

Curso

Herramientas de Computación en la Nube

Clase 5

Febrero 14 de 2026

RUSTFS

Almacenamiento de Objetos tipo S3

Guía de estudio

Profesor

Alvaro Mauricio Montenegro Díaz, Ph.D.
Universidad de la Sabana

Febrero 2026

1. Propósito de la Lección

Esta lección no tiene como objetivo aprender a instalar RustFS, sino comprender:

- el modelo de almacenamiento de objetos,
- la arquitectura peer-to-peer,
- las implicaciones del teorema CAP,
- el diseño de consistencia fuerte,
- y el rol estratégico del almacenamiento en sistemas cloud modernos.

El enfoque es arquitectónico, no operacional.

2. Competencias que Debe Desarrollar el Estudiante

Al finalizar la lección, el estudiante debe ser capaz de:

1. Explicar el modelo Object Storage frente a POSIX y bases de datos.
2. Describir una arquitectura peer-to-peer real.
3. Analizar el posicionamiento de RustFS frente al teorema CAP.
4. Explicar la diferencia entre replicación y erasure coding.
5. Identificar los trade-offs arquitectónicos de un sistema distribuido.
6. Ubicar RustFS dentro de una arquitectura cloud moderna.
7. Argumentar cuándo RustFS es apropiado y cuándo no lo es.

3. Conceptos Fundamentales que Deben Dominarse

Nivel 1 – Modelo

- Object Storage
- Bucket
- Objeto
- Namespace
- Inmutabilidad

Nivel 2 – Distribución

- Peer-to-Peer

- Clúster
- Set
- Erasure Coding
- Quórum
- Read-after-write consistency

Nivel 3 – Teoría

- CAP Theorem
- Consistencia fuerte
- Disponibilidad
- Tolerancia a particiones
- Trade-offs arquitectónicos

Nivel 4 – Seguridad

- TLS
- IAM
- Cifrado en reposo
- Versionado
- WORM

4. Preguntas de Autoevaluación Técnica

Conceptuales

1. ¿Por qué RustFS no necesita un nodo maestro?
2. ¿Qué implica priorizar CP frente a AP en CAP?
3. ¿Qué diferencia operativa existe entre replicación y erasure coding?
4. ¿Por qué el almacenamiento de objetos desacopla cómputo y persistencia?
5. ¿Qué riesgos introduce una mala configuración de quórum?

Arquitectónicas

6. ¿En qué escenarios empresariales RustFS sería superior a una solución tradicional?
7. ¿Qué ocurre ante la caída simultánea de múltiples discos dentro de un set?
8. ¿Qué ventajas ofrece Rust como lenguaje para infraestructura crítica?
9. ¿Cuándo un sistema POSIX sería más adecuado que object storage?

10. ¿Cómo se integra RustFS dentro de Kubernetes?

5. Ejercicio de Pensamiento Arquitectónico

Escenario 1

Un clúster RustFS con:

- 8 nodos
- 16 discos por set
- Erasure Coding 12+4

Pregunta:

- ¿Cuántos discos pueden fallar sin pérdida de datos?
- ¿Qué pasa si fallan 5?
- ¿El sistema sigue disponible?
- ¿Se mantiene consistencia fuerte?

Escenario 2

Una empresa requiere:

- Durabilidad extremadamente alta
- Multi-tenant
- Auditoría estricta
- Cumplimiento regulatorio

Pregunta:

- ¿RustFS es adecuado?
- ¿Qué configuraciones de seguridad son imprescindibles?

6. Lecturas Complementarias Recomendadas

Para comprensión profunda:

- Brewer, Eric — CAP Theorem
- Kleppmann — Designing Data-Intensive Applications
- Ghemawat — Google File System
- Dynamo Paper — Amazon

- Documentación técnica Rust (memory safety)
- Documentación oficial RustFS

7. Nivel de Profundización Esperado

Esta lección debe permitir al estudiante:

- Entender cómo funciona un sistema distribuido real
- Analizar arquitectura más allá de marketing tecnológico
- Evaluar trade-offs estructurales
- Diseñar almacenamiento para sistemas de datos modernos

No basta con “saber que RustFS es S3-compatible”. Debe comprender:

- por qué está diseñado así,
- qué problemas resuelve,
- qué riesgos evita,
- qué riesgos introduce.

8. Reflexión Final para el Estudiante

La nube moderna no es un conjunto de productos, sino un conjunto de decisiones arquitectónicas coherentes.

RustFS es un caso de estudio sobre cómo:

- la descentralización,
- la consistencia fuerte,
- y la seguridad por construcción

pueden integrarse en una infraestructura distribuida contemporánea.

Comprender RustFS es comprender cómo se construyen los cimientos del almacenamiento cloud actual.