

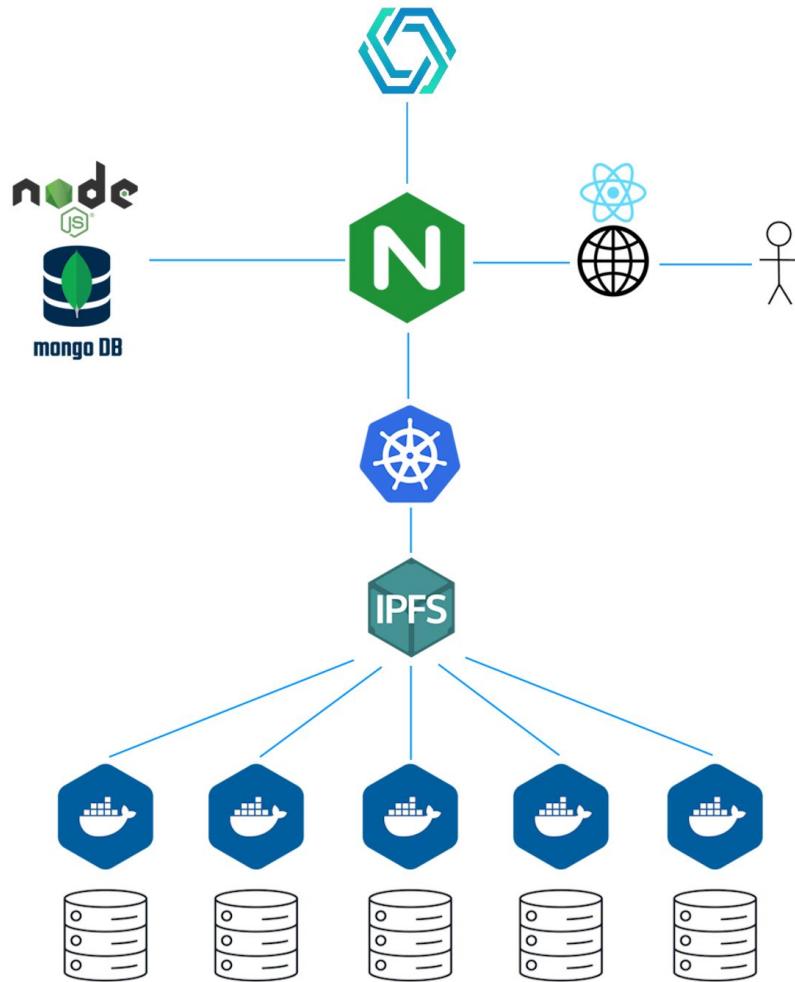
UPC Cloud

Javier Armaza Bravo
Sergi Colomer Segalés
Sofía López Olivares
Pablo Rosas Roda

Índice

- Propuesta inicial
- Paquetes trabajo
 - Nginx
 - FrontEnd
 - Backend
 - IPFS
 - Kubernetes
 - Blockchain
- Estructura final

Propuesta inicial

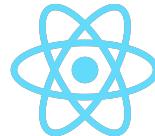


Paquetes trabajo

1. Nginx



2. FrontEnd



3. BackEnd

4. IPFS



5. Kubernetes



6. Blockchain



Nginx

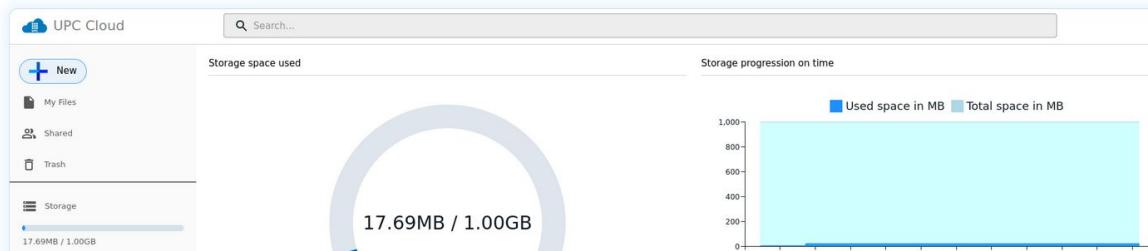
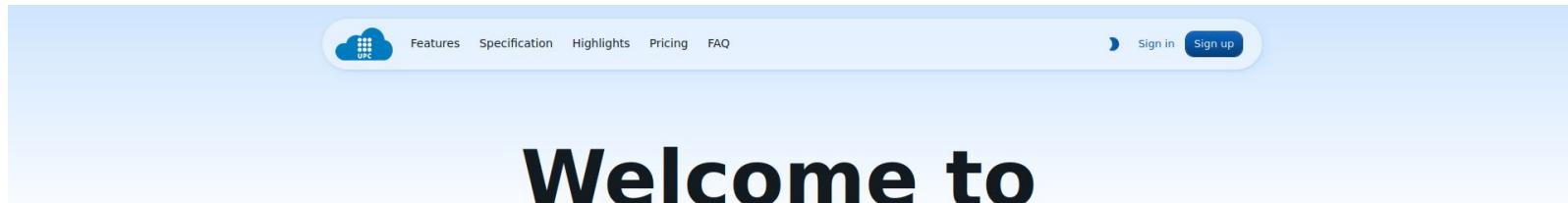
```
server {  
    listen 443 ssl default_server;  
    include snippets/self-signed.conf;  
    include snippets/ssl-params.conf;  
  
    root /var/www/upccloud/html;  
  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name upccloud.com www.upccloud.com;  
  
    location / {  
        try_files $uri /index.html;  
    }  
}  
  
server {  
    listen 80;  
    server_name upccloud.com www.upccloud.com;  
    return 301 https://$host$request_uri;  
}
```

Nginx

```
server {  
    listen 443 ssl default_server;  
    include snippets/self-signed.conf;  
    include snippets/ssl-params.conf;  
  
    root /var/www/upccloud/html;  
  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name upccloud.com www.upccloud.com;  
  
    location / {  
        try_files $uri /index.html;  
    }  
}  
  
server {  
    listen 80;  
    server_name upccloud.com www.upccloud.com;  
    return 301 https://$host$request_uri;  
}
```

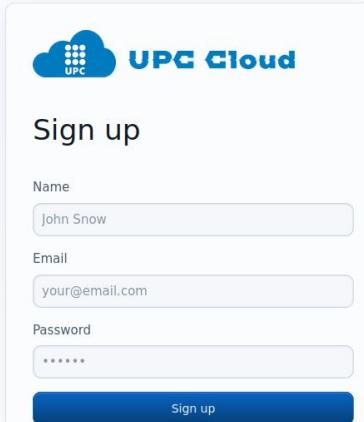
```
upstream backend {  
    server 10.4.41.68:3001;  
}  
  
server {  
    listen 443 ssl;  
    include snippets/self-signed.conf;  
    include snippets/ssl-params.conf;  
    location / {  
        proxy_pass http://backend/;  
        proxy_set_header Host $host;  
    }  
}  
  
server {  
    listen 80;  
    return 302 https://10.4.41.68$request_uri;  
}
```

FrontEnd



FrontEnd

← Back



The image shows a sign-up form for UPC Cloud. The form is contained within a white rectangular box with rounded corners. At the top left is the UPC logo, which consists of a blue cloud icon with white dots and the text "UPC Cloud". Below the logo is the title "Sign up". The form has three input fields: "Name" with the placeholder "John Snow", "Email" with the placeholder "your@email.com", and "Password" with five asterisks as a placeholder. At the bottom of the form is a blue "Sign up" button. Below the button, there is a link "Already have an account? Sign in".

UPC Cloud

Sign up

Name

John Snow

Email

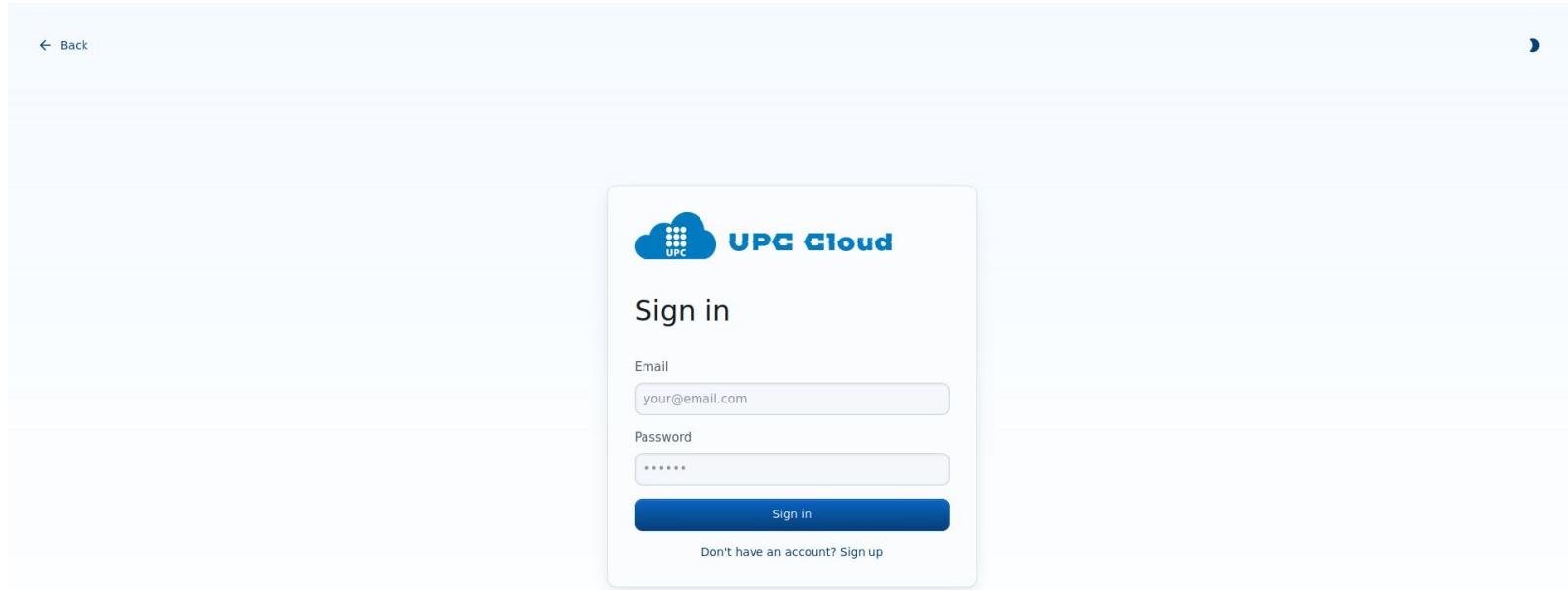
your@email.com

Password

Sign up

Already have an account? [Sign in](#)

FrontEnd



FrontEnd

The screenshot shows the UPC Cloud storage interface. On the left, there's a sidebar with a 'New' button, 'My Files', 'Shared', and 'Trash' sections. Below that is a 'Storage' section showing 91.75MB / 1.00GB. The main area is titled 'My Storage' and contains two preview cards for 'google-chrome-stable_c...deb' (file icon) and 'si-lab4.tar' (folder icon). Below these are two table rows showing the details of the files.

| Name | Owner | Last Modified | File Size |
|--|--------------------------|---------------------|-----------|
| google-chrome-stable_current_amd64.deb | rubenapruebame@gmail.com | 2024-05-09 11:15:51 | 90.27MB |
| si-lab4.tar | rubenapruebame@gmail.com | 2024-05-09 11:12:49 | 1.47MB |

Base de datos - MongoDB

MongoDB Compass - localhost:27017/Proyecto

Connect Edit View Help

localhost:27017 Proyecto +

My Queries localhost:27017 > Proyecto

Create collection Refresh

View Collection Name Sort by

archivo

| Storage size: | Documents: | Avg. document size: | Indexes: | Total index size: |
|---------------|------------|---------------------|----------|-------------------|
| 20.48 kB | 2 | 159.00 B | 1 | 36.86 kB |

Compartido

| Storage size: | Documents: | Avg. document size: | Indexes: | Total index size: |
|---------------|------------|---------------------|----------|-------------------|
| 20.48 kB | 1 | 221.00 B | 1 | 24.58 kB |

usuario

| Storage size: | Documents: | Avg. document size: | Indexes: | Total index size: |
|---------------|------------|---------------------|----------|-------------------|
| 20.48 kB | 2 | 96.00 B | 1 | 36.86 kB |

My Queries Performance Databases Search Projeto + -

¿Por qué MongoDB?



Servicio de almacenamiento distribuido robusto y escalable que está preparado para crecer y evolucionar con el tiempo.



Flexibilidad para experimentar y pivotar

Escalabilidad horizontal



- Agregar nuevos servidores.
- Distribución de carga eficiente.
- Flexibilidad para crecer sin problemas
- Facilita la expansión de infraestructura

Optimizado para un alto rendimiento



Acceso y manipular de datos rápido y eficiente para el usuario.

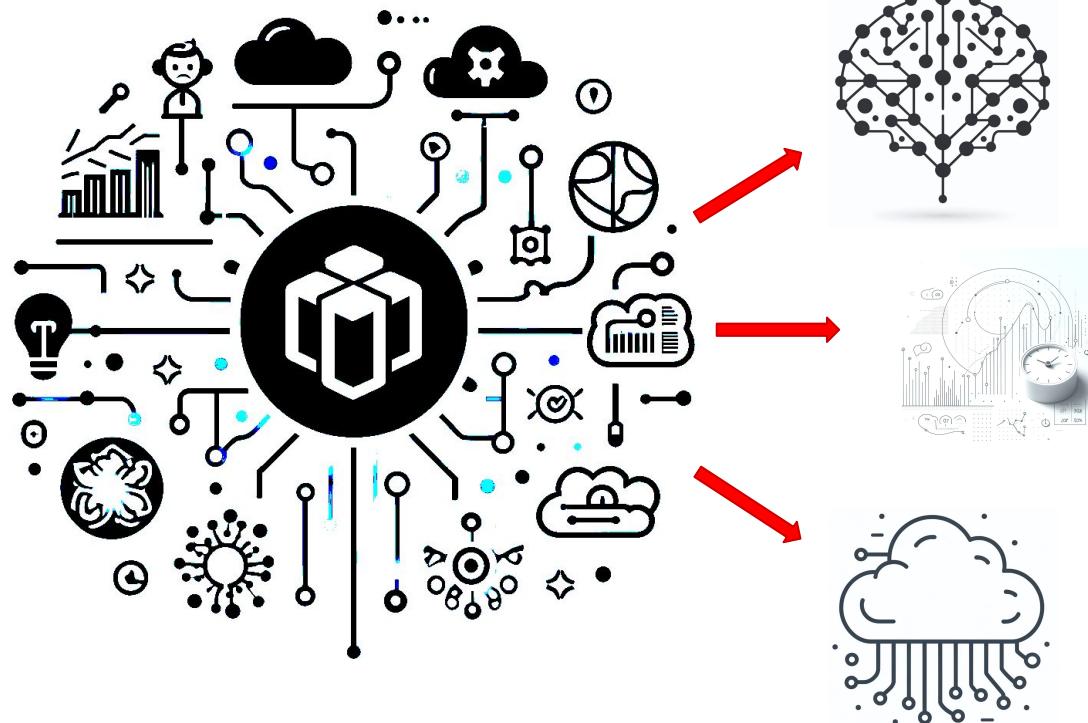


Una experiencia fluida y receptiva para los usuarios



Mejor satisfacción y posible fidelidad.

Integración fácil con tecnologías emergentes



Permite:

- Aprovechar nuevas oportunidades y tendencias tecnológicas
- Mantener relevancia y competencia en el mercado.

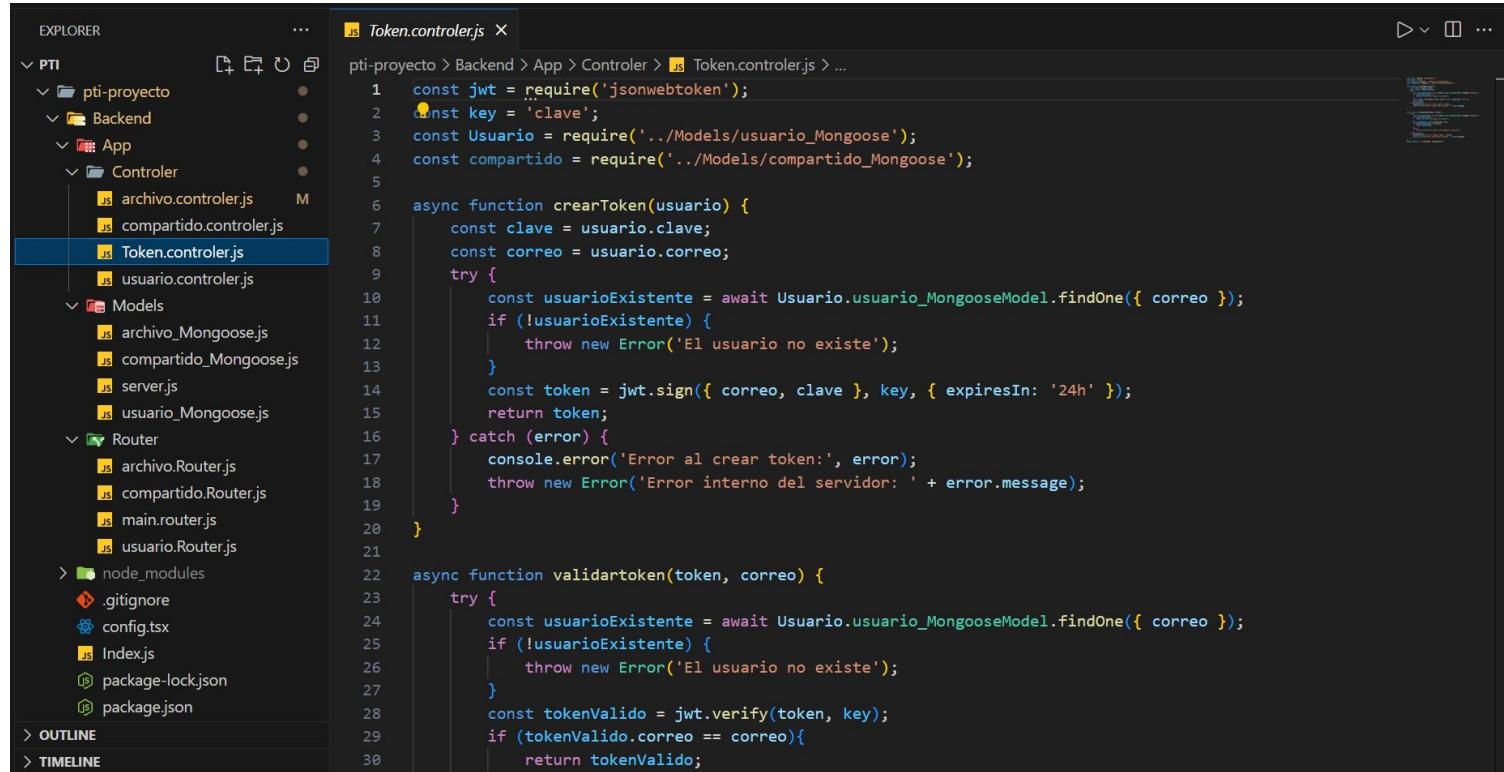
Capacidad para experimentar y pivotar



Crucial para el éxito a largo plazo del proyecto

- Flexibilidad para iterar rápidamente el servicio
- Probar nuevas características
- Adaptación a los comentarios de los usuarios, sin verse limitado
- Permitiendo mantener la agilidad y la capacidad de respuesta

Backend - Node

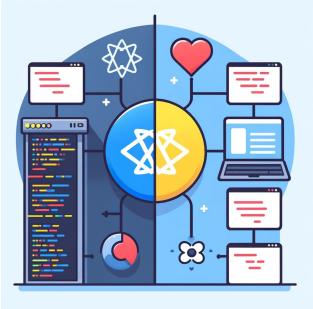


The screenshot shows the Visual Studio Code interface with the following details:

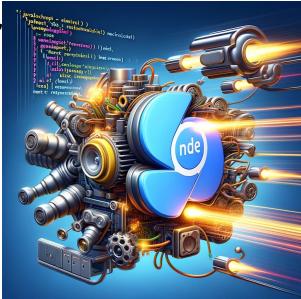
- EXPLORER** sidebar: Shows the project structure under "PTI". The "Token.controller.js" file is selected in the "Controller" folder.
- Token.controller.js** (File Preview): Displays the code for the controller. It uses `jsonwebtoken` and `mongoose` modules. The code defines two asynchronous functions: `crearToken` (creates a token for a user) and `validartoken` (validates a token).
- OUTPUT**: Shows the terminal output area.
- TIMELINE**: Shows the history of changes made to the file.

```
js Token.controller.js x
pti-proyecto > Backend > App > Controller > Token.controller.js > ...
1 const jwt = require('jsonwebtoken');
2 const key = 'clave';
3 const Usuario = require('../Models/usuario_Mongoose');
4 const compartido = require('../Models/compartido_Mongoose');
5
6 async function crearToken(usuario) {
7     const clave = usuario.clave;
8     const correo = usuario.correo;
9     try {
10         const usuarioExistente = await Usuario.usuario_MongooseModel.findOne({ correo });
11         if (!usuarioExistente) {
12             throw new Error('El usuario no existe');
13         }
14         const token = jwt.sign({ correo, clave }, key, { expiresIn: '24h' });
15         return token;
16     } catch (error) {
17         console.error('Error al crear token:', error);
18         throw new Error('Error interno del servidor: ' + error.message);
19     }
20 }
21
22 async function validartoken(token, correo) {
23     try {
24         const usuarioExistente = await Usuario.usuario_MongooseModel.findOne({ correo });
25         if (!usuarioExistente) {
26             throw new Error('El usuario no existe');
27         }
28         const tokenValido = jwt.verify(token, key);
29         if (tokenValido.correo == correo){
30             return tokenValido;
31         }
32     } catch (error) {
33         console.error('Error al validar token:', error);
34         throw new Error('Error interno del servidor: ' + error.message);
35     }
36 }
37
38 module.exports = {
39     crearToken,
40     validartoken
41 }
```

¿Por qué Node?



JavaScript en el servidor
y en el cliente



Tiempo de ejecución rápido
y eficiente



Modelo de E/S no
bloqueante y orientado
a eventos

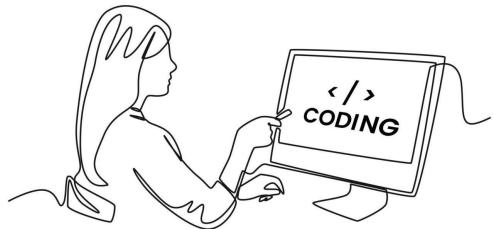


Amplio ecosistema de paquetes y
módulos

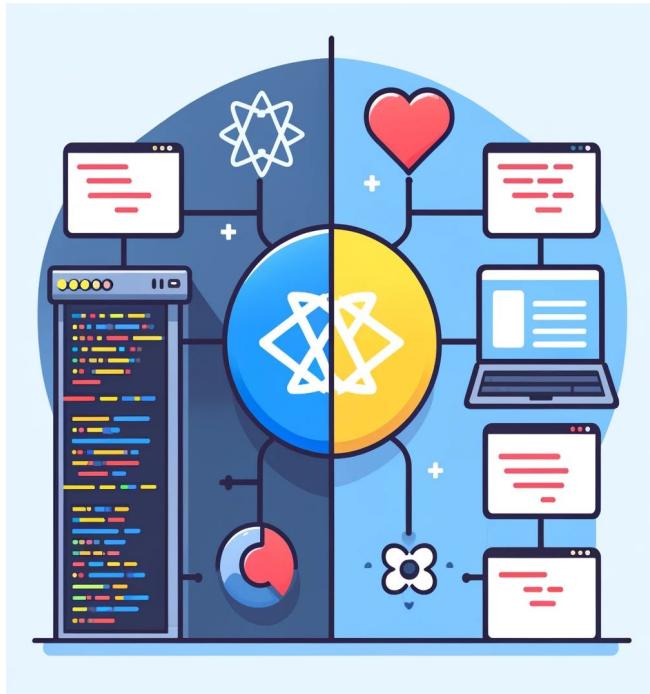


Comunidad activa y
soporte continuo

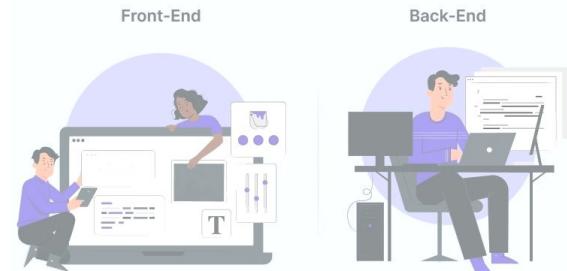
Node.js utiliza JavaScript tanto en el lado del servidor como en el lado del cliente



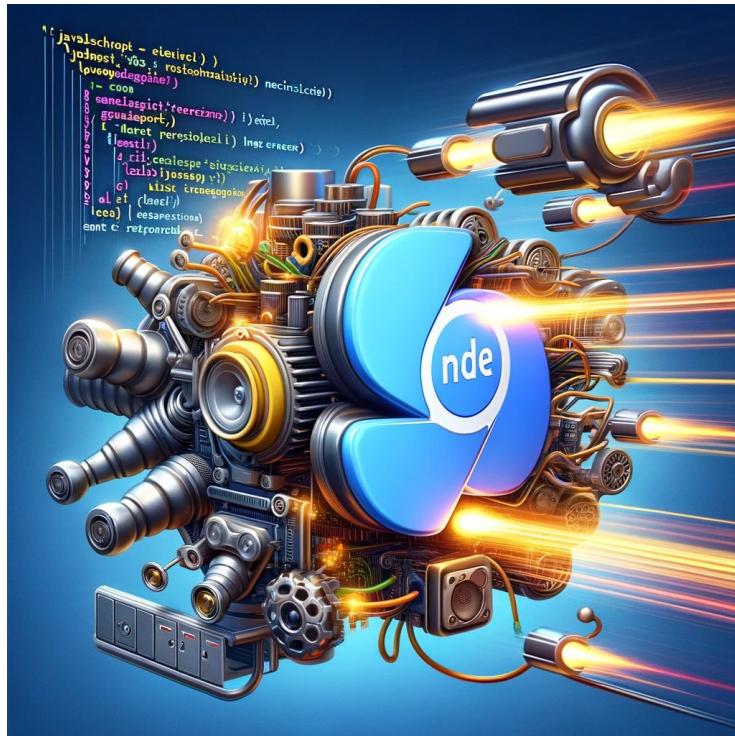
Lenguaje coherente en todo el stack de tecnología



- Simplifica el desarrollo
- Simplifica la colaboración



Tiempo de ejecución rápido y eficiente



Construido sobre el motor
V8 de Google Chrome



Compila el código
JavaScript



Código de máquina
altamente optimizado.



Tiempo de ejecución
rápido y eficiente.

Manejo de **grandes volúmenes** de solicitudes de manera **simultánea** con **bajo consumo** de recursos.

Modelo de E/S no bloqueante y orientado a eventos



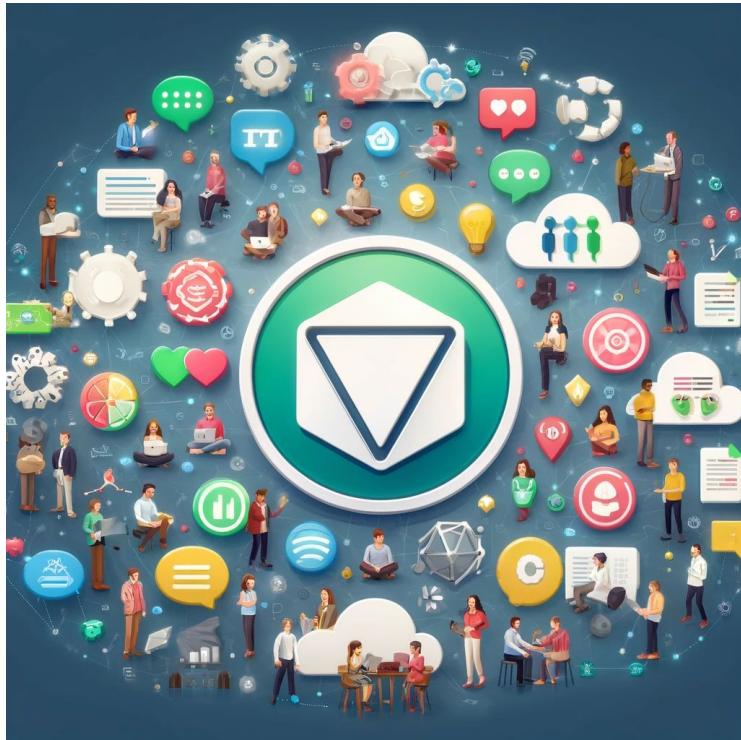
- Puede manejar múltiples solicitudes de forma simultánea sin bloquear el hilo de ejecución.
 - Ideal para aplicaciones en tiempo real

Ecosistema robusto de paquetes y módulos a través de npm



- Mayor registro de paquetes de código abierto del mundo.
- Facilita la integración de nuevas funcionalidades
- Facilita la reutilización de código
- Acelera el desarrollo y la innovación

Comunidad activa de desarrolladores y contribuyentes



- Continuamente trabajan en mejorar y expandir la plataforma.
- Respaldado por empresas líderes en tecnología
- Garantiza un soporte continuo y actualizaciones regulares de seguridad y rendimiento.

¿Por qué Node?



Escalabilidad horizontal



Adopción de tecnologías emergentes

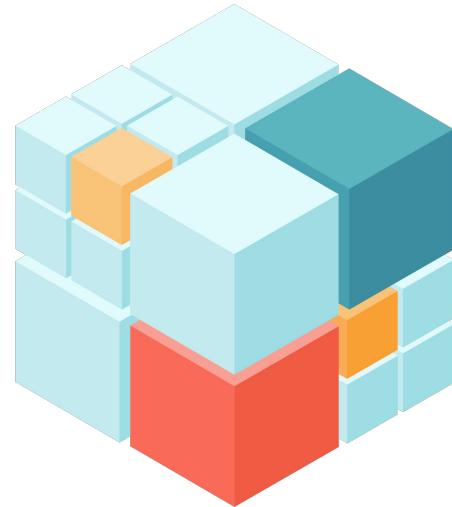
IPFS

- Tecnología de almacenamiento en red
- Almacenamiento descentralizado
- Capacidad escalable



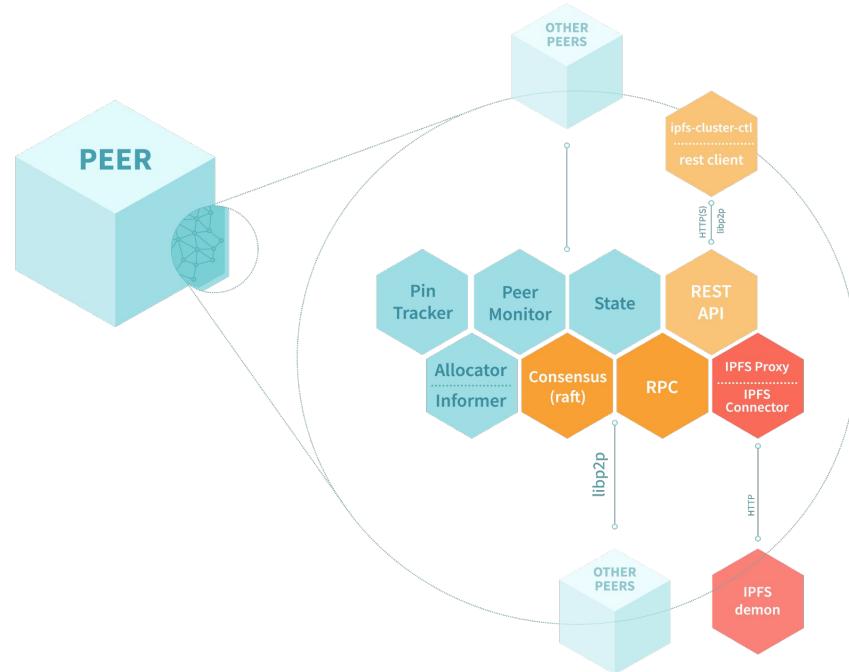
IPFS Cluster

- Sistema orquestación IPFS
- Encargado de distribuir archivos
- Caché de archivos



IPFS + IPFS Cluster

- Red privada UPC Cloud
- Acceso rápido
- Escalabilidad



Kubernetes



Dar soporte a IPFS

Kubernetes

| NAME | STATUS | ROLES | AGE |
|-----------|--------|---------------|-----|
| ninetales | Ready | control-plane | 24d |
| persian | Ready | <none> | 13d |
| pikachu | Ready | <none> | 24d |
| venomoth | Ready | <none> | 15d |

```
alumne@ninetales:~/kube-ipfs$ kubectl get sts
NAME           READY   AGE
ipfs-cluster   2/2     13d
```

```
alumne@ninetales:~/kube-ipfs$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
ipfs-cluster-0  2/2     Running   20 (174m ago)  13d
ipfs-cluster-1  2/2     Running   69 (4h44m ago)  13d
```

Kubernetes

```
volumeClaimTemplates:  
- metadata:  
    name: cluster-storage  
  spec:  
    storageClassName: local-storage  
    accessModes: ["ReadWriteOnce"]  
    resources:  
      requests:  
        storage: 500Mi  
- metadata:  
    name: ipfs-storage  
  spec:  
    storageClassName: local-storage  
    accessModes: ["ReadWriteOnce"]  
    resources:  
      requests:  
        storage: 1Gi
```

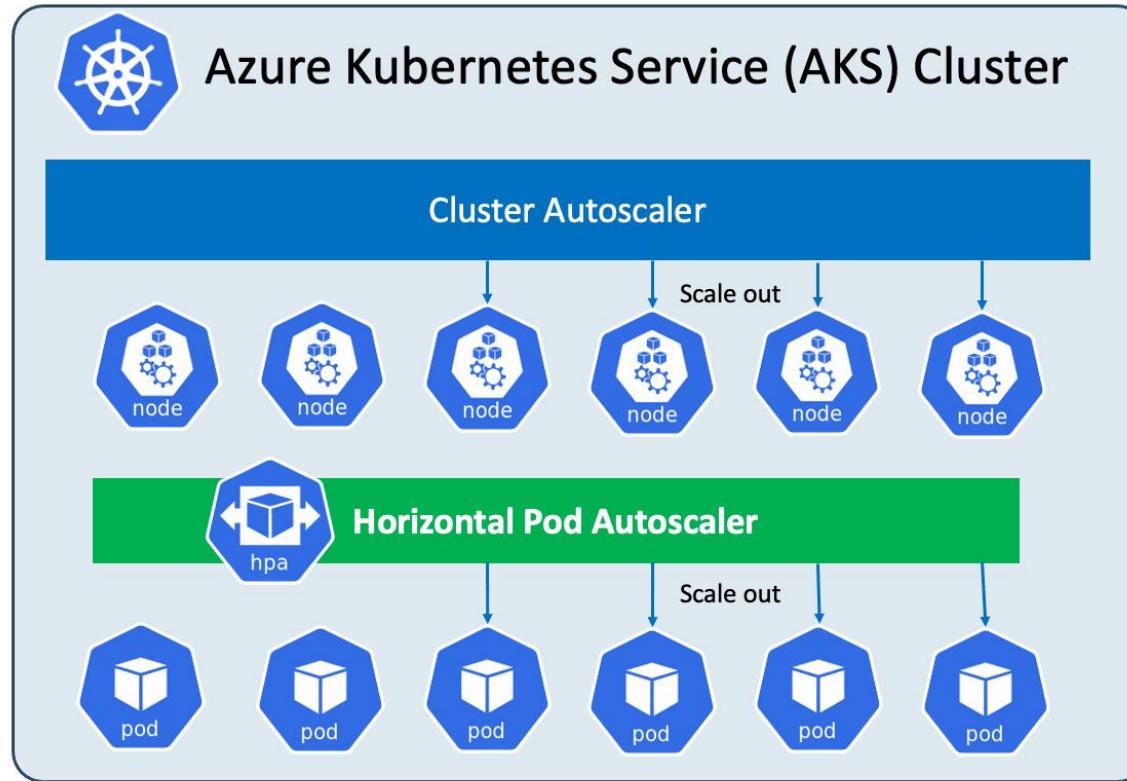
```
alumne@ninetales:~/kube-ipfs$ kubectl get pvc  
NAME           STATUS  VOLUME   CAPACITY  ACCESS MODES  STORAGECLASS  
cluster-storage-ipfs-cluster-0  Bound   cluster-pv-0  500Mi     RWO          local-storage  
cluster-storage-ipfs-cluster-1  Bound   cluster-pv-1  500Mi     RWO          local-storage  
ipfs-storage-ipfs-cluster-0   Bound   ipfs-pv-0    1Gi       RWO          local-storage  
ipfs-storage-ipfs-cluster-1   Bound   ipfs-pv-1    1Gi       RWO          local-storage
```

Kubernetes

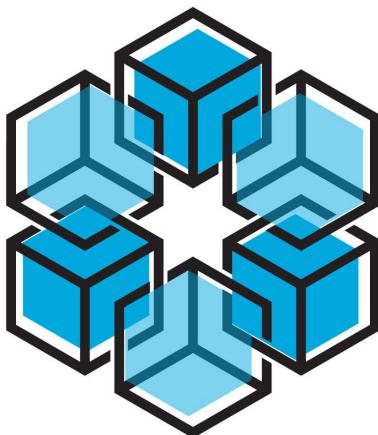
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: ipfs-pv-0
spec:
  capacity:
    storage: 1Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  storageClassName: local-storage
  local:
    path: /home/alumne/kube-pv/
nodeAffinity:
  required:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - venomoth
```

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: cluster-pv-0
spec:
  capacity:
    storage: 500Mi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  storageClassName: local-storage
  local:
    path: /home/alumne/kube-pv/
nodeAffinity:
  required:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - venomoth
```

Autoscaler



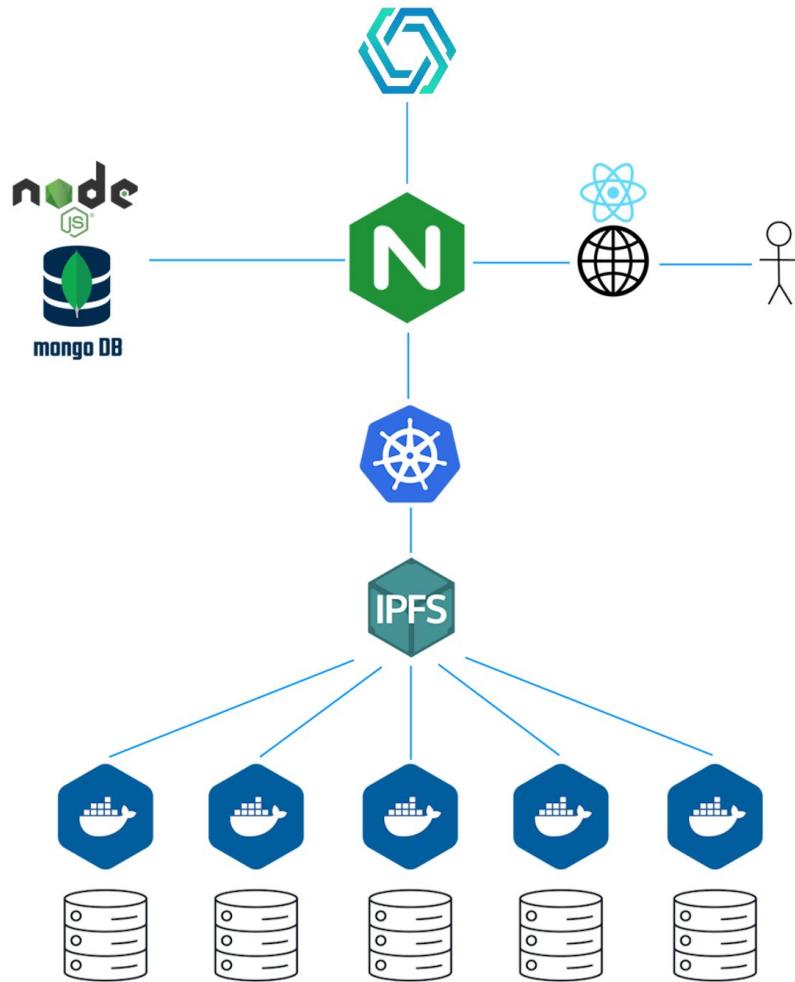
Datos Popularidad



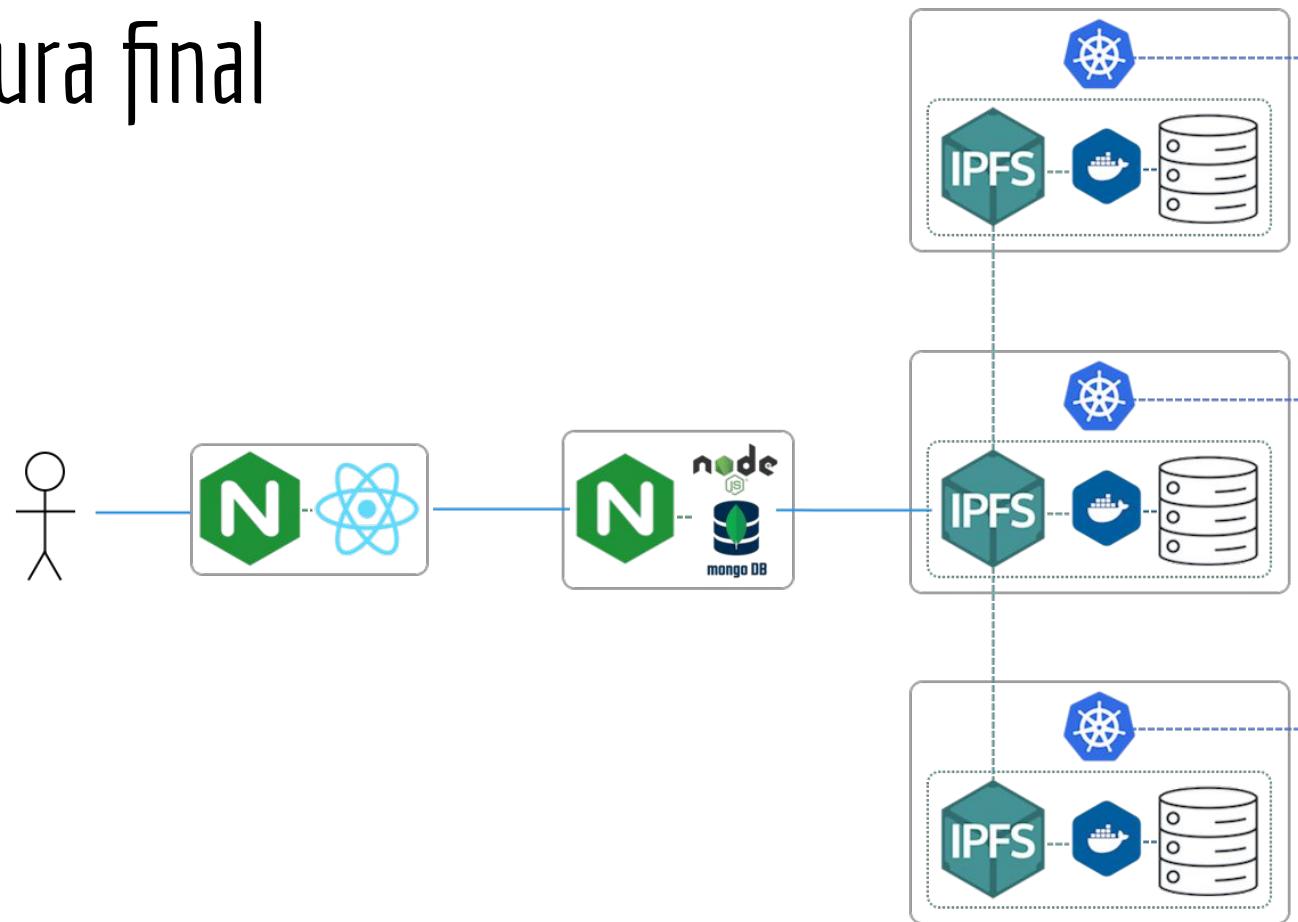
BLOCKCHAIN

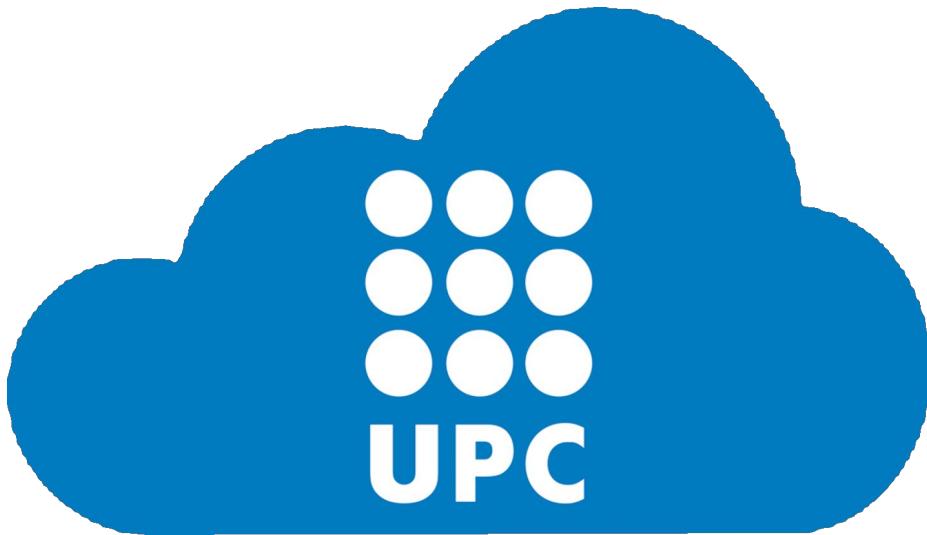


Propuesta inicial



Estructura final





UPC Cloud

Javier Armaza Bravo
Sergi Colomer Segalés
Sofía López Olivares
Pablo Rosas Roda