

# Implementación de algoritmos distribuidos en redes P2P

## Objetivos generales:

- Desarrollar e implementar algoritmos distribuidos en una red P2P utilizando ZeroMQ.
- Optimizar el rendimiento y escalabilidad de estos algoritmos.
- Asegurar la tolerancia a fallos en la red P2P.

## Objetivos específicos:

- Implementar los algoritmos Gather y Broadcast.
- Desarrollar un panel de control para monitorear la red en tiempo real.
- Realizar pruebas de rendimiento y optimización.



# Metodología

Enfoque ágil:

- Uso de metodologías ágiles para estructurar el trabajo en sprints.
- Revisión y adaptación continua de las tareas según los resultados y retroalimentación.

Estructura de sprints:

- Sprint 1: Implementación básica y pruebas iniciales.
- Sprint 2: Optimización y pruebas de estrés.
- Sprint 3: Desarrollo del panel de control y documentación.

Herramientas y tecnologías utilizadas:

- ZeroMQ: Biblioteca de mensajería asincrónica para comunicación en la red P2P.
- Flask: Framework de Python para el desarrollo del servidor web.
- Chart.js: Biblioteca de JavaScript para visualización de gráficos.
- Zlib: Biblioteca para compresión de datos.
- Threading en Python: Para manejar concurrencia y paralelismo.

# Objetivos y logros del Sprint 1

## Objetivos:

- Configuración del entorno de desarrollo.
- Implementación inicial de los algoritmos Gather y Broadcast.
- Realización de pruebas iniciales de funcionalidad.

## Logros:

- Entorno configurado exitosamente.
- Algoritmos Gather y Broadcast implementados.
- Pruebas iniciales exitosas, asegurando la correcta transmisión de mensajes.

## Detalles adicionales:

- Configuración del entorno: Utilización de 'virtualenv' para aislar dependencias y asegurar un entorno de desarrollo limpio.
- Pruebas iniciales: Validación de la correcta transmisión de mensajes y medición básica de latencia.



# Objetivos y logros del Sprint 2

## Objetivos:

- Optimización de los algoritmos Gather y Broadcast.
- Implementación de técnicas de replicación y recuperación.
- Realización de pruebas de estrés.

## Logros:

- Algoritmos optimizados para mejor rendimiento.
- Técnicas de replicación implementadas, mejorando la tolerancia a fallos.
- Pruebas de estrés demostraron la escalabilidad del sistema con múltiples nodos.

## Detalles adicionales:

- Optimización de algoritmos: Reducción de latencia en un 25% para Broadcast y 30% para Gather.
- Pruebas de estrés: Evaluación con hasta 100 nodos, midiendo el uso de ancho de banda y la latencia bajo carga.





# Objetivos y logros del Sprint 3

## Objetivos:

- Desarrollar un panel de control para monitorear la red en tiempo real.
- Documentar completamente el sistema y los algoritmos implementados.
- Preparar y realizar una presentación sobre los resultados obtenidos.

## Logros:

- Panel de control desarrollado utilizando Flask y Chart.js.
- Documentación detallada del sistema y algoritmos.
- Presentación preparada, incluyendo demostraciones en vivo.

## Detalles adicionales:

- Panel de control: Capacidad para agregar nodos dinámicamente y visualizar métricas en tiempo real.
- Documentación: Guías detalladas de instalación, configuración y uso del sistema.

# Resultados

## Funcionalidades desarrolladas:

- Implementación de algoritmos Gather y Broadcast.
- Optimización de algoritmos para mejorar el rendimiento.
- Panel de control para monitorear y gestionar la red P2P en tiempo real.

## Resultados de pruebas:

- Latencia y ancho de banda: Se midió el rendimiento bajo diferentes condiciones de red.
- Replicación: Verificación de la replicación de datos entre nodos.
- Estrés: Evaluación de la escalabilidad con múltiples nodos.

## Detalles adicionales:

- Pruebas de replicación: Aseguraron que los datos se mantuvieran íntegros y disponibles incluso en caso de fallos de nodos.
- Pruebas de latencia: Resultados consistentemente bajos y estables, validando la eficiencia de los algoritmos.



# Demostración en vivo

Demostración en tiempo real:

- Ejecución del panel de control.
- Visualización de métricas de latencia y ancho de banda.
- Gestión de nodos en la red P2P.

Detalles adicionales:

- Preparación de la demostración: Ensayos previos para asegurar que todos los componentes funcionen correctamente.
- Interactividad: Posibilidad de que la audiencia vea en tiempo real cómo se agregan nodos y cómo se actualizan las métricas.

# Análisis y evaluación

## Lecciones aprendidas:

- Importancia de una planificación detallada y pruebas exhaustivas.
- Manejo de concurrencia y replicación de datos en sistemas distribuidos.

## Desafíos y soluciones:

- Integración de datos en tiempo real: Utilización de AJAX y Chart.js para actualización dinámica.
- Optimización de rendimiento: Implementación de compresión de datos y técnicas de replicación.

## Detalles adicionales:

- Monitoreo y gestión: La implementación del panel de control facilitó la identificación y resolución de problemas en tiempo real.



# Conclusión y futuro trabajo

Resumen de los logros:

- Implementación exitosa de algoritmos distribuidos en red P2P.
- Desarrollo de un panel de control eficiente y funcional.
- Pruebas de rendimiento y optimización realizadas exitosamente.

Posibles mejoras y expansiones futuras:

- Mejorar la visualización de datos y la gestión de nodos.
- Continuar optimizando los algoritmos para mayor escala de nodos y datos.
- Implementación de nuevas funcionalidades en el panel de control.

Detalles adicionales:

- Expansión futura: Investigación sobre la integración de machine learning para predicción de fallos y optimización automática del rendimiento.

# Preguntas y respuestas

Interacción con la audiencia:

- Espacio para responder preguntas y recibir retroalimentación.
- Invitación a la audiencia para sugerir mejoras y nuevas ideas.

Detalles adicionales:

- Preparación para preguntas: Anticipación de posibles preguntas y preparación de respuestas detalladas.

**GRACIAS**

