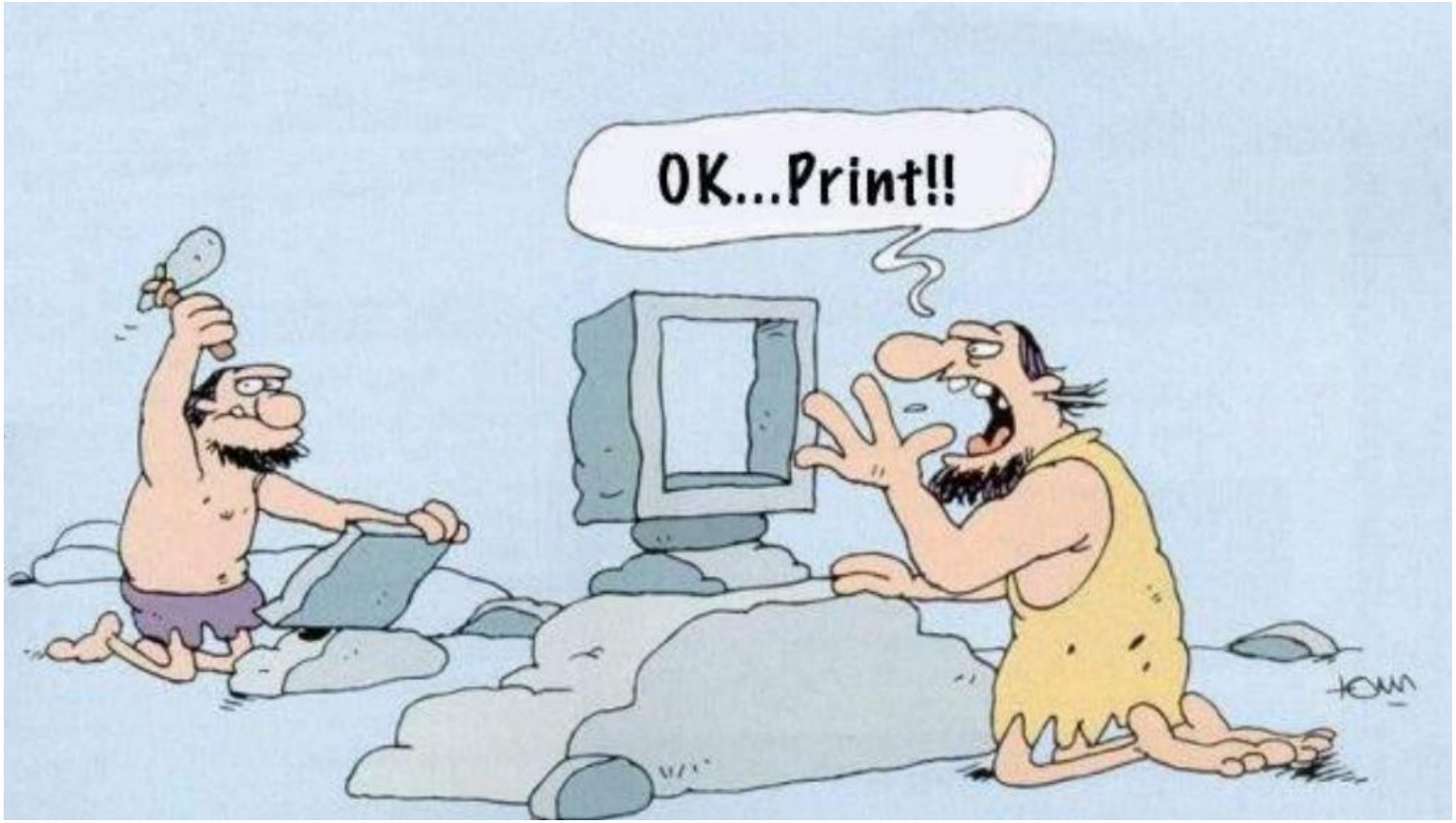
Desafío de los Sistemas Distribuidos

Sistemas Distribuidos



Operaciones bursátiles tan rápidas que dejan en desventaja a los que operan desde más lejos

Sistema Distribuido

- Un sistema distribuido se define como una colección de computadoras separadas físicamente y conectadas entre sí por una red de comunicaciones; cada máquina posee sus componentes de hardware y software que el programador percibe como un solo sistema (no necesita saber qué cosas están en qué máquinas). El programador accede a los componentes de software (objetos) remotos, de la misma manera en que accedería a componentes locales, en un grupo de computadoras que usan un middleware entre los que destacan (RPC) y SOAP para conseguir un objetivo.

Desafío de los Sistemas Distribuidos

- Heterogeneidad
- Extensibilidad
- Seguridad
- Escalabilidad
- Tolerancia a Fallas
- Concurrencia
- Transparencia

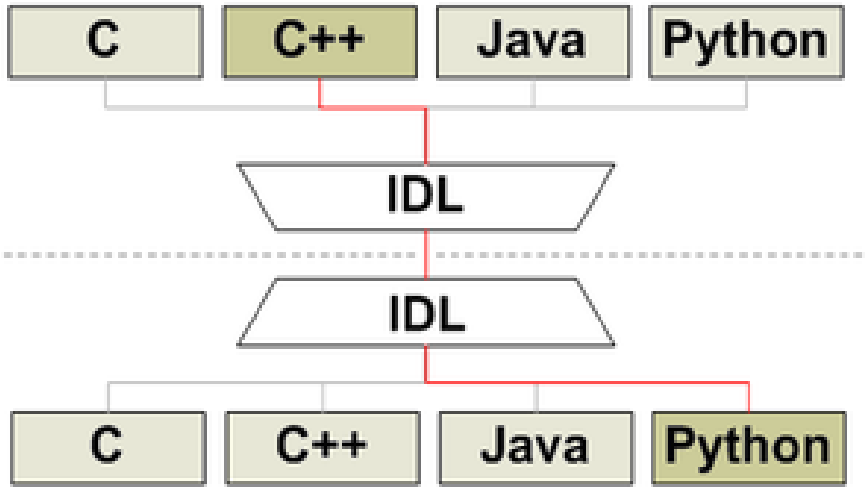
Heterogeneidad

La Heterogeneidad se aplica en los siguientes elementos:

- Redes
- Hardware de computadores
- Sistemas operativos
- Lenguajes de programación
- Implementaciones de diferentes Desarrolladores

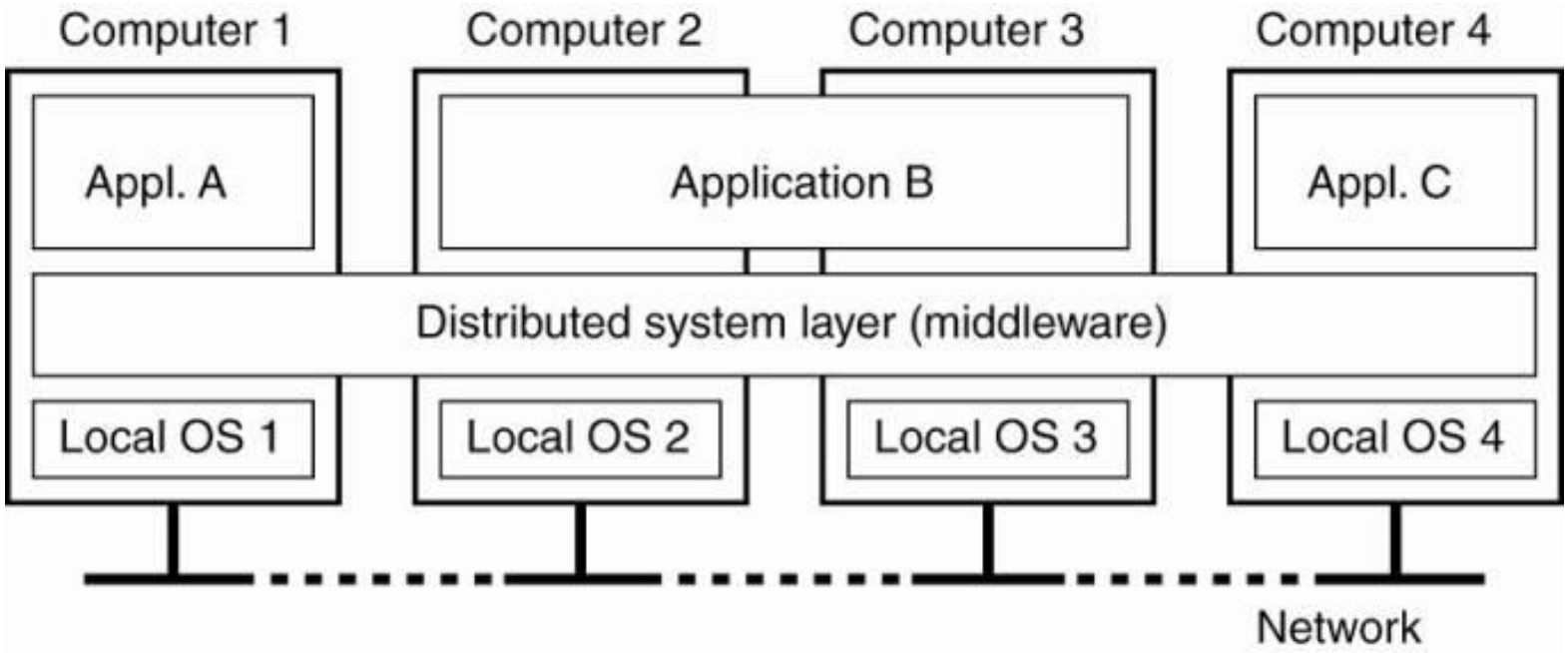
Heterogeneidad

- Middleware: es el estrato de software que provee una abstracción de programación, así como un enmascaramiento de la heterogeneidad subyacente de las redes, hardware, sistemas operativos y lenguajes de programación. Ejem: Corba, Java RMI



Heterogeneidad

- Middleware



Heterogeneidad

- Heterogeneidad y código móvil. Código que puede enviarse desde un computador a otro y ejecutarse en este último. El concepto de máquina virtual ofrece un modo de crear código ejecutable sobre cualquier hardware.

Extensibilidad

- Es la característica que determina si el sistema puede extenderse de varias maneras. Un sistema puede ser abierto o cerrado con respecto a extensiones de hardware o de software. Para lograr la extensibilidad es imprescindible que las interfaces clave sean publicadas.
- Los Sistemas Distribuidos Abiertos pueden extenderse a nivel de hardware mediante la inclusión de computadoras a la red y a nivel de software por la introducción de nuevos servicios y la re implementación de los antiguos. Otro beneficio de los sistemas abiertos es su independencia de proveedores concretos.

Seguridad

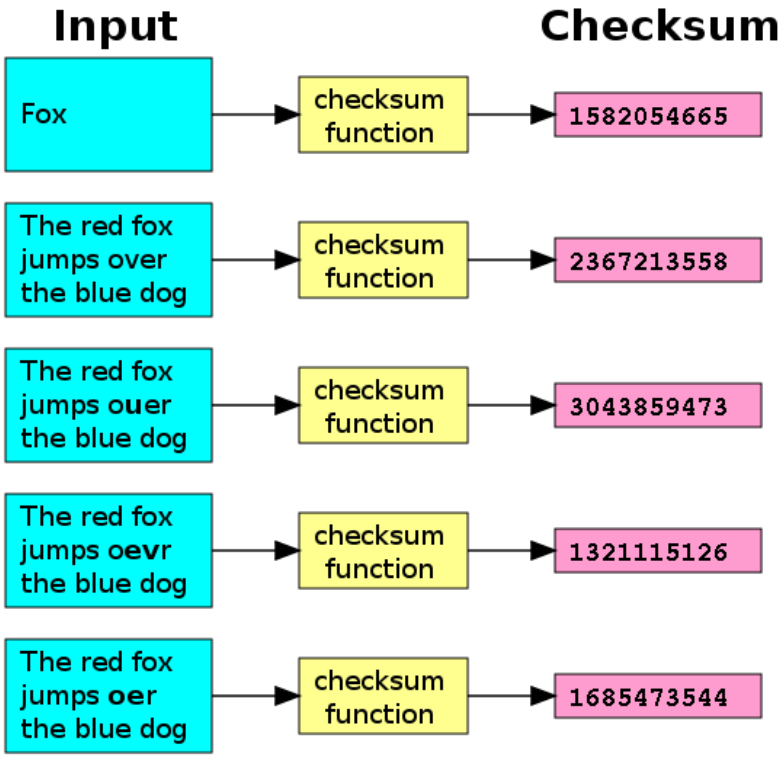
- La seguridad tiene tres componentes: Confidencialidad: protección contra individuos no autorizados,
- Integridad: protección contra la alteración o corrupción, Disponibilidad: protección contra la interferencia que impide el acceso a los recursos

Escalabilidad

- Se dice que un sistema es escalable si conserva su efectividad cuando ocurre un incremento significativo en el número de recursos y en el número de usuarios.

Tratamiento de Fallos

- Detección de fallos: Ejem. Se pueden utilizar sumas de comprobación (checksums) para detectar datos corruptos en un mensaje.



Tratamiento de Fallos

- Enmarascamiento de fallos: Ejem: Los mensajes pueden retransmitirse, Replicar los datos.
- Tolerancia de fallos: los programas clientes de los servicios pueden diseñarse para tolerar ciertos fallos. Esto implica que los usuarios tendrán también que tolerarlos.
- Recuperación de fallos: implica el diseño de software en el que, tras una caída del servidor, el estado de los datos puede reponerse o retractarse (rollback) a una situación anterior.
- Redundancia: emplear componentes redundantes.

Concurrencia

- Existe la posibilidad de acceso concurrente a un mismo recurso. La concurrencia en los servidores se puede lograr a través de threads. Cada objeto que represente un recurso compartido debe responsabilizarse de garantizar que opera correctamente en un entorno concurrente. Para que un objeto sea seguro en un entorno concurrente, sus operaciones deben sincronizarse de forma que sus datos permanezcan consistentes.

Transparencia

- Transparencia de acceso: permite acceder a los recursos locales y remotos empleando operaciones idénticas.
- Transparencia de ubicación: permite acceder a los recursos sin conocer su localización.
- Transparencia de concurrencia: permite que varios procesos operen concurrentemente sobre recursos compartidos sin interferencia mutua.
- Transparencia de replicación: permite replicar los recursos sin que los usuarios y los programadores necesiten su conocimiento.

Transparencia

- Transparencia frente a fallos: permite ocultar fallos.
- Transparencia de movilidad: permite la reubicación de recursos y clientes en un sistema sin afectar la operación de los usuarios y los programas.
- Transparencia de rendimiento: permite reconfigurar el sistema para mejorar el desempeño según varíe su carga.
- Transparencia al escalado: permite al sistema y a las aplicaciones expandirse en tamaño sin cambiar la estructura del sistema o los algoritmos de aplicación.

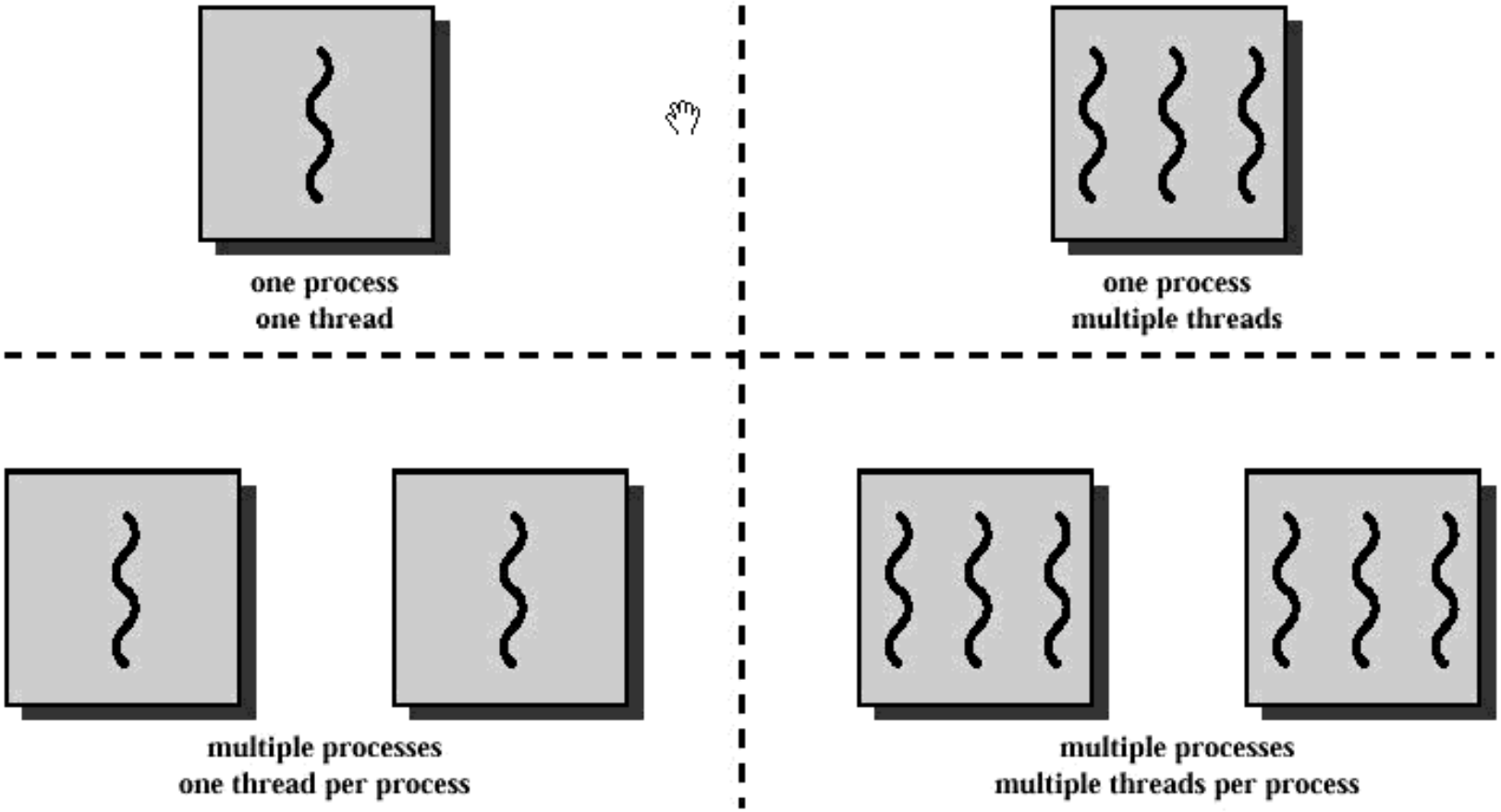
Programación Concurrente

Diferencia entre Hilos y Procesos

- Un **proceso** es un programa en ejecución. Se compone de: Instrucciones del programa: Estado de ejecución; Memoria de trabajo; Otra información para controlar su planificación.
- Un **hilo** de ejecución es la unidad de procesamiento más pequeña que puede ser planificada por un sistema operativo. Permite a una aplicación realizar varias tareas a la vez (concurrencia).

	PROCESOS	HILOS
Creación	Costosa	Ligera
Recursos y memoria	Independiente	Compartida
Comunicación	Compleja	Sencilla
Complejidad en programación	Reducida	Alta

Procesos e Hilos



Programa secuencial

- Es aquel en el que sus instrucciones se van ejecutando en serie, una tras otra, desde su inicio hasta el final. Por tanto, en un programa secuencial solo existe una única actividad o hilo de ejecución.

Programa concurrente

- Se llegaran a crear múltiples actividades que progresaran de manera simultanea. Cada una de esas actividades será secuencial pero ahora el programa no avanzará siguiendo una única secuencia pues cada actividad podrá seguir una serie de instrucciones distinta y todas las actividades avanzan de manera simultanea (o, al menos, esa es la imagen que ofrecen al usuario).
- Para que en un determinado proceso haya múltiples actividades, cada una de esas actividades estará soportada por un *hilo de ejecución*. *Para que* un conjunto de actividades constituya un programa concurrente, todas ellas deben cooperar entre si para realizar una tarea común.

Concurrencia

- Paralelismo real : Se dará cuando tengamos disponibles múltiples procesadores, de manera que ubicaremos a cada actividad en un procesador diferente.
- En las arquitecturas modernas, los procesadores pueden tener múltiples núcleos. En ese caso, basta con asignar un núcleo diferente a cada actividad. Todas ellas podrán avanzar simultáneamente sin ningún problema.
- Si solo disponemos de ordenadores con un solo procesador y un único núcleo, podremos todavía lograr la concurrencia real utilizando múltiples ordenadores.

Concurrencia

- Paralelismo lógico: Como los procesadores actuales son rápidos y las operaciones de E/S suspenden temporalmente el avance del hilo de ejecución que las haya programado, se puede intercalar la ejecución de múltiples actividades y ofrecer la imagen de que éstas progresan simultáneamente.
- El usuario sera incapaz de distinguir entre un sistema informático con un solo procesador y un solo núcleo de otro que tenga múltiples procesadores, pues los sistemas operativos ocultan estos detalles.

Ventajas

- Eficiencia: El hecho de disponer de múltiples actividades dentro de la aplicación permite que esta pueda concluir su trabajo de manera más rápida.
- Escalabilidad : Si una determinada aplicación puede diseñarse e implantarse como un conjunto de componentes que interactúen y colaboren entre si, generando al menos una actividad por cada componente, se facilitara la distribución de la aplicación sobre diferentes ordenadores.
- Gestión de las comunicaciones: El uso eficiente de los recursos, ya comentado en la primera ventaja citada en este apartado, permite que aquellos recursos relacionados con la comunicación entre actividades sean explotados de manera sencilla en una aplicación concurrente.

Ventajas

- Flexibilidad: Las aplicaciones concurrentes suelen utilizar un diseño modular y estructurado, en el que se presta especial atención a que es lo que debe hacer cada actividad, que recursos requerir 'a y que módulos del programa necesitara ejecutar
- Menor hueco semántico: Un buen numero de aplicaciones informáticas requieren el uso de varias actividades simultaneas (por ejemplo, en un videojuego suele utilizarse un hilo por cada elemento móvil que haya en la pantalla; en una hoja de calculo tenemos un hilo para gestionar la entrada de datos, otro para recalculo, otro para la gestión de los menús, otro para la actualización de las celdas visibles, ...).

Inconvenientes

- Programación delicada: Durante el desarrollo de aplicaciones concurrentes pueden llegar a surgir algunos problemas. "condición de carrera" y puede generar inconsistencias imprevistas en el valor de las variables o atributos compartidos entre las actividades, cuando estas los modifiquen. Un segundo problema son los interbloqueos
- Depuración compleja: Una aplicación concurrente, al estar compuesta por múltiples actividades, puede intercalar de diferentes maneras en cada ejecución las sentencias que ejecuten cada una de sus actividades.

Aplicaciones reales

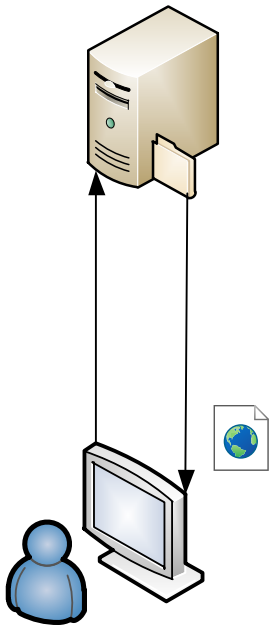
- Programa de control del acelerador lineal*. Los aceleradores* lineales se utilizan en los tratamientos de radioterapia para algunos tipos de cáncer, eliminando mediante ciertos clases de radiación (radiación de electrones, rayos X y rayos gamma, principalmente) a las células cancerígenas.
- Servidor web Apache*. Apache es un servidor web que emplea programación* concurrente. El objetivo de un servidor web es la gestión de cierto conjunto de paginas web ubicadas en ese servidor. En un servidor de este tipo interesa tener múltiples hilos de ejecución, de manera que cada uno de ellos pueda atender a un cliente distinto, paralelizando asi la gestion de multiples clientes.

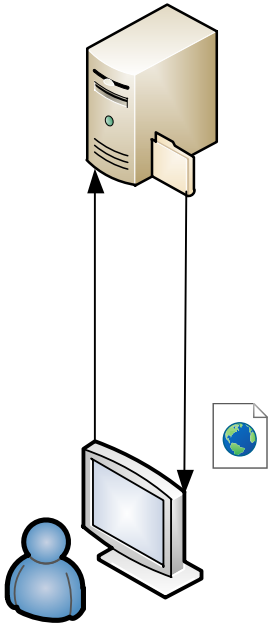
Aplicaciones

- Videojuegos. La mayoría de los videojuegos actuales (tanto para ordenadores personales como para algunas consolas) se estructuran como aplicaciones concurrentes compuestas por múltiples hilos de ejecución.
- Navegador web. Cuando Google presento su navegador Chrome (en diciembre de 2008), su arquitectura] era muy distinta a la del resto de navegadores web (Microsoft Internet Explorer, Mozilla Firefox, Opera, ...). En Chrome, cada pestaña abierta esta soportada por un proceso independiente.

Librería para Servidor Web

Interpreta la accion01 y ejecuta una funcion de la librería que esta en el Servidor Web responde el resultado mediante una pagina web





http://pagina.html

Petición de un pagina web

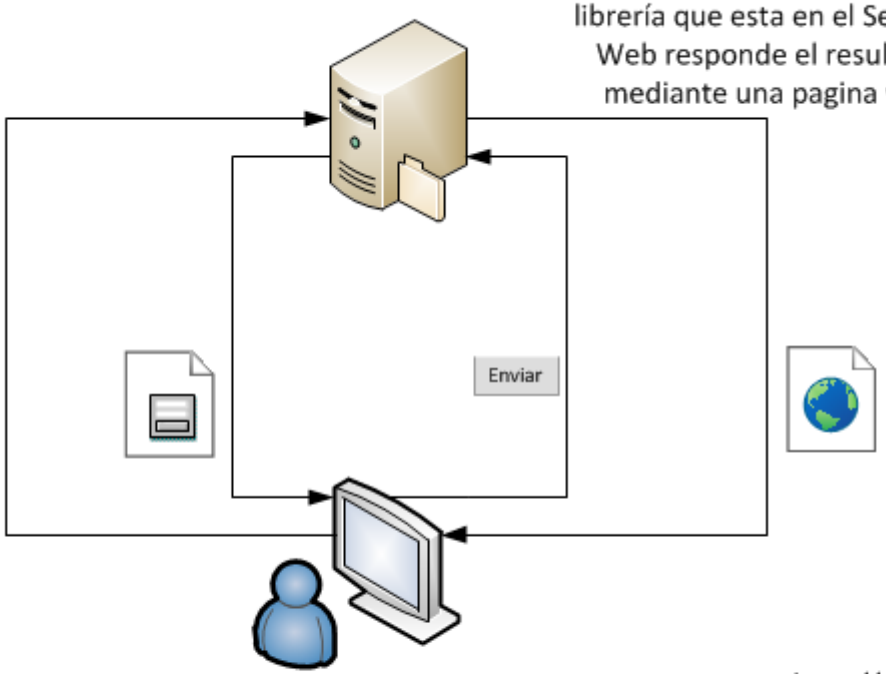
http://accion01

Petición de una funcionalidad de la librería

Devuelve una pagina respuesta01.html

Librería para Servidor Web

Interpreta la accion02 con los parametros de name y price y ejecuta una funcion de la librería que esta en el Servidor Web responde el resultado mediante una pagina web



http://form01.html

Petición de un pagina web

Recepciona form01.html la que tiene un formulario

http://accion02?name=texto&price=12.2

Se envía parámetros a la librería con el parámetro accion02

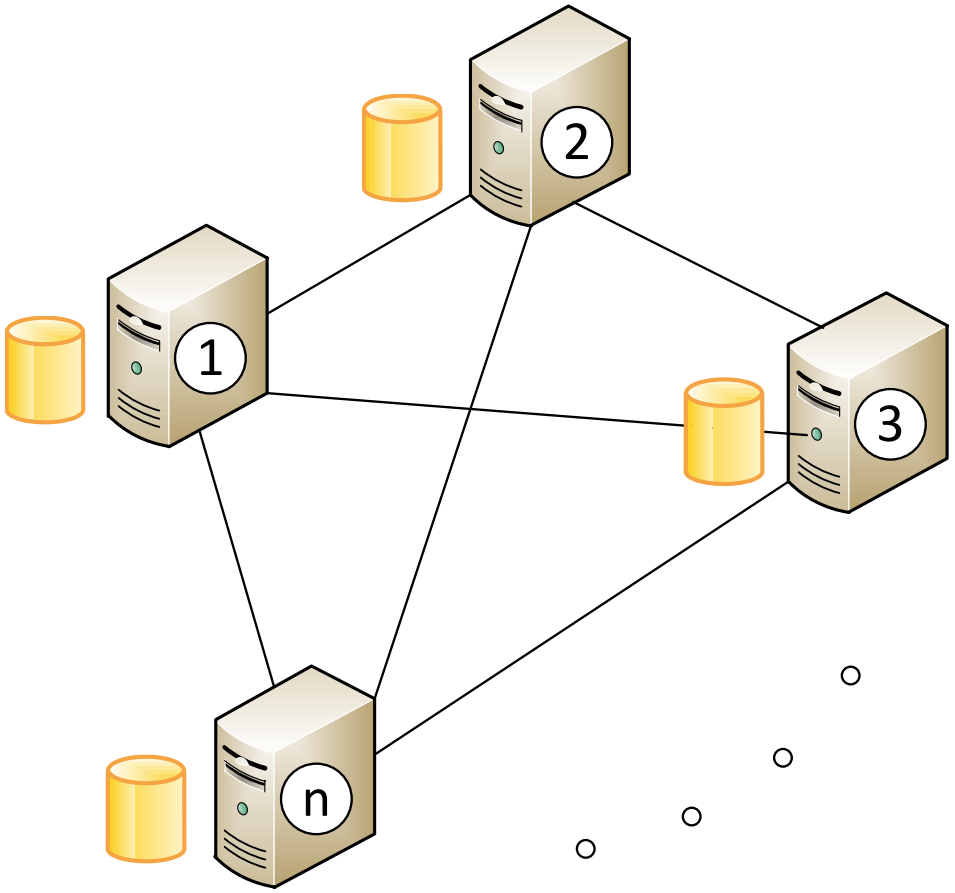
http://respuesta02.html

La accion02 responde una pagina web respuesta01.html que da el resultado

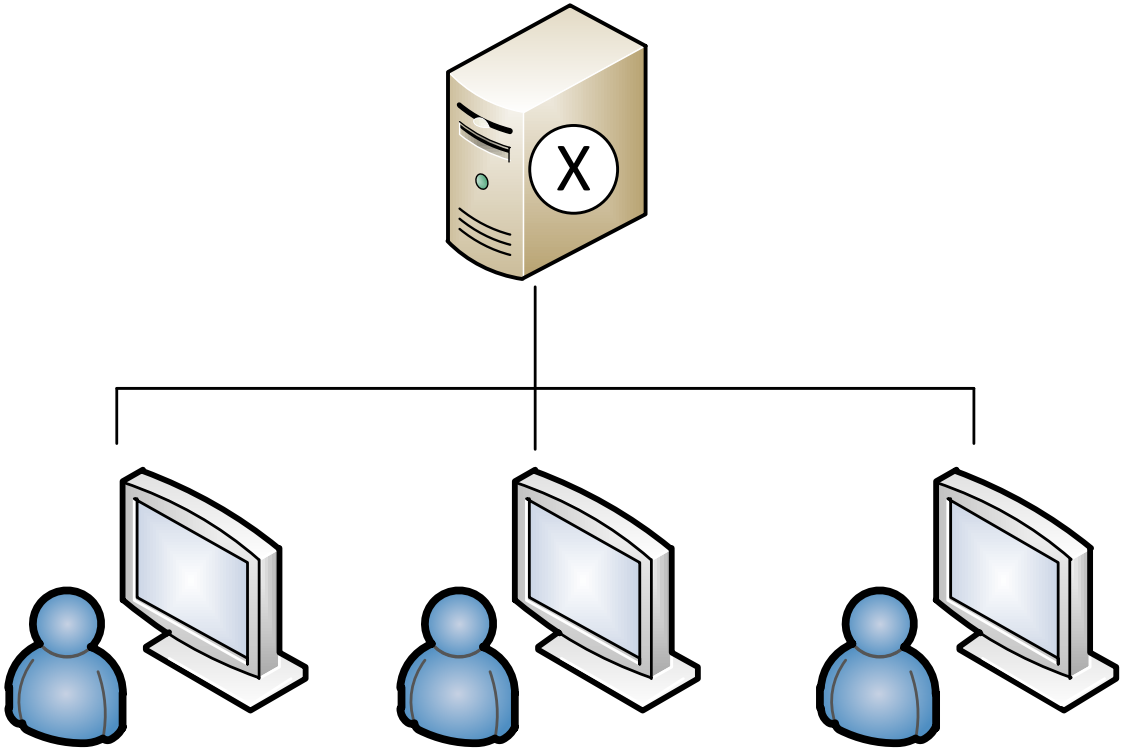
form01.html

```
<html>
<body>
<form action="accion02" >
Nombre: <input type="text" name="name"><br>
Precio: <input type="text" name="price"><br>
<input type="submit" value="Enviar" />
</form>
</body>
</html>
```

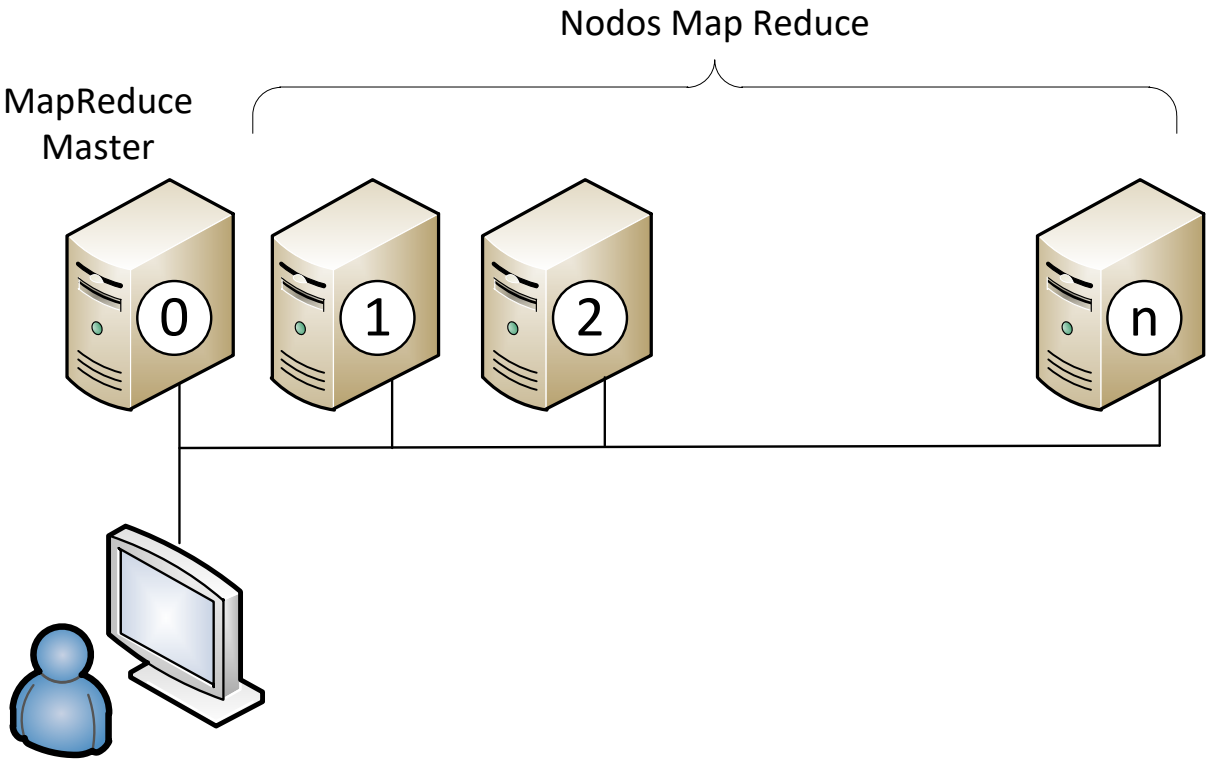
Base de Datos de Alta Disponibilidad



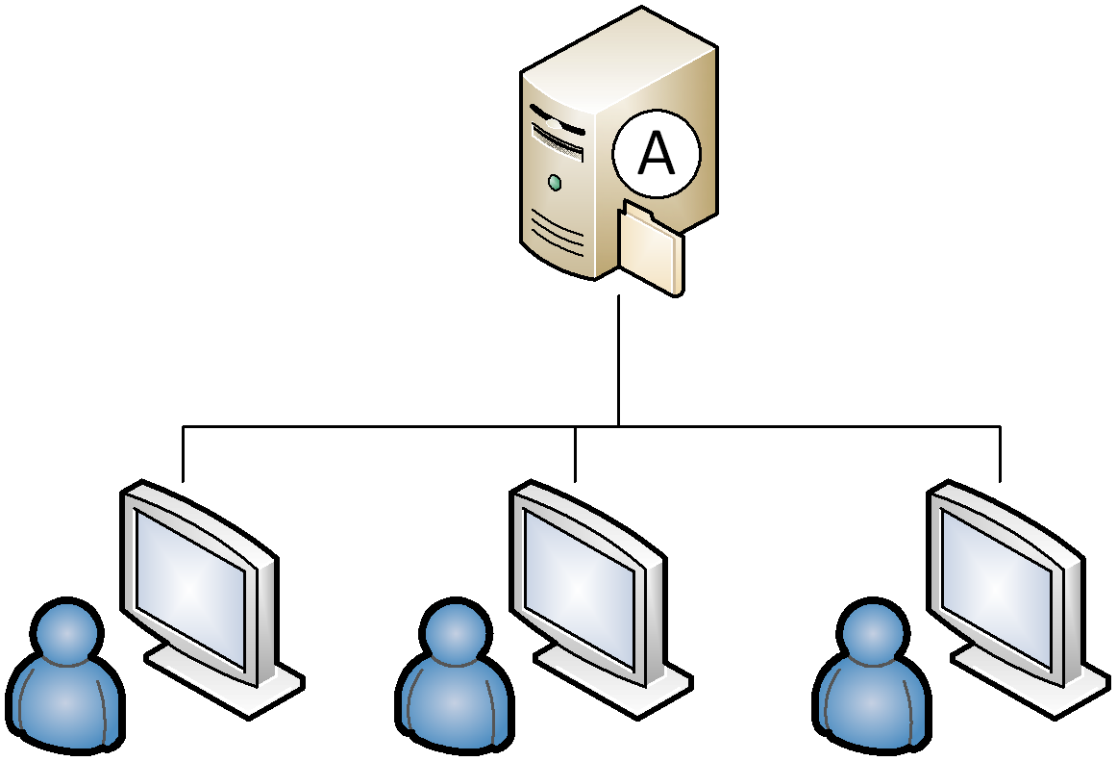
Base de Datos de Alta Disponibilidad



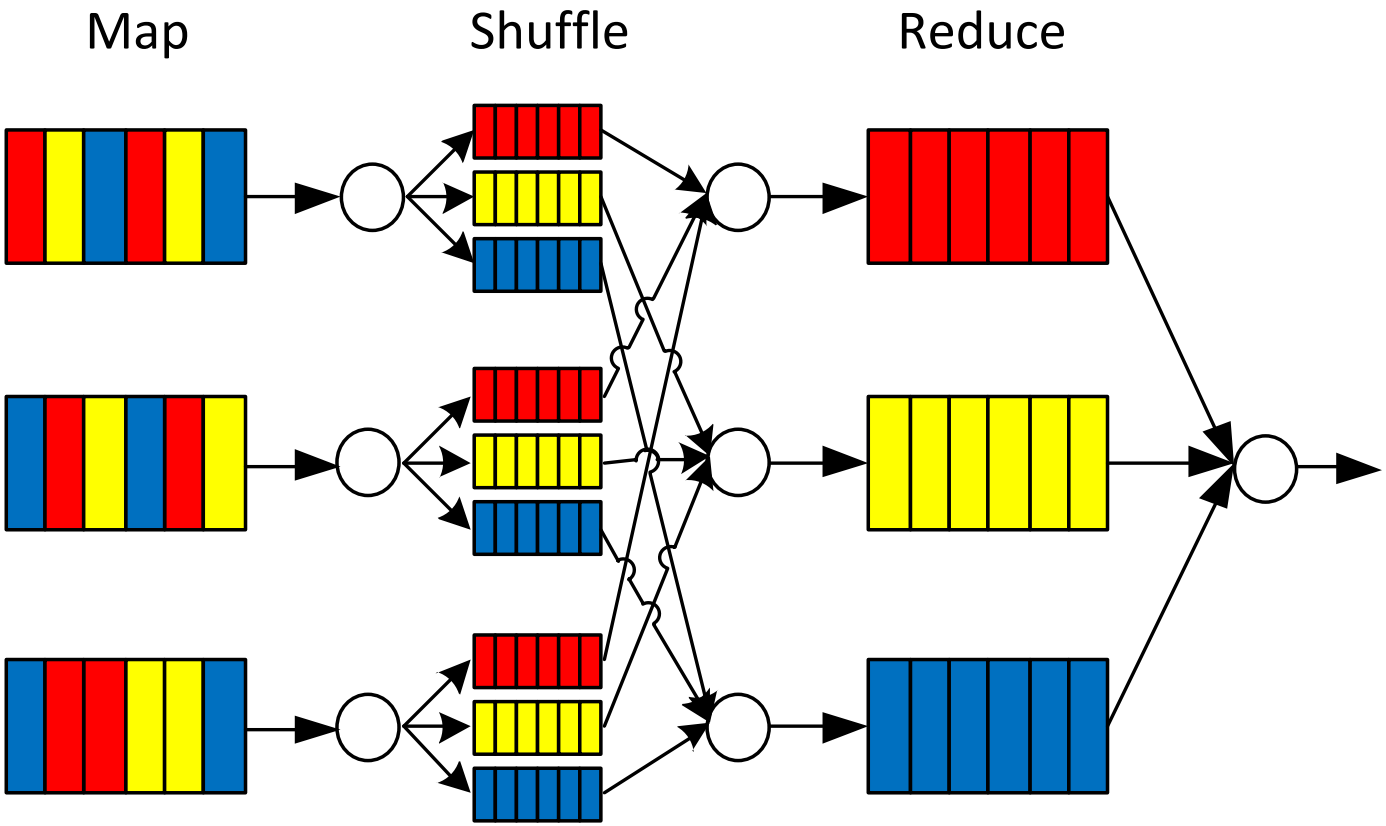
MapReduce



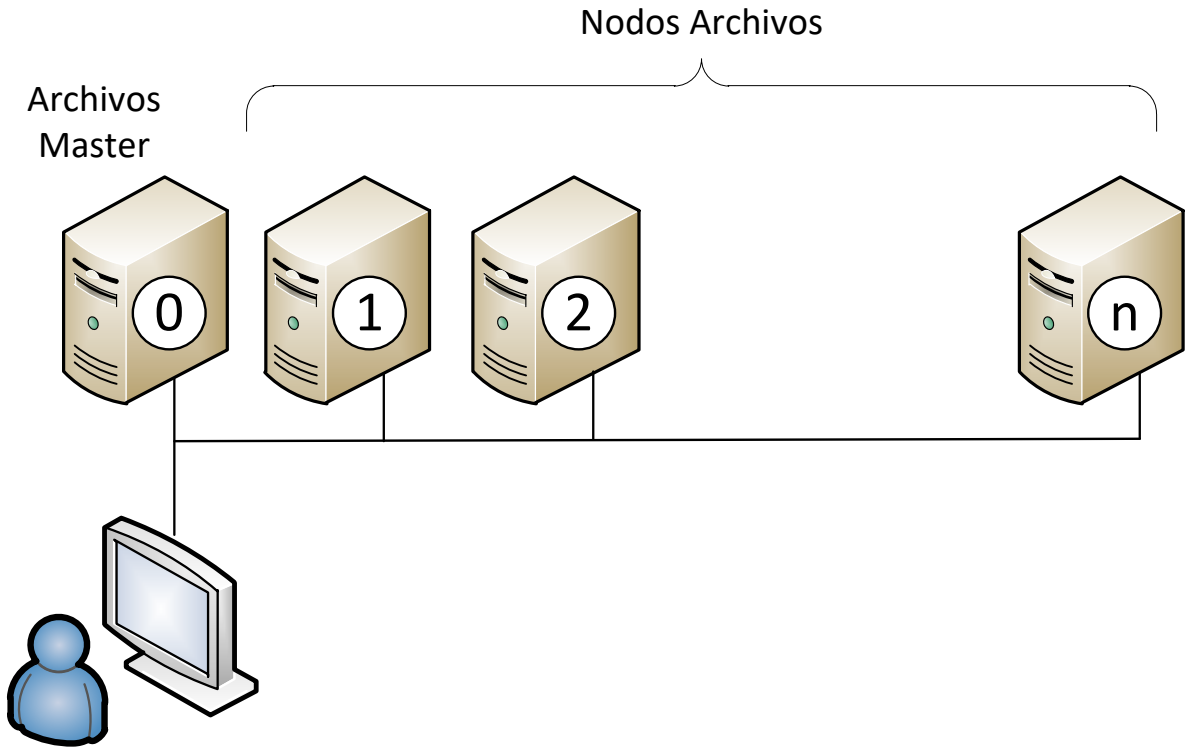
MapReduce



MapReduce

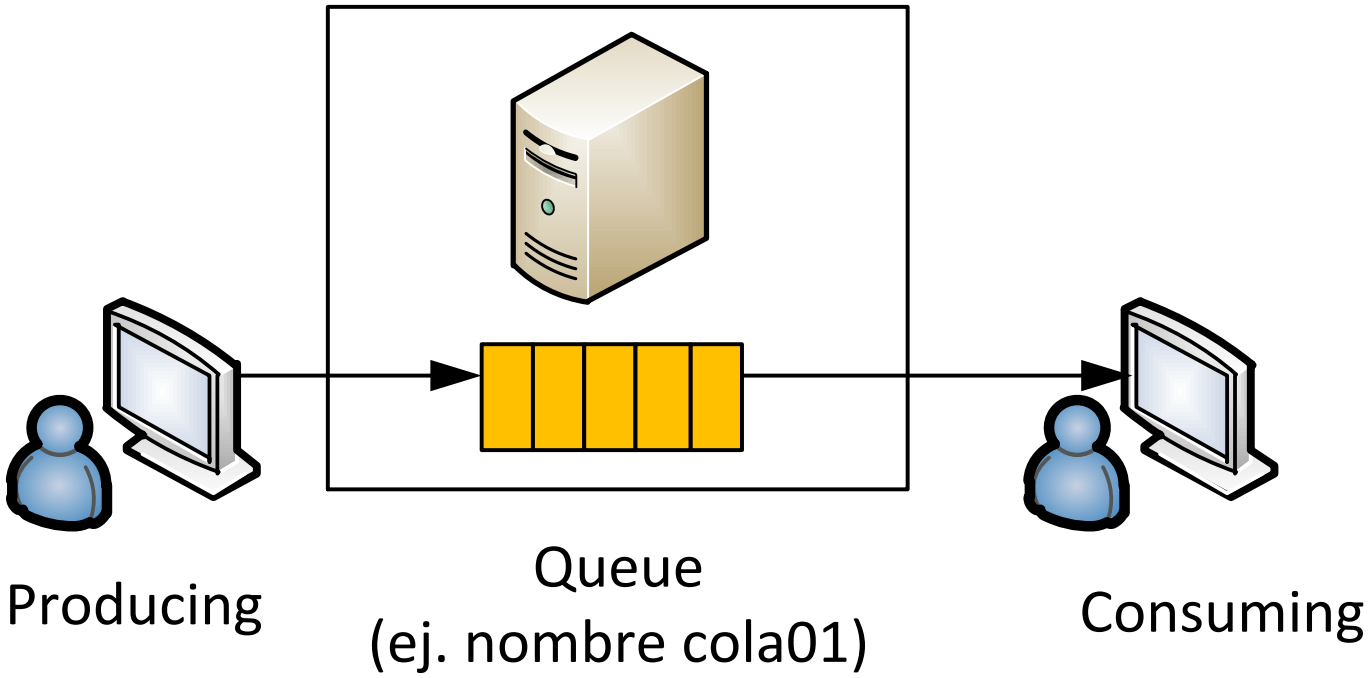


El Servidor de Archivos, Correos y Contraseñas de alta Disponibilidad



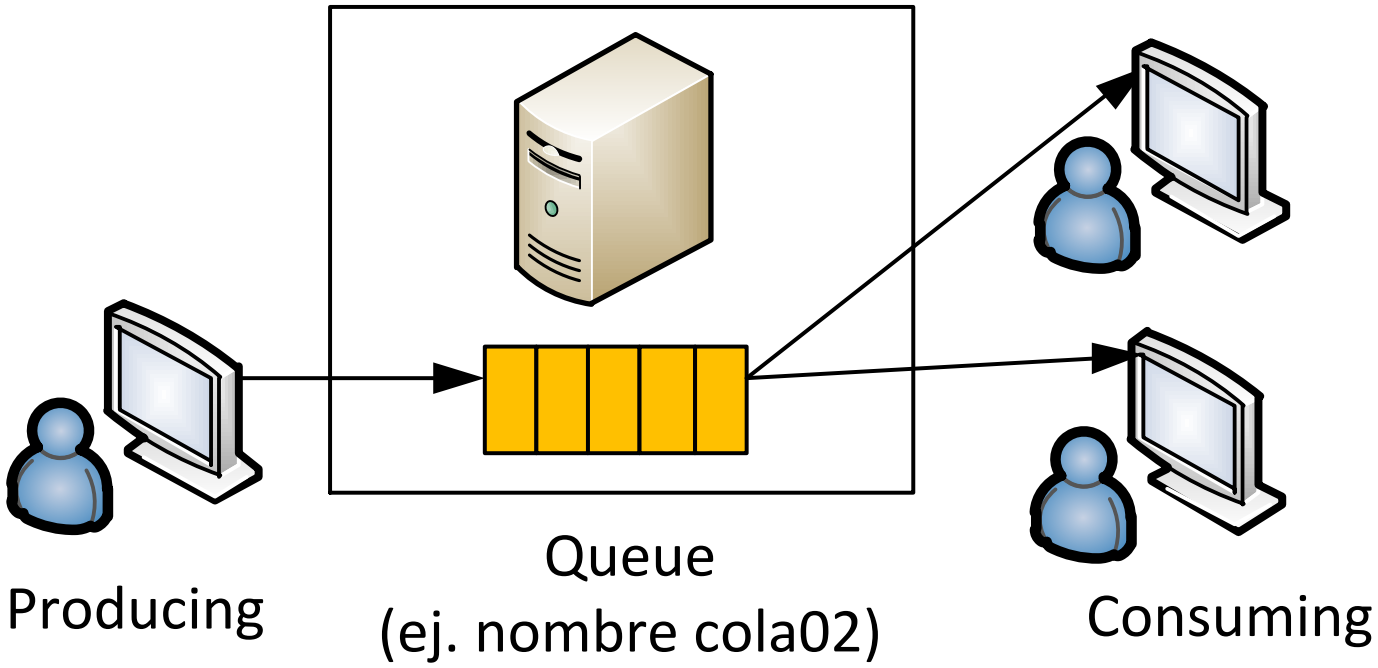
Librería MQ (Message Queue)

Simple Producer y Consuming



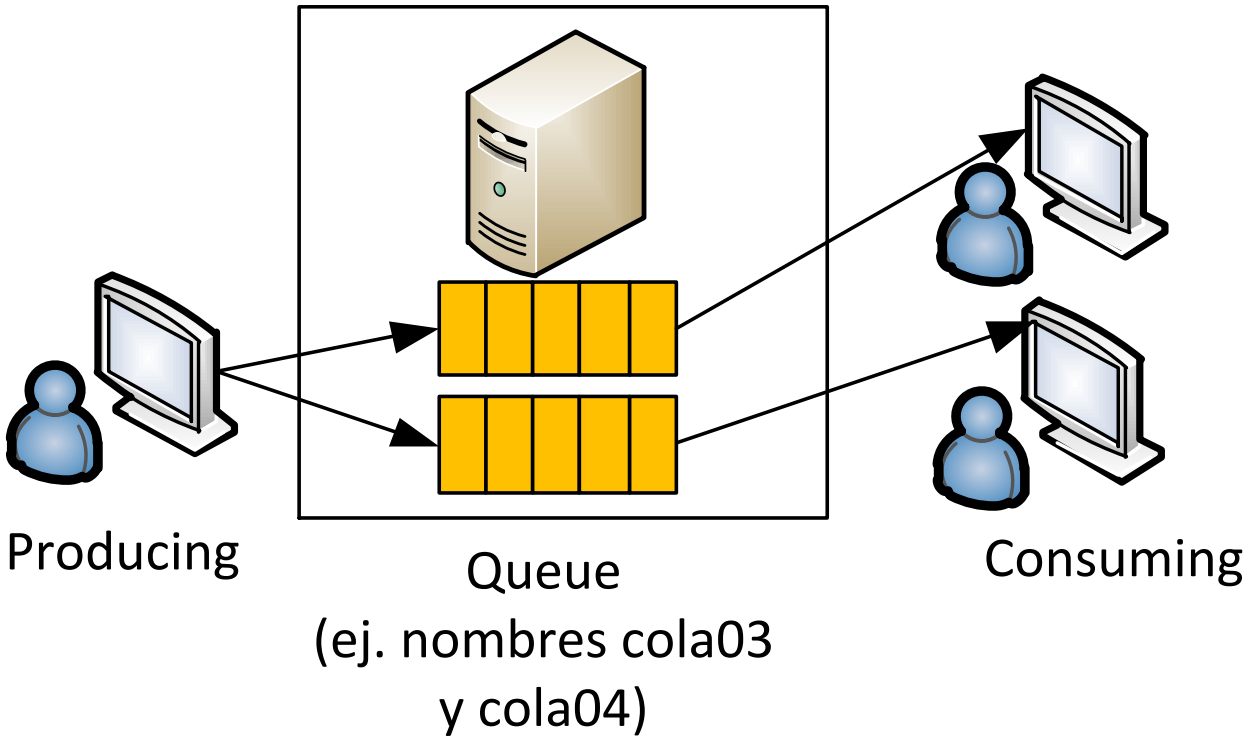
Librería MQ (Message Queue)

Simple Producer y Varios Consuming

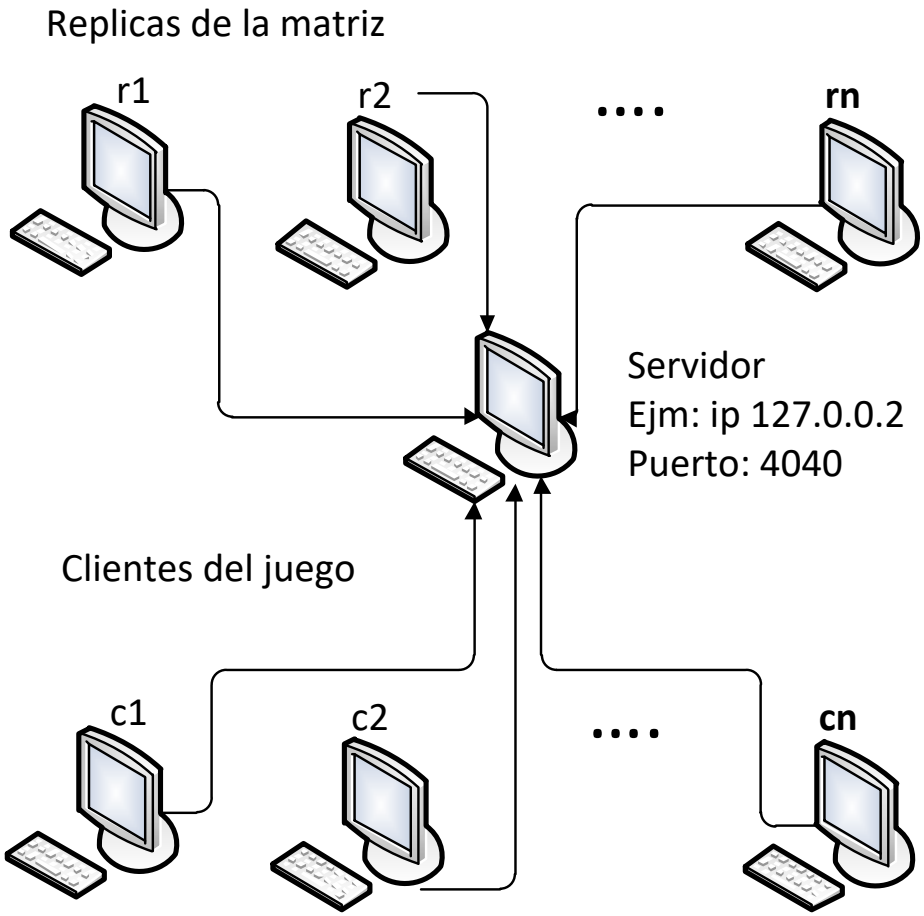


Librería MQ (Message Queue)

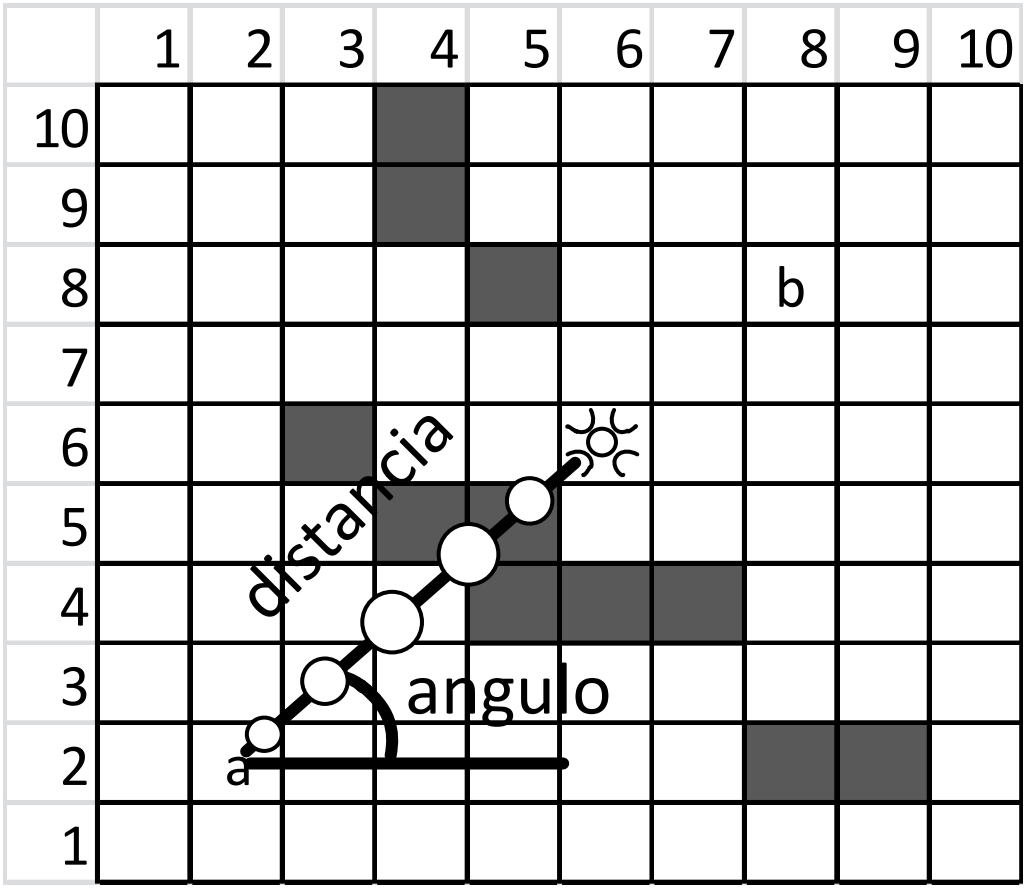
Simple Producer varios Queue y varios Consuming



Juego Desktop



Juego Desktop



Punto de Ventas en Movil

recibo

idr	clir	total	idcodbanco	estado
1	juan ramirez	320	5001	1
2	rosa rodriguez	955	9100	2
3	pepe gonzales	125	2002	2
4	mirian gomez	440	0	0
5	roberto wu		0	0

detallerecibo

idr	idp	numdr
1	2	5
1	4	10
2	3	15
2	4	5
2	5	20
3	1	5
4	2	20

Juego 2D y 3D Movil