

Esquema de Distribución Sistema de Gestión Hospitalaria Inteligente (HIS)

Versión: 1.0
Fecha: Diciembre 2024
Curso: Bases de Datos

1. Arquitectura Distribuida del Sistema HIS+

1.1. Concepto Fundamental: Fragmentación Horizontal (Sharding)

El sistema HIS+ implementa una base de datos distribuida siguiendo el principio de “una biblioteca con múltiples sucursales”:

- **Sistema Centralizado Tradicional:** Un solo servidor almacena todos los datos del país
- **Sistema Distribuido HIS+:** Tres nodos independientes pero interconectados, cada uno gestionando datos locales pero accesibles globalmente mediante **Fragmentación Horizontal por id_sede**

1.2. Topología de Red

Nodo	Ubicación Física	Rol en la Red	Tipo de Despliegue	Dirección IP
Nodo Central	Bogotá (Centro de Datos)	Maestro/Coordinador	Nube Pública (AWS/GCP)	192.168.1.100
Nodo Norte	Medellín (Sede Clínica)	Secundario	Nube Privada	192.168.2.100
Nodo Sur	Cali (Urgencias Móvil)	Secundario	On-premise/Local	192.168.3.100

Cuadro 1: Topología de nodos de la red hospitalaria

2. Estrategia de Fragmentación de Datos

2.1. Fragmentación Horizontal por Sede (Sharding)

Se implementa fragmentación horizontal basada en el campo `id_sede`:

```
1 -- Criterio de fragmentación para todas las tablas transaccionales
2 WHERE id_sede = {1, 2, 3}
```

2.2. Tablas Fragmentadas por Sede

Nodo	Datos Locales (<code>id_sede = ?</code>)
Nodo Central (Bogotá)	<code>id_sede = 1</code>
Nodo Norte (Medellín)	<code>id_sede = 2</code>
Nodo Sur (Cali)	<code>id_sede = 3</code>

Cuadro 2: Distribución de datos por sede

Tablas que siguen fragmentación horizontal:

- Citas - Fragmentada por sede

- Empleados - Fragmentada por sede
- Equipamiento - Fragmentada por sede
- Inventario_Farmacia - Fragmentada por sede
- Reportes_Generados - Fragmentada por sede

2.3. Catálogos Replicados Manualmente

Las tablas maestras/catálogos existen completas en cada nodo para garantizar consistencia referencial:

Tabla	Estrategia
Roles	Copia completa en los 3 nodos
Especialidades	Copia completa en los 3 nodos
Enfermedades	Copia completa en los 3 nodos
Catalogo_Medicamentos	Copia completa en los 3 nodos
Sedes_Hospitalarias	Copia completa en los 3 nodos

Cuadro 3: Tablas maestras replicadas manualmente

2.4. Datos Replicados Críticos

Datos que existen en múltiples nodos por necesidad operativa:

Tabla	Estrategia	Justificación
Historias_Clinicas	Vistas distribuidas	Acceso médico entre sedes para continuidad de atención
Auditoria_Accesos	Replicación parcial	Seguridad y trazabilidad de accesos en toda la red
Pacientes	Vistas distribuidas	Pacientes pueden atenderse en múltiples sedes

Cuadro 4: Estrategia de replicación de datos críticos

3. Implementación Técnica con PostgreSQL FDW

3.1. Configuración del Foreign Data Wrapper

```

1 -- En el Nodo Central (Bogot ) configurar acceso a nodos remotos
2
3 -- 1. Habilitar extensi n
4 CREATE EXTENSION IF NOT EXISTS postgres_fdw;
5
6 -- 2. Crear servidores externos para cada nodo secundario
7 CREATE SERVER nodo_medellin_server
8 FOREIGN DATA WRAPPER postgres_fdw
9 OPTIONS (
10   host '192.168.2.100',
11   dbname 'hisplus_db',
12   port '5432'
13 );
14
15 CREATE SERVER nodo_cali_server
16 FOREIGN DATA WRAPPER postgres_fdw
17 OPTIONS (
18   host '192.168.3.100',
19   dbname 'hisplus_db',
20   port '5432'
21 );
22
23 -- 3. Mapeo de usuarios para autenticaci n cruzada
24 CREATE USER MAPPING FOR administrador
25 SERVER nodo_medellin_server

```

```

26 OPTIONS (user 'admin_norte', password 'admin123_secure');
27
28 CREATE USER MAPPING FOR administrador
29 SERVER nodo_cali_server
30 OPTIONS (user 'admin_moreno', password 'admin123_secure');

```

3.2. Creación de Tablas Externas

```

1 -- 4. Crear tablas externas para acceder a datos remotos
2
3 -- Tablas del Nodo Medellín
4 CREATE FOREIGN TABLE citas_medellin (
5     id_cita BIGINT,
6     id_sede INT,
7     id_dept INT,
8     id_emp INT,
9     cod_pac INT,
10    fecha_hora TIMESTAMP,
11    fecha_hora_solicitada TIMESTAMP,
12    tipo_servicio VARCHAR(50),
13    estado VARCHAR(20),
14    motivo VARCHAR(200)
15 ) SERVER nodo_medellin_server
16 OPTIONS (schema_name 'public', table_name 'citas');
17
18 -- Tablas del Nodo Cali
19 CREATE FOREIGN TABLE citas_cali (
20     id_cita BIGINT,
21     id_sede INT,
22     id_dept INT,
23     id_emp INT,
24     cod_pac INT,
25     fecha_hora TIMESTAMP,
26     fecha_hora_solicitada TIMESTAMP,
27     tipo_servicio VARCHAR(50),
28     estado VARCHAR(20),
29     motivo VARCHAR(200)
30 ) SERVER nodo_cali_server
31 OPTIONS (schema_name 'public', table_name 'citas');

```

3.3. Vistas Distribuidas Consolidadas

```

1 -- 5. Crear vistas distribuidas que consoliden datos de los 3 nodos
2
3 CREATE VIEW vista_citas_consolidadas AS
4 SELECT * FROM citas WHERE id_sede = 1 -- Datos locales Bogotá
5 UNION ALL
6 SELECT * FROM citas_medellin -- Datos remotos Medellín
7 UNION ALL
8 SELECT * FROM citas_cali; -- Datos remotos Cali
9
10 -- Vista para historias clínicas consolidadas
11 CREATE VIEW vista_historias_consolidadas AS
12 SELECT * FROM Historias_Clinicas hc
13 WHERE EXISTS (SELECT 1 FROM Citas c WHERE c.id_cita = hc.id_cita AND c.id_sede = 1)
14 UNION ALL
15 SELECT * FROM historias_medellin hm
16 WHERE EXISTS (SELECT 1 FROM citas_medellin cm WHERE cm.id_cita = hm.id_cita)
17 UNION ALL
18 SELECT * FROM historias_cali hcal
19 WHERE EXISTS (SELECT 1 FROM citas_cali ccsl WHERE ccsl.id_cita = hcal.id_cita);

```

4. Mapa de Nodos y Estrategia de Sincronización

4.1. Mapa de Nodos

NODO CENTRAL (Bogotá)
Rol: Maestro/Coordinador
IP: 192.168.1.100:5432
Datos: id_sede = 1

NODO NORTE (Medellín)	CONEXIÓN FDW	VISTAS DISTRIBUIDAS
Rol: Secundario	postgres_fdw	• Historias
IP: 192.168.2.100		• Citas
Datos: id_sede=2		• Pacientes

NODO SUR (Cali)	CONEXIÓN FDW	SINCRONIZACIÓN MANUAL
Rol: Secundario	postgres_fdw	• Catálogos
IP: 192.168.3.100		• Auditoría
Datos: id_sede=3		

Figura 1: Arquitectura de nodos y conexiones

4.2. Estrategia de Sincronización

4.2.1. Sincronización Automática (FDW)

- **Tipo:** Consultas en tiempo real via postgres_fdw
- **Latencia:** Baja (depende de la red)
- **Datos:** Tablas transaccionales (Citaciones, Historias)
- **Ventaja:** Datos siempre actualizados

4.2.2. Sincronización Manual (Catálogos)

- **Tipo:** Scripts SQL periódicos
- **Frecuencia:** Diaria o según cambios
- **Datos:** Tablas maestras (Medicamentos, Enfermedades)
- **Ventaja:** Control total sobre consistencia

4.2.3. Sincronización por Vistas

- **Tipo:** Vistas distribuidas
- **Datos:** Datos consolidados para reportes
- **Ejemplo:** vista.historias_consolidadas

5. Flujo de Consultas Distribuidas

5.1. Ejemplo 1: Consulta Local

Usuario en Bogotá consulta citas locales:

1. Cliente → Nodo Central (Bogotá)
2. SELECT * FROM Citas WHERE id_sede = 1
3. Resultado directo (sin consultas remotas)

5.2. Ejemplo 2: Consulta Distribuida

Administrador consulta todas las citas del sistema:

1. Cliente → Nodo Central (Bogotá)
2. SELECT * FROM vista_citas_consolidadas
3. Nodo Central ejecuta:
 - SELECT local (id_sede = 1)
 - Consulta remota a Medellín via FDW
 - Consulta remota a Cali via FDW
4. Consolida resultados
5. Devuelve datos unificados

5.3. Ejemplo 3: Consulta Analítica

Reporte de enfermedades por sede:

1. Cliente → Nodo Central
2. Ejecuta vista_enfermedades_por_sede
3. Vista accede a:
 - Diagnósticos locales (Bogotá)
 - Diagnósticos remotos (Medellín via FDW)
 - Diagnósticos remotos (Cali via FDW)
4. Agrupa por sede y enfermedad
5. Genera reporte consolidado

6. Consideraciones de Red y Seguridad

6.1. Configuración de Red

Parámetro	Valor	Descripción
Puerto PostgreSQL	5432	Puerto estándar
Ancho de banda mínimo	10 Mbps	Para consultas FDW
Latencia máxima	200ms	Para tiempo real
Protocolo	SSL/TLS	Conexiones encriptadas

6.2. Reglas de Firewall

```
1 # Nodo Central (Bogot )
2 Permitir entrada: 192.168.2.100:5432 (Medell n)
3 Permitir entrada: 192.168.3.100:5432 (Cali)
4 Permitir salida: Todas las conexiones
5
6 # Nodo Norte (Medell n)
7 Permitir entrada: 192.168.1.100:5432 (Bogot )
8 Permitir salida: 192.168.1.100:5432
9
10 # Nodo Sur (Cali)
11 Permitir entrada: 192.168.1.100:5432 (Bogot )
12 Permitir salida: 192.168.1.100:5432
```

6.3. Seguridad de Conexiones FDW

- Autenticación con usuarios específicos por nodo
- Contraseñas almacenadas encriptadas
- Conexiones SSL entre nodos Registro de todas las conexiones en `Auditoria_Accesos`

7. Plan de Contingencia y Recuperación

7.1. Escenario 1: Caída del Nodo Central

1. **Detección:** Monitoreo continuo de conectividad
2. **Acción:** Redireccionar tráfico a Nodo Norte (Medellín)
3. **Recuperación:** Restaurar desde backup diario
4. **Tiempo estimado:** 30 minutos

7.2. Escenario 2: Pérdida de Conexión entre Nodos

1. **Detección:** Timeout en consultas FDW
2. **Acción:** Modo degradado (solo datos locales)
3. **Recuperación:** Reintentos automáticos cada 5 minutos
4. **Sincronización:** Catch-up al restaurar conexión

7.3. Escenario 3: Inconsistencia de Datos

1. **Detección:** Checksums periódicos
2. **Acción:** Ejecutar script de verificación
3. **Corrección:** Sincronización manual desde última copia buena
4. **Prevención:** Transacciones atómicas en escrituras

8. Procedimientos de Mantenimiento

8.1. Mantenimiento Diario

- Verificación de conexiones FDW
- Revisión de logs de errores
- Backup incremental de datos locales

8.2. Mantenimiento Semanal

- Sincronización manual de catálogos
- Optimización de vistas distribuidas
- Rotación de logs de auditoría

8.3. Mantenimiento Mensual

- Actualización de estadísticas
- Revisión de permisos y roles
- Pruebas de recuperación de desastres

9. Monitoreo y Métricas

9.1. Métricas Clave

Métrica	Umbral	Acción
Latencia FDW	>500ms	Investigar red
Consultas fallidas	>5/min	Revisar conexión
Uso de CPU Nodo Central	>80 %	Escalar recursos
Espacio disco Nodo Sur	<10 % libre	Limpiar logs

9.2. Herramientas de Monitoreo

- PostgreSQL built-in: `pg_stat_activity`
- Custom: Scripts de verificación de nodos
- External: Nagios/Zabbix para monitoreo de red

10. Conclusión

El esquema de distribución implementado en HIS+ permite:

1. **Alta disponibilidad:** Fallos en un nodo no afectan a los demás
2. **Escalabilidad:** Pueden añadirse nuevas sedes fácilmente
3. **Desempeño:** Consultas locales rápidas, distribuidas cuando sea necesario
4. **Consistencia:** Datos críticos accesibles en toda la red
5. **Seguridad:** Control granular de accesos por rol y sede

La arquitectura cumple con todos los requisitos del proyecto manteniendo la simplicidad operativa y garantizando la interoperabilidad entre los diferentes centros médicos de la red hospitalaria.