

Qin - Cluster

Autores:

Barrabino, Diego

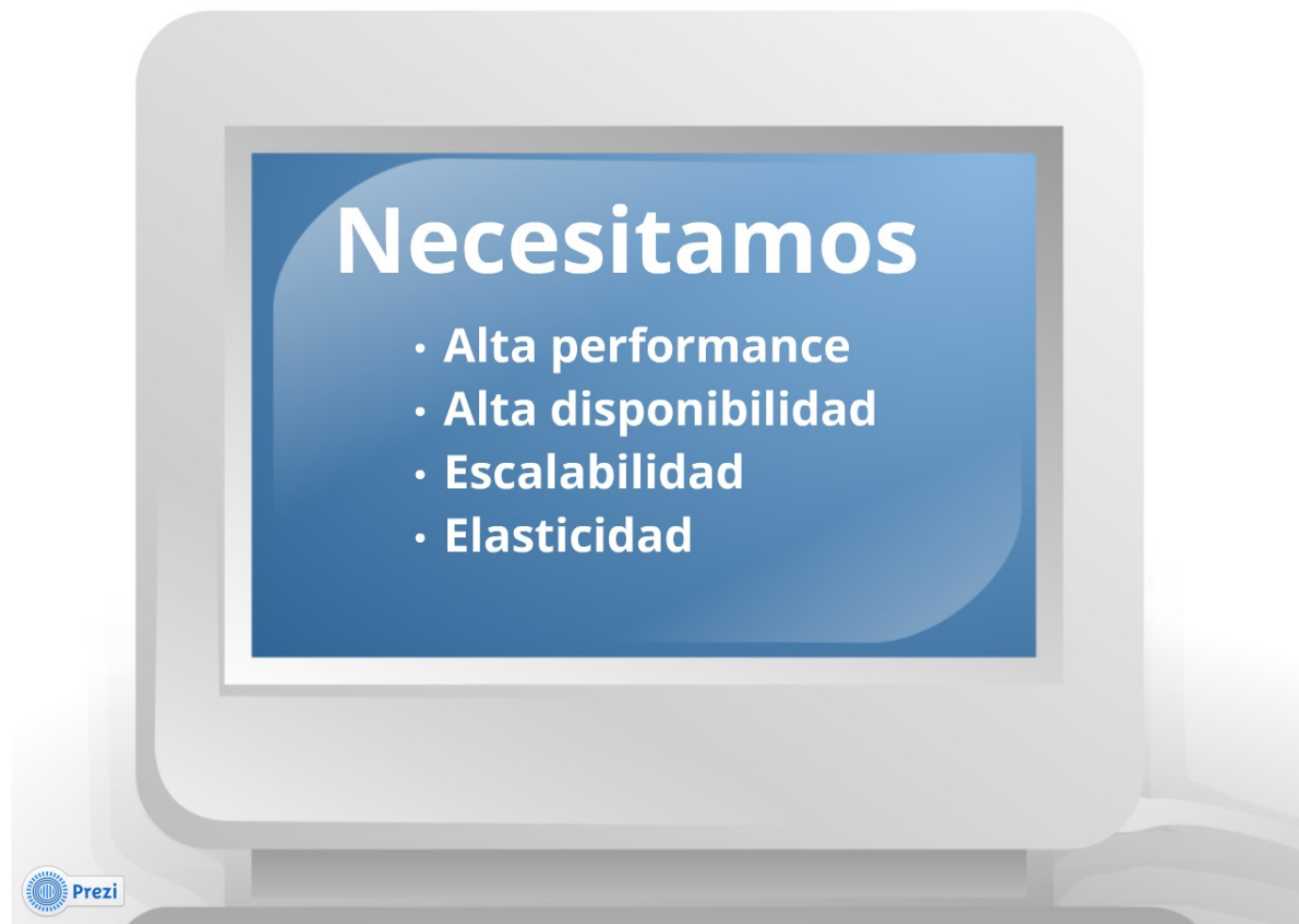
Moreyra, Martín Jorge

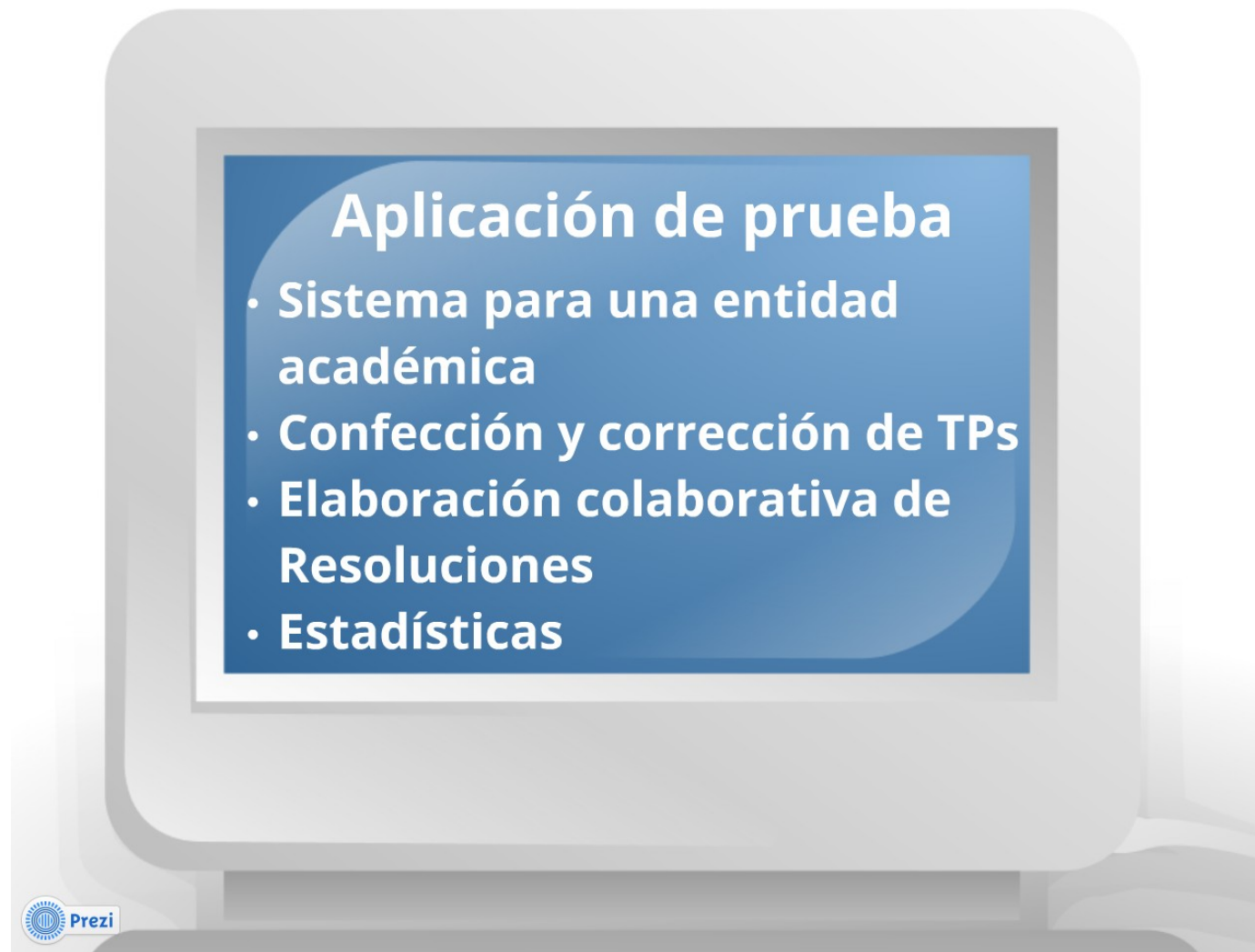


Problema

- **Sistemas críticos**
- **Gran carga de usuarios**
- **Expectativa y compromiso de servicio**

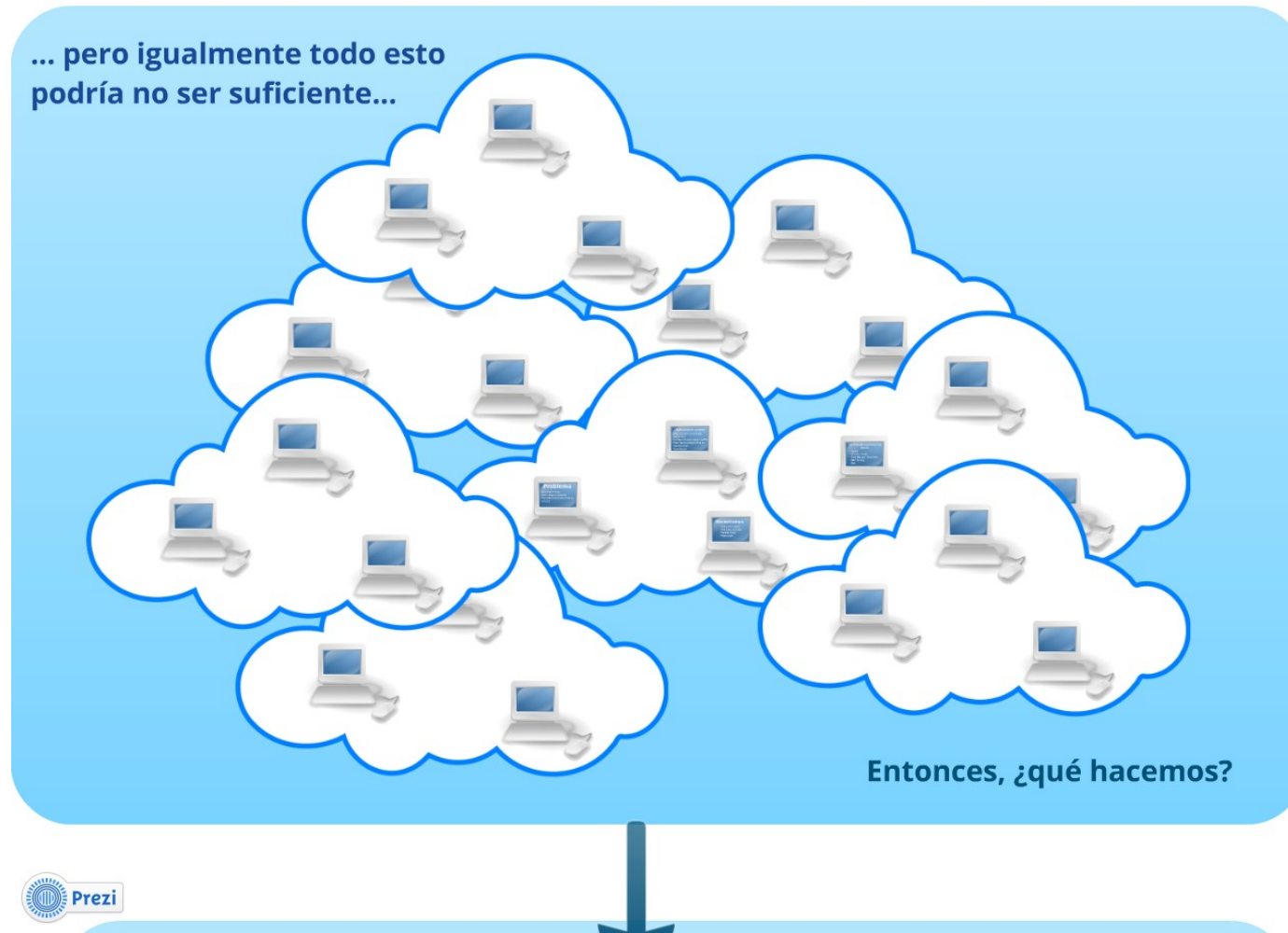


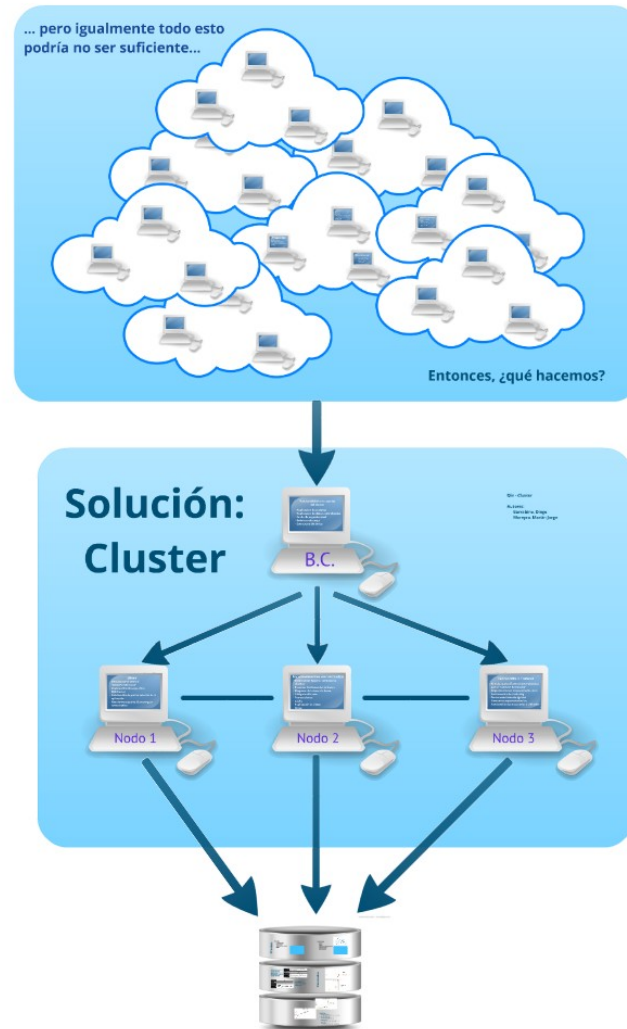


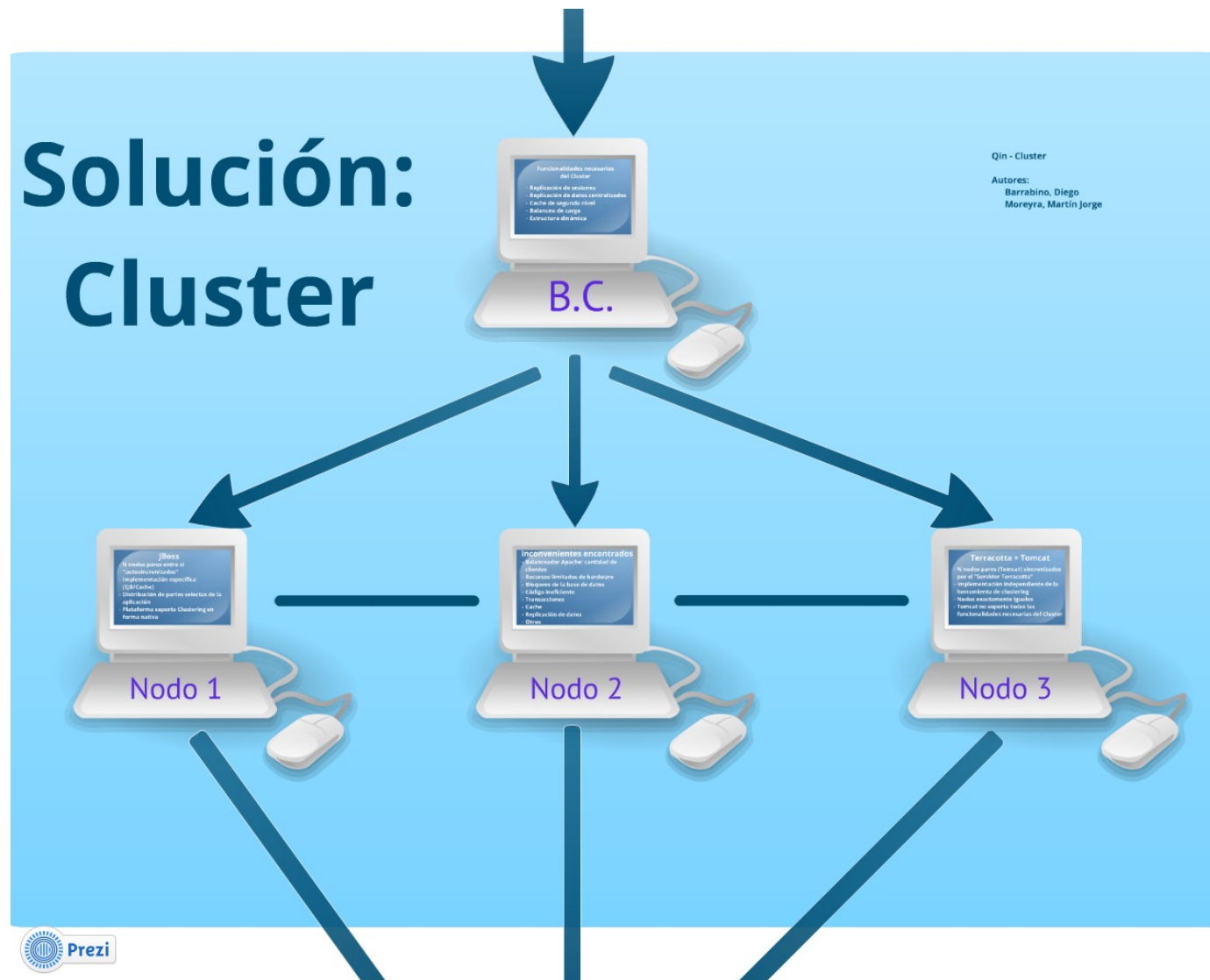


04 - Aplicación de prueba









08 - Solución: Cluster (2)

Funcionalidades necesarias del Cluster

- Replicación de sesiones
- Replicación de datos centralizados
- Cache de segundo nivel
- Balanceo de carga
- Estructura dinámica



JBoss

- N nodos pares entre sí "autosincronizados"
- Implementación específica (EJB/Cache)
- Distribución de partes selectas de la aplicación
- Plataforma soporta Clustering en forma nativa



Terracotta + Tomcat

- N nodos pares (Tomcat) sincronizados por el "Servidor Terracotta"
- Implementación independiente de la herramienta de clustering
- Nodos exactamente iguales
- Tomcat no soporta todas las funcionalidades necesarias del Cluster



Inconvenientes encontrados

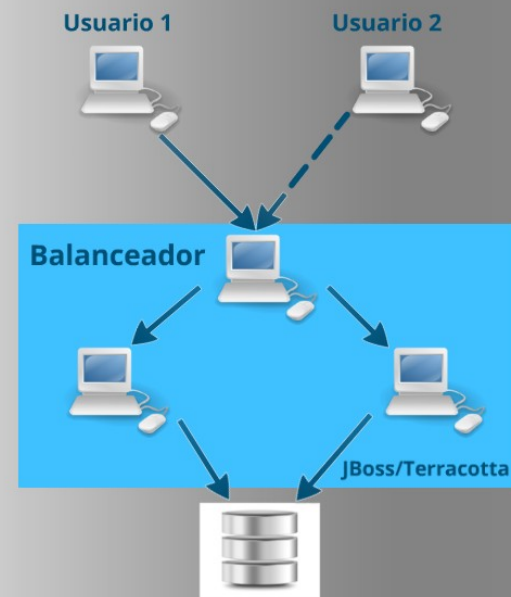
- **Balanceador Apache: cantidad de clientes**
- **Recursos limitados de hardware**
- **Bloqueos de la base de datos**
- **Código ineficiente**
- **Transacciones**
- **Cache**
- **Replicación de datos**
- **Otros**



Pruebas

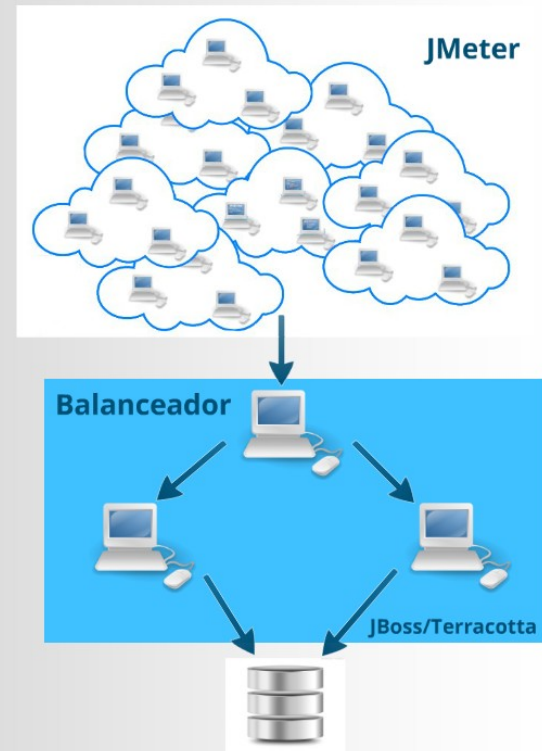
Manuales

- Disponibilidad - desconexión
- Replicación de sesion
- Cache

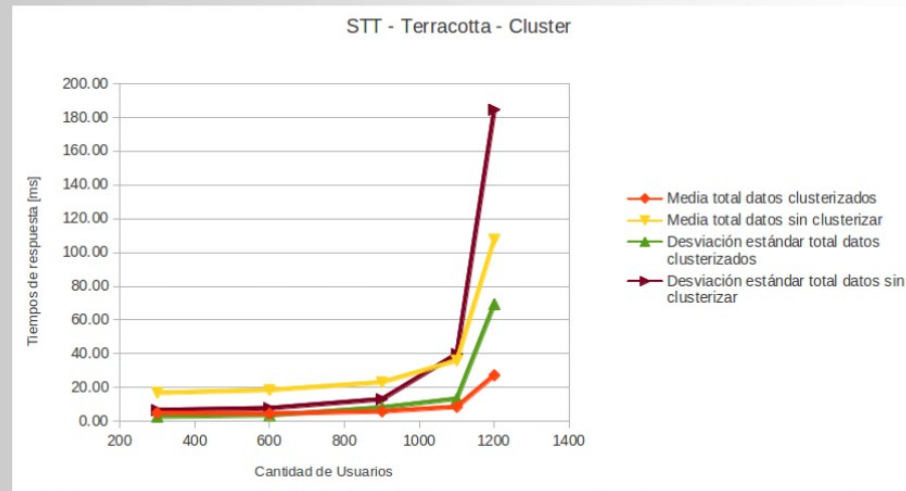
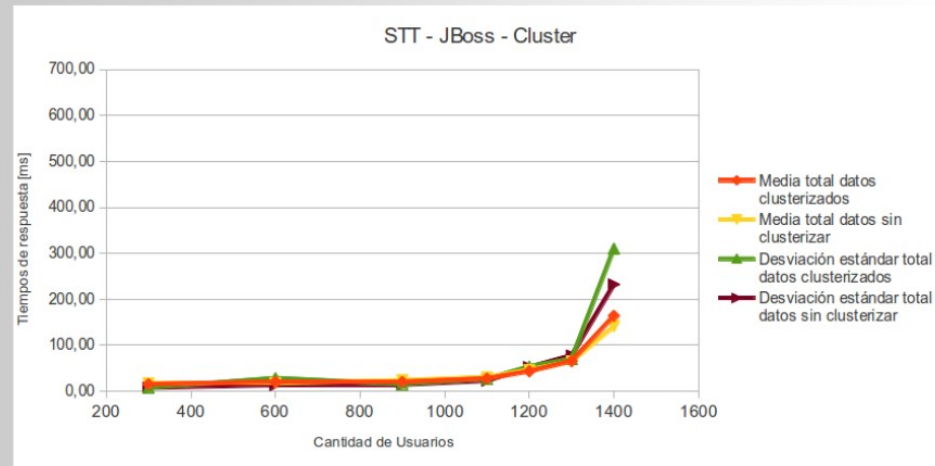


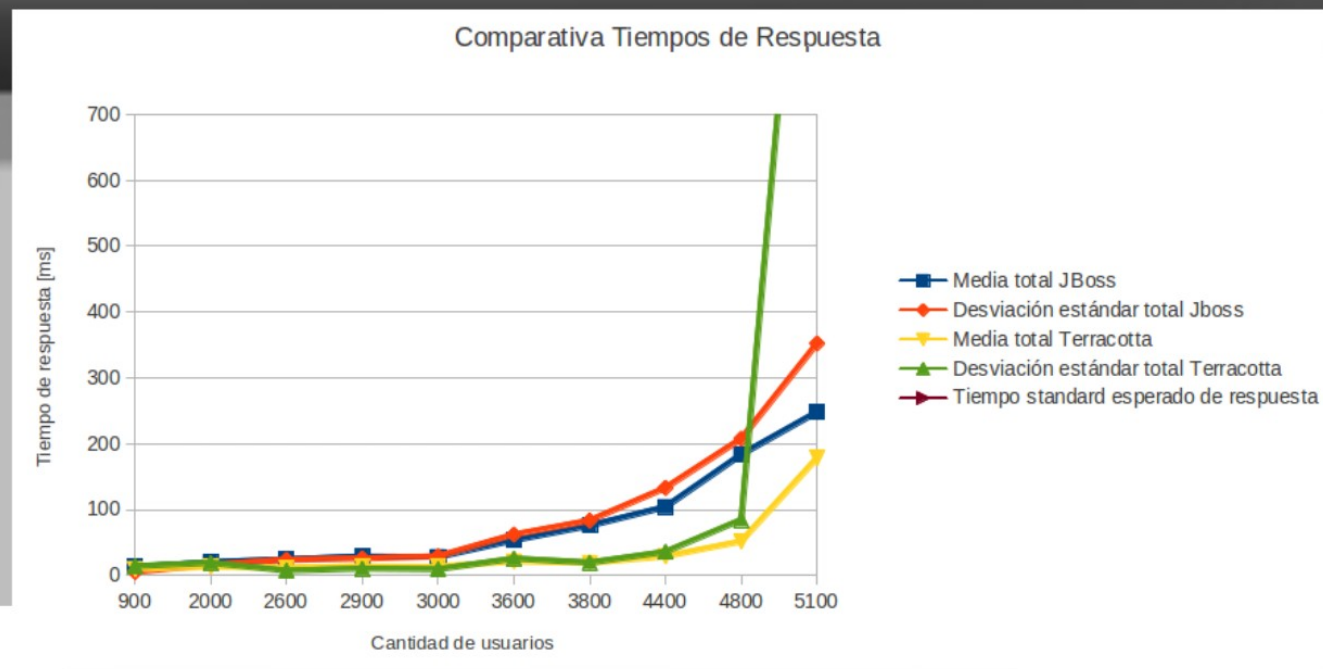
Automáticas

- Stress
- Carga
- Carga datos sincronizados
- Carga datos no sincronizados

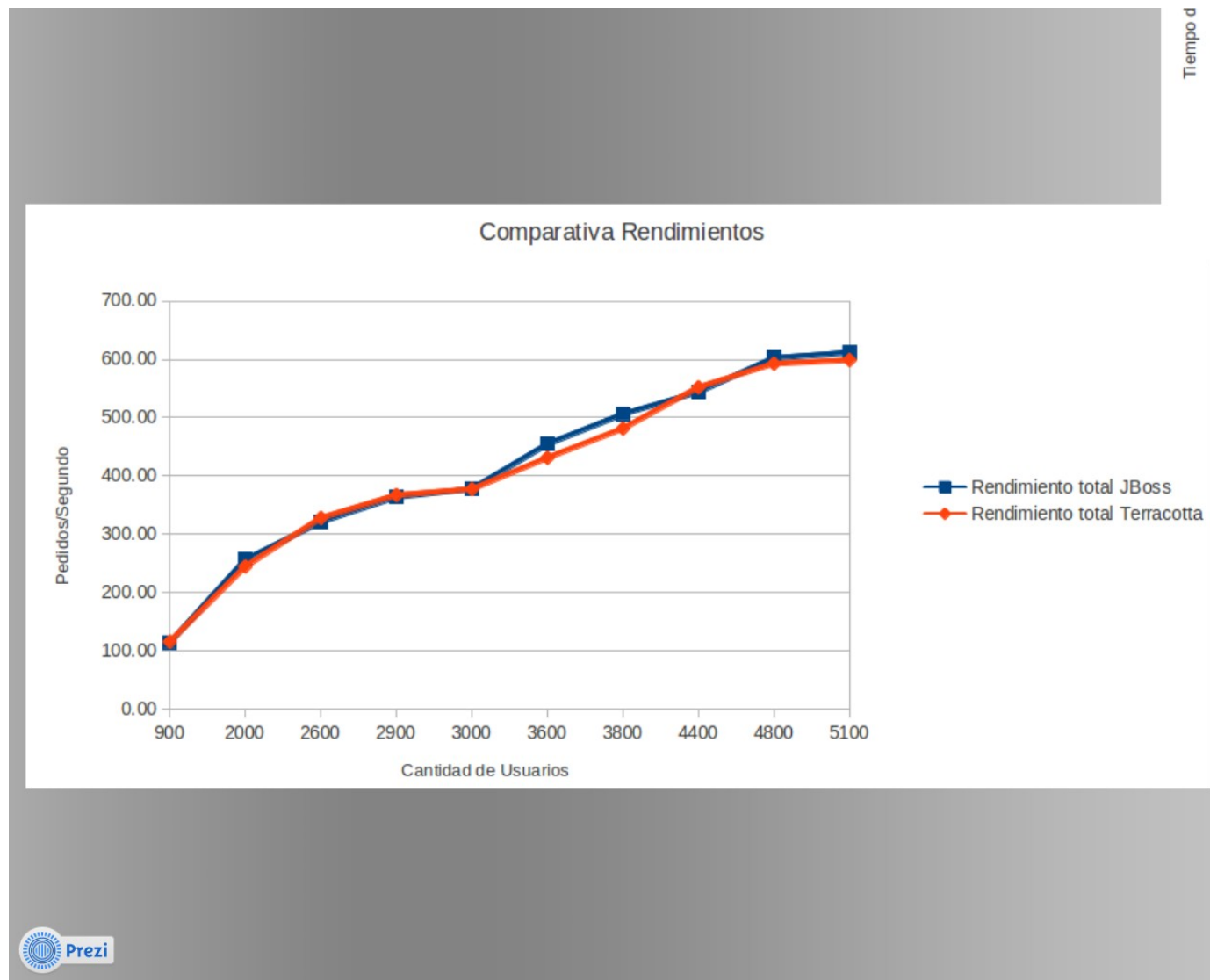


Resultados

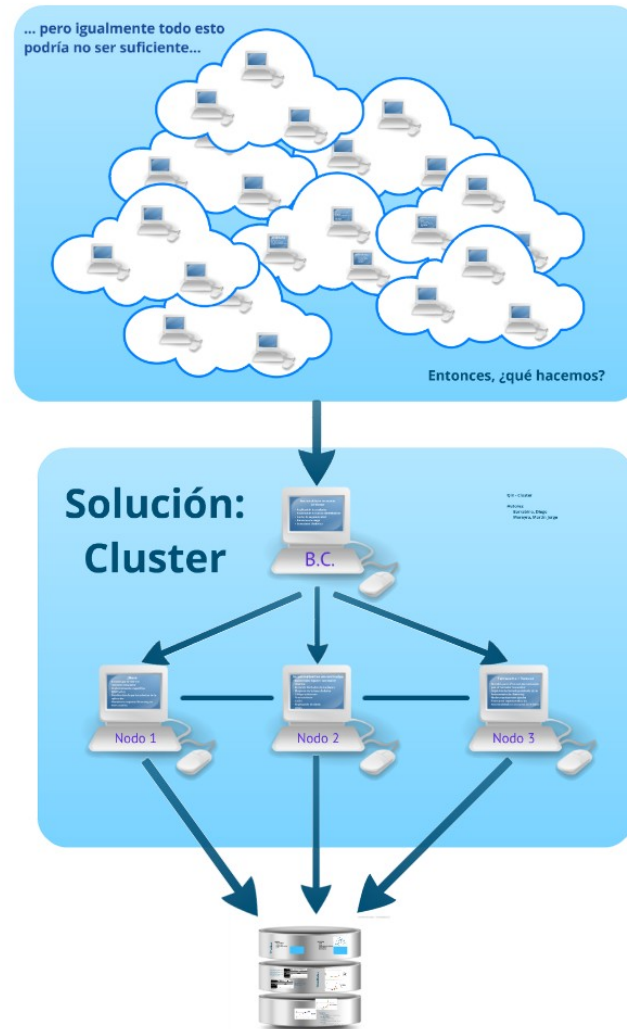




16 - Comparativa Tiempos de respuesta



17 - Comparativa - Rendimientos



18 - Cierre y paso a la demo

Conclusiones

- **Alta performance**

4500 - 6000 usuarios en entorno de pruebas

- **Alta disponibilidad**

Redundancia datos y sesiones

- **Escalabilidad**

Herramientas que permiten agregar recursos

- **Elasticidad**

JBoss provee una arquitectura más elástica



