**Universidad Nacional Mayor de San Marcos  
Facultad de Ingeniería de Sistemas e Informática  
E.P. de Ingeniería de Software**

****

**SS-DAS: Documento de Arquitectura del Software**

**Integrantes**

Alva Sáenz, Rodrigo José 22200232  
Del Carpio Martinez, Damaris Marian 22200082  
Diaz Ingol, Jesus Stevan 22200083  
Maylle Colaca, Luis Fernando 21200221  
Monzon Argüelles, Oscar Sebastian 22200094  
Rojas Castañeda, Ruth Camila 22200239  
Rojas Rojas, Sebastian Alberto 22200261  
Zegarra Medina, Jose Antonio 22200102

**Documento de Arquitectura del Software**Tabla de contenidos

[**Introducción 3**](#_aahdooayerup)

[Propósito 3](#_87bnbenpstdk)

[Scope 3](#_wetnvqcxe1vg)

[**Representación 3**](#_u38f0nh21vct)

[**Metas de proyecto y limitaciones 3**](#_pt3zv6jp31ag)

[**Diagrama de casos de uso 3**](#_nx87v2en9s6i)

[Recibir sugerencias de escritura 5](#_aaqsznfafw5f)

[Recibir correcciones de escritura 5](#_wem7hehrmkv5)

[Utilizar funciones cíclicas 5](#_hltmgwwjqjq2)

[Interactuar con feed RSS web 5](#_8n3lziofceev)

[**Diagrama de subsistemas 6**](#_rjev3mleubqy)

**Historial de revisiones**

| **Versión** | **Autores** | **Descripción de la versión** | **Fecha** |
| --- | --- | --- | --- |
| v1.0a | A.S.R.J. / M.A.O.S. / D.M.D.M. | Diseño inicial de la Arquitectura de Software | 02/05/2024 |
| v1.1. | A.S.R.J. / M.A.O.S. / D.M.D.M. / Z.M.J.A. | Diseño final de la Arquitectura de Software | 27/05/2024 |

**Documento de Arquitectura del Software**

# Introducción

## Propósito

Este documento provee una descripción de la arquitectura de software del proyecto a desarrollar. El documento describe las decisiones principales que se tomaron respecto de la arquitectura, así como descripciones de diseños, procesos, etc.

## Scope

El documento describe las decisiones arquitecturales y de diseño arquitectural de TCP, una toolkit de herramientas para mejorar la productividad y calidad de trabajo y comunicación de los trabajadores de la empresa “Kalhec”.

# Representación

Este documento divide a la arquitectura del sistema en distintos grupos relacionados con diagramas específicos: el diagrama de casos de uso, que une a los actores definidos en los requisitos funcionales junto con la cadena de acciones que pueden desencadenar, el diagrama de subsistemas y capas del sistema, los diagramas de procesos y el diagrama de deployment.

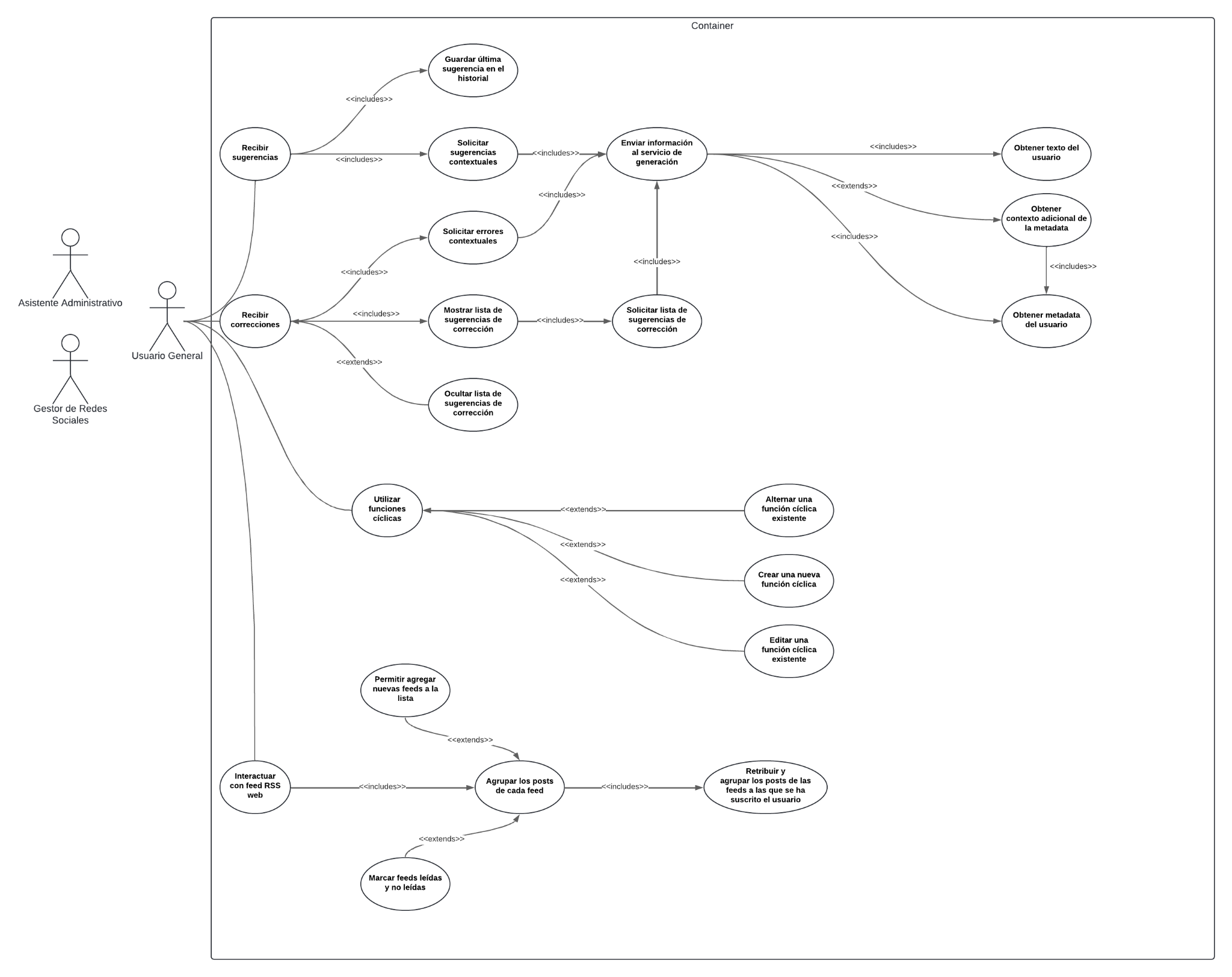
# Metas de proyecto y limitaciones

La funcionalidad base de “TPC” debe estar contenida en una extensión de navegador, de ese modo, la capa de interactividad con el usuario también será parte de ella. Internamente, además, se debe mantener la información contextual de cada usuario, así como información extra de otros módulos, asegurada en una base de datos, añadiendo una capa de seguridad criptográfica a las contraseñas de usuario.

# Diagrama de casos de uso

Para elaborar el diagrama de casos de uso, se tomó en cuenta al usuario general como el único usuario final, dado que las funcionalidades que se acabaron quedando como parte del sistema eran de carácter universal, es decir, cualquier tipo de trabajador puede hacer uso de estas. Existen 4 casos de uso primordiales, que comprenden los módulos principales del sistema y que se dividen en tareas más pequeñas:

1. Recibir sugerencias de escritura
2. Recibir correcciones de escritura
3. Utilizar funciones cíclicas
4. Interactuar con feed RSS web

[](https://lucid.app/lucidchart/ed05f5b7-a374-4b1e-afdc-4d4c7f8b1972/edit?viewport_loc=312%2C700%2C3284%2C1552%2C0_0&invitationId=inv_137ea962-a13e-48cd-bbc1-734696f7aae1)

## Recibir sugerencias de escritura

El sistema va a observar en todos los bloques de entrada de texto de las páginas web en las que se use el sistema. De ese modo, cuando el usuario empiece a escribir, el sistema va a generar posibles palabras que podrían proseguir del texto original, similar a las autosugerencias del bloque de texto de gmail, pero con la diferencia de que este servicio de sugerencias está basado en el contexto tanto del mensaje como del remitente. Por ejemplo, este caso de uso engloba al sistema cuando éste sugiere palabras menos amenazantes al redactar un mensaje delicado por correo desde un cargo alto de una empresa.

## Recibir correcciones de escritura

De forma similar a como funciona el caso de uso para las sugerencias, el sistema observará todas las entradas de texto y, en lugar de mostrar sugerencias de escritura, mostrará sugerencias de corrección, es decir, el sistema mostrará una lista de sugerencias de cambios en el texto, como pequeñas indicaciones. Por ejemplo, dentro del mismo caso anterior, al terminar de redactar el correo, el usuario desea poder ver en concreto qué debería cambiar del mismo.

## Utilizar funciones cíclicas

Una función cíclica es una tarea determinada que se ejecuta cada cierto intervalo de tiempo, emitiendo señales o notificaciones a su término. El sistema tendrá algunos presets de tareas determinadas como el de pomodoro, que se ejecuta como una task de “descanso” de una duración corta tras un número determinado de intervalos de “trabajo” de duración larga. El usuario debe poder agregar, editar o eliminar las tareas del sistema.

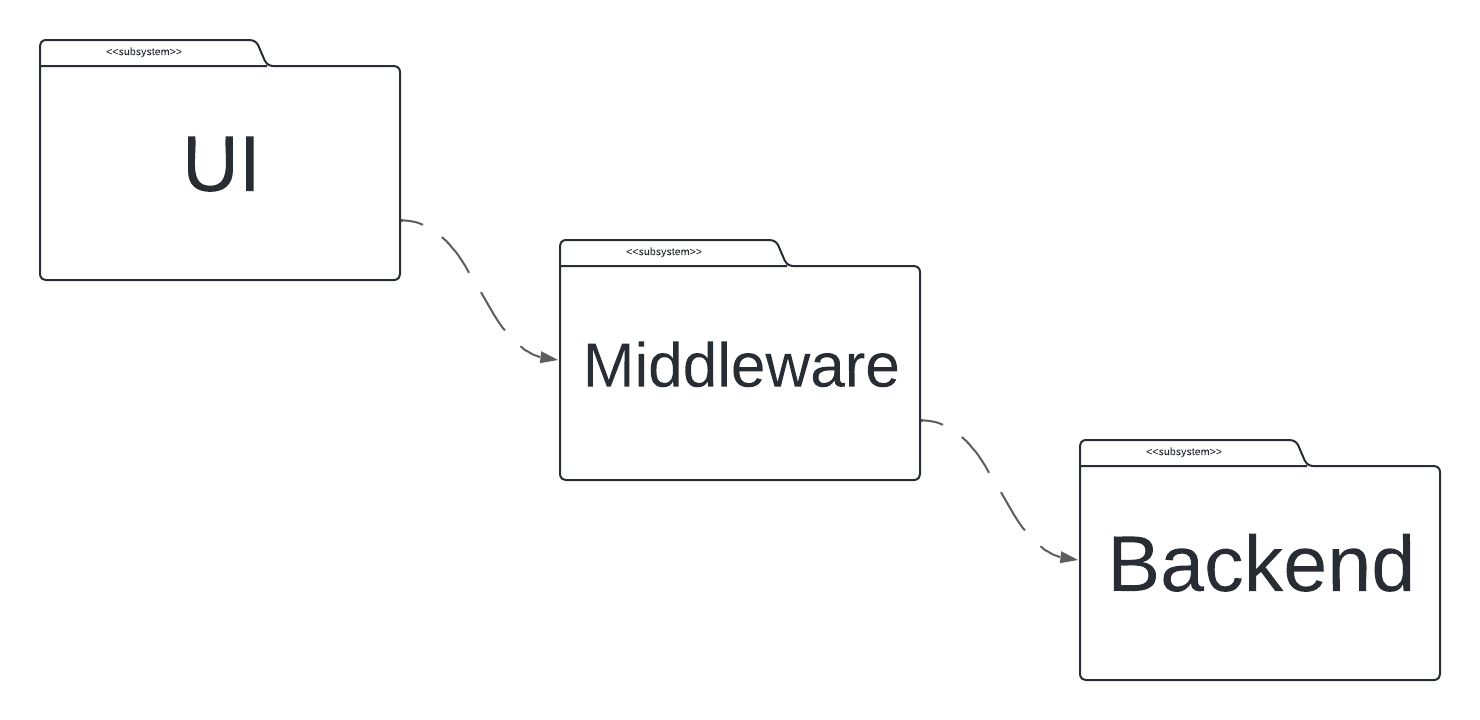
## Interactuar con feed RSS web

El protocolo “RSS” permite seguir canales de publicaciones como los de noticias o periódicos de forma más minimalista y plana, estas publicaciones se sirven como archivos de formato “xml” que son interpretados por el sistema, agrupados, renderizados y mostrados al usuario. El usuario debe poder suscribirse a varias “feeds” o fuentes de publicaciones, así como poder marcas publicaciones que ya leyó como “leídas”, de forma similar a los correos electrónicos.

# 

# Diagrama de subsistemas

Se han definido inicialmente 3 subsistemas para el proyecto “TPC”, el subsistema de “UI”, que contiene a los componentes de la UI de la extensión de navegador, el subsistema de “Middleware” que contiene al servicio intermediario para los datos de la base de datos, y el subsistema de “Backend”, que engloba a la base de datos principal.

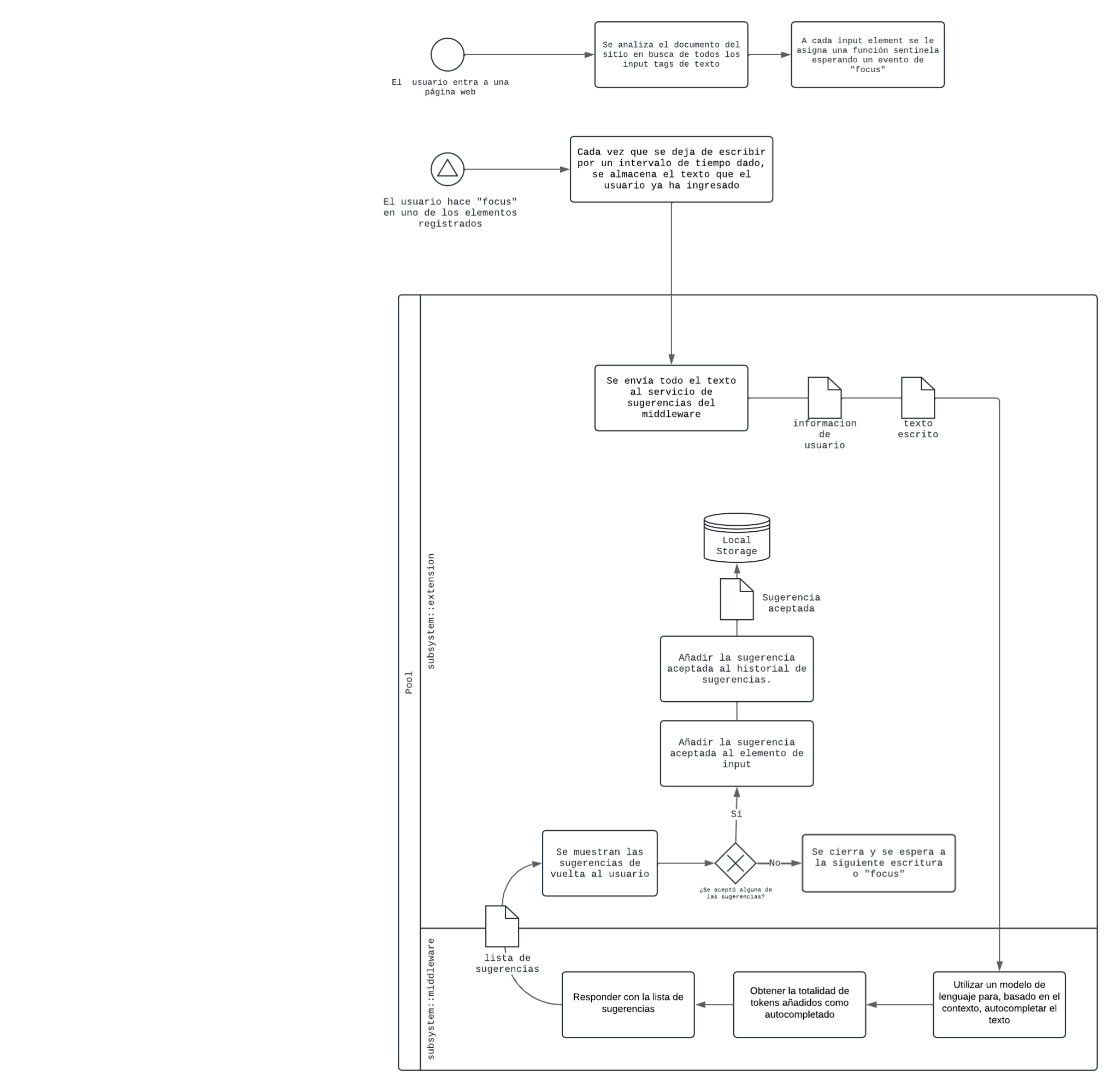


Existen 2 razones principales por las que consideramos necesario el uso de un servicio intermedio o “Middleware”. La primera razón es por seguridad, ya que se necesita de información sensible como “API keys” para acceder a servicios determinados. Además, se suele exponer el backend (la base de datos) a 1 sola conexión por seguridad (el middleware). La segunda razón es porque generalmente el backend no tiene forma de comunicarse directamente con la “UI” o el “frontend”.

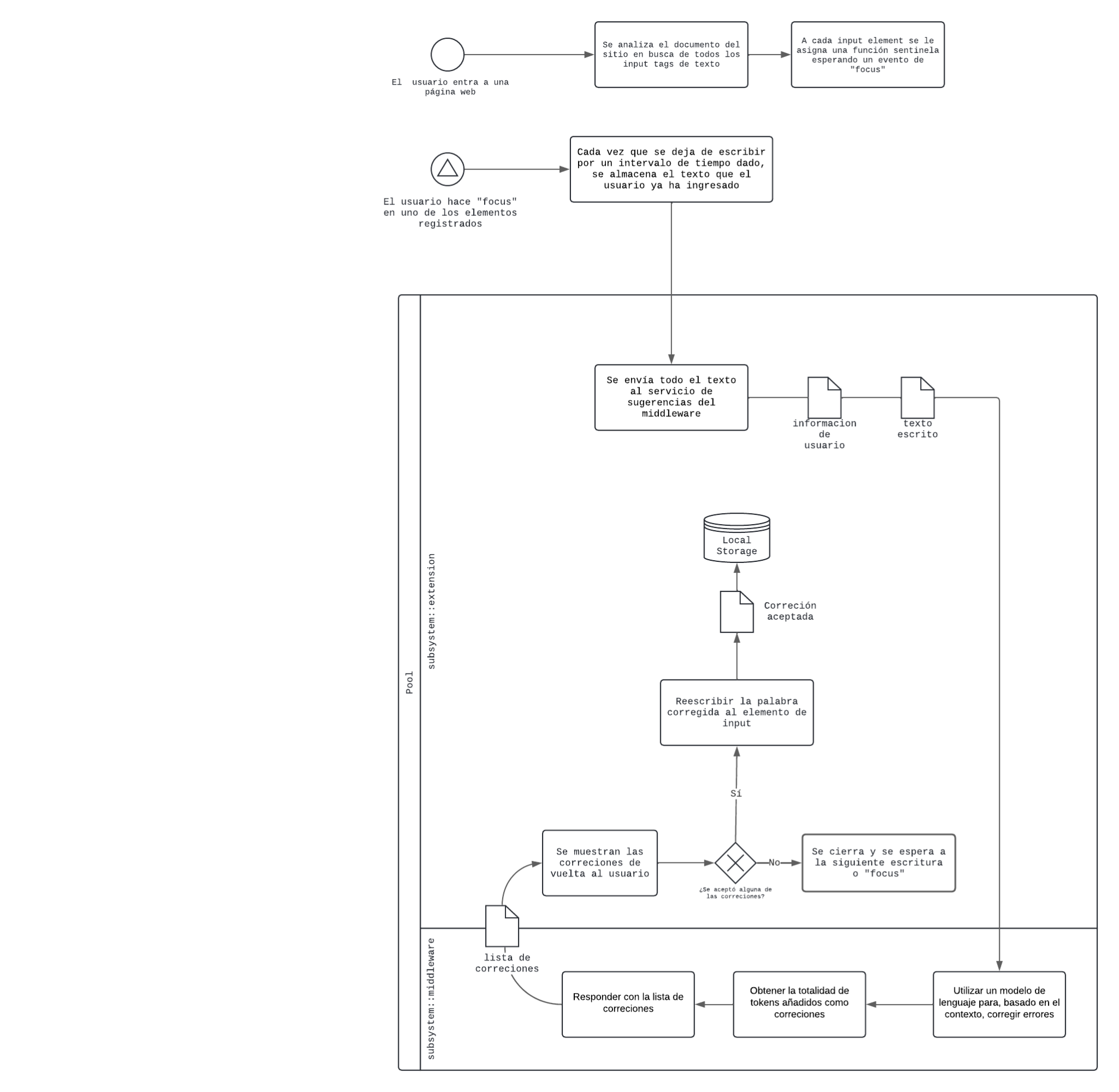
# 

# Diagramas de procesos

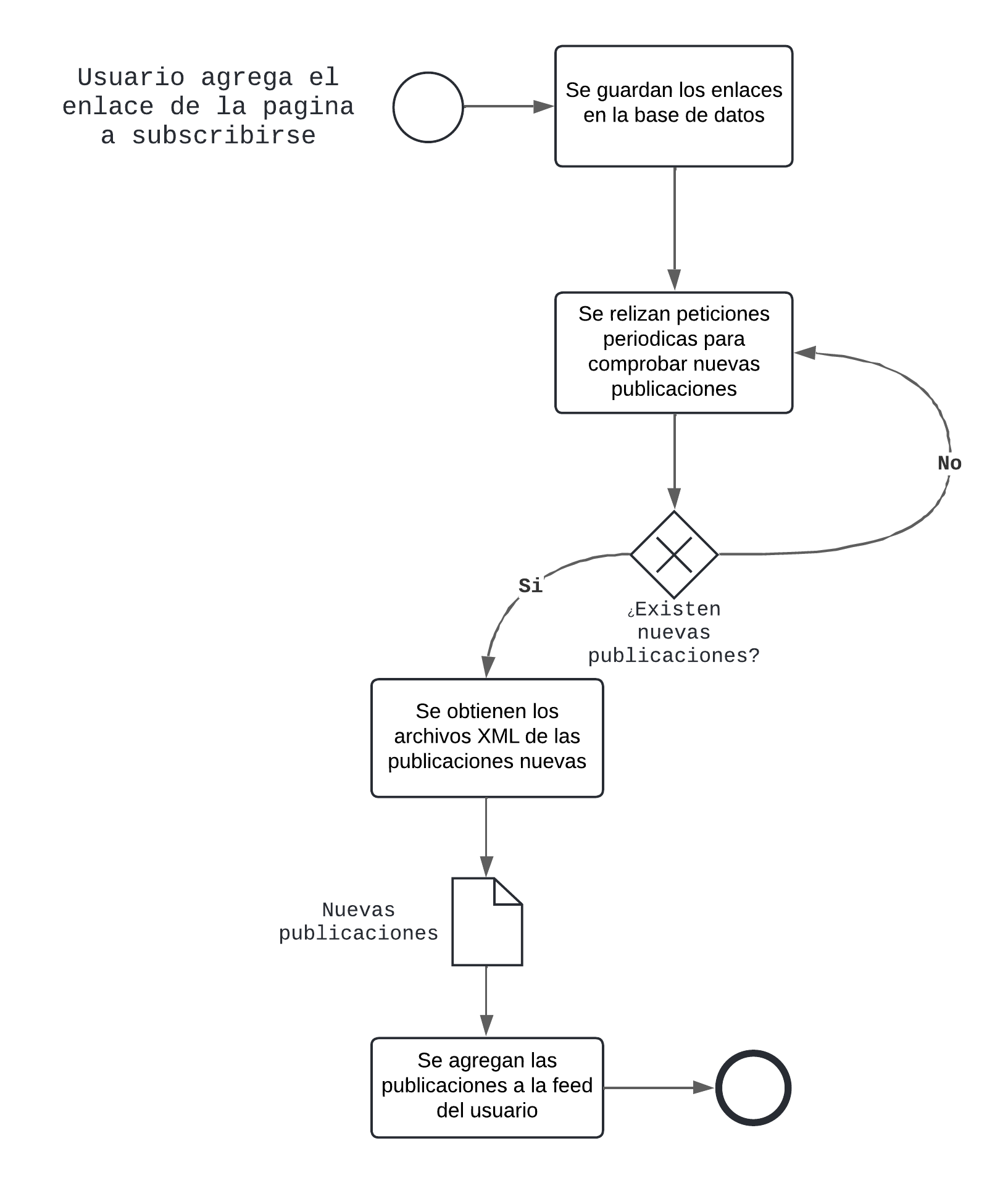
## Proceso de generación de sugerencias de texto



## Proceso de generación de sugerencias de corrección de texto

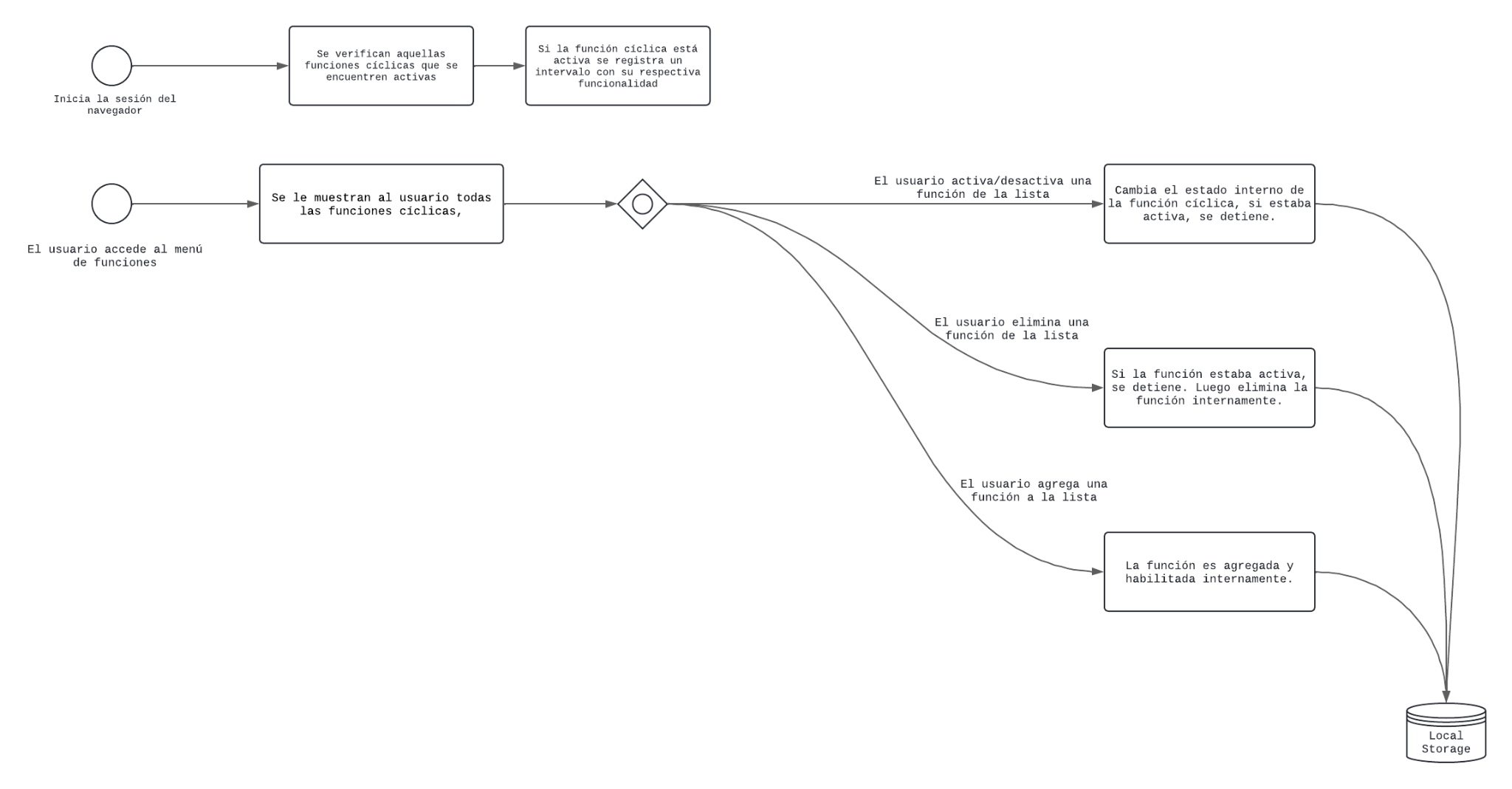


## Renderizado de web feeds RSS



## 

## Generación de funciones cíclicas



# Diagrama de deployment

En este diagrama se detallan tanto los artefactos que vamos a generar como parte del despliegue como la forma en la que estos interactúan. A través de peticiones HTTP, se comunicarán los servicios como detallados en diagramas anteriores, se eligió, además, PocketBase como Backend as a Service (BaaS) por la practicidad del mismo, que usa SQLite por detrás. Se organiza través de Colecciones (Collections) que representarían las tablas de la base de datos SQLite, y estas contendrían los campos (Fields).

