.

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

##### Componente bloque

Grupos de datos necesarios a nivel visual (Ej: datos de vehículo ligero, datos de vehículo pesado). Permite la captura de varios items con el mismo formato.

**Configuración**:

* El bloque podrá configurarse como bloque de repetición, por lo que, en la fase de cumplimentación de solicitud, una vez completado un bloque, podrá añadir un nuevo bloque (“añadir otro”) o duplicar el último marcado para facilitar el marcaje (“duplicar”).
* Si el componente bloque se define como de repetición, los componentes internos no podrán se referenciados para validaciones o acciones condicionadas.
* Podrá ser colapsable, mostrándose abierto por defecto.

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

##### Componente módulo: mínimo nivel de agrupación de componentes.

**Configuración**:

* Podrá ser colapsable, mostrándose abierto por defecto.
* El módulo podrá configurarse como módulo de repetición, por lo que, en la fase de cumplimentación de solicitud, una vez completado un módulo, podrá añadir un nuevo Módulo (“añadir otro”) o duplicar el último marcado para facilitar el marcaje (“duplicar”).
* Si el componente módulo se define como de repetición, los componentes internos no podrán se referenciados para validaciones o acciones condicionadas.

Los componentes organizativos podrán asociarse a acciones condicionales para su exposición en base a datos introducidos (Ej: *radiobutton* de selección).

Icono

Descripción generada automáticamente

##### Componente GRID: permitirá añadir dos componentes en la misma línea.

Por defecto, los contenidos del formulario se mostrarán secuencialmente uno debajo del otro ocupando cada componente una línea. El componente GRID permitirá insertar dos componentes en la misma línea bajo 3 posibles casuísticas:

Funcionalidad Grid:

* 50/50--->Está opción preseleccionada
* 30/70
* 70/30

Opciones de componente:

* Input
* Select
* Date picker
* Multiselect (Cuando se coloque en un lado, el otro lado tiene que ir vacío --> Si selecciona componente1=multiselect, componente2 queda deshabilitado. En el segundo componente no se puede dar opción de multiselect
* El grid de la izquierda tiene que estar completo/relleno, pero el de la derecha es opcional.

Interfaz de usuario gráfica, Aplicación

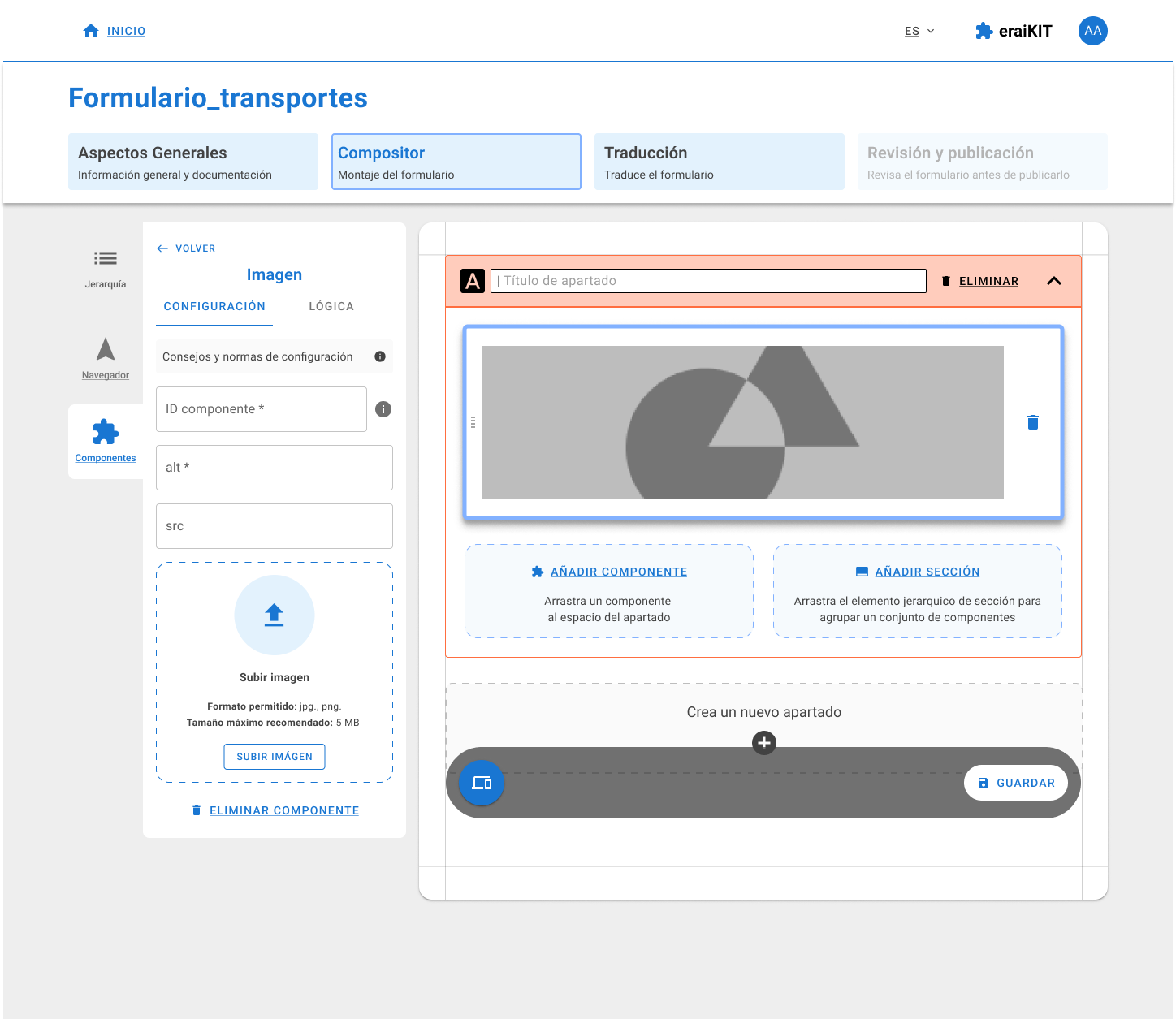
El contenido generado por IA puede ser incorrecto.

**Configuración**:

* Selección del formato de GRID 50/50, 30/70, 70/30
* Permitirá incluir dos componentes en cada GRID controlando los tamaños.
* El componente GRID sólo admitirá añadir componentes Input, Select y Date.

Componentes Informativos

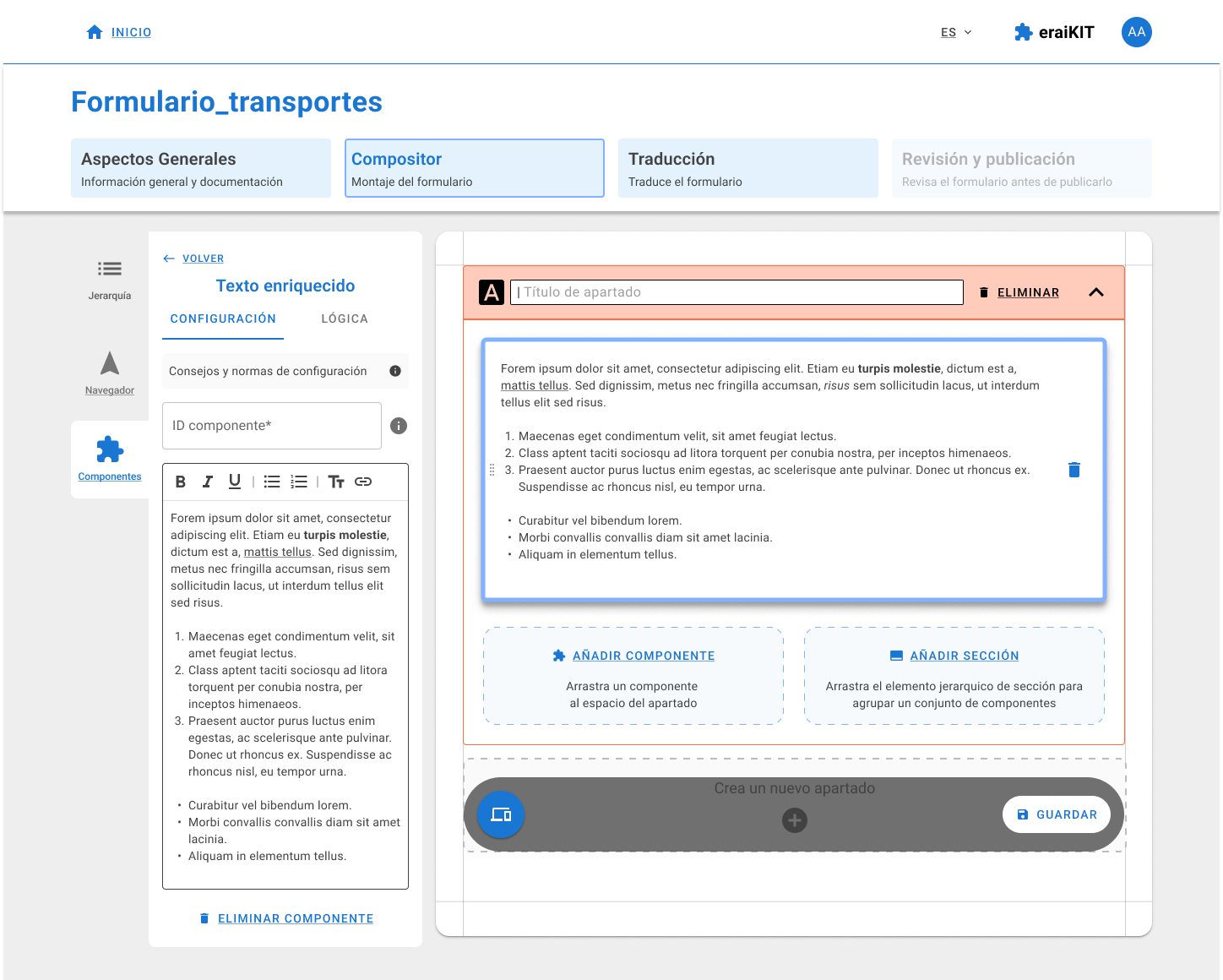
##### Componente Imagen



* Permite incorporar imágenes, logos o iconos que puedan incorporarse al cuerpo del formulario.
* Las imágenes se podrán añadir desde el PC (*drag&drop* o explorador de Windows, a conveniencia de la persona usuaria).
* Las imágenes podrán ser diferentes por idioma, por lo que se tendrán que definir y asociar al componente de manera independiente para cada idioma.
* Las imágenes se guardarán en el formulario como un contenido más.
* Será necesario establecer, si fuera necesario, una restricción de tamaño de la imagen para poder incluirla en el formulario sin que afecte a su visualización, Por ejemplo: establecer una talla para su visualización con unos tamaños prestablecidos: S, M, L, XL.

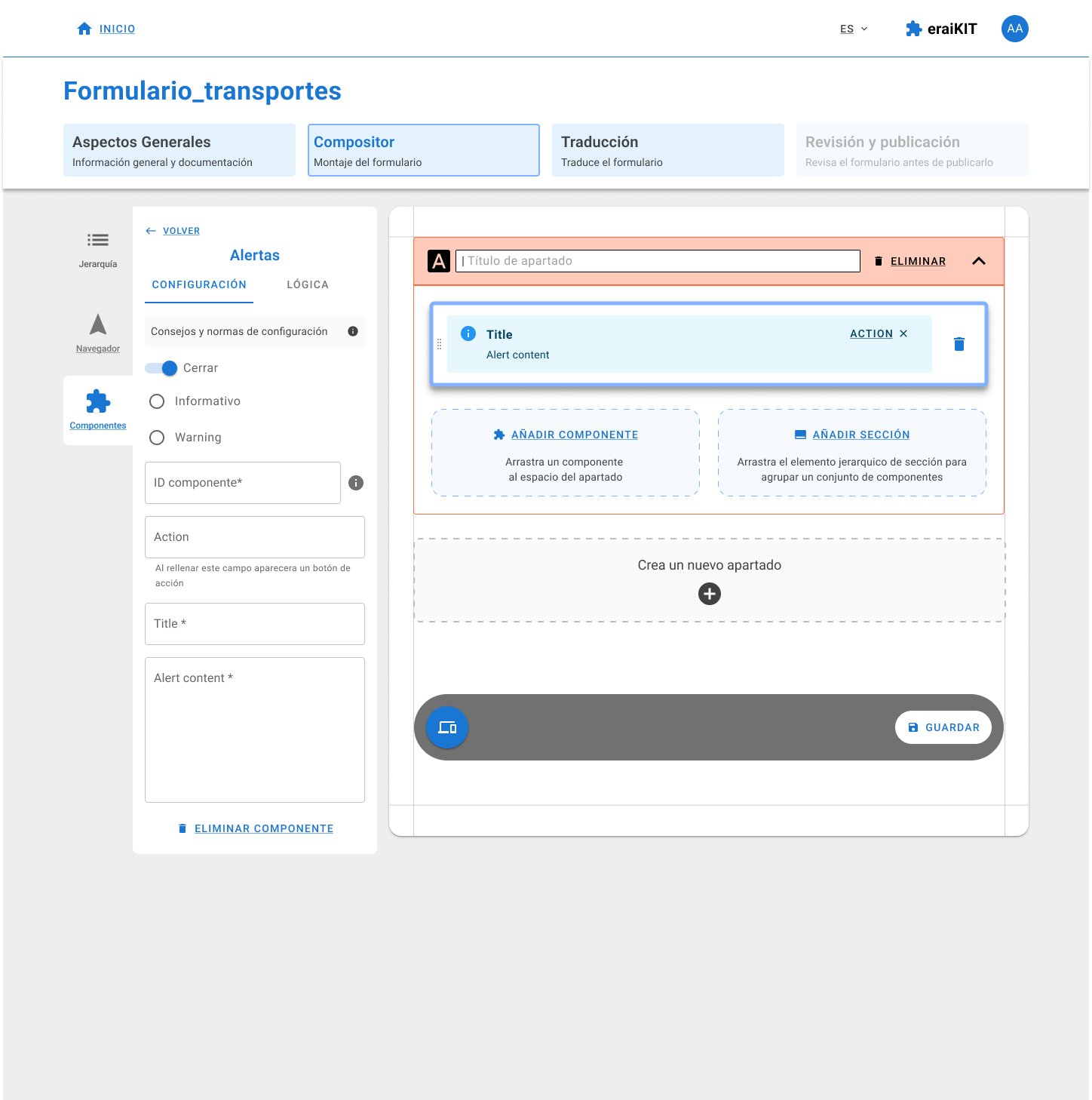
##### Texto enriquecido

Tanto en el componente Párrafo como en las etiquetas de ayuda de los componentes se incorporará el uso de texto enriquecido, que permitirá aplicar listas (ordenadas y no ordenadas), negrita, cursiva y subrayado, generar enlaces y definir estilos de título y párrafo según los tamaños y estilos predefinidos en el formulario: Párrafo (body-1) y Título (body-1-bold).



##### Componente Alertas

Se empleará cuando en alguno de los pasos posteriores se vaya a tener que aportar documentación en función de lo introducido en un componente, así como cuando sea importante informar al usuario sobre algo.



Mensajes de aviso ante finalizaciones exitosas o validaciones de datos incorrectos, incompletos o incongruentes a nivel de formulario.





**Configuración:**

* Selección del tipo de aviso: informativo (azul) o warning (naranja)

Componentes de captura de datos

##### Componente Input

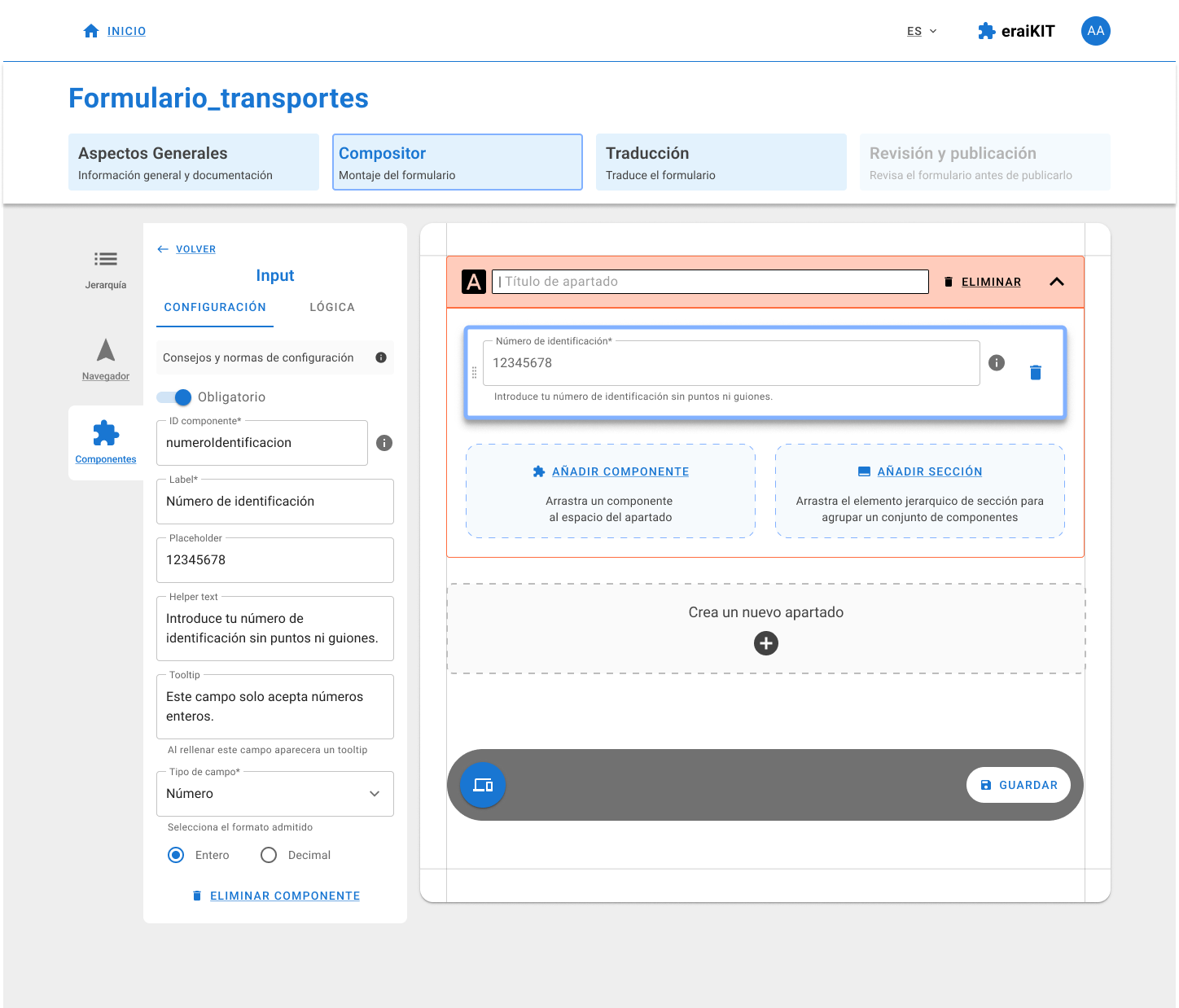
El componente **Input** se emplea cuando se necesita capturar **información breve y precisa** de parte del usuario. Es ideal para datos estructurados como nombres, correos, fechas o cantidades. Permite establecer restricciones como el tipo de dato (texto, numérico, email, contraseña, etc.), lo cual garantiza una mejor validación del formulario.

Se recomienda su uso cuando:

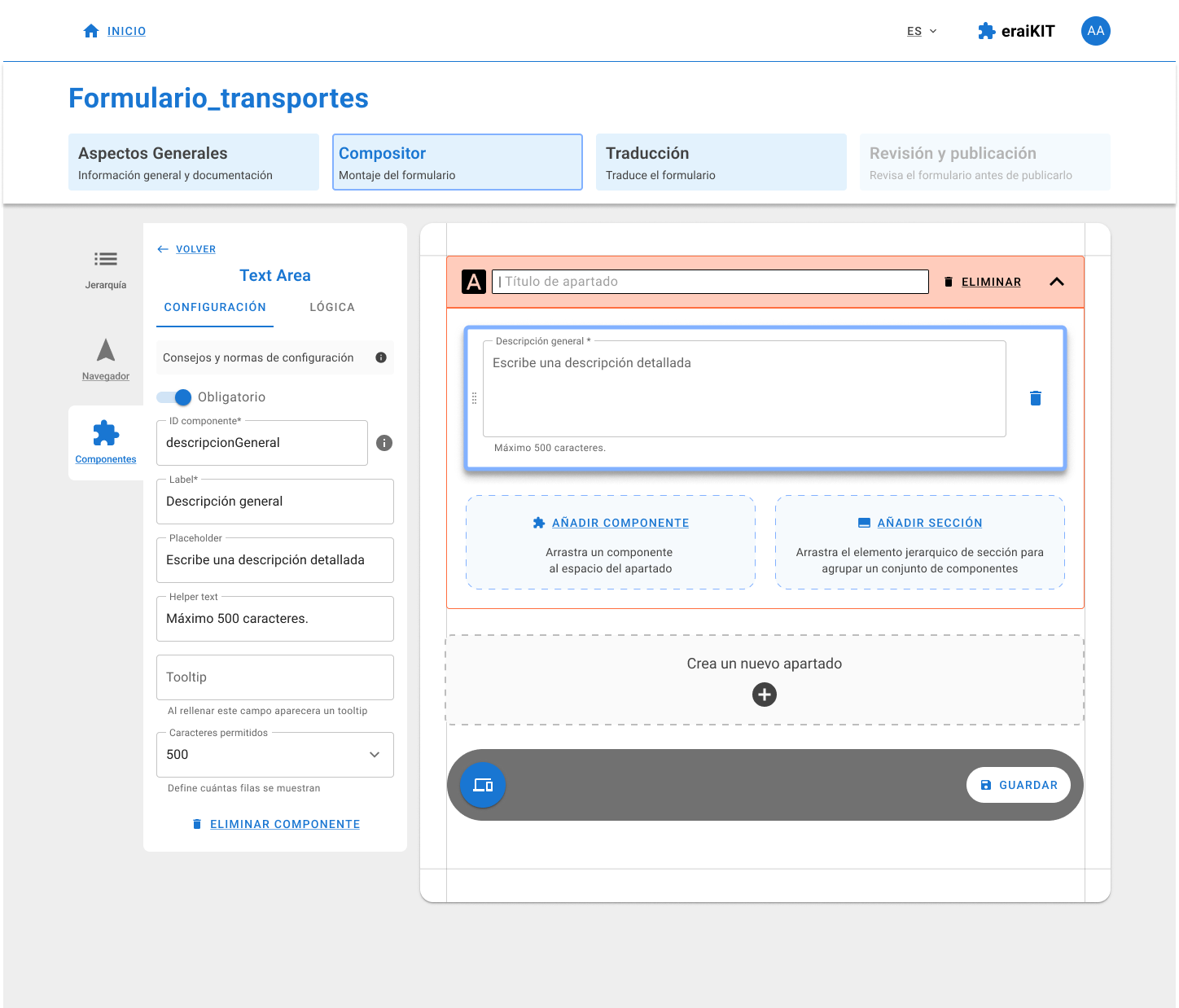
* Se espera una única línea de texto.
* El dato es obligatorio para continuar el proceso.
* El usuario está familiarizado con el tipo de entrada.
* El campo puede ser configurado como:
  + Fijo (fixed): El valor incorporado no puede ser modificado por el ciudadano
  + Oculto (hidden): Es un campo oculto, no visible por el ciudadano.

El reso de atributos aparecerán o no en función de que el campo sea especificado cmoo oculto o fijo.

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| **Obligatorio\*** | Define si es obligatorio completarlo. |
| **ID de componente\*** | Name, configurable y obligatorio |
| **Tipo** | Normal, Fijo y Oculto |
| **Label\*** | Título o descripción breve del campo. Obligatorio. |
| **Placeholder** | Texto sugerido dentro del campo que indica el tipo de dato esperado. |
| **Helper Text** | Mensaje auxiliar que aclara el propósito o uso del campo. |
| **Tooltip** | Información adicional al pasar el cursor sobre un ícono. Si este atributo no se rellena, no aparece tooltip |
| **Tipo de campo\*** | Define el formato: texto, número, fecha, contraseña, moneda, etc. Configuracion de decimal o entero |



##### Componente Text Area



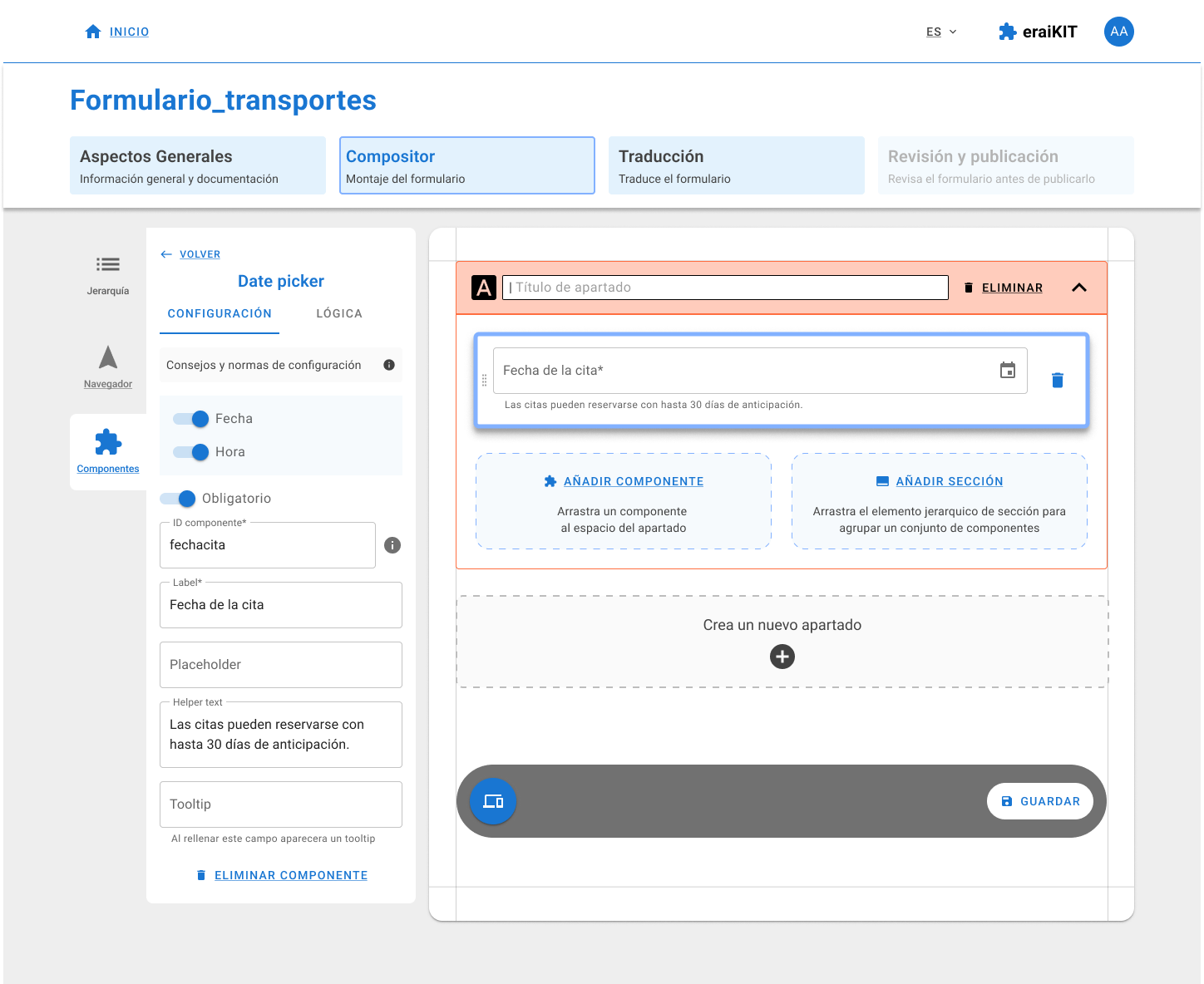
El componente **Text Area** es perfecto para recoger **entradas largas, descriptivas o de redacción libre**. A diferencia del input, permite saltos de línea y mayor extensión, lo cual es clave para encuestas, formularios de contacto o evaluaciones cualitativas.

Se recomienda su uso cuando:

* Se requiere redactar comentarios, sugerencias o descripciones.
* El contenido puede ocupar varias líneas.
* No hay una estructura rígida del contenido esperado.

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| **Obligatorio\*** | Define si debe completarse para enviar el formulario. |
| **ID de componente\*** | Name, configurable y obligatorio |
| **Label\*** | Identifica el campo. |
| **Placeholder** | Guía al usuario sobre qué escribir. |
| **Helper Text** | Ayuda adicional sobre el propósito del campo. |
| **Tooltip** | Detalles informativos adicionales. |

##### Componente Date Picker



El **Date Picker** (selector de fecha) se utiliza cuando es necesario que el usuario seleccione una fecha o un rango de fechas. Es especialmente útil para formularios que gestionan datos personales, nacimientos,eventos, reservas, vencimientos, estancias, etc. Este componente mejora la precisión y evita errores en el formato de entrada de fecha, a la vez que permite validaciones automáticas (por ejemplo, no permitir fechas pasadas).

**Por defecto Hora + fecha seleccionado**. se puede personalizar con un switch; solo fecha, o solo hora pero siempre tiene que haber uno.

Se recomienda su uso para:

* Indicar fecha de nacimiento
* Agendar citas
* Establecer fechas de entrega
* Seleccionar periodos

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| **Obligatorio\*** | Define si debe marcarse al menos una opción. |
| **ID de componente\*** | Name, configurable y obligatorio |
| **Label** | Texto que identifica el propósito del selector de fecha. |
| **Placeholder** | Texto que aparece cuando no se ha seleccionado ninguna fecha. |
| **Helper text** | Información adicional para ayudar al usuario a elegir correctamente. |
| **Tooltip** | Información contextual sobre cómo usar el campo, visible al pasar el cursor. |
| **Fecha por defecto** | Valor preseleccionado al cargar el formulario. |

Variable que permite introducir una Fecha/hora.​

Calendario

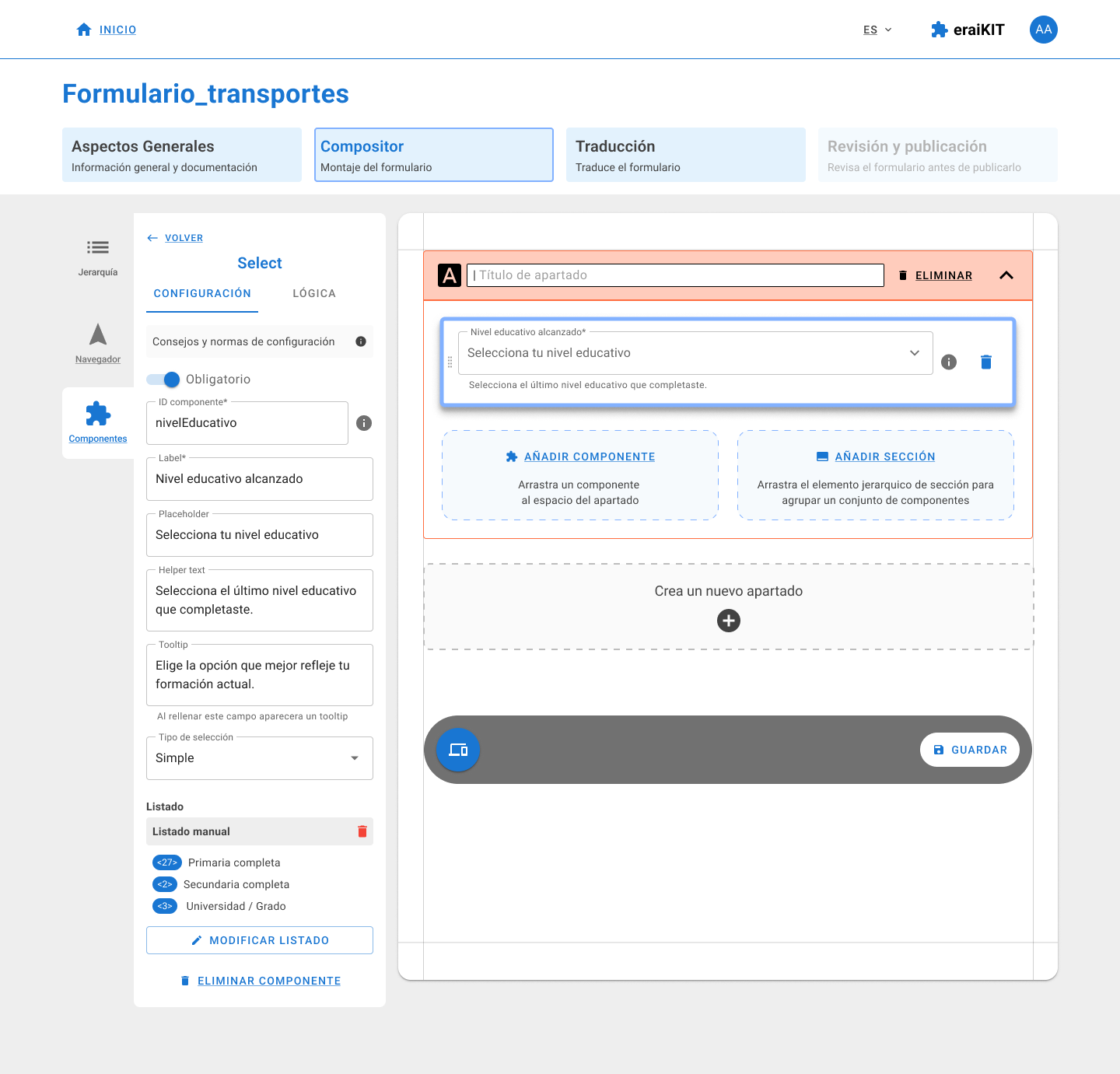
Descripción generada automáticamente

**Configuración​**

* Validará que la fecha es correcta contra calendario y si, además de la fecha, se requiere la hora. Abrirá tanto un calendario como un combo horario para facilitar el marcaje sin teclado.​
* Permitirá aplicar funciones de fecha para sumar y restar días, meses o años, calcular número de días entre dos fechas, etc.

##### Componente Select

El componente Select se utilizará cuando es necesario mostrar un desplegables con varios valores o elementos de entre los cuales el usuario podrá seleccionar uno o varios de ellos.



La configuración del componente permite decidir si la selección de elementos puede ser simple o múltiple.

1. **Select simple**

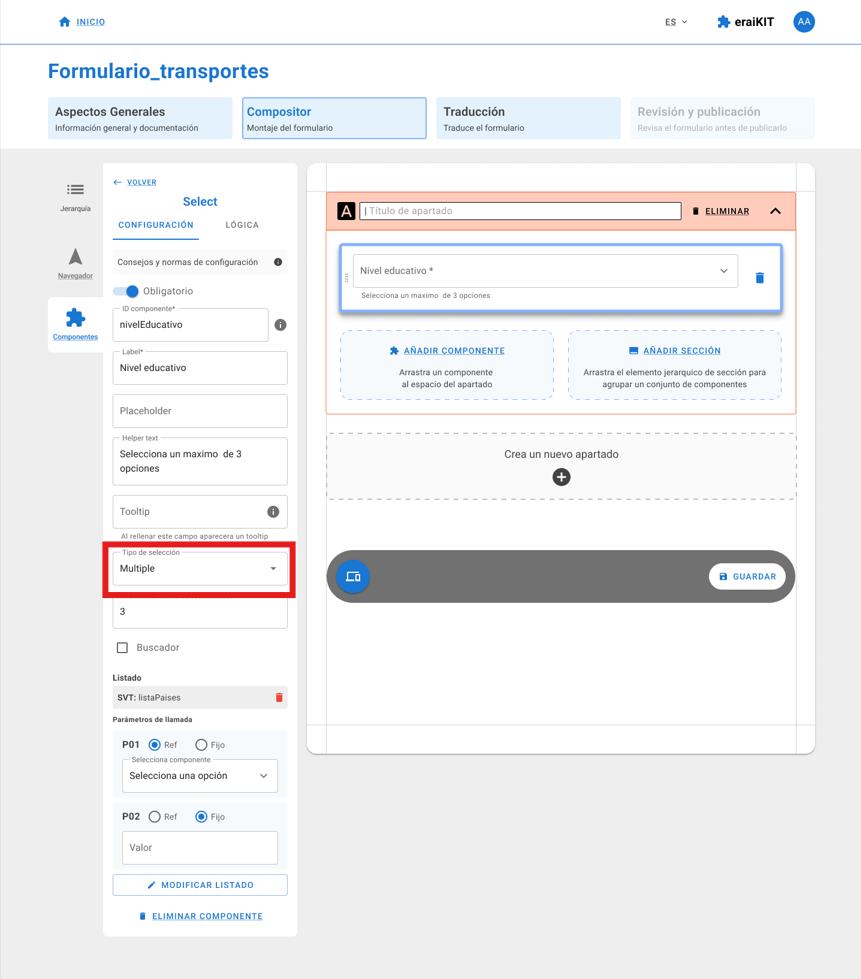
El **Select** simple permite que el usuario elija **una única opción de una lista predefinida**. Resulta útil para reducir errores de entrada, estandarizar los datos y simplificar la interfaz, especialmente cuando hay muchas opciones (más de cinco), o si estas pueden ordenarse alfabéticamente. El componente dispondrá de un buscador cuando tenga N opciones

 Se recomienda su uso cuando:

* Solo se debe seleccionar una opción de una lista extensa.
* Se desea evitar errores de tipeo.
* El listado puede actualizarse dinámicamente desde una fuente de datos.

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| **Obligatorio\*** | Si debe seleccionarse una opción. |
| **ID de componente\*** | Name, configurable y obligatorio |
| **Label** | Título del selector. |
| **Placeholder** | Texto que se muestra cuando no se ha hecho una selección. |
| **Helper Text** | Instrucciones o ayuda contextual. |
| **Tooltip** | Información complementaria al pasar el cursor. |
| **Tipo de selección** | Simple o múltiple |
| **Cargar listado** | Opción a seleccionar, pueden ser manuales,desde SVT o desde listados Toolkit |

1. **Select múltiple o multiselect**



Este componente es ideal cuando el usuario necesita **seleccionar múltiples elementos simultáneamente** desde una lista. Aporta flexibilidad y permite una mejor personalización de las elecciones. Es común en configuraciones de usuario, selección de tags o intereses, y paneles de preferencias.

 Se recomienda su uso cuando:

* Se permite o se necesita seleccionar más de una opción.
* La lista de opciones es manejable visualmente.
* Se requiere granularidad en las selecciones.

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| **Obligatorio\*** | Define si al menos una opción debe ser seleccionada. |
| **ID de componente\*** | Name, configurable y obligatorio |
| **Label\*** | Nombre del campo. |
| **Placeholder** | Texto visible si no se ha hecho ninguna selección. |
| **Valores por defecto** | Opciones preseleccionadas. |
| **Helper Text** | Texto adicional explicativo. |
| **Tipo de selección** | Simple o múltiple |
| **Buscador** | Opción de añadir buscador |
| **Límite de selección** | Cantidad máxima de ítems que se pueden seleccionar. |
| **Cargar listado** | Opción a seleccionar, pueden ser manuales, desde SVT o desde listados Toolkit |

Permitirá la selección de un valor de la lista.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

**Configuración**

* Admitirá que la lista se pueda crear de forma manual o mediante la selección de una llamada a un servicio de terceros.
* Permitirá mostrar la opción en blanco (o “seleccione una opción”). En el caso de ser obligatorio y mostrar sólo una opción en la lista, se autorrellenará directamente.
* Se asociará cada valor de la lista a un valor de backend para su guardado.
* Se guardará la descripción del item seleccionado además del código.

Visualización en los formularios

Interfaz de usuario gráfica, Texto, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

##### Componente Checkbox

Los **checkboxes** permiten marcar **una o varias opciones independientes**, lo que los hace ideales para campos de selección múltiple donde **no es obligatorio elegir todas las opciones**. Su ventaja es que son siempre visibles, lo que permite a los usuarios comprender todas las alternativas rápidamente. Orientación siempre en vertical.

 Se recomienda su uso cuando:

* El usuario puede seleccionar múltiples ítems opcionales.
* No hay un orden estricto entre las opciones.
* Las opciones son fácilmente comprensibles sin interacción adicional.

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| **Obligatorio\*** | Define si debe marcarse al menos una opción. |
| **ID de componente\*** | Name, configurable y obligatorio |
| **Label del grupo\*** | Nombre general del conjunto de checkboxes. |
| **Texto** | Por si requiere de un texto que acompañe a las opciones |
| **Opciones\*** | Lista de elementos marcables. |
| **Valores por defecto** | Opciones marcadas por defecto. |
| **Helper Text** | Ayuda contextual. |
| **Tooltip** | Información emergente adicional. |
| **Cargar listado** | Opción a seleccionar, pueden ser manuales, desde SVT o desde listados Toolkit |

Muestra un listado que permite la selección de múltiples opciones.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

**Configuración**

* + Se definirá el número de checkbox a mostrar, con los textos asociados a cada checkbox.
  + Los checkbox siempre van configurados en vertical.
  + Se asociará un valor de backend a cada punto de checkbox.

##### Componente Radio Button

Los **radio buttons** son excelentes cuando el usuario debe elegir **una opción entre pocas disponibles**, y es importante que **todas estén visibles al mismo tiempo**. Brindan claridad y accesibilidad inmediata sin necesidad de desplegar una lista.

 Se recomienda su uso cuando:

* Se tienen entre 2 y 5 opciones.
* La elección debe ser evidente sin interacción adicional.
* La presentación visual facilita la comparación rápida.

|  |  |
| --- | --- |
| **Atributo** | **Descripción** |
| **Obligatorio\*** | Define si es necesario seleccionar una opción. |
| **ID de componente\*** | Name, configurable y obligatorio |
| **Label del grupo\*** | Título del conjunto de opciones. |
| **Texto** | Por si requiere de un texto que acompañe a las opciones |
| **Opciones\*** | Lista de valores posibles. |
| **Valor por defecto** | Valor preseleccionado. |
| **Orientación\*** | Disposición vertical u horizontal. |
| **Helper Text** | Ayuda contextual adicional. |
| **Tooltip** | Información expandible. |
| **Cargar listado** | Opción a seleccionar, pueden ser manuales, desde SVT o desde listados Toolkit |

Muestra una serie de botones de selección que permite escoger un valor determinado para capturar un dato con un click. Admitirá la opción “otro…” para poder ingresar un valor personalizado.

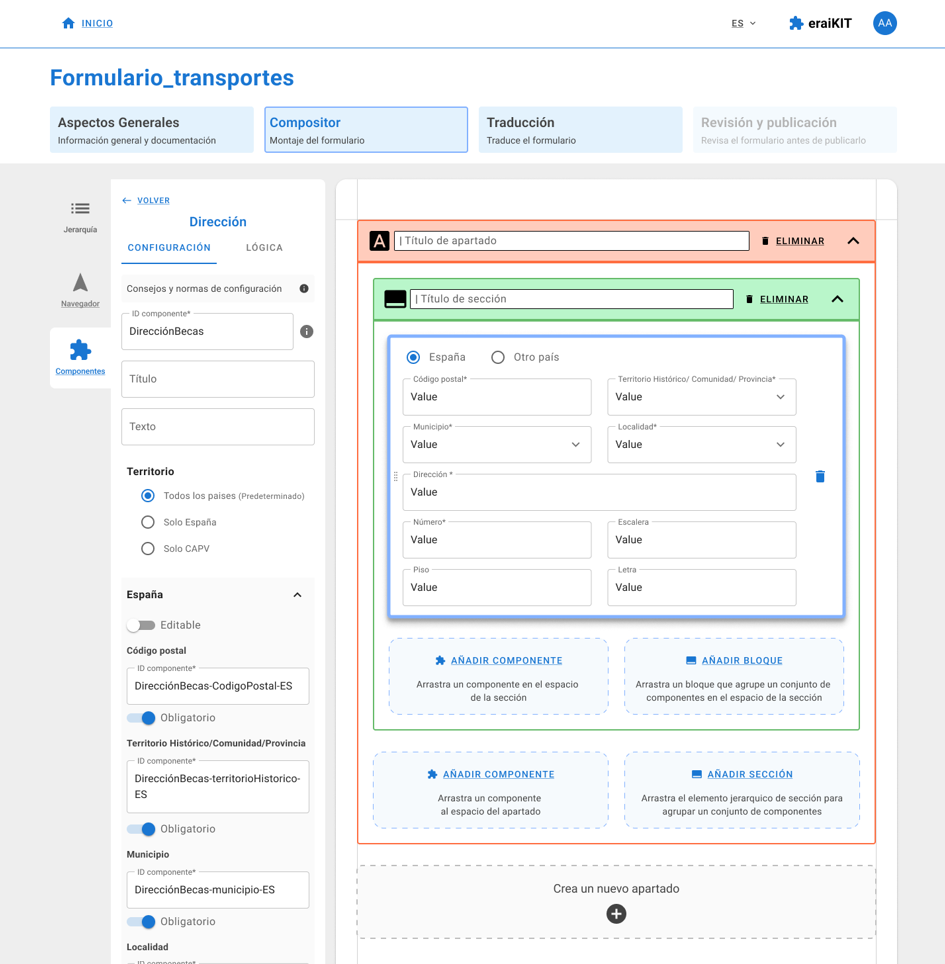
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Configuración**

* Se definirá el número de Radio Button a mostrar, con los textos asociados a cada uno de ellos.
* Si hay 2 Radio Button, por defecto irán alineados en horizontal, a no ser que el espacio no lo permita. En caso de que haya más de 2, se alinearán verticalmente.
* Se asociará un valor de backend a cada Radio Button. Esta alineación no lo configura el funcionario, si no que el compositor detecta la situación lo ordena según necesidad.
* Se permitirá definir un RadioButton con el texto “otro” para que el usuario pueda introducir la descripción de la selección

##### Componente Dirección Postal



* Captura la dirección mediante el uso del callejero de NORA.
* Los datos se mostrarán en el siguiente orden: Código postal, País, Territorio histórico, Municipio, Localidad, Tipo de calle, Dirección, Número (con bloque), Escalera, Piso, Puerta.
* Realizará el autocompletado de los valores deducibles: por ejemplo,en base al CP que se introduzca: Territorio histórico, Municipio, Localidad.
* Los componentes que conforman el Componente Dirección Postal son específicos para conformar este componente, es decir, no tendrán opciones de configuración ni lógica propias (subcomponentes)
* De los subcomponentes no se podrá modificar la configuración (obligatoriedad, configuración de select, radiobutton, etiquetas), ni la lógica (validaciones,acciones, SVT’s) ya que vendrá preestablecida por el propio componente dirección.
* Desde el componente Dirección Postal se establecerán configuraciones como obligatoriedad o visualizable que afectan a cada uno de los subcomponentes.
* El componente Dirección Postal tendrá un “name” específico (Ej: direccionEmpresa o direccionTitular)
* Los subcomponentes se nombrarán automáticamente añadiendo el “name” padre y los nombres de cada uno de los atributos que siempre serán fijo:

Ej: direccionEmpresa\_codigoPostal, direccionEmpresa\_país, direccionEmpresa\_territorioHistorico, direccionEmpresa\_municipio, direccionEmpresa\_localidad, direccionEmpresa\_dirección, direccionEmpresa\_numeroBloque, direccionEmpresa\_escalera, direccionEmpresa\_piso, direccionEmpresa\_puerta, direccionEmpresa\_latitud, direccionEmpresa\_longitud

De cara a los formularios a migrar, se respetarán los nombres actuales de los componentes para evitar el impacto en el intercambio de datos (*detailInfo*)

**Configuración**

* Poder introducir una dirección extranjera
* Poder introducir SOLO direcciones de la Comunidad Autónoma del País Vasco.
* La dirección será editable o no: Configura si la dirección será sólo autocompletable y si además admite la opción de introducir manualmente (dirección no existente en NORA).
* Definición de qué campos se quieren mostrar
* Definición de qué campos son obligatorios
* Definición de opción de mostrar mapa para facilitar la búsqueda de la calle.
* Si se requieren coordenadas geográficas se obtendrán mediante un componente input o pulsando sobre el mapa.

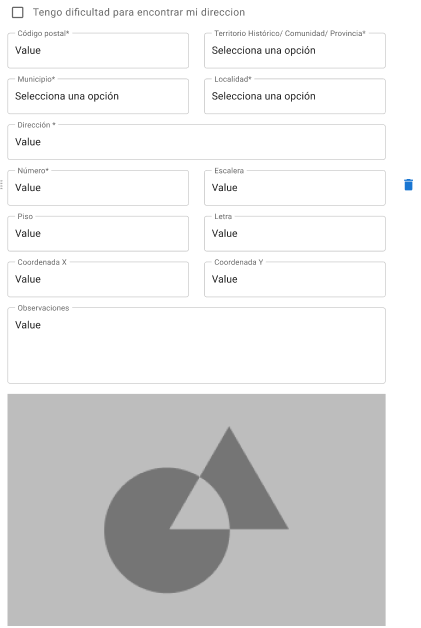
Opciones de configuración del componente Dirección Postal:

* Si está activo el flag “sólo España”, no aparece la opción país y sólo se podrá introducir una dirección nacional.
* Si está activo el flag de “sólo CAPV”, la lista de provincias se llamará Territorio Histórico y sólo mostrarán Alava, Bizkaia y Gipuzkoa. Si no se activa, se mostrará Territorio histórico/Comunidad/Provincia y mostrará todas las de España.
* Si está activo el flag de editable, podrá meterse cualquier dirección aunque no esté en NORA.
* Si no está activo el flag de editable, la dirección y el resto de campos se validará contra NORA. En el caso de que no se pueda completar una dirección a través de la información de NORA (Por ejemplo: no se encuentra la calle o el portal por no estar codificada), el usuario podrá introducir estos datos de manera manual. Para ello podrá activar el check que aparecerá con el texto “***Tengo dificultad para encontrar mi dirección***” y el componente dirección le mostrará los nuevos campos donde incorporar los datos necesarios: dirección, número, escalera, piso y letra.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Si está activa la opción de GPS, aparecerán las coordenadas, observaciones y mapa. Cuando se pulsa en el mapa, se recuperan las coordenadas y la dirección completa del punto marcado si esta estuviera disponible.



Según las diferentes combinaciones en la configuración, se identifican los siguientes casos:

**Sólo España y editable**: introduce el CP, (si es CAPV autorrellena hasta localidad. Si no es CAPV, a mano con selects hasta localidad).

* SI CP es de CAPV, la dirección y resto de campos son select de NORA que, si no encuentra, admite rellenar con otro valor y el resto texto libre.
* SI CP es del resto de España, la dirección y resto de campos se podrán marcar libremente.

**Sólo España y no editable**: introduce el CP. (Si es CAPV autorrellena hasta localidad. Si no es CAPV, a mano con selects hasta localidad).

* SI CP es de CAPV, la dirección y resto de campos son select de NORA. Si no lo encuentra tendrá que ir a la opción de “¿Ha podido completar los campos obligatorios de este apartado?”
* SI CP es del resto de España, no podrá meter la dirección así que tendrá que ir a la opción de “¿Ha podido completar los campos obligatorios de este apartado?”

**Solo CAPV y editable**: Introduce el CP, y autorrellena hasta localidad. Sólo admite CP de las 3 provincias vascas.

* SI CP es de CAPV, la dirección y resto de campos son select de NORA que si no encuentra, admite rellenar con otro valor y el resto texto libre

**Solo CAPV y NO editable**: introduce el CP, y autorrellena hasta localidad. Sólo admite CP de las 3 provincias vascas.

* SI CP es de CAPV, la dirección y resto de campos son select de NORA. Si no lo encuentra tendrá que ir a la opción de “¿Ha podido completar los campos obligatorios de este apartado?”

**NO Sólo España y editable**: aparece el componente button: España - Otro País

* Si elige España funcionará como Sólo España y Editable
* SI elige otro país abre la modal de extranjero con los campos
* **PAIS (Select) Obligatorio**
* Provincia: campo libre
* Ciudad: campo libre
* Dirección: campo libre
* Código Postal: campo libre

**NO Sólo España y No Editable**: Aparece el componente button: España - Otro País

* Si elige España funcionará como Sólo España y Editable
* SI elige otro país abre la modal de extranjero con los campos
* **PAIS (Select) Obligatorio**
* Provincia: campo libre
* Ciudad: campo libre
* Dirección: campo libre
* Código Postal: campo libre

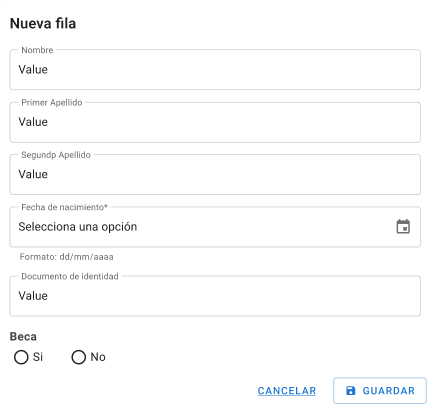
##### Componente Tabla

Permitirá añadir filas a una tabla en base a la configuración de campos establecida en la composición del formulario y generar campos “totales” donde se realizará el cálculo de la operación seleccionada sobre la columna.

Tabla

Descripción generada automáticamente

Al pulsar una línea, se podrá editar o borrar el contenido de la línea. Tanto la incorporación de un nuevo registro como la edición de uno existente se realizará desde una ventana modal como la del ejemplo**:**



**Configuración**

* El componente tabla podrá tener la acción condicional (mostrar)
* Definición de columnas
* El número de columnas de la tabla estará limitado al ancho de la página (5 o 6 columnas máximo dependiendo del tamaño)
  + Las columnas podrán incorporar componentes input, date, select o radiobutton, con sus propias validaciones.
  + Los componentes internos no podrán se referenciados para validaciones o acciones condicionadas
  + Las columnas podrán incorporar un total (media, sumatorio, máximo y mínimo).
  + Las columnas mostradas en la hoja de cálculo podrán ser intercambiables, es decir, se permitirá su reordenación mediante drag&drop.
  + Registros repetidos: Se identificarán en el compositor los campos clave por los que se deba evaluar si hay registros repetidos.
* Definición de filas
  + Cada fila llevará una numeración, no asociada al registro, sino a la posición. (La numeración podrá ser activada o desactivada por el funcionario al crear el formulario)
  + Las filas podrán ordenarse en base a una columna
  + Las filas podrán subirse o bajarse en la tabla de manera manual
  + Se validará la existencia de registros repetidos (las columnas clave podrán ser una o el combinado de varias columnas).
* Pie de la tabla
  + Contendrá un contador de filas (filas totales).
  + Podrán definirse totales para las columnas numéricas (suma, media aritmética).
  + Se guardarán los datos de totales en un “name” para que puedan ser consumidos en otros componentes.
* Documentos Asociados: la tabla podrá configurarse con la opción para admitir documentos asociados. Esta configuración permitirá, en la ejecución del formulario, incluir documentos en el paso 4 – Documentos asociados a cada uno de los registros que cumplimente el ciudadano en la propioa tabla.

Para ello, el paso 4 incorporará un nuevo bloque con el nombre de la tabla en la que se podrán incorporar dichos documentos. Por cada uno de ellos el ciudadano podrá especificar a qué registro de la tabla corresponde.

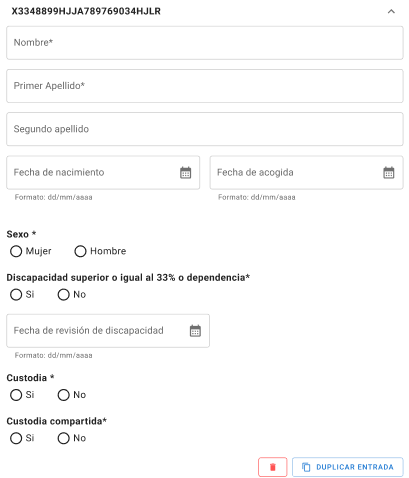
* Cargar desde CSV: en la configuración de la tabla se habilitará una opción para identificar si va a admitir la carga desde CSV. En la solicitud se mostrará la opción de cargar desde CSV, pudiendo obtener un fichero aportado por el ciudadano para su carga automática. El sistema aportará el formato de CSV a requerir, con todas las columnas de la tabla.

##### Componente Grupo Repetición

Cuando se necesita un componente tabla que supere las 6 columnas, se utilizará el Componente de Grupo de Repetición, que mostrará una cabecera con los campos identificativos y un pìe con los totales de los campos numéricos que se identifiquen:

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.



**Configuración**

Mostrará todos los datos de cada registro:

* Cabecera
* Mostrará nombre del grupo repetición y número de items
* Cabecera de registro
* Los registros se configurarán en una sábana colapsable. En la cabecera no colapsable se mostrará:
  + El campo clave de la tabla podrá ser un campo único (Ej.: Matrícula) o la concatenación de varios campos (Ej.: Nombre+Apellido1+Apellido2) que se seleccionarán de entre los campos input y date
  + Tres pills correspondientes a columnas seleccionables (sólo pueden ser componentes input, date, radiobutton o select).
* Definición de columnas
  + Se podrán incorporar diferentes componentes como si fuera un componente bloque.
  + Podrá incorporar cualquier tipología de componente (también tabla), a excepción de componentes organizativos y la anidación de otro grupo de repetición
  + Los componentes internos no podrán se referenciados para validaciones o acciones condicionadas.
* Definición de filas
  + Podrán añadirse tantos registros como el ciudadano considere
  + Las filas tendrán botones de opción para duplicar o borrar la entrada.
* Pie de grupo:
  + Podrán definirse totales para las columnas numéricas (suma, media aritmética, máximo o mínimo). Se mostrarán en una línea debajo de añadir nueva entrada con el formato:
  + Si se trata del campo “importe” aparecerá: “Importe (total): XXXXX” o “Importe (medio): XXXXXX”
  + Se guardarán los datos de totales en un “name” para que puedan ser consumidos en otros componentes
  + Permitirá añadir nueva entrada en el grupo de repetición.

En el momento de la composición y configuración del componente , el grupo de repetición se visualizará como un componente jerárquico que permita incluir diferentes componentes.

### Formatos específicos

Un componente podrá configurarse con formatos específicos dada la naturaleza del valor que va a contener. Los formatos incluídos en la configuración son los siguientes:

* **Formato Moneda**: numérico con 2 decimales y el símbolo € al final
* **Formato Moneda 2**: numérico con 2 decimales sin el símbolo € al final
* **Formato cuenta bancaria**: 9999 9999 99 9999999999
* **Formato IBAN**: XX99 9999 9999 99 9999999999
* **Número Seguridad Social**
* **Número entero**
* **Máscara customizable** para que el usuario defina cómo va a ser el formato del contenido

### Fragmentos

El concepto “fragmento” es un nuevo elemento de ayuda a la composición de formularios que permite guardar conjuntos de componentes relacionados y configurados con sus validaciones, funciones y acciones reutilizables en múltiples formularios, agilizando el tiempo de diseño y composición.

Se distinguen dos tipos de fragmentos con funcionamientos distintos:

* **Fragmentos comunes**, gestionados por el grupo funcional “Platea-Tramitación” y usable por todos los grupos funcionales.
* **Fragmentos específicos**, gestionados por cada uno de los grupos funcionales y usables sólo por el grupo funcional que lo ha creado.

A la hora de componer un formulario, se podrán utilizar desde el compositor a través de la opción que aparece en el bloque de “componentes”..

Las diferencias entre los dos tipos de fragmentos definidos se basan en la seguridad para su creación y gestión y en el modo en el que se incorporan en los formularios. La siguiente tabla muestra dichas diferencias

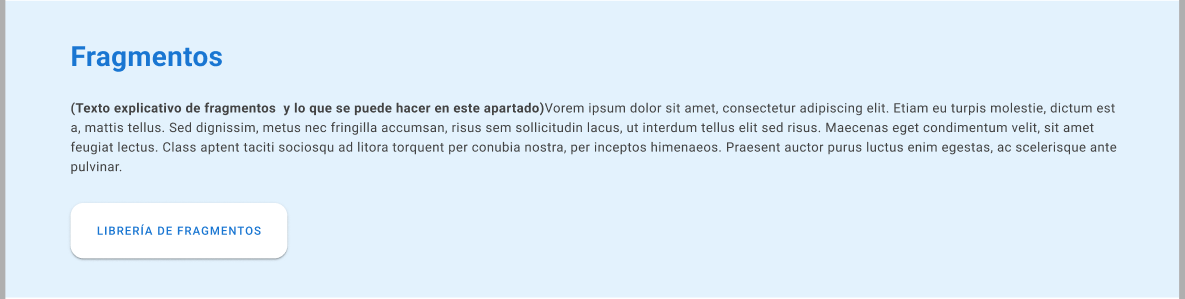
|  |  |  |
| --- | --- | --- |
|  | Fragmento específico | Fragmento común |
| ¿Quién lo gestiona? | Usuarios de un grupo funcional. Pueden ser creados y modificados por cualquier usuario del grupo funcional al que pertenece | Usuarios administradores (usuarios del grupo funcional “Platea Tramitación”). Sólo creados y editados por los usuarios de este grupo funcional. |
| ¿Quién lo puede usar? | Cualquier usuario del mismo grupo funcional al que pertenece el fragmento | Todos los usuarios de cualquier grupo funcional. |
| ¿Cómo se integra en el formulario? | Se copia en el formulario expandiendo todos los componentes del fragmento, pasando a ser parte del nuevo formulario donde puede modificarse y adaptarse. | Se añade al formulario como una referencia. |
| ¿Cómo le afectan los cambios? | Un cambio afecta a los próximos formularios que lo incorporen, ya que, en los anteriores, el contenido del fragmento ha pasado a formar parte del propio formulario. | Un cambio impacta en todos los formularios que lo usan. Al estar referenciado, en el momento de la ejecución del formulario, se incorporará el contenido del fragmento referenciado. Los cambios se incorporarán en sucesivas versiones del fragmento. |
| ¿Se pueden eliminar? | Pueden eliminarse en cualquier momento que se considere. | No pueden eliminarse si están en uso por algún formulario. |

La edición y gestión de los fragmentos ya sean comunes o específicos, tendrá asociado un propio editor desde donde el usuario podrá consultar los fragmentos disponibles, su estado (publicado o borrador) y podrá iniciar la creación de un nuevo fragmento.

Este editor tendrá la misma apariencia que el propio compositor ya que el modo de operación para su creación es similar a la de crear un formulario completo. En este caso, este compositor será una versión reducida dado que en la creación de los fragmentos no está contemplado la utilización de componentes jerárquicos, es decir, no se pueden utilizar los componentes para la creación de apartados, bloques, secciones….

Los pasos para la creación de un fragmento también seguirán la misma estructura que la de un formulario: creación del fragmento, traducción y revisión y publicación, momento en el que ya estará disponible para su uso en los formularios.

El acceso a la gestión de la librería de fragmentos a los que tiene acceso el usuario conectado se realiza desde la pantalla principal de EraiKIT.



La pantalla principal de la gestión de fragmentos presentará dos “pestañas” diferenciando los fragmentos comunes y los fragmentos específicos.

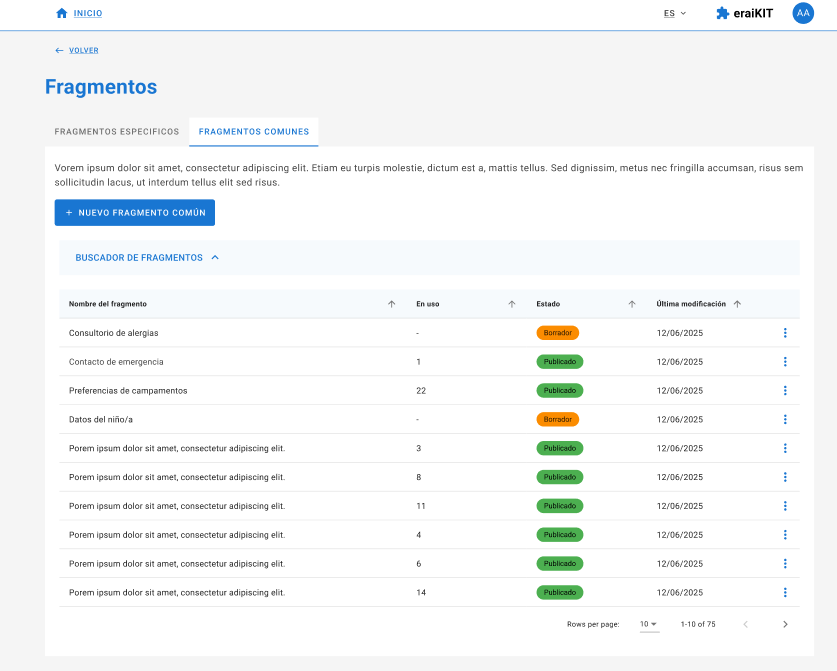
Estas pestañas darán acceso a la lista de fragmentos de cada tipo junto a:

* Un botón para iniciar el proceso de incluir un nuevo fragmento
* Un buscador para localizar fragmentos por nombre
* Tabla con todos los fragmentos existentes.

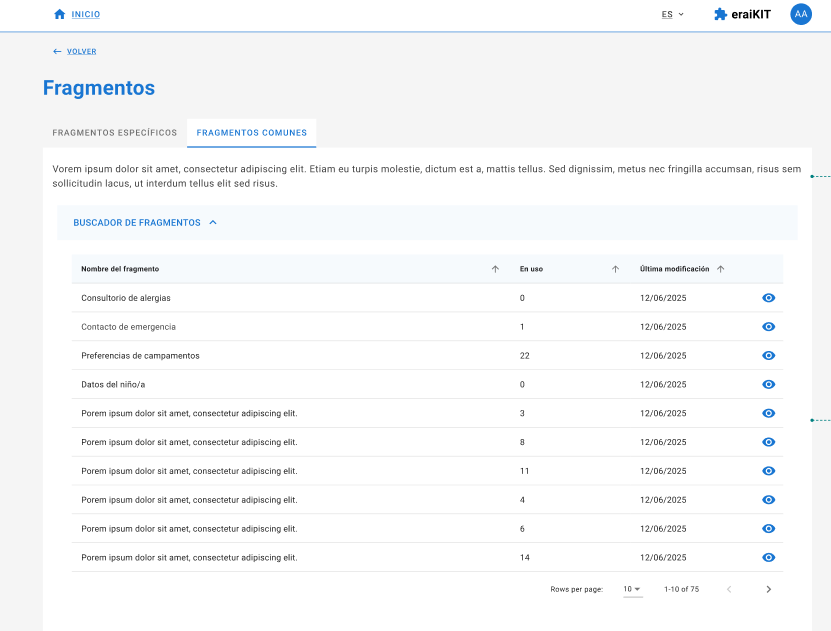
Esta tabla presentará diferentes opciones en función del grupo funcional al que pertenezca el usuario ya que:

* Si pertenece al grupo funcional “Platea Tramitación” podrá crear y modificar fragmentos comunes pero sólo podrá visualizar y consultar los fragmentos del resto de grupos funcionales.
* Si no pertenece al grupo funcional “Platea Tramitación” sólo podrá crear y editar fragmentos específicos de sus grupos funcionales. Sólo podrá visualizar y consultar los fragmentos comunes.

**Pantalla de “fragmentos comunes” para un usuario del grupo funcional “Platea Tramitación”**



**Pantalla de “fragmentos comunes” para un usuario de otro grupo funcional:**

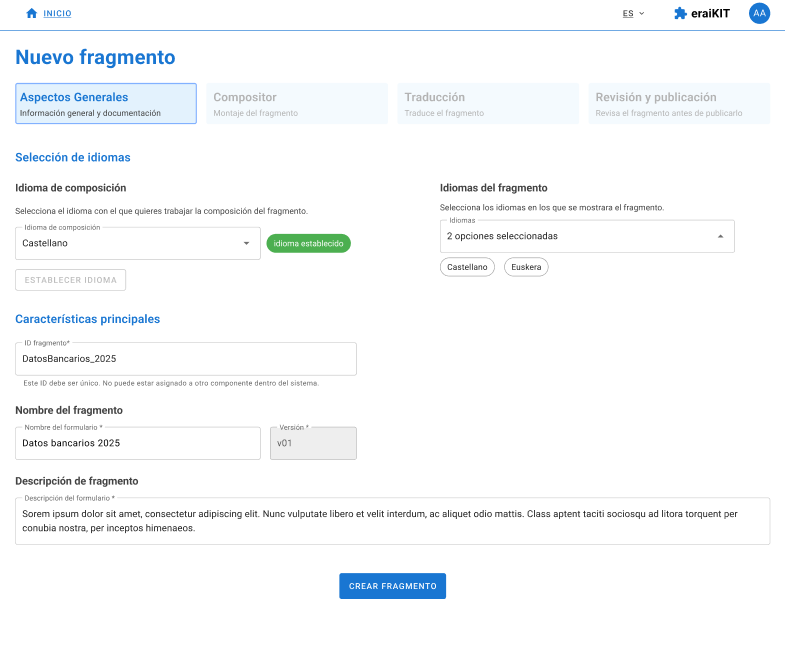


Nuevo Fragmento

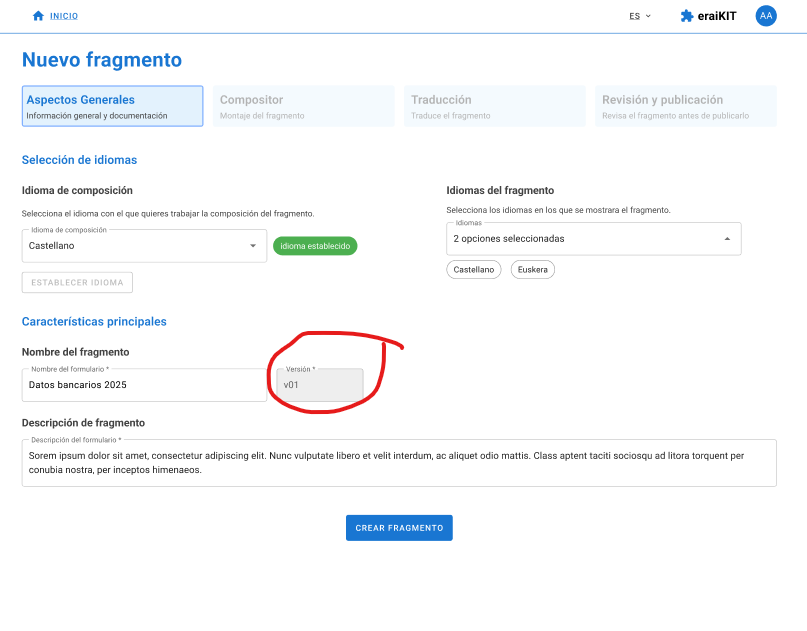
Para construir un nuevo fragmento, ya sea común o específico, EraiKIT presentará un compositor similar a la composición de los formularios, dado que la creación de los fragmentos sigue los mismos pasos que la composición de un formulario. Existen 4 pasos principales:

* **Aspectos generales**, donde se podrán incluir los datos principales: nombre, idiomas, grupo funcinal al que pertenece, versión en el caso de los fragmentos comunes...

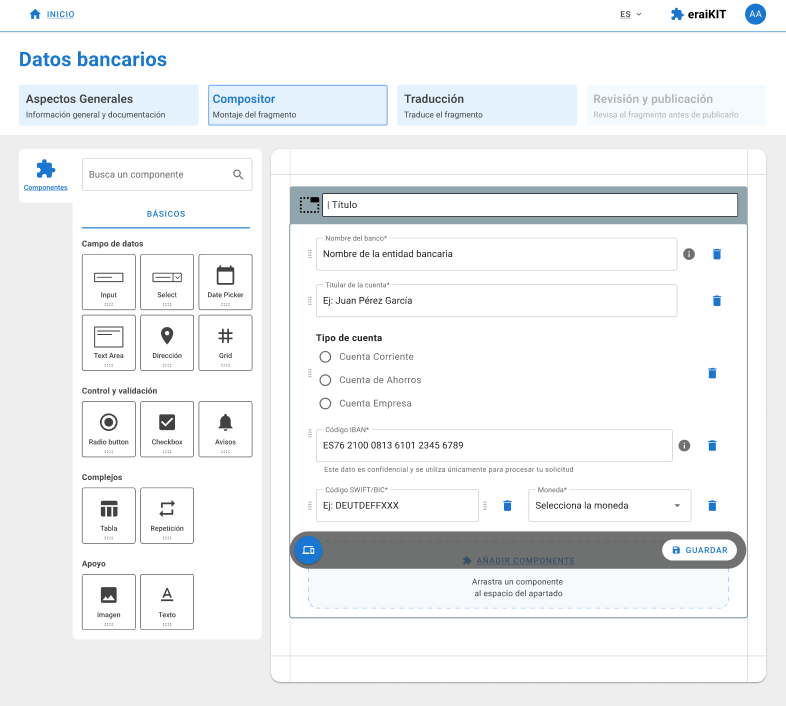
**Nuevo fragmento común**



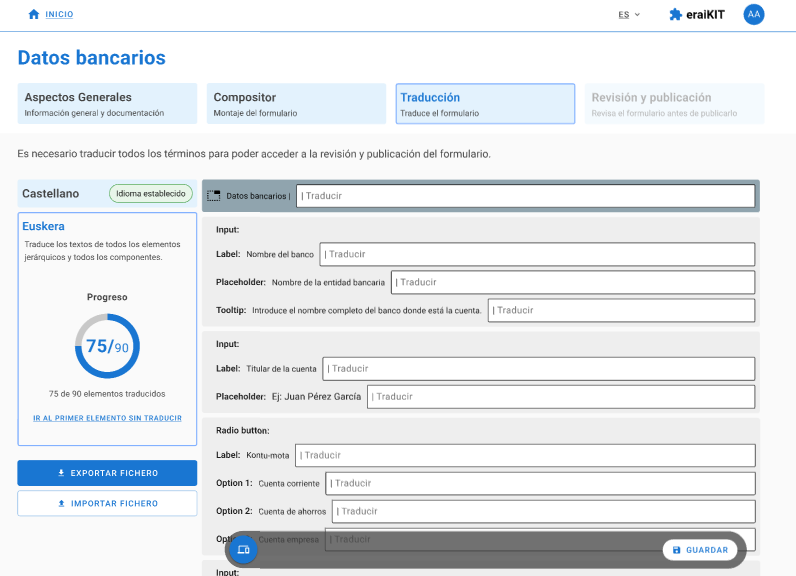
**Nuevo fragmento específico**



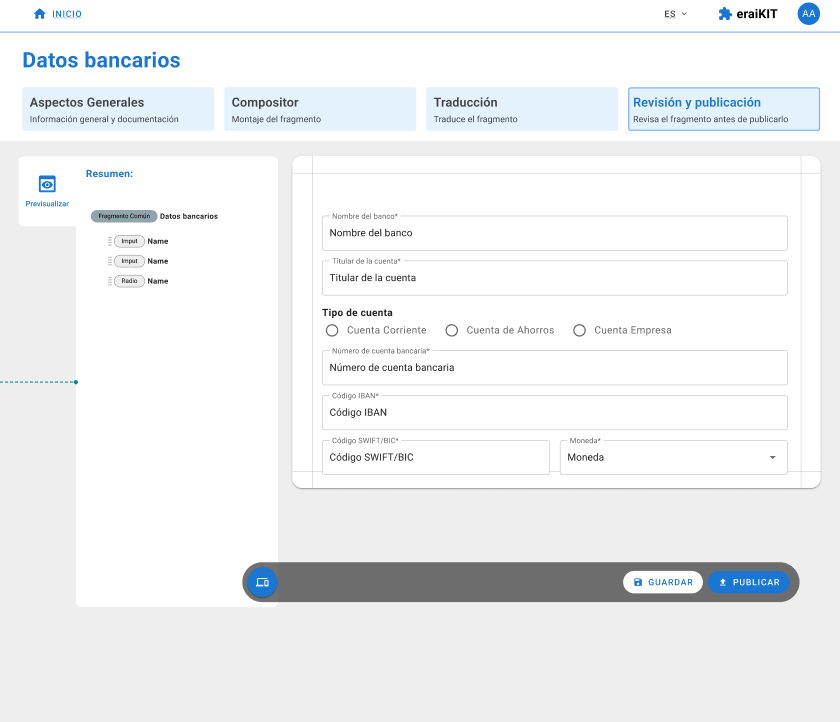
* **Compositor**, pantalla en la que se creará el fragmento en sí. Mostrará la lista de componentes que se pueden incluir. En este caso, no es posible utilizar componentes de tipo organizativo (apartados, secciones, bloques, ...)



* **Traducción**. Una vez creado el fragmento se deberá traducir a través del paso de traducción y no podrá pasarse al siguiente paso y publicarlo mientras no esté totalmente traducido.



* **Revisión y publicación**. En este paso se podrá revisar el funcionamiento y configuración del fragmento visualizando cómo aparecerá en el formulario y cómo funcionarán las validaciones y acciones condicionales incorporadas en el mismo. Una vez revisado, se podrá “publicar” para que esté disponible para su uso.





Editar un fragmento

El acceso a la edición de un fragmento se basará en la tipología del mismo (común o específico) y a la seguridad del usuario conectado (a qué grupo o grupos funcionales pertenece).

Fragmentos comunes

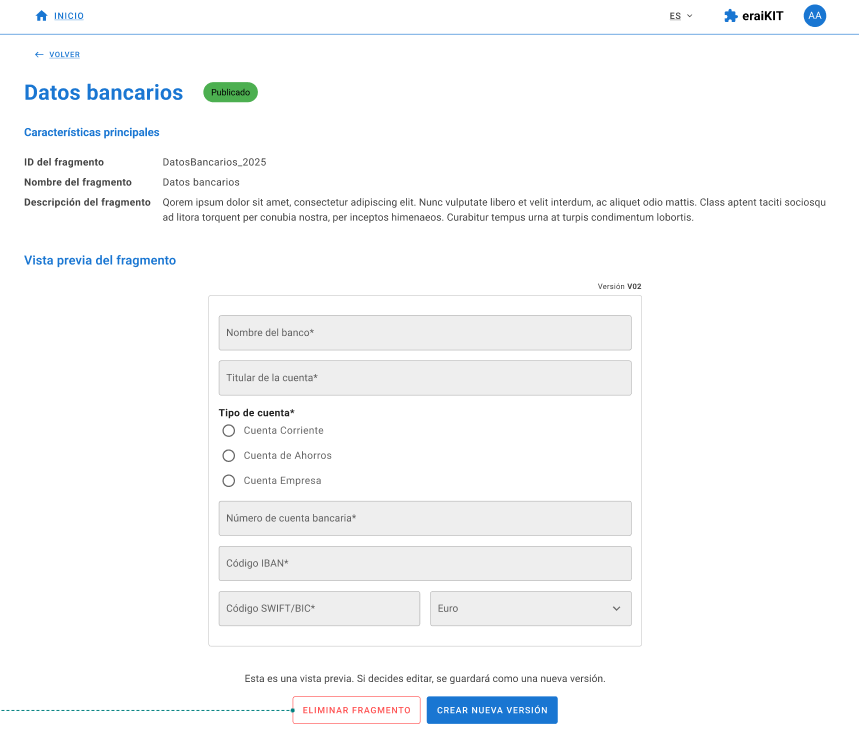
* Sólo podrán ser modificados por los usuarios del grupo funcional “Platea Tramitación”.
* La modificación del fragmento estará disponible mientras se encuentre en estado “borrador”.
* Una vez publicado, sólo será posible crear una nueva versión para incluir cambios en la definición del fragmento.

Fragmentos específicos

* Podrán ser modificados por los usuarios del grupo funcional al que pertenece el fragmento.
* Se podrán modificar en cualquier situación ya que las modificaciones no impactan en los formularios en los que ya se incluyeron.
* Estado “borrador” / “publicado”

Vista previa de un fragmento

Antes de entrar a modificar un fragmento, o desde el propio listado de fragmentos, es posible realizar una vista previa del contenido de un fragmento. Esta pantalla mostrará visualmente la composición del mismo y, en base a la seguridad de acceso del usuario, le mostrará las opciones para eliminarlo y/o modificarlo. En el caso de los fragmentos comunes, esta modificación se realizará en base a la creación de una nueva versión del mismo.



Gestión de versiones

La creación de versiones de los fragmentos sólo estará disponible para los fragmentos comunes. Dado que los fragmentos comunes se incluyen en los formularios en base a la referencia al nombre del fragmento, no es posible modificarlos una vez que están en estado “publicado”.

Para poder incluir cambios, de aspecto o de funcionamiento, es necesario crear una nueva versión del mismo que será ejecutada de forma automática en todos los formularios que referencien al fragmento una vez pase a estado “publicada”.

Por consiguiente, en la ejecución de un formulario que incluya un fragmento común se incluirá siempre el contenido de la última versión publicada de dicho fragmento.

Uso de fragmentos

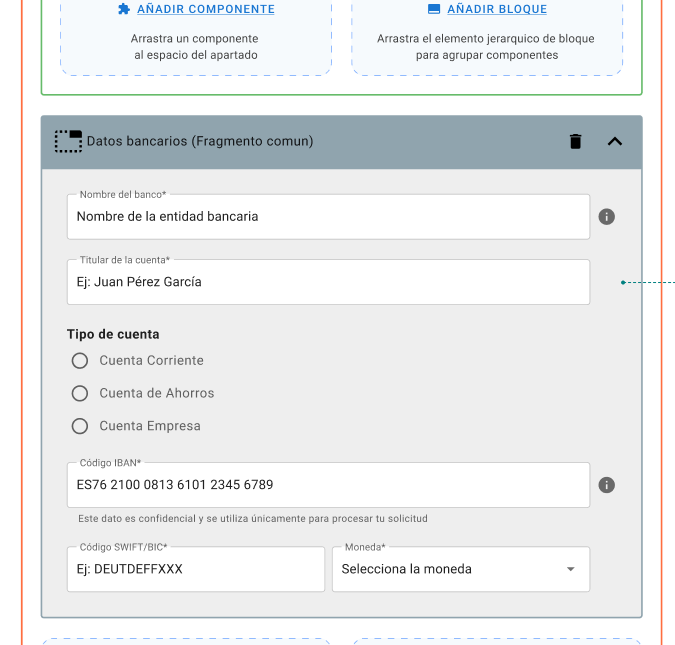
Una vez creados los fragmentos, éstos se pueden incluir en los formularios como si de un componente más se tratase. Para ello, aparecerá una opción “fragmentos” en el compositor:  


Esta opción mostrará todos los fragmentos disponibles (aquellos que estén publicados), comunes y específicos, en dos bloques diferenciados. En el caso de los fragmentos “específicos” mostrará sólo aquellos que pertenezcan al mismo grupo funcional que el propio formulario.

Para incluir un fragmento en el formulario que se está editando tan sólo hace falta seleccionarlo y arrastrarlo a la zona en al que se desea que se visualice.

**Fragmentos comunes**

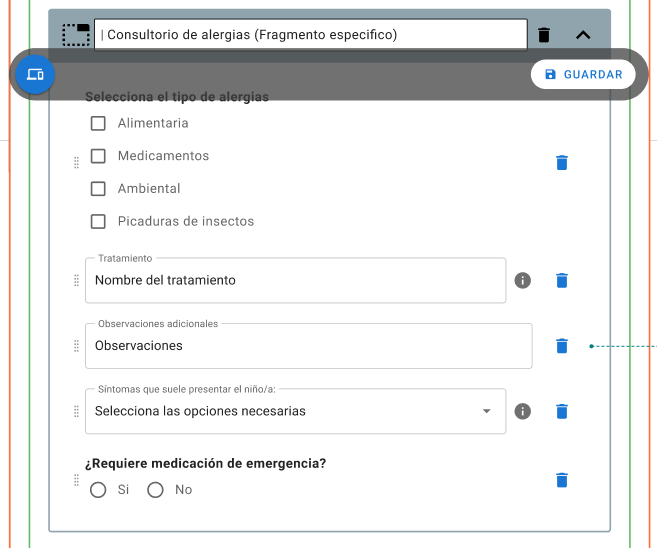
Los fragmentos comunes se incorporan en el formulario como una referencia. Esto quiere decir que los componentes internos del fragmento no se incluyen directamente en el propio formulario, sino que se crea una referencia al nombre del fragmento. Esto permite que, a la hora de mostrarse el formulario, se incluyan dinámicamente los componentes, validaciones y configuración de la última versión de el fragmento referenciado.



No es posible la edición ni modificación de ningún parámetro incluido en el fragmento, siempre se ejecutará según la configuración y disposición configurada.

**Fragmentos específicos**

Los fragmentos específicos, en cambio, se incorporan de forma distinta en el formulario, Esta incorporación no se realiza a través de referencia, sino que el contenido completo del fragmento se incluye directamente en el formulario en el que se desea usar. Se realiza una copia del contenido pasando a formar parte del formulario. Esto permite al usuario poder modificar cualquier parámetro de los componentes del fragmento, eliminarlos, incluir nuevos, cambiar validaciones o acciones, ...



### Validaciones de Componente

Sobre un componente podrán realizarse diferentes validaciones.

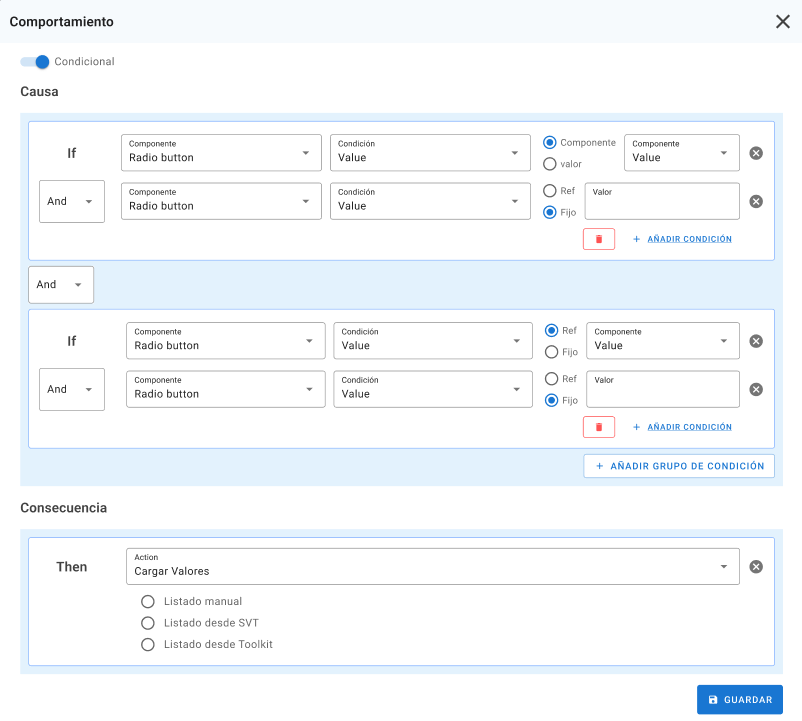
Una validación podrá realizarse directamente sobre el componente o bien realizarse si se cumple una condición previa:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Para configurar las validaciones se aplicarán fórmulas sobre el dato o los datos del componente:

* Fórmulas: validaciones en base a reglas IF/ELSE creadas mediante operadores de comparación (=, <>, >,<, >=, <=) y puertas lógicas (AND, OR) aplicadas contra otros campos ya marcados o contra valores fijos.
* Validaciones en base a la consulta de un servicio de terceros.
* Las validaciones podrán anidarse y dentro de cada anidación podrá haber varias validaciones que siempre estarán asociadas a la misma puerta lógica. (Ejemplo: en una anidación todas las validaciones se asocian mediante AND. No puede combinar AND y OR en la misma anidación).



Cada validación definida llevará un texto asociado para mostrar en caso de error.

* Valor Obligatorio
* Email
* Número de Teléfono
* Número Móvil
* DNI/NIE/CIF/NIF
* IBAN
* Código Postal
* Tamaño mínimo
* Tamaño máximo
* Igual
* Distinto
* Numéricos: =, <>, >=, <=, >, <
* Expresión regular (Pendiente analizar el caso si no está cubierto) Ejemplo: "(^[0-9]{1,6}(\,[0-9]{1,2})?$)"

Sobre los campos fecha, las validaciones podrán ser:

* equalToHoy
* notEqualToHoy
* fechaMenorEqualQueHoy
* fechaMayorEqualQueHoy
* fechaMenorEqual
* fechaMayorEqual
* fechaMayor
* fechaMenor
* fechaMayorQueHoy
* fechaMenorQueHoy
* fechaIgualA
* fechaDistintaA

En relación a la manera de gestionar los avisos de las validaciones de los componentes en la solicitud:

* Se harán después de quitar el foco en el campo
  + **Solo mostrar errores si ya interactuó con el campo:** (touched: true para saber si el campo ha sido clickado antes de mostrar errores).
  + Si el formato es erróneo, mostrará mensaje de error con sugerencia de formato, **mensajes de error claros y específicos.** Ejemplo: “formato: usuario@dominio.com”.
* Si el campo tiene un formato especifico y numérico
  + IBAN: “Formato:  XX00 0000 0000 00 00000000000”
  + Cuenta Bancaria: “Formato:  0000 0000 00 00000000000”
  + Moneda: “Formato: X,XX €”
  + Moneda 2: “Formato: XX”
  + Número seguridad social: “Formato: XXXXXXXX”
  + Fecha: “Formato: “DD/MM/AAAA” en Euskera será “Formato: AAAA/MM/DD”
  + Si el usuario configura que un input sea un entero NO hay que poner como texto de ayuda

**Como son formatos preestablecidos no puede saltárselos, por lo que no necesitan aviso de error por formato.** (Otra cosa es el error por validación de un iban incorrecto)

### Acciones Condicionales

**Acciones condicionales**: conjunto de acciones asociadas a un componente que pueden estar condicionadas al valor aportado en otro componente o que directamente se apliquen sobre el componente sin condición ninguna.

Ejemplo: *ocultar* o *mostrar* un componente en base a una entrada de datos anterior.

Existirá la posibilidad de aplicar acciones condicionales a todas las tipologías de componentes.

La definición de una acción condicional permitirá realizar una acción como resultado de una validación o no.

Ejemplo: si se cumple la condición que se indica, se muestra o se oculta el componente. La condición a cumplir puede hacer referencia a cualquier campo del formulario.

Acción condicionada:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Acción NO condicionada:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Las acciones condicionales definidas son:

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

En base a la imagen anterior, todas las acciones pueden estar condicionadas. Además, la aacción “concatenar” y “Cargar Grupo Repetición” pueden ejecutarse sin necesidad de que se cumpla una condición previa.

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

**Descripción**

Esta acción muestra el elemento si se cumplen las condiciones.

**Solicitud**

* Si se cumplen las condiciones se muestra el componente.

**Componentes**

* Aplica a todos los componentes de captura de datos, informativos y jerárquicos.

**Configuración Compositor**:

* Será una opción seleccionable de la lista de acciones.

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

**Descripción**

Esta acción habilita el elemento si se cumplen las condiciones.

**Solicitud**

* Si se cumplen las condiciones se habilita el campo. Si no se cumplen, el campo aparece como deshabilitado.

**Componentes**

* Aplica a todos los componentes de captura de datos.

**Configuración Compositor:**

* Será una opción seleccionable de la lista de acciones.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Descripción**

El usuario quiere concatenar una serie de valores extraídos de uno o varios componentes del formulario o especificado como textos fijos.

**Solicitud**

* Muestra la concatenación en el componente.

**Componentes**

* Aplica a input, textArea

**Configuración Compositor:**

* Requiere incluir en la pantalla de concatenación la expresión que va realizar el cálculo y la concatenación de valores a partir de los componentes o textos fijos especificados.

Para indicar un valor fijo, éste vendrá entre “” comillas. Por ejemplo:

“Texto fijo”

Para indicar una variable correspondiente a un componente, vendrá entre punto y coma ;;. Por ejemplo:

;nombreSelect-key;

La forma de referenciar los valores de un componente es la siguiente:

**nombrecomponente-value**: hace referencia al valor introducido por el usuario en los campos donde el usuario escribe texto o selecciona fecha, y también hace referencia a la clave del elemento seleccionado en componentes de tipo selección: select, radiobutton y checkbox

**nombre-componente-label**: hace referencia, no al valor introducido ni seleccionado, sino a la etiqueta del componente, siendo esta un valor fijo configurad como “label” en el componente.

* Selector componente:
  + Si es input podrá seleccionar etiqueta o valor
  + Si es radio button, checkbox o select podrá elegir la label o el value del mismo.

Por ejemplo:

Teniendo una lista de valores de un select en el que se incluyen “animales”:

* "[";miRadioMascotaFavorita-value;"]" ---> 1, 2
* "[";miRadioMascotaFavorita-label;"]" --> "Gato", "Perro"
  + Podrá concatenar e intercalar valores fijos

Un ejemplo completo sería:

"["; miRadioMascotaFavorita -value;"-"; miRadioMascotaFavorita -label;"]"

----> Genera algo tipo [1-Gato] o [2-Perro]

A la hora de almacenar esta expresión de concatenación introducida por el funcionario en el compositor, ésta se guardará como una estructura JSON donde:

* FIXED: indicará un valor fijo especificado en la expresión
* KEY: Es el valor del atributo key del componente identificado como viene en value
* LABEL: Es el valor de la etiqueta i18n del componente identificado como viene en value

De esta forma, la expresión anterior:

"["; miRadioMascotaFavorita -value;"-"; miRadioMascotaFavorita -label;"]"

Se almacenaría como:

|  |
| --- |
| [      {          "type": "FIXED",          "value": "[",      },      {          "type": "KEY",          "value": "miRadioMascotaFavorita",      },      {          "type": "FIXED",          "value": "-",      },      {          "type": "LABEL",          "value": "miRadioMascotaFavorita",      },      {          "type": "FIXED",          "value": "]",      }  ] |

**Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

**Descripción**

Esta acción cambia el texto (etiqueta) del elemento si se cumplen las condiciones.

**Solicitud**

* En base al cumplimiento de la condición, se muestra el texto definido.

**Componentes**

* Aplica a input, date, textArea, select, radio, Checkbox

**Configuración Compositor:**

* Cuando se selecciona la acción CambiarTexto se incluirá el nuevo valor que tomará la etiqueta del componente de cumplirse las condiciones.
* Podrá haber varias validaciones con diferentes acciones y, por tanto, podrán aplicar diferentes textos.

Aplicación

El contenido generado por IA puede ser incorrecto.

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

**Descripción**

Esta acción establece un valor por defecto o extraído de otro componente si se cumplen las condiciones.

**Solicitud**

* En base al cumplimiento de la condición, se muestra el valor preestablecido. Una vez mostrado, puede ser modificable.

**Componentes**

* Aplica a input, inputfixed, date, textArea, select, radio, Checkbox.

**Configuración Compositor:**

* Requiere la asociación de un valor fijo o selección de un componente para aplicar el valor constante al componente. Podrá haber varias validaciones con diferentes acciones y, por tanto, podrán aplicar diferentes valores.

**Aplicación

El contenido generado por IA puede ser incorrecto.**

**Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.**

**Descripción**

Esta acción carga valores en el elemento si se cumplen las condiciones: en base a una condición, se cambia el SVT, el listado toolkit o cargar a mano los valores de la select.

**Solicitud**

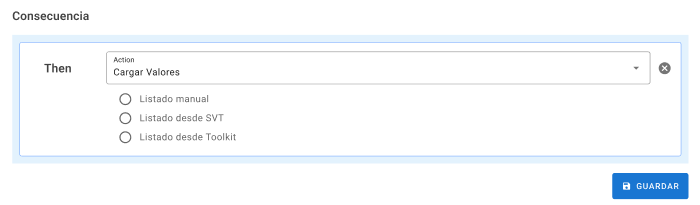
* En base al cumplimiento de la condición, se muestra una lista diferente.

**Componentes**

* select, radio, Checkbox

**Configuración Compositor:**

* Requiere la asociación y definición de llamada a un SVT diferente al establecido en el componente



**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

**Descripción**

Esta acción recupera datos desde un SVT para cargar cada componente de un Grupo. Orientado a Tabla y grupo repetición

**Solicitud**

* En base al cumplimiento de una condición se rellena el Grupo.
* Puede rellenar varias “filas” del grupo de repetición.

**Componentes**

Se aplicaría a nivel de Grupo de Repetición (también bloque y módulo, si es de repetición) o tabla. Y dentro de ellos a los componentes input, inputfixed, date, textArea, hidden, select, radio, Checkbox

**Configuración Compositor:**

* Requiere la asociación y definición de llamada a un SVT a nivel de componente.
* Se mapean los datos del SVT con los datos del GR o Tabla

### Funciones de Cálculo

Las funciones serán reglas de cálculo que se aplicarán a un componente como acciones en base a una condición o directamente sobre un componente sin condiciones.

Por ejemplo, un componente podrá realizar un sumatorio de los valores de otros componentes o bien realizar dicho cálculo en base al cumplimiento de una condición.

Se establecen las siguientes acciones de cálculo:

* **Funciones de Cálculo Matemático**

Acciones a realizar sobre un componente en base al valor de otro componente o componentes:

* Suma, resta, multiplica, divide, porcentaje, máximo, mínimo, media, redondeo.
* También admitirá la definición de fórmulas con anidación de cálculos definidas con paréntesis.
* **Funciones de Cálculo sobre Fechas**

Acciones a realizar sobre un componente teniendo en cuenta los valores de otros componentes fecha:

* Días naturales entre dos fechas, meses entre dos fechas, años entre dos fechas, edad (años entre una fecha y la fecha del día), fecha a la que se suma o restan días, meses o años.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

**Descripción**

Cálculo a realizar sobre un componente teniendo en cuenta los valores de otros componentes: suma, resta, multiplica, divide, porcentaje, redondeo, máximo, mínimo, media.

**Solicitud**

* Se realiza el cálculo y el resultado se quedará en el propio componente como valor.

**Componentes**

* Aplica a input

**Configuración Compositor**

* Requiere la selección de uno o varios componentes o valores fijos.
* Las funciones se mostrarán en la lista de acciones (condicionadas y no condicionadas) con los siguientes nombres:
* **sum**: suma una serie de valores y el total lo asigna a un campo del formulario. Ejemplo sum (name1, name2).
* **dif**: resta una serie de valores y el total lo asigna a un campo del formulario.
* **mult**: multiplica una serie de valores y el total lo asigna a un campo del formulario
* **div**: divide una serie de valores y el total lo asigna a un campo del formulario.
* **Max**: obtiene el máximo de una serie de valores y el resultado lo asigna al componente.
* **Min**: obtiene el mínimo de una serie de valores y el resultado lo asigna al componente.
* **avg**: obtiene el promedio de una serie de valores y el resultado lo asigna al componente.
* **round**: solicitará el número de decimales a aplicar a un número.
* **Percent**: calcula el porcentaje de un componente (Ejemplo: percent (name, numporcentaje))

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

**Descripción**

Cálculo a realizar mediante la aplicación de una fórmula. Permite al usuario establecer una fórmula matemática compleja para realizar el cálculo del valor de un componente.

**Solicitud**

* Se realiza el cálculo y el resultado se quedará en el propio componente como valor
* Se desarrollará una lógica que sepa resolver cálculos con anidación

**Componentes**

* Aplica a input

**Configuración Compositor**:

* La función se mostrará en la lista de acciones (condicionadas y no condicionadas) con el nombre **Combinado**
* Requiere la selección de varios componentes o valores fijos
* Guarda el cálculo mediante “líneas de código” similar a lo existente actualmente, validando que los componentes existen y que las funciones están escritas correctamente. :

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

**Descripción**

Cálculo a realizar sobre un componente teniendo en cuenta los valores de otros componentes: días naturales entre dos fechas, meses entre dos fechas, años entre dos fechas, edad (años entre una fecha y la fecha del día), fecha a la que se suma o restan días, meses o años.

**Solicitud**

* Se realiza el cálculo y el resultado se quedará en el propio componente como valor

**Componentes**

* Se aplicaría a nivel de Grupo de Repetición o tabla.
* Aplica a inputfixed, hidden

**Configuración Compositor**:

* Requiere la selección de varios componentes o valores fijos con la posibilidad de concatenarlos.
* Las funciones se mostrarán en la lista de acciones (condicionadas y no condicionadas) con los siguientes nombres:
* calendarNow: devuelve la fecha y hora actual, según el reloj y zona horaria del ordenador donde se ejecuta el formulario.
* calendarIntervalDays: calcula el intervalo en días naturales entre dos fechas y el resultado lo asigna a un campo del formulario.
* calendarIntervalMonths: calcula el intervalo entre dos fechas, medido en periodos de 30 días naturales, y el resultado lo asigna a un campo del formulario.
* calendarIntervalYears: calcula el intervalo entre dos fechas, medido en periodos de 365 días naturales, y el resultado lo asigna a un campo del formulario.
* calendarAge: calcula la edad actual de una persona a partir de su fecha de nacimiento. El resultado se asigna a un campo del formulario.
* dateModify: calcula la suma o resta de un valor de días, meses o años a una fecha. El resultado se asigna a un campo del formulario.

**FUNCION EEFFC**

Existirá una función de cálculo para validar si un IBAN pertenece a una Entidad Financiera Colaboradora.

La función se llamará comprobarEEFFC (nameIban)

Ejemplo:

Se aplicará a un campo hidden que guardará “0” o “1” dependiendo de si el IBAN pertenece a una Entidad Financiera Colaboradora. La función validará el IBAN contra un fichero de *properties* donde residirán los códigos de las Entidades Financieras Colaboradoras.

Los códigos son: 2095, 0049, 3035, 2100, 0182, 0081, 3008, 2080

### Carga de valores en componentes

La carga de valores en un componente se podrá realizar de manera manual, específica para ese componente, mediante la llamada a un SVT o mediante un listado predeterminado en el toolkit.

Este tipo de cargas aplicará tanto al componente select como a los componentes radiobutton y checkbox y la acción condicional de cargarValores..

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

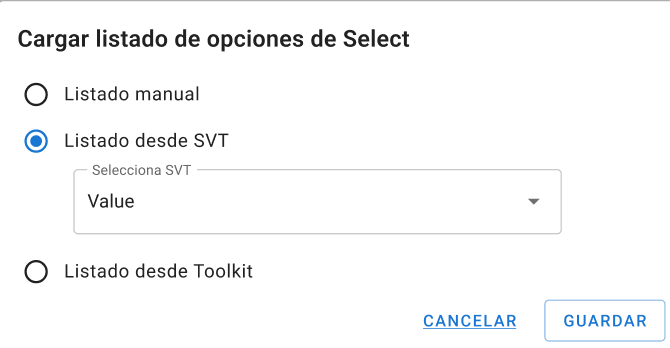
Listado manual

Para realizar la carga manual de la lista de datos a mostrar por el componente, desde el compositor se podrán ir añadiendo valores uno a uno indicando una clave y valor asociado:



Listado desde SVT

Si lo que se desea es obtener la información mediante el acceso a un SVT seleccionado de recuperación de datos, el compositor mostrará un desplegable con la lista de servicios disponibles para su selección



En la composición de un formulario el usuario podrá escoger del listado de servicios disponibles en el catálogo aquel que desee usar para establecer los datos de un componente, por ejemplo, un combo; seguidamente se solicitará que indiquen de dónde obtener los parámetros establecidos para ese servicio, pudiendo ser tanto datos fijos como obtenidos desde otro componente.

El compositor mostrará si la configuración del servicio no concuerda con el uso que se esté haciendo del mismo; por ejemplo, que un servicio que retorna una lista de valores se intente configurar como fuente de datos de un campo de texto.

Igualmente se mostrarán avisos si se intentan configurar parámetros cuyos tipos no concuerdan con los que espera el servicio; por ejemplo, si el servicio espera ser invocado con un valor de tipo booleano que se indique un valor de tipo string al componer el formulario.

Los SVT’s pueden ser utilizados para validar información previamente marcada y/o para recuperar datos que faciliten el rellenado automático de una lista (select, radiobutton,…).

El catálogo mostrará como datos principales de identificación:

• Nombre del servicio

• Descripción

• Datos de entrada

• Datos de salida

Por ejemplo, los datos estructurales relativos a las direcciones postales se obtendrán mediante la llamada a un servicio de NORA para recuperar la información de callejero en el componente de Dirección Postal.

Durante la ejecución en la solicitud, el componente que tenga asociado un SVT (por ejemplo para rellenar los valores de un select o realizar una validación), ejecutará la acción de llamada al SVT para recuperar la lista de valores o realizar la validación que corresponda y recibir una respuesta para mostrar en la solicitud.

Listados desde Toolkit

Cuando se seleccione esta opción, se mostrará una lista con las opciones posibles para que sea seleccionada y guardada como configuración del componente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



getSexo

getSexoRRI

getSiNo

getProvincia

getTipoId

getTipoIdDNI

getTipoIdCIF

getTipoAportacion

Durante la ejecución de la solicitud, cuando se identifique que los datos se recuperarán de un fichero toolkit, el componente recuperará los datos de un fichero de properties específico para cada listado que contendrá los valores en castellano y euskera:

|  |  |
| --- | --- |
| **Opción de listado** | **Valor (código valor)** |
| getSex | Varón (Varón), Mujer (Mujer). |
| getSexRRI | Hombre (HOMBRE), Mujer (MUJER), Otra identidad sexual (OTRA) |
| getSiNoOptions | Si (Si), No (No). |
| getProvinces | ARABA (01), BIZKAIA (48), GIPUZKOA (20) |
| getIdTypes | DNI (1), Pasaporte (4), CIF (2), NIE (3), Otros (5). |
| getIdTypesWithNI | DNI (1), Pasaporte (4), CIF (2), NIE (3), Otros (5) y N/I (6) |
| getIdTypeDNI | sólo DNI’s y NIE’s |
| getIdTypeCIF | sólo CIF’s (esta opción NO está permitida en solicitudes presenciales de la ciudadanía) |

getAddendumType : Devuelve la lista de posibles aportaciones

|  |  |  |
| --- | --- | --- |
| **ACTO\_ID** | **NOMBRE\_ACTO\_ES** | **NOMBRE\_ACTO\_EU** |
| ALEGACION | Alegación | Alegazioa |
| APLAZAMIENTO | Aplazamiento | Atzeratzea |
| APORTACIONDOC | Aportación de documentos | Dokumentuak aurkeztea |
| DESISTIMIENTO | Desistimiento | Atzera egitea |
| RECURSOALZ | Recurso de alzada | Gora Jotzeko Errekurtsoa |
| RECURSOPOT | Recurso potestativo de reposición | Berraztertzeko aukerako errekurtsoa |
| RECURSOEXT | Recurso extraordinario de revisión | Berrikusteko errekurtso berezia |
| RENUNCIA | Renuncia | Uko egitea |
| SUBSANACION | Subsanación | Zuzentzea |
| JUSTIFICACION | Justificación | Zuzenketa |
| ACEPTACION | Aceptación | Onarpena |
| AMPLIACIONINF | Ampliación de información | Informazioa gehitu |
| REQUERIMIENTO | Requerimiento | Errekerimendua |
| SUBSANACIONJUS | Subsanación de justificación | Zuzenketako zuzentzea |

# Catálogo de formularios

## Alcance

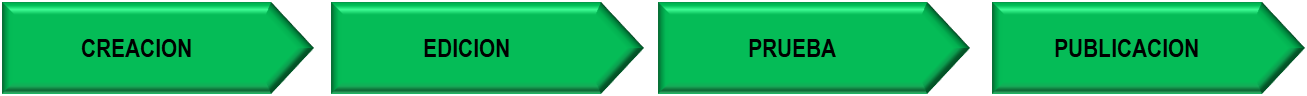
Los formularios residirán en un catálogo donde se guardará la identificación y la definición del formulario con todos sus componentes.

Sobre los formularios se gestionarán los diferentes estados por los que podrá pasar desde su diseño y desarrollo hasta su publicación.

El catálogo de formularios ofrecerá un servicio que permitirá a CCP consultar los formularios en estado “publicado” para facilitar su asociación a un procedimiento.

## Descripción

El catálogo de formularios permitirá mantener los diferentes estados por los que podrá pasar un formulario desde su creación hasta su publicación:



Estado: -------------------EDICION---------------------------------------------------- ------PUBLICADO-------

Desde el Compositor se podrá crear, modificar la información, dar de baja y actualizar los cambios de estado de los formularios utilizando los servicios que ofrece el Catálogo de Formularios.

1. **Creación de un formulario:**

Desde el Compositor de formularios podrá iniciarse la creación de un nuevo formulario.

El sistema ofrecerá la posibilidad de copiar un formulario ya existente para utilizarlo como punto de partida y agilizar la composición de formularios similares.

La creación de un nuevo formulario pedirá el registro de la información identificativa:

* Nombre del formulario
* Descripción del formulario
* Idioma de composición
* Grupo funcional (salud, justicia, educación, etc)
* Estado (edición/publicación)
* Fecha inicio de vigencia (sólo para formularios en estado “publicado”)
* Fecha fin de vigencia (sólo para formularios en estado “publicado”)

El sistema creará un Id de Formulario con la siguiente máscara:

**F-OOOO-NNNNNNNNN-V99.99**

Que corresponde a:

F-Grupo Funcional-Nombre-Versión.Release

Si la creación comienza con un copiado de formulario, se mostrará la información de creación para su modificación. Si no se cambia el Grupo Funcional ni el Nombre del Formulario, se considerará como una nueva versión del mismo formulario y se numerará automáticamente.

El versionado de formularios se compondrá de 2 números de dos cifras cada uno: El primer número indicará la **versión**. El segundo número indicará la **release.** La diferencia entre versión y release reside en el impacto del cambio realizado. (La versión y release se iniciarán numerando por 1).

Será un cambio de versión cuando se añaden nuevos campos en el formulario y, por tanto, variará la información compartida con la aplicación departamental.

Si el cambio no supone nuevos campos y sólo varía la estructura, ordenación o formato del formulario, se tratará como una release, ya que el cambio seguirá manteniendo la compatibilidad del formulario con el procedimiento de negocio actual que lo consume. De esta manera, el cambio de release no afectará al procedimiento en producción, y los cambios de formato se realizarán de manera transparente.

Los estados por los que pasará el formulario son:

**Edición**: desde el momento de su creación durante la fase de diseño (permitirá realizar previsualizaciones parciales)

**Publicado**: el formulario está publicado y puede asociarse a un procedimiento en CCP.

**El Catálogo de formularios guardará la identificación de los formularios y su estado, así como las referencias a los componentes, SVT`s y configuraciones relacionadas con el formulario.**

1. **Modificación de Formulario**

Permitirá la modificación de los datos relacionados con el catálogo de Formularios.

1. **Consulta de formularios**

El compositor de formularios mostrará una lista de todos los formularios en sus diferentes estados mostrando aquellos a los que el usuario tenga permisos. Podrá aplicar filtros y búsquedas.

Un formulario en estado publicado podrá recuperarse para ser copiado o para crear una nueva versión o release.

**Servicio expuesto de consulta de formularios**

Se habilitará un servicio externo (API) de consulta de formularios para ser consumido por terceros (CCP) con el objetivo de ofrecer la lista de formularios en estado “publicado” y vigentes, para que puedan asociarlos al procedimiento de negocio que consideren. CCP sólo escogerá el código de formulario y versión.

Cuando se inicie una solicitud, CCP devolverá el nombre de formulario (versión). Al acceder al catálogo para recuperar el formulario, se devolverá siempre la última reléase vigente.

La idea es poder publicar una nueva release de manera planificada para que el cambio se aplique a partir de la fecha de inicio de vigencia sin afectar a CCP ni al procedimiento.

1. **Eliminación de formulario**

Los formularios que todavía no han pasado a estado “publicado”, podrán borrarse físicamente. No se permitirá eliminar formularios publicados, ni formularios que pertenezcan a un procedimiento.

En el caso de haber sido publicados, la baja será lógica (estado publicado, fecha de fin de vigencia actualizada a la fecha del día) aunque se encuentren asociados a un procedimiento.

Sólo se podrán de de baja formularios de grupos funcionales a los que se pertenezca o, si se pertenece al grupo funcional de PLATEA, se podrá dar de baja cualquier formulario.

Por tanto, el ciclo de vida del formulario será:

* Un formulario publicado tendrá un periodo de vigencia desde <fecha de publicación> hasta <9999-12-31>.
* Si se elimina, pasará a periodo de vigencia <fecha de publicación> hasta <fecha del día de la baja> sin validar si está asociado a un procedimiento o no.
* Los servicios de consulta de formularios para CCP sólo mostrarán los formularios publicados y vigentes a la fecha de consulta.

## Traspaso de formularios

**Pendiente de incluir el nuevo escenario de gestión de versiones, configuración en el CCP y traspso completo.**

Generalmente, la labor de configuración de procedimientos y acciones telemáticas, así como la composición y pruebas de todos los formularios se realiza en el entorno de desarrollo. Una vez generados los formularios, estos son asociados a las distintas acciones telemáticas de los procedimientos relacionados desde el CCP.

Por esta razón se hace necesario disponer de un proceso que traslade la configuración y especificación de estos formularios hasta el entorno de producción que es donde se van a utilizar realmente.

Este proceso de traspaso está vinculado, a su vez, al traspaso de la configuración de los procedimientos y sus acciones telemáticas, por lo que se hace necesario establecer un proceso capaz de trasladar esta información entre entornos de manera que la foto final de configuración de procedimientos-acciones telemáticas-formularios en el entorno de producción sea la adecuada.

En el compositor se gestionan dos tipos de “versiones” de un formulario:

* La propia “versión” del formulario, que indica cambios importantes en el mismo, componentes, funcionalidad entre ellos, validaciones que hacen que, en realidad sean dos formularios distintos
* La “release” de un formulario, que trata cambios menores en el mismo que no afectan a la introducción de datos o relaciones entre campos, por ejemplo, cambio de textos, reorganización, etc…

A la hora de relacionar una acción telemática de un procedimiento con su formulario, esta relación se realiza con una “versión” de formulario, de manera que, cualquier “release” que se haga sobre el mismo, no afecte a la configuración en el CCP, sino que será ejecutada de manera automática a la hora de presentarle el formulario al ciudadano.

Con este escenario, el proceso de traspaso de datos de un procedimiento junto con el formulario/formularios asociados a las acciones telemáticas del mismo implicará lo siguiente:

* Cuando se haga el traspaso de un procedimiento, se traspasarán también la versión y releases existentes para el formulario/s asignado/s a las acciones telemáticas de dicho procedimiento
* En el caso de que la versión o releases traspasadas ya existan en el entorno productivo, no se actualizará ninguna información.
* Si no estuvieran creadas, se darán de alta en el catálogo de formularios del compositor de producción quedando así relacionadas de forma correcta con el procedimiento-acción telemática.
* Si los formularios son compartidos por varios procedimientos:
  + En el caso de que se traspase una versión completa es necesario traspasar todos los procedimientos, aunque el traspaso del formulario solo se realizará de forma efectiva con el primer procedimiento.
  + En el caso de que se traspasen releases, sólo será necesario realizar el traspaso con un procedimiento, ya que la relación procedimiento-acción telemática-formulario no varía.

Para llevar a cabo esta labor de traspaso de la configuración de los formularios, se desarrollarán los servicios correspondientes para:

* Serializar el contenido de un formulario de manera que pueda ser traspasado entre entornos a través de un archivo (vía PIF)
* Deserializar el contenido de ese archivo realizando las actualizaciones necesarias en la base de datos del compositor en el entorno destino.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.

## Servicio de consulta de formularios para CCP

El catálogo de formularios ofrecerá un servicio que permitirá a CCP consultar los formularios en estado “publicado” para facilitar su asociación a un procedimiento y acción telemática.

El CCP guardará el nombre del formulario con el formato F-OOOO-NNNNNNNNN-V99.

Ruta:

<https://ae50.lab.euskadi.eus/ae50a/swagger-ui/index.html#/osbForm/getOsbPublishedForms>

* La invocación al API debe enviar los grupos funcionales que requiere.
* Este servicio filtrará y retornará los formularios de dichos grupos funcionales indicados teniendo en cuenta que:
  + Si el grupo funcional es *PLATEA-Tramitacion* devolverá *TODOS* los formularios de TODOS los grupos funcionales.
  + En caso contrario, devolverá los relacionados a los grupos funcionales recibidos como filtro más los correspondientes al grupo funcional *PLATEA-Tramitacion*.
* Se añadirán filtros para poder consultar por grupo funcional o nombre del formulario.
* Los campos que devolverá el servicio con:
  + grupo funcional
  + nombre
* Está paginado para no devolver todos los almacenados, sino un listado de 20 en 20 (dado que 20 es el valor por defecto).

# Catálogo de SVT’s

## Alcance

De cara a la integración con servicios de validación de terceros la propuesta es desarrollar un catálogo de servicios disponibles junto con un servicio que realice su invocación y valide las respuestas obtenidas.

## Descripción

El catálogo de SVTs gestionará los microservicios o APIS que ofrezcan las aplicaciones departamentales, describiendo la funcionalidad y los campos de entrada/salida del servicio que puedan ser utilizados en la configuración de un formulario.

Este catálogo permitirá al usuario escoger el servicio que quiere utilizar y asociarlo al formulario durante la definición.Posteriormente el formulario pasará a estado “publicado” con la configuración de la llamada al servicio correspondientes ya incorporada.

Los SVT’s pueden ser utilizados para validar información marcada y/o para recuperar datos que faciliten el rellenado automático de los componentes que conformen el formulario.

De cara a facilitar su configuración, existirá la posibilidad de incluirlos en el catálogo de SVTs previo a su exposición en el OSB, configurándolos en estado "Mock". La idea es que pueda crearse el acceso al servicio y su configuración sin interferir en la construcción del formulario., Este estado no permitirá al formulario su progreso a estado “publicado” si tienen algún SVT configurado como “Mock”.

## Funciones principales del catálogo de SVTs

El catálogo de SVTs implementará las funciones principales de creación, modificación y eliminación de servicios (en este último caso, sólo si no está siendo utilizado por algún formulario).

Los datos que se almacenarán por cada servicio son los siguientes:

* **Nombre del SVT:** Indica el nombre con el que se va a identificar el servicio y se referenciará en los formularios.
* **Texto descriptivo:** Texto breve que describe la funcionalidad/objetivo del SVT
* **Grupo funcional:** Grupo funcional al que pertenece. Permite especificar el criterio de uso del propio SVT.
  + Un SVT perteneciente al grupo funcional “Platea Tramitación”
    - podrá ser usado en cualquier formulario, sea del grupo funcional que sea.
  + Un SVT perteneciente a otro grupo funcional
    - Sólo podrá ser usado en los formularios del mismo grupo funcional
* **Endpoint:** URL del servicio (en ambos tipos)
* **Tipo:** SOAP / REST
* **Operación:** Método u operación para la ejecución del servicio (sólo si el tipo=SOAP)
* **TargetNamespace:** (solo si el tipo=SOAP)
* **Método HTTP:** GET/POST (solo si el tipo=REST)
* **Parámetros de entrada:** por cada uno de ellos:
  + Nombre del parámetro
  + Tipo de dato del parámetro:
    - Texto (ID\_TIPO=STRING)
    - Número (ID\_TIPO=NUMBER)
    - Verdadero/falso (ID\_TIPO=BOOLEAN)
    - Fecha/hora (ID\_TIPO=DATETIME)
* **Retorno:**
  + Esquema de retorno: indicará qué tipo de esquema seguirá la respuesta del SVT
    - Validación (ID\_TIPO=VALIDACION): sólo devuelve un valor que, además en este caso, es de tipo boolean.
    - Valor simple (ID\_TIPO=UNIBASIC): sólo devuelve un único valor.
    - Lista simple (ID\_TIPO=MULTIBASIC): Devuelve una lista de textos simples.
    - Lista Clave-Valor (ID\_TIPO=UNIDUPLA): Devuelve un array de elementos clave-valor (Un único valor).
    - Lista Clave-Valores (ID\_TIPO=MULTIDUPLA): Devuelve un array de elementos clave-valor con varios valores por clave (multiidioma, por ejemplo).
    - Tabla de datos (ID\_TIPO=TABLA): Devuelve una lista con registros que pueden tener múltiples campos.
* **Mock:** indicador de si el servicio está o no expuesto y es consumible. Si está en estado “mock” no se invocará, pero habrá que especificar un bloque de respuesta para poder testearlo y configurarlo. Los SVTs que se encuentren en este estado podrán utilizarse en los formularios que se necesiten, pero dichos formularios no podrán pasar a estado “publicado” hasta que todos los SVTs utilizados en ellos dejen de estar en estado “mock”.
* **Bloque de respuesta,** permitirá incluir la respuesta que sustituirá a la llamada real del servicio. Esta respuesta será la que devuelva el servicio SOAP/REST invocado. Mientras el SVT se encuentre en este estado “mock”, ésta será la respuesta que se utilice en todos los sitios donde esté configurado.

Además de las funcionalidades básicas de creación, edición y eliminación de servicios el catálogo proporcionará las siguientes funcionalidades:

* **Prueba / Test de Comprobación de un servicio:** permitirá verificar que el servicio definido es correcto, accesible y que responde satisfactoriamente.
* **“Mockeo” de un servicio:** permitirá indicar que las llamadas al servicio indicado deberán ser simuladas, devolviendo los datos facilitados como ejemplo de respuesta. Se espera que esta funcionalidad permita componer formularios antes de que un servicio concreto esté desarrollado y sea accesible.

## Configuración de traducción del esquema de retorno al esquema utilizado por la aplicación

A la hora de configurar los datos de respuesta de un SVT, será necesario especificar el esquema de retorno del mismo. Este esquema de retorno permitirá al sistema ejecutor de SVTs transformar la respuesta real del servicio (XML, JSON, …) en un esquema JSON estándar interpretable por ToolkitBerria permitiendo el uso de los datos en el componente en el que se encuentre configurado.

Para algunos de los tipos de esquema de retorno es necesario especificar, además, una serie de datos:

* **Validación (ID\_TIPO=VALIDACION):** No será necesaria ninguna información extra.
* **Valor simple (ID\_TIPO=UNIBASIC):** Será necesario identificar el tipo del valor retornado (STRING, DATETIME, NUMBER, BOOLEAN).
* **Lista simple (ID\_TIPO=MULTIBASIC):** Será necesario identificar el tipo del valor retornado (STRING, DATETIME, NUMBER, BOOLEAN).
* **Lista Clave-Valor (ID\_TIPO=UNIDUPLA):** Será necesario identificar el tipo del valor del elemento que actúa como “clave” (STRING, NUMBER).
* **Lista Clave-Valores (ID\_TIPO=MULTIDUPLA):** Será necesario identificar el tipo del valor del elemento que actúa como “clave” (STRING, NUMBER).
* **Tabla de datos (ID\_TIPO=TABLA):** Deberá especificarse tantas columnas como tenga cada registro y la tipología de cada una de ellas (STRING, DATETIME, NUMBER, BOOLEAN)..

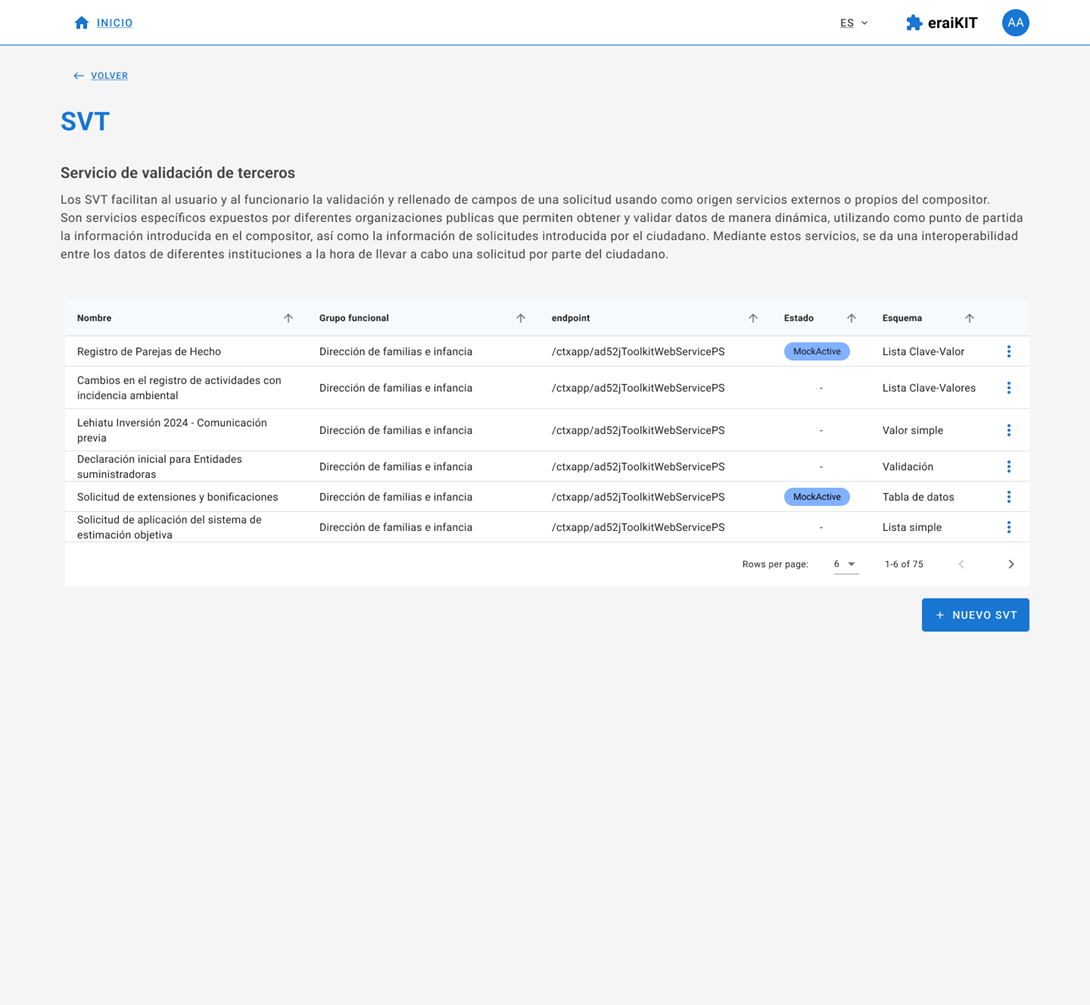
## Listado de SVTs

El acceso al Catálogo de SVTs mostrará la lista completa de SVTs existentes en base al grupo funcional del usuario conectado:

* Si el usuario pertenece al grupo funcional Platea Tramitación, mostrará todos los SVTS incluidos en el catálogo, pertenezcan al grupo funcional que pertenezcan.
* Si el usuario pertenece a otro grupo funcional, se mostrarán los SVTs que pertenezcan a su grupo funcional más los pertenecientes al grupo funcional Platea Tramitación.

En ambos casos, sólo podrá editar y modificar aquellos que pertenezcan a su propio grupo funcional.

La tabla con la información de todos los SVTs se muestra en la siguiente imagen:



La tabla mostrará:

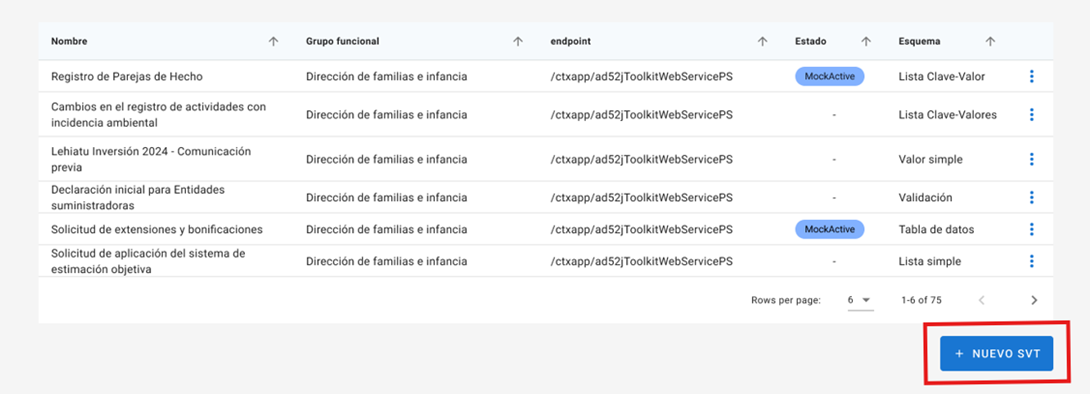
* El nombre del SVT
* El grupo funcional al que pertenece
* Endpoint del SVT
* Si se encuentran o no en estado “mock”
* El esquema de retorno
* Menú contextual donde se ubicarán las siguientes opciones:
  + Ejecutar, que permitirá ejecutar una prueba/test de ejecución del servicio
  + Editar, sólo aparecerá en aquellos sobre los que se tenga permiso de edición en base al grupo funcional del usuario y del SVT
  + Eliminar, sólo aparecerá disponible si:
    - El SVT pertenece al mismo grupo funcional del usuario conectado
    - El SVT está en estado borrador o está publicado, pero no está en uso en ningún formulario.

## Consulta de un SVT

La consulta de los datos de un SVT se podrá realizar desde la lista de servicios de la pantalla principal del catálogo de SVTs. La consulta dará acceso a la pantalla de configuración de un servicio en la que se mostrarán todos los parámetros configurados para el mismo.

## Creación de un SVT

Desde la página principal del catálogo de SVTs se podrá acceder a dar de alta un nuevo SVT



Este botón dará acceso al formulario de alta de un nuevo SVT y mostrará el formulario para la inclusión de todos los datos necesarios.



El alta de un SVT debería poder completarse cuando el servicio ya exista y esté operativo. En el caso de que se desee incluir la configuración del mismo mientras no exista o no esté publicado para su ejecución en el aplicativo origen, se deberá activar el estado “Mock”. La activación de este estado implicará la obligatoriedad de incluir una respuesta que sustituirá la ejecución real del servicio mientras esté en este estado.

El nombre del SVT conformará parte del identificador único del mismo, que tendrá la siguiente nomenclatura:

**SVT-OOOOOOOOOOOOOOOO-NNNNNNNNN** **SVT-Grupo Funcional-Nombre**

Donde:

* **OOOOOOOOOOOOOOOO:** corresponderá al grupo funcional al que pertenece.
* **NNNNNNNNN:** corresponderá al nombre indicado en el alta del mismo.

## Modificación de un SVT

Desde el listado de SVT’s la persona usuaria podrá seleccionar un SVT para su edición. Existen tres posibles escenarios en los que se puede realizar una modificación:

1. SVT que no está siendo utilizado por ningún formulario publicado. Si el SVT es de nueva creación, podrán actualizarse todos los campos.
2. Si el SVT está siendo utilizado por otros formularios publicados, podrá cambiarse sólo el campo descripción.
3. Si el SVT está siendo utilizado por otros formularios publicados y se desea modificar los campos de entrada/salida, el endpoint, o cualquier otra característica que afecte a la ejecución, la actualización deberá realizarse mediante la creación de un nuevo SVT, de manera que los formularios puedan actualizar la llamada al nuevo SVT en los casos en los que se considere necesario e incorporar la nueva funcionalidad por necesidades de uso en el formulario.

Siempre que un SVT cambie de estructura, y para evitar tener que cambiar todos los formularios que lo están usando, se creará un nuevo SVT. Los formularios con el SVT antiguo podrán seguir funcionando y “migrarán” al nuevo SVT cuando consideren oportuno y sólo si fuera necesario

En ningún caso un SVT en uso podrá modificarse para pasarlo a estado “mock”.

## Eliminación de un SVT

Por regla general, un SVT que está en uso por un formulario publicado no podrá eliminarse. En aquellos SVT que sí se pueden eliminar, aparecerá la opción en el menú contextual de la lista de SVT.

## Prueba/Test de un SVT

El catálogo de SVTs gestionará losmicroservicios o APIS que ofrezcan las aplicaciones departamentales, describiendo la funcionalidad y los campos de entrada/salida del servicio que puedan ser utilizados en la configuración de un formulario.

Este catálogo permitirá al usuario escoger el servicio que quiere utilizar y asociarlo al formulario durante la definición.

Posteriormente el formulario pasará a producción con la configuración de la llamada al servicio correspondientes ya incorporada.

Los SVT’s pueden ser utilizados para **validar** información marcada y/o para **recuperar datos** que faciliten el rellenado automático de los componentes que conformen el formulario,De cara a facilitar su configuración, existirá la posibilidad de incluirlos en el catálogo de SVTs previo a su exposición en el OSB, configurándolso en estado "Mock". La idea es que pueda crearse el acceso al servicio y su configuración en la construcción del formulario., Este estado no permitirá al formulairo su progreso a estado “publicado” si no tiene todos los SVT's que utiliza publicados.

**Funciones principales del catálogo de servicios**

El catálogo de SVTs implementará las funciones principales de creación, modificación y eliminación de servicios (en este último caso, sólo si no está siendo utilizado por algún formulario).

Los datos que se almacenarán por cada servicio son los siguientes:

\*Nombre del SVT: Indica el nombre con el que se va a identificar el servicio y se referenciará en los formularios. (\*)

\*Texto descriptivo: Texto breve que describe la funcionalidad/objetivo del SVT

\*Grupo funcional: Grupo funcional al que pertenece. Permite especificar el criterio de uso del propio SVT.

Un SVT perteneciente al grupo funcional “Platea Tramitación”

podrá ser usado en cualquier formulario, sea del grupo funcional que sea.

Un SVT perteneciente a otro grupo funcional

Sólo podrá ser usado en los formularios del mismo grupo funcional

\*Endpoint: URL del servicio (en ambos tipos)

\*Tipo: SOAP / REST

\*Operación: Método u operación para la ejecución del servicio (SOAP)

\*TargetNamespace: (SOAP)

\*Método HTTP: GET/POST (REST)

* 1. Identificador
  2. Grupo funcional (salud, justicia, educación, PLATEA-TRAMITACIÓN, etc)
  3. Descripción del servicio
  4. Endpoint
  5. Tipo de servicio (SOAP, REST)
  6. Parámetros de invocación (nombre y tipo).
  7. Tipo de respuesta (lista, valor único, booleano).
  8. Respuesta de ejemplo (ayudará al mapeo de campos a rellenar)
  9. Formato de respuesta: nombre de los campos devueltos por el SVT.
  10. Mensaje de error para casos de error en la llamada al servicio.
  11. Valor “por defecto” para casos de error en la llamada al servicio.
  12. Estado Publicado/Mock.
  13. Cacheable S/N.

Además de las funcionalidades básicas de creación, edición y eliminación de servicios el catálogo tendrá las siguientes características:

* + Prueba / Test de un servicio: permitirá verificar que el servicio se ha configurado de forma correcta, es accesible y que responde satisfactoriamente.
  + “Mockeo” de un servicio: permitirá indicar que las llamadas al servicio indicado deberán ser simuladas, devolviendo los datos facilitados como ejemplo de respuesta. Se espera que esta funcionalidad permita componer formularios antes de que un servicio concreto esté desarrollado y sea accesible.

Al igual que otros componentes de los formularios, existirán servicios definidos a nivel de plataforma que todos podrán usar (Ej: PLATEA-tramitacion, NORA), y otros definidos a nivel de grupo de usuarios que sólo podrán usar los formularios asociados a ese mismo grupo.

El catálogo de servicios requerirá un mantenimiento que permita asegurar que todos los servicios que se incorporen al catálogo están vigentes y en estado productivos (NO en estado MOCK).

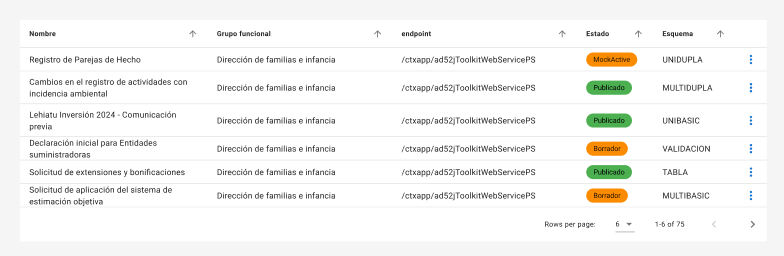
Los servicios consumidos por formularios no podrán eliminarse mientras exista un formulario que los utilice.

La actualización de un servicio deberá realizarse mediante versionado, de manera que los formularios puedan corregirse y poder actualizar la llamada a la nueva versión en los casos que se considere necesario incorporar la nueva funcionalidad por necesidades de uso en el formulario.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Los servicios podrán darse de alta en estado mock: podrá crearse el SVT y asociarse a un formulario para facilitar la fase de diseño de formulario en el compositor (devolviendo un código de respuesta prefijado). Para que el formulario progrese a producción, el SVT deberá estar ya creado.



### Creación de un nuevo SVT

El usuario accede al catálogo de SVT’s y marcará la opción de alta de SVT.

En la siguiente pantalla podrá cumplimentar toda la información requerida para la definición del SVT: su nombre, el grupo funcional al que pertenece, los datos del servicio web o endpoint que se ejecutará, la lista de parámetros para la invocación, etc.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

La configuración de un SVT en el compositor debería poder completarse cuando el servicio ya exista y esté operativo. En el caso de que se desee incluir la configuración del mismo mientras no exista relamente, se deberá activar el estado “Mock”.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

El identificador de los SVT’s llevará la siguiente nomenclatura:

**SVT-OOOO-NNNNNNNNN**

SVT-Grupo Funcional-Nombre

Creará un nuevo SVT. El campo Endpoint sólo podrá completarse cuando el servicio exista y esté operativo. Mientras no exista el endpoint, el SVT estará en estado “Mock”.

### Validación de un SVT

Proceso de validación de existencia de los servicios creados.

Se comprobará que un servicio mediante una llamada al endpoint (ping): permitirá verificar que el servicio definido es correcto, accesible y que responde satisfactoriamente.

* Si el servicio no responde, pasará a estado “Mock”
* Si el servicio responde, pasará a estado “Publicado”

### Modificación de un SVT

Desde el listado de SVT’s vigentes, la persona usuaria podrá filtrar los SVT’s por Grupo Funcional (salud, justicia, educación, etc) de aquellos a los que tenga acceso.

La persona usuaria podrá seleccionar un SVT para su edición.

Existen tres posibles escenarios en los que se puede realizar una modificación:

1. SVT que no está siendo utilizado por ningún formulario en producción. Si el **SVT es de nueva creación,** podrán actualizarse todos los campos.
2. Si el SVT está siendo utilizado por otros formularios en producción, podrá cambiarse sólo el campo *descripción*.
3. Si el SVT está siendo utilizado por otros formularios en producción y se desea modificar los campos de entrada/salida, el endpoint, o cualquier otra característica que afecte a la ejecución, la actualización deberá realizarse mediante versionado, de manera que los formularios puedan corregirse y actualizar la llamada a la nueva versión en los casos en los que se considere necesario e incorporar la nueva funcionalidad por necesidades de uso en el formulario.

Si un SVT cambia de estructura, para evitar tener que cambiar todos los formularios que lo están usando, se crea un nuevo SVT (nueva versión) para el formulario que se esté componiendo. Los formularios con el SVT antiguo podrán seguir funcionando y “migrarán” a la nueva versión del SVT cuando consideren oportuno y si necesitan incorporar los datos de la nueva versión.

En el caso de los SVT’s ya publicados el sistema no permitirá pasar a estado “Mock”.

### Consulta de SVT’s

Desde el compositor de formularios, en la pantalla de configuración de componentes, podrá invocarse un servicio del catálogo de SVT’s que mostrará la lista completa de SVT’s vigentes. El usuario podrá filtrar los SVT’s por Grupo Funcional (salud, justicia, educación, etc).

El usuario podrá seleccionar y relacionar el SVT en la configuración del componente.

### Baja de un SVT

Los SVT’s consumidos por formularios no podrán eliminarse (borrado físico) mientras exista un formulario que los utilice.

Si el SVT es consumido por un formulario que estuvo publicado, pero ya no está vigente, el SVT podrá marcarse como no vigente (baja lógica) y no aparecerán en las consultas al catálogo realizadas durante la composición de formularios.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

### Ayuda a la utilización de un SVT en la composición

En la composición de un formulario el usuario podrá escoger del listado de servicios disponibles en el catálogo aquel que desee usar para establecer los datos de un componente, por ejemplo, un combo; seguidamente se solicitará que indiquen de dónde obtener los parámetros establecidos para ese servicio, pudiendo ser tanto datos fijos como obtenidos desde otro componente.

El compositor mostrará si la configuración del servicio no concuerda con el uso que se esté haciendo del mismo; por ejemplo, que un servicio que retorna una lista de valores se intente configurar como fuente de datos de un campo de texto.

Igualmente se mostrarán avisos si se intentan configurar parámetros cuyos tipos no concuerden con los que espera el servicio; por ejemplo, si el servicio espera ser invocado con un valor de tipo booleano que se indique un valor de tipo string al componer el formulario.

### Definición de esquemas para las respuestas de SVTs

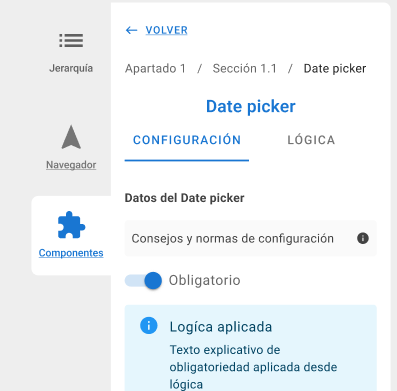
Los servicios que estén enfocados a su uso como SVTs en el Toolkit deberán responder siguiendo los esquemas definidos y según el tipo de respuesta definido en el catálogo.

# Ayudas en la composición de formularios

Dada la novedad del Compositor como herramienta para la creación, composición y configuración de los formularios, se hace necesaria la incorporación de ayudas contextuales, guías interactivas y demás elementos de ayuda que faciliten, al menos de manera inicial, el uso y entendimiento de la herramienta por parte de los usuarios que vayan a trabajar con ella.

## Consejos y Normas de Configuración

Se ha incorporado una sección de Consejos y Normas de Configuración dentro del apartado de configuración de cada componente. En este espacio se presentará una guía de uso recomendada, con indicaciones prácticas que facilitarán la correcta configuración y el aprovechamiento óptimo de cada elemento.



# Configuración del CCP y Traspaso de formularios

## Configuración en el CCP

Una vez ”publicado” el formulario podrá ser seleccionado en el CCP para configurar la acción telemática de un procedimiento. Esta relación se almacenará en el CCP guardando el identificador del formulario junto con la versión del formulario seleccionado para la acción telemática corespondiente. Es decir:

Procedimiento Acción Telemática - Formulario

Para poder realizar esta configuración, EraiKIT proporcionará un endpoint que permitirá recuperar todos los formularios en estado “publicado” seleccionables para configurar la acción telemática del procedimiento.

**TODO**-Especificar Endpoint

## Proceso de traspaso de formularios entre entornos

EraiKIT estará disponible únicamente en el entorno de desarrollo, donde se llevará a cabo la creación de formularios y la configuración de procedimientos, incluyendo la relación entre ambos. Esta limitación implica la necesidad de establecer un proceso que permita trasladar los formularios a los entornos de pruebas y producción para que puedan ser utilizados.

Actualmente, el Catálogo de Configuración de Procedimientos (CCP) es el responsable de iniciar este proceso de traspaso, trasladando tanto la configuración del procedimiento como los formularios en formato XML a los entornos de pruebas y producción.

La propuesta consiste en mantener este esquema, incorporando los ajustes necesarios para que EraiKIT pueda gestionar el traslado del contenido y la configuración de los formularios y SVTs entre los diferentes entornos. Todo ello se realizará siempre bajo la dirección del CCP, que seguirá actuando como desencadenante del proceso.

De esta manera:

* El usuario del CCP seleccionará el/los procedimientos a traspasar a pruebas/producción
* El CCP comunicará a EraiKIT qué formularios se deben traspasar a los siguientes entornos
* EraiKIT “empaqueta” la información de los formularios y los SVTs relacionados y los traspasa al siguiente entorno
* EraiKIT confirma el éxito o no de la operación al CCP
* El CCP procede con el traspaso de la configuración de los procedimineot sne caso de éxito en la operación de traspaso de formularios.

Para que esta secuencia de pasos se pueda completar y el CCP pueda dirigir la operación es necesario que EraiKIT exponga los endpoints necesarios:

* Un endpoint que permita recibir la orden de traspaso con la lista de formularios a traspasar (OfficialFormId)
* Un endpoint que permite al CCP consultar si el formulario en cuestión se ha traspasado satisfactoriamente o no al siguiente entorno, de manera que pueda continuar con su proceso.

**TODO**-Especificar Endpoint