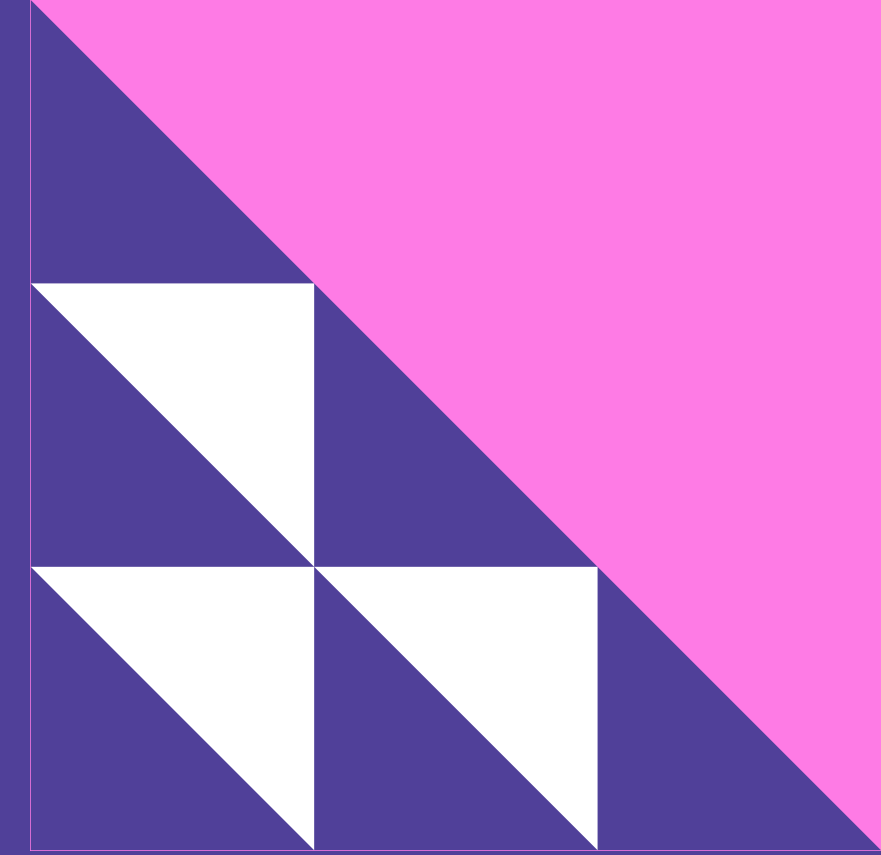


Ayudantía 3

Entrega 3 

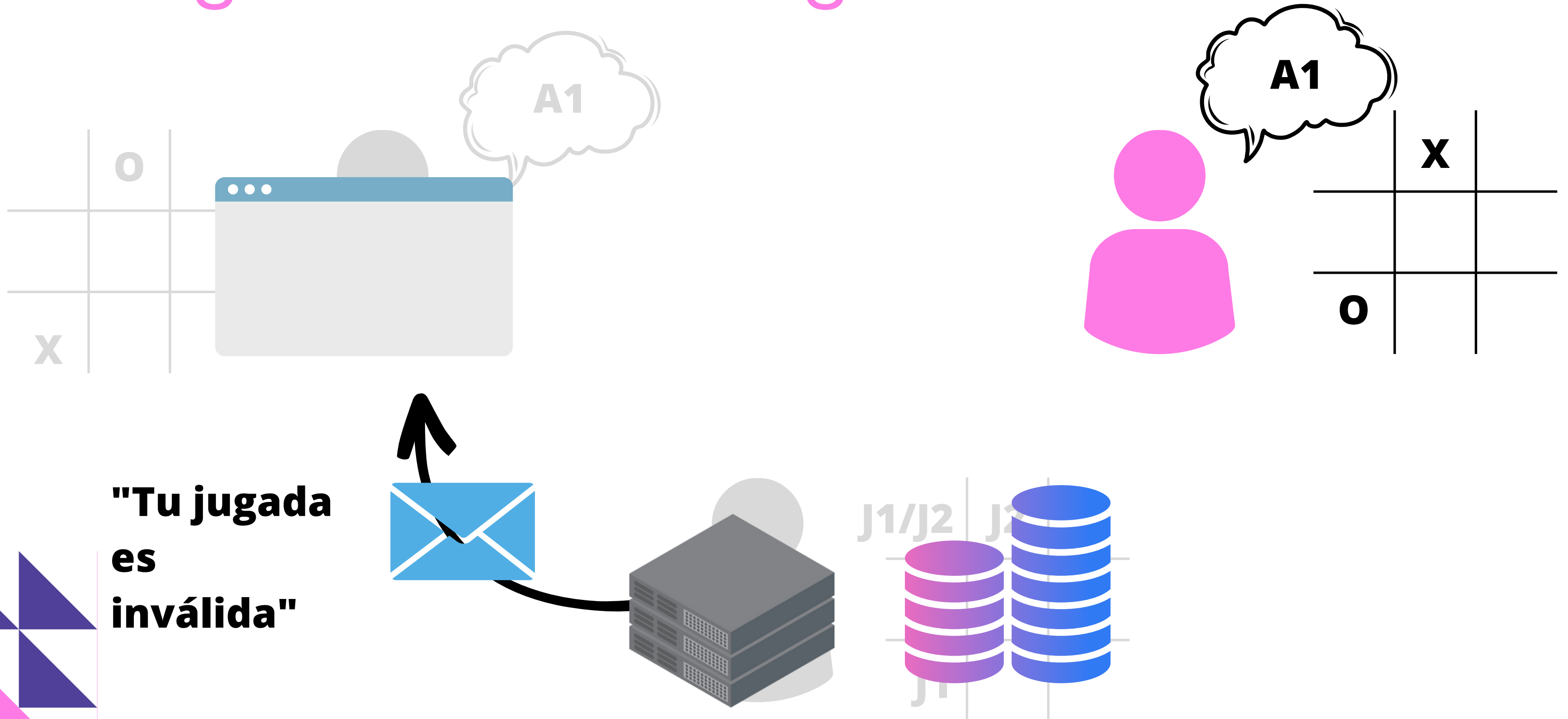
IIC2513 - 2

2023-2



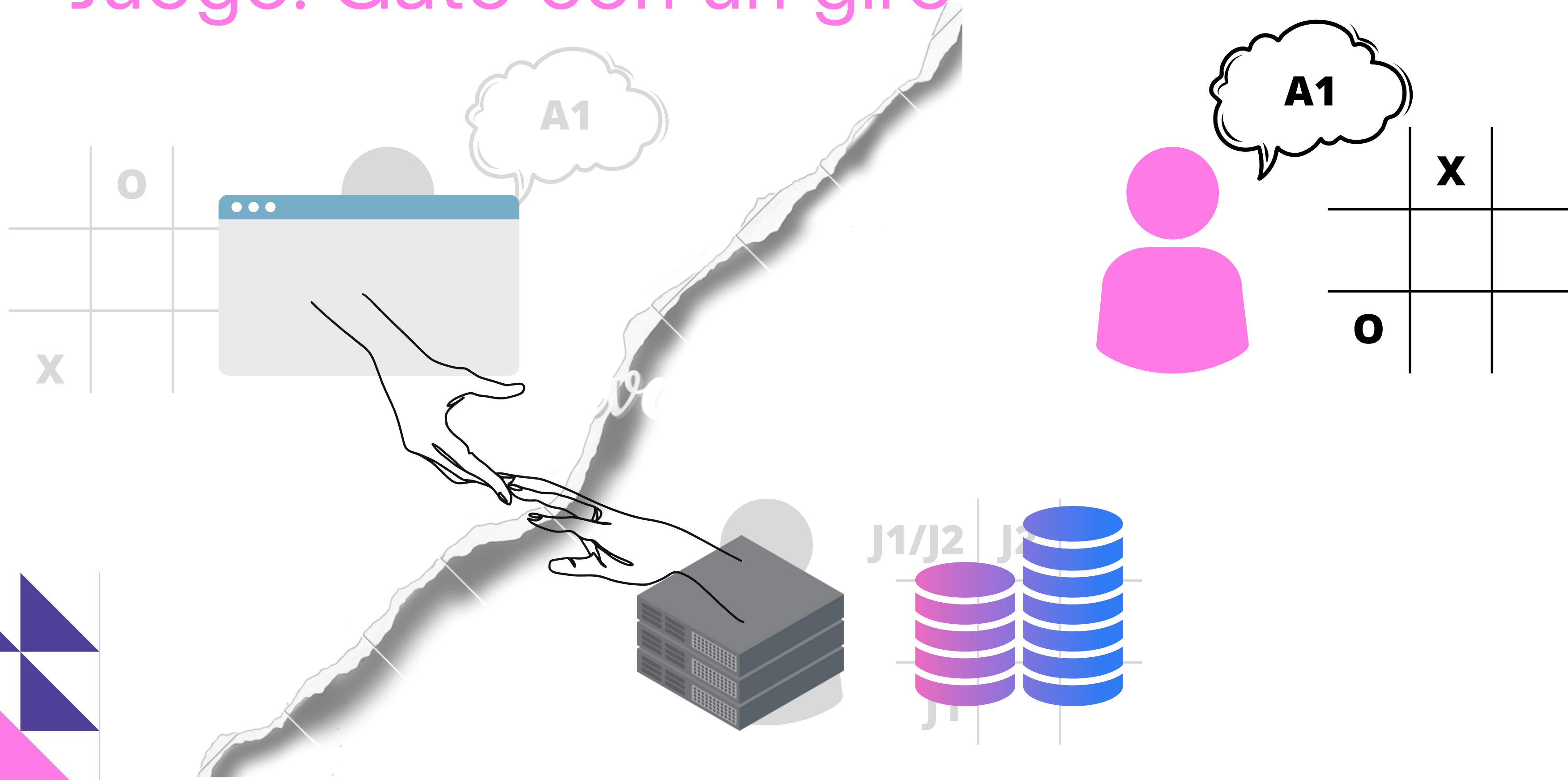
Ayudantía 1

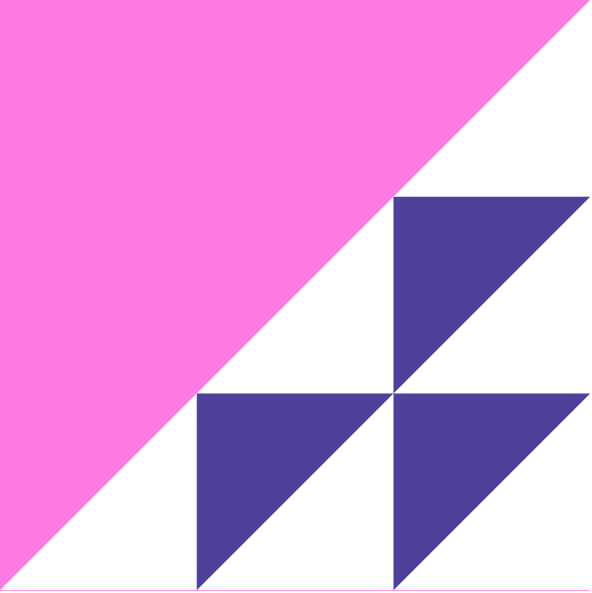
Juego: Gato con un giro



Ayudantía 1

Juego: Gato con un giro

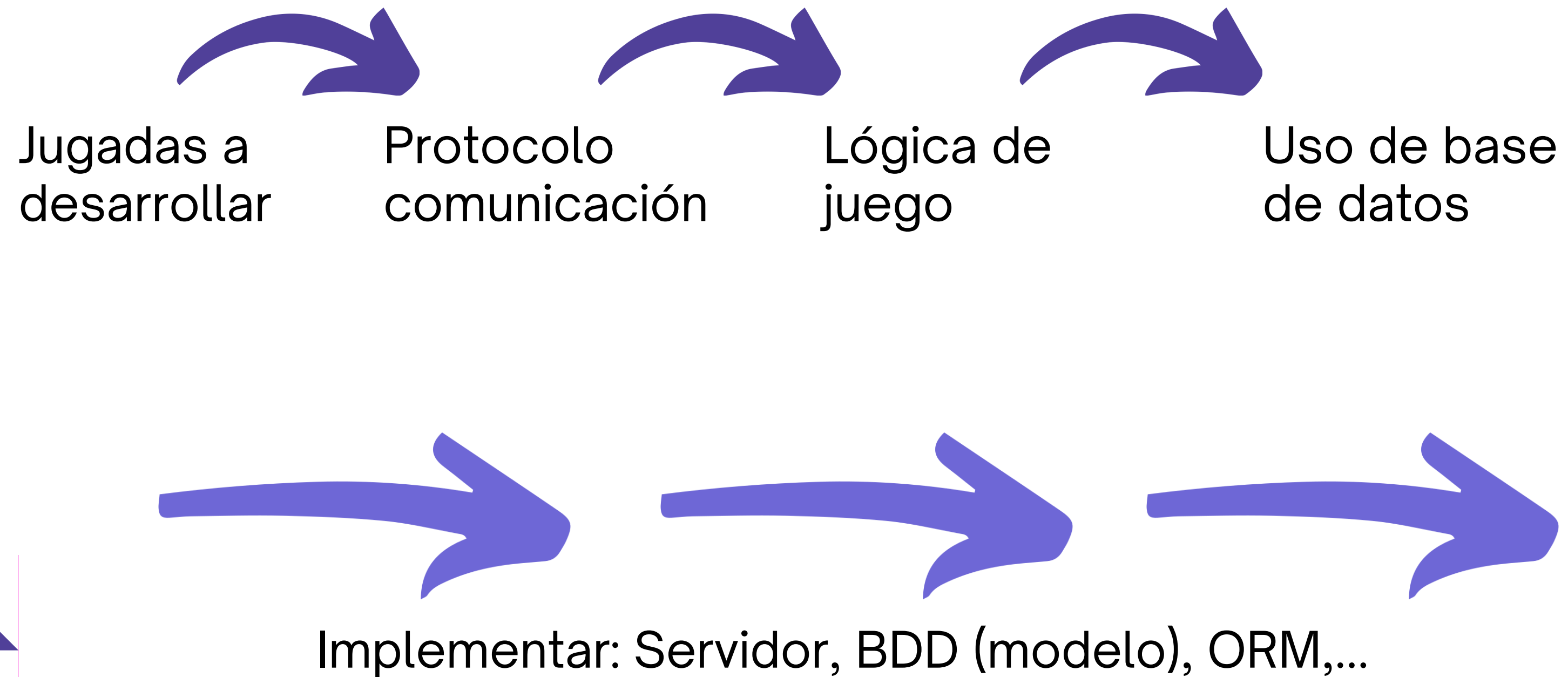




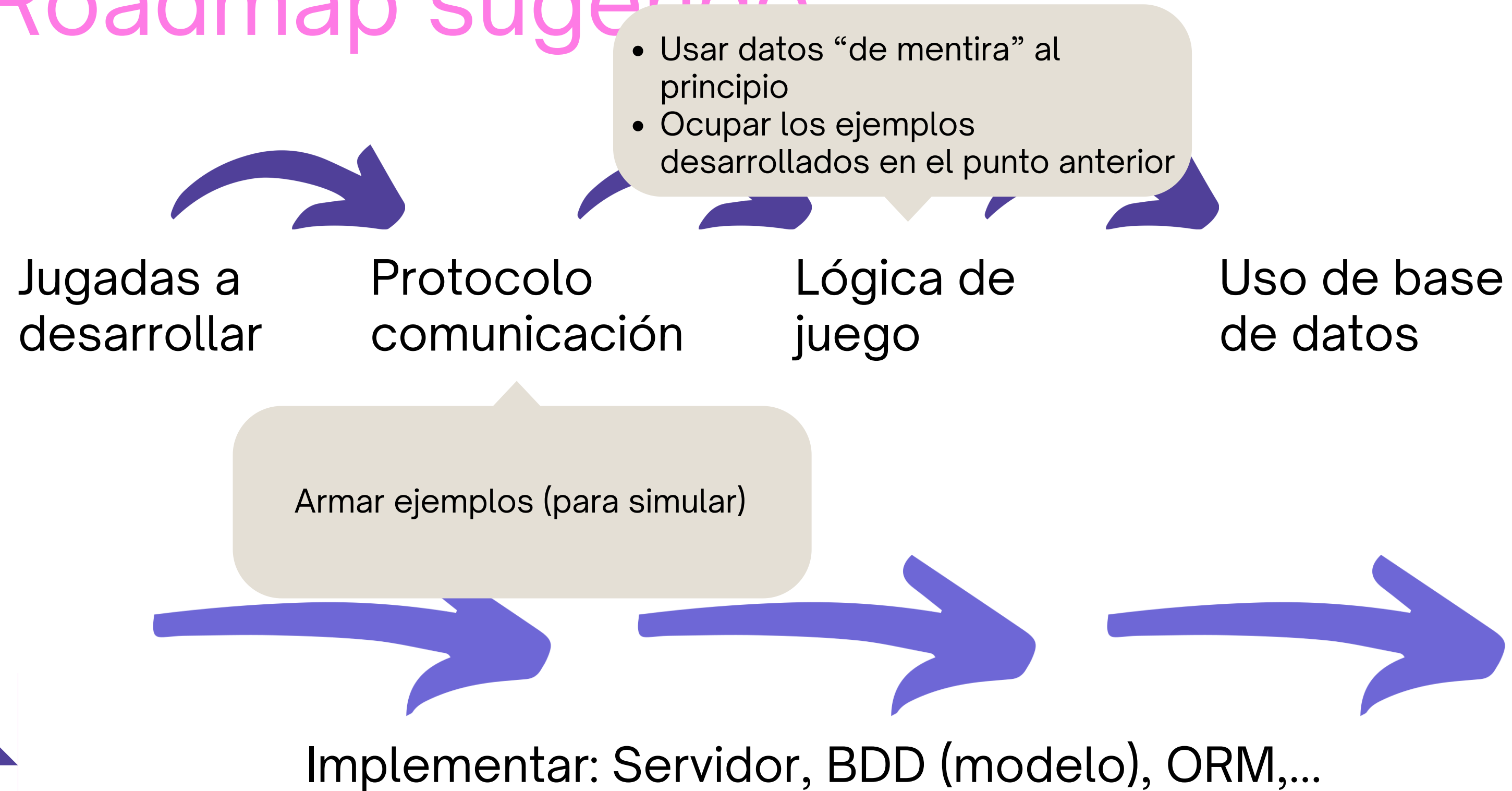
Elementos a desarrollar

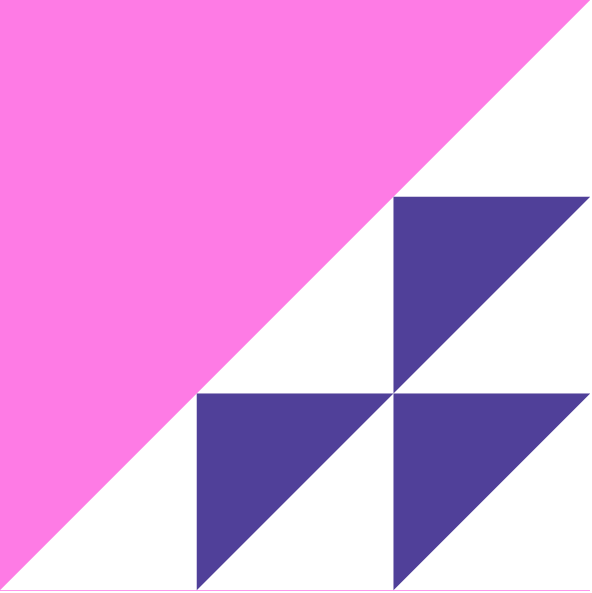
- Front más desarrollado (en base a lo que ya entregaron en la E2)
- Servidor básico
 - Lógica de juego
 - Endpoints (PRÓXIMA ENTREGA)
- Protocolo de comunicación
- Archivos para simular
- Base de datos

Roadmap sugerido



Roadmap sugerido

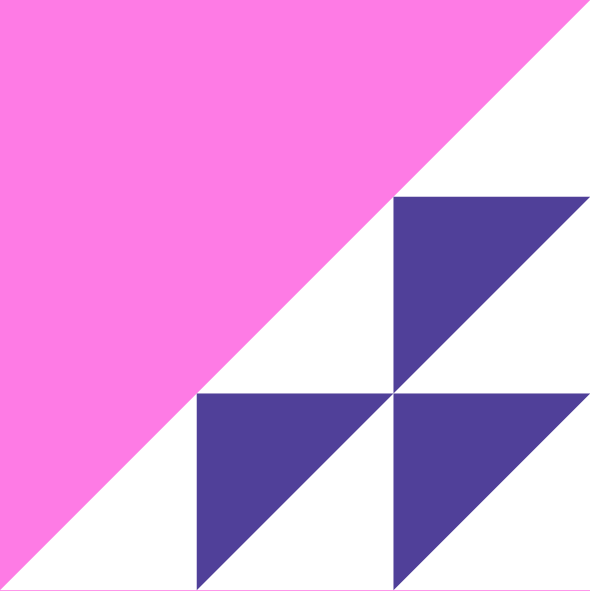




Servidor

En el contexto de Node.js y Koa, un servidor se refiere a un programa o una aplicación que está diseñada para recibir y manejar solicitudes HTTP (Hypertext Transfer Protocol) entrantes y enviar respuestas a esas solicitudes.

Básicamente, un servidor web es una pieza de software que **actúa como intermediario entre un cliente (generalmente un navegador web) y una aplicación web o un recurso en el servidor.**

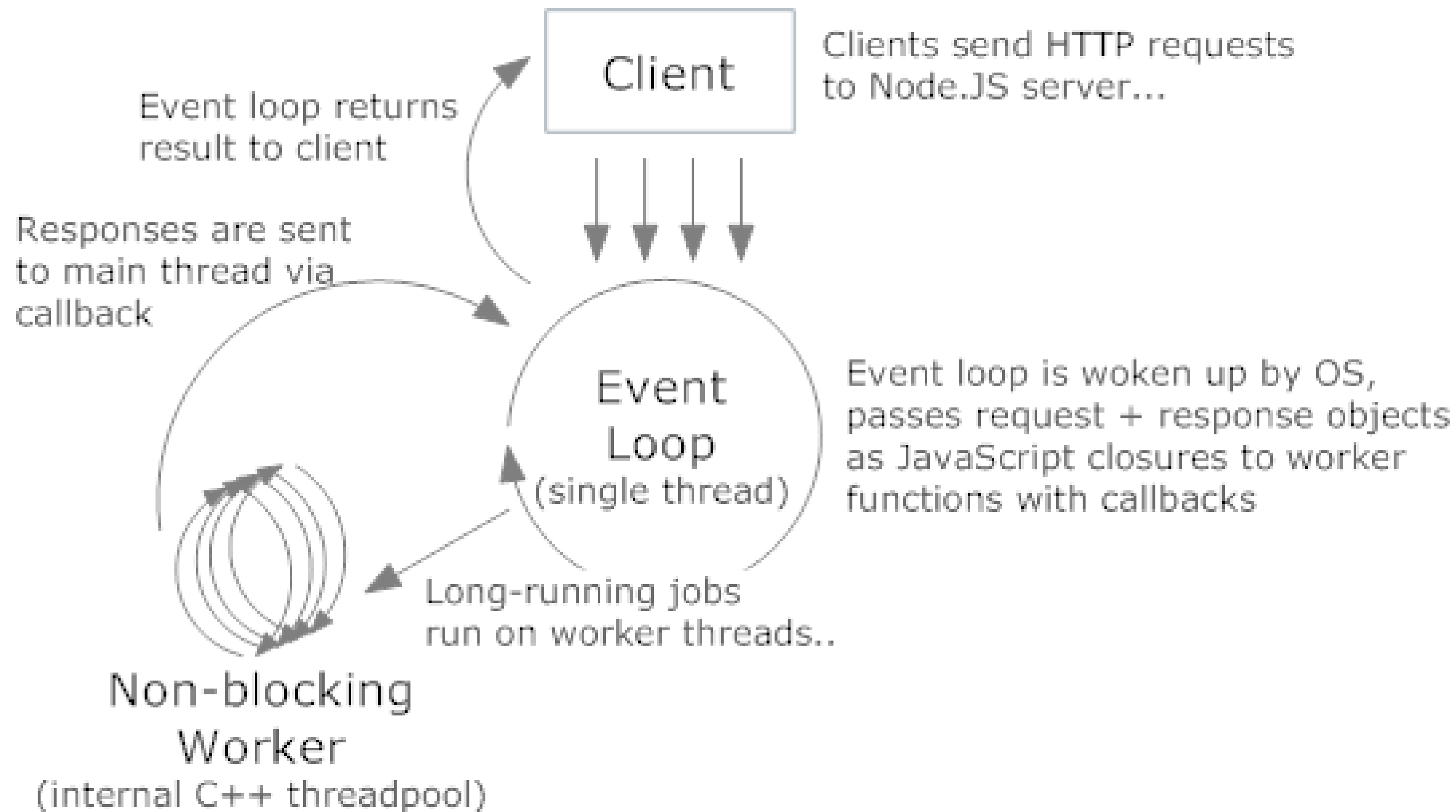


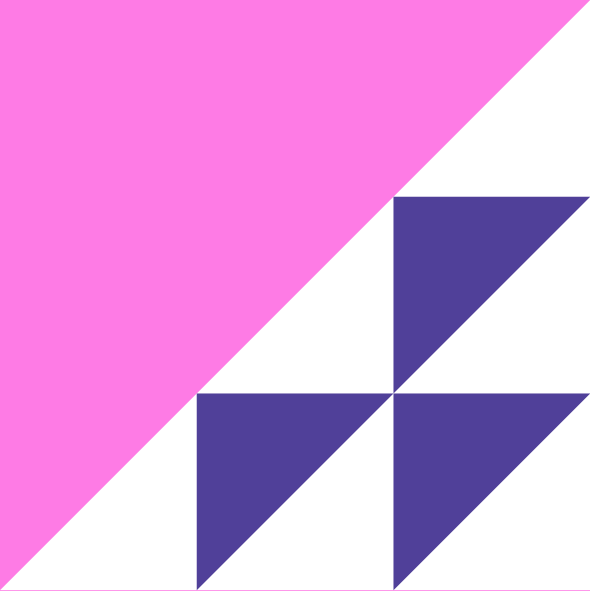
Asincronía

En la mayoría de los casos, las consultas a una base de datos pueden ser operaciones que requieren tiempo, ya que involucran la comunicación con la base de datos, la ejecución de consultas y la espera de los resultados.

Async/Await: Esta es una sintaxis moderna de JavaScript que simplifica la escritura de código asíncrono. Permite escribir código que parece síncrono mientras que, en realidad, se ejecuta de manera asíncrona.

Node.JS Processing Model



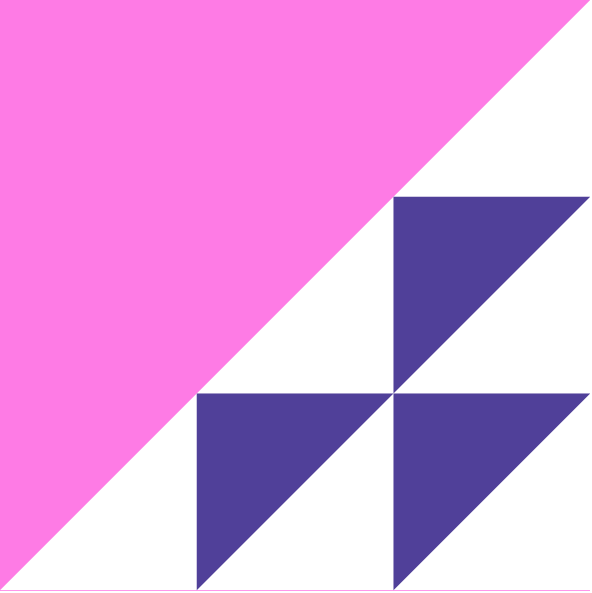


Context (ctx)

Dentro de Koa, el context (ctx) es un objeto que **encapsula la petición (request) y la respuesta (response) de Node en un solo objeto.**

Proporciona una interfaz más amigable y más rica en funcionalidades que la interfaz de solicitud y respuesta de Node por sí misma.

El context es una parte central de Koa y es fundamental para el manejo de solicitudes. Cada vez que llega una solicitud, Koa crea un objeto context, lo pasa a través de una serie de funciones middleware, y finalmente produce una respuesta.



¡Código!