

# Aplicación de la federación de bases de datos para la gestión de revistas en una biblioteca universitaria

José Manuel Barrueco

Jose.Barrueco@uv.es

Universitat de Valencia. Biblioteca de Ciencias Sociales  
Campus del Tarongers, s/n 46071 Valencia

Pedro Blesa

pblesa@dsic.upv.es

Universitat Politècnica de València. DSIC  
Camino de Vera, s/n 46022 Valencia

**Resumen** En las bibliotecas conviven múltiples sistemas de gestión de bases de datos para la administración de la colección. En muchos casos la misma información está duplicada en varias bases de datos. En este trabajo aportamos una solución a este problema utilizando técnicas de federación de bases de datos. Se hace un estudio de un caso concreto de una biblioteca universitaria y se ofrece una propuesta de arquitectura federada utilizando Oracle.

Keywords: bases de datos federadas, SGBD, bibliotecas universitarias, publicaciones seriadas

## 1. Introducción

En un escenario ideal el sistema integrado de automatización de una biblioteca (en adelante catálogo) debería permitir la gestión de todos los trabajos técnicos que se realizan en la misma. Lamentablemente es tal la diversidad de bibliotecas, de intereses, de objetivos y de funcionamiento que es un objetivo no cumplido por la mayoría de sistemas. En algunos casos, aunque el sistema permita gestionar una determinada tarea, los condicionantes locales o la forma de trabajar de la biblioteca hacen que sea imposible aplicarlo.

Por todo ello en las bibliotecas convive el sistema de automatización con un número más o menos grande de bases de datos destinadas a gestionar aspectos concretos de la colección. Así por ejemplo, frente al catálogo podemos encontrarnos con una base de datos para gestionar las adquisiciones, otra para llevar el registro de monografías, otra para gestionar la adquisición y registro de publicaciones periódicas, otra para gestionar duplicados, otra para gestionar el préstamo

interbibliotecario, etc etc. Bases de datos que normalmente suelen estar dispersas en varios sistemas de gestión como MS Access, DBase o FileMaker. Es frecuente encontrarnos también con softwares especialmente diseñando para cumplir una tarea determinada (SOD para obtención de documentos, GLAS para gestión de revistas, etc).

Los problemas planteados por esta dispersión de datos son evidentes. Como más importantes podríamos señalar dos: principalmente la consistencia de la información y la redundancia de los datos.

Este problema de dispersión y la consiguiente necesidad de acceso, inserción y modificación de los datos en diferentes sistemas de gestión, no es exclusivo de las bibliotecas. Como afirma [6], en el ámbito empresarial se plantea también con frecuencia. Los vendedores de Sistemas de Gestión de Bases de Datos (SGBD), como Oracle o IBM, han trabajado durante mucho tiempo en la creación de aplicaciones que permitan acceder a datos dispersos no solo en diferentes sistemas gestores sino incluso disponibles en ficheros estructurados como por ejemplo XML. Incluso la ISO consciente de este problema está trabajando en una extensión al lenguaje SQL denominado *Management of External Data* que permita este tipo de acceso [1]. La especificación actual solamente permite accesos de lectura a los datos. Una versión completa con lectura y modificación de datos está prevista que aparezca este año.

Desde el punto de vista de las bases de datos la arquitectura que permite el acceso a bases de datos heterogéneas (como el escenario descrito anteriormente) son las bases de datos federadas. Una base de datos federada permite integrar en una sola base global un conjunto de bases preexistentes gestionadas

por SGBD diferentes, ya sean relacionales o en red, e incluso datos estructurados almacenados en ficheros texto, XML, etc [5].

El objetivo del trabajo que presentamos es aplicar los principios de las bases de datos federadas (BDF) para solucionar un problema concreto de integración de bases de datos heterogéneas en una biblioteca universitaria. Autores como [4] o [8] han trabajado en problemas similares. Los primeros proponen un sistema federado para acceder a una variedad de recursos bibliográficos en Internet utilizando un sistema denominado FraQL. Los segundos aplican la federación al enlace de referencias dentro de una biblioteca digital especializada en contenidos científicos y técnicos.

El objeto de nuestro trabajo es mucho más modesto en el sentido que trabajaremos en un problema particular de las bibliotecas como es el control de los duplicados de publicaciones periódicas. En el presente trabajo ofrecemos una solución teórica al mismo al formular una arquitectura federada cuya implementación está en curso.

Aunque con la generalización de las publicaciones electrónicas esta tarea desaparecerá en un futuro, por el momento sigue siendo una parte necesaria de la gestión de la colección impresa. En nuestra biblioteca nos encontraremos necesariamente con fascículos duplicados de una publicación. Estos se producen por varias causas como errores en el envío por parte del distribuidor o lo que es más frecuente, en el caso de publicaciones que se reciben por donación, generalmente publicaciones oficiales, por el envío de varios ejemplares por parte de la editorial. Es necesario mantener un control de estos fascículos porque podemos necesitarlos en el futuro. Por ejemplo para sustituir fascículos deteriorados o extraviados, o también para realizar intercambio con otras bibliotecas. El intercambio es una de las formas de adquisición de publicaciones como así lo reconoce [10].

A pesar de esta importancia, los programas comerciales diseñados para la gestión de publicaciones periódicas no ofrecen una buena solución a esta operación por lo que normalmente las bibliotecas se ven obligadas a recurrir a bases de datos adicionales diseñadas expresamente para esta tarea. El objetivo de este trabajo es presentar una solución particular a este problema, utilizando la federación de bases de datos, para simplificar la tarea de gestión de duplicados.

El trabajo se estructura de la siguiente forma. En la sección 2 hacemos una breve introducción a las BDF. En la sección 3 se describe el escenario sobre el que vamos a trabajar. Seguidamente hacemos una descripción de la arquitectura del sistema federado siguiendo el marco teórico que hemos esbozado en la sección 2. El punto 5 se dedica a la implementación del sistema.

## 2. Bases de datos federadas

El término "base de datos federada" se usa para designar una técnica que permite el acceso integrado a un conjunto de bases de datos distribuidas, heterogéneas y autónomas.

La autonomía de sistemas basados en múltiples nodos, llamados componentes (sistemas fuentes) es crítica. Se pueden consignar varios tipos de autonomía: de diseño (cada componente es independiente de los demás y se puede modificar), de comunicación (en cualquier momento un componente puede entrar o dejar la federación), de ejecución (casi imposible si hay gestión global de las transacciones).

La heterogeneidad por su parte puede ser de distintos tipos [9]: sintáctica, del modelo de datos o lógica.

Podemos diferenciar dos tipos de federación: estricta y relajada.

Estricta es cuando se dispone de un esquema global unificado, que usan los usuarios para el acceso. Puede cubrir total o parcialmente los componentes. Para asegurar la equivalencia semántica el sistema federado debe conocer las correspondencias entre las consultas y entre el esquema federado y los de los componentes a través de reglas. Estas reglas debe definir las un experto en el dominio. Las federaciones estrictas son cómodas para el usuario ya que sólo necesita conocer el esquema federado.

Es relajada cuando no existe tal esquema global pero se dispone de un lenguaje unificado de consulta (MDBQL). Estos sistemas suelen usar vistas integradas de forma que los usuarios pueden definir vistas sobre los componentes. En una integración estricta los esquemas fuentes no son visibles, mientras que en una ligera si lo son. No obstante pueden definir vistas adicionales para acceder de una forma predefinida.

Por otro lado podemos diferenciar también entre federaciones de estrategia ascendente

(bottom-up) o descendente (top-down). La segunda funciona de acuerdo a unas necesidades globales. Por ejemplo, una empresa de venta de libros quiere integrar información de bases de datos diferentes. Esta solución tiene ventajas en escenarios donde las fuentes evolucionan muy rápidamente. En la estrategia ascendente se busca un acceso integrado a un número dado de fuentes de datos. Esta solución da lugar a sistemas bien integrados semánticamente porque se supone que se conocen bien los componentes. Las actualizaciones son sólo posibles si la conexión es muy estricta.

Los sistemas de bases de datos federadas ofrecen la funcionalidad típica de un SGBD como accesos de lectura y escritura. Presentan una estructura clásica en cinco capas que se pueden reducir a cuatro cuando todas las fuentes de datos utilizan el mismo modelo, por ejemplo el relacional. Gráficamente lo podemos ver en la figura 1.

### 3. Descripción del escenario de trabajo

Dado que no es posible establecer un modelo válido para cualquier biblioteca puesto que son muy diversas las técnicas y las soluciones aportadas localmente, es necesario adaptarse a las particularidades de cada caso.

En nuestro trabajo vamos a realizar un estudio del caso de una biblioteca universitaria donde para la gestión de la colección de publicaciones seriadas se utilizan principalmente dos sistemas informáticos: el catálogo general de la biblioteca, donde se incluyen todas las publicaciones tanto activas como canceladas. Sirve fundamentalmente de información a los usuarios a la hora de consultar la disponibilidad de una publicación, periodo cronológico del que se disponen fondos o situación física de los fondos a través de la signatura. Por otro lado tenemos un software específico para la gestión administrativa de la colección, denominado GLAS. A través de este programa se lleva a cabo el registro de entrada, el seguimiento de los fascículos, las reclamaciones, la gestión de proveedores, etc. Finalmente existen otras aplicaciones para tratar aspectos concretos de la colección como el que nos afecta en este trabajo: el control de duplicados. Este se lleva a cabo mediante una base de datos en MS Access diseñada en la biblioteca y denominada Duplicados.

Claramente los tres sistemas comparte un

núcleo básico de información que está duplicado en todos ellos: los datos bibliográficos de cada publicación. El problema más grave que causa es que cuando una nueva publicación es añadida a la colección es necesario incluir los mismos datos en todos ellos. Problema que se solucionaría federando los tres sistemas de tal forma que las operaciones de adición y borrado de datos bibliográficos se pudiera realizar una sola vez para todos los sistemas. No obstante y como señalamos más adelante existen limitaciones impuestas por el administrador de un componente que hacen que estas operaciones solamente se puedan realizar en dos de las tres fuentes de datos (GLAS y Duplicados).

La arquitectura de bases de datos federadas que se propone se describe en la sección siguiente. A continuación describimos someramente los esquemas locales que entran en juego en el proceso. Estos son bastante extensos en los casos de GLAS y Catálogo por lo que aquí solamente mencionamos las tablas que utilizaremos para la federación.

El sistema GLAS para la gestión de publicaciones seriadas funciona sobre un SGBD DBase. De él utilizaremos dos tablas. Una denominada SERLIST que almacena los datos bibliográficos básicos de cada publicación y otra SERCHECK que almacena los datos de cada fascículo recibido en la biblioteca.

La base de datos MS Access tiene una estructura sumamente sencilla ya que está compuesta por dos únicas tablas: FONDOS y REVISTA. La primera almacena los fascículos duplicados y la segunda una información bibliográfica básica para identificar la publicación a la que pertenecen y realizar listados para el intercambio con otras bibliotecas.

El catálogo de la biblioteca es un sistema diseñado por el servicio de informática de la universidad que corre sobre Oracle. El módulo de hemeroteca es sumamente complejo con 34 tablas diferentes. No obstante el esquema exportado solamente utilizará una de esas tablas: REV\_REVISTAS que contiene la información bibliográfica de cada publicación.

Por lo que respecta a los permisos de lectura y escritura, podremos leer y escribir en las dos primeras mientras que en el caso del catálogo solamente podremos leer. Esta es una limitación impuesta por cuestiones de seguridad que no reviste importancia puesto que el primer lugar donde deben incluirse las publicaciones es el catálogo y el resto de bases

deben nutrirse de éste.

#### 4. Arquitectura propuesta

La arquitectura típica de un sistema federado consta de las cuatro capas que pueden apreciarse en la figura 1.

Partimos de los datos fuente que en este caso están compuestos por las tres bases de datos descritas anteriormente. Todas ellas siguen el modelo de datos relacional y tienen sus correspondientes esquemas locales. Sólo una parte de esos esquemas locales es la que va a utilizarse en la federación. Esa parte es la que denominamos esquema exportado. A partir de los esquemas exportados de cada componente se elabora un sólo esquema federado que incluirá las equivalencias entre los atributos de todos los participantes. Finalmente, el acceso al sistema federado se hará a través de uno o varios esquemas externos.

Según las categorías de sistemas federados vistas en el punto 2, lo que pretendemos hacer es una base de datos federada estricta y ascendente. Estricta porque dispondrá de un esquema global federado y todos los componentes siguen el modelo relacional, y ascendente porque partimos de un número definido y limitado de fuentes de datos.

El proceso de diseño de una base de datos federada consta de una serie de pasos definidos por [9].

1. Traducción de esquemas. Se realiza cuando un esquema componente no está en el mismo modelo de datos que usamos como general. En nuestro caso, todos los modelos de datos son relacionales por lo que no es necesaria tal traducción.
2. Definición de esquemas exportados. Cada componente negocia qué parte de su base de datos autoriza que se incluya en la federación. El administrador de la base federada debe llegar a un acuerdo sobre los contenidos y las operaciones permitidas sobre los esquemas exportados. En concreto cada componente debe especificar si concede derechos de escritura o de sólo lectura. En nuestro caso habrá derechos de escritura y lectura sobre GLAS y Duplicados pero sólo de lectura sobre el Catálogo. En las figuras 2, 3 y 4 relacionamos los esquemas exportados de todos los componentes.
3. Integración de esquemas. La integración de los diferentes esquemas exportados

dará lugar a un esquema federado. Dicha integración se divide en cinco pasos: preintegración, comparación, conformación, mezcla y reestructuración. En nuestro caso no hay preintegración ya que ello supone la traducción de un modelo de datos particular al general.

El paso de comparación, también llamado análisis de esquemas, incluye dos actividades, en primer lugar analizar y comparar los objetos de los esquemas, identificando los problemas de nombres (como homónimos y sinónimos), conflictos en los dominios (tipos de valores) o restricciones distintas. En segundo lugar especificar las interrelaciones entre los objetos de los esquemas. El paso de conformación está muy ligado al de comparación, ya que resulta difícil comparar a no ser que la información se represente de forma similar en los diferentes esquemas.

El proceso de integración no puede ser automático ya que, primero la semántica del modelo relacional no puede capturar totalmente el estado del mundo real, segundo sería necesaria mucha más información que la que da el modelo, y tercero puede haber diferentes interpretaciones del mundo real, que además pueden cambiar con el tiempo.

De acuerdo con [3] el esquema integrado resultante debe cumplir una serie de requisitos como son: ser completo, correcto, minimalista y comprensible.

[7] ofrece el siguiente procedimiento, utilizando Modelo Genérico de Integración, para la integración de esquemas 5. Partiendo de los esquemas exportados, se analizan los elementos que pertenecen a todos los componentes y se crea una tupla con los mismos. El resto de elementos que no tiene equivalentes en algún componente se incluyen en tuplas separadas relacionadas con la anterior.

El esquema integrado resultante en nuestro caso sería el que aparece en la figura 6. Podemos apreciar como los componentes son bastante homogéneos ya que todos los objetos representados en cada esquema exportado tienen su equivalente, en muchos casos con atributos nombrados de la misma forma, en al menos dos esquemas. En algunos casos

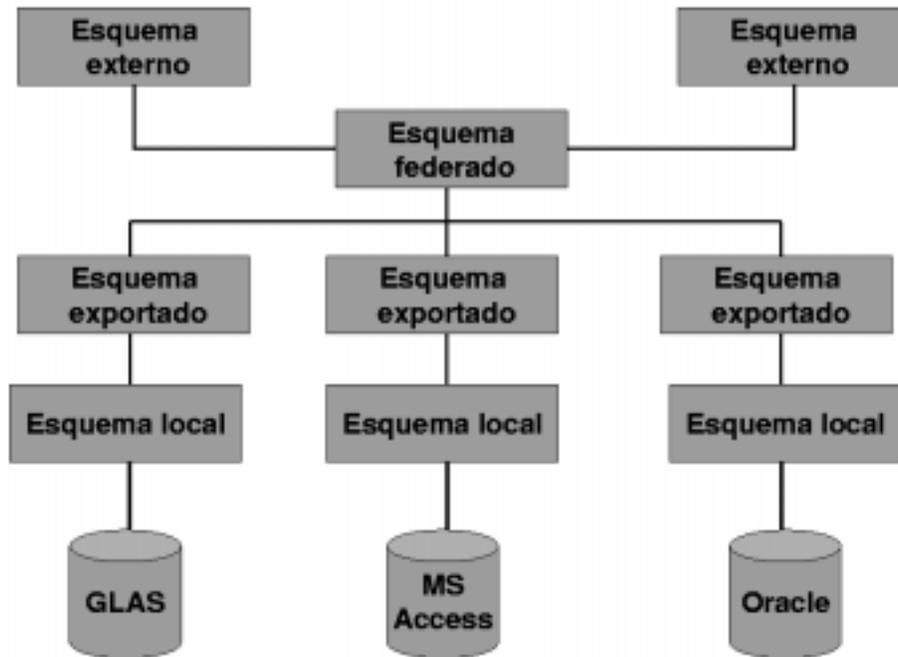


Figura 1: Arquitectura Bases de Datos Federadas

Tabla	Campo	Tipo	Longitud
<b>SERLIST</b>			
	TI	Memo	10
	FR	Carácter	2
	NI	Carácter	15
	TP	Carácter	10
<b>SERCHECK</b>			
	IS	Carácter	31
	ST	Carácter	2
	PR	Carácter	2
	PO	Numérico	4

Figura 2: Esquema exportado GLAS

Tabla	Campo	Tipo	Longitud
<b>REVISTA</b>			
	Título	Texto	255
	ISSN	Texto	10
	Signatura	Texto	55
<b>FONDOS</b>			
	Numero	Texto	10
	Volumen	Texto	10
	Año	Numérico	Entero
	Estación	Texto	50

Figura 3: Esquema exportado Duplicados

Tabla	Campo	Tipo	Longitud
<b>REV_REVISTAS</b>			
título	Carácter		150
estado	Carácter		1
issn	Carácter		8

Figura 4: Esquema exportado Catálogo

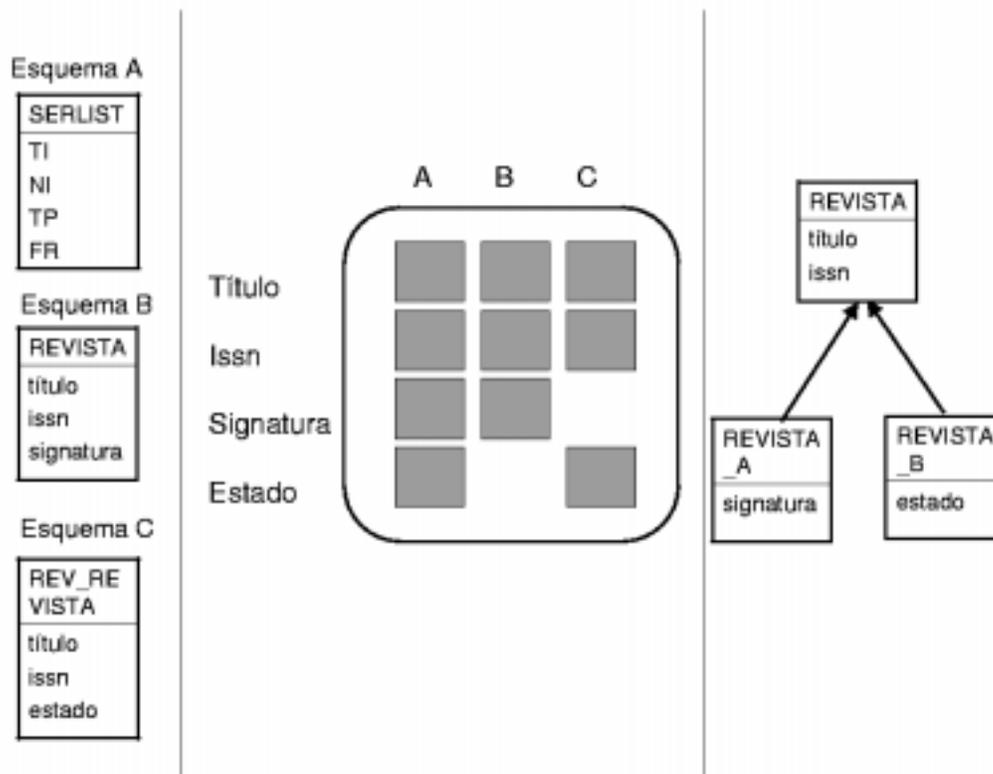


Figura 5: Modelo Genérico de Integración

se han producido conflictos en el tipo de atributo y en su longitud. Se han solucionado utilizando el tipo carácter allí donde éste entraba en conflicto con numéricos o el más grande donde el conflicto se refería a la longitud del atributo.

4. El último punto sería la definición de esquemas externos allí donde sea necesario.

## 5. Implementación

Para la implementación del modelo descrito estamos utilizando Oracle9i. Dado que uno de los componentes utiliza este mismo SGBD el problema se plantea en la integración del resto de componentes basados en DBase y MS Access. Oracle ofrece una serie de herramientas para facilitar el acceso a este tipo

de sistemas que se engloban bajo el nombre de Heterogeneous Services (HS) [2]. HS proporciona la tecnología para que el SGBD pueda conectar a sistemas ajenos a Oracle. Proporciona dos tipos de traducciones: es capaz de traducir las instrucciones SQL de Oracle al dialecto del sistema ajeno así como traducciones referidas a los diccionarios, los cuales muestran los metadatos de los sistemas ajenos en el formato local.

Dependiendo del tipo de sistema que contenga los componentes HS proporciona dos soluciones diferentes: Transparent Gateways con una mayor funcionalidad ya que accede los sistemas ajenos utilizando su propio interfaz. Está destinado a acceder fuentes del tipo de Sybase, DB2, Informix, Ingress, etc. Y en segundo lugar Generic Connectivity que constituye una solución más gen-

ESQUEMA FEDERADO				EQUIVALENCIA		
				GLAS	Duplicados	Catálogo
<b>REVISTAS</b>				<b>SERLIST</b>	<b>REVISTA</b>	<b>REV_REVISTA</b>
	título	Carácter	255	TI	título	título
	issn	Carácter	15	NI	issn	issn
<b>REVISTAS_A</b>				<b>SERLIST</b>	<b>REVISTA</b>	<b>REV_REVISTA</b>
	signatura	Carácter	55	TP	signatura	
<b>REVISTAS_B</b>				<b>SERLIST</b>	<b>REVISTA</b>	<b>REV_REVISTA</b>
	estado	Carácter	2	FR		estado
<b>FONDOS</b>				<b>SERCHECK</b>	<b>FONDOS</b>	
	Año	Numérico	4	PO	Año	
	Número	Carácter	10	PR	Número	
	Volumen	Carácter	10	ST	Volumen	
	Estación	Carácter	50	IS	Estación	

Figura 6: Esquema integrado

eral que no incluye todas las funcionalidades del anterior y que permite el acceso a fuentes utilizando estándares industriales como ODBC o OLEDB. Permite acceder a cualquier fuente de datos que sea compatible con estos estándares.

Es esta última solución la que hemos escogido para implementar nuestro sistema. Gráficamente el resultado se puede ver en la figura 7.

El proceso de implementación consta de las siguientes fases:

- Configuración de las fuentes de datos. Hemos de crear orígenes de datos ODBC para cada uno de los componentes. En nuestro caso, tanto GLAS como Duplicados se encuentran en un servidor Windows NT.
- Configuración del HS de Oracle para acceder a los componentes. Esta configuración es sencilla ya que únicamente tenemos que especificar el driver utilizado, el servidor y el puerto.
- Crear un enlace entre la base de datos Oracle y la fuente de datos. Esto lo haremos por cada componente. Por ejem. utilizando el comando *CREATE DATABASE REVISTAS LINK Revistas USING 'MSAccess\_Serlist'*;
- Definición mediante las instrucciones SQL correspondientes del esquema federado descrito en el punto anterior.

## 6. Conclusiones

En el presente trabajo hemos presentado una posible solución al problema de la duplicación de información en bibliotecas universitarias. Este es un problema que se plantea no solamente en bibliotecas sino en cualquier organización. Los principios de las bases de datos federadas se establecieron a finales de los años ochenta para solucionarlo. No obstante desde entonces no se ha avanzado mucho en el mismo y no existe un estándar que permita esa integración de forma sencilla y eficiente.

Hemos descrito un problema concreto que se podría solucionar con la federación de las bases de datos componentes. Hemos definido una arquitectura federada para el mismo y hemos apuntado cómo se podría implementar utilizando un SGBD concreto, Oracle.

El trabajo está en curso y en estos momentos estamos desarrollando la fase de implementación.

## Referencias

- [1] Iso/iec 9075-9:2000 information technology - database languages - sql - part 9: Management of external data (sql/med). Technical report, International Organization for Standardization, 2000.
- [2] Oracle9i heterogeneous connectivity administrator's guide. Technical report, Oracle, 2002.

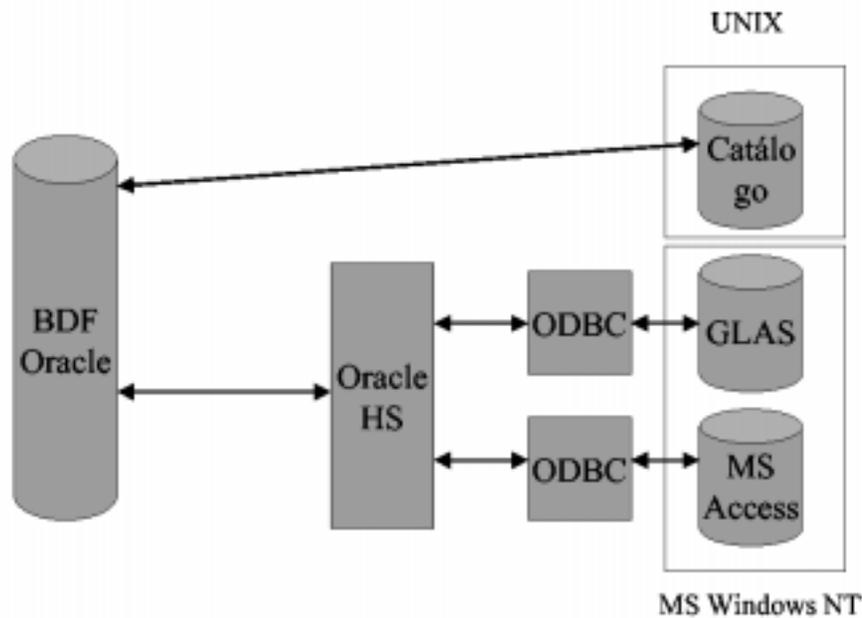


Figura 7: Implementación

- [3] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323–364, 1986.
- [4] Martin Endig, Michael Hoding, Gunter Saake, Kai-Uwe Sattler, and Eike Schallehn. Federation services for heterogeneous digital libraries accessing cooperative and non-cooperative sources. Technical report, 2000.
- [5] S. Busse et al. Federated information systems: Concepts, terminology and architectures. Technical report, Technische Universität Berlin, 1999.
- [6] Jim Melton and Jan-Eike Michels. Sql and management of external data. Technical report, 2001.
- [7] et al. Michael Hoding. Sigma fdb: Overview of the magdeburg-approach to database federations. Technical report, 1999.
- [8] Eike Schallehn, Martin Endig, and Kai-Uwe Sattler. Citation linking in federated digital libraries. Technical report, 1999.
- [9] Amit Sheth and James Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 1990.
- [10] Judith Szilvassy. *Manual básico de gestión de publicaciones seriadas*. AN-ABAD, 1998.