

DESARROLLO DE APLICACIONES CON BASES DE DATOS

Licenciatura en Informática
Trabajo Práctico 3

Prof. Titular Disciplinar: Silvia Laura Castelli
Prof. Titular Experto: Ricardo Ramón Daubrowsky
Alumno: Pablo Alejandro Hamann
Legajo: VINF010782
Año: 2025

Tabla de contenido

Introducción.....	1
Repositorio en GitHub	1
Sección 1: Diagrama de arquitectura de implementación	2
Sección 2: Tabla con detalle de función de cada componente	3
Sección 3: Software sugerido para implementación de servidores de datos	4

Introducción

Este documento corresponde al desarrollo de las consignas planteadas en el Trabajo Práctico 3, y retoma a partir de las tareas realizadas en los TPs 1 y 2. Al igual que en el práctico anterior, complementan este documento, el archivo SQL correspondiente, que contiene todas las sentencias necesarias para crear la base de datos (esta vez en Oracle), su estructura, poblarla con algunos datos modelo, y realizar las consultas solicitadas.

Repositorio en GitHub

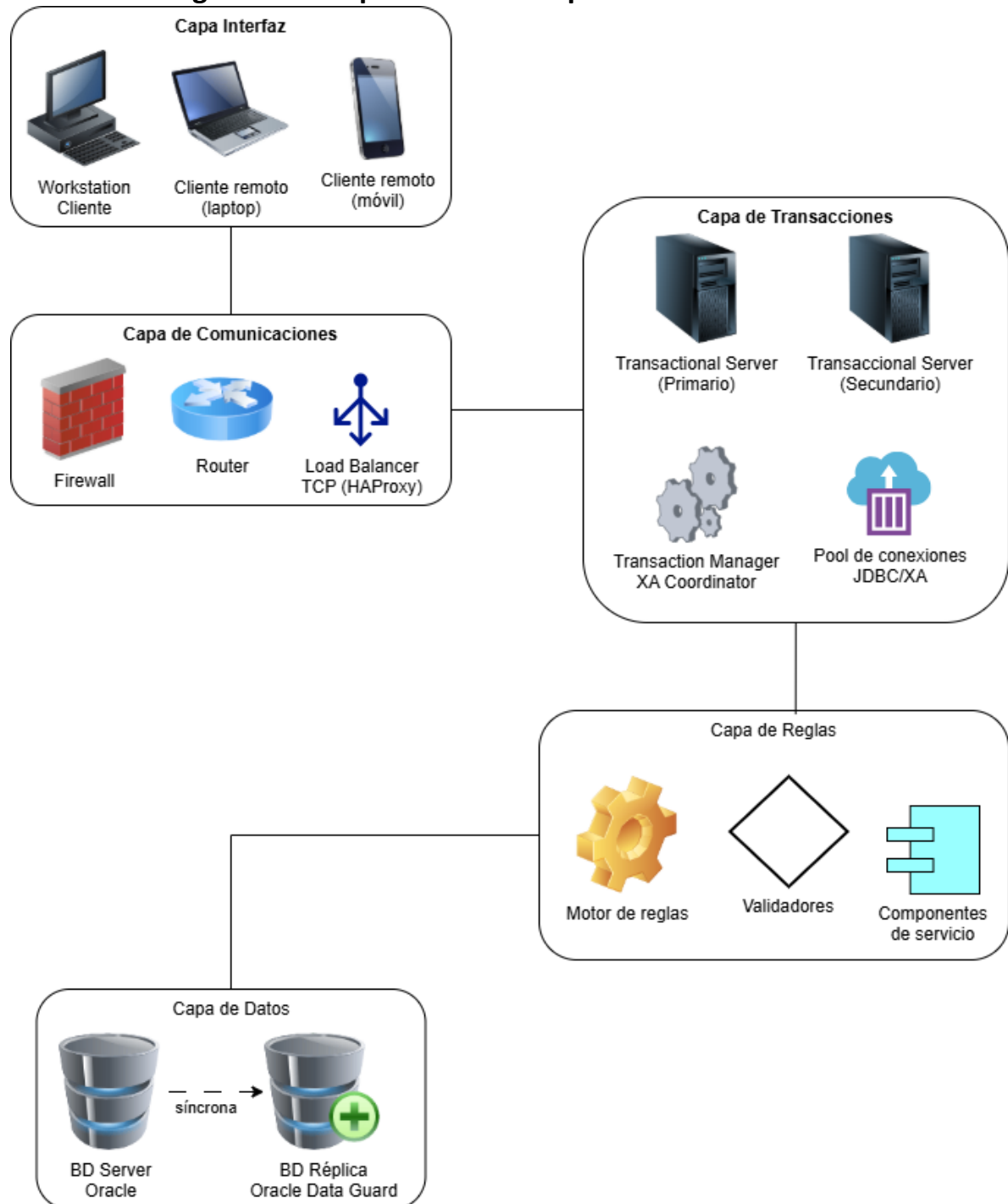
Todo lo producido, tanto para este presente TP, como para el anterior, se encuentra en un repositorio en GitHub creado para el cursado de esta materia. Allí, se mantienen actualizadas tanto las actividades prácticas como los TPs y cualquier otro tipo de actividad que implique desarrollo (de documentación, programación, etc.), que se dé durante el cursado de la materia. El repositorio se puede acceder mediante el siguiente enlace:

<https://github.com/linkstat/dabd/tree/main>

Archivos principales del proyecto **dabd** (este proyecto):

- Este documento en formato PDF:
 - <https://github.com/linkstat/dabd/raw/refs/heads/main/docs/HAMANN-PABLO-ALEJANDRO-TP3.pdf>
- Este documento en formato DOCX de Word:
 - <https://github.com/linkstat/dabd/raw/refs/heads/main/docs/HAMANN-PABLO-ALEJANDRO-TP3.docx>
- Archivo de diagrama en formato Drawio:
 - <https://raw.githubusercontent.com/linkstat/dabd/refs/heads/main/docs/TP3-Diagrama.drawio>

También es posible ver el historial de *commits* realizado a los archivos (y a toda la estructura del proyecto), ya que se trata de un repositorio público y se actualizando de forma regular, sobre todo cuando se aplican muchos cambios.

Sección 1: Diagrama de arquitectura de implementación

Sección 2: Tabla con detalle de función de cada componente

Componente	Descripción función
Workstation Cliente Cliente remoto (laptop / móvil)	Interfaz de usuario (PC, laptop o móvil) desde la que se envían las solicitudes al sistema de registro de pedidos.
Firewall	Filtra y controla el acceso de red, aplicando políticas de seguridad entre la capa de interfaz y la capa de comunicaciones.
Router	Dirige el tráfico IP entre la capa de interfaz y la capa de comunicaciones, asegurando conectividad de red.
Load Balancer TCP (<i>HAProxy</i> en modo TCP u otros)	Reparte las conexiones TCP entrantes de manera uniforme entre los Servidores Transaccionales, garantizando alta disponibilidad y balanceo de carga. <i>HAProxy</i> es uno de tantos productos, solo que este tiene la particularidad de poder trabajar tanto en capa 4 (modo TCP, donde puede funcionar como un balanceador de carga) como en capa 7 (a nivel de aplicación: HTTP SSL, en donde sería un proxy reverso más tradicional, como <i>nginx</i> en modo proxy reverso o similares).
Transactional Server / Servidor Transaccional (Primario)	Ejecuta la lógica de aplicación y coordina el inicio/commit/rollback de transacciones; punto principal de procesamiento de pedidos.
Transactional Server / Servidor Transaccional (Secundario)	Réplica en «caliente» (o en vivo) del servidor primario para <i>fail-over</i> ; se mantiene sincronizado y puede tomar el relevo en caso de caída o baja del primario.
Transaction Manager / XA Coordinator	Orquesta transacciones distribuidas en dos fases (2PC) entre la base de datos y otros recursos (pools XA, colas), garantizando ACID (A tomicidad, C onsistencia, I solación, D urabilidad).
Pool de conexiones JDBC/XA	Gestiona y reutiliza conexiones a Oracle con soporte XA, optimizando el rendimiento y habilitando transacciones distribuidas.
Motor de reglas	Subsistema que evalúa las reglas de negocio configurables (validaciones, descuentos, etc.) antes de confirmar el pedido.
Validadores	Componentes dedicados a chequeos específicos (como control de stock disponible, formatos de datos, etc.), que se invocan desde los servicios de dominio.
Componentes de servicio	Servicios de dominio (e.g. <i>PedidoService</i> , <i>ClienteService</i>) que encapsulan casos de uso (crear pedido, anular pedido, actualizar stock, etc.).
BD Server Oracle (Primary)	Base de datos principal (Oracle XE 21c sobre Oracle Linux), almacena tablas de pedidos, productos, logs y soporta todas las operaciones de escritura.
BD Réplica Oracle Data Guard (Standby)	Instancia en modo <i>read-only</i> que replica de forma síncrona la BD primaria. Principalmente se usará como <i>fail-over</i> ante caída o baja de la BD principal; aunque también se puede aprovechar para realizar tareas de análisis y reportes, sin sobrecargar la BD principal.

Sección 3: Software sugerido para implementación de servidores de datos

Como ya venía anticipando al identificar los componentes en el diagrama de la sección 1 (y posteriormente en la tabla de descripción), sugiero los siguientes productos de software para todos los componentes antes mencionados:

Componente	Producto / Tecnología	Justificación
Base de datos primaria	Oracle Database 21c Express Edition v.21.3.0 sobre OS: Oracle Linux 8.6	Motor OLTP, admite tablas, PL/SQL, Data Guard. Liviano y gratuito para entornos de desarrollo y pruebas.
Base de datos réplica	Oracle Data Guard (Broker) v.21.3.0 sobre OS: Oracle Linux 8.6	Réplica síncrona o asíncrona de la primaria, para <i>fail-over</i> automático. Es configurable con Data Guard Broker.
Servidor transaccional	Red Hat JBoss EAP v.7.4 sobre OS: Oracle Linux 8.6 (o bien RHEL 8)	Es el contenedor Java EE/Jakarta EE. Aloja los servicios de dominio, el gestor de transacciones y el pool JDBC/XA (IronJacamar).
Transaction Manager / XA Coordinator	Narayana (incorporado en JBoss EAP) v.6.10.x, corre sobre JVM: OpenJDK 11+	Orquesta Two-Phase-Commit entre la BD y otros recursos XA.
Pool de conexiones JDBC/XA	IronJacamar (JBoss), ver. IronJacamar 1.4.x, corre sobre JVM: OpenJDK 11+	Reutiliza conexiones, gestiona transacciones XA y optimiza el rendimiento de acceso a la BD.
Motor de reglas de negocio	Red Hat Decision Manager v.7.72.0.Final, corre sobre JVM OpenJDK 11+	Es una plataforma completa para modelar, versionar y ejecutar reglas de negocio separadas del código.
Validadores (JSR-380 Bean Validation)	Hibernate Validator v.6.2.0.Final, correo sobre JVM OpenJDK 11+	Implementa validaciones declarativas (anotaciones) para campos de entidades: stock, formatos, rangos, etc.
Componentes de servicio (Domain Services)	Spring Boot v.2.6.6, correo sobre JVM OpenJDK 11+	Framework para microservicios o aplicaciones monolíticas: inyecta dependencias, controladores REST, servicios transaccionales.
Servidor de comunicaciones / proxy	HAProxy v.2.4.x, corriendo sobre FreeBSD 14.2	Balanceo TCP/L4 en modo passthrough TLS. Firewall de aplicaciones, health-checks, alta disponibilidad. FreeBSD es el sistema operativo que proporciona una plataforma de alta confiabilidad para correr cualquier servidor servicio de la capa de comunicaciones
Java Runtime	Oracle OpenJDK v.11.0.x LTS	Oracle Java 11 LTS es la JVM sobre la cual corren los servidores Java EE, y es ampliamente soportada por todos los componentes (JBoss EAP, Spring Boot, Drools e Hibernate Validator).