

ARQUITECTURA DE DIAGRAMAS

-Álvarez Menéndez, David

U0288705@uniovi.es

-Fernández Vior, Ángel David

U0287998@uniovi.es

-Jirout Cid, Alfredo

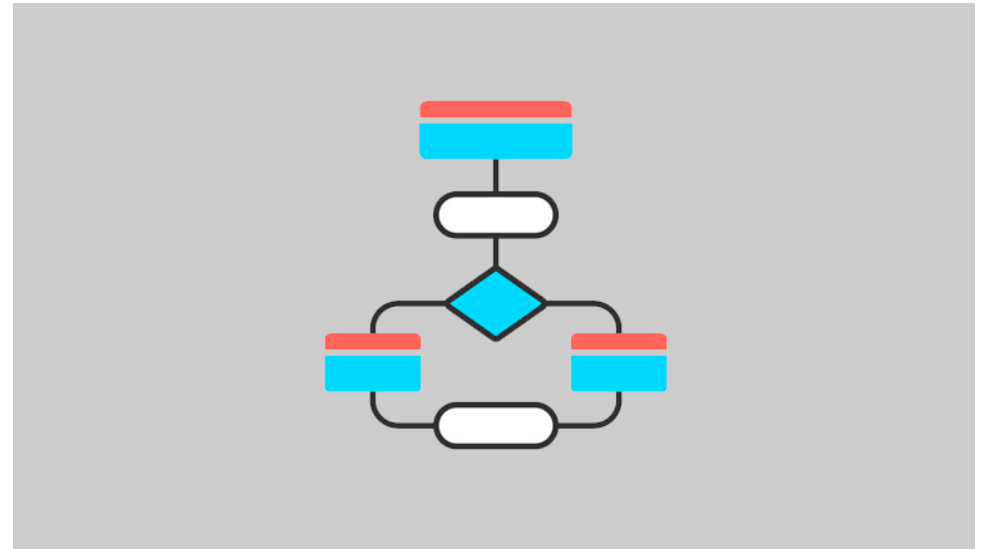
U0288443@uniovi.es

- **ÍNDICE**

- Introducción
- Tipos de Diagramas
- Notación y tipo de diagrama
- Pasado vs Presente
- Mermaid
- Modelo C4
- Miro
- Ahley Peacock
- Técnicas de arquitectura de diagramas

INTRODUCCIÓN

- Diagramas => Muy importantes
- Soporte => Consultarse en cualquier momento
- Claros y legibles para todas las personas que participan en el proyecto
- Hay gente que le tiene algo de odio ya que se ve como algo antiguo a estas alturas, pero es algo muy útil.



TIPOS DE DIAGRAMA



De corta duración: se hacen en reuniones cortas, hechos para que se entienda el código o lo que se tenga que hacer



De larga duración: más formales y mejor estructurados. Se incluyen en el código fuente del proyecto

NOTACIÓN Y TIPO DE DIAGRAMA

Más importante: tenga sentido y transmita la idea correcta del proyecto.

Tener en cuenta: No nos sirve un diagrama con notación perfecta si no transmite la idea que tenemos en mente.

UML: Hay 15 tipos de diagrama, pero hay 2 o 3 que son los más importantes y cubren el 95% de los casos de uso.

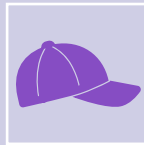
Diagrama de clases: Estructura e interacciones en el código.

Diagramas de secuencia: su favorito, permite entender como interaccionan los usuarios con su sistema.

PRESENTE VS PASADO



Al principio: se hacían en papel y boli, el problema era que una modificación en el diagrama acarreaba una gran cantidad de tiempo de modificación.



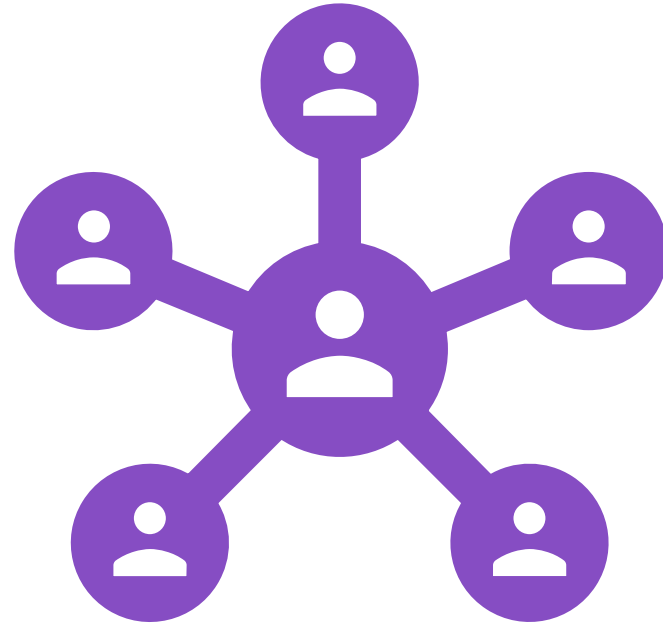
Actualmente: Multitud de herramientas, Visio, Draw.io o PlantUML.



Mermaid: Permite generar diagramas a partir de código, fácil de usar y capaz de hacer los diagramas en GitHub, código abierto aportando versatilidad, nuevas características y adaptables a las necesidades de los usuarios.

MERMAID

- Uso de "actores" o "participantes"
- Representación de servicios involucrados y el navegador
- Buena representación de la secuencia e interacciones
- Diagramas UML
- Modelo C4



MODELO C4

Creado por Simon Brown

Consiste en 4 diagramas diferentes:

Modelo C4



Contexto del sistema



Contenedores



Componentes



Código

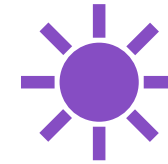
MIRO



Sesiones de tormenta
de ideas
"brainstorming"



Pizarra en línea



Facilita la
planificación de ideas
en tiempo real



Pizarra física vs
herramientas online

AHSLEY PEACOCK



Modelos de dominio. Sirven para comprender mejor el proyecto sobre el que se está trabajando.



Modelo C4. Proporciona una buena visión de la arquitectura del sistema.

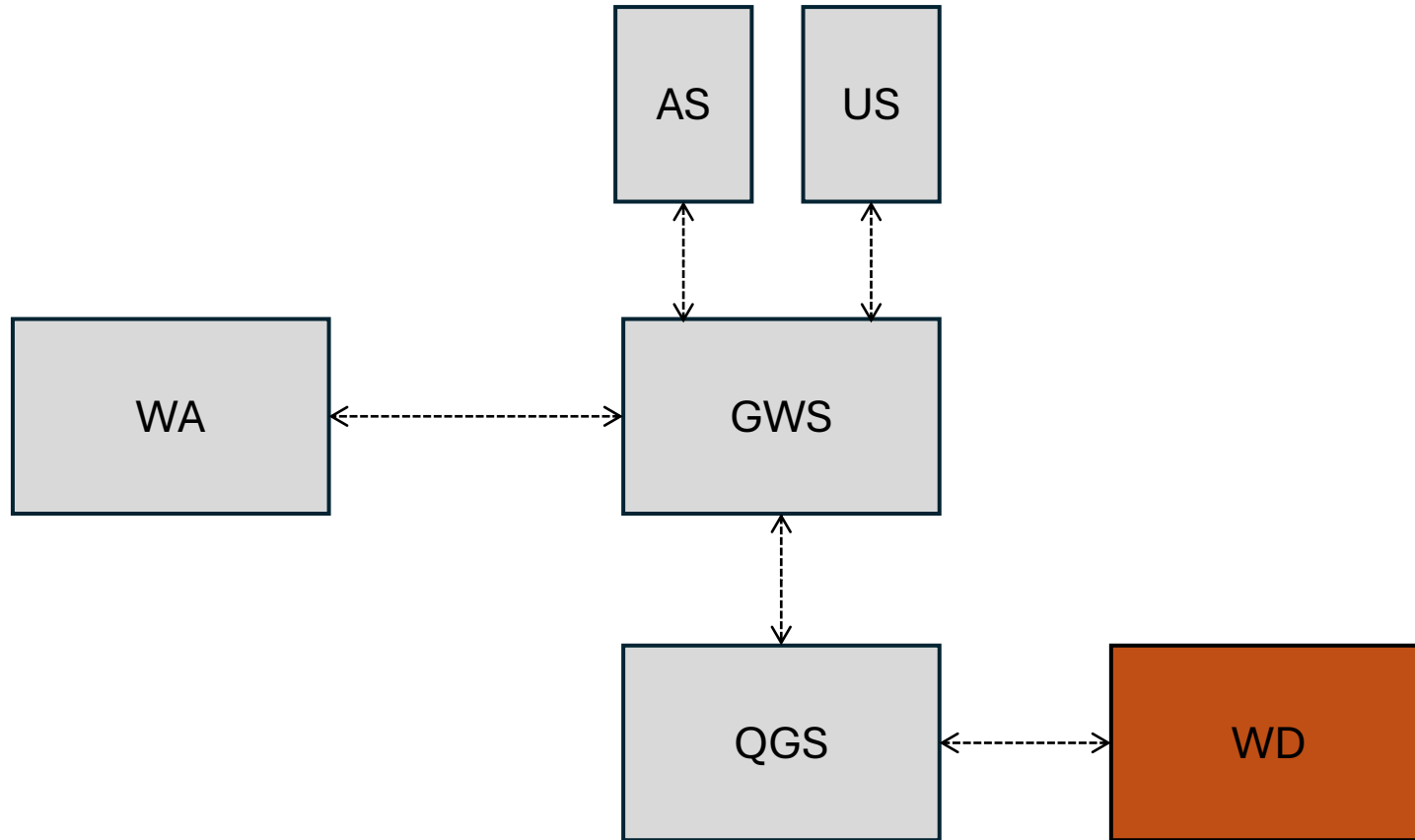


Documentación. Labor muy importante, pero difícil.



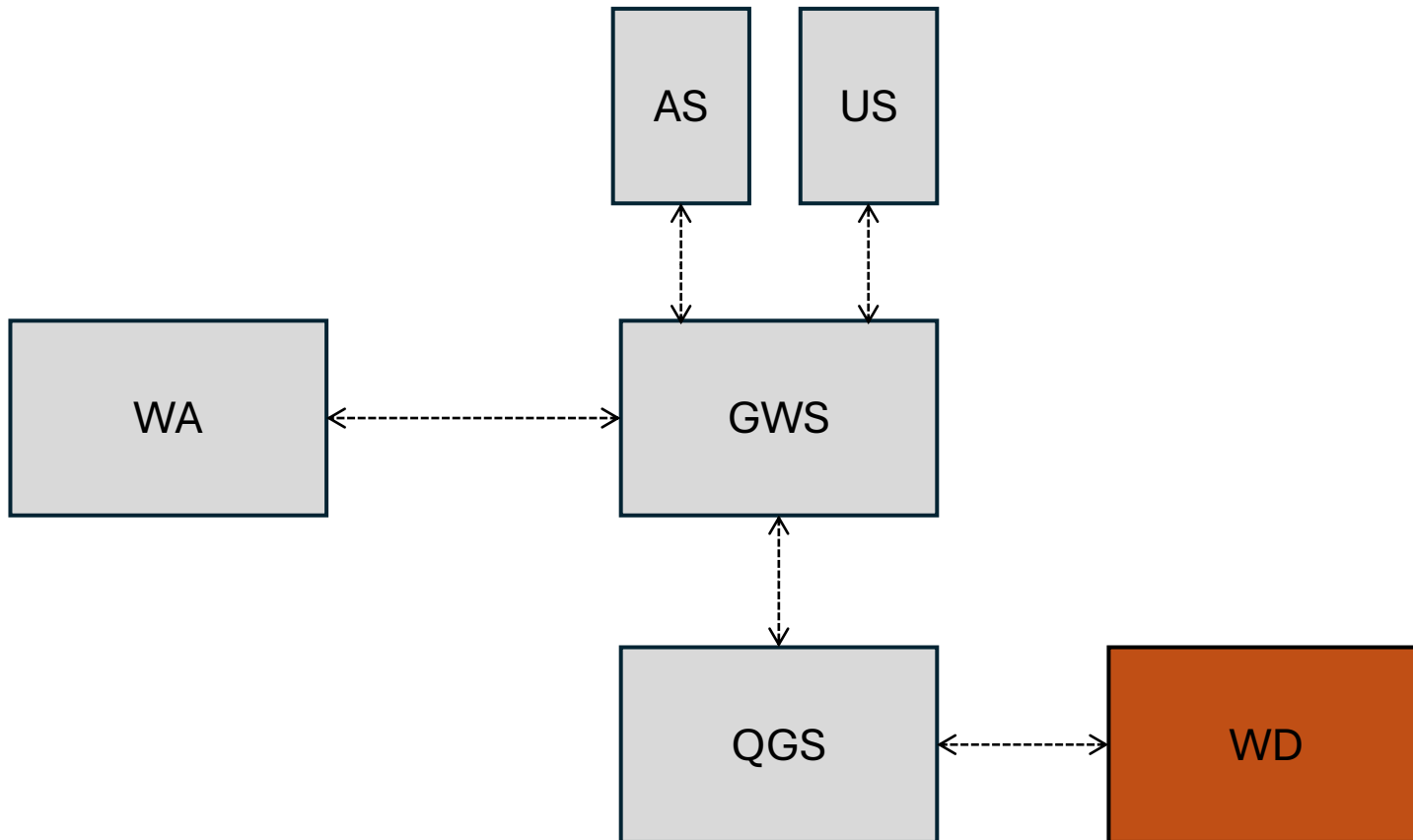
Diagramas de secuencia. Útiles para comprender mejor el código.

DIAGRAMAS DE ARQUITECTURA EFECTIVOS – MARK RICHARDS



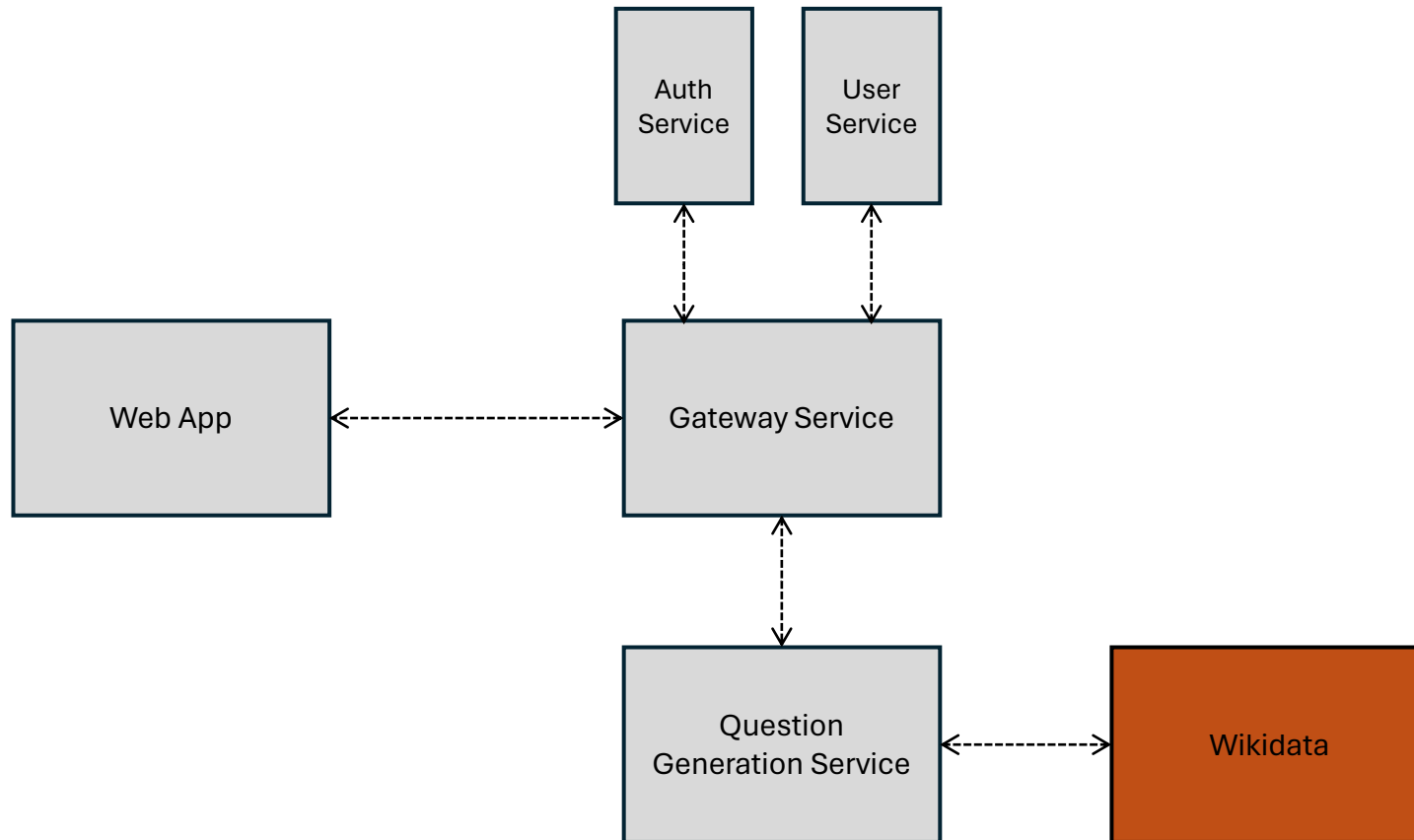
1 - TÍTULO DEL DIAGRAMA

Arquitectura en microservicios de un juego de preguntas



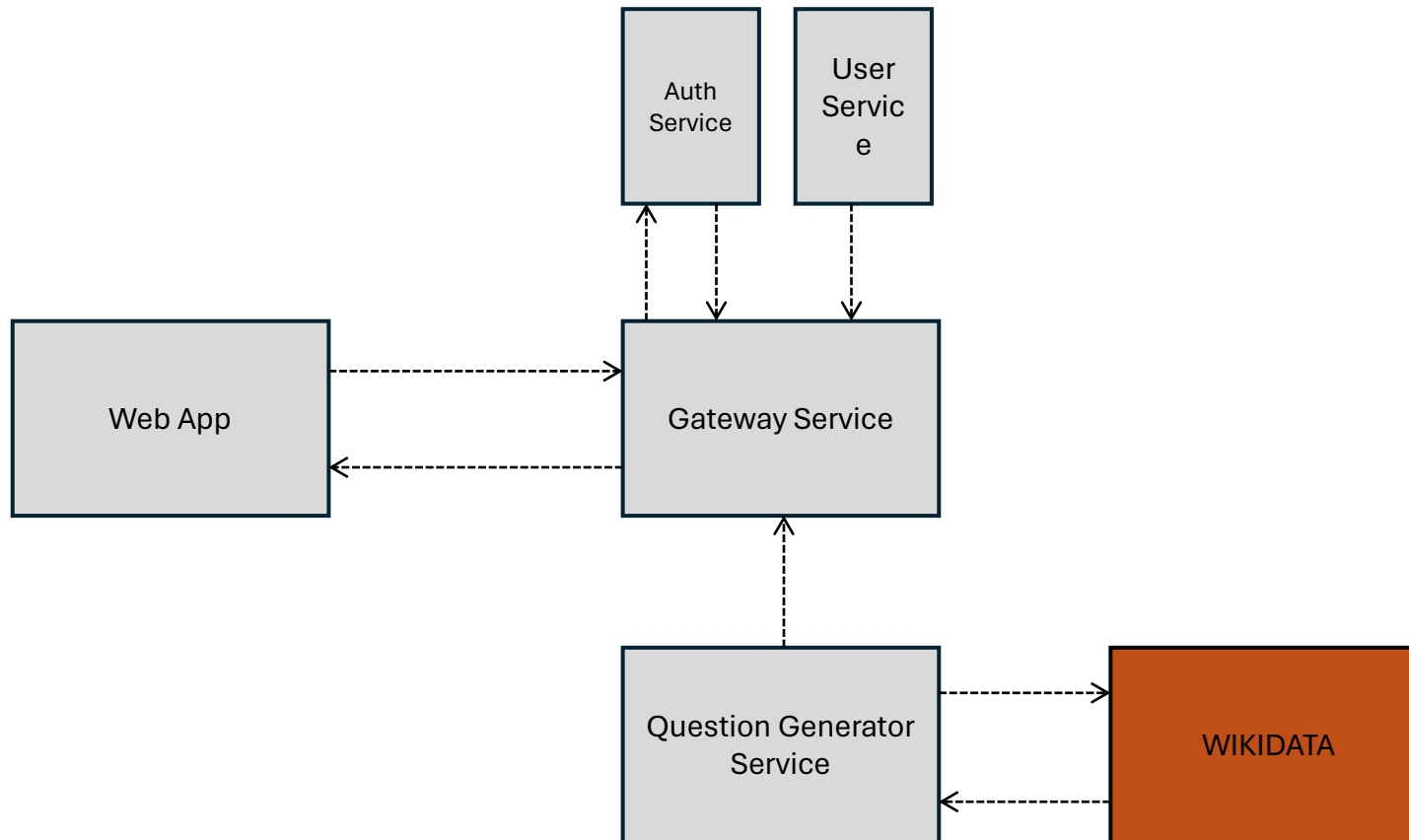
2 – NOMBRES DE LOS COMPONENTES

Arquitectura en microservicios de un juego de preguntas



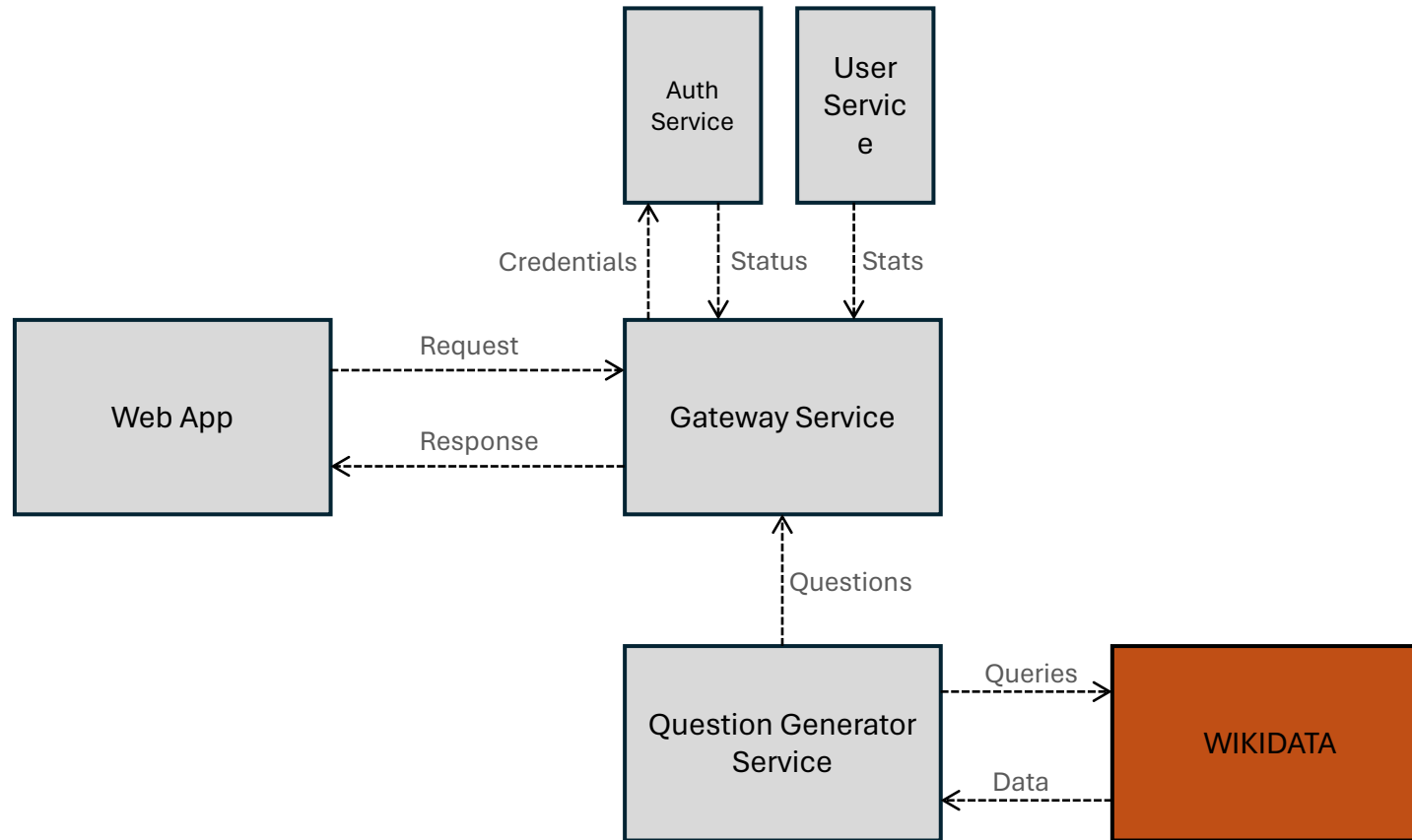
3 – COMUNICACIONES UNIDIRECCIONALES

Arquitectura en microservicios de un juego de preguntas



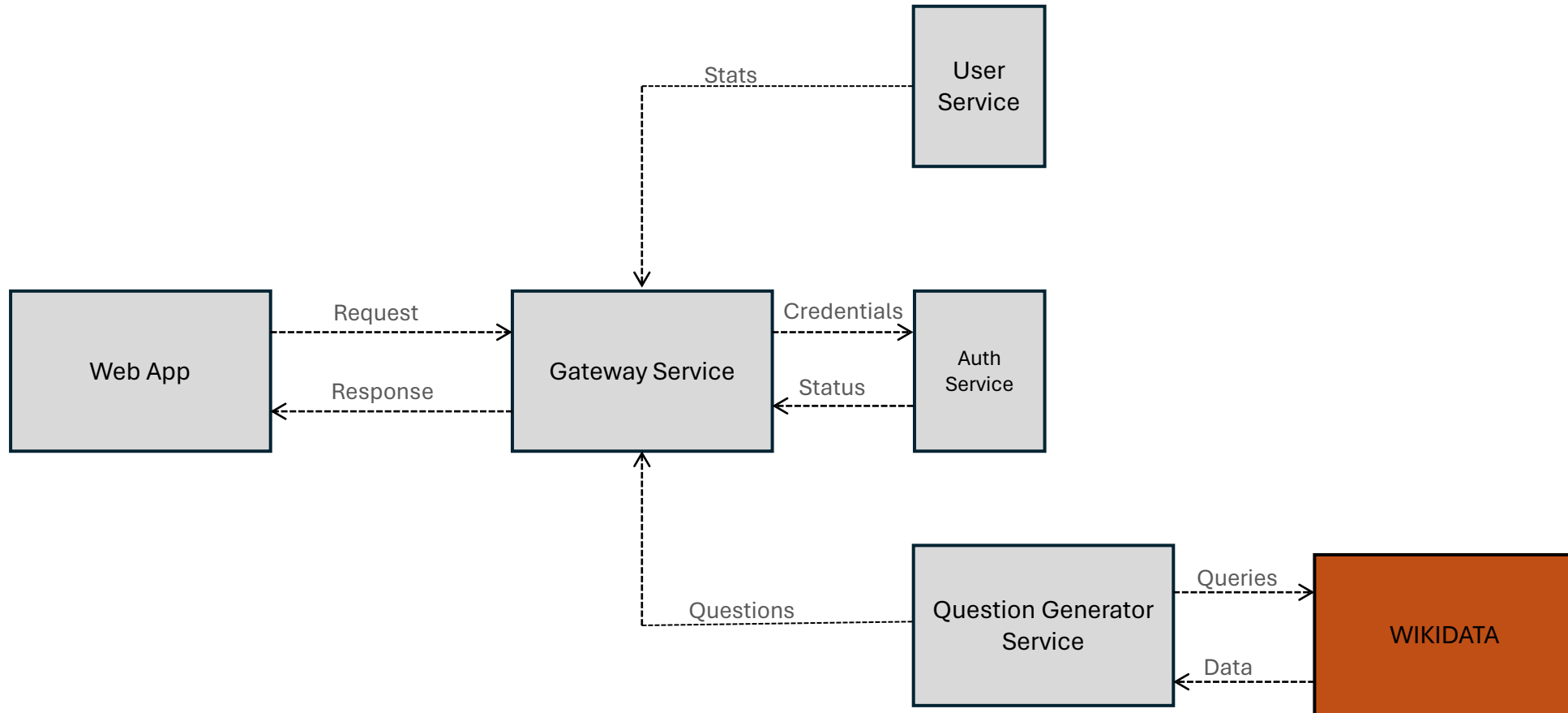
4 – NOMBRES DE LAS COMUNICACIONES

Arquitectura en microservicios de un juego de preguntas



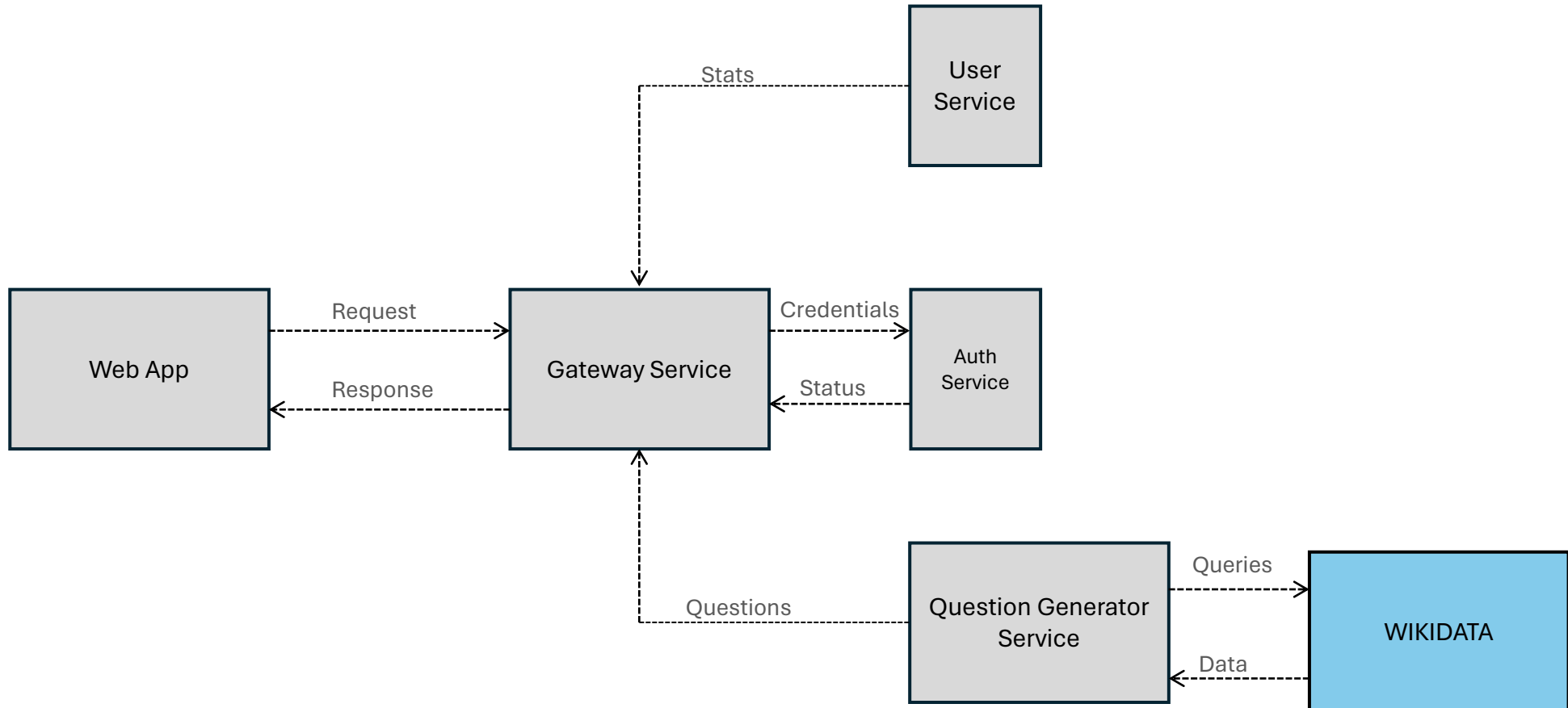
5 - ORIENTACIÓN

Arquitectura en microservicios de un juego de preguntas



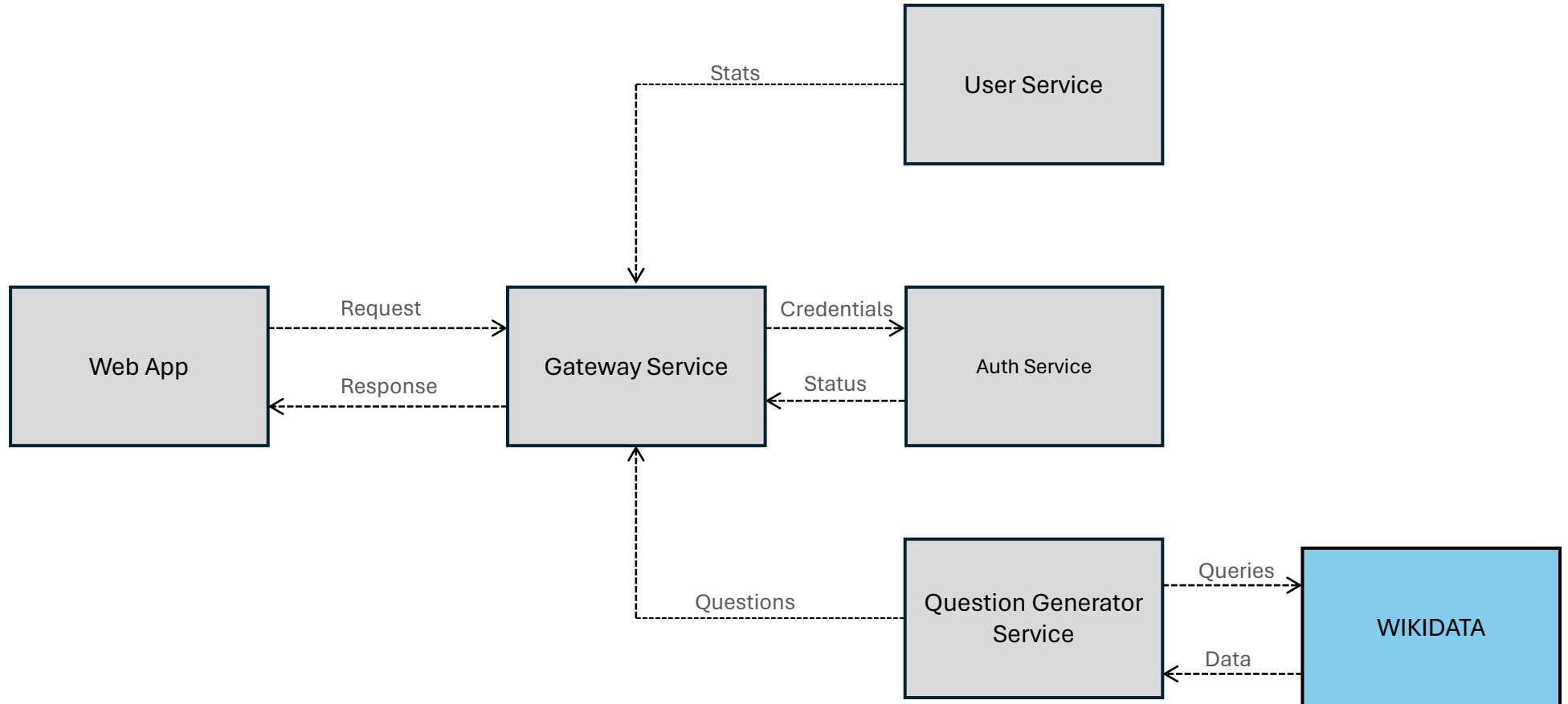
6 - COLORES

Arquitectura en microservicios de un juego de preguntas



7 - FORMAS

Arquitectura en microservicios de un juego de preguntas



8 - LEYENDA

Arquitectura en microservicios de un juego de preguntas

