

Universidad de las Ciencias Informáticas
Facultad 4



Título: Sistema para la Administración del Proceso de
Réplica del Proyecto SIGEP

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Yanet Del Risco Batista
Yandry Alberto Terry

Tutor(es): Lic. Eddy Manuel Infante Alonso

Consultante: Ing. Aláin Fernández del Toro

19 de junio de 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 19 días del mes de junio del año 2007

Yandry Alberto Terry
Yanet Del Risco Batista

Lic. Eddy Manuel Infante Alonso

RESUMEN

El presente trabajo consiste en el análisis y el diseño de un sistema encargado de administrar y monitorear el proceso de réplica del proyecto Sistema de Gestión Penitenciaria (SIGEP) para el Sistema Penitenciario de la República Venezolana.

El proyecto SIGEP utiliza el Oracle 10g como sistema gestor de base de datos. Oracle brinda junto con su servidor, la herramienta necesaria para replicar entre servidores de su tipo, pero para poder replicar de manera by-direccional se necesita tener la versión Enterprise Edition instalada en cada servidor de base de datos que se vaya a utilizar en el proceso de replicación. La licencia para la versión Enterprise de Oracle cuesta una cifra milenaria que multiplicada por la cantidad de servidores que necesita el SIGEP puede llegar a ser millonario el gasto por concepto de licencias. Como alternativa a este problema la UCI ha propuesto utilizar la versión Standard Edition, la cual no permite realizar el proceso de replicación de forma by-direccional, pero es mucho más económico y ha desarrollado un nuevo algoritmo de replicación.

El trabajo surge a partir de la necesidad de crear una herramienta que realice la replicación de datos utilizando el algoritmo desarrollado por la UCI en aras de garantizar que el proceso de réplica sea administrado y monitoreado eficientemente por los administradores de las bases de datos de los establecimientos penitenciarios.

El objetivo de este trabajo es realizar el análisis y diseño de un software que permita realizar la administración y monitoreo de los procesos de replicación de los datos, monitorear el flujo de transacciones por cada establecimiento penitenciario, vigilar los errores que puedan aparecer tanto en las prisiones como en el centro de datos, generar reportes que permita a los administradores de la base de datos poder brindar información sobre el estado de la replicación de datos por cada establecimiento penitenciario o de manera general e implementar un mecanismo de alerta que pueda informar en caso de que ocurran fallos que impidan realizar este proceso.

ÍNDICE

| | |
|---|----|
| RESUMEN | II |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA | 3 |
| 1.1.1 Replicación | 3 |
| 1.1.2 Las herramientas para la administración de procesos de réplica | 3 |
| 1.1.3 La herramienta de administración de réplica del Oracle10g..... | 4 |
| 1.1.4 Otras herramientas propietarias..... | 4 |
| 1.1.4 Las herramientas de administración de réplica para gestores de bases de datos libres..... | 5 |
| 1.2 HERRAMIENTAS PROPUESTAS PARA EL DESARROLLO DEL SISTEMA | 7 |
| 1.2.1 La tecnología Java..... | 7 |
| CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA | 8 |
| 2.1 Descripción de los procesos que serán objeto de automatización: | 9 |
| 2.2 Descripción general de la propuesta de sistema, como debe funcionar..... | 11 |
| 2.3 Requerimientos Funcionales | 12 |
| 2.4 Requerimientos no Funcionales | 13 |
| CAPÍTULO 3: ANÁLISIS Y DISEÑO..... | 16 |
| 3.1 Análisis..... | 16 |
| 3.2 DISEÑO. | 21 |
| 3.2 .2 Diagramas de Clases..... | 21 |
| .3.2.3 Descripción de las Clases..... | 28 |
| 3.2.4 Diseño de la Base de Datos..... | 59 |
| 3.2.5 Descripción de las Tablas de la Base de Datos..... | 60 |
| CONCLUSIONES | 67 |
| RECOMENDACIONES..... | 68 |
| BIBLIOGRAFÍA | 69 |

| | |
|--------------------------------|----|
| ANEXO 1: | 70 |
| ANEXO 2 | 90 |
| Diagramas de interacción. | 90 |

INTRODUCCIÓN

El proyecto SIGEP es un sistema distribuido, que utiliza como gestor de bases de datos a Oracle. Para el mantenimiento del mismo es necesario realizar la replicación de datos hacia un centro de datos que guarde toda la información del sistema penitenciario.

Para realizar este proceso utilizando Oracle10g es necesario instalar en cada servidor de base de datos que se vaya a utilizar la versión Enterprise Edition que cuesta una cifra millonaria la licencia de cada servidor, esto supondría para el proyecto un gasto millonario si tenemos en cuenta que el sistema penitenciario venezolano cuenta con aproximadamente 81 centros penitenciarios y cada uno tendría un servidor de base de datos.

Por razones económicas, la UCI ha propuesto utilizar la versión Standard Edition de este software, que es mucho más económica y ha desarrollado un algoritmo que permite realizar la replicación de esta versión hacia la Enterprise Edition. Utilizando esta variante el costo por concepto de licencia sería unos pocos miles lo que implicaría un ahorro para el proyecto de varios millones.

El trabajo surge a partir de la necesidad de crear una herramienta que realice la replicación de datos utilizando el algoritmo desarrollado por la UCI en aras de proporcionarles a los administradores de base de datos de los establecimientos penitenciarios un artefacto que les facilite el trabajo, que sea seguro y fácil de utilizar.

Con la culminación de este trabajo, se contará con el diseño de una herramienta que permitirá a los administradores de las bases de datos realizar la administración y monitoreo de los procesos de replicación de los datos, monitorear el flujo de transacciones por cada establecimiento penitenciario, vigilar los errores que puedan aparecer tanto en las prisiones como en el centro de datos, generar reportes que permita a los administradores de la base de datos poder brindar información sobre el estado de la replicación de datos por cada establecimiento penitenciario o de manera general e implementar un mecanismo de alerta que pueda informar en caso de que ocurran fallos que impidan realizar este proceso.

Para ello es necesario estudiar profundamente los procesos de réplica de datos con Oracle, los sistemas de monitoreo de réplicas con Oracle y el algoritmo de réplica implementado por la UCI, que se está aplicando actualmente en los proyectos del Ministerio del Poder Popular para Relaciones Interiores y de Justicia (MPPRIJ).

El objetivo general de este trabajo es realizar el análisis y el diseño de una herramienta para la administración y monitoreo de los procesos de réplica del proyecto SIGEP.

Como objetivos específicos tenemos:

- Realizar el análisis y diseño una aplicación que permita administrar el proceso de réplica entre servidores de bases de datos Oracle, a través de interfaces de usuario.
- Diseñar un prototipo de interfaz de usuario.

Para esto nos hemos propuesto las siguientes tareas:

- Realizar el estudio de las ventajas y desventajas de las herramientas que utiliza Oracle para el monitoreo y la administración del proceso de réplica.
- Realizar un estudio de los tipos de réplica que se van a desarrollar, así como el algoritmo implementado por la UCI.
- Realizar un estudio de las principales arquitecturas de software.
- Realizar la captura de requisitos del software a diseñar.
- Realizar el análisis y diseño de la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1.1 Replicación

La replicación de datos es el copiado de la información a unas o más bases de datos de manera tal que la información sea constante a través de todos los sitios.

Hay dos tipos básicos de réplica: sincrónica y asincrónica. Con la réplica sincrónica, todas las copias o réplicas de datos se mantienen sincronizadas exactamente y constantes. Si se actualiza cualquier copia los cambios serán aplicados inmediatamente al resto de las bases de datos dentro de la misma transacción. La réplica sincrónica es utilizada cuando se necesita tener exactamente la misma información en todas las bases de datos.

Con la réplica asincrónica o réplica de almacenamiento y envío, las copias de datos se convierten temporalmente fuera de la sincronización, es decir si una copia es actualizada, el cambio es propagado y aplicado a otras copias como segundo paso. Este tipo de réplica el tiempo entre transacciones puede ser desde segundos hasta días, por lo que puede suceder que las copias en algún momento puedan estar fuera de sincronización, pero los datos estarán actualizados en los dos sitios en un periodo de tiempo determinado.[1].

1.1.2 Las herramientas para la administración de procesos de réplica.

En la actualidad existen muchas herramientas que gestionan los procesos de réplica, sincronización o copia de datos en tiempo real, cada uno de los principales gestores de bases de datos a nivel mundial tienen una herramienta para dicha funcionalidad. De ahí que las herramientas de gestión de la replicación de datos se agrupen en dos grandes grupos: las herramientas que gestionan los procesos de réplica para sistemas de gestión de bases de datos propietario y las de gestores libres. Es importante destacar que los dos gestores de bases de datos propietarios más populares: Oracle y SQL Server tienen una herramienta incluida que les permite replicar datos entre sus servidores. El caso específico de Oracle, en el Oracle Enterprise Manager (OEM) trae una sección dedicada a configurar y mantener la replicación de datos entre servidores. Para el caso de los gestores libres podemos decir que por lo general no traen

una herramienta que brinda dicha funcionalidad dentro de su servidor, por lo que existe una gran cantidad de herramientas desarrolladas por terceros dedicadas a solucionar este problema.

1.1.3 La herramienta de administración de réplica del Oracle10g

Oracle 10g tiene para administrar el proceso de réplica el Streams el cual esta integrado en el EMC y posee una interfaz grafica, para la creación y administración de los tipos de réplica que implementa Oracle. Además incluye asistentes que guían los usuarios en todas las operaciones que se realizan en el proceso de replicación. Permite configurar y replicar la base de datos completa, esquemas específicos o tablas específicas entre dos bases de datos y la creación y la configuración de colas. Además el Streams permite mediante una interfaz de usuario administrar los objetos que son replicables. [2]

Oracle Streams permiten la propagación y la administración de datos, de transacciones y de eventos en una secuencia de datos dentro de una base de datos, o de una base de datos a otra. El Streams enruta la información publicada a los destinos suscritos. El resultado es una nueva característica que proporciona mayor funcionalidad y flexibilidad que las soluciones tradicionales para la captura y el manejo de los eventos compartidos con otras bases de datos y aplicaciones. Mientras que las necesidades de los usuarios cambian, pueden poner simplemente una nueva capacidad del Oracle Streams en ejecución, sin sacrificar capacidades existentes. Especificando la configuración de los elementos que actúan en Streams, un usuario puede tratar requisitos específicos. [2]

1.1.4 Otras herramientas propietarias

InterBase Replication Manager

El servidor de InterBase Replication está compuesto por el Replication Manager y por el Replication Server. El servidor de replicación es escrito en el lenguaje ' C ', y usa directamente llamadas del API de InterBase. Esto quiere decir que puede correr en cualquier plataforma soportada por InterBase, y sin necesidad de un software intermedio (como BDE u ODBC) para conectarse a InterBase.[3]

El Replication Manager cuenta con una serie de funciones básicas, muy útiles a los que utilizan esta herramienta, como son:

- Configurar herramientas de la replicación.

- Especificar los servidores de bases de datos que van a replicar.
- Especificar base de datos origen y destino de la replicación.
- Definir los parámetros de configuración de la base de datos origen
- Especificar las tablas que van a replicar
- Especificar los campos de cada tabla que va a replicar.
- Definir las llaves primarias de cada tabla
- Autenticar los usuarios que replican.
- Establecer condiciones para la replicación.

Esta herramienta cuenta además con una serie de facilidades para los administradores como son:

- Implementa la resolución de conflictos en la replicación.
- Permite realizar gráficamente el monitoreo de la replicación.
- Cuenta con un sistema de alertas que permite conocer si hubo algún error en la replicación.
- Muestra una serie de reportes que ayudan a la toma de decisiones de los administradores.

Después de haber analizado la herramienta podemos llegar a la conclusión que la misma tiene todas las funcionalidades necesarias para realizar una administración y un monitoreo eficiente y seguro del proceso de replicación. La interfaz con el usuario es muy profesional para el trabajo de administración, ya que permite configurar y monitorear fácilmente. Presenta además los tres tipos de réplica en una dirección, en ambas y en muchas al mismo tiempo. La herramienta es una buena solución para el gestor de bases de datos InterBase.

1.1.4 Las herramientas de administración de réplica para gestores de bases de datos libres

Entre las principales herramientas para la administración del proceso de replicación para gestores de bases de datos de libre acceso se encuentran:

Slony-I

Slony es un sistema de replicación para PostgreSQL. Esta basado en un sistema de un maestro y múltiples esclavos. Slony permite realizar la replicación para bases de datos grandes, pero solo en un número limitado de servidores (cerca de una docena). Este sistema esta creado básicamente para bases

de datos distribuidas, en las que se tenga un centro de datos y varios servidores conectados a este, donde cada nodo este asegurado.

Entre las principales limitaciones de esta herramienta encontramos que no implementa la resolución de conflictos ni el monitoreo de errores, no propaga los cambios en los esquemas ni replica objetos grandes. Slony-I implementa un modelo particular para realizar la replicación asincrónica, utilizando disparadores para recopilar las tablas que han sido modificadas y esto lo replica a los otros servidores conectados al servidor central y los que puedan estar conectados a ellos. [4]

En forma de resumen podemos decir que esta herramienta solo se encarga de realizar la replicación, recomendando que el número de servidores al que se va a replicar no sea grande y no brinda una serie de funcionalidades necesarias para el apoyo al trabajo de los administradores como el monitoreo y la resolución de conflictos.

Otra herramienta permite la migración de datos es Monarca Enterprise 2.0. Y lo hace desde cualquier base de datos hacia cualquier base de datos utilizando tecnología JDBC. Posee un conjunto de características tales como permitir transformar y validar los datos antes de que estos lleguen a la base de datos destino. Provee dos métodos para realizar migraciones de datos, el primero es a través del ADM Wizard (por sus siglas, Automated Database Migration) y el segundo es a través de DMD (por sus siglas, Database Migration Diagrams). Una de la característica más popular es que permite guardar los registros fallidos durante cualquier proceso de migración para una revisión manual posterior. Una vez que el usuario logre corregir cualquier error en los registros fallidos, la herramienta se encarga de enviarlo nuevamente a su base de datos destino. La interfaz con el usuario es un poco pobre en cuanto al monitoreo del proceso y además no permite que se puedan definir reglas para replicar los datos.

Como conclusión debemos decir que la mayoría de las herramientas de replicación propietarias brindan una mayor cantidad de funcionalidades que las libres, debido principalmente a que se centran en un servidor de bases de datos específico y que además que implementan el algoritmo del proceso de réplica dentro de su servidor.

1.2 HERRAMIENTAS PROPUESTAS PARA EL DESARROLLO DEL SISTEMA

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta la tendencia actual y las novedades en este campo se hace la siguiente propuesta.

1.2.1 La tecnología Java

El lenguaje de programación Java es actualmente un lenguaje de alto nivel, el mismo tiene una de las comunidades más grandes del planeta. Todo esto se debe al desarrollo que ha tenido dicho lenguaje en los últimos años y por que es libre. El lenguaje permite utilizar de manera eficaz los principales patrones de arquitectura de software actuales. Es uno de los lenguajes que es constantemente mejorado por la Sun Microsystems, la cual lo creo, es un de los lenguajes más recomendados para el trabajo con bases de datos, Oracle esta implementado en Java por poner un ejemplo.

Sin embargo, la característica más atractiva de Java desde el punto de vista de Internet, son sus applets, que se pueden ejecutar en cualquier plataforma con cualquier sistema operativo, constituyen también su talón de Aquiles. Si desde cualquier página Web que visitamos nos pueden enviar una applet que se ejecute en nuestro ordenador sin nuestro conocimiento, todo tipo de ataques podría tener lugar: esas applets podrían cifrar el contenido de nuestro disco duro y luego su autor pedirnos una suma millonaria por entregarnos la clave; podrían introducirnos virus; podrían robar todo tipo de información de nuestro ordenador; podrían explotar recursos de nuestro sistema, como ciclos de CPU, y un largo etcétera fácil de imaginar.

Con el fin de que estos escenarios no fueran posibles, se implementó un sistema de seguridad integrado por cuatro líneas de defensa. Otra de las razones por la cual escogemos esta tecnología es las posibilidades que brinda la gran cantidad de frameworks que existen para Java hoy en día, los casos más conocidos son Spring 3.0, Hibernate, JFreeChart, JasperReports, Struts, entre otros. La utilización de los mismos permite una mayor rapidez a la hora de implementar cualquier aplicación empresarial.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Actualmente la República Bolivariana de Venezuela realiza grandes esfuerzos para mejorar las condiciones de vida del pueblo venezolano. Dentro de este marco, el gobierno realiza esfuerzos por revertir la crítica e inhumana situación que presentan los establecimientos penitenciarios, como consecuencia del deterioro progresivo de su sistema penitenciario.

En el marco de estos esfuerzos gubernamentales, el 24 noviembre de 2004 se decreta la Emergencia Penitenciaria (Decreto N° 3.265) con el propósito de crear condiciones para: a) acometer un conjunto de acciones de impacto directo sobre las problemáticas más comprometedoras que presenta el Sistema Penitenciario; y b) la ejecución del Proyecto de Modernización del Sistema Penitenciario.

Para darle cumplimiento al segundo inciso del decreto ley, se propone la realización del proyecto SIGEP. El Sistema de Gestión Penitenciaria (SIGEP), dará respuesta a las necesidades de gestión, información y apoyo a la toma de decisiones de la Dirección General de Custodia y Rehabilitación del Recluso (DGCRR), en sus tres niveles, que se describen a continuación:

- **Operativo:** Integrado por los Establecimientos penitenciarios (Internados Judiciales y Centros Penitenciarios), Centros de Tratamiento Comunitario (CTC) y Unidades Técnicas de Apoyo al Sistema Penitenciario (UTASP).
- **Táctico:** integrado por las Coordinaciones Regionales.
- **Estratégico:** integrado por la Dirección General de Custodia y Rehabilitación del Recluso.

Con este sistema automatizado se busca:

- Aumentar la eficacia, profesionalismo y equidad en el sistema penitenciario venezolano, de manera de lograr un incremento de la confianza en el sistema penitenciario en general.
- Generar y diseminar información vital para el funcionamiento de los establecimientos penitenciarios. Esto permitirá generar estadísticas confiables y actualizadas sobre la situación jurídica de los privados de libertad, condiciones de vida y salud, actividades de rehabilitación y reinserción, la situación operativa, la actividad administrativa, entre otras.

- Permitir la comunicación en línea con tribunales, sistemas de identificación y antecedentes penales, que complementan la información necesaria para la gestión de los procesos vinculados a los privados de libertad.

Por la ubicación distante de la mayoría de los establecimientos penitenciarios y la necesidad de mantener actualizada la información del centro de datos, es imprescindible garantizar con este punto, una conexión disponible y de calidad, desde los establecimientos penitenciarios. Incluso algunas tareas administrativas remotas no podrán ejecutarse si la conectividad no es buena, como tampoco podrían realizarse consultas a la base de datos central, debido a esto se realizara replicación automática de los datos desde el centro de datos hacia los establecimientos penitenciarios y viceversa.

Entre los principales procesos a automatizar se encuentran:

1. El proceso de replicación de datos entre las prisiones y el centro de datos.
2. Generación de reportes
3. Limpieza y corrección de conflictos ocurridos en el proceso de replicación.
4. Monitoreo de Transacciones que llegan al centro de datos.

2.1 Descripción de los procesos que serán objeto de automatización:

1. El proceso de configuración de la base para la replicación de datos entre los establecimientos penitenciarios y el centro de datos.

El proceso se inicia cuando el administrador de bases de datos una vez instalada la aplicación en cada uno de los nodos y en el centro de datos, decide iniciar el proceso de configuración de la réplica entre cada uno de los nodos y el centro de datos, para ello, debe contar con el script que tiene implementado el algoritmo de réplica de la UCI, el cual esta sin configurar todavía. Después de instalar un software cliente para conectarse al Oracle, abre el script con el cliente y procede a configurar el mismo editándolo, en el script se especifican los usuarios y las contraseñas que hacen falta para conectarse desde la aplicación a la base de datos y los usuarios que van a replicar, además se especifican los parámetros de

almacenamiento y envío de las transacciones, en el caso del centro de datos se especifican los servidores autorizados a replicar con sus datos correspondientes, y por último se especifican las tablas que van a replicar así como las operaciones por las cuales van a replicar cada una. Se crean las tablas que intervienen en el proceso de la réplica, los paquetes, las colas de envío y procedimientos almacenados. Luego de terminada la configuración del script se procede a ejecutar el mismo en el cliente de Oracle. Luego de ejecutarse el script, en cada uno de los nodos y en el centro de datos, se puede realizar la réplica de los datos desde los nodos hacia el centro de datos, para comprobar que la configuración tuvo éxito se replican datos de manera aleatoria unos datos de prueba, y luego se verifica que estén en cada uno de los servidores de bases de datos de todo el sistema. Este proceso puede tardar un día entero si no ocurren fallos a la hora de editar el script, lo cual es muy frecuente. Para desconfigurar el ambiente de replicación el administrador debe borrar de manera manual todos los esquemas y los tablespaces que intervienen en el proceso de réplica.

2. Generación de reportes:

La generación de los reportes asociados al proceso de replicación se hace de manera manual, cada vez que el administrador necesita tener un conocimiento de como esta funcionando la réplica entre los establecimientos penitenciarios y el centro de datos tiene que construir una serie de consultas SQL, para luego copiar los resultados y pasarlos a hojas de cálculos y documentos Word.

3. Limpieza y corrección de conflictos ocurridos en el proceso de replicación.

Este proceso comienza cuando una transacción no es ejecutada en uno de los servidores de base de datos tanto del centro de datos como de los centros penitenciarios, cada transacción fallida es almacenada junto con todas sus operaciones asociadas a esta.

Luego el administrador tiene que revisar las tablas donde se almacenaron las transacciones fallidas para luego analizar el mensaje de error junto con la operación que ocasiono el fallo de la transacción tabla es consultada esto ocurre cada vez que el administrador de la réplica decide hacer la limpieza y corrección de las transacciones.

Una vez que el administrador le da solución al problema ejecuta de nuevo la transacción ya arreglada por el. El proceso ocurre de la misma manera a la hora de realizar la limpieza y corrección de los errores ocurridos en el llenado de las colas de envío.

4. Monitoreo de Transacciones que Llegan al centro de datos.

Cuando se termina de configurar el ambiente de replicación el administrador de bases de datos abre varios de clientes de Oracle, en cada uno de ellos abre las tablas que guardan los datos de las excepciones que llegan al centro de datos, cada cierto tiempo el administrador debe autenticarse debido a que se desconecta el cliente Oracle. Este monitoreo el administrador solo lo realiza para ver las excepciones que ocurren en el proceso de replicación.

2.2 Descripción general de la propuesta de sistema, como debe funcionar.

El sistema será una aplicación Web, en la cual los usuarios se conectan a ella a través del cualquier navegador. Una vez en la página de autenticación, se procede a acceder a la misma introduciendo usuario y contraseña, si la autenticación es satisfactoria, en la aplicación aparece una pantalla que contiene todas las opciones que tiene el usuario para poder administrar el proceso de réplica esto ocurre en el caso de que ya este configurado la base para replicar en caso contrario aparece una interface indicándole al usuario que debe configurar el ambiente de la réplica para poder realizar la administración y el monitoreo de la misma.

El usuario escoge en que ambiente a va configurar la base de la réplica, si en el centro de datos o en uno de los establecimientos penitenciarios, y aparece una pantalla describiéndole el proceso a iniciar. Durante el proceso el usuario puede observar el código SQL que se va creando a medida que va configurando la base de la réplica este código de muestra aparece en una pestaña adyacente a la pestaña de configuración.

En el sistema que se instala en el centro de datos se puede configurar los servidores autorizados a replicar. El sistema permite realizar el monitoreo de las transacciones que llegan al centro de datos, en tiempo real a través de un monitor que esta refrescándose cada el tiempo que escoja el administrador además se puede hacer la corrección de las transacciones que fallan en cada uno de los establecimientos penitenciarios a través del monitoreo, ya que cada ves que falla una transacción se genera una alerta que es almacenada. El sistema permite generar reportes del proceso de replicación, los mismos pueden se pueden mostrar en diferentes formatos, hojas de calculo, pdf y otros. El sistema puede publicarse el mismo servidor de aplicación de la bases de datos o en el servidor Web del SIGEP.

2.3 Requerimientos Funcionales

1. Preparar el ambiente de réplica en el centro de datos
 - 1.1 Especificar parámetros de almacenamiento en el centro de datos
 - 1.2 Especificar los servidores que van a replicar
 - 1.4 Introducir parámetros de almacenamiento y envió.
2. Preparar el ambiente de réplica en los establecimientos penitenciarios
 - 2.1 Especificar los parámetros de almacenamiento
 - 2.2 Especificar las tablas que van a replicar
3. Realizar limpieza y corrección de conflictos ocurridos en el proceso de replicación
 - 3.1 Almacenar transacciones por servidor
 - 3.2 Mostrar transacciones fallidas
 - 3.3 Mostrar transacciones con inconsistencia
 - 3.4 Identificar fuente de la transacción
 - 3.5 Identificar destino transacción
 - 3.6 Aplicar la transacción
4. Monitorear el flujo de transacciones en el centro de datos
 - 4.1 Mostrar las transacciones que van llegando al centro de datos.
 - 4.2 Mostrar estado de las transacciones
 - 4.3 Mostrar las operaciones de una transacción.
 - 4.4 Mostrar de manera grafica las transacciones en tiempo real.
5. Administrar Usuarios

- 5.1 Crear usuarios de la réplica
- 5.2 Modificar usuarios de la réplica
- 5.3 Eliminar usuarios de la réplica
- 6. Administrar Objetos de la réplica
 - 6.1 Crear tablas de la replicación
 - 6.2 Modificar tablas de la replicación
 - 6.3 Eliminar tablas de replicación
 - 6.4 Crear tablas para especificar servidores de la réplica
 - 6.5 Modificar tablas donde se especifican servidores de la réplica.
 - 6.6 Eliminar tablas para eliminar servidores de la réplica
 - 6.7 Mostrar servidores registrados.
- 7. Administrar Tablespace
 - 7.1 Crear tablespace
 - 7.2 Modificar tablespace
 - 7.3 Eliminar tablespace
- 8. Autenticar Usuarios
 - 8.1. Permitir autenticar un usuario
 - 8.2 Comprobar usuario y contraseña
 - 8.3 Permitir al usuario acceder a las páginas que tenga permisos.
- 9. Listar Reportes
 - 9.1 Mostrar cantidad de transacciones que llegaron al centro de datos de un establecimiento penitenciario en un periodo de tiempo determinado.
 - 9.2 Mostrar que centro penitenciario generó la mayor cantidad de transacciones en un periodo de tiempo determinado.

2.4 Requerimientos no Funcionales

1. Apariencia o interfaz externa

- El sistema debe ser simple de usar para administradores de bases de datos, legible, profesional, tanto los colores como los estilos que se utilizan son los mismos del sistema de gestión penitenciaria.

2. Usabilidad

- Este sistema está diseñado para personas con conocimientos previos de Oracle y réplica de datos, con la introducción del producto se mejora el proceso de réplica y se agiliza el trabajo de los administradores de la misma, reduciendo grandemente los errores en la administración del proceso.

3. Rendimiento

- El sistema debe ser capaz de procesar y mostrar la información requerida por el usuario en un tiempo que no exceda los 20 segundos.

4. Soporte

- Una vez terminado el sistema se le realizarán pruebas, se continuará extendiendo las funcionalidades del mismo con el objetivo de mejorar la administración y el monitoreo del proceso de replicación, se valorará la posibilidad de hacer internacionalización del sistema a otros idiomas, se le dará mantenimiento cada vez que el usuario presente alguna sugerencia o el sistema de bases de datos cambie.

5. Portabilidad

- Al sistema se podrá acceder a través de cualquier navegador Web, siendo este independiente de la plataforma.

6. Seguridad

- Confidencialidad: La información manejada por el sistema estará protegida de acceso no autorizado.
- Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.
- Disponibilidad: Significa que los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.
- El sistema estará protegido en un cuarto donde solo tendrán acceso personal autorizado, la comunicación con la aplicación será a través de red inalámbrica.

7. Políticos Culturales

- El sistema esta implementado para una de las organizaciones gubernamentales de Venezuela, el sistema penitenciario, por esta razón el sistema tendrá el logo de la organización como identificación y los colores que se utilizarán son los de la bandera de Venezuela.

8. Ayuda y documentación en línea

- El sistema contará con una ayuda incluida en el mismo ya que por cada una de las pantallas aparecerá asociado una página de ayuda explicando la pantalla al usuario. Además junto con el software vendrá un manual de usuario e instalación.

9. Software

- El sistema es una aplicación implementada en Java y utiliza los frameworks Spring, JfreeChart y JasperReports, como servidor Web utiliza Apache Tomcat o el servidor de aplicaciones OC4J del Oracle 10g.

10. Hardware

- Puede correr en cualquier maquina con procesador Pentium III o superior, el mínimo de RAM recomendado es 512 MB, necesita como mínimo 20 MB de espacio libre en disco.

CAPÍTULO 3: ANÁLISIS Y DISEÑO

3.1 Análisis.

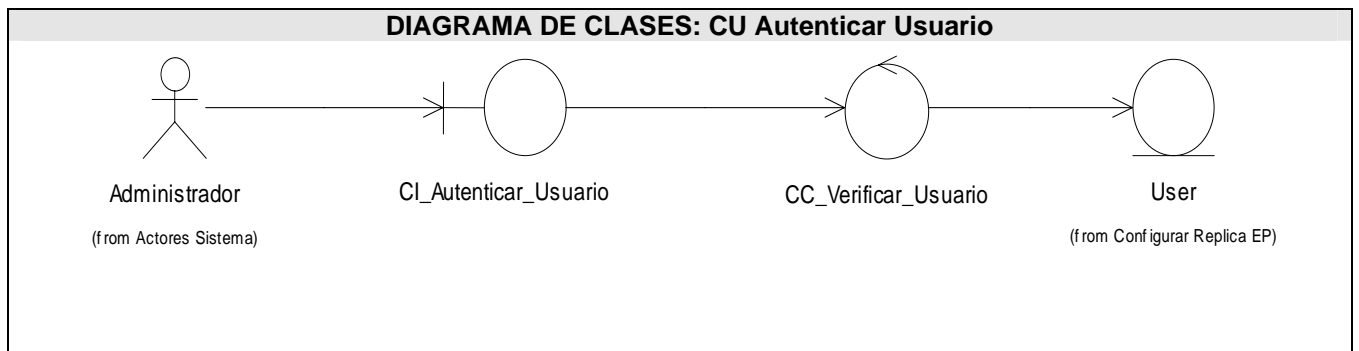
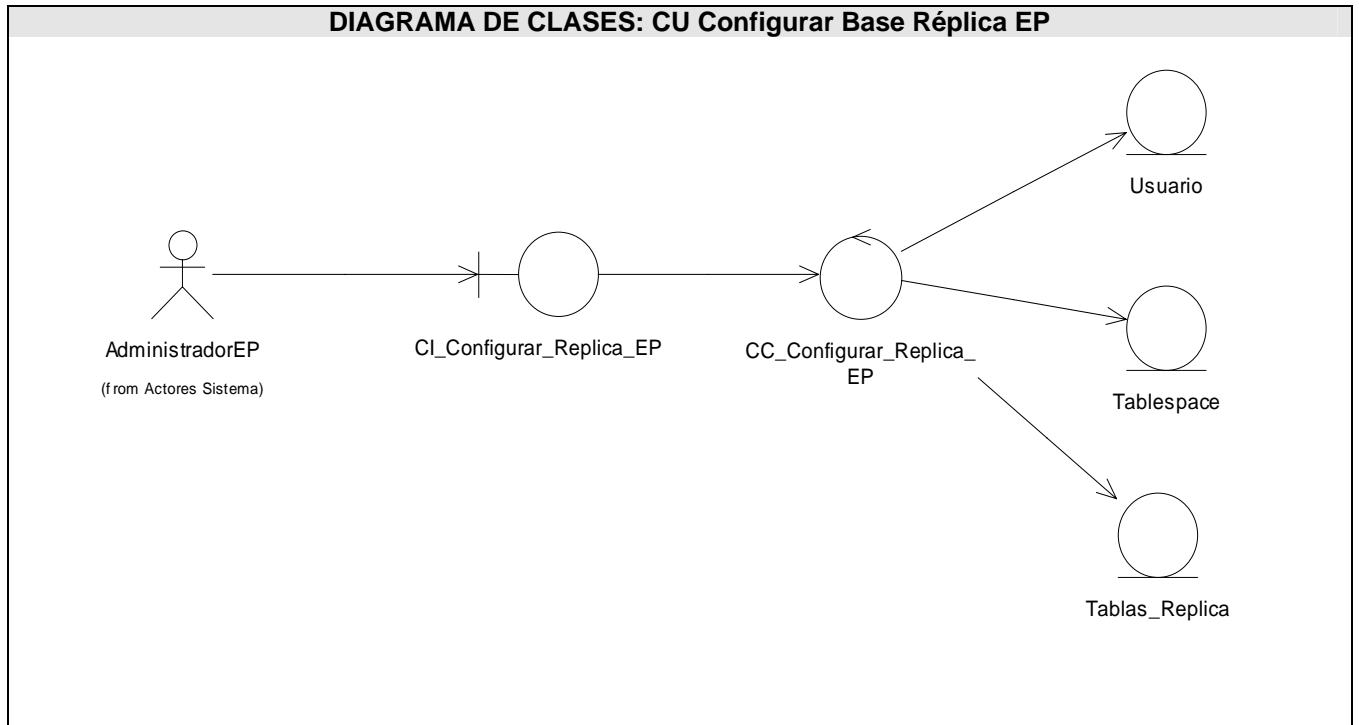


DIAGRAMA DE CLASES: CU Mostrar Reportes

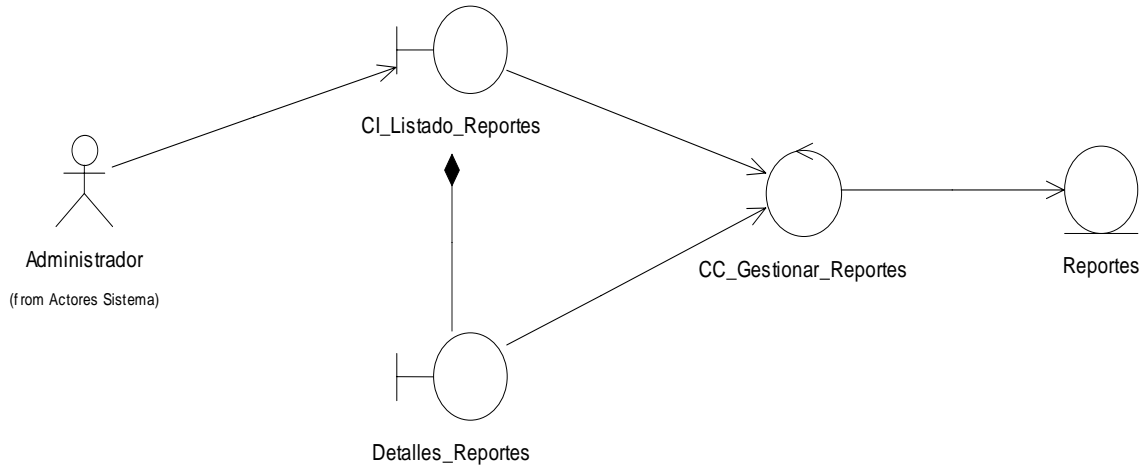


DIAGRAMA DE CLASES: CU Gestionar Servidores Replicación

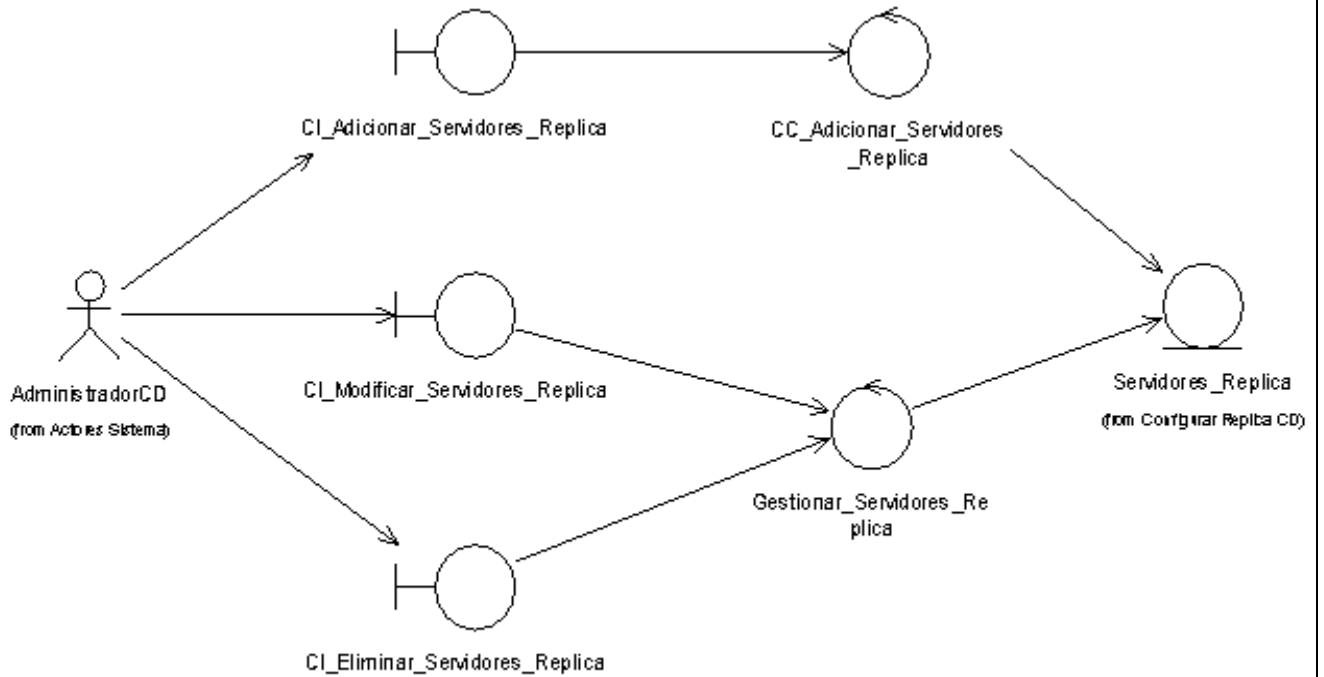


DIAGRAMA DE CLASES: CU Gestionar Tablas Répica

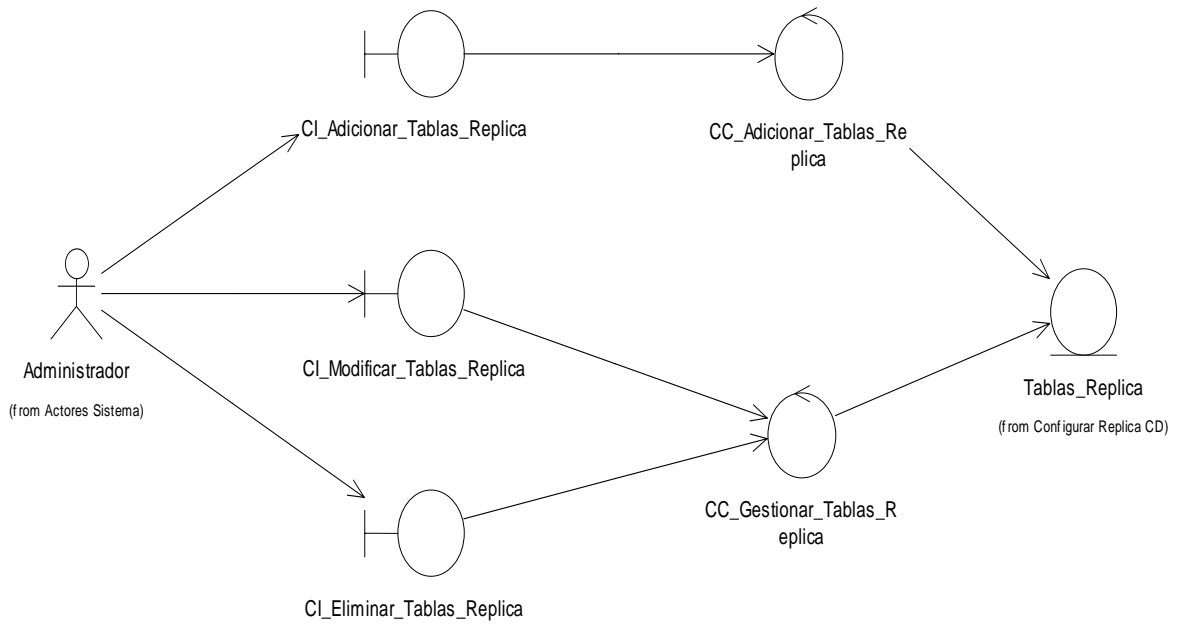


DIAGRAMA DE CLASES: CU Gestionar Usuario

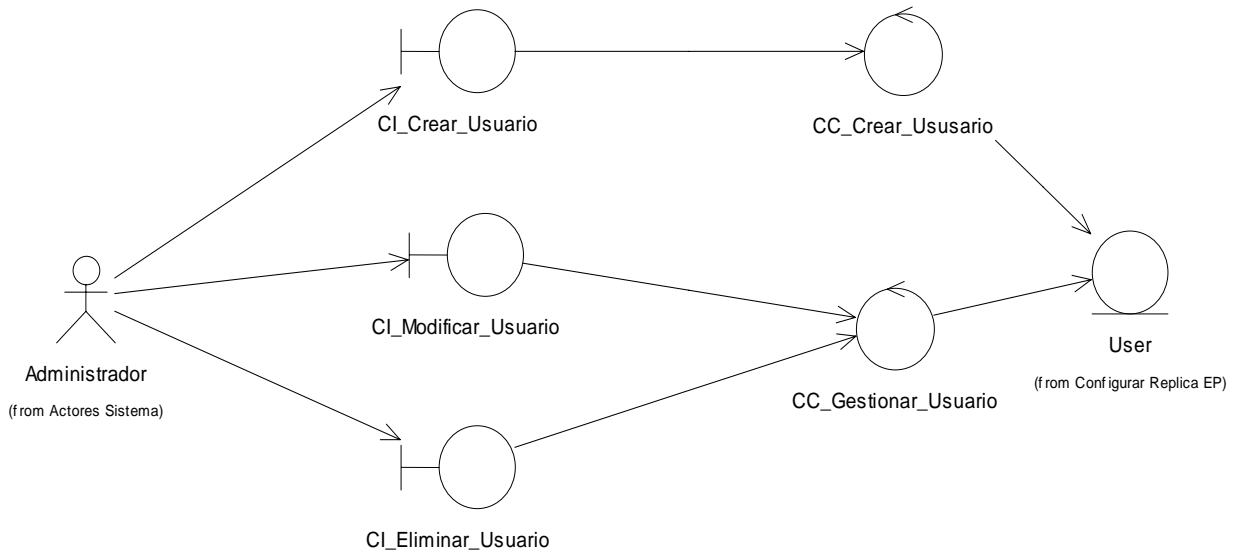


DIAGRAMA DE CLASES: CU Monitorear Transacciones CD

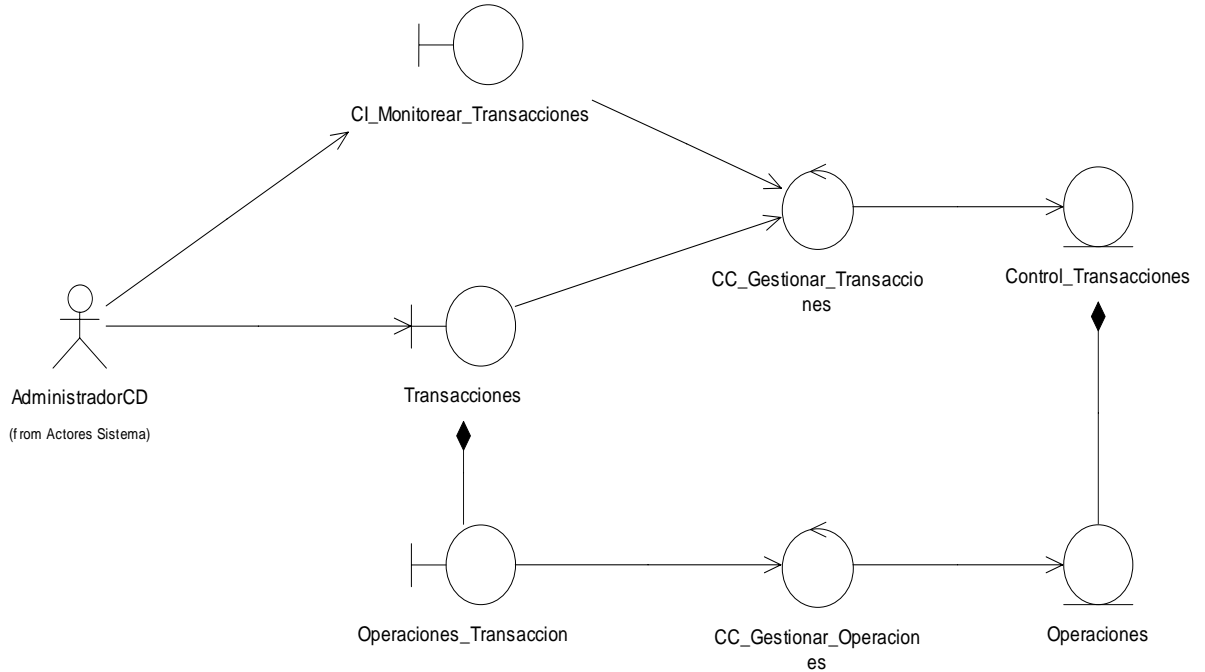


DIAGRAMA DE CLASES: CU Gestionar Tablespace

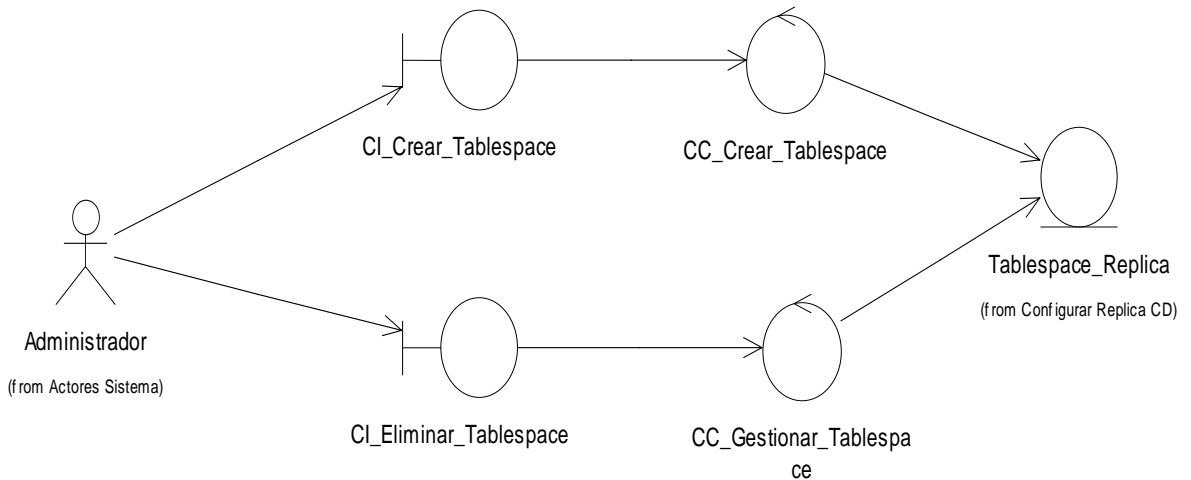
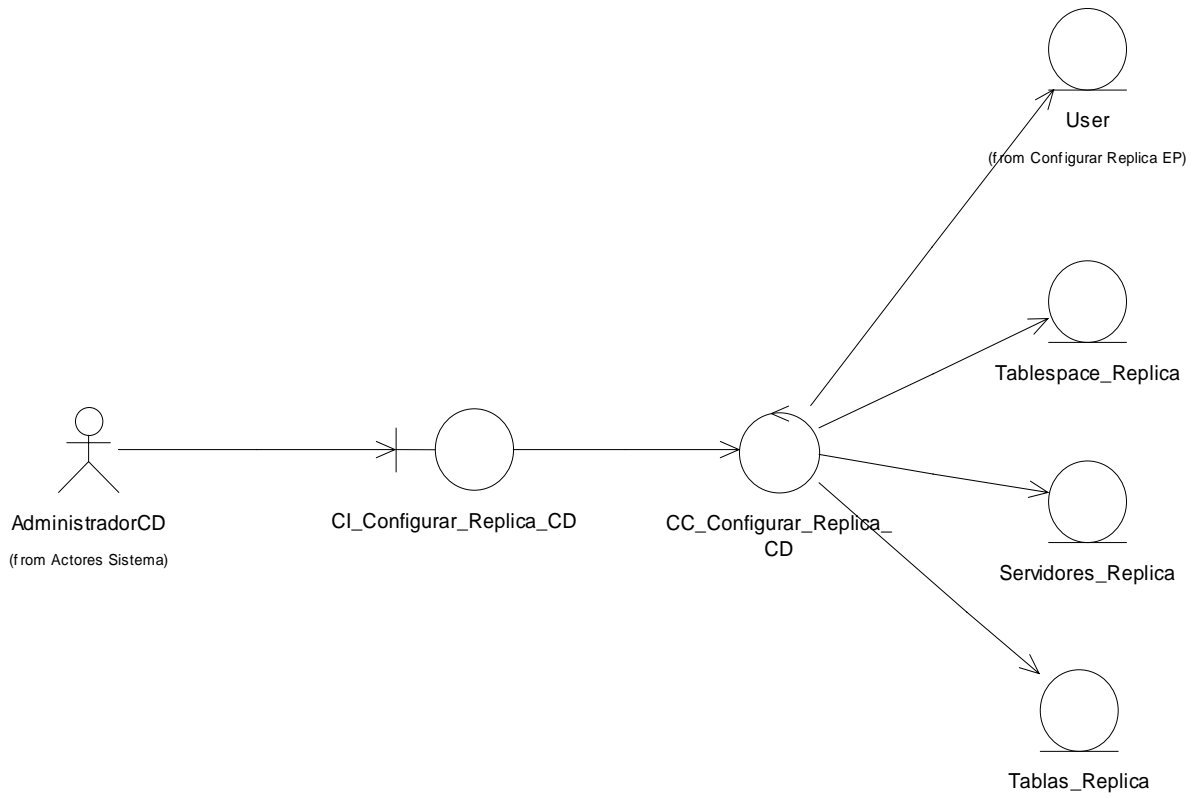


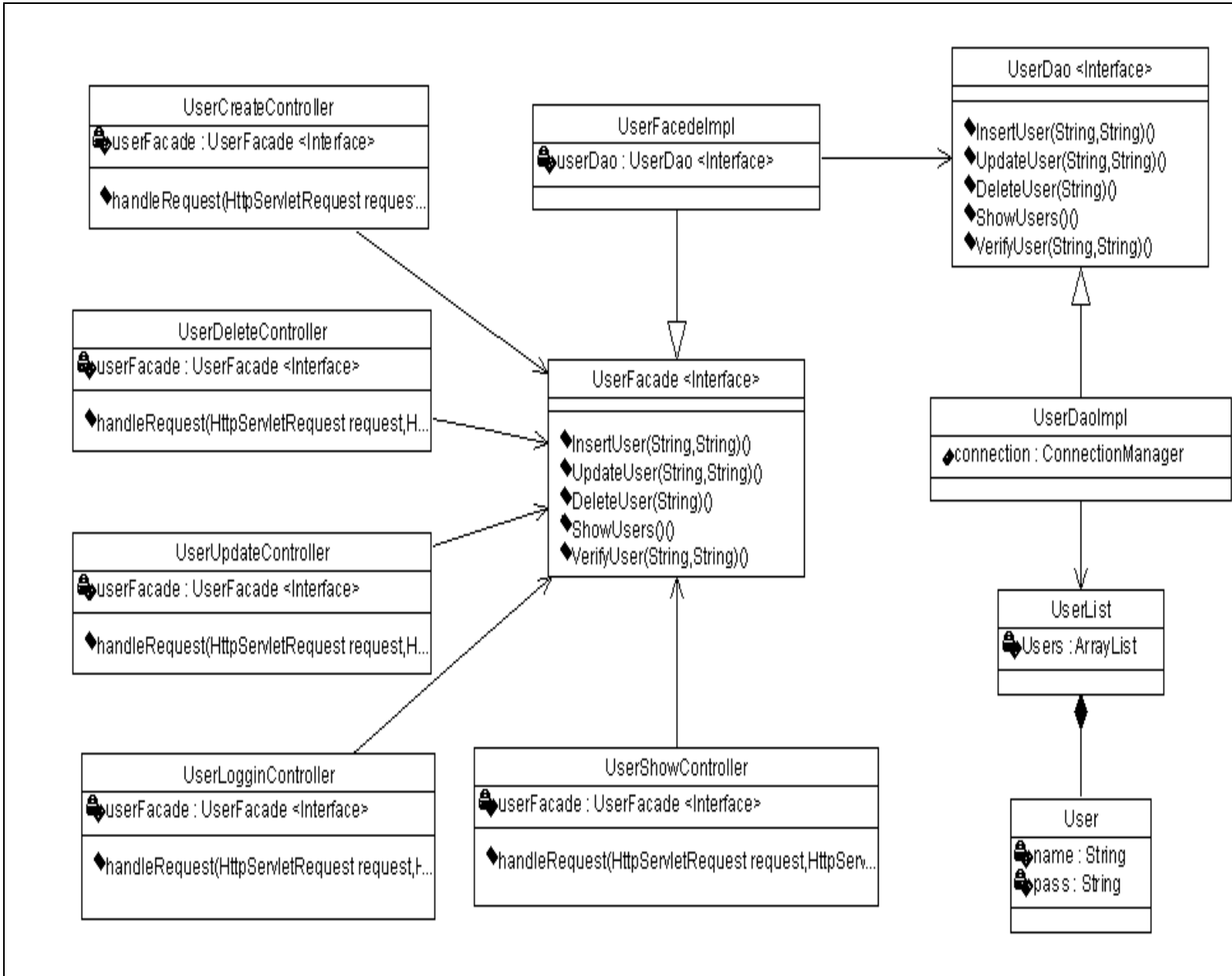
DIAGRAMA DE CLASES: CU Configurar Base Réplica CD



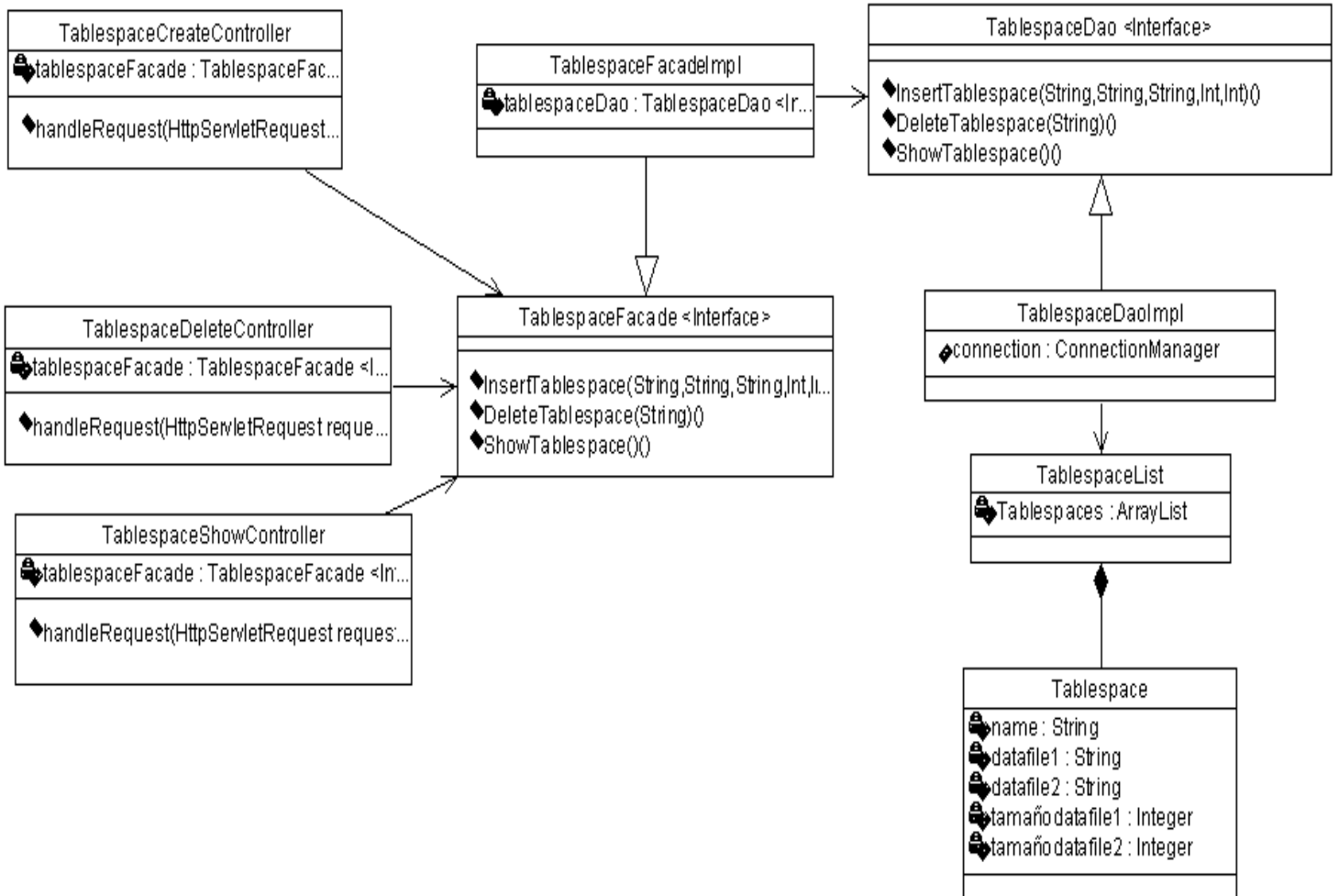
3.2 DISEÑO.

3.2 .2 Diagramas de Clases

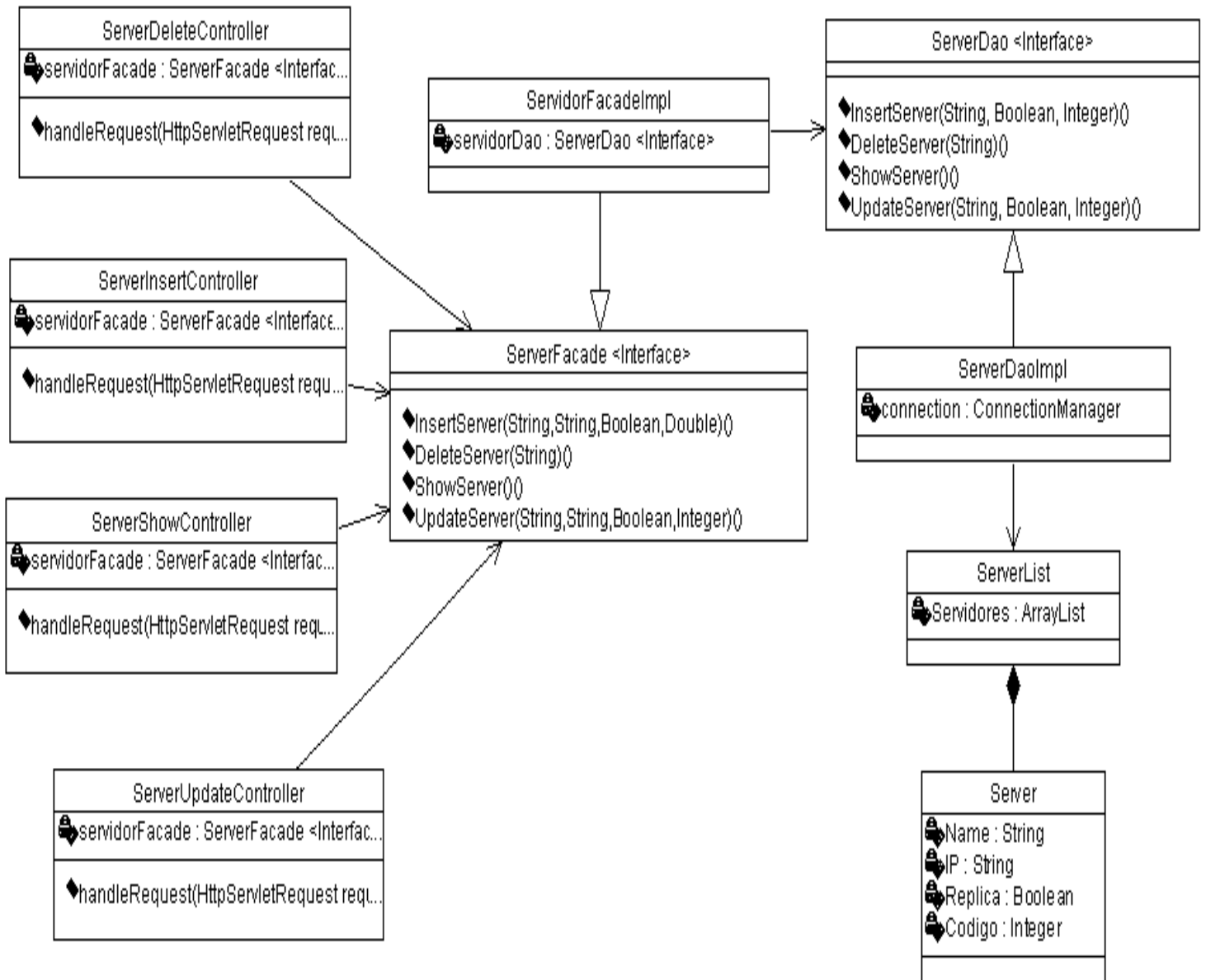
Gestionar Usuario



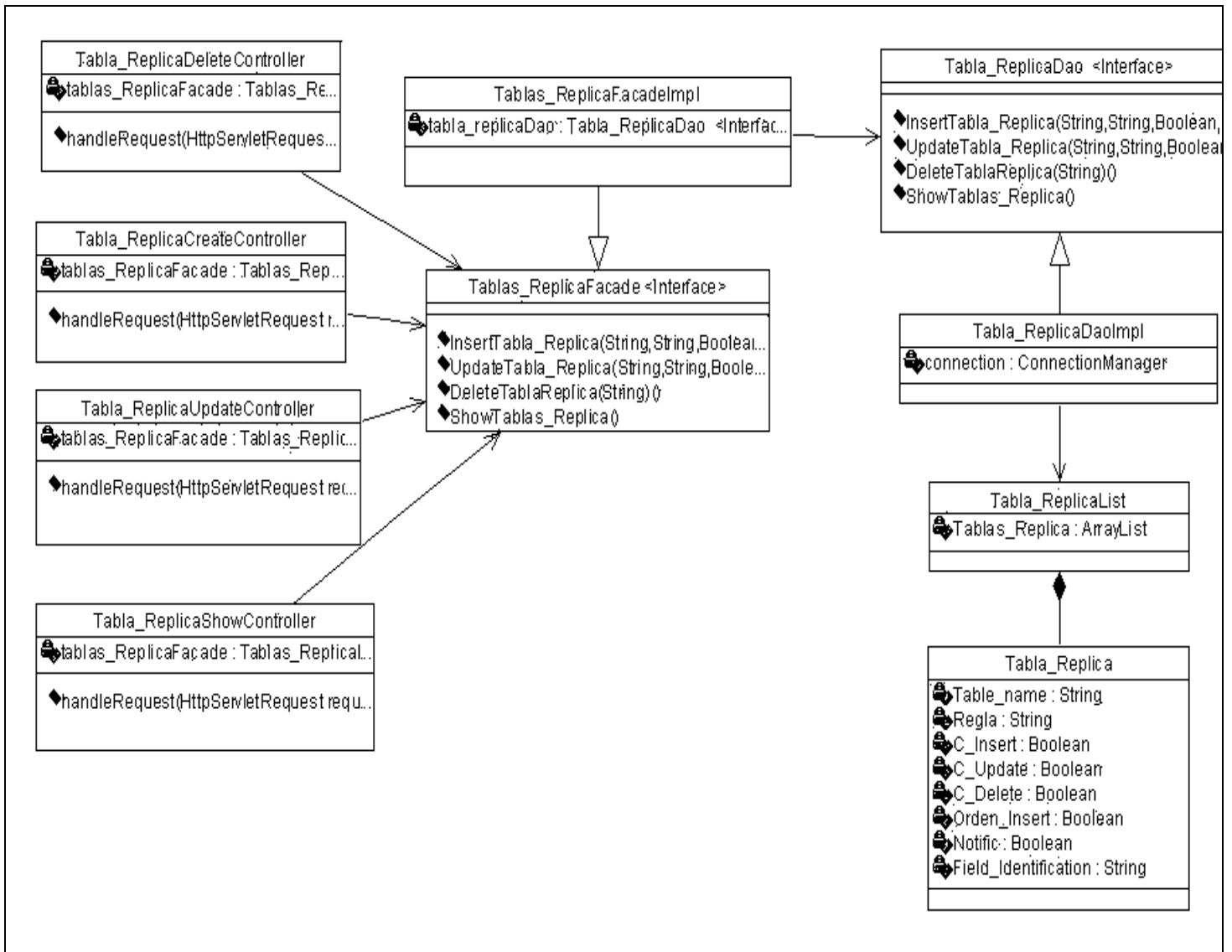
Gestionar Tablespace



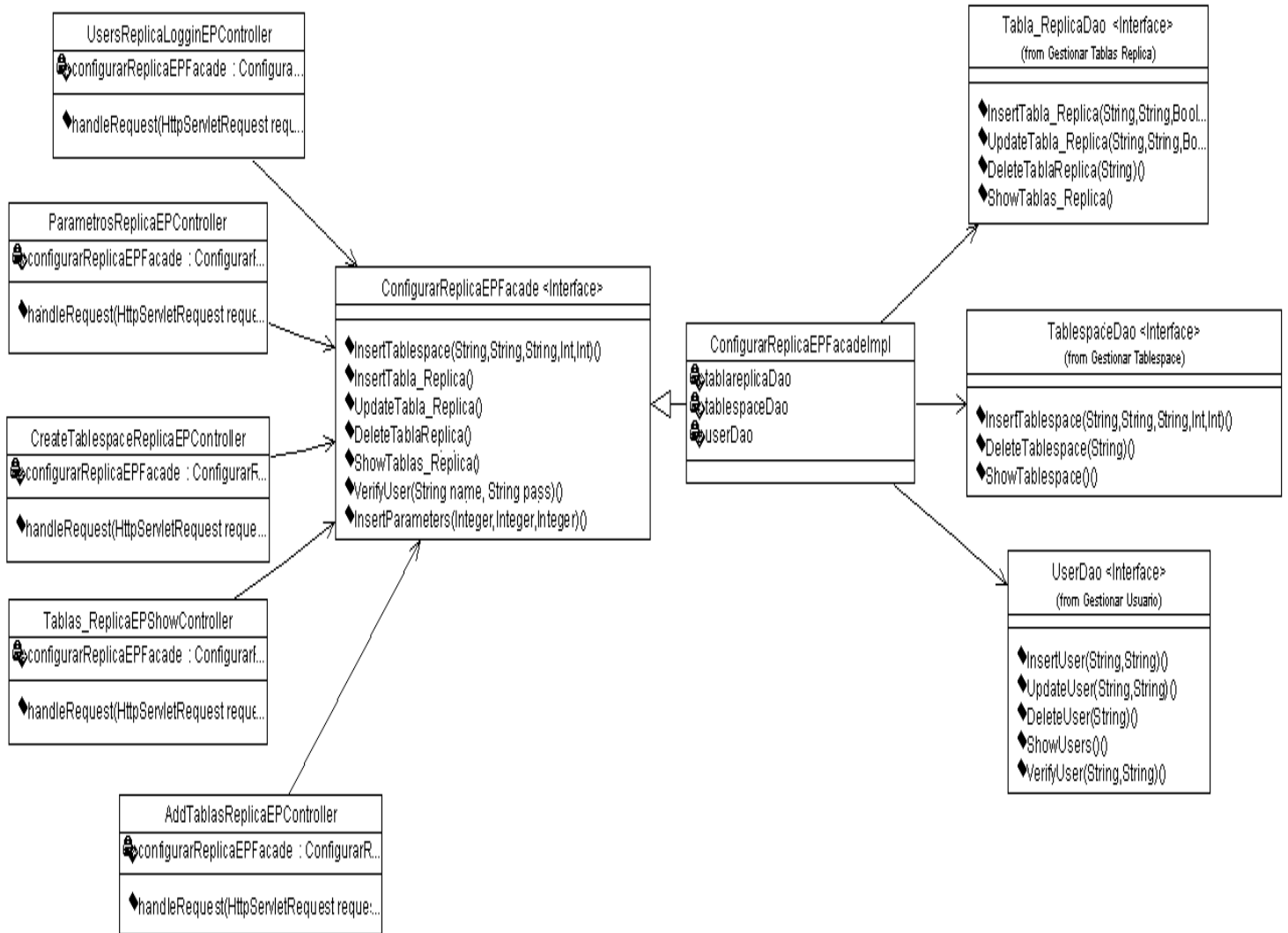
Gestionar Servidores



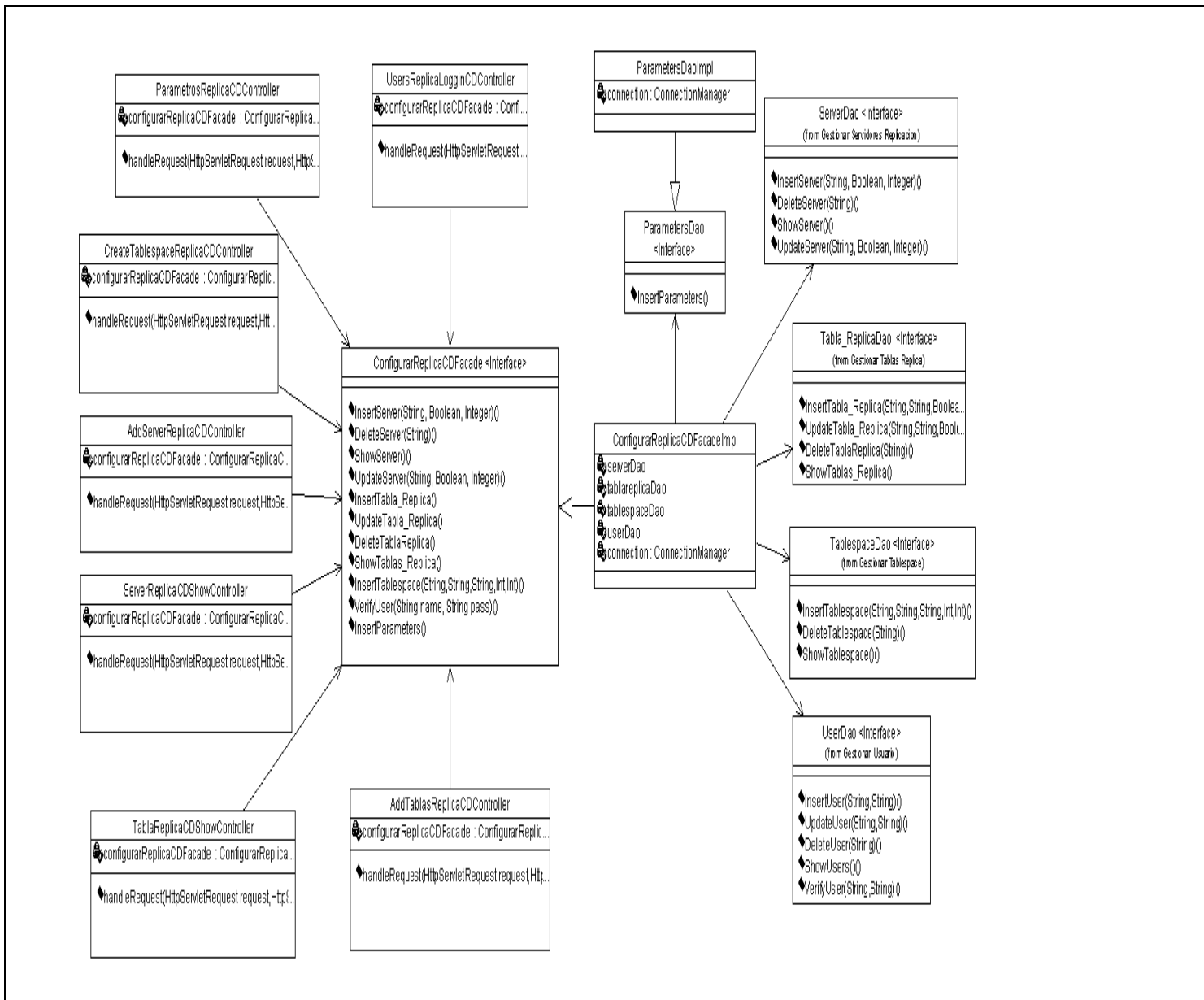
Gestionar Tablas Réplica



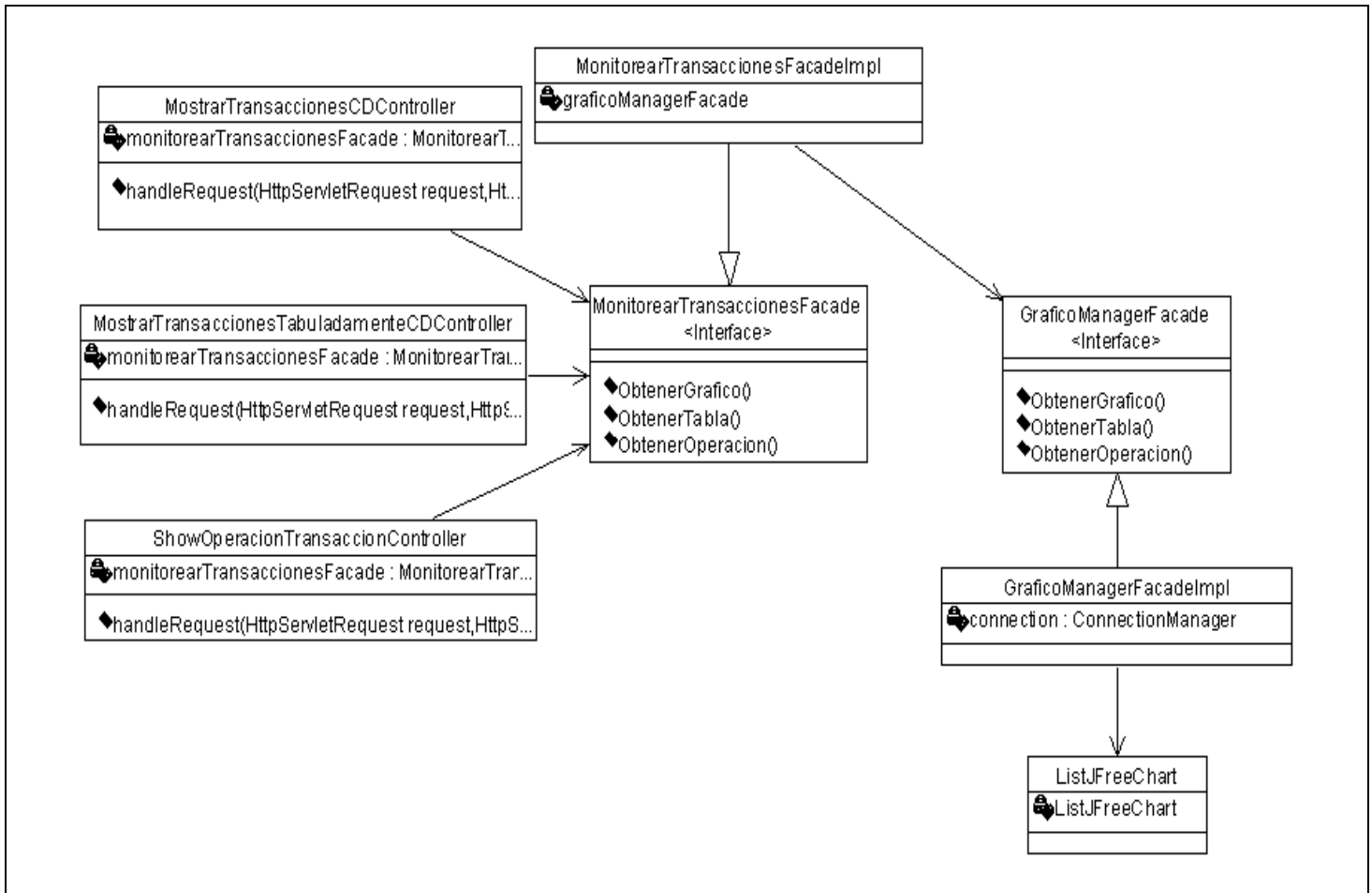
Configurar Réplica EP



Configurar Réplica CD



Monitorear Transacciones



.3.2.3 Descripción de las Clases

| | |
|-------------------------------|-------------|
| Nombre: User | |
| Tipo de clase: Entidad | |
| Atributo | Tipo |
| name | String |
| pass | String |

| | |
|-------------------------|-----------------|
| Nombre: UserList | |
| Tipo de clase: | |
| Atributo | Tipo |
| users | ArrayList<user> |

| | |
|-----------------------------------|--|
| Nombre: Interface_UserDao | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre: | InsertUser(String name, String pass) |
| Descripción: | Este es método abstracto que es implementado en la clase UserDaoImpl |
| Nombre: | UpdateUser(String name, String pass) |
| Descripción: | Este es método abstracto que es implementado en la clase UserDaoImpl |
| Nombre: | DeleteUser(String name) |
| Descripción: | Este es método abstracto que es implementado en la clase UserDaoImpl |
| Nombre: | ShowUsers() |
| Descripción: | Este método muestra todo los usuarios de la aplicación. |
| Nombre: | VerifyUser(String name, String pass) |
| Descripción: | Este es método abstracto que es implementado en la clase UserDaoImpl |

| | |
|----------------------------|-------------------|
| Nombre: UserDaoImpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| Conecction | ConecctionManager |

| | |
|-----------------------------------|--|
| Para cada responsabilidad: | |
| Nombre: | InsertUser(String name, String pass) |
| Descripción: | Este método se conecta a la base de datos, inserta los datos del usuario y retorna verdadero si los inserto correctamente o falso si no los pudo insertar. |
| Nombre: | UpdateUser(String name, String pass) |
| Descripción: | Este método se conecta a la base de datos, modifica los datos del usuario y retorna: verdadero si los modifíco correctamente, falso si no lo pudo modificar. |
| Nombre: | DeleteUser(String name) |
| Descripción: | Este método se conecta a la base de datos y elimina el usuario cuyo nombre es pasado como parámetro. |
| Nombre: | ShowUsers() |
| Descripción: | Recupera de la base de datos todos los usuarios, crea un objeto de tipo UserList, el cual es llenado con los datos de la base de datos para luego retornarlo. |
| Nombre: | VerifyUser(String name, String pass) |
| Descripción: | Este método se conecta a la base de datos, busca el usuario por los datos pasados como parámetros, en caso de encontrarlos crea un objeto de tipo User y lo retorna. |

| | |
|-----------------------------------|--|
| Nombre: UserFacade | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre: | InsertUser(String name, String pass) |
| Descripción: | Este es un método abstracto que es implementado en la clase UserFacadeImpl |
| Nombre: | UpdateUser(String name, String pass) |
| Descripción: | Este es un método abstracto que es implementado en la clase UserFacadeImpl |
| Nombre: | DeleteUser(String name) |
| Descripción: | Este es un método abstracto que es implementado en la clase UserFacadeImpl |
| Nombre: | ShowUsers() |

| | |
|---------------------|--|
| Descripción: | Este es un método abstracto que es implementado en la clase UserFacadeImpl |
| Nombre: | VerifyUser(String name, String pass) |
| Descripción: | Este es un método abstracto que es implementado en la clase UserFacadeImpl |

| | |
|-----------------------------------|---|
| Nombre: UserFacadeImpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| UserDao | Interface_UserDao |
| Para cada responsabilidad: | |
| Nombre: | InsertUser(String name, String pass) |
| Descripción: | Este método es el encargado de llamar al método InsertUser que se encuentra en la Interfaz UserDao. |
| Nombre: | UpdateUser(String name, String pass) |
| Descripción: | Este método es el encargado de llamar al método UpdateUser que se encuentra en la Interfaz UserDao. |
| Nombre: | DeleteUser(String name) |
| Descripción: | Este método es el encargado de llamar al método DeleteUser que se encuentra en la Interfaz UserDao. |
| Nombre: | ShowUsers() |
| Descripción: | Este método es el encargado de llamar al método ShowUsers que se encuentra en la Interfaz UserDao. |
| Nombre: | VerifyUser(String name, String pass) |
| Descripción: | Este método es el encargado de llamar al método VerifyUser que se encuentra en la Interfaz UserDao. |

| | |
|-------------------------------------|--|
| Nombre: UserCreateController | |
| Tipo de clase: Controlador | |
| Atributo | Tipo |
| UserFacade | Interface_UserFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertUser de la fachada UserFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|-------------------------------------|--|
| Nombre: UserDeleteController | |
| Tipo de clase: Controlador | |
| Atributo | Tipo |
| UserFacade | Interface_UserFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método DeleteUser de la fachada UserFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|-------------------------------------|----------------------|
| Nombre: UserUpdateController | |
| Tipo de clase: Controlador | |
| Atributo | Tipo |
| UserFacade | Interface_UserFacade |

| | |
|-----------------------------------|--|
| Para cada responsabilidad: | |
| Nombre | handleRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método updateUser de la fachada UserFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|-----------------------------------|---|
| Nombre: UseLoginController | |
| Tipo de clase: Controlador | |
| Atributo | Tipo |
| UserFacade | Interface_UserFacade |
| Para cada responsabilidad: | |
| Nombre | handleRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método verifyUser de la fachada UserFacade, este método retorna un objeto de tipo User si el objeto es null se le pasa una URL con un error al ModelAndView sino se retorna el ModelAndView con una URL válida. |

| | |
|-----------------------------------|---|
| Nombre: UseShowController | |
| Tipo de clase: Controlador | |
| Atributo | Tipo |
| UserFacade | Interface_UserFacade |
| Para cada responsabilidad: | |
| Nombre | handleRequest(HttpServletRequest request) |
| Descripción | Este método llama al método userShow de la fachada y obtiene una lista de usuarios que son pasados al ModelAndView junto con la URL que envía hacia el jsp donde se muestran los usuarios |

| | |
|-------------------------------|-------------|
| Nombre: Tablespace | |
| Tipo de clase: Entidad | |
| Atributo | Tipo |
| name | String |
| datafile1 | String |
| Datafile2 | String |
| tamañodatafile1 | Integer |
| tamañodatafile2 | Integer |

| | |
|-------------------------------|-----------------------|
| Nombre: TablespaceList | |
| Tipo de clase: | |
| Atributo | Tipo |
| tablespace | ArrayList<Tablespace> |

| | |
|-----------------------------------|--|
| Nombre: TablespaceDao | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | InsertTablespace(String, String, String, Integer, Integer) |
| Descripción | Este es un método abstracto que es implementado en la clase TablespaceDaolmpl. |
| Nombre | DeleteTablespace(String) |
| Descripción | Este es un método abstracto que es implementado en la clase TablespaceDaolmpl. |
| Nombre | ShowTablespace() |
| Descripción | Este es un método abstracto que es implementado en la clase TablespaceDaolmpl. |

| | |
|-----------------------------------|---|
| Nombre: TablespaceDaolmpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| connection | ConecctionManager |
| Para cada responsabilidad: | |
| Nombre | InsertTablespace(String name, String datafile1, String datafile2, Integer tamañodatafile1, Integer tamañodatafile2) |
| Descripción | Este método se conecta a la base de datos, inserta los datos del tablespace y retorna verdadero si los inserto correctamente o falso si no los pudo insertar. |
| Nombre | ShowTablespace() |
| Descripción | Recupera de la base de datos todos los usuarios, crea un objeto de tipo TablespaceList, el cual es llenado con los datos de la base de datos para luego retornarlo. |
| Nombre | DeleteTablespace(String name) |
| Descripción | Este método se conecta a la base de datos y elimina el tablespace cuyo nombre es pasado como parámetro. |

| | |
|-----------------------------------|--|
| Nombre: TablespaceFacade | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | InsertTablespace(String, String, String, Integer, Integer) |
| Descripción | Este es un método abstracto que es implementado en la clase TablespaceFacadeImpl |
| Nombre | DeleteTablespace(String) |
| Descripción | Este es un método abstracto que es implementado en la clase TablespaceFacadeImpl |
| Nombre | ShowTablespace() |
| Descripción | Este es un método abstracto que es implementado en la clase TablespaceFacadeImpl |

| | |
|-------------------------------------|---|
| Nombre: TablespaceFacadeImpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| tablespaceDao | TablespaceDao |
| Para cada responsabilidad: | |
| Nombre | InsertTablespace(String name, String datafile1, String datafile2, Integer tamañodatafile1, Integer tamañodatafile2) |
| Descripción | Este método es el encargado de llamar al método InsertTablespace que se encuentra en la Interfaz TablespaceDao. |
| Nombre | ShowTablespace() |
| Descripción | Este método es el encargado de llamar al método ShowTablespace () que se encuentra en la Interfaz TablespaceDao. |
| Nombre | DeleteTablespace(String name) |
| Descripción | Este método es el encargado de llamar al método DeleteTablespace () que se encuentra en la Interfaz TablespaceDao. |

| | |
|---|--|
| Nombre: TablespaceCreateController | |
| Tipo de clase: Controlador | |
| Atributo | Tipo |
| tablespaceFacade | TablespaceFacade |
| Para cada responsabilidad: | |
| Nombre | handleRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertTablespace de la fachada TablespaceFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|---|--|
| Nombre: TablespaceDeleteController | |
| Tipo de clase: Controlador | |
| Atributo | Atributo |
| tablespaceFacade | tablespaceFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método DeleteTablespace de la fachada TablespaceFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|---|---|
| Nombre: TablespaceShowController | |
| Tipo de clase: Controlador | |
| Atributo | Tipo |
| tablespaceFacade | tablespaceFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | Este método llama al método ShowTablespace () de la fachada y obtiene una lista de tablespace que son pasados al ModelAndView junto con la URL que envía hacia el jsp donde se muestran los tablespace. |

| | |
|-------------------------------|-------------|
| Nombre: Server | |
| Tipo de clase: Entidad | |
| Atributo | Tipo |
| Name | String |
| IP | String |
| Replica | Boolean |

| | |
|--------|---------|
| Código | Integer |
|--------|---------|

| | |
|---------------------------|---------------------|
| Nombre: ServerList | |
| Tipo de clase: | |
| Atributo | Tipo |
| Servidores | ArrayList<Servidor> |

| | |
|-----------------------------------|--|
| Nombre: ServerDao | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | InsertServer(String, String, Boolean, Integer) |
| Descripción | Este es un método abstracto que es implementado en la clase ServerDaoImpl. |
| Nombre | DeleteServer(String) |
| Descripción | Este es un método abstracto que es implementado en la clase ServerDaoImpl. |
| Nombre | ShowServer() |
| Descripción | Este es un método abstracto que es implementado en la clase ServerDaoImpl. |
| Nombre | UpdateServer(String, String, Boolean, Integer) |
| Descripción | Este es un método abstracto que es implementado en la clase ServerDaoImpl. |

| | |
|-----------------------------------|---|
| Nombre: ServerDaoImpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| conecction | ConecctionManager |
| Para cada responsabilidad: | |
| Nombre | InsertServer(String name, String IP, Boolean replica, Integer código) |
| Descripción | Este método se conecta a la base de datos, inserta los datos del servidor y retorna verdadero si los insertó correctamente o falso si no los pudo insertar. |
| Nombre | DeleteServer(String name) |

| | |
|--------------------|---|
| Descripción | Este método se conecta a la base de datos y elimina el servidor cuyo nombre es pasado como parámetro. |
| Nombre | ShowServer() |
| Descripción | Recupera de la base de datos todos los servidores, crea un objeto de tipo TablespaceList, el cual es llenado con los datos de la base de datos para luego retornarlo. |
| Nombre | UpdateServer(String name, String IP, Boolean replica, Integer código) |
| Descripción | Este método se conecta a la base de datos, modifica los datos del servidor y retorna: verdadero si los modificó correctamente, falso si no lo pudo modificar. |

| | |
|-----------------------------------|---|
| Nombre: ServerFacade | |
| Tipo de clase: | |
| Para cada responsabilidad: | |
| Nombre | InsertServer(String , String , Boolean , Integer) |
| Descripción | Este es un método abstracto que se implementa en la clase ServerFacadeImpl. |
| Nombre | DeleteServer(String) |
| Descripción | Este es un método abstracto que se implementa en la clase ServerFacadeImpl. |
| Nombre | ShowServer() |
| Descripción | Este es un método abstracto que se implementa en la clase ServerFacadeImpl. |
| Nombre | UpdateServer(String , String , Boolean , Integer) |
| Descripción | |

| | |
|-----------------------------------|--|
| Nombre: ServerFacadeImpl | |
| Tipo de clase: | |
| Para cada responsabilidad: | |
| Nombre | InsertServer(String name , String IP, Boolean replica, Integer código) |
| Descripción | Este método es el encargado de llamar al método InsertServer () que se encuentra en la Interfaz ServerDao. |

| | |
|--------------------|--|
| Nombre | DeleteServer(String name) |
| Descripción | Este método es el encargado de llamar al método DeleteServer () que se encuentra en la Interfaz ServerDao. |
| Nombre | ShowServer() |
| Descripción | Este método es el encargado de llamar al método ShowServer () que se encuentra en la Interfaz ServerDao. |
| Nombre | UpdateServer(String name , String IP , Boolean replica, Integer código) |
| Descripción | Este método es el encargado de llamar al método UpdateServer () que se encuentra en la Interfaz ServerDao. |

| | |
|---------------------------------------|---|
| Nombre: ServerDeleteController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| serverFacade | ServerFacade |
| Para cada responsabilidad: | |
| Nombre | handleRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método DeleteServer () de la fachada TablespaceFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|---------------------------------------|---|
| Nombre: ServerInsertController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| serverFacade | ServerFacade |
| Para cada responsabilidad: | |
| Nombre | handleRequest(HttpServletRequest request) |

| | |
|--------------------|---|
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertServer () de la fachada TablespaceFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |
|--------------------|---|

| | |
|-------------------------------------|---|
| Nombre: ServerShowController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| serverFacade | ServerFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | Este método llama al método ShowServer () de la fachada y obtiene una lista de servidores que son pasados al ModelAndView junto con la URL que envía hacia el jsp donde se muestran los servidores. |

| | |
|-------------------------------------|--|
| Nombre: ServerShowController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| serverFacade | ServerFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método UpdateServer de la fachada ServerFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|-------------------------------|-------------|
| Nombre: Tabla_Réplica | |
| Tipo de clase: Entidad | |
| Atributo | Tipo |
| table_name | String |
| regla | String |
| insert | Boolean |
| update | Boolean |
| delete | Boolean |
| Orden_Insert | Boolean |
| Notify | Boolean |
| Fiel_Identification | String |

| | |
|----------------------------------|--------------------------|
| Nombre: Tabla_ReplicaList | |
| Tipo de clase: | |
| Atributo | Tipo |
| Tablas_Replica | ArrayList<Tabla_Replica> |

| | |
|-----------------------------------|---|
| Nombre: Tabla_ReplicaDao | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | InsertTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tabla_ReplicaDaoImpl. |
| Nombre | UpdateTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tabla_ReplicaDaoImpl. |
| Nombre | DeleteTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tabla_ReplicaDaoImpl. |
| Nombre | ShowTablas_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tabla_ReplicaDaoImpl. |

| | |
|-------------------------------------|--|
| Nombre: Tabla_ReplicaDaolmpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| connection | ConecctionManager |
| Para cada responsabilidad: | |
| Nombre | InsertTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método se conecta a la base de datos, inserta los datos de Tabla_Replica y retorna verdadero si los insertó correctamente o falso si no los pudo insertar. |
| Nombre | UpdateTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método se conecta a la base de datos, modifica los datos de Tabla_Replica y retorna: verdadero si los modificó correctamente, falso si no lo pudo modificar. |
| Nombre | DeleteTabla_Replica(String table_name) |
| Descripción | Este método se conecta a la base de datos y elimina la tabla cuyo nombre es pasado como parámetro. |
| Nombre | ShowTabla_Replica() |
| Descripción | Recupera de la base de datos de todas las tablas que replican, crea un objeto de tipo Tabla_ReplicaList, el cual es llenado con los datos de la base de datos para luego retornarlo. |

| | |
|-------------------------------------|--|
| Nombre: Tablas_ReplicaFacade | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | InsertTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tablas_ReplicaFacadeImpl |

| | |
|--------------------|--|
| Nombre | UpdateTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tablas_ReplicaFacadeImpl |
| Nombre | DeleteTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tablas_ReplicaFacadeImpl |
| Nombre | ShowTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase Tablas_ReplicaFacadeImpl |

| | |
|---|--|
| Nombre: Tablas_ReplicaFacadeImpl | |
| Tipo de clase: Interface | |
| Atributo | Tipo |
| tabla_replicaDao | Tabla_ReplicaDao |
| Para cada responsabilidad: | |
| Nombre | InsertTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método es el encargado de llamar al método I InsertTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | UpdateTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método es el encargado de llamar al método I UpdateTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | DeleteTabla_Replica(String table_name) |
| Descripción | Este método es el encargado de llamar al método I DeleteTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | ShowTabla_Replica() |
| Descripción | Este método es el encargado de llamar al método I ShowTabla_Replica () que se |

| | |
|--|--|
| | encuentra en la Interfaz Tabla_ReplicaDao. |
|--|--|

| | |
|--|---|
| Nombre: Tabla_ReplicaDeleteController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| tablas_ReplicaFacade | Tablas_ReplicaFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método Tabla_ReplicaDelete () de la fachada Tabla_ReplicaFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|--|---|
| Nombre: Tabla_ReplicaCreateController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| tablas_ReplicaFacade | Tablas_ReplicaFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método Tabla_ReplicaCreate () de la fachada Tabla_ReplicaFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|--|--|
| Nombre: Tabla_ReplicaUpdateController | |
| Tipo de clase: Controladora | |

| Atributo | Tipo |
|-----------------------------------|---|
| tablas_ReplicaFacade | Tablas_ReplicaFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método Tabla_ReplicaUpdate () de la fachada Tabla_ReplicaFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| Nombre: Tabla_ReplicaShowController | |
|--|---|
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| tablas_ReplicaFacade | Tablas_ReplicaFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método Tabla_ReplicaShow () de la fachada Tabla_ReplicaFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| Nombre: ConfigurarReplicaEPFacadeImpl | |
|--|-----------------|
| Tipo de clase: | |
| Atributo | Tipo |
| tablareplicaDao | TablaReplicaDao |
| tablespaceDao | TablespaceDao |
| Para cada responsabilidad: | |

| | |
|--------------------|--|
| Nombre | InsertTablespace(String name, String datafile1, String datafile2, Integer tamañodatafile1, Integer tamañodatafile2) |
| Descripción | Este método es el encargado de llamar al método InsertTablespace que se encuentra en la Interfaz TablespaceDao. |
| Nombre | InsertTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método es el encargado de llamar al método InsertTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | UpdateTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método es el encargado de llamar al método UpdateTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | DeleteTabla_Replica(String table_name) |
| Descripción | Este método es el encargado de llamar al método DeleteTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | ShowTablas_Replica() |
| Descripción | Este método es el encargado de llamar al método ShowTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |

| | |
|--|--|
| Nombre: ConfigurarReplicaEPFacade | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | InsertTablespace(String, String, String, Integer, Integer) |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaEPFacadeImpl |
| Nombre | InsertTabla_Replica(String, String, Boolean, Boolean, Boolean, Boolean, Boolean, String) |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaEPFacadeImpl |

| | |
|--------------------|--|
| Nombre | UpdateTabla_Replica(String, String, Boolean, Boolean, Boolean, Boolean, Boolean, String) |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaEPFacadeImpl |
| Nombre | DeleteTabla_Replica(String) |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaEPFacadeImpl |
| Nombre | ShowTablas_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaEPFacadeImpl |

| | |
|--|--|
| Nombre: UsersReplicaLoginEPController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| configurarReplicaEPFacade | ConfigurarReplicaEPFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método VerifyUser () de la fachada Tabla_ReplicaFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|--|--|
| Nombre: ParametrosReplicaEPController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| configurarReplicaEPFacade | ConfigurarReplicaEPFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |

| | |
|--------------------|---|
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método ValidarParametros () de la clase ConfigurarReplicaFacadeImpl, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |
|--------------------|---|

| | |
|--|---|
| Nombre: CreateTablespaceReplicaEPController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| configurarReplicaEPFacade | ConfigurarReplicaEPFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertTablespace () de la interfaz ConfigurraReplicaEPFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|--|--|
| Nombre: TablasReplicaEPShowController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| configurarReplicaEPFacade | ConfigurarReplicaEPFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método ShowTablas_Replica () de la interfaz ConfigurraReplicaEPFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa |

| | |
|--|--|
| | como parámetro al objeto ModelAndView que es el que se retorna en el método. |
|--|--|

| | |
|---|--|
| Nombre: AddTablasReplicaEPController | |
| Tipo de clase: Controladora | |
| Atributo | Tipo |
| configurarReplicaEPFacade | ConfigurarReplicaEPFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertTabla_Replica () de la interfaz ConfigurraReplicaEPFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|---|---|
| Nombre: ConfigurarReplicaCDFFacade | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | InsertServer(String , String , Boolean , Integer) |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaCDFFacadeImpl. |
| Nombre | DeleteServer(String) |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaCDFFacadeImpl |
| Nombre | ShowServer() |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaCDFFacadeImpl. |
| Nombre | UpdateServer(String , String , Boolean , Integer) |
| Descripción | Este es un método abstracto que se implementa en la clase |

| | |
|--------------------|--|
| | ConfigurarReplicaCDFacadeImpl. |
| Nombre | InsertTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaCDFacadeImpl. |
| Nombre | UpdateTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaCDFacadeImpl. |
| Nombre | DeleteTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaCDFacadeImpl. |
| Nombre | ShowTabla_Replica() |
| Descripción | Este es un método abstracto que se implementa en la clase ConfigurarReplicaCDFacadeImpl. |
| Nombre | InsertTablespace(String, String, String, Integer, Integer) |
| Descripción | Este es un método abstracto que es implementado en la clase ConfigurarReplicaCDFacadeImpl. |

| | |
|---|--|
| Nombre: ConfigurarReplicaCDFacadeImpl. | |
| Tipo de clase: | |
| Atributo | Tipo |
| serverDao | ServerDao |
| tablareplicaDao | TablareplicaDao |
| tablespaceDao | TablespaceDao |
| Para cada responsabilidad: | |
| Nombre | InsertServer(String name , String IP, Boolean replica, Integer código) |
| Descripción | Este método es el encargado de llamar al método InsertServer () que se encuentra en la Interfaz ServerDao. |
| Nombre | DeleteServer(String name) |

| | |
|--------------------|--|
| Descripción | Este método es el encargado de llamar al método DeleteServer () que se encuentra en la Interfaz ServerDao. |
| Nombre | ShowServer() |
| Descripción | Este método es el encargado de llamar al método ShowServer () que se encuentra en la Interfaz ServerDao. |
| Nombre | UpdateServer(String name , String IP , Boolean replica, Integer código) |
| Descripción | Este método es el encargado de llamar al método UpdateServer () que se encuentra en la Interfaz ServerDao. |
| Nombre | InsertTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método es el encargado de llamar al método InsertTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | UpdateTabla_Replica(String table_name, String regla, Boolean insert, Boolean update, Boolean delete, Boolean ordenInsert, Boolean Notify, String Fiel_Identification) |
| Descripción | Este método es el encargado de llamar al método UpdateTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | DeleteTabla_Replica(String table_name) |
| Descripción | Este método es el encargado de llamar al método DeleteTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | ShowTablas_Replica() |
| Descripción | Este método es el encargado de llamar al método ShowTabla_Replica () que se encuentra en la Interfaz Tabla_ReplicaDao. |
| Nombre | InsertTablespace(String name, String datafile1, String datafile2, Integer tamañodatafile1, Integer tamañodatafile2) |
| Descripción | Este método es el encargado de llamar al método InsertTablespace que se encuentra en la Interfaz TablespaceDao. |

| | |
|---|---|
| Nombre: UsersReplicaLogginCDController | |
| Tipo de clase: | |
| Atributo | Tipo |
| configurarReplicaCDFacade | ConfigurarReplicaCDFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método VerifyUser () de la interfaz ConfigurraReplicaCDFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|--|--|
| Nombre: ParametrosReplicaCDController | |
| Tipo de clase: | |
| Atributo | Tipo |
| configurarReplicaCDFacade | ConfigurarReplicaCDFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método ParametrosReplica () de la interfaz ConfigurraReplicaCDFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|--|---------------------------|
| Nombre: CreateTablespaceReplicaCDController | |
| Tipo de clase: | |
| Atributo | Tipo |
| configurarReplicaCDFacade | ConfigurarReplicaCDFacade |

| | |
|-----------------------------------|---|
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertTablespace () de la interfaz ConfigurraReplicaCDFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|---|---|
| Nombre: AddServerReplicaCDController | |
| Tipo de clase: | |
| Atributo | Tipo |
| configurarReplicaCDFacade | ConfigurarReplicaCDFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertServer () de la interfaz ConfigurraReplicaCDFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|--|--|
| Nombre: ServerReplicaCDShowController | |
| Tipo de clase: | |
| Atributo | Tipo |
| configurarReplicaCDFacade | ConfigurarReplicaCDFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método |

| | |
|--|--|
| | ShowServer () de la interfaz ConfigurraReplicaCDFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |
|--|--|

| | |
|---|---|
| Nombre: TablaReplicaCDShowController | |
| Tipo de clase: | |
| Atributo | Tipo |
| configurarReplicaCDFacade | ConfigurarReplicaCDFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método ShowTablas_Replica () de la interfaz ConfigurraReplicaCDFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|---|--|
| Nombre: AddTablasReplicaCDController | |
| Tipo de clase: | |
| Atributo | Tipo |
| configurarReplicaCDFacade | ConfigurarReplicaCDFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de obtener del jsp los datos enviados a través del formulario en el objeto request, los datos se le pasan como parámetro al método InsertTabla_Replica () de la interfaz ConfigurraReplicaCDFacade, este método retorna un valor boolean el cual se utiliza para escoger que URL que se le pasa como parámetro al objeto ModelAndView que es el que se retorna en el método. |

| | |
|---|---|
| Nombre: MostrarTransaccionesCDController | |
| Tipo de clase: | |
| Atributo | Tipo |
| monitorearTransaccionesFacade | MonitorearTransaccionesFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de llamar al método ObtenerGrafico() de la interfaz MonitorearTransaccionesFacade, este método es el encargado cargar la imagen que contiene las transacciones que vienen entrando desde los demás centros penitenciarios al centro de datos, para después retornar en el ModelAndView la vista correspondiente al jsp monitor.jsp. |

| | |
|--|--|
| Nombre: MostrarTransaccionesTabuladamenteCDController | |
| Tipo de clase: | |
| Atributo | Tipo |
| monitorearTransaccionesFacade | MonitorearTransaccionesFacade |
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServlet request) |
| Descripción | El método es el encargado de llamar al método ObtenerTabla() de la interfaz MonitorearTransaccionesFacade, este método es el encargado recuperar todas las transacciones que vienen entrando desde los demás centros penitenciarios al centro de datos, para después retornar en el ModelAndView la vista correspondiente al jsp tablatransaccion.jsp. |

| | |
|---|-------------------------------|
| Nombre: ShowOperacionTransaccionController | |
| Tipo de clase: | |
| Atributo | Tipo |
| monitorearTransaccionesFacade | MonitorearTransaccionesFacade |

| | |
|-----------------------------------|---|
| Para cada responsabilidad: | |
| Nombre | handeRequest(HttpServletRequest request) |
| Descripción | El método es el encargado de llamar al método ObtenerOperacion () de la interfaz MonitorearTransaccionesFacade, este método es el encargado recuperar todas las operaciones de una transacción específica seleccionada por el usuario, para después retornar en el ModelAndView la vista correspondiente al jsp tablaoperaciones.jsp. |

| | |
|--|--|
| Nombre: MonitorearTransaccionesFacade | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | ObtenerGrafico() |
| Descripción | Este es un método abstracto que se implementa en la clase MonitorearTransaccionesFacadeImpl. |
| Nombre | ObtenerTabla() |
| Descripción | Este es un método abstracto que se implementa en la clase MonitorearTransaccionesFacadeImpl |
| Nombre | ObtenerOperacion() |
| Descripción | Este es un método abstracto que se implementa en la clase MonitorearTransaccionesFacadeImpl. |

| | |
|--|--|
| Nombre: MonitorearTransaccionesFacadeImpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| graficoManagerFacade | GraficoManagerFacade |
| Para cada responsabilidad: | |
| Nombre | ObtenerGrafico() |
| Descripción | Este es un método se encarga de llamar ala método ObtenerGrafico () que se |

| | |
|--------------------|---|
| | encuentra en la interfaz GraficoManagerFacade. |
| Nombre | ObtenerTabla() |
| Descripción | Este es un método se encarga de llamar ala método ObtenerTabla() que se encuentra en la interfaz GraficoManagerFacade |
| Nombre | ObtenerOperacion() |
| Descripción | Este es un método se encarga de llamar ala método ObtenerOperacion () que se encuentra en la interfaz GraficoManagerFacade. |

| | |
|-------------------------------------|---|
| Nombre: GraficoManagerFacade | |
| Tipo de clase: Interface | |
| Para cada responsabilidad: | |
| Nombre | ObtenerGrafico() |
| Descripción | Este es un método abstracto que se implementa en la clase GraficoManagerFacadeImpl. |
| Nombre | ObtenerTabla() |
| Descripción | Este es un método abstracto que se implementa en la clase GraficoManagerFacadeImpl |
| Nombre | ObtenerOperacion() |
| Descripción | Este es un método abstracto que se implementa en la clase GraficoManagerFacadeImpl. |

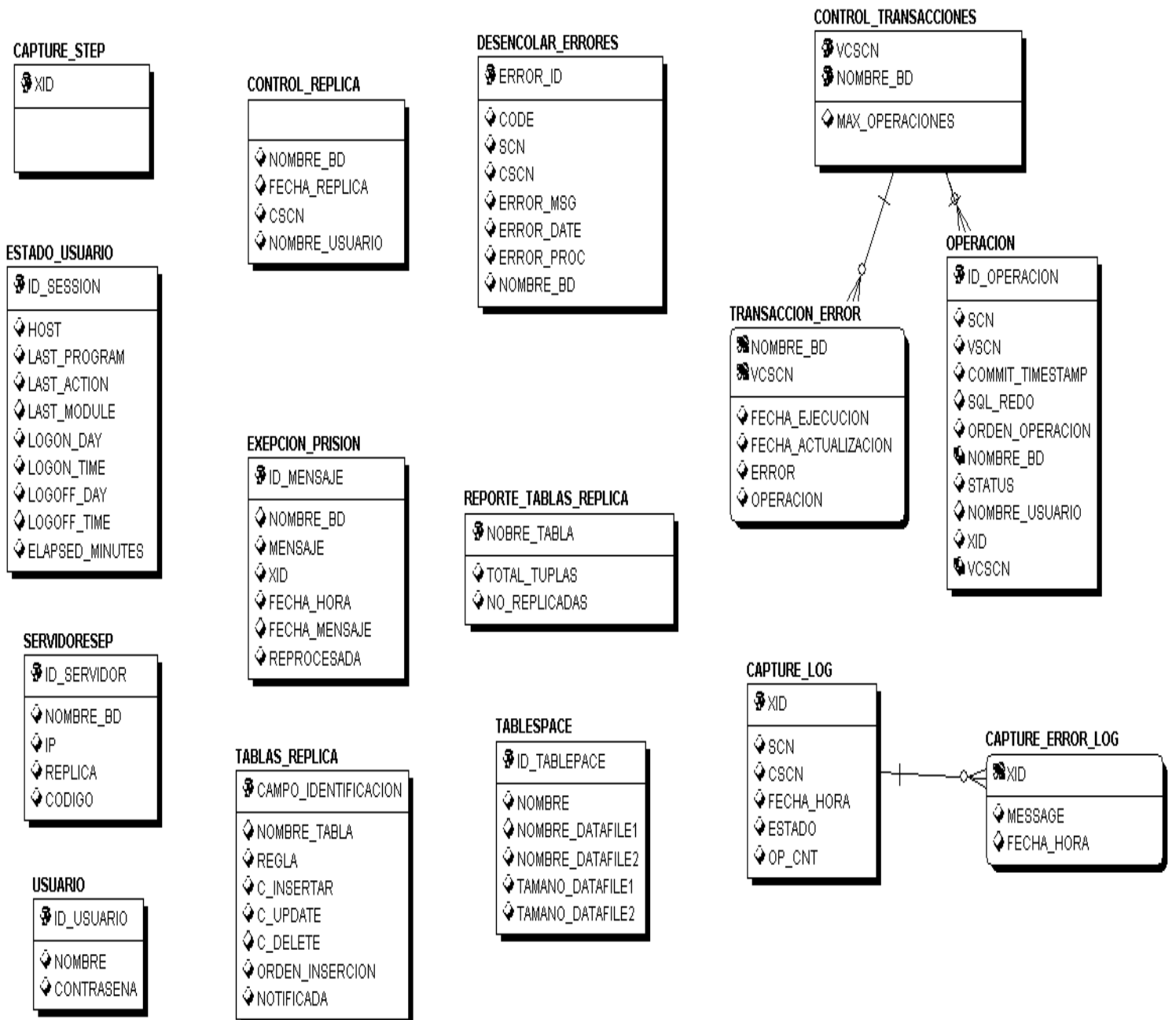
| | |
|---|--|
| Nombre: GraficoManagerFacadeImpl | |
| Tipo de clase: | |
| Atributo | Tipo |
| connection | ConecctionManager |
| Para cada responsabilidad: | |
| Nombre | ObtenerGrafico() |
| Descripción | Este es un método se encarga de conectarse a la base de datos y generar una lista de |

| | |
|--------------------|--|
| | imágenes que son almacenadas un objeto de tipo ListJFreeChart el cual se retorna para luego obtener las imágenes con el reporte. |
| Nombre | ObtenerTabla() |
| Descripción | Este es un método se encarga de conectarse a la base de datos y recuperar todas las transacciones que están guardadas en la tabla CONTROL _ TRANSACCIONES y retornarlas en un ArrayList. |
| Nombre | ObtenerOperacion() |
| Descripción | Este es un método se encarga de conectarse a la base de datos y recuperar todas las operaciones de una transacción específica estas operaciones están guardadas en la tabla OPERACION y retornarlas en un ArrayList. |

| | |
|-------------------------------|----------------|
| Nombre: ListJFreeChart | |
| Tipo de clase: | |
| Atributo | Tipo |
| listaJFreeChart | ListJFreeChart |

3.2.4 Diseño de la Base de Datos

DIAGRAMA ENTIDAD RELACION



3.2.5 Descripción de las Tablas de la Base de Datos

| Nombre: Usuario | | |
|--|-------------|---|
| Descripción: En esta tabla se guarda la información de los usuarios del sistema: sus nombres y sus contraseñas. | | |
| Atributo | Tipo | Descripción |
| Id_usuario | Varchar(20) | Este es el campo de identificación de los usuarios, la llave primaria de la tabla |
| nombre | Varchar(30) | Guarda el nombre del usuario |
| contrasena | Varchar(30) | Guarda la contraseña del usuario |

| Nombre: Tablespace | | |
|--|---------------|--|
| Descripción: En esta tabla se guarda la información de los tablespace de la réplica | | |
| Atributo | Tipo | Descripción |
| Id_tablespace | Varchar(20) | Este es el campo de identificación del tablepace, la llave primaria de la tabla. |
| nombre | Varchar(30) | Guarda el nombre del tablepace |
| nombre_datafile1 | Varchar(30) | Guarda el nombre del primer datafile asociado al tablepace. |
| nombre_datafile2 | Varchar(30) | Guarda el nombre del segundo datafile asociado al tablepace. |
| tamano_datafile1 | Numeric(10,0) | Guarda el tamaño del primer datafile del tablespace. |
| tamano_datafile1 | Numeric(10,0) | Guarda el tamaño del primer datafile del tablespace. |

| Nombre: ServidoresEP | | |
|--|---------------|--|
| Descripción: En esta tabla se guarda la información de los servidores autorizados a replicar. | | |
| Atributo | Tipo | Descripción |
| Id_servidor | Varchar(20) | Este es el campo de identificación del servidor, la llave primaria de la tabla. |
| Nombre_bd | Varchar(50) | Guarda el nombre de la base de datos instanciada en el servidor |
| IP | Varchar(15) | Guarda el numero IP del servidor. |
| Replica | boolean | Guarda si el servidor va a replicar o no |
| código | Numeric(10,0) | Guarda el código que identifica a la prisión en la que se encuentra el servidor. |

| Nombre: Tablas_Replica | | |
|--|--------------|--|
| Descripción: En esta tabla se guarda toda la información de las tablas que van a replicar | | |
| Atributo | Tipo | Descripción |
| nombre_tabla | Varchar(30) | Este es el campo que guarda el nombre de la tabla que va a replicar. |
| regla | Varchar(50) | Campo utilizado para definir que reglas se pueden aplicar en la replica de Stream. |
| c_insert | Numeric(1,0) | Guarda 1 si la tabla va a realizar operación de inserción, o si no va a realizar la operación. |
| c_update | Numeric(1,0) | Guarda 1 si la tabla va a realizar operación de actualización, o si no va a realizar la operación. |
| c_delete | Numeric(1,0) | Guarda 1 si la tabla va a realizar operación de eliminar, o si no va a realizar la operación. |
| orden _ inserción | Numeric | Este campo almacena el orden en que se va a |

| | | |
|------------------------|--------------|--|
| | | realizar la inserción de datos en las tablas. |
| notificada | Numeric(1,0) | Especifica que tablas van a ser notificadas en la configuración inicial de la réplica. |
| campo _ identificación | Varchar(50) | Guarda un campo único que identifica cada tabla. |

Nombre: Control_Transacciones

Descripción: Esta es una tabla de trabajo temporal que guarda de que servidor viene la transacción, el numero de identificación y el máximo numero de operaciones que trae la transacción.

| Atributo | Tipo | Descripción |
|-----------------|-------------|--|
| VCSCN | Varchar(20) | Numero de identificación de la transacción, es llave primaria |
| Nombre_BD | Varchar(50) | Nombre de la base de datos de la cual viene la transacción, es llave primaria. |
| Max_Operaciones | Numeric() | Máximo numero de operaciones que trae la transacción. |

Nombre: Control_Réplica

Descripción: En esta tabla se guarda el código de identificación de las transacciones, el servidor del que viene, la fecha en que se realizo la replicación y el usuario que la realizó.

| Atributo | Tipo | Descripción |
|----------------|-------------|--|
| Nombre_BD | Varchar(50) | Nombre de la base de datos de la cual viene la transacción |
| Fecha_Replica | Date | Fecha en que se realizó la replicación |
| CSCN | Numeric() | Identificador de la transacción |
| Nombre_Usuario | Varchar(30) | Nombre del usuario que realizó la réplica. |

Nombre: Excepción_Prisiones

Descripción: Guarda todas las excepciones que ocurren en las prisiones cuando se realiza el

| proceso de replicación. | | |
|-------------------------|---------------|--|
| Atributo | Tipo | Descripción |
| Id_Mensaje | Raw(18) | Identificador único del mensaje, llave primaria |
| Mensaje | Varchar(4000) | Mensaje de error que se genero en la prisión. |
| XID | Raw(18) | Numero interno de la transacción. |
| Fecha_Hora | Date | Fecha y hora a la que la excepción llevo al centro de datos. |
| Fecha_Mensaje | Date | Fecha y hora a la que dio la excepción. |
| Reprocesada | Numeric(1,0) | Si la excepción fue reprocesada o no. |

| Nombre: Operación | | |
|---|---------------|--|
| Descripción: Guarda todas las operaciones de las transacciones que vienen llegando al centro de datos. | | |
| Atributo | Tipo | Descripción |
| Id _ operación | Raw(18) | Identificador único de la operación, llave primaria. |
| SCN | Numeric() | SCN (System Change Number) Numero unívoco que se le asigna a una transacción cuando es confirmada. |
| CSCN | Numeric() | |
| Commit_timestamp | Date | Fecha en que se le dio commit a la operación en la prisión. |
| SQL_Redo | Varchar(4000) | Almacena la sentencia SQL de la operación. |
| Orden_Operacion | Numeric() | Orden de la ejecución de las operaciones |
| Nombre_DB | Varchar(50) | Nombre de la base de datos que viene la operación. |
| Status | Numeric(1,0) | Bandera interna utilizada por uno de los procedimientos. |

| | | |
|----------------|-------------|---|
| Nombre_Usuario | Varchar(30) | Nombre del usuario que generó la transacción. |
| XID | RAW(18) | Identificador de la transacción. |

| Nombre: Transacción_Error | | |
|---|---------------|---|
| Descripción: sirve para registrar el error cuando existe un conflicto de las transacciones | | |
| Atributo | Tipo | Descripción |
| Nombre_BD | Varchar(50) | Nombre de la base de datos que viene el error de la transacción. |
| VCSCN | Numeric() | Numero de identificación de la transacción, es llave primaria |
| Fecha _ ejecución | Date | Fecha en la que se trato de ejecutar la transacción por primera vez en el centro de datos |
| Fecha _ actualización | Date | Fecha de la última vez que se trató de ejecutar. |
| Error | Varchar(4000) | El error que lanzó la transacción. |
| Operación | Numeric() | Numero de la operación dentro de la transacción que dio el error. |

| Nombre: Desencolar_Errores | | |
|---|---------------|---|
| Descripción: Almacena todos los errores que ocurren a la de desencolar los mensajes. | | |
| Atributo | Tipo | Descripción |
| Error_id | Raw(18) | Identificador del error, llave primaria |
| Code | Varchar(4000) | Código del error |
| Error_msg | Varchar(4000) | Mensaje de error |
| SCN | Numeric() | SCN (System Change Number) Número unívoco que se le asigna a una operación dentro de la transacción cuando es confirmada. |

| | | |
|------------|-------------|--|
| CSCN | Numeric() | CSCN (Current System Change Number) Número unívoco que se le asigna a una transacción cuando es confirmada. |
| Error_Date | Date | Fecha del error |
| Error_Proc | Numeric() | |
| Nombre_BD | Varchar(50) | Nombre de la base de datos que viene el error. |

| Nombre: Capture_Log | | |
|--|--------------|--|
| Descripción: Cuando una tabla en las prisión se notifica, se guarda en esta tabla los datos asociados a la operación que origino la notificación. | | |
| Atributo | Tipo | Descripción |
| XID | Raw(18) | Identificador de la transacción. |
| SCN | Numeric() | SCN (System Change Number) Numero unívoco que se le asigna a una transacción cuando es confirmada. |
| Fecha_Hora | Date | Fecha en que ocurrió el cambio en la tabla |
| Estado | Numeric(1.0) | Estado en que se encuentra la notificación: 0: Notificada 1: Datos Extraídos del Flashback y enviada a la cola de capture. 2: Extraída de los redologs y en proceso de filtrado 3: Propagada y completada. |
| OP_CNT | Numeric() | Cantidad de operaciones. |

| Nombre: Capture_Error_Log | | |
|---|-------------|----------------------------------|
| Descripción: Esta tabla guarda los errores que se registran en la captura. | | |
| Atributo | Tipo | Descripción |
| XID | Raw(18) | Identificador de la transacción. |

| | | |
|------------|---------------|---------------------------------------|
| Message | Varchar(4000) | Mensaje del error. |
| Fecha_Hora | date | Fecha y hora en que ocurrió el error. |

| Nombre: Reporte_Tablas_Réplica | | |
|---|-------------|---|
| Descripción: En esta tabla se guarda un reporte de cómo ocurrió la réplica | | |
| Atributo | Tipo | Descripción |
| Nombre_Tabla | Varchar(30) | Nombre de la tabla que replicó |
| Total_Tuplas | Numeric() | Total de campos que tiene la tabla |
| No_Replicadas | Numeric() | Total de campos de la tabla que no replicaron |

| Nombre: Estado_Usuario | | |
|--|-------------|--|
| Descripción: Esta tabla guarda todos los datos de un usuario que se conecta a la aplicación. Para cada vez que se conecta un usuario, un trigger llena la tabla con los parámetros. | | |
| Atributo | Tipo | Descripción |
| Session_Id | Varchar(30) | Identificador de la sesión por la que se conectó el usuario. |
| Id_Usuario | Varchar(20) | Identificador del usuario que se conectó. |
| Host | Varchar(30) | IP de la PC que se conecto a la aplicación. |
| Last_Program | Varchar(50) | Ultimo programa que corrió el usuario. |
| Last_Action | Varchar(30) | Ultima acción que realizo el usuario. |
| Last_Module | Varchar(50) | |
| Logon_Day | Date | Fecha en que se conectó el usuario. |
| Logon_Time | Varchar(20) | Hora a la que se conectó el usuario. |
| Logoff_Day | Date | Fecha en la que se desconectó el usuario |
| Logoff_Time | Varchar(20) | Hora a la que se desconecto el usuario. |
| Elapsed_Minutes | Numeric() | |

CONCLUSIONES

Después de haber evaluado el beneficio económico que le reportaba a la UCI, implementar un algoritmo alternativo poder realizar replicar datos con Oracle, se tomo la dedición de realizar el análisis y el diseño de un sistema que permitiera administrar y monitorear la replica de datos. En el presente trabajo se lograron todos los objetivos propuestos.

RECOMENDACIONES

Se recomienda que se realice la implementación del sistema aquí propuesto, así como la implantación del mismo en cada uno de los proyectos de la UCI que realicen réplica de datos con Oracle 10g. Además que se extienda la aplicación a otros Sistemas Gestores de Bases de Datos para que la UCI la pueda utilizar con fines comerciales.

BIBLIOGRAFIA

1. Urbano, R., *Oracle Database Advanced Replication 10g Release 2 (10.2)*. Primera Edición ed. June 2005. 246.
2. Urbano, R., *Oracle Streams Concepts and Administration, 10g Release 2 (10.2)*. Junio 2006. 614.
3. Vince Duggan, B.D. (2000) *InterBase Replication: Fundamentals*.
4. Herrera, A. (2005) *PostgreSQL Aplicaciones Avanzadas*.

ANEXO 1:

| Actores | Justificación |
|------------------|--|
| administrador | Es un actor general, que generaliza los procesos que son comunes para el centro de datos y los establecimientos penitenciarios. |
| Administrador CD | Es el encargado de realizar todos los procesos del centro de datos: Configurar la base de la réplica, gestionar los servidores que van a replicar, monitorear las transacciones que llegan provenientes de los establecimientos penitenciarios y realizar la corrección de conflictos. |
| Administrador EP | Es el encargado de realizar todas las tareas de los proceso de los establecimientos penitenciarios: Configurar la base de la réplica. |

| | |
|--------------------|---|
| CU-1 | Configurar Base Réplica EP |
| Actor | Administrador EP |
| Descripción | Este caso de uso es el que crea la base para realizar el proceso de réplica, en el se crea el vinculo con el centro de datos, se especifican los parámetros de almacenamiento para los establecimientos penitenciarios, y los objetos necesarios para el proceso. |
| Referencia | RF 2.1, 2.2 |

| | |
|--------------------|---|
| CU-2 | Configurar Base Réplica CD |
| Actor | Administrador CD |
| Descripción | El caso de uso es el que prepara el ambiente para realizar el proceso de réplica en el centro de datos, en el se especifican todos los parámetros de almacenamiento y se especifican los tablespaces. |
| Referencia | RF 1.1, 1.2, 1.3, 1.4 |

| | |
|--------------------|---|
| CU-3 | Gestionar Servidores Replicación |
| Actor | Administrador CD |
| Descripción | Es aquí donde se adicionan, modifican o eliminan los datos de los servidores que se van a autorizar a replicar. |
| Referencia | RF 6.4, 6.5, 6.6, 6.7 |

| | |
|--------------------|--|
| CU-4 | Gestionar Tablas Replicación |
| Actor | Administrador |
| Descripción | Es aquí donde el administrador selecciona las tablas que el desea que repliquen, especifica las operaciones que va a realizar cada tabla, la regla por la que va a replicar. |
| Referencia | RF 6.1, 6.2, 6.3 |

| | |
|--------------------|---|
| CU-5 | Monitorear Transacciones CD |
| Actor | Administrador CD |
| Descripción | Mostrarle al administrador del centro de datos una grafica de las transacciones que van llegando al centro de datos para que el pueda monitorearlas en tiempo real. |
| Referencia | 4.1, 4.2, 4.3, 4.4 |

| | |
|--------------------|--|
| CU-6 | Autenticar Usuario |
| Actor | Administrador |
| Descripción | El propósito general de este caso es autenticar los usuarios que se conecten a la aplicación para poder definir los niveles de acceso de cada uno y establecer los permisos correspondientes al usuario. |
| Referencia | 8.1, 8.2, 8.3 |

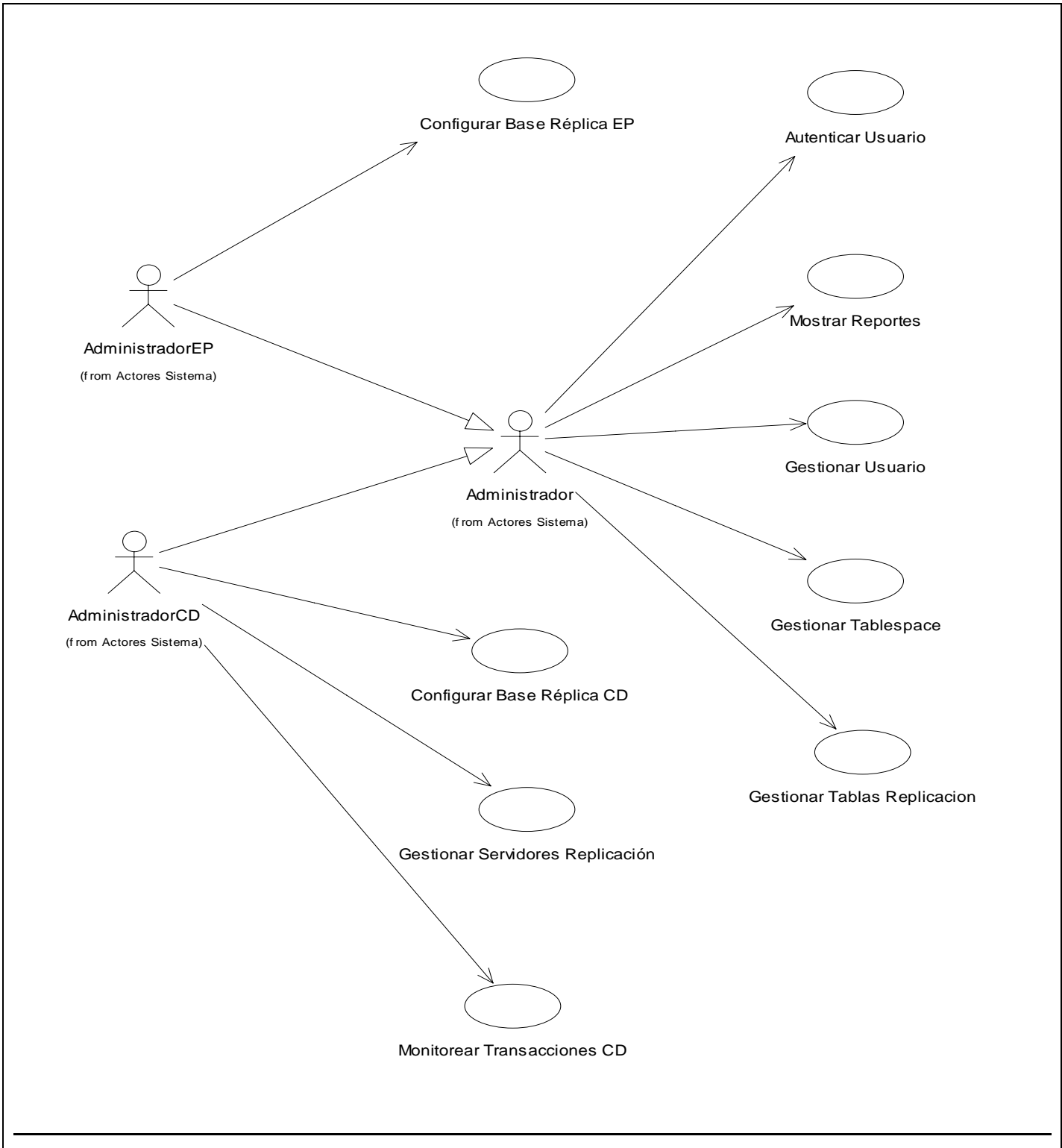
| | |
|--------------------|---|
| CU-7 | Mostrar Reportes |
| Actor | Administrador |
| Descripción | El propósito de este caso de uso es permitirles a los administradores ver una serie de reportes del estado de la réplica, los conflictos ocurridos, las bases de datos que mas conflictos presentan para la replicación, entre otros aspectos importantes para la administración de la réplica. |
| Referencia | 9.1, 9.2, 9.3, 9.4, 9.5 |

| | |
|--------------------|---|
| CU-8 | Gestionar Usuario |
| Actor | Administrador |
| Descripción | El propósito de este caso es administrar todos los usuarios que se conecten a la réplica, crear, modificar y eliminar usuarios de la réplica. |
| Referencia | 5.1, 5.2, 5.3 |

| | |
|--------------------|---|
| CU-9 | Gestionar Tablespace |
| Actor | Administrador |
| Descripción | El propósito de este caso es administrar los tablespace para la réplica. En este se crean, modifican o eliminan los tablespace de la réplica. |

| | |
|-------------------|---------------|
| Referencia | 7.1, 7.2, 7.3 |
|-------------------|---------------|

Diagrama de Casos de Usos



Casos de uso expandidos

| Caso de uso | |
|---|---|
| CU-1 | Configurar Base Réplica EP |
| Propósito | El Objetivo de este caso de uso es preparar el ambiente el establecimiento penitenciario para que se pueda realizar la réplica hacia el CD. |
| Actores: Administrador EP | |
| Resumen: Este caso de uso es el que crea la base para realizar el proceso de réplica, en el se crea el vínculo con el centro de datos, se especifican los parámetros de almacenamiento para los establecimientos penitenciarios, y los objetos necesarios para el proceso. | |
| Referencias | RF 2.1, 2.2 |
| Acción del actor | Respuesta del sistema |
| 1- Acciona la opción "Configurar Réplica EP" | 1.1-El sistema muestra la interfaz autenticación para que el usuario especifique usuario sys y su contraseña y un usuario de la aplicación con su contraseña. |
| 2- Teclea los usuarios con sus contraseñas correspondientes y acciona el botón "siguiente". | 2.1- Comprueba que los usuarios y las contraseñas sean válidos. |
| | 3- Muestra la Interfaz para especificar los parámetros de iniciación, donde el usuario debe introducir parámetros de almacenamiento y envío. |
| 4- Teclea los parámetros de iniciación y pulsa el botón "siguiente". | 4.1- Muestra una interfaz para especificar el tablespace para la réplica, donde el usuario debe introducir los datos del tablespace de la réplica. |
| 5- Teclea los datos del tablespace y pulsa el botón "siguiente". | 5.1- Muestra una interfaz para especificar las tablas de la réplica. 5.2 Muestra una interfaz para especificar las tablas de chequeo de la réplica. |

| | |
|---|--|
| Flujo alternativo | |
| Línea 4 | |
| Acción del actor | Respuesta del sistema |
| | 1- Muestra un mensaje de error informando que el usuario o la contraseña que ha especificado no son válidos. |
| 2- Introduce nuevamente el usuario y la contraseña | |
| Puntos de extensión. | |
| Línea 5.1 Muestra la interfaz para especificar las tablas de la réplica. Ver CU Gestionar Tablas Replicación. | |
| Línea 5.2 Muestra la interfaz para especificar las tablas de chequeo de la réplica. Ver CU Gestionar Tablas Chequeo Replicación. | |

| | |
|--|--|
| Caso de uso | |
| CU-2 | Configurar Base Réplica CD |
| Propósito | El Objetivo de este caso de uso es preparar el ambiente en el centro de datos para que se pueda realizar la replicación de datos entre dos bases de datos. |
| Actores: Administrador CD | |
| Resumen: El caso de uso es el que prepara el ambiente para realizar el proceso de réplica en el centro de datos, en el se especifican todos los parámetros de almacenamiento y se especifican los tablespace. | |
| Referencias | RF 1.1, 1.2, 1.3, 1.4 |
| Acción del actor | Respuesta del sistema |
| 1- Acciona la opción "Configurar Réplica CD" | 1.1- El sistema muestra una interfaz de autenticación donde se especifican los nombre de usuario y las contraseñas de sys y el usuario de la aplicación respectivamente. |
| 2- Teclea los usuarios con sus contraseñas | 2.1- Comprueba que los usuarios y las contraseñas |

| | |
|---|---|
| correspondientes y acciona el botón “siguiente”. | sean válidos. |
| | 3- Muestra una Interfaz donde el usuario especifica los parámetros de iniciación, donde se debe introducir los parámetros de almacenamiento y envío. |
| 4- Teclea los parámetros de iniciación y pulsa el botón “siguiente”. | 4.1- Muestra una interfaz para especificar el tablespace para la réplica, donde el usuario debe introducir Nombre del tablespace, nombre de los datafiles y tamaño de los mismos. |
| 5- Teclea los datos del tablespace y pulsa el botón “siguiente”. | 5.1- Muestra la interfaz para especificar los servidores. |
| | 5.2- Muestra una interfaz para especificar las tablas de la réplica. |
| Flujo alternativo | |
| Línea 4 | |
| Acción del actor | Respuesta del sistema |
| | 1- Muestra un mensaje de error informando que el usuario o la contraseña que ha especificado no son válidos. |
| 2- Introduce nuevamente el usuario y la contraseña | |
| Puntos de extensión. | |
| Línea 5.1 Muestra la interfaz para especificar los servidores. Ver CU Gestionar Servidores Replicación. | |
| Línea 5.2 Muestra una interfaz para especificar las tablas de la réplica. Ver CU Gestionar Tablas Replicación. | |

| Caso de uso | |
|---|--|
| CU-3 | Gestionar Servidores Réplica |
| Propósito | Permitir Adicionar, modificar o eliminar servidores para la réplica |
| Actores: Administrador CD | |
| Resumen: Es aquí donde se adicionan, modifican o eliminan los datos de los servidores que se van a autorizar a replicar. | |
| Referencias | RF 6.4, 6.5, 6.6, 6.7 |
| Acción del actor | Respuesta del sistema |
| 1- el actor del sistema necesita registrar, modificar o eliminar un servidor para la réplica. | 1.1 El sistema ejecuta alguna de las siguientes acciones: a) Si se decide registrar un servidor nuevo sección "Adicionar Servidor". b) Si se decide modificar un tabla ir a la sección "Modificar Servidor" c) Si se decide eliminar una tabla ir a la sección "Eliminar Servidor". |
| Sección "Adicionar Servidor" | |
| 2- El administrador selecciona la opción "Adicionar Servidor" | 2.1 El sistema muestra la interfaz que permite adicionar un servidor, donde el usuario debe introducir el nombre de la base de datos, el IP, especificar si el servidor réplica, y el código del establecimiento penitenciario. |
| 3- El administrador presiona el botón aceptar. | 3.1 El sistema guarda la información del servidor. 3.2 El sistema muestra un mensaje informando que los datos se guardaron satisfactoriamente. |
| Sección "Modificar Servidor" | |
| 4- El administrador selecciona la opción "Modificar Servidor" | 4.1 El sistema muestra un listado de todos los servidores. |
| 5- El administrador selecciona el servidor que | 5.1 El sistema muestra los datos del servidor |

| | |
|---|---|
| quiere modificar | seleccionado |
| 6- El administrador modifica los datos y presiona el botón "Aceptar" | 6.1 El sistema guarda los datos del servidor 6.2 El sistema muestra un mensaje informando que los datos se guardaron satisfactoriamente. |
| Sección "Eliminar Servidor" | |
| 7- El administrador selecciona la opción "Eliminar Servidor" | 7.1 El sistema muestra un listado de todos los servidores. |
| 8- El administrador selecciona el servidor que desea eliminar 9- Presiona el botón aceptar | 9.1 El Sistema muestra un mensaje alertando que el objeto será eliminado |
| 10- El administrador presiona el botón aceptar | 10.1 El sistema muestra un mensaje informando que el servidor se eliminó satisfactoriamente. |

| | |
|--|--|
| Flujo alternativo | |
| Acción del actor | Respuesta del sistema |
| Sección “Nuevo Servidor” | |
| Línea 3 | |
| | 3.3 El sistema muestra un mensaje de error informando que los datos no fueron guardados correctamente. |
| Sección “Modificar Servidor” | |
| Línea 6 | |
| | 6.3 El sistema muestra un mensaje de error informando que los datos no fueron guardados correctamente. |
| Sección “Eliminar Servidor” | |
| Línea 9.1 | |
| 9.2 El administrador cancela la acción | |
| Línea 10.1 | |
| | 10.2 El sistema muestra un mensaje de error informando que el servidor no fue eliminado correctamente. |
| Puntos de extensión. | |
| | |

| | |
|--|--|
| Caso de uso | |
| CU-4 | Gestionar Tablas Replicación |
| Propósito | Permitir al administrador especificar las tablas que va a replicar así como las operaciones de las mismas. |
| Actores: Administrador | |
| Resumen: Es aquí donde el administrador selecciona las tablas que el desea que repliquen, especifica las operaciones que va a realizar cada tabla, la regla por la que va a replicar. | |

| | |
|--|--|
| Referencias | RF- 6.1, 6.2, 6.3 |
| Acción del actor | Respuesta del sistema |
| 1- El administrador necesita especificar, modificar o eliminar la información de las tablas que va a replicar. | 1.1 El sistema ejecuta las siguientes acciones: a) Si se decide especificar las tablas que van a replicar ver sección “Adicionar Tablas Réplica” b) Si se decide modificar la información sobre las tablas que van a replicar ver sección: “Modificar Tablas Réplica” c) Si se decide eliminar la información sobre las tablas que van a replicar ver sección “Eliminar Tablas Réplica” |
| Sección “Adicionar Tablas Réplica” | |
| 2- El administrador selecciona la opción “Adicionar Tablas Réplica” | 2.1 El sistema muestra una interfaz con un listado con todas las tablas de la base de datos. |
| 3- El administrador selecciona las tablas que el desea replicar. | 3.1 El sistema activa los campos que permiten introducir la información de la tabla para la réplica: regla por la que va a replicar, operaciones que va a realizar, si la inserción va a ser de forma ordenada, y el campo identificador de la tabla. |
| 4- El administrador introduce la información de cada tabla que va a replicar. | |
| 5- El administrador presiona el botón “Aceptar” | 5.1 El sistema muestra un mensaje informando que la información se guardó satisfactoriamente. |
| Sección “Modificar Tablas Réplica” | |
| 6- El administrador selecciona la opción “Modificar Tablas Réplica” | 6.1 El sistema muestra una interfaz con un listado con todas las tablas de la réplica. |
| 7- El administrador selecciona las tablas que el desea modificar. | 7.1 El sistema activa los campos con la información de la tabla seleccionada. |

| | |
|---|--|
| 8- El administrador modifica la información de la tabla | |
| 9- El administrador presiona el botón aceptar. | 9.1 El sistema muestra un mensaje informando que la información se guardó satisfactoriamente. |
| Sección “Eliminar Tablas Réplica” | |
| 10- El administrador selecciona la opción “Eliminar Tablas Réplica” | 10.1 El sistema muestra una interfaz con un listado con todas las tablas de la réplica. |
| 11- El administrador selecciona las tablas que el desea eliminar. | |
| 12- El administrador presiona el botón aceptar. | 12.1 El sistema muestra un mensaje alertando que esta a punto de eliminar dichas tablas |
| 13- El administrador presiona el botón aceptar. | 13.1 El sistema muestra un mensaje informando que las tablas se eliminaron satisfactoriamente de la réplica. |
| Flujo alternativo | |
| Sección “Adicionar Tablas Réplica” | |
| Línea 5 | |
| | 5.2 El sistema muestra un mensaje informando que la información no se guardó satisfactoriamente. |
| Sección “Adicionar Tablas Réplica” | |
| Línea 9 | |
| | 9.2 El sistema muestra un mensaje informando que la información no se guardó satisfactoriamente. |
| Sección “Eliminar Tablas Réplica” | |
| Línea 13 | |
| | 13.2 El sistema muestra un mensaje de error informando que no se pudo eliminar satisfactoriamente el objeto seleccionado |

| Caso de uso | |
|---|--|
| CU-5 | Monitorear Transacciones CD |
| Propósito | Mostrarle a los administrador del centro de datos una grafica de la transacciones que van llegando al centro de datos para que el pueda monitorearlas en tiempo real. |
| Actores: Administrador CD | |
| Resumen: Su objetivo es monitorear todas las transacciones que ocurren en el proceso de replicación, esto les permitirá a los administradores ver si en algún momento hay un pico de transacciones (un aumento repentino de llegada de transacciones al centro de datos), poder ver los detalles de las mismas, mostrar las operaciones de las transacciones y sus detalles. | |
| Referencias | RF 4.1, 4.2, 4.3, 4.4 |
| Acción del actor | Respuesta del sistema |
| 1-El administrador presiona la opción "Monitorear Transacciones" | 1.1- El sistema muestra un gráfico donde se puede observar las transacciones que van llegando al centro de datos en tiempo real. |
| 2. El administrador hace clic sobre la parte de la grafica que desea ver mas detalles. | 2.1 El sistema muestra la parte seleccionada en un grafico ampliado. 2.2 El sistema muestra un listado de las transacciones que pertenecen a la parte del gráfico seleccionado. |
| 3- El administrador hace clic sobre la transacción que desea obtener más detalles. | 3.1 El sistema muestra todas las operaciones detalladas de la transacción. |

| Caso de uso | |
|---|---|
| CU-6 | Autenticar Usuario |
| Propósito | El propósito general de este caso de uso es autenticar los usuarios que se conecten a la aplicación para poder definir los niveles de acceso de cada uno y establecer los permisos correspondientes al usuario. |
| Actores: Administrador | |
| Resumen: Su objetivo es permitir autenticar a los usuarios que se conecten a la aplicación. Comprobar que los usuarios y sus contraseñas sean validos, tengan los permisos necesarios para realizar la replicación y estén accediendo desde un servidor autorizado a replicar. | |
| Referencias | |
| Acción del actor | Respuesta del sistema |
| 1- el administrador intenta acceder a la aplicación de la réplica. | 1.1 El sistema muestra una interfaz de autenticación, donde el administrador debe introducir un usuario u una contraseña validos para acceder a la aplicación. |
| 2- El administrador introduce usuario y contraseña | 2.1 El sistema comprueba que el usuario y la contraseña sean correctos. |
| | 2.2 Muestra la página principal de la aplicación |
| Flujo alternativo | |
| Línea 2.2 | |
| Acción del actor | Respuesta del sistema |
| | 2.3 Muestra un mensaje de error informando que el usuario o la contraseña son incorrectos. |
| Puntos de extensión. | |
| | |

| Caso de uso | |
|--|--|
| CU-7 | Mostrar Reportes |
| Propósito | El propósito de este caso de uso es permitirles a los administradores listar reportes de la réplica. |
| Actores: Administrador | |
| Resumen: Su objetivo es permitirles a los administradores ver una serie de reportes del estado de la réplica, los conflictos ocurridos, las bases de datos que mas conflictos presentan para la replicación, entre otros aspectos importantes para la administración de la réplica. | |
| Referencias | RF 9.1, 9.2, 9.3, 9.4, 9.5 |
| Acción del actor | Respuesta del sistema |
| 1- El administrador selecciona la acción “Mostrar Reportes” | 1.1- El sistema muestra una interfaz donde se muestran todos los reportes de la aplicación. |
| 2- Hace clic sobre el reporte que desea ver mas detalles | 2.2 Muestra los detalles del reporte seleccionado. |

| Caso de uso | |
|---|---|
| CU-8 | Gestionar Usuario |
| Propósito | El propósito de este caso es administrar todos los usuarios que se conecten a la réplica; crear, modificar y eliminar usuarios de la réplica. |
| Actores: Administrador | |
| Resumen: En este caso de uso se crean todos los usuarios para el proceso de replicación, se le asignan los permisos necesarios para realizar las operaciones, se modifica su información y se eliminan los mismos. | |
| Referencias | RF 5.1, 5.2, 5.3 |
| Acción del actor | Respuesta del sistema |
| 1- El administrador necesita crear, modificar o eliminar un usuario | 1.1 El sistema muestra las siguientes opciones: a) Si se desea crear un nuevo usuario ver sección “Crear Usuario”. |

| | |
|--|--|
| | <p>b) Si se desea modificar un usuario ver sección “Modificar Usuario”</p> <p>c) Si se desea eliminar un usuario ver sección “Eliminar Usuario”.</p> <p>d) Si se va a autenticar un usuario ver sección “Autenticar Usuario”</p> |
| Sección “Crear Usuario” | |
| 2- El administrador escoge la opción crear usuario. | 2.1 Muestra una interfaz para la creación del usuario donde el administrador debe introducir: nombre del usuario, contraseña, tablespace por defecto, tablespace temporal y el tablespace para el cual va a tener cuota ilimitada. |
| 3- El administrador introduce los datos solicitados y presiona el botón “Aceptar” | 3.1 El sistema muestra un mensaje informando que el usuario se creo satisfactoriamente. |
| Sección “Modificar Usuario” | |
| 4- El administrador escoge la opción modificar usuario | 4.1 El sistema muestra un listado con todos los usuarios de la aplicación. |
| 5- El administrador selecciona el usuario que desea modificar | 5.1 Muestra los datos del usuario seleccionado |
| 6- Modifica los datos que desea cambiar en el usuario y presiona el botón aceptar. | 6.1 Muestra un mensaje informando que el usuario se actualizó correctamente. |
| Sección “Eliminar Usuario” | |
| 7- El administrador selecciona la opción eliminar usuario | 7.1 El sistema muestra un listado de todos los usuarios de la aplicación. |
| 8- Selecciona el usuario que desea eliminar y presiona el botón “Aceptar” | 8.1 el sistema muestra un mensaje de alerta informando que esta a punto de eliminar el usuario seleccionado. |
| 9- Selecciona la opción aceptar | 9.1 Muestra un mensaje informando que el usuario se eliminó satisfactoriamente. |

| | |
|------------------------------------|---|
| Flujo alternativo | |
| Sección “Crear Usuario” | |
| Línea 3 | |
| | 3.2 El sistema muestra un mensaje de error informando que el usuario no se creo satisfactoriamente. |
| Sección “Modificar Usuario” | |
| Línea 6 | |
| | 6.2 Muestra un mensaje informando que el usuario no se actualizó correctamente. |
| Sección “Eliminar Usuario” | |
| Línea 9 | |
| 10- Selecciona la opción cancelar | 10.1 Cancela la operación indicada |
| Línea 9.1 | |
| | 9.2 Muestra un mensaje informando que el usuario no se eliminó satisfactoriamente. |
| Puntos de extensión. | |
| | |

| | |
|--|---|
| Caso de uso | |
| CU-9 | Gestionar Tablespace |
| Propósito | El propósito de este caso es administrar los tablespace para la réplica |
| Actores: Administrador | |
| Resumen: Aquí es donde se administran los tablespace de la réplica, en el se crean los tablespace con sus datafiles, se modifican sus parámetros y se eliminan. | |
| Referencias | RF 7.1, 7.2, 7.3 |
| Acción del actor | Respuesta del sistema |
| 1- El administrador necesita crear, modificar o eliminar un tablespace. | 1.1 El sistema muestra las siguientes opciones: a) Si se desea crear un tablespace ir a la sección |

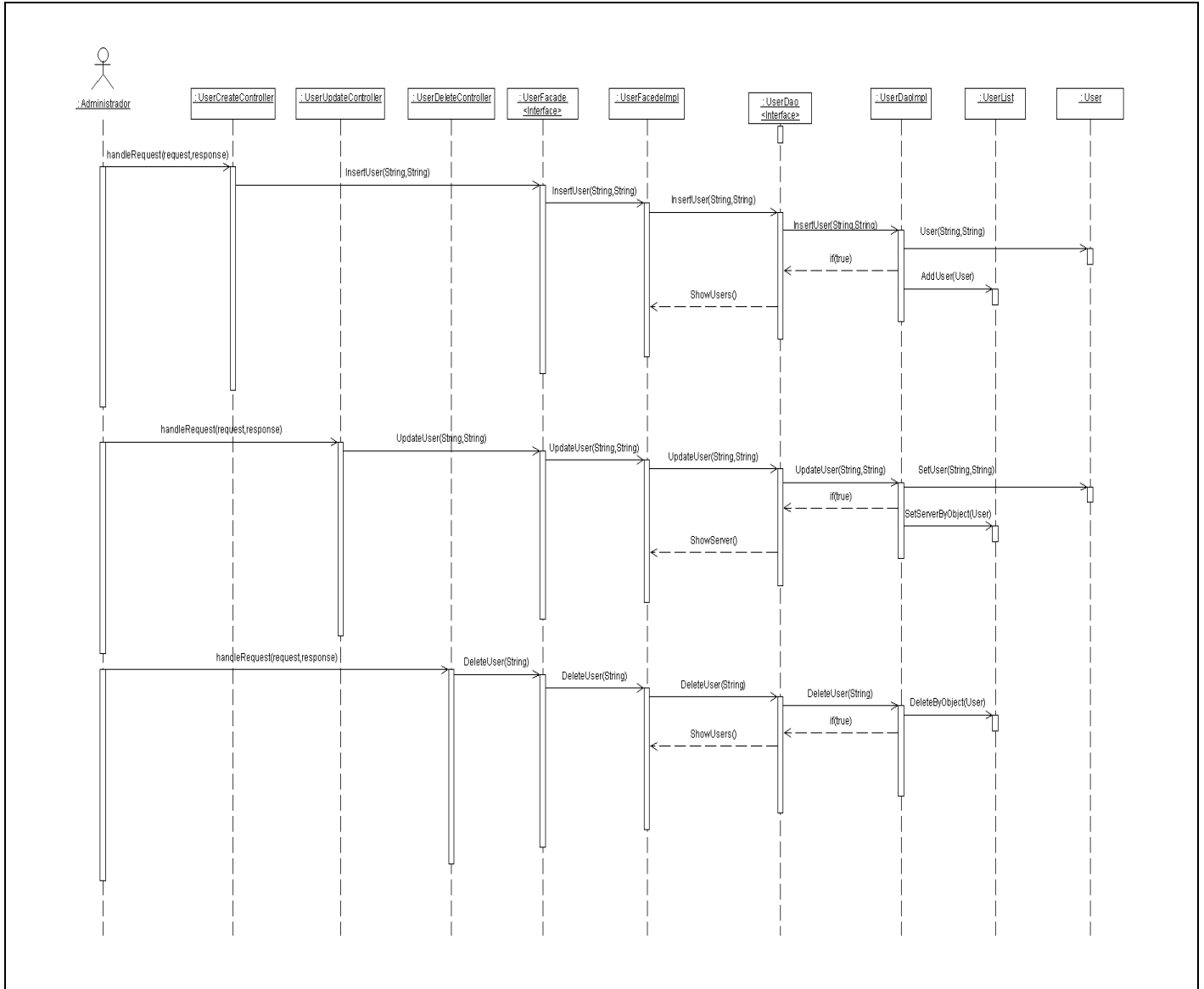
| | |
|---|--|
| | <p>“Crear Tablespace”</p> <p>b) Si se desea modificar un tablespace ir a la sección “Modificar Tablespace”</p> <p>c) Si se desea eliminar un tablespace ir a sección “Eliminar Tablespace”</p> |
| Sección “Crear Tablespace” | |
| 2- El administrador selecciona la opción “Crear Tablespace”. | 2.1 Muestra una interfaz para crear un tablespace, donde el administrador debe introducir: nombre del tablespace, especificar el tipo, nombre de los datafiles y tamaño de los datafiles. |
| 3- El usuario introduce los datos solicitados y presiona el botón “Aceptar” | 3.1 El sistema muestra un mensaje informando que el tablespace se creo satisfactoriamente. |
| Sección “Modificar Tablespace” | |
| 4- El administrador selecciona la opción “Modificar Tablespace”. | 4.1 El sistema muestra un listado con todos los tablespace de la réplica. |
| 5- Selecciona el tablespace que desea modificar. | 5.1 Muestra los datos del tablespace seleccionado. |
| 6- Modifica los datos y presiona el botón “Aceptar”. | 6.1 Muestra un mensaje informando que el tablespace se modificó correctamente. |
| Sección “Eliminar Tablespace” | |
| 7- El administrador selecciona la opción “Eliminar Tablespace”. | 7.1 El sistema muestra un listado con todos los tablespace de la réplica. |
| 8- Selecciona el tablespace que desea eliminar y presiona el botón “Aceptar”. | 8.1 Muestra un mensaje alertando que esta apunto de eliminar el objeto seleccionado y pregunta si desea continuar con al operación. |
| 9- Presiona el botón aceptar | 9.1 Muestra un mensaje informando que el objeto fue eliminado satisfactoriamente. |

| | |
|---------------------------------------|---|
| Flujo alternativo | |
| Sección “Crear Tablespace” | |
| Línea 3 | |
| | 3.2 El sistema muestra un mensaje informando que el tablespace no se creo satisfactoriamente. |
| Sección “Modificar Tablespace” | |
| Línea 6 | |
| | 6.2 Muestra un mensaje de error informando que el tablespace no se pudo modificar. |
| Sección “Eliminar Tablespace” | |
| Línea 9 | |
| 10- Presiona el botón “Cancelar” | 10.1 Cancela la operación indicada. |
| | 9.2 Muestra un mensaje informando que el objeto no se pudo eliminar. |

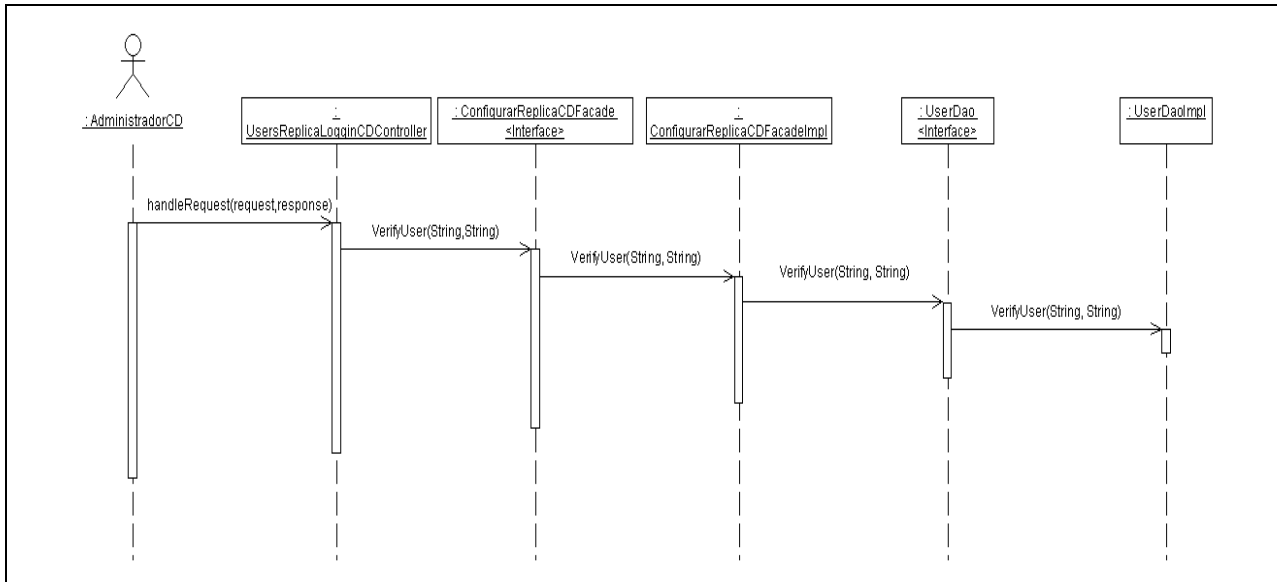
ANEXO 2

Diagramas de interacción.

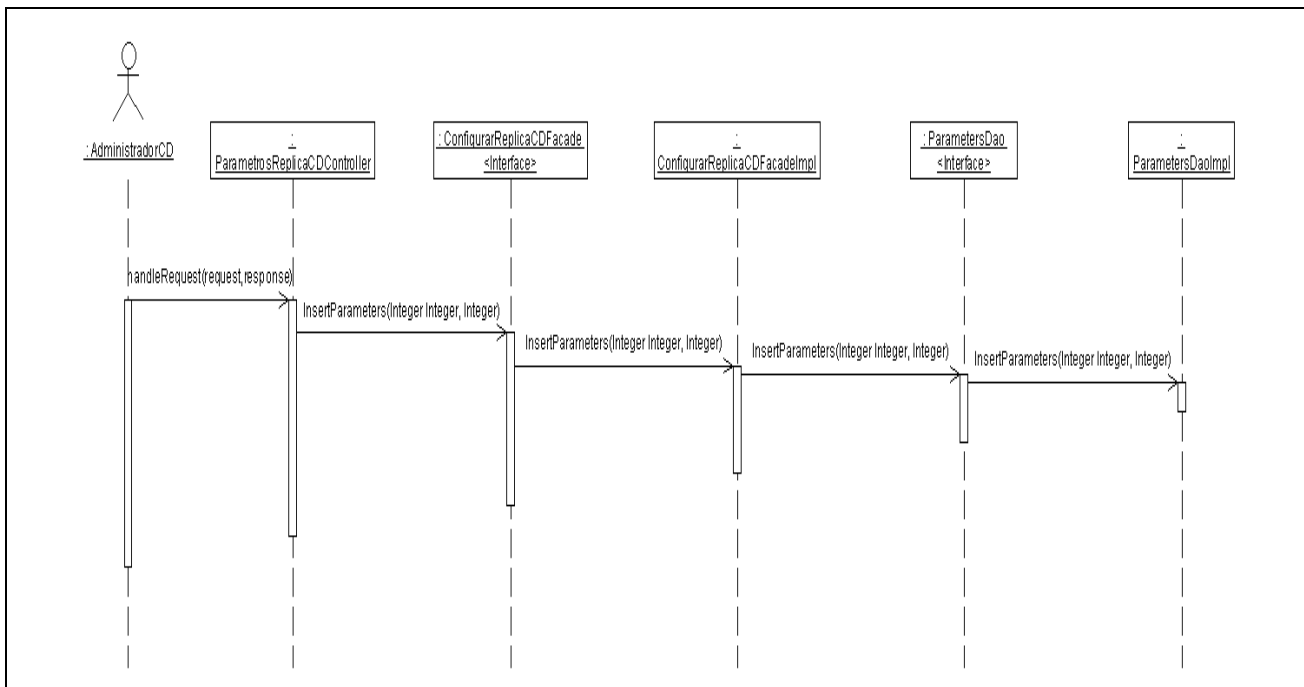
Gestionar Usuario



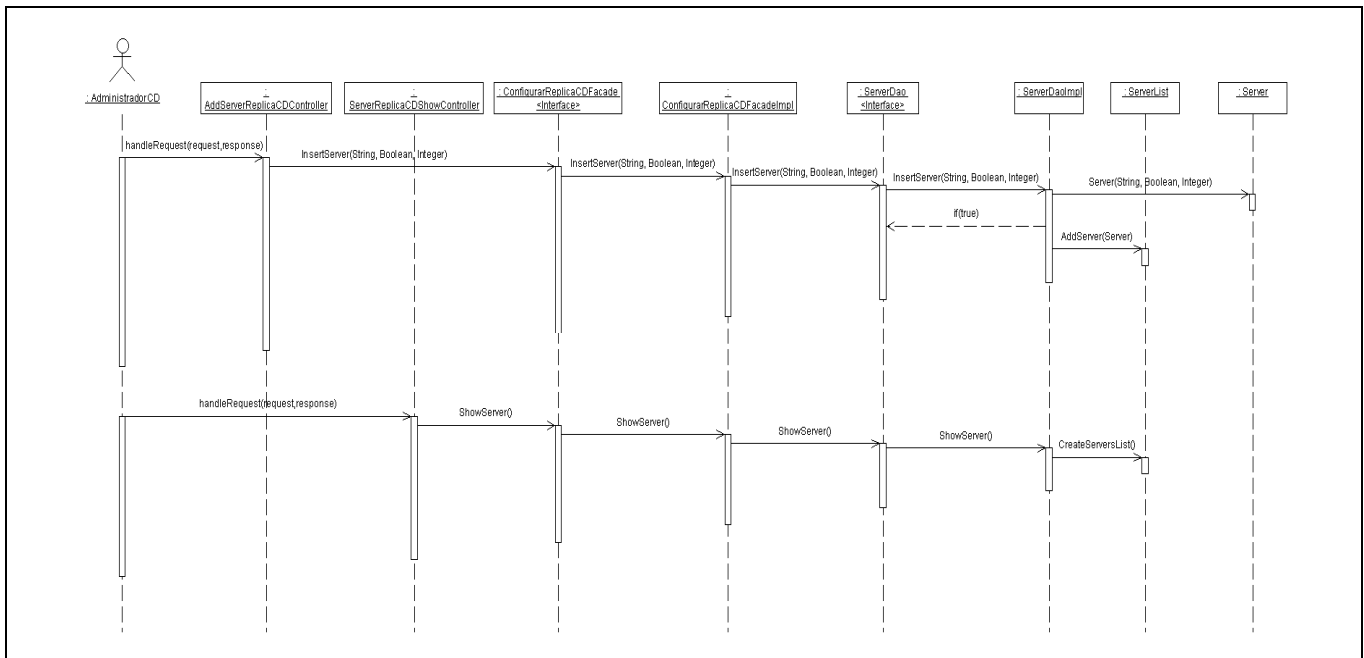
Configurar Réplica CD escenario: Loggin.



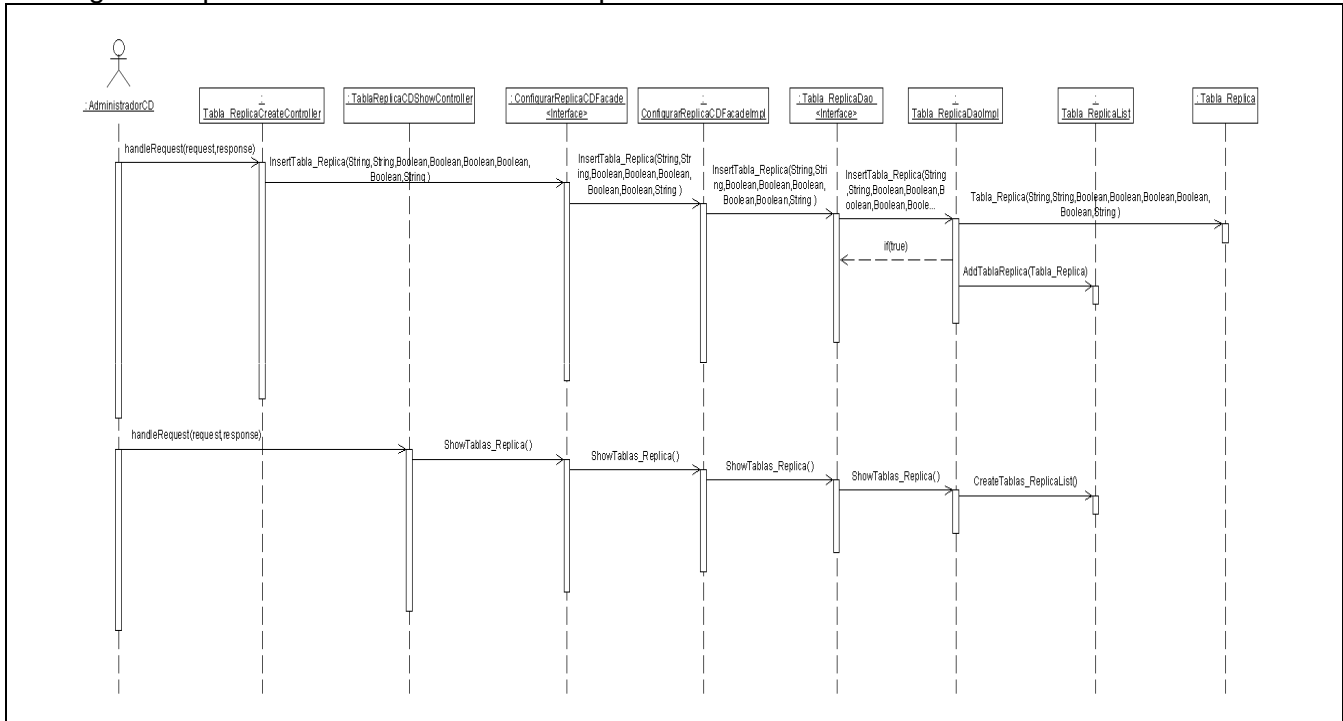
Configurar Réplica CD escenario: Parámetros de Almacenamiento



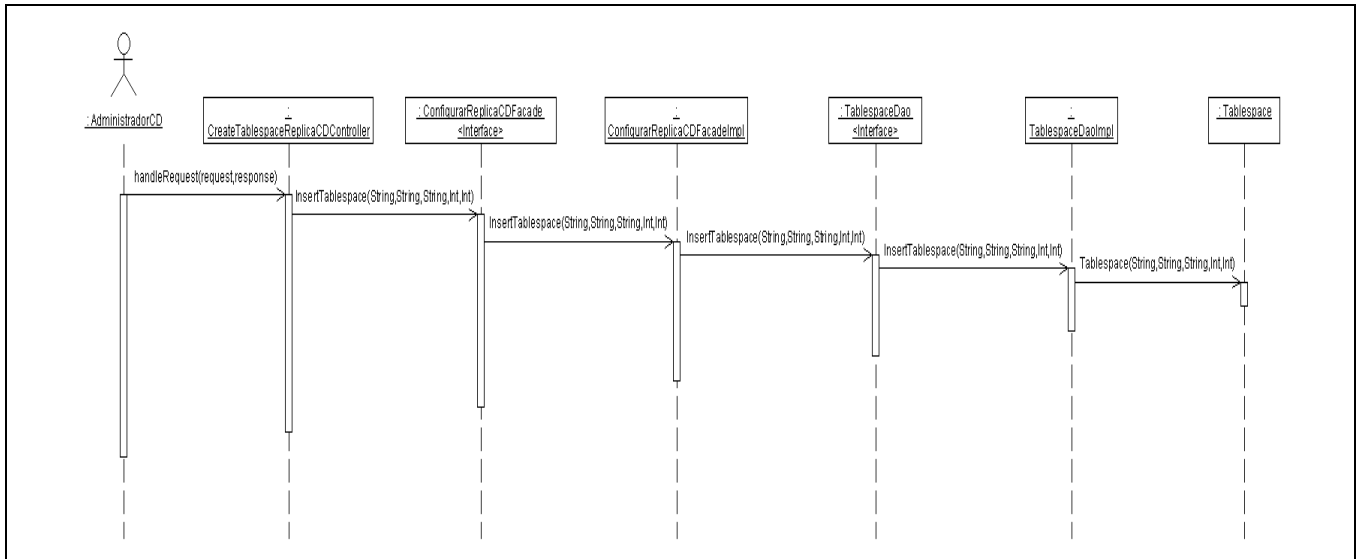
Configurar Réplica CD escenario: Servidores Replicación.



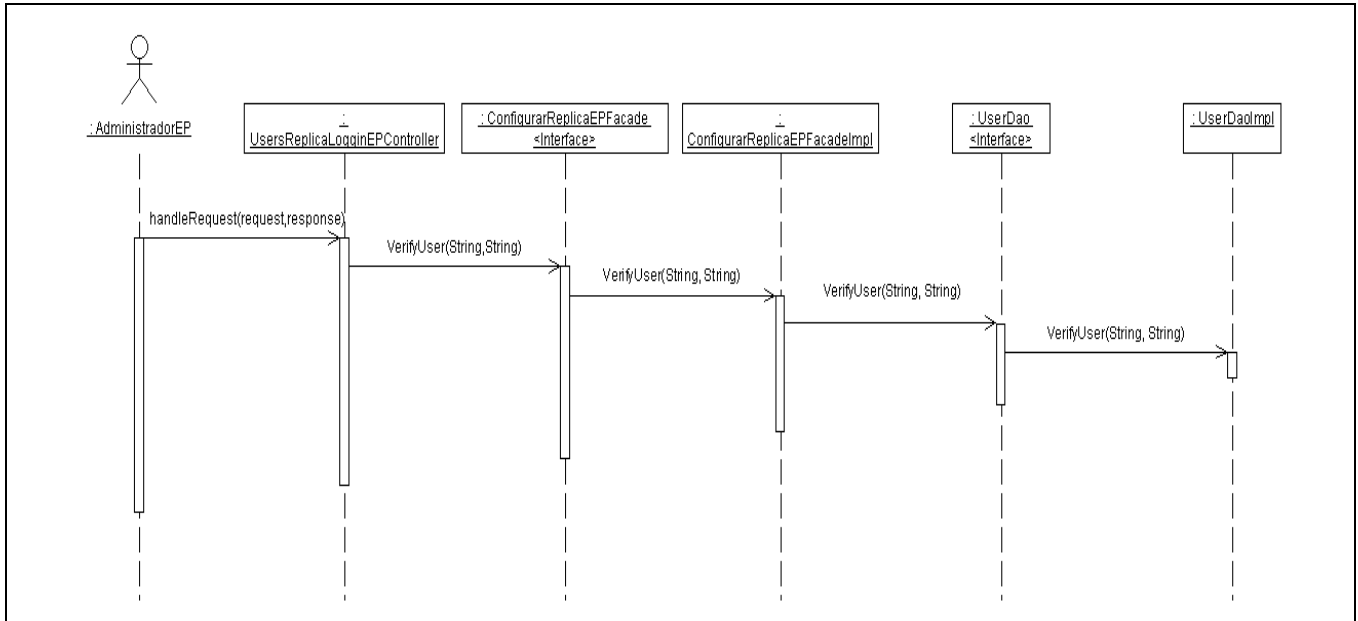
Configurar Réplica CD escenario: Tablas Réplica.



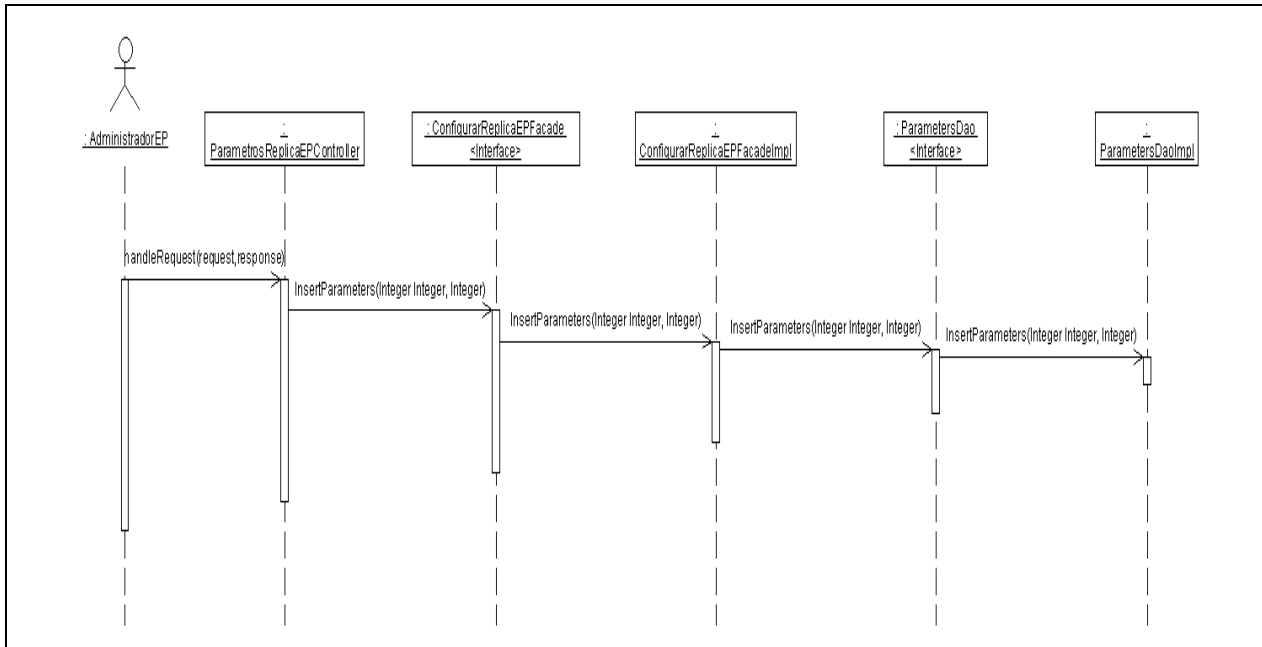
Configurar Réplica CD escenario: Tablespace



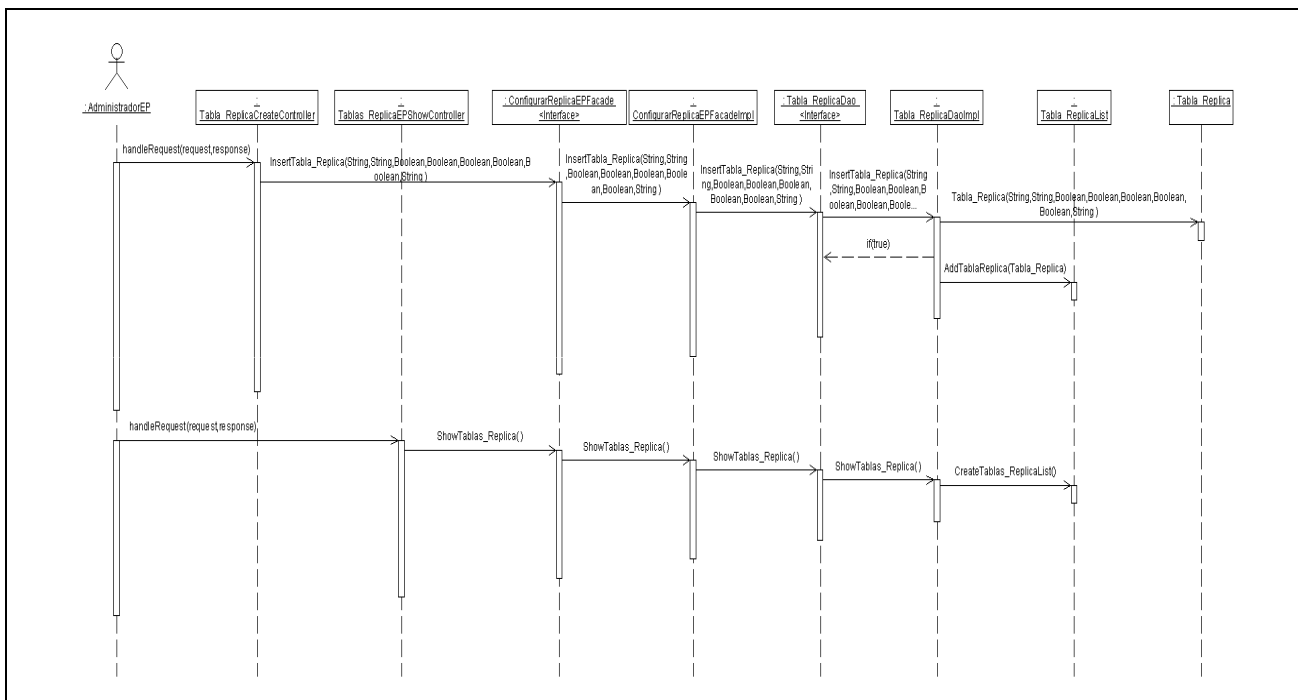
Configurar Réplica EP escenario: Login



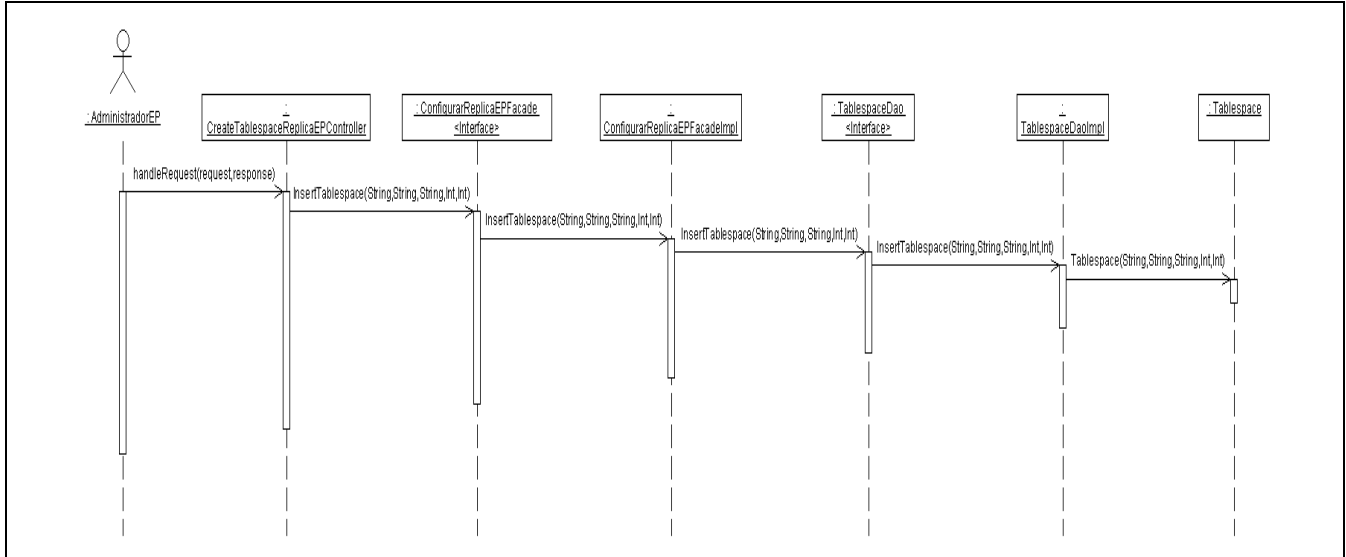
Configurar Réplica EP escenario: Parámetros de Almacenamiento



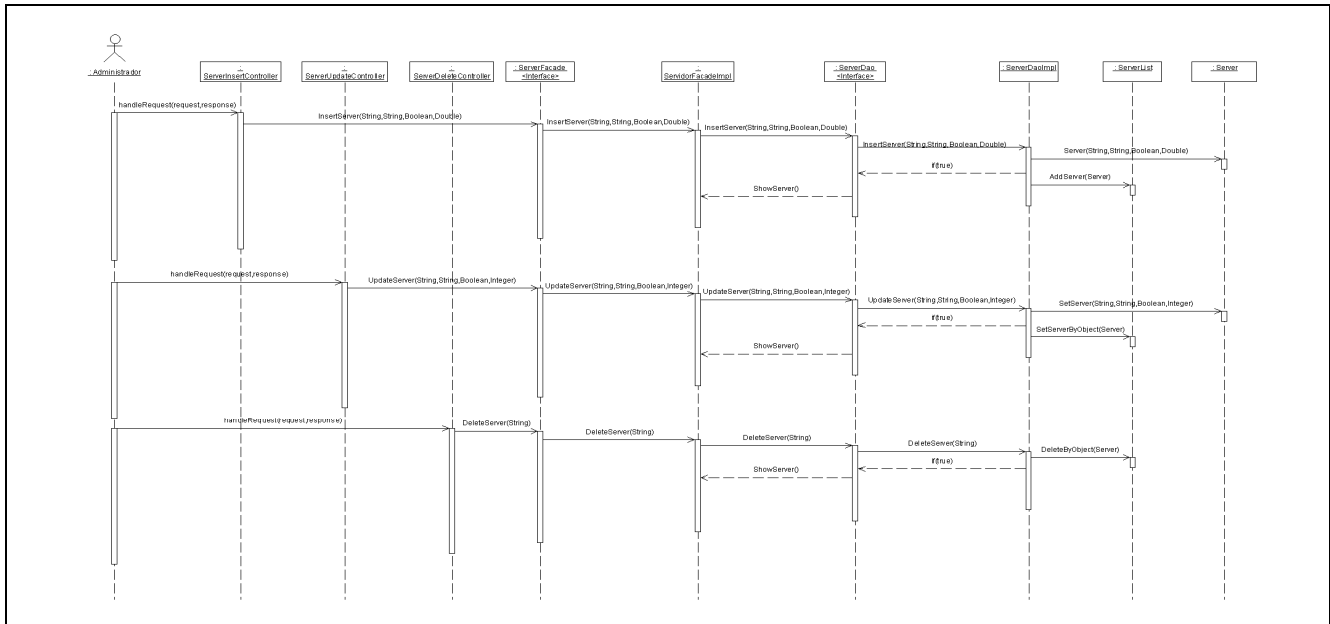
Configurar Réplica EP escenario: Tablas Réplica



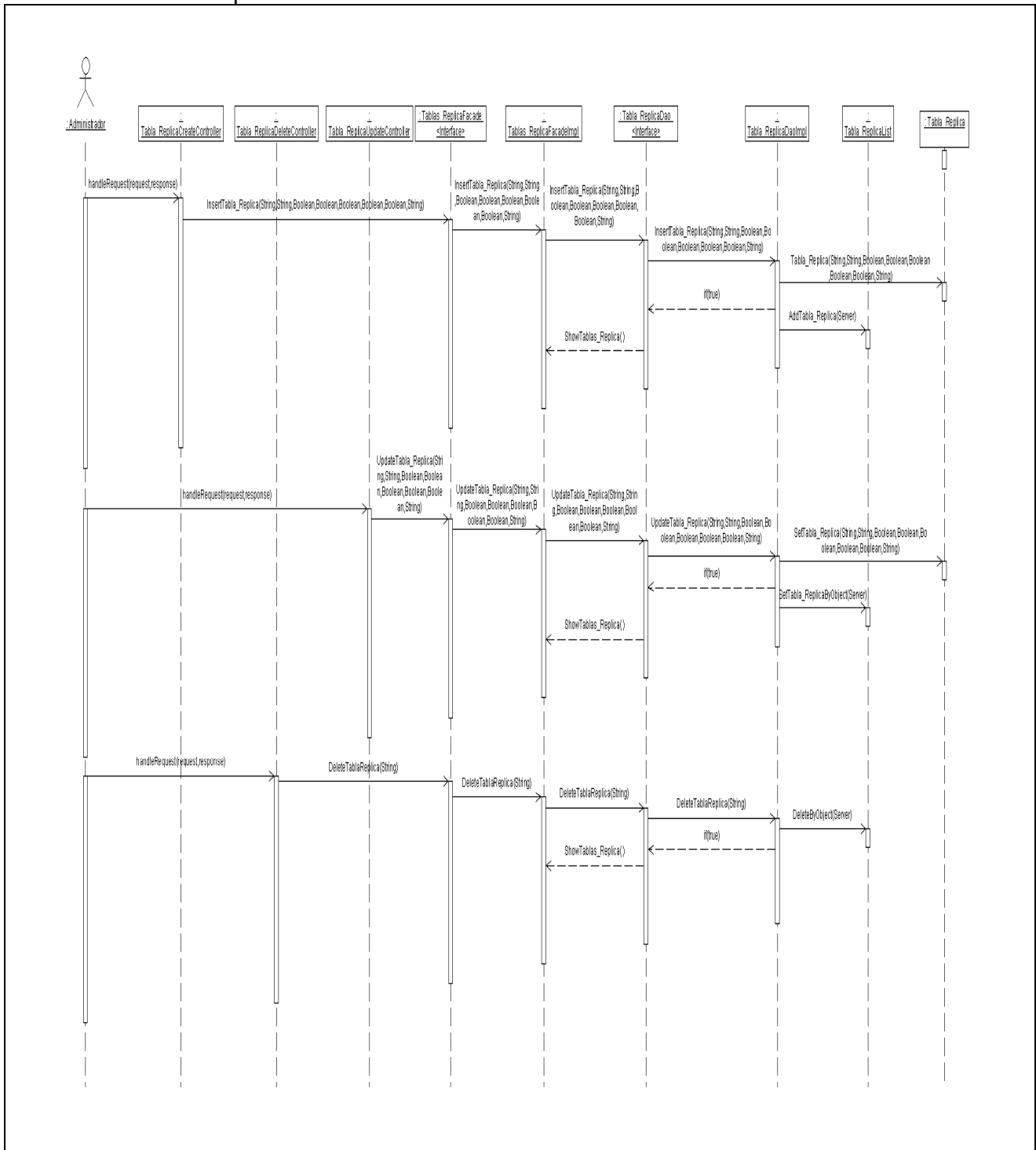
Configurar Réplica EP escenario: Tablespace Réplica



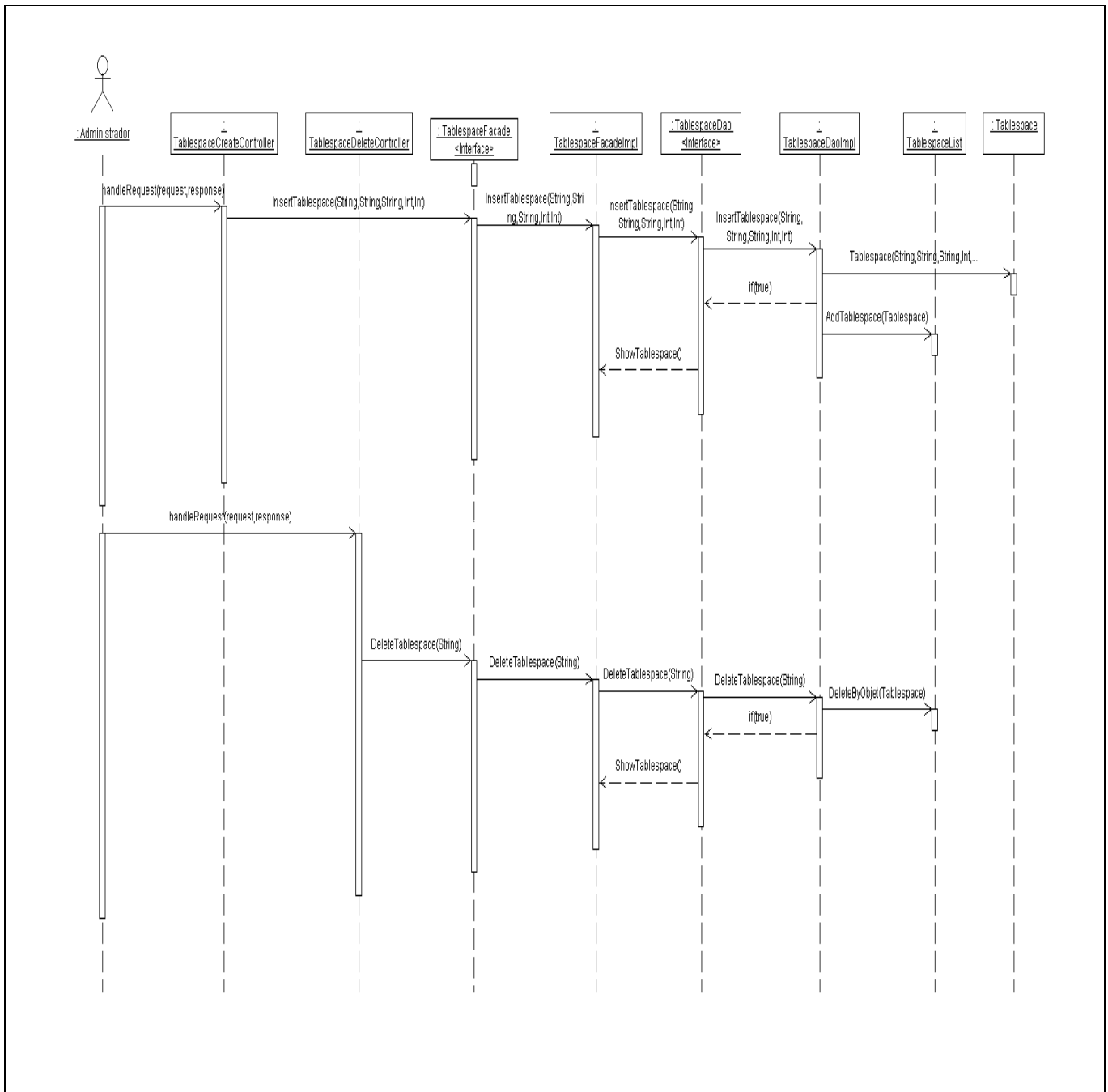
Gestionar Servidor



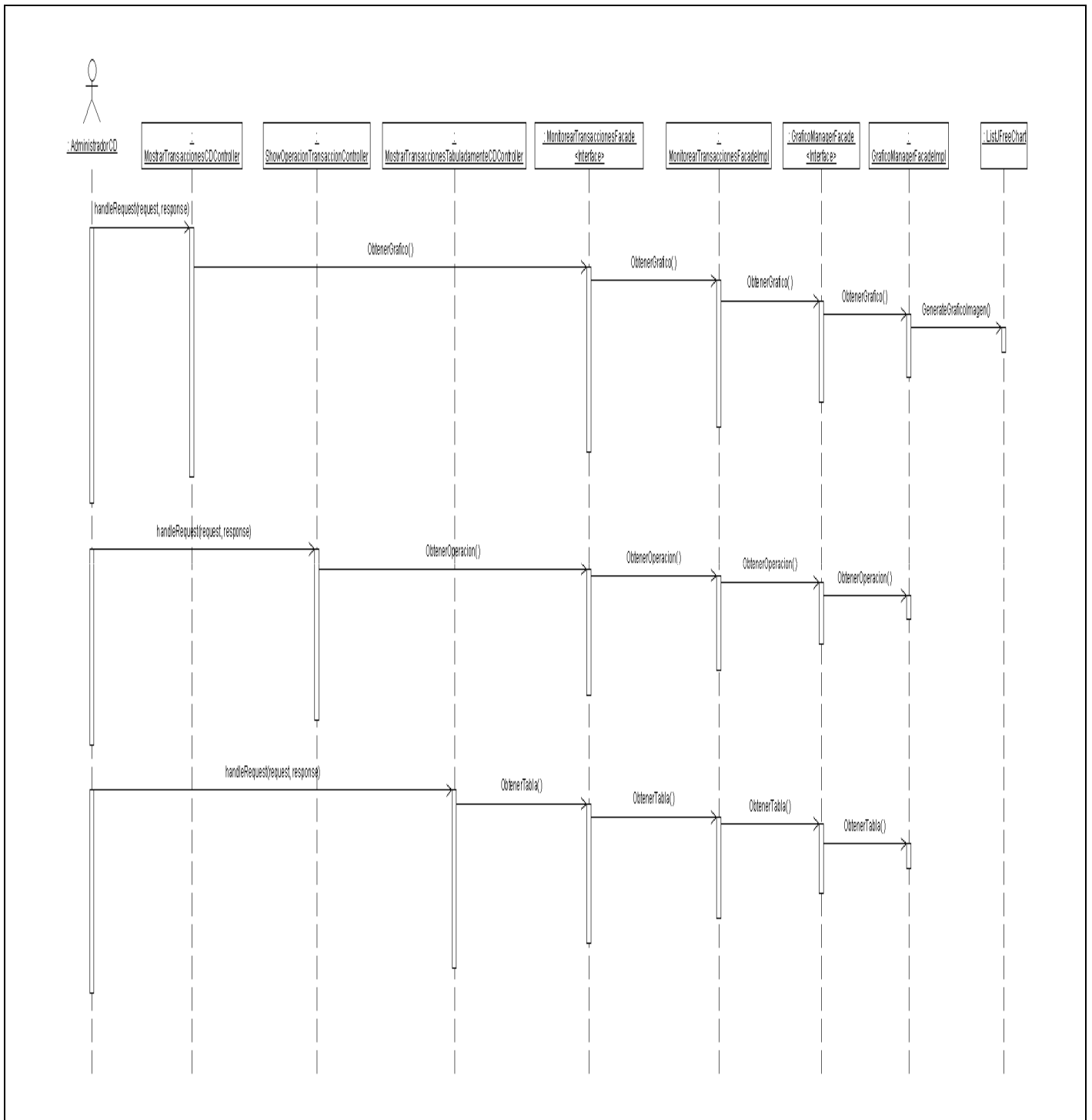
Gestionar Tablas Réplica



Gestionar Tablespace



Monitorear Transacciones CD



Mostrar Reportes

