

# Documentar nuestras APIs

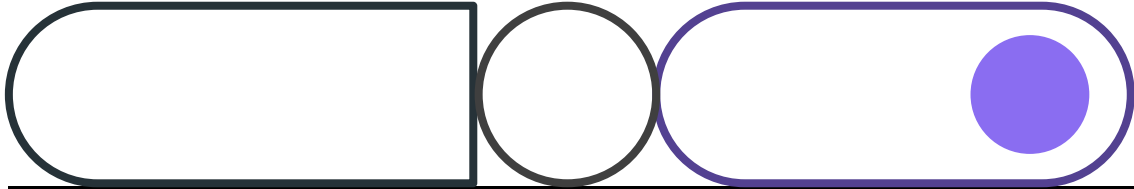
# Índice

- 01**    [¿Para qué sirve?](#)
- 02**    [Herramientas de documentación](#)
- 03**    [Buenas prácticas](#)



01

¿Para qué sirve?



Documentar correctamente una API permite a los desarrolladores (internos o externos a nuestro equipo) un acercamiento accesible al proceso de aprendizaje de nuestra API/servicio.

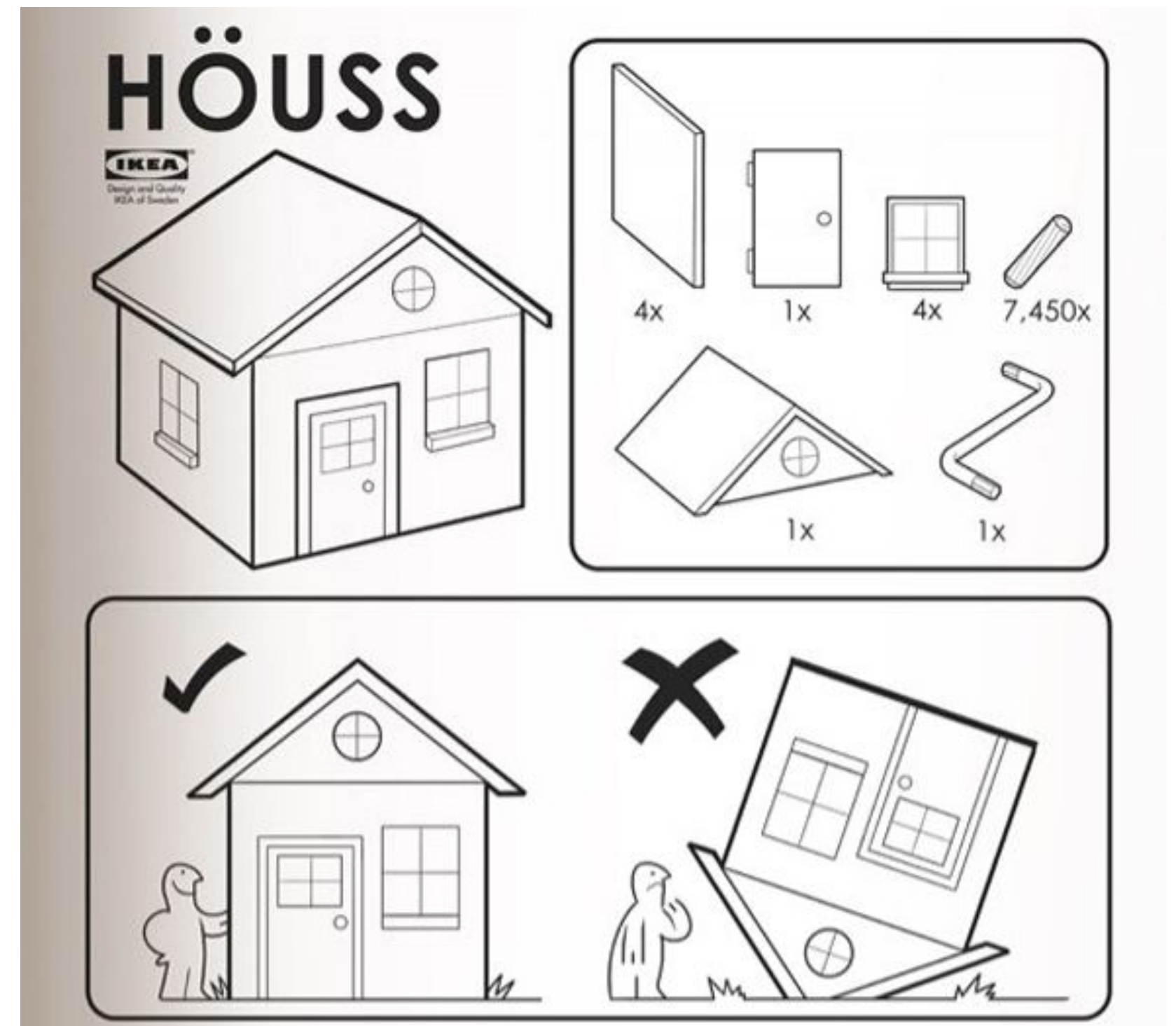


# Documentación

Una documentación bien estructurada facilita el proceso de aprendizaje sobre nuestro servicio API. Es una de las piezas básicas que determinan la experiencia de un desarrollador al hacer uso de nuestros servicios API.

Al igual que lo hacen las instrucciones de armado de un mueble, esta debe cumplir ciertos requisitos que diferencian una buena documentación “simplificada” de una mala documentación con páginas indescifrables, en idiomas de extraña procedencia.

Va más allá del formato de respuesta API o el método de autenticación. La documentación debe ser **interactiva**, mostrando a los desarrolladores **la estructura y la organización** que les facilitará el trabajo. De esta manera no tendrán que pasar tiempo investigando las funciones o perderlo en pruebas totalmente improductivas.



02

# Herramientas de documentación

# Herramientas de documentación API

Desarrollar una API requiere un duro esfuerzo. Tenemos que dedicarnos al diseño y la implementación, a la documentación y a su mantenimiento con el objetivo de que los desarrolladores que la utilicen tengan claro su funcionamiento.

Por este motivo existen herramientas de documentación API que nos facilitarán el trabajo. Las más conocidas son **RAML** (lenguaje de modelado API RESTful), **Slate** y **Swagger**. Esta última es uno de los estándares más utilizados y fue elegida por **Open API Initiative** como modelo de especificación API.

03

# Buenas prácticas



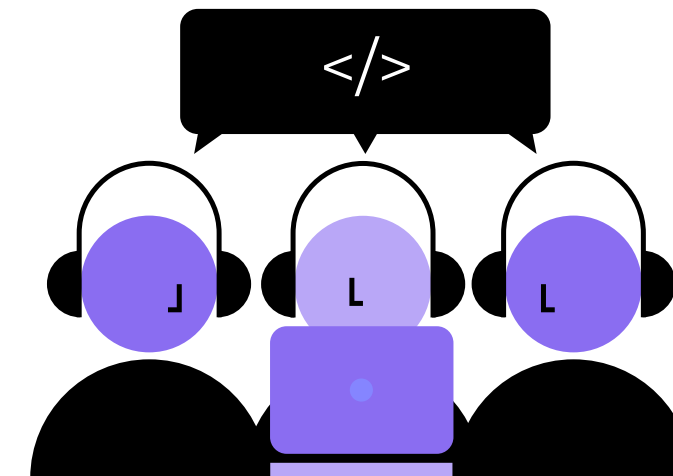
# Descripción endpoint detallada

Amplía la descripción de los endpoints con texto explicativo de la forma más detallada posible, utilizando un lenguaje comprensible, intentando no enfocarse en lo técnico, no todos tienen la misma experiencia o están familiarizados en el entorno de nuestra API.

Aunque las APIs actúan como una capa de abstracción que sirve para ocultar ciertos detalles de las operaciones que se ejecutan, lo mejor para no confundir a los desarrolladores es precisamente evitar abstracciones en la documentación.

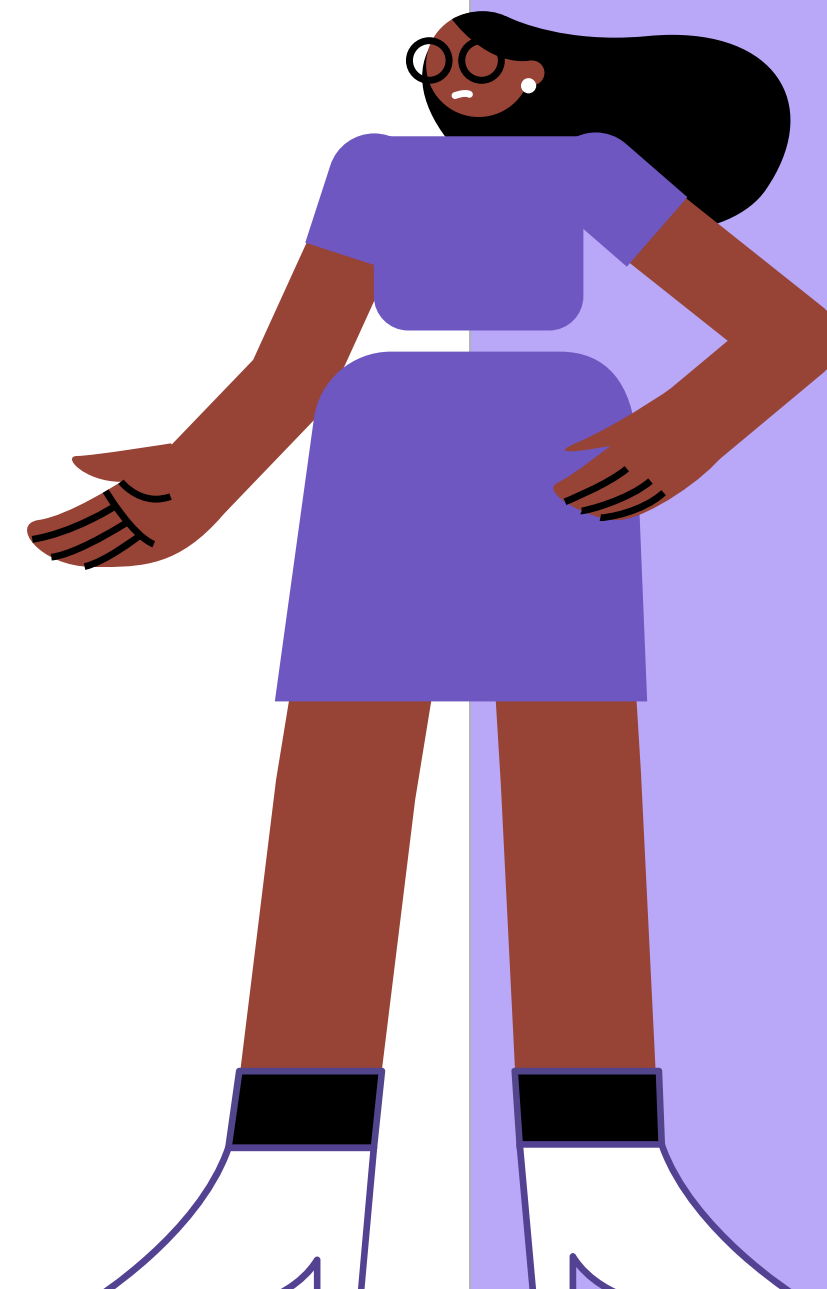
# Detallar los parámetros y las respuestas esperadas

Listar todos los parámetros de entrada y salida que proporcione cada endpoint, además de los tipos de respuesta HTTP esperados. Esto evitará sorpresas a los desarrolladores que consuman la API, se tendrán en cuenta las diferentes casuísticas que deberán valorar en el momento de desarrollar un front end apropiado.



# Conclusiones

Documentar correctamente nuestra API permite a los desarrolladores, internos o externos, mantener una única fuente de información, evitando errores de versionado.



¡Muchas gracias!