



Facultad 5

Centro de Consultoría y Desarrollo de Arquitecturas Empresariales

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

Título: Configuración y Monitorización de la réplica de estructura para el Replicador de datos REKO

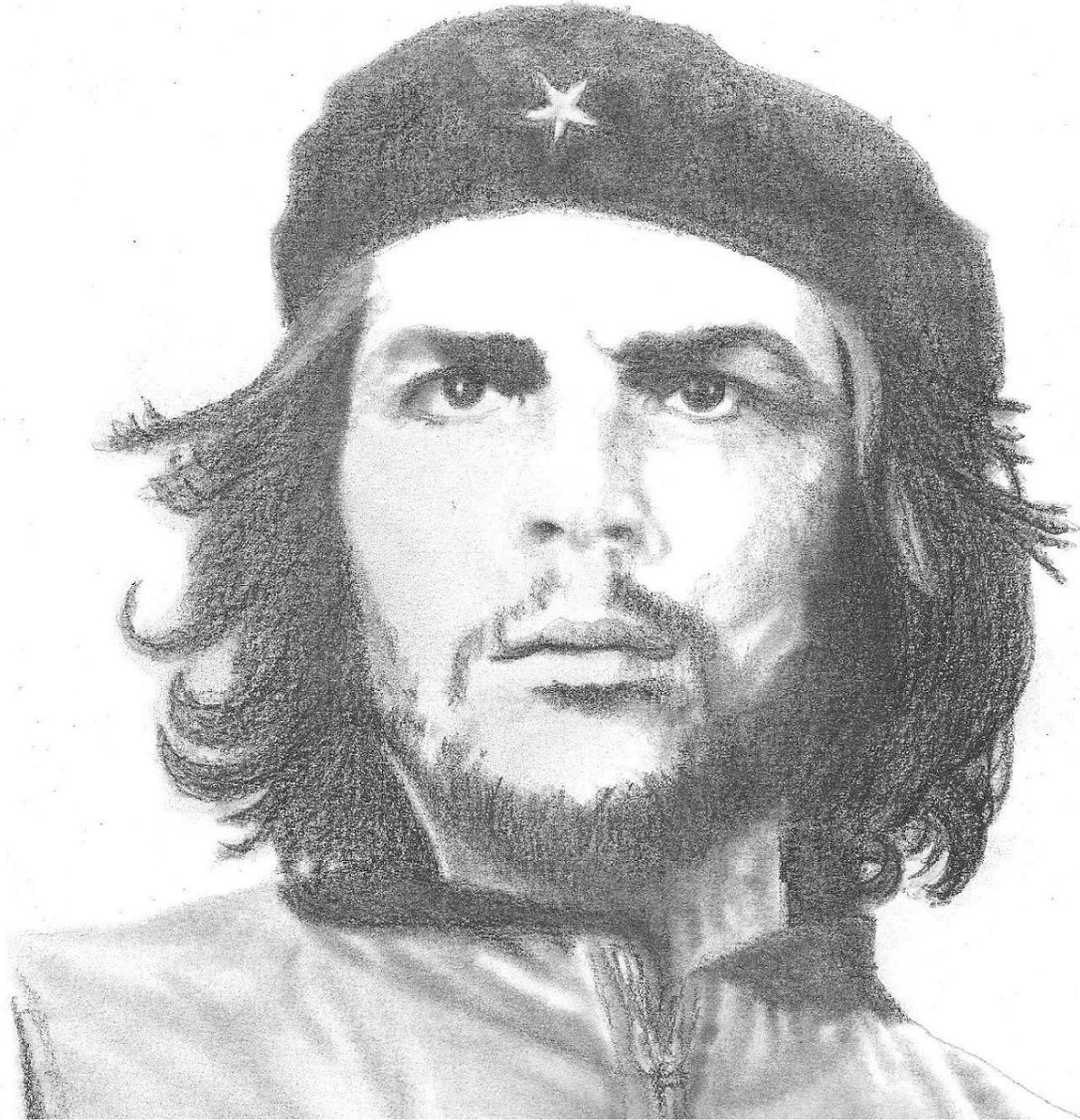
Autor: Alejandro García Santiesteban

Tutores: Ing. Katia Roselló Díaz

Ing. Gloria Raquel Leyva Jerez

La Habana, junio de 2015
"Año 57 de la Revolución"

PENSAMIENTO



“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte”.

Ernesto Guevara de la Serna.

DECLARACIÓN DE AUTORÍA

Declaro ser autor del presente trabajo de diploma y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste, firmo la presente a los ___ días del mes de _____ del año _____.

Firma del autor:

Alejandro García Santiesteban

Firma de los tutores:

Ing. Katia Roselló Díaz

Ing. Gloria Raquel Leyva Jerez

AGRADECIMIENTOS

Agradezco primeramente a Dios por darme la oportunidad de vivir cada minuto de esta vida.

A mi madre y mi padre que han sido muy importantes en el desempeño de este trabajo de diploma, por enseñarme a ser una mejor persona y nunca dudar de mí.

A mi amigo Arisney, que más que un amigo, se comportó como mi mamá, mi papá y mi hermano durante gran parte de la universidad, brindándome su amistad incondicional, sus buenísimos consejos cuando estaba deprimido o desorientado.

A mis amigos Lisardo, Roberto, Jordanis que siempre estuvieron en los momentos difíciles cuando los necesité.

A mis tutoras Gloria y Katia que se jodieron tanto con mi tesis, que tanto me halaron las orejas para que me pusiera las pilas, que más que tutoras son mis amigas.

A mi prima Lusmeidi que más que una prima es mi hermana, que la amo con mi vida y siempre estuvo presente.

A toda mi familia y amigos que de alguna forma contribuyeron a mi formación como profesional.

DEDICATORIA

A mi abuela Luisa González que Dios me la tenga en la gloria, por ser mucho más que una abuela, por ser madre, padre y amiga, es por ella que me mantengo con las fuerzas para seguir viviendo y luchando por siempre salir adelante. A mi mamá Oneida Santiesteban que la amo con todas las fuerzas de mi corazón por apoyarme y aconsejarme siempre, guiándome por el buen camino y dándome ánimos en los momentos más difíciles de mi vida. A mi papá por mostrarme su apoyo incondicional en todo momento, por su confianza en mí y siempre darme sus consejos. A mi hermanita querida por su amor incondicional, por ser una parte importante de mi vida. A mi familia en general que ha sido tan importante en toda la carrera.

RESUMEN

El Replicador de datos REKO realiza el proceso de réplica tanto de datos como de estructura. Cuando en la base de datos se genera cualquier cambio de esquema sobre las acciones especificadas en la configuración de réplica, todos los cambios son replicados. Al replicar todas las estructuras, cambios no deseados por el cliente son enviados hacia las instancias que intervienen en el proceso de replicación. Por otra parte, durante la réplica el administrador no puede visualizar el comportamiento del proceso, dificultando el trabajo del personal involucrado en la administración del proceso de réplica.

El presente trabajo tiene como objetivo extender los módulos de Configuración y Monitor de Réplica para la réplica de estructura en el software Replicador de datos REKO, que permita hacer efectivos los tiempos de respuesta y evitar pérdida de información.

En la investigación se describe la problemática inicial, la utilización de los métodos científicos y el estudio de replicadores actuales con características similares que determinó como resultado la extensión de estos componentes. Se aplicó la metodología AUP adaptada al ciclo de vida definido para la actividad productiva que permite describir el diseño e implementación de los componentes. Los módulos fueron desarrollados haciendo uso de herramientas *Open Source*¹ y librerías de clases con licencias gratuitas. El módulo Configuración de Réplica permite establecer condiciones sobre las configuraciones para restringir la réplica de estructura a las necesidades del cliente, mientras que el módulo Monitor de Réplica permite observar el comportamiento de la réplica de estructura durante este proceso.

Palabras claves: configuración, lenguaje de definición de datos, monitorización, réplica de estructura.

¹ De código abierto, término con el cual se conoce al software distribuido y desarrollado libremente. Esto significa que cuando los programadores pueden leer, modificar y redistribuir el código fuente de un programa, éste evoluciona y mejora.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	6
1.1 Conceptos asociados al objeto de estudio del problema.....	6
1.1.1 Base de datos	6
1.1.2 Sistemas distribuidos.....	6
1.1.3 DDL.....	6
1.1.4 Replicación	7
1.1.5 Configuración	7
1.1.6 Monitorización.....	7
1.2 Replicadores homólogos.....	8
1.3 Metodología de Desarrollo.....	9
1.3.1 Proceso Unificado Ágil (AUP)	9
1.4 Herramientas, tecnologías y lenguajes de desarrollo.....	12
1.4.1 Lenguajes de programación	13
1.4.2 Herramientas de programación.....	13
1.4.3 Sistema de Gestión de Base de Datos	13
1.4.4 Tecnologías de desarrollo.....	14
1.4.5 Herramientas de modelado	14
1.4.6 Lenguaje de modelado	15
1.4.7 Servidor de mensajería	15
1.4.8 Servidor de aplicación	15
1.5 Consideraciones parciales del capítulo	16
CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.....	17
2.1 Descripción detallada del proceso a desarrollar	17
2.2 Modelo de dominio.....	17
2.2.1 Descripción de las clases de dominio	18

2.3	Especificación de requisitos	19
2.3.1	Requisitos funcionales.....	19
2.3.2	Requisitos no funcionales	19
2.4	Propuesta de solución	22
2.5	Historias de Usuario.....	24
2.5.1	Descripción de las HU	24
2.6	Descripción de la arquitectura.....	27
2.6.1	Arquitectura basada en componentes	27
2.6.2	Arquitectura en capas.....	28
2.6.3	Patrones de diseño	29
2.7	Modelo de diseño	32
2.7.1	Diagramas de paquetes	32
2.7.2	Diagrama de clases	33
2.7.3	Descripción de las clases.....	35
2.8	Modelo de despliegue	37
2.9	Consideraciones parciales del capítulo	38
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS AL SISTEMA.....		39
3.1	Implementación de la solución.....	39
3.1.1	Diagramas de componentes.....	39
3.2	Pruebas al sistema.....	41
3.2.1	Pruebas de Caja negra.....	42
3.2.2	Partición Equivalente	42
3.2.3	Pruebas de aceptación.....	49
3.2.4	Resultados obtenidos	51
3.3	Conclusiones parciales del capítulo	51
CONCLUSIONES GENERALES		52
RECOMENDACIONES		53
REFERENCIAS BIBLIOGRÁFICAS.....		54

ANEXOS..... 58

ÍNDICE DE TABLAS

Tabla 1 Fases de AUP adaptada a la actividad productiva de la UCI..... 9

Tabla 2 Roles y artefactos de salida11

Tabla 3 Requisitos no funcionales.....20

Tabla 4 HU1 Adicionar filtros a la configuración de réplica de estructura25

Tabla 5 HU4 Aplicar filtros de esquemas a los grupos de estructura replicable.....25

Tabla 6 HU5 Monitorizar la recepción de los grupos del proceso de réplica de estructura26

Tabla 7 Descripción textual de la clase ConfigurationManager36

Tabla 8 Descripción textual de la clase SenderManager.....36

Tabla 9 Descripción textual de la clase MonitoringManager.....37

Tabla 10 Prueba de partición equivalente sobre la HU1 Adicionar filtros a la configuración de réplica de estructura42

Tabla 11 Resultados obtenidos en la primera iteración de pruebas sobre el módulo "Monitor de Réplica"44

Tabla 12 Resultados obtenidos en la segunda iteración de pruebas sobre el módulo "Monitor de Réplica"44

Tabla 13 Resultados obtenidos en la tercera iteración de pruebas sobre el módulo "Monitor de Réplica"45

Tabla 14 Resultados obtenidos en la primera iteración de pruebas sobre el módulo "Configuración de Réplica"45

Tabla 15 Resultados obtenidos en la segunda iteración de pruebas sobre el módulo "Configuración de Réplica".....47

Tabla 16 Resultados obtenidos en la tercera iteración de pruebas sobre el módulo "Configuración de Réplica".....48

Tabla 18 Prueba de aceptación sobre la HU1 Adicionar filtros a la configuración de réplica de estructura.....49

Tabla 19 Prueba de aceptación sobre la HU4 Aplicar filtros de esquemas a los grupos de estructura replicable.....50

Tabla 20 Prueba de aceptación sobre la HU5 Monitorizar la recepción de los grupos del proceso de réplica de estructura50

ÍNDICE DE FIGURAS

Figura 1 Modelo de dominio.....18

Figura 2 Vista que muestra los principales componentes del Replicador de datos REKO.....28

Figura 3 Diagrama de paquetes de los RF 1, RF 2 y RF 3.33

Figura 4 Diagrama de paquetes del RF 5.....33

Figura 5 Diagrama de clases del Diagrama de paquetes de los RF 1, RF 2 y RF 3.34

Figura 6 Diagrama de clases del Diagrama de paquetes del RF 5.....35

Figura 7 Diagrama de despliegue38

Figura 8 Diagrama de componentes para la Consola de Administración.....40

Figura 9 Diagrama de componentes para el Distribuidor de cambios.....41

Figura 10 Diagrama de componentes para el Aplicador de cambios.....41

INTRODUCCIÓN

Como resultado del proceso de informatización de la sociedad se ha revolucionado la manera en la que se gestiona la información en las organizaciones y entidades. El crecimiento de los volúmenes de información, así como su difícil manejo, ha hecho imprescindible el uso de Sistemas Gestores de Bases de Datos (SGBD) para almacenar y acceder de manera rápida a toda la información generada. La descentralización geográfica de las organizaciones y entidades ha impulsado el empleo de los Sistemas de Bases de Datos Distribuidas (SBDD).

Los Sistemas Distribuidos necesitan un mecanismo que les facilite la gestión de la información entre las diferentes Bases de Datos (BD), para lo cual utilizan replicadores de datos. Para realizar la réplica, los replicadores requieren una configuración que les permite a los administradores adecuar la réplica a sus entornos, logrando que sea menor el tiempo y esfuerzo al evitar que se replique toda la información almacenada en la BD. Por otra parte, es muy común el uso de mecanismos de monitorización y control en la mayoría de estos sistemas, y los replicadores de datos no son la excepción. Con los mecanismos de monitorización y control se observa cómo se comporta la réplica de la información entre las diferentes BD, contribuyendo a la detección de los problemas.

En la Universidad de las Ciencias Informáticas (UCI), en el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), se viene desarrollando un software para la réplica en ambientes distribuidos llamado Replicador de datos REKO, el cual tiene como objetivo brindar una herramienta multiplataforma que le permita al usuario cubrir las necesidades fundamentales de réplica de datos, tales como: sincronización, transferencia de datos entre diversas localizaciones, centralización de la información en una única localización, así como protección y recuperación de los datos. El software ha sido implementado utilizando herramientas y librerías completamente libres, permitiendo que los costos para el desarrollo sean ínfimos, pues sería más costoso utilizar herramientas y librerías con licencias privadas. En sus inicios fue desarrollado para dar solución a las necesidades de réplica del Sistema de Gestión Penitenciario Venezolano (SIGEP), aunque en el presente 2015, está siendo usado en el Sistema de Gestión de los Tribunales, el Sistema de Gestión Fiscal, entre otros. El replicador cuenta con un módulo de Configuración de Réplica y un Monitor de Réplica.

El módulo de Configuración de Réplica permite la creación de configuraciones para las acciones DDL (Lenguaje de Definición de Datos, del inglés *Data Definition Language*), donde

se define la captura de los cambios de esquemas sobre toda la información. Cuando en la BD se genera cualquier cambio de estructura sobre la acción que es especificada en la configuración, todos los cambios son replicados, por lo que los cambios no deseados por el cliente son enviados hacia las instancias que intervienen en el proceso de réplica de estructura. Otro de los inconvenientes presentados por el módulo es que aumenta el tiempo de respuesta del sistema para realizar el proceso de envío de los cambios, al comparar toda la información y no solamente la que el cliente necesita.

El Monitor de Réplica permite observar todas las acciones del proceso de réplica de datos. Sin embargo, durante el proceso de réplica de estructura, el administrador de réplica no puede visualizar el comportamiento del proceso de dicha réplica, lo que puede provocar pérdida de información durante eventualidades tales como: fallos en la red o la interrupción de alguna instancia. Se dificulta el trabajo del personal encargado de administrar el sistema, debido a que tienen que invertir tiempo y esfuerzo para verificar de forma manual la integridad de los datos replicados, para evitar la pérdida de información e identificar con exactitud el origen del problema.

Dada la situación antes expuesta, se plantea el siguiente **problema a resolver**:

¿Cómo condicionar y monitorizar las acciones que intervienen en el proceso de réplica de estructura para hacer efectivos los tiempos de respuesta y evitar pérdida de información?

Para dar respuesta al problema a resolver se plantea como **objetivo general**:

Extender los módulos de Configuración y Monitor de Réplica para el proceso de réplica de estructura para el Replicador de datos REKO, que permita hacer efectivos los tiempos de respuesta y evitar pérdida de información.

El cual ha sido desglosado en los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación para organizar los conocimientos científicos adquiridos, los principales autores que trabajan la temática y asentar posición respecto al tema.
- Diseñar e implementar la propuesta de solución para extender el funcionamiento de los módulos definidos, aplicando la metodología y herramientas seleccionadas para lograr un producto acorde a las necesidades del cliente.
- Validar la solución propuesta mediante la aplicación de pruebas de caja negra, para el aseguramiento de la calidad del producto.

A partir del problema enunciado anteriormente, se define como **objeto de estudio**: el proceso de réplica de estructura en los SBDD, enmarcando el **campo de acción en**: el proceso de configuración de réplica y monitorización de las acciones en el Replicador de datos REKO.

Para cumplir con los objetivos planteados se proponen las siguientes **tareas de la investigación**:

- Realización del marco conceptual para precisar los principales conceptos que se emplean en la investigación.
- Estudio del estado del arte sobre las tendencias actuales en la utilización de sistemas de réplica, para sintetizar los resultados alcanzados en la revisión bibliográfica e indagaciones realizadas relacionadas con el tema.
- Selección de la metodología para definir los métodos y técnicas necesarias que guiarán el desarrollo.
- Descripción de las herramientas, tecnologías y lenguajes a utilizar para definir el ambiente de desarrollo.
- Realización del análisis y diseño del módulo para describir los artefactos, modelar el diagrama de clases y establecer los patrones de diseño.
- Implementación de las funcionalidades que dan cumplimiento a los requisitos identificados.
- Realización de pruebas para valorar la calidad del producto implementado.
- Valoración de los resultados obtenidos a partir de las pruebas aplicadas y objetivos planteados.

Teniendo en cuenta el objetivo general se plantea como **idea a defender** que con el desarrollo de los módulos de Configuración y Monitorización de réplica de estructura se evitará pérdida de información y reducirán considerablemente los tiempos de respuesta del Replicador de datos REKO.

Los **métodos científicos** son empleados para explicar fenómenos, establecer relaciones entre los hechos, enunciar leyes que expliquen los fenómenos físicos del mundo y permitan obtener, con estos conocimientos, aplicaciones útiles al hombre. En la presente investigación se hacen uso de algunos métodos, entre los que se encuentran:

Métodos teóricos

- Analítico – Sintético: el uso de este método permite desarrollar una profundización en la revisión bibliográfica para realizar análisis de elementos esenciales con el objetivo de estudiar cómo se desarrollan otros replicadores en un entorno de trabajo, analizando los resultados de sus deficiencias y características generales para así obtener un resultado eficiente que cumpla con los requisitos pedidos por el cliente.
- Modelación: método mediante el cual se generan abstracciones tales como los modelos, diagramas, artefactos o productos de trabajos que permiten entender las necesidades, definir metas, objetivos, procesos como parte del desarrollo de los módulos de configuración y monitorización de la réplica de estructura y explicar la solución que se propone en la investigación.

Métodos empíricos

- Observación: se utilizó para observar de forma práctica el funcionamiento de replicadores de datos. Se observa principalmente las acciones de réplica y monitorización de datos.
- Consulta de información: se utilizó para abordar los aspectos conceptuales y metodológicos de la investigación.
- Análisis de documentos: se expresa en la revisión realizada de los documentos rectores del software Replicador de datos REKO, literatura relacionada con la configuración y monitorización, y tesis de pregrado con temáticas relacionadas con la investigación.

Con la presente investigación se espera como **resultado** que el Replicador de datos REKO cuente con funcionalidades para establecer condiciones en las configuraciones de esquemas y monitorizar el proceso de réplica de estructura, que permita hacer efectivos los tiempos de respuesta y evitar pérdida de información, brindando así una solución más completa ante las necesidades de replicación.

A continuación se presenta la estructura capitular que guiará la presente investigación:

El **Capítulo 1. Fundamentación teórica** presenta un análisis del estado del arte sobre replicadores de datos. Se describen la metodología de desarrollo de software, herramientas y lenguajes a utilizar durante el desarrollo de la solución.

El **Capítulo 2. Propuesta de solución** muestra una descripción de la solución propuesta para la problemática planteada. Se describen los requisitos funcionales y no funcionales del sistema, además de los diagramas de clases e historias de usuario (HU). Se explica la arquitectura y los patrones de diseño empleados.

El **Capítulo 3. Implementación y Pruebas al Sistema** presenta parte del código fuente de las principales clases y la descripción del flujo de implementación. Se describen las pruebas realizadas al sistema con el objetivo de validar que la solución funcione de manera correcta y cumpla con las especificaciones planteadas en las descripciones de las HU.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se reflejan distintos elementos que permiten fundamentar el tema abordado en la investigación. Se hace alusión a los conceptos fundamentales que contribuyen al esclarecimiento de su objeto de estudio. Se realiza un análisis más profundo de la problemática antes planteada con el objetivo de lograr una mayor comprensión de la necesidad actual. Se efectúa un estudio sobre algunas de las soluciones de réplica, analizando sus principales ventajas y desventajas. Además, se describen las herramientas y tecnologías que sustentarán el desarrollo de la solución.

1.1 Conceptos asociados al objeto de estudio del problema

En esta sección se ofrecen algunos conceptos para el mejor entendimiento y comprensión de la investigación por parte del lector.

1.1.1 Base de datos

“Es un almacén que permite guardar grandes cantidades de información de forma organizada para que luego se puede encontrar y utilizar fácilmente. La información almacenada se encuentra agrupada o estructurada”.(1)

1.1.2 Sistemas distribuidos

Conjunto de computadores interconectados por una red que comparten un estado, ofreciendo una visión de sistema único. Estos sistemas cumplen con las propiedades de transparencia, escalabilidad, consistencia, fiabilidad y tolerancia a fallos.(2)

1.1.3 DDL

Es la parte del SQL² dedicada a la definición de la base de datos por lo que se especializa en la escritura de esquemas. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos.(3, 4)

Posee tres comandos básicos:

- *Create*: para crear nuevas tablas, campos e índices.
- *Alter*: para modificar las tablas agregando campos o cambiando la definición de los campos.

² Lenguaje de Consulta Estructurado, de inglés *Structured Query Language*. Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

- *Drop*: para eliminar tablas e índices.

1.1.4 Replicación

La replicación de datos consiste en el transporte de datos entre dos o más servidores, permitiendo que ciertos datos de la base de datos estén almacenados en más de un sitio, y así aumentar la disponibilidad de los datos y mejorar el rendimiento de las consultas globales.(5)

Randy Urbano³ en el año 2010 expresó que *“la replicación en el contexto de los SGBD, se entiende como el proceso de copiar y mantener objetos de una base de datos, entiéndase tablas, secuencias y/o triggers⁴, etcétera, en múltiples bases de datos, de tal forma que los cambios realizados localmente, sean enviados a las bases de datos remotas y luego sean aplicados”*. (6)

A consideración del autor, en la investigación se estará utilizando el concepto dado por Randy Urbano, debido a que considera que es el más relacionado con la presente investigación, además de estar actualizado.

En un sistema de información si durante la generación de reportes se ejecuta un comando DDL sobre la base de datos, la estructura de la base de datos cambia instantáneamente. Las diferencias de estructuras que se generan con respecto al resto de las bases de datos distribuidas, es posible actualizarlas con un componente que permita replicar los cambios de estructura o con un replicador de datos que posea o adquiera esta funcionalidad.(7)

1.1.5 Configuración

Es lo que determine cómo, a través de qué medios y con qué recursos funcionará un elemento. Se utiliza para adaptar una aplicación o un elemento al resto de los elementos del entorno y a las necesidades específicas del usuario. Es una tarea esencial antes de trabajar con cualquier nuevo elemento.(8)

1.1.6 Monitorización

Según el diccionario de la Real Academia Española se define como: *“Observar mediante aparatos especiales el curso de uno o varios parámetros fisiológicos o de otra naturaleza para detectar posibles anomalías”*.

³ Escritor técnico de Oracle.

⁴ Conocido como desencadenador. Procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación DML (*Lenguaje de Modelado de Datos*).

El diccionario Webster define la monitorización como: *“Para ver, comprobar y observar con un propósito especial”*.

El diccionario de Oxford la define como: *“Observar, supervisar o mantener en examen, a medida o prueba a intervalos, especialmente con el propósito de la regulación o control, o para controlar o regular la calidad técnica de algo. Cualquier instrumento o dispositivo para el seguimiento de algún proceso o cantidad”*.

El autor del presente trabajo llega a la conclusión de que el término monitorización está relacionado con la acción de controlar o supervisar continuamente una o varias actividades durante un tiempo establecido para comparar los resultados con los que se planificó y detectar posibles anomalías.

1.2 Replicadores homólogos

Luego de definir y describir los principales conceptos relacionados con la investigación científica, se hace un estudio acerca de los replicadores homólogos, atendiendo particularmente la forma en la que realizan la configuración y monitorización de la réplica.

Entre los replicadores estudiados se encuentran Tungsten Replicator v 2.0.4, Oracle Streams, SymmetricDS v 3.1.9, Slony-I v 2.0.7 y el Replicador de datos REKO v 4.0. Fueron estudiados con el objetivo de conocer su comportamiento a nivel mundial en cuanto a la realización de la configuración y monitorización de la réplica de estructura, es decir, conocer si no existe alguno que realice los procesos mencionados. En el caso de SymetricDS está limitado en cuanto a usabilidad, debido a que las configuraciones se realizan mediante comandos en una consola de administración, lo que implica que los usuarios deban poseer un conocimiento avanzado sobre el tema. En la información consultada de los replicadores Tungsten Replicator v 2.0.4, Oracle Streams, y Slony-I v 2.0.7 se hacía alusión a la realización de los procesos, sin ofrecer detalles en cuanto a su funcionamiento interno, ni de cómo se realiza.(7, 9-11) El Replicador de datos REKO aportó información valiosa para realizar los procesos, debido a que este software cuenta con una interfaz web para realizar los procesos de configuración y monitorización de réplica de datos. Además, brindó la posibilidad de estudiar el código fuente, lo cual resultó de vital importancia para el desarrollo de la solución.

1.3 Metodología de Desarrollo

Una metodología define “quién” debe hacer, “qué”, “cuándo” y “cómo” debe hacerlo, imponiendo un proceso disciplinado con el fin de aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Establece qué procedimientos, reglas y herramientas se deben utilizar durante el desarrollo del software.

Para guiar el proceso de desarrollo de la solución planteada se empleará la metodología AUP (Proceso Unificado Ágil, del inglés *Agile Unified Process*) adaptada al ciclo de vida definido para la actividad productiva, debido a que es la metodología establecida por la universidad para los proyectos de desarrollo como es el caso del Replicador de datos REKO, proyecto al cual se le integrará la solución propuesta en su próxima versión. Por ende, el proyecto requiere que el flujo de trabajo y artefactos generados como parte de la solución, se correspondan al desarrollo del proyecto. (12)

1.3.1 Proceso Unificado Ágil (AUP)

El Proceso Unificado Ágil de Scott Ambler del inglés *Agile Unified Process* es una versión simplificada de RUP (Proceso Unificado de Desarrollo, del inglés *Rational Unified Process*). Esta metodología describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Además, aplica técnicas ágiles incluyendo TDD (Desarrollo Dirigido por Pruebas, del inglés *Test Driven Development*), modelado ágil, gestión de cambios ágil, y refactorización de BD para mejorar la productividad. En la variación de AUP para la UCI se definen las siguientes fases, las cuales se aplicarán para guiar el desarrollo del presente trabajo.

Tabla 1 Fases de AUP adaptada a la actividad productiva de la UCI

Fases	Objetivos
Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo, además de decidir si se ejecuta o no el proyecto.

Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.
Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Fuente: Metodología de desarrollo para la actividad productiva de la UCI(12)

Se consideran las siguientes disciplinas:(12)

Modelado de negocio (opcional): el Modelado del negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito.

Requisitos: el esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.

Análisis y diseño: en esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.

Implementación: en la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.

Pruebas internas: en esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como

las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posibles componentes de prueba ejecutables para automatizar las pruebas.

Pruebas de liberación: pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

Pruebas de aceptación: es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Despliegue (opcional): constituye la instalación, configuración, adecuación, puesta en marcha de soluciones informáticas y entrenamiento al personal del cliente.

Los roles de AUP-UCI, que reúnen el conjunto esencial de habilidades para realizar el desarrollo, se describen a continuación:(12)

Tabla 2 Roles y artefactos de salida

Rol	Función principal	Artefactos de salida
Líder del proyecto Planificador	Lidera la planeación del proyecto, coordina interacciones con los interesados y conserva el equipo del proyecto enfocado en alcanzar los objetivos del proyecto.	Registro de proveedores de requisitos. Criterios para validar requisitos del cliente. Criterios para validar requisitos del producto. Estándar de codificación. Plan de desarrollo de software. Método de estimación.
Analista	Realizar tareas de relevamiento, análisis y diseño de los requerimientos y requisitos en el proyecto.	Reporte de trazabilidad. Evaluación de requisitos. Requisitos rechazados. Mapa de procesos. Especificación de requisitos de software. HU. Manual de usuario. Glosario de términos.

		Diseño de casos de prueba.
Desarrollador	Desarrollar una parte del sistema o completamente, incluyendo su diseño para que se ajuste a la arquitectura, implementación de pruebas unitarias y de integración para los componentes.	Código fuente.
Arquitecto	Diseñar la arquitectura del software, la cual incluye tomar las principales decisiones técnicas que condicionan globalmente el diseño y la implementación del proyecto.	Modelo de diseño. Arquitectura de software.
Probador	Identificar, definir, implementar y dirigir las pruebas necesarias, así como verificar y analizar sus resultados.	Diseño de casos de prueba.

La metodología AUP presenta algunos **beneficios** entre los que se encuentran:

- El personal sabe lo que está haciendo: no obliga a conocer detalles.
- Simplicidad: Apuntes concisos.
- Agilidad: Procesos simplificados de RUP.
- Herramientas independientes: A disposición del usuario.
- Fácil adaptación de este producto. (12)

Entre las técnicas ágiles que utiliza AUP se encuentra el Modelado ágil que permite encapsular los requisitos funcionales en HU o en Descripción de requisitos por procesos. La otra forma de encapsular los requisitos es por Casos de Uso. El proyecto Replicador de datos REKO en su versión 4.0 realiza la descripción de las HU. Por tanto, se hará uso de esta técnica para encapsular los requisitos funcionales.

1.4 Herramientas, tecnologías y lenguajes de desarrollo

El uso de herramientas en el mundo de la informática es imprescindible, pues permiten realizar programas y sistemas para el avance de este campo en la sociedad. Con el objetivo de realizar buenas prácticas en el uso de herramientas, es necesario contar con una tecnología adecuada para así guiar todo el proceso de investigación.

Para el desarrollo de la solución se continuará usando los lenguajes y herramientas que han sido empleadas desde la versión 1.0 del Replicador de datos REKO, debido a que se disminuye la curva de aprendizaje y el tiempo de desarrollo.

1.4.1 Lenguajes de programación

Java es un lenguaje de programación orientada a objetos, sólida y de código abierto. Basado en el lenguaje C++⁵ pero donde se eliminan muchas de las características de la OOP (Programación Orientada a Objetos, del inglés *Object Oriented Programming*) que se utilizan esporádicamente y que creaban frecuentes problemas a los programadores. Esta eliminación de causas de error y problemas de mantenimiento facilita y reduce el coste del desarrollo de software. Además posee mecanismos para garantizar la seguridad durante la ejecución comprobando, antes de ejecutar código, que este no viola ninguna restricción de seguridad del sistema donde se va a ejecutar. (13)

JavaScript es un lenguaje de programación que se puede utilizar para construir sitios Web y hacerlos más interactivos. Es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java. Es un lenguaje *Open Source*, por lo que cualquier persona puede utilizarlo sin comprar una licencia.(14, 15)

1.4.2 Herramientas de programación

Son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del computador u ordenador, funcione y pueda producir resultados.(16)

El **Eclipse STS (*Spring Tool Suite*) v 3.6.1** es una herramienta de código abierto y con licencia bajo los términos de la Licencia Pública Eclipse que incluye herramientas para el desarrollo del lenguaje Java. Cuenta con una consola que ofrece una visión en tiempo real y una gráfica de los parámetros de rendimiento de aplicaciones, que permite a los desarrolladores identificar y diagnosticar los problemas desde sus escritorios. (17)

1.4.3 Sistema de Gestión de Base de Datos

PostgreSQL v 9.1 es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Distribución de Software Berkeley, del inglés *Berkeley Software Distribution*) y con su código fuente disponible libremente. Es el SGBD de código abierto más potente del

⁵ Lenguaje de programación creado a partir del lenguaje de programación C y con mecanismos que permiten la manipulación de objetos.

mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi – hilos para garantizar la estabilidad del sistema. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.(18, 19)

1.4.4 Tecnologías de desarrollo

Java Enterprise Edition (JEE) v 1.7.0_71 es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.(20)

Dojo v 1.2.0 es un *framework*⁶ que contiene APIs (Interfaz de programación de aplicaciones, del inglés *Application Programming Interface*) y controles para soportar el desarrollo de aplicaciones web. Resuelve asuntos de usabilidad comunes como son la navegación y detección del buscador. Es utilizado en la creación y manejo de las vistas del software facilitando la utilización de componentes ya creados.(21)

Spring v 2.5 es un *framework* que proporciona un modelo de programación y configuración completo para aplicaciones de empresa moderna basada en Java en cualquier tipo de plataforma de despliegue. *Framework Open Source* que proporciona un marco de trabajo para el desarrollo de aplicaciones J2EE. Está basado en el uso de ficheros planos JavaBeans para la lógica de aplicación y archivos XML (Lenguaje de Marcas Extensible, del inglés *eXtensible Markup Language*) para la configuración. Se utiliza para la inyección de dependencias, o sea, para establecer la comunicación entre la capa de presentación y la capa modelo a través de parámetros. También se basa en un conjunto de módulos que proporcionan todo lo necesario para desarrollar una aplicación empresarial.(22)

1.4.5 Herramientas de modelado

Diversas son las aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del

⁶ Marco de trabajo.

software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos o detección de errores.

Visual Paradigm v 8 es una herramienta CASE (Ingeniería de Software Asistida por Computadora, del inglés *Computer Aided Software Engineering*) que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades, aunque también existe una alternativa libre y gratuita de este software. La versión Visual Paradigm UML (Lenguaje de Modelado Unificado, del inglés *Unified Modeling Language*) 6.4 Community Edition ha sido diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. (23)

1.4.6 Lenguaje de modelado

Lenguaje Unificado de Modelado v 2.0 (UML por sus siglas en inglés) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar elementos conceptuales como lo son procesos de negocio y funciones de sistema, además de sucesos concretos como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.(24)

1.4.7 Servidor de mensajería

Apache ActiveMQ v 5.9.0 es un popular, potente y rápido servidor de mensajería de código abierto que se encuentra distribuido bajo la licencia de Apache y soporta gran variedad de lenguajes de programación como Java, C⁷, C++, C#⁸, Python⁹, entre otros.(25)

1.4.8 Servidor de aplicación

Apache Tomcat v 7.0.47 es un servidor web multiplataforma que funciona como contenedor de servlets¹⁰. Dicho servidor es mantenido y desarrollado por miembros de la fundación y voluntarios independientes, los cuales tienen libre acceso al código fuente bajo los términos

⁷ Lenguaje de programación orientado a la implementación de sistemas operativos.

⁸ Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft.

⁹ Lenguaje de programación interpretado y multiparadigma.

¹⁰ Programa Java que se ejecuta dentro de un servidor Web. Reciben y atienden las solicitudes de los clientes web.

establecidos por la Fundación de Software Apache, del inglés *Apache Software Foundation*.(26)

1.5 Consideraciones parciales del capítulo

Una vez realizada la investigación científica y haber analizado los principales aspectos de este capítulo, se puede concluir que:

- Luego de estudiar los replicadores homólogos se determinó que no es factible la utilización de algún replicador de datos ya desarrollado, pues adquirir una herramienta de estas sería un riesgo, ya que no se tendría dominio del funcionamiento interno del sistema adquirido debido a la poca documentación existente.
- La metodología utilizada para el desarrollo de la solución fue justificada y descrita haciendo uso de bibliografía actualizada.
- Para el desarrollo de la solución se utilizaron las herramientas, tecnologías y lenguajes con los que se ha trabajado en el sistema desde su versión inicial, lo que trajo consigo que el tiempo de desarrollo fuera menor.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Una vez finalizada la fundamentación teórica de la investigación se procede a la descripción del proceso de desarrollo del módulo de configuración y monitorización de la réplica de estructura para el Replicador de datos REKO. En el presente capítulo se realiza la propuesta del sistema y se determinan los requisitos del mismo.

2.1 Descripción detallada del proceso a desarrollar

Por cada configuración, se registran las configuraciones independientes de las tablas que se desean replicar. Cada tabla puede ser configurada para replicar según la acción que se efectúe sobre ella: inserción, actualización y/o eliminación. Además pueden definirse filtros SQL para determinar por cada acción si se replicará o no el cambio. Sin embargo, al crear una configuración que contenga cambios de esquemas no se pueden establecer condiciones sobre las acciones DDL que favorecen el ambiente de réplica. Ante esta limitante se requiere brindar al Replicador de datos REKO una solución que permita adicionar filtros para las acciones DDL del proceso de réplica de estructura.

El sistema monitoriza todas las acciones del proceso de réplica de datos. Sin embargo el Monitor de Réplica no permite controlar el comportamiento de las acciones que contienen los cambios de esquemas. Ante esta limitante se requiere adicionar al Monitor de Réplica funcionalidades que permitan monitorizar en tiempo real las acciones del proceso de réplica de estructura.

2.2 Modelo de dominio

Es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis.(27)

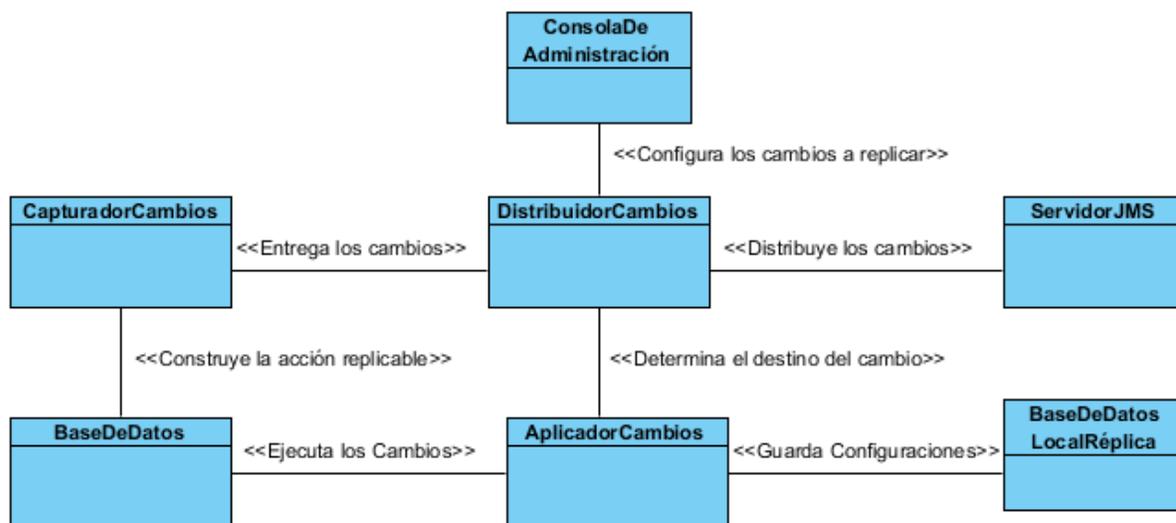


Figura 1 Modelo de dominio

Fuente: elaboración propia

2.2.1 Descripción de las clases de dominio

Consola web de administración: representa la interfaz del software. Permite realizar las configuraciones principales del software como el registro de nodos, configuración de las tablas y esquemas a replicar, y la monitorización del funcionamiento del software.

Capturador de cambios: concepto encargado de capturar los cambios realizados a la BD y de entregarlos al Distribuidor.

Distribuidor de cambios: concepto que determina para dónde debe ser enviado cada cambio realizado en la BD, los envía y se responsabiliza de que cada cambio llegue a su destino.

Aplicador de cambios: concepto que ejecutará sobre la Base de Datos los cambios que sean replicados hacia ella.

Base de Datos local de réplica: concepto que se refiere a guardar las configuraciones propias de la réplica, así como acciones sobre la BD que han provocado conflicto al aplicarse y transacciones que no han llegado a su destino.

Base de Datos: concepto que representa la BD que se está replicando. Los cambios ejecutados sobre ella serán enviados así como serán aplicados otros cambios provenientes de otros nodos de réplica.

Servidor JMS: concepto que representa un servidor de mensajería, utilizado como punto intermedio para la distribución de la información enviada. (28)

2.3 Especificación de requisitos

Describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento. Constituyen propiedades o restricciones determinadas de forma precisa que deben satisfacerse. Es una tarea de ingeniería de software que permite especificar las características operacionales del sistema, indicar su interfaz con otros elementos del sistema y establecer las restricciones que debe cumplir el software. (29)

2.3.1 Requisitos funcionales

Durante el proceso de captura de requisitos funcionales fueron detectados siete requisitos, los cuales fueron descritos a través de las HU. Los requisitos capturados son los siguientes:

RF 1 Adicionar filtros a la configuración de réplica de estructura.

RF 2 Modificar filtros de la configuración de réplica de estructura.

RF 3 Eliminar filtros de estructura de la configuración de réplica.

RF 4 Aplicar filtros de esquemas a los grupos de estructura replicable.

RF 5 Monitorizar el envío de los grupos del proceso de réplica de estructura.

RF 6 Monitorizar la recepción de los grupos del proceso de réplica de estructura.

RF 7 Monitorizar la aplicación de los cambios de estructura.

2.3.2 Requisitos no funcionales

En la tabla 3 se describen los requisitos no funcionales y sus atributos de calidad. (Para una mejor comprensión de los requisitos no funcionales y sus atributos de calidad ver Anexo 32). Hay requisitos de calidad que, con sus respectivos atributos, no han sido ejemplificados porque están incluidos en el sistema actual y su interpretación se describe en el documento Especificación de requisitos.

Tabla 3 Requisitos no funcionales

No	Nombre	Descripción
[RnF1.]	Usabilidad	<p>El módulo debe ser de fácil manejo para los usuarios que tengan niveles básicos de trabajo con ordenadores y en la Web. Tendrá una correcta estructuración y organización de la información, con el objetivo de que los usuarios tengan el conocimiento y dominio requerido del lugar del sistema en el que se encuentran y no se pierdan.</p> <ul style="list-style-type: none"> - Operabilidad: capacidad del producto que permite al usuario operarlo y controlarlo con facilidad. - Protección contra errores de usuarios: capacidad del producto para proteger a los usuarios de cometer errores.
[RnF2.]	Fiabilidad	<p>El módulo debe ser capaz de recuperarse ante fallos, además de notificar al usuario cualquier error o excepción que ocurra durante la ejecución de las acciones solicitadas.</p> <ul style="list-style-type: none"> - Tolerancia a fallos y recuperabilidad: capacidad del producto para operar según lo previsto en presencia de fallos de hardware o software. - Madurez: capacidad del producto para satisfacer las necesidades de fiabilidad en condiciones normales. - Disponibilidad: capacidad del producto de estar operativo y accesible para su uso cuando se requiere.
[RnF3.]	Mantenibilidad	<p>Mide la facilidad con que puede darse mantenimiento al producto con la finalidad de desarrollar nuevos requerimientos, corregir defectos y atender las demandas del entorno cambiante.</p> <ul style="list-style-type: none"> - Analizabilidad: facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar. - Modificabilidad: capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir

		<p>defectos o degradar el desempeño.</p> <ul style="list-style-type: none"> - Capacidad para ser probado: facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.
[RnF4.]	Portabilidad	<p>El módulo debe ser capaz de adaptarse de forma efectiva y eficiente a diferentes entornos.</p> <ul style="list-style-type: none"> - Adaptabilidad: capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso. - Instalabilidad: facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno. - Reemplazabilidad: grado en que un producto puede sustituir a otro producto de software especificado para el mismo propósito en el mismo entorno.
[RnF5.]	Compatibilidad	<p>El módulo debe ser compatible con los diferentes sistemas operativos, de forma que no sea una limitante para los usuarios a la hora de utilizarlo.</p> <ul style="list-style-type: none"> - Co-existencia: grado en que un producto puede realizar sus funciones necesarias de manera eficiente mientras comparte un entorno común y recursos con otros productos, sin impacto perjudicial en cualquier otro producto.
[RnF6.]	Adecuación funcional	<p>El desarrollo del módulo está guiado por las necesidades expresadas por parte de los proveedores de requisitos, dándole cumplimiento a sus especificaciones.</p> <ul style="list-style-type: none"> - Integridad funcional: grado en el que el conjunto de funciones cubre todas las tareas y objetivos del usuario especificado. - Corrección funcional: grado en que un producto o sistema proporciona los resultados correctos con el grado necesario

		de precisión.
[RnF7.]	Seguridad	<p>Grado de protección de los datos, software y plataforma de tecnología de posibles pérdidas, actividades no permitidas o uso para propósitos no establecidos previamente.</p> <ul style="list-style-type: none"> - Confidencialidad: la información manejada por el sistema está protegida de acceso no autorizado y divulgación. - 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. - No repudio: grado en que las acciones o eventos pueden ser probados al haber tenido lugar, por lo que los eventos o acciones no pueden ser repudiados más tarde.
[RnF8.]	Eficiencia en el rendimiento	<p>El módulo debe estar concebido para el consumo mínimo de recursos. Los clientes no necesitarán más de 128MB de RAM, lo suficiente para ejecutar un navegador Web.</p> <ul style="list-style-type: none"> - Comportamiento en el tiempo: grado en que los tiempos de respuesta, procesamiento y las tasas de rendimiento de un producto o sistema, al realizar sus funciones cumplen con los requisitos. - Utilización de recursos: grado en el que las cantidades y tipos de recursos utilizados por un producto o sistema, al realizar sus funciones cumplen con los requisitos.

Fuente:(30)

2.4 Propuesta de solución

Se extendió el componente Consola de Administración para adicionar nuevas funcionalidades al módulo de Configuración de Réplica y Monitor de Réplica. Para cumplir con los requisitos planteados es necesario extender el funcionamiento del componente Distribuidor para monitorizar los cambios enviados y recibidos. Para monitorizar la aplicación de los cambios se debe extender el funcionamiento del Aplicador.

En el módulo Configuración de Réplica al adicionar la configuración de réplica de estructura, se podrán establecer filtros para cada tipo de acción (creación, modificación, eliminación, creación

de un nuevo esquema, modificación del nombre de algún esquema y eliminación de algún esquema). Se podrán establecer filtros de nombres, prefijos y sufijos para todas las acciones y para la acción de modificación se podrá seleccionar un tipo de modificación para condicionar el proceso de captura. A continuación se muestran los diferentes tipos de filtros de estructura:

Nombre: condiciona la réplica solamente para el nombre de la estructura especificado.

Prefijo: condiciona la réplica solamente para aquellas estructuras que comiencen con los datos especificados.

Sufijo: condiciona la réplica solamente a seleccionar las tablas que terminan con el sufijo especificado.

Tipo de acción: especifica el tipo de acción ALTER que se va a replicar.

En la ventana que se muestra al adicionar los filtros se podrán establecer las diferentes condiciones. Los operados permitirán condicionar el comportamiento de las restricciones. Si se determina el operador “=” se replicará solamente los cambios sobre las estructuras establecidas, mientras que si se determina el operador “<>” se replicará toda la información exceptuando las estructuras establecidas.

El Monitor de Réplica visualizará el comportamiento del proceso de réplica de estructura. De las acciones de estructura replicable que se envían y reciben se mostrarán los datos necesarios, como son:

Dirección: indica la trayectoria del cambio, si el cambio entra o sale.

Fecha: muestra la fecha y hora en que la acción de enviar, recibir o aplicar es realizada.

Remitente: contiene el identificador del nodo que originó el cambio.

Destinos: contiene el identificador del nodo que procesa el cambio, además del estado en que se encuentra (enviando, enviada, recibiendo o recibida, aplicada).

SQL: sentencia SQL de la acción correspondiente al cambio a ejecutar en la BD.

Esquema: el esquema que se afecta con el cambio.

Tabla: la tabla que se afecta con el cambio.

Tipo de acción: el tipo de acción a ejecutar en la BD.

Observación: información adicional que se desee mostrar.

2.5 Historias de Usuario

Las HU son un instrumento para el levantamiento de requerimientos para el desarrollo de un software. Son descripciones cortas de una necesidad de un cliente del software que se esté desarrollando, que ha emergido con la aparición de los nuevos marcos de trabajo de desarrollo ágil, como por ejemplo las metodologías Scrum, AUP o las diferentes técnicas que comprenden el XP (Programación Extrema, del inglés *eXtreme Programming*).⁽³¹⁾

Las HU quedan estructuradas de la siguiente manera:

Nombre: nombre descriptivo de la HU.

Prioridad: grado de prioridad que le asigna el cliente a la HU en dependencia de sus necesidades. Los valores que puede tomar son (Alta, Media o Baja).

Estimación: unidades de tiempo estimadas por el equipo de desarrollo para darle cumplimiento a la HU.

Iteración: número de la iteración en la cual será implementada la HU.

Descripción: descripción simple que brinda el cliente sobre lo que debe hacer la funcionalidad en cuestión.

Adicionalmente a cada HU se le asigna un número para facilitar su identificación por parte del equipo de desarrollo y se le puede incluir alguna información adicional que pueda ser útil para la comprensión de la misma. El tiempo de desarrollo ideal para una HU es entre una y tres semanas.

Uno de los beneficios de las HU es que pueden ser escritas en diferentes niveles de detalle. Es posible escribir historias que cubren múltiples funcionalidades.

El autor del trabajo selecciona las HU como técnica de descripción de requisitos por ser una de las técnicas más sencillas y completas, además de seguir la línea de trabajo del proyecto de desarrollo del Replicador de datos REKO.

2.5.1 Descripción de las HU

Durante el diseño de la propuesta de solución se identificaron siete HU que responden a las diferentes funcionalidades solicitadas por el cliente y presentan una descripción para que el equipo de desarrollo conozca su posterior implementación. A continuación se describen algunas de las HU identificadas. (Para consultar las restantes dirigirse a los Anexos del 4 al 7).

Tabla 4 HU1 Adicionar filtros a la configuración de réplica de estructura

Número: HU 1	Nombre del requisito: Adicionar filtros a la configuración de réplica de estructura.
Programador: Alejandro García Santiesteban	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 15 días
Riesgo en Desarrollo: Fallos eléctricos.	Tiempo Real: 120 horas
<p>Descripción: Al hacer clic en “Nueva Configuración” del módulo de Configuración de Réplica, específicamente en la opción “Esquemas”, el sistema debe brindar la posibilidad de “Adicionar filtros” de estructura para cada tipo de acción. El sistema debe almacenar filtros por nombre, prefijo o sufijo de una tabla para el esquema seleccionado, en el caso de las acciones de “Creación”, “Modificación” y “Eliminación”. Para la acción “Modificación” se debe permitir además filtrar por los diferentes tipos de acciones de modificación. De igual forma, el sistema debe establecer filtros por nombre, prefijo o sufijo de esquemas para las acciones de la opción “Configurar la réplica de esquemas”. El sistema debe permitir la adición y eliminación de condiciones, y para cada condición se debe brindar la posibilidad de indicar los operadores de igualdad y desigualdad, para restringir la réplica.</p>	
Observaciones: Ningún campo vacío.	
<p>Prototipo de interfaz:</p> 	

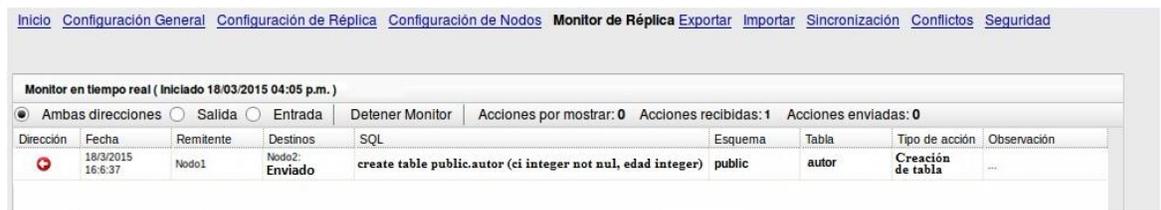
Tabla 5 HU4 Aplicar filtros de esquemas a los grupos de estructura replicable

Número: HU 4	Nombre del requisito: Aplicar filtros de esquemas a los grupos de

estructura replicable.	
Programador: Alejandro García Santiesteban	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 15 días
Riesgo en Desarrollo: Fallos eléctricos.	Tiempo Real: 120 horas
Descripción: El sistema debe aplicar los filtros de estructura a los grupos de estructura replicables antes de ser enviados. Cuando se inicie el envío de los grupos de estructura replicable, el sistema debe acceder a las configuraciones de réplica de estructura y determinar las acciones de estructura permitidas por los filtros, para enviar solamente aquellas acciones que se correspondan a los filtros definidos por el usuario.	
Observaciones: Ningún campo vacío.	
Prototipo de interfaz:	

Tabla 6 HU5 Monitorizar la recepción de los grupos del proceso de réplica de estructura

Número: HU 5	Nombre del requisito: Monitorizar el envío de los grupos del proceso de réplica de estructura.
Programador: Alejandro García	Iteración Asignada: 1

Santiesteban.	
Prioridad: Alta	Tiempo Estimado: 15 días
Riesgo en Desarrollo: Fallos eléctricos.	Tiempo Real: 40 horas
<p>Descripción: En el módulo de “Monitor de Réplica”, al realizarse el envío de los cambios del proceso del proceso de réplica de estructura, el sistema debe publicar el evento Enviado, el cual confirma al usuario que el envío de las acciones fue realizado con éxito. El sistema debe mostrar además los detalles del envío. De las acciones que son enviadas se debe mostrar la dirección de la réplica, la fecha en que se envía, el nodo que envía, el destino, el estado (enviando o enviado), la sentencia SQL, el esquema y la tabla que se afecta con el cambio, así como el tipo de acción.</p>	
<p>Observaciones: En tiempo real solo se podrá observar el evento si en ese momento el usuario se encuentra en la vista “Monitor en tiempo real”.</p>	
<p>Prototipo de interfaz:</p>  <p>The screenshot shows a web interface for monitoring replication. At the top, there are navigation links: Inicio, Configuración General, Configuración de Réplica, Configuración de Nodos, Monitor de Réplica, Exportar, Importar, Sincronización, Conflictos, and Seguridad. Below the links, there is a section titled 'Monitor en tiempo real (Iniciado 18/03/2015 04:05 p.m.)'. This section includes a table with columns: Dirección, Fecha, Remitente, Destinos, SQL, Esquema, Tabla, Tipo de acción, and Observación. The table contains one row with a red plus icon in the 'Dirección' column, a date of 18/3/2015 16:6:37, 'Nodo1' as the sender, 'Nodo2: Enviado' as the destination, a SQL statement 'create table public.autor (ci integer not nul, edad integer)', 'public' as the schema, 'autor' as the table, and 'Creación de tabla' as the action type.</p>	

2.6 Descripción de la arquitectura

En el ámbito taller se enseñan tanto la teoría como la práctica del diseño arquitectónico. La arquitectura es necesaria para comprender el sistema, organizar el desarrollo, fomentar la reutilización y hacer evolucionar el sistema. Para definirla es necesario seleccionar y combinar patrones.(7)

2.6.1 Arquitectura basada en componentes

Un componente es una *“unidad de composición de aplicaciones software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio”*.(32)

El desarrollo basado en componentes permite alcanzar un mayor nivel de reutilización de software y que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados, también simplifica el mantenimiento del sistema, entre otras cuestiones.(33)

Con las definiciones expuestas se pretende describir de forma general la arquitectura del software Replicador de datos REKO la cual es basada en componente, teniendo en cuenta que está compuesto por diferentes módulos, los cuales pueden ser separados o añadidos sin que perjudique a los demás. En el presente desarrollo se han extendido los componentes Consola web de administración, Capturador de cambios, Aplicador y Distribuidor que se muestran en la figura 5, con el objetivo de que el software permita realizar la configuración y monitorización de la réplica de estructura.

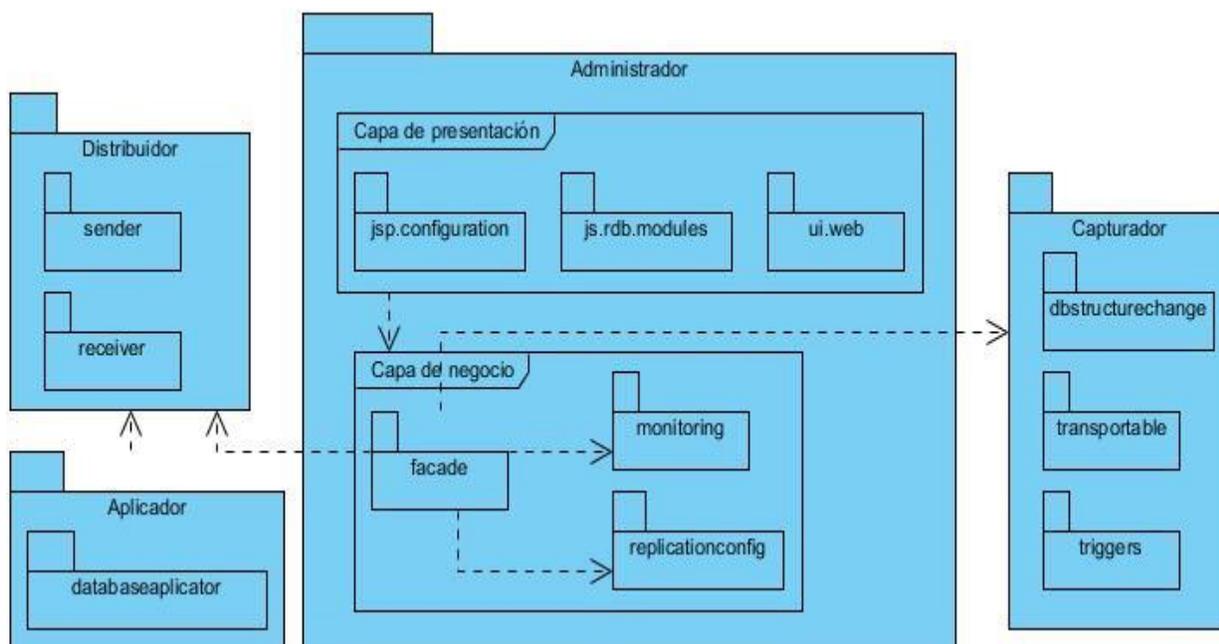


Figura 2 Vista que muestra los principales componentes del Replicador de datos REKO

Fuente: elaboración propia

2.6.2 Arquitectura en capas

El componente Administrador responde a un modelo multicapas. Cada capa tiene funcionalidades y objetivos específicos.

La capa de presentación está formada por la lógica de aplicación, que prepara datos para su envío a la capa de cliente y procesa solicitudes desde la capa de cliente para su envío a la lógica de negocios del servidor. La lógica en la **capa de presentación** está formada

normalmente por componentes de J2EE¹¹ como, por ejemplo, Java Servlet¹² o los componentes de JSP (del inglés *Java Server Pages*) que preparan los datos para enviarlos en formato HTML (Lenguaje de Marcas de Hipertexto del inglés *HyperText Markup Language*), o XML (Lenguaje de Marcas Extensible, del inglés *eXtensible Markup Language*). Es la capa que el usuario ve en su ordenador y se comunica únicamente con la capa de negocio. La **capa de negocio** consiste en la lógica que realiza las funciones principales de la aplicación: procesamiento de datos, implementación de funciones de negocios, coordinación de varios usuarios y administración de recursos externos como, por ejemplo, bases de datos o sistemas heredados. Esta capa suele estar formada por componentes firmemente acoplados que se ajustan al modelo de componentes distribuidos de J2EE como, por ejemplo, los objetos Java, los componentes EJB¹³ o los *beans*¹⁴ conducidos mediante mensajes. Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina además como lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse.(34, 35)

2.6.3 Patrones de diseño

Describen un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse varias veces sin tener que hacer dos veces lo mismo. Un patrón de diseño identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades.(32, 36)

Patrón DAO (Objeto de Acceso a Datos, del inglés *Data Access Object*)

Es el encargado de separar la aplicación de la BD, de tal forma que el cliente no tenga que preocuparse por cómo se realiza la conexión, solamente centrándose en manejar los datos que necesitase. El uso de este patrón hace que no haya código de acceso a datos en otro lugar que no sea en las clases DAO. Oculta completamente los detalles de implementación de la fuente de datos a sus clientes, de esta manera cuando la capa de lógica de negocio necesite interactuar con la base de datos, ésta utilizará los métodos ofrecidos por DAO. (Ver Anexo 1)
(37)

¹¹ Arquitectura multicapa para implementar aplicaciones de tipo empresarial y aplicaciones basadas en la Web, del inglés *Java 2 Enterprise Edition*.

¹² Aplicaciones Java que corren en un entorno de servidor.

¹³ Clase java que implementa determinada interface, del inglés *Enterprise JavaBeans*

¹⁴ Componente que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje Java.

Patrones GRASP (Del inglés *General Responsibility Assignment Software Patterns*)

Describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. (38)

✓ Experto: propone asignar una responsabilidad a la clase que tiene la información necesaria para realizar estas acciones. Para citar algún ejemplo de patrón Experto se puede mencionar la *ConfigurationManager*, atendiendo a que cuenta con la información necesaria para gestionar las configuraciones de réplica, mientras que la *MonitoringManager* es la responsable solamente de monitorizar el proceso de réplica.

El uso de este patrón aporta los siguientes beneficios:

- La información permanece encapsulada, puesto que los objetos utilizan su propia información para llevar a cabo las tareas. Esto soporta un Bajo Acoplamiento, lo que favorece el hecho de tener sistemas más robustos y de fácil mantenimiento.
- Las clases son más cohesivas, ya que el comportamiento se distribuye entre las clases que contienen la información requerida, brindando así una Alta Cohesión.(38)

✓ Creador: el propósito fundamental es encontrar un creador, el cual se debe conectar con el objeto producido en cualquier evento. Se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando:

- B contiene a A.
- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor).
- B usa a A.(38)

Se muestra como ejemplo la clase *DBConfiguration* la cual es la encargada de crear las configuraciones de esquema porque:

- *DBConfiguration* agrega los objetos de *SchemaConfig*.
- *DBConfiguration* es una agregación (o composición) de *SchemaConfig*.
- *DBConfiguration* almacena a *SchemaConfig*.
- *DBConfiguration* tiene los datos de inicialización de *SchemaConfig* (datos que requiere su constructor).

- *DBConfiguration* usa a *SchemaConfig*. (Ver Anexo 2)
- ✓ Controlador: su uso garantiza que los procesos del dominio sean manejados por la capa lógica del negocio y no por la interfaz, sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocio debe estar separada de la capa de presentación, lo que aumenta la reutilización de código y permite a la vez tener un mayor control.(38)

Se evidencia con la clase *ManageSchemaActionFilterController*.

- ✓ Alta cohesión: el objetivo de este patrón es asignar una responsabilidad de modo que la cohesión siga siendo alta. Explica cómo mantener la complejidad dentro de límites manejables. Dentro del sistema se aplica para mantener una alta coherencia entre las clases del mismo, por ejemplo, cada vista accede a los datos del controlador a través de una instancia recibida como parámetro.(39)

- ✓ Bajo acoplamiento: explica como asignar una responsabilidad para mantener un bajo acoplamiento. Se aplica para dar soporte a una dependencia escasa y a un aumento de la reutilización. Dentro del sistema está presente en las clases del modelo, las cuales son completamente independientes de la vista y el controlador, lo que garantiza una amplia reutilización de las mismas.(39)

Por ejemplo, para monitorizar el envío de las acciones del proceso de réplica de estructura, la clase *SenderManager* se comunica con la menor cantidad de clases posibles para publicar la acción, presentando así un bajo acoplamiento. (Ver Anexo 3)

Patrones GoF (Pandilla de los Cuatro, del inglés *Gang of Four*)

Pueden ser vistos como complementos de los patrones GRASP y en ocasiones se puede encontrar una contraposición entre este tipo de patrones, e incluso, podría inferirse que algunos patrones GoF son variantes de los patrones GRASP, es por ello que la decisión de utilizar uno u otro debe tomarse con precaución y aplicarse sólo en el ámbito necesario. Los patrones GoF proponen soluciones a ciertos problemas a la hora de construir algún software complejo.

- Fachada (Facade): Proporciona una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. Dentro del sistema el patrón fachada se utiliza para que el usuario acceda a la gran

mayoría de las interfaces del sistema a través de la interfaz principal del módulo. Este patrón especifica que puede darse el caso de vistas a las que se deba acceder a través de otras vistas.(39)

El Replicador de datos REKO tiene el componente *facade* que aprovisiona estas ventajas, la clase *TransactionProcessor* del paquete *facade* ofrece un punto de acceso para acceder a las configuraciones de esquemas.

2.7 Modelo de diseño

El modelo de diseño se encuentra más cerca de la solución que se desea obtener y adquiere alguna de las entidades reflejadas en el modelo de dominio para convertirlas en clases. Sirve como abstracción del modelo de implementación y el código fuente. Representa la solución a las HU definidas y es utilizado como entrada a las tareas de implementación.(40)

2.7.1 Diagramas de paquetes

Un paquete es un mecanismo utilizado para agrupar elementos de UML. Permite organizar los elementos modelados con UML, facilitando de ésta forma el manejo de los modelos de un sistema complejo. Define un espacio de nombres, dos elementos de UML pueden tener el mismo nombre, con tal y estén en paquetes distintos.(41)

A continuación se muestran los diagramas de paquetes correspondientes a los RF 1 y RF 5. (Para analizar los demás Diagramas de Paquetes (DP) se pueden consultar los Anexos del 8 al 10).

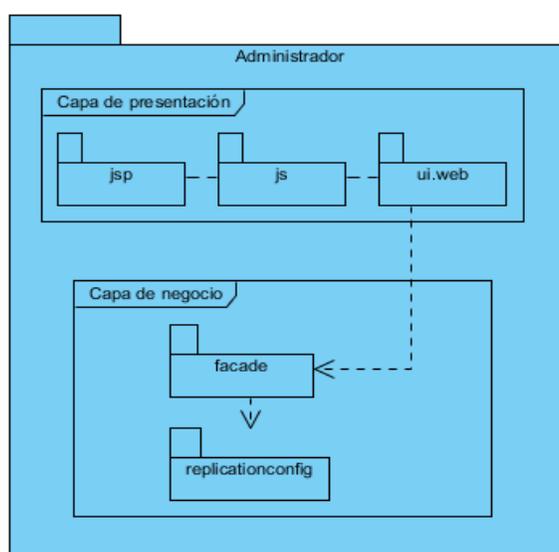


Figura 3 Diagrama de paquetes de los RF 1, RF 2 y RF 3.

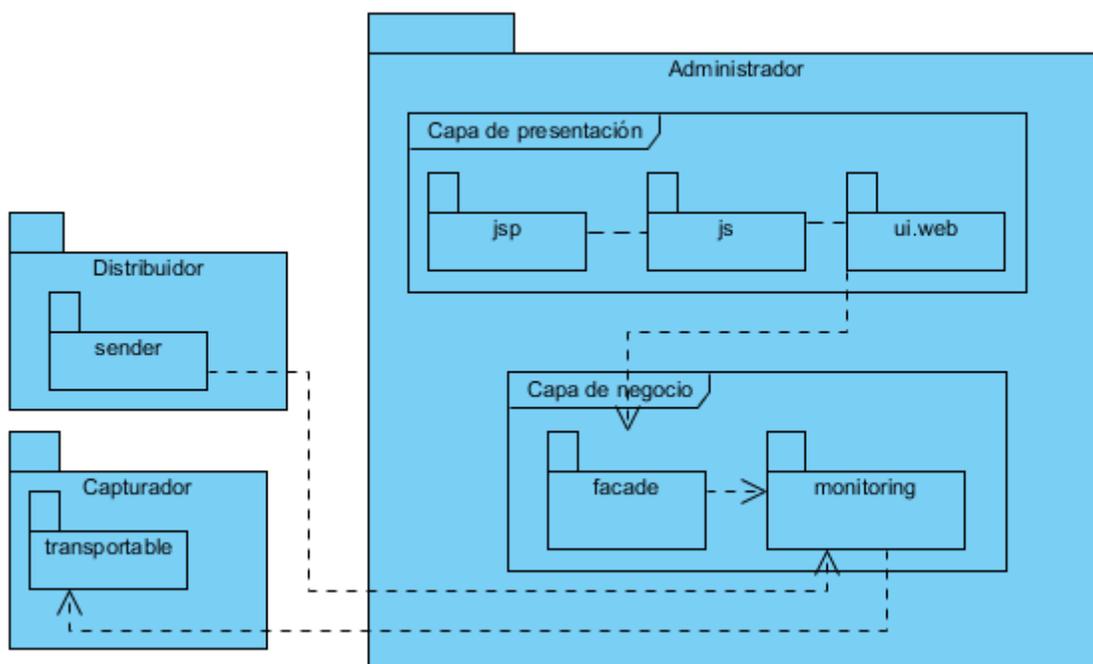


Figura 4 Diagrama de paquetes del RF 5

2.7.2 Diagrama de clases

Los diagramas de clases (DC) describen de forma visual los posibles sistemas. El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para completar su tarea. La clase define el ámbito de definición de un conjunto de objetos. Un diagrama de clases y un diagrama de objetos son las alternativas de representación de modelos de objetos, aunque los diagramas de clases prevalecen más que los de objetos. Normalmente se puede construir un diagrama de clases y ocasionalmente uno de objetos para ilustrar las estructuras de datos más complejas. Contienen íconos que representan las clases.(42)

A continuación se muestran los diagramas de clases correspondientes a los diagramas de paquetes expuestos anteriormente. (Para consultar los demás dirigirse a los Anexos del 11 al 13).

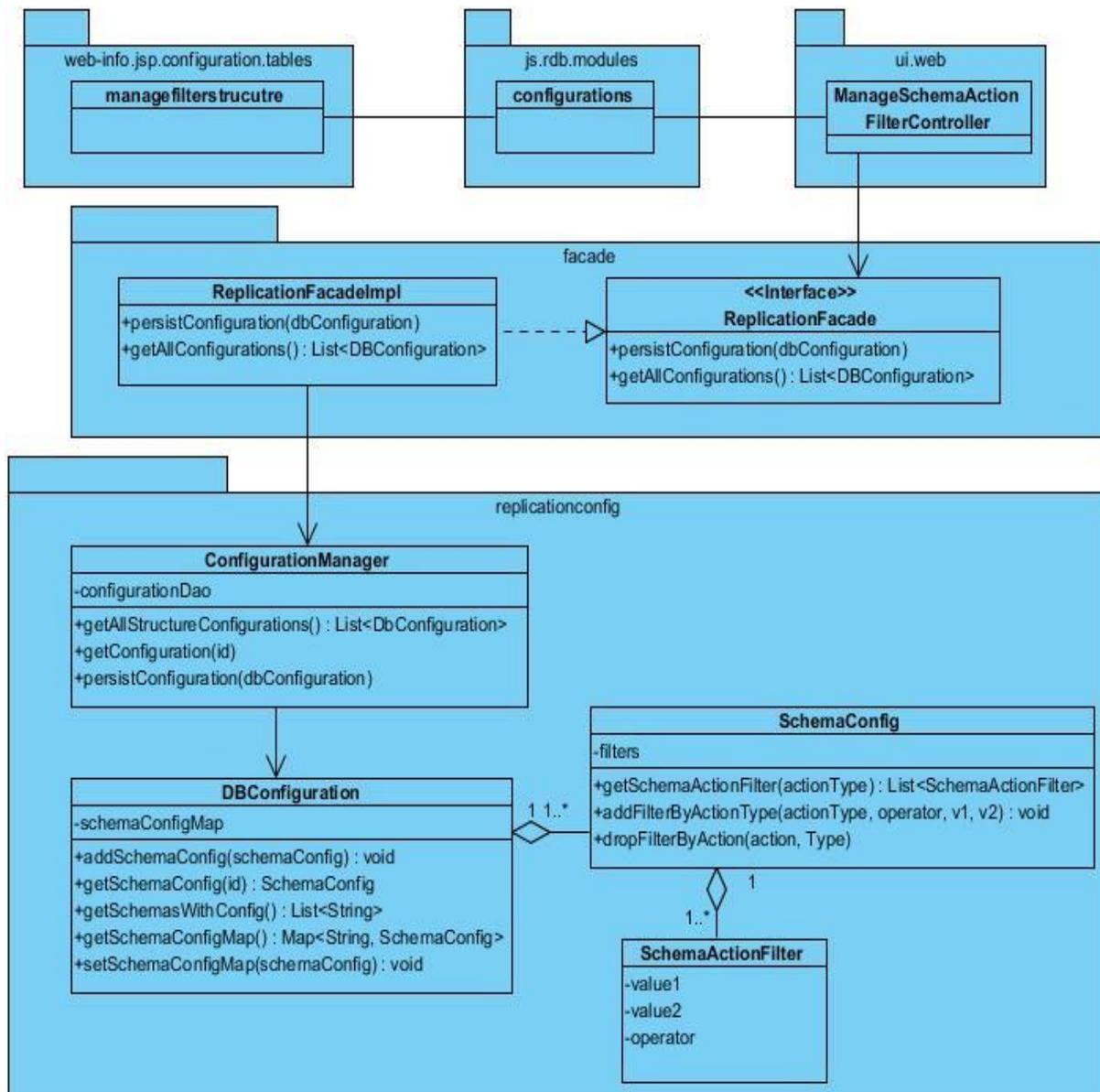


Figura 5 Diagrama de clases del Diagrama de paquetes de los RF 1, RF 2 y RF 3.

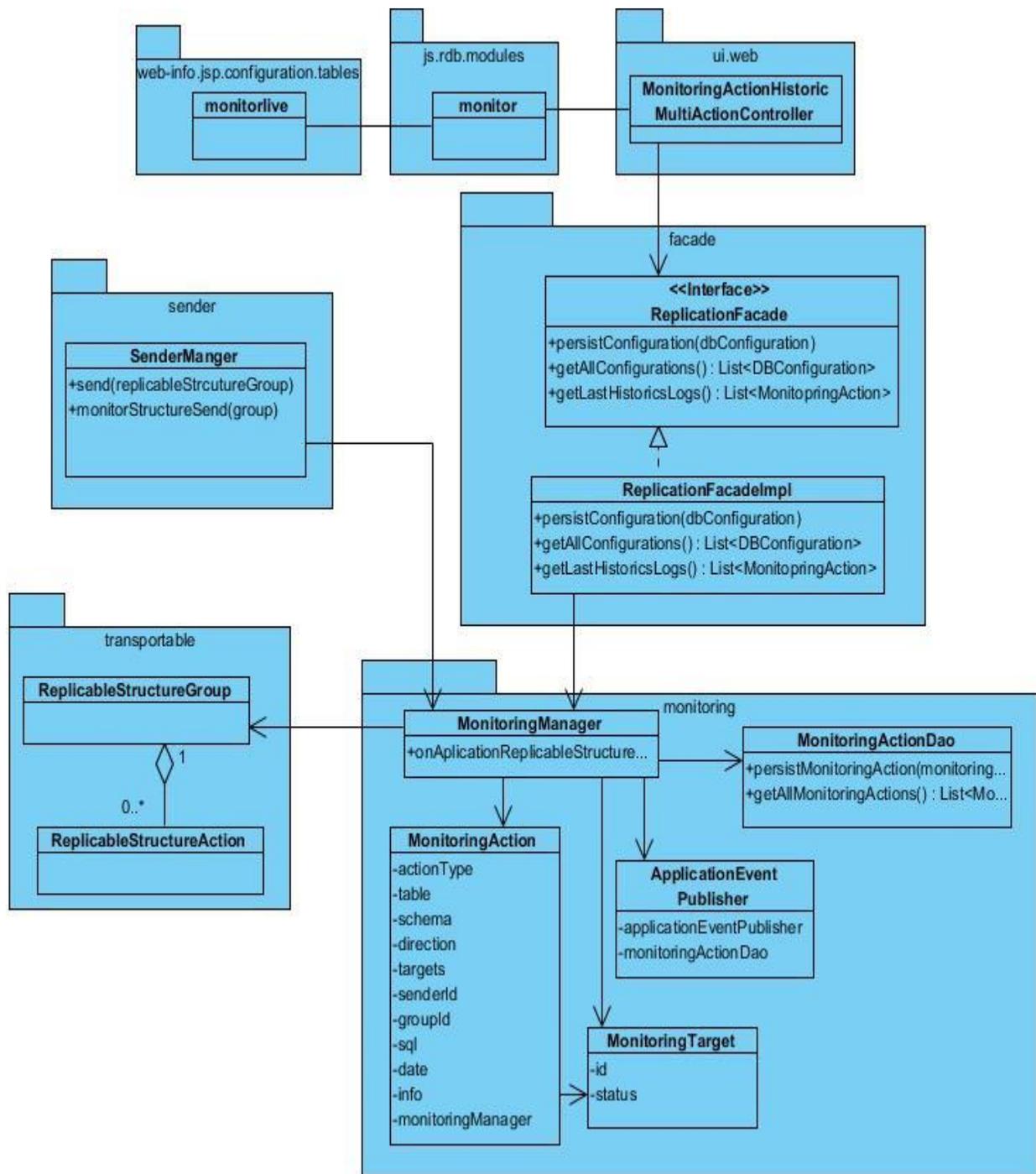


Figura 6 Diagrama de clases del Diagrama de paquetes del RF 5

2.7.3 Descripción de las clases

A continuación se muestra la descripción de las clases correspondientes a los diagramas de clases expuestos anteriormente.

Tabla 7 Descripción textual de la clase ConfigurationManager

Descripción de la Clase ConfigurationManager	
Nombre: ConfigurationManager	
Tipo de clase: Controladora	
Atributos	Tipos
configurationDao	private
exportConfig	private
Responsabilidades	
Nombre:	persistConfiguration(dbConfiguration: cu.uci.dbreplication.replicationconfig.DBConfiguration)
Descripción:	Guarda la configuración con los parámetros establecidos.
Nombre:	getConfiguration(id: String): cu.uci.dbreplication.replicationconfig.DBConfiguration
Descripción:	Obtiene la configuración correspondiente al identificador especificado por parámetro.
Nombre:	getAllConfigurations(): java.util.List< cu.uci.dbreplication.replicationconfig.DBConfiguration >
Descripción:	Obtiene todas las configuraciones que se tienen almacenadas.
Nombre:	getAllStructureConfigurations java.util.List< cu.uci.dbreplication.replicationconfig.DBConfiguration >
Descripción:	Obtiene todas las configuraciones de réplica de estructura que se tienen almacenadas.

Tabla 8 Descripción textual de la clase SenderManager

Descripción de la Clase SenderManager	
Nombre: SenderManager	
Tipo de clase: Controladora	
Atributos	Tipos
monitoringManager	private
exportConfig	private
Responsabilidades	
Nombre:	afterPropertiesSet()
Descripción:	Método a implementar de la interfaz InitializingBean. Inicia las tareas de ejecución periódica. Garantiza que en caso de interrupciones, las agrupaciones pendientes sean enviadas antes de las nuevas.
Nombre:	send (replicableStructureGroup: cu.uci.dbreplication.transportable.ReplicableStructureGroup)

Descripción:	Mientras existen agrupaciones las envía.
Nombre:	applyFilterToStructure (aggrupation: cu.uci.dbreplication.transportable.ReplicableStructureGroup)
Descripción:	Aplica los filtros de destinatario al ReplicableStructureGroup que se tienen almacenados en las configuraciones de réplica.
Nombre:	monitorSend(replicableStructureGroup: cu.uci.dbreplication.transportable.ReplicableStructureGroup, targets : java.util.Set<String>)
Descripción:	Publica el envío de los RSG hacia los objetivos especificados con sus detalles en el módulo Monitor de Réplica.

Tabla 9 Descripción textual de la clase MonitoringManager

Descripción de la Clase MonitoringManager	
Nombre:	MonitoringManager
Tipo de clase:	Controladora
Atributos	Tipos
applicationEventPublisher	private
Responsabilidades	
Nombre:	onSendingStructureGrup(group: cu.uci.cbreplication.transportable.ReplicableStrcutureGroup)
Descripción:	Publica el evento “Enviado” en el Monitor de Réplica.
Nombre:	onReceivingStructureGrup(group: cu.uci.cbreplication.transportable.ReplicableStrcutureGroup)
Descripción:	Publica el evento “Recibido” en el Monitor de Réplica.
Nombre:	onAplicationStructureGrup(group: cu.uci.cbreplication.transportable.ReplicableStrcutureGroup)
Descripción:	Publica el evento “Aplicado” en el Monitor de Réplica.

Se realizó la descripción de otras clases en donde se recurrió a métodos ya implementados y en otros casos fue necesaria la implementación de nuevas operaciones. (Ver Anexos del 14 al 27)

2.8 Modelo de despliegue

Muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los enlaces de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos. El modelo de despliegue se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos.(43)



Figura 7 Diagrama de despliegue

2.9 Consideraciones parciales del capítulo

Durante el desarrollo de la propuesta de solución y después de un análisis de las características del Replicador de datos REKO, se puede concluir que:

- Se hace necesario extender las funcionalidades de los componentes Consola de Administración, Distribuidor y Aplicador de cambios para permitir la creación de configuraciones y monitorización del proceso de réplica de estructura, debido a que hasta el momento su funcionamiento estaba limitado a la réplica de datos.
- El desarrollo de las nuevas funcionalidades para extender los módulos de Configuración y Monitor de Réplica está dividido en dos partes fundamentales: una encargada de la aplicación de los filtros de estructura y la otra a la monitorización del proceso de réplica de estructura.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS AL SISTEMA

En el presente capítulo, una vez concluida la propuesta de solución, se procede a la implementación de las clases y a realizarle las pruebas al sistema. Se describen los componentes extendidos para el desarrollo de los módulos “Configuración de Réplica” y “Monitor de Réplica” y se determina si las funcionalidades implementadas cumplen con las características establecidas y con las descripciones de las HU anteriormente expuestas.

3.1 Implementación de la solución

Durante la fase de implementación se vinculan las HU con las tareas concretas de desarrollo. Su objetivo fundamental es desarrollar la arquitectura y el sistema como un todo.

3.1.1 Diagramas de componentes

Los diagramas de componentes muestran las organizaciones y dependencias lógicas entre componentes de software, sean estos de código fuente, binarios, archivos, bibliotecas cargadas dinámicamente o ejecutables. Modelan la vista estática de un sistema.(44)

Según Madeline Tracy García Saavedra, *“Los diagramas de componentes describen los elementos físicos y sus realizaciones en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros”*.(45)

Como el Replicador de datos REKO presenta una arquitectura basada en componentes, se emplearon los subsistemas de implementación para realizar los diagramas de componente, dividiéndolo en partes más pequeñas que puedan ser integradas y probadas de forma separada.

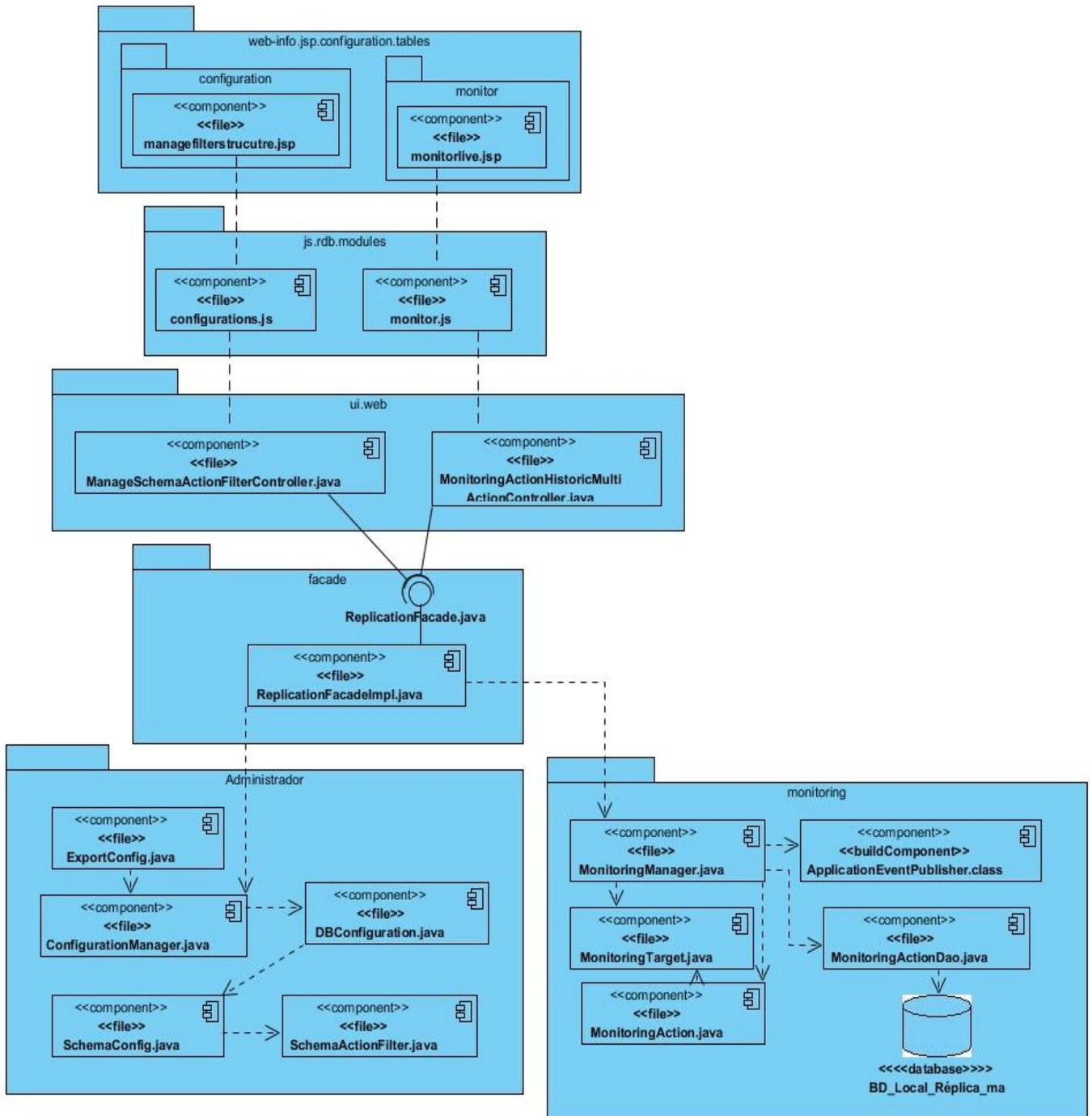


Figura 8 Diagrama de componentes para la Consola de Administración

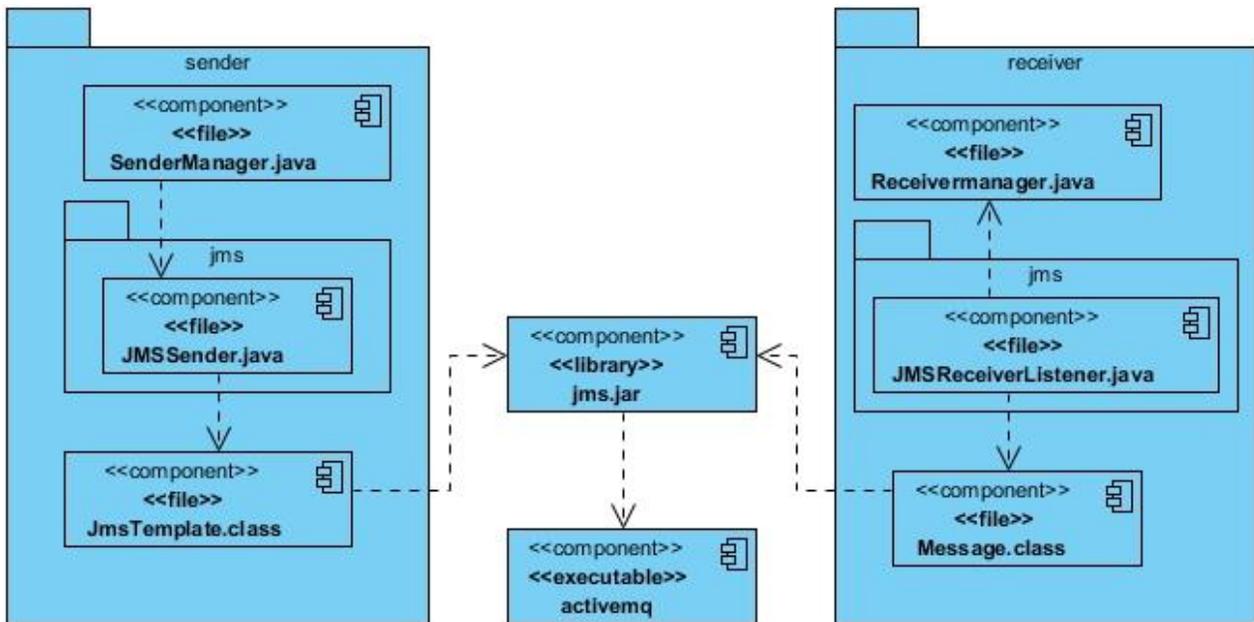


Figura 9 Diagrama de componentes para el Distribuidor de cambios

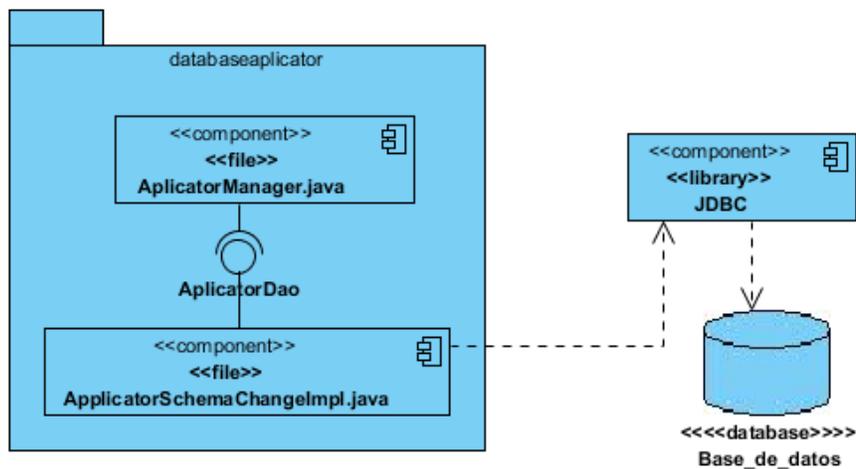


Figura 10 Diagrama de componentes para el Aplicador de cambios

3.2 Pruebas al sistema

Las pruebas de software consisten en una serie de acciones en las que el sistema es ejecutado bajo condiciones y requisitos específicos. Los resultados son observados y registrados realizando una evaluación de la calidad del producto para ser entregado.(46)

Una vez concluida la disciplina de implementación y con el objetivo de validar el correcto funcionamiento de los requisitos implementados se realizan pruebas de caja negra y aceptación.

3.2.1 Pruebas de Caja negra

Son pruebas realizadas sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Es decir, a través de ellas se pretende demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y que se produce un resultado correcto. En general, este método se centra principalmente en los requisitos funcionales del software. Dentro de los métodos de pruebas de caja negra se encuentra el de partición equivalente, el cual será usado para la realización de las pruebas al sistema.(47)

3.2.2 Partición Equivalente

Se define como un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.(48)

A continuación se muestran los casos de prueba realizados para las HU que fueron mostradas en el capítulo anterior, si se desea consultar los restantes casos de prueba dirigirse a los Anexos del 27 al 32.

Tabla 10 Prueba de partición equivalente sobre la HU1 Adicionar filtros a la configuración de réplica de estructura

Caso de prueba	
Código: HU1	Historia de usuario: 1
Nombre: Adicionar filtros a la configuración de réplica de estructura.	
Descripción: Prueba para la funcionalidad de adicionar filtros a la configuración de réplica de esquemas.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” seleccionando al menos un tipo de acción.	
Variables: “Esquema”, “Replicar por” y “Filtros”.	
Clase de equivalencia válida: Adicionar filtros a la configuración de réplica de estructura correctamente.	

<p>Entradas/Pasos de Ejecución: Todas las variables tienen sus valores de entrada correctos. Al seleccionar el esquema sobre el cual desea replicar y seleccionar el tipo de acción que desea, se procede a seleccionar la opción “Adicionar filtro”, donde el sistema muestra la ventana “Filtros de Estructura” para que el usuario determine las condiciones de la réplica para el tipo de acción seleccionada. El usuario establece las condiciones deseadas mediante los diferentes filtros (nombre, prefijo, sufijo y/o tipo de acción). El sistema almacena los filtros y muestra la pantalla “Nueva configuración”, modificando el enlace “Adicionar filtro” por los enlaces “Editar filtros” y “Eliminar filtros” seguido del enlace “Usuarios”.</p>
<p>Resultado Esperado: El sistema almacena los filtros en la configuración adicionada o cargada.</p>
<p>Clase de equivalencia válida: Adicionar filtros a la configuración de réplica de estructura incorrectamente.</p>
<p>Entradas/Pasos de Ejecución: La variable “Filtros” no obtuvo el valor esperado, o sea, se quedó algún campo vacío. Al seleccionar el esquema sobre el cual desea replicar y seleccionar el tipo de acción que desea, se procede a seleccionar la opción “Adicionar filtro”, donde el sistema muestra la ventana “Filtros de Estructura” para que el usuario determine las condiciones de la réplica para el tipo de acción seleccionada. El usuario establece las condiciones deseadas mediante los diferentes filtros (nombre, prefijo, sufijo y/o tipo de acción), dejando algún campo vacío y presionando el botón “Aceptar”. El sistema muestra al usuario un mensaje de alerta informando que faltan datos por ingresar manteniendo la misma pantalla.</p>
<p>Resultado Esperado: El sistema muestra un mensaje de alerta informando que faltan datos por ingresar.</p>
<p>Evaluación de la Prueba: La prueba fue realizada con éxito.</p>

En el proceso de pruebas de caja negra al módulo “Monitor de Réplica” se aplicaron tres iteraciones de prueba con el objetivo de verificar el correcto funcionamiento del módulo y su cumplimiento con las funcionalidades. Se obtuvo un total de tres no conformidades las cuales se explican a continuación:

1ra Iteración:

Durante la primera iteración de pruebas al módulo “Monitor de Réplica”, se detectó una no conformidad, la cual se describe a continuación:

Al enviarse una acción los resultados obtenidos son los siguientes:

Tabla 11 Resultados obtenidos en la primera iteración de pruebas sobre el módulo "Monitor de Réplica"

Funcionalidades del Monitor de Réplica	Resultado	
	Tiempo Real	Histórico
Monitorizar el envío de los grupos del proceso de réplica de estructura	Satisfactorio, se mostró el evento "Enviado".	Satisfactorio, se mostró el evento "Enviado".
Monitorizar la recepción de los grupos del proceso de réplica de estructura.	Satisfactorio, se mostró el evento "Recibido".	Satisfactorio, se mostró el evento "Recibido".
Monitorizar la aplicación de los cambios de estructura.	No Satisfactorio, no se mostró el evento "Aplicado".	Satisfactorio, se mostró el evento "Aplicado".

2da Iteración

Durante la segunda iteración de pruebas al módulo "Monitor de Réplica", se corrigió la no conformidad detectada en la iteración anterior, sin embargo se detectaron dos no conformidades, las cuales se describen a continuación:

Al enviarse dos acciones los resultados obtenidos son los siguientes:

Tabla 12 Resultados obtenidos en la segunda iteración de pruebas sobre el módulo "Monitor de Réplica"

Funcionalidades del Monitor de Réplica	Resultado	
	Tiempo Real	Histórico
Monitorizar el envío de los grupos del proceso de réplica de estructura	Satisfactorio, se mostró el evento "Enviado".	Satisfactorio, se mostró el evento "Enviado".
Monitorizar la recepción de los grupos del proceso de réplica de estructura.	Satisfactorio, se mostró el evento "Recibido".	Satisfactorio, se mostró el evento "Recibido".
Monitorizar la aplicación de los cambios de estructura.	No Satisfactorio, se muestra el evento "Aplicado" solamente para un cambio.	No Satisfactorio, se muestra el evento "Aplicado" solamente para un cambio.

3ra iteración

Durante la tercera iteración de pruebas al módulo “Monitor de Réplica”, se corrigieron las no conformidades detectadas en la iteración anterior, cumpliendo correctamente con todas las funcionalidades especificadas para este módulo, las cuales se describen a continuación:

Al enviarse diez acciones los resultados obtenidos son los siguientes:

Tabla 13 Resultados obtenidos en la tercera iteración de pruebas sobre el módulo "Monitor de Réplica"

Funcionalidades del Monitor de Réplica	Resultado	
	Tiempo Real	Histórico
Monitorizar el envío de los grupos del proceso de réplica de estructura	Satisfactorio, se mostró el evento “Enviado”.	Satisfactorio, se mostró el evento “Enviado”.
Monitorizar la recepción de los grupos del proceso de réplica de estructura.	Satisfactorio, se mostró el evento “Recibido”.	Satisfactorio, se mostró el evento “Recibido”.
Monitorizar la aplicación de los cambios de estructura.	Satisfactorio, se mostró el evento “Aplicado” para cada una de las acciones.	Satisfactorio, se mostró el evento “Aplicado” para cada una de las acciones.

En el proceso de pruebas de caja negra al módulo “Configuración de Réplica” se aplicaron tres iteraciones de pruebas, con el objetivo de verificar el correcto funcionamiento del módulo y su cumplimiento con las funcionalidades. Se obtuvo un total de cuatro no conformidades las cuales se explican a continuación.

1ra iteración

Durante la primera iteración de pruebas al módulo “Configuración de Réplica”, se detectaron tres no conformidades, las cuales se describen a continuación:

Tabla 14 Resultados obtenidos en la primera iteración de pruebas sobre el módulo "Configuración de Réplica"

Funcionalidades del módulo “Configuración de Réplica”	Resultado obtenido	Resultado esperado	No conformidades
Adicionar filtros a la	Al adicionar un filtro	El sistema debe	No satisfactorio, el

configuración de réplica de estructura.	y guardar la configuración, ésta no se persistió en la BD.	adicionar los filtros de estructura a la configuración.	sistema no guardó la configuración cuando se añadieron filtros de estructura.
Modificar filtros de la configuración de réplica de estructura.	Al presionar el enlace "Editar filtros" en algún tipo de acción, no se cargaron los filtros añadidos.	El sistema debe permitir cargar los filtros añadidos para editarlos.	No satisfactorio, el sistema no cargó los filtros añadidos para ningún tipo de acción.
Eliminar filtros de estructura de la configuración de réplica.	El sistema eliminó correctamente los filtros de estructura añadidos a la configuración.	El sistema debe permitir eliminar los filtros añadidos.	Satisfactorio.
Aplicar filtros de esquemas a los grupos de estructura replicable.	Se realizaron siete cambios de estructura y se inició el proceso de réplica de estructura. La configuración para ese destino permitía sólo cuatro de esos cambios, sin embargo, fueron aplicados seis.	El sistema debe aplicar los cambios a partir de los filtros de estructura establecidos para ese destino.	No satisfactorio, el sistema no aplicó los cambios correspondientes a la configuración establecida para el destino especificado.

2da iteración

Durante la segunda iteración de pruebas al módulo "Configuración de Réplica", fueron corregidas las no conformidades de la iteración anterior, sin embargo, en esta iteración se detectó una no conformidad, la cual se describe a continuación:

Tabla 15 Resultados obtenidos en la segunda iteración de pruebas sobre el módulo "Configuración de Réplica"

Funcionalidades del módulo Configuración de Réplica	Resultado obtenido	Resultado esperado	No conformidades
Adicionar filtros a la configuración de réplica de estructura.	El sistema guardó los filtros de estructura en la configuración correctamente.	El sistema debe adicionar los filtros de estructura a la configuración.	Satisfactorio.
Modificar filtros de la configuración de réplica de estructura.	Al editar un filtro en algún tipo de acción el sistema lo editaba correctamente, sin embargo, afectaba los enlaces "Adicionar filtro" de los demás tipos de acciones.	El sistema debe permitir editar los filtros añadidos para cualquier tipo de acción sin afectar los demás tipos de acciones.	No satisfactorio, el sistema solamente permitió editar un filtro en un tipo de acción, afectando los demás tipos de acciones.
Eliminar filtros de estructura de la configuración de réplica.	El sistema eliminó correctamente los filtros de estructura añadidos a la configuración.	El sistema debe permitir eliminar los filtros añadidos.	Satisfactorio.
Aplicar filtros de esquemas a los grupos de estructura replicable.	Se realizaron siete cambios de estructura y se inició el proceso de réplica de estructura. La configuración para ese destino permitía sólo cuatro de esos cambios, los cuales fueron aplicados correctamente.	El sistema debe aplicar los cambios a partir de los filtros establecidos para ese destino.	Satisfactorio.

3ra Iteración

Durante la tercera iteración de pruebas al módulo "Configuración de Réplica", fue corregida la no conformidad detectada en la iteración anterior, cumpliendo correctamente con todas las funcionalidades especificadas para este módulo, las cuales se describen a continuación:

Tabla 16 Resultados obtenidos en la tercera iteración de pruebas sobre el módulo "Configuración de Réplica"

Funcionalidades del módulo Configuración de Réplica	Resultado obtenido	Resultado esperado	No conformidades
Adicionar filtros a la configuración de réplica de estructura.	El sistema guardó los filtros de estructura en la configuración correctamente.	El sistema debe adicionar los filtros de estructura a la configuración.	Satisfactorio.
Modificar filtros de la configuración de réplica de estructura.	El sistema editó los filtros de estructura correctamente.	El sistema debe permitir editar los filtros de estructura añadidos en la configuración.	Satisfactorio.
Eliminar filtros de estructura de la configuración de réplica.	El sistema eliminó correctamente los filtros de estructura añadidos a la configuración.	El sistema debe permitir eliminar los filtros añadidos.	Satisfactorio.
Aplicar filtros de esquemas a los grupos de estructura replicable.	Se realizaron siete cambios de estructura y se inició el proceso de réplica de estructura. La configuración para ese destino permitía sólo cuatro de esos cambios, los cuales fueron aplicados correctamente.	El sistema debe aplicar los cambios a partir de los filtros de estructura establecidos para ese destino.	Satisfactorio.

Como resultado de las pruebas de caja negra se realizó un total de seis iteraciones, donde se obtuvieron siete no conformidades, las cuales fueron solucionadas en su totalidad, demostrando que los módulos extendidos cumplen con los requerimientos pedidos por el cliente.

3.2.3 Pruebas de aceptación

El método de pruebas de aceptación se refiere a las pruebas llevadas a cabo por parte del cliente, labor fundamental para que valide el sistema, como último paso, previo a la puesta en explotación. Se debe insistir, principalmente, en los criterios de aceptación del sistema que sirven de base para asegurar que satisface los requisitos exigidos.(49)

El ciclo de vida de las pruebas de aceptación es el siguiente:

- Diseñar casos de prueba de aceptación basados en los requerimientos que presenta el cliente.
- Preparar datos de las pruebas de aceptación.
- Validar datos de los casos de prueba de aceptación.
- Procedimiento de la prueba.
- Ejecutar las pruebas de aceptación para validar el análisis de requerimientos del cliente.
- Comparar los resultados de las pruebas con los casos de prueba iniciales.(50)

A continuación se muestran los casos de pruebas de aceptación para las HU1, HU4 y HU5, si se desea consultar los restantes casos de prueba dirigirse a los Anexos del 33 al 36.

Tabla 17 Prueba de aceptación sobre la HU1 Adicionar filtros a la configuración de réplica de estructura

Caso de prueba	
Código: HU1_E1	Historia de usuario: 1
Nombre: Adicionar filtros a la configuración de réplica de estructura.	
Descripción: Prueba para la funcionalidad de adicionar filtros a la configuración de réplica de esquemas.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” seleccionando al menos un tipo de acción.	
Entradas/Pasos de Ejecución: Al seleccionar la opción “Adicionar filtros” el sistema muestra la ventana “Filtros de Estructura” para que el usuario determine las condiciones de la réplica para el tipo de acción seleccionada. El usuario establece las condiciones deseadas mediante los diferentes filtros (nombre, prefijo, sufijo y/o tipo de acción).	
Resultado Esperado: El sistema almacena los filtros en la configuración adicionada o cargada.	

Evaluación de la Prueba: La prueba fue realizada con éxito.

Tabla 18 Prueba de aceptación sobre la HU4 Aplicar filtros de esquemas a los grupos de estructura replicable

Caso de prueba	
Código: HU4_P1_E1	Historia de usuario: 4
Nombre: Aplicar filtros de esquemas a los grupos de estructura replicable.	
Descripción: Prueba para la funcionalidad de aplicar filtros a los grupos de estructura replicables antes de ser enviados.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” al menos un tipo de acción y adicionarle filtros. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de estructura.	
Entradas/Pasos de Ejecución: Al hacer clic en el botón “Iniciar” el sistema debe comprobar que las acciones contenidas en los grupos de estructuras replicables están permitidas por los filtros asociados a las configuraciones registradas.	
Resultado Esperado: El sistema aplica los filtros de estructura y envía aquellas acciones que se corresponden a las configuraciones registradas.	
Evaluación de la Prueba: La prueba fue realizada con éxito.	

Tabla 19 Prueba de aceptación sobre la HU5 Monitorizar la recepción de los grupos del proceso de réplica de estructura

Caso de prueba	
Código: HU5_P1_E1	Historia de usuario: 5
Nombre: Monitorizar la recepción de los datos del proceso de réplica de estructura.	
Descripción: Prueba para la funcionalidad de monitorizar el recibo de los grupos de estructura replicables.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de estructura.	
Entradas/Pasos de Ejecución: Al hacer clic en el botón “Iniciar” si hay cambios de estructura y el servidor de mensajería está online, cuando el sistema recibe el grupo de estructura replicable debe monitorizar su recibo.	
Resultado Esperado: El sistema publica el evento “Recibido” en el Monitor de Réplica.	

Evaluación de la Prueba: La prueba fue realizada con éxito.

Como resultado de estas pruebas se obtiene un acta de Aceptación emitida por el cliente, donde se manifiesta su satisfacción con las funcionalidades implementadas, además es firmada por el Jefe del departamento al que pertenece el proyecto.

3.2.4 Resultados obtenidos

Después de realizadas las pruebas de caja negra y aceptación al sistema se obtuvieron los siguientes resultados:

- Todas las no conformidades detectadas durante las iteraciones de pruebas fueron solucionadas.
- El sistema mostró ser completamente funcional, dando solución a todos los requisitos planteados.

3.3 Conclusiones parciales del capítulo

Una vez culminada la etapa de implementación y haberle realizado las pruebas correspondientes al sistema, se puede concluir que:

- Se logró implementar las funcionalidades con éxito, presentándose el Modelo de Implementación, conformado por los diagramas correspondientes a los componentes Distribuidor, Aplicador y Administrador, que fueron extendidos.
- Se describieron las pruebas de caja negra y aceptación efectuadas. Dichas pruebas corroboraron el cumplimiento de los requerimientos planteados por el cliente así como la obtención de los datos que se esperaban como respuesta de la aplicación, demostrando la solidez de su implementación.

CONCLUSIONES GENERALES

1. El estudio del estado del arte de los sistemas de réplica tanto dentro como fuera del país, arrojó como resultado que la mayoría no brinda el conocimiento necesario para ser utilizados, a excepción del Replicador de datos REKO que aportó información necesaria para el desarrollo de la solución.
2. Concluido el análisis de las soluciones existentes se decidió extender los módulos “Configuración de Réplica” y “Monitor de Réplica” para realizar los procesos de réplica de estructura en el Replicador de datos REKO, utilizando herramientas y tecnologías completamente libres, generando los artefactos correspondientes a la metodología empleada.
3. La solución fue sometida a un proceso de pruebas de funcionamiento guiado por casos de pruebas. La realización de las mismas corroboraron la solidez del sistema, cumpliendo con las características pedidas por el cliente.

RECOMENDACIONES

Después de haber concluido el trabajo de investigación se busca mejorar el sistema y ampliar sus prestaciones, por lo que se recomienda:

- Agregar al sistema la posibilidad de realizar la configuración a través de expresiones regulares y de establecer filtros con condiciones compuestas.

REFERENCIAS BIBLIOGRÁFICAS

1. Valdés DP. ¿Qué son las bases de datos? 2007. Disponible en: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos>.
2. Lafuente A. Introducción a los sistemas distribuidos. Disponible en: <http://www.sc.ehu.es/acwlaroa/SDI/Apuntes/Cap1.pdf>.
3. AulaClic. El DDL, Lenguaje de Definición de Datos (I) 2010. Disponible en: http://www.aulacli.es/sqlserver/t_8_1.htm.
4. Argentina DdId. DDL y DML. Disponible en: <http://www.ub.edu.ar/catedras/ingenieria/Datos/capitulo4/cap42.htm>.
5. Soto MCDE. Base de Datos. Replicación de Datos en SQL Server. Disponible en: www.desarrollopro.com/tec/Unidad%20III%20BDD%20B.ppt.
6. Urbano R. Database Advances Replication 2010. Disponible en: http://docs.oracle.com/cd/E11882_01/server.112/e10706.pdf.
7. Jerez GRL. Componente para el envío y recepción de datos del proceso de réplica de estructura para el software de replicación Reko [Investigación]: Universidad de las Ciencias Informáticas; 2012.
8. Mastermagazine. Disponible en: <http://www.mastermagazine.info/termino/4404.php>.
9. PostgreSQL GdDGd. Slony-I 1.2.23 Documentation 2010. Disponible en: <http://www.slony.info/documentation/1.2/slonyadmin.html>.
10. Long E. SymmetricDS 2015. Disponible en: <http://www.symmetricds.org/>.
11. CDAE. Manual de usuario de REKO: Universidad de las Ciencias Informáticas; 2014.
12. Sánchez TR. Metodología de desarrollo para la actividad productiva de la UCI: Universidad de las Ciencias Informáticas; 2013.
13. Fernández OB. Introducción a los lenguajes de programación Java. Una guía básica 2004. Disponible en: <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>.
14. Masadelante Eddd. ¿Qué es JavaScript? - Definición de Javascript 1999. Disponible en: <https://www.masadelante.com/faqs/javascript>.
15. ¿Que es JavaScript ? 2010. Disponible en: <http://www.efectosjavascript.com/javascript.html>.
16. Herramientas de programación 2009. Disponible en: <http://www.lenguajes-de-programacion.com/herramientas-de-programacion.shtml>.
17. Spring Dd. Spring Tool Suite 2010. Disponible en: <http://springla.io/spring-tool-suite>.

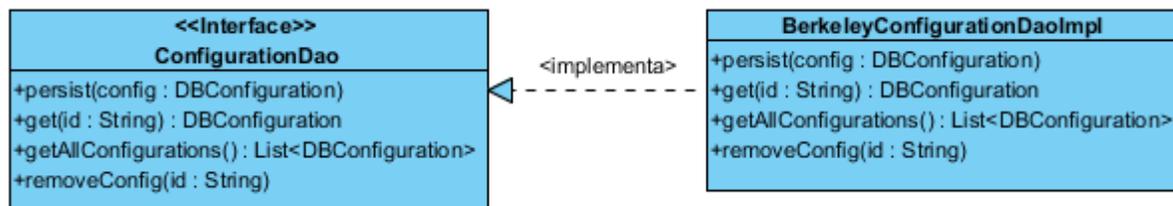
18. Martínez R. Sobre PostgreSQL 2013. Disponible en:
http://www.postgresql.org.es/sobre_postgresql#intro.
19. Marc Gibert Ginestà OPM. Bases de Datos en Postgres. Disponible en:
http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf.
20. Groussard T. Java Enterprise Edition. Desarrollo de aplicaciones web con JEE6 2007. Disponible en: <http://www.ediciones-eni.com/libros/java-enterprise-edition-desarrollo-de-aplicaciones-web-con-jee-6/.a8611edb1118d274965600ba195c0d71.html>.
21. JCCSC-II. Kit de herramientas de código abierto DHTML escrito en JavaScript 2011. Disponible en: http://www.ecured.cu/index.php/Dojo_toolkit.
22. Andalucía Jd. Spring. Disponible en:
<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/93>.
23. Squires IVMCIERC. Sistema para el manejo de órdenes de servicio para los especialistas del centro de ingeniería clínica y electromedicina: Universidad de las Ciencias Informáticas; 2013. Disponible en:
<http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/download/167/32>
24. Yolanda ILPE. Modelado de Sistemas con UML. Disponible en:
<http://www.monografias.com/trabajos94/modelado-sistemas-uml/modelado-sistemas-uml.shtml>.
25. Activemq. Active MQ 2011. Disponible en: <http://activemq.apache.org/index.html>.
26. Foundation A. Apache Tomcat. Disponible en:
<http://apachefoundation.wikispaces.com/Apache+Tomcat>.
27. Larman. C, Hall. P. UML y Patrones. 2da Edición. 2003.
28. Companioni Sardiña Y. y otros. REKO: Una solución de réplica para Sistemas de Bases de Datos Relacionales Distribuidos. Universidad de las Ciencias Informáticas; 2012.
29. Sommerville I. Requerimientos. En Ingeniería del Software. 7 ed. Madrid: Pearson Education S.A, 2005, p. 712.
30. CDAE. Especificación de requisitos de software del proyecto Replicador de datos REKO v4.0. Universidad de las Ciencias Informáticas; 2014.
31. Informática OdPd. La Oficina de Proyectos de Informática 2013. Disponible en:
<http://www.pmoinformatica.com/2013/04/que-son-las-historias-de-usuario-7.html>.

32. Jerez GRL. Componente para el envío y recepción de datos del proceso de réplica de estructura para el software de replicación Reko [Investigación]: Universidad de las Ciencias Informáticas; 2012.
33. Pressman RS. Ingeniería del Software. Un enfoque práctico 2002. 7ma edición:[Disponible en: <http://es.slideshare.net/jes4791/ingenieria-delsoftware-un-enfoque-practico>.
34. Reynoso N. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. 2004. Disponible en:
http://univirtual.unicauca.edu.co/moodle/pluginfile.php/24153/mod_resource/content/0/Lectura/Estilos_y_patrones.doc.
35. Oracle. Dimension 2: capas lógicas 2010. Disponible en: <http://docs.oracle.com/cd/E19528-01/820-0888/auto22/index.html>.
36. Chávez MAP. Replicacion SQL Server 2011. Disponible en:
<http://basesdedatosues.blogspot.com/2011/06/replicacion-sql-server.html>.
37. Bustamante C. Patron Singleton, DAO. Disponible en:
http://www.librosdigitales.net/Fphp_dev/Flecciones/PHP_POO_Leccion_5.pdf.
38. González O. Patrones de Diseño GRASP. Disponible en:
<http://es.scribd.com/doc/53315954/Patrones-de-Disenio-GRASP>.
39. Landaburo JJR. Sistema informático para el ensamblaje y edición de secuencias de ácidos nucleicos [Investigación]. Universidad de las Ciencias Informáticas; 2013.
40. Peña NM. Desarrollo de un dialecto de SQL Server para la réplica de datos en el software Reko [Investigación]. Universidad de las Ciencias Informáticas; 2013.
41. Gutierrez D. UML Diagrama de Paquetes 2009. Disponible en:
http://www.codecompiling.net/files/slides/UML_clase_05_UML_paquetes.pdf.
42. Montecinos ACFI. Diagrama de Clases. Disponible en:
http://users.dcc.uchile.cl/~afierro/d_clases.html.
43. Hernandez DCAYT. Conferencia #1: Continuación de la Disciplina Análisis y Diseño. En Ingeniería de software II: Universidad de las Ciencias Informáticas; 2011.
44. Tema II. Fase de Construcción. Universidad de las Ciencias Informáticas; 2011.
45. Astudillo Visconti HM. Fundamentos de Ingeniería de Software 2012. Disponible en:
<http://www.inf.utfsm.cl/~visconti/ili236/Documentos/15-Implementacion.pdf>.
46. Ivar Jacobson GB, James Rumbaugh. El proceso Unificado de Desarrollo de Software 2000.

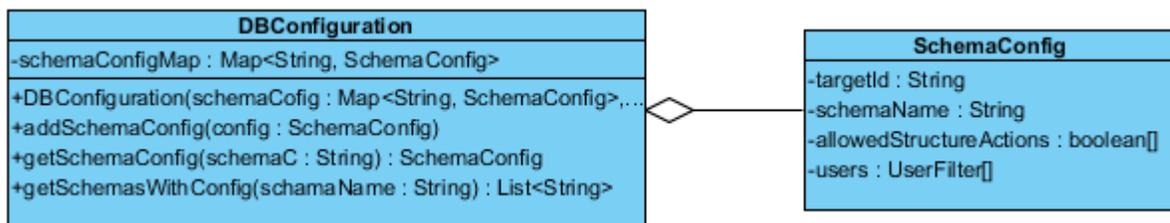
47. Torres M. Técnicas de prueba 2005. Disponible en:
<http://indalog.ual.es/mtorres/LP/Prueba.pdf>.
48. UCI. Ingeniería de Software II Conferencia 4: Flujo de trabajo de Prueba. 2011. Español.
49. Synexia. Definición de las pruebas de aceptación del sistema 2007. Disponible en:
<http://www.synexia.net/tecnologia/cvs/descriptivo/478/5101>.
50. CopSoftware S. Plan de Pruebas de Aceptación. Versión 2.0 2006. Disponible en:
http://www.acis.org.co/fileadmin/Curso_Memorias/Curso_CMMI_Sept06/Modulo%205%20-%20Corporate%20Maturity%20Ejemplos/Ejemplo%20Plan%20de%20Pruebas.doc.

ANEXOS

Anexo 1 Ejemplo del patrón DAO



Anexo 2 Ejemplo del patrón Creador



Anexo 3 Ejemplo del patrón Bajo acoplamiento

1: Invoca las funcionalidades necesarias para monitorizar el envío



2: Publica el evento

Anexo 4 Descripción de la HU 2 Modificar filtros de la configuración de réplica de estructura

Número: HU 2	Nombre del requisito: Modificar filtros de la configuración de réplica de estructura.
Programador: Alejandro García Santiesteban	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 15 días
Riesgo en Desarrollo: Fallos eléctricos.	Tiempo Real: 120 horas
Descripción: En la opción “Esquemas” del módulo de “Configuración de Réplica”, el sistema debe brindar la posibilidad de “Editar filtros” en cada tipo de acción que posea filtros de estructura, en la configuración de réplica almacenada. Al seleccionar esta	

opción, se debe cargar automáticamente la ventana “Filtro de Estructura” con los filtros que se tienen almacenados para la configuración y acción seleccionada, brindando al usuario la posibilidad de actualizar los parámetros editables. Se debe brindar al usuario la posibilidad de modificar el nombre, prefijo, sufijo o tipo de acción en el filtro. Se debe permitir la adición de nuevas condiciones o la eliminación de las existentes. Al aceptar, el sistema debe actualizar la configuración con los filtros de estructura actualizados.

Observaciones: La opción “Editar filtro” debe mostrarse en cada tipo de acción que tenga almacenados filtros. Por ejemplo, en la siguiente figura solamente posee filtros la acción de “Creación”, por tanto las restantes opciones no deben brindar la posibilidad de “Editar filtros” debido a que no tienen almacenado filtros.



Prototipo de interfaz:



Anexo 5 Descripción de la HU 3 Eliminar filtros de estructura de la configuración de réplica

Número: HU 3	Nombre del requisito: Eliminar filtros de estructura de la configuración de réplica.
Programador: Alejandro García Santiesteban	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 15 días

Riesgo en Desarrollo: Fallos eléctricos.	Tiempo Real: 120 horas												
Descripción: En la opción “Esquemas” del módulo de “Configuración de Réplica”, el sistema debe brindar la posibilidad de “Eliminar filtros” en cada tipo de acción que posea filtros de estructura en la configuración de réplica almacenada. Al seleccionar esta opción, el sistema debe eliminar los filtros creados dependiendo de la acción en la que se realice.													
Observaciones: Ningún campo vacío.													
Prototipo de interfaz:													
 <p>The screenshot shows a web interface for configuring replication schemas. At the top, there are navigation links: Inicio, Configuración General, Configuración de Réplica (selected), Configuración de Nodos, Monitor de Réplica, Exportar, Importar, Sincronización, Conflictos, and Seguridad. Below these are buttons for 'Nueva Configuración' and 'Cargar Configuración'. A secondary row contains 'Nueva Configuración de Entrada' and 'Cargar Configuración de Entrada'. A third row has 'Guardar', 'Guardar como', and 'Cancelar' buttons, with 'Nueva configuración' on the right. The main area is titled 'Esquemas' and includes a 'Filtrar esquemas' section with dropdowns for 'Replicar por', 'No replicar por', 'Seleccionar', and 'Configurar Usuarios'. Below this is a table with the following data:</p> <table border="1"> <thead> <tr> <th>Esquemas</th> <th>Creación</th> <th>Modificación</th> <th>Eliminación</th> </tr> </thead> <tbody> <tr> <td>pg_catalog</td> <td>No</td> <td>No</td> <td>No</td> </tr> <tr> <td>public</td> <td>Sí</td> <td>No</td> <td>No</td> </tr> </tbody> </table> <p>Under the 'public' row, there are links for 'Editar filtro', 'Eliminar filtro', and 'Usuarios'.</p>		Esquemas	Creación	Modificación	Eliminación	pg_catalog	No	No	No	public	Sí	No	No
Esquemas	Creación	Modificación	Eliminación										
pg_catalog	No	No	No										
public	Sí	No	No										

Anexo 6 Descripción de la HU 6 Monitorizar la recepción de los datos del proceso de réplica de estructura

Número: HU 6		Nombre del requisito: Monitorizar la recepción de los datos del proceso de réplica de estructura.	
Programador: Alejandro García Santiesteban.		Iteración Asignada: 1	
Prioridad: Media		Tiempo Estimado: 15 días	
Riesgo en Desarrollo: Fallos eléctricos.		Tiempo Real: 120 horas	
Descripción: En el módulo de “Monitor de Réplica”, al realizarse la recepción de los cambios a aplicar, el sistema debe publicar el evento “Recibido”, el cual confirma al usuario que la recepción de las acciones fue realizada con éxito. El sistema debe mostrar además los detalles de su llegada. De las acciones que son recibidas se debe mostrar la dirección de la réplica, la fecha en que se recibe la acción, el nodo del que proviene, el destino, el estado (recibiendo o recibido), la sentencia SQL, el esquema y la tabla que se afecta con el cambio, así como el tipo de acción.			
Observaciones: En tiempo real solo se podrá observar el evento si en ese momento el			

usuario se encuentra en la vista “Monitor en tiempo real”.

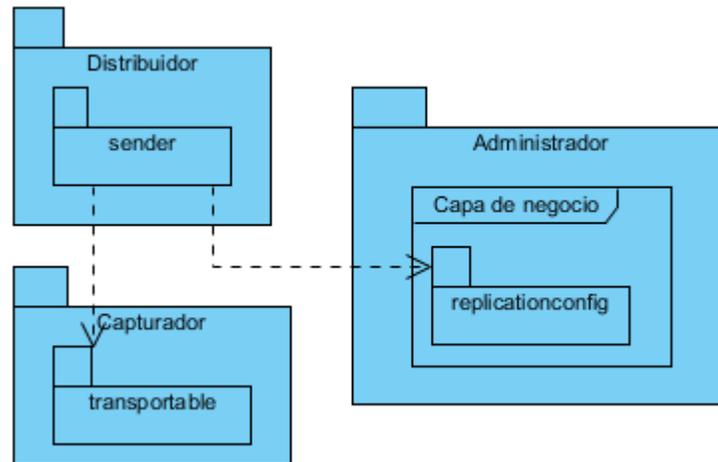
Prototipo de interfaz:

Dirección	Fecha	Remitente	Destinos	SQL	Esquema	Tabla	Tipo de acción	Observación
	18/3/2015 16:6:37	Nodo1	Nodo2: Recibido	create table public.autor (ci integer not nul, edad integer)	public	autor	Creación de tabla	...

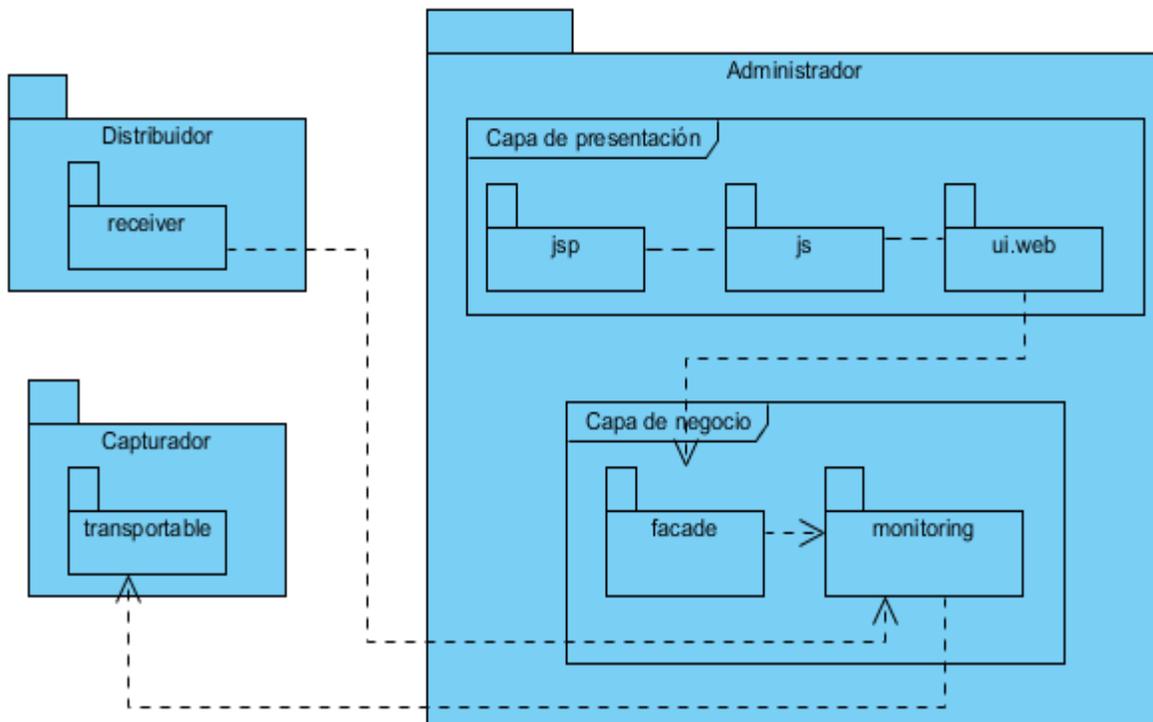
Anexo 7 Descripción de la HU 7 Monitorizar la aplicación de los cambios de estructura

Número: HU 6		Nombre del requisito: Monitorizar la aplicación de los cambios de estructura.																												
Programador: Alejandro García Santiesteban.		Iteración Asignada: 1																												
Prioridad: Media		Tiempo Estimado: 15 días																												
Riesgo en Desarrollo: Fallos eléctricos.		Tiempo Real: 120 horas																												
<p>Descripción: Cuando el grupo de estructura replicable arriba a su destino inicia la aplicación de los cambios que contiene. Al iniciar la aplicación de los cambios, el sistema debe publicar el evento “Aplicado”, el cual confirma al usuario que la aplicación de las acciones fue realizado con éxito, mostrando además los detalles de la acción realizada. Se debe mostrar la dirección, la fecha, el remitente, el destino, el SQL de la acción, el esquema y/o tabla en el que se realiza el cambio y el tipo de acción ejecutada sobre la BD.</p>																														
Observaciones: NA																														
Prototipo de interfaz:																														
<table border="1"> <thead> <tr> <th>Dirección</th> <th>Fecha</th> <th>Remitente</th> <th>Destinos</th> <th>SQL</th> <th>Esquema</th> <th>Tabla</th> <th>Tipo de acción</th> <th>Observación</th> </tr> </thead> <tbody> <tr> <td></td> <td>18/3/2015 16:6:37</td> <td>Nodo1</td> <td>Nodo2: Recibido</td> <td>create table public.autor (ci integer not nul, edad integer)</td> <td>public</td> <td>autor</td> <td>Creación de tabla</td> <td>...</td> </tr> <tr> <td></td> <td>18/3/2015 16:6:37</td> <td>Nodo1</td> <td>Nodo2: Aplicado</td> <td>create table public.autor (ci integer not nul, edad integer)</td> <td>public</td> <td>autor</td> <td>Creación de tabla</td> <td>...</td> </tr> </tbody> </table>				Dirección	Fecha	Remitente	Destinos	SQL	Esquema	Tabla	Tipo de acción	Observación		18/3/2015 16:6:37	Nodo1	Nodo2: Recibido	create table public.autor (ci integer not nul, edad integer)	public	autor	Creación de tabla	...		18/3/2015 16:6:37	Nodo1	Nodo2: Aplicado	create table public.autor (ci integer not nul, edad integer)	public	autor	Creación de tabla	...
Dirección	Fecha	Remitente	Destinos	SQL	Esquema	Tabla	Tipo de acción	Observación																						
	18/3/2015 16:6:37	Nodo1	Nodo2: Recibido	create table public.autor (ci integer not nul, edad integer)	public	autor	Creación de tabla	...																						
	18/3/2015 16:6:37	Nodo1	Nodo2: Aplicado	create table public.autor (ci integer not nul, edad integer)	public	autor	Creación de tabla	...																						

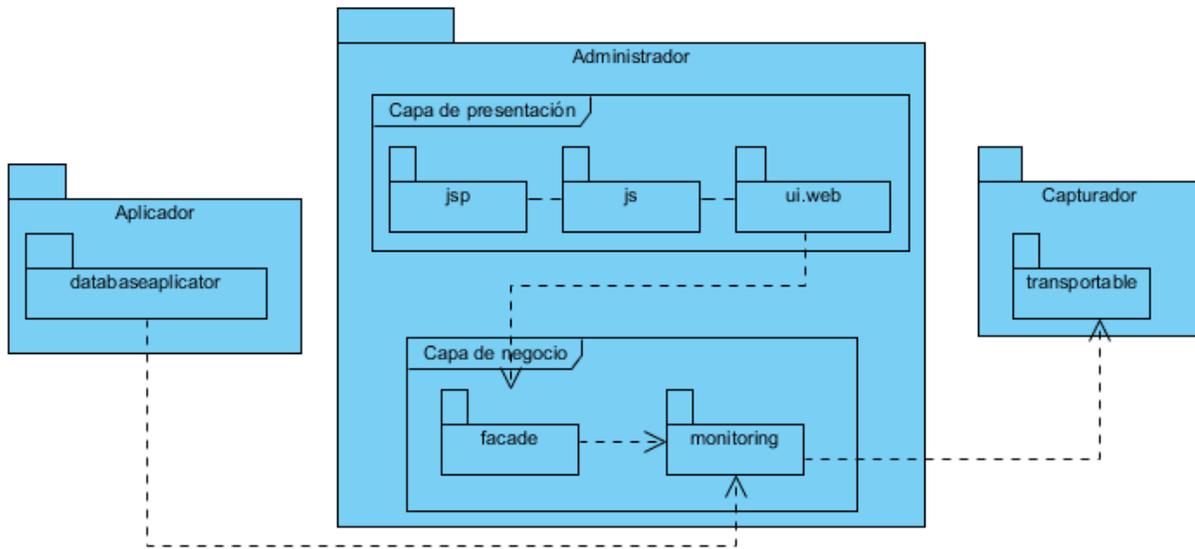
Anexo 8 Diagrama de paquetes del RF 4



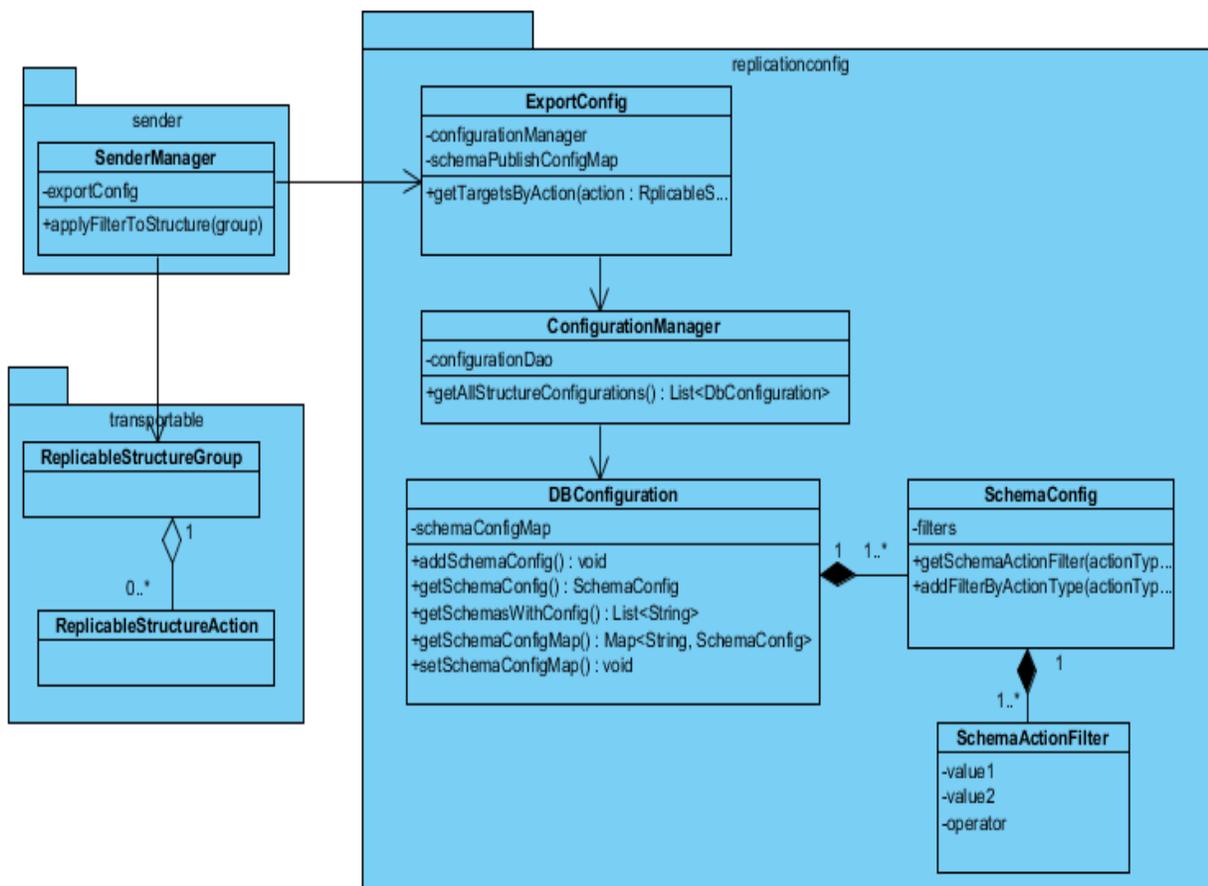
Anexo 9 Diagrama de paquetes del RF 6



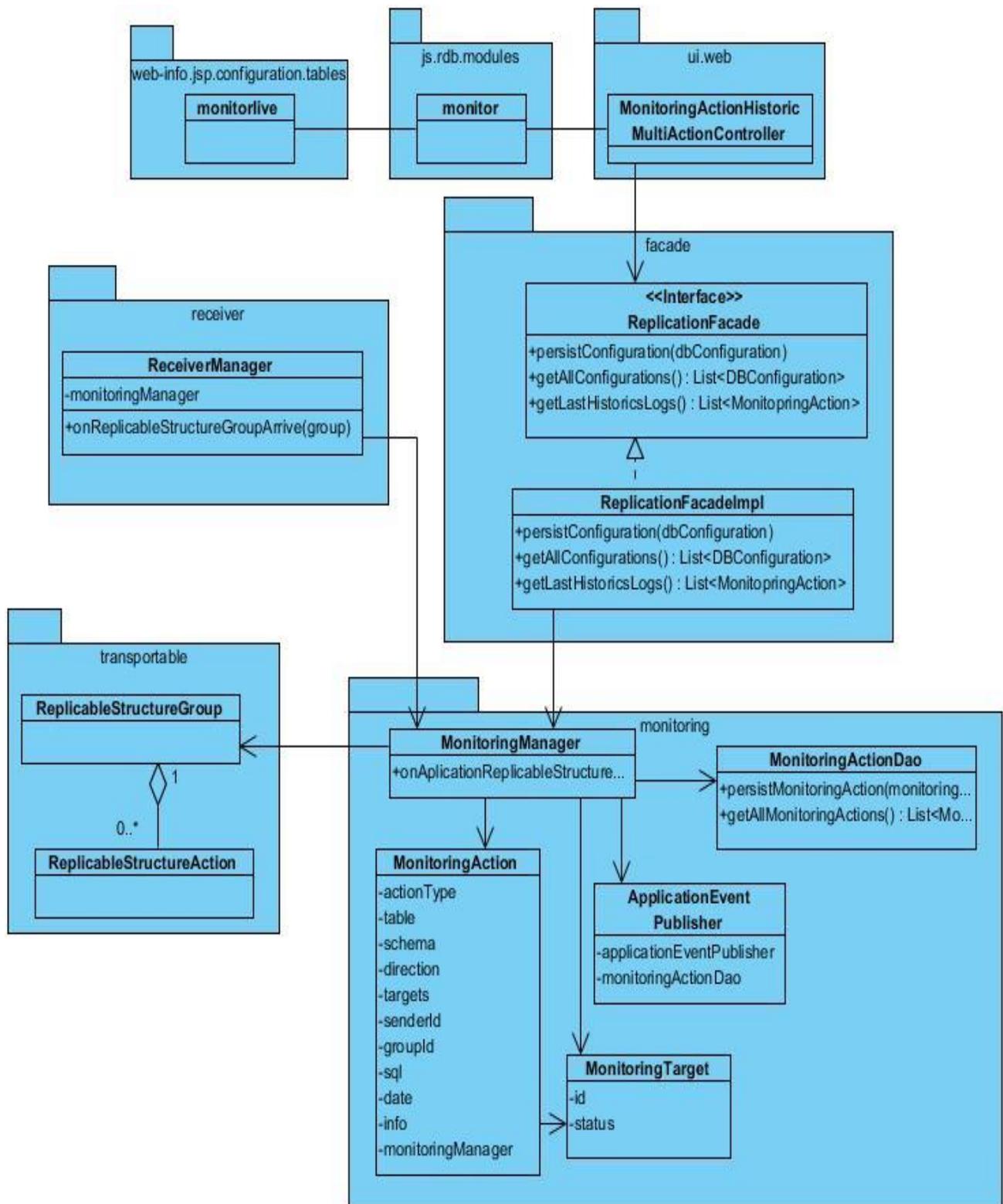
Anexo 10 Diagrama de paquetes del RF 7



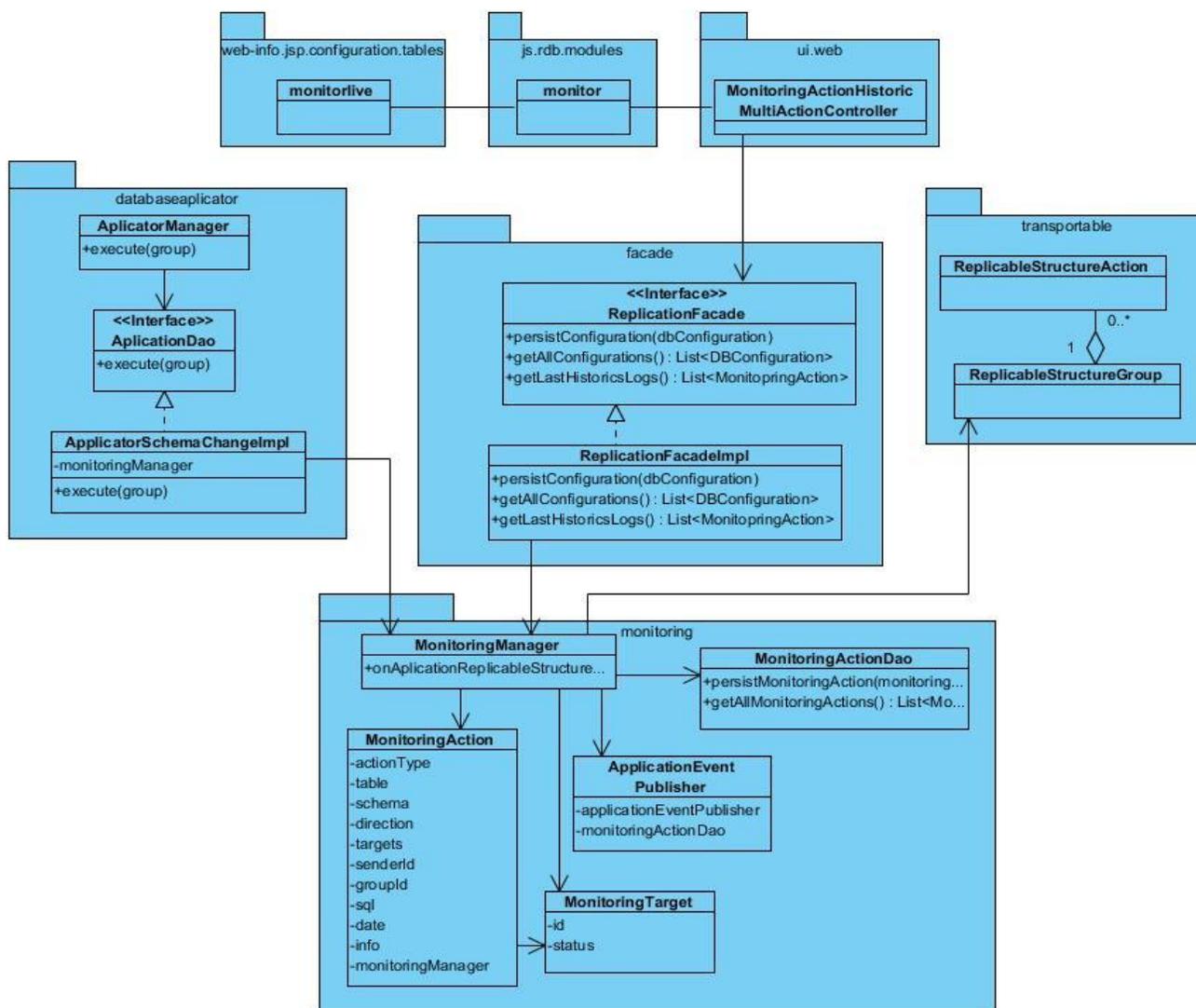
Anexo 11 Diagrama de clases del Diagrama de paquetes del RF 4



Anexo 12 Diagrama de clases del Diagrama de paquetes del RF 6



Anexo 13 Diagrama de clases del Diagrama de paquetes del RF 7



Anexo 14 Descripción textual de la clase DBConfiguration

Descripción de la Clase DBConfiguration	
Nombre: DBConfiguration	
Tipo de clase: Modelo	
Atributos	Tipos
schemaConfigMap	private
Responsabilidades	
Nombre:	addSchemaConfig(schemaConfig: cu.uci.replicationconfig. replicationconfig.SchemaConfig)
Descripción:	Guarda las configuraciones de esquemas en la configuración de réplica.
Nombre:	getSchemaConfig(id: String): cu.uci.replicationconfig. replicationconfig.SchemaConfig

Descripción:	Obtiene la configuración de esquema correspondiente al identificador especificado por parámetro.
Nombre:	getSchemasWithConfig() : java.util.List<String>
Descripción:	Obtiene todos los esquemas que contienen una configuración de réplica de estructura.
Nombre:	getSchemaConfigMap() : Map<String, SchemaConfig>
Descripción:	Obtiene todas las configuraciones de réplica de estructura.
Nombre:	setSchemaConfigMap schemaConfig: cu.uci.replicationconfig. replicationconfig.SchemaConfig ()
Descripción:	Actualiza la configuración de réplica de estructura establecida.

Anexo 15 Descripción textual de la clase SchemaConfig

Descripción de la Clase SchemaConfig	
Nombre:	SchemaConfig
Tipo de clase:	Modelo
Atributos	Tipos
filters	private
Responsabilidades	
Nombre:	getSchemaActionFilter(actionType) : List<SchemaActionFilter>
Descripción:	Obtiene los filtros de estructura correspondiente al tipo de acción especificada por parámetros.
Nombre:	addFilterByActionType(actionType, operator, v1, v2) : void
Descripción:	Adiciona los filtros de esquema con los parámetros establecidos a la configuración de réplica, particularmente al que se corresponde con el tipo de acción especificada.
Nombre:	dropFilterByAction(action, Type)
Descripción:	Elimina los filtros de esquema de la configuración de réplica, específicamente el que se corresponde con el tipo de acción especificada.

Anexo 16 Descripción textual de la clase ReceiverManager

Descripción de la clase ReceiverManager	
Nombre:	ReceiverManager.
Tipo de clase:	Controladora.
Atributos	Tipos
logger	protected
monitoringManager	private

Responsabilidades	
Nombre:	afterPropertiesSet()
Descripción:	Método a implementar de la interfaz InitializingBean.
Nombre:	onReplicableStructureGroupArrive(replicableStructureGroup: cu.uci.dbreplication.transportable.ReplicableStructureGroup)
Descripción:	Recibe el RSG enviado por otros nodos de réplica y notifica la llegada.

Anexo 17 Descripción textual de la clase ApplicatorManager

Descripción de la clase ApplicatorManager	
Nombre:	ApplicatorManager
Tipo de clase:	Controladora
Atributos	Tipos
applicatorSchemaChange	private
Responsabilidades	
Nombre:	execute(replicableStructureGroup: cu.uci.dbreplication.transportable.ReplicableStructureGroup)
Descripción:	Envía los grupos para aplicar los cambios en dependencia del tipo de replicación especificado.

Anexo 18 Descripción textual de la clase ApplicatorDao

Descripción de la clase ApplicatorDao	
Nombre:	ApplicatorDao
Tipo de clase:	Interfaz
Responsabilidades	
Nombre:	execute(replicableStructureGroup: cu.uci.dbreplication.transportable.ReplicableStructureGroup)
Descripción:	Envía aplicar los cambios para el tipo de replicación especificada.

Anexo 19 Descripción textual de la clase ApplicatorSchemaChangeImpl

Descripción de la clase ApplicatorSchemaChangeImpl	
Nombre:	ApplicatorSchemaChangeImpl
Tipo de clase:	Controladora
Atributos	Tipos
monitoringManager	private
Responsabilidades	
Nombre:	execute(replicableStructureGroup:

	cu.uci.dbreplication.transportable.ReplicableStructureGroup)
Descripción:	Envía aplicar los cambios para el tipo de replicación especificada.

Anexo 20 Descripción textual de la clase ReplicableStructureAction

Descripción de la clase ReplicableStructureAction	
Nombre:	ReplicationStructureAction
Tipo de clase:	Modelo
Atributos	Tipos
table	private
column_name	private
destination	private
executedByUser	private
destFilter	private
user	private
actionType	private
CREATE_TABLE	public
DROP_TABLE	public
ALTER_ADD_COLUMN	public
ALTER_DROP_COLUMN	public
ALTER_COLUMN	public
Responsabilidades	
Nombre:	getTable():cu.uci.dbreplication.dbmetadata.Table
Descripción:	Obtiene la tabla de esa acción.
Nombre:	setTable(table: cu.uci.dbreplication.dbmetadata.Table)
Descripción:	Cambia la tabla por la especificada por parámetro.
Nombre:	getColumnName():String
Descripción:	Obtiene la columna que en la acción Alter se está modificando.
Nombre:	setColumnName(columnName: String)
Descripción:	Modifica el nombre de la columna.
Nombre:	getDestination(): java.util.Set<String>
Descripción:	Obtiene los destinos de los nodos remotos.
Nombre:	setDestination(targets: java.util.Set<String>)
Descripción:	Cambia los destinos por los especificados.
Nombre:	getActionType(): int
Descripción:	Obtiene el tipo de acción a replicar.

Nombre:	setActionType(actionType: int)
Descripción:	Modifica el tipo de acción.
Nombre:	getExecutedByUser(): String
Descripción:	Obtiene el usuario que replicó.
Nombre:	setExecutedByUser(user: String)
Descripción:	Modifica el usuario que replica el cambio.
Nombre:	getUser(): String
Descripción:	Obtiene el usuario del Replicador de datos REKO.
Nombre:	setUser(user: String)
Descripción:	Modifica al usuario del Replicador de datos REKO.
Nombre:	getDestFilter(): HashMap<String, Long>
Descripción:	Obtiene los filtros asociados a la acción.
Nombre:	setDestFilter(destFilter: HashMap<String, Long>)
Descripción:	Modifica el filtro asociado a la acción.
Nombre:	containsFilter(target: String): boolean
Descripción:	Comprueba si la acción contiene algún filtro para el destino especificado.

Anexo 21 Descripción textual de la clase ReplicableStructureGroup

Descripción de la clase ReplicableStructureGroup	
Nombre:	ReplicationStructureGroup
Tipo de clase:	Modelo
Atributos	Tipos
id	private
senderId	private
creatorId	private
replicatorType	private
REPLICATOR_COMPARE	public
REPLICATOR_TYPE_TRIGGERS	public
actions	private
targets	private
Responsabilidades	
Nombre:	getId(): String
Descripción:	Obtiene el identificador del grupo replicable.
Nombre:	setId(id: String)
Descripción:	Modifica el identificador del grupo.

Nombre:	getSenderId(): String
Descripción:	Obtener el identificador del Sender que envía esta agrupación.
Nombre:	setSenderId(id: String)
Descripción:	Modifica el identificador del Sender que envía esta agrupación por el especificado por parámetro.
Nombre:	getCreatorId(): String
Descripción:	Obtiene el identificador del Creator que aplica los cambios contenidos en esta agrupación.
Nombre:	setCreatorId(id: String)
Descripción:	Modifica el identificador del Creator que aplica los cambios contenidos en esta agrupación por el especificado por parámetro.
Nombre:	getTypeReplicator(): int
Descripción:	Obtiene el tipo de réplica.
Nombre:	setTypeReplicator(typeReplicator: int)
Descripción:	Modifica el tipo de réplica por el especificado en el parámetro.
Nombre:	getActions(): java.util.Set< cu.uci.dbreplication.transportable.ReplicableStructureAction>
Descripción:	Obtiene las acciones del grupo.
Nombre:	getActionsByTargetId(targetId: java.util.Set<String>): java.util.Set< cu.uci.dbreplication.transportable.ReplicableStructureAction>
Descripción:	Obtiene las acciones de un destino específico.
Nombre:	getTargets(): HashMap<String, Long>
Descripción:	Obtiene los destinos de los nodos remotos.
Nombre:	setTargets(targets: HashMap<String, Long>)
Descripción:	Cambia los destinos por los especificados.
Nombre:	addSentence(action: cu.uci.dbreplication.transportable.ReplicableStructureAction)
Descripción:	Adiciona la nueva acción al listado que contiene la agrupación a replicar.
Nombre:	setSentTime(long sentTime)
Descripción:	Modifica el tiempo de envío.
Nombre:	getCounter(idTarget: String): long
Descripción:	Informa si la agrupación contiene el destino especificado.

Anexo 22 Descripción textual de la clase MonitoringAction

Descripción de la clase MonitoringAction	
Nombre:	MonitoringAction
Tipo de clase:	Modelo

Atributos		Tipos
id		private
sql		private
groupId		private
actionType		private
table		private
esquema		private
direction		private
targets		private
senderId		private
date		private
info		private
Responsabilidades		
Nombre:	getId(): String	
Descripción:	Obtiene el identificador de la acción monitorizada.	
Nombre:	getSql(): String	
Descripción:	Obtiene el sql de la acción monitorizada.	
Nombre:	getGroupId(): String	
Descripción:	Obtiene el identificador del grupo monitorizado.	
Nombre:	getActionType(): String	
Descripción:	Obtiene el tipo de acción.	
Nombre:	getEsquema(): String	
Descripción:	Obtiene el nombre del esquema afectado.	
Nombre:	getTable(): String	
Descripción:	Obtiene el nombre de la tabla afectada.	
Nombre:	getDireccion(): String	
Descripción:	Obtiene el sentido de la acción monitorizada.	
Nombre:	getTargets(): List<cu.uci.dbreplication.monitoring.MonitoringTarget>	
Descripción:	Obtiene los destinos de la acción monitorizada.	
Nombre:	getSenderId(): String	
Descripción:	Obtiene el identificador del nodo que envía la acción.	
Nombre:	getDate(): String	
Descripción:	Obtiene la fecha en que se publica la acción.	
Nombre:	getInfo(): String	
Descripción:	Obtiene el nombre del evento a publicar.	

Nombre:	setId(id: String)
Descripción:	Modifica el identificador de la acción monitorizada.
Nombre:	getsql(sql: String)
Descripción:	Modifica el sql de la acción monitorizada.
Nombre:	getGroupId(groupId: String)
Descripción:	Modifica el identificador del grupo monitorizado.
Nombre:	getActionType(action: String)
Descripción:	Modifica el tipo de acción.
Nombre:	getEsquema(esquemaName: String)
Descripción:	Modifica el nombre del esquema afectado.
Nombre:	getTable(tableName: String)
Descripción:	Modifica el nombre de la tabla afectada.
Nombre:	getDireccion(inOut: String)
Descripción:	Modifica el sentido de la acción monitorizada.
Nombre:	getTargets(targets: List<cu.uci.dbreplication.monitoring.MonitoringTarget>)
Descripción:	Modifica los destinos de la acción monitorizada.
Nombre:	getSenderId(senderId: String)
Descripción:	Modifica el identificador del nodo que envía la acción.
Nombre:	getDate(fechaHora: String)
Descripción:	Modifica la fecha en que se publica la acción.
Nombre:	getInfo(info: String)
Descripción:	Modifica el nombre del evento a publicar.

Anexo 23 Descripción textual de la clase MonitoringTarget

Descripción de la clase MonitoringTarget	
Nombre:	MonitoringTarget
Tipo de clase:	Modelo
Atributos	Tipos
id	private
status	private
sql	private
Responsabilidades	
Nombre:	getId(): String
Descripción:	Obtiene el identificador de la acción monitorizada.
Nombre:	getsql(): String

Descripción:	Obtiene el sql de la acción monitorizada.
Nombre:	getStatus (): String
Descripción:	Obtiene el estado del grupo monitorizado.
Nombre:	getId(id: String)
Descripción:	Modifica el identificador del nodo que envía la acción.
Nombre:	getSql(sql: String)
Descripción:	Modifica la fecha en que se publica la acción.
Nombre:	getStatus(status: String)
Descripción:	Modifica el nombre del evento a publicar.

Anexo 24 Descripción textual de la clase MonitoringActionDao

Descripción de la clase MonitoringActionDao	
Nombre:	MonitoringActionDao
Tipo de clase:	Auxiliar
Atributos	Tipos
EnvironmentFactory environmentFactory;	private
Environment dbEnvironment;	private
Database monitoringActionsDatabase;	private
Responsabilidades	
Nombre:	persistMonitoringAction(MonitoringAction monitoringAction): void
Descripción:	Persiste la acción monitorizada
Nombre:	deleteMonitoringAction(MonitoringAction monitoringAction): void
Descripción:	Elimina la acción monitorizada que se le especifique.
Nombre:	deleteAllMonitoringAction():void
Descripción:	Elimina todas las acciones monitorizadas.
Nombre:	getAllMonitoringActions():List<MonitoringAction>
Descripción:	Obtiene todas las acciones monitorizadas.
Nombre:	getFirst20MonitoringActions():List<MonitoringAction>
Descripción:	Obtiene las primeras 20 acciones que se han monitorizado.
Nombre:	getLastMonitoringActions(int count): List<MonitoringAction>
Descripción:	Obtiene las últimas acciones monitorizadas, especificando la cantidad deseada.

Anexo 25 Descripción textual de la clase ReplicationFacade

Descripción de la clase ReplicationFacade
Nombre: ReplicationFacade

Tipo de clase: Interfaz	
Responsabilidades	
Nombre:	getDataBaseSchemas():List<String>
Descripción:	Obtiene los esquemas existentes en la base de datos.
Nombre:	getDataBaseStructure(String schema): DataBaseStructure
Descripción:	Obtiene la estructura con la que cuenta un esquema determinado en la base de datos.
Nombre:	persistConfiguration(DBConfiguration configuration): void
Descripción:	Persiste la configuración especificada en la base de datos.

Anexo 26 Descripción textual de la clase ReplicationFacadeImpl

Descripción de la clase ReplicationFacadeImpl	
Nombre:	ReplicationFacadeImpl
Tipo de clase:	Auxiliar
Responsabilidades	
Nombre:	getDataBaseSchemas():List<String>
Descripción:	Obtiene la estructura de los esquemas de la base de datos.
Nombre:	persistConfiguration(DBConfiguration configuration): void
Descripción:	Persiste la configuración especificada en la base de datos.

Anexo 27 Prueba de caja negra sobre la HU2 Modificar filtros de la configuración de réplica de estructura

Caso de prueba	
Código: HU2	Historia de usuario: 2
Nombre: Modificar filtros de la configuración de réplica de estructura.	
Descripción: Prueba para la funcionalidad de modificar filtros de la configuración de réplica de esquemas.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” seleccionando al menos un tipo de acción. Se debe adicionar un filtro de estructura.	
Variables: “Esquema”, “Replicar por” y “Filtros”.	
Clase de equivalencia válida: Modificar filtros a la configuración de réplica de estructura correctamente.	
Entradas/Pasos de Ejecución: Todas las variables tienen sus valores de entrada	

correctos. Se selecciona la opción “Editar filtros”, donde el sistema muestra la ventana “Filtros de Estructura” para que el usuario modifique las condiciones de la réplica para el tipo de acción seleccionada. El usuario establece las condiciones deseadas mediante los diferentes filtros (nombre, prefijo, sufijo y/o tipo de acción). El sistema almacena los filtros modificados y muestra la pantalla “Nueva configuración”, manteniendo los enlaces “Editar filtros” y “Eliminar filtros” seguidos del enlace “Usuarios”.
Resultado Esperado: El sistema almacena los filtros en la configuración adicionada o cargada.
Clase de equivalencia inválida: Modificar filtros a la configuración de réplica de estructura incorrectamente.
Entradas/Pasos de Ejecución: La variable “Filtros” no obtuvo el valor esperado, o sea, se quedó algún campo vacío. Se selecciona la opción “Editar filtros”, donde el sistema muestra la ventana “Filtros de Estructura” para que el usuario modifique las condiciones de la réplica para el tipo de acción seleccionada. El usuario deja algún campo vacío mientras realiza el proceso de editar los filtros y presiona el botón “Aceptar”. El sistema muestra al usuario un mensaje informando que faltan datos por ingresar, manteniendo la misma pantalla.
Resultado Esperado: Mensaje de alerta informando que faltan datos por ingresar.
Evaluación de la Prueba: La prueba fue realizada con éxito.

Anexo 28 Prueba de caja negra sobre la HU3 Eliminar filtros de la configuración de réplica de estructura

Caso de prueba	
Código: HU3	Historia de usuario: 3
Nombre: Eliminar filtros de la configuración de réplica de estructura.	
Descripción: Prueba para la funcionalidad de eliminar filtros de la configuración de réplica de esquemas.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” seleccionando al menos un tipo de acción. Se debe adicionar un filtro de estructura.	
Variables: “Eliminar”.	
Clase de equivalencia válida: Eliminar filtros a la configuración de réplica de estructura	

correctamente.
Entradas/Pasos de Ejecución: Se selecciona el enlace “Eliminar filtros”, donde el sistema modifica los enlaces “Editar filtros” y “Eliminar filtros” por el enlace “Adicionar filtro” seguido del enlace “Usuarios”, y a su vez elimina los filtros creados para el tipo de acción especificado.
Resultado Esperado: El sistema elimina los filtros en la configuración adicionada o cargada.
Evaluación de la Prueba: La prueba fue realizada con éxito.

Anexo 29 Prueba de caja negra sobre la HU4 Aplicar filtros de esquemas a los grupos de estructura replicable

Caso de prueba	
Código: HU4	Historia de usuario: 4
Nombre: Aplicar filtros de esquemas a los grupos de estructura replicable.	
Descripción: Prueba para la funcionalidad de aplicar filtros de esquemas a los grupos de estructura replicable.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” por al menos un tipo de acción y adicionarle filtros. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de estructura.	
Variables: “Iniciar”.	
Clase de equivalencia válida: Aplicar filtros de esquemas a los grupos de estructura replicable correctamente.	
Entradas/Pasos de Ejecución: Se selecciona el botón “Iniciar”. El sistema comprueba que las acciones contenidas en los grupos de estructuras replicables están permitidas por los filtros asociados a las configuraciones registradas. El sistema aplica los cambios que son permitidos por los filtros.	
Resultado Esperado: El sistema aplica los cambios que son permitidos por los filtros asociados a las configuraciones registradas.	
Evaluación de la Prueba: La prueba fue realizada con éxito.	

Anexo 30 Prueba de caja negra sobre la HU5 Monitorizar el envío de los grupos del proceso de réplica de estructura

Caso de prueba

Código: HU5	Historia de usuario: 5
Nombre: Monitorizar el envío de los grupos del proceso de réplica de estructura.	
Descripción: Prueba para la funcionalidad de monitorizar el envío de los grupos del proceso de réplica de estructura.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de estructura.	
Variables: “Iniciar”.	
Clase de equivalencia válida: Monitorizar el envío de los grupos del proceso de réplica de estructura correctamente.	
Entradas/Pasos de Ejecución: Se selecciona el botón “Iniciar”. Se selecciona el módulo “Monitor de Réplica” donde el sistema muestra el evento “Enviado”, especificando los detalles del envío.	
Resultado Esperado: El sistema monitoriza el envío de los grupos.	
Evaluación de la Prueba: La prueba fue realizada con éxito.	

Anexo 31 Prueba de caja negra sobre la HU6 Monitorizar la recepción de los grupos del proceso de réplica de estructura

Caso de prueba	
Código: HU6	Historia de usuario: 6
Nombre: Monitorizar la recepción de los grupos del proceso de réplica de estructura.	
Descripción: Prueba para la funcionalidad de monitorizar la recepción de los grupos del proceso de réplica de estructura.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de estructura.	
Variables: “Iniciar”.	
Clase de equivalencia válida: Monitorizar la recepción de los grupos del proceso de réplica de estructura correctamente.	
Entradas/Pasos de Ejecución: Se selecciona el módulo “Monitor de Réplica” donde el sistema muestra el evento “Recibido”, especificando los detalles de la recepción.	
Resultado Esperado: El sistema monitoriza la recepción de los grupos.	

Evaluación de la Prueba: La prueba fue realizada con éxito.
--

Anexo 32 Prueba de caja negra sobre la HU7 Monitorizar la aplicación de los cambios de estructura

Caso de prueba	
Código: HU7	Historia de usuario: 7
Nombre: Monitorizar la aplicación de los cambios de estructura.	
Descripción: Prueba para la funcionalidad de monitorizar la aplicación de los cambios de estructura.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de estructura.	
Variables: “Iniciar”.	
Clase de equivalencia válida: Monitorizar la aplicación de los cambios de estructura correctamente.	
Entradas/Pasos de Ejecución: Al hacer clic en el botón “Iniciar” si hay cambios de estructura y el servidor de mensajería está online, cuando el sistema aplique el cambio sobre el grupo de estructura replicable debe monitorizar su aplicación.	
Resultado Esperado: El sistema monitoriza la aplicación de los cambios.	
Evaluación de la Prueba: La prueba fue realizada con éxito.	

Anexo 33 Prueba de aceptación sobre la HU2 Modificar filtros de la configuración de réplica de estructura

Caso de prueba	
Código: HU2_P1	Historia de usuario: 2
Nombre: Modificar filtros de la configuración de réplica de estructura.	
Descripción: Prueba para la funcionalidad de modificar filtros de la configuración de réplica de esquemas.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” por al menos un tipo de acción. Se debe adicionar un filtro en la opción “Adicionar filtros”.	
Entradas/Pasos de Ejecución: Al seleccionar la opción “Editar filtros” el sistema muestra la	

ventana “Filtros de Esquema” para que el usuario modifique las condiciones de la réplica para el tipo de acción seleccionada. El usuario modifica las condiciones deseadas mediante los diferentes filtros (nombre, prefijo, sufijo y/o tipo de acción).
Resultado Esperado: El sistema almacena los filtros en la configuración adicionada o cargada.
Evaluación de la Prueba: La prueba fue realizada con éxito.

Anexo 34 Prueba de aceptación sobre la HU3 Eliminar filtros de la configuración de réplica de estructura

Caso de prueba	
Código: HU3_P1	Historia de usuario: 3
Nombre: Eliminar filtros de la configuración de réplica de estructura.	
Descripción: Prueba para la funcionalidad de eliminar filtros de la configuración de réplica de esquemas.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura en la opción “Nueva configuración” del módulo “Configuración de Réplica”. Se debe seleccionar la opción “Replicar por” por al menos un tipo de acción. Se debe adicionar un filtro en la opción “Adicionar filtros”.	
Entradas/Pasos de Ejecución: Al seleccionar la opción “Eliminar filtros” el sistema mantiene la pantalla “Nueva configuración”, sustituyendo las opciones “Editar filtros” y “Eliminar filtros” por la opción “Adicionar filtros” en el tipo de acción seleccionada.	
Resultado Esperado: El sistema elimina los filtros de la configuración.	
Evaluación de la Prueba: La prueba fue realizada con éxito.	

Anexo 35 Prueba de aceptación sobre la HU6 Monitorizar el envío de los datos del proceso de réplica de estructura

Caso de prueba	
Código: HU6_P1	Historia de usuario: 6
Nombre: Monitorizar el envío de los datos del proceso de réplica de estructura.	
Descripción: Prueba para la funcionalidad de monitorizar el envío de los grupos de estructura replicables.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de	

estructura.
Entradas/Pasos de Ejecución: Al hacer clic en el botón “Iniciar” si hay cambios de estructura y el servidor de mensajería está online, cuando el sistema envíe el grupo de estructura replicable debe monitorizar su envío.
Resultado Esperado: El sistema publica el evento “Enviado” en el Monitor de Réplica.
Evaluación de la Prueba: La prueba fue realizada con éxito.

Anexo 36 Prueba de aceptación sobre la HU7 Monitorizar la aplicación de los cambios de estructura

Caso de prueba	
Código: HU7_P1	Historia de usuario: 7
Nombre: Monitorizar la aplicación de los cambios del proceso de réplica de estructura.	
Descripción: Prueba para la funcionalidad de monitorizar la aplicación de los cambios sobre los grupos de estructura replicables.	
Condiciones de Ejecución: Se debe adicionar una configuración de réplica de estructura. En el módulo “Configuración General” se debe “Iniciar” el envío de los cambios de estructura.	
Entradas/Pasos de Ejecución: Al hacer clic en el botón “Iniciar” si hay cambios de estructura y el servidor de mensajería está online, cuando el sistema aplique el cambio sobre el grupo de estructura replicable debe monitorizar su aplicación.	
Resultado Esperado: El sistema publica el evento “Aplicado” en el Monitor de Réplica.	
Evaluación de la Prueba: La prueba fue realizada con éxito.	

Anexo 37 Descripción del sub-requisito “Operabilidad”

Atributo de Calidad	Usabilidad.
Sub-atributos/Sub-característica	Operabilidad.
Objetivo	Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
Origen	Proveedor de requisitos.
Artefacto	Las interfaces de usuario pertenecientes a todos los requisitos del sistema.
Entorno	El sistema desplegado.

Estímulo	Respuesta: Flujo de eventos (Escenarios)
<<1>>. <<a>>< Número de pasos a ejecutar para llegar a los requerimientos del sistema >	
Interactuar con el módulo Configuración de Réplica.	<ol style="list-style-type: none"> 1. Una vez que el usuario se ha autenticado, el sistema brinda la opción de acceder al módulo de Configuración de Réplica, donde selecciona la opción “Nueva configuración”. 2. El usuario escribe las condiciones que desea para la réplica y presiona el botón “Guardar”. 3. El sistema guarda la configuración y muestra las configuraciones guardadas hasta el momento.
Medida de respuesta	
Navegar en el módulo “Configuración de Réplica”.	

Anexo 33 Descripción del sub-requisito “Protección contra errores de usuarios”

Atributo de Calidad	Usabilidad.
Sub-atributos/Sub-característica	Protección contra errores de usuarios.
Objetivo	Capacidad del producto para proteger a los usuarios de cometer errores.
Origen	Proveedor de requisitos.
Artefacto	Los mensajes de alerta para prevenir al usuario ante un error en la configuración y administración del producto.
Entorno	El sistema desplegado.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
<<1>>. <<a>>< Número de pasos a ejecutar para llegar a los requerimientos del sistema >	
El usuario intenta crear un filtro dejando algún campo sin datos.	1. El usuario selecciona los esquemas a replicar, especifica qué acciones desea

	<p>replicar y presiona el enlace “Adicionar filtro”.</p> <p>2. El sistema muestra la pantalla para adicionar filtros.</p> <p>3. El usuario presiona el botón “Aceptar”.</p> <p>4. El sistema muestra un mensaje de alerta, informándole al usuario que existen campos vacíos.</p>
Medida de respuesta	
Navegar en el módulo “Configuración de Réplica”.	

Anexo 34 Descripción del sub-requisito “Madurez”

Atributo de Calidad	Fiabilidad.
Sub-atributos/Sub-característica	Madurez.
Objetivo	Capacidad del producto para satisfacer las necesidades de fiabilidad en condiciones normales.
Origen	Proveedor de requisitos.
Artefacto	Los módulos Configuración de Réplica y Monitor de Réplica.
Entorno	El sistema desplegado.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
<<1>>. <<a>>< Número de pasos a ejecutar para llegar a los requerimientos del sistema >	
Los módulos funcionan correctamente bajo condiciones y períodos de tiempo determinados por el cliente.	NA
Medida de respuesta	
Desplegar el sistema con los módulos “Configuración de Réplica” y “Monitor de Réplica” integrados.	

Anexo 35 Descripción del sub-requisito “Analizabilidad”

Atributo de Calidad	Mantenibilidad.
Sub-atributos/Sub-característica	Analizabilidad.
Objetivo	Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias, causas de fallos en el software, o identificar las partes a modificar.
Origen	Equipo de desarrollo.
Artefacto	El código fuente.
Entorno	El ambiente de desarrollo del sistema.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
<<1>>. <<a>>< Impacto de un determinado cambio sobre el resto del sistema>	
Se evalúa el impacto de un cambio en los módulos.	<ol style="list-style-type: none"> 1. El equipo de desarrollo identifica los paquetes y clases implicados en el cambio, así como las funciones que se reutilizarán o se crearán para introducir el cambio. 2. Se evalúa el impacto partiendo de los resultados que arrojó el análisis anterior con respecto a la arquitectura del sistema.
<<1>>. <>< Diagnosticar las deficiencias o causas de fallos en el sistema>	Respuesta: Flujo de eventos (Escenarios)
Se diagnostica las deficiencias o causas de fallos en el sistema.	<ol style="list-style-type: none"> 1. El equipo de desarrollo identifica las posibles deficiencias o causas de fallos que se pueden originar en los módulos. 2. Se diagnostican las deficiencias o causas de fallos partiendo de los resultados que arrojó el análisis anterior.
<<1>>. <<c>>< Identificar las partes a modificar>	Respuesta: Flujo de eventos (Escenarios)
Se identifica las partes a modificar en el sistema.	<ol style="list-style-type: none"> 1. El equipo de desarrollo identifica los paquetes y clases implicados en el cambio, así como las funciones que se reutilizarán o se crearán para introducir el cambio.

Medida de respuesta
Analizar el cambio en los módulos “Configuración de Réplica” y “Monitor de Réplica”.

Anexo 36 Descripción del sub-requisito “Modificabilidad”

Atributo de Calidad	Mantenibilidad.
Sub-atributos/Sub-característica	Modificabilidad.
Objetivo	Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
Origen	Equipo de desarrollo.
Artefacto	El código fuente.
Entorno	El ambiente de desarrollo de los módulos.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
<<1>>. <<a>>< Capacidad de modificación del sistema>	
La arquitectura de los módulos está diseñada para brindar facilidades a la hora de introducir modificaciones en los módulos. Esto permite que se puedan introducir cambios o modificaciones, que no afecten el correcto desempeño del resto de la solución.	NA
Medida de respuesta	
Introducir una modificación a los módulos “Configuración de Réplica” y “Monitor de Réplica”.	

Anexo 37 Descripción del sub-requisito “Modificabilidad”

Atributo de Calidad	Mantenibilidad.
Sub-atributos/Sub-característica	Capacidad para ser probado.
Objetivo	Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

Origen	Equipo de desarrollo.
Artefacto	Diseños de casos de pruebas.
Entorno	El sistema desplegado.
Estímulo	Respuesta: Flujo de eventos
<<1>>. <<a>><< Facilidad con la que se pueden establecer criterios de prueba para el sistema>	(Escenarios)
Desde la concepción inicial de los módulos se definen las funciones según los requisitos funcionales y no funcionales a implementar, especificando los argumentos o variables de entrada y salida de los módulos, elementos que favorecen el proceso de identificación y elaboración de los distintos escenarios de prueba.	NA
Medida de respuesta	
Escenario de prueba de los módulos “Configuración de Réplica” y “Monitor de Réplica”.	

Anexo 38 Descripción del sub-requisito “Integridad funcional”

Atributo de Calidad	Adecuación funcional.
Sub-atributos/Sub-característica	Integridad funcional.
Objetivo	Grado en que el conjunto de funciones cubre todas las tareas y objetivos del usuario especificados.
Origen	Proveedor de requisitos.
Artefacto	Los módulos Configuración de Réplica y Monitor de Réplica.
Entorno	El sistema desplegado.
Estímulo	Respuesta: Flujo de eventos
<<1>>. <<a>><< Grado en el que el sistema cubre todas las tareas y objetivos especificados>	(Escenarios)

El desarrollo de los módulos está guiado por las necesidades expresadas por parte de los proveedores de requisitos, dándole total cumplimiento a sus especificaciones.	NA
Medida de respuesta	
Analizar las funcionalidades de los módulos “Configuración de Réplica” y “Monitor de Réplica”.	

Anexo 38 Descripción del sub-requisito “Corrección funcional”

Atributo de Calidad	Adecuación funcional.
Sub-atributos/Sub-característica	Corrección funcional.
Objetivo	Grado en que un producto o sistema proporciona los resultados correctos con el grado necesario de precisión.
Origen	Proveedor de requisitos.
Artefacto	Los módulos.
Entorno	El sistema desplegado.
Estímulo	Respuesta: Flujo de eventos (Escenarios)
<<1>>. <<a>>< Grado en el que el sistema proporciona los resultados correctos con precisión>	
El sistema realiza el proceso de réplica con la exactitud necesaria en cada uno de los nodos de réplica. En caso de que existan errores, la solución es capaz de gestionar los conflictos para mantener la integridad de las BD implicadas en el proceso de réplica de estructura.	NA
Medida de respuesta	
Navegar en el sistema.	

Anexo 38 Fragmento de código de la clase SenderManager para monitorizar el envío de las acciones

```
Sender Manager
/* Cuando el grupo se envía se monitoriza el envío */
public boolean send(ReplicableStructureGroup replicableStructureGroup) {
```

```

try {
    System.out.println(sendingStructure);
    if (!SendingStructure) {
        Set<String> targets =
            getTargetsForStructureReplics(replicableStructureGroup);
        if (targets.size() > 0) {
            if (sender.send(replicableStructureGroup, targets)) {
                logger
                    .info("Enviado al JMS el grupo de estructura replicable ["
                        + replicableStructureGroup.toString()+ "]);
                // monitorizar el envío del grupo de estructura replicable
                monitorStructureSend(replicableStructureGroup, targets);
            } else {
                logger.info("NO hay conexión con el JMS");
            }
        }
    }

    sendingStructure = true;
    logger.info(sendingStructure);
    return true; }
} catch (Exception e) {
    // TODO: Manipular el error mandando
    sendingStructure = false;
    logger.info(sendingStructure);
}
return false;
}
}

/** Monitoriza las acciones enviadas con sus destinos */
private void monitorStructureSend(ReplicableStructureGroup rsg,
    Set<String> targets) {
    // Compara si existen destinos para enviar el grupo de estructura
    // replicable hacia el destino
    if (monitoringManager.isListening() && targets.size() != 0) {
        monitoringManager.onSendingReplicableStructureGrup(
            replicableStructureGroup, targets);
        monitoringManager.onStoredSentGroupSizeChange();
        monitoringManager.onSentGroup
            (replicableStructureGroup.getId());
    }
}
}

```