



# Sistemas Distribuidos I (75.74)

## Introducción a Sistemas Distribuidos

Tendencias. Definición. Áreas de Aplicación. Características.  
Virtualización.

### Docentes

- Pablo D. Roca
- Ezequiel Torres Feyuk

- Ana Czarnitzki
- Cristian Raña

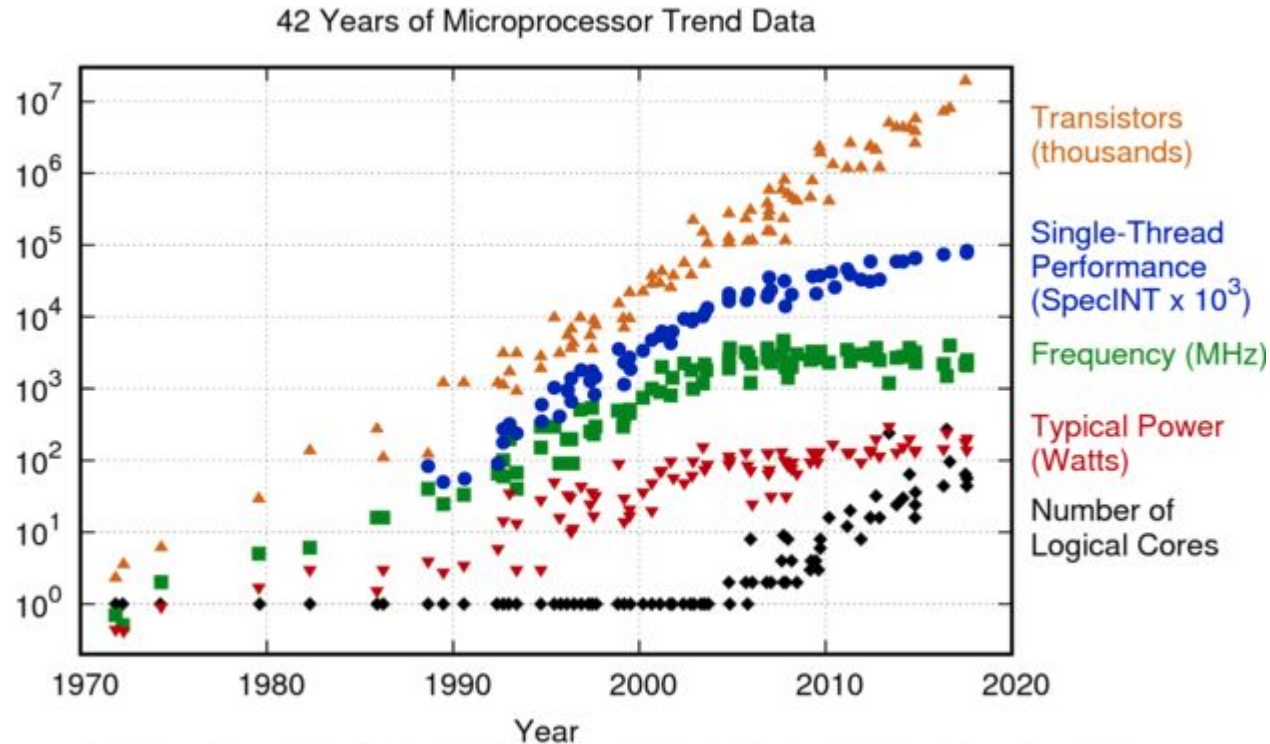
# Agenda



- Tendencias de Hardware y Software
- Definición de Sistemas Distribuidos
- Características Sistemas Distribuidos
- Virtualización



# Tendencias | La ley de Moore no nos garantiza *performance*

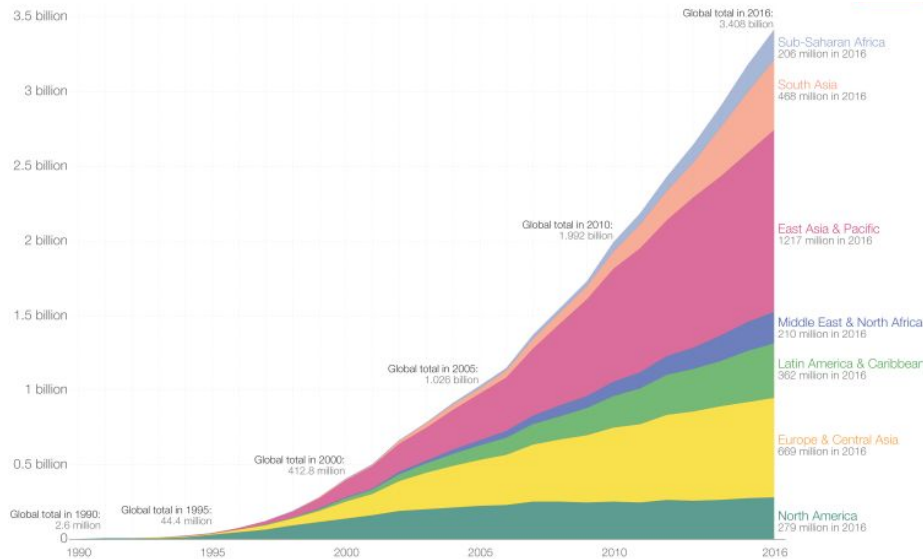


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2017 by K. Rupp

# Tendencias | Internet como medio de integración indiscutido

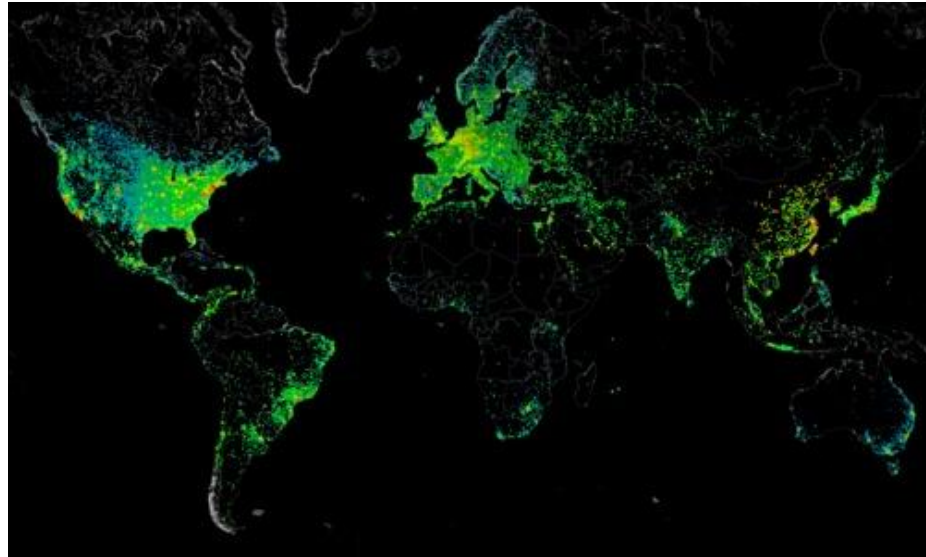


## Internet Users by Region



Source: OurWorldInData.org/internet, 2018

## Carna Botnet Census (2012): ~460M IPs

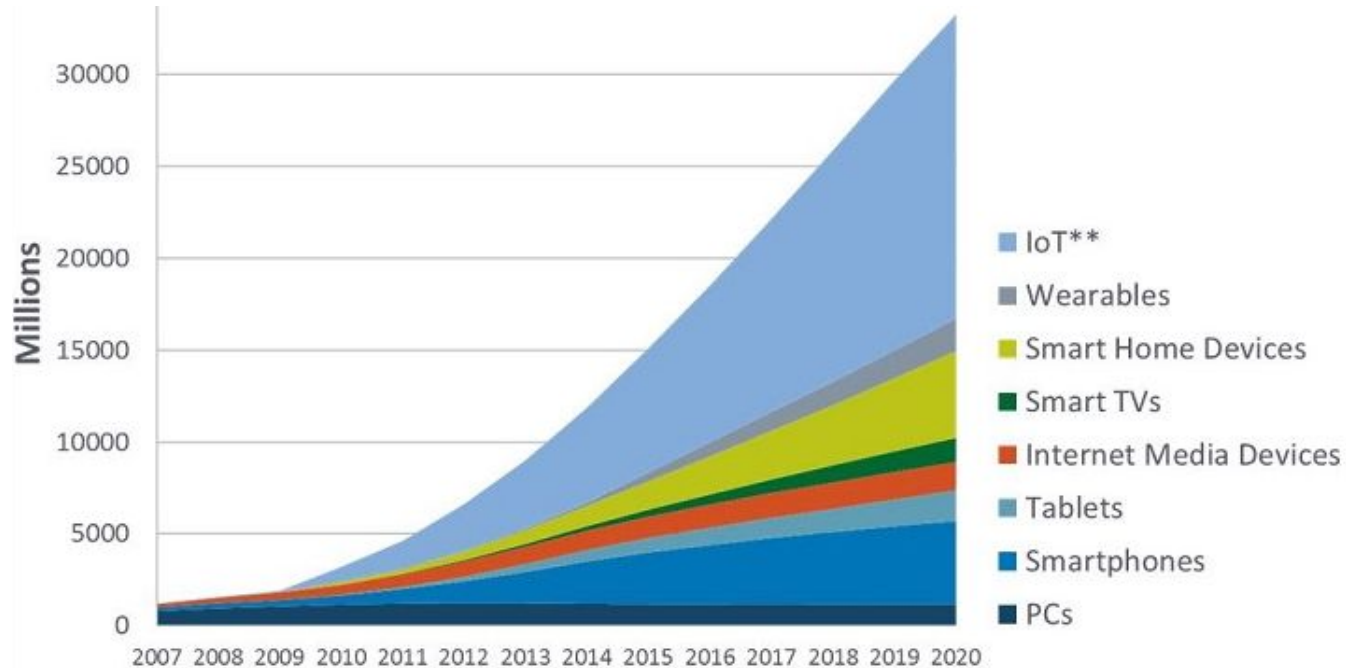


Source: Wikipedia, CarnaBotnet, <http://internetcensus2012.bitbucket.org/paper.html>

# Tendencias | Las PCs ya no conducen el movimiento...



Forecast of Installed Internet Devices



Source: Strategic Analytics, Oct. 2014



Cobran gran relevancia debido al constante crecimiento del nivel de integración e interdependencia entre sistemas, volúmenes de datos a procesar, capacidades de cómputo, paralelismo y virtualización.

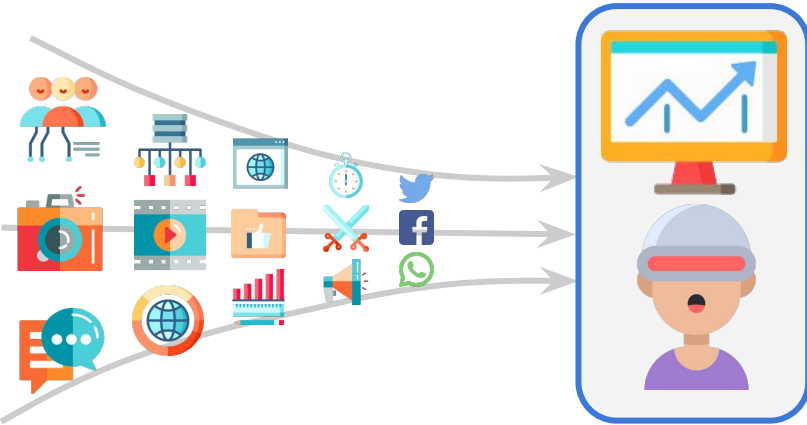
Se encuentra presente en:

- Big Data
- Data Analytics
- Scalable Architectures
- Elastic Architectures
- Machine Learning
- Internet of Things (IoT)
- Wearables
- High Performance Computer

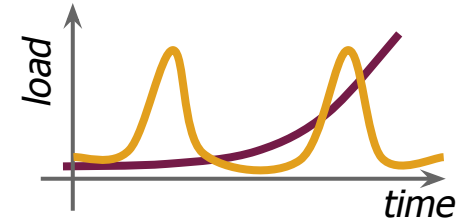


# Distribuidos | Áreas de Aplicación (I)

## Big Data, Analytics & Machine Learning

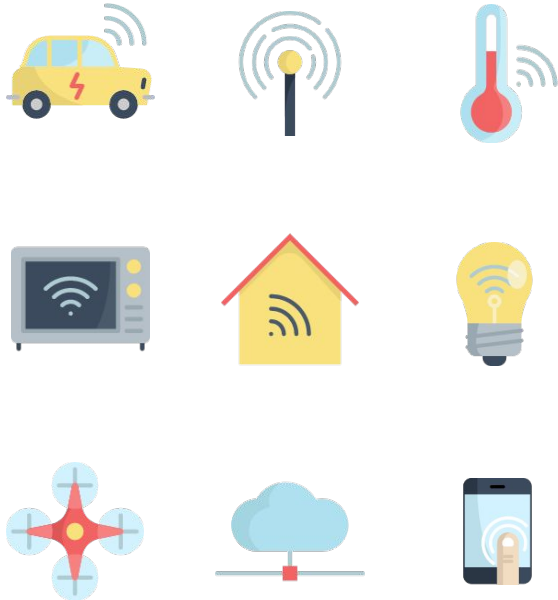


## Elastic & Scalable Apps





## Internet of Things (IoT)



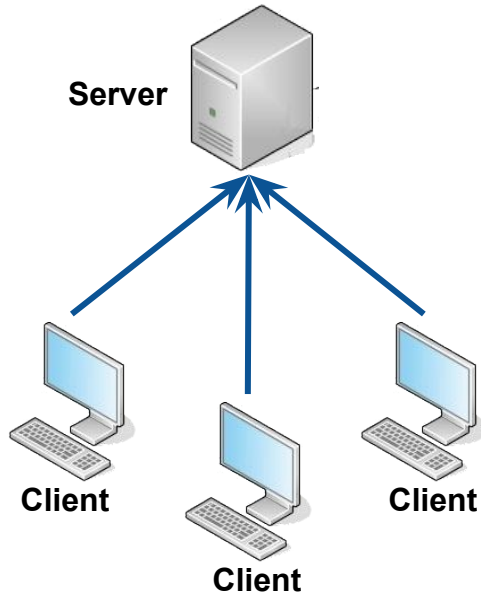
## Wearables



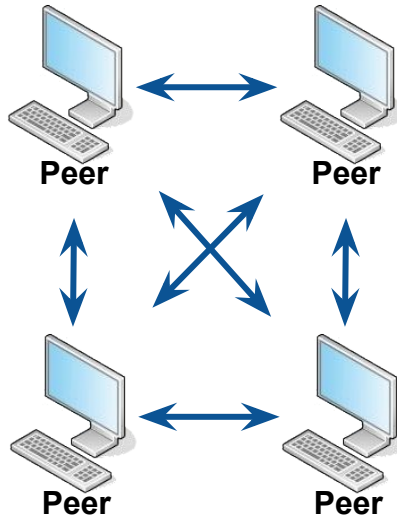




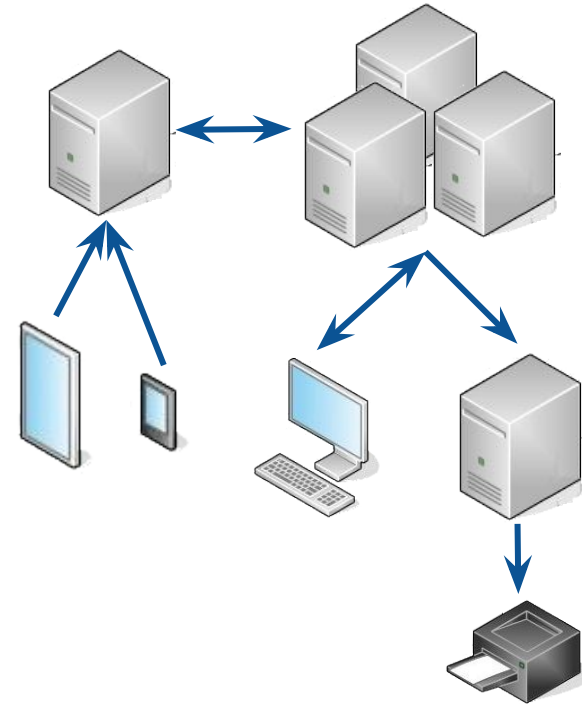
## Client - Server



## peer-to-peer (p2p)



## Heterogéneos



# Agenda



- ☐ Tendencias de Hardware y Software
- ☒ Definición de Sistemas Distribuidos
- ☐ Características Sistemas Distribuidos
- ☐ Virtualización



# Distribuidos | Definiciones de Sistemas Distribuidos

“Colección de computadoras independientes que el usuario ve como un solo sistema coherente” (Tanenbaum)

“Es un sistema de computadoras interconectadas por una red que se comunican y coordinan sus acciones intercambiando mensajes” (Coulouris)

**“Aquel en el que el fallo de un computador que ni siquiera sabes que existe, puede dejar tu propio computador inutilizable” (Lamport)**



# Distribuidos | Definición (II)

Desglose de las definiciones:

- Colección de computadoras => multiprogramación
- Independientes => autónomos
- Un solo sistema => el usuario no conoce su distribución
- Interconectadas por red => sistemas aislados no son distribuidos
- Comunican y coordinan acciones => colaborativos
- Intercambiando mensajes => protocolos de comunicación
- Fallo de un computador => nuevos problemas no determinísticos



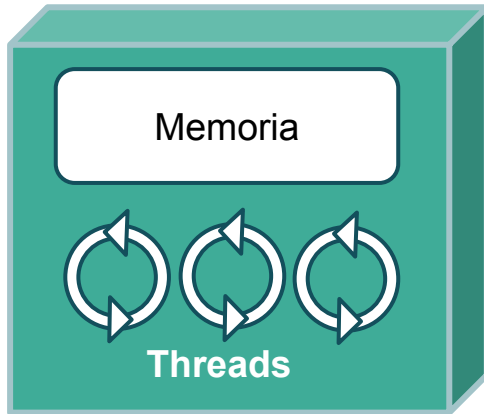
# Distribuidos | Parámetros de Diseño

- Transparencia
- Escalabilidad
- Tolerancia a Fallos
  - *Availability, Reliability, Safety, Maintainability*
- Acceso a Recursos Compartidos
- Sistemas distribuidos abiertos
  - *Interfaces, Interoperability, Portability*

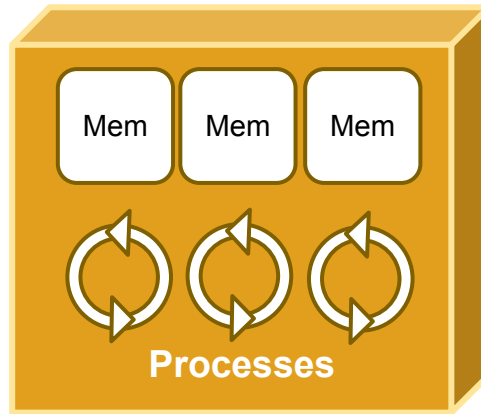


# Distribuidos | Variantes de Multiprogramación

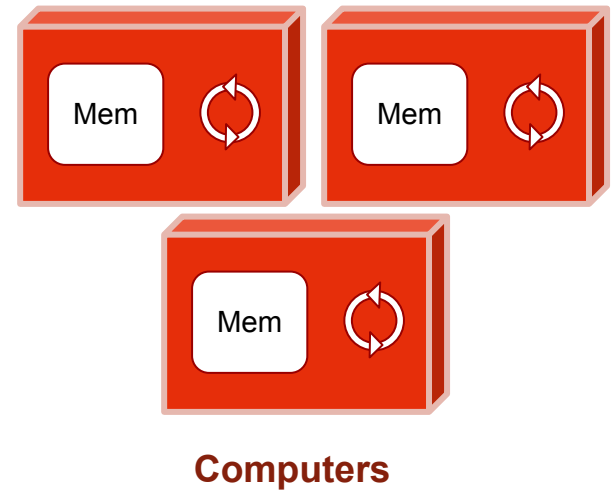
## Multi-threading



## Multi-processing

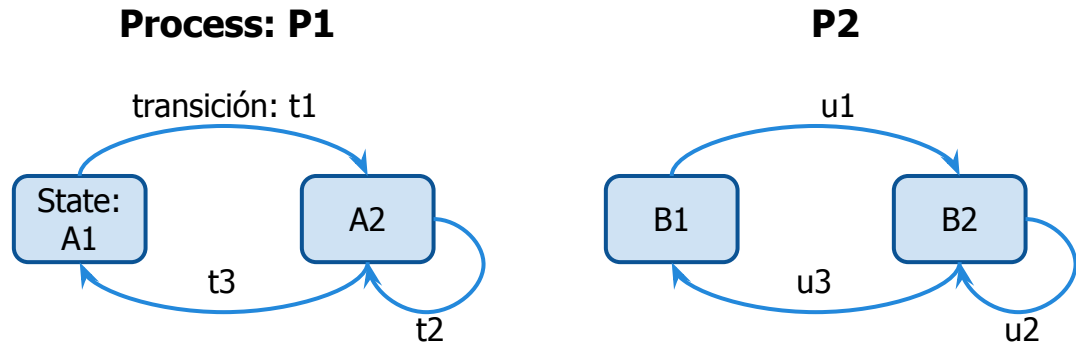


## Multi-computing

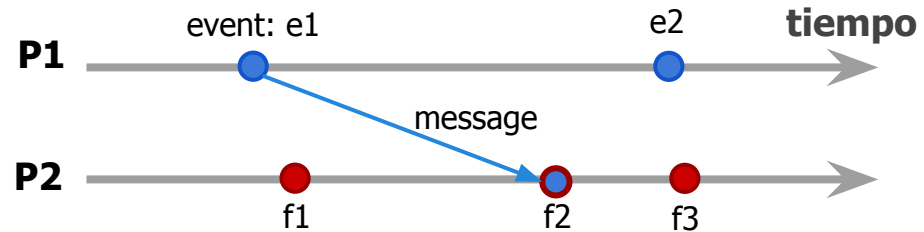




## Modelo de Estados (interleaved model)



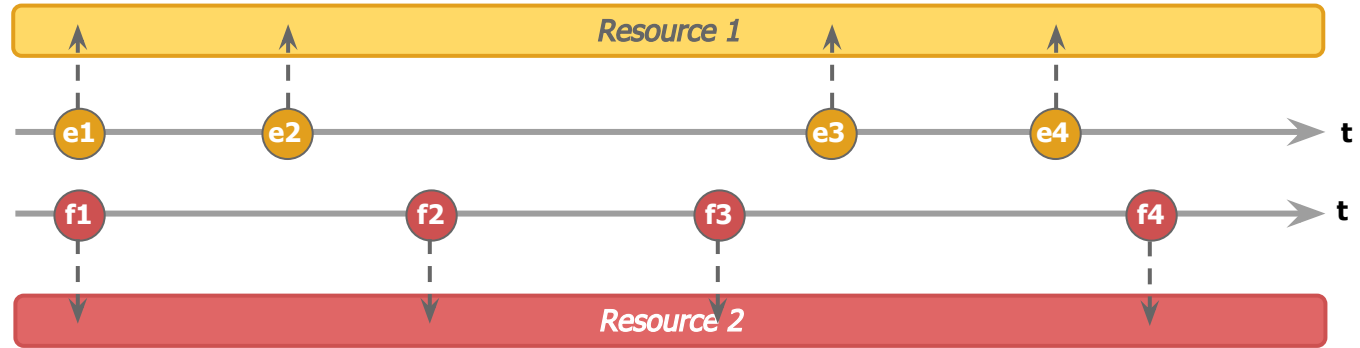
## Modelo de Eventos (happened before)



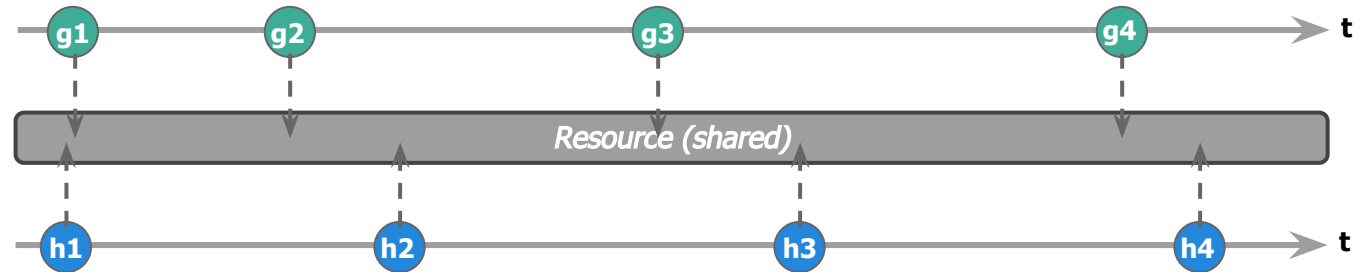
# Distribuidos | Paralelismo vs Concurrency



## Paralelo



## Concurrente





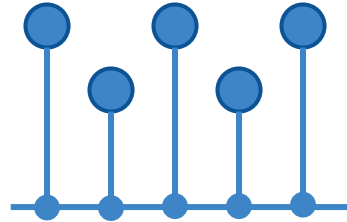
# Agenda



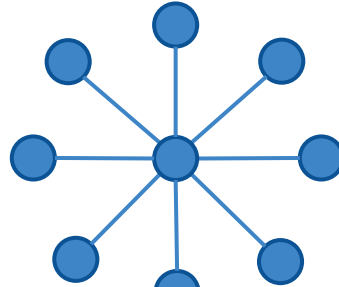
- ☐ Tendencias de Hardware y Software
- ☐ Definición de Sistemas Distribuidos
- ☒ Características Sistemas Distribuidos
- ☐ Virtualización



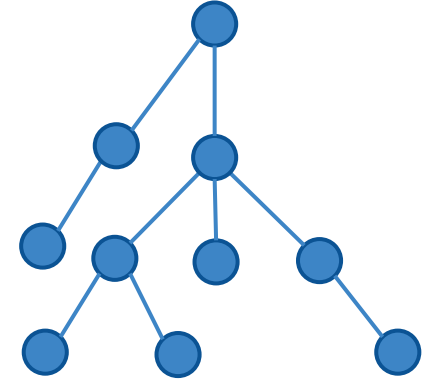
# Distribuidos | Topologías de Comunicación



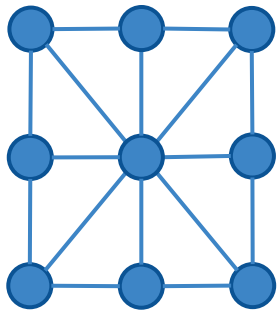
**Bus**



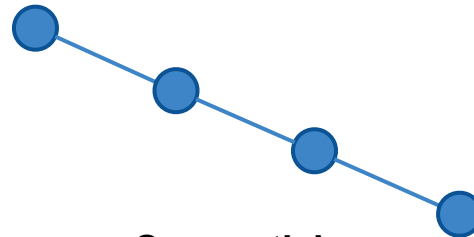
**Star**



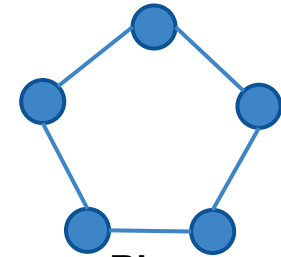
**Tree**



**Mesh**



**Sequential**



**Ring**



## Sist. Centralizados

Sin conexiones

Con conexiones pero:

- Sin trabajo colaborativo
- Sin un objetivo común
- Sistemas de tiempo compartido
- 'Terminales de conexión'

Muy difíciles de escalar:

- CPUs, Memoria, HD

## Sist. Distribuidos

Componentes conectados y:

- realizando trabajo colaborativo
- buscando un objetivo común

Escalan distribuyendo trabajo y recursos:

- Nodos, regiones, canales



## Centralizados vs Distribuidos | Ventajas de Centralizar

- **Control:** lógica de control muy simple, efectiva y, en ocasiones, eficiente.
- **Homogeneidad:** la centralización incita a definir estándares para software y hardware.
- **Consistencia:** es posible definir fuertes políticas de consistencia de información y monitoreo del estado global del sistema.
- **Seguridad:** se disminuye la 'superficie de ataque' frente a amenazas.



# Centralizados vs Distribuidos | Ventajas de Distribuir

- **Disponibilidad:** aún frente a fallos aislados, el sistema general puede prestar servicios.
- **Escalabilidad:** mejores alternativas de adaptarse a nuevas escalas
- **Reducción de Latencia:** al favorecer principios de localidad de recursos.
- **Colaboración:** permite interacciones entre sistemas de forma orgánica y natural.
- **Movilidad:** no están circunscriptos al alcance de un único computador.
- **Costo:** componentes más simples. Subsistemas delegados en servicios terceros.



# Descentralizar vs Distribuir

**Centralizar** implica la concentración de la autoridad en los niveles más altos de un jerarquía.

**Descentralizar** implica transferir la toma de decisiones a eslabones inferiores de cierta organización.

**Distribuir** implica utilizar un modelo descentralizado de control de computadoras para la coordinación de actividades con una coherencia dada.



# Pensamiento Distribuido | Ley de Conway

“Cualquier organización que diseñe un sistema, inevitablemente producirá un diseño cuya estructura será una copia de la estructura de comunicación de la organización”

(Conway M., *How do committees invent*, Datamation, 1968 )

- Este enunciado fue demostrado empíricamente en distintos relevamientos de arquitecturas de software corporativo.
- Corolario: diseñamos de acuerdo a lo que conocemos y estamos acostumbrados a hacer en el día a día.
- No es necesariamente negativo: en su trabajo, el hombre tiende a encontrar soluciones distribuidas y paralelas eficientes (minimizan costo, energía, tiempo, etc.)

# Agenda

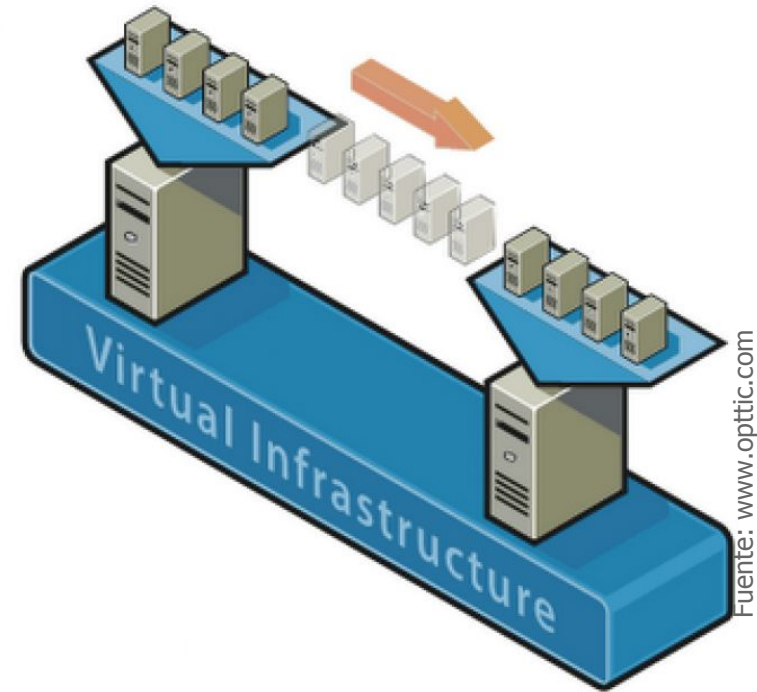


- Tendencias de Hardware y Software
- Definición de Sistemas Distribuidos
- Características Sistemas Distribuidos
- Virtualización



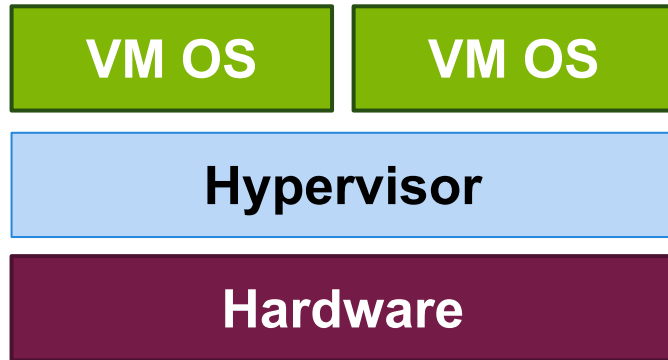


- Necesidad de independencia real de los recursos
  - Seguridad en el acceso
  - Diferenciación entre instrucciones con privilegios vs sin privilegios.
- Hypervisor (Virtual Machine Monitor)
  - Manager de VMs
  - Emulación de Hardware Capabilities
  - Administración de recursos del Host OS hacia los Guest OS
  - Implementación de mecanismos de seguridad

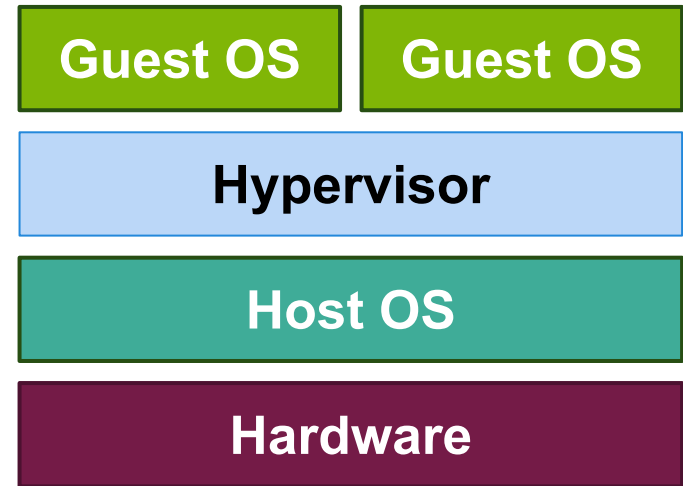




## Bare Metal (70s-00s)

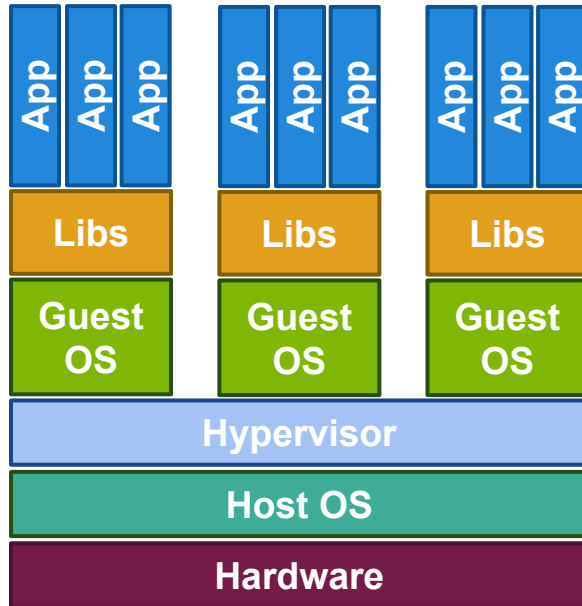


## Hosted (00s)

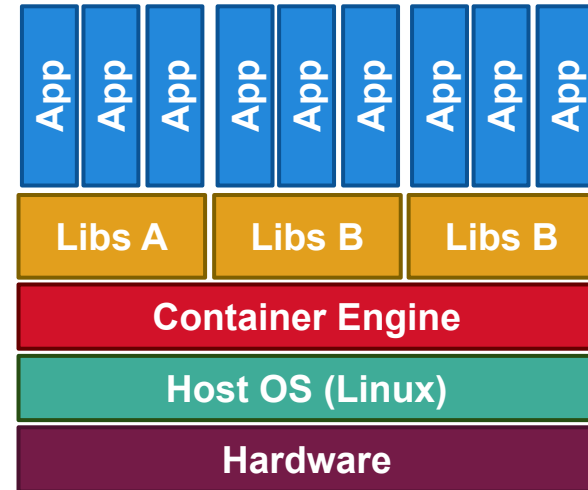




## Hosted (00s)

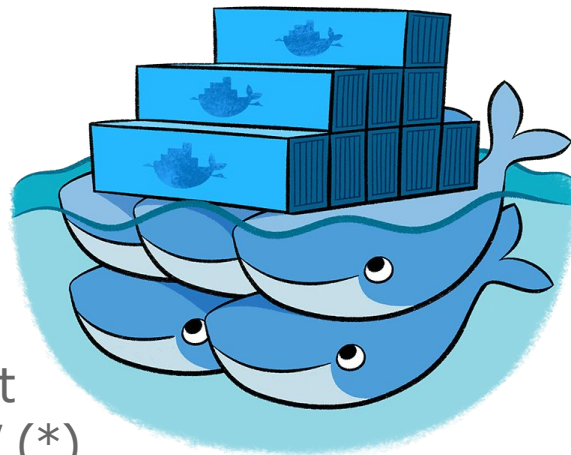


## Containers (10s)



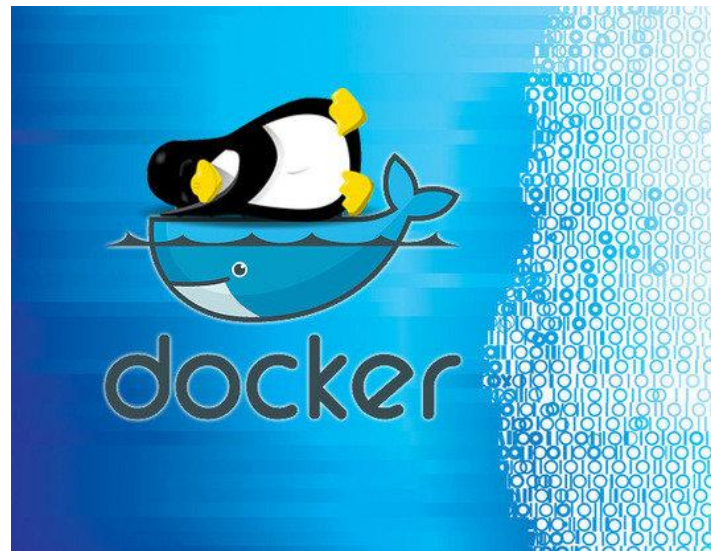


- Desarrollado en Golang
- Primera versión estable en 2013
- Basado fuertemente en features provistos por el kernel del OS y el Hypervisor:
  - Linux: cgroups, namespaces, union mount
  - Mac: recientemente soportado con HyperKit
  - Win: recientemente soportado con Hyper-V (\*)
- Rápida adopción por parte de la industria
  - Kubernetes, AWS, GCP, Azure, etc. soportan Docker





- **Namespaces:** Aislamiento de Recursos
  - NET (Network Stack)
  - PID (Procesos)
  - MNT (Mount)
  - IPC (Semáforos, Shared Memory, etc.)
  - USER (user and group ids)
- **Cgroups:** Seguridad y administración de recursos
- **Union Mount:** Múltiples directorios vistos como uno





- **Docker Daemon**

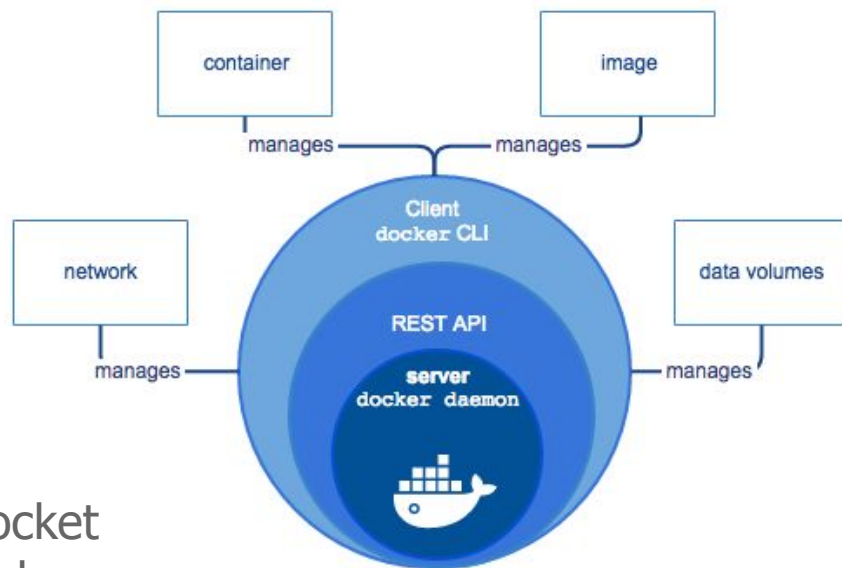
- Manejo de todos los recursos (images, containers, volumes, network, etc.)
- Nueva funcionalidad a través de Plugins (Ej. [convoy](#))

- **Docker REST API**

- Sumamente flexible
- Fácil crear nuevos clientes con funcionalidades específicas

- **Docker CLI**

- Acceso REST API a través de unix socket
- Gran cantidad de comandos disponibles





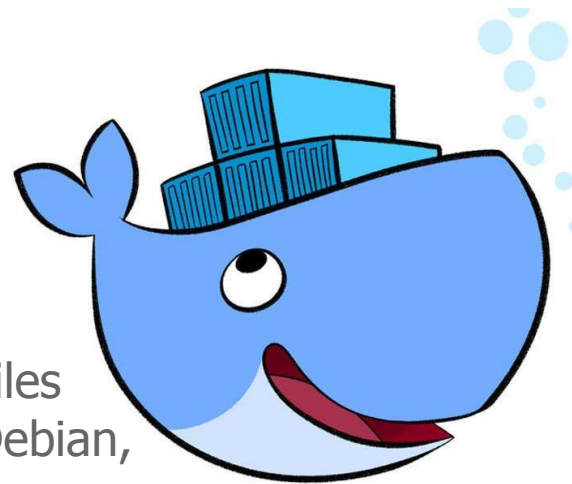
# Docker | Conceptos Básicos (I)

- **Dockerfile**

- Secuencia de comandos que generan Images
- Instalación de dependencias, librerías
- Cada comando agrega un nuevo Layer
- Agregado de archivos a utilizar

- **Image**

- Bloques estáticos contruidos a través de Dockerfiles
- Distros de Linux proveen base Images (Ubuntu, Debian, Centos, Alpine, etc.)
- Se pueden reutilizar para construir nuevas Images





- **Container**

- Ejecución de comandos dentro de una Image
- Kernel compartido entre Host OS y Container
- Al terminar el comando a ejecutar, el container deja de existir
- No proveen persistencia

- **Volume**

- Directorios compartidos entre el Host OS y container
- Necesarios para dar persistencia a aplicaciones







```
FROM centos:latest
WORKDIR /tmp

# Install wget to be able to download OpenMPI
RUN yum update -y && \
    yum groupinstall "Development Tools" -y && \
    yum install wget vim -y

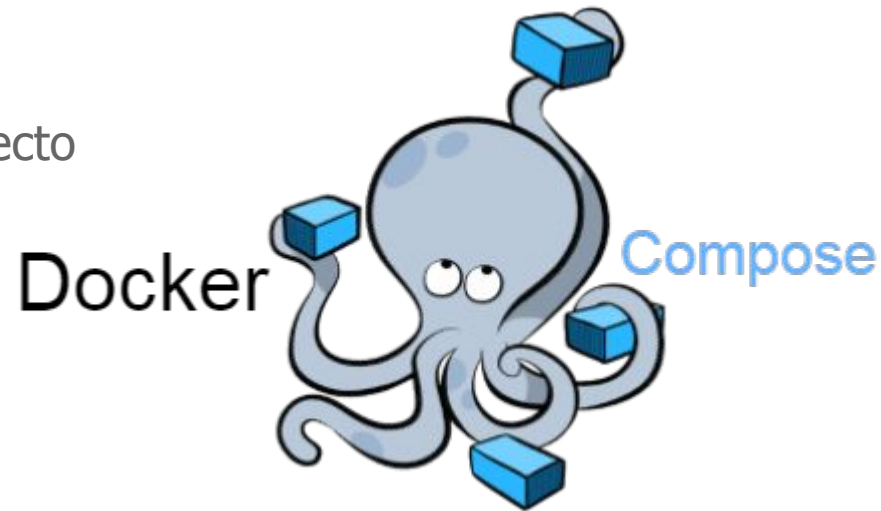
# Compile and install OpenMPI
RUN wget https://www.open-mpi.org/software/ompi/v3/openmpi-3.1.0.tar.gz && \
    tar -xvzf openmpi-3.0.0.tar.gz && \
    cd openmpi-3.0.0 && \
    ./configure --prefix=/opt/openmpi-3.0.0 && \
    make -j4 && make install && \
    cd /tmp && rm -rf /tmp/openmpi-3.0.0*

COPY env /opt/env
```



# Docker Compose

- Desarrollado en Python
- Administración de múltiples *containers*
  - DNS Resolution
  - *Containers* en la misma red por defecto
- Configuración a través de YAMLS
  - Intuitivo
  - Mayoría de los comandos Docker disponibles





# Docker Compose | docker-compose.yml

```
web:
  image: nginx:latest
  ports:
    - "8080:80"
  volumes:
    - ./code:/code
    - ./nginx.conf:
        /etc/nginx/conf.d/default.conf
  depends_on:
    - php

php:
  build: . #Requires a Dockerfile
  volumes:
    - ./code:/code
  environment:
    XDEBUG_CONFIG: remote_host=
                    host.docker.internal
  depends_on:
    - mysql
```

```
mysql:
  image: mysql:latest
  ports:
    - "3306:3306"
  environment:
    - MYSQL_ROOT_PASSWORD=password

phpmyadmin:
  image: phpmyadmin/phpmyadmin
  depends_on:
    - mysql:db
  ports:
    - "8082:80"
  environment:
    - PMA_USER=root
    - PMA_PASSWORD=password
    - PHP_UPLOAD_MAX_FILESIZE=100MB
```



# Bibliografía

- P. Verissimo, L. Rodriguez: Distributed Systems for Systems Architects, Kluwer Academic Publishers, 2001.
- G. Coulouris, J. Dollimore, t. Kindberg, G. Blair: Distributed Systems. Concepts and Design, 5th Edition, Addison Wesley, 2012.
- M. Van Steen, A. Tanenbaum: Distributed Systems. 3rd Edition. Pearson Education, 2017.