

# SISTEMA DE ALMACENAMIENTO DISTRIBUIDO SEGURO

CURSO: COMPUTACION PARALELA Y  
DISTRIBUIDA

PRESENTADOR: GILMAR OVIEDO CHAHUA

PROFESOR: CESAR JESUS LARA AVILA



# INTRODUCCIÓN

Desarrollo de un sistema distribuido para la gestión de archivos, con enfoque en seguridad y replicación de datos

## RELEVANCIA:

Asegura la integridad y disponibilidad de los datos en un entorno distribuido.

## OBJETIVOS

01. Implementar cifrado de datos.
02. Asegurar la replicación y disponibilidad de archivos.
03. Mejorar el rendimiento y la resiliencia del sistema.

# METODOLOGÍA

- **Enfoque Ágil:** Utilizamos una metodología ágil para iterar y mejorar continuamente.
- **Estructura de los Sprints:** El proyecto se dividió en varios sprints, cada uno con objetivos específicos.

## Sprint 1

Configuración inicial y desarrollo básico de la API.

## Sprint 2

Implementación de cifrado y replicación.

## Sprint 3

Optimización y pruebas de resiliencia y seguridad.

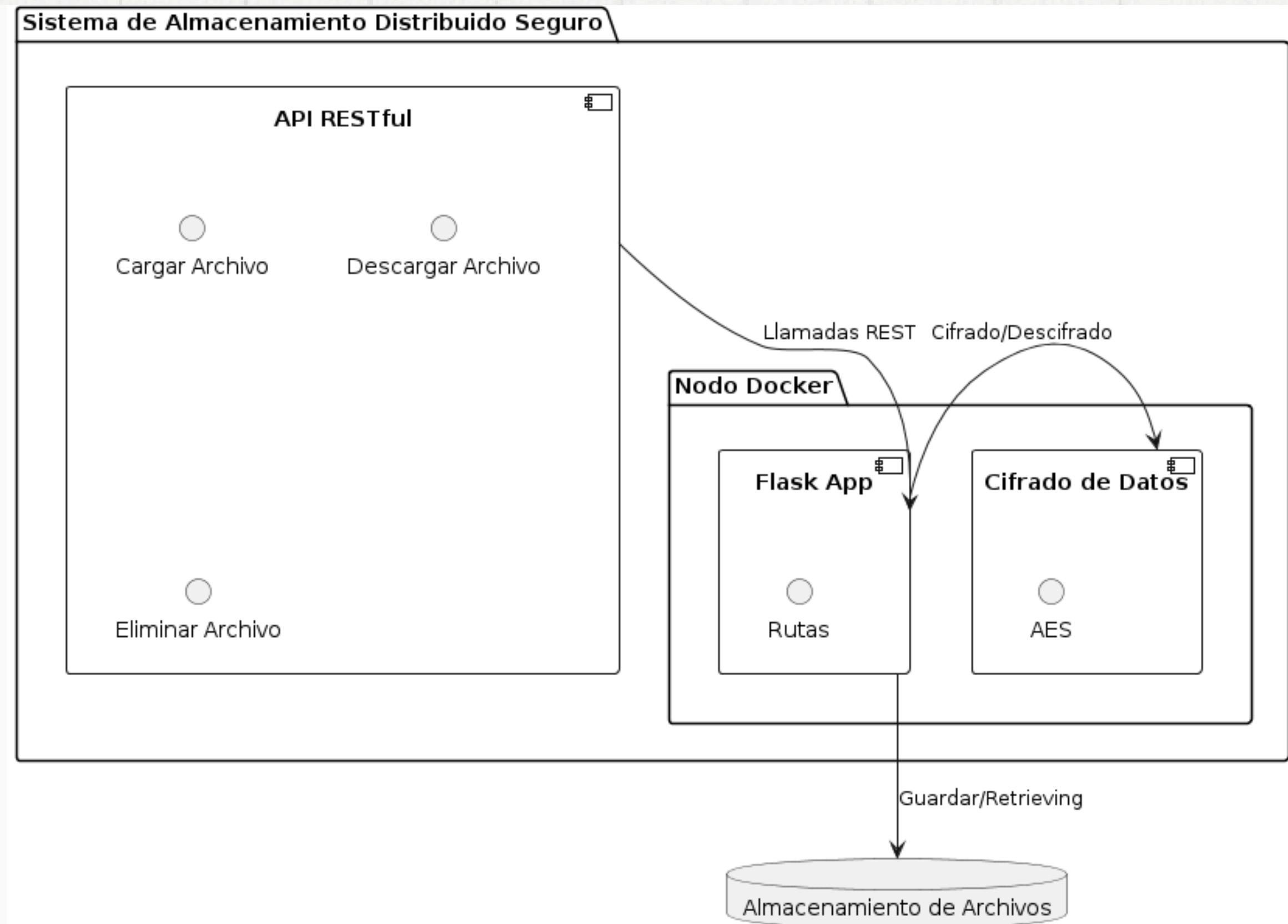
# SPRINT 1: CONFIGURACIÓN Y DESARROLLO BÁSICO

## Objetivos:

- Configurar el entorno de desarrollo.
- Implementar la API RESTful para la gestión de archivos.
- Configurar la comunicación básica entre nodos.

## Logros:

- API funcional para cargar, descargar y eliminar archivos.
- Estructura de red establecida para la comunicación entre contenedores Docker.



# SPRINT 2: CIFRADO Y REPLICACIÓN

## Objetivos:

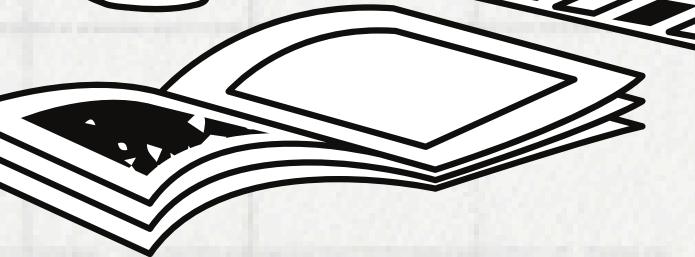
- Implementar cifrado de datos con AES.
- Desarrollar funciones de replicación entre nodos.
- Configurar la sincronización asíncrona y concurrente.

## Logros

- Cifrado y descifrado de archivos implementado.
- Replicación de archivos entre nodos asegurada.
- Sincronización eficiente usando asyncio y threading.

```
async def replicar_archivo_async(nodo, ruta_archivo, nonce, tag):  
    try:  
        form = aiohttp.FormData()  
        with open(ruta_archivo, 'rb') as file_enc:  
            data = file_enc.read()  
        form.add_field('archivo', data, filename=os.path.basename(ruta_archivo))  
        form.add_field('nonce', nonce.hex())  
        form.add_field('tag', tag.hex())  
        async with aiohttp.ClientSession() as session:  
            async with session.post(f"{nodo}/cargar", data=form) as response:  
                if response.status != 200:  
                    logging.error(f"Error replicando en {nodo}: {response.status}")  
                else:  
                    logging.debug(f"Replicación en {nodo} completada con estado {response.status}")  
    except Exception as e:  
        logging.error(f"Error replicando en {nodo}: {e}")  
  
def replicar_archivo(nodo, ruta_archivo, nonce, tag):  
    loop = asyncio.new_event_loop()  
    asyncio.set_event_loop(loop)  
    loop.run_until_complete(replicar_archivo_async(nodo, ruta_archivo, nonce, tag))  
    loop.close()
```

# SPRINT 3: OPTIMIZACIÓN Y PRUEBAS



## Objetivos:

- Optimizar el algoritmo de replicación.
- Realizar pruebas de seguridad y resiliencia.

## Logros:

```
# Cargar archivo
files = {'archivo': open('Actividad.txt', 'rb')}
response = requests.post(url_cargar, files=files)
print("Archivo cargado exitosamente. Ejecutando prueba de fallo de nodo...")
```

- Sistema optimizado para reducir la latencia.
- Pruebas de seguridad y resiliencia exitosas.

# RESULTADOS Y DEMOSTRACIÓN

## Resultados Clave:

- Seguridad y cifrado de datos efectivo.
- Alta disponibilidad y replicación de archivos.
- Rendimiento optimizado bajo condiciones de carga.

## Demostración en Vivo:

- Cargar un archivo cifrado.
- Descargar un archivo replicado.
- Mostrar la resiliencia del sistema al fallar un nodo.

```
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ docker run -d --network storage-network --name storage-node-1 -p 5000:5000 storage-system  
52c364f9182b6aacc454858756064fac87ee1edebc42fdd754fe0975869219ec  
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ docker run -d --network storage-network --name storage-node-2 -p 5001:5000 storage-system  
91b8f2b1ae1e109aff1ecb067c0ef70b1affe80a2223411a9ceebe9d2fda0bab  
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ docker run -d --network storage-network --name storage-node-3 -p 5002:5000 storage-system  
aabef2dd264875a7c16a5ddec32c382201e3f00f84fbb93cca5ceb4785298fdc  
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ |
```

```
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ curl -X POST -F 'archivo=@Actividad.txt' http://localhost:5000/cargar  
Archivo cargado y cifrado exitosamenteoviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ |
```

```
Archivo cargado y cifrado exitosamenteoviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ curl http://localhost:5000/descargar/Actividad.txt --o  
tput descargado.txt  
curl http://localhost:5000/descargar/Actividad.txt --output descargado.txt  
% Total    % Received % Xferd  Average Speed   Time   Time     Current  
          Dload Upload Total Spent   Left Speed  
100  94  100  94    0      0  8545    0  --:--:--  --:--:-- 8545  
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ |
```

```
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ curl -X DELETE http://localhost:5000/eliminar/Actividad.txt  
Archivo eliminado exitosamenteoviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ |
```

# ANÁLISIS Y EVALUACIÓN

## Logros:

- Implementación exitosa de cifrado y replicación.
- Mejoras significativas en rendimiento y seguridad.
- Mejoras Futuras:
- Implementación de una cola de tareas.
- Optimización adicional del algoritmo de replicación.

## Objetivos del Próximo Sprint:

- Pruebas de carga y estrés.
- Mecanismos de monitoreo y alerta.
- Documentación y presentación de resultados finales

```
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ python3 test_seguridad.py
Carga exitosa. Probando acceso sin clave...
Error al descargar el archivo para la prueba de seguridad.
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ |
```

```
Creating storage-node-1 ... done
Creating storage-node-2 ... done
Creating storage-node-3 ... done
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS          NAMES
78cf7f063797        src_storage-node-2   "python app.py"    28 seconds ago   Up 23 seconds   0.0.0.0:5001->5000/tcp, :::5001->5000/tcp   storage-node-2
c983885bc7dc        src_storage-node-3   "python app.py"    28 seconds ago   Up 24 seconds   0.0.0.0:5002->5000/tcp, :::5002->5000/tcp   storage-node-3
ecd36d10e84c        src_storage-node-1   "python app.py"    28 seconds ago   Up 24 seconds   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   storage-node-1
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ python3 test_resiliencia.py
```

```
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ docker exec -it storage-node-2 /bin/bash
root@78cf7f063797:/app# ls /app/cargas
Actividad.txt
root@78cf7f063797:/app# exit
exit
oviedo@oviedo:~/Evaluaci-n-C/Proyecto_Final/sprint3/src$ docker exec -it storage-node-3 /bin/bash
root@c983885bc7dc:/app# ls /app/cargas
Actividad.txt
```

# CONCLUSIÓN Y FUTURO TRABAJO

## Logros:

- Implementación exitosa de cifrado y replicación.
- Mejoras significativas en rendimiento y seguridad.

## Mejoras Futuras:

- Implementación de una cola de tareas.
- Optimización adicional del algoritmo de replicación.

## Objetivos del Próximo Sprint:

- Pruebas de carga
- Mecanismos de monitoreo y alerta.
- Documentación y presentación de resultados finales.

# **Gracias**