

Universidad de las Ciencias Informáticas

Facultad 1



Título: Sistema para la administración y supervisión del sistema de réplica “Magic@ Data Replication eXtensible Solution”

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

Autores:

Martha María Rondón Regalado

Madiela Cabeza Beltran

Tutor:

Ing. Joe Luis Borrero Senciales

Ciudad de La Habana, Cuba

Curso 2011-2012

Dedicatoria

Dedico este trabajo a toda mi familia especialmente a mis padres y mis hermanos que siempre confiaron en mí y dieron todo de sí, para que pudiera llegar hasta aquí.

Martha

Dedico este trabajo a toda mi familia, en especial a mis padres, mis hermanos y mi novio por ser mi sostén en todo momento y por ser una familia tan maravillosa.

Madiela

Agradecimientos

A mis padres que siempre han estado cerca de mí, apoyándome y dándome todo su amor y confianza. En especial a mi mamá por educarme con todos los valores y principios de una persona de bien. Y a mi hermanita por estar siempre conmigo. A mi papito lindo por todos sus consejos y apoyo incondicional. A mi manito por ser el mejor hijo y hermano del mundo.

A mi putito por ser un ejemplo de perseverancia, responsabilidad y por estar cada día a mi lado, en las buenas y en las malas y por quererme en todas mis facetas.

A todos mis amigos, que siempre estuvieron ahí, para brindarme el consejo oportuno en las diferentes situaciones del destino.

A mi tutor que fue un apoyo fundamental en el desarrollo de esta investigación.

A Madi por aceptarme como su compañera de tesis y apoyarme en todo.

Martha

Agradecimientos

A mi mamita y a mi papito les agradezco toda mi vida porque siempre han estado cerca de mí en los buenos y en los malos momentos, constituyen mi principal guía y sin ellos no hubiera alcanzado mi sueño, por brindarme siempre su apoyo y confianza. A ustedes que representan mi más valioso tesoro.

A mis hermanos por brindarme su apoyo en todo momento.

A mi tuty por estar siempre a mi lado en las buenas y en las malas, por brindarme todo su comprensión, amor y cariño en estos seis años.

A mis abuelos y demás familiares que siempre han estado ahí para mí.

A todos mis amigos de la UCI que siempre los voy a llevar en mi corazón, por compartir momentos inolvidables en estos cinco años.

A mi tutor por confiar en nosotras y apoyarnos en esta investigación.

A Martica por soportarme y tener paciencia conmigo.

Madiela

Declaramos ser las autoras del trabajo titulado Sistema para la administración y supervisión del sistema de réplica “Mágic@ Data Replication eXtensible Solution” y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo el presente a los ____ días del mes de _____ del año 2012.

Martha María Rondón Regalado

Madiela Cabeza Beltran

Joe Luis Borrero Senciales

Resumen

En el Centro de Identificación y Seguridad Digital (CISED) se desarrolló una aplicación que optimiza y simplifica el proceso de sincronización de la información entre servidores de base de datos, que tiene por nombre “Magic@ Data Replication eXtensible Solution” y es conocida como “Réplica Mágic@”. El presente trabajo de diploma tiene como objetivo desarrollar una aplicación web para la administración y supervisión de esta réplica.

La investigación estuvo enmarcada fundamentalmente en el estudio del proceso de administración y supervisión del sistema de la “Réplica Mágic@”, para ello se utilizaron métodos empíricos y teóricos, los cuales posibilitaron una mejor comprensión de este proceso.

En estos momentos el mecanismo para llevar a cabo el proceso de administración y supervisión de la réplica se realiza a través de una terminal de comandos en Linux, lo que dificulta el trabajo con esta solución, además ante eventuales fallos en algunos de los servidores de réplica se ve afectada la disponibilidad de restaurar el servicio en el menor tiempo posible.

El sistema se desarrolla utilizando herramientas libres que consumen pocos recursos y permiten agilizar el proceso de implementación.

El despliegue de la aplicación beneficiará al centro con la reducción del tiempo a la hora de restablecer el servicio cuando ocurren fallos en los servidores de réplica, además visualizará de forma más clara el proceso de administración y supervisión de la réplica y permite que se pueda acceder a ella desde diferentes lugares, garantizando los niveles de seguridad en la información y mayor agilidad en los resultados.

Palabras clave: administración, réplica, supervisión.

Contenido	
Introducción	1
Capítulo I: Fundamentación Teórica.....	5
1.1. Introducción	5
1.2. Conceptos asociados al dominio del problema.....	5
1.2.1. Base de Datos	5
1.2.2. Base de Datos Distribuidas	6
1.2.3. Réplica de Datos	7
1.2.4. Magic@ Data Replication eXtensible Solution	9
1.2.5. Aplicación Web	10
1.2.6. Sistemas similares para la administración de réplicas.....	11
1.3. Estudio de las metodologías y estándares para el desarrollo del software.....	12
1.3.1. Proceso Unificado de Desarrollo del software (RUP).....	12
1.3.2. Programación Extrema (XP).....	13
1.3.3. Solución propuesta	15
1.4. Herramientas y lenguajes de desarrollo	15
1.4.1. Selección de lenguaje de programación	15
1.4.2. El Lenguaje Unificado de Modelado (UML).....	17
1.4.3. Framework para PHP	18
1.4.4. Uso de <i>software</i> libre	21
1.4.5. Entorno de Desarrollo Integrado (IDE).....	21
1.4.6. Doctrine como Modelo de Objeto-Relacional (ORM).....	24
1.4.7. Sistema Gestor de Base de Datos (SGBD)	25
1.4.8. Servidor web.....	27
1.4.9. Herramienta para el modelado	29
1.5. Conclusiones parciales.....	30
Capítulo II: Propuesta de Solución.....	31
2.1. Introducción	31
2.2. Descripción del sistema propuesto	31
2.2.1. Modelo de Dominio	32
2.3. Requisitos funcionales del sistema.....	34
2.4. Requisitos no funcionales del sistema.....	38
2.5. Fase de Exploración	39
2.5.1. Historias de Usuarios (HU)	39
2.6. Fase de Planeación	41
2.6.1. Estimación del esfuerzo por Historia de Usuario.....	41
2.6.2. Plan de iteraciones.....	42
Iteración 1	42
Iteración 2	43
Iteración 3	43
2.6.3. Plan de duración de las iteraciones	43

2.6.4.	Plan de Entrega.....	44
2.7.	Descripción de la Arquitectura	44
2.8.	Conclusiones parciales.....	46
Capítulo III: Implementación y Prueba		47
1.1.	Introducción	47
1.2.	Diseño de la solución	47
1.2.1.	Tarjetas CRC.....	47
1.2.2.	Diseño de la base de datos.....	48
1.3.	Implementación del sistema	50
1.3.1.	Iteración 1.....	50
1.3.1.1.	Tares de la ingeniería iteración 1	50
1.3.1.2.	Tareas de la ingeniería detalladas Iteración 1	51
1.3.2.	Iteración 2.....	51
1.3.2.1.	Tares de la ingeniería iteración 2	51
1.3.2.2.	Tareas de la ingeniería detalladas Iteración 2	52
1.3.3.	Iteración 3.....	52
1.3.3.1.	Tares de la ingeniería iteración 3.....	53
1.4.	Prueba.....	54
1.4.1.	Pruebas unitarias	55
1.4.2.	Pruebas de aceptación	55
1.4.2.1.	Iteración 1.....	56
1.4.2.2.	Iteración 2.....	58
1.4.2.3.	Iteración 3.....	59
1.5.	Conclusiones parciales.....	62
Conclusiones.....		63
Recomendaciones.....		64
Bibliografía		65
Glosario de Términos.....		68
Anexos.....		69

Índice de Figuras

Figura 1: Modelo de Dominio.....	33
Figura 2: Patrón MVC en Symfony.....	45
Figura 3 : Diagrama Entidad Relación	48
Figura 4: Diagrama de Despliegue.....	54
Figura 5: Portal de Hitachi Replication Manager.....	69
Figura 6: Prueba unitaria: testShowSincronizacionesIndex	104
Figura 7: Prueba unitaria: testLoginIndex	105
Figura 8: Prueba unitaria: testAdminIndex	105
Figura 9: Resultado de la prueba.....	105

Índice de Tablas

Tabla 1: Comparación entre RUP y XP	15
Tabla 2: Comparación entre PHP y JAVA	17
Tabla 3: Comparación entre Symfony, Codeigniter y Zend Framework	20
Tabla 4: Comparación entre Eclipse, Zend Studio y NetBeans.....	24
Tabla 5: Comparación entre PostgreSQL y MySQL.....	27
Tabla 6: Comparación entre Apache y Nginx	28
Tabla 7: HU Gestionar servidores de réplica	40
Tabla 8: HU Sincronizar cambios	40
Tabla 9: HU Gestionar usuarios	41
Tabla 10: Estimación del esfuerzo por HU	42
Tabla 11: Plan de duración de las iteraciones	43
Tabla 12: Plan de entrega	44
Tabla 13: Tarjeta CRC ClienteController	47
Tabla 14: Tareas de la ingeniería en la primera iteración	50
Tabla 15: HU2_T1	51
Tabla 16: Tareas de la ingeniería en la segunda iteración	51
Tabla 17: HU8_T1	52
Tabla 18: Tareas de la ingeniería en la tercera iteración	53
Tabla 19: HU12_T1	53
Tabla 20: Prueba de aceptación Gestionar servidores de réplica: Agregar Nuevo	56
Tabla 21: Prueba de aceptación Gestionar servidores de réplica: Listar	57
Tabla 22: Prueba de aceptación Gestionar servidores de réplica: Borrar	58
Tabla 23: Prueba de aceptación Sincronizar cambios	58
Tabla 24: Prueba de aceptación: Gestionar usuarios: Agregar Nuevo	59
Tabla 25: Prueba de aceptación Gestionar usuarios: Listar	60
Tabla 26: Prueba de aceptación Gestionar usuarios: Borrar	61
Tabla 27: HU Gestionar Rangos de IP	69
Tabla 28: HU Gestionar servidores de base de datos	70
Tabla 29: HU Gestionar ficheros.....	70
Tabla 30: HU Gestionar tipo de base de datos	71
Tabla 31: HU Gestionar esquema de sincronización	71
Tabla 32: HU Gestionar sincronización	72
Tabla 33: HU Planificar las tareas programadas.....	72
Tabla 34: HU Listar servidores de base de datos.....	73

Tabla 35: HU Listar sincronizaciones	73
Tabla 36: HU Gestionar grupos de usuarios	74
Tabla 37: HU Mostrar log de las sincronizaciones	74
Tabla 38: HU Autenticar usuario	75
Tabla 39: Tarjeta CRC AdminController	75
Tabla 40: Tarjeta CRC clase DsincronizacionRepository	75
Tabla 41: Tarjeta CRC DservidorDbRepository	76
Tabla 42: Tarjeta CRC DcambiosRepository	76
Tabla 43: Tarjeta CRC ActualizarListerner	76
Tabla 44: HU4_T1	76
Tabla 45: HU5_T1	77
Tabla 46: HU6_T1	77
Tabla 47: HU9_T1	78
Tabla 48: HU3_T1	78
Tabla 49: HU7_T1	78
Tabla 50: HU10_T1	79
Tabla 51: HU11_T1	79
Tabla 52: HU13_T1	80
Tabla 53: HU1_T1	80
Tabla 54: HU14_T1	80
Tabla 55: HU15_T1	81
Tabla 56: Prueba de aceptación Gestionar servidores de base de datos: Agregar nuevo.....	81
Tabla 57: Prueba de aceptación Gestionar servidores de base de datos: Listar	82
Tabla 58: Prueba de aceptación Gestionar servidores de base de datos: Borrar	83
Tabla 59: Prueba de aceptación Gestionar ficheros: Agregar nuevo	84
Tabla 60: Prueba de aceptación Gestionar ficheros: Listar	85
Tabla 61: Prueba de aceptación Gestionar ficheros: Borrar	86
Tabla 62: Prueba de aceptación Gestionar tipo de base de datos: Agregar nuevo	86
Tabla 63: Prueba de aceptación Gestionar tipo de base de datos: Listar	87
Tabla 64: Prueba de aceptación Gestionar tipo de base de datos: Borrar	88
Tabla 65: Prueba de aceptación Gestionar esquema de sincronización: Agregar nuevo	89
Tabla 66: Prueba de aceptación Gestionar esquema de sincronización: Listar	90
Tabla 67: Prueba de aceptación Gestionar esquema de sincronización: Borrar.....	91
Tabla 68: Prueba de aceptación Planificar las tareas programadas: Agregar nuevo	91
Tabla 69: Prueba de aceptación Planificar las tareas programadas: Listar	92
Tabla 70: Prueba de aceptación Planificar las tareas programadas: Borrar	93

Tabla 71: Prueba de aceptación Listar servidores de base de datos.....	94
Tabla 72: Prueba de aceptación Listar sincronizaciones.....	95
Tabla 73: Prueba de aceptación Gestionar sincronización: Agregar nuevo.....	95
Tabla 74: Prueba de aceptación Gestionar sincronización: Listar.....	96
Tabla 75: Prueba de aceptación Gestionar sincronización: Borrar.....	97
Tabla 76: Prueba de aceptación Gestionar Rangos de IP: Agregar nuevo.....	98
Tabla 77: Prueba de aceptación Gestionar Rangos de IP: Listar.....	99
Tabla 78: Prueba de aceptación Gestionar rangos de IP: Borrar.....	100
Tabla 79: Prueba de aceptación Gestionar grupo de usuarios: Agregar nuevo.....	100
Tabla 80: Prueba de aceptación Gestionar grupo de usuarios: Listar.....	101
Tabla 81: Prueba de aceptación Gestionar grupo de usuarios: Borrar.....	102
Tabla 82: Prueba de aceptación Mostrar log de las sincronizaciones.....	103
Tabla 83: Prueba de aceptación Autenticar usuario.....	103

Introducción

El uso de sistemas de bases de datos se desarrolló a partir de la necesidad de almacenar grandes cantidades de datos para su posterior consulta, producidas por las nuevas industrias que creaban gran cantidad de información. Estos sistemas no son más que: un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una base de datos puede considerarse una colección de datos variables en el tiempo. (Matos García, 1999)

Con el crecimiento de las organizaciones por medio de nuevos departamentos y el manejo de una gran cantidad de datos entre unidades dispersas geográficamente dentro de una misma empresa, surge la necesidad de una tecnología que sea capaz de acceder y procesar de forma rápida y segura toda la información. Aprovechando la evolución de los sistemas informáticos en las últimas décadas y el surgimiento de grandes redes de comunicación surgen las bases de datos distribuidas, que se definen como: colecciones de datos relacionados lógicamente, pero dispersos sobre diferentes sitios de una red de computadoras. Cada sitio en la red tiene capacidad de procesamiento autónomo y puede ejecutar aplicaciones locales. (Álvarez Carrión, 1999)

Estos sistemas de base de datos distribuidos permiten agregar nuevos nodos existiendo independencia y autonomía entre ellos, de fallar un nodo el sistema no se ve afectado. Estos sistemas posibilitan acceder de manera más rápida a los datos para consultarlos, surgiendo así la necesidad de tener todos los nodos sincronizados con la última información mediante diferentes mecanismos.

La técnica más usada para esto es la réplica de datos, que no es más que un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de base de datos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia. La replicación permite distribuir datos entre diferentes ubicaciones y entre usuarios remotos o móviles, mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e Internet. (Microsoft, 2012)

La mayoría de estos sistemas de réplicas están orientados a bases de datos relacionales en las cuales los datos están relacionados entre sí mediante interconexiones (relaciones entre las diferentes tablas) y casi en su totalidad son *software* propietario con un alto costo, además de estar implementados para gestores de base de datos específicos.

Por tal motivo se ha desarrollado una aplicación en el Centro de Identificación y Seguridad Digital (CISED) perteneciente a la Facultad 1 de la Universidad de las Ciencias Informáticas (UCI), que brinda un mecanismo de replicación de datos entre sistemas de base de datos distribuidas, independiente del gestor de base de datos utilizado. Este sistema tiene por nombre “Magic@ Data Replication eXtensible Solution” y es conocida como “Réplica Mágic@”. Los objetivos principales de esta solución son simplificar y optimizar el proceso de sincronización de la información entre servidores de base de datos.

Como **situación problemática** se tiene que todos los procesos para la administración y supervisión de la aplicación, que permite llevar a cabo el mecanismo de réplicas de datos, se realizan a través de una terminal de comandos en Linux.

- Esta forma de administrar el sistema de replicación tiene como principal desventaja que requiere de conocimientos del sistema operativo GNU/LINUX, dificultando así el trabajo con la misma.
- No se visualizan de forma clara los procesos de administración y supervisión.
- Se requiere de personal calificado o con dominio de los comando empleados para llevar a cabo los proceso.
- Se ve afectada la disponibilidad de restaurar el servicio en el menor tiempo posible ante eventuales fallos en algunos de los servidores de réplica.
- No permite el acceso desde diferentes lugares.

Para darle solución al problema generado se quiere desarrollar en el Centro de Identificación y Seguridad Digital (CISED) una aplicación web, que facilite el trabajo con los procesos de administración y supervisión del sistema de la réplica “Magic@ Data Replication eXtensible Solution”.

Queda definido como **problema científico**: ¿Cómo mejorar los procesos de administración y supervisión del sistema de la réplica “Magic@ Data Replication eXtensible Solution”?

Se tiene como **objeto de estudio**: los procesos para la administración y supervisión de la replicación de datos.

Se define como **objetivo general**: desarrollar una aplicación web en el Centro de Identificación y Seguridad Digital (CISED) para la administración y supervisión del sistema de la réplica “Magic@ Data Replication eXtensible Solution”.

El **campo de acción** queda definido como: los procesos para la administración y supervisión del sistema de la réplica “Magic@ Data Replication eXtensible Solution”.

La presente investigación propone la siguiente **idea a defender**: La implementación de un sistema para los procesos de administración y supervisión del sistema de la réplica de datos “Magic@ Data Replication eXtensible Solution”, garantizará los niveles de seguridad en la información y mayor rapidez en los resultados.

Para el desarrollo y cumplimiento al objetivo general se cuenta con las siguientes **tareas de investigación**:

- Fundamentación de los diferentes sistemas de administración y supervisión para la replicación de datos que existen a nivel mundial.
- Estudio teórico sobre los procesos de administración y supervisión del sistema de la réplica “Magic@ Data Replication eXtensible Solution”.
- Selección del Modelo de Objeto Relacional (ORM) para el trabajo con la base de datos.
- Fundamentación de las tecnologías, herramientas, metodología y lenguaje necesarios para el desarrollo de la aplicación.
- Modelación del ciclo del software.
- Elaboración de la base de datos.
- Implementación de los módulos del sistema para la administración y supervisión del sistema de la réplica “Magic@ Data Replication eXtensible Solution”.
- Validación de la propuesta de solución.

Los métodos de la investigación quedan definidos como:

Métodos teóricos:

- Histórico - Lógico: se origina un análisis y razonamiento de las herramientas utilizadas en la réplica de datos para lograr una mejor comprensión sobre el tema.

- Analítico - Sintético: se consultan varias bibliografías para profundizar y obtener una mejor visión de los objetivos planteados y así lograr una mejor propuesta de solución.

Métodos empíricos:

- La observación: se utiliza para recopilar una serie de datos que ayudan a la comprensión del problema planteado y al mismo tiempo posibilitan enmarcar la investigación hacia lo que se necesita alcanzar y como alcanzarlo.

Se define como **justificación de la investigación**: proporcionar al sistema de la réplica “Magic@ Data Replication eXtensible Solution” una forma rápida y segura de administrarlo y supervisarlos, a través de una aplicación web que le permita al usuario el acceso desde diferentes lugares y que cuente con una interfaz amigable para su uso. Además posibilita que los procesos de administración y supervisión del sistema de la réplica se realice de forma más clara y sencilla para el cliente, logrando rapidez en los resultados y mayor seguridad en la información.

Este documento está estructurado por tres capítulos, el contenido de los cuales se describe a continuación:

Capítulo 1: Fundamentación Teórica: Se exponen los principales conceptos como resultado de la investigación realizada sobre el objeto de estudio. Se realiza un análisis de las herramientas metodologías y tecnologías más utilizadas para analizar sus ventajas, desventajas y características y de esta manera proponer la adecuada para el desarrollo de la investigación. También se realiza un estudio del arte sobre los diferentes sistemas de réplica de datos que existen a nivel nacional e internacional.

Capítulo 2: Análisis y Diseño: Se incluye los requisitos funcionales y no funcionales del sistema. Además se obtienen las historias de usuarios como artefacto generado por la metodología utilizada, se realiza un análisis de la estimación del esfuerzo por cada una de las historias de usuario y se identifican las iteraciones para la implementación del sistema.

Capítulo 3: Implementación y Pruebas: Se hace alusión a las fases de implementación y pruebas propias de la metodología de desarrollo para guiar la implementación del sistema que se propone. Se exponen un conjunto de casos de pruebas mediante las cuales el cliente podrá validar la correcta implementación de cada uno de los requisitos funcionales.

Capítulo I: Fundamentación Teórica

1.1. Introducción

En el presente capítulo se abordan los conceptos fundamentales relacionados con las bases de datos, la réplica de datos y además se expone una breve explicación sobre la “Réplica Mágic@”. También se realiza un estudio del arte con relación a algunos sistemas que existen en el mundo y que abarcan el tema de réplica de datos, ofreciendo una visión general de las mismas. Además se analizan las metodologías más usadas en la actualidad y las herramientas más importantes a tener en cuenta para el desarrollo de la aplicación, brindando sus principales características para escoger las más indicadas a utilizar en la solución del problema a resolver.

1.2. Conceptos asociados al dominio del problema

Para un mejor entendimiento del problema inicial y de la solución propuesta se hace necesario anunciar los principales conceptos asociados con la replicación de datos.

1.2.1. Base de Datos

En la actualidad las bases de datos se han convertido en una de las principales herramientas que permiten tener una mayor disponibilidad de los datos. Dependiendo de los permisos otorgados a los usuarios les proporcionan el acceso a los mismos.

Las bases de datos, surgidas como respuesta al nuevo planteamiento de los sistemas orientados a la gestión de los datos, para mejorar la calidad de las prestaciones de los sistemas informáticos y aumentar su rendimiento, presentan una multitud de ventajas frente a los sistemas clásicos de ficheros, debido, sobre todo, a que se basan en una estructura de datos integrada y centralizada, eliminando así los problemas de redundancia y control de los datos. (Trejo Miranda, 2011)

Las ventajas de las bases de datos son:

- Múltiples usuarios puedan tener acceso a ella al mismo tiempo.
- Posibilita un control sobre la redundancia de los datos y la consistencia de los mismos.

- Proporcionan una mejora en cuanto a la accesibilidad, gracias a que estos proporcionan lenguajes de consultas que permiten al usuario realizar cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador implemente una aplicación para ejecutar dicha tarea.

1.2.1.1. Sistema de Gestión de Base de Datos (SGBD)

Surge por la necesidad de contar con una técnica de administración que permita llevar un control no solo en los datos sino también en los usuarios. Se denomina SGBD al software que permite la utilización y/o actualización de los datos almacenados en una o varias bases de datos, por uno o varios usuarios desde diferentes puntos de vista y a la vez. (Matos García, 1999)

Además cuidan la seguridad de la información almacenada en las base de datos ante posibles fallos del sistema y ante los intentos de acceso no autorizado. Su misión es proporcionar métodos eficientes para definir, almacenar y recuperar la información contenida en la base de datos.

1.2.2. Base de Datos Distribuidas

Las bases de datos pueden encontrarse almacenadas en una computadora y gestionadas por una aplicación instalada en la propia computadora; en tal caso se le denomina a esta base de datos local o centralizada. En el caso de que las bases de datos a las que se tendrá acceso desde un ordenador determinado, se encuentren situados en varios de estos, se dice que el sistema de base de datos es distribuido. (Núñez, 2004)

Una base de datos distribuida no es más que un conjunto de múltiples bases de datos lógicamente relacionadas, las cuales se encuentran distribuidas entre diferentes sitios interconectados por una red de comunicaciones. (Martínez Martín, 2010)

Existen varias razones para desarrollar sistemas distribuidos que incluyen, el compartimento de información y distribución del control, donde cada localidad tiene cierto grado de control sobre sus datos. Además permiten una mayor fiabilidad y disponibilidad de los datos, puesto que si se produce un fallo en alguno de los nodos o localidades lo más probable es que los restantes puedan seguir funcionando.

Las bases de datos distribuidas también tienen sus desventajas como, el aumento de la complejidad para garantizar una coordinación adecuada entre localidades, la cual se refleja en el desarrollo de *software* cada vez más costosos. Mayor probabilidad de errores, ya que los nodos que constituyen el sistema funcionan en paralelo.

También cuentan con una mayor sobrecarga de procesamiento, donde el intercambio de mensajes y ejecución de algoritmos, para el mantenimiento de la coordinación entre los nodos, es una forma de tiempo extra que no existe en los sistemas centralizados.

1.2.2.1. Sistema Gestor de Bases de Datos Distribuidas (SGBDD)

El SGBDD es el *software* que gestiona todo lo relacionado con las bases de datos distribuidas y subministra mecanismos de acceso que hacen la distribución transparente para el usuario. (Delgado, et al., 2008)

Un SGBDD está compuesto por una única base de datos lógica dividida en una serie de fragmentos. Cada fragmento se almacena en una o más computadoras bajo el control de un SGBD independiente, estando dichas computadoras conectadas mediante una red de comunicaciones. (Delgado, et al., 2008)

El SGBDD proporciona servicios avanzados de recuperación ante posibles fallos de alguno de los nodos o en algunos enlaces de comunicación, la capacidad de decidir qué versión de la copia de un elemento de información es la que tiene que ser accedida en un momento determinado. Brinda mecanismos avanzados para el control de la concurrencia, permitiendo mantener la coherencia de los datos distribuidos. Además permite tener acceso a nodos remotos, facilitando la transferencia de consultas y datos entre estos, también aporta un control estricto de la seguridad a la hora de otorgar los privilegios de acceso a los datos.

1.2.3. Réplica de Datos

La replicación de los datos se ha convertido en una parte fundamental dentro de la estrategia de protección de datos, que llevan a cabo las organizaciones, proporcionando una manera rápida y confiable de diseminar la información corporativa entre múltiples localizaciones en un ambiente de negocio distribuido, permitiendo distribuir la información de manera confiable. (Microsoft, 2012)

La réplica de datos permite transparencia ante los usuarios, estos no necesitan saber sobre la replicación. Si existen réplicas de objetos en una base de datos, esta debe ser controlada por el sistema manejador de base de datos distribuida y no por el usuario. La función principal de la transparencia es mantener la consistencia entre las copias, y que funcione de forma transparente a las aplicaciones.

Las razones para replicar una base de datos incluyen:

- **Disponibilidad:** los datos van a ser más accesibles al tener varias copias.
- **Mayor rendimiento:** se mejora para las transacciones de consulta, cuando se introduce la replicación en un sistema que estuviera afectado por sobrecarga de recursos centralizados.
- **Fiabilidad:** al haber múltiples copias de los datos disponibles en el sistema, se dispone de un mecanismo excelente de recuperación cuando existan fallos en algunos de los nodos. (SlideShare Inc, 2008)

Existen dificultades a la hora de replicar datos, pues se necesita mantener actualizadas todas las copias de ese objeto o dato replicado y esto trae consigo una sobrecarga en las actualizaciones.

Modelo de la replicación

Sincrónico (Fraire, 2005)

Este tipo de replicación proyecta el proceso de actualización de réplicas en el momento en que una réplica es modificada. En este momento, se establece contacto con el servidor responsable de la réplica y se le informa de la actualización. El servidor actualiza la copia primaria del dato modificado y esta información es enviada a todos los sitios en que existe una copia de la réplica modificada.

Este tipo de estrategia de replicación se caracteriza por mantener un alto nivel de consistencia entre las réplicas. Genera un alto grado de sobrecarga en la red y no permite que los usuarios puedan trabajar desconectados. Un sistema que opere con este mecanismo de replicación solo soporta clientes tradicionales ya que no admite manejar clientes móviles. Además la sincronización sólo se suele hacer en un centro de datos o a una distancia muy corta, inferior a 75 kilómetros.

Asincrónico (Fraire, 2005)

La tecnología de replicación asíncrona es la más reciente alternativa para proporcionar tolerancia ante fallos en servidores y almacenamiento en red. Este tipo de replicación emite el proceso de actualización de réplicas en intervalos regulares de tiempo. Requiere que todos los servidores almacenen localmente una bitácora con las modificaciones que se van realizando sobre las réplicas.

Una vez que inicia el proceso de sincronización con los servidores primarios, las bitácoras son enviadas a dichos servidores y estos realizan el proceso de reconciliación de réplicas para determinar sus valores actuales.

Finalmente se realiza el proceso de actualización de todas las réplicas administradas por cada servidor primario modificado. En el momento en que se modifica una de las réplicas, se establece contacto con el servidor responsable de la réplica y se le informa de la actualización. El servidor actualiza la copia primaria del dato modificado y esta información es enviada a todos los sitios en que existe una copia de la réplica modificada.

1.2.4. Magic@ Data Replication eXtensible Solution

Magic@: Es una suite de aplicaciones que permiten la configuración e implantación de una red compleja de replicación de datos. Esta réplica es una solución de réplica multi-maestra asincrónica que se basa en el control de cambios en las bases de datos relacionales y la replicación de las mismas. Brinda solución a la problemática de la réplica entre bases de datos implementadas en cualquier gestor y es multiplataforma. Permite la implementación de sistemas de bases de datos distribuidas, aislando la complejidad del sistema del mecanismo de replicación de los datos. (Landrian, 2009)

Se compone de tres herramientas:

- **RDBTool:** Permite la conexión con diferentes gestores de base de datos e incorpora plantilla de generación de scripts que son necesarias para la réplica de datos.
- **ReplicationSchemaTool:** Se encarga del diseño visual del mapeo de las tablas y permite la creación de filtros de la información a replicar.
- **DataSynchronizaer:** Esta aplicación es la encargada de realizar el proceso de sincronización entre dos servidores, realiza en paralelo el proceso de sincronización desde la base de datos local a la base de datos remota y el proceso en sentido contrario. Es una herramienta fácilmente configurable.

Esta investigación se basa específicamente en el componente DataSynchronizaer porque es la herramienta encargada de desarrollar el proceso de sincronización en la réplica "Magic@ Data Replication eXtensible Solution".

1.2.5. Aplicación Web

En sus inicios la web solo consistía en una colección de páginas estáticas, documentos, entre otros elementos, para su consulta o descarga. Con el desarrollo de la ciencia y las nuevas tecnologías se ha hecho necesario mantener una comunicación activa entre el usuario y la información, además se demandan cada vez más aplicaciones ligeras, robustas y rápidas que permitan utilizar la web. Una aplicación web es una aplicación informática distribuida cuya interfaz de usuario es accesible desde un cliente web, normalmente un navegador web. (Software, 2004)

1.2.5.1. Ventajas de las aplicaciones web

Los medios distribuidos han tenido un alto auge en los últimos años colocándose en primer lugar ante las pesadas aplicaciones de escritorio que requieren muchas veces de instalaciones previas y dispositivos auxiliares para la perfecta visualización de la misma. Las páginas web brindan la posibilidad del libre acceso de navegación y de interacción, sin tener que instalar ni copiar nada en la máquina.

Algunas de las ventajas que ofrecen las aplicaciones web son:

- Permiten ahorrar tiempo, sin la necesidad de instalar ningún programa.
- Pueden realizar tareas muy sencillas.
- Multiplataforma, ya que se pueden usar desde cualquier sistema operativo.
- No ocupan espacio en las computadoras personales de los usuarios, ni consumen muchos recursos. Para utilizar estas aplicaciones solo se requiere de una computadora con un buen navegador web y acceso a Internet.
- Se facilita el trabajo a distancia.
- Actualizaciones inmediatas: Como el cliente utiliza solamente el navegador web, todas las actualizaciones de *software* se realizan en el servidor, por lo que siempre se estará usando la última versión disponible.

1.2.5.2. Estructura de una aplicación Web

Una aplicación web está normalmente estructurada como una aplicación de tres-capas. La primera capa la brinda el navegador web y la capa intermedia es la encargada de manejar y gestionar toda la información y constituye un motor capaz de usar alguna tecnología web dinámica como por ejemplo PHP y ASP.NET.

La última capa la constituye una base de datos donde se guarda toda la información. La primera capa envía peticiones a la segunda o intermedia y proporciona una interfaz de usuario valiéndose de actualizaciones y consultas a la base de datos.

1.2.6. Sistemas similares para la administración de réplicas

Las aplicaciones web pueden ser utilizadas para la gestión de la información o del negocio de una empresa, ofreciendo soluciones para optimizar el rendimiento y la relación con los clientes. Brindan un acceso seguro y rápido a los datos desde cualquier ubicación, posibilitando realizar servicios online que sustituyan muchas de las tareas administrativas.

EMC Replication Manager

EMC Corporation es un fabricante de *software* y sistemas para administración y almacenamiento de información. Uno de sus productos es EMC Replication Manager que administra la replicación a través de una consola de administración centralizada. Este producto simplifica la replicación y pone los datos correctos en el lugar indicado en el momento adecuado. Ahorra tiempo y dinero al tiempo que aumenta la protección de datos. Este proceso consiste en crear réplicas de una aplicación coherente de Microsoft Exchange, SQL Server, SharePoint, VMware, Oracle, en los sistemas de archivos personalizados. También agiliza la gestión y automatización de EMC Recover Point, Time Finder, Snap View, Snap Sure Celerra Replicator y SAN Copy. Gestiona las tecnologías de replicación de EMC y coordina el proceso de replicación para el funcionamiento de múltiples réplicas basadas en discos. (EMC Corporation , 2012)

Este sistema realiza la administración de la replicación a través de una terminal de comandos y lo que se pretende desarrollar es una aplicación web, por lo que esta herramienta solo se estudia para profundizar en cómo realiza el proceso de administración.

Hitachi Replication Manager

Hitachi Replication Manager es una parte de Hitachi Storage Command Suite, esta configura, controla y gestiona dicho sistema. Aprovecha las potentes capacidades de almacenamiento del sistema de replicación de Hitachi y combina de forma sólida las aplicaciones de protección e información, lo que refleja las características previamente disponibles en ofertas separadas.

También disminuye la complejidad de la gestión y aumenta la productividad del personal, proporcionando un mayor control a través de una única interfaz de usuario consistente. Permite configurar todos los productos de replicación de Hitachi sin necesidad de otras herramientas, sistemas de almacenamiento local y remoto, incluyendo configuraciones de puerto de comunicaciones, ofrece soporte para múltiples centros de datos y varios sistemas de almacenamiento en cada centro de datos. El diseño basado en funciones de control de acceso, logra el estricto control de los usuarios múltiples. La gestión de tareas permite la programación y configuración automática de los datos replicados. (Hitachi Data Systems Corporation, 2012) Ver Anexo # 1.

Esta herramienta se desarrolló específicamente para la suite Hitachi, por lo que no sería viable utilizar este sistema, ya que se quiere desarrollar una aplicación específicamente para la “Réplica Mágic@”.

1.3. Estudio de las metodologías y estándares para el desarrollo del software.

Ante la necesidad de utilizar una serie de procedimientos, herramientas, soporte documental y técnicas a la hora de desarrollar un producto, surgen las metodologías de desarrollo. Estas se encargan de guiar a los desarrolladores a la hora de realizar un *software* nuevo.

Existen dos tipos de metodologías las ágiles y las tradicionales, esta última es necesaria en proyectos donde la organización juega el papel fundamental en cuanto al modelado, redacción de la documentación, definición de roles, actividades y artefactos. Sin embargo las metodologías ágiles están orientadas al intercambio entre el cliente y el desarrollo del *software*, pues en intervalos de corto tiempo se le muestra al cliente versiones funcionales del producto para que pueda valorar y proponer cambios en el mismo.

1.3.1. Proceso Unificado de Desarrollo del software (RUP)

Dentro de las metodologías tradicionales, se encuentra RUP, llamada así por sus siglas en inglés *Rational Unified Process*, es uno de los métodos más utilizados dentro de los métodos tradicionales. RUP es un proceso formal: provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto).

Dentro de las características fundamentales de RUP se puede encontrar que es dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental.

RUP divide el proceso en cuatro fases:

-**Inicio:** se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

-**Elaboración:** fase en la que se define la arquitectura del sistema.

-**Construcción:** se obtiene un producto listo para su utilización, documentado y con un manual de usuario.

-**Transición:** el *release* ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Además define nueve flujos de trabajo distintos, separados en dos grupos: los flujos de trabajo de ingeniería y los flujos de trabajo de apoyo. (G. Figueroa, y otros, 2010)

1.3.2. Programación Extrema (XP)

XP por sus siglas de inglés *Extreme Programming*. Es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación permanente entre cliente y equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Además se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el *software* que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. Como segundo objetivo se encuentra potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del *software*.

A continuación se muestran las fases en la que se divide el ciclo de vida de XP:

Fase de la exploración: en esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Esta fase toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase del planeamiento: se priorizan las historias de usuario y se acuerda el alcance del *release*. Los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma. La duración del cronograma de la primera entrega del producto no excede normalmente dos meses. La fase de planeamiento toma un par de días. Se deben incluir varias iteraciones para lograr una primera versión del producto.

El cronograma fijado en la etapa de planeamiento se realiza a un número de iteraciones, cada una toma de una a cuatro semanas en ejecución. La primera iteración crea un sistema con la arquitectura del sistema completo, seleccionando las historias que harán cumplir la construcción de la estructura para el sistema completo. El cliente decide las historias que se seleccionarán para cada iteración. Las pruebas funcionales creadas por el cliente se ejecutan al final de cada iteración. Al final de la última iteración el sistema está listo para producción.

Fase de producción: requiere prueba y comprobación extra del funcionamiento del sistema antes de que este se pueda liberar al cliente. En esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de si se incluyen o no en el *release* actual. Durante esta fase, las iteraciones pueden ser aceleradas de una a tres semanas. Las ideas y las sugerencias postpuestas se documentan para una puesta en práctica posterior por ejemplo en la fase de mantenimiento. Después de que se realice el primer *release* productivo para uso del cliente, el proyecto de XP debe mantener el funcionamiento del sistema mientras que realiza nuevas iteraciones.

Fase de mantenimiento: requiere de un mayor esfuerzo para satisfacer también las tareas del cliente. Así, la velocidad del desarrollo puede desacelerar después de que el sistema esté en la producción. La fase de mantenimiento puede requerir la incorporación de nueva gente y cambiar la estructura del equipo.

Fase de muerte: es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo. (Solis Calero, 2003)

1.3.3. Solución propuesta

Tabla 1: Comparación entre RUP y XP

Aspectos	RUP	XP
Artefactos y roles	Más	Pocos
Cliente	Interactúa con el equipo de desarrollo mediante reuniones	Parte del equipo de desarrollo
Proceso	Mucho más controlado, con numerosas políticas.	Menos controlado, con pocos principios.
Arquitectura de software	Esencial y se expresa mediante modelos.	Menos énfasis
Cantidad de personas	Grupo grandes.	Grupos pequeños (menos de 10) y trabajando en el mismo sitio.

Después de un análisis se escoge una metodología de desarrollo ágil, ya que se cuenta con un grupo resumido de personas en el equipo de trabajo y se decide utilizar la metodología XP, esta específicamente siempre está en constante interacción con el cliente lo cual favorece mucho al desarrollo de la aplicación web, para establecer cambios en caso que sea necesario e informar al cliente de forma muy rápida en el momento requerido para satisfacerlo.

1.4. Herramientas y lenguajes de desarrollo

Para la selección de las herramientas y lenguajes de desarrollo empleados en la propuesta de solución se realizó un estudio, para escoger aquellas que respondieran a las necesidades del sistema a implementar. A continuación se mostrara el resultado del estudio realizado.

1.4.1. Selección de lenguaje de programación

Cuando se elabora un sistema informático es necesario utilizar uno o varios lenguajes de programación para lograr las funcionalidades deseadas. A continuación se tratan brevemente los lenguajes estudiados haciendo énfasis en sus principales características, para escoger el más indicado en el desarrollo de la aplicación.

PHP

Hoy en día existen varios lenguajes de programación del lado del servidor, entre los que se encuentran: ASP.net, JSP, Perl y PHP. Este último es un acrónimo que significa por sus siglas en inglés: *Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*)

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (*open source*). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML. (GÓMEZ, y otros, 2005)

Este lenguaje en sus inicios fue creado para que los desarrolladores elaboren páginas web dinámicas, y así garantizar que se desarrollen de forma más eficiente y rápidas.

PHP es un producto de libre distribución que permite a los programadores un fácil aprendizaje y utilización ya que emplea técnicas de Programación Orientada a Objeto (POO) y no requiere definición de tipos de variables, por lo que se hace más fácil su uso. Además puede interactuar con la mayoría de los servidores web como Apache, está disponible para muchos sistemas como GNU/Linux, Windows, UNIX, entre otros.

Hay varios *framework* desarrollados en PHP posibilitando a los desarrolladores que pueden crear aplicaciones web de forma rápida, eficiente y estructurada. Día a día van ganando más popularidad en cuanto a su utilización, dentro de los que se pueden encontrar: Kumbia, CakePHP, Zend Framework y Symfony.

JAVA

Desde el principio JAVA fue diseñado como un lenguaje orientado a objetos. Es un lenguaje multiplataforma pues el mismo código java que funciona en un sistema operativo, funcionará en cualquier otro que tenga instalada la máquina virtual java, en otras palabras soportan aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows NT, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Fue diseñado para crear *software* altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. (Álvarez Marañón, 1999)

Este lenguaje de programación es multihilo, permite realizar muchas actividades simultáneas en un programa. Además se trabaja con cálculos de alto nivel de complejidad y con grandes cantidades de datos.

1.4.1.1. Solución propuesta

Tabla 2: Comparación entre PHP y JAVA

Aspectos	PHP	JAVA
Fácil aprendizaje	X	
Páginas más rápidas y ligeras	X	
Seguro y estable	X	
Software Libre	X	X
Integración Externa		X

Después de un análisis se concluye que PHP presenta las características y ventajas necesarias para desarrollar la aplicación web, las páginas PHP son muchos más rápidas y ligeras, también existen muchas más plataformas de desarrollo, desde VIM, ZendStudio, NetBeans, Eclipse, para PHP que para otro lenguaje de programación. Es un *software* libre, no es necesario pagar para poder utilizarlo. Debido a su sintaxis simple y entendible, es muy fácil de aprender. PHP es un lenguaje muy utilizado en el mundo y muchos programadores están vinculados para la mejora de su motor, por lo cual es cada vez más seguro y estable a medida que pasa el tiempo y aumenta su versión. La aplicación a desarrollar no requiere de grandes cálculos, ni de realizar muchos procesos a la vez por lo que PHP se adapta mejor como lenguaje de programación.

1.4.2. El Lenguaje Unificado de Modelado (UML)

Como soporte de la modelación se escogió el Lenguaje Unificado de Modelado UML, llamado así por sus siglas en inglés *Unified Modeling Language*, es el lenguaje de modelado más conocido en el mundo. Para solucionar un problema, UML permite a través de diagramas y esquemas, representar conceptualmente el sistema.

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos (modelos) de un sistema, desde una perspectiva orientada a objetos. (Molina García, 2008)

Para conformar los diagramas el UML combina muchos elementos gráficos para así presentar diversas perspectivas de un sistema. Un modelo UML describe lo que supuestamente hará el *software*, pero no especifica cómo implementarlo. Posteriormente ha surgido una nueva versión de este lenguaje, la versión 2.0, tiene como objetivo hacer el lenguaje de modelado mucho más extensible de lo que era anteriormente y permitir la validación y ejecución de modelos creados mediante el UML, también se plantea cubrir otros campos como el modelado de negocio, de sistemas de tiempo real basado en componentes, entre otros. (Mora, y otros, 2006)

Deja atrás la creencia de que este lenguaje no es de programación, pues permite la creación de herramientas que crean códigos ejecutables a partir de sus modelos.

1.4.3. Framework para PHP

Un *framework* simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un *framework* facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (Zaninotto, y otros, 2008)

Para brindarles a los programadores y diseñadores una mejor estructura y organización en los proyectos, son desarrollados los *frameworks*. Estos proporcionan una estructura bien definida para crear aplicaciones web con mayor rapidez.

Symfony

Symfony es uno de los *frameworks* para PHP más utilizados en el mundo entero y está precisamente desarrollado para optimizar el desarrollo de aplicaciones web, este separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web por lo que emplea el tradicional patrón de diseño Modelo Vista Controlador (MVC). El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos. Para reducir el tiempo en el desarrollo de aplicaciones web complejas, este *framework* proporciona varias herramientas y clases. Además automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, entre otras.) como en plataformas Windows. (Zaninotto, y otros, 2008)

Zend Framework

Zend Framework (ZF) es un marco de desarrollo (*framework*) para PHP, libre, orientado a objetos que brinda una arquitectura flexible. Provee un núcleo para el manejo del patrón de diseño MVC y posee algunas librerías que lo hacen perder acoplamiento y ganar flexibilidad. (Tupe, y otros, 2008)

La estructura de los componentes es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. Aunque se pueden utilizar de forma individual, al combinar los componentes de la biblioteca estándar de Zend Framework, conforman un potente y extensible *framework* para el desarrollo de aplicaciones web, que ofrece un gran rendimiento y una robusta implementación del patrón MVC; una capa de abstracción de datos fácil de utilizar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. (Nazep, 2011)

Codeigniter

Codeigniter es un *framework* para el desarrollo de aplicaciones dirigidas a personas que quieran construir sitios usando PHP. Su objetivo es poder hacer los proyectos más rápidos de lo que se pueden hacer, suministrando un rico conjunto de librerías para tareas comunes, con una simple interfaz y estructura lógica para acceder a estas librerías. Codeigniter deja que su creatividad se centre en el proyecto minimizando la cantidad de código necesitado para ejecutar una tarea. Está liberado bajo la licencia de código abierto Apache-BSD¹ lo que significa que es totalmente libre y puede ser usado.

¹ Otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*), pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercano al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Entre sus características se puede encontrar su compatibilidad con PHP 4 y PHP 5, posee soporte para múltiples bases de datos entre las que se pueden mencionar PostgreSQL, MySQL, MSSQL, además se le pueden incluir *drivers* para otros gestores de bases de datos que se programen, incluye también plantillas, validaciones, no requiere instalación, se puede encontrar una librería con un número de clases que incluye para trabajar con distintos objetos: calendario, bases de datos, correo electrónico, manipulación de imágenes, FTP, lenguaje, tablas, sesiones y compresión ZIP.

Requiere sólo de unas bibliotecas muy pequeñas. Esto está en contraste con muchos *frameworks* que requieren significativamente más recursos. Las bibliotecas adicionales serán dinámicamente cargadas según la demanda, basada en sus necesidades por un proceso dado, por lo que es bastante rápido. Puede extenderse fácilmente a través del uso de *plugins* y bibliotecas de ayuda, o a través de extensiones de clases. Está basado en el patrón Modelo Vista Controlador (MVC). (ExpressionEngine, 2006)

1.4.3.1. Solución propuesta

Tabla 3: Comparación entre Symfony, Codeigniter y Zend Framework

Aspectos	Symfony	Codeigniter	Zend Framework
Rapidez	X		
Estructura de programación (MVC)	X	X	X
Soporte para varios lenguajes en los archivos de configuración	X		
Generación de clases controladoras de forma automática	X		X
Software libre	X	X	X
Mapeado de objetos a bases de datos relacionales (ORM)	X	X	

Después de un análisis se escoge como *framework* de desarrollo Symfony, este contiene una gran variedad de herramientas y clases para conseguir acortar el tiempo de desarrollo de aplicaciones web complejas.

Específicamente se elige en su versión 2.0.6, donde los archivos de configuración se pueden escribir en PHP, XML o YAML. Internamente Symfony 2 crea una nueva estructura de directorios para los proyectos, desaparecen los *plugins* y se sustituyen por un elemento mucho más potente llamado *bundle*. Posee buenas prácticas no solo de PHP, sino que coge ideas de diversos *frameworks* como Django, Spring, Java entre otros.

Symfony2 va más allá de mostrar el mensaje de error, muestra el seguimiento de la pila entera y las excepciones vinculadas, para facilitar el trabajo del programador; tanto es así que divide la lógica del negocio y la presentación de la información con la utilización del modelo vista controlador integrado. Esta versión de Symfony es mucho más rápida que las anteriores y requiere utilizar como mínimo una versión de PHP 5.3.2.

1.4.4. Uso de *software* libre

El uso de las computadoras hoy en día a nivel mundial se ha convertido en un hecho fundamental del proceso de modernización de la sociedad en general. Cuba no escapa de este proceso que se ha hecho muy creciente. Con la utilización de nuevos *software* libre tampoco ha dejado de hacerlo.

Ubuntu 10.04

El tema de *software* libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el *software*. Ubuntu es una de las grandes distribuciones de Linux y precisamente una de las nuevas versiones Ubuntu 10.04 hace que el sistema sea seguro, robusto y flexible. Con ello se mejora en la velocidad de arranque, encender o reiniciar la PC, es ahora mucho más rápido. Además tiene un aspecto gráfico mejorado, superando las expectativas de los usuarios. También facilita a los nuevos beneficiarios la instalación de aplicaciones en este sistema, ya que con esta versión se han hecho mejoras en el centro de *software* de Ubuntu.

1.4.5. Entorno de Desarrollo Integrado (IDE)

Los Entornos de Desarrollo Integrados, en inglés *Integrated Development Environment* (IDE) son programas que agrupan un conjunto de herramientas que viabilizan la labor de los programadores.

Los componentes esenciales que los definen son:

- Editor de texto (código).
- Compilador.
- Intérprete.
- Depurador.
- Herramientas de automatización (completamiento de código y navegación rápida dentro del código).
- Sistema de control de versiones.
- Diseñador de interfaces gráficas.
- Soporte para varios lenguajes de programación y plataformas de desarrollo.

No todos poseen las propiedades anteriores, ni las implementan al mismo nivel, pero estas son las más usuales.

Eclipse

Eclipse sirve como IDE para Java y cuenta con numerosas herramientas de desarrollo de *software*. También da soporte a otros lenguajes de programación, como son: C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (*plugins*) para extender la funcionalidad. Eclipse es una plataforma libre, robusta, completa, y de calidad para el desarrollo de aplicaciones.

La última versión estable de Eclipse se encuentra disponible para los sistemas operativos Windows, Linux, Solaris, AIX, HP-UX y Mac OSX. Todas las versiones de Eclipse necesitan tener instalado en el sistema una máquina virtual Java (JVM), preferiblemente JRE (*Java Runtime Environment*) o JDK (*Java Developer Kit*) de Sun. (Seoane Pascual, y otros, 2007)

A pesar de que las funcionalidades de Eclipse son variadas, gran parte de estas permite que nuevos componentes puedan utilizar nuevos tipos de contenido, para realizar nuevas tareas con contenidos existentes. La plataforma Eclipse permite descubrir, e invocar funcionalidad implementada en componentes llamados *plugins*. Cuando la plataforma se inicializa, se le mostrará al usuario además del entorno de Eclipse todos los *plugins* que tiene instalado el entorno. (Garrido Montero, 2004)

Zend Studio

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. (Alvarez, 2003)

Zend Studio es el único IDE para desarrolladores profesionales que ofrecen las capacidades necesarias para desarrollar aplicaciones de negocio. Características como refactorización, la generación de código, asistente de código y análisis semántico se combinan para permitir el desarrollo rápido de aplicaciones tanto en el lado del servidor (en PHP) como al lado del cliente (en JavaScript). Zend Studio también hace que sea fácil de desarrollar en un entorno de producción como por la integración con VMware Workstation para ejecutar y depurar aplicaciones PHP.

NetBeans

NetBeans es un entorno de desarrollo visual que les permite a los programadores escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación como PHP, JavaScript, Groovy y Ruby. Es un producto de *software* libre, sin restricciones de uso.

La programación mediante NetBeans se realiza a través de componentes de *software* modulares, también llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las APIs de NetBeans y un archivo especial *manifest file* que lo identifica como módulo. Es fácil de usar, instalar y puede ser ejecutado en múltiples plataformas como Windows, Linux, Mac OS X y Solaris. (Parra, 2011)

Características de NetBeans:

- Instalación y actualización más simple.
- Características visuales para desarrollo web con interfaces amigables y completamiento automático de código PHP, así como coloreado de código sintáctico y semántico.
- Mejoras para SOA y UML.
- Permite depurar el código usando Xdebug.
- Soporte para PHP y Symfony.

1.4.5.1. Solución propuesta

Tabla 4: Comparación entre Eclipse, Zend Studio y NetBeans

Aspectos	Eclipse	Zend Studio	NetBeans
Soporte para Symfony			X
Software libre	X		X
Instalación y actualización más simple			X
Consume menos recursos			X
Depuración del código en aplicaciones PHP	X	X	X

Después de un análisis se concluye que para el desarrollo de la aplicación se elige como entorno de desarrollo NetBeans, por sus características que la convierten en la herramienta indicada. Brinda soporte para Symfony, *framework* con el que se pretende trabajar. Además es ligero y consume menos recursos, con pocos módulos activados puede funcionar correctamente. Sus actualizaciones y complementos son más fáciles de instalar.

1.4.6. Doctrine como Modelo de Objeto-Relacional (ORM)

Un ORM consiste en una serie de objetos que permiten acceder a los datos y que contienen en su interior cierta lógica de negocio.

En concreto es una técnica, o patrón arquitectónico, que permite comunicar dos sistemas distintos, típicamente una base de datos relacional con objetos de un lenguaje orientado a objetos, mediante un sistema que mapea o vincula ambos sistemas.

Una de las ventajas de utilizar estas capas de abstracción de objetos/relacional es que evita utilizar una sintaxis específica de un sistema de bases de datos concreto. Esta capa transforma automáticamente las llamadas a los objetos en consultas SQL optimizadas para el sistema gestor de bases de datos que se está utilizando en cada momento. De esta forma, es muy sencillo cambiar a otro sistema de bases de datos completamente diferente en mitad del desarrollo de un proyecto. (Zaninotto, y otros, 2008)

Existen ORM para la mayoría de lenguajes de programación. Por ejemplo, en Java es muy popular Hibernate, en .NET se utiliza ADO.NET Entity Framework, y en PHP el más usado es Doctrine, aunque este está inspirado en Hibernate.

Se propone utilizar Doctrine específicamente la versión 2.0 ya que es una librería para PHP que permite trabajar con un esquema de base de datos como si fuese un conjunto de objetos, y no de tablas y registros.

Entre sus principales características está que tiene su propio dialecto de SQL, llamado DQL (*Doctrine Query Language*) que es un lenguaje creado para ayudar al programador a extraer objetos de la base de datos. También es capaz de generar los modelos a partir de una base de datos con solo definir las dependencias entre ellos, o también se puede generar de forma inversa. Es muy sencillo de configurar, no es necesario mantener complejos ficheros de configuración XML. (Jonathan H. Wage, 2007)

1.4.7. Sistema Gestor de Base de Datos (SGBD)

Un SGBD es un grupo de herramientas que permite la gestión de los archivos de las bases de datos. Existen varios SGBD como son, MySQL, PostgreSQL, entre otros, que facilitan el trabajo con una base de datos. A continuación se analizan estos dos SGBD, haciendo énfasis en sus principales características, que ayuden a la selección del más indicado para el desarrollo de la aplicación.

PostgreSQL (Pecos, 2002)

Es un sistema de gestión de bases de datos objeto-relacional, y libre. Tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos, como son la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

Es hoy, el sistema gestor de bases de datos libre más avanzado que existe, contando con características tales como:

- Soporta bloques de código o funciones que se ejecutan en el lado del servidor, armándose de esta forma de todas las ventajas que ofrecen los lenguajes de programación soportados como: Java, C, C++, Python, Ruby, Perl, etc.
- Multiplataforma, está disponible para casi todos los principales SO: Linux, FreeBSD, Solaris, MS Windows y Mac OS X.

- Soporta concurrencia mediante el sistema MVCC (Acceso Concurrente Multi-Versión), el cual permite que mientras un proceso modifica una tabla, otro proceso pueda acceder a la misma tabla sin necesidad de bloqueos.
- PostgreSQL es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

Inconvenientes:

- Consume muchos recursos y carga más el sistema.
- Es muy lento.
- Menos funciones en PHP.

MySQL (Pecos, 2002)

Es un sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga, de forma muy eficiente. MySQL es patrocinado por una empresa sueca MySQL AB Presenta características que lo convierten en uno de los gestores de base de datos más estables.

- Mayor rendimiento y mayor velocidad al conectar con el servidor.
- Mejores utilidades de administración, backup y recuperación de errores.
- No suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- MySQL se comporta mejor que PostgreSQL a la hora de modificar o añadir campos a una tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.

Inconvenientes:

- No soporta transacciones, *roll-backs*, ni *subselects*.

- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

1.4.7.1. Solución propuesta

Tabla 5: Comparación entre PostgreSQL y MySQL

Aspectos	PostgreSQL	MySQL
Rendimiento	2 a 3 veces más lento	Mayor rendimiento
Consumo de recursos	Alto	Bajo
Facilidad de configuración y uso	Baja	Alta
Control de acceso	Bajo	Alto
Integración con PHP	Baja	Alta

Después de un análisis se concluye que a pesar de que PostgreSQL hoy en día es uno de los gestores de bases de datos más avanzados que existe, tiende a sobrecargar mucho el sistema y a desperdiciar memoria, además de que es muy lenta su velocidad de respuesta. Por tal motivo y dependiendo de las características que tendrá la aplicación a realizar, se elige como servidor de base de datos MySQL, puesto que presenta una mejor integración con PHP, que es el lenguaje con el cual se pretende desarrollar el sistema. Además presenta un mayor rendimiento y rapidez a la hora de hacerle una petición al servidor. Y todos estos mecanismos o servicios los brinda con muy poca utilización tanto de la unidad central de procesamiento como de memoria.

1.4.8. Servidor web

Un servidor web es un programa que implementa el protocolo HTTP y está diseñado para transferir hipertextos, páginas web y páginas HTML. Cuando se ejecuta una petición HTTP por parte de un navegador, el servidor responde enviando el código HTML de la página, el navegador lo interpreta y muestra el resultado en una página web. A continuación se analizan dos de los servidores web más utilizados, exponiendo sus principales características que ayuden a la selección del más indicado para el desarrollo de la aplicación.

Apache

Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, entre otras), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP. Entre sus características se encuentra que es altamente configurable y su arquitectura es modular. El servidor consta de diversos módulos que aportan muchas de las funcionalidades que podrían considerarse básicas para un servidor web. Tiene la capacidad de crear sitios privados seguros, es el servidor web más implementado entre los distintos servidores que ofrecen servicios a internet. Es muy usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. (McCool, 2011)

Nginx

Nginx es un servidor HTTP, *reverse proxy* y servidor *proxy* IMAP/POP3. Desarrollado bajo licencia BSD, va por la versión 1.0 para desarrolladores tiene buena estabilidad y alto rendimiento entre sus muchas características esta que tiene una configuración sencilla y poco consumo de recursos. Entre las características HTTP se encuentran: manejo de ficheros estáticos, índices y auto índices, *reverse proxying* (en español proxy inverso), acelerado sin caché y con balanceo de carga y tolerancia a fallos, arquitectura modular y soporte SSL². El acceso se basa en la dirección IP del cliente y en la autenticación básica HTTP. (Softpedia, 2012)

1.4.8.1. Solución propuesta

Tabla 6: Comparación entre Apache y Nginx

Aspectos	Apache	Nginx
Interfaz gráfica para su configuración		X
Alto consumo de memoria RAM	X	
Liviano		X
Alto rendimiento		X

² *Secure Socket Layer* (SSL en español: capa de conexión segura) es un protocolo de seguridad desarrollado por la empresa Netscape Communications para lograr que la transmisión de datos entre un servidor y un usuario, o viceversa, a través de Internet, sea completamente segura.

A pesar de que Apache es uno de los más usados en el mundo por su fácil uso, se puede tornar a veces muy pesado puesto que carga muchos módulos que le son indispensables para un buen funcionamiento. No cuenta con una interfaz gráfica que ayude a su configuración. Se escogió como servidor web Nginx puesto que es mucho más sólido y robusto, con una comunidad más activa y con mayores facilidades de configuración, es más rápido a la hora de proveer todo tipo de ficheros como imágenes y contenidos estáticos. No brinda tantos servicios como Apache, pero los que oferta lo hace de una manera rápida, consume menos recursos a medida que las peticiones aumentan por lo que se torna más liviano.

1.4.9. Herramienta para el modelado

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todo el ciclo de vida de desarrollo de un producto. (Murillo Alfaro, 1999)

La realización de un nuevo *software* requiere que las tareas sean organizadas y completadas en forma correcta y eficiente. Las herramientas CASE fueron desarrolladas para automatizar los procesos y facilitar las tareas de coordinación de los eventos, que necesitan ser mejorados en el ciclo de desarrollo de *software*. Permite la realización de un buen diseño para el proyecto, implementando a partir de él, parte del código automáticamente. Para el sistema que se va a desarrollar se utiliza Visual Paradigm como herramienta CASE.

Visual Paradigm para UML (VP-UML) está diseñado para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas, analistas de negocios, arquitectos, al igual que para aquellas personas interesadas en la construcción de sistemas de *software* de forma fiable a través de la utilización del enfoque Orientado a Objetos (OO).

Esta herramienta, además de soportar modelado UML, provee el modelado de procesos de negocio *Business Process Modeling Notation* (BPMN) y el mapeo relacional de objetos para Java, .Net y PHP. En su edición *Enterprise Edition* soporta la última versión de BPMN, la cual es una sencilla pero potente notación para el modelado de flujos de trabajo de un negocio. Visual Paradigm permite realizar ingeniería tanto directa como inversa, es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto.

Genera la documentación del proyecto automáticamente en varios formatos como Word o PDF, es además una herramienta multiplataforma. Tiene integración con la mayoría de los IDEs y permite trabajar con sistema de control de versiones. (Grupo Satelite s.a, 2011)

1.5. Conclusiones parciales

La fundamentación teórica de esta investigación trajo consigo que se definieran los conceptos y definiciones para posibilitar una mejor comprensión del proceso de replicación de datos. Además permitió:

- Analizar los sistemas existentes que abarcan el tema de administración de una réplica de datos, corroborando que el desarrollo de un nuevo sistema será la mejor solución para cumplir con las expectativas del cliente.
- Realizar un estudio teórico de los procesos de administración y supervisión del sistema de réplica “Magic@ Data Replication eXtensible Solution”, para comprender con mayor claridad cómo se realizan estos procesos y enfocar la investigación hacia los mismos.
- Desarrollar una investigación de las metodologías y herramientas a utilizar, donde se analizaron sus principales características para llegar a una correcta selección y tener una propuesta de solución bien justificada.

Capítulo II: Propuesta de Solución

2.1. Introducción

Este capítulo tiene como principal objetivo solucionar el problema que originó la investigación, por lo que se hace necesario realizar un análisis del comportamiento del negocio. Se detallan las necesidades de los usuarios, describiéndose las funcionalidades que serán objeto de automatización, además se generan los artefactos como resultado de la metodología ágil XP, basándose fundamentalmente en sus dos primeras fases: Exploración y Planeación. Se presentará una propuesta del *software* a implementar.

2.2. Descripción del sistema propuesto

Para lograr solucionar la problemática planteada, se propone realizar la implementación de una aplicación web a través de la cual se accederá a toda la información para la administración y supervisión de la réplica de datos “Magic@ Data Replication eXtensible Solution”, que permita gestionar la administración y supervisión del sistema de réplica. Compuesta por dos módulos que visualicen de una forma más clara y entendible la realización de estos procesos:

Módulo administración: este módulo se encarga de administrar todos los procesos para realizar la réplica, donde se gestionan los servidores de réplica, servidores de base de datos, las sincronizaciones, así como otros elementos para llevar a cabo el mismo. Se gestionan los usuarios y los rangos de IP que tienen acceso a la aplicación, asignándole a cada una de estos un grupo específico, del cual dependen los permisos que tenga a la hora de interactuar con la aplicación. También permite sincronizar los cambios en los servidores de réplica.

Módulo supervisión: este módulo se encarga de monitorear los servidores de base de datos que estén activos, así como también las sincronizaciones asociadas a estos, dependiendo del grupo al cual pertenezca el usuario y desde el IP del cual esté accediendo a la aplicación, específicamente para el grupo que tenga rol supervisor. Además permite para las sincronizaciones que se encuentren detenidas ver el log³ de error para cada una de estas, donde el usuario puede ver por qué se detuvo.

³ Es un registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema. (Martínez López, M. A. 2010)

A partir de ver con antelación el log de error de la sincronización, le facilita al usuario poder activarla. También posibilita que se supervisen para cada una de las sincronizaciones lo que está ocurriendo con ella en tiempo real en el servidor de réplica, a través de su log generado.

2.2.1. Modelo de Dominio

Debido al bajo conocimiento de los procesos del negocio se plantea un modelo de dominio para lograr una mejor comprensión de los conceptos del sistema. Su objetivo es lograr un mejor entendimiento entre usuarios, clientes y desarrolladores. Además de especificar y conocer las clases y objetos más significativos dentro del entorno del problema, lo que ayuda a determinar los procesos y roles más relevante para el sistema a desarrollar.

2.2.1.1. Conceptos principales del modelo de dominio

Usuario: Persona que interactúa con la aplicación web para la navegación.

- **Administrador:** Usuario con el completo acceso y privilegios en la aplicación que gestiona y controla todo lo referente al proceso de la réplica.
- **Usuario_Supervisor:** Persona que dependiendo del grupo en que se encuentre y del IP del cual esté accediendo a la aplicación podrá supervisar solo los servidores de base de datos y sincronizaciones que estén vinculados a su grupo.

Aplicación_Web: Es una aplicación informática que está destinada para la interacción con el usuario a través de las funcionalidades que brinda, la cual es accesible desde un navegador web⁴.

Esquemas: Son esquemas de sincronizaciones que permite definir el mapeo unidireccional de datos entre dos servidores y es posible definir las tablas de donde se extraen los datos y hacia qué tablas van dirigidos.

Tareas: Son las actividades programadas por el cliente para planificar en qué período de tiempo se desarrollan las sincronizaciones.

⁴ Es un programa informático que permite visualizar la información contenida en una página web, ya sea alojada en Internet o en un servidor local.

Ficheros: Son archivos indispensables para crear los esquemas de sincronización y configurar las bases de datos. Estos pueden ser ficheros de tipo (xml, config.xml, xrplschem y xtblstruct).

Sincronizaciones: Es el proceso que permite mantener la información actualizada entre servidores, ante eventuales cambios que se realicen en algún servidor.

Servidores de Réplica: Son los servidores encargados de realizar la réplica entre los servidores de base de datos.

Servidores de Base de Datos: Son servidores de base de datos a los cuales se le realizará la sincronización.

2.2.1.2. Diagrama de Clases del Modelo de Dominio

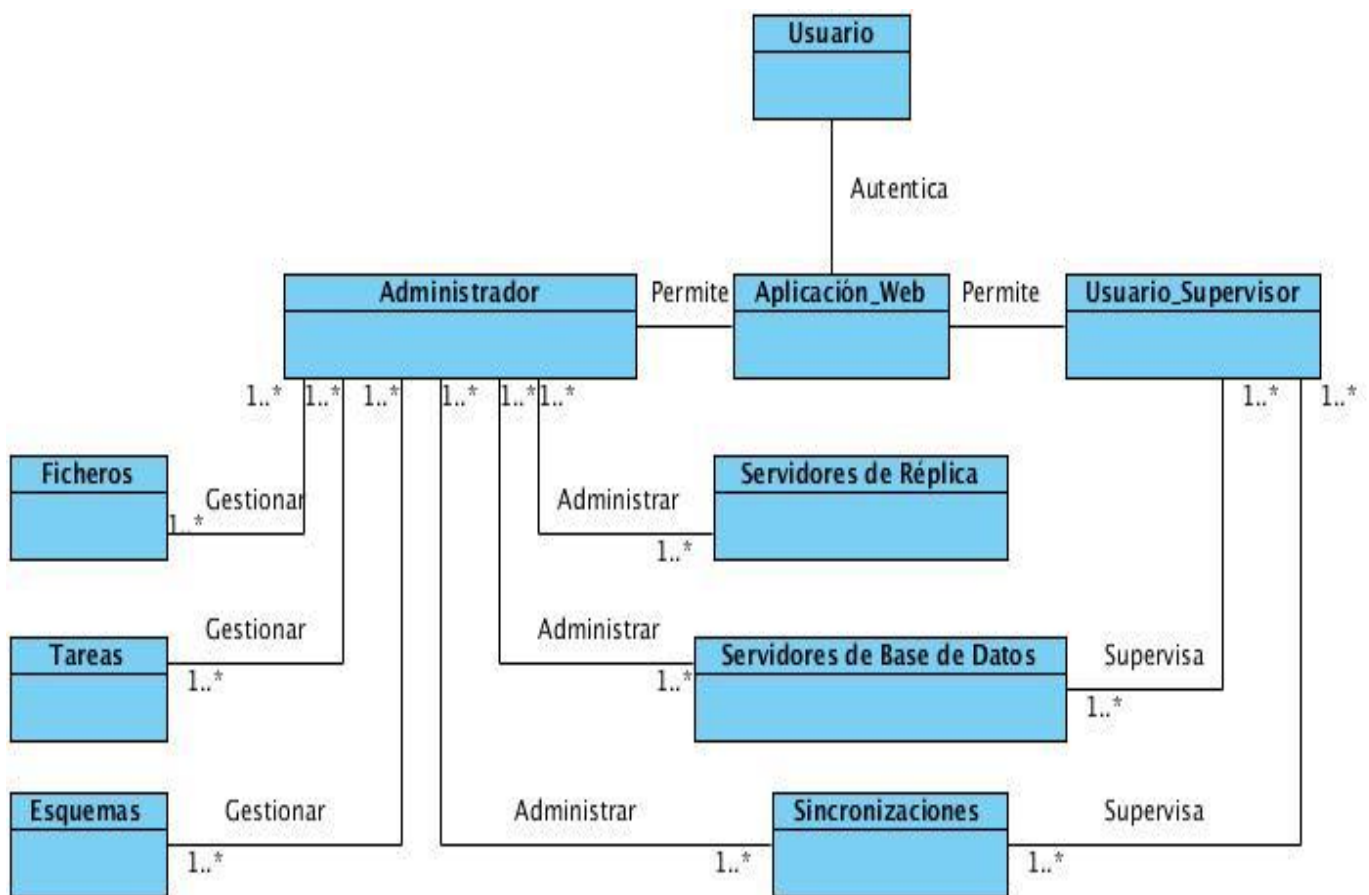


Figura 1: Modelo de Dominio

2.3. Requisitos funcionales del sistema

R.1-Gestionar rangos de IP

R1.1-Registrar nuevos rangos de IP (Nombre, Descripción, Dirección IP, Grupo).

R1.2-Listar rangos de IP.

R1.2.2-Modificar información de rangos de IP.

R1.2.3-Eliminar información de rangos de IP.

R1.2.4-Mostrar información de rangos de IP.

R1.3-Filtrar rangos de IP por: (Nombre, Descripción, Dirección IP).

R.2-Gestionar servidores de réplica

R2.1-Registrar nuevos servidores de réplica (Nombre, Descripción, Activo, Dirección IP, Usuario, Contraseña, Path, Sistema Operativo, Mac).

R 2.2-Listar servidores de réplica.

R2.2.1-Mostrar información del servidor de réplica.

R2.2.2-Editar información del servidor de réplica.

R2.2.3-Eliminar servidor de réplica.

R 2.3-Filtar servidor de réplica por: (Nombre, Descripción, Dirección IP, Activo o no).

R3-Gestionar servidores de base de datos

R 3.1-Registrar nuevos servidores de base de datos (Nombre, Descripción, Dirección IP, Sistema Operativo, Mac, Activo, Fichero de configuración, Tipo de Base de Datos, Grupo).

R3.2-Listar servidores de base de datos.

R3.2.1-Mostrar información del servidor de base de datos.

R3.2.2-Editar información del servidor de base de datos.

R3.2.3-Eliminar servidor de base de datos.

R3.3-Filtrar servidores de base de datos por (Nombre, Fichero, Dirección IP, Tipo de base de datos, Activos).

R4-Gestionar ficheros

R4.1-Registrar nuevo fichero (Fichero, Descripción, Tipo de fichero).

R4.2-Listar ficheros.

R4.2.1-Mostrar información del fichero.

R4.2.2-Editar información del fichero.

R4.2.3-Borrar fichero.

R4.3-Filtrar ficheros por (Nombre, Descripción, Tipo de fichero).

R5-Gestionar tipo de base de datos

R5.1-Registrar nuevo tipo de base de datos (Nombre, Descripción).

R5.2-Listar tipos de base de datos.

R5.2.1-Mostrar información del tipo de base de datos.

R5.2.2-Editar información del tipo de base de datos.

R 5.2.3-Borrar tipo de base de datos.

R5.3-Filtrar tipos de base de datos por (Nombre, Descripción).

R6-Gestionar esquema de sincronización

R6.1-Registrar nuevo esquema (Nombre, Descripción, Esquema de sincronización, Estructura a sincronizar, Destino de la estructura a sincronizar).

R6.2-Listar esquemas de sincronización.

R6.2.1-Mostrar información del esquema de sincronización.

R6.2.2-Editar información del esquema de sincronización.

R6.2.3-Borrar esquema de sincronización.

R 6.3-Filtrar esquemas de sincronización por (Nombre, Descripción, Esquema de sincronización, Estructura a sincronizar, Destino estructura a sincronizar).

R7- Gestionar sincronización

R7.1-Registrar nueva sincronización (Nombre, Descripción, Servidor de Réplica, Activo, Fichero de configuración, Esquema de subida, Esquema de descarga, Base de datos local, Base de datos remoto, Tareas programadas).

R7.2-Listar sincronizaciones.

R 7.2.1-Mostrar información de la sincronización.

R 7.2.2-Editar información de la sincronización.

R 7.2.3-Eliminar sincronización.

R7.3-Filtrar sincronización por (Nombre, Servidor de réplica, Activo, Fichero de configuración, Esquema de subida, Esquema de descarga, Base de datos local, Base de datos remota).

R8-Sincronizar cambios

R8.1-Listar los cambios realizados en los servidores de réplica.

R8.2-Copiar en el servidor web los cambios realizados en los servidores de réplica.

R8.3- Conectar el sistema a los servidores de réplica.

R8.4-Sincronizar los cambios con los servidores de réplica.

R9-Planificar las tareas programadas

R9.1-Registrar una nueva tarea (Nombre, Descripción, Minuto, Hora, Día, Mes, Día de la semana).

R9.2-Listar tareas programadas.

R9.2.1-Mostrar información de la tarea programada.

R9.2.2-Editar información de la tarea programada.

R9.2.3-Borrar la tarea programada.

R9.3-Filtrar tarea programada por (Nombre, Minuto, Hora, Día, Mes, Día de la semana).

R10-Listar servidores de base de datos

R10.1-Mostrar información de los servidores de base de datos activos, dependiendo del grupo en que se encuentre el usuario.

R10.2-Mostrar sincronizaciones asociadas al servidor de base de datos escogido por el usuario, dependiendo al grupo en que se encuentre el usuario antes mencionado.

R10.2.1-Mostrar información de las sincronizaciones.

R10.2.1.1-Mostrar el log de error de cada sincronización.

R10.2.1.2-Activar sincronizaciones detenidas.

R11-Listar sincronizaciones

R11.1.-Mostrar información de las sincronizaciones dependiendo al grupo en que se encuentre el usuario.

R11.1.1.-Mostrar log de error de cada sincronización.

R11.1.2.-Activar sincronizaciones detenidas.

R12-Gestionar usuarios

R12.1.-Registrar un nuevo usuario (Usuario, Correo electrónico, Contraseña, Grupo, Cerrado Expirado, Habilitado, Credencial caducada).

R12.2-Listar información de los usuarios.

R13.2.1-Mostrar información de los usuarios.

R13.2.2-Editar información de los usuarios.

R13.2.3-Borrar información de los usuarios.

R12.3 -Filtrar usuario por (Usuario, Habilitado, Correo).

RF13. Gestionar grupo de usuario

R13.1.-Registrar un nuevo grupo de usuario (Nombre, Roles).

R13.2-Listar información de los grupos de usuarios.

R13.2.1-Mostrar información de los grupos de usuarios.

R13.2.2-Editar información de los grupos de usuarios.

R13.2.3-Borrar información de los grupos de usuarios.

R13.3 -Filtrar grupo de usuario por (Nombre, Roles).

RF 14. Mostrar log de las sincronizaciones

R14.1.-Mostrar un listado de las sincronizaciones con su contenido.

R14.2-Mostrar el log en tiempo real para cada una de las sincronizaciones.

RF 15. Autenticar usuario

RF15.1- Permitir introducir los datos del usuario (Usuario, Contraseña).

2.4. Requisitos no funcionales del sistema

Apariencia

- La corrección de errores en la introducción de datos será clara y fácil de realizar.
- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- El diseño responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.

Usabilidad

- La aplicación deberá poseer una interfaz y navegación asequibles y funcionales tanto para usuarios expertos como para los que no tienen conocimientos profundos de informática.

Rendimiento

- El sistema requiere una rápida velocidad de procesamiento, debido a que su objetivo es la gestión y supervisión, por lo que sus funciones tienen que tener tiempos de respuestas cortos, lo que aumentará su operatividad.

Requisitos de software

- Sistemas operativos multiplataforma.
- Navegador web: Internet Explorer, Mozilla.
- Servidor web HTTP Nginx.
- Servidor gestor de base de datos MySQL.

Requisitos de hardware

- Los servidores Web y de Bases de Datos deben poseer un gigabyte de memoria RAM como mínimo, una Motherboard Asus P5LD2, además de un micro IP 3.0.

Estación de trabajo. (PC cliente)

- Periféricos: Mouse y Teclado.
- Tarjeta de Red.
- 256 MB de Memoria RAM. (Mínimo).
- Procesador Intel Celeron.

Restricciones en el diseño y la implementación

- Las herramientas de desarrollo de la aplicación serán el *framework* Symfony, NetBeans, Visual Paradigm y como mapeo de objeto relacional Doctrine.
- El lenguaje de programación será PHP 5.3.2 o superior.
- Se utilizará la Programación Orientada Objetos.

2.5. Fase de Exploración

Durante esta fase se realiza el proceso de identificación de las historias de usuario (HU), así como la familiarización con los equipos de trabajo, tecnologías y herramientas seleccionadas para la construcción del sistema. Dependiendo del conocimiento de los desarrolladores y las habilidades que estos tengan con la tecnología a utilizar, esta fase puede variar en el período de tiempo de su desarrollo, es decir, toma de pocas semanas a pocos meses.

2.5.1. Historias de Usuarios (HU)

En la fase de exploración de XP, las HU son las encargadas de representar los requisitos del sistema de una forma clara y concreta sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer, permitiendo responder al principio de los requerimientos cambiantes.

A continuación se describen algunas de las HU identificadas durante esta fase:

Tabla 7: HU Gestionar servidores de réplica

Historia de Usuario	
Número: HU2	Usuario: Desarrollador
Nombre historia: Gestionar servidores de réplica.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Martha María Rondón Regalado.	
Descripción: Permitirá añadir nuevos servidores de réplica, donde el usuario podrá insertar los datos, además se mostrará una lista de los servidores existentes, permitiendo para cada uno de estos la opción de modificar, eliminar y mostrar su contenido. También se podrán filtrar los servidores de réplica por el nombre, descripción, dirección IP o si están activos o no.	
Observaciones:	

Tabla 8: HU Sincronizar cambios

Historia de Usuario	
Número: HU8	Usuario: Desarrollador
Nombre historia: Sincronizar cambios.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Madiela Cabeza Beltran.	
Descripción: Permitirá listar los cambios en los servidores de réplica y sincronizar los mismos, para esto se realiza una copia local en el servidor web de las modificaciones realizadas. Luego el sistema se conecta con los servidores de réplica y realiza la sincronización con los cambios existentes.	
Observaciones:	

Tabla 9: HU Gestionar usuarios

Historia de Usuario	
Número: HU12	Usuario: Desarrollador
Nombre historia: Gestionar usuarios.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Madiela Cabeza Beltran.	
Descripción: Permitirá añadir nuevos usuarios, también se mostrará una lista con los todo los usuarios existentes, permitiendo además para cada uno de estos la opción de modificar, eliminar, y mostrar su contenido. Además se podrán filtrar los usuarios por el nombre, grupo o rol.	
Observaciones:	

Para darle seguimiento a las historias de usuarios puede ver el Anexo # 2.

2.6. Fase de Planeación

En esta fase se establece la prioridad de cada historia de usuario y se realiza la estimación del esfuerzo necesario para la realización de cada una de ellas, además se identifican las iteraciones con las cuales se estima implementar el sistema y un plan de entrega para fijar el período de tiempo que puede tardar el desarrollo del mismo.

2.6.1. Estimación del esfuerzo por Historia de Usuario

En el sistema propuesto se realiza un análisis de la estimación del esfuerzo por cada una de las HU, los programadores estiman el tiempo que necesitan para desarrollar cada una de estas. Este valor se expresa en semanas y se irá aproximando a la realidad con el transcurso de las iteraciones.

Una HU no debe desarrollarse en menos de una, ni en más de dos semanas; en otro caso, será necesario acoplar o dividir las HU. Llegándose a los resultados que se muestran en la siguiente tabla:

Tabla 10: Estimación del esfuerzo por HU

Historia de Usuario	Estimación(semanas)
Gestionar Rangos de IP	1
Gestionar servidores de réplica	1
Gestionar servidores de base de datos	1
Gestionar ficheros	1
Gestionar tipo de base de datos	1
Gestionar esquema de sincronización	1
Gestionar sincronización	1
Sincronizar cambios	1
Planificar las tareas programadas	1
Listar servidores de base de datos	1
Listar sincronizaciones	1
Gestionar usuarios	1
Gestionar grupo de usuario	1
Mostrar log de las sincronizaciones	1
Autenticar usuario	1
Total	15

2.6.2. Plan de iteraciones

Después de haber sido estimado el esfuerzo dedicado a la realización de cada una de las HU identificadas, se procede a la creación del plan de iteraciones. Este plan tiene como objetivo mostrar la duración y el orden en que serán implementadas las HU dentro de cada iteración. Las funcionalidades son planificadas en esta fase, generando al final de cada una un entregable funcional que implementa las HU asignadas a la iteración. En la presente solución se han identificado 15 historias de usuario y se definieron tres iteraciones; con una duración de seis semanas en la primera iteración, cuatro en la segunda y cinco en la tercera iteración.

Iteración 1

En la presente iteración se establecerá la arquitectura base del sistema y se implementarán las HU con mayor prioridad y que sean de suma importancia en la primera etapa de implementación. Al finalizar la misma se contará con una primera iteración de prueba correspondiente a esta iteración para ser mostrada al cliente y así obtener una evaluación del mismo.

Iteración 2

Esta iteración tiene como objetivo la implementación de las HU que tienen mediana prioridad dentro del sistema a desarrollar. Y como resultado de esta iteración se contará con la segunda versión del producto final. Esta versión junto a las implementaciones anteriores, serán mostradas al cliente con el objetivo de realizar cambios en base a la aceptación del mismo.

Iteración 3

El objetivo de esta iteración es implementar las HU de baja complejidad. Como resultado de esta iteración se tendrá la versión 1.0 del producto final, el cual será puesto en funcionamiento para evaluar su rendimiento en función de satisfacer las expectativas del cliente.

2.6.3. Plan de duración de las iteraciones

Este plan se encarga de mostrar el tiempo de demora de cada una de las iteraciones y las HU que serán abordadas en cada una de ellas con su respectivo orden de implementación, lo que ayuda a obtener una idea aproximada del tiempo que durará la confección del sistema en su totalidad.

Tabla 11: Plan de duración de las iteraciones

Iteración	No. HU	Historia de Usuario	Duración Estimada
Iteración 1		Gestionar servidores de réplica	6 semanas
		Gestionar servidores de base de datos	
		Gestionar ficheros	
		Gestionar tipo de base de datos	
		Gestionar esquema de sincronización	
		Planificar las tareas programadas	

Iteración 2	Gestionar sincronización	4 semanas
	Sincronizar cambios	
	Listar servidores de base de datos	
	Listar sincronizaciones	
Iteración 3	Gestionar usuarios	5 semanas
	Gestionar grupo de usuarios	
	Gestionar Rangos de IP	
	Mostrar log de las sincronizaciones	
	Autenticar usuario	

2.6.4. Plan de Entrega

Luego de haberse estimado la duración de cada una de las iteraciones, se crea un plan de entregas ideado específicamente para la fase de planificación donde se tiene como principal objetivo presentar una propuesta al finalizar cada iteración, de la fecha aproximada en que se harán versiones al sistema a implementar.

Tabla 12: Plan de entrega

Producto	Final 1ra Iteración (23 de marzo)	Final 2da Iteración (20 de abril)	Final 3ra Iteración (25 de mayo)
Sistema para la administración y supervisión de la réplica “Magic@ Data Replication eXtensible Solution”.	0.1	0.2	1.0

2.7. Descripción de la Arquitectura

El diseño arquitectónico de una aplicación requiere un firme conocimiento de las tecnologías aplicadas y lenguajes de programación, dado que estas están en constante desarrollo y avanzan día a día. Un mal diseño de la arquitectura puede causar problemas usuales como bajo rendimiento, insuficiente mantenimiento y baja disponibilidad.

Modelo Vista Controlador (MVC) (Eguiluz, 2011)

El Modelo Vista Controlador es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista son las páginas HTML y el código que provee de datos dinámicos a la página; el modelo es la representación de los datos con que trabaja la aplicación y la lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Modelo: es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo.

Vista: es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio modelo.

Controlador: es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

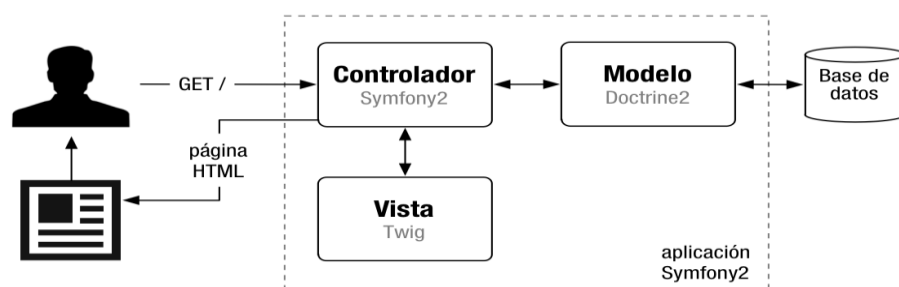


Figura 2: Patrón MVC en Symfony

Cuando el usuario solicita ver la portada del sitio, internamente sucede lo siguiente:

1. El sistema de enrutamiento determina qué **Controlador** está asociado con la página de la portada.
2. Symfony2 ejecuta el **Controlador** asociado a la portada. Un controlador no es más que una clase PHP en la que puedes ejecutar cualquier código que quieras.

3. El **Controlador** solicita al **Modelo** los datos de la oferta del día. El modelo no es más que una clase PHP especializada en obtener información, normalmente de una base de datos.

4. Con los datos devueltos por el **Modelo**, el **Controlador** solicita a la **Vista** que cree una página mediante una plantilla y que inserte los datos del **Modelo**.

5. El **Controlador** entrega al servidor la página creada por la **Vista**.

A pesar de que se pueda llegar a hacer cosas muy complejas con Symfony2, el funcionamiento interno siempre es el mismo:

1. El **Controlador** manda y ordena.
2. El **Modelo** busca la información que se le pide.
3. La **Vista** crea páginas con plantillas y datos.

Se utilizará un patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC), ya que el marco de trabajo Symfony que será utilizado en el desarrollo del sistema está basado arquitectónicamente en este patrón.

2.8. Conclusiones parciales

Con la realización de este capítulo fueron desarrolladas las primeras dos fases de la metodología XP permitiendo:

- Generar los artefactos correspondientes, así como las historias de usuario que caracterizan el sistema y el análisis de la estimación del tiempo de desarrollo para cada una de estas, logando realizar una buena comprensión del negocio y una interpretación correcta de las necesidades del cliente.
- Identificar tres iteraciones para la implementación del sistema, donde a partir de ello, se creó el plan de duración para las mismas y el plan de entrega del producto final.

Capítulo III: Implementación y Prueba

1.1. Introducción

Después de culminar la fase de Planificación o Planeación, XP propone que la implementación debe desarrollarse de forma iterativa, obteniéndose al terminar cada iteración un producto que debe ser mostrado al cliente. En este capítulo se detalla específicamente las tres iteraciones llevadas a cabo durante la etapa de construcción de la aplicación web, así como también las pruebas de aceptación efectuadas sobre el sistema.

1.2. Diseño de la solución

La metodología XP propone que el diseño del sistema se realice de forma sencilla, hacerlo lo menos complicado posible y conseguir un diseño fácil de entender, además no requiere la representación del sistema mediante diagramas de clases utilizando notación UML. Usa tarjetas CRC (Contenido, Responsabilidad y Colaboración) para reemplazar los diagramas de modelos. Estas tarjetas permiten trabajar con una metodología basada en objetos. Proporciona que el equipo completo contribuya en la tarea del diseño. La tarjeta CRC se caracteriza por representar un objeto. El nombre de la clase se coloca en forma de título en la tarjeta, en la parte izquierda las funcionalidades y las clases que se implican en cada funcionalidad se ponen en la derecha.

1.2.1. Tarjetas CRC

Tabla 13: Tarjeta CRC ClienteController

ClienteController	
Funcionalidades	Colaboraciones
indexAction homeAction replica_magicaAction showAction playAction mostrarAction showTodasAction playTodasAction consolelogAction	DservidorDbRepository DsincronizacionRepository EntityManager

Para darle seguimiento a las Tarjetas CRC puede ver el Anexo # 3.

1.2.2. Diseño de la base de datos

El modelo entidad relación es el más utilizado para el diseño conceptual de una bases de datos, este proporciona mucha ayuda a la hora de visualizar las diferentes entidades que intervienen, así como también las relaciones entre ellas. Estos modelos expresan entidades relevantes para un sistema de información.

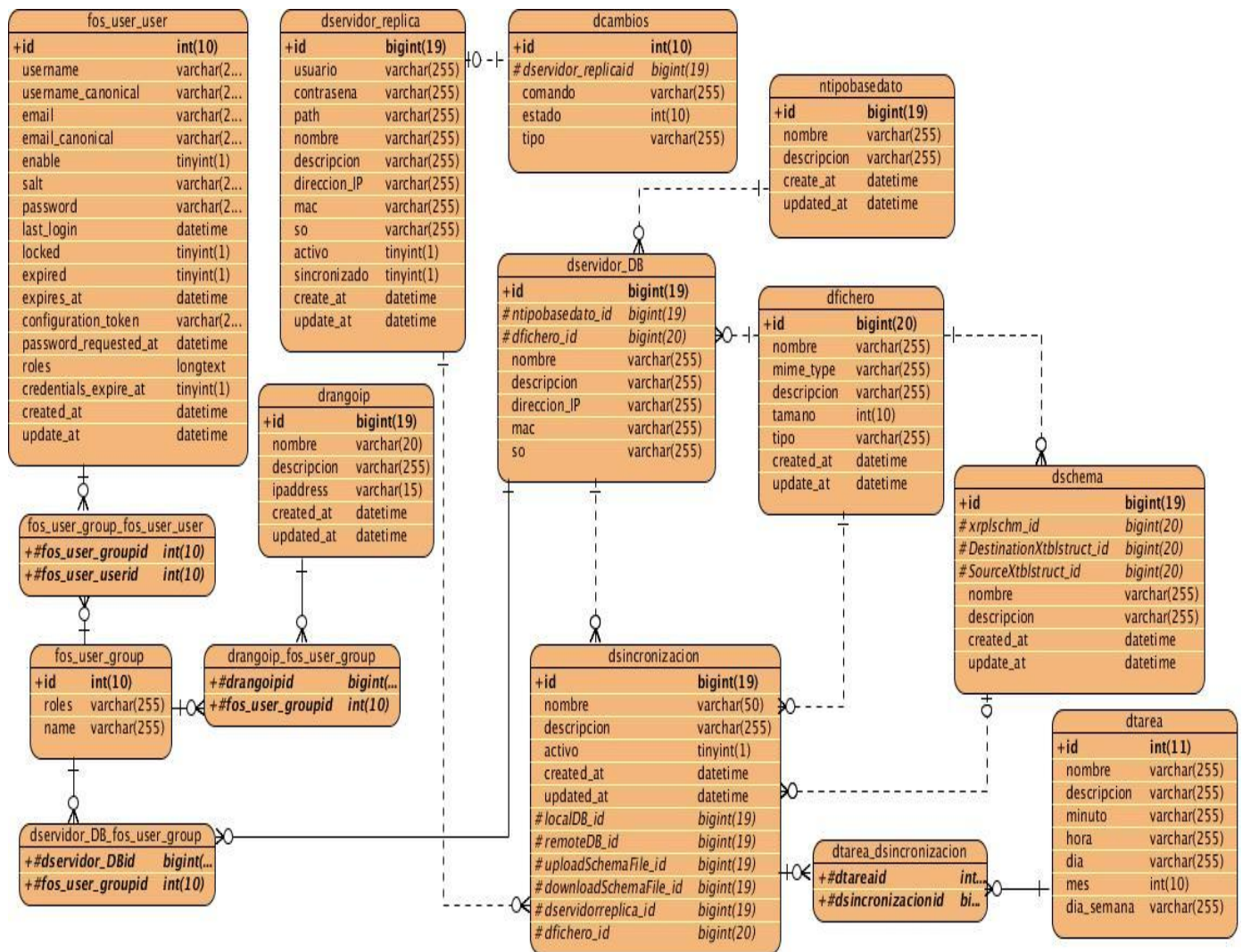


Figura 3 : Diagrama Entidad Relación

Descripción de las tablas de la base de datos:

fos_user_user: Guarda todos los usuarios del sistema.

fos_user_group: Guarda todos los grupos de usuarios del sistema.

fos_user_group_fos_user_user: Guarda los indicadores de las tablas fos_user_user y fos_user_group producto de la relación de muchos a muchos que existe entre ellas.

drangoip: Guarda los datos de los rangos de IP para los grupos de usuarios del sistema.

drangoip_fos_user_group: Guarda los indicadores de las tablas fos_user_group y drangoip producto de la relación de muchos a muchos que existe entre ellas.

dservidor_DB: Guarda los datos de los servidores de base de datos del sistema.

dservidor_DB_fos_user_group: Guarda los indicadores de las tablas fos_user_group y dservidor_DB producto de la relación de muchos a muchos que existe entre ellas

dservidor_replica: Guarda todos los datos de los servidores de réplica del sistema.

dcambios: Guarda todos los cambios o modificaciones que ocurran en los servidores de réplica.

dfichero: Guarda los datos de los ficheros necesarios para las sincronizaciones y las bases de datos del sistema.

ntipobasedato: Guarda los datos de los tipos de base de datos que existen en el sistema.

dsincronizacion: Guarda los datos de las sincronizaciones hechas a los servidores de réplica.

dtarea: Guarda los datos relacionados con las tareas programadas de cada sincronización.

dtarea_dsincronizacion: Guarda los indicadores de las tablas dtarea y dsincronizacion producto de la relación de muchos a muchos que existe entre ellas.

dschema: Guarda los datos referentes a los esquemas de sincronización.

1.3. Implementación del sistema

La fase de desarrollo del sistema es uno de los principales pilares para obtener un producto final deseado con una óptima calidad. En esta fase, XP propone la implementación de cada una de las HU en el transcurso de la iteración a la cual pertenecen. Para ayudar y organizar la implementación de las HU se crean tareas de desarrollo, garantizando de esta forma asignar responsables de la implementación a un grupo de desarrolladores.

Para la implementación de un software, XP propone que al terminar cada iteración se obtenga del producto final una versión que debe ser mostrado y probado al cliente. A continuación se muestran detalladamente las tres iteraciones generadas por la planificación descrita en el capítulo anterior.

1.3.1. Iteración 1

Como principal objetivo en esta iteración se desarrollarán aquellas HU que tienen mayor importancia para el sitio web, convirtiéndose en las de máximo interés para el cliente y que cubran la mayoría de las funcionalidades básicas del sistema.

1.3.1.1. Tares de la ingeniería iteración 1

Tabla 14: Tareas de la ingeniería en la primera iteración

Iteración	No	Historia de Usuario	Tareas
Iteración 1	HU2	Gestionar servidores de réplica.	– Agregar, modificar, eliminar, mostrar y listar los servidores de réplica.
	HU3	Gestionar servidores de base de datos.	– Agregar, modificar, eliminar, mostrar y listar las bases de datos.
	HU4	Gestionar ficheros.	– Agregar, modificar, eliminar, mostrar y listar los ficheros de configuración.
	HU5	Gestionar tipo de base de datos.	– Agregar, modificar, eliminar, mostrar y listar los tipos de base de datos.
	HU6	Gestionar esquema de sincronización.	– Agregar, modificar, eliminar, mostrar y listar los esquemas de sincronización.
	HU9	Planificar las tareas programadas.	– Agregar, modificar, eliminar, mostrar y listar las tareas programadas.

1.3.1.2. Tareas de la ingeniería detalladas Iteración 1

Tabla 15: HU2_T1

Tarea	
Número de la Tarea: HU2_T1	HU: Gestionar servidores de réplica.
Nombre: Agregar, modificar, eliminar, mostrar y listar los servidores de réplica.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10 de febrero	Fecha Fin: 17 de febrero
Responsable: Martha Rondón Regalado.	
Descripción: Deberá agregar, eliminar, mostrar y listar todos los datos asociados a los servidores de réplica. El sistema debe validar que no se adicionen servidores con igual nombre, que no se dejen campos vacíos y en caso contrario mostrará un mensaje de error informando al usuario. Se deben guardar los datos insertados.	

Para darle seguimiento a las tareas realizadas y ayudar a la implementación de los módulos de la primera iteración puede ver el Anexo # 4.

1.3.2. Iteración 2

En esta iteración se encuentran aquellas historias de usuario de prioridad mediana, que no cuentan en su mayoría con un nivel de riesgo muy elevado para el programador. En esta iteración se culmina la implementación de una parte del módulo de administración, pues fueron divididas sus funcionalidades en las tres primeras iteraciones para distribuir mejor el trabajo.

1.3.2.1. Tares de la ingeniería iteración 2

Tabla 16: Tareas de la ingeniería en la segunda iteración

Iteración	No	Historia de Usuario	Tareas
Iteración 2	HU7	Gestionar sincronización.	– Agregar, modificar, eliminar, mostrar y listar las sincronizaciones.
	HU8	Sincronizar cambios.	– Listar y sincronizar los cambios realizados en los servidores de réplica.

	HU10	Listar servidores de base de datos.	– Listar los servidores de base de datos y para cada uno de estos, mostrar las sincronizaciones asociadas al mismo.
	HU11	Listar sincronizaciones.	– Listar las sincronizaciones y mostrar el log de error por cada una de las que se encuentren detenidas, y permitir activarlas.

1.3.2.2. Tareas de la ingeniería detalladas Iteración 2

Tabla 17: HU8_T1

Tarea	
Número de la Tarea: HU8_T1	HU: Sincronizar cambios.
Nombre: Listar los cambios realizados en los servidores de réplica.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 30 de marzo	Fecha Fin: 6 de abril
Responsable: Martha Rondón Regalado.	
Descripción: Deberá listar todos los cambios realizados en los servidores de réplica. El sistema debe realizar una copia local con todos estos cambios, como por ejemplo modificar el fichero de configuración de una sincronización o los ficheros de la base de datos para luego realizar el proceso de sincronización con los servidores de réplica.	

Para darle seguimiento a las tareas realizadas y ayudar a la implementación de la segunda iteración puede ver el Anexo # 5.

1.3.3. Iteración 3

En esta iteración se desarrollan las historias de usuario que tienen una baja prioridad. En su mayoría presenta funcionalidades que son de un ambiente más amigable y agradable para el cliente. En ella se recogen los procesos referidos a los usuarios y grupos de usuarios, así como sus roles.

1.3.3.1. Tares de la ingeniería iteración 3

Tabla 18: Tareas de la ingeniería en la tercera iteración

Iteración	No	Historia de Usuario	Tareas
Iteración 3	HU12	Gestionar usuarios.	– Agregar, modificar, eliminar, mostrar y listar los usuarios.
	HU13	Gestionar grupo de usuarios.	– Agregar, modificar, eliminar, mostrar y listar los grupos de usuarios.
	HU1	Gestionar Rangos de IP.	– Agregar, modificar, eliminar, mostrar y listar los rangos de IP.
	H14	Mostrar log de las sincronizaciones.	– Listar las todas las sincronizaciones existentes y para cada una de ellas mostrar el log en tiempo real que se encuentra en el servidor de réplica.
	H15	Autenticar usuario.	– Controlar el acceso a la aplicación.

Tabla 19: HU12_T1

Tarea	
Número de la Tarea: HU12_T1	HU: Gestionar usuarios.
Nombre: Agregar, modificar, eliminar, mostrar y listar los usuarios.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20 de abril	Fecha Fin: 27 de abril
Responsable: Martha Rondón Regalado.	
Descripción: Se podrán agregar, eliminar, mostrar y listar todos los usuarios. Se deberá validar que no se dejen campos vacíos, que el correo sea válido y en caso contrario mostrará un mensaje de error informando al usuario. Los datos insertados correctamente deben ser guardados en la base de datos. Además posibilitará asignarle a un usuario un grupo determinado al cual va a pertenecer y con el cual se va a identificar.	

Para darle seguimiento a las tareas realizadas para ayudar a la implementación de la tercera iteración puede ver el Anexo # 6.

1.3.4. Diagrama de despliegue

Para evaluar de forma visual cómo se encuentran los componentes de la aplicación relacionados físicamente se elabora un diagrama de despliegue, este representa las relaciones físicas entre componentes de *software* y *hardware* en el sistema. En este caso la aplicación se encuentra hospedada en un servidor Web NGINX y se comunica con un gestor de base de datos MySQL y con uno o varios servidores de réplica, por el protocolo seguro ssh⁵.

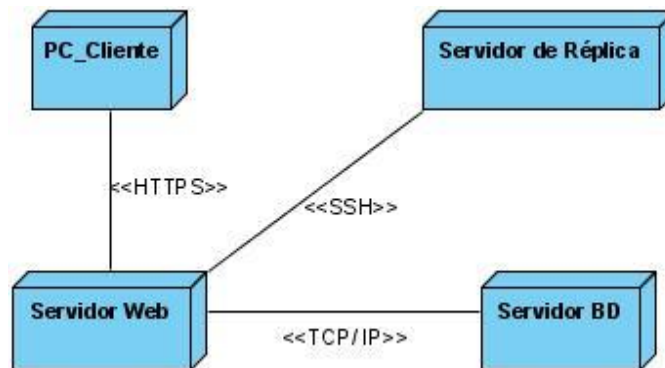


Figura 4: Diagrama de Despliegue

1.4. Prueba

Lo más importante para revelar la calidad de un *software* es desarrollar un proceso de pruebas, esto contribuye con los programadores y con los clientes a estar constantemente probando el producto como sea posible. Así se pueden detectar y corregir fallos y errores encontrados a tiempo. Las pruebas de *software* permiten además identificar detalles en aquellas historias de usuarios que no fueron obvias para el cliente. Una prueba tiene éxito si descubre un error no detectado hasta entonces. La metodología XP se encarga de dividir las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación.

⁵ Es tanto una aplicación como un protocolo, que permite conectar dos ordenadores a través de una red, ejecutar comandos de manera remota y mover ficheros entre los mismos. Proporciona autenticación fuerte y comunicaciones seguras sobre canales no seguros. (García Alfaro, Joaquín; Perramon Tornil, Xavier, 2009).

1.4.1. Pruebas unitarias

Las pruebas unitarias son las encargadas de probar el correcto funcionamiento del código, desarrollado por los programadores. Estas pruebas son especificadas antes de realizarse la implementación del código y en ellas se describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo mejorar la calidad de las funcionalidades de un sistema. Es de gran importancia que todos los módulos de la aplicación deben de pasar por las pruebas unitarias antes de ser divulgados, de no ser así se concluye que el código no está completo.

A partir de la versión 2, Symfony utilizará PHPUnit para realizar todas las pruebas unitarias y funcionales. De este modo deja de lado a Lime, la librería para pruebas unitarias y funcionales creada para la versión 1.0. (Eguiluz, 2010)

Puesto que Symfony y Doctrine comparten la misma plataforma de pruebas, es muy fácil implementar las pruebas unitarias en un proyecto Symfony. El modelo de objeto-relacional viene con su propio conjunto de herramientas para facilitar las pruebas unitarias y simular todo lo que necesites, tal como una conexión y un gestor de entidades. Al usar los componentes de prueba proporcionados por Doctrine, junto con algunas configuraciones básicas, se puede aprovechar las herramientas de Doctrine para las pruebas unitarias de los repositorios. Si se desea probar la ejecución real de las consultas, se necesita una prueba funcional. Las pruebas unitarias sólo son posibles cuando pruebas un método que construye una consulta. (Pacheco, 2011)

Se realizaron varias pruebas unitarias, en el Anexo # 10 se puede encontrar un ejemplo de ello, donde se obtuvo tres pruebas satisfactorias.

1.4.2. Pruebas de aceptación

Estas pruebas tienen como propósito validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. En las iteraciones, cada historia de usuario se convierte en un caso de prueba, aunque puede tener todas las pruebas de aceptación que desee para asegurar su mejor funcionamiento.

La realización de este tipo de pruebas y la publicación de los resultados deben ser lo más rápido posible, para que los desarrolladores puedan realizar con la mayor rapidez posible, los cambios que sean necesarios. A las pruebas de aceptación también se las conoce con el nombre de pruebas de funcionalidad, y constituyen la garantía de que los requerimientos fijados por los usuarios han sido reflejados en el sistema. (Mendía Arriola, 2006)

A continuación se muestran las pruebas de aceptación por cada iteración:

1.4.2.1. Iteración 1

Tabla 20: Prueba de aceptación Gestionar servidores de réplica: Agregar Nuevo

Caso de Prueba de Aceptación	
Código: HU2_P1	HU: Gestionar servidores de réplica.
Nombre: Gestionar servidores de réplica.	
Descripción: Prueba de funcionalidad para gestionar los servidores de réplica, con el fin de agregar nuevos servidores de réplica.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un nuevo servidor de réplica. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). <ol style="list-style-type: none"> 3.1. El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio). Actualizar y Cerrar (Se actualizan los cambios, se listan todos los servidores de réplica y muestra mensaje satisfactorio). Borrar (Ver Prueba 3 paso1). Mostrar (Ver paso 5). Agregar nuevo (Vuelve al paso 2). Volver a la lista (Ver Prueba 2 paso 2). 4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1). 5. El administrador observa los datos del servidor de réplica insertado. 	

6. El administrador escoge la opción que desea: Editar (Ver paso 3.1). Agregar nuevo (Vuelve al paso 2).
Resultado esperado: Se guardan los datos del nuevo servidor de réplica y si se editan sus datos se guardan los cambios realizados, además se muestran si el administrador desea.
No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un servidor de réplica.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 21: Prueba de aceptación Gestionar servidores de réplica: Listar

Caso de Prueba de Aceptación	
Código: HU2_P2	HU: Gestionar servidores de réplica.
Nombre: Gestionar servidores de réplica.	
Descripción: Prueba de funcionalidad para gestionar los servidores de réplica, para listar todos los servidores de réplica.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Listar. 2. Selecciona para cada servidor de réplica que se encuentra en la lista, la opción que desea: <ul style="list-style-type: none"> Editar (Ver Prueba 1 paso 3.1). Borrar (Ver Prueba 3 paso1). Agregar nuevo (Prueba 1 paso 2). Borrar todos los elementos. (Se muestra un mensaje de confirmación). 2.1. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Si, borrar (Se muestra un mensaje satisfactorio). Volver a la lista (Volver paso 2). Agregar nuevo (Volver prueba 1 paso 2). 3. El administrador escoge la opción: Filtros. 3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción: <ul style="list-style-type: none"> Filtrar (Filtra el listado de servidores de réplica por el campo defino). Resetear (Muestra el listado original). 	

Resultado esperado: Se listan todos los servidores de réplica con todos sus datos.
No conformidades:
Evaluación de la prueba: Prueba satisfactoria.

Tabla 22: Prueba de aceptación Gestionar servidores de réplica: Borrar

Caso de Prueba de Aceptación	
Código: HU2_P3	HU: Gestionar servidores de réplica.
Nombre: Gestionar servidores de réplica.	
Descripción: Prueba de funcionalidad para gestionar los servidores de réplica, para borrar los servidores de réplica del sistema.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el servidor de réplica y se muestra un mensaje satisfactorio). 2. Puede escoger la opción que desea: <ul style="list-style-type: none"> Editar (Prueba 1 paso 3.1). Agregar nuevo (Prueba 1 paso 2). 	
Resultado esperado: Se borran los servidores de réplica con todos sus datos y el sistema muestra un mensaje de confirmación satisfactoria.	
No conformidades:	
Evaluación de la prueba: Prueba satisfactoria.	

Para darle seguimiento al resto de las pruebas de la primera iteración puede ver el Anexo # 7.

1.4.2.2. Iteración 2

Tabla 23: Prueba de aceptación Sincronizar cambios

Caso de Prueba de Aceptación	
Código: HU8_P1	HU: Sincronizar cambios.
Nombre: Sincronizar cambios.	
Descripción: Prueba de funcionalidad para sincronizar los cambios en los servidores de réplica.	

Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El administrador selecciona la opción Sincronizar cambios. 2. El administrador observa una lista de servidores de réplica a los cuales se les va a realizar la sincronización. 3. El administrador selecciona la opción Sincronizar.
Resultado esperado: Se realiza la sincronización con los servidores de réplica.
No conformidades:
Evaluación de la prueba: Prueba satisfactoria.

Para darle seguimiento al resto de las pruebas de la segunda iteración puede ver el Anexo # 8.

1.4.2.3. Iteración 3

Tabla 24: Prueba de aceptación: Gestionar usuarios: Agregar Nuevo

Caso de Prueba de Aceptación	
Código: HU13_P1	HU: Gestionar usuarios.
Nombre: Gestionar usuarios.	
Descripción: Prueba de funcionalidad para gestionar usuarios, con el fin de agregar nuevos usuarios al sistema.	
Condiciones de ejecución: El administrados debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un nuevo usuario. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). <ol style="list-style-type: none"> 3.1 El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio). Actualizar y Cerrar (Se actualizan los cambios, se listan todos los usuarios y muestra mensaje satisfactorio). Borrar (Ver Prueba 3 paso1). 	

<p>Mostrar (Ver paso 5).</p> <p>Agregar nuevo (Vuelve al paso 2).</p> <p>Volver a la lista (Ver Prueba 2 paso 2).</p> <p>4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1).</p> <p>5. El administrador observa los datos del usuario.</p> <p>6. El administrador escoge la opción que desea:</p> <p>Editar (Ver paso 3.1).</p> <p>Agregar nuevo (Vuelve al paso 2).</p>
<p>Resultado esperado: Se guardan los datos del nuevo usuario y si se editan los datos, se guardan los cambios realizados, además se muestran si el administrador lo desea.</p>
<p>No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un usuario.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 25: Prueba de aceptación Gestionar usuarios: Listar

Caso de Prueba de Aceptación	
Código: HU13_P2	HU: Gestionar usuarios.
Nombre: Gestionar usuarios.	
Descripción: Prueba de funcionalidad para gestionar los usuarios, con el fin de listar todos los usuarios.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<p>1. El administrador selecciona la opción que desea: Listar.</p> <p>2. Selecciona para cada usuario que se encuentra en la lista, la opción que desea:</p> <p>Editar (Ver Prueba 1 paso 3.1).</p> <p>Borrar (Ver Prueba 3 paso1).</p> <p>Agregar nuevo (Prueba 1 paso 2).</p> <p>Borrar todos los elementos (Se muestra un mensaje de confirmación).</p> <p>2.1. El administrador escoge la opción que desea:</p> <p>Si, borrar (Se muestra un mensaje satisfactorio).</p>	

<p>Volver a la lista (Volver paso 2).</p> <p>Agregar nuevo (Volver prueba 1 paso 2).</p> <p>3. El administrador escoge la opción: Filtros.</p> <p>3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción:</p> <p>Filtrar (Filtra el listado de usuarios por el campo defino).</p> <p>Resetear (Muestra el listado original).</p>
<p>Resultado esperado: Se listan todos los usuarios con todos sus datos.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 26: Prueba de aceptación Gestionar usuarios: Borrar

Caso de Prueba de Aceptación	
Código: HU13_P3	HU: Gestionar usuarios.
Nombre: Gestionar usuarios.	
Descripción: Prueba de funcionalidad para gestionar los usuarios, con el fin de borrarlos.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el usuario y se muestra un mensaje satisfactorio). Puede escoger la opción que desea: <ul style="list-style-type: none"> Editar (Prueba 1 paso 3.1). Agregar nuevo (Prueba 1 paso 2). 	
Resultado esperado: Se borran los usuarios con todos sus datos y el sistema muestra un mensaje de confirmación satisfactoria.	
No conformidades:	
Evaluación de la prueba: Prueba satisfactoria.	

Para darle seguimiento al resto de las pruebas de aceptación correspondientes a la tercera iteración puede ver el Anexo # 9.

1.5. Conclusiones parciales

Culminado el ciclo de desarrollo del sistema para la administración y supervisión de la “Réplica Mágic@” se concluye que:

- Se realizó el diseño de la aplicación procurando que fuera lo más simple y lo menos complicado posible, utilizando para esto Tarjetas CRC, esto ayudó a proporcionar que el equipo de trabajo completo contribuya en la tarea del diseño.
- Se diseñaron las clases persistentes que permitieron realizar el diagrama de entidad-relación en el sistema de gestión de bases de datos.
- La utilización de las tareas de ingeniería en la implementación del sistema fue una buena práctica utilizada para los programadores, pues mostró las principales funcionalidades a implementar.
- El proceso de pruebas ayudó a validar las necesidades del cliente y a que se cumplieran todos los requisitos planteados por el mismo.

Conclusiones

A partir de los objetivos planteados y el trabajo realizado en esta investigación para llevar a cabo los procesos de administración y supervisión del sistema de la réplica “Magic@ Data Replication eXtensible Solution”, se arribó a las siguientes conclusiones:

- El empleo de los métodos teóricos y empíricos facilitó conocer el estado de los procesos para la administración y supervisión de la replicación de datos.
- Las tareas investigativas proporcionaron mayor organización durante la investigación.
- El objetivo elaborado en el diseño de la investigación fue cumplido, comprobándose la idea a defender como respuesta del problema a resolver que originó la investigación.
- La aplicación web facilita que se acceda desde diferentes lugares y que cuente con una interfaz amigable para su uso, garantizando los niveles de seguridad en la información y mayor rapidez en los resultados.
- Terminada la etapa de pruebas del sistema con resultados satisfactorios se pudo comprobar que cumple con los requisitos funcionales y no funcionales del sistema, por lo que es posible realizar el despliegue de su primera versión.

Recomendaciones

Luego de la investigación realizada en el presente trabajo y teniendo en cuenta las ideas que surgieron durante el progreso de la misma se recomienda:

- Seguir actualizando la aplicación con las nuevas versiones de Symfony y sus componentes.
- Realizar la sincronización con servidores de réplica que estén en el sistema operativo Windows, utilizando los protocolos de Samba y FTP.

Bibliografía

Delgado, Alejandro y Corbera Navas , Francisco Gallego. 2008. *MODELOS AVANZADOS DE BASES DE DATOS*. 2008.

Murillo Alfaro, Félix. 1999. *Herramientas Case*. 1999.

Pacheco, Nacho. 2011. Cómo probar repositorios Doctrine. *Cómo probar repositorios Doctrine*. [En línea] 2011. [Citado el: 20 de abril de 2012.] <http://gitnacho.github.com/symfony-docs-es/cookbook/testing/doctrine.html#como-probar-repositorios-doctrine>.

Seoane Pascual, Joaquín y González Barahona, Jesús M. 2007 . *Introducción al software libre*. Madrid : s.n., 2007 .

Tupe, Cirley y Cisneros, Juan. 2008. *Evaluación y Selección de Frameworkde Desarrollo PHP:Symfony, Kumbia, CakePHP y Zend*. 2008.

Álvarez Carrión. 1999. *Integración de esquemas en bases de datos heterogéneas fuertemente acopladas*. 1999.

Álvarez Marañón, Gonzalo. 1999. Características del lenguaje java. *Características del lenguaje java*. [En línea] 1999. [Citado el: 15 de febrero de 2012.] <http://www.iec.csic.es/cryptonomicon/java/quesjava.html>.

Alvarez, Miguel Angel. 2003. Zend Studio. *Zend Studio*. [En línea] 2003. [Citado el: 2 de marzo de 2012.] <http://www.desarrolloweb.com/articulos/1178.php>.

Brito Zambrano, Paola Nataly y Caguana Romero, Mariuxi Beatriz. 2011. *IDES para Java*. 2011.

Castro, Fernando Lorenzo. 2000. *MODELOS DE DATOS, CONCEPTOS Y CLASIFICACIONES*. 2000.

Eguiluz, Javier . 2010. Symfony.es. *Symfony.es*. [En línea] 25 de febrero de 2010. [Citado el: 20 de abril de 2012.] <http://www.symfony.es/2010/02/24/symfony-2-utilizara-phpunit/>.

Eguiluz, Javier. 2011. *Desarrollo web ágil con Symfony2*. 2011.

EMC Corporation . 2012. EMC Replication Manager - Data Replication Software - EMC. *EMC Replication Manager - Data Replication Software - EMC*. [En línea] 2012. [Citado el: 20 de febrero de 2012.] <http://www.emc.com/replication/replication-manager.htm..>

ExpressionEngine. 2006. CodeIgniter User Guide Version 2.1.0. *CodeIgniter User Guide Version 2.1.0*. [En línea] 2006. [Citado el: 25 de febrero de 2012.] http://codeigniter.com/user_guide/index.html.

Fraire, Joaquín Huacuja. 2005. *Automatización del Diseño de la Fragmentación Horizontal en Bases de Datos Distribuidas*. 2005.

- G. Figueroa, Roberth , J. Solís, Camilo y A. Cabrera, Armando . 2010.** *Metodologías Tradicionales vs. Metodologías Ágiles.* 2010.
- Garrido Montero, Jesús Manuel. 2004.** *Plataforma Eclipse Introducción Técnica.* 2004.
- GÓMEZ, PATRICIA , y otros. 2005.** *PHP y MySQL. Tecnologías para el desarrollo de aplicaciones web.* s.l. : ÁNGEL COBO, PATRICIA GÓMEZ, DANIEL PÉREZ, ROCÍO ROCHA, 2005.
- Grupo Satelite s.a. 2011.** Modelamiento de Datos. *Modelamiento de Datos.* [En línea] 2011. [Citado el: 15 de marzo de 2012.]
file:///C:/Documents%20and%20Settings/martha/Mis%20documentos/Descargas/visual%20paradigm/index.php.htm.
- Gutmans, Andi y Rethans, Derick . 2004.** *PHP 5 Power Programming.* 2004.
- Hansen, Gary W y Hansen, James V.** *Diseño y Administración de Bases de Datos.* s.l. : Prentice Hall.
- Hitachi Data Systems Corporation. 2012.** Hitachi Data Systems. *Hitachi Data Systems .* [En línea] 2012. [Citado el: 20 de febrero de 2012.] <http://www.hds.com/products/storage-/hitachi-replication-manager.html>.
- Jonathan H. Wage. 2007.** doctrine. *doctrine.* [En línea] 2007. [Citado el: 2 de marzo de 2012.]
www.doctrine-project.org.
- Landrian, Jorge. 2009.** Magic@ Data Replication eXtensible Solution. 2009.
- Martínez Martín, Laura. 2010.** *Diseño y construcción de bases de datos distribuidas heterogéneas sobre Oracle y SQL Server.* s.l. : Universidad Carlos III de Madrid. Departamento de Informática, 2010.
- Matos García, Rosa María. 1999.** *DISEÑO de BASES DE DATOS.* 1999.
- McCool, Robert . 2011.** *Servidor HTTP Apache.* 2011.
- Mendía Arriola, Héctor Alberto Heber. 2006.** *METODOLOGÍAS ÁGILES INCORPORADAS A LAS NECESIDADES DE LAS EMPRESAS QUE DESARROLLAN SOFTWARE EN GUATEMALA.* Guatemala : s.n., 2006.
- Microsoft. 2012.** compucaja.net. [En línea] 2012. [Citado el: 5 de febrero de 2012.]
<http://www.compucaja.net/>.
- . 2012.** MSND-Explorar el desarrollo de escritorio, web, nube y teléfono. [En línea] 2012. [Citado el: 25 de febrero de 2012.] <http://msdn.microsoft.com/es-es/library/ms151198.aspx..>
- Molina García, Jesús. 2008.** *El Lenguaje Unificado de Modelado, UML.* s.l. : Departamento de Informática y Sistemas Universidad de Murcia., 2008.

Mora, Beatriz , y otros. 2006. *Definición de Lenguajes de Modelos MDA vs DSL*. España : Universidad de Castilla-La Mancha, 2006.

Nazep. 2011. Comunidad PHP. *Comunidad PHP*. [En línea] 2011. [Citado el: 5 de marzo de 2012.] http://www.comunidadphp.org/index.php?sec=8&clave_articulo=106.

Núñez, Camellea. 2004. *Gestión de base de datos con ado.Net*. 2004.

Oren Ben-Kiki, Clark Evans, and Brian Ingerson. 2004. *YAML Ain't Markup Language (YAML™)*. 2004.

Parra, Eduardo . 2011. Portal Ubuntu. *Portal Ubuntu*. [En línea] 2011. [Citado el: febrero de 20 de 2012.] <http://www.portalubuntu.com/2011/04/instalar-netbeans-70-en-espanol-en.html>.

Pecos, Daniel. 2002. *Comparativa MySQL vs PostgreSQL*. 2002.

Penadés, Patricio Letelier y M^a, Carmen. 2006. *Métodologías ágiles para el desarrollo de :eXtreme Programming (XP)*. Valencia : s.n., 2006.

SlideShare Inc. 2008. Replicación Base De Datos. *Replicación Base De Datos*. [En línea] 2008. [Citado el: 25 de febrero de 2012.] <http://www.slideshare.net/luisfe/replicacin-base-de-datos-488210#>.

Softpedia. 2012. Softpedia. *Softpedia*. [En línea] 2012. [Citado el: 20 de marzo de 2012.] <http://www.softpedia.es/programa-Nginx-138898.html>.

Software, Grupo de Ingeniería del. 2004. *Introducción a las Aplicaciones Web*. s.l. : Departamento de Lenguajes y Sistemas Informáticos, 2004.

Solis Calero, Manuel. 2003. *Una explicación de la programación extrema (XP)*. Madrid : s.n., 2003.

Total Publishing Network S.A. 2008. muyLinux. *muyLinux*. [En línea] 2008. [Citado el: 02 de marzo de 2012.] <http://www.muylinux.com/2011/04/19/disponible-el-servidor-web-nginx-1-0-0/>.

Trejo Miranda, María Isabel. 2011. *Ventaja de las bases de datos fente a los ficheros clásicos*. 2011.

Zaninotto, François y Potencier, Fabien . 2008. *Symfony 1.2, la guía definitiva*. 2008.

Glosario de Términos

API: (*Application Program Interface*, Interfaz de Programación de Aplicaciones). Herramientas de programación para rutinas, protocolos y software. Propuesta por Microsoft como una alternativa más rápida que el **CGI**. Dicha Interfaz proporciona funciones que suelen permitir entre otras cosas el acceso a BBDD y la gestión de sesiones de usuario.

Bundle: Son una mezcla entre los módulos y los plugins. Es aquí donde se encuentra la programación realmente y la idea del Bundle es que si necesito una misma funcionalidad dentro de otro proyecto debería poder copiarlo completo al nuevo proyecto y debería ser lo suficientemente independiente como para funcionar.

CGI:(*Common Gateway Interface*, Interfaz de entrada común). Es tecnología web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web.

Frameworks: Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Hardware: Conjunto de componentes físicos de una computadora. Refiérase a objetos tangibles y palpables como son los discos, lectores de discos, monitores, teclados, las impresoras, tarjetas y chips.

ORM: Acrónimo de *Object-Relational Mapping* (Mapeo Objeto-Relacional). Framework que utiliza técnicas de programación para convertir datos a objetos y viceversa, permitiendo el trabajo con datos persistentes como si formaran parte de una Base de Datos orientada a objetos.

Plugin: Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Software libre: Software que puede ser distribuido, modificado, redistribuido, copiado y usado libremente.

Software: Es un término genérico que designa al conjunto de programas de distintos tipos (sistema operativo y aplicaciones diversas) que hacen posible operar con el ordenador.

Anexos

Anexo # 1:

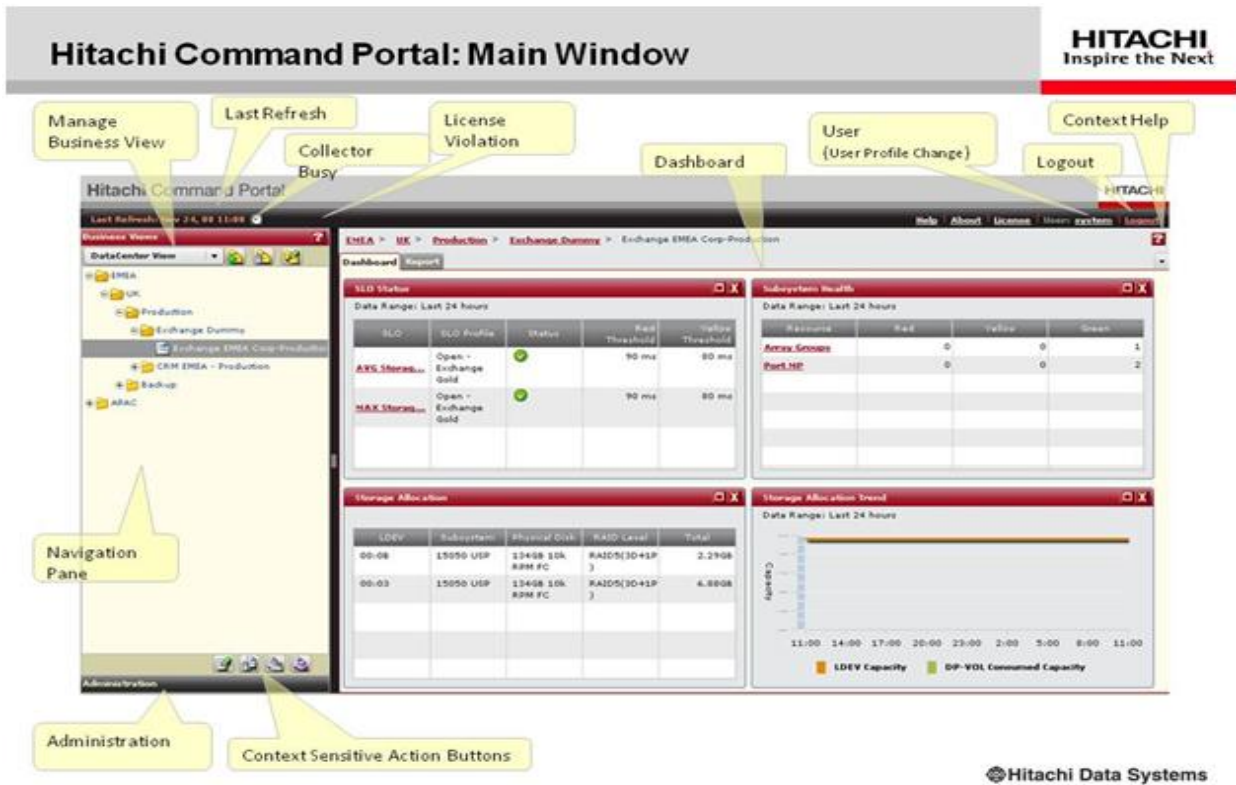


Figura 5: Portal de Hitachi Replication Manager

Anexo # 2 Historias de usuarios

Tabla 27: HU Gestionar Rangos de IP

Historia de Usuario	
Número: HU1	Usuario: Desarrollador
Nombre historia: Gestionar Rangos de IP.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Martha María Rondón Regalado.	

<p>Descripción:</p> <p>Permitirá agregar nuevos rangos de IP, donde el usuario podrá insertar los datos y podrá asignarle un nuevo grupo de usuario al mismo, además de mostrar una lista de rangos existentes, permitiendo para cada uno de estos la opción de modificar, eliminar y mostrar su contenido. También se podrá filtrar los rangos de IP por el nombre, descripción o dirección IP.</p>
<p>Observaciones:</p>

Tabla 28: HU Gestionar servidores de base de datos

Historia de Usuario	
Número: HU3	Usuario: Desarrollador
Nombre historia: Gestionar servidores de base de datos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Martha María Rondón Regalado.	
<p>Descripción:</p> <p>Permitirá añadir nuevos servidores de base de datos, donde el usuario podrá insertar los datos, además se mostrará una lista de los servidores de base de datos existentes, permitiendo para cada uno de estos la opción de modificar, eliminar y mostrar su contenido. También se podrán filtrar los mismos por el nombre, descripción, dirección IP, si están activos o no y por el tipo de base de datos.</p>	
Observaciones:	

Tabla 29: HU Gestionar ficheros

Historia de Usuario	
Número: HU4	Usuario: Desarrollador
Nombre historia: Gestionar ficheros.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Martha María Rondón Regalado.	

<p>Descripción:</p> <p>Permitirá agregar nuevos ficheros de configuración, donde el usuario podrá insertar los datos, además se mostrará una lista de los ficheros existentes, permitiendo para cada uno de estos la opción de modificar, eliminar y mostrar su contenido. También se podrán filtrar los ficheros por el nombre, descripción, tipo de ficheros de configuración.</p>
<p>Observaciones:</p>

Tabla 30: HU Gestionar tipo de base de datos

Historia de Usuario	
Número: HU5	Usuario: Desarrollador
Nombre historia: Gestionar tipo de base de datos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Martha María Rondón Regalado.	
<p>Descripción:</p> <p>Permitirá añadir nuevos tipos de base de datos, además se mostrará una lista con los tipos de base de datos existentes, permitiendo para cada uno de estos la opción de modificar, eliminar y mostrar su contenido. También se podrán filtrar los tipos de base de datos por el nombre y su descripción.</p>	
Observaciones:	

Tabla 31: HU Gestionar esquema de sincronización

Historia de Usuario	
Número: HU6	Usuario: Desarrollador
Nombre historia: Gestionar esquema de sincronización.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Martha María Rondón Regalado.	

<p>Descripción:</p> <p>Permitirá añadir nuevos esquemas de sincronización, también se mostrará una lista con los esquemas de sincronización existentes, permitiendo además para cada uno de estos la opción de modificar, eliminar y mostrar su contenido. Además se podrán filtrar los esquemas por el nombre, descripción, esquema de sincronización, estructura a sincronizar o el destino estructura a sincronizar.</p>
<p>Observaciones:</p>

Tabla 32: HU Gestionar sincronización

Historia de Usuario	
Número: HU7	Usuario: Desarrollador
Nombre historia: Gestionar sincronización.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Martha María Rondón Regalado.	
<p>Descripción:</p> <p>Permitirá añadir nuevas sincronizaciones, también se mostrará una lista con las sincronizaciones existentes, permitiendo además para cada uno de estas la opción de modificar, eliminar y mostrar su contenido. Se podrán filtrar los esquemas por el nombre, servidor de réplica, fichero de configuración, esquema de subida, esquema de descarga, base de dato local o base de datos remota.</p>	
Observaciones:	

Tabla 33: HU Planificar las tareas programadas

Historia de Usuario	
Número: HU9	Usuario: Desarrollador
Nombre historia: Planificar las tareas programadas.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Madiela Cabeza Beltran.	

<p>Descripción:</p> <p>Permitirá añadir nuevas tareas, también se mostrará una lista con todas las tareas existentes, permitiendo además para cada uno de estas la opción de modificar, eliminar y mostrar su contenido. Se podrán filtrar las tareas por el nombre o la descripción.</p>
<p>Observaciones:</p>

Tabla 34: HU Listar servidores de base de datos

Historia de Usuario	
Número: HU10	Usuario: Desarrollador
Nombre historia: Listar servidores de base de datos.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Madiela Cabeza Beltran.	
<p>Descripción:</p> <p>Mostrará un listado con todos los servidores de base de datos activos dependiendo del grupo al que pertenece el usuario que se autenticó, y para cada uno de los servidores, debe permitir listar las sincronizaciones asociadas a cada uno de ellos. También brindará la posibilidad de mostrar al usuario el log de error de cada una de las sincronizaciones detenidas, además permitirá activarlas.</p>	
Observaciones:	

Tabla 35: HU Listar sincronizaciones

Historia de Usuario	
Número: HU11	Usuario: Desarrollador
Nombre historia: Listar sincronizaciones.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Madiela Cabeza Beltran.	

<p>Descripción:</p> <p>Mostrará un listado con todas las sincronizaciones existentes dependiendo del grupo al que pertenece el usuario que se autenticó, permitiendo además mostrar el log de error para aquellas que se encuentren detenidas, además debe permitir activarla.</p>
<p>Observaciones:</p>

Tabla 36: HU Gestionar grupos de usuarios

Historia de Usuario	
Número: HU13	Usuario: Desarrollador
Nombre historia: Gestionar grupos de usuarios.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Madiela Cabeza Beltran.	
<p>Descripción:</p> <p>Permitirá añadir nuevos grupos de usuarios al sistema, también se mostrará una lista con los todo los grupos existentes, permitiendo además para cada uno de estos, la opción de modificar, eliminar y mostrar su contenido. Además se podrán filtrar los grupos de usuario por el nombre.</p>	
Observaciones:	

Tabla 37: HU Mostrar log de las sincronizaciones

Historia de Usuario	
Número: HU14	Usuario: Desarrollador
Nombre historia: Mostrar log de las sincronizaciones.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Madiela Cabeza Beltran.	

<p>Descripción:</p> <p>El sistema mostrará un listado de todas las sincronizaciones con todos sus datos, permitiéndole al usuario escoger una de ellas para ver su log en tiempo real, es decir, lo que está pasando en el servidor de réplica con ella en ese instante.</p>
<p>Observaciones:</p>

Tabla 38: HU Autenticar usuario

Historia de Usuario	
Número: HU15	Usuario: Desarrollador
Nombre historia: Autenticar usuario.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Madiela Cabeza Beltran.	
<p>Descripción:</p> <p>El sistema permitirá introducir los datos del usuario y validará que los mismos sean correctos para permitirle el acceso a la aplicación dependiendo del grupo al que pertenezca.</p>	
<p>Observaciones:</p>	

Anexo # 3 Tarjetas CRC

Tabla 39: Tarjeta CRC AdminController

AdminController	
Funcionalidades	Colaboraciones
listarAction sincronizarAction	DsincronizacionRepository folder ssh DcambioRepository DservidorReplicaRepository

Tabla 40: Tarjeta CRC clase DsincronizacionRepository

DsincronizacionRepository

Funcionalidades	Colaboración
-----------------	--------------

Tabla 41: Tarjeta CRC DservidorDbRepository

DservidorDbRepository	
Funcionalidades	Colaboraciones
findTodaSincronizaciones buscarNoSincronizados actualizarByFilename modificarServerReplica GetServidoresPorGrupo existeServidores	DservidorDb EntityManager DsincronizacionRepository

Tabla 42: Tarjeta CRC DcambiosRepository

DcambiosRepository:	
Funcionalidades	Colaboraciones
existecambio buscarcambios saveCambio	Dcambio EntityManager

Tabla 43: Tarjeta CRC ActualizarListerner

ActualizarListerner:	
Funcionalidades	Colaboraciones
preUpdate prePersist postRemove	DcambioRepository DsincronizacionRepository DservidorDbRepository

Anexo # 4: Tareas de ingeniería en la iteración 1.

Tabla 44: HU4_T1

Tarea	
Número de la Tarea: HU4_T1	HU: Gestionar ficheros.

Nombre: Agregar, modificar, eliminar, mostrar y listar los ficheros de configuración.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 24 de febrero	Fecha Fin: 2 de marzo
Responsable: Martha María Rondón Regalado.	
Descripción: Deberá agregar, eliminar, mostrar y listar todos los ficheros de configuración. Se deberá validar que no se adicionen ficheros con igual nombre y que no se dejen campos vacíos, en caso contrario mostrará un mensaje de error informando al usuario. Los datos insertados deben ser guardados.	

Tabla 45: HU5_T1

Tarea	
Número de la Tarea: HU5_T1	HU: Gestionar tipo de base de datos.
Nombre: Agregar, modificar, eliminar, mostrar y listar los tipos de base de datos.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 2 de marzo	Fecha Fin: 9 de marzo
Responsable: Madiela Cabeza Beltran.	
Descripción: Deberá agregar, eliminar, mostrar y listar todos los tipos de base de datos. También se debe validar que no se adicionen tipos de base de datos existentes con igual nombre y que no se dejen campos vacíos, en caso que exista se debe mostrar al usuario un mensaje de notificación informando al usuario. Los datos insertados deben ser guardados.	

Tabla 46: HU6_T1

Tarea	
Número de la Tarea: HU6_T1	HU: Gestionar esquema de sincronización.
Nombre: Agregar, modificar, eliminar, mostrar y listar los esquemas de sincronización.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 9 de marzo	Fecha Fin: 16 de marzo
Responsable: Madiela Cabeza Beltran.	
Descripción: Deberá agregar, eliminar, mostrar y listar todos los esquemas de sincronización. Se deberá validar que no se adicionen esquemas existentes y que no se dejen campos vacíos, en caso que exista se debe mostrar al usuario un mensaje de error al usuario. Los datos insertados deben ser guardados.	

Tabla 47: HU9_T1

Tarea	
Número de la Tarea: HU9_T1	HU: Planificar las tareas programadas.
Nombre: Agregar, modificar, eliminar, mostrar y listar los esquemas de sincronización.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 16 de marzo	Fecha Fin: 23 de marzo
Responsable: Martha María Rondón Regalado.	
Descripción: Deberá agregar, eliminar, mostrar y listar todas las tareas programadas por el cliente para desarrollar las sincronizaciones. Se deberá validar que no se dejen campos vacíos y en caso contrario se debe mostrar al usuario un mensaje de error al usuario. Los datos insertados deben ser guardados.	

Tabla 48: HU3_T1

Tarea	
Número de la Tarea: HU3_T1	HU: Gestionar servidores de base de datos.
Nombre: Agregar, modificar, eliminar, mostrar y listar los servidores de base de datos.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 17 de febrero	Fecha Fin: 24 de febrero
Responsable: Martha María Rondón Regalado.	
Descripción: Deberá agregar, eliminar, mostrar y listar todos los datos asociados a los servidores de base de datos. Se deberá validar que no se adicionen servidores con igual nombre, que no se dejen campos vacíos y en caso contrario mostrará un mensaje de error informando al usuario. Se deben guardar los datos insertados.	

Anexo # 5: Tareas de ingeniería en la iteración 2.

Tabla 49: HU7_T1

Tarea	
Número de la Tarea: HU7_T1	HU: Gestionar sincronización.
Nombre: Agregar, modificar, eliminar, mostrar y listar las sincronizaciones.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 23 de marzo	Fecha Fin: 30 de marzo

Responsable: Madiela Cabeza Beltran.
Descripción: Deberá agregar, eliminar, mostrar y listar todos los datos asociados a las sincronizaciones. Se deberá validar que no se adicionen sincronizaciones con igual nombre, que no se dejen campos vacíos y en caso contrario mostrará un mensaje de error informando al usuario. Se deben guardar los datos insertados.

Tabla 50: HU10_T1

Tarea	
Número de la Tarea: HU9_T1	HU: Listar servidores de base de datos.
Nombre: Listar los servidores de base de datos y para cada uno de estos, mostrar las sincronizaciones asociadas al mismo.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 6 de abril	Fecha Fin: 13 de abril
Responsable: Madiela Cabeza Beltran.	
Descripción: Deberá mostrar y listar todos los servidores de base de datos que existen activos dependiendo del grupo en que se encuentre el usuario autenticado y para cada uno de ellos deberá mostrar todas las sincronizaciones asociadas al servidor escogido, permitiéndole ver el log de error de la sincronización detenida para luego activarla.	

Tabla 51: HU11_T1

Tarea	
Número de la Tarea: HU11_T1	HU: Listar sincronizaciones.
Nombre: Listar las sincronizaciones y mostrar por cada una de ellas los log de error y permitir activar las que se encuentren detenidas.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 13 de abril	Fecha Fin: 20 de abril
Responsable: Martha María Rondón Regalado.	
Descripción: Deberá mostrar y listar todas las sincronizaciones dependiendo del grupo en que se encuentre el usuario autenticado, y para cada una de ellas deberá mostrar los log de error, para la sincronización detenida para luego activarla.	

Anexo # 6: Tareas de ingeniería en la iteración 3.

Tabla 52: HU13_T1

Tarea	
Número de la Tarea: HU13_T1	HU: Gestionar grupo de usuarios.
Nombre: Agregar, modificar, eliminar, mostrar y listar los grupos de usuarios.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 27 de abril	Fecha Fin: 4 de mayo
Responsable: Madiela Cabeza Beltran.	
Descripción: Deberá agregar, eliminar, mostrar y listar todos los grupos de usuarios. Se deberá validar que no se dejen campos vacíos, en caso contrario mostrará un mensaje de error informando al usuario. Los datos insertados deben ser guardados.	

Tabla 53: HU1_T1

Tarea	
Número de la Tarea: HU1_T1	HU: Gestionar Rangos de IP.
Nombre: Agregar, modificar, eliminar, mostrar y listar los rangos de IP.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 4 de mayo	Fecha Fin: 11 de mayo
Responsable: Madiela Cabeza Beltran.	
Descripción: Deberá agregar, eliminar, mostrar y listar todos los rangos de IP. Se deberá validar que no se dejen campos vacíos y que el IP sea válido, en caso contrario mostrará un mensaje de error informando al usuario. Los datos insertados deben ser guardados. Además posibilitará asignarle a un rango de IP determinado, un grupo de usuario que serán los que tendrán acceso a la aplicación desde un IP que se encuentre en ese rango.	

Tabla 54: HU14_T1

Tarea	
Número de la Tarea: HU14_T1	HU: Mostrar log de las sincronizaciones.
Nombre: Listar las todas las sincronizaciones existentes y para cada una de ellas mostrar el log en tiempo real que se encuentra en el servidor de réplica.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 11 de mayo	Fecha Fin: 18 de mayo

Responsable: Martha Rondón Regalado.
Descripción: Deberá mostrar un listado de todas las sincronizaciones dependiendo del grupo en que se encuentre el usuario y para cada una de ellas brindará la posibilidad de ver en tiempo real lo que está sucediendo con la misma en el servidor de réplica al que pertenece.

Tabla 55: HU15_T1

Tarea	
Número de la Tarea: HU15_T1	HU: Autenticar usuario.
Nombre: Controlar el acceso a la aplicación.	
Tipo: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 18 de mayo	Fecha Fin: 25 de mayo
Responsable: Martha Rondón Regalado.	
Descripción: Deberá permitir al usuario acceder a la aplicación y en dependencia al grupo que este pertenezca podrá interactuar con ella.	

Anexo # 7: Pruebas de aceptación en la iteración 1.

Tabla 56: Prueba de aceptación Gestionar servidores de base de datos: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU3_P1	HU: Gestionar servidores de base de datos.
Nombre: Gestionar servidores de base de datos.	
Descripción: Prueba de funcionalidad para gestionar servidores de base de datos, específicamente lograr agregar nuevos usuarios al sistema.	
Condiciones de ejecución: El administrados debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un nuevo servidor de base de datos. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). <ol style="list-style-type: none"> 3.1 El administrador introduce los datos a editar y escoge la opción que prefiere: <p style="text-align: center;">Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio).</p> 	

<p>Actualizar y Cerrar (Se actualizan los cambios, se listan todos los servidores de base de datos y muestra mensaje satisfactorio).</p> <p>Borrar (Ver Prueba 3 paso1).</p> <p>Mostrar (Ver paso 5).</p> <p>Agregar nuevo (Vuelve al paso 2).</p> <p>Volver a la lista (Ver Prueba 2 paso 2).</p> <p>4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1).</p> <p>5. El administrador observa los datos del servidor de base de datos.</p> <p>6. El administrador escoge la opción que desea:</p> <p style="padding-left: 40px;">Editar (Ver paso 3.1).</p> <p style="padding-left: 40px;">Agregar nuevo (Vuelve al paso 2).</p>
<p>Resultado esperado: Se guardan los datos del nuevo servidor de base de datos y si se editan los datos, se guardan los cambios realizados, además se muestran los mismos si el administrador desea.</p>
<p>No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un servidor de base de datos.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 57: Prueba de aceptación Gestionar servidores de base de datos: Listar

Caso de Prueba de Aceptación	
Código: HU3_P2	HU: Gestionar servidores de base de datos.
Nombre: Gestionar servidores de base de datos.	
Descripción: Prueba de funcionalidad para gestionar servidores de base de datos, con el fin de listar todos los servidores de base de datos que existen en el sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Listar. 2. Selecciona para cada servidor de base de datos que se encuentra en la lista, la opción que desea: <p style="padding-left: 40px;">Editar (Ver Prueba 1 paso 3.1).</p> 	

<p>Borrar (Ver Prueba 3 paso1).</p> <p>Agregar nuevo (Prueba 1 paso 2).</p> <p>Borrar todos los elementos. (Se muestra un mensaje de confirmación).</p> <p>2.1. El administrador escoge la opción que desea:</p> <p>Si, borrar (Se muestra un mensaje satisfactorio).</p> <p>Volver a la lista (Volver paso 2).</p> <p>Agregar nuevo (Volver prueba 1 paso 2).</p> <p>3. El administrador escoge la opción: Filtros.</p> <p>3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción:</p> <p>Filtrar (Filtra el listado de servidores de base de datos por el campo defino).</p> <p>Resetear (Muestra el listado original).</p>
<p>Resultado esperado: Se listan todos los servidores de base de datos con todos sus contenidos.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 58: Prueba de aceptación Gestionar servidores de base de datos: Borrar

Caso de Prueba de Aceptación	
Código: HU3_P3	HU: Gestionar servidores de base de datos.
Nombre: Gestionar servidores de base de datos.	
Descripción: Prueba de funcionalidad para gestionar servidores de base de datos, con el fin de borrarlos del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el servidor de base de datos y se muestra un mensaje satisfactorio). Puede escoger la opción que desea: <ul style="list-style-type: none"> Editar (Prueba 1 paso 3.1). Agregar nuevo (Prueba 1 paso 2). 	
Resultado esperado: Se borran los servidores de base de datos con el contenido que poseía y el sistema	

muestra un mensaje de confirmación satisfactoria.
No conformidades:
Evaluación de la prueba: Prueba satisfactoria.

Tabla 59: Prueba de aceptación Gestionar ficheros: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU4_P1	HU: Gestionar ficheros.
Nombre: Gestionar ficheros.	
Descripción: Prueba de funcionalidad para gestionar los ficheros de configuración, específicamente lograr agregar nuevos ficheros al sistema.	
Condiciones de ejecución: El administrados debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un nuevo fichero. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). <ol style="list-style-type: none"> 3.1 El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio). Actualizar y Cerrar (Se actualizan los cambios, se listan todos los ficheros y muestra mensaje satisfactorio). Borrar (Ver Prueba 3 paso1). Mostrar (Ver paso 5). Agregar nuevo (Vuelve al paso 2). Volver a la lista (Ver Prueba 2 paso 2). 4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1). 5. El administrador observa los datos del fichero. 6. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Editar (Ver paso 3.1). Agregar nuevo (Vuelve al paso 2). 	

Resultado esperado: Se guardan los datos del nuevo fichero de configuración y si se editan sus datos, se guardan los cambios realizados, además se muestran los mismos si el administrador desea.
No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un fichero.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 60: Prueba de aceptación Gestionar ficheros: Listar

Caso de Prueba de Aceptación	
Código: HU4_P2	HU: Gestionar ficheros.
Nombre: Gestionar ficheros.	
Descripción: Prueba de funcionalidad para gestionar los ficheros, con el fin de listar todos los ficheros de configuración que existen en el sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Listar. 2. Selecciona para cada fichero que se encuentra en la lista, la opción que desea: <ul style="list-style-type: none"> Editar (Ver Prueba 1 paso 3.1). Borrar (Ver Prueba 3 paso1). Agregar nuevo (Prueba 1 paso 2). Borrar todos los elementos. (Se muestra un mensaje de confirmación). 2.1. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Si, borrar (Se muestra un mensaje satisfactorio). Volver a la lista (Volver paso 2). Agregar nuevo (Volver prueba 1 paso 2). 3. El administrador escoge la opción: Filtros. <ol style="list-style-type: none"> 3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción: <ul style="list-style-type: none"> Filtrar (Filtra el listado de ficheros por el campo defino). Resetear (Muestra el listado original). 	
Resultado esperado: Se listan todos los ficheros de configuración con todo su contenido.	
No conformidades:	

Evaluación de la prueba: Prueba satisfactoria.

Tabla 61: Prueba de aceptación Gestionar ficheros: Borrar

Caso de Prueba de Aceptación	
Código: HU4_P3	HU: Gestionar ficheros.
Nombre: Gestionar ficheros.	
Descripción: Prueba de funcionalidad para gestionar ficheros, con el fin de borrarlos del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el fichero y se muestra un mensaje satisfactorio). 2. Puede escoger la opción que desea: <ul style="list-style-type: none"> Editar (Prueba 1 paso 3.1). Agregar nuevo (Prueba 1 paso 2). 	
Resultado esperado: Se borran los ficheros de configuración con el contenido que poseía y el sistema muestra un mensaje de confirmación satisfactoria.	
No conformidades:	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 62: Prueba de aceptación Gestionar tipo de base de datos: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU5_P1	HU: Gestionar tipo de base de datos
Nombre: Gestionar tipo de base de datos.	
Descripción: Prueba de funcionalidad para gestionar los tipos de base de datos, específicamente lograr agregar un nuevo tipo al sistema.	
Condiciones de ejecución: El administrados debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un nuevo tipo de base de datos. 	

<p>3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio).</p> <p>3.1 El administrador introduce los datos a editar y escoge la opción que prefiere:</p> <p style="padding-left: 40px;">Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio).</p> <p style="padding-left: 40px;">Actualizar y Cerrar (Se actualizan los cambios, se listan todos los tipo de base de datos y muestra mensaje satisfactorio).</p> <p style="padding-left: 40px;">Borrar (Ver Prueba 3 paso1).</p> <p style="padding-left: 40px;">Mostrar (Ver paso 5).</p> <p style="padding-left: 40px;">Agregar nuevo (Vuelve al paso 2).</p> <p style="padding-left: 40px;">Volver a la lista (Ver Prueba 2 paso 2).</p> <p>4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1).</p> <p>5. El administrador observa los datos del tipo de base de datos.</p> <p>6. El administrador escoge la opción que desea:</p> <p style="padding-left: 40px;">Editar (Ver paso 3.1).</p> <p style="padding-left: 40px;">Agregar nuevo (Vuelve al paso 2).</p>
<p>Resultado esperado: Se guardan los datos del nuevo tipo de base de base de datos y si se editan los datos, se guardan los cambios realizados, además se muestran los mismos si el administrador desea.</p>
<p>No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un tipo de base de base de datos.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 63: Prueba de aceptación Gestionar tipo de base de datos: Listar

Caso de Prueba de Aceptación	
Código: HU5_P2	HU: Gestionar tipo de base de datos.
Nombre: Gestionar tipo de base de datos.	
Descripción: Prueba de funcionalidad para gestionar los tipos de base de datos, con el fin de listar todos los tipos que existen en el sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	

<p>1. El administrador selecciona la opción que desea: Listar.</p> <p>2. Selecciona para cada tipo de base de datos que se encuentra en la lista, la opción que desea:</p> <p style="padding-left: 40px;">Editar (Ver Prueba 1 paso 3.1).</p> <p style="padding-left: 40px;">Borrar (Ver Prueba 3 paso1).</p> <p style="padding-left: 40px;">Agregar nuevo (Prueba 1 paso 2).</p> <p style="padding-left: 40px;">Borrar todos los elementos. (Se muestra un mensaje de confirmación).</p> <p>2.1. El administrador escoge la opción que desea:</p> <p style="padding-left: 40px;">Si, borrar (Se muestra un mensaje satisfactorio).</p> <p style="padding-left: 40px;">Volver a la lista (Volver paso 2).</p> <p style="padding-left: 40px;">Agregar nuevo (Volver prueba 1 paso 2).</p> <p>3. El administrador escoge la opción: Filtros.</p> <p>3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción:</p> <p style="padding-left: 40px;">Filtrar (Filtra el listado del tipo de base de datos por el campo defino).</p> <p style="padding-left: 40px;">Resetear (Muestra el listado original).</p>
Resultado esperado: Se listan todos los tipos de base de datos con todo su contenido.
No conformidades:
Evaluación de la prueba: Prueba satisfactoria.

Tabla 64: Prueba de aceptación Gestionar tipo de base de datos: Borrar

Caso de Prueba de Aceptación	
Código: HU5_P3	HU: Gestionar tipo de base de datos.
Nombre: Gestionar tipo de base de datos.	
Descripción: Prueba de funcionalidad para gestionar los tipo de base de datos, con el fin de borrarlos del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el tipo de base de datos y se muestra un mensaje satisfactorio). 2. Puede escoger la opción que desea: 	

<p>Editar (Prueba 1 paso 3.1).</p> <p>Agregar nuevo (Prueba 1 paso 2).</p>
<p>Resultado esperado: Se borran los tipos de base de datos con el contenido que poseía y el sistema muestra un mensaje de confirmación satisfactoria.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 65: Prueba de aceptación Gestionar esquema de sincronización: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU6_P1	HU: Gestionar esquema de sincronización
Nombre: Gestionar esquema de sincronización.	
Descripción: Prueba de funcionalidad para gestionar los esquema de sincronización, específicamente lograr agregar un nuevo esquema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un nuevo esquema de sincronización. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). 3.1 El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio). Actualizar y Cerrar (Se actualizan los cambios, se listan todos los esquema de sincronización y muestra mensaje satisfactorio). Borrar (Ver Prueba 3 paso1). Mostrar (Ver paso 5). Agregar nuevo (Vuelve al paso 2). Volver a la lista (Ver Prueba 2 paso 2). 4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1). 5. El administrador observa los datos del esquema de sincronización. 	

<p>6. El administrador escoge la opción que desea:</p> <p style="padding-left: 40px;">Editar (Ver paso 3.1).</p> <p style="padding-left: 40px;">Agregar nuevo (Vuelve al paso 2).</p>
<p>Resultado esperado: Se guardan los datos del nuevo esquema de sincronización y si se editan sus datos, se guardan los cambios realizados, además se muestran los mismos si el administrador desea.</p>
<p>No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un esquema de sincronización.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 66: Prueba de aceptación Gestionar esquema de sincronización: Listar

Caso de Prueba de Aceptación	
Código: HU9_P2	HU: Gestionar esquema de sincronización.
Nombre: Gestionar esquema de sincronización.	
Descripción: Prueba de funcionalidad para gestionar los esquemas de sincronización, con el fin de listar todos los que existen en el sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Listar. 2. Selecciona para cada esquema de sincronización que se encuentra en la lista, la opción que desea: <ul style="list-style-type: none"> Editar (Ver Prueba 1 paso 3.1). Borrar (Ver Prueba 3 paso1). Agregar nuevo (Prueba 1 paso 2). Borrar todos los elementos. (Se muestra un mensaje de confirmación). 2.1. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Si, borrar (Se muestra un mensaje satisfactorio). Volver a la lista (Volver paso 2). Agregar nuevo (Volver prueba 1 paso 2). 3. El administrador escoge la opción: Filtros. 3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción: 	

<p>Filtrar (Filtra el listado de esquemas de sincronización por el campo defino).</p> <p>Resetear (Muestra el listado original).</p>
<p>Resultado esperado: Se listan todos los esquemas de sincronización con todo su contenido.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 67: Prueba de aceptación Gestionar esquema de sincronización: Borrar

Caso de Prueba de Aceptación	
Código: HU9_P3	HU: Gestionar esquema de sincronización.
Nombre: Gestionar esquema de sincronización.	
Descripción: Prueba de funcionalidad para gestionar los esquema de sincronización, con el fin de borrarlos del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el esquema de sincronización y se muestra un mensaje satisfactorio). 2. Puede escoger la opción que desea: <ul style="list-style-type: none"> Editar (Prueba 1 paso 3.1). Agregar nuevo (Prueba 1 paso 2). 	
Resultado esperado: Se borran los esquemas de sincronización con el contenido que poseía y el sistema muestra un mensaje de confirmación satisfactoria.	
No conformidades:	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 68: Prueba de aceptación Planificar las tareas programadas: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU9_P1	HU: Planificar las tareas programadas
Nombre: Planificar las tareas programadas.	
Descripción: Prueba de funcionalidad para planificar las tareas programadas, específicamente lograr agregar nuevas tareas para llevar a cabo las sincronizaciones.	

<p>Condiciones de ejecución: El administrados debe estar autenticado. Se utilizará un usuario con datos válidos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar una nueva tarea programada. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). 3.1 El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio). Actualizar y Cerrar (Se actualizan los cambios, se listan todos las tareas programadas y muestra mensaje satisfactorio). Borrar (Ver Prueba 3 paso1). Mostrar (Ver paso 5). Agregar nuevo (Vuelve al paso 2). Volver a la lista (Ver Prueba 2 paso 2). 4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1). 5. El administrador observa los datos de las tareas programadas. 6. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Editar (Ver paso 3.1). Agregar nuevo (Vuelve al paso 2).
<p>Resultado esperado: Se guardan los datos de la nueva tarea programada y si se editan sus datos se guardan los cambios realizados, además se muestran los mismos si el administrador desea.</p>
<p>No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear una tarea programada.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 69: Prueba de aceptación Planificar las tareas programadas: Listar

Caso de Prueba de Aceptación	
Código: HU9_P2	HU: Planificar las tareas programadas.
Nombre: Planificar las tareas programadas.	
Descripción: Prueba de funcionalidad para planificar tareas programadas, con el fin de listar todas las tareas que existen en el sistema.	

<p>Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.</p>
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Listar. 2. Selecciona para cada tarea programada que se encuentra en la lista, la opción que desea: <ul style="list-style-type: none"> Editar (Ver Prueba 1 paso 3.1). Borrar (Ver Prueba 3 paso1). Agregar nuevo (Prueba 1 paso 2). Borrar todos los elementos (Se muestra un mensaje de confirmación). <ol style="list-style-type: none"> 2.1. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Si, borrar (Se muestra un mensaje satisfactorio). Volver a la lista (Volver paso 2). Agregar nuevo (Volver prueba 1 paso 2). 3. El administrador escoge la opción: Filtros. <ol style="list-style-type: none"> 3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción: <ul style="list-style-type: none"> Filtrar (Filtra el listado de las tareas programadas por el campo defino). Resetear (Muestra el listado original).
<p>Resultado esperado: Se listan todas las tareas planificadas por el administrador con todo su contenido.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 70: Prueba de aceptación Planificar las tareas programadas: Borrar

Caso de Prueba de Aceptación	
Código: HU9_P3	HU: Planificar las tareas programadas.
Nombre: Planificar las tareas programadas.	
Descripción: Prueba de funcionalidad para gestionar las tareas programadas, con el fin de borrarlas del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	

<ol style="list-style-type: none"> 1. El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina la tarea programada y se muestra un mensaje satisfactorio). 2. Puede escoger la opción que desea: <ul style="list-style-type: none"> Editar (Prueba 1 paso 3.1). Agregar nuevo (Prueba 1 paso 2).
Resultado esperado: Se borran las tareas programadas con el contenido que poseía y el sistema muestra un mensaje de confirmación satisfactoria.
No conformidades:
Evaluación de la prueba: Prueba satisfactoria.

Anexo # 8: Pruebas de aceptación en la iteración 2.

Tabla 71: Prueba de aceptación Listar servidores de base de datos.

Caso de Prueba de Aceptación	
Código: HU10_P1	HU: Listar servidores de base de datos.
Nombre: Listar servidores de base de datos.	
Descripción: Prueba de funcionalidad para listar todos los servidores de base datos que estén activos y en el caso de un usuario pertenezca al grupo supervisor solo podrá ver aquellos servidores de base de datos que se encuentren dentro de este grupo.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El usuario selecciona la opción Listado de servidores de base datos. 2. El usuario de la lista de servidores de base datos activos, escoge la opción: Ver sincronizaciones. 3. El usuario puede ver todas las sincronizaciones asociadas al servidor escogido anteriormente dependiendo del grupo al que pertenezca el usuario. 4. El usuario puede ver el log de error de la sincronización detenida, escogiendo la opción: Ver log. 5. El usuario activa las sincronizaciones detenidas, escogiendo la opción: Activar. 6. El usuario puede ver que el estado de la sincronización cambia a activo. 	
Resultado esperado: Se listan los servidores de base datos activos y se activan aquellas sincronizaciones detenidas asociadas al servidor escogido, si el administrador desea.	

No conformidades: Solamente se podrá ver el log si se descarga en fichero.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 72: Prueba de aceptación Listar sincronizaciones.

Caso de Prueba de Aceptación	
Código: HU11_P1	HU: Listar sincronizaciones.
Nombre: Listar sincronizaciones.	
Descripción: Prueba de funcionalidad para listar todas las sincronizaciones.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción Listado de sincronizaciones. 2. El usuario dependiendo al grupo en que se encuentre, puede ver las sincronizaciones asociadas a él. 3. El usuario puede ver el log de error de la sincronización detenida, escogiendo la opción: Ver log. 4. El usuario activa las sincronizaciones detenidas, escogiendo la opción: Activar. 5. El usuario puede ver que el estado de la sincronización cambia a activo. 	
Resultado esperado: Se listan todas las sincronizaciones y aquellas que se encuentran detenidas el usuario las podrá activar si lo desea.	
No conformidades: Solamente se podrá ver el log si se descarga en fichero.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 73: Prueba de aceptación Gestionar sincronización: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU7_P1	HU: Gestionar sincronización.
Nombre: Gestionar sincronización.	
Descripción: Prueba de funcionalidad para gestionar las sincronizaciones, específicamente lograr agregar nuevas.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar una sincronización. 	

<p>3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio).</p> <p>3.1 El administrador introduce los datos a editar y escoge la opción que prefiere:</p> <p style="padding-left: 40px;">Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio).</p> <p style="padding-left: 40px;">Actualizar y Cerrar (Se actualizan los cambios, se listan todas las sincronizaciones y muestra mensaje satisfactorio).</p> <p style="padding-left: 40px;">Borrar (Ver Prueba 3 paso1).</p> <p style="padding-left: 40px;">Mostrar (Ver paso 5).</p> <p style="padding-left: 40px;">Agregar nuevo (Vuelve al paso 2).</p> <p style="padding-left: 40px;">Volver a la lista (Ver Prueba 2 paso 2).</p> <p>4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1).</p> <p>5. El administrador observa los datos de las sincronizaciones.</p> <p>6. El administrador escoge la opción que desea:</p> <p style="padding-left: 40px;">Editar (Ver paso 3.1).</p> <p style="padding-left: 40px;">Agregar nuevo (Vuelve al paso 2).</p>
<p>Resultado esperado: Se guardan los datos de la nueva sincronización y si se editan los datos se guardan los cambios realizados, además se muestran los mismos si el administrador desea.</p>
<p>No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear una sincronización.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 74: Prueba de aceptación Gestionar sincronización: Listar

Caso de Prueba de Aceptación	
Código: HU7_P2	HU: Gestionar sincronización.
Nombre: Gestionar sincronización.	
Descripción: Prueba de funcionalidad para gestionar sincronización, con el fin de listar todas las sincronizaciones que existen en el sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<p>1. El administrador selecciona la opción que desea: Listar.</p>	

<p>2. Selecciona para cada sincronización que se encuentra en la lista, la opción que desea:</p> <p>Editar (Ver Prueba 1 paso 3.1).</p> <p>Borrar (Ver Prueba 3 paso1).</p> <p>Agregar nuevo (Prueba 1 paso 2).</p> <p>Borrar todos los elementos (Se muestra un mensaje de confirmación).</p> <p>2.1. El administrador escoge la opción que desea:</p> <p>Si, borrar (Se muestra un mensaje satisfactorio).</p> <p>Volver a la lista (Volver paso 2).</p> <p>Agregar nuevo (Volver prueba 1 paso 2).</p> <p>3. El administrador escoge la opción: Filtros.</p> <p>3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción:</p> <p>Filtrar (Filtra el listado de las sincronizaciones por el campo defino).</p> <p>Resetear (Muestra el listado original).</p>
<p>Resultado esperado: Se listan todas las sincronizaciones con todo su contenido.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 75: Prueba de aceptación Gestionar sincronización: Borrar

Caso de Prueba de Aceptación	
Código: HU7_P3	HU: Gestionar sincronización.
Nombre: Gestionar sincronización.	
Descripción: Prueba de funcionalidad para gestionar ficheros, con el fin de borrarlos del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<p>1. El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina la sincronización y se muestra un mensaje satisfactorio).</p> <p>2. Puede escoger la opción que desea:</p> <p>Editar (Prueba 1 paso 3.1).</p> <p>Agregar nuevo (Prueba 1 paso 2).</p>	

Resultado esperado: Se borran las sincronizaciones con el contenido que poseía y el sistema muestra un mensaje de confirmación satisfactoria.
No conformidades:
Evaluación de la prueba: Prueba satisfactoria.

Anexo # 9: Pruebas de aceptación en la iteración 3.

Tabla 76: Prueba de aceptación Gestionar Rangos de IP: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU1_P1	HU: Gestionar Rangos de IP.
Nombre: Gestionar Rangos de IP.	
Descripción: Prueba de funcionalidad para gestionar los rangos de IP, específicamente lograr agregar nuevos rangos.	
Condiciones de ejecución: El administrados debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un rango de IP. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). <ol style="list-style-type: none"> 3.1 El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio). Actualizar y Cerrar (Se actualizan los cambios, se listan todos los rangos y muestra mensaje satisfactorio). Borrar (Ver Prueba 3 paso1). Mostrar (Ver paso 5). Agregar nuevo (Vuelve al paso 2). Volver a la lista (Ver Prueba 2 paso 2). 4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1). 5. El administrador observa los datos de los rangos de IP. 6. El administrador escoge la opción que desea: 	

<p>Editar (Ver paso 3.1).</p> <p>Agregar nuevo (Vuelve al paso 2).</p>
<p>Resultado esperado: Se guardan los datos del nuevo rango de IP y si se editan los datos, se guardan los cambios realizados, además se muestran los mismos si el administrador desea.</p>
<p>No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un rango de IP.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 77: Prueba de aceptación Gestionar Rangos de IP: Listar

Caso de Prueba de Aceptación	
Código: HU1_P2	HU: Gestionar Rangos de IP.
Nombre: Gestionar Rangos de IP.	
Descripción: Prueba de funcionalidad para gestionar los rangos de IP, con el fin de listar todos los rangos que existen en el sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Listar. 2. Selecciona para cada rango que se encuentra en la lista, la opción que desea: <ul style="list-style-type: none"> Editar (Ver Prueba 1 paso 3.1). Borrar (Ver Prueba 3 paso1). Agregar nuevo (Prueba 1 paso 2). Borrar todos los elementos (Se muestra un mensaje de confirmación). <ol style="list-style-type: none"> 2.1. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Si, borrar (Se muestra un mensaje satisfactorio). Volver a la lista (Volver paso 2). Agregar nuevo (Volver prueba 1 paso 2). 3. El administrador escoge la opción: Filtros. <ol style="list-style-type: none"> 3.1. El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción: <ul style="list-style-type: none"> Filtrar (Filtra el listado de los rangos por el campo defino). Resetear (Muestra el listado original). 	

Resultado esperado: Se listan todos rangos de IP con todos sus datos.
No conformidades:
Evaluación de la prueba: Prueba satisfactoria.

Tabla 78: Prueba de aceptación Gestionar rangos de IP: Borrar

Caso de Prueba de Aceptación	
Código: HU3_P3	HU: Gestionar Rangos de IP.
Nombre: Gestionar Rangos de IP.	
Descripción: Prueba de funcionalidad para gestionar los rangos de IP, con el fin de borrarlos del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el rango y se muestra un mensaje satisfactorio). 2. Puede escoger la opción que desea: <ul style="list-style-type: none"> Editar (Prueba 1 paso 3.1). Agregar nuevo (Prueba 1 paso 2). 	
Resultado esperado: Se borran los rangos de IP con el contenido que poseía y el sistema muestra un mensaje de confirmación satisfactoria.	
No conformidades:	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 79: Prueba de aceptación Gestionar grupo de usuarios: Agregar nuevo

Caso de Prueba de Aceptación	
Código: HU13_P1	HU: Gestionar grupo de usuarios.
Nombre: Gestionar grupo de usuarios.	
Descripción: Prueba de funcionalidad para gestionar grupos de usuarios, con el fin de agregar nuevos grupos al sistema.	
Condiciones de ejecución: El administrados debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	

<ol style="list-style-type: none"> 1. El administrador selecciona la opción que desea: Agregar nuevo. 2. El administrador introduce los datos para agregar un grupo de usuarios. 3. El administrador selecciona la opción: Crear y Editar (Se muestra un mensaje satisfactorio). 3.1 El administrador introduce los datos a editar y escoge la opción que prefiere: <ul style="list-style-type: none"> Actualizar (Se actualizan los cambios y se muestra un mensaje satisfactorio). Actualizar y Cerrar (Se actualizan los cambios, se listan todos los grupos de usuarios y muestra mensaje satisfactorio). Borrar (Ver Prueba 3 paso1). Mostrar (Ver paso 5). Agregar nuevo (Vuelve al paso 2). Volver a la lista (Ver Prueba 2 paso 2). 4. El administrador selecciona la opción: Crear y agregar otro (Se muestra un mensaje satisfactorio y vuelve al paso 1). 5. El administrador observa los datos de los grupos de usuarios. 6. El administrador escoge la opción que desea: <ul style="list-style-type: none"> Editar (Ver paso 3.1). Agregar nuevo (Vuelve al paso 2).
Resultado esperado: Se guardan los datos del nuevo grupo de usuario y si se editan los datos, se guardan los cambios realizados, además se muestran si el administrador desea.
No conformidades: Aparecen palabras en inglés a la hora de mostrar y crear un grupo de usuario.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 80: Prueba de aceptación Gestionar grupo de usuarios: Listar

Caso de Prueba de Aceptación	
Código: HU13_P2	HU: Gestionar grupo de usuarios.
Nombre: Gestionar grupo de usuarios.	
Descripción: Prueba de funcionalidad para gestionar los grupos de usuarios, con el fin de listar todos los grupos.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	

<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> El administrador selecciona la opción que desea: Listar. Selecciona para cada grupo que se encuentra en la lista, la opción que desea: <ul style="list-style-type: none"> Editar (Ver Prueba 1 paso 3.1). Borrar (Ver Prueba 3 paso1). Agregar nuevo (Prueba 1 paso 2). Borrar todos los elementos (Se muestra un mensaje de confirmación). <ol style="list-style-type: none"> El administrador escoge la opción que desea: <ul style="list-style-type: none"> Si, borrar (Se muestra un mensaje satisfactorio). Volver a la lista (Volver paso 2). Agregar nuevo (Volver prueba 1 paso 2). El administrador escoge la opción : Filtros <ol style="list-style-type: none"> El administrador puede filtrar por los campos definidos en la aplicación y escoge la opción: <ul style="list-style-type: none"> Filtrar (Filtra el listado de los grupos de usuarios por el campo defino). Resetear (Muestra el listado original).
<p>Resultado esperado: Se listan todos los grupos de usuarios que existen en el sistema con todos sus datos.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 81: Prueba de aceptación Gestionar grupo de usuarios: Borrar

Caso de Prueba de Aceptación	
Código: HU13_P3	HU: Gestionar grupos de usuarios.
Nombre: Gestionar grupos de usuarios.	
Descripción: Prueba de funcionalidad para gestionar los grupos de usuarios, con el fin de borrarlos del sistema.	
Condiciones de ejecución: El administrador debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> El administrador confirma el borrado, escogiendo la opción: Si, borrar (Se elimina el rango y se 	

<p>muestra un mensaje satisfactorio).</p> <p>2. Puede escoger la opción que desea:</p> <p style="padding-left: 40px;">Editar (Prueba 1 paso 3.1).</p> <p style="padding-left: 40px;">Agregar nuevo (Prueba 1 paso 2).</p>
<p>Resultado esperado: Se borran los grupos de usuarios con todos sus datos y el sistema muestra un mensaje de confirmación satisfactoria.</p>
<p>No conformidades:</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 82: Prueba de aceptación Mostrar log de las sincronizaciones

Caso de Prueba de Aceptación	
Código: HU14_P1	HU: Mostrar log de las sincronizaciones
Nombre: Mostrar log de las sincronizaciones.	
Descripción: Prueba de funcionalidad para mostrar todos los log de todas las sincronizaciones.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción: Mostrar log de las sincronizaciones. 2. El usuario observa una lista todas las sincronizaciones dependiendo del grupo que el usuario autenticado se encuentre. 3. El usuario selecciona la opción Ver. 4. El usuario observa los datos de la sincronización escogida y además puede observar lo que está pasando en tiempo real con la sincronización en el servidor de réplica. 	
Resultado esperado: Se muestra un listado de las sincronizaciones que el usuario tiene acceso y para cada una de ellas el log en tiempo real.	
No conformidades:	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 83: Prueba de aceptación Autenticar usuario

Caso de Prueba de Aceptación	
Código: HU14_P1	HU: Autenticar usuario
Nombre: Autenticar usuario.	

Descripción: Prueba de funcionalidad para controlar el acceso a la aplicación.
Condiciones de ejecución: El usuario debe estar registrado en la base de datos.
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario introduce sus datos: usuario y contraseña. 3. Dependiendo al grupo al que el usuario pertenezca tiene permisos a interactuar con la aplicación.
Resultado esperado: Se autentica el usuario.
No conformidades: Se muestra un mensaje en inglés cuando el usuario o la contraseña son incorrectos, también cuando el usuario no se encuentra registrado en la aplicación.
Evaluación de la prueba: Prueba satisfactoria.

Anexo # 10: Prueba unitaria:

```

public function testShowSincronizacionesIndex() {
    $sessionTemp=$this->session;
    if (!$sessionTemp) {
        $sessionTemp=$this->getSession();
    }
    $sessionTemp->setBasicAuth("admin", "admin");
    $sessionTemp->visit($this->base.'/showTodas');

    $page=$sessionTemp->getPage();

    $this->assertTrue($page->hasContent('Inicio'));
}

```

Figura 6: Prueba unitaria: testShowSincronizacionesIndex

```

public function testLoginIndex() {
    $sessionTemp=$this->session;
    if (!$sessionTemp) {
        $sessionTemp=$this->getSession();
    }
    $sessionTemp->setBasicAuth("admin", "admin");
    $sessionTemp->visit($this->base.'/');

    $page=$sessionTemp->getPage();

    $this->assertTrue($page->hasContent('Inicio '));
}

```

Figura 7: Prueba unitaria: testLoginIndex

```

public function testAdminIndex() {
    $sessionTemp=$this->session;
    if (!$sessionTemp) {
        $sessionTemp=$this->getSession();
    }
    $sessionTemp->setBasicAuth("admin", "admin");
    $sessionTemp->visit($this->base.'/admin/dashboard');

    $page=$sessionTemp->getPage();

    $this->assertTrue($page->hasContent('Inicio '));
}

```

Figura 8: Prueba unitaria: testAdminIndex

```

root@IDENTIDAD Tesis_Replica # phpunit -c app/ -v
PHP Deprecated: Comments starting with '#' are deprecated in /etc/php5/cli/conf
.d/mcrypt.ini on line 1 in Unknown on line 0
PHPUnit 3.6.10 by Sebastian Bergmann.

```

```

Configuration read from /var/www/Tesis_Replica/app/phpunit.xml.dist

```

```

.
..

```

```

Time: 3 seconds, Memory: 22.25Mb

```

```

OK (3 tests, 3 assertions)

```

```

root@IDENTIDAD Tesis_Replica # █

```

Figura 9: Resultado de la prueba