

CS 2031 - DBP

# Desarrollo Basado en Plataformas

Jorge Ríos

# ¿Qué haremos hoy?

Auditorio

- 1 Conoce al equipo**
  - 2 Logros del curso**
  - 3 Temario**
  - 4 Semana 0 y E2E**
  - 5 Sistema de evaluación**
- Break 5 min —
- 6 Clase Teoría**

# Conoce al equipo



**Jesus Bellido**



**Jorge Rios**



**Mateo Noel**



**Mauricio Jacobo**



**Gabriel Romero**



**Jeffrey Monja**

# Logros del curso



**Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional para alcanzar las necesidades deseadas.**



**Trabajar efectivamente en equipos para cumplir con un objetivo común.**



**Que el alumno sea capaz de desarrollar servicios y aplicaciones web utilizando herramientas y lenguajes como HTML, CSS, JavaScript, back-end scripting y una base de datos, a un nivel intermedio.**



**Desarrollar aplicaciones móviles, administrar servidores web en sistemas basados en UNIX y aplicar técnicas de seguridad en la web a un nivel intermedio.**

# Sesiones Auditorio

Computing Platform				
Architectural Constraints	Client - Server	Stateless Stateful	Mobile	
REST Representational State Transfer	Uniform Interface	Testing	Events	
Security	Separation of Concerns	UI/UX	Layered System	

# Sesiones Laboratorio

Back End		Front End	
ORM	Security	React	React Native
Testing	Events	Sensors	Expo
DTOs	Handle Exceptions	UI/UX	External API

# Semana 0

The screenshot shows a web browser window with the following details:

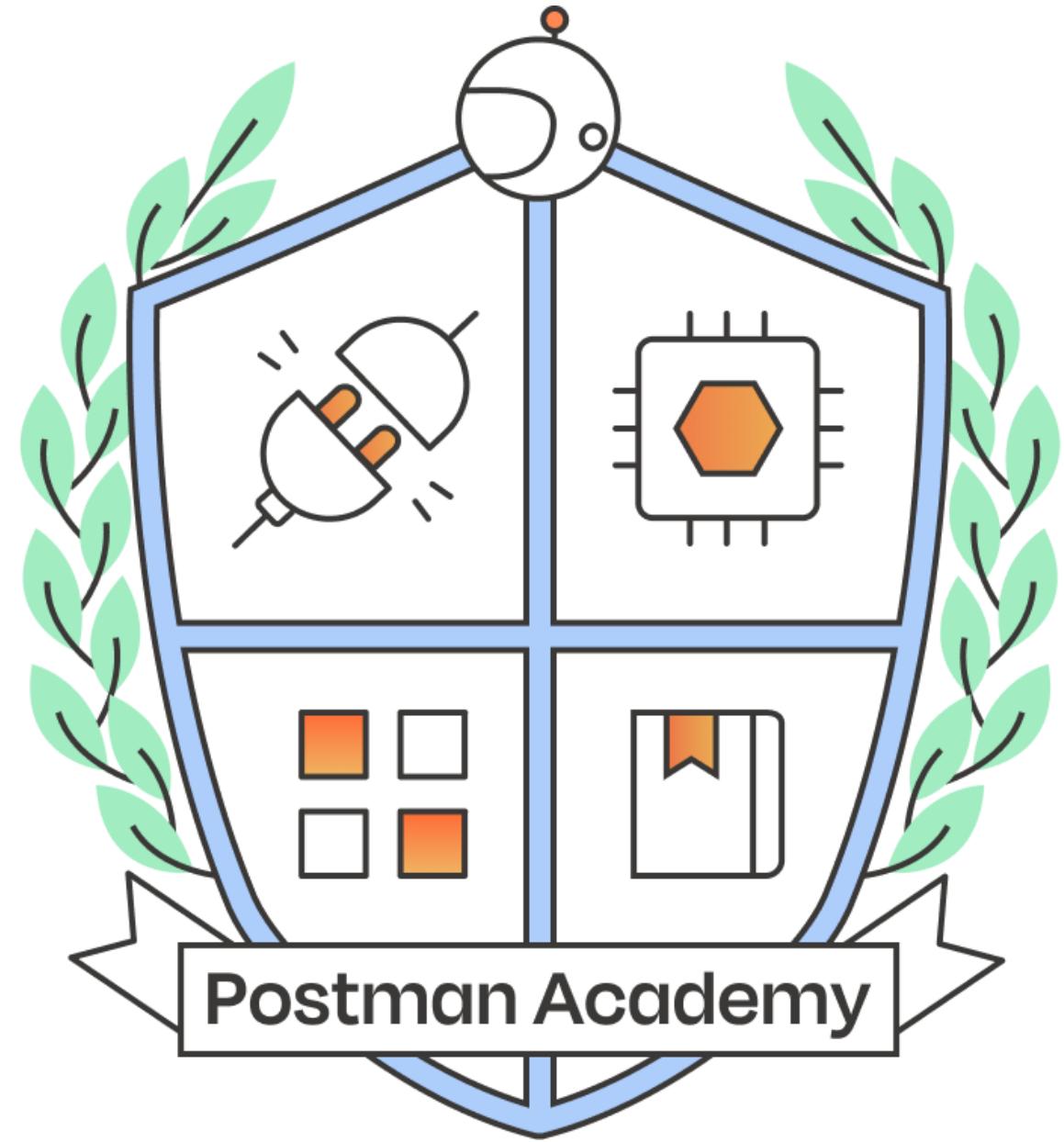
- Header:** The title "Desarrollo Basado en Plataformas (CS2031)" and subtitle "Ciencia de la Computación - UTEC" are displayed above a large, colorful illustration of a bird.
- Body:**
  - A main heading "¡Bienvenidos a CS2031 - DBP (2024-I) en UTEC!" is centered.
  - Text describing the course's purpose: "Este curso está diseñado para equiparlos con habilidades críticas en el desarrollo de aplicaciones de software utilizando diversas plataformas (web y mobile), preparándolos para enfrentar los desafíos tecnológicos del futuro."
  - Information about the course structure:
    - "Este semestre, el curso se compone de:"
    - A bulleted list detailing the course components: 1. Clase de teoría semanal, 2. Clases de laboratorio semanales, 3. Sesiones de asesoría semanales, and Tareas de Laboratorio.
- Left Sidebar:** A navigation menu titled "CS2031-DBP" with the following items:
  - Página principal
  - Semana 0
    - Temario
    - Instalación
  - Master Class
    - Git y Github
    - Terminal
    - Frontend
    - Docker
    - Entidad-Relación
    - Java + POO
    - Postman
  - Sparky
  - Referencias

## CS2031 Web

- **Tutoriales de instalación**
- **Lecturas y videos de refuerzo y nivelación**
- **Material complementario**

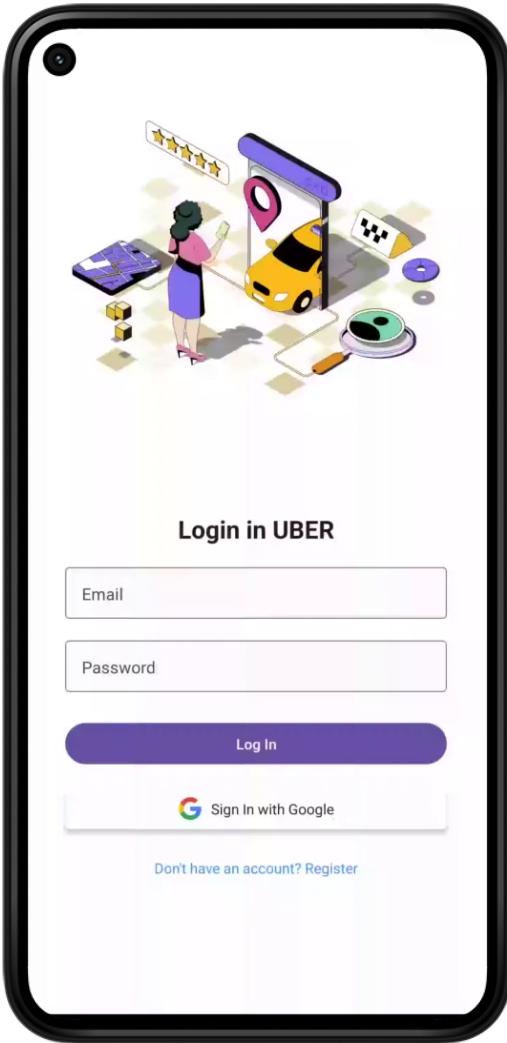


**Github Campus Program**



**CS 2031**

# Proyecto End to End (E2E)



- **Web + Mobile**
- **WebSockets**
- **Google Maps API**
- **Uber H3**

# Sistema de Evaluación

Teoría (60%)

- **Evaluación Continua (C1)**

$$C1 = \frac{1}{12} \sum_{i=1}^{12} (EA) \times 0.60 + HKT1 \times 0.20 + HKT2 \times 0.20$$

Evaluación en Auditorio (EA)

Hackathones (HKT)

- **Entrega de Proyectos (P1 y P2)**

Cada uno conforma el 20% de la teoría. Se desarrollan en grupos, aplicando conocimientos progresivos y habilidades de trabajo en equipo.

# Sistema de Evaluación

## Laboratorio (40%)

- **Prácticas Calificadas**

Son dos pruebas que se llevan a cabo durante el ciclo, cada una constituyendo el 10% de la parte práctica.

- **Evaluación Continua (C2)**

Equivale al 20% de la práctica, basada en la media ponderada de los laboratorios semanales (LS), que son laboratorios asincrónicos parte del proyecto E2E.

$$C2 = \frac{1}{10} \sum_{i=1}^{10} (LS) \times 0.20$$

# **Asesorías con TAs**

**Mon Apr 01**

**Tue Apr 02**

**9 AM – 11 AM  
(A401)**

**Wed Apr 03**

**3 PM – 5 PM  
(M604)**

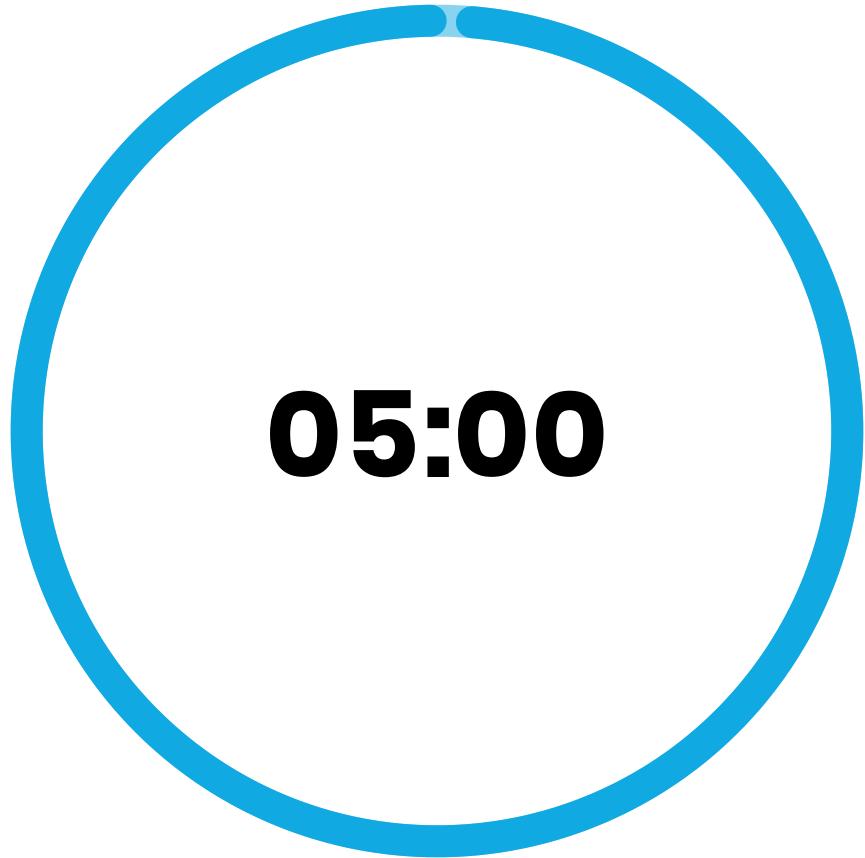
**Thu Apr 04**

**4 PM – 6 PM  
(A502)**

**Fri Apr 05**

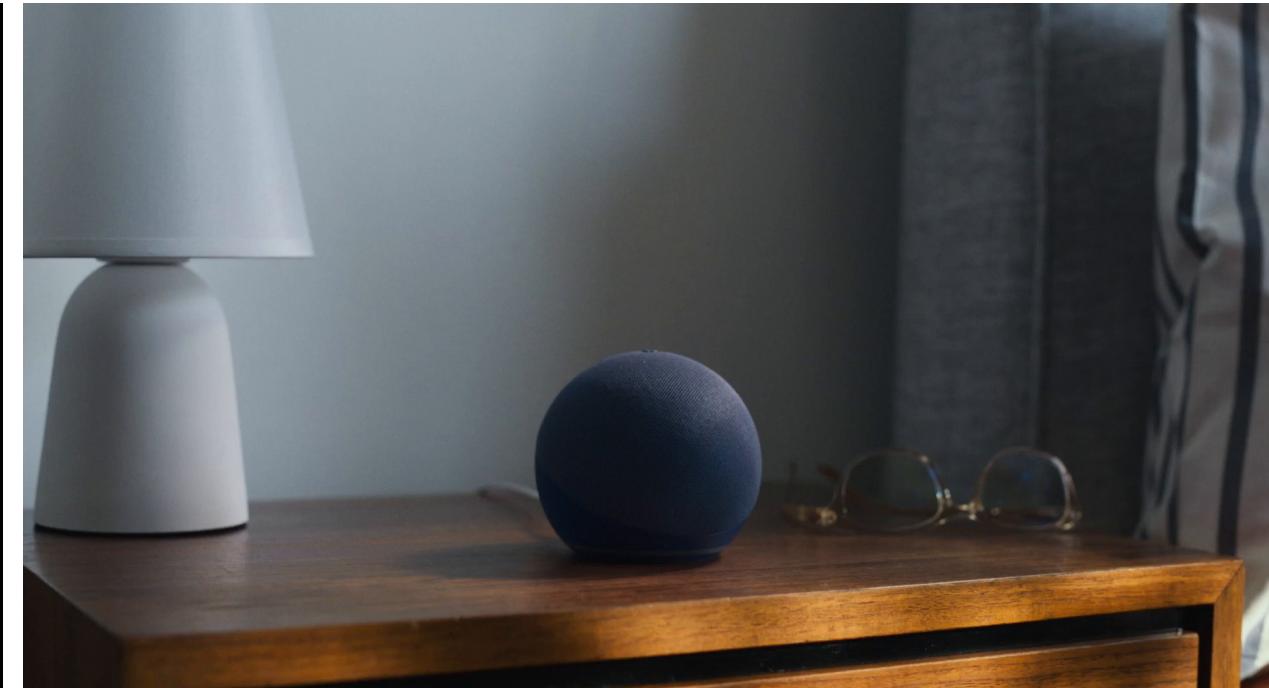
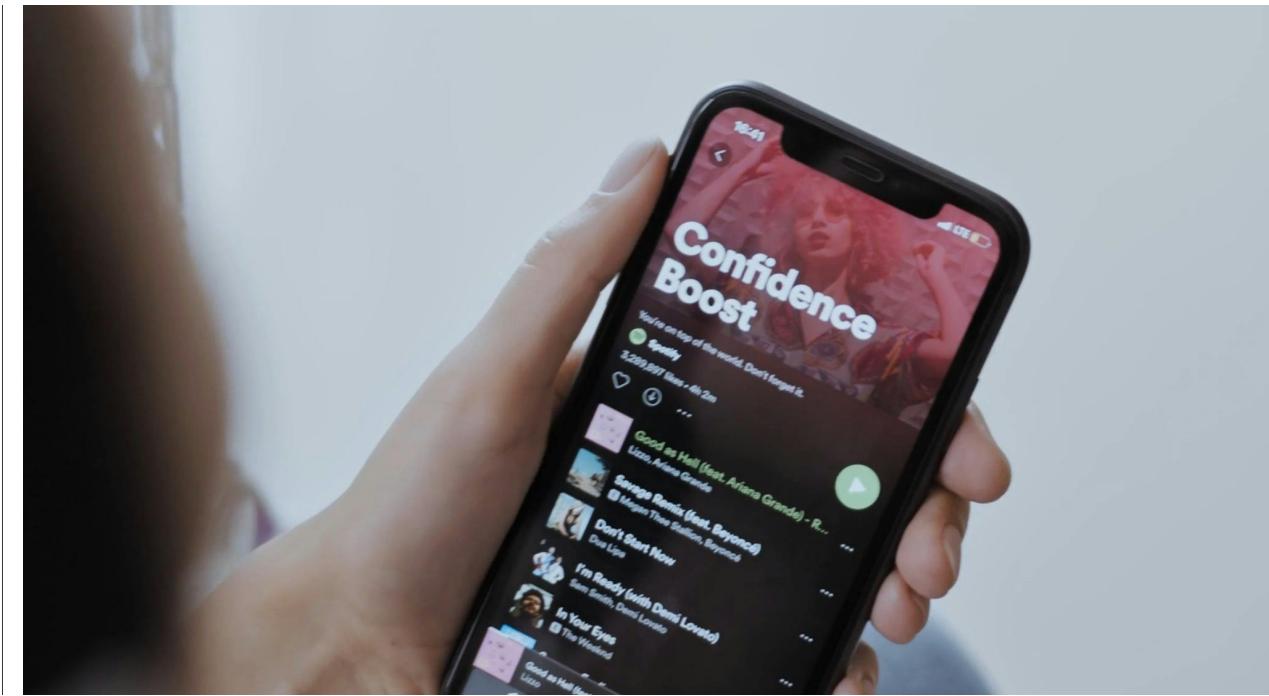
**11 AM – 1 PM  
(M604)**

**Break  
5 min**



Teoría

# Desarrollo Basado en Plataformas



# ¿Cómo funciona el Internet?

# El mundo antes del Internet



El telégrafo emitía código Morse usando señales eléctricas que viajaban por cables conectados entre un origen y un destino.



Redes de computadoras en el ámbito académico, en universidades como MIT y UCLA. Estas redes también se utilizaban en instituciones de investigación y departamentos de defensa de USA.

# ARPANET

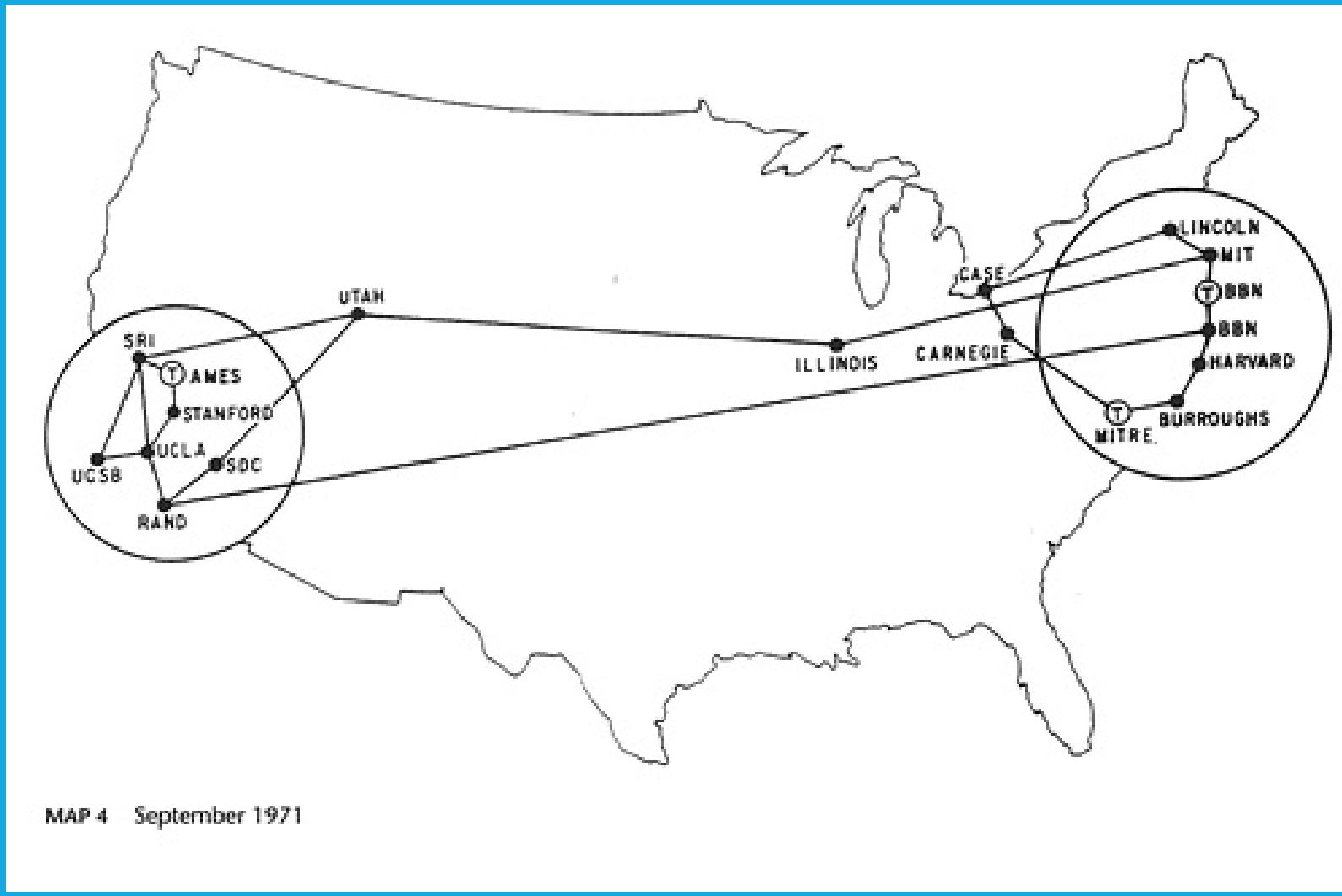
**ARPANET fue desarrollada por la Agencia de Proyectos de Investigación Avanzados (ARPA) a finales de los años 60.**

**Actuó como precursor de Internet, siendo la primera red operativa de conmutación de paquetes del mundo.**

**Permitió que computadoras geográficamente separadas pudieran comunicarse e intercambiar datos a través de la red.**

**El primer mensaje enviado a través de ARPANET en 1969 conectó computadoras en UCLA y Stanford.**

**El desarrollo de protocolos como TCP/IP fue fundamental para el crecimiento de ARPANET y eventualmente del Internet.**



# ¿Problemas con ARPANET?

Limitaciones de NCP  
(Network Control Protocol)

- **Inflexibilidad: ARPANET no era capaz de interconectar diferentes tipos de red.**
- **No había mecanismos sólidos para manejar errores o control de la congestión de la red. A medida que aumentaba el tráfico en la red, también aumentaba la pérdida de paquetes y la corrupción de datos.**
- **El direccionamiento de ARPANET no era escalable. La asignación fija de direcciones limitaba la cantidad de hosts y dificultaba la incorporación de redes nuevas.**

# TCP/IP

1

Diseñado para **interconexión de redes heterogéneas**, permitiendo la formación de una red global de múltiples redes menores con diferentes arquitecturas.

2

IP introdujo un **sistema de direccionamiento** que permitía la **identificación** única de dispositivos y superó las limitaciones de NCP.

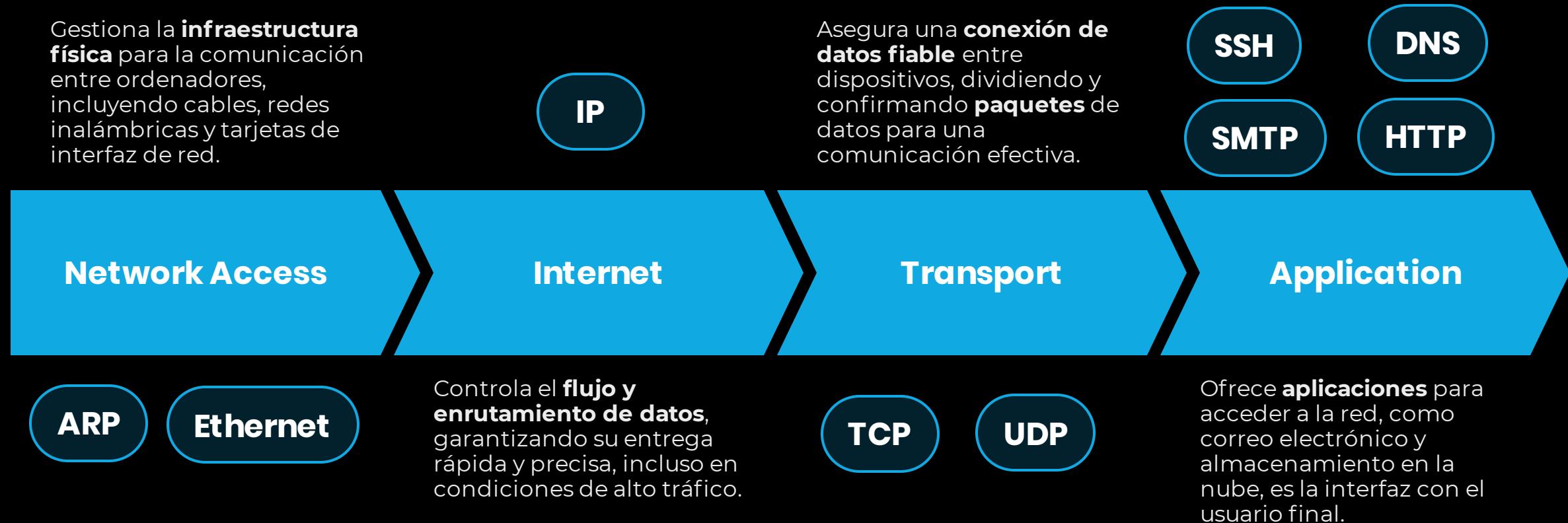
3

TCP/IP introdujo un **modelo de capas**, separando funciones para flexibilidad, escalabilidad y adaptación a nuevas tecnologías.

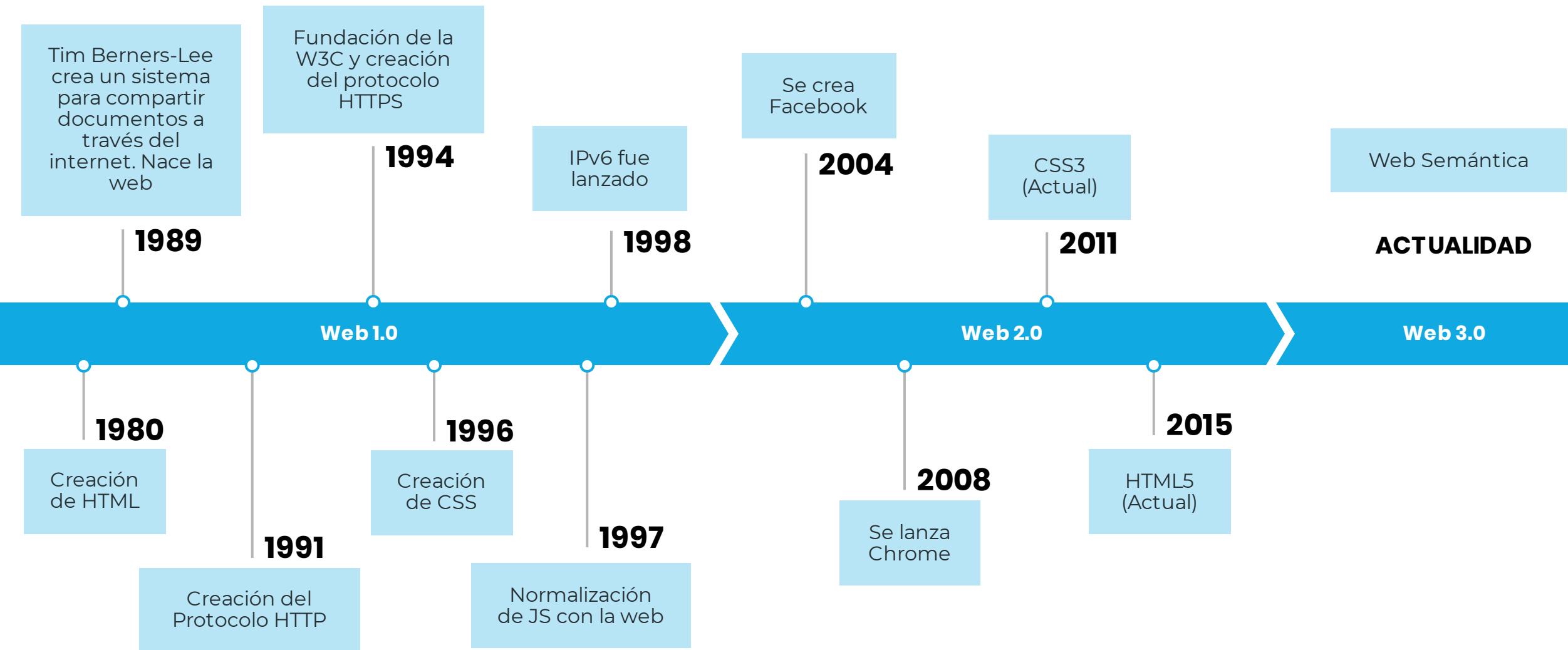
4

TCP/IP fue diseñado para soportar una amplia gama de aplicaciones, promoviendo la flexibilidad y adaptabilidad, y abriendo camino al desarrollo de la World Wide Web.

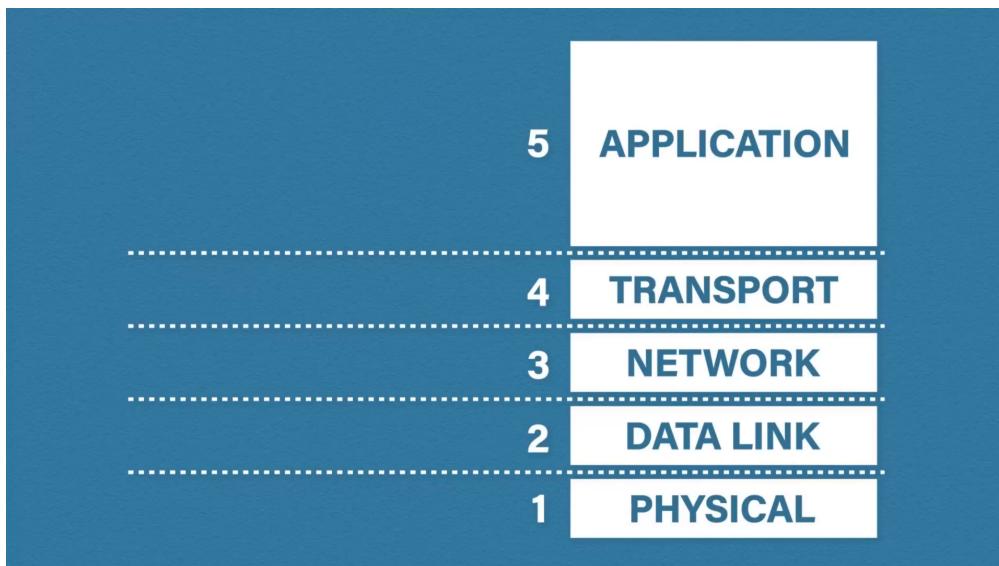
# Capas TCP/IP



# World Wide Web



# Standardized Network Communication



## TCP/IP

- Protocolo fundamental que subyace a la comunicación en Internet.
- Proporciona una conexión confiable para la transmisión de datos en redes.
- IP facilita el direccionamiento de los paquetes de datos hasta su destino final.

## HTTP

- HTTP se basa en TCP/IP para la transmisión de datos.
- Transmisión de datos basado en la arquitectura *Client-Server*.
- Usa los *métodos HTTP* como GET, POST, PUT y DELETE

# Principios REST

## Client-Server

**Añade portabilidad a la interfaz de usuario entre múltiples plataformas y mejora la escalabilidad simplificando los componentes del servidor**

## Stateless

Cada solicitud del cliente al servidor debe incluir toda la información necesaria para comprender la solicitud.

## Uniform Interface

El servidor envía/recibe datos en un formato estándar

## Code On Demand

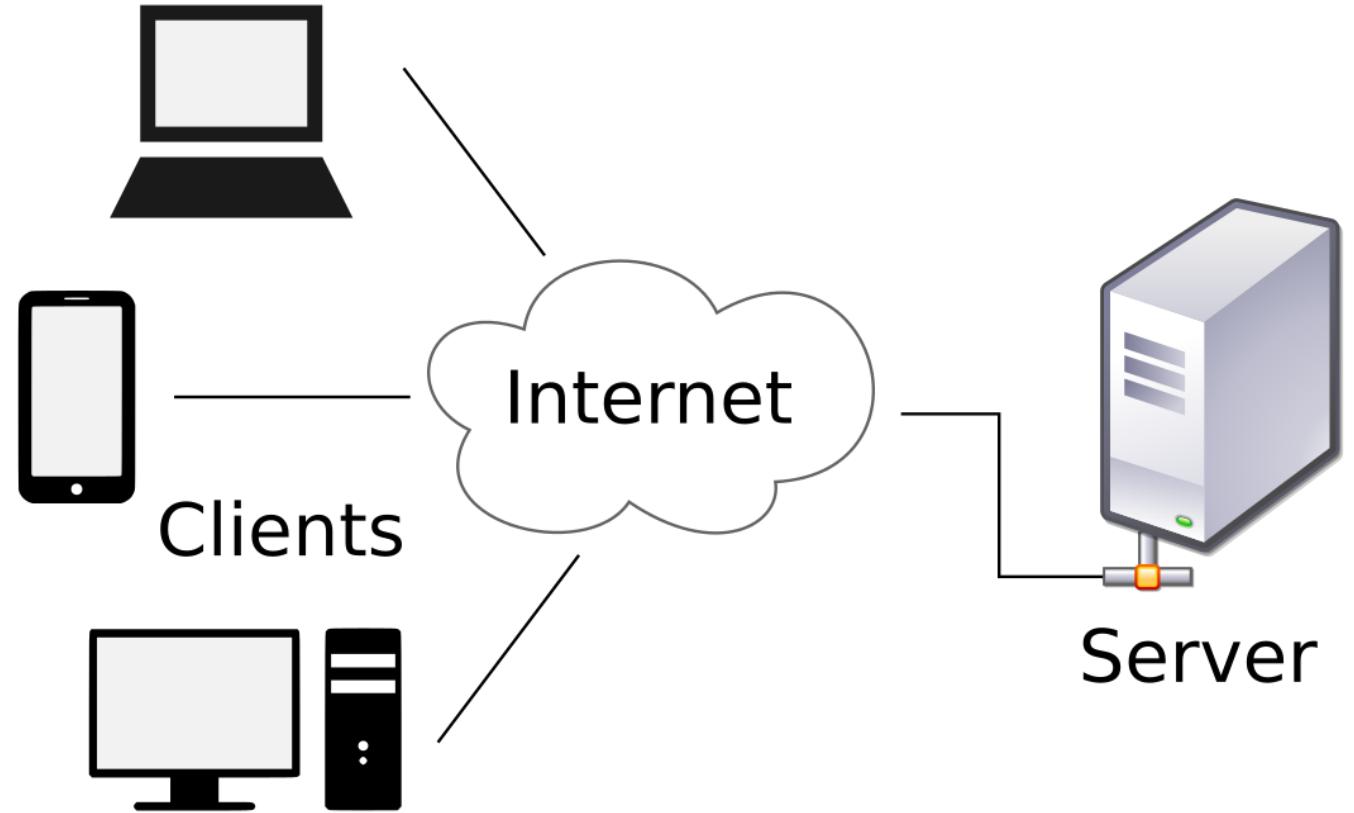
Mejorar la funcionalidad del cliente al permitir la descarga y ejecución de código.

## Cache

Usar memoria cache para mejorar la eficiencia, escalabilidad y el rendimiento percibido por el usuario.

**CS 2031**

# **Client- Server**



# Gracias