

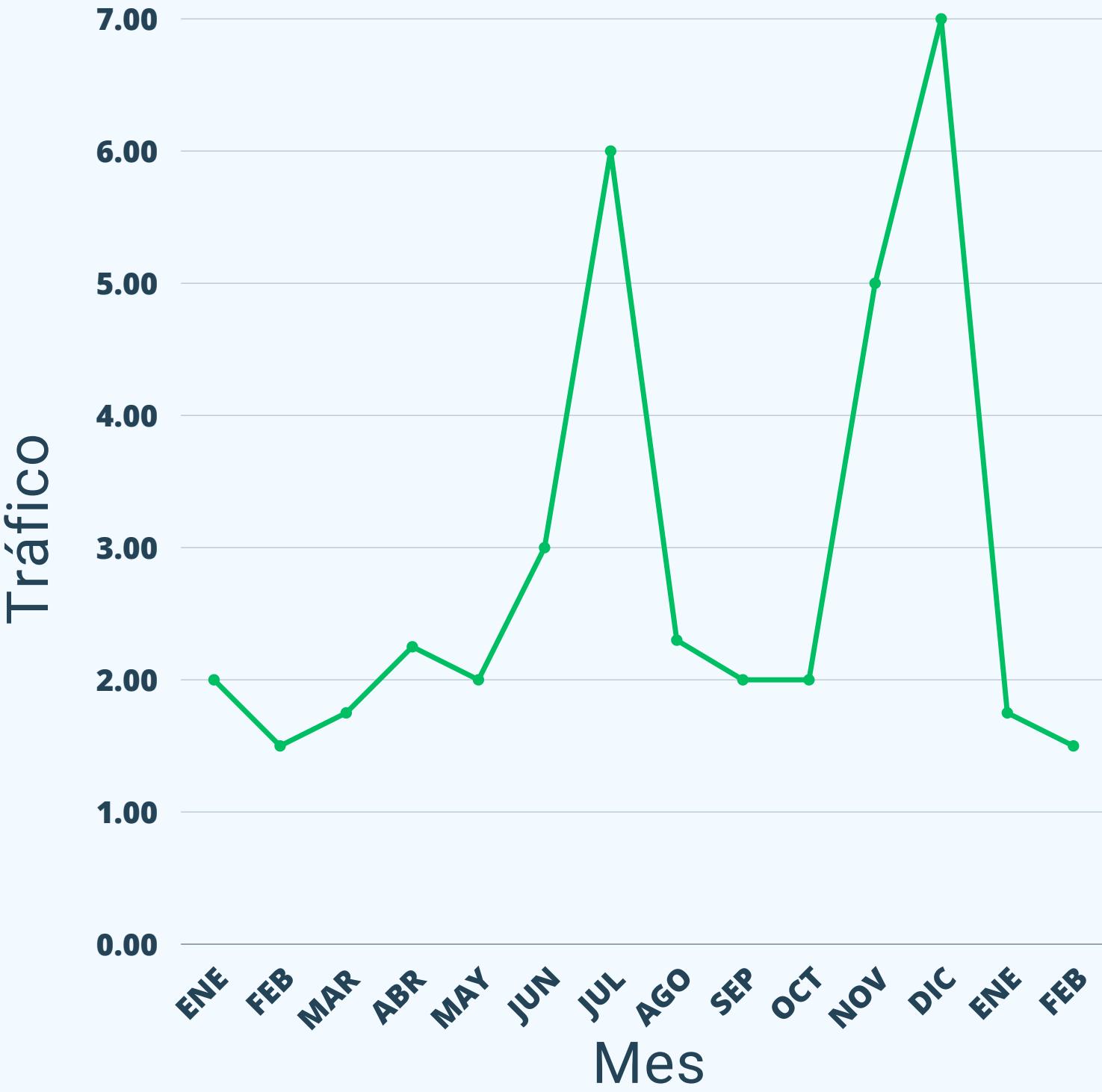
INTRODUCCIÓN

ESCALABILIDAD

Daniel Blanco Calviño

ESCALABILIDAD

- Imprescindible un **buen rendimiento**.
 - No es suficiente con que rinda bien bajo ciertas condiciones.
 - Debe hacerlo bajo **cualquier contexto**.



ESCALABILIDAD

- Imprescindible un **buen rendimiento**.
 - No es suficiente con que rinda bien bajo ciertas condiciones.
 - Debe hacerlo bajo **cualquier contexto**.
- La escalabilidad es la **habilidad** de un sistema **para soportar más carga añadiendo recursos** al mismo y **sin verse afectado su rendimiento**.
 - Puntual por picos de uso en nuestro sistema.
 - Permanente por crecimiento en el número de usuarios de la plataforma.

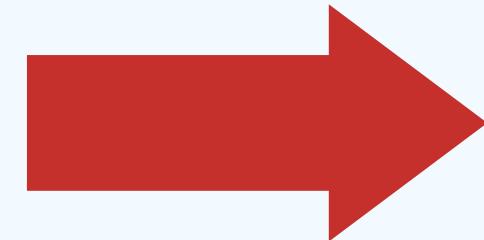
ESCALABILIDAD

- Imprescindible un **buen rendimiento**.
 - No es suficiente con que rinda bien bajo ciertas condiciones.
 - Debe hacerlo bajo **cualquier contexto**.
- La escalabilidad es la **habilidad** de un sistema **para soportar más carga añadiendo recursos** al mismo y **sin verse afectado su rendimiento**.
 - Puntual por picos de uso en nuestro sistema.
 - Permanente por crecimiento en el número de usuarios de la plataforma.
- Escalabilidad **vertical**, escalabilidad **horizontal** y escalabilidad en la **organización**.

ESCALABILIDAD VERTICAL

- Se añaden más recursos a un servidor.

x CPU
y memoria



2x CPU
2y memoria

ESCALABILIDAD VERTICAL



Muy sencilla de implementar. No se necesita adaptar el código.



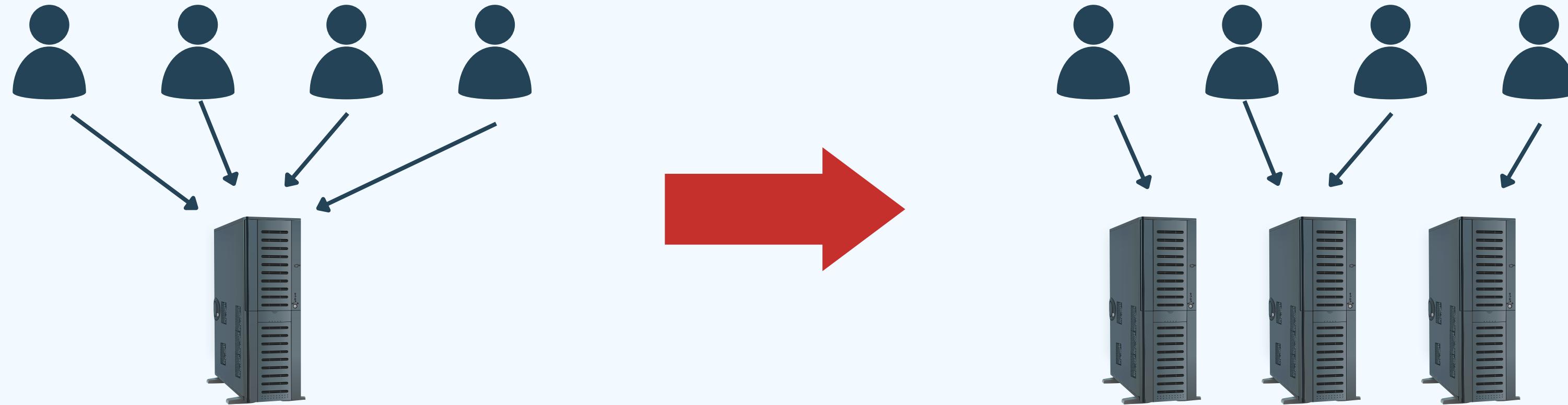
Límite en los recursos que podemos añadir a un único nodo.



No es tolerante a fallos.

ESCALABILIDAD HORIZONTAL

- **Se añaden más servidores.** Se ejecutan múltiples instancias de nuestros servicios.



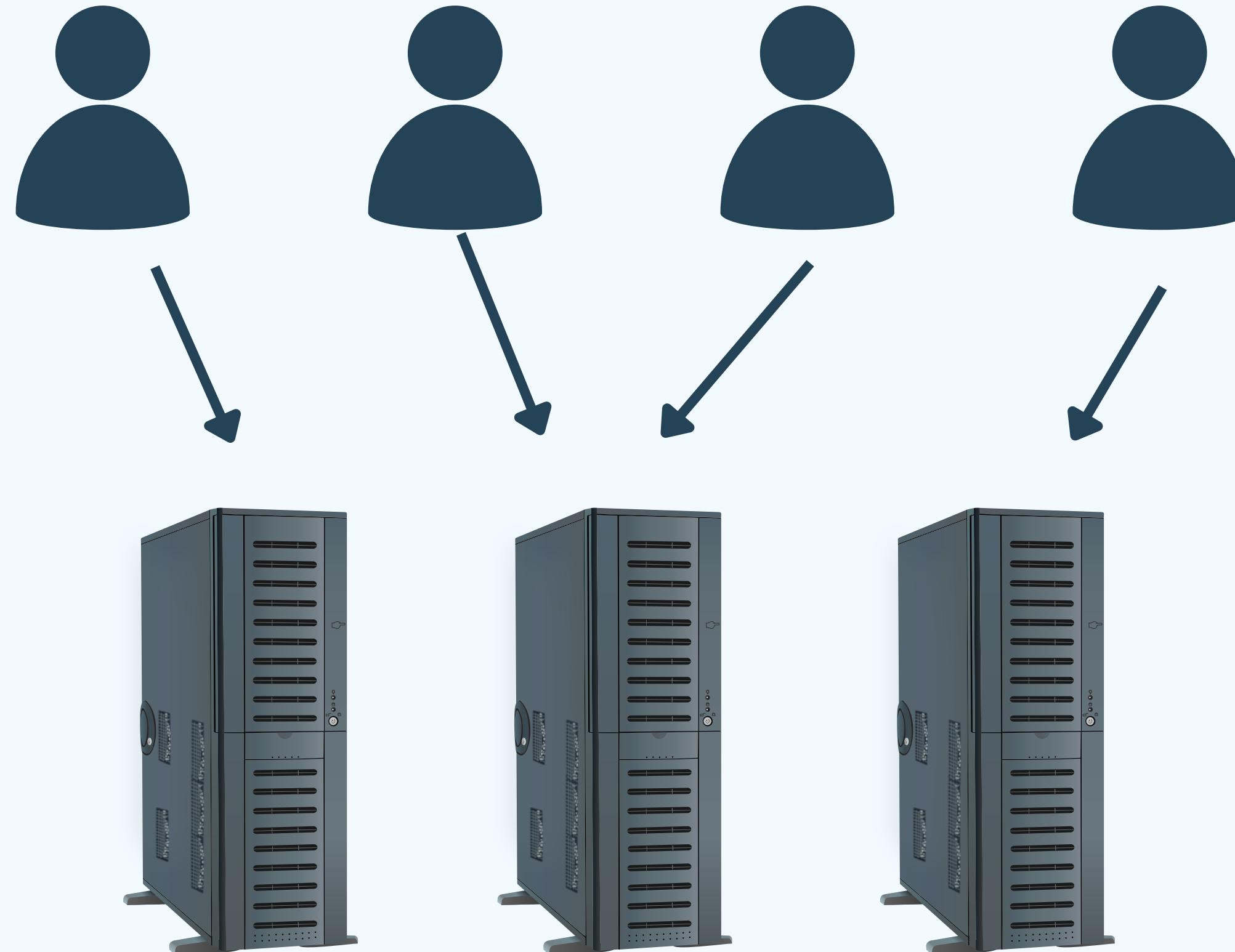
ESCALABILIDAD HORIZONTAL

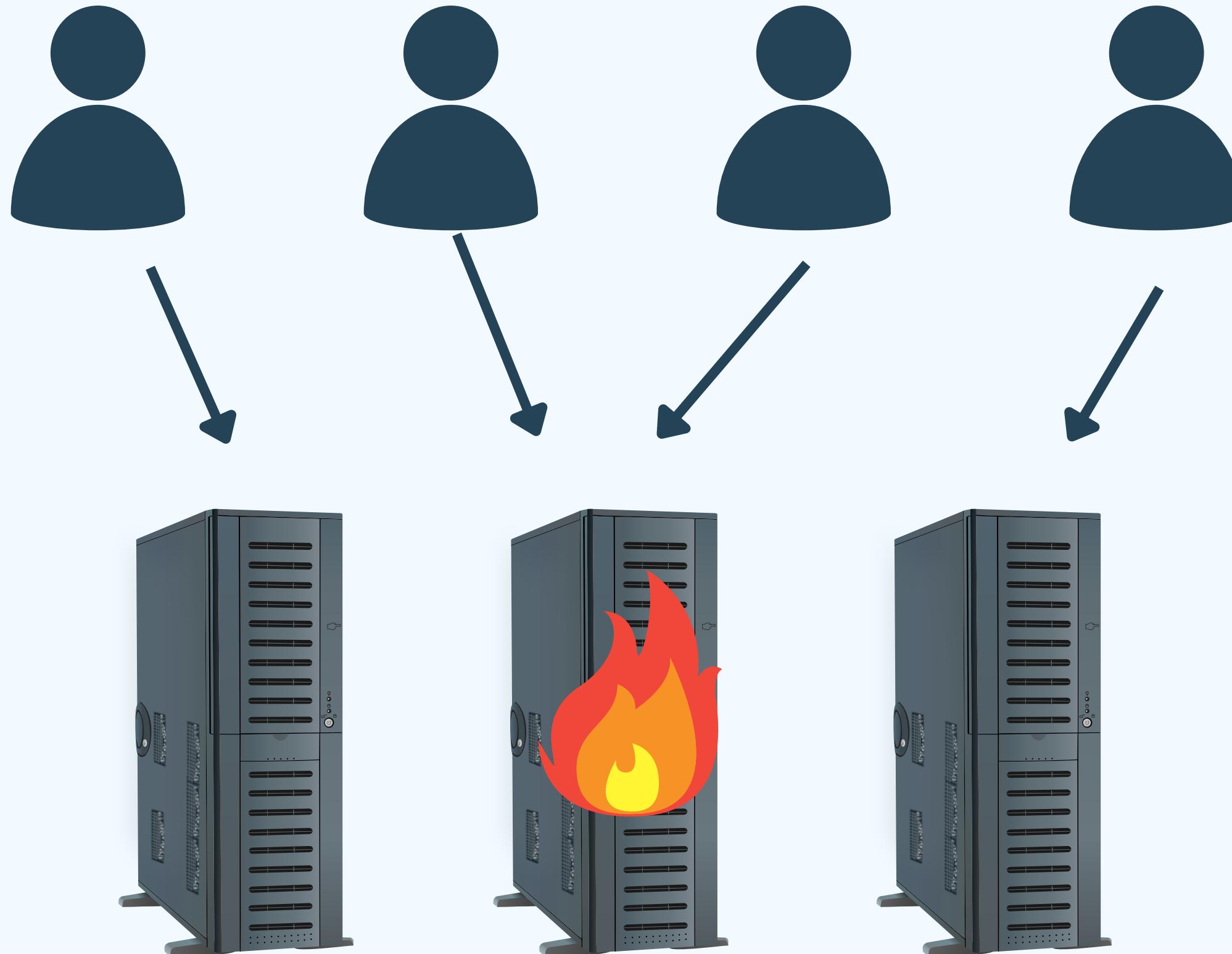


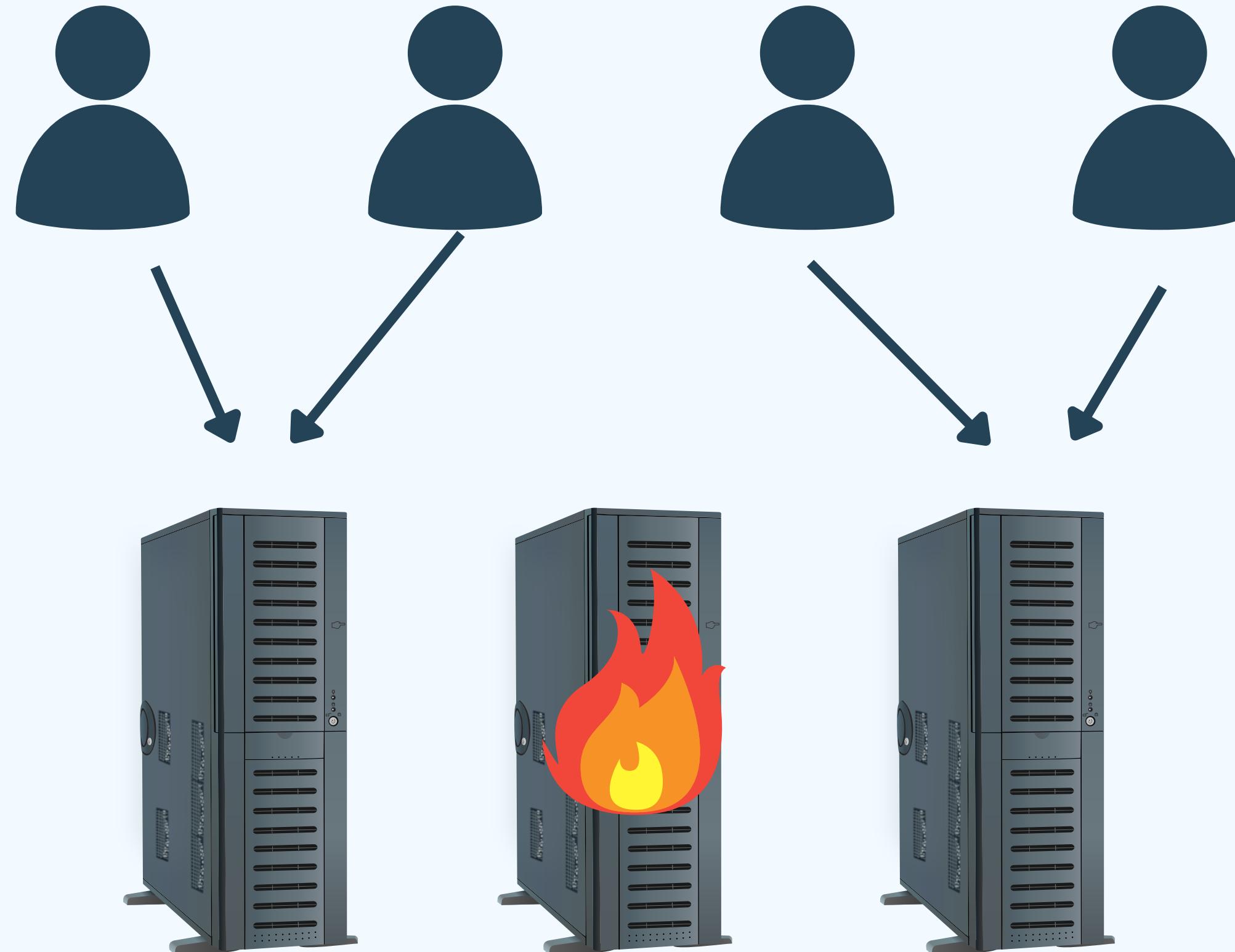
No tiene límite teórico.

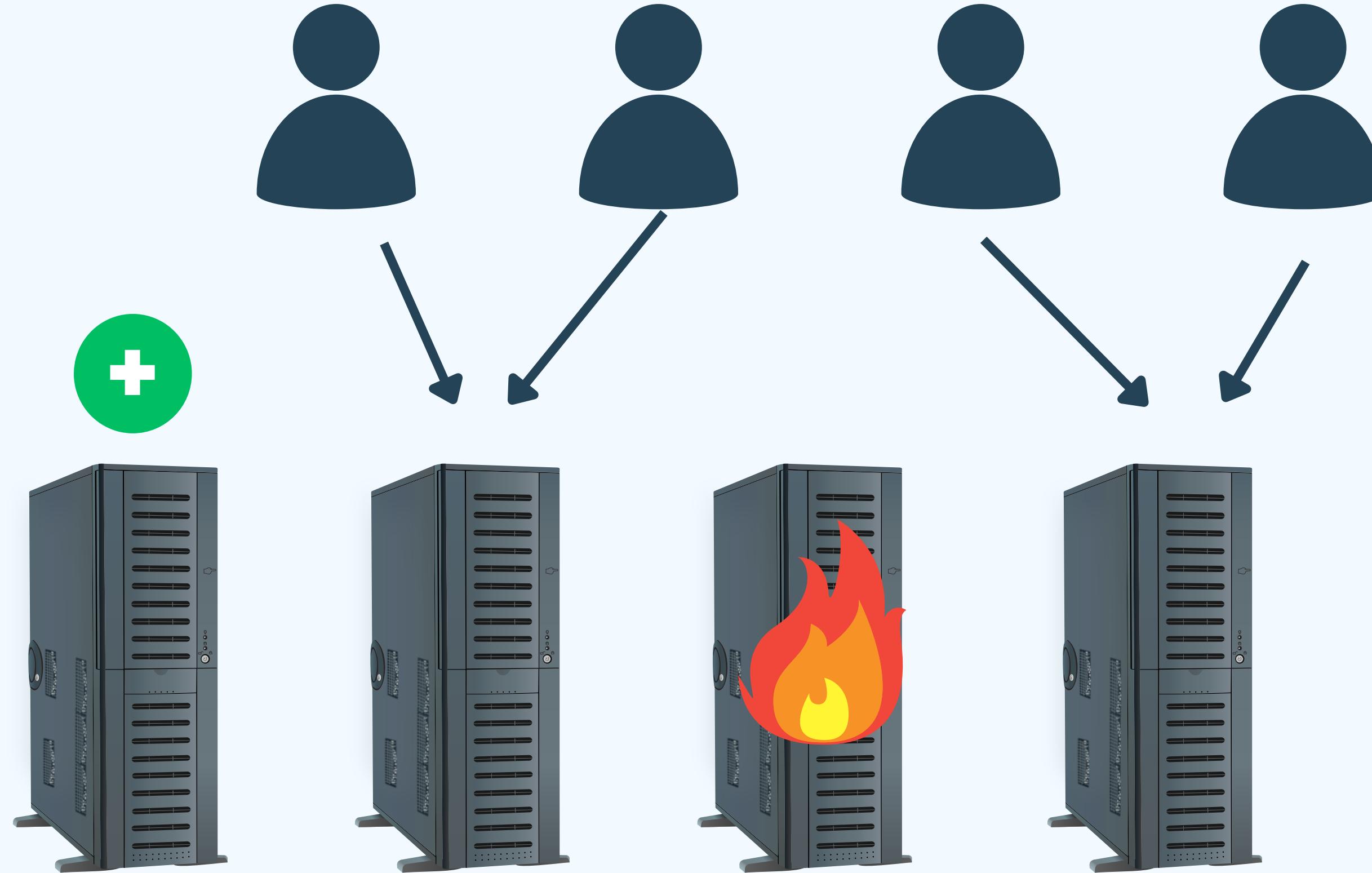


Hace que nuestro sistema sea tolerante a fallos.









ESCALABILIDAD HORIZONTAL



No tiene límite teórico.



Hace que nuestro sistema sea **tolerante a fallos**.



Puede ser necesario **adaptar nuestro código**.



Mayor **complejidad**.

GESTIÓN DEL ESTADO

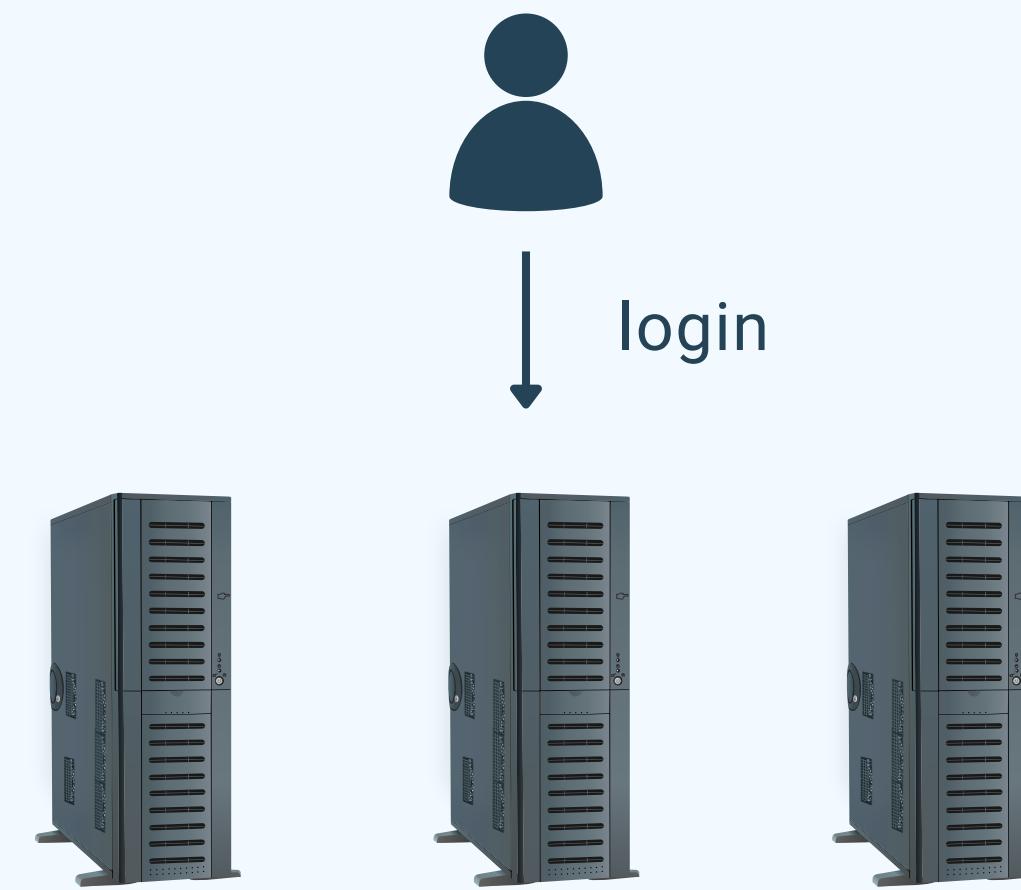
- Dos formas de gestionar el estado.
 - Servicios **stateful**.
 - Servicios **stateless**.

SERVICIOS STATEFUL

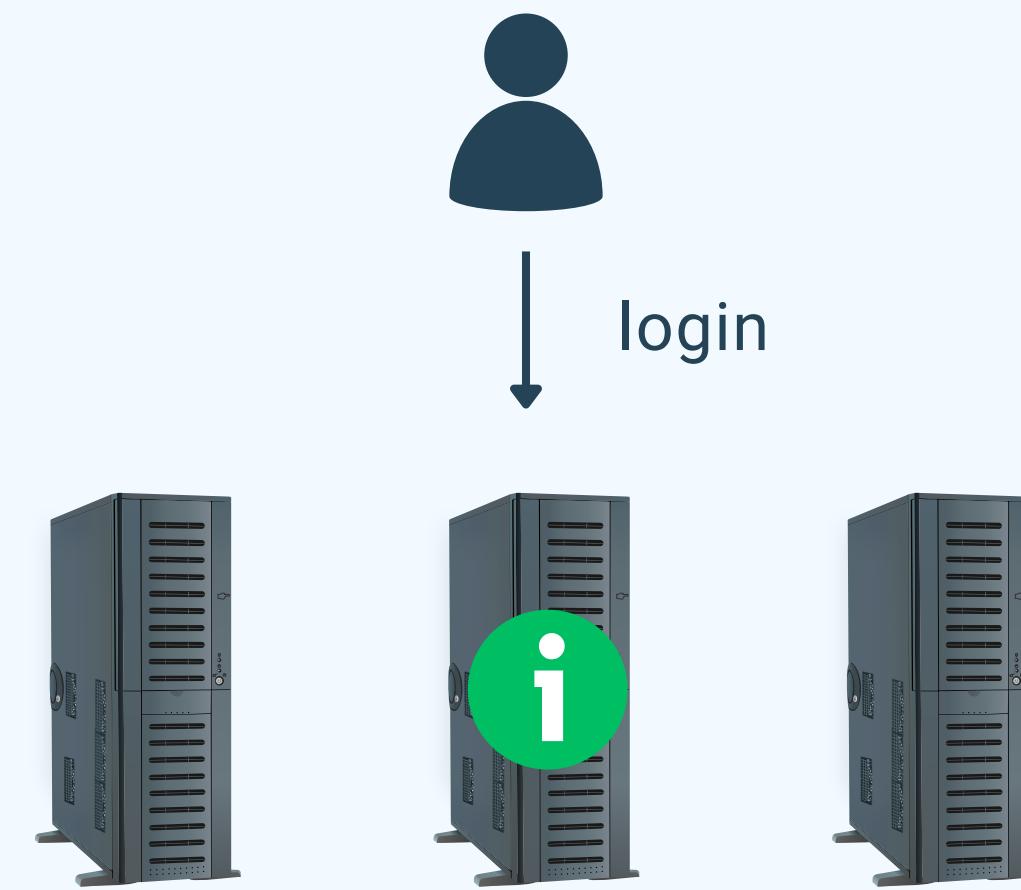
- Se almacena información en la memoria del servidor (ej. sesión de usuarios)



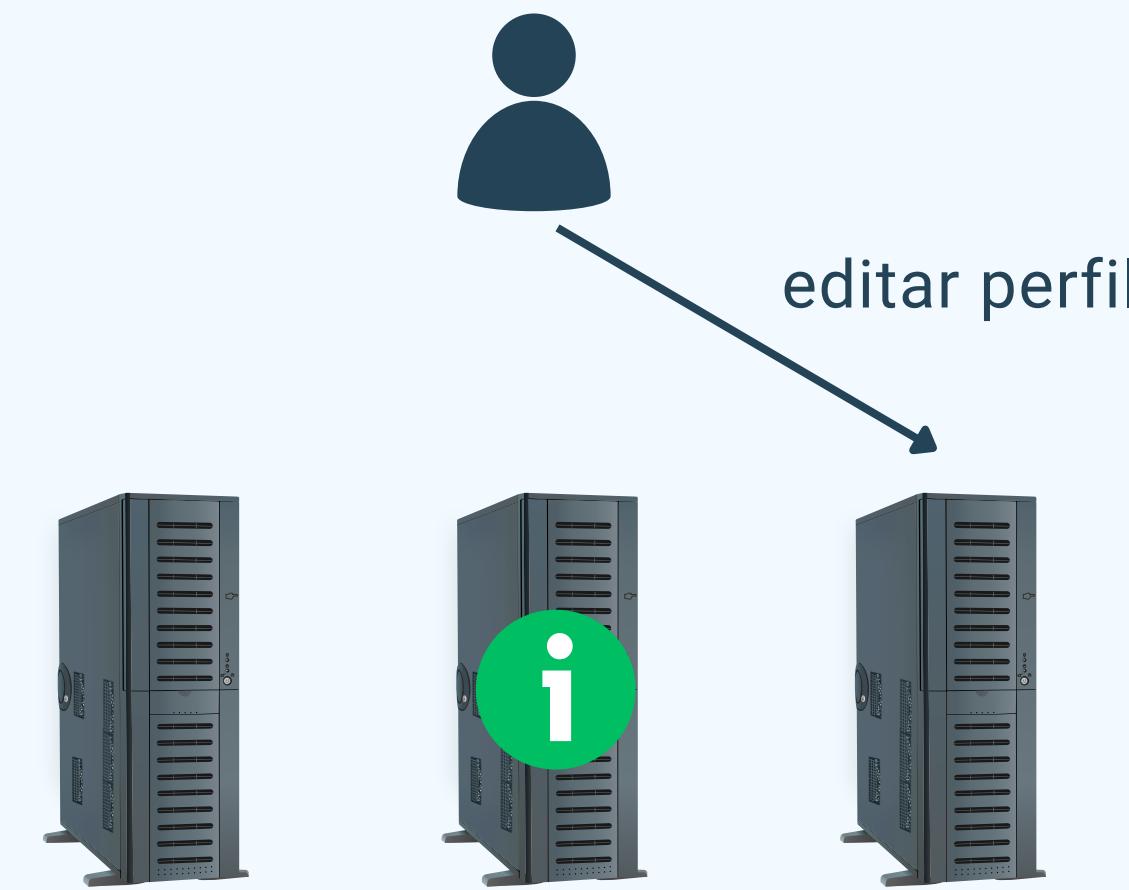
SERVICIOS STATEFUL



SERVICIOS STATEFUL



SERVICIOS STATEFUL



SERVICIOS STATEFUL

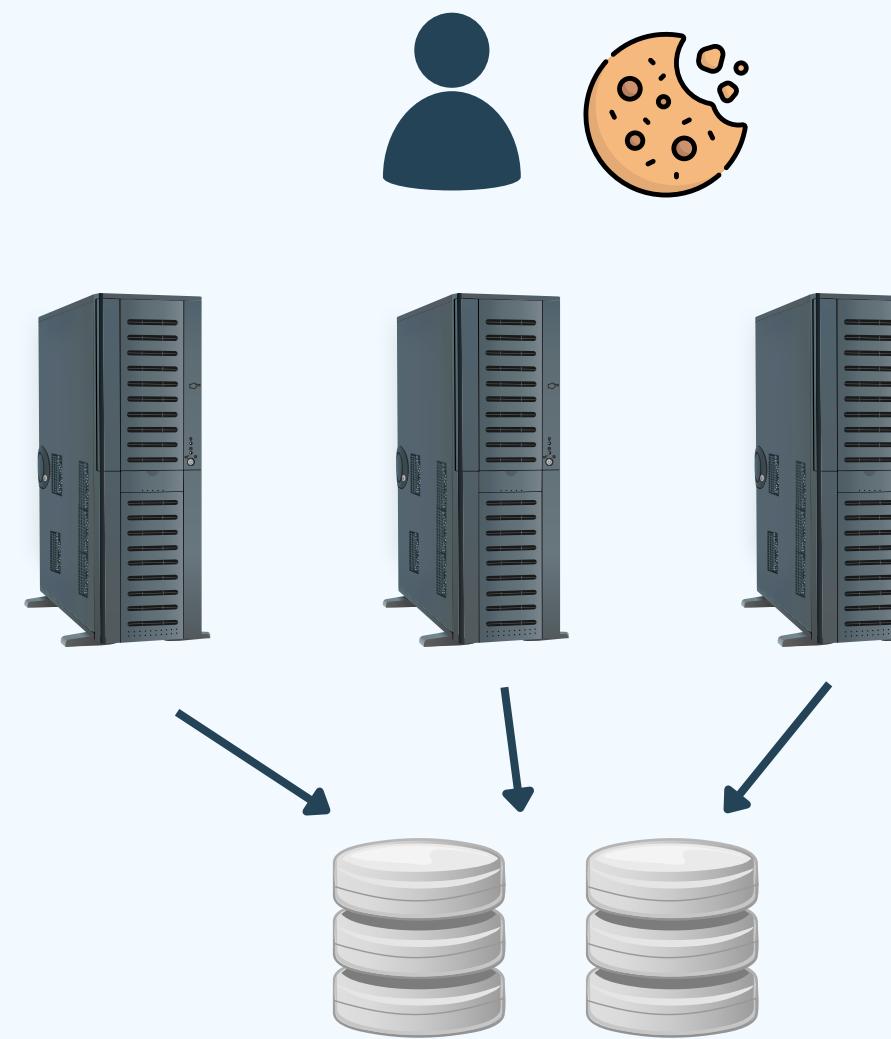


SERVICIOS STATEFUL

- Se almacena información en la memoria del servidor (ej. sesión de usuarios)
- **No escalan bien horizontalmente.**
 - Podríamos redirigir a los usuarios a un único servidor.
 - No existiría tolerancia a fallos.

SERVICIOS STATELESS

- **No** se almacena información en la memoria del servidor.
 - Se almacena en el **cliente** (cookies).
 - **Almacenes de datos.** Todos los servidores tienen acceso.

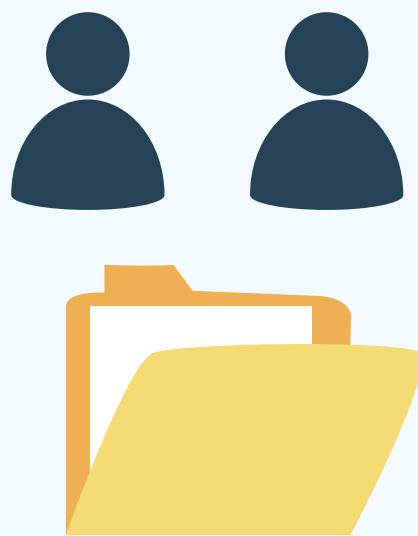


ESCALABILIDAD ORGANIZACIÓN

- Es vital poder añadir nuevos integrantes al equipo de forma rápida y sencilla.
 - Buen proceso de **onboarding**.
 - Buena **documentación**.
 - **Seguimiento** del nuevo integrante.
 - **Pair programming**.
 - Código limpio y legible.

ESCALABILIDAD ORGANIZACIÓN

- El diseño de nuestro sistema también influye.



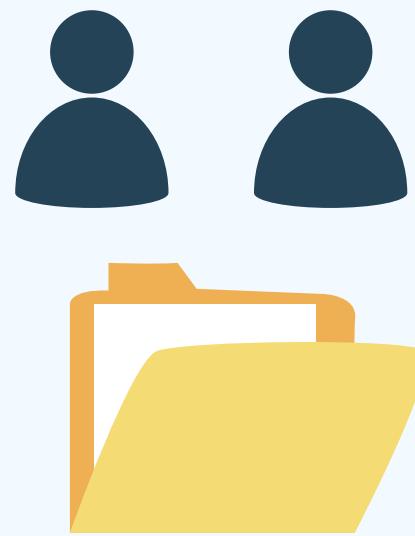
ESCALABILIDAD ORGANIZACIÓN

- El diseño de nuestro sistema también influye.



ESCALABILIDAD ORGANIZACIÓN

- El diseño de nuestro sistema también influye.



CONSIDERACIONES AL ESCALAR UN SISTEMA

- Las tres estrategias **no son excluyentes entre sí.**
 - Podemos escalar en **una, dos o tres formas a la vez.**

CONSIDERACIONES AL ESCALAR UN SISTEMA

- Las tres estrategias **no son excluyentes entre sí.**
 - Podemos escalar en **una, dos o tres formas a la vez.**
- **No hay relación lineal** directa **entre los recursos y la carga soportada.**
 - Tener el doble de recursos no significa obtener el doble de resultados.
 - Podemos llegar al punto en el que añadir recursos empeore la situación.

CONSIDERACIONES AL ESCALAR UN SISTEMA

