

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA DE FORMACIÓN PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“COMPONENTE SOFTWARE PARA MEJORAR EL ACCESO A LAS
BASES DE DATOS DISTRIBUIDA, SOFTWARE MIDDLEWARE, 2016”**

Tesis presentado por : Bach. Abilio Cáceres Curo

Para optar el título profesional de : Ingeniero de Sistemas

Tipo de investigación : Observacional, Prospectiva y Transversal.

Área de investigación : Ingeniería de Software

Asesor : MSc. Ing. Efraín Porras Flores

Ayacucho, Junio del 2016

DEDICATORIA

A mis padres, quienes siempre me acompañan en los distintos momentos de mi vida, por todo el apoyo brindado y confiar en mí a pesar de tantas dificultades.

A mis hermanos, por haber fomentado en mí el deseo por el estudio y las ganas de romper barreras de superación frente a cualquier limitación.

A mis amigos que copartieron los años de aprendizaje, que confiaron en cada pequeño logro siempre en favor de la sociedad.

AGRADECIMIENTO

A la Universidad Nacional de San Cristóbal de Huamanga, mi alma máter, a todos los docentes que de alguna forma han contribuido en mi formación profesional.

CONTENIDO

	Pág.
DEDICATORIA.....	i
AGRADECIMIENTO	ii
CONTENIDO	iii
RESUMEN	v
INTRODUCCIÓN.....	vi

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1	DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA.....	1
1.2	FORMULACIÓN DEL PROBLEMA DE LA INVESTIGACIÓN.....	3
1.3	OBJETIVOS DE LA INVESTIGACIÓN	3
1.4	HIPÓTESIS DE LA INVESTIGACIÓN.....	4
1.5	JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN	4

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1	ANTECEDENTES DE LA INVESTIGACIÓN.....	6
2.2	MARCO TEÓRICO	7
2.2.1	COMPONENTE SOFTWARE	7
2.2.2	BASE DE DATOS DISTRIBUIDA	10
2.2.3	MÉTRICA DE EVALUACIÓN DE EFICIENCIA Y RENDIMIENTO.....	15
2.2.4	TÉCNICAS DE OPTIMIZACIÓN DE ACCESO A BASE DE DATOS	17
2.2.5	ARQUITECTURA ORIENTADA A SERVICIO.....	17
2.2.6	SOFTWARE MIDDLEWARE	19
2.2.7	MARCO DE TRABAJO SCRUM	20
2.2.8	LENGUAJE DE MODELADO UML.....	26
2.2.9	SISTEMA GESTOR DE BASE DE DATOS	28
2.2.10	LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS	29
2.2.11	TECNOLOGÍAS DE INTERNET	29
2.2.11	POBLACIÓN	32
2.2.11	MUESTRA	32
2.2.11	MUESTREO POR CONVENIENCIA.....	32

CAPÍTULO III
METODOLOGÍA DE LA INVESTIGACIÓN

3.1	TIPO Y NIVEL DE LA INVESTIGACIÓN.....	34
3.2	DISEÑO DE LA INVESTIGACIÓN.....	35
3.3	POBLACIÓN Y MUESTRA	36
3.4	VARIABLES E INDICADORES	36
3.5	TÉCNICAS E INSTRUMENTOS PARA RECOLECTAR INFORMACIÓN.....	38
3.5.1	TÉCNICAS PARA RECOLECTAR INFORMACIÓN.....	38
3.5.2	INSTRUMENTOS PARA RECOLECTAR INFORMACIÓN.....	38
3.5.3	HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN ..	39
3.5.4	TÉCNICAS PARA APLICAR SCRUM.....	40

CAPÍTULO IV
ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

4.1.	RESULTADOS	41
4.1.2.	ARTEFACTOS SCRUM	42
4.1.3.	ITERACIÓN I: MODELO ARQUITECTÓNICO Y CONFIGURACIÓN DEL ENTORNO DEL PROYECTO	45
4.1.4.	ITERACIÓN II: DESARROLLO DEL COMPONENTE SOFTWARE MIDDLEWARE SQL	57
4.1.5.	ITERACIÓN III: MEDICIÓN Y FORMALIZACIÓN DE TABLAS DE EFICIENCIA Y RENDIMIENTO.....	121
4.2.	ANÁLISIS ESTADÍSTICO	131
4.3.	DISCUSIÓN.....	145

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1	CONCLUSIONES.....	149
5.1	RECOMENDACIONES	149
	REFERENCIA BIBLIOGRÁFICA.....	152
	ANEXO A.....	158
	ANEXO B.....	160
	ANEXO C.....	161

RESUMEN

En el entorno actual, el desarrollo de software en los negocios y el manejo de datos es un punto crucial para la toma de decisiones y los accesos a los datos son a fuentes de datos homogéneos o heterogéneos en federación, haciendo que incremente la complejidad de desarrollo debido a las diversas limitaciones que se presenta en cada tecnología de base de datos que mantiene su propia denotación en el manejo de optimización y manipulación de datos de forma homogénea.

El objetivo de este trabajo de investigación es desarrollar un componente software denominado Middleware SQL, para mejorar el acceso a las bases de datos distribuida de forma transparente y homogénea. El tipo de investigación es observacional, prospectiva y transversal

De acuerdo al capítulo IV se obtuvo los artefactos de la pila del proyecto en la tabla N° 4.1, modelo arquitectónico de la Figura N° 4.7 y 4.8, los requisitos funcionales de la tabla N° 4.9 y 4.18, Diagrama de clases de la figura N° 4.11 y 4.24, el diagrama relacional de base de datos de la figura N° 4.12, las tablas, de las medidas de los indicadores de eficiencia N° 4.21 y 4.22 utilizando el proceso Scrum integrado con la notación UML, para generar artefactos que a través de 3 iteraciones permita una correcta abstracción para poder finalmente obtener el Middleware SQL, un método optimizado para mejorar el acceso a las bases de datos distribuidos de forma homogénea en entornos distribuidos y heterogéneos. Ccontrastados, inferidos y demostrados en las secciones 4.2, Análisis estadístico y la sección 4.3 discusión de la investigación. En conclusión el Middleware SQL desarrollado en comparación con Hibernate, ORM más óptimo, tiene menor tiempo de acceso a las bases de dato distribuidas, resuelve el problema de incompatibilidad de tipos de datos, y brinda la homogeneidad en múltiples plataformas de sistemas bases de datos distribuida.

Palabras claves: Componente Software, Base de Datos Distribuida, Sistema Distribuido, SOA, Middleware, Scrum, Java, Base de Datos Relacional.

INTRODUCCIÓN

La comunicación entre bases de datos heterogéneas, resalta una problemática muy común en los sistemas de bases de datos distribuidos, en la que manipular los datos de forma homogénea sin importar su procedencia tiene alta criticidad. Según Velasco (2008), el componente software “se ubica en la capa intermedia de la arquitectura de aplicaciones Cliente/Servidor, proporcionando una forma única de comunicación e interacción entre aplicaciones destinadas para el usuario final con los diversos sistemas administradores de base de datos a través de en middleware, proporcionando un soporte tecnológico autónomo y homogéneo al acceso a las bases de datos distribuidas”, la misma que es definido por Álvarez (2012), como un conjunto de múltiples bases de datos lógicamente relacionadas, las cuales se encuentran distribuidas en diferentes espacios lógicos y geográficos e interconectados por una red de comunicaciones. Scrum está basada en un proceso constructivo iterativo e incremental donde las iteraciones tienen duración fija (Pham y Phuong, 2010).

Mi motivación, para desarrollar un componente software denominado Middleware SQL, es contar con un método alternativo para la creación de aplicaciones empresariales con alta criticidad en el acceso a datos en entornos distribuidos con autonomía e independencia de la plataforma de software y hardware.

Actualmente existen métodos, arquitecturas de software y patrones que permiten acceso a las bases de datos distribuidos, pero con diversas limitaciones como tipos de datos incompatibles, alta complejidad de procesamiento, carga excesiva en memoria del servidor, errores de tiempo de ejecución, excesiva cantidad de tiempo y costos de implementación.

Los principales objetivos son: a) Analizar y diseñar el modelo arquitectónico, con la finalidad de mejorar la eficiencia de acceso a las bases de datos distribuida. b) Analizar, diseñar, implementar y probar la capa de abstracción, con la finalidad de mejorar el acceso a las bases de datos distribuida. c) Implementar y probar las interfaces, con la finalidad de mejorar el acceso a las bases de datos distribuida.

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1 DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA

La información es una palabra común en todas las organizaciones actuales. El Diccionario de la Lengua Española (RAE, 2001) define la palabra Dato como: “Hechos conocidos que pueden registrarse y que tienen un significado implícito” y también como: “Antecedente necesario para llegar al conocimiento exacto de algo o para deducir las consecuencias legítimas de un hecho”. Los datos al ser analizados y cuestionados se les proveen de un significado que produce información y conocimiento, lo cual es necesario para la toma de óptimas decisiones a cualquier nivel y sobre cualquier situación que se presente. Actualmente, los datos o la descripción parcial o total sobre cualquier objeto de análisis, que tenga importancia en la vida cotidiana y laboral, son registrados, administrados y compartidos con otras personas u organizaciones por sistemas de cómputo. Esto hace que en el proceso de desarrollo de software, el correcto y eficiente funcionamiento de la bases de datos homogéneos o heterogéneos y por supuesto el acceso a éstas sea un aspecto vital.

El establecimiento de estándares computacionales como el Lenguaje Estructurado de Consultas que utilizan los Sistemas de Gestión de Bases de Datos en general, ha ayudado al intercambio de información de manera transparente para el usuario final y para las aplicaciones que lo utilizan. Sin embargo, toda herramienta de software es propensa a modificaciones, lo que hace que existan diferentes versiones de éste y que cada uno de los sistemas mencionados realice ampliaciones diversas al mismo, dando como resultado un conjunto de combinaciones entre estos elementos, lo que aumenta la incompatibilidad en los procesos de comunicación e intercambio de datos. El “tipo de dato” es quizá el concepto más importante que existe en el mundo de las bases de datos puesto que ellos son los responsables de asemejarse lo más fielmente posible a la realidad que se requiere almacenar dentro de un Sistema de Base de Datos. Sin embargo, cada fabricante de estas herramientas también tienen su propia perspectiva sobre cómo representarlos e interpretarlos y, aunque es válido, esta situación conlleva

a incompatibilidades y pérdida de significado en la transferencia de los mismos. Las grandes empresas o cualquier otra agrupación actualmente se ayudan de los Sistemas de Bases de Datos para la toma de decisiones de alto riesgo y al crecer éstas aumentan su complejidad, por lo que la distribución de todo ese conjunto de activos se hace indispensable; por ende, un Sistema de Base de Datos Distribuida es el concepto útil para tal fin, sin embargo, cuáles software de gestión y cómo integrarlos para dar la apariencia de un sistema único o componente centralizado, hace que esta elección deba de hacerse con cuidado y principalmente si se trata de un ambiente heterogéneo de cómputo.

Actualmente, existen componentes software llamados frameworks para la mayoría de los lenguajes de programación; su definición varía, pero la mayoría apunta a exponer que es una estructura de soporte con características genéricas, el cual puede ser utilizado en un proyecto de software de iguales características técnicas convirtiéndose el uso de los ORMs en una práctica necesaria y generalizada, con la manipulación de objetos entre la memoria y su ubicación real en la base de datos, permitiendo al desarrollador definir de forma clara la gestión entre el modelo de objetos y el esquema de base de datos, además de expresar las operaciones de ésta. Por otro lado, es crucial la forma correcta de manejar los datos haciendo que la persistencia no esté limitado a ninguna plataforma hardware o software concreta; caso que no sucede con la mayoría de los frameworks de persistencia de datos puesto que están ligadas completamente a un lenguaje de programación, no están completamente desacoplados de la lógica de negocio obligando que en tiempo de ejecución ayudado por el API realice la traducción de consultas a objetos al lenguaje SQL. No ofrece todas las funcionalidades que ofrece al utilizar JDBC en su forma nativa, las consultas cargadas a memoria para ser procesada hacen que se incremente de forma exponencial el consumo de memoria y el trabajo de procesamiento; la configuración está muy ligada a la tecnología de base de datos usada. Frente a esto, otros arquitectos de software optan la utilización de JDBC nativo, pero ello incrementa el tiempo de desarrollo, incrementa errores en tiempo de ejecución y a la vez incrementa las líneas de código haciendo tedioso el mantenimiento.

En el mundo de los negocios de hoy, las empresas necesitan tener una gran flexibilidad para poder adaptarse ágilmente a lo que un entorno muy exigente demanda, surgiendo

así incógnitas como: ¿Qué componente software utilizar?, ¿Cuál de los componentes software de persistencia brinda mayor rendimiento?, ¿Cómo reducir el trabajo físico y de procesamiento en el acceso a datos sin reducir la eficiencia?, ¿De qué manera armar una arquitectura para desacoplar la lógica de negocio de la persistencia a las bases de datos distribuidas? Esto dificulta, a su vez, la rápida adaptación de los procesos para poder aprovechar nuevas oportunidades de negocio o responder a las amenazas externas. Debido a estas interrogantes ha surgido una nueva forma de concebir los sistemas de información denominada Arquitectura Orientada a Servicios (SOA) que parece brindar la mejor respuesta de la que se dispone hasta ahora para dotar a las organizaciones de la flexibilidad requerida. En una SOA, la funcionalidad aplicativa se brinda a través de componentes denominados servicios, que presentan interfaces estándar bien definidas y que representan funciones de negocio. Los servicios se pueden combinar en lo que se denomina coreografía de servicios permitiendo implementar de ese modo procesos de negocio sumamente ágiles y flexibles.

1.2 FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN

PROBLEMA PRINCIPAL

¿De qué manera desarrollar un componente software para mejorar el acceso a las bases de datos distribuida, software middleware, 2016?

PROBLEMAS ESPECÍFICOS

- a. ¿Qué modelo arquitectónico diseñar para mejorar el acceso a las bases de datos distribuida?;
- b. ¿Cómo desarrollar la capa de abstracción para mejorar el acceso a las bases de datos distribuida?;
- c. ¿Cómo desarrollar las interfaces para mejorar el acceso a las bases de datos distribuida?

1.3 OBJETIVOS DE LA INVESTIGACIÓN

OBJETIVO GENERAL

Desarrollar un componente software mediante técnicas e instrumentos, el proceso Scrum, la notación UML, sistemas distribuidos, arquitectura orientada a servicios, lenguaje orientado a objetos, base de dato relacional y tecnologías de internet, con la

finalidad de mejorar el acceso a las bases de datos distribuidas, software middleware, 2016.

OBJETIVOS ESPECÍFICOS

- a. Analizar y diseñar un modelo arquitectónico, con la finalidad de mejorar el acceso a las bases de datos distribuida.
- b. Analizar, diseñar, implementar y probar la capa de abstracción, con la finalidad de mejorar el acceso a las bases de datos distribuida.
- c. Implementar y probar las interfaces, con la finalidad de mejorar el acceso a las bases de datos distribuida.

1.4 HIPÓTESIS DE LA INVESTIGACIÓN

Si desarrollamos un componente software basado en técnicas de optimización, entonces se mejora el acceso a las bases de datos distribuida, software middleware, 2016.

1.5 JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN

1.5.1 IMPORTANCIA DEL TEMA

IMPORTANCIA TÉCNICA

Al contar con el componente software se podrá tener una alternativa para ser usada en cualquier proyecto de software, simplificando el esfuerzo de procesamiento sin quitarle eficiencia en el acceso a las bases de datos distribuidas homogéneas o heterogéneas, transparencia y compatibilidad de los tipos de dato, dotando de mayor flexibilidad en la adaptación a entornos de negocios cambiantes y heterogéneos. Así mismo, se dispone de un método de desarrollo para los estudiantes de ingeniería informática, sistemas, para los desarrolladores de software, arquitectos de software e investigadores.

IMPORTANCIA ECONÓMICA

La implementación de este componente de software permitirá reducir el esfuerzo físico de implementación en nuevos softwares que cuenten con acceso a base de datos distribuida, de esta manera se procederá a reducir el tiempo y costo en la implementación.

1.5.2 JUSTIFICACIÓN

Existe una variedad de técnicas y métodos que permiten acceder a bases de datos distribuidas, pero ninguno de estos componentes software permite tratar de forma modular el acceso al llamado Sistemas de base de datos en federación o distribuidas. Además Orfali et al. (2002), en su libro “Cliente/Servidor. Guía de supervivencia”, determinan que con un middleware o interfaz específico para base de datos se puede tener una solución autónoma, configurable e independiente de hardware y software.

Por lo tanto, es necesario contar con un componente software implementado bajo un modelo arquitectónico, unos niveles de abstracción organizada en capas de abstracción, conjunto de interfaces por medio de las cuales una aplicación cliente invoca un servicio que transmita la solicitud por la red de computadoras hasta llegar a los servidores que soportan a la base de datos distribuida para obtener la información que se requiere o manipular los datos que la componen previa configuración para mantener su autonomía.

1.5.3 DELIMITACIÓN

La investigación se realizará sobre los componentes software, se estudiarán los componentes software de acceso a base de datos distribuida basado en técnicas de optimización, software middleware y la arquitectura orientada a servicios. Los componentes software de acceso serán frameworks (hibernate), jdbc, middleware sql; los sistemas operativos que se usarán serán Windows 7 y Ubuntu 15.10; las bases de datos serán: Sql Server 2012, Oracle 11g, PostgreSQL 9.0.1; también se usarán indicadores para medir la eficiencia de acceso.

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

Según la investigación de Callejas et al. (2011), concluyen que en la preocupación de la selección correcta del uso de un determinado componente software de persistencia, previamente se deben realizar distintas comparaciones entre éstos. Los aspectos comúnmente evaluados son los que tienen que ver con cuál tiene un mayor rendimiento en cuanto a uso de memoria, tiempo de ejecución, entre otros. De igual manera es necesario tener en cuenta aspectos como throughput o volumen de trabajo o de información que fluye a través del sistema, así como el tiempo de respuesta, el costo por transacción y medidas de utilización de recursos, para evaluar el rendimiento de una aplicación.

Según Velasco (2008), en su investigación interfaz común SQL, afirma que las especificaciones técnicas que conforman a un componente denominado Interfaz Común SQL se ubica en la capa intermedia de la arquitectura de aplicaciones Cliente/Servidor, proporcionando una forma única de comunicación e interacción entre aplicaciones destinadas para el usuario final con diversos Sistemas para la Administración de Base de Datos y proporcionan el soporte tecnológico heterogéneo de una base de datos distribuida. Para lograrlo se basó en Ingeniería de Software Distribuido, Lenguaje Unificado de Modelado, la Programación Orientada a Objetos y el Lenguaje de Programación Java, con los que realizó y documentó este arquetipo o modelo. Con estas herramientas se logró controlar la estructura estática y el comportamiento dinámico para obtener información y manipular los datos de manera remota; además hizo posible probar el desenvolvimiento de este elemento en ambientes heterogéneos para la creación de nuevas aplicaciones basadas en esta tecnología, de esta forma se consiguió sistemas compatibles, integrados y uniformes.

Según la investigación Ramírez (2010), concluye que las aplicaciones muy robustas para acceder a los datos sin importar que se encuentren almacenados en una base de

datos local, en una base de datos distribuida o si los datos se encuentran dispersos y dependen en gran medida de los dispositivos o componentes que lo forman (Hardware), se subsana con un sistema informático donde se simulen los procesos de acceso a los datos mediante un modelo basado en entidades informáticas para garantizar una respuesta optimizada al acceso concurrente de forma segura y logrando que los procesos de transacción sean rápidos – entendiéndose que “Entidad Informática”, en la presente tesis, se define como un objeto cuya estructura permite la gestión a un conjunto de servicios para realizar transacciones de información en una base de datos desde cualquier lugar y desde cualquier dispositivo.

Además según la investigación de López (2010), concluye que la implementación de una Base de Datos Distribuidas mejora el rendimiento y la accesibilidad de los datos, esto debido a una disminución de tráfico de la red y a un trabajo colaborativo de varios servidores como si fuera uno solo, ya que puede distribuir y replicar en diferentes nodos locales, tienen capacidades de almacenamiento y procesamiento limitado, pero tienen la ventaja de poder integrarse a otros nodos remotos con mejores recursos por medio de una red de comunicación con el objetivo de centrar la lógica de negocios sin importar las fuentes de datos a través del Modelo MVC y ADF Business Components.

2.2 MARCO TEÓRICO

2.2.1 COMPONENTE SOFTWARE

Según Beneken et al. (2003), el término componente de software se utiliza para denotar cosas diversas. En general, un componente es una entidad reutilizable con una interfaz bien definida y un comportamiento específico, pensado para combinarse con otros componentes y formar un sistema software (una aplicación). Un componente puede tener múltiples interfaces: algunas para proporcionar los servicios a los clientes del componente y otras para requerir servicios del entorno. Los componentes se pueden agrupar formando jerarquías. Un componente de alto nivel puede estar integrado por varios componentes interconectados mutuamente, o por varios subcomponentes cooperativos.

Según Sommerville (2005), es una unidad de software independiente que puede estar compuesta por otros componentes y que se utiliza para crear un sistema software.

Un elemento software que se ajusta a un modelo de componentes, que puede ser desplegado y compuesto de forma independiente sin modificación, según un estándar de composición (Councill y Heineman, 2001).

Un componente software es una unidad de composición de interfaces especificadas contractualmente y dependencias del contexto explícitas únicamente. Un componente software puede ser desplegado de forma independiente y está sujeto a la composición por terceras partes (Szyperski, 2002).

A. MODELO ARQUITECTÓNICO

Según Pressman (2011), es un modelo de software. Es la descripción y esquematización de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Es un patrón arquitectónico que expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones logrando un nivel de abstracción mayor.

Según Jacobson, Booch y Rumbaugh (2006), un arquetipo es un patrón o prototipo que representa una abstracción fundamental de importancia crítica para el diseño de una arquitectura para el sistema objetivo. Un modelo que captura los tipos más importantes de objetos en el contexto del sistema.

Según Sommerville (2005), es el establecimiento de un marco de trabajo estructural básico para un sistema, dicha estructura ofrece una integridad conceptual al sistema. Está compuesta por la estructura jerárquica de los componentes, la manera en la que dichos componentes interactúan y la estructura de datos que es utilizada por dichos componentes; dicho modelo abstraído proporciona de forma clara una idea de lo que es el software.

B. CAPA DE ABSTRACCIÓN

Según Pressman (2005), la capa de abstracción consiste en aislar un elemento de su contexto o del resto de los elementos que lo acompañan; es el nivel de abstracción de cada uno de ellos, separando en estructuras organizacionales o capas. Es una medida de la interconexión entre los módulos de la estructura de un programa y

depende de la complejidad del diseño arquitectónico de los módulos, haciendo posible conseguir el menor nivel posible de acoplamiento, y las conexiones sencillas entre los módulos hacen que el software sea más fácil de entender.

Según Jacobson, Booch y Rumbaugh (2000), el nivel de abstracción o también llamado capa de abstracción, es el proceso mediante el cual se identifican los aspectos importantes de un fenómeno ignorando sus detalles, de esta forma el diseñador puede reducir un problema a interacciones entre un grupo de entidades con comportamientos establecidos para cumplir con los objetivos iniciales. Estas entidades son a su vez nuevos problemas, aunque claro, de menores dimensiones que el original. Algunos de ellos posiblemente pueden resolverse en forma directa al tener una complejidad accesible al entendimiento humano; mientras que otras quizás requieran su descomposición en nuevas abstracciones, en un proceso iterativo mediante el que establecemos nuevos niveles de abstracción para concentrarnos en los aspectos del problema que nos interesa en un momento particular.

La arquitectura de software, son un conjunto de servicios especializados que pueden ser accesibles por múltiples clientes y que deben ser fácilmente reutilizable; para lograr dicho objetivo debe organizarse en niveles de abstracción (Rosenberg y Stephens, 2007).

C. INTERFAZ

Según Balek (2002), las interfaces son los medios para interconectar componentes, además de especificar los servicios que proporciona el componente; la interfaz también especifica los servicios que requiere de otros componentes. Esta característica es una diferencia fundamental entre los componentes y las bibliotecas tradicionales o los objetos ordinarios, donde estos aspectos se ocultan generalmente en su implementación. Por esta razón, las conexiones entre los componentes son más simétricas que las conexiones entre los objetos.

Según Rosenberg y Scott (2005), a las clases abstractas puras, es decir, a las clases que no contienen ninguna implementación, se les llama interfaces. En UML, una interfaz es una colección de operaciones que sirven para especificar los servicios de una clase o un componente. Una interfaz sólo contiene las cabeceras de las operaciones, más no

su implementación.

Cheng (2010), afirma que la interfaz del servicio debe ser abstracta u oculta, y de alguna manera especifican a las organizaciones su trabajo actual. Esto hace más sencillo cambiar, actualizar o cambiar la implementación sin romper la interoperabilidad desde la interfaz, que puede permanecer tal cual está. Estas interfaces son declarativas y de fácil entendimiento.

2.2.2 BASE DE DATOS DISTRIBUIDA

Según Álvarez (2012) una base de datos distribuida (BDD), es un conjunto de múltiples bases de datos lógicamente relacionadas, las cuales se encuentran distribuidas en diferentes espacios lógicos y geográficos e interconectados por una red de comunicaciones. Dichas BDD tienen la capacidad de realizar procesamiento autónomo, esto permite realizar operaciones locales o distribuidas. Las bases de datos distribuida (BDD), es un sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones, de tal forma que un usuario en cualquier sitio puede acceder a los datos en cualquier parte de la red, exactamente como si éstos fueran accedidos de forma local.

La base de datos distribuida en realidad es un tipo de base de datos “virtual” cuyas partes que la componen están almacenadas en varias bases de datos “reales” distintas que se encuentran en varios sitios distintos; de hecho, es la unión lógica de esas bases de datos reales; cada sitio tiene sus propias bases de datos reales, sus propios usuarios locales, su propio DBMS local y software de administración de transacciones; incluyendo su propio software local para bloqueo, registro de bitácora, recuperación, etcétera; así como su propio administrador de DB/DC local. En particular, un usuario determinado puede realizar operaciones sobre los datos desde su propio sitio local, tal como si ese sitio no participara nunca en el sistema distribuido (Date, 2001).

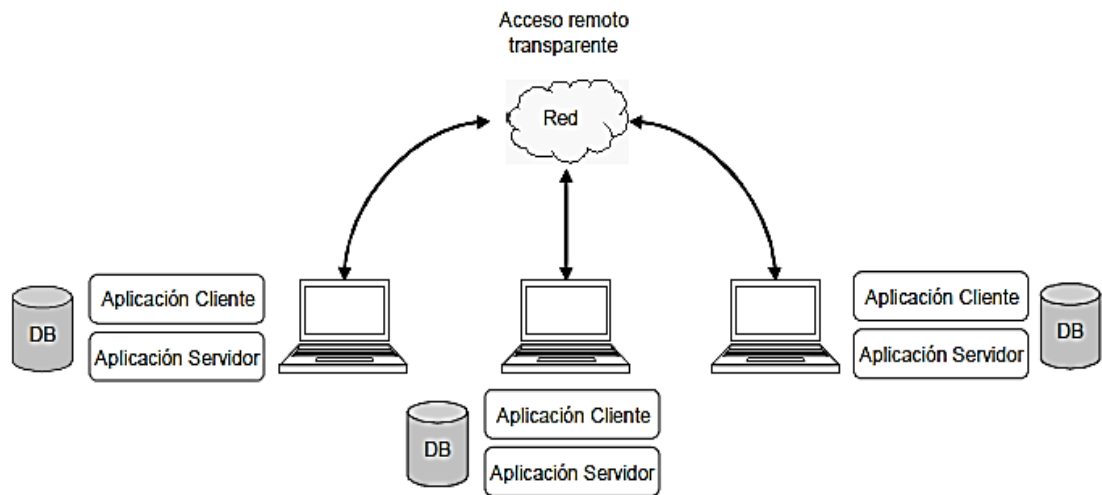


Figura N° 1.1. Operación de una computadora como Cliente/Servidor (Date, 2001).

A. SISTEMA DISTRIBUIDO

Un sistema es un conjunto de elementos interrelacionados entre sí, que a través de la aplicación de ciertos procesos predefinidos a un conjunto de datos del entorno o ambiente en el que se desenvuelve, persigue lograr objetivos establecidos con anterioridad. Estos procesos son llevados a cabo por los llamados “subsistemas”, elementos o sistemas más pequeños que no interfieren para funcionar entre sí, pero que el resultado de cada uno de ellos sirve para generar u obtener el resultado que el sistema en general necesita. Los “Sistemas de Cómputo”, son un conjunto de subsistemas, llamados hardware y software, que trabajan juntos para lograr los objetivos para los cuales fueron diseñados (Kendall y Kendall, 1997).

Según Silberschatz et al. (1998), dependiendo del contexto en el que se mencionen, es una colección de computadoras separadas físicamente y conectadas entre sí por una red de comunicaciones; cada máquina posee sus componentes de hardware y software que el programador percibe como un solo sistema.

a. SISTEMAS OPERATIVOS

Un Sistema Operativo es un conjunto de programas que efectúan la gestión de los procesos básicos de una computadora y permite la normal ejecución del resto de las operaciones. Es una capa compleja entre el hardware y el usuario que le facilita al usuario las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndolo de los complicados procesos necesarios para llevarlos a cabo. Por

ejemplo, un usuario normal simplemente abre los archivos grabados en un disco sin preocuparse por la disposición de los bits en el medio físico de almacenamiento como los disquetes, CDROM, discos duros, tarjetas de memoria portátil, etcétera. Este elemento es un software responsable del control o administración del hardware. También recibe el nombre de núcleo, conocido en términos informáticos como kernel (Tanenbaum, 1997).

Las RPC funcionan cuando un proceso Cliente llama a una función en un Servidor remoto y se detiene hasta que recibe los resultados. Los parámetros se transmiten como en cualquier procedimiento ordinario y el proceso es sincrónico. El proceso que emite la llamada espera hasta que obtiene los resultados. En forma oculta, el software de RPC de tiempo de ejecución reúne los valores para los parámetros, forma un mensaje y lo envía al Servidor remoto. El Servidor recibe la solicitud, desempaqueta los parámetros, llama al procedimiento y envía la respuesta al Cliente (Pressman, 2005).

b. APLICACIONES CLIENTE/SERVIDOR

La tecnología Cliente/Servidor engloba entonces los elementos siguientes: una red de computadoras, un protocolo de comunicación, varias computadoras y los sistemas operativos de red correspondientes a cada una de ellas; todos ellos son elementos que deben ser correctamente elegidos para no invertir recursos en unir piezas que sean incompatibles entre sí. Cliente/Servidor se refiere a una arquitectura o división de responsabilidades en las aplicaciones distribuidas de software (Pressman, 2011).

Según Pressman (2005), Cliente/Servidor es un tipo especial de Sistemas Distribuidos. El Cliente es la aplicación (a quién se le conoce como parte frontal o front-end) y el Servidor (back – end) que puede ser de diferentes tipos dependiendo del área de interés del constructor del sistema distribuido.

c. PROTÓCOLOS DE COMUNICACIÓN

Según Licesio y Rodríguez (2013), es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud

física.

d. RED DE COMPUTADORAS

Es un conjunto de equipos informáticos y software conectados entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios (Groth y Skandier, 2005).

Según Stallings (2004), es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre computadoras.

B. BASE DE DATOS RELACIONAL

Una base de datos es un conjunto de datos relacionados entre sí, datos que representan algún aspecto del mundo real, que pueden registrarse de manera lógica y coherente para que tengan un significado implícito y un propósito específico para un grupo de usuarios que la necesite (Elmasri y Navathe, 2000).

Osorio (2008), asevera que el usuario de un sistema de base de datos puede realizar consultas de tablas, inserción de nuevas tupas, actualizaciones o borrado de las existentes. Estas operaciones se realizan mediante unos lenguajes conocidos como lenguajes de consulta relacional.

“La existencia de los lenguajes denominados álgebra relacional y cálculo relacional son la base del éxito comercial de los SGBDR (Sistemas Gestores de Base de Datos Relaciones)” (Cobo, s.f., p.49).

Los principales conceptos sobre el cual se basan son:

a. LENGUAJE SQL

Según Eisenberg et al. (2003), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas.

SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones (ISO/IEC, 2005).

b. TIPO DE DATO

Según Cardelli (1985), un tipo de dato informático o simplemente tipo es un atributo de los datos que indica al ordenador (y/o al programador) sobre la clase de datos que se va a trabajar. En esencia, un espacio en memoria con restricciones sobre el cual se puede operar para un fin.

c. PROCESAMIENTO DE CONSULTAS SQL

Según Silberschatz, Korth y Sudarshan (2002), es una serie de actividades implicadas en la extracción y acceso de datos en una base de datos. Estas actividades incluyen la traducción de consultas expresadas en lenguajes de bases de datos de alto nivel en expresiones implementadas en el nivel físico del sistema, así como transformaciones de optimización de consultas y la evaluación real de las mismas. El costo del procesamiento de una consulta está determinado por la cantidad de veces que se accede al disco duro de la computadora, comparativamente más lento que el acceso a la memoria principal. Normalmente hay muchas estrategias posibles para procesar una consulta dada, especialmente si la consulta es compleja. Por esta razón, vale la pena que el sistema gaste una cantidad sustancial de tiempo en la selección de una buena estrategia para el procesamiento de la consulta, aunque ésta se ejecute solamente una vez.

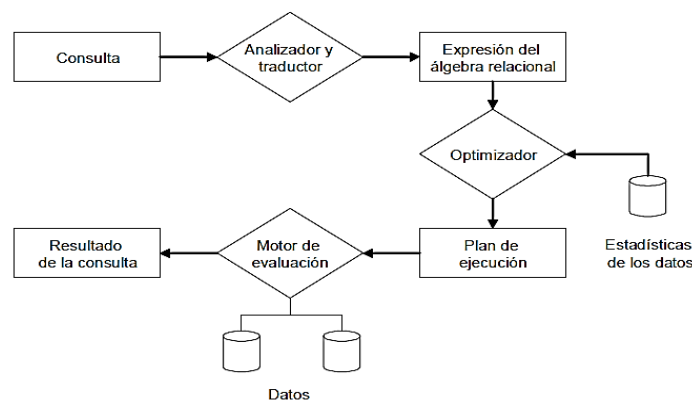


Figura N° 1.2: Procesamiento de una consulta SQL (Silberschatz, 2002).

En las instrucciones SELECT, INSERT, UPDATE y DELETE de Transact-SQL, se puede hacer referencia directa a las tablas y vistas de orígenes de datos externos evaluando la expresión algebraica relacional según (Silberschatz, 1998).

d. SERVIDOR DE BASE DE DATOS

Un Servidor de bases de datos SQL administra el control y ejecución de comandos SQL. Proporciona las vistas lógica y física de los datos además de generar planes de acceso optimizado para la ejecución de los comandos de SQL. Además, la mayoría de los servidores de bases de datos proporcionan características y utilerías de administración del servidor que facilitan el manejo de los datos. Mantiene tablas dinámicas de catálogos que contienen información sobre los objetos SQL alojados en él. Administra los aspectos de recuperación, concurrencia, seguridad y consistencia de una base de datos. Esto supone controlar la ejecución de una transacción y anular sus efectos en caso de fallas. Implica, asimismo, la obtención y entrega de candados durante la ejecución de una transacción y la protección de los objetos de la base de datos contra acceso no autorizado (ISO/IEC, 2005).

Este término servidor se usa para caracterizar a un DBMS cuando la aplicación se ejecuta físicamente en una computadora llamada cliente y cuando otra computadora llamada Servidor se encarga del almacenamiento y acceso a los datos (Pressman, 2005).

2.2.3 MÉTRICA DE EVALUACIÓN DE EFICIENCIA Y RENDIMIENTO

En términos generales de sistemas de información, el rendimiento es considerado como la «reducción de uso de la CPU, reducción del uso de la memoria heap y la búsqueda de métodos más óptimos para el sistema» (Nolte, 1995). De igual manera es necesario tener en cuenta aspectos como throughput o volumen de trabajo o de información que fluye a través del sistema, así como el tiempo de respuesta, el costo por transacción y medidas de utilización de recursos para evaluar el rendimiento de una aplicación (Throughput, 2010).

Este tipo de pruebas se enfocan en la determinación o validez de las características de rendimiento del sistema cuando se someten a condiciones extremas durante las operaciones de producción (Alonso et al., 2005).

Número	Métrica	,Unidad
1	Tiempo de acceso a los datos o tiempo de respuesta empleado en la ejecución de determinadas instrucciones.	Milisegundos
2	Líneas de código implementado.	Cantidad
3	Alto acoplamiento de lógica y acceso a datos.	s/n
4	Overhead o volumen de trabajo o de información que fluye a través del sistema.	s/n
5	Incompatibilidad de tipos de dato.	s/n
6	Multiplataforma y transparencia del componente.	s/n

Tabla N° 1.1: Métricas para evaluar la eficiencia al acceso a datos (Alonso et al., 2005).

Los ORMs se han convirtiéndose en una práctica necesaria en la manipulación de objetos en la memoria y su ubicación real en la base de datos, permitiendo al desarrollador definir de forma clara la gestión entre el modelo de objetos y el esquema de base de datos, sin incrementar su acoplamiento, además de expresar las operaciones de esta última en términos de objetos (Richardson, 2009). Estos frameworks de persistencia buscan simplificar el proceso de desarrollo y las líneas de código implementado (Benjelloun, 2006).

Según Callejas et al. (2011), el framework Hibernate es robusto, capaz de soportar cantidades inmensas de instrucciones SQL, sin afectar su rendimiento frente a otros frameworks de persistencia.

Según Callejas et al. (2011), Hibernate carga los objetos relacionados reduciendo rendimiento en su carga inicial.

Como lo indica Tuya et al. (2007, 43), afirman que las pruebas para determinar si los tiempos de respuesta del sistema tanto en condiciones normales como en condiciones especiales, se encuentran dentro de los límites predefinidos; de forma similar Alonso (2005), afirma que las pruebas tienen como propósito evaluar el tiempo de respuesta del sistema o memoria heap que ocupa.

Por otro lado, Pressman (2010), argumenta que es inaceptable que un software proporcione las funciones requeridas pero no se ajuste a los requisitos de rendimiento, por esto la importancia de estas pruebas diseñadas para probar el rendimiento del

software en tiempo de ejecución dentro del contexto de un sistema integrado.

Existen inconvenientes en los tipos de datos de los diferentes DBMS a pesar de manejar el estándar SQL-92 como los campos numéricos fechas generando incompatibilidad de los sistemas (Velasco, 2010).

2.2.4 TÉCNICAS DE OPTIMIZACIÓN DE ACCESO A BASE DE DATOS

Según Pagani (2012), define que son técnicas que permiten mejorar el performance de las aplicaciones software, puesto que las consultas SQL como el resto del código de la aplicación influirán directamente en la performance del motor de base de datos. Si bien, es bastante engorroso analizar una aplicación que puede tener decenas de miles de líneas de código, es importante trabajar en este punto de forma tal de optimizar tanto como sea posible el acceso a la base de datos.

Número	Recomendaciones
1	Pool de conexiones.
2	Utilizar un caché entre la aplicación y el motor de base de datos.
3	Optimizar las consultas SQL mediante plan de ejecución.
4	Fragmentación de consultas.
5	Optimizar motores de base de datos.
6	Diseñar modelos de base de datos relacional optimizado.
7	Uso de índices de ordenamiento en base de datos.

Tabla N° 1.2: Recomendaciones de optimización para acceso a base de datos (Pagani, 2005).

Los aspectos comúnmente evaluados son los que tienen que ver con cual tiene un mayor rendimiento en cuanto a uso memoria, tiempo de ejecución, entre otros aspectos (Zyl et al., 2006).

2.2.5 ARQUITECTURA ORIENTADA A SERVICIO

Bieberstein (2008), define que SOA es un término que representa una nueva generación de una plataforma distribuida, como también involucra muchas cosas, incluyendo su propio diseño de paradigmas y principales diseños, patrones, un modelo de arquitectura distinta, conceptos relacionados y tecnologías.

Erl (2009), define que la computación orientada a servicios construye una plataforma distribuida y agrega nuevas capas, consideraciones de desarrollo y una vasta implementación de tecnologías, muchas de las cuales están basadas en el marco de desarrollo de los Web Services.

Gao (2007) , afirma que la orientación es un paradigma de diseño entendido para la creación de soluciones lógicas, unidades que son individualmente plasmadas, tal que colectivamente y repetidamente son utilizadas en el soporte de la realización de objetivos estratégicos específicos y los beneficios asociados con la orientación a servicio y la computación orientada a servicios.

Erl (2009), menciona que la solución lógica diseñada acorde con la orientación a servicios puede ser calificado con Orientación a servicios, y las unidades de servicio orientadas a referirse como servicio. Como un paradigma de diseño para la computación distribuida, la orientación a servicio puede ser comparado con la orientación a objetos. De hecho tiene muchas raíces en la orientación a objetos y también ha sido influenciado por la industria de desarrollo.

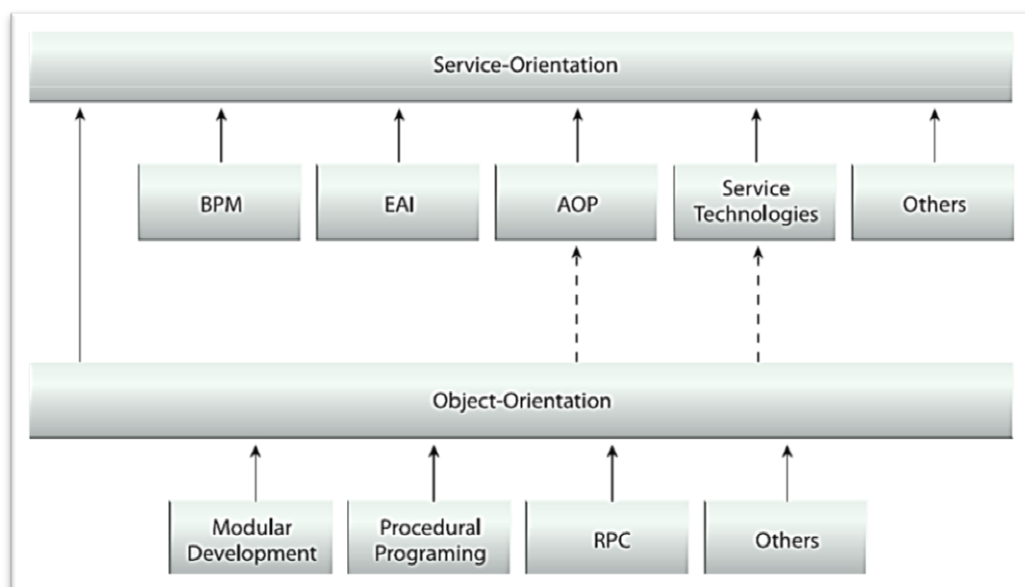


Figura 2.3: Evolución del diseño de paradigmas. (Erl, 2009).

Bieberstein (2008), afirma que la arquitectura orientada a servicios representa un modelo de Arquitectura que tiene como misión mejorar la agilidad y el costo efectivo

de una empresa mientras se reduce la carga de TI para toda la organización. SOA soporta la orientación a servicios en la realización de objetivos estratégicos, históricamente el término “Orientado a servicios” ha sido usado por los medios como casi el sinónimo con Computación Orientada a Servicios.

Dikmans et al. (2012), menciona que la implementación de SOA consiste en la combinación de tecnologías, productos, APIs, extensiones del soporte de infraestructura y varias otras partes. La puesta en práctica de la arquitectura orientada a servicios es única dentro de cada organización; sin embargo, es tipificado por la introducción de nuevas tecnologías y plataformas que especifican la creación, ejecución y evolución de soluciones orientados a servicios. Como resultado, construir una arquitectura alrededor de la arquitectura orientada a servicios, establece un entorno amigable para la solución lógica que ha sido diseñado con los principales diseños.

2.2.6 SOFTWARE MIDDLEWARE

Según la IEFT (2006), un Middleware puede ser vista como un conjunto de servicios y funciones reutilizables, expandibles, que son comúnmente utilizadas por muchas aplicaciones para funcionar bien dentro de un ambiente interconectado.

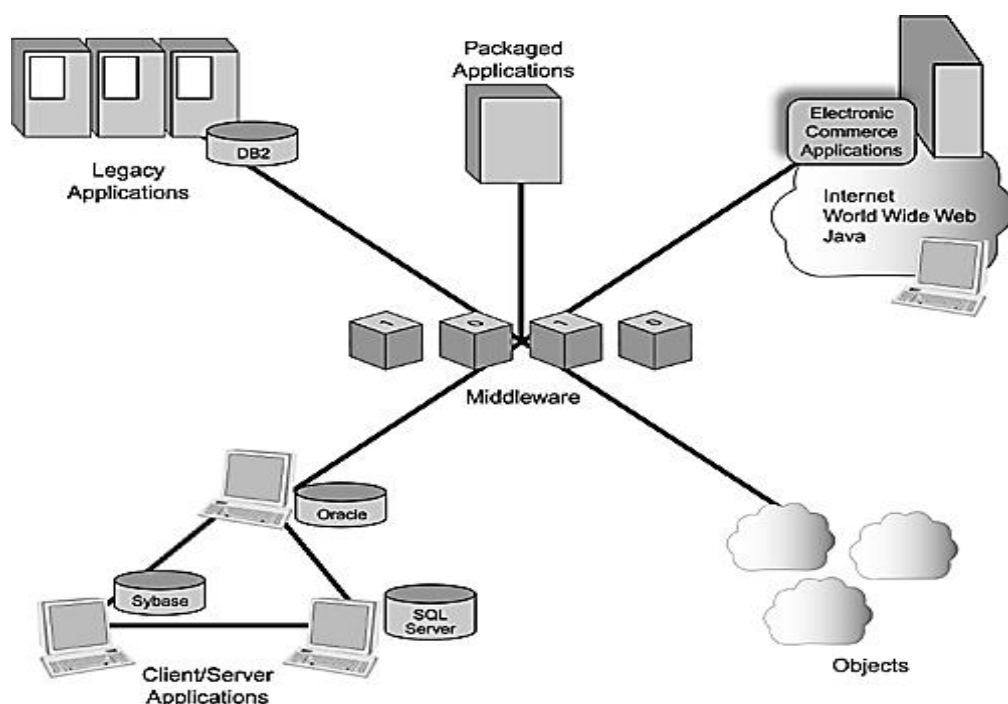


Figura 2.4: Esquema de funcionamiento de un middleware. (IEFT, 2006).

Es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones y sincronizaciones que son necesarias en los sistemas distribuidos (Bougettaya et al., 2006).

2.2.7 MARCO DE TRABAJO SCRUM

Según Palacio (2008), Scrum es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Reduce al máximo la burocracia y actividades no orientadas a producir software que funcionen y produce resultados en periodos muy breves de tiempo. Como método, Scrum enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, prácticas de desarrollo, implementación y demás cuestiones técnicas. Más bien delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles.

Scrum define un conjunto de prácticas y roles, que pueden tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Se basa en: el desarrollo de los requisitos del proyecto en bloques temporales cortos y fijos; la priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración; potenciación del equipo, que se compromete a entregar unos requisitos y para ello se le otorga la autoridad necesaria para organizar su trabajo; y la sistematización de la colaboración y la comunicación tanto entre el equipo y con el cliente (Münch, 2010).

Según Kent et al. (2001), Scrum se basa en los principios del manifiesto ágiles: privilegiar el valor de la gente sobre el valor de los procesos; entregar software funcional lo más pronto posible; predisposición y respuesta al cambio; fortalecer la comunicación y la colaboración; comunicación verbal directa entre los implicados en el proyecto; simplicidad; supresión de artefactos innecesarios en la gestión del proyecto.

Según Schwaber y Sutherland (2011), Scrum es un marco de trabajo de procesos que ha sido utilizado para gestionar el desarrollo de productos complejos desde principios

de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum hace patente la eficacia relativa de tus prácticas de gestión de producto y de desarrollo de modo que puedas mejorarlas.

A. VALORES SCRUM

Según Palacio (2008), los valores sobre el que se enfoca son:

a. FOCO

Los individuos y sus interacciones son más importantes que el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto de software (Palacio, 2008).

b. COMUNICACIÓN

Scrum pone en comunicación directa y continua a clientes y desarrolladores. El cliente se integra en el equipo para establecer prioridades y resolver dudas. De esta forma, ve el avance día a día y es posible ajustar la agenda y las funcionalidades de forma consecuente (Münch, 2010).

c. RESPETO

Scrum diferencia claramente entre dos grupos para garantizar que quienes tienen la responsabilidad tienen también la autoridad necesaria para poder lograr el éxito (cerdos), y que quienes no tienen la responsabilidad, los observadores externos (gallinas), no produzcan interferencias innecesarias (Pham, 2010).

d. CORAJE

El coraje implica saber tomar decisiones difíciles. Reparar un error cuando se detecta. Mejorar el código siempre que el feedback y las sucesivas iteraciones se manifiesten susceptibles de mejora (Münch, 2010).

B. PILARES SCRUM

Según Palacios (2010) los pilares sobre el que se basa Scrum es considera: la transferencia, inspección y adaptación.

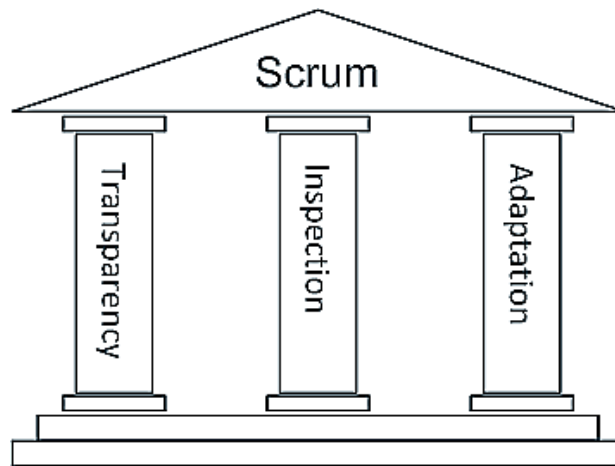


Figura N° 2.5: pilares de Scrum (palacio 2008).

C. PROCESO SCRUM

Según Pham y Phuong (2010), está basada en un proceso constructivo iterativo e incremental donde las iteraciones tienen duración fija.

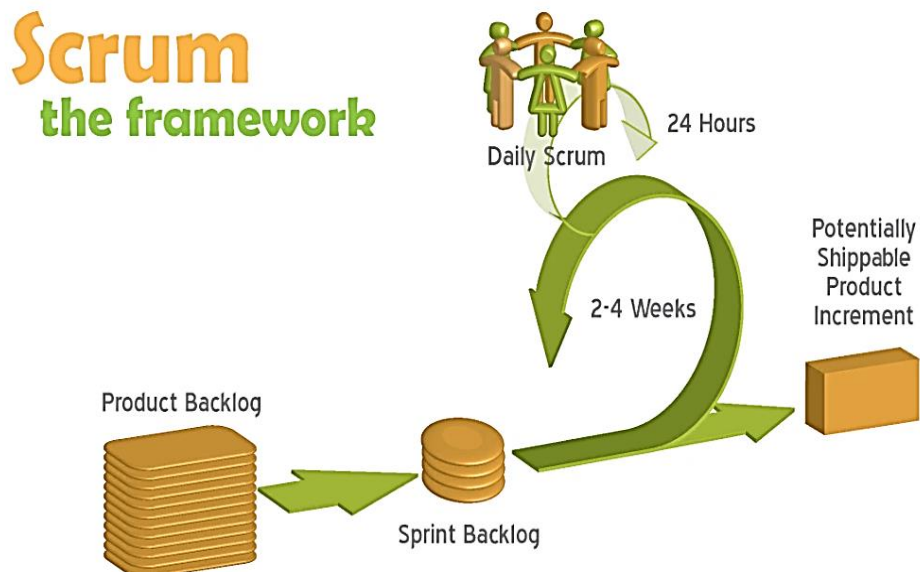


Figura N° 2.6: Proceso Scrum, Siglo Iterativo (Pham, 2010).

D. ROLES SCRUM

Según Pham (2010), los roles y responsabilidades que Scrum define en un proyecto son:

a. PROPIETARIO DEL PRODUCTO (PRODUCT OWNER)

Es el representante de todas las personas interesadas en el resultado del

proyecto. Se encarga de la definición de los objetivos del proyecto y colabora con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración (Palacios, 2008).

b. FACILITADOR (SCRUM MASTER)

Es el líder del equipo responsable de que todos los participantes sigan las reglas y los procesos de Scrum. Asegura la disponibilidad de las herramientas en cada iteración, facilita las reuniones de Scrum y está encargado de eliminar los impedimentos que el equipo tenga para continuar con su trabajo (Pham, 2010).

c. EQUIPO (TEAM)

Es el grupo de personas, responsable de transformar el Backlog de la iteración en un incremento de la funcionalidad del software. Tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos: Auto-gestionado, Auto-organizado, Multi-funcional (Palacios, 2010).

La dimensión del equipo total de Scrum no debería ser superior a veinte, el número ideal es diez, más o menos dos. Si hay más, lo más recomendable es formar varios equipos. No hay una técnica oficial para coordinar equipos múltiples, pero se han documentado experiencias de hasta 800 miembros, divididos en Scrums de Scrums, definiendo un equipo central que se encarga de la coordinación, las pruebas cruzadas y la rotación de los miembros (Münch, 2010).

d. USUARIO (USER)

Potencial usuario del sistema, se debe tomar en cuenta las características de los posibles usuarios al momento de desarrollar el aplicativo (Münch, 2010).

e. CLIENTE (CUSTUMER)

Grupo interesado en el desarrollo del proyecto del cual es parte el propietario del producto (Palacios, 2008).

f. ADMINISTRACIÓN (MANAGEMENT)

Encargado de suministrar los recursos necesarios para la realización del producto (Palacios, 2008).

E. ACTIVIDADES SCRUM

Según Pham (2010), las actividades son el listado de las tareas que un equipo debe llevar a cabo dentro de una iteración, puntualmente son:

a. REUNIÓN DE PLANIFICACIÓN DEL SPRINT (SPRING PLANNING MEETING)

Donde el cliente presenta al equipo la lista de requisitos, indicando la prioridad de cada uno de ellos. Además el equipo debe realizar una planificación de cómo se realizarán y en qué momento se abordarán cada uno de los requisitos planteados, estimando tiempo y asignando tareas a cada uno de los integrantes (Pham, 2010).

b. REUNIÓN DIARIA DE SCRUM (DAILY SCRUM)

Reunión a diario que se debe realizar para lograr la sincronización, cada miembro inspecciona el trabajo de los demás para poder hacer las adaptaciones necesarias, actualiza el estado de la pila de iteración (Sprint Backlog) y el gráfico de trabajo pendiente (Palacios, 2008).

c. REUNIÓN DE REVISIÓN DE LA ITERACIÓN (SPRINT REVIEW MEETING)

Reunión informal donde el equipo presenta al cliente los requisitos completados en la iteración, de modo que se pueda apreciar un esbozo del producto final. Aquí el cliente puede replantear los objetivos y prioridades para la siguiente iteración o para algún cambio dentro del proyecto final (Pham, 2010).

d. REPLANTEACIÓN DEL PROYECTO

Mientras se está desarrollando una iteración, el cliente puede empezar a realizar cambios a los requisitos priorizados a fin de enfocar de mejor manera la siguiente iteración del proyecto, modificaciones de la manera que era conveniente y cambiando incluso el contexto del proyecto si así lo requiere (Pham, 2010).

F. ARTEFACTOS SCRUM

Según Pham (2010), lo clasifica en:

a. PILA DEL PRODUCTO (PRODUCT BACKLOG)

Representa la visión y expectativas del cliente respecto a los objetivos y entregas del proyecto. El cliente es el responsable de crear y gestionar esta lista, y plasma en ella las expectativas de los resultados a obtener. Contiene los requisitos de alto nivel, el esfuerzo estimado en horas de trabajo para cada requisito, las posibles iteraciones, las entregas a realizar, y si es necesario que se considere dentro de cada requisito (Pham, 2010).

b. PILA DE LA ITERACIÓN (SPRINT BACKLOG)

Elaborada en la reunión de planificación del Sprint (Sprint Planning Meeting), es el plan para completar los requisitos seleccionados para la iteración y que se compromete a demostrar al cliente al finalizar la iteración, en forma de un entregable. Permite identificar las tareas de negocio y los problemas que se presentan a lo largo de una iteración (Palacios, 2010).

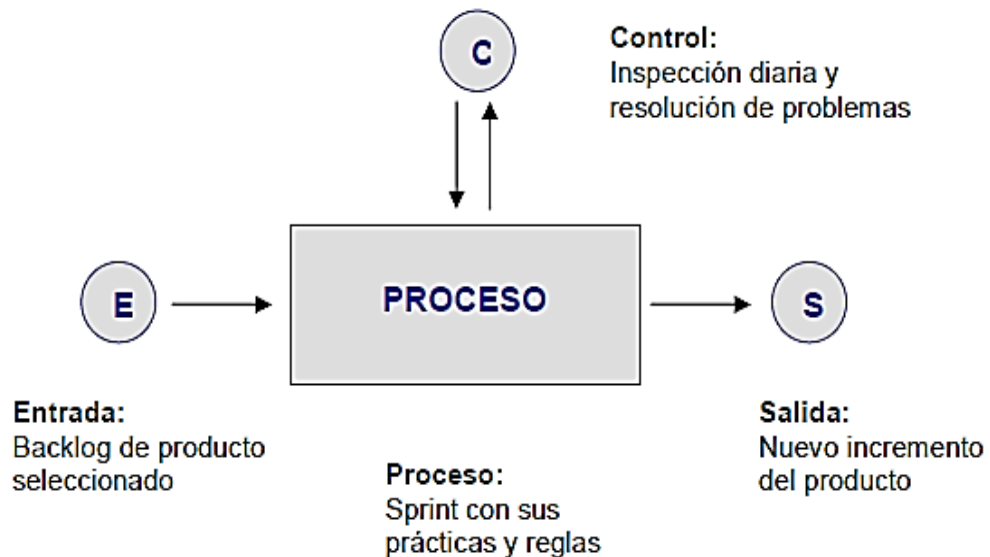


Figura N° 2.7: Flujo de iteración Scrum (Palacio, 2008).

c. GRÁFICOS DE TRABAJO PENDIENTE (BURNDOWN CHARTS)

Muestra la velocidad a la que se están completando los requisitos. Permite deducir si el tiempo es el requerido para la finalización del proyecto o iteración. Se puede representarlo en intervalos de días u horas pendientes para la culminación de las tareas de iteración (Münch, 2010).

2.2.8 LENGUAJE DE MODELADO UML

“El Lenguaje de Modelado Unificado (UML-Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s.UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos, de un lenguaje de modelado y de un proceso. El UML fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE”. (Booch et al., 1999).

Debrauwer y Van (2005), afirman que es un lenguaje semánticamente rico y que resulta bastante difícil retener todos los conceptos en los que se basa.

Para Fowler (1999), el lenguaje unificado es el sucesor de la oleada de métodos de análisis y diseño orientados a objetos (OOA&D) que surgió a finales de la década de 1980 y principios de la siguiente. Este lenguaje unifica los métodos Booch, Rumbaugh (OMT) y Jacobson, pero su alcance llegará a ser mucho más amplio.

García (2009), asevera que una forma de especificar, visualizar, construir y documentar sistemas de software es mediante el llamado Lenguaje Modelado Unificado (UML), que es usado para entender, diseñar, configurar, mantener y controlar información relacionada con éstos.

Para Marcos, Vela y Vara (2005), UML define las siguiente técnicas: a) Diagramas de clase, b) Diagramas de objetos, c) Diagrama de actividades, d) Diagrama de caso de uso, e) Diagrama de colaboración, f) Diagrama de componentes. g) Diagrama de despliegue, h) Diagrama de estado transiciones, finalmente i) Diagramas de secuencia.

ARTEFACTOS UML INTEGRADOS CON SCRUM

A. CASOS DE USO

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema (Sommerville, 2005).

Es una representación de una unidad discreta de trabajo realizada por un usuario (u otro sistema) usando el sistema en operación. Se ejecuta en su totalidad o no se ejecuta nada, devolviendo algo de valor al usuario. Un caso de uso define una secuencia de acciones que un sistema ejecuta, produciendo un resultado observable de valor para un usuario (actor) del sistema (Cockburn, 2006).

B. DIAGRAMA DE ACTIVIDAD

Un diagrama de actividades muestra un flujo de acciones; los nodos ejecutan un proceso, generalmente secuenciales, además presenta los resultados de dicha acción (Sommerville, 2005).

Larman (2010), sostiene que es una especialización de los Diagramas de Colaboración ya que contiene los mensajes y las reacciones de los objetos ante estímulos internos de sus pares. Este diagrama se usa para evitar que la documentación de las actividades del sistema se limite únicamente al texto informal de los casos de uso. Un caso de uso puede estar acompañado por uno o más Diagramas de Actividades.

C. ARQUITECTURA DE SOFTWARE

Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco (Kruchten y Philippe, 1995).

Una arquitectura de software se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real (Jacobson, Ivar, Booch, y Rumbaugh, 2000).

D. MODELO DE CLASES

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro (Jacobson, 1999).

Una clase es una descripción de un grupo de objetos con propiedades en común (atributos), comportamiento similar (operaciones), la misma manera de relacionarse entre objetos (asociaciones y agregados) y una semántica en común (Luza, 2010).

E. MODELO ENTIDAD RELACIÓN

Es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades (Sommerville, 2005).

Un diagrama de entidad-relación o DER es un modelo de red que describe la distribución de los datos almacenados en un sistema de forma abstracta. Es el "lenguaje" utilizado para crear diagramas de entidad-relación (Cockburn, 2006).

2.2.9 SISTEMA GESTOR DE BASE DE DATOS

Nevado (s.f.), un SGBD tiene como funciones principales: la descripción, manipulación y control, que en conjunto permiten: a) Definir los datos a los distintos niveles de abstracción (físico, lógico y externo), b) Manipular los datos en la base de datos, permitiendo insertar, modificar, borrar y consultar los datos, c) Mantener la integridad de la base de datos, y d) Controlar de la privacidad y seguridad de los datos en la base de datos.

Para Pons, Marín, Medina, Acid y Vila (2005), aseveran que hay dos tipos de tablas en un gestor de bases de datos relacionales: tablas de usuario que contienen los datos operativos y tablas del sistema que contienen los datos que describen la estructura de la base de datos (catálogo).

Osorio (2008), asevera que el usuario de un sistema de base de datos puede realizar

consultas de tablas, inserción de nuevas tupas o actualizaciones o borrado de las existentes. Estas operaciones se realizan mediante unos lenguajes conocidos como lenguajes de consulta relacional.

“La existencia de los lenguajes denominados álgebra relacional y cálculo relacional, son la base del éxito comercial de los SGBDR (Sistemas Gestores de Base de Datos Relaciones)” (Cobo, s.f., p.49).

2.2.10 LENGUAJE DE PROGRAMACIÓN ORIENTADA A OBJETOS

Weitzenfeld (s.f.), asevera que los lenguajes de programación orientados a objetos varían en sus estructuras y flujos de control. Y a pesar de que estos estén diseñados para un mismo tipo de programación, existen aspectos que hacen que ciertos lenguajes ofrezcan mejor apoyo que otros durante el desarrollo de un sistema de software.

“Es un conjunto de símbolos, palabras y reglas que permiten implementar un algoritmo en una computadora. Dicha implementación se conoce como programa y se escribe como un secuencia de frases del lenguaje de programación” (Osorio, s.f., p.368).

Los lenguajes de programación orientados a objetos ofrecen las ventajas potenciales de los códigos reutilizables, costos inferiores, menos pruebas y rapidez en la puesta en marcha. Los programadores pueden combinar, modificar e integrar módulos pre-desarrollados en un programa unificado. Entre los más importantes están: Smalltalk, C++ y Java (Stair y Reynolds, 1999, p.461).

Para Craing (2002), los lenguajes orientados a objetos están definidos por un conjunto de propiedades. La medida en la que un lenguaje particular satisfaga estas propiedades define en cuánto es un lenguaje orientado a objetos.

2.2.11 TECNOLOGÍAS DE INTERNET

Internet, también llamado autopista de información, designa un conjunto de redes informáticas relacionadas entre sí, para permitir que los usuarios puedan comunicarse entre sí; es una red abierta. Su principio básico es la trasmisión de datos

de manera fiable entre ordenadores (Colección Esencial, 2011).

Luján (2001) asevera que contrario a otros servicio online, que se controlan de forma centralizada, la Internet posee un diseño descentralizado. Dado que cada ordenador (host) en la Internet es independiente. Los operadores pueden elegir qué servicio usar y qué servicios locales proporcionar.

Cafassi (1998) asevera que Internet es una tecnología. Sin embargo no es la acepción más correcta y su consecuencia se ve a la hora de analizar las implicaciones sociales que conlleva su uso.

A. APLICACIÓN WEB

Para Luján (2001), una aplicación web es un tipo especial de aplicación cliente / servidor, donde el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones.

Para el Equipo Vértice (2010), una aplicación web es una aplicación informática que se utiliza accediendo a través de un sistema de red como puede ser internet o una intranet. La estructura común se basa en tres capas: Cliente, servidor y sistema de administración de base de datos.

B. PROTOCOLO

“Proviene del griego <<protókolon>>, que significa <<la primera hoja o tapa, encolada, de un manuscrito importante con notas sobre su contenido>>” (Antonio, 2001).

Según Gutiérrez y Tena (2003), un protocolo es un conjunto definido de etapas, que implica a dos o más partes, designado para realizar una tarea específica.

a. HTTP

Luján (2001), el protocolo forma parte de la familia de protocolos de comunicación TCP / IP, que son los empleados en internet. Permiten la conexión de sistemas heterogéneos, facilitando el intercambio de información ente distintos

ordenadores.

“El protocolo de transferencia de hipertexto es un sencillo protocolo cliente–servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP” (Romero, 1997, p. 203).

Para Kroenke (2003), hay dos características importantes en ese protocolo: a) Orientado a peticiones, pues los servidores HTTP esperan peticiones para realizar alguna acción y generar una respuesta y b) No mantienen un estado, ya que reciben una pregunta, la procesan y luego olvidan cuál cliente la formuló.

b. TCP

Según Colección Esencial (2011), este protocolo, descompone el mensaje en paquetes y asegura la fiabilidad de la transmisión. Cuando los paquetes llegan a su destino, se agrupan automáticamente para formar un único mensaje, idéntico al original.

TCP (Transmission Control Protocol), utiliza mensajes IP para lograr una transferencia de datos libre de errores. Ambos establecen un diálogo con otro sistema a base de enviar servicios de mensajes IP (Romero, 1997, p.22).

c. IP

En el protocolo de internet, cada equipo posee una sola dirección única en la red a la que pertenece y cada red posee una dirección única en internet. Por lo tanto, cada ordenador tiene asignado un nombre y un número IP, y en algunos casos un nombre de dominio (Colección Esencial, 2011).

“IP (Internet Protocol) es capaz de enviar mensajes de pequeño tamaño (denominado datagrama) entre dos ordenadores conectados en red. No ofrece garantías de que los mensajes alcancen su destino, debido a los posibles fallos de las redes de comunicaciones.” (Romero, 1997, p.20).

d. WEB SOCKET

WebSocket es una tecnología que proporciona un canal de comunicación

bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor. La API de WebSocket está siendo normalizada por el W3C, mientras que el protocolo WebSocket ya fue normalizado por la IETF como el RFC 6455 (IETF, 2012).

La especificación WebSocket define un API que establece conexiones "socket" entre un navegador web y un servidor. Dicho con otras palabras: existe una conexión persistente entre el cliente y el servidor, y ambas partes pueden empezar a enviar datos en cualquier momento (IANA, 2012).

2.2.12 POBLACIÓN

Para Chávez (2007), la población “es el universo de estudio de la investigación, sobre el cual se pretende generalizar los resultados, constituida por características o estratos que le permiten distinguir los sujetos, unos de otros”.

Según Tamayo y Tamayo, (1997), la población se define como la totalidad del fenómeno a estudiar donde las unidades de población posee una característica común la cual se estudia y da origen a los datos de la investigación.

2.2.13 MUESTRA

Respecto a esto, Bavaresco (2006), refiere que “cuando se hace difícil el estudio de toda la población, es necesario extraer una muestra, la cual no es más que un subconjunto de la población, con la que se va a trabajar”.

Según Tamayo y Tamayo (1997), afirma que la muestra es el grupo de individuos que se toma de la población, para estudiar un fenómeno estadístico.

2.2.14 MUESTREO POR CONVENIENCIA

También recibe el nombre de sesgado. El investigador selecciona los elementos que a su juicio son representativos, lo que exige un conocimiento previo de la población que se investiga (Tamayo y Tamayo, 1997).

Según Bavaresco (2006), los sujetos de una investigación específica, son

seleccionados para el estudio sólo porque son más fáciles de reclutar y el investigador no está considerando las características de inclusión de los sujetos que los hace representativos de toda la población.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1 TIPO Y NIVEL DE INVESTIGACIÓN

A. TIPO DE INVESTIGACIÓN

Según Hernández, Fernández y Baptista (2006), define la investigación observacional como aquella “investigación que se sustenta en el uso de técnicas que permiten al investigador adquirir información por medio de la observación directa y el registro de fenómenos, pero sin ejercer ninguna intervención”.

Según Gómez et al. (2005), concluyen que mediante la investigación observacional se intenta obtener información cuantitativa sobre una población utilizando diseños que controlen de modo externo las condiciones de producción de la conducta mediante la selección adecuada de las unidades de análisis y la sistematización de la recogida de información.

Según Bernal (2006), la investigación prospectiva “es aquella en los cuales la información se va registrando en la medida que va ocurriendo a propósito de la investigación el fenómeno o los hechos programados para observar”.

Según Hernández, Fernández y Baptista (2006), el tipo de investigación transaccional o transversal recolecta datos en un solo momento, en un tiempo único. Su propósito es describir variables, y analizar su incidencia e interrelación en un momento dado.

Según Hernández (2000), los estudios transaccionales nos presentan un panorama del estado de una o más variables en uno o más grupos de personas, o indicadores en un determinado momento.

En la presente investigación no pretendemos manipular las variables para tratar de producir resultados. En su lugar, observamos y comparamos los sujetos con un grupo de control cuyos datos se obtendrán y serán sometidos a métricas y estadísticas en un

momento determinado. De acuerdo a estas consideraciones la investigación es de tipo observacional, prospectivo y transversal.

B. NIVEL DE INVESTIGACIÓN

Según Hernández, Fernández y Baptista (2006), la investigación aplicada “es una investigación cuya finalidad es la resolución de problemas prácticos inmediatos en orden a transformar las condiciones del acto productivo y a mejorar la calidad del producto tomando como criterio el grado de abstracción del trabajo y para otros según el uso que se pretende dar al conocimiento”.

Para Hernández (2000), la investigación aplicada “busca la aplicación o utilización de los conocimientos que se adquieren. La investigación aplicada se encuentra estrechamente vinculada con la investigación básica, pues depende de los resultados y avances de esta última”.

La investigación aplicada se entiende como aquella que genera conocimientos o métodos dirigidos al sector productivo de bienes y servicios, ya sea con el fin de mejorarlo y hacerlo más eficiente, o con el fin de obtener productos nuevos y competitivos en dicho sector (Arias, 2006).

Se desarrollará un producto software; componente software para mejorar el acceso a las bases de datos distribuidas, software middleware, 2016; usando el marco de trabajo Scrum integrado con la notación UML. La utilización prácticas de los conocimientos implica buscar el conocer para hacer, para actuar, para comparar, para construir y para modificar, permitiendo lograr un producto o servicio (Zorrilla, 1993). Por estas consideraciones el nivel de la investigación es aplicada.

3.2 DISEÑO DE LA INVESTIGACIÓN

Según Hernández (2000), los estudios prospectivos transaccionales nos presentan un panorama del estado de una o más variables en uno o más grupos de personas, o indicadores en determinado momento único.

En este sentido, para Manterola (2014), son “diseños de investigación cuyo objetivo es "la observación y registro" de acontecimientos sin intervenir en el curso natural de

estos. Las mediciones, se pueden realizar a lo largo del tiempo (estudio longitudinal), ya sea de forma prospectiva o retrospectiva; o de forma única (estudio transversal)”.

En la investigación observamos, generamos los datos en un momento determinado de la investigación, medimos dichos valores en un único momento y sometemos a medidas a través de técnicas para concluir. En tal sentido, la presente investigación de acuerdo con los objetivos planteados, se ubicó el diseño no experimental, observacional y prospectivo transversal.

3.3 POBLACIÓN Y MUESTRA

POBLACIÓN.- La población de estudio analizada, estuvo compuesta los patrones de diseño e implementación para construir el componente software que permite el acceso a las bases de datos distribuidos en el año 2016.

MUESTRA.- Se tomó una muestra por conveniencia de los patrones de diseño e implementación para construir el componente software que permite el acceso a base de datos distribuida en el año 2016.

3.4 VARIABLES E INDICADORES

DEFINICIÓN CONCEPTUAL DE LAS VARIABLES

VARIABLE INDEPENDIENTE

Componente software.- Es una unidad constituida por interfaces reutilizables y comportamiento específico, es un modelo de componentes o arquitecturas que puede ser desplegado y compuesto de forma independiente sin modificación según un estándar para definir determinada estructura de software y funcionalidad.

INDICADORES DE LA VARIABLE INDEPENDIENTE

Modelo arquitectónico.- Modelo de resolución de problemas de ingeniería de software, es analizar, describir y esquematizar un prototipo de arquitectura para el desarrollo del sistema objetivo cuyo esquema abstraído visualiza la organización estructural y esencial para la construcción de un sistema software.

Capa de abstracción.- Proceso mediante el cual se identifica aspectos importantes de un fenómeno y se encapsula su detalle en niveles de abstracción o capas, de la misma

forma posibilita la especialización funcional y el bajo acoplamiento de acuerdo a algún estándar de calidad de software.

Interfaz.- Medio para interconectar componentes software haciendo posible la simetría, la conexión de los objetos o componentes y altamente escalable. Es la exposición declarativa de métodos o servicios de fácil entendimiento. En una interfaz, su uso es intuitivo y delega la implementación a clases especializadas dentro un determinado nivel de abstracción en ambientes cambiantes y heterogéneos.

VARIABLE DEPENDIENTE

Base de datos distribuida.- Conjunto de múltiples bases de datos lógicamente relacionadas, las cuales se encuentran en diferentes espacios lógicos y geográficos e interconectados por una red de comunicaciones, protocolos y sistemas operativos, posibilitando el procesamiento de forma autónoma y homogénea.

INDICADORES DE LA VARIABLE DEPENDIENTE

Sistemas distribuidos.- Conjunto de sistemas y subsistemas, llamados hardware y software que trabajan juntos para lograr los objetivos para los cuales fueron diseñados. Dicho sistemas están conectadas entre sí por una red de comunicaciones haciendo posible que el usuario lo perciba como un solo sistema.

Base de datos relacional.- Es un conjunto de datos relacionados entre sí; datos que representan algún aspecto del mundo real, que pueden registrarse de manera lógica y coherente para que tengan un significado implícito y un propósito específico. Así mismo, las bases de datos utilizan los sistemas gestores de base de datos para gestionar los planes de ejecución de procesamiento de consultas SQL, tempos, concurrencias, tipos de datos y sesiones de conexión.

DEFINICIÓN OPERACIONAL DE LAS VARIABLES

VARIABLE INDEPENDIENTE

X: Componente software

INDICADORES

X1: Modelo arquitectónico

X2: Capa de abstracción

X3: Interfaz

VARIABLE DEPENDIENTE

Y: Base de datos distribuida

INDICADORES

Y1: Sistema distribuido

Y2: Base de datos relacional

OPERACIONALIZACIÓN DE LAS VARIABLES

Se muestra en el Anexo A.

3.5 TÉCNICAS E INSTRUMENTOS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN

3.5.1 TÉCNICAS PARA RECOLECTAR INFORMACIÓN

Se consideró la técnica “Análisis Documental”, para documentar las arquitecturas y patrones de diseño e implementación para construir los componentes software que usen modelos arquitectónicos, capas de abstracción, interfaces, metodologías y tecnologías que permiten el acceso a las bases de datos distribuida.

Se considera la técnica “Encuesta a arquitectónicos de software”, para realizar la encuesta a los arquitectos de software que usan las arquitecturas y patrones de diseño e implementación para construir el componentes software que usen modelos arquitectónicos, capas de abstracción, interfaces, metodologías y tecnologías que permiten el acceso a las bases de datos distribuida.

3.5.2 INSTRUMENTOS PARA RECOLECTAR INFORMACIÓN

Se ha diseñado el instrumento “Registro de ficha”, para el estudio de arquitecturas y patrones de diseño e implementación que permita construir los componentes software que permitan identificar los tiempos de demora en la programación, tecnologías y errores en tiempo de ejecución que permiten el acceso a las bases de datos distribuidos, presentado en el Anexo B.

Se ha diseñado el instrumento “Cuestionario”, para la técnica “Encuesta a arquitectos de software”, presentado en el Anexo C. Muestra la información sobre las arquitecturas y patrones de diseño e implementación que permita construir el componente software que permita el acceso a base de datos distribuida.

3.5.3 HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN

Las herramientas tecnológicas que se utilizan son seleccionadas debido a la interoperabilidad que presenta los componentes software, patrón arquitectónico, arquitectura orientada a servicios, soporte continuo a base de datos distribuido homogéneos o heterogéneos; se propone la implementación de un componente software llamado (Middleware SQL), para la construcción de nuevas aplicaciones distribuidas. Por lo cual seleccionamos las tecnologías según la tabla 3.1.

NOMBRE	FABRICANTE	SERVICIO
Windows 7	Microsoft Corporación.	Sistema Operativo de Microsoft, línea de sistemas operativos con licencia producida por Microsoft Corporación.
Linux Ubuntu 15.10	Comunidad de Software Libre.	Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre bajo la licencia GPL.
PostgreSql	PostgreSQL Global.	PostgreSQL es un SGBD relacional y libre desarrollada por el Development Group (PGDG), publicado bajo la licencia BSD.
Microsoft Sql Server 2012	Microsoft Corporación.	SGBD, línea de sistemas gestor de base de datos con licencia producida por Microsoft Corporación.
Oracle 11g	Oracle Corporación.	SGBD, línea de sistemas gestor de base de datos con licencia Oracle Corporación.
JAVA 8	Desarrollado por Sun Microsystem, en la actualidad fue comprada por ORACLE.	Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystem, que cuanta con una máquina virtual encargada de interpretar y ejecutar los archivos para ser compilados en tiempo real.
Eclipse 4.5	Eclipse Foundation.	Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma.
Spring 4	Apache License 2.0.	Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma

		Java.
Primefaces	Primefaces Foundation.	PrimeFaces es un framework de componentes para JavaServer Faces (JSF) de código abierto.
Java Script	Netscape Communications Corp, Mozilla Foundation.	JavaScript, es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Estándar para genera JQuery, Ajax, Json, WebSocet.
Apache Tomcat 8	Apache Software Foundation.	Apache Tomcat, funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation.
Enterprise Architect	Grupo de Desarrollo EAPrincipals.	Enterprise Architect es una herramienta comprensible de diseño y análisis UML, que cubre el desarrollo de software desde la captura de requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento.

Tabla N° 3.1: Herramientas para el desarrollo del componente software (Middleware SQL).

3.5.4 TÉCNICAS PARA APLICAR SCRUM

Revisando el marco teórico en el capítulo II, sección 2.2.7, formulamos el proceso, que considera las fases para construir aplicaciones, usando Scrum donde dictamina la utilización de tres artefactos esenciales en un proyecto: la pila del proyecto, con los requerimientos totales del aplicativo a desarrollar; la pila de iteración, que determina los requerimientos a desarrollar en cada iteración; y la gráfica de trabajo pendiente, que representa el trabajo que falta por hacer en un proyecto en el tiempo (Laura, 2006), como se muestra en las tablas N° 3.2 y 3.4.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLE
Asignar roles primarios y secundarios	Propietarios del proyecto	Identificar los responsables de los roles primarios y secundarios	Propietario del Producto y equipo
Elaboración de la pila del proyecto	Matriz de la pila del Proyecto	Revisión de requerimientos y elaboración de la matriz o pila de Iteraciones	equipo
Definir la cantidad de iteraciones	Integrar las iteraciones a la pila del proyecto	Clasificar las tareas de la pila del proyecto	equipo
Asignar prioridades del proyecto	Integrar las prioridades a la pila del proyecto	Evaluar y asignar prioridades a todas las tareas de la pila del	equipo

		proyecto.	
Asignar horas esfuerzo	Integrar las horas esfuerzo a la pila del proyecto	Evaluar y estimar las horas esfuerzo para cada actividad de la pila del proyecto	equipo
Elaborar la matriz de la pila del proyecto	Pila del proyecto integrado	Matriz o pila de proyecto concluido	equipo

Tabla N° 3.2: Entregables y sub-entregables de artefactos Scrum (Laura, 2006).

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLE
Reunión de planificación del Sprint.	Pila de iteración aprobada por ítem	Reunión de planificación, disolución de dudas, distribución de tareas	Propietario, Facilitador y equipo
Elaborar la pila de iteración primaria	Pila de iteración primaria	Ítem de la pila del proyecto e identificación de las tareas en cada ítem.	Facilitador y equipo
Revisión de la pila de iteración	Pila de iteración actualizada	Revisión del avance y recálculo del esfuerzo.	equipo
Mostrar resultados para cada tarea.	Capturas por cada iteración	Mostrar los avances capturadas	equipo
Realizar el gráfico de tarea pendiente de la primera iteración	Gráfica de tarea pendiente de la primera iteración	Revisar la actualización de la primera iteración y consolidar los esfuerzos realizados para elaborar el gráfico de trabajo pendiente.	Equipo
Realizar la reunión de aprobación de la primera iteración	Aprobación de la primera pila de iteración	Reunión de aprobación	Propietario, facilitador y equipo
Pasar a la siguiente iteración	Elaborar las matrices de la siguiente iteración	Revisar faces de desarrollo de software y UML	Facilitador y equipo

Tabla N° 3.3: Entregables de Scrum en el ciclo de desarrollo (Guzmán et al., 2012).

CAPÍTULO IV

ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

4.1 RESULTADOS

Según las fases del desarrollo de componentes software usando el marco de trabajo Scrum integrándolo con la notación UML, desarrolladas en las tablas 3.2 y 3.3, descrito en el capítulo II sección 2.2.7, se diseñan las características, roles y artefactos Scrum con el análisis y diseño de la pila de iteración, implementación, y prueba. A continuación mostramos los resultados de los artefactos de acuerdo al proceso Scrum y la notación UML.

4.1.1. CARACTERÍSTICAS DE SCRUM

Prácticas y roles pre-definidos, los cuales fueron debidamente determinados para el presente proyecto se detalla a continuación.

4.1.1.1. ROLES

A. ROLES PRINCIPALES

En el grupo tenemos los roles que están comprometidos enteramente con el desarrollo del componente software para lograr el éxito del aplicativo.

PROPIETARIO DEL PRODUCTO (PRODUCT OWNER)

Responsable: Bach. Abilio Cáceres Curo.

El Bach. Abilio Cáceres Curo, representa al propietario del producto, como representante con el rol de arquitecto de software. Representa a todos los interesados con el proyecto, verificando continuamente el avance del mismo, evaluando los diferentes logros alcanzados al final de cada iteración y tomando decisiones respecto a las futuras iteraciones, de tal modo que el componente software denominado Middleware SQL, sea acorde a las expectativas de los Stakeholders.

FACILITADOR (SCRUM MASTER)

Responsable: Bach. Abilio Cáceres Curo.

Responsable de la correcta dirección de los avances del proyecto. Durante cada iteración se comprometió a verificar la disponibilidad de todos los elementos que requiera el equipo para su correcto desarrollo del proyecto. Además actúa como intermediario para facilitar una correcta comunicación entre el equipo de trabajo y la propietaria del producto.

EQUIPO (TEAM)

Responsable: Bach. Abilio Cáceres Curo.

Encargado del desarrollo del aplicativo en cada una de las iteraciones. Comprometido con el correcto funcionamiento del producto entregado al final de cada iteración. En concordancia con las buenas prácticas de Scrum, no está a cargo una tarea específica dentro del proyecto, realiza el modelado arquitectónico, análisis, diseño, implementación, evaluación y puesta en marcha el componente software.

B. ROLES SECUNDARIOS

Los roles externos, dan un mayor grado de satisfacción y eficiencia del componente software (Middleware SQL) que va a ser entregado y usado en nuevas aplicaciones de software empresarial.

USUARIO (USER)

Los usuarios que optaron por el componente software como una alternativa o técnica de persistencia de datos en sistemas informáticos con uso de base de datos distribuidos homogéneo o heterogéneo.

CLIENTE (CUSTOMER)

El cliente representa una generalización mayor del propietario del producto. Dentro del presente proyecto es el arquitecto de software al que será entregado el componente software completo para su posterior uso y puesta en marcha en nuevos proyectos.

ADMINISTRACIÓN (MANAGEMENT)

En este proyecto se consideró en la administración a Abilio Cáceres Curo, persona que facilitó la información necesaria para el desarrollo del componente software.

4.1.2. ARTEFACTOS SCRUM

Scrum dictamina la utilización de 3 artefactos esenciales en un proyecto: la pila del proyecto, con los requisitos totales del aplicativo a desarrollar; la pila de la iteración, que determina los requisitos a desarrollar en cada iteración; y la gráfica de trabajo pendiente, que representa el trabajo que falta por hacer en un proyecto en el tiempo.

A. PILA DEL PROYECTO

ITERACIÓN	ÍTEM	DESCRIPCIÓN	ESFUERZO ESTIMADO (Horas)	PRIORIDAD
I	1	Constituir el proyecto.	20	ALTA
	2	Diseñar el modelo arquitectónico.	50	ALTA
	3	Instalación de las herramientas tecnológicas.	20	MEDIA
	4	Configurar el entorno de gestión de la configuración (CM).	20	ALTA
II	5	Diseñar y desarrollar el componente software, Middleware SQL.	100	ALTA
	6	Generar el instalador del componente software.	10	ALTA
	7	Elaborar manual de usuario del componente software.	10	ALTA
III	8	Desarrollar nueva aplicación SpcMid con Middleware SQL.	40	MEDIA
	9	Desarrollar nueva aplicación SpcH con Framewok Hibernate.	40	MEDIA
	10	Medir y formalizar las tablas de eficiencia.	10	ALTA

Tabla N°4.1: Pila del proyecto integrado para desarrollo del componente Software.

Los otros dos artefactos de la pila de iteración y gráfica de trabajo pendiente se detallarán individualmente dentro de cada iteración.

B. ITERACIONES DE SCRUM

Mediante una reunión entre los principales roles del proyecto; Propietario,

Facilitador y Equipo; se han determinado tres iteraciones para el desarrollo del proyecto. La primera iteración consiste en la formalización del proyecto, diseño del modelo arquitectónico y configuración del entorno del proyecto; la segunda iteración consiste en análisis, diseño, implementación y documentación de componente software denominado Middleware SQL; la tercera iteración de medidas y formalización de las tablas de eficiencia de acceso a las bases de datos distribuidas.

4.1.3. ITERACIÓN I: MODELO ARQUITECTÓNICO Y CONFIGURACIÓN DEL ENTORNO DEL PROYECTO

4.1.3.1. RESUMEN

La primera iteración del proyecto consistió en la investigación de arquitecturas de software, patrones de diseño, desarrollo y la elaboración del diseño de un modelo arquitectónico para la construcción del componente software denominado Middleware SQL, así mismo con la configuración del entorno del proyecto. Se trabajó en conjunto con el propietario del producto, quien facilitó el proceso de inspección, investigación de las diversas arquitecturas de software, patrones de diseño e implementación que constituyen los componentes software para permitir el acceso a las BDD. Con toda esta fuente de información obtenida se procedió a diseñar un nuevo modelo o modelos arquitectónicos que subsana las ineficiencias de los demás modelos arquitectónicos, y posteriormente se realizará la configuración del entorno del proyecto. Así mismo, la instalación y configuración de herramientas, controles de versiones cuyo resultado final viene a ser el incremento obtenido de esta iteración.

4.1.3.2. REUNIÓN DE PLANIFICACIÓN DEL SPRINT

Durante la primera reunión del proyecto se revisó la pila del producto con el propietario, facilitador y equipo, en la que se definió como meta el diseño del modelo arquitectónico para el desarrollo del componente software denominado Middleware SQL.

La iteración tomaría un esfuerzo de 110 horas, fue dividido a lo largo de 18 días no consecutivos, asumiendo un trabajo diario de 6 horas sin tomar en cuenta descansos ni recesos, asimismo ésta fue dividida en 4 ítems. Al final de esta reunión se generó la pila de iteración.

4.1.3.3. PILA DE LA ITERACIÓN

La primera pila de iteración consta de todos los requerimientos que están relacionados con la investigación y diseño de un modelo o modelos arquitectónicos para construir el componente software y su posterior configuración del entorno del proyecto. En la tabla N° 4.2 se detalla la pila de iteración para los 10 primeros días, y el resto de días se detallan en la tabla N° 4.3.

Iteración I: Modelo arquitectónico y configuración del entorno del proyecto											
ÍTEM	Tarea	Días de la iteración/esfuerzo restante									
		0	1	2	3	4	5	6	7	8	9
		110	104	98	92	86	78	69	60	52	46
1	Constituir el proyecto.										
	Concebir un nuevo proyecto.	2	0	0	0	0	0	0	0	0	0
	Identificar los objetivos del negocio.	2	0	0	0	0	0	0	0	0	0
	Elaborar el cronograma de actividades.	4	2	0	0	0	0	0	0	0	0
2	Diseñar el modelo arquitectónico.										
	Análisis patrones arquitectónicos.	15	15	11	8	4	0	0	0	0	0
	Identificar componentes del modelo arquitectónico.	10	10	10	7	5	2	1	0	0	0
	Diseñar un nuevo modelo arquitectónico.	25	25	25	25	25	24	20	15	10	6
	Formalizar y generar el documento de diseño.	16	16	16	16	16	16	12	10	8	6
3	Instalación de las herramientas tecnológicas.										
	Seleccionar las herramientas tecnológicas.	4	4	4	4	4	4	4	3	2	2
	Instalar y configurar las herramientas tecnológicas.	10	10	10	10	10	10	10	10	10	10
4	Configurar el entorno de gestión de la configuración (CM).										
	Definir parámetros de configuración.	10	10	10	10	10	10	10	10	10	10
	Configurar servidores	6	6	6	6	6	6	6	6	6	6
	Configurar control de versiones	6	6	6	6	6	6	6	6	6	6

Tabla N°4.2: Pila de iteración I – Primera parte.

Iteración I: Modelo arquitectónico y configuración del entorno del proyecto									
ÍTEM	Tarea	Días de la iteración/esfuerzo restante							
		10	11	12	13	14	15	16	17
		38	34	28	24	20	12	6	0
1	Constituir el proyecto.								
	Concebir el nuevo proyecto.	0	0	0	0	0	0	0	0
	Identificar los objetivos del negocio.	0	0	0	0	0	0	0	0
	Elaborar el cronograma de actividades.	0							
2	Diseñar el modelo arquitectónico.								
	Análisis de los patrones arquitectónicos.	0	0	0	0	0	0	0	0
	Identificar componentes del modelo arquitectónico.	0	0	0	0	0	0	0	0
	Diseñar un nuevo modelo arquitectónico.	2	0	0	0	0	0	0	0
	Formalizar y generar el documento de diseño.	2	0	0	0	0	0	0	0
3	Instalación de las herramientas tecnológicas.								
	Seleccionar las herramientas tecnológicas.	2	2	0	0	0	0	0	0
	Instalar y configurar las herramientas tecnológicas.	10	10	8	8	6	2	0	0
4	Configurar el entorno de gestión de la configuración (CM).								
	Definir parámetros de configuración.	10	10	8	8	6	6	4	0
	Configurar servidores.	6	6	6	4	2	0	0	0
	Configurar control de versiones.	6	6	6	4	6	4	2	0

Tabla N°4.3: Pila de Iteración I – Segunda parte.

4.1.3.4. TAREAS

A. ÍTEM 1: Constituir el proyecto

a. Concebir el nuevo proyecto

Se identificó las limitaciones que surge cuando un sistema informático tiene que acceder a las bases de datos distribuida y su vez se investigó logrando identificar posibles soluciones basadas en middleware.

Número	Limitaciones	Unidad
1	Tiempo empleado en la ejecución consultas críticas afecta directamente al rendimiento.	Milisegundos
2	Incrementa la cantidad de código implementado durante el tiempo de ejecución	
3	Memoria Heap consumida en la ejecución instrucciones	Megabyte

	críticas.	
4	Especialización de consultas y gestores de base de datos.	Ninguno
5	Líneas de código implementado excesivo.	Cantidad
6	Es simple. Esto es bueno y malo al mismo tiempo. Con lógica de negocio compleja, este patrón pierde coherencia.	Porcentaje
7	Acoplado a la estructura de la clase, un cambio en el diseño de la tabla, implica cambiar la clase.	Porcentaje
8	En situaciones de operaciones de alto volumen de datos, el overhead que se paga en el pasaje y carga de datos, es innecesario. Esta desventaja aplica tanto a Active Record, como a cualquier otro diseño orientado a objetos	Megabyte
9	Muchos “puristas” de OOP critican que ActiveRecord tiene una misma clase tanto la responsabilidad de acceder a la base como de manejar la lógica de negocio y “ensucia” el código.	Porcentaje
10	Excesivo % de CPU utilizado por el proceso.	Porcentaje

Tabla N°4.4: Limitaciones en el acceso a datos.

Así mismo se analizó los diversos incompatibilidades y compatibilidades de los tipos de datos manejados por las bases de datos; Oracle 11g, Microsoft Sql Server 2012, PostgreSQL 9; y el lenguaje Java, la misma que se detalla en la tabla 4.5.

Java 8	SQL-92
byte	Ninguno
short	SMALLINT
int	INTEGER
long	Ninguno
float	REAL
double	DOUBLE PRECISION
char	NATIONAL CHARACTER (Dimensión: 1)
boolean	BOOLEAN
java.lang.Integer	SMALLINT INTEGER INT
java.math.BigDecimal	DECIMAL DEC NUMERIC
java.lang.Double	FLOAT DOUBLE PRECISION
java.lang.Float	REAL
java.lang.String	CHARACTER CHAR NATIONAL CHARACTER NATIONAL CHAR NCHAR NATIONAL CHARACTER VARYING

	NATIONAL CHAR VARYING NCHAR VARYING
java.lang.Boolean	BIT BOOLEAN
Ninguno	BIT VARYING
java.sql.Date	DATE
java.sql.Time	TIME
java.sql.Timestamp	TIMESTAMP
Ninguno	INTERVAL
java.sql.Clob	CHARACTER LARGE OBJECT CHAR LARGE OBJECT CLOB NATIONAL CHARACTER LARGE OBJECT NCHAR LARGE OBJECT NCLOB
java.sql.Blob	BINARY LARGE OBJECT BLOB
java.sql.Array	ARRAY
Ninguno	ROW
java.sql.Ref	REF
java.lang.Long	BIGINT
Ninguno	SCOPE MULTISET

Tabla N°4.5: Equivalencias de los tipos de dato Java y Sql.

SQL-92	PostgresSql 9	Microsoft Sql Server 2012	Oracle 11g
CHARATER CHAR	CHAR	CHAR	CHAR
CHARACTER VARYING CHAR VARYING VARCHAR	VARYING	VARCHAR	VARCHAR
NATIONAL CHARACTER NATIONAL CHAR NCHAR	CHAR	NCHAR	NCHAR
NATIONAL CHARACTER VARYING NATIONAL CHAR VARYING NCHAR VARYING	VARYING	NVARCHAR	NVARCHAR2
NUMERIC DECIMAL DEC	NUMERIC DECIMAL	NUMERIC DECIMAL	NUMBER
INTEGER INT	INTEGER	INT	NUMBER

SMALLINT	NUMERIC	SMALLINT	NUMBER
FLOAT	NUMERIC	FLOAT	NUMBER
REAL DOUBLE PRECISION	NUMERIC	REAL FLOAT	NUMBER
BIT BIT VARYING	BIT VARYING	BINARY VAR BINARY	No aplicable.
DATE	TIMESTAMP	DATETIME O SMALLTIME	DATE
TIME	TIMESTAMP	TIMESTAMP	TIMESTAMP
INTERVAL	No aplicable.	No aplicable.	No aplicable.

Tabla N°4.6: Equivalencias de los tipos de dato de los DBMS seleccionados.

b. Identificar los objetivos del negocio

Se identificó los 3 objetivos frente a estas las limitaciones identificadas anteriormente; analizar y diseñar un modelo arquitectónico, con la finalidad de mejorar el acceso a las bases de datos distribuida, analizar y diseñar, implementar y probar la capa de abstracción, con la finalidad de mejorar el acceso a las bases de datos distribuida, desarrollar y probar las interfaces, con la finalidad de mejorar el acceso a las bases de datos distribuida.

c. Elaborar el cronograma de actividades

Se elaboró un cronograma de actividades asociadas a la pila del proyecto estimando las horas de esfuerzo que tomarían dichas actividades que se identificaron y las prioridades en el desarrollo del componente software denominado Middleware SQL, estas horas fueron estimadas en la primera reunión junto con el administrador, facilitador, dueño del proyecto y el equipo que por convención logró elaborar la pila del proyecto.

B. ÍTEM 2: Diseñar el modelo arquitectónico

a. Análisis de los patrones arquitectónicos

Se investigó y analizó los diseños de patrones arquitectónicos existentes actuales, como los modelos arquitectónicos de los frameworks de persistencia, los ODBC, JDBC nativos y middleware. Los patrones de acceso a datos identificados son; **Active Record**.- Este patrón de persistencia es quizás uno de los más habituales y es usado por frameworks como Rails. La idea detrás del patrón es bastante sencilla. Se trata de una clase que se encarga de implementar todas las operaciones de consulta y modificación de una tabla concreta de la base de datos como se visualiza en la Figura

N° 4.1; **Data Access Object**.- Este patrón de diseño divide más las responsabilidades en la aplicación de tal forma que tendremos unas clases que se encargarán de la lógica de negocio y otras clases la responsabilidad de persistencia como se visualiza en Figura N° 4.2.

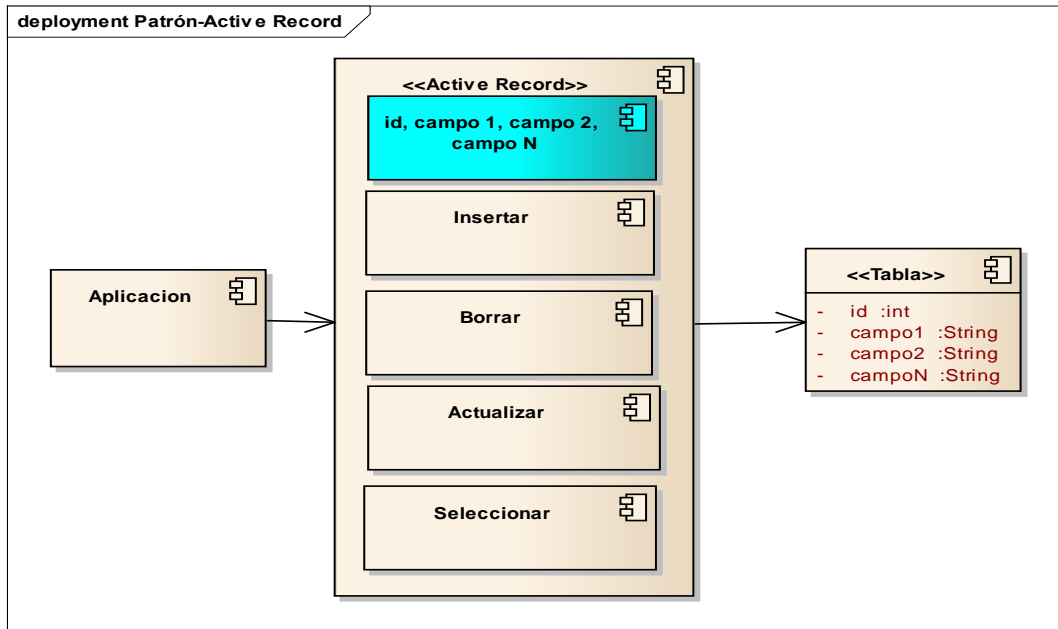


Figura N° 4.1: Patrón active record para acceso a datos (Fowler, 2003).

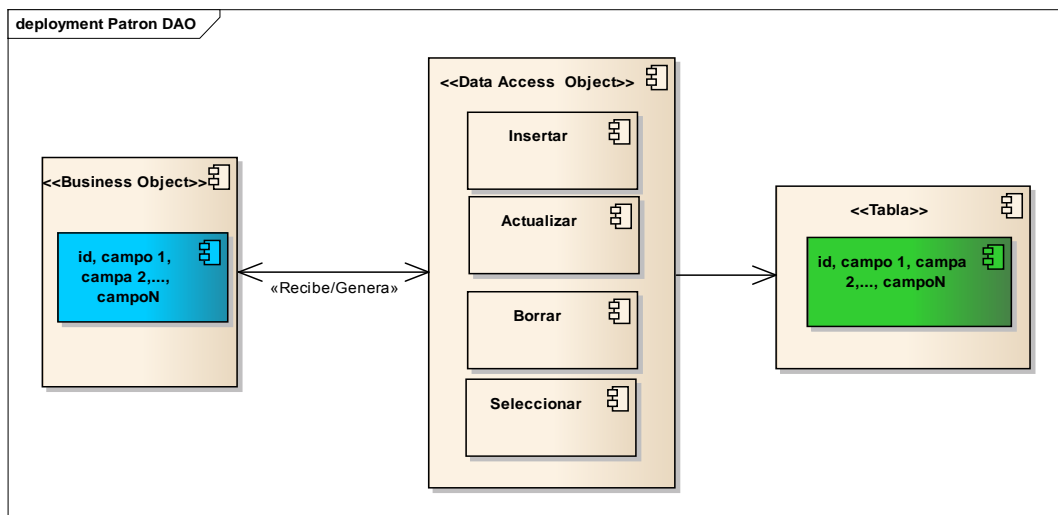


Figura N° 4.2: Patrón Data Access Object para acceso a datos (Sun Microsystems, 2007).

b. Identificar componentes del modelo arquitectónico

Se identificó los componentes que se abstraen de los patrones existentes, así mismo se empieza diseñar subcomponentes para un diseño posterior.



Figura N° 4.3 El middleware dentro del entorno cliente/servidor.

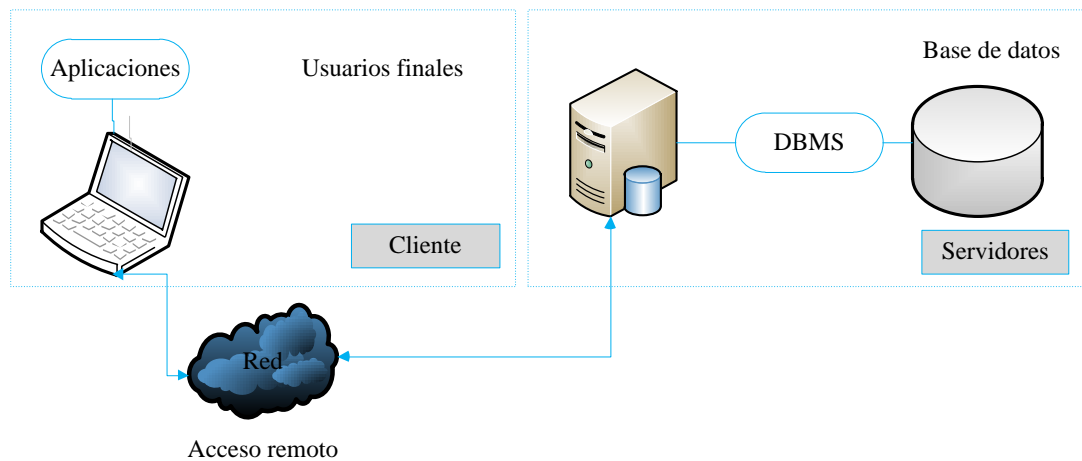


Figura N° 4.4: Función de Cliente/Servidor de un Sistema de Bases de Datos Velasco (2010).

Las bases de datos distribuidos permiten las siguientes características: Autonomía local, No-dependencia de un sitio central, Operación continua, Independencia de ubicación, Independencia de fragmentación, Independencia de replicación, Procesamiento de consultas distribuidas, Administración de transacciones distribuidas, Independencia del hardware, Independencia del OS, Independencia de red e Independencia del DBMS.

El Cliente podría ser capaz de acceder a varios servidores en forma simultánea (es decir, una sola petición de base de datos podría combinar datos de varios servidores). En este caso, los servidores ven al Cliente – desde un punto de vista lógico – como si en realidad fuera un solo Servidor y el usuario no tiene que saber qué computadoras contienen qué piezas de datos (Date, 2001).

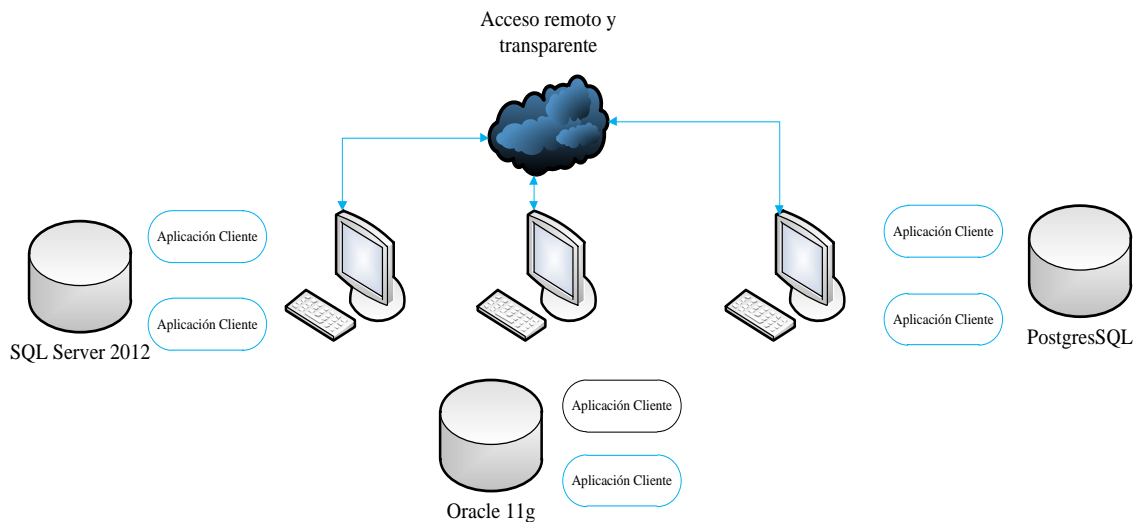


Figura N° 4.5: Modelo de acceso remoto transparente (Date, 2001).

El procesamiento de consultas hace referencia a la serie de actividades implicadas en la extracción de datos en una base de datos. Estas actividades incluyen la traducción de consultas expresadas en lenguajes de bases de datos de alto nivel en expresiones implementadas en el nivel físico del sistema, así como técnicas de optimización de consultas y la evaluación real de las mismas. El costo del procesamiento de una consulta está determinado por la cantidad de veces que se accede al disco duro de la computadora, comparativamente, más lento que el acceso a la memoria principal. Normalmente hay muchas estrategias posibles para procesar una consulta dada, especialmente si la consulta es compleja.

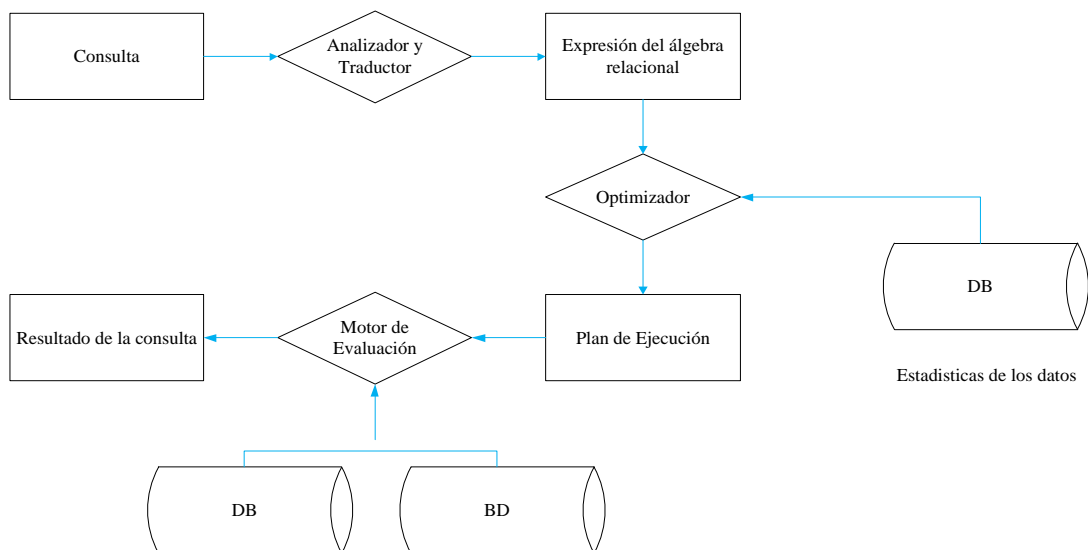


Figura N° 4.6: Procesos en el procesamiento de una consulta (Date, 2001).

c. Diseñar un nuevo modelo arquitectónico

Se diseñó un modelo arquitectónico para desarrollar un componente software denominado Middleware SQL que acceda a una base de datos distribuida.

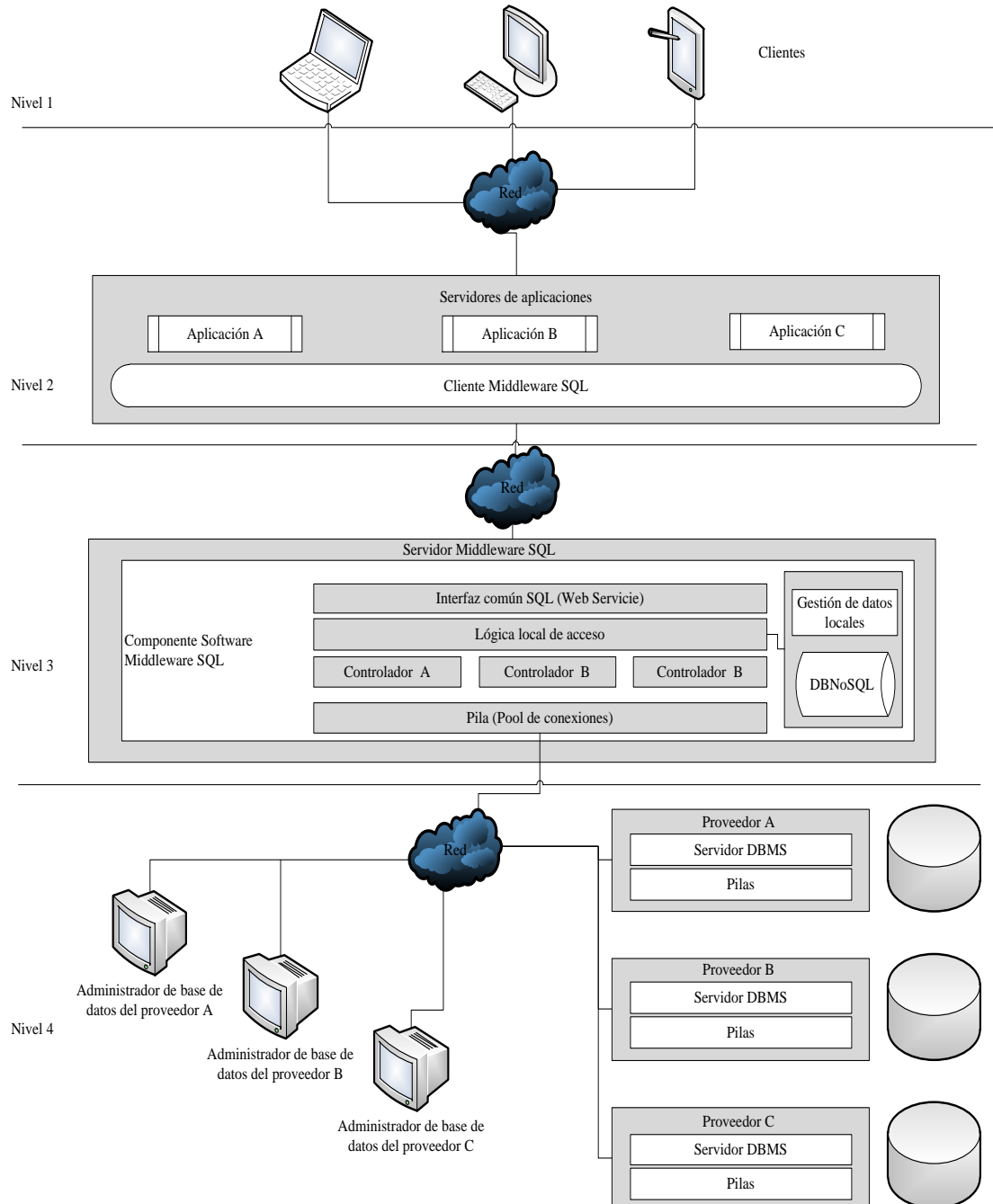


Figura N° 4.7: Modelo arquitectónico del funcionamiento del Middleware SQL.

Se detalla la funcionalidad interna del componente software, en la que se visualiza cómo funciona la gestión de la configuración y el proceso de la obtención de los datos de entrada, ejecución de la persistencia de la misma y la respuesta de dicho proceso.

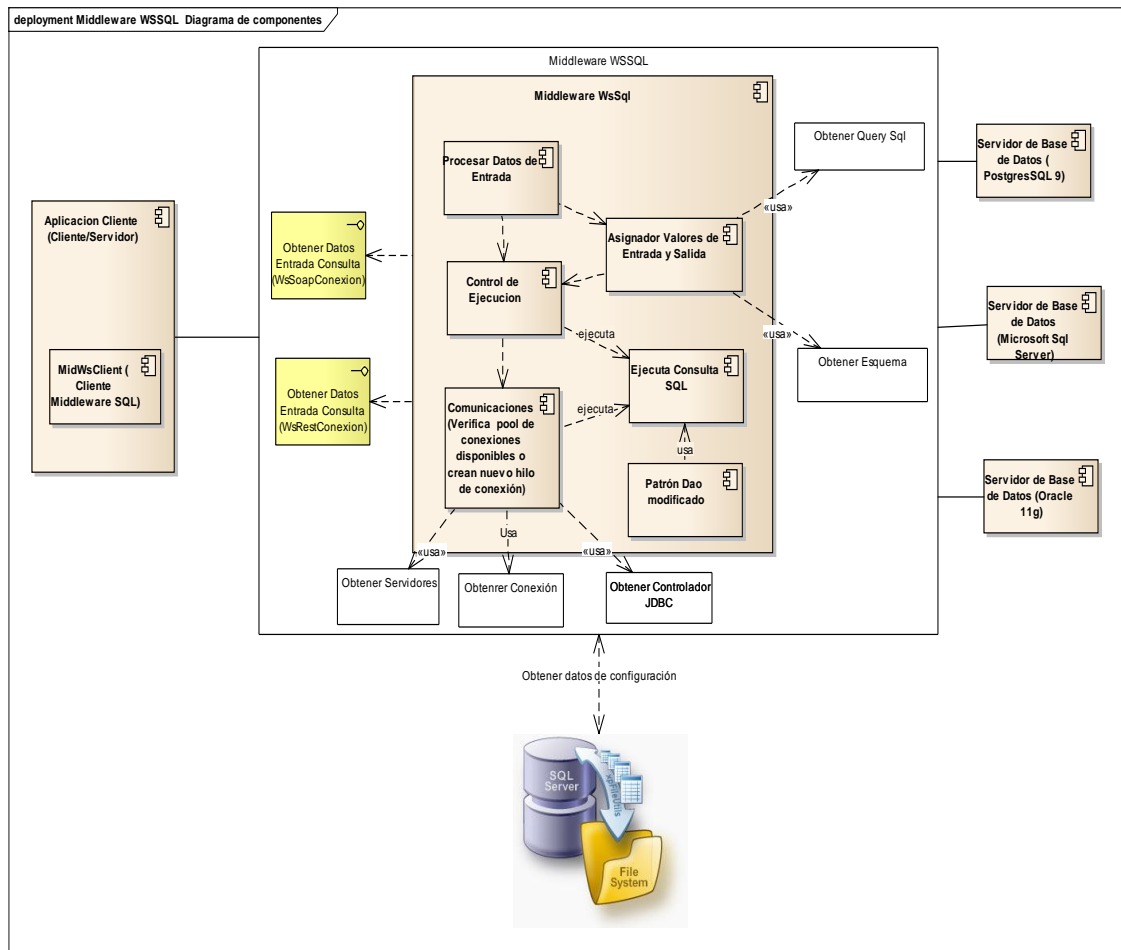


Figura N° 4.8: Modelo arquitectónico interno del Middleware SQL.

d. Formalizar y generar el documento de diseño

Se formalizó el diseño el modelo arquitectónico, así mismo se procedió a generar el modelo gráfico de la arquitectura focalizándonos en la gestión de información que el componente software realiza para mantener su autonomía cumpliendo sus objetivos en el acceso a los datos en entornos distribuidos y tecnologías de base de datos diversos.

C. ÍTEM 3: Instalación de las herramientas tecnológicas

a. Seleccionar las herramientas tecnológicas

Se seleccionó las herramientas tecnológicas que nos permitirá el desarrollo del proyecto.

b. Instalar y configurar las herramientas tecnológicas

Se instaló las herramientas tecnológicas seleccionadas, se configuraron los

sistemas operativos Microsoft Windows 7 y Linux 15.10; los servidores web apache tomcat 8; entorno de desarrollo eclipse 4; bases de datos Microsoft sql server 2012, Oracle 11g y PostgresSql 9; IDEs de modelado y desarrollo de software como Architec Enterprise y Microsoft Visio.

D. ÍTEM 4: Configurar el entorno de gestión de la configuración (CM)

a. Definir parámetros de configuración

Se definió parámetros de configuración para el desarrollo del componente software como son los servidores web, servidores de archivos y servidor de versiones.

b. Configurar servidores

Se configuró los servidores web, el servidor de desarrollo, los servidores de base de datos, en lo mínimo básico funcional para garantizar un sistema distribuido heterogéneo que permitió el desarrollo del componte software.

c. Configurar control de versiones

Se configuró un control de versiones para tener flexibilidad en los cambios y versiones del proyecto para ello se optó por SVN, que nos permitirá tener el control de versiones de documentos, fuentes y diseños del componente software.

4.1.3.5. GRÁFICA DE TRABAJO PENDIENTE

Se muestra el esfuerzo restante desde el inicio de la iteración en el día cero, hasta el final de la iteración que fue determinada en 18 días.

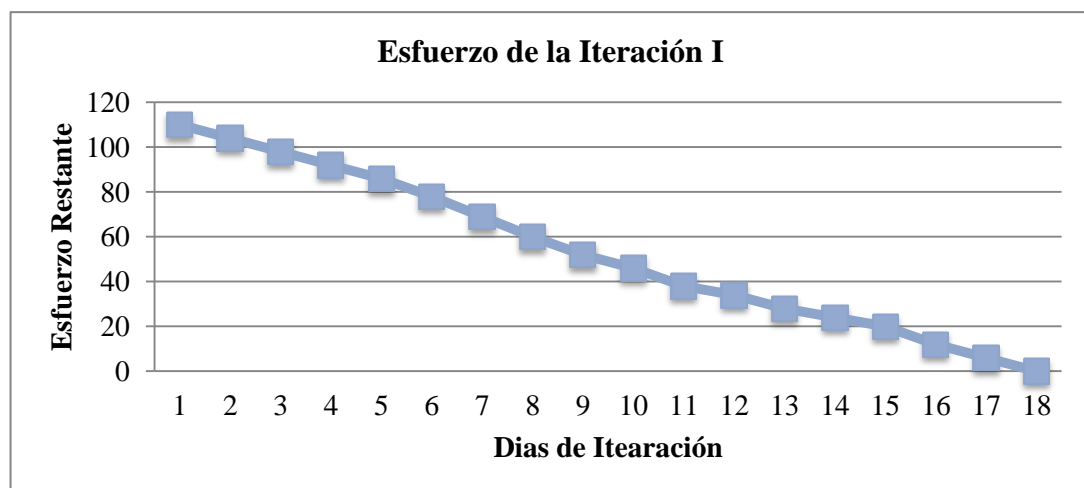


Figura N° 4.9: Gráfica de trabajo pendiente de la iteración I.

Existe un decremento constante del esfuerzo requerido en el tiempo, para lograr completar todos los requerimientos al final de los 18 días de la iteración.

4.1.3.6. REUNIÓN DE REVISIÓN DE LA ITERACIÓN

La primera reunión de revisión fue realizada para mostrar el incremento del desarrollo a lo largo de la primera iteración, junto con el facilitador y propietario del producto se revisó los resultados, obteniendo una evaluación favorable del detalle en el diseño alcanzado, se presentó en modelo arquitectónico y los patrones utilizados. Así mismo se acordó la secuencia de la próxima iteración teniendo como objetivo el desarrollo del componente software denominado Middleware SQL.

4.1.4. ITERACIÓN II: DESARROLLO DEL COMPONENTE SOTWARE MIDDLEWARE SQL

4.1.4.1. RESUMEN

En esta iteración se enfoca el trabajo en la realización del componente software Middleware SQL para acceder a las bases de datos distribuida, basada en el modelo arquitectónico diseñado en la iteración I; el lenguaje de programación orientada a objetos Java; los frameworks Spring 4, Primefaces 5, patrón DAO modificada; estas herramientas permitió el análisis, diseño e implementación. Al final de esta iteración se logró entregar el Middleware SQL que pueda ser utilizada en nuevas aplicaciones empresariales.

4.1.4.2. REUNIÓN DE PLANIFICACIÓN DEL SPRINT

En esta reunión se seleccionaron los requerimientos que están enfocadas al desarrollo del Middleware SQL. La meta fue entregar el middleware SQL funcional para que pueda ser utilizado en las aplicaciones cliente y primordialmente aquellas con alta criticidad con acceso a las bases de datos distribuida. Al finalizar esta fase de iteración el Middleware SQL publicó dos servicios denominados WsSoapConexionService y WsRestConexionService endpoints del servicio web que permite la comunicación al crear nuevas aplicaciones empresariales.

Por segunda ocasión se estima un esfuerzo de 120 horas, divididas en 20 días sin tomar en cuenta los fines de semana. Se estimó 100 horas para diseñar y desarrollar el componente software Middleware SQL, 10 horas para preparar y generar el instalador

y 10 horas para documentar el manual de usuario del Middleware SQL.

4.1.4.3. PILA DE LA ITERACIÓN

La pila de iteración en la tabla N° 4.7 y 4.8, está compuesta de todos los requerimientos que estén relacionados con el desarrollo del Middleware SQL, la configuración para su funcionamiento como la publicación de servicios denominado Middleware WSSQL para las consultas de acceso a las bases de datos distribuidos de forma transparente y optimizada.

Iteración II: Desarrollo del Componente Software Middleware SQL											
ÍTEM	Tarea	días de la iteración/esfuerzo restante									
		0	1	2	3	4	5	6	7	8	9
		120	112	106	98	90	82	76	70	64	58
5	Diseñar y desarrollar el componente software, Middleware SQL.										
	Diseñar y desarrollar el Middleware SQL.	60	52	50	44	36	28	26	20	14	12
	Diseñar y desarrollar los contratos web services.	40	40	36	34	34	34	30	30	30	26
6	Generar el instalador del componente software.										
	Definir parámetros de compilación.	4	4	4	4	4	4	4	4	4	4
	Instalar el middleware SQL y probar en varios ambientes	6	6	6	6	6	6	6	6	6	6
7	Elaborar manual de usuario del componente software.										
	Documentar las opciones y el manejo de Middleware SQL.	10	10	10	10	10	10	10	10	10	10

Tabla N°4.7: Pila de Iteración II – Primera parte.

Iteración II: Desarrollo del Componente Software Middleware SQL											
ÍTEM	Tarea	días de la iteración/esfuerzo restante									
		10	11	12	13	14	15	16	17	18	19
		50	45	37	31	24	19	12	8	4	0
5	Diseñar y desarrollar el componente software, Middleware SQL.										
	Diseñar y desarrollar el Middleware SQL.	12	8	6	0	0	0	0	0	0	0
	Diseñar y desarrollar los contratos	18	18	12	12	6	4	0	0	0	0

	web services.										
6	Generar el instalador del componente software.										
	Definir parámetros de compilación.	4	3	3	3	2	1	0	0	0	0
	Instalar el middleware SQL y probar en varios ambientes.	6	6	6	6	6	6	4	2	0	0
7	Elaborar manual de usuario del componente software.										
	Documentar las opciones y el manejo de Middleware SQL.	10	10	10	10	10	8	8	6	4	0

Tabla N°4.8: Pila de Iteración II – Segunda parte.

4.1.4.4. TAREAS

A. ÍTEM 5: Diseñar y desarrollar el componente software Middleware SQL

a. Diseñar y desarrollar el Middleware SQL

Se ha utilizado la notación UML para generar los artefactos de análisis, diseño e implementación que se genera en la ingeniería de software, para ello Scrum declara que los artefactos UML generados son a voluntad de las necesidades para el entendimiento y desarrollo que está a cargo del equipo.

REQUISITOS FUNCIONALES

Requisito	Descripción
1	El usuario podrá gestionar los Antisondeos de una conexión.
2	El usuario podrá gestionar los Servidores de base de datos.
3	El usuario podrá gestionar los Queries SQL
4	Se le permitirá al podrá gestionar las Empresa Mid, entidad de la empresa y la aplicación asociada.
5	El usuario podrá gestionar las conexiones.
6	El usuario podrá asociar a las conexiones a la empresa mid.
7	El usuario podrá asociar las queries SQL a la empresa mid.
8	El usuario podrá generar los datos locales de configuración del middleware SQL.
9	El usuario podrá gestionar las incidencias generadas por el Middleware SQL al ejecutar las conexiones y los queries.

Tabla N°4.9: Requisitos funcionales del Middleware SQL.

CASOS DE USO

Documentan el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir,

representan las funciones que Middleware SQL ofrece en el en su funcionalidad para mantener su autonomía en su configuración.

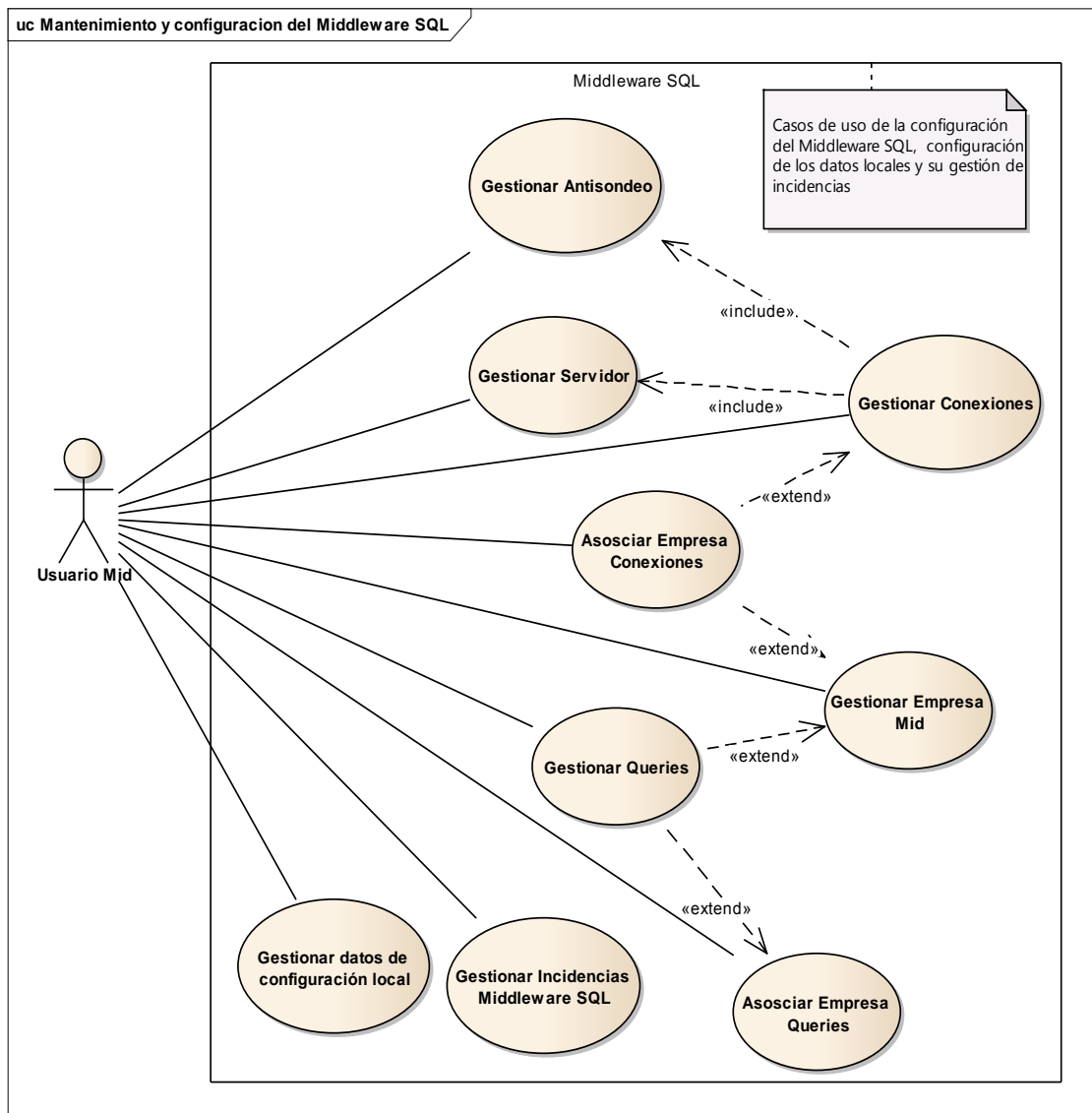


Figura N° 4.10: Diagrama de Casos de uso del mantenimiento, configuración y gestión del Middleware SQL.

Actor	Primario: Usuario Mid
Descripción	Se administrará un antisondeo que posteriormente, podrá asociarse a una conexión.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”.
Post-condiciones	Ninguna
Req. Especiales	Ninguno
Flujo Normal – Buscar Antisondeos	
Escenario	El actor (Usuario Mid) podrá aplicar un conjunto de condiciones

	sobre la lista de antisondeos visualizada.
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado una opción”. • Ex.2: Visualizará la alerta: “No se encontraron resultados”. • Ex.3: Visualizará la alerta: “El antisondeo está siendo usada en una conexión”
Actor	Sistema
1. Seleccionará la opción Antisondeos del menú izquierdo Mantenimiento.	
	<p>2. Visualizará la pantalla del Prototipo N° 1, con los siguientes controles:</p> <ul style="list-style-type: none"> • Filtros de Antisondeo: sección que mostrará los filtros de búsqueda de antisondeos: <ul style="list-style-type: none"> o Número de Consultas: campo numérico con una longitud máxima de 10 dígitos. o Tiempo Máximo: campo numérico con una longitud máxima de 10 dígitos. o Estado: lista desplegable de selección simple que mostrará los estados de un antisondeo, sus valores serán: <ul style="list-style-type: none"> ▪ CREADO ▪ ACTIVADO ▪ DESACTIVADO <p>3. También mostrará los botones:</p> <ul style="list-style-type: none"> • Buscar: permitirá realizar la búsqueda de antisondeos, su funcionalidad será descrita más adelante. • Limpiar: permitirá limpiar los campos de la pantalla, colocando los valores en: <ul style="list-style-type: none"> o Número de consultas: blanco o Tiempo de consulta: blanco o Estado: Seleccionar • Nuevo: permitirá crear nuevos antisondeos, ver Flujo Normal – Nuevo Antisondeo.
4. Seleccionará o ingresará los filtros que desee aplicar a su búsqueda. 5. Seleccionará el botón Buscar .	
	<p>6. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresará ningún criterio de búsqueda, generará la excepción Ex1. <p>7. Si no se generara ninguna excepción, buscará los antisondeos que coincidan con los criterios de búsqueda ingresados y mostrará la sección Resultados de la Búsqueda con los registros encontrados. La sección Resultados de Búsqueda</p>

	<p>mostrará las siguientes columnas:</p> <ul style="list-style-type: none"> • Código: mostrará el código del antisondeo, los resultados obtenidos estarán ordenados según esta columna. • Número de consultas: mostrará el número de consultas que permite el antisondeo. • Tiempo Máximo: Mostrará el tiempo que es asignado para las consultas. • Estado: mostrará el estado del antisondeo. <p>Ver Prototipo N° 2</p> <p>8. Si no se encontraran resultados para la búsqueda, mostrará el mensaje “No se encontraron resultados”, dentro de la tabla de resultados.</p> <p>9. También se muestra los botones:</p> <ul style="list-style-type: none"> • Ver Detalle: Deshabilitada, permitirá ver el detalle del antisondeo que será descrita más adelante. • Eliminar: Deshabilitada, permitirá borrar el antisondeo que será descrita más adelante.
<p>10. Elegirá un antisondeo de la sección de resultados.</p> <p>11. Seleccionará el botón Ver Detalle.</p>	
	<p>12. Si no seleccionara un antisondeo de la sección de resultados, los botones Ver Detalle y Eliminar seguirán deshabilitadas.</p> <p>13. Visualizará la pantalla del prototipo N° 3, con los siguientes controles:</p> <ul style="list-style-type: none"> • Propiedades de Antisondeo: sección que mostrará las propiedades del antisondeo seleccionado: <ul style="list-style-type: none"> o Código: código del antisondeo, campo no editable. o Número de Consultas: campo numérico, campo deshabilitado. o Tiempo Máximo: campo deshabilitado. o Estado: campo no editable. • También mostrará los botones: <ul style="list-style-type: none"> o Editar: permitirá editar las propiedades de un antisondeo, Ver Flujo Normal – Editar Propiedades de un Antisondeo. o Regresar: regresará a la pantalla de Listado de Antisondeos.
<p>14. Elegirá un antisondeo de la sección de resultados.</p> <p>15. Seleccionará el botón Eliminar.</p>	

16. Visualizará una alerta para confirmar con el mensaje “¿Está seguro que desea eliminar este Antisondeo?”, con las opciones:

- Sí: comprueba si el antisondeo no está siendo asociada a alguna conexión, si existe asociación se generará la excepción **Ex3**, en caso contrario se procede a eliminar y cerrar la alerta de confirmación.
- No: cierra la alerta de confirmación.

Prototipo

Prototipo N° 1:

Lista de Antisondeos

Q Filtros de Antisondeo [-]

Número de Consultas

Tiempo Máximo

Estado

Buscar

Limpiar

Nuevo

☰ Antisondeos Encontrados

Prototipo N° 2:

Lista de Antisondeos

Q Filtros de Antisondeo [-]

Número de Consultas

Tiempo Máximo

Estado

Buscar

Limpiar

Nuevo

☰ Antisondeos Encontrados

▶ Ver Detalle

Código	Número de Consultas	Tiempo Máximo	Estado
2	30	240	ACTIVADO

1

Prototipo N° 3:

Propiedades de Antisondeo

Datos de Antisondeo -

Código 2

Número de Consultas*

Tiempo Máximo*

Estado ACT

Flujo Normal – Nuevo Antisondeo

Escenario	El Usuario Mid ingresará al sistema y registrará la configuración de un antisondeo.
Excepciones	<ul style="list-style-type: none"> • Ex3: “Número de Consultas es un campo requerido.” • Ex4: “Tiempo Máximo es un campo requerido.”
Actor	Sistema
1. Seleccionará el botón Nuevo de la pantalla de Listado de Antisondeos. Ver Flujo Normal – Buscar Antisondeos	
	<p>2. Visualizará la pantalla del Prototipo N° 4, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: mostrará el código del antisondeo. este campo solo se visualizará después de guardar el antisondeo. • Número de Consultas: campo numérico de longitud máxima de 10 dígitos (requerido). • Tiempo Máximo: campo numérico de longitud máxima de 10 dígitos, representará tiempo máximo que una consulta se ejecuta en una conexión (requerido). • Estado: mostrará el estado del antisondeo, por defecto mostrará el estado CREADO. <p>3. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: permitirá guardar la información del antisondeo, su funcionalidad será descrita más adelante. • Regresar: retornará a la pantalla Listado de Antisondeos.
4. Ingresará los datos del antisondeo y dará clic en el botón Guardar .	
	<p>5. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresara el número de consulta del antisondeo, generará la excepción Ex4.

	<ul style="list-style-type: none"> • Si no ingresara el tiempo máximo de consulta, generará la excepción Ex5. <p>6. Si no se generara ninguna excepción, guardará la información del antisondeo, mostrará un mensaje de confirmación “La información del antisondeo ([Código]) ha sido guardada con éxito” y mostrará el Prototipo N° 5.</p> <p>7. Mostrará el botón Cambiar Estado, que permitirá modificar el estado del antisondeo, su funcionalidad será descrita en el Flujo Normal – Editar Propiedades de un Antisondeo.</p>
--	---

Prototipo

Prototipo N° 4:

Propiedades de Antisondeo

Datos de Antisondeo -

Número de Consultas*

Tiempo Máximo*

Estado CRE

Prototipo N° 5:

Propiedades de Antisondeo

Datos de Antisondeo -

Código 3

Número de Consultas*

Tiempo Máximo*

Estado CRE

Flujo Normal – Editar Propiedades de un Antisondeo

Escenario	El Usuario Mid ingresará al sistema y modificará los datos de un antisondeo. Solo podrá modificar el número de consultas, tiempo máximo y el estado del antisondeo.
Excepciones	<ul style="list-style-type: none"> • Ex6: “Número de Consultas es un campo requerido.” • Ex7: “Tiempo Máximo es un campo requerido.”
Actor	Sistema
<ol style="list-style-type: none"> 1. <incluir> Flujo Normal – Buscar Antisondeos. 2. Elegirá el Antisondeo y seleccionará el botón Ver Detalle. 	

	<p>3. Visualizará la pantalla del prototipo N° 6, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: campo no editable (requerido). • Número de Consultas: campo numérico (requerido). • Tiempo Máximo: campo numérico (requerido). • Estado: mostrará el estado del antisondeo. • Botón Editar • Botón Regresar
<p>4. Seleccionará el botón Editar.</p>	
	<p>5. Visualizará la pantalla del Prototipo N° 7, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: campo no editable (requerido). • Número de Consultas: campo numérico (requerido). • Tiempo Máximo: campo numérico (requerido). • Estado: mostrará el estado del antisondeo. • Botón Guardar. • Botón Cambiar Estado. • Botón Regresar. <p>6. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: guardará la actualización de la información del antisondeo, será descrita más adelante. • Cambiar Estado: modificará el estado del Antisondeo, será descrita más adelante • Regresar: mostrará la pantalla Listado de Antisondeos.
<p>7. Ingresará los valores a modificar. 8. Seleccionará el botón Guardar.</p>	
	<p>9. Validará los datos ingresados:</p> <ul style="list-style-type: none"> • Si no ingresara el número de consultas del antisondeo, generará la excepción Ex6. • Si no ingresara el tiempo máximo del antisondeo, generará la excepción Ex7. <p>10. Si no se generara ninguna excepción, actualizará la información del antisondeo y mostrará un mensaje de confirmación “La información del antisondeo ([Código]) ha sido guardada con éxito”.</p>
<p>11. Seleccionará el botón Cambiar Estado.</p>	
	<p>12. Mostrará la lista de procesos, de acuerdo al estado del antisondeo:</p>

	<ul style="list-style-type: none"> • ACTIVAR: cambiará el estado del antisondeo ha ACTIVADO y mostrará el mensaje: “El estado del antisondeo [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado del antisondeo fuera CREADO O DESACTIVADO. • DESACTIVAR: cambiará el estado del antisondeo ha DESACTIVADO y mostrará el mensaje: “El estado del antisondeo [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado del antisondeo fuese CREADO O ACTIVADO.
--	---

Prototipo

Prototipo N° 6:

Prototipo N° 7:

Tabla N° 4.10: CU01 - Gestionar Antisondeos.

Actor	Primario: Usuario Mid
Descripción	Se administrará un servidor que posteriormente, podrá asociarse a una conexión.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”.
Post-condiciones	Ninguna
Req. Especiales	Ninguno
Flujo Normal – Buscar Servidores	
Escenario	El actor (Usuario Mid) podrá aplicar un conjunto de condiciones

	sobre la lista de servidores visualizada.
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado una opción”. • Ex.2: Visualizará la alerta: “No se encontraron resultados”. • Ex.3: Visualizará la alerta: “El servidor está siendo usada en una conexión”
Actor	Sistema
1. Seleccionará la opción Servidores del menú izquierdo Mantenimiento.	
	<p>2. Visualizará la pantalla del Prototipo N° 1, con los siguientes controles:</p> <ul style="list-style-type: none"> • Filtros de Servidor: sección que mostrará los filtros de búsqueda de los servidores: <ul style="list-style-type: none"> o IP Servidor: campo alfanumérico con una longitud máxima de 80 dígitos. o Puerto: campo numérico con una longitud máxima de 10 dígitos. o Nombre: campo alfanumérico que representa nombre del servidor con una longitud máxima de 80 dígitos. o Estado: lista desplegable de selección simple que mostrará los estados de un servidor, sus valores serán: <ul style="list-style-type: none"> ▪ CREADO ▪ ACTIVADO ▪ DESACTIVADO <p>3. También mostrará los botones:</p> <ul style="list-style-type: none"> • Buscar: permitirá realizar la búsqueda de servidor, su funcionalidad será descrita más adelante. • Limpiar: permitirá limpiar los campos de la pantalla, colocando los valores en: <ul style="list-style-type: none"> o IP Servidor: blanco o Nombre: blanco o Estado: Seleccionar • Nuevo: permitirá crear nuevos servidores, Ver Flujo Normal – Nuevo Servidor.
4. Seleccionará o ingresará los filtros que desee aplicar a su búsqueda. 5. Seleccionará el botón Buscar .	
	<p>6. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresará ningún criterio de búsqueda, generará la excepción Ex1. <p>7. Si no se generara ninguna excepción, buscará los servidores que coincidan con los criterios de búsqueda ingresados y mostrará la sección Resultados de la Búsqueda con los registros encontrados.</p>

	<p>8. La sección Resultados de Búsqueda mostrará las siguientes columnas:</p> <ul style="list-style-type: none"> • Código: mostrará el código del servidor, los resultados obtenidos estarán ordenados según esta columna. • IP Servidor: Ip del servidor de base de datos. • Puerto: Puerto del servidor de base de datos. • Nombre: Mostrará el nombre del servidor. • Estado: mostrará el estado del servidor. <p>Ver Prototipo N° 2</p> <p>9. Si no se encontraran resultados para la búsqueda, mostrará el mensaje “No se encontraron resultados”, dentro de la tabla de resultados.</p> <p>10. También se muestra los botones:</p> <ul style="list-style-type: none"> • Ver Detalle: Deshabilitada, permitirá ver el detalle del servidor que será descrita más adelante. • Eliminar: Deshabilitada, permitirá borrar el servidor que será descrita más adelante.
<p>11. Elegirá un servidor de la sección de resultados.</p> <p>12. Seleccionará el botón Ver Detalle.</p>	
	<p>13. Si no seleccionara un servidor de la sección de resultados, los botones Ver Detalle y Eliminar seguirán deshabilitadas.</p> <p>14. Visualizará la pantalla del prototipo N° 3, con los siguientes controles:</p> <ul style="list-style-type: none"> • Propiedades de Servidor: sección que mostrará las propiedades del servidor seleccionado: <ul style="list-style-type: none"> o Código: código del servidor, campo no editable. o IP Servidor: campo alfanumérico, campo deshabilitado. o Puerto: campo numérico, campo deshabilitado. o Nombre: campo deshabilitado. o Estado: campo no editable. • También mostrará los botones: <ul style="list-style-type: none"> o Editar: permitirá editar las propiedades de un servidor, ver Flujo Normal – Editar Propiedades de un Servidor. o Regresar: regresará a la pantalla de

	Listado de Servidores.
15.Elegirá un servidor de la sección de resultados. 16.Seleccionará el botón Eliminar .	
	17. Visualizará una alerta para confirmar con el mensaje “¿Está seguro que desea eliminar este Servidor?”, con las opciones: <ul style="list-style-type: none"> • Si: comprueba si el servidor no esté siendo asociada a alguna conexión, si existe asociación se generará la excepción Ex3, en caso contrario se procede a eliminar y cerrar la alerta de confirmación. • No: cierra la alerta de confirmación.

Prototipo

Prototipo N° 1:

Lista de Servidores

-

Q Filtros de Servidor

IP Servidor

Nombre

Estado Seleccionar

🔍 Buscar

🧼 Limpiar

➕ Nuevo

☰ Servidores Encontrados

Prototipo N° 2:

Lista de Servidores

-

Q Filtros de Servidor

IP Servidor

Nombre

Estado ACTIVADO

🔍 Buscar

🧼 Limpiar

➕ Nuevo

☰ Servidores Encontrados

▶ Ver Detalle
🗑 Eliminar

Código	IP Servidor	Puerto	Nombre	Estado
1	localhost	5432	ACC-SERVER	ACTIVADO
3	180.169.1.2	1432	SQL_SERVER_2012	ACTIVADO
4	192.168.1.3	1521	ACC_SERVER_ORACLE	ACTIVADO

Prototipo N° 3:
Propiedades de Servidores

Datos de Servidor -

Código	1	
IP Servidor*	localhost	▶ Editar
Puerto*	5432	◀ Regresar
Nombre*	ACC-SERVER	
Estado	ACT	

Flujo Normal – Nuevo Servidor

Escenario	El Usuario Mid ingresará al sistema y registrará la configuración de un servidor.
------------------	---

Excepciones	<ul style="list-style-type: none"> • Ex3: “IP Servidor es un campo requerido.” • Ex4: “Nombre es un campo requerido.”
--------------------	---

Actor	Sistema
-------	---------

<p>1. Seleccionará el botón Nuevo de la pantalla de Listado de Servidores. Ver Flujo Normal – Buscar Servidores</p>	
---	--

	<p>2. Visualizará la pantalla del Prototipo N° 4, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: mostrará el código del servidor, este campo solo se visualizará después de guardar el servidor. • IP Servidor: campo alfanumérico de longitud máxima de 80 dígitos (requerido). • Puerto: campo numérico de longitud máxima de 10 dígitos, representará el puerto del servidor de base de datos. • Nombre: campo alfanumérico de longitud máxima de 80 dígitos, representará el nombre del servidor de base de datos (requerido). • Estado: mostrará el estado del servidor, por defecto mostrará el estado CREADO. <p>3. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: permitirá guardar la información del servidor, su funcionalidad será descrita más adelante. • Regresar: retornará a la pantalla Listado de Servidores.
--	---

<p>4. Ingresará los datos del servidor y dará clic en el botón Guardar.</p>	
--	--

	<p>5. Validará los datos ingresados</p>
--	---

	<ul style="list-style-type: none"> • Si no ingresara el IP del servidor, generará la excepción Ex4. • Si no ingresara el nombre del, generará la excepción Ex5. <p>6. Si no se generara ninguna excepción, guardará la información del servidor, mostrará un mensaje de confirmación “La información del servidor ([Código]) ha sido guardada con éxito” y mostrará el Prototipo N° 5.</p> <p>7. Mostrará el botón Cambiar Estado, que permitirá modificar el estado del servidor, su funcionalidad será descrita en el Flujo Normal – Editar Propiedades de un Servidor.</p>
--	---

Prototipo

Prototipo N° 4:

Propiedades de Servidores

Datos de Servidor

IP Servidor*

Puerto*

Nombre*

Estado CRE

Guardar

Regresar

Prototipo N° 5:

Propiedades de Servidores

Datos de Servidor

Código 1

IP Servidor* localhost

Puerto* 5432

Nombre* ACC-SERVER

Estado ACT

Cambiar Estado

Guardar

Regresar

Flujo Normal – Editar Propiedades de un Servidor

Escenario	El Usuario Mid ingresará al sistema y modificará los datos de un servidor. Solo podrá modificar el ip servidor, nombre y el estado del servidor.
Excepciones	<ul style="list-style-type: none"> • Ex6: “IP Servidor es un campo requerido.” • Ex7: “Nombre es un campo requerido.”
Actor	Sistema
<ol style="list-style-type: none"> 1. <incluir> Flujo Normal – Buscar Servidores. 2. Elegirá el servidor y seleccionará el botón Ver Detalle. 	

	<p>3. Visualizará la pantalla del prototipo N° 6, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: campo no editable (requerido). • IP Servidor: campo alfanumérico (requerido). • Puerto: campo numérico (requerido). • Nombre: campo alfanumérico (requerido). • Estado: mostrará el estado del servidor. • Botón Editar • Botón Regresar
<p>4. Seleccionará el botón Editar.</p>	
	<p>5. Visualizará la pantalla del Prototipo N° 7, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: campo no editable (requerido). • IP Servidor: campo alfanumérico (requerido). • Puerto: campo numérico (requerido). • Nombre: campo alfanumérico (requerido). • Estado: mostrará el estado del servidor. • Botón Guardar. • Botón Cambiar Estado. • Botón Regresar. <p>6. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: guardará la actualización de la información del servidor, será descrita más adelante. • Cambiar Estado: modificará el estado del Servidor, será descrita más adelante • Regresar: mostrará la pantalla Listado de Servidores.
<p>7. Ingresará los valores a modificar. 8. Seleccionará el botón Guardar.</p>	
	<p>9. Validará los datos ingresados:</p> <ul style="list-style-type: none"> • Si no ingresara el ip del servidor, generará la excepción Ex6. • Si no ingresara el nombre del servidor, generará la excepción Ex7. <p>10. Si no se generara ninguna excepción, actualizará la información del servidor y mostrará un mensaje de confirmación “La información del servidor ([Código]) ha sido guardada con éxito”.</p>
<p>11. Seleccionará el botón Cambiar Estado.</p>	
	<p>12. Mostrará la lista de procesos, de acuerdo al estado del servidor:</p>

	<ul style="list-style-type: none"> • ACTIVAR: cambiará el estado del servidor a ACTIVADO y mostrará el mensaje: “El estado del servidor [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado del servidor fuera CREADO O DESACTIVADO. • DESACTIVAR: cambiará el estado del servidor a DESACTIVADO y mostrará el mensaje: “El estado del servidor [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado del servidor fuese CREADO O ACTIVADO.
--	---

Prototipo

Prototipo N° 6:

Propiedades de Servidores

Datos de Servidor

Código: 1

IP Servidor*: localhost

Puerto*: 5432

Nombre*: ACC-SERVER

Estado: ACT

Buttons: Editar, Regresar

Prototipo N° 7:

Propiedades de Servidores

Datos de Servidor

Código: 1

IP Servidor*: localhost

Puerto*: 5432

Nombre*: ACC-SERVER

Estado: ACT

Buttons: Cambiar Estado (DESACTIVAR), Regresar

Tabla N° 4.11: CU02 - Gestionar Servidores.

Actor	Primario: Usuario Mid
Descripción	Se administrará un los datos de una empresa mid que posteriormente, podrá asociarse a las conexiones y queries.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”.
Post-condiciones	Ninguna
Req. Especiales	Ninguno
Flujo Normal – Buscar Empresas Mid	
Escenario	El actor (Usuario Mid) podrá aplicar un conjunto de condiciones sobre la lista de empresas mid visualizada.

Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado una opción”. • Ex.2: Visualizará la alerta: “No se encontraron resultados”. • Ex.3: Visualizará la alerta: “La empresa mid está siendo usada en una conexión”
Actor	Sistema
1. Seleccionará la opción Empresas Mid del menú izquierdo Mantenimiento.	
	<p>2. Visualizará la pantalla del Prototipo N° 1, con los siguientes controles:</p> <ul style="list-style-type: none"> • Filtros de Empresa Mid: sección que mostrará los filtros de búsqueda de las empresas mid: <ul style="list-style-type: none"> o Empresa: lista desplegable de selección simple que mostrará empresa de los módulos de seguridad. o Nombre: campo alfanumérico que representa nombre de la empresa mid con una longitud máxima de 80 dígitos. o Estado: lista desplegable de selección simple que mostrará los estados de una empresa mid, sus valores serán: <ul style="list-style-type: none"> ▪ CREADO ▪ ACTIVADO ▪ DESACTIVADO <p>3. También mostrará los botones:</p> <ul style="list-style-type: none"> • Buscar: permitirá realizar la búsqueda de empresa mid, su funcionalidad será descrita más adelante. • Limpiar: permitirá limpiar los campos de la pantalla, colocando los valores en: <ul style="list-style-type: none"> o Empresa: Seleccionar o Nombre: blanco o Estado: Seleccionar • Nuevo: permitirá crear nuevas empresa mid, ver Flujo Normal – Nueva Empresa Mid.
4. Seleccionará o ingresará los filtros que desee aplicar a su búsqueda.	
5. Seleccionará el botón Buscar.	
	<p>6. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresará ningún criterio de búsqueda, generará la excepción Ex1. <p>7. Si no se generara ninguna excepción, buscará las empresas mid que coincidan con los criterios de búsqueda ingresados y mostrará la sección Resultados de la Búsqueda con los registros encontrados.</p>

	<p>8. La sección Resultados de Búsqueda mostrará las siguientes columnas:</p> <ul style="list-style-type: none"> • Código: mostrará el código de la empresa mid, los resultados obtenidos estarán ordenados según esta columna. • Empresa: mostrará la empresa al cual fue asocia la empresa mid. • Nombre: mostrará el nombre de la empresa mid. • Estado: mostrará el estado de la empresa mid. <p>Ver Prototipo N° 2</p> <p>9. Si no se encontraran resultados para la búsqueda, mostrará el mensaje “No se encontraron resultados”, dentro de la tabla de resultados.</p> <p>10. También se muestra los botones:</p> <ul style="list-style-type: none"> • Ver Detalle: Deshabilitada, permitirá ver el detalle de la empresa mid que será descrita más adelante. • Eliminar: Deshabilitada, permitirá borrar la empresa mid que será descrita más adelante.
<p>11. Elegirá una empresa mid de la sección de resultados.</p> <p>12. Seleccionará el botón Ver Detalle.</p>	
	<p>13. Si no seleccionara una empresa mid de la sección de resultados, los botones Ver Detalle y Eliminar seguirán deshabilitadas.</p> <p>14. Visualizará la pantalla del prototipo N° 3, con los siguientes controles:</p> <ul style="list-style-type: none"> • Propiedades de Empresa Mid: sección que mostrará las propiedades de la empresa mid seleccionado: <ul style="list-style-type: none"> o Código: código del servidor, campo no editable. o Empresa: lista de selección simple, campo deshabilitado. o Nombre: campo deshabilitado. o Estado: campo no editable. • También mostrará los botones: <ul style="list-style-type: none"> o Editar: permitirá editar las propiedades de una empresa mid, ver Flujo Normal – Editar Propiedades de una Empresa Mid. o Regresar: regresará a la pantalla de Listado de Empresas Mid.

<p>15. Elegirá una empresa mid de la sección de resultados.</p> <p>16. Seleccionará el botón Eliminar.</p>	<p>17. Visualizará una alerta para confirmar con el mensaje “¿Está seguro que desea eliminar esta Empresa Mid?”, con las opciones:</p> <ul style="list-style-type: none"> • Si: comprueba si la empresa mid no esté siendo asociada a alguna conexión o query, si existe asociación se generará la excepción Ex3, en caso contrario se procede a eliminar y cerrar la alerta de confirmación. • No: cierra la alerta de confirmación.
---	--

Prototipo

Prototipo N° 1:

Lista de Empresas Mid

Filtros de Empresa Mid -

Empresa* 🔍 Buscar

Nombre*

Estado 🧼 Limpiar

Nuevo

☰ Empresas Mid Encontrados

Prototipo N° 2:

Lista de Empresas Mid

Filtros de Empresa Mid -

Empresa* 🔍 Buscar

Nombre*

Estado 🧼 Limpiar

Nuevo

☰ Empresas Mid Encontrados

▶ Ver Detalle
🗑 Eliminar

Código	Empresa	Nombre	Estado
1	Consultoria y Asesoría en Tecnología Prueba Empresa Prueba	CONASTEC	ACTIVADO

Prototipo N° 3:

Propiedades de Empresa Mid

Flujo Normal – Nueva Empresa Mid

Escenario	El Usuario Mid ingresará al sistema y registrará la configuración de una empresa mid.
Excepciones	<ul style="list-style-type: none"> • Ex3: “Empresa es un campo requerido.” • Ex4: “Nombre es un campo requerido.”
Actor	Sistema
1. Seleccionará el botón Nuevo de la pantalla de Listado de Empresas Mid. Ver Flujo Normal – Buscar Empresas Mid	
	2. Visualizará la pantalla del Prototipo N° 4 , con los siguientes controles: <ul style="list-style-type: none"> • Código: mostrará el código de la empresa mid, este campo solo se visualizará después de guardar la empresa mid. • Empresa: lista de selección simple, (requerido). • Nombre: campo alfanumérico de longitud máxima de 80 dígitos, representará el nombre de la empresa mid, (requerido). • Estado: mostrará el estado de la empresa mid, por defecto mostrará el estado CREADO. 3. También mostrará las opciones: <ul style="list-style-type: none"> • Guardar: permitirá guardar la información de la empresa mid, su funcionalidad será descrita más adelante. • Regresar: retornará a la pantalla Listado de Empresas Mid.
4. Ingresará los datos de la empresa mid y dará clic en el botón Guardar .	
	5. Validará los datos ingresados <ul style="list-style-type: none"> • Si no se seleccionará la empresa, generará la excepción Ex4. • Si no ingresara el nombre de la empresa mid, generará la excepción Ex5.

	<p>6. Si no se generara ninguna excepción, guardará la información de la empresa mid, mostrará un mensaje de confirmación “La información de la empresa mid ([Código]) ha sido guardada con éxito” y mostrará el Prototipo N° 5.</p> <p>7. Mostrará el botón Cambiar Estado, que permitirá modificar el estado de la empresa mid, su funcionalidad será descrita en el Flujo Normal – Editar Propiedades de una Empresa Mid.</p>
--	--

Prototipo

Prototipo N° 4:

Propiedades de Empresa Mid

Datos de Empresa Mid -

Empresa* Guardar

Nombre*

Estado Regresar

Prototipo N° 5:

Propiedades de Empresa Mid

Datos de Empresa Mid -

Código Cambiar Estado

Empresa* Guardar

Nombre* Regresar

Estado

Flujo Normal – Editar Propiedades de una Empresa Mid

Escenario	El Usuario Mid ingresará al sistema y modificará los datos de una empresa mid. Solo podrá modificar la empresa, nombre y el estado de la empresa.
Excepciones	<ul style="list-style-type: none"> • Ex6: “Empresa es un campo requerido.” • Ex7: “Nombre es un campo requerido.”
Actor	Sistema
<ol style="list-style-type: none"> 1. <incluir> Flujo Normal – Buscar Empresas Mid. 2. Elegirá una empresa mid y seleccionará el botón Ver Detalle. 	
	<ol style="list-style-type: none"> 3. Visualizará la pantalla del prototipo N° 6, con los siguientes controles: <ul style="list-style-type: none"> • Código: campo no editable (requerido). • Empresa: campo de selección simple

	<p>(requerido).</p> <ul style="list-style-type: none"> • Nombre: campo alfanumérico (requerido). • Estado: mostrará el estado de la empresa mid. • Botón Editar • Botón Regresar
4. Seleccionará el botón Editar .	
	<p>5. Visualizará la pantalla del Prototipo N° 7, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: campo no editable (requerido). • Empresa: campo de selección simple (requerido). • Nombre: campo alfanumérico (requerido). • Estado: mostrará el estado de la empresa mid. • Botón Guardar. • Botón Cambiar Estado. • Botón Regresar. <p>6. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: guardará la actualización de la información de la empresa mid, será descrita más adelante. • Cambiar Estado: modificará el estado de la empresa mid, será descrita más adelante • Regresar: mostrará la pantalla Listado de Empresas Mid.
7. Ingresará los valores a modificar.	
8. Seleccionará el botón Guardar .	
	<p>9. Validará los datos ingresados:</p> <ul style="list-style-type: none"> • Si no se seleccionará la empresa, generará la excepción Ex6. • Si no ingresara el nombre de la empresa mid, generará la excepción Ex7. <p>10. Si no se generara ninguna excepción, actualizará la información de la empresa mid y mostrará un mensaje de confirmación “La información de la empresa mid ([Código]) ha sido guardada con éxito”.</p>
11. Seleccionará el botón Cambiar Estado .	
	<p>12. Mostrará la lista de procesos, de acuerdo al estado de la empresa mid:</p> <ul style="list-style-type: none"> • ACTIVAR: cambiará el estado de la empresa mid a ACTIVADO y mostrará el mensaje: “El estado de la empresa mid [código] ha sido modificado con éxito.”

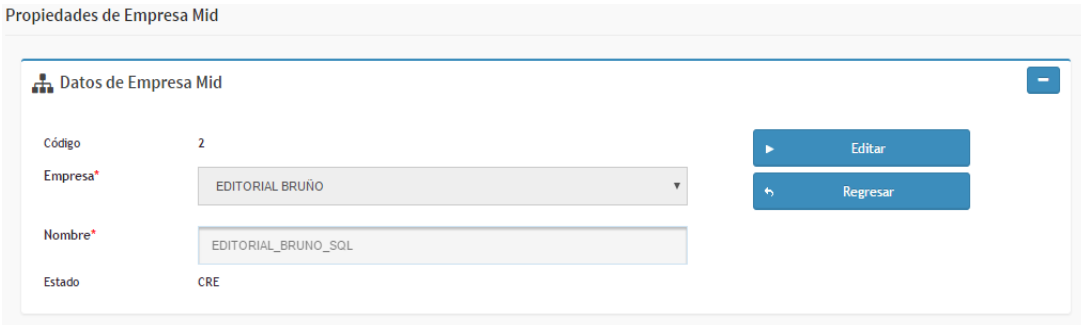
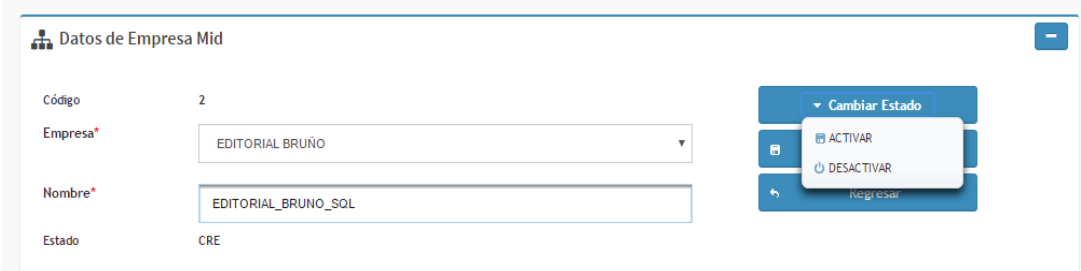
	<p>Esto solo se realizará, si el estado de la empresa mid fuera CREADO O DESACTIVADO.</p> <ul style="list-style-type: none"> • DESACTIVAR: cambiará el estado de la empresa mid a DESACTIVADO y mostrará el mensaje: “El estado de la empresa mid [código] ha sido modificado con éxito.” <p>Esto solo se realizará, si el estado de la empresa mid fuese CREADO O ACTIVADO.</p>
Prototipo	
<p>Prototipo N° 6:</p> <p>Propiedades de Empresa Mid</p> 	
<p>Prototipo N° 7:</p> <p>Propiedades de Empresa Mid</p> 	

Tabla N° 4.12: CU03 - Gestionar Empresas Mid.

Actor	Primario: Usuario Mid
Descripción	Se administrará un query sql que posteriormente, podrá asociarse a una empresa mid.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”.
Post-condiciones	Ninguna
Req. Especiales	Ninguno
Flujo Normal – Buscar Queries	
Escenario	El actor (Usuario Mid) podrá aplicar un conjunto de condiciones sobre la lista de queries sql visualizada.
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado una opción”. • Ex.2: Visualizará la alerta: “No se encontraron resultados”.

	<ul style="list-style-type: none"> • Ex.3: Visualizará la alerta: “El Query Sql está siendo usada por una empresa”
Actor	Sistema
1. Seleccionará la opción Queries del menú izquierdo Mantenimiento.	
	<p>2. Visualizará la pantalla del Prototipo N° 1, con los siguientes controles:</p> <ul style="list-style-type: none"> • Filtros de Queries Sql: sección que mostrará los filtros de búsqueda de los queries sql: <ul style="list-style-type: none"> o Código Query: campo alfanumérico con una longitud máxima de 80 dígitos. o Alias: campo alfanumérico que representa nombre del query sql con una longitud máxima de 80 dígitos. o Estado: lista desplegable de selección simple que mostrará los estados de un query sql, sus valores serán: <ul style="list-style-type: none"> ▪ CREADO ▪ ACTIVADO ▪ DESACTIVADO <p>3. También mostrará los botones:</p> <ul style="list-style-type: none"> • Buscar: permitirá realizar la búsqueda de query sql, su funcionalidad será descrita más adelante. • Limpiar: permitirá limpiar los campos de la pantalla, colocando los valores en: <ul style="list-style-type: none"> o Código Query: blanco o Alias: blanco o Estado: Seleccionar • Nuevo: permitirá crear nuevos queries, ver Flujo Normal – Nuevo Query Sql.
4. Seleccionará o ingresará los filtros que desee aplicar a su búsqueda.	
5. Seleccionará el botón Buscar .	
	<p>6. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresará ningún criterio de búsqueda, generará la excepción Ex1. <p>7. Si no se generara ninguna excepción, buscará los queries sql que coincidan con los criterios de búsqueda ingresados y mostrará la sección Resultados de la Búsqueda con los registros encontrados.</p> <p>8. La sección Resultados de Búsqueda mostrará las siguientes columnas:</p> <ul style="list-style-type: none"> • Código Query: mostrará el código del Query Sql, los resultados obtenidos

	<p>estarán ordenados según esta columna.</p> <ul style="list-style-type: none"> • Alias: Nombre del Query Sql. • Instrucción: Mostrará la instrucción Sql. • Estado: mostrará el estado del Query Sql. <p>Ver Prototipo N° 2</p> <p>9. Si no se encontraran resultados para la búsqueda, mostrará el mensaje “No se encontraron resultados”, dentro de la tabla de resultados.</p> <p>10. También se muestra los botones:</p> <ul style="list-style-type: none"> • Ver Detalle: Deshabilitada, permitirá ver el detalle del query sql que será descrita más adelante. • Eliminar: Deshabilitada, permitirá borrar el query sql que será descrita más adelante.
<p>11. Elegirá un query sql de la sección de resultados.</p> <p>12. Seleccionará el botón Ver Detalle.</p>	
	<p>13. Si no seleccionara un query sql de la sección de resultados, los botones Ver Detalle y Eliminar seguirán deshabilitadas.</p> <p>14. Visualizará la pantalla del prototipo N° 3, con los siguientes controles:</p> <ul style="list-style-type: none"> • Propiedades de Query Sql: sección que mostrará las propiedades del query sql seleccionado: <ul style="list-style-type: none"> o Código Query: código del query sql, campo no editable. o Alias: campo alfanumérico, campo deshabilitado. o Instrucción: campo deshabilitado. o Estado: campo no editable. • También mostrará los botones: <ul style="list-style-type: none"> o Editar: permitirá editar las propiedades de un query sql, ver Flujo Normal – Editar Propiedades de un Query Sql. o Regresar: regresará a la pantalla de Listado de Queries Sql.
<p>15. Elegirá un query sql de la sección de resultados.</p> <p>16. Seleccionará el botón Eliminar.</p>	
	<p>17. Visualizará una alerta para confirmar con el mensaje “¿Está seguro que desea eliminar este Query Sql?”, con las opciones:</p> <ul style="list-style-type: none"> • Si: comprueba si el query sql no esté siendo asociada a alguna empresa mid, si

existe asociación se generará la excepción **Ex3**, en caso contrario se procede a eliminar y cerrar la alerta de confirmación.

- No: cierra la alerta de confirmación.

Prototipo

Prototipo N° 1:

Lista de Queries Sql

Filtros de Query Sql

Código Query

Alias

Estado

Buscar

Limpiar

Nuevo

Queries Sql Encontrados

Prototipo N° 2:

Lista de Queries Sql

Filtros de Query Sql

Código Query

Alias

Estado

Buscar

Limpiar

Nuevo

Queries Sql Encontrados

Ver Detalle Eliminar

Código	Alias	Instrucción	Estado
CONASTEC_SAS_1	INSETAR USUARIO	SELECT GETDATE()	ACTIVADO
CONASTEC_SAS_2	SELECCIONAR USUARIO	SELECT * FROM USUARIO	ACTIVADO
CONASTEC_SAS_3	ELIMINAR UAUARIO	DELETE FROM USUARIO WERE ID=1	ACTIVADO

Prototipo N° 3:

Propiedades de Query Sql

Datos de Query Sql

Código Query*

Alias*

Instrucción*

Descripción*

Estado

Editar

Regresar

Flujo Normal – Nuevo Query Sql	
Escenario	El Usuario Mid ingresará al sistema y registrará la configuración de un query sql.
Excepciones	<ul style="list-style-type: none"> • Ex4: “Código Query es un campo requerido.” • Ex5: “Alias es un campo requerido.” • Ex6: “Instrucción es un campo requerido.” • Ex7: “Descripción es un campo requerido.”
Actor	Sistema
1. Seleccionará el botón Nuevo de la pantalla de Listado de Queries Sql. Ver Flujo Normal – Buscar Queries Sql	
	<p>2. Visualizará la pantalla del Prototipo N° 4, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código Query: mostrará el código del query sql, este campo solo se deshabilitará después de guardar el query. • Alias: campo alfanumérico de longitud máxima de 60 dígitos (requerido). • Instrucción: campo alfanumérico de longitud máxima de 800 dígitos, representará la instrucción sql (requerido). • Estado: mostrará el estado del query sql, por defecto mostrará el estado CREADO. <p>3. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: permitirá guardar la información del query sql, su funcionalidad será descrita más adelante. • Regresar: retornará a la pantalla Listado de Queries Sql.
4. Ingresará los datos del query sql y dará clic en el botón Guardar .	
	<p>5. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresara el código query, generará la excepción Ex4. • Si no ingresara el alias del, generará la excepción Ex5. • Si no ingresara la instrucción, generará la excepción Ex6. • Si no ingresara la descripción del query sql, generará la excepción Ex7. <p>6. Si no se generara ninguna excepción, guardará la información del query sql, mostrará un mensaje de confirmación “La información del query sql ([Código]) ha</p>

“ sido guardada con éxito” y mostrará el **Prototipo N° 5**.

- Mostrará el botón Cambiar Estado, que permitirá modificar el estado del query sql, su funcionalidad será descrita en el **Flujo Normal – Editar Propiedades de un Query Sql**.

Prototipo

Prototipo N° 4:

Propiedades de Query Sql

Datos de Query Sql

Código Query*

Alias*

Instrucción*

Descripción*

Estado CRE

Guardar

Regresar

Prototipo N° 5:

Propiedades de Query Sql

Datos de Query Sql

Código Query* CONASTEC_SAS_1

Alias* INSETAR USUARIO

Instrucción* SELECT GETDATE()

Descripción* SELECCIONAR LA FECHA

Estado ACT

Cambiar Estado

Guardar

Regresar

Flujo Normal – Editar Propiedades de un Query Sql

Escenario	El Usuario Mid ingresará al sistema y modificará los datos de un query sql. Solo podrá modificar el alias, instrucción, descripción y el estado del query sql.
Excepciones	<ul style="list-style-type: none"> • Ex8: “Código Query es un campo requerido.” • Ex9: “Alias es un campo requerido.” • Ex10: “Instrucción es un campo requerido.” • Ex11: “Descripción es un campo requerido.”
Actor	Sistema

<p>1. <incluir> Flujo Normal – Buscar Queries Sql.</p> <p>2. Elegirá el Query Sql y seleccionará el botón Ver Detalle.</p>	
	<p>3. Visualizará la pantalla del prototipo N° 6, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código Query: campo no editable (requerido). • Alias: campo alfanumérico (requerido). • Instrucción: campo alfanumérico (requerido). • Descripción: campo alfanumérico (Requerido). • Estado: mostrará el estado del query sql. • Botón Editar • Botón Regresar
<p>4. Seleccionará el botón Editar.</p>	
	<p>5. Visualizará la pantalla del Prototipo N° 7, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código Query: campo no editable (requerido). • Alias: campo alfanumérico (requerido). • Instrucción: campo alfanumérico (requerido). • Descripción: campo alfanumérico (Requerido). • Estado: mostrará el estado del query sql. • Botón Guardar • Botón Cambiar Estado • Botón Regresar. <p>6. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: guardará la actualización de la información del query sql, será descrita más adelante. • Cambiar Estado: modificará el estado del Query Sql, será descrita más adelante • Regresar: mostrará la pantalla Listado de Queries Sql.
<p>7. Ingresará los valores a modificar.</p> <p>8. Seleccionará el botón Guardar.</p>	
	<p>9. Validará los datos ingresados:</p> <ul style="list-style-type: none"> • Si no ingresara el alias del, generará la excepción Ex9. • Si no ingresara la instrucción, generará la excepción Ex10. • Si no ingresara la descripción del query sql,

	<p>generará la excepción Ex11.</p> <p>10. Si no se generara ninguna excepción, actualizará la información del query sql y mostrará un mensaje de confirmación “La información del query sql ([Código]) ha sido guardada con éxito”.</p>
11. Seleccionará el botón Cambiar Estado .	
	<p>12. Mostrará la lista de procesos, de acuerdo al estado del query sql:</p> <ul style="list-style-type: none"> • ACTIVAR: cambiará el estado del query sql a ACTIVADO y mostrará el mensaje: “El estado del query sql [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado del query fuera CREADO O DESACTIVADO. • DESACTIVAR: cambiará el estado del query sql a DESACTIVADO y mostrará el mensaje: “El estado del query sql [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado del query sql fuese CREADO O ACTIVADO.

Prototipo

Prototipo N° 6:

Propiedades de Query Sql

Datos de Query Sql

Código Query* CONASTEC_SAS_1

Alias* INSETAR USUARIO

Instrucción* SELECT GETDATE()

Descripción* SELECCIONAR LA FECHA

Estado ACT

Editar

Regresar

Prototipo N° 7:

Propiedades de Query Sql

Datos de Query Sql

Código Query* CONASTEC_SAS_1

Alias* INSETAR USUARIO

Instrucción* SELECT GETDATE()

Descripción* SELECCIONAR LA FECHA

Estado ACT

Cambiar Estado

DESACTIVAR

Regresar

Tabla N° 4.13: CU04 - Gestionar Queries Sql.

Actor	Primario: Usuario Mid
Descripción	Se administrará una conexión que posteriormente, podrá asociarse a una empresa mid.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”.
Post-condiciones	Ninguna
Req. Especiales	Ninguno
Flujo Normal – Buscar Conexiones	
Escenario	El actor (Usuario Mid) podrá aplicar un conjunto de condiciones sobre la lista de conexiones visualizada.
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado una opción”. • Ex.2: Visualizará la alerta: “No se encontraron resultados”. • Ex.3: Visualizará la alerta: “La Conexión está siendo usada por una empresa mid”
Actor	Sistema
1. Seleccionará la opción Conexiones del menú izquierdo Mantenimiento.	
	<p>2. Visualizará la pantalla del Prototipo N° 1, con los siguientes controles:</p> <ul style="list-style-type: none"> • Filtros de Conexión: sección que mostrará los filtros de búsqueda de las conexiones. <ul style="list-style-type: none"> o Código: campo alfanumérico con una longitud máxima de 20 dígitos. o Alias: campo alfanumérico que representa nombre de la conexión con una longitud máxima de 60 dígitos. o Servidor: lista desplegable de selección simple que mostrará los servidores de base de datos activos. o Estado: lista desplegable de selección simple que mostrará los estados de una conexión, sus valores serán: <ul style="list-style-type: none"> ▪ CREADO ▪ ACTIVADO ▪ DESACTIVADO <p>3. También mostrará los botones:</p> <ul style="list-style-type: none"> • Buscar: permitirá realizar la búsqueda de conexión, su funcionalidad será descrita más adelante. • Limpiar: permitirá limpiar los campos de la pantalla, colocando los valores en: <ul style="list-style-type: none"> o Código: blanco o Alias: blanco

	<p>o Estado: Seleccionar</p> <ul style="list-style-type: none"> • Nuevo: permitirá crear nuevos conexión, ver Flujo Normal – Nuevo Conexión.
<p>4. Seleccionará o ingresará los filtros que desee aplicar a su búsqueda.</p> <p>5. Seleccionará el botón Buscar.</p>	
	<p>6. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresará ningún criterio de búsqueda, generará la excepción Ex1. <p>7. Si no se generara ninguna excepción, buscará las conexiones sql que coincidan con los criterios de búsqueda ingresados y mostrará la sección Resultados de la Búsqueda con los registros encontrados.</p> <p>8. La sección Resultados de Búsqueda mostrará las siguientes columnas:</p> <ul style="list-style-type: none"> • Código: mostrará el código de la conexión, los resultados obtenidos estarán ordenados según esta columna. • Alias: Nombre de la conexión. • Test Sql: Mostrará la instrucción Sql de prueba. • IP Servidor: Mostrará el Ip del servidor. • Base de Datos: Mostrará la base de datos física. • Puerto: muestra el puerto del servidor de base de datos. • Estado: mostrará el estado de la conexión. <p>Ver Prototipo N° 2</p> <p>9. Si no se encontraran resultados para la búsqueda, mostrará el mensaje “No se encontraron resultados”, dentro de la tabla de resultados.</p> <p>10. También se muestra los botones:</p> <ul style="list-style-type: none"> • Ver Detalle: Deshabilitada, permitirá ver el detalle de la conexión que será descrita más adelante. • Eliminar: Deshabilitada, permitirá borrar la conexión que será descrita más adelante.
<p>11. Elegirá una conexión de la sección de resultados.</p> <p>12. Seleccionará el botón Ver Detalle.</p>	
	<p>13. Si no seleccionara una conexión de la</p>

	<p>sección de resultados, los botones Ver Detalle y Eliminar seguirán deshabilitadas.</p> <p>14. Visualizará la pantalla del prototipo N° 3 con los siguientes controles:</p> <ul style="list-style-type: none"> • Propiedades de Conexión: sección que mostrará las propiedades de la conexión seleccionada: <ul style="list-style-type: none"> o Código: código de la conexión, campo no editable. o Alias: campo alfanumérico, campo deshabilitado, campo deshabilitado. o Conexiones Máximas: Número máximo de conexiones aceptadas, campo deshabilitado. o Conexiones Mínimas: Número mínimo de conexiones aceptadas, campo deshabilitado. o Test Sql: Instrucción Sql de prueba para la conexión, campo deshabilitado. o Usuario: Usuario de la base de datos. o Password: Clave de la base de datos, campo deshabilitado. o Servidor: Servidor de la base de datos, campo deshabilitado. o Base de datos: instancia de base de datos, campo deshabilitado. o Driver Class: Driver de conexión a la base de datos, campo deshabilitado. o Driver Url: Url de conexión de base de datos, campo deshabilitado. o Estado: campo no editable. • También mostrará los botones: <ul style="list-style-type: none"> o Editar: permitirá editar las propiedades de una conexión, ver Flujo Normal – Editar Propiedades de una Conexión. o Regresar: regresará a la pantalla de Listado de Queries Sql.
<p>15. Elegirá una conexión de la sección de resultados.</p> <p>16. Seleccionará el botón Eliminar.</p>	
	<p>17. Visualizará una alerta para confirmar con el mensaje “¿Está seguro que desea eliminar este Conexión?”, con las opciones:</p>

- Si: comprueba si la conexión no esté siendo asociada a alguna empresa mid, si existe asociación se generará la excepción **Ex3**, en caso contrario se procede a eliminar y cerrar la alerta de confirmación.
- No: cierra la alerta de confirmación.

Prototipo

Prototipo N° 1:

Lista de Conexiones

Q Filtros de Conexión -

Código	<input type="text"/>	<input type="button" value="Buscar"/>
Alias	<input type="text"/>	<input type="button" value="Limpiar"/>
Servidor	Seleccionar ▼	<input type="button" value="Nuevo"/>
Estado	Seleccionar ▼	

☰ Conexiones Encontrados

Prototipo N° 2:

Lista de Conexiones

Q Filtros de Conexión -

Código	<input type="text"/>	<input type="button" value="Buscar"/>
Alias	<input type="text"/>	<input type="button" value="Limpiar"/>
Servidor	ACC-SERVER - localhost : 5432 ▼	<input type="button" value="Nuevo"/>
Estado	Seleccionar ▼	

☰ Conexiones Encontrados

Código	Alias	Test Sql	IP Servidor	Puerto	Base de Datos	Estado
CXN_SPCMIDSQL	CONEXION PARA SPC CON MIDDLEWARE	SELECT GETDATE()	localhost	5432	DESA_SSIIT_BASE_PRUEBAMID	ACTIVADO
CONASTEC_1	SSIIT_MID_POSTGRES	SELECT NOW()	localhost	5432	DESA_SSIIT_BASE_PRUEBAMID	ACTIVADO

Prototipo N° 3:

Propiedades de Conexión

Datos de Conexión

Código* CXN_SPCMIDSQLS_1 ▶ Editar

Alias* CONEXION PARA SPC CON MIDDLEWARE ◀ Regresar

Conexiones Maximas* 100

Conexiones Mínimas* 20

Test Sql* SELECT GETDATE()

Usuario* mid

Password* 654321

Antisondeo* 240 (ms) ▼

Servidor* ACC-SERVER - localhost : 5432 ▼

Base de Datos* DESA_SSIT_BASE_PRUEBAMID

Driver Class* com.microsoft.sqlserver.jdbc.SQLServerDriver

Driver Uri* jdbc:sqlserver://<host>:databaseName=<database>;user=<user>;password=<password>;

Estado ACT

Flujo Normal – Nueva Conexión

Escenario	El Usuario Mid ingresará al sistema y registrará la configuración de una conexión.
Excepciones	• Ex4: “[Nombre Campo] es un campo requerido.”
Actor	Sistema
1. Seleccionará el botón Nuevo de la pantalla de Listado de Conexiones. Ver Flujo Normal – Buscar Conexiones	2. Visualizará la pantalla del Prototipo N° 4 , con los siguientes controles: <ul style="list-style-type: none"> • Código: mostrará el código de la conexión, este campo solo se deshabilitará después de guardar la conexión. • Alias: campo alfanumérico de longitud máxima de 60 dígitos (requerido). • Conexiones máximas: campo numérico de longitud máxima de 10 dígitos, representará número de conexiones máximas (requerido). • Conexiones mínimas: campo numérico de longitud máxima de 10 dígitos, representará número de conexiones mínimas (requerido). • Test Sql: campo alfanumérico, instrucción Sql de prueba (requerido). • Usuario: campo alfanumérico de longitud 60, usuario de la base de

	<p>datos (requerido).</p> <ul style="list-style-type: none"> • Password: campo alfanumérico de longitud 60, clave de la base de datos (requerido). • Antisondeo: lista de selección simple, muestra el antisondeo de una conexión (requerido). • Servidor: lista de selección simple, muestra los servidores de base de datos (requerido). • Base de Datos: campo alfanumérico de longitud 200 (requerido). • Driver Class: campo alfanumérico de longitud 200 (requerido). • Driver Url: campo alfanumérico de longitud 800 (requerido). • Estado: mostrará el estado de la conexión, por defecto mostrará el estado CREADO. <p>3. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: permitirá guardar la información de la conexión, su funcionalidad será descrita más adelante. • Regresar: retornará a la pantalla Listado de Conexiones.
<p>4. Ingresará los datos de la conexión y dará clic en el botón Guardar.</p>	
	<p>5. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresara un campo requerido, generará la excepción Ex4. <p>6. Si no se generara ninguna excepción, guardará la información de la conexión, mostrará un mensaje de confirmación “La información de la conexión ([Código]) ha sido guardada con éxito” y mostrará el Prototipo N° 5.</p> <p>7. Mostrará el botón Cambiar Estado, que permitirá modificar el estado de la conexión, su funcionalidad será descrita en el Flujo Normal – Editar Propiedades de una Conexión.</p>
<p>Prototipo</p>	

Prototipo N° 4:

Propiedades de Conexión

Datos de Conexión
-

Código*	<input type="text"/>	<div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;">Guardar</div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;">Regresar</div> </div>
Alias*	<input type="text"/>	
Conexiones Maximas*	<input type="text"/>	
Conexiones Minimas*	<input type="text"/>	
Test Sql*	<input type="text"/>	
Usuario*	<input type="text"/>	
Password*	<input type="text"/>	
Antisondeo*	<input type="text" value="Seleccionar"/>	
Servidor*	<input type="text" value="Seleccionar"/>	
Base de Datos*	<input type="text"/>	
Driver Class*	<input type="text"/>	
Driver Uri*	<input type="text"/>	
Estado	CRE	

Prototipo N° 5:

Propiedades de Conexión

Datos de Conexión
-

Código*	<input type="text" value="CXN_SPCMIDSQLS_1"/>	<div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;"> ▼ Cambiar Estado </div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;">Guardar</div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;">Iniciar</div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;">Detener</div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;">Test</div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; text-align: center; width: 100px;">Regresar</div> </div>
Alias*	<input type="text" value="CONEXION PARA SPC CON MIDDLEWARE"/>	
Conexiones Maximas*	<input type="text" value="100"/>	
Conexiones Minimas*	<input type="text" value="20"/>	
Test Sql*	<input type="text" value="SELECT GETDATE()"/>	
Usuario*	<input type="text" value="mid"/>	
Password*	<input type="text" value="654321"/>	
Antisondeo*	<input type="text" value="240 (ms)"/>	
Servidor*	<input type="text" value="ACC-SERVER - localhost : 5432"/>	
Base de Datos*	<input type="text" value="DEGA_SSIT_BASE_PRUEBAMID"/>	
Driver Class*	<input type="text" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>	
Driver Uri*	<input type="text" value="jdbc:sqlserver://<host>;databaseName=<database>;user=<user>;password=<password>;"/>	
Estado	ACT	

Flujo Normal – Editar Propiedades de una Conexión

Escenario	El Usuario Mid ingresará al sistema y modificará los datos de una conexión. Solo podrá modificar los campos habilitados para la modificación.
Excepciones	• Ex5: “[Campo Requerido] es un campo requerido.”
Actor	Sistema
1. <incluir> Flujo Normal – Buscar Conexiones. 2. Elegirá la conexión y seleccionará el	

botón Ver Detalle .	
	<p>3. Visualizará la pantalla del prototipo N° 6, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: campo no editable (requerido). • Alias: campo alfanumérico deshabilitado (requerido). • Conexiones máximas: campo numérico deshabilitado (requerido). • Conexiones mínimas: campo numérico deshabilitado (requerido). • Test Sql: campo alfanumérico, deshabilitado (requerido). • Usuario: campo alfanumérico deshabilitado (requerido). • Password: campo alfanumérico deshabilitado (requerido). • Antisondeo: lista de selección simple, deshabilitado (requerido). • Servidor: lista de selección simple, deshabilitado (requerido). • Base de Datos: campo alfanumérico deshabilitado (requerido). • Driver Class: campo alfanumérico deshabilitado (requerido). • Driver Url: campo alfanumérico deshabilitado (requerido). • Estado: campo no editable. • Botón Editar • Botón Regresar
4. Seleccionará el botón Editar .	
	<p>5. Visualizará la pantalla del Prototipo N° 7, con los siguientes controles:</p> <ul style="list-style-type: none"> • Código: campo no editable (requerido). • Alias: campo alfanumérico deshabilitado (requerido). • Conexiones máximas: campo numérico deshabilitado (requerido). • Conexiones mínimas: campo numérico deshabilitado (requerido). • Test Sql: campo alfanumérico, deshabilitado (requerido). • Usuario: campo alfanumérico deshabilitado (requerido). • Password: campo alfanumérico deshabilitado (requerido). • Antisondeo: lista de selección simple,

	<p>deshabilitado (requerido).</p> <ul style="list-style-type: none"> • Servidor: lista de selección simple, deshabilitado (requerido). • Base de Datos: campo alfanumérico deshabilitado (requerido). • Driver Class: campo alfanumérico deshabilitado (requerido). • Driver Url: campo alfanumérico deshabilitado (requerido). • Estado: campo no editable. • Botón Guardar. • Botón Cambiar Estado. • Botón Regresar. <p>6. También mostrará las opciones:</p> <ul style="list-style-type: none"> • Guardar: guardará la actualización de la información de la conexión, será descrita más adelante. • Cambiar Estado: modificará el estado de la conexión, será descrita más adelante • Regresar: mostrará la pantalla Listado de Conexiones.
<p>7. Ingresará los valores a modificar. 8. Seleccionará el botón Guardar.</p>	
	<p>9. Validará los datos ingresados:</p> <ul style="list-style-type: none"> • Si no ingresara un campo requerido, generará la excepción Ex5. <p>10. Si no se generara ninguna excepción, actualizará la información de la conexión y mostrará un mensaje de confirmación “La información de la conexión ([Código]) ha sido guardada con éxito”.</p>
<p>11. Seleccionará el botón Cambiar Estado.</p>	
	<p>12. Mostrará la lista de procesos, de acuerdo al estado de la conexión:</p> <ul style="list-style-type: none"> • ACTIVAR: cambiará el estado de la conexión a ACTIVADO y mostrará el mensaje: “El estado de la conexión [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado de la conexión fuera CREADO O DESACTIVADO. • DESACTIVAR: cambiará el estado de la conexión a DESACTIVADO y mostrará el mensaje: “El estado del de la conexión [código] ha sido modificado con éxito.” Esto solo se realizará, si el estado de la

conexión fuese CREADO O
ACTIVADO.

Prototipo

Prototipo N° 6:
Propiedades de Conexión

Datos de Conexión

Código*	CXN_SPCMIDSOLS_1	▶ Editar
Alias*	CONEXION PARA SPC CON MIDDLEWARE	↶ Regresar
Conexiones Maximias*	100	
Conexiones Mínimas*	20	
Test Sql*	SELECT GETDATE()	
Usuario*	mid	
Password*	654321	
Antisondeo*	240 (ms)	
Servidor*	ACC-SERVER - localhost : 5432	
Base de Datos*	DESA_SSIT_BASE_PRUEBAMID	
Driver Class*	com.microsoft.sqlserver.jdbc.SQLServerDriver	
Driver Uri*	jdbc:sqlserver://<host>;databaseName=<database>;user=<user>;password=<password>;	
Estado	ACT	

Prototipo N° 7:
Propiedades de Conexión

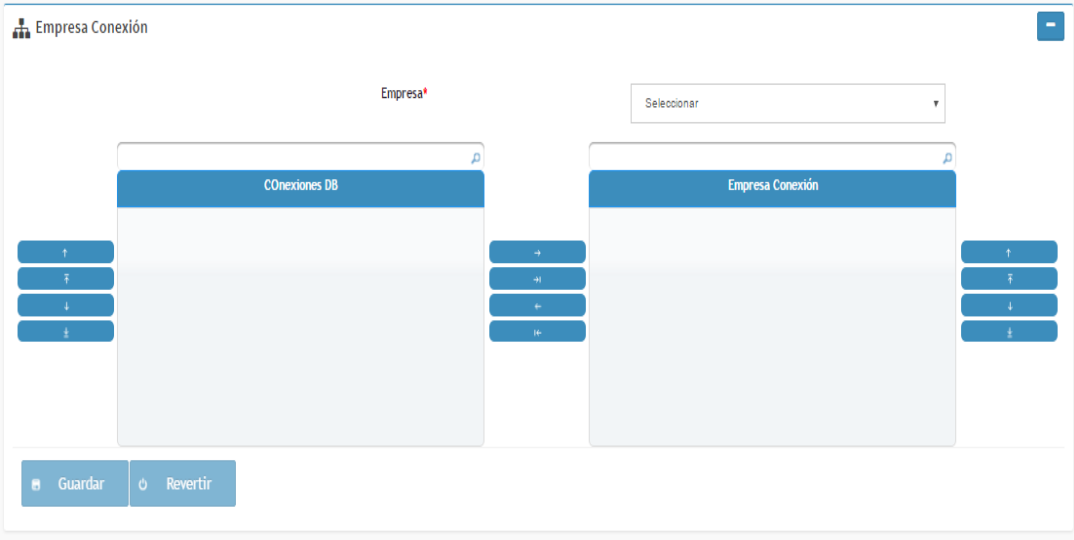
Datos de Conexión

Código*	CXN_SPCMIDSOLS_1	▼ Cambiar Estado DESACTIVAR
Alias*	CONEXION PARA SPC CON MIDDLEWARE	▶ Iniciar
Conexiones Maximias*	100	⏸ Detener
Conexiones Mínimas*	20	✓ Test
Test Sql*	SELECT GETDATE()	↶ Regresar
Usuario*	mid	
Password*	654321	
Antisondeo*	240 (ms)	
Servidor*	ACC-SERVER - localhost : 5432	
Base de Datos*	DESA_SSIT_BASE_PRUEBAMID	
Driver Class*	com.microsoft.sqlserver.jdbc.SQLServerDriver	
Driver Uri*	jdbc:sqlserver://<host>;databaseName=<database>;user=<user>;password=<password>;	
Estado	ACT	

Tabla N° 4.14: CU05 - Gestionar Conexiones.

Actor	Primario: Usuario Mid
Descripción	Se asignar una a una empresa mid.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”.

	Tener gestionado las conexiones y empresas mid.	
Post-condiciones	Ninguna	
Req. Especiales	Ninguno	
Flujo Normal – Asociar Conexiones a Empresa Mid		
Escenario	El actor (Usuario Mid) podrá aplicar el filtro de empresa mid y alias de conexión sobre las conexiones disponibles y conexiones asociadas.	
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado una Empresa Mid”. 	
Actor	Sistema	
1. Seleccionará la opción Empresa Conexión del menú izquierdo Configuración.		
	2. Visualizará la pantalla del Prototipo N° 1 , con los siguientes controles: <ul style="list-style-type: none"> o Empresa: lista de selección simple campo requerido. o Conexiones BD: componente que visualiza las conexiones disponibles. o Empresa Conexión: componente que visualiza las conexiones asociados a empresa mid. 3. También mostrará los botones: <ul style="list-style-type: none"> • Guardar: permitirá realizar la búsqueda de conexión, su funcionalidad será descrita más adelante. • Revertir: permitirá revertir los cambios realizados en la los componentes de conexión disponible y conexiones empresa asociadas. 	
4. Elegirá una conexión o seleccionar varias conexiones a través de los checks de la sección de conexiones disponibles. 5. Seleccionará los botones direccionales (>>), ver prototipo N° 2 .		
	6. Si no seleccionara una conexión de la sección de resultados, los botones Guardar y Revertir seguirán deshabilitadas. 7. Visualizará la pantalla del prototipo N° 3 , con los siguientes controles: <ul style="list-style-type: none"> o Empresa Conexión: componente que asocia una conexión a empresa mid. • También mostrará los botones: <ul style="list-style-type: none"> o Guardar: permitirá guardar la asociación de conexión a una empresa mid que será 	

	<p>descrita más adelante.</p> <ul style="list-style-type: none"> o Revertir: revertirá los cambios en los componentes de conexiones disponibles y asociadas.
<p>8. Elegirá una conexión de la sección de Empresa Conexión.</p> <p>9. Seleccionará los botones direccionales (<<)</p>	
	<p>10. Visualizará la pantalla del prototipo N° 4, con los siguientes controles:</p> <ul style="list-style-type: none"> o Conexión: componente conexión disponible. • También mostrará los botones: o Guardar: permitirá guardar la asociación de conexión a una empresa mid que será descrita más adelante. • Revertir: revertirá los cambios en los componentes de conexiones disponibles y asociadas.
<p>11. Seleccionará la opción Guardar</p>	
	<p>12. Si no se seleccionará la empresa se mostrará la excepción Ex.1.</p> <p>13. Si no se genera la excepción se procede guardar los cambios.</p> <ul style="list-style-type: none"> • Si una conexión deshabile es asociada recientemente se guarda dicha asociación. • Si una conexión asociada a sido desasociada se elimina de la asociación.
Prototipo	
<p>Prototipo N° 1:</p> 	
<p>Prototipo N° 2:</p>	



Prototipo N° 3:



Prototipo N° 4:



Tabla N° 4.15: CU06 - Asociar Conexión a Empresa Mid.

Actor	Primario: Usuario Mid
Descripción	Se asigna un query sql a una empresa mid.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”. Tener gestionado los query sql y empresas mid.
Post-condiciones	Ninguna

Req. Especiales	Ninguno	
Flujo Normal – Asociar Query Sql a Empresa Mid		
Escenario	El actor (Usuario Mid) podrá aplicar el filtro de empresa mid y alias de query sql sobre los queries sql disponibles y queries sql asociadas.	
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “La Empresa Mid es un capo requerido” 	
Actor	Sistema	
1. Seleccionará la opción Empresa Query del menú izquierdo Configuración.		
	2. Visualizará la pantalla del Prototipo N° 1 , con los siguientes controles: <ul style="list-style-type: none"> o Empresa: lista de selección simple campo requerido. o Queries: componente que visualiza los queries sql disponibles. o Empresa Query: componente que visualiza los queries asociados a empresa mid. 3. También mostrará los botones: <ul style="list-style-type: none"> • Guardar: permitirá realizar la búsqueda de query sql, su funcionalidad será descrita más adelante. • Revertir: permitirá revertir los cambios realizados en la los componentes de query sql disponible y empresa query sql asociadas. 	
4. Elegirá un query sql o seleccionar varios queries a través de los checks de la sección de queries disponibles.		
5. Seleccionará los botones direccionales (>>), ver prototipo N° 2 .		
	6. Si no seleccionara una conexión de la sección de resultados, los botones Guardar y Revertir seguirán deshabilitadas. 7. Visualizará la pantalla del prototipo N° 3 , con los siguientes controles: <ul style="list-style-type: none"> o Empresa Query: componente que asocia un query sql a empresa mid. • También mostrará los botones: <ul style="list-style-type: none"> o Guardar: permitirá guardar la asociación de un query sql a una empresa mid que será descrita más adelante. o Revertir: revertirá los cambios en los componentes de queries sql disponibles y asociadas. 	
8. Elegirá un query sql de la sección de		

<p>Empresa Query.</p> <p>9. Seleccionará los botones direccionales (<<)</p>	
	<p>10. Visualizará la pantalla del prototipo N° 4, con los siguientes controles:</p> <ul style="list-style-type: none"> o Queries: componente queries sql disponible. • También mostrará los botones: <ul style="list-style-type: none"> o Guardar: permitirá guardar la asociación de un query sql a una empresa mid que será descrita más adelante. • Revertir: revertirá los cambios en los componentes de queries sql disponibles y asociadas.
<p>11. Seleccionará la opción Guardar</p>	
	<p>12. Si no se seleccionará la empresa se mostrará la excepción Ex.1.</p> <p>13. Si no se genera la excepción se procede guardar los cambios.</p> <ul style="list-style-type: none"> • Si un query sql deshabile es asociada recientemente se guarda dicha asociación. • Si un query sql asociada a sido desasociada se elimina de la asociación.

Prototipo

Prototipo N° 1:



Prototipo N° 2:



Prototipo N° 3:



Prototipo N° 4:

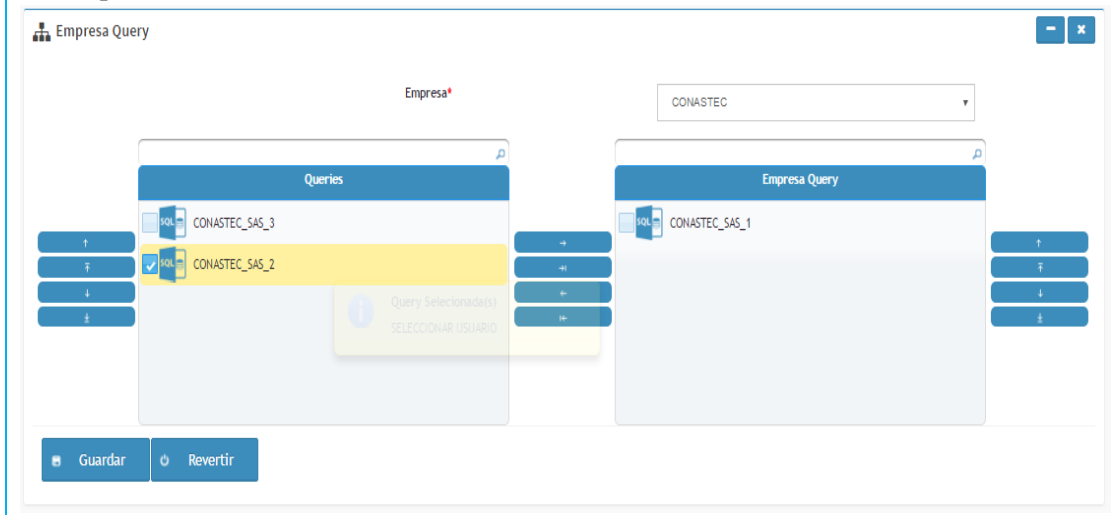


Tabla N° 4.16: CU07 - Asociar Query Sql a Empresa Mid.

Actor	Primario: Usuario Mid
Descripción	Se genera los datos locales para que el middleware no acceda a una base de datos para cada consulta de configuración.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”. Tener gestionado los query sql y las conexiones.
Post-condiciones	Ninguna
Req. Especiales	Ninguno
Flujo Normal – Generar datos de configuración local	
Escenario	El actor (Usuario Mid) podrá seleccionar las tablas y esquemas que quiera generar de forma local.
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado ninguna opción”
Actor	Sistema
1. Seleccionará la opción Base de Datos	

Local del menú izquierdo Configuración.	
	<p>2. Visualizará la pantalla del Prototipo N° 1, con los siguientes controles:</p> <ul style="list-style-type: none"> o Gestionar Base de Datos Local: tabla que muestra las estructuras disponibles que se majara de forma local. <ul style="list-style-type: none"> ▪ Opción: check que permite seleccionar. ▪ Nombre: nombre de las estructuras. ▪ Descripción: Descripción de la estructura. <p>3. También mostrará los botones:</p> <ul style="list-style-type: none"> • Esquema: permitirá generar la estructura local de almacenamiento. • Generar: permitirá permite copiar los datos de query Sql y Conexión a una base de datos local previamente generado.
<p>4. Elegirá seleccionara un esquema disponible o varios a través de los checks de la sección de Esquemas Disponibles.</p> <p>5. Seleccionará el botón Esquema.</p>	
	<p>6. Si no seleccionara una conexión de la sección de resultados, se generará la excepción Ex1.</p> <p>7. Si no hay excepciones se procede a generar los esquemas seleccionados en el entorno local y muestra su mensaje de confirmación “Los esquemas han sido gestionados con éxito”.</p>
<p>8. Elegirá seleccionara un esquema disponible o varios a través de los checks de la sección de Esquemas Disponibles.</p> <p>9. Seleccionará el botón Generar.</p>	
	<p>10.Si no seleccionara una conexión de la sección de resultados, se generará la excepción Ex1.</p> <p>11.Si no hay excepciones se procede a poblar los esquemas con los datos de la base de datos relacional a los esquemas seleccionados en el entorno local y procede a muestra el mensaje de confirmación “Los esquemas han sido gestionados con éxito”.</p>
Prototipo	

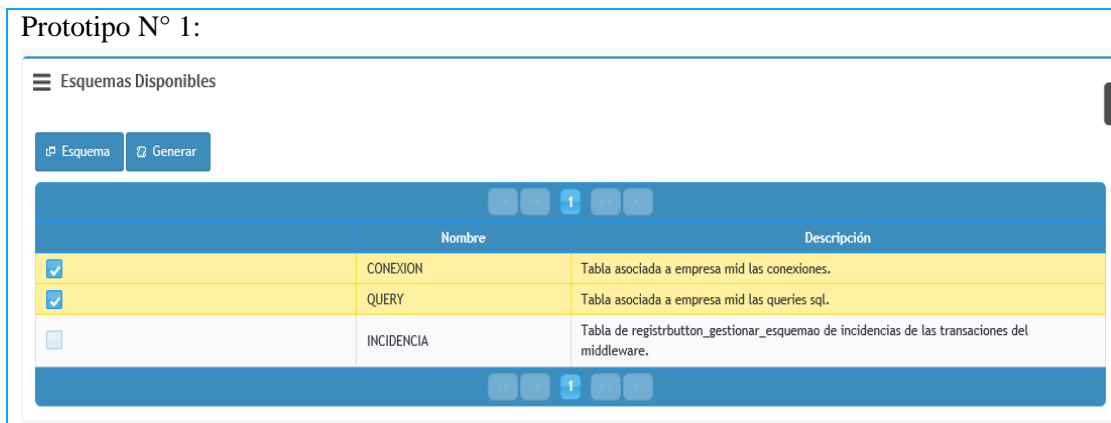


Tabla N° 4.17: CU08 – Gestionar datos de configuración local.

Actor	Primario: Usuario Mid
Descripción	Se gestiona todas las incidencias reportadas por el componente software denominado Middleware SQL.
Pre-condiciones	El actor se encontrará en la pantalla principal del módulo “Middleware SQL”.
Post-condiciones	Ninguna
Req. Especiales	Ninguno
Flujo Normal – Buscar Incidencias Middleware SQL	
Escenario	El actor (Usuario Mid) podrá seleccionar los filtros para obtener las incidencias generadas por el Middleware SQL.
Excepciones	<ul style="list-style-type: none"> • Ex.1: Visualizará la alerta: “No se ha seleccionado ninguna opción” • Ex.2: Visualizará la alerta: “No se seleccionaron incidencias”
Actor	Sistema
1. Seleccionará la opción Incidencias del menú izquierdo Gestión.	
	<p>2. Visualizará la pantalla del Prototipo N° 1, con los siguientes controles:</p> <ul style="list-style-type: none"> • Filtros de Gestión de incidencias Middleware SQL: sección que mostrará los filtros de búsqueda de las incidencias. <ul style="list-style-type: none"> o Empresa: lista desplegable de selección simple que mostrará la denominación de empresas y las aplicaciones. o Conexión DB: lista desplegable de selección simple que mostrará las conexiones activas. o Query Sql: lista desplegable de selección simple que mostrará los queries sql activos. o Estado: lista desplegable de selección simple que mostrará las incidencias del Middleware SQL sus valores serán:

	<ul style="list-style-type: none"> ▪ TRANSFERIDO ▪ PENDIENTE <p>3. También mostrará los botones:</p> <ul style="list-style-type: none"> • Buscar: permitirá realizar la búsqueda de conexión, su funcionalidad será descrita más adelante. • Limpiar: permitirá limpiar los campos de la pantalla, colocando los valores en: <ul style="list-style-type: none"> o Empresa: blanco. o Conexión DB: blanco. o Query Sql: blanco. o Estado: Seleccionar.
<p>4. Seleccionará o ingresará los filtros que desee aplicar a su búsqueda.</p> <p>5. Seleccionará el botón Buscar.</p>	
	<p>6. Validará los datos ingresados</p> <ul style="list-style-type: none"> • Si no ingresará ningún criterio de búsqueda, generará la excepción Ex1. <p>7. Si no se generara ninguna excepción, buscará las incidencias que coincidan con los criterios de búsqueda ingresados y mostrará la sección Resultados de la Búsqueda con los registros encontrados. La sección Resultados de Búsqueda mostrará las siguientes columnas:</p> <ul style="list-style-type: none"> • Código: mostrará el código de la incidencia, los resultados obtenidos estarán ordenados según esta columna. • Empresa: mostrará la empresa aplicación asociada a la incidencia. • Conexión DB: mostrará la conexión asociada al middleware. • Query Sql: mostrará la Query Sql asociada al middleware. • Estado: mostrará el estado del antisondeo. <p>Ver Prototipo N° 2</p> <p>8. Si no se encontraran resultados para la búsqueda, mostrará el mensaje “No se encontraron resultados”, dentro de la tabla de resultados.</p> <p>9. También se muestra los botones:</p> <ul style="list-style-type: none"> • Ver Detalle: Deshabilitada, permitirá ver el detalle del antisondeo que será descrita más adelante. • Transferir: Deshabilitada, permite transferir de forma manual las incidencias en estado PENDIENTE a la base de datos

	<p>correspondiente.</p> <ul style="list-style-type: none"> • Eliminar: Deshabilitada, permitirá borrar las incidencias ya transferidas que será descrita más adelante.
<p>10. Elegirá una incidencia de la sección de resultados.</p> <p>11. Seleccionará el botón Ver Detalle.</p>	
	<p>12. Si no seleccionara una incidencia de la sección de resultados, los botones Ver Detalle, Transferir y Eliminar seguirán deshabilitadas.</p> <p>13. Visualizará la pantalla del prototipo N° 3, con los siguientes controles:</p> <ul style="list-style-type: none"> • Propiedades de una Incidencia Middleware SQL: sección que mostrará las propiedades de una incidencia: <ul style="list-style-type: none"> o Código: código del antisondeo, campo no editable. o Empresa: campo numérico, campo deshabilitado. o Conexión: campo deshabilitado. o Query Sql: campo deshabilitado. o Usuario: campo deshabilitada. o Estado: campo no editable.
<p>14. Elegirá una o varias incidencias en estado PENDIENTE de la sección de resultados.</p> <p>15. Seleccionará el botón Transferir.</p>	
	<p>16. El sistema trasfieren todas los queries a las conexiones respectivas de las aplicaciones y confirma la trasferencia con el mensaje “Se trasfirieron [Núm. Incidencias] Incidencias”.</p>

Prototipo

Prototipo N° 1:

Lista de Incidencias Middleware SQL

Q Gestión de Incidencias Middleware SQL

Empresa

Conexión DB

Query Sql

Estado

Buscar

Limpiar

☰ Incidencias Middleware SQL Encontrados

Prototipo N° 2:

Lista de Incidencias Middleware SQL

Gestión de Incidencias Middleware SQL

Empresa: Buscar

Conexión DB: Limpiar

Query Sql:

Estado:

Incidencias Middleware SQL Encontrados

Ver Detalle Transferir Eliminar

Código	Empresa	Conexión DB	Query Sql	Estado
No se encontraron resultados.				

Tabla N° 4.18: CU09 – Gestionar Incidencias Middleware SQL.

DIAGRAMA DE CLASES

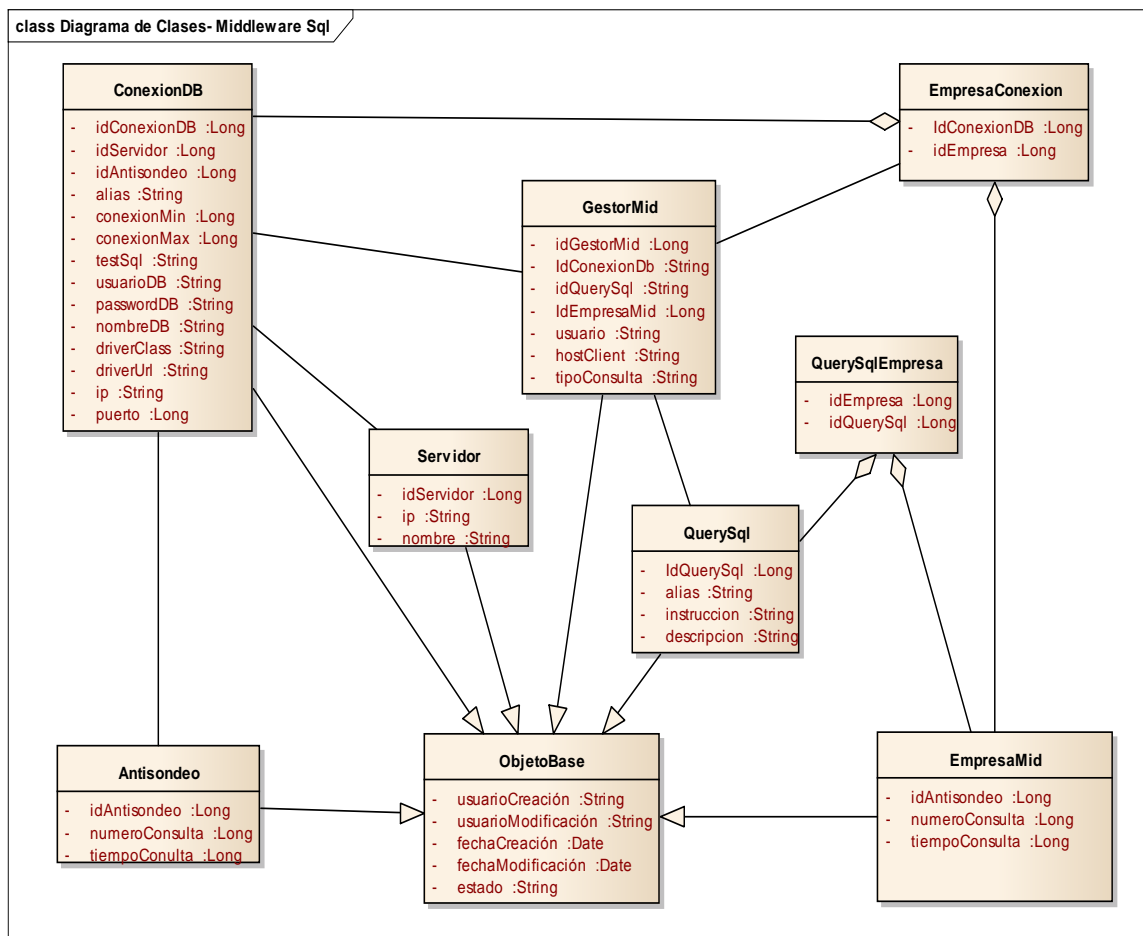


Figura N° 4.11: Diagrama de clases del Middleware SQL.

DIAGRAMA RELACIONAL DE BASE DE DATOS

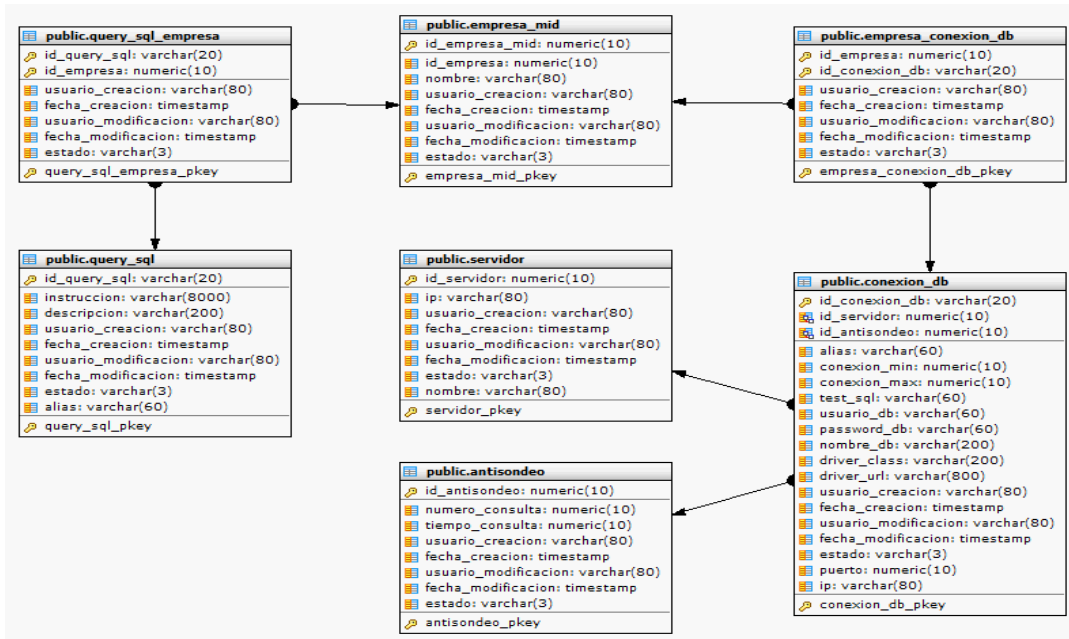


Figura N° 4.12: Base de datos relacional del Middleware SQL.

ARQUITECTURA TÉCNICA FUNCIONAL

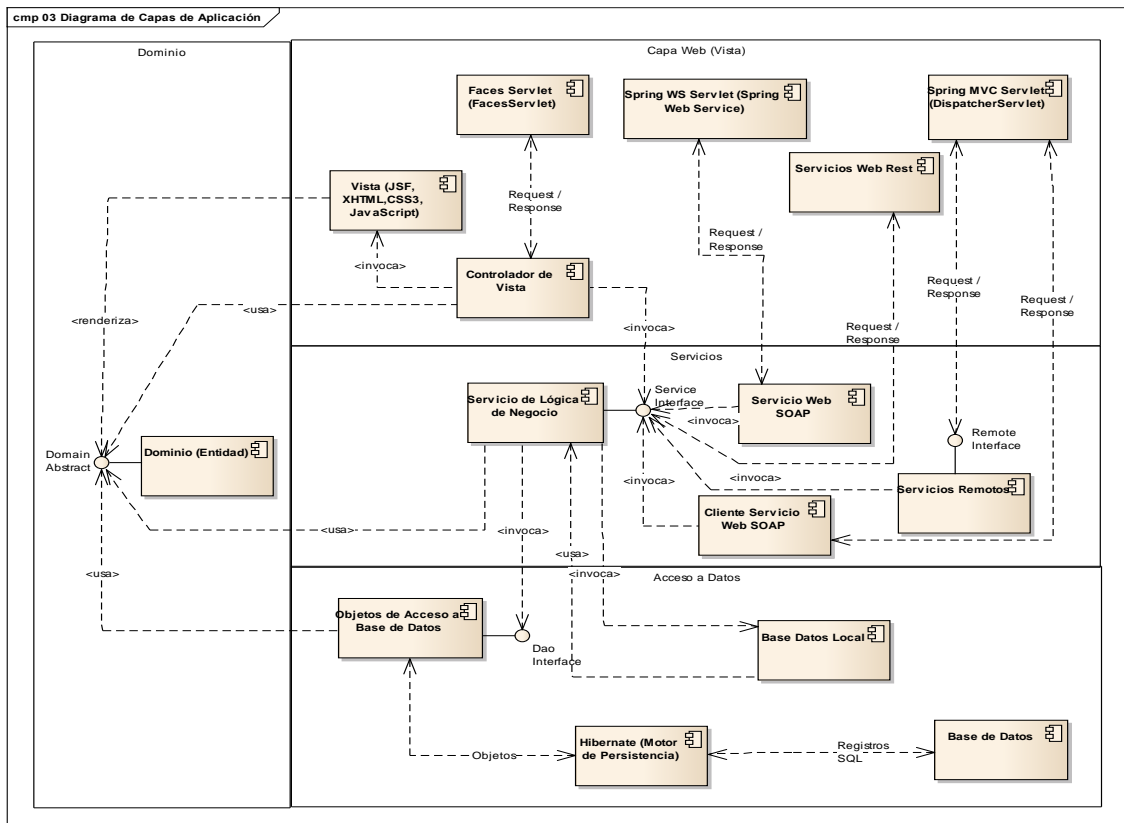


Figura N° 4.13: Diagrama de componentes del Middleware SQL.

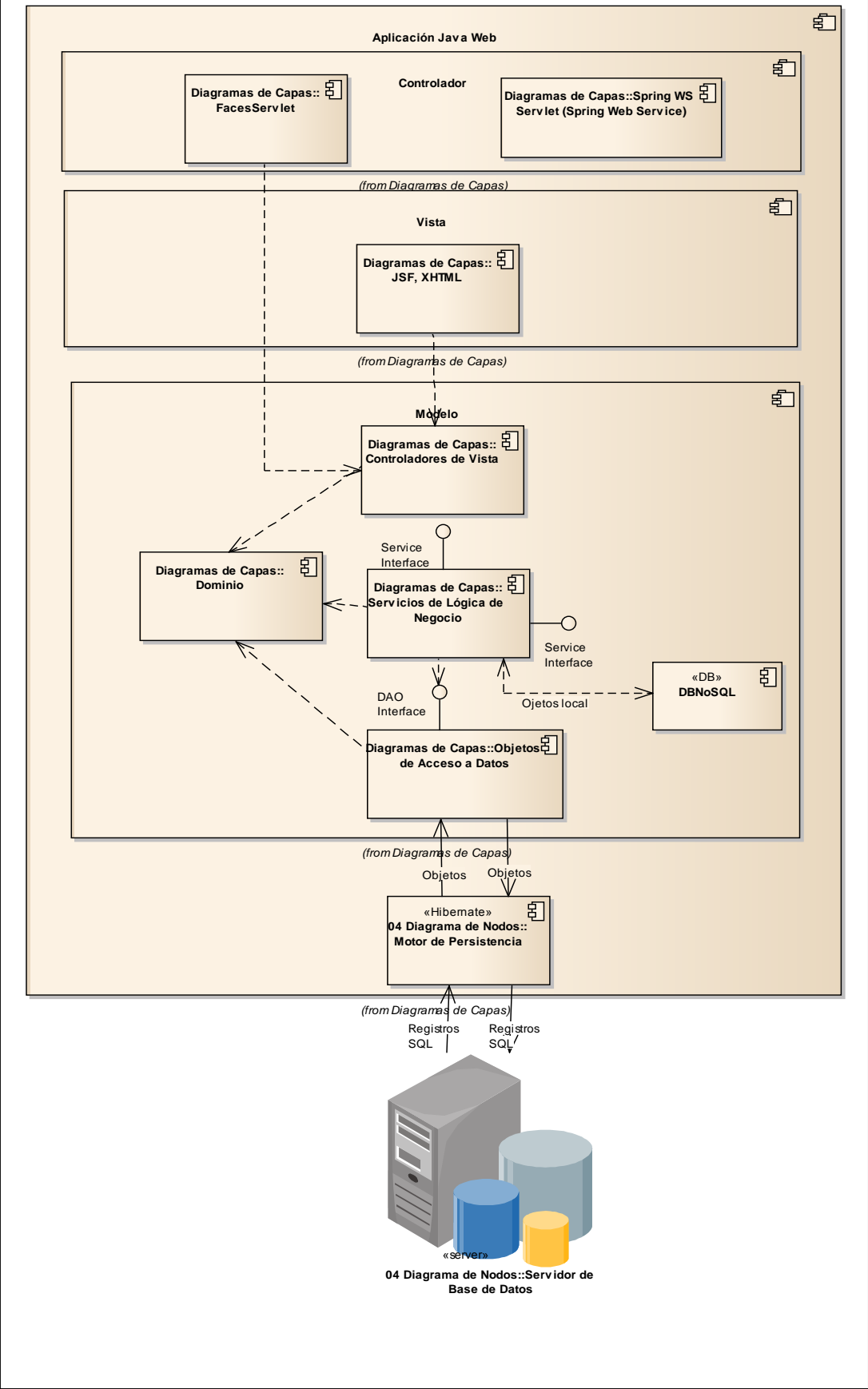


Figura N° 4.14: Diagrama de Capas MVC del Middleware SQL.

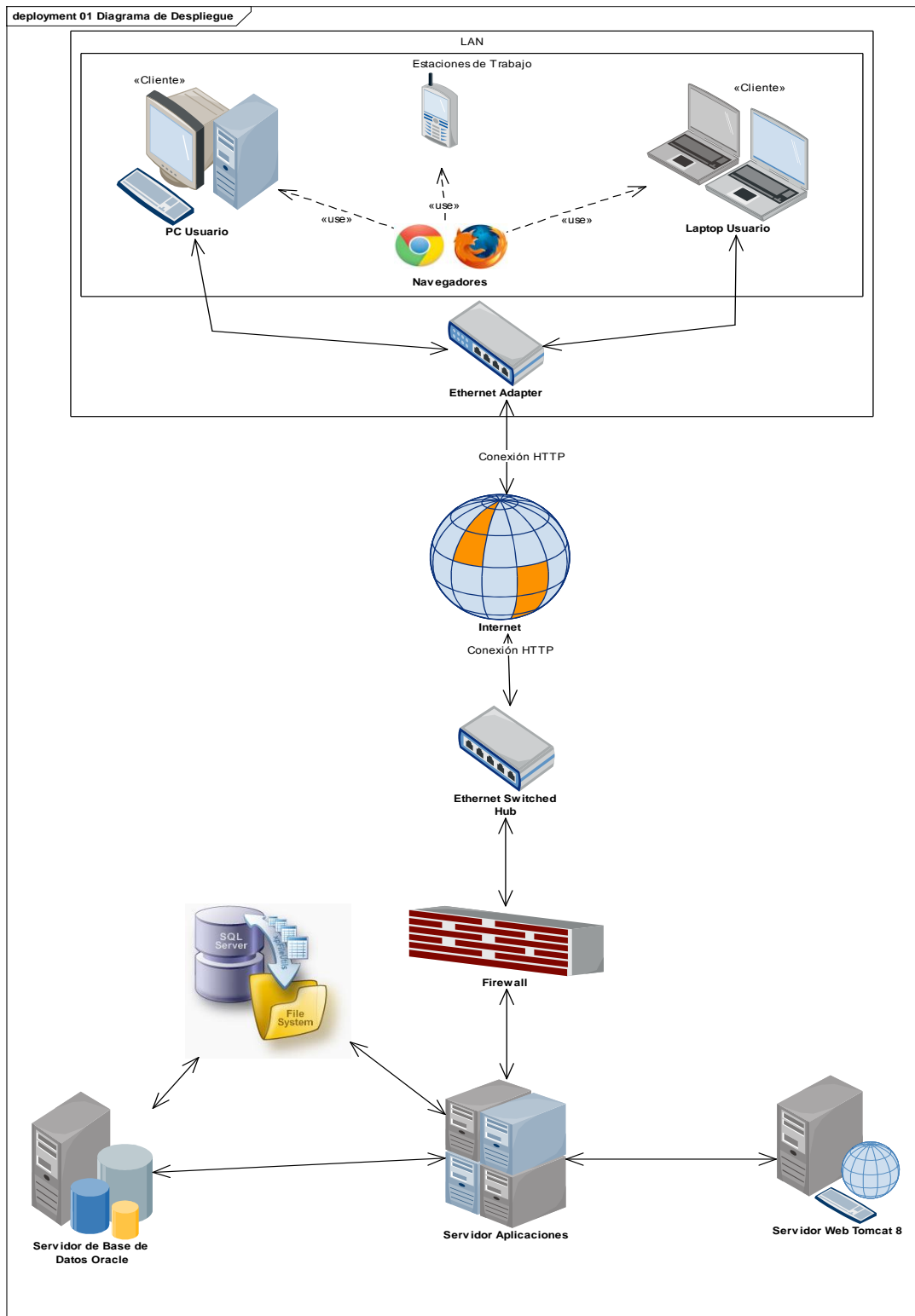


Figura N° 4.15: Diagrama de Despliegue del Middleware SQL.

b. Diseñar y desarrollar los contratos web services

Se ha utilizado la notación UML para generar los artefactos de análisis, diseño e implementación que se genera en la ingeniería de software para ello Scrum

declara que los artefactos UML generados son a voluntad de las necesidades para el entendimiento y desarrollo de los contratos web service al que denominamos Middleware WSSQL modulo interno que permite publicar los servicios WsSoapConexionService y WsRestConexionService los cuales han estado a cargo del Equipo.

REQUISITOS FUNCIONALES

Requisito	Descripción
1	El usuario wssql podrá gestionar las conexiones y comunicaciones a las bases de dato distribuida.
2	El usuario wssql podrá obtener los datos de entrada.
3	El usuario wssql gestionar y procesar datos de entrada.
4	El usuario podrá gestionar la ejecución de la consulta SQL

Tabla N°4.19: Requisitos funcionales del Middleware WSSQL.

CASOS DE USO

Documentan el comportamiento del sistema desde el punto de vista del usuario, usuario que será otra aplicación que use este middleware. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

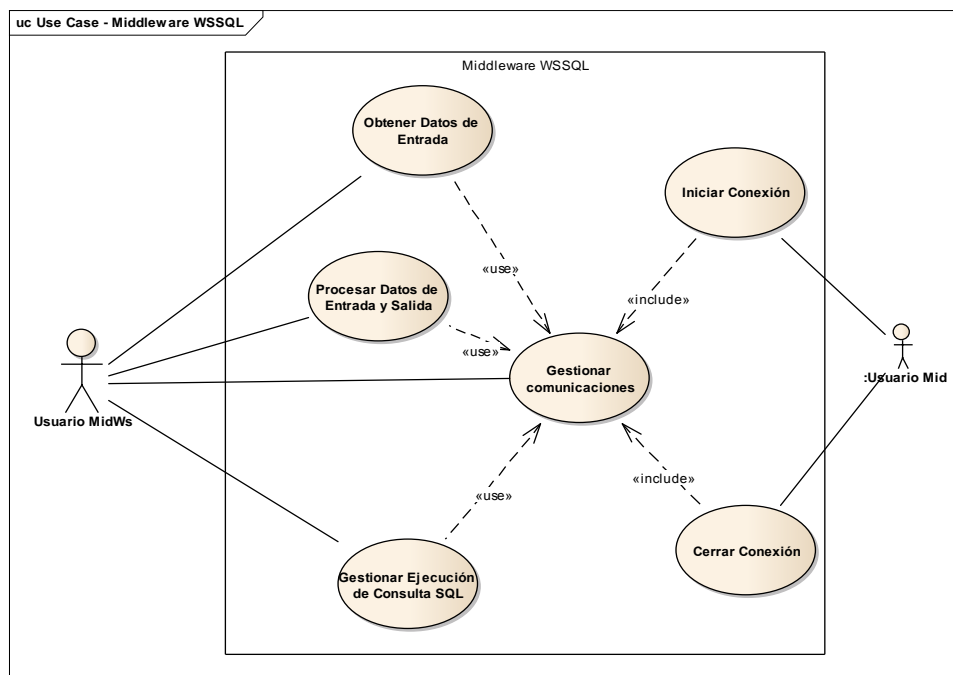


Figura N° 4.16: Diagrama de Casos de Uso del Middleware WSSQL.

DIAGRAMA DE ACTIVIDAD

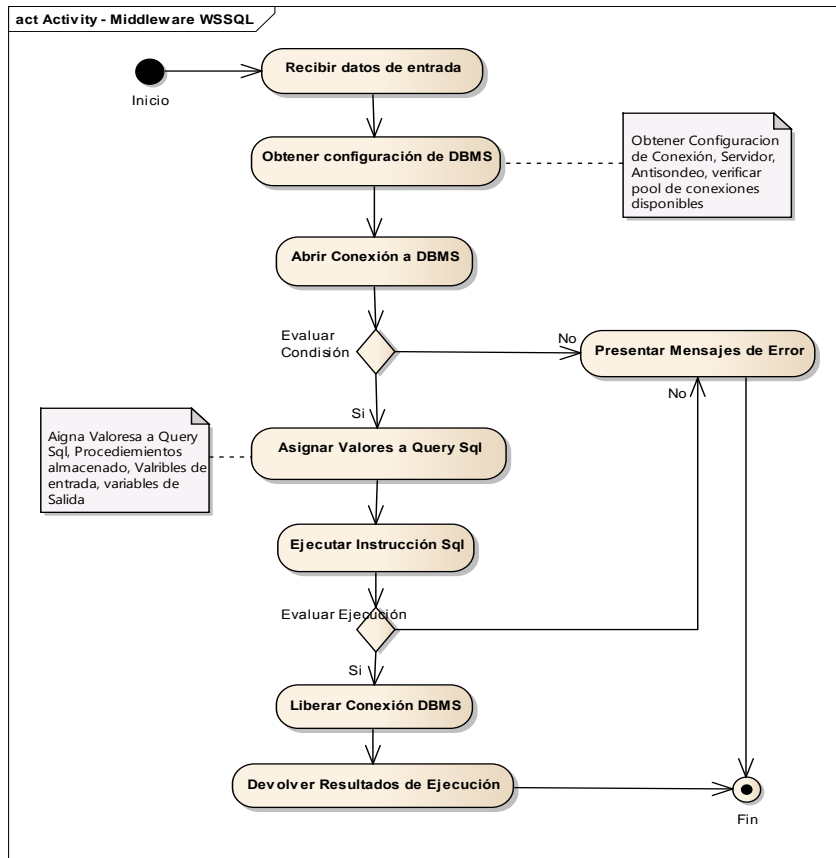


Figura N° 4.17: Diagrama de Actividades Generales del Middleware WSSQL.

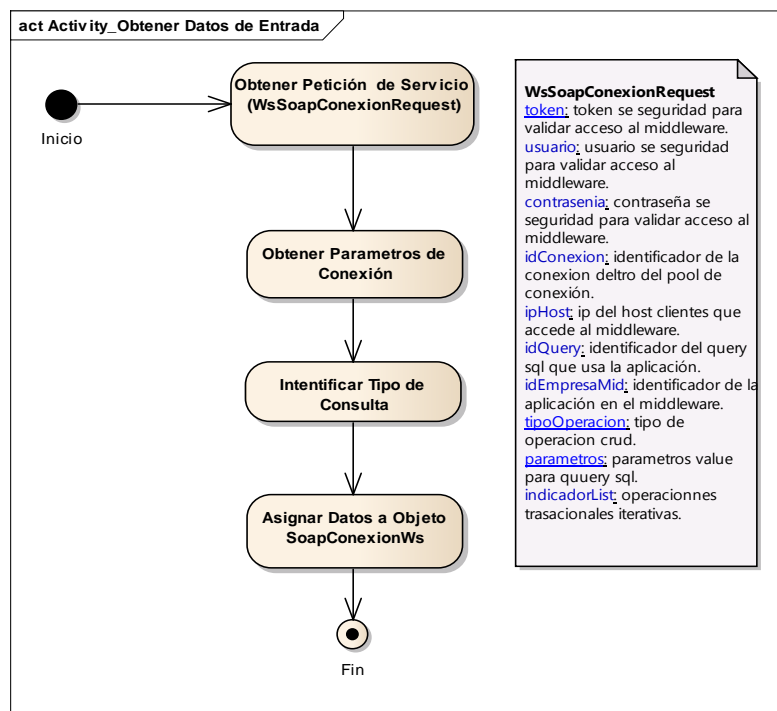


Figura N° 4.18: Actividades para Obtener Datos en el Middleware WSSQL.

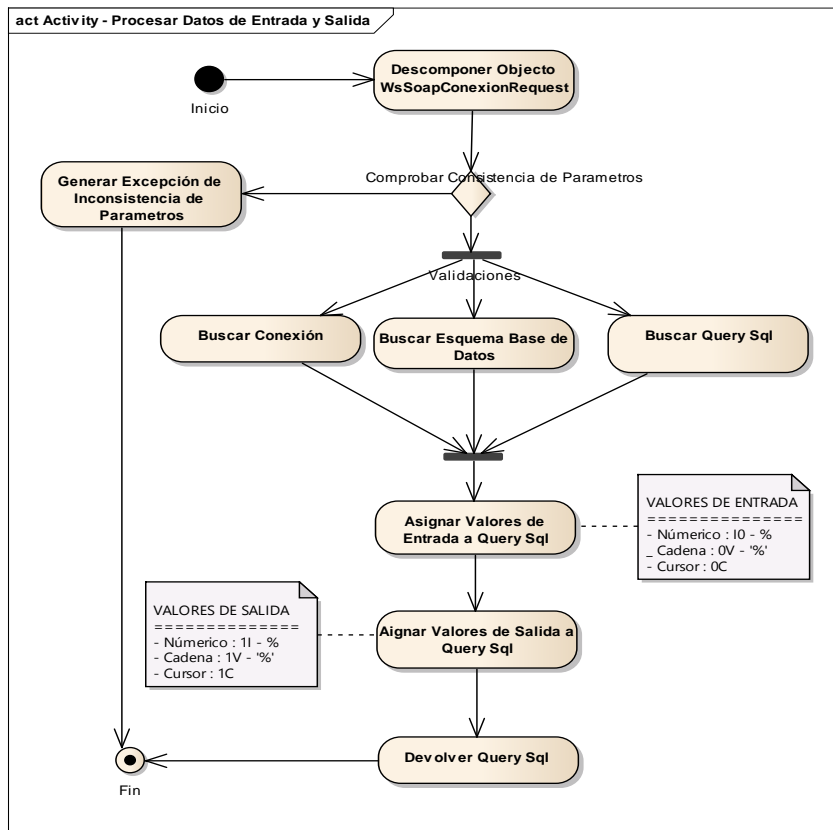


Figura N° 4.19: Actividades para Procesar Datos de E/S en el Middleware WSSQL.

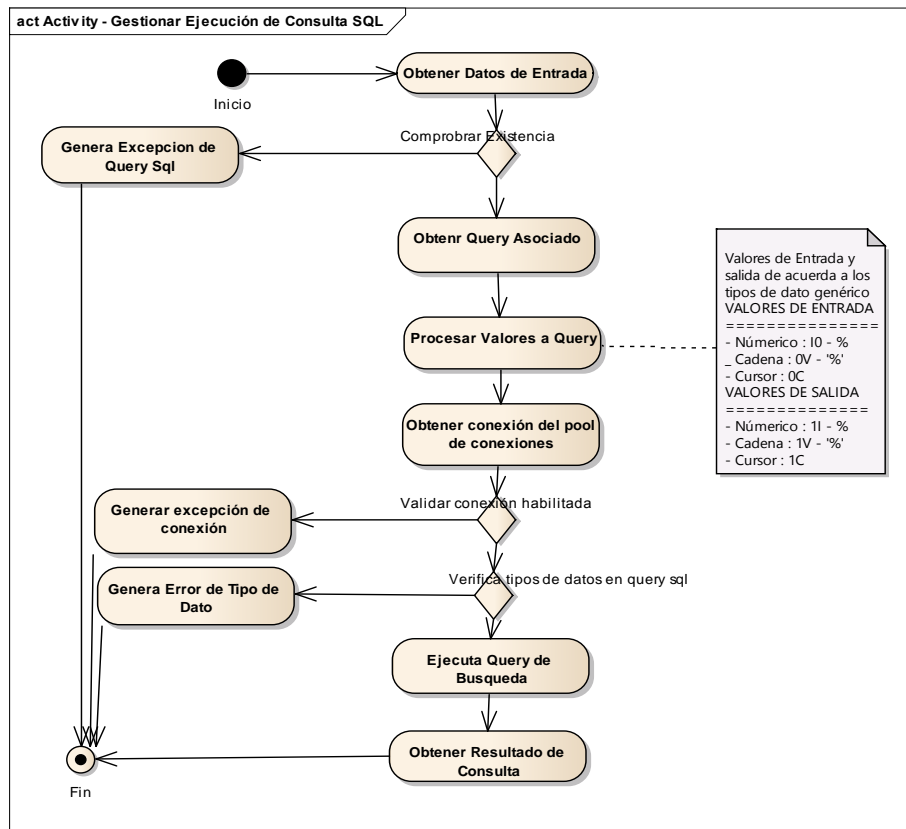


Figura N° 4.20: Actividades para Consulta Query Sql en el Middleware WSSQL.

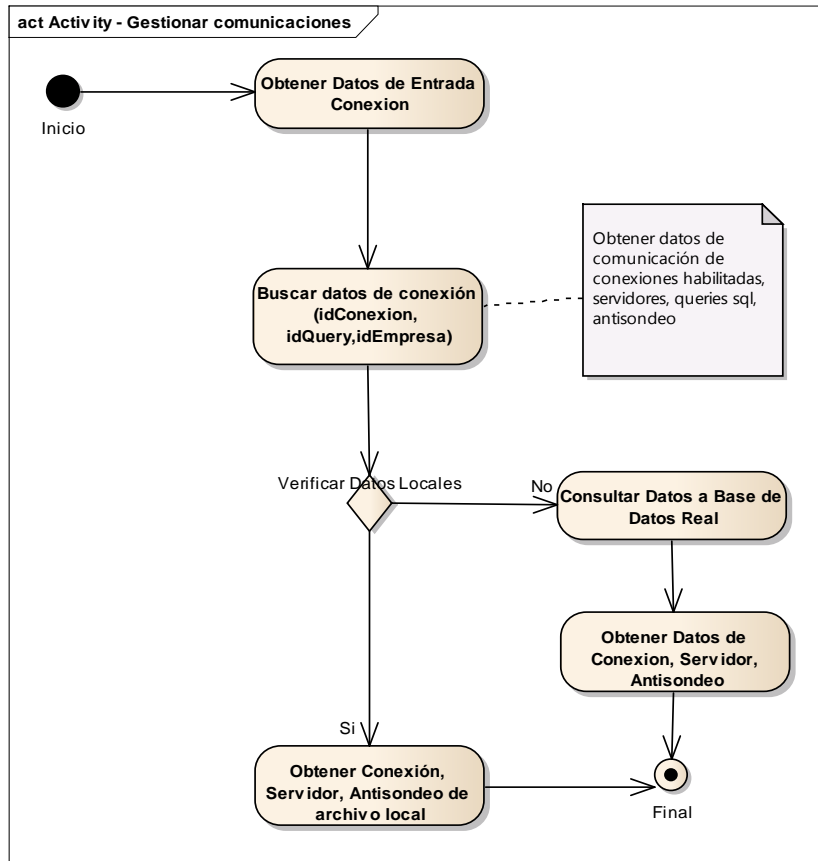


Figura N° 4.21: Actividades para Gestionar Comunicación en el Middleware WSQL.

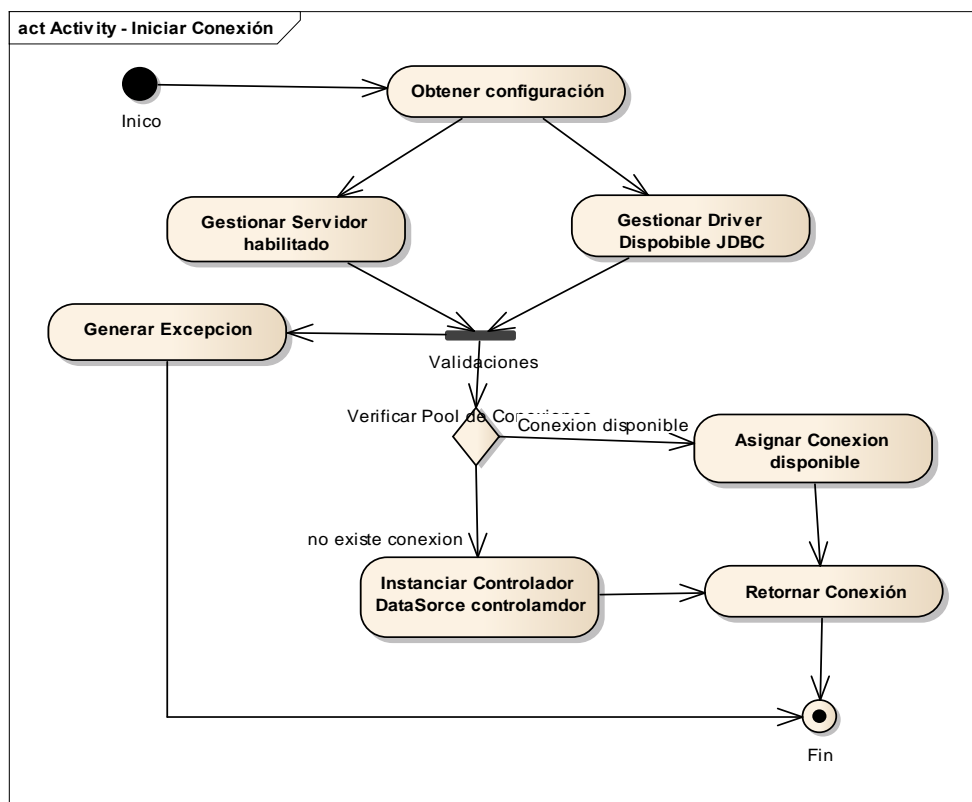


Figura N° 4.22: Actividades para Inicializar Conexión en el Middleware WSQL.

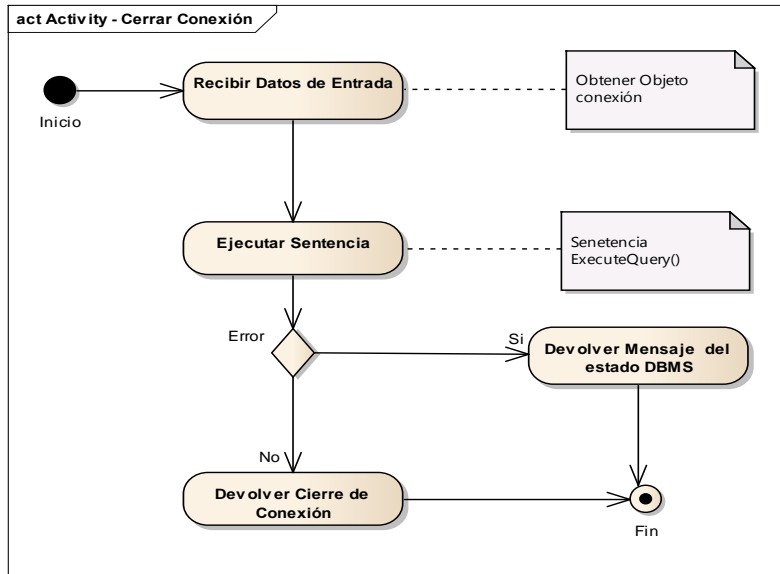


Figura N° 4.23: Actividades para Cerrar Conexión en el Middleware WSSQL.

DIAGRAMA DE CLASES

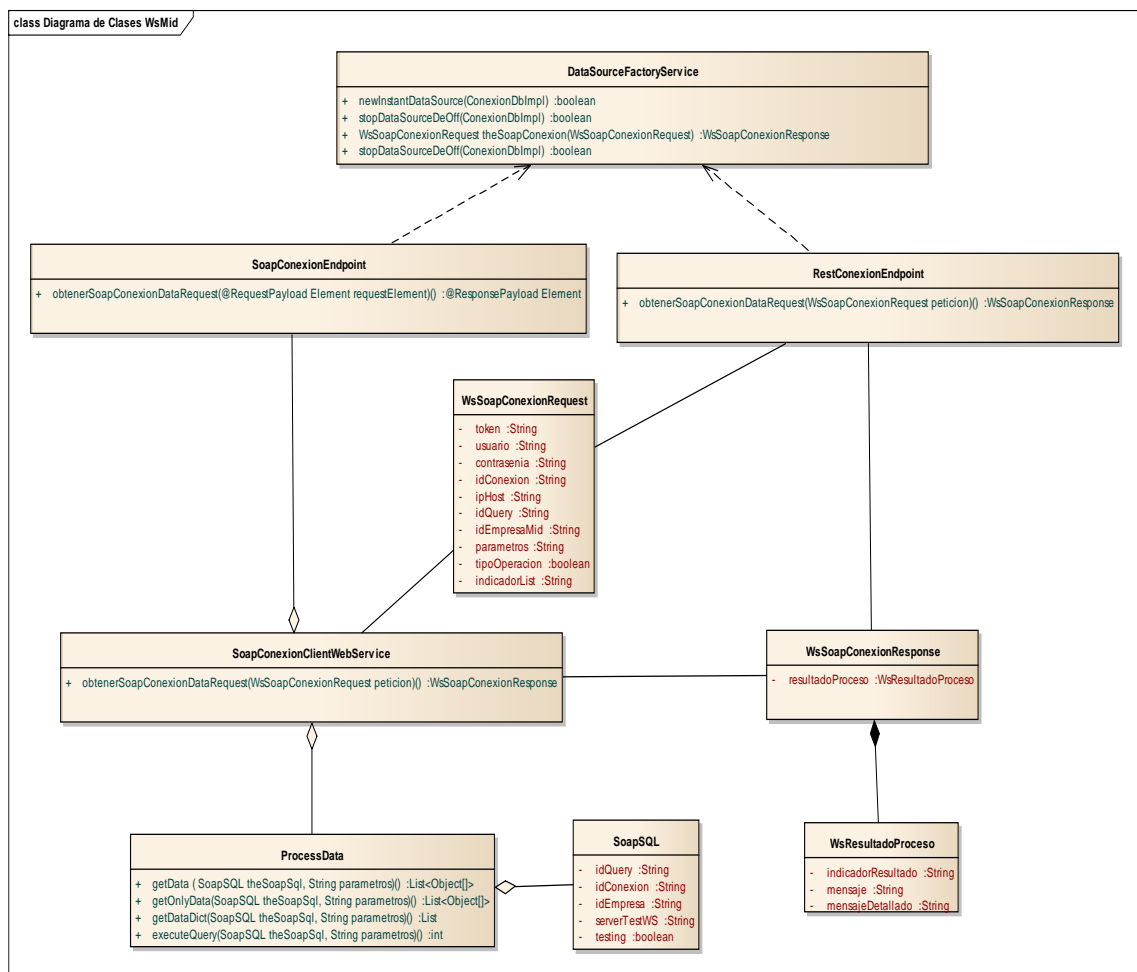


Figura N° 4.24: Diagrama de clases del Middleware WSSQL.

ARQUITECTURA TÉCNICA FUNCIONAL

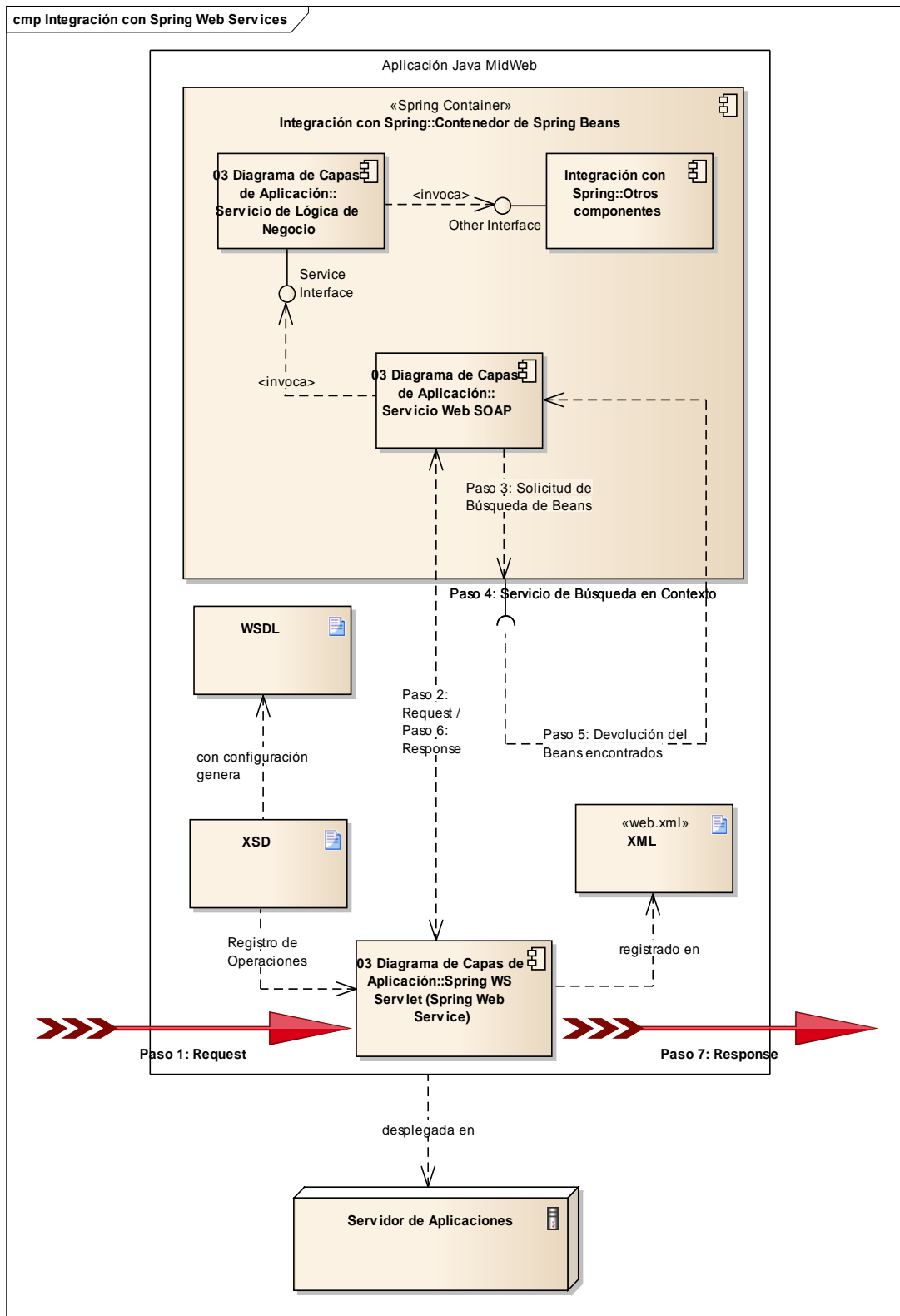


Figura N° 4.25: Diagrama de componentes para Integrar y generar la notación WSDL.

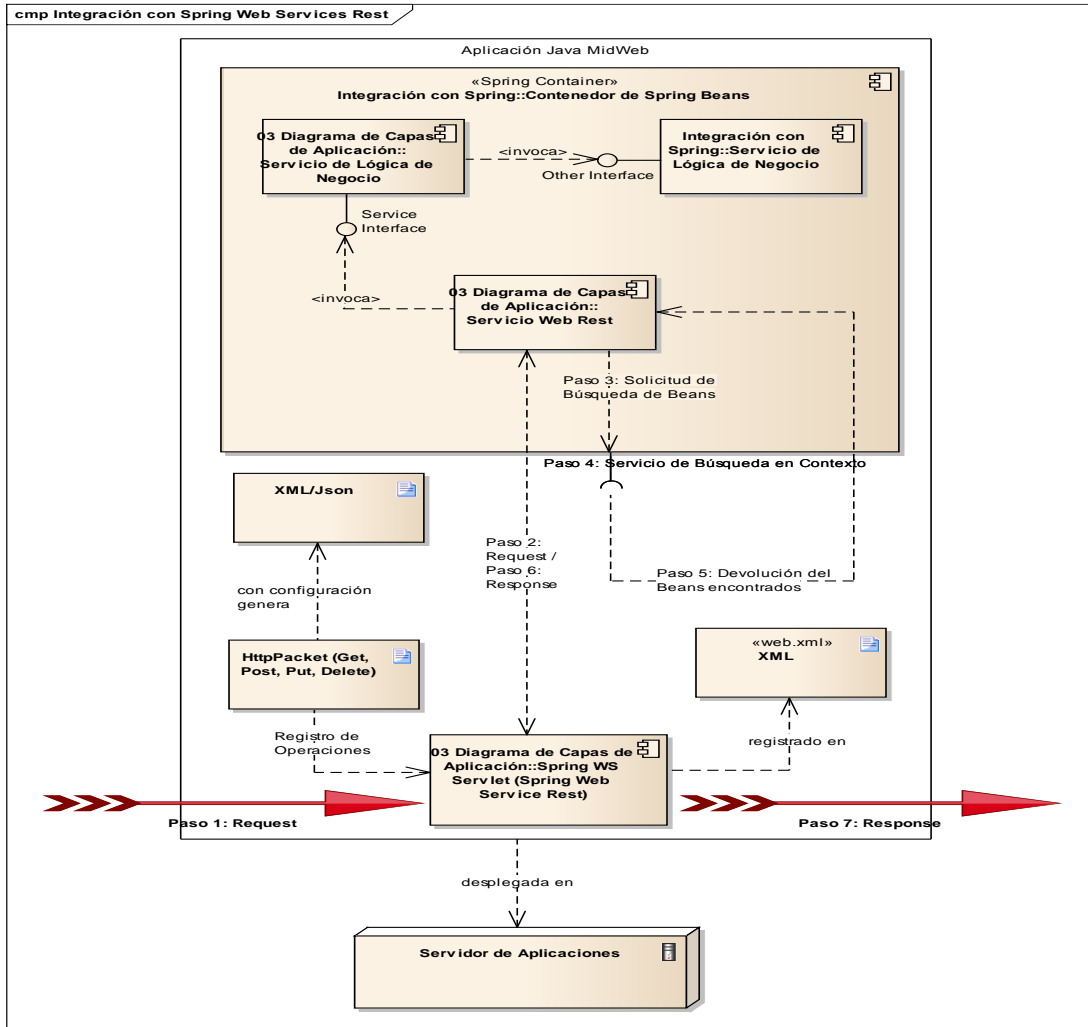


Figura N° 4.26: Diagrama de componentes para Integrar y generar Xml/Json.

PRUEBA UNITARIA DEL MIDDLEWARE SQL

```

ClienteDaoHibernate.java  ClienteDaoTest.java
1 package com.spring.pizza.dao;
2
3 import static org.junit.Assert.*;
14
15 @RunWith(SpringJUnit4ClassRunner.class)
16 @ContextConfiguration(locations={"classpath:com/spring/pizza/system
17 public class ClienteDaoTest {
18
19     @Autowired
20     private ClienteDao clienteDao;
21
22     @Test
23     public void obtenerListaDeClientesDePizzeria(){
24
25         List<Cliente> clientes = clienteDao.obtenerListaDeClientes
26         assertNotNull(clientes);
27     }
28 }

```

Console | Markers | Progress | Task List | Outline | Spring Explorer | Servers | Search | JUnit

Finished after 3,059 seconds

Runs: 1/1 | Errors: 0 | Failures: 0

com.spring.pizza.dao.ClienteDaoTest [Runner: JUnit 4] (3,003 s) | Failure Trace

Figura N° 4.27: Prueba unitaria de listar clientes con uso del middleware sql.

B. ÍTEM 6: Generar el instalador del componente software

a. Definir parámetros de compilación

La finalización del desarrollo nos permitió generar los instaladores requeridas para el funcionamiento autónomo del componente. Para ello se seleccionó los instaladores com.pe.ssit.acc.core.jar, com.pe.ssit.acc.pg.jar, com.pe.ssit.acc.sqlite.jar, MidWeb.war, MidClientWs.jar, PacWeb.war.

b. Instalar el middleware SQL y probar en varios ambientes

Los instaladores definidos y generados se empaquetaron y se instalaron en el servidor web Apache tomcat 8, configurado anteriormente en la Iteración I, se procede a la prueba del correcto funcionamiento del componte Middleware SQL.

C. ÍTEM 7: Elaborar manual de usuario del componente software

a. Documentar las opciones y el manejo de Middleware SQL

Se procede a elaborar el manual de funcionamiento y opciones que nos brinda el Middleware SQL y el Middleware WSSQL para su uso posterior en la creación de un nuevo software empresarial.

4.1.4.5. GRÁFICA DE TRABAJO PENDIENTE

Se muestra el esfuerzo restante desde el inicio de la iteración en el día cero, hasta el final de la iteración que fue determinada en 20 días.

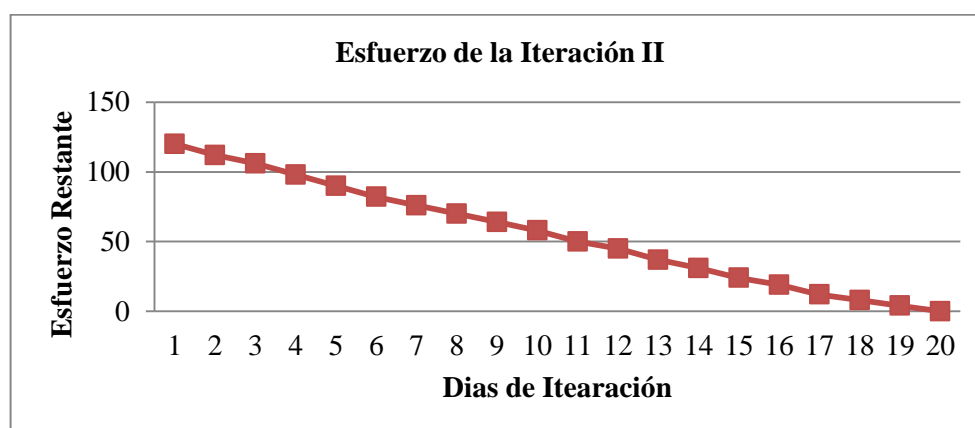


Figura N° 4.28: Gráfica de trabajo pendiente de la iteración II.

Existe un decremento constante del esfuerzo requerido en el tiempo, para lograr completar todos los requerimientos al final de los 20 días de la iteración.

4.1.4.6. REUNIÓN DE REVISIÓN DE LA ITERACIÓN

La segunda reunión de revisión fue realizada para mostrar el incremento del desarrollo a lo largo de la segunda iteración, junto con el facilitador y propietario del producto se revisó los resultados, obteniendo una evaluación favorable del detalle en el desarrollo del componente software, se presentó en la disponibilidad de los servicios de acceso a los datos. Así mismo se acordó la secuencia de la próxima iteración teniendo como objetivo en la medición de las eficiencias de acceso con diferentes software cliente, se desarrollara una aplicación web con Middleware SQL, una aplicación con Hibernate y se procederá a realizar la medias de eficiencia en el acceso a datos.

4.1.5. ITERACIÓN III: MEDICIÓN Y FORMALIZACIÓN DE TABLAS DE EFICIENCIA Y RENDIMIENTO

4.1.5.1. RESUMEN

En esta iteración se enfoca el trabajo de medición de eficiencia del software Middleware SQL para acceder a las bases de datos distribuida, para cumplir dicho cometido se desarrolló en paralelo un sistema web crud, uno con la técnica del Middleware SQL y el otro con la técnica de framework Hibernate en el acceso a datos; los procesos de registro, activación, desactivación, eliminación y consulta de registros se medirán con de acuerdo a los indicadores elaborados en esta misma iteración. Al final de esta iteración se logró tener tablas formalizadas de las mediadas realizadas con los indicadores de eficiencia respectiva.

4.1.5.2. REUNIÓN DE PLANIFICACIÓN DEL SPRINT

En esta reunión se seleccionaron los requerimientos que están que están enfocadas a desarrollar las aplicaciones web cliente y medir el acceso a los datos con dos técnicas diferentes. La meta es entregar las tablas formalizada de medidas de eficiencia en el acceso.

Por tercera ocasión se estima un esfuerzo de 90 horas, divididas en 10 días sin tomar en cuenta los fines de semana. Se estimó 40 horas para desarrollar el software web cliente con técnica de Middleware SQL, 40 horas para desarrollar el software web cliente con la técnica de Framework Hibernate y 10 horas medir y formalizar en las tablas de eficiencia de acceso a datos distribuidos.

4.1.5.3. PILA DE LA ITERACIÓN

La pila de iteración en la tabla N° 4.15, está compuesta por el desarrollo de los softwares cliente con las dos técnicas usadas para el acceso a los datos distribuidos y las medidas que han sido formalizadas las tablas de eficiencia.

Iteración I: Medición y Formalización de las Tablas de Eficiencia y Rendimiento											
ITEM	Tarea	Días de la iteración/esfuerzo restante									
		0	1	2	3	4	5	6	7	8	9
		90	82	76	66	60	52	34	26	12	0
8	Desarrollar nueva aplicación SpcMid con Middleware SQL.										
	Desarrollar la aplicación SpcMid.	36	28	24	16	12	6	0	0	0	
	Instalar la aplicación SpcMid.	4	4	4	4	4	4	0	0	0	
9	Desarrollar nueva aplicación SpcH con Framewok Hibernate.										
	Desarrollar la aplicación SpcH.	36	36	34	32	30	28	20	12	0	
	Instalar la aplicación SpcH.	4	4	4	4	4	4	4	4	0	
10	Medir y formalizar tablas.										
	Realizar las mediciones.	4	4	4	4	4	4	4	4	0	
	Elaborar las tablas comparativas de eficiencia.	6	6	6	6	6	6	6	6	4	

Tabla N°4.20: Pila de Iteración III.

4.1.5.4. TAREAS

A. ÍTEM 8: Desarrollar nueva aplicación SpcMid con Middleware SQL

a. Desarrollar la aplicación SpcMid

Se desarrolló un software cliente denominado SpcMid (Sistema procesador de consultas con Middleware SQL), sistema muestra 5 funcionalidades básicas que consta en la inserción de 10 mil registros, activar o actualizar 10 mil registros, desactivar o actualizar 10 mil registros, eliminar 10 mil registros, buscar 10 mil registros. Este sistema nos permitió poner a prueba el Middleware SQL que desarrollamos en la Iteración II y obtener las medidas respectivas de eficiencia en un ambiente distribuido heterogéneo.

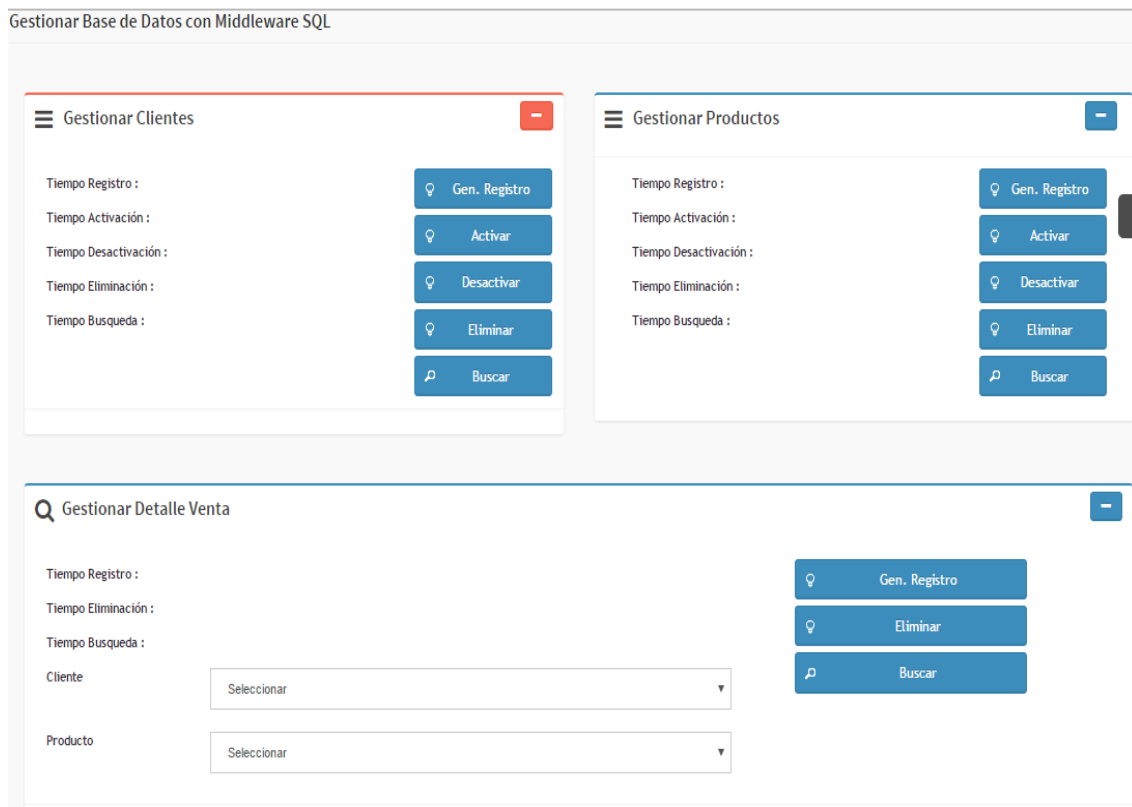


Figura N° 4.28: Prototipo de la aplicación web SpcMid.

b. Instalar la aplicación SpcMid

La instalación de la aplicación web cliente SpcMid se realiza en el servidor tomcat 8 para su prueba respectiva.

B. ÍTEM 8: Desarrollar nueva aplicación SpcH con Framework Hibernate

a. Desarrollar la aplicación SpcH

Se desarrolló un software denominado SpcH (Sistema procesador de consultas con Hibernate), Según Callejas (2011), en los estudios realizados “Evaluación y análisis de rendimiento de los frameworks de persistencia” concluye que “El framework hibernate es el más estable, robusto y de mayor performance capaz de soportar cantidades inmensas de instrucciones SQL, sin afectar su rendimiento”. El sistema muestra 5 funcionalidades básicas que consta en la inserción de 10 mil registros, activar o actualizar 10 mil registros, desactivar o actualizar 10 mil registros, eliminar 10 mil registros, buscar 10 mil registros.

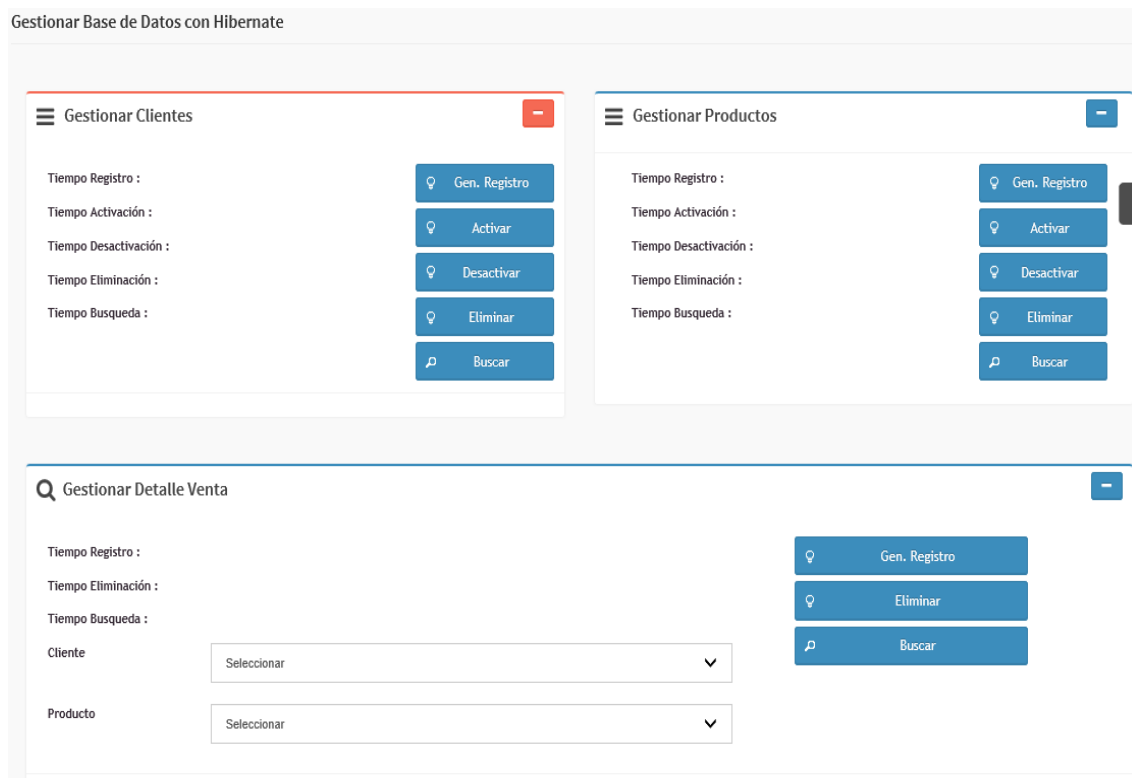


Figura N° 4.29: Prototipo de la aplicación web SpcH.

b. Instalar la aplicación SpcH

La instalación de la aplicación web SpcH se realizó luego de finalizar el desarrollo en el servidor tomcat 8 para su prueba respectiva.

C. ITEM 8: Medir y formalizar las tablas de eficiencia

a. Realizar las mediciones

Los sistemas SpcMid y SpcH instalados se procederá a realizar las mediciones respectivas de acuerdo a los indicadores o métricas elaborados.

MEDIDAS REALIZADAS CON SPCMID

En estas dos secciones, generar clientes, generar productos se midió el tiempo de respuesta en el acceso a las bases de datos distribuidos de forma transparente de acuerdo a cada evento de Gen. Registro, Activar, Desactivar, Eliminar y Buscar.

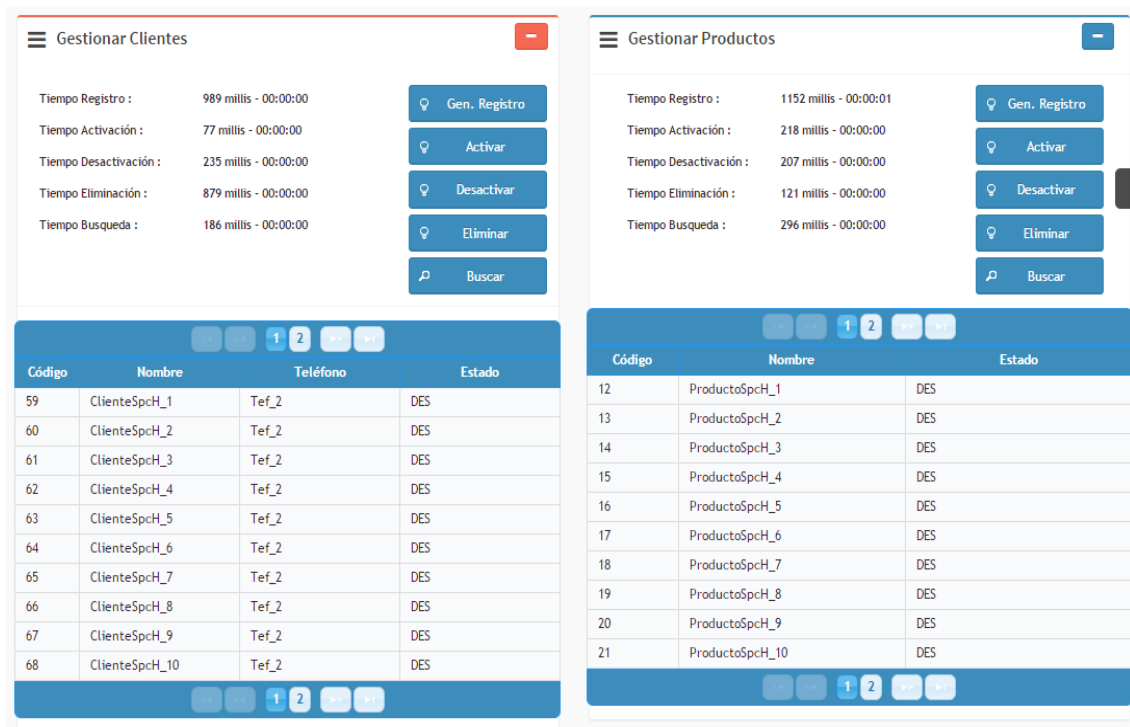


Figura N° 4.27: Tiempos en la generación de clientes y productos con SpcMid.

En esta tercera sección se genera datos para el detalle de venta, se elimina los detalles y se busca los detalles, estas funcionalidades permitieron obtener medidas reales en el acceso a las bases de datos distribuidas de forma transparente.

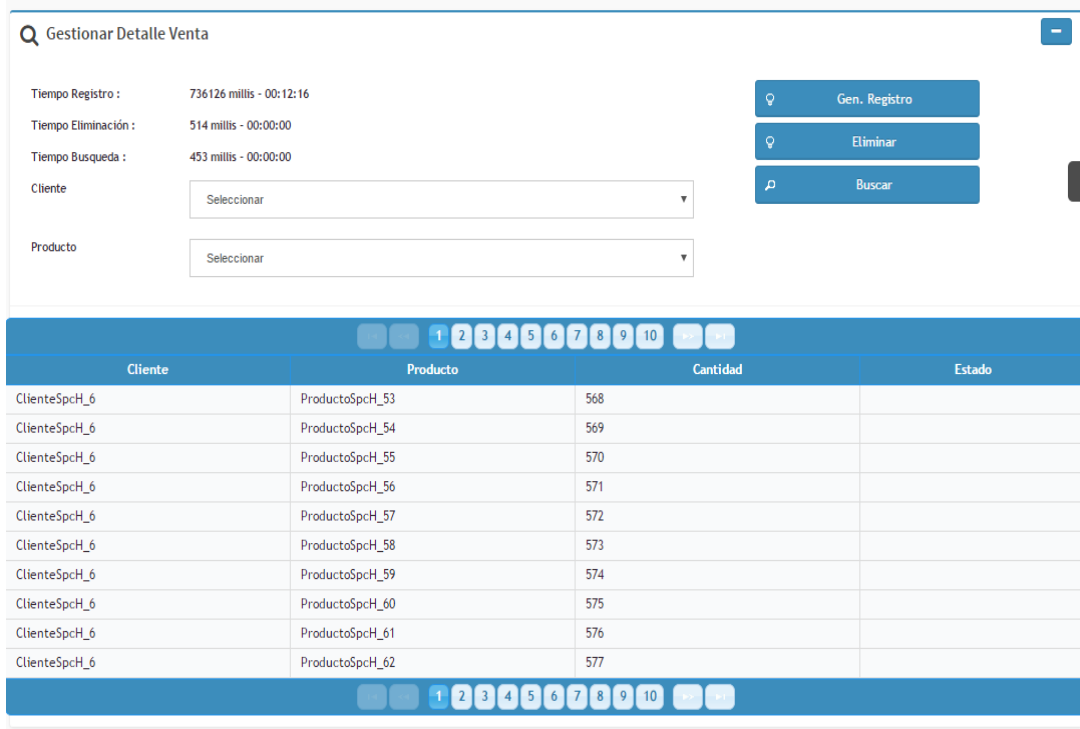


Figura N° 4.30: Tiempos en la generación de detalle de ventas con SpcMid.

MEDIDAS REALIZADAS CON SPCH

En estas dos secciones, generar clientes, generar productos se midió el tiempo de respuesta en el acceso a las bases de datos distribuidos de forma transparente de acuerdo a cada evento de Gen. Registro, Activar, Desactivar, Eliminar y Buscar.

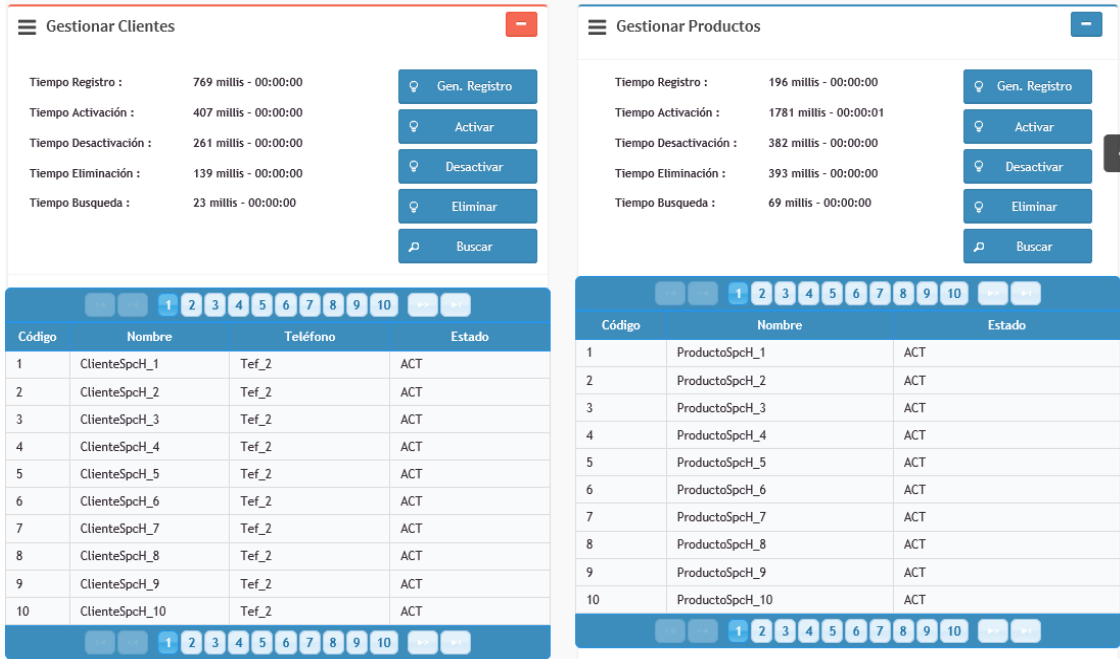


Figura N° 4.31: Tiempos en la generación de clientes y productos con SpcH.

En esta tercera sección se genera datos para el detalle de venta, se elimina los detalles y se busca los detalles, estas funcionalidades permitieron obtener medidas reales en el acceso a las bases de datos distribuidas de forma transparente.

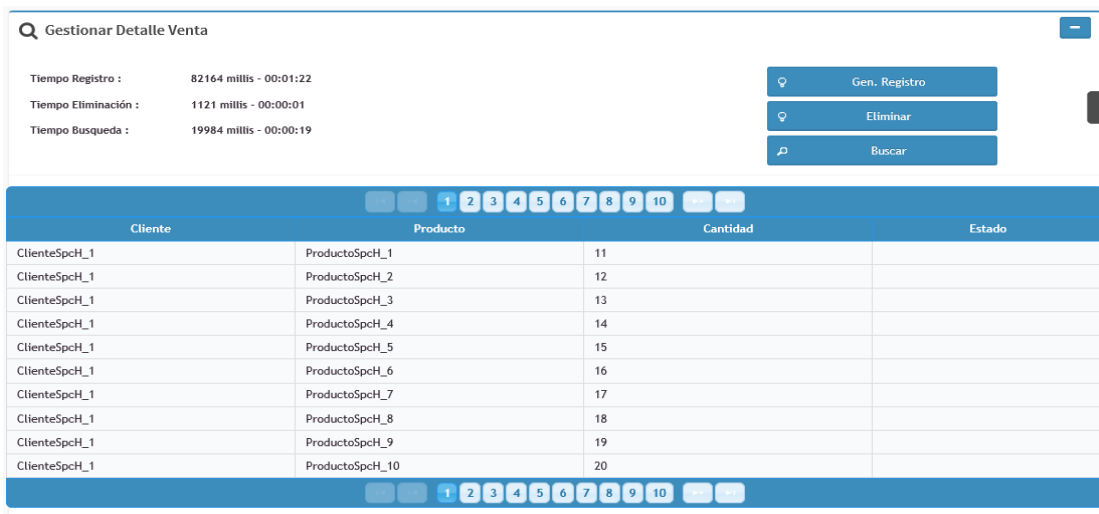


Figura N° 4.32: Tiempos en la generación de detalle de ventas con SpcH.

b. Elaborar las tablas comparativas de eficiencia

Se procede a formalizar las medidas comparativas en dos tablas elaboradas con los indicadores de eficiencia definida en el acceso a las bases de datos distribuida.

Número	Indicador	Abreviatura	Unidad	Abreviatura
1	Tiempo de acceso a los datos.	TA	Milisegundos	ms
2	Líneas de código implementado.	LC	Número	#
3	Memoria heap usada por el proceso.	MU	Megabyte	mb
4	CPU utilizado por el proceso	CP	Porcentaje	%
5	Cantidad de nodos recorridos.	CN	Número	#
6	Alto acoplamiento de lógica y acceso a datos.	AA	Si/No	s/n
7	Overhead o carga innecesaria.	OV	Si/No	s/n
8	Incompatibilidad de tipos de dato.	ID	Si/No	s/n
9	Multiplataforma y transparencia del componente	MT	Si/No	s/n

Tabla N°4.21: Indicadores de eficiencia de acceso a datos distribuidos.

CONSULTAS		INDICADORES									
N°	Consultas y operaciones a base de datos distribuida.	TA (ms)	LC (#)	MU (mb)	CP (%)	NC (#)	AA (s/n)	OV (mb)	ID (s/n)	MT (s/n)	
1	Generar 100 clientes.	391	1	6	15	2	n	n	n	s	
2	Activar 100 clientes.	250	1	6	15	2	n	n	n	s	
3	Desactivar 100 clientes.	250	1	6	15	2	n	n	n	s	
4	Eliminar 100 clientes.	297	1	1	10	2	n	n	n	s	
5	Buscar 100 clientes.	1322	12	16	18	2	n	n	n	s	
6	Buscar 100 clientes activados.	192	12	16	18	2	n	n	n	s	
7	Buscar 100 clientes desactivados.	219	12	16	18	2	n	n	n	s	
8	Generar 100 productos.	406	1	6	15	2	n	n	n	s	
9	Activar 100 productos.	172	1	6	15	2	n	n	n	s	
10	Desactivar 100 productos.	160	1	6	15	2	n	n	n	s	
11	Eliminar 100 productos.	275	1	6	12	2	n	n	n	s	
12	Buscar 100	972	12	16	15	2	n	n	n	s	

	productos.									
13	Buscar 100 productos activados.	276	12	16	15	2	n	n	n	s
14	Buscar 100 productos desactivados.	200	12	16	15	2	n	n	n	s
15	Generar 100* 100 ventas detalle.	9513	1	28	25	2	n	n	n	s
16	Buscar 100* 100 ventas.	989	19	50	26	2	n	n	n	s
17	Eliminar 100* 100 ventas.	218	1	10	20	2	n	n	n	s
18	Buscar ventas por cliente.	172	19	30	24	2	n	n	n	s
19	Buscar ventas por producto.	141	19	30	24	2	n	n	n	s
20	Buscar ventas por cliente y productos.	86	19	30	24	2	n	n	n	s
21	Buscar 10000 clientes.	954	12	52	25	2	n	n	n	s
22	Buscar 10000 productos.	919	12	52	35	2	n	n	n	s
23	Buscar 100000 clientes.	1005	12	62	38	2	n	n	n	s
24	Buscar 100000 productos.	1012	12	62	38	2	n	n	n	s
25	Buscar 10000* 10000 ventas.	1002	19	70	48	2	n	n	n	s
26	Buscar 50000 * 50000 ventas.	13473	19	72	48	2	n	n	n	s
27	Eliminar 1000 ventas.	315	1	12	30	2	n	n	n	s
28	Eliminar 100000 ventas.	380	1	15	32	2	n	n	n	s
29	Eliminar 10000* 10000 ventas.	450	1	19	38	2	n	n	n	s
30	Eliminar 50000 * 50000 ventas.	510	1	20	40	2	n	n	n	s

Tabla N°4.22: Medias de los indicadores del componente software desarrollado, Middleware SQL, para el acceso a las BDD.

CONSULTAS		INDICADORES								
N°	Consultas y operaciones a base de datos distribuida.	TA (ms)	LC (#)	MU (mb)	CP (%)	NC (#)	AA (s/n)	OV (mb)	ID (s/n)	MT (s/n)
1	Generar 100 clientes.	1710	2	126	40	3	s	s	s	n
2	Activar 100	424	7	126	40	3	s	s	s	n

	clientes.										
3	Desactivar 100 clientes.	438	7	126	40	3	s	s	s	n	
4	Eliminar 100 clientes.	291	2	124	20	3	n	n	s	n	
5	Buscar 100 clientes.	817	21	124	42	3	s	s	s	n	
6	Buscar 100 clientes activados.	68	21	124	42	3	s	s	s	n	
7	Buscar 100 clientes desactivados.	31	21	124	45	3	s	s	s	n	
8	Generar 100 productos.	1204	2	126	40	3	s	s	s	n	
9	Activar 100 productos.	268	7	126	41	3	s	s	s	n	
10	Desactivar 100 productos.	250	7	126	41	3	s	s	s	n	
11	Eliminar 100 productos.	138	2	124	40	3	n	n	s	n	
12	Buscar 100 productos	44	21	124	42	3	s	s	s	n	
13	Buscar 100 productos activados.	25	21	124	45	3	s	s	s	n	
14	Buscar 100 productos desactivados.	36	21	124	45	3	s	s	s	n	
15	Generar 100* 100 ventas detalle.	83570	2	147	60	3	s	s	s	n	
16	Buscar 100* 100 ventas.	22970	12	135	57	3	s	s	s	n	
17	Eliminar 100* 100 ventas.	57345	2	130	55	3	n	n	s	n	
18	Buscar ventas por cliente.	229	2	110	80	3	s	s	s	n	
19	Buscar ventas por producto.	231	12	110	45	3	s	s	s	n	
20	Buscar ventas por cliente y productos.	45	12	110	45	3	s	s	s	n	
21	Buscar 10000 clientes	27743	21	135	50	3	s	s	s	n	
22	Buscar 10000 productos.	18088	21	135	50	3	s	s	s	n	
23	Buscar 100000 clientes.	22566	21	139	60	3	s	s	s	n	
24	Buscar 100000 productos.	22766	21	139	60	3	s	s	s	n	
25	Buscar 10000* 10000 ventas.	227629	12	205	75	3	s	s	s	n	
26	Buscar 50000 * 50000 ventas.	2783838	12	270	78	3	s	s	s	n	

27	Eliminar 1000 ventas.	5034	2	134	80	3	n	n	s	n
28	Eliminar 100000 ventas.	8735	2	137	52	3	n	n	s	n
29	Eliminar 10000* 10000 ventas.	9735	2	142	60	3	n	n	s	n
30	Eliminar 50000 * 50000 ventas.	10454	2	145	62	3	n	n	s	n

Tabla N°4.23: Medias de los indicadores del componente software, Framework Hibernate, para el acceso a las BDD.

4.1.5.5. GRÁFICA DE TRABAJO PENDIENTE

Se muestra el esfuerzo restante desde el inicio de la iteración en el día cero, hasta el final de la iteración que fue determinada en 10 días.

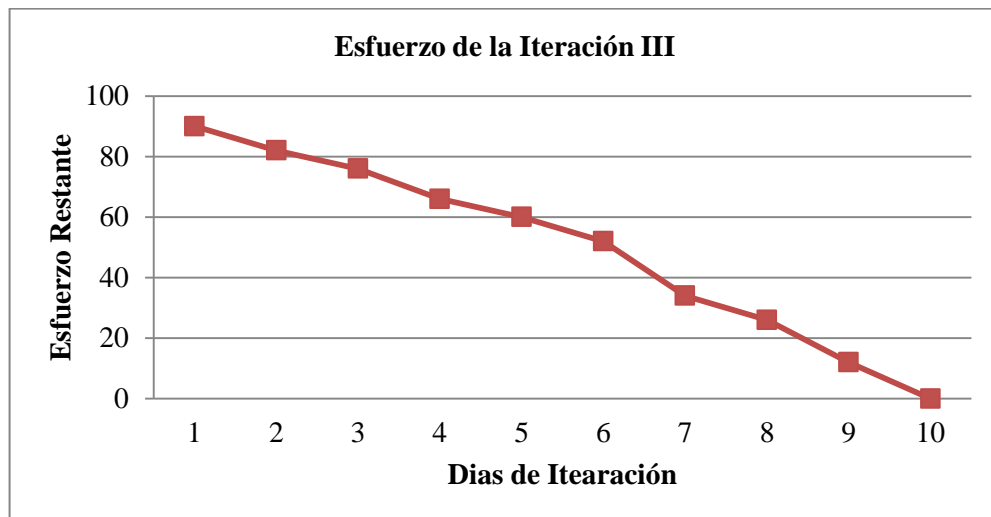


Figura N° 4.33: Gráfica de trabajo pendiente de la iteración III.

Existe un decremento constante del esfuerzo requerido en el tiempo, para lograr completar todos los requerimientos al final de los 10 días de la iteración.

4.1.5.6. REUNIÓN DE REVISIÓN DE LA ITERACIÓN

La tercera reunión de revisión fue realizada para mostrar el incremento del desarrollo del proyecto a lo largo de la tercera iteración, junto con el facilitador y propietario del producto se revisó los resultados, obteniendo una evaluación favorable del detalle en el diseño alcanzado, se presentó la matriz de comparación de indicadores de eficiencia. Así mismo al finalizar la reunión se a levantar la acta de finalización del proyecto quedado satisfecho los responsables del proyecto.

4.2 ANÁLISIS ESTADÍSTICO

A. PRUEBA DE HIPÓTESIS PARA TIEMPO DE ACCESO (TA)

Para las muestras del indicador tiempo de acceso, del componente software denominado Middleware SQL y el Framework Hibernate se aplica la diferencia de medias y el estadístico t student puesto que las muestras son memores o iguales 30 y no se conoce la varianza poblacional.

Datos tomados de la tabla N° 4.21 y 4.22.

Tamaños de la muestra:

$n_1 = 30$ Tiempos de acceso con Middleware SQL.

$n_2 = 30$ Tiempos de acceso con Framework Hibernate.

Cálculo de las medias muestrales:

$$\overline{TA}_1 = \frac{\sum_{i=1}^{n_1} TA_i}{n_1} = 1217.37$$

$$\overline{TA}_2 = \frac{\sum_{i=1}^{n_2} TA_i}{n_2} = 110224.07$$

Plantear hipótesis para la diferencia de medias:

$$H_0 : u_m < u_h$$

$$H_1 : u_m \geq u_h$$

Cálculo de las varianzas muestrales:

$$S_1 = \sqrt{\frac{\sum_{i=1}^{n_1} (\overline{TA}_1 - TA_i)^2}{n_1 - 1}} = 2862.94$$

$$S_2 = \sqrt{\frac{\sum_{i=1}^{n_2} (\overline{TA}_2 - TA_i)^2}{n_2 - 2}} = 506860.67$$

Calculamos la variación muestral para la diferencia de medias:

$$S = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} = 358410.34$$

Cálculo del valor crítico para un nivel de confianza del 95%:

$$t_{\alpha/2, n_1 + n_2 - 2} = t_{0.025, 58} = 2.002 \text{ Valor crítico con 58 grados de libertad.}$$

Calculamos el estadístico t para diferencia de medias:

$$t = \frac{(\overline{TA}_1 - \overline{TA}_2)}{s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = -1.67$$

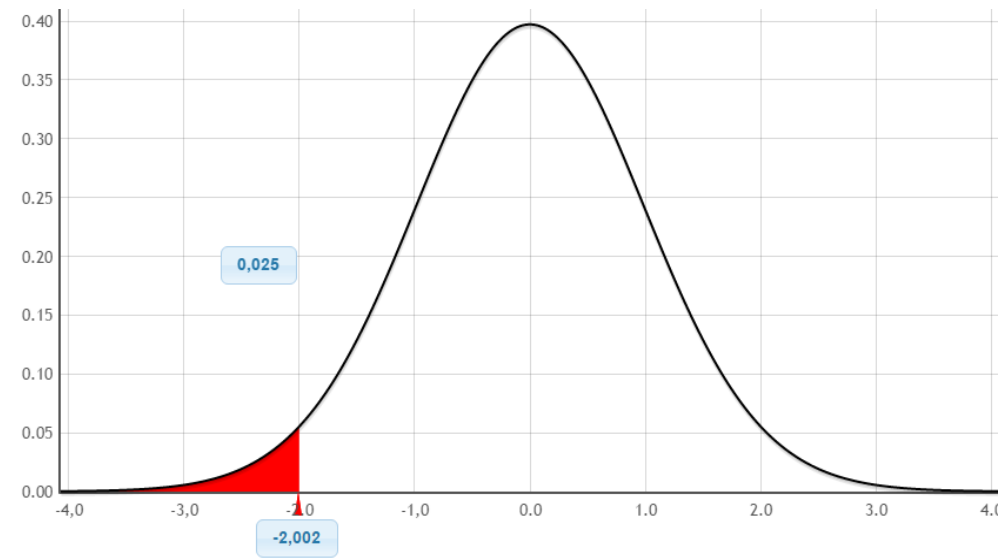


Figura N° 4.34: Curva t – student con 58 grados de libertad para TA.

Como el estadístico $t^* < t_{0,025,58}$ y $-1.67 \in RA$, se rechaza H_1 y se acepta H_0 .

∴ Según la muestra, se usa el estadístico t con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, los tiempos de acceso a datos de los softwares que utiliza la técnica de Middleware SQL es significativamente inferior a los que utilizan la técnica de Framework Hibernate.

B. PRUEBA DE HIPÓTESIS PARA LÍNEAS DE CÓDIGO (LC)

Para las muestras del indicador líneas de código implementado en acceso a datos del componente software denominada Middleware SQL y el Framework Hibernate se aplica la diferencia de medias y el estadístico t student, puesto que las muestras son memores o iguales 30 y no se conoce la varianza poblacional.

Datos tomados de la tabla N° 4.21 y 4.22.

Tamaños de la muestra:

$n_1 = 30$ Lineas de código de acceso a los datos con Middleware SQL.

$n_2 = 30$ Lineas de código de acceso a los datos con Framework Hibernate.

Cálculo de las medias muestrales:

$$\overline{LC}_1 = \frac{\sum_{i=1}^{n_1} LC_i}{n_1} = 8.27$$

$$\overline{LC}_2 = \frac{\sum_{i=1}^{n_2} LC_i}{n_2} = 10.67$$

Plantear hipótesis para la diferencia de medias:

$$H_0 : u_m \leq u_h$$

$$H_1 : u_m > u_h$$

Cálculo de las varianzas muestrales:

$$S_1 = \sqrt{\frac{\sum_{i=1}^{n_1} (\overline{LC}_1 - LC_i)^2}{n_1 - 1}} = 7.36$$

$$S_2 = \sqrt{\frac{\sum_{i=1}^{n_2} (\overline{LC}_2 - LC_i)^2}{n_2 - 2}} = 8.21$$

Calculamos la variación muestral para la diferencia de medias:

$$S = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} = 7.80$$

Cálculo del valor crítico para un nivel de confianza del 95%:

$$t_{\alpha/2, n_1 + n_2 - 2} = t_{0.025, 58} = 2.002 \text{ Valor crítico con 58 grados de libertad.}$$

Calculamos el estadístico t para diferencia de medias:

$$t = \frac{(\overline{LC}_1 - \overline{LC}_2)}{s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = -1.69$$

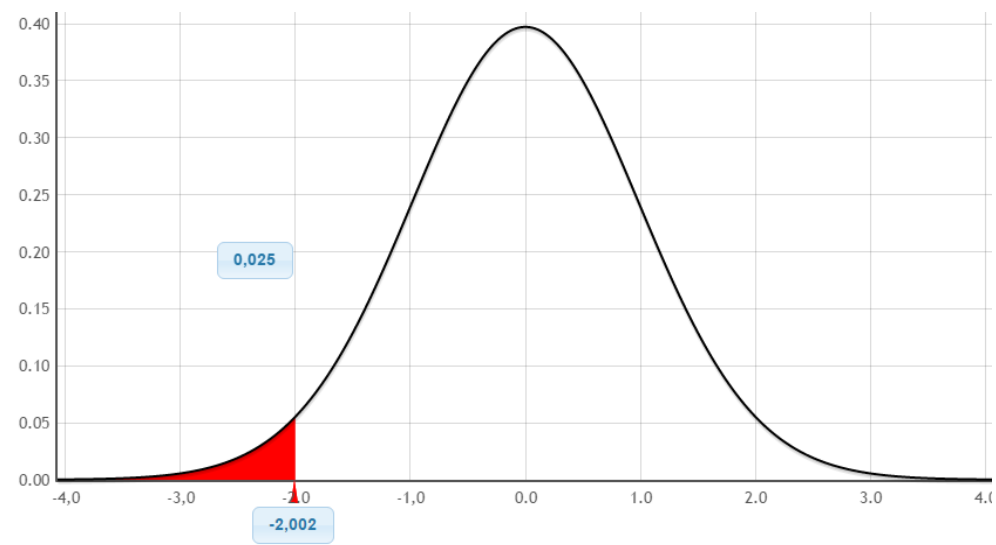


Figura N° 4.35: Curva t – student con 58 grados de libertad para LC.

Como el estadístico $t^* < t_{0.025,58} y - 1.69 \in RA$, se acepta H_0 y se rechaza H_1 .

∴ Según la muestra, se usa el estadístico t con 58 grados de libertad y un nivel de confianza del 95%, $u_m \leq u_h$, las líneas de código implementado en el acceso a datos de los softwares que utiliza la técnica de Middleware SQL son significativamente menores o iguales a los que utilizan la técnica de Framework Hibernate.

C. PRUEBA DE HIPÓTESIS PARA MEMORIA UTILIZADA (MU)

Para las muestras del indicador memoria heap utilizada del componente software denominada Middleware SQL y el Framework Hibernate, se aplica la diferencia de medias y el estadístico t student puesto que las muestras son memores o iguales 30 y no se conoce la varianza poblacional.

Datos tomados de la tabla N° 4.21 y 4.22.

Tamaños de la muestra:

$n_1 = 30$ Memoria heap utilizada con Middleware SQL.

$n_2 = 30$ Memoria heap utilizada con Framework Hibernate.

Cálculo de las medias muestrales:

$$\overline{MU}_1 = \frac{\sum_{i=1}^{n_1} MU_i}{n_1} = 25.1$$

$$\overline{MU}_2 = \frac{\sum_{i=1}^{n_2} MU_i}{n_2} = 135.7$$

Plantear hipótesis para la diferencia de medias:

$$H_0 : u_m < u_h$$

$$H_1 : u_m \geq u_h$$

Calculo de las varianzas muestrales:

$$S_1 = \sqrt{\frac{\sum_{i=1}^{n_1} (\overline{MU}_1 - MU_i)^2}{n_1 - 1}} = 21.40$$

$$S_2 = \sqrt{\frac{\sum_{i=1}^{n_2} (\overline{MU}_2 - MU_i)^2}{n_2 - 2}} = 30.40$$

Calculamos la variación muestral para la diferencia de medias:

$$S = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} = 26.29$$

Cálculo del valor crítico para un nivel de confianza del 95%:

$$t_{\alpha/2, n_1 + n_2 - 2} = t_{0.025, 58} = 2.002, \text{ valor crítico con 58 grados de libertad.}$$

Calculamos el estadístico t para diferencia de medias:

$$t = \frac{(\overline{MU}_2 - \overline{MU}_1)}{s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = 23.05$$

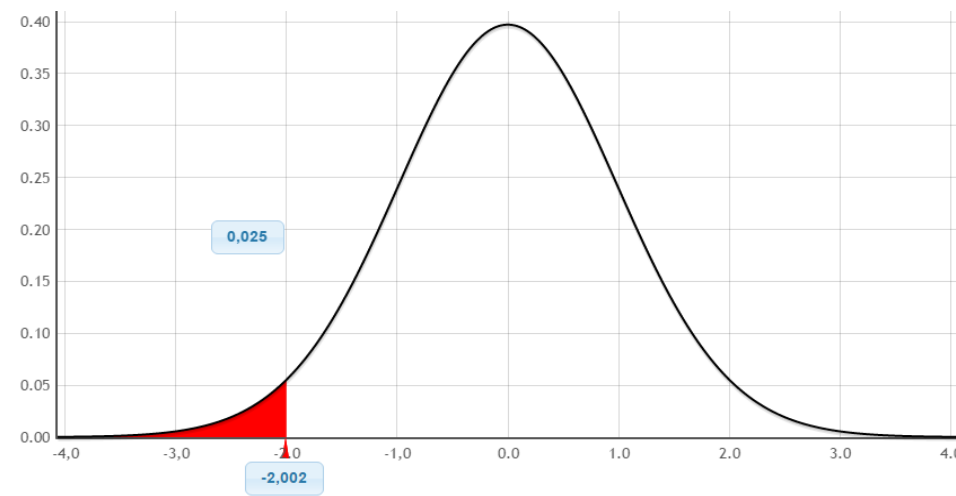


Figura N° 4.36: Curva t – student con 58 grados de libertad para MU.

Como el estadístico $t^* < t_{0.025, 58}$ y $23.05 \in RA$ se rechaza H_1 y se acepta H_0 .

∴ Según la muestra se usa el estadístico t con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, la memoria heap utilizada en el acceso a datos de los softwares que utiliza la técnica de Middleware SQL es significativamente menor a los que utilizan la técnica de Framework Hibernate.

D. PRUEBA DE HIPÓTESIS PARA CPU UTILIZADA (CP)

Para las muestras del indicador CPU utilizada por componente software denominada Middleware SQL y el Framework Hibernate, se aplica la diferencia de medias y el estadístico t student puesto que las muestras son menores o iguales 30 y no se conoce la varianza poblacional.

Datos tomados de la tabla N° 4.21 y 4.22.

Tamaños de la muestra:

$n_1 = 30$ CPU utilizada con Middleware SQL.

$n_2 = 30$ CPU utilizada con Framework Hibernate.

Cálculo de las medias muestrales:

$$\overline{CP}_1 = \frac{\sum_{i=1}^{n_1} CP_i}{n_1} = 24.2$$

$$\overline{CP}_2 = \frac{\sum_{i=1}^{n_2} CP_i}{n_2} = 51.07$$

Plantear hipótesis para la diferencia de medias:

$$H_0 : u_m < u_h$$

$$H_1 : u_m \geq u_h$$

Calculo de las varianzas muestrales:

$$S_1 = \sqrt{\frac{\sum_{i=1}^{n_1} (\overline{CP}_1 - CP_i)^2}{n_1 - 1}} = 10.85$$

$$S_2 = \sqrt{\frac{\sum_{i=1}^{n_2} (\overline{CP}_2 - CP_i)^2}{n_2 - 2}} = 13.99$$

Calculamos la variación muestral para la diferencia de medias:

$$S = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} = 12.52$$

Cálculo del valor crítico para un nivel de confianza del 95%:

$t_{\alpha/2, n_1 + n_2 - 2} = t_{0.025, 58} = 2.002$, valor crítico con 58 grados de libertad.

Calculamos el estadístico t para diferencia de medias:

$$t = \frac{(\overline{CP}_2 - \overline{CP}_1)}{s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = 11.75$$

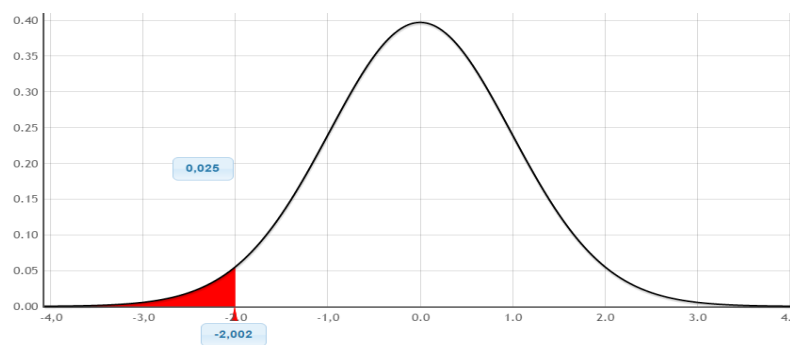


Figura N° 4.37: Curva t – student con 58 grados de libertad para CP.

Como el estadístico $t^* < t_{0.025,58}$ y $11.75 \notin RR$, se rechaza H_1 y se acepta H_0 .

\therefore Según la muestra se usa el estadístico t con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, la CPU utilizada en el acceso a datos de los softwares que utiliza la técnica de Middleware SQL es significativamente menor a los que utilizan la técnica de Framework Hibernate.

E. PRUEBA DE HIPÓTESIS PARA NODOS RECORRIDOS (CN)

Para las muestras del indicador Nodos Recorridos por los Servicios para acceso a los datos utilizada del componente software denominado Middleware SQL y el Framework Hibernate, se aplica la diferencia de medias y el estadístico t student puesto que las muestras son menores o iguales 30 y no se conoce la varianza poblacional.

Datos tomados de la tabla N° 4.21 y 4.22.

Tamaños de la muestra:

$n_1 = 30$ Cantidad de nodos recorridos utilizando Middleware SQL.

$n_2 = 30$ Cantidad de nodos recorridos utilizando Framework Hibernate.

Cálculo de las medias muestrales:

$$\overline{CN}_1 = \frac{\sum_{i=1}^{n_1} CN_i}{n_1} = 2$$

$$\overline{CN}_2 = \frac{\sum_{i=1}^{n_2} CN_i}{n_2} = 3$$

Plantear hipótesis para la diferencia de medias:

$$H_0 : u_m < u_h$$

$$H_1 : u_m \geq u_h$$

Calculo de las varianzas muestrales:

$$S_1 = \sqrt{\frac{\sum_{i=1}^{n_1} (\overline{CN}_1 - CN_i)^2}{n_1 - 1}} = 0$$

$$S_2 = \sqrt{\frac{\sum_{i=1}^{n_2} (\overline{CN}_2 - CN_i)^2}{n_2 - 2}} = 0$$

Calculamos la variación muestral para la diferencia de medias:

$$S = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} = 0$$

Cálculo del valor crítico para un nivel de confianza del 95%:

$$t_{\alpha/2, n_1 + n_2 - 2} = t_{0,025,58} = 2.002, \text{ valor crítico con 58 grados de libertad.}$$

Calculamos el estadístico t para diferencia de medias:

$$t = \frac{(\overline{CN}_2 - \overline{CN}_1)}{s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = \alpha$$

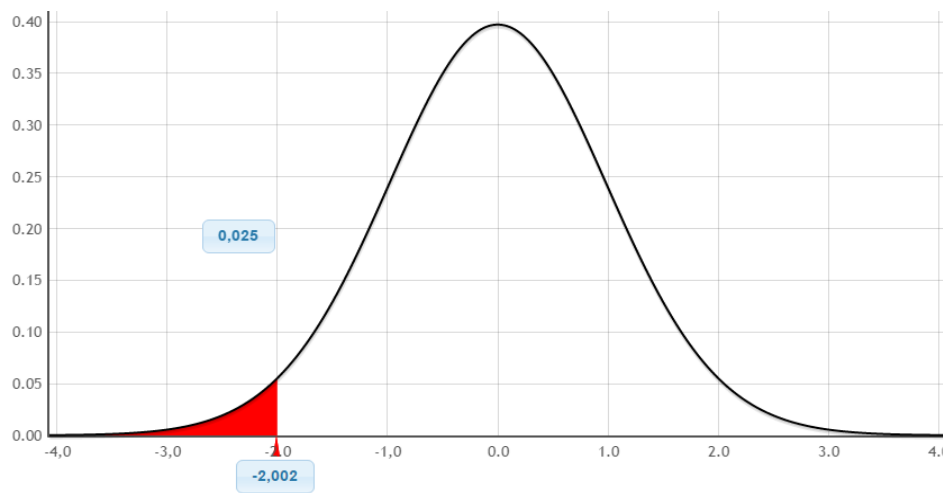


Figura N° 4.38: Curva t – student con 58 grados de libertad para CN.

Como $t < t_{0,025,58}$ y $\alpha \notin RR$ se rechaza H_1 y se acepta H_0 .

∴ Según la muestra se usa el estadístico t con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, la cantidad de nodos recorridos para el acceso a datos por los softwares que utiliza la técnica de Middleware SQL es significativamente menor a los que utilizan la técnica de Framework Hibernate.

F. PRUEBA DE HIPÓTESIS PARA ALTO ACOPLAMIENTO DE LÓGICA Y ACCESO A DATOS (AA)

Para las muestras del indicador alto acoplamiento en el acceso a datos del componente software desarrollado denominada Middleware SQL y el Framework Hibernate. Se utiliza el estadístico ji-cuadrado, puesto que las muestras son memores o iguales 30, no paramétricas y cualitativas.

Datos tomados de la tabla N° 4.21 y 4.22.

Alto Acoplamiento de lógica y acceso a datos			
Componente Software	Si	No	Total
Middleware SQL	a=30	b=0	a+b=30
Framework Hibernate	c=7	d=21	c+d=30
Total	a+c=37	b+d=21	n=60

Tabla N°4.24: Tabla de consistencia para prueba de hipótesis de AA.

Planteamos las hipótesis:

H_0 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente igual o mayor acoplamiento de lógica y acceso a datos que con la técnica de Framework Hibernate.

H_1 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor acoplamiento de lógica y acceso a datos que con la técnica de Framework Hibernate.

Calculamos los valores estimados:

$$E_{11} = \frac{(a+b)x(a+c)}{n} = 18.5$$

$$E_{12} = \frac{(a+b)x(b+d)}{n} = 10.5$$

$$E_{21} = \frac{(c+d)x(a+c)}{n} = 18.5$$

$$E_{22} = \frac{(c+d)x(b+d)}{n} = 10.5$$

Calcular el valor del estadístico χ^2 :

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^k \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{(30 - 18.5)^2}{18.5} + \frac{(0 - 10.5)^2}{10.5} + \frac{(7 - 18.5)^2}{18.5} + \frac{(21 - 10.5)^2}{10.5}$$

$$\chi^2 = 7.15 + 10.5 + 7.15 + 10.5 = 35.3$$

Calculamos los grados de libertad, región crítica para un nivel de confianza del 95%:

$$gl = (2 - 1) \times (2 - 1) = 1 \text{ Y el nivel de confianza } \alpha = 0.05$$

ji-cuadrado para la región crítica es 3.84

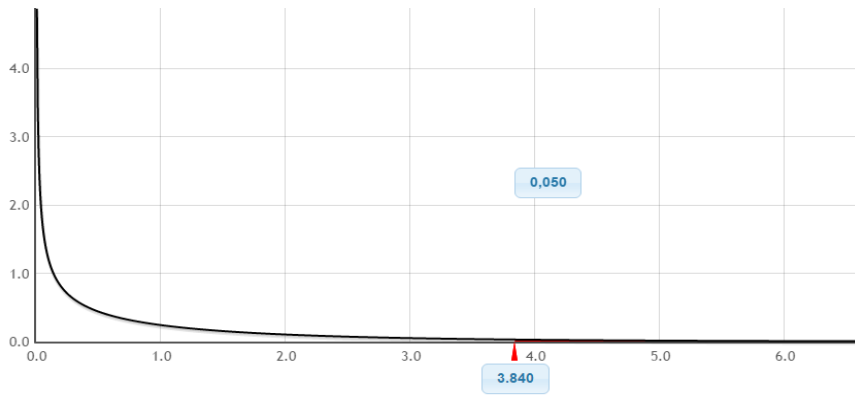


Figura N° 4.39: Curva ji – cuadrado con un grado de libertad para AA.

Como se observa $\chi_{tabla}^2 < \chi_{obs}^2 : 3.840 < 35.30$ se rechaza la H_0 y se acepta H_1 .

∴ Según la muestra se usa χ^2 , ji-cuadrada, con un grado de libertad, los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor acoplamiento de lógica y acceso a datos que con la técnica de Framework Hibernate.

G. PRUEBA DE HIPÓTESIS PARA CARGA INNECESARIA (OV)

Para las muestras del indicador overhead o carga innecesaria en el acceso a datos del componente software desarrollado denominada Middleware SQL y el Framework Hibernate. Se utiliza el estadístico ji-cuadrado, puesto que las muestras son menores o iguales 30, no paramétricas y cualitativas.

Datos tomados de la tabla N° 4.21 y 4.22.

Overhead o Carga Innecesaria			
Componente Software	Si	No	Total
Middleware SQL	a=30	b=0	a+b=30
Framework Hibernate	c=7	d=21	c+d=30
Total	a+c=37	b+d=21	n=60

Tabla N°4.25: Tabla de consistencia para prueba de hipótesis de OV.

Planteamos las hipótesis:

H_0 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente igual o mayor carga innecesaria en el acceso a datos que con la técnica de Framework Hibernate.

H_1 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor carga innecesaria en el acceso a datos que con la técnica de Framework Hibernate.

Calculamos los valores estimados:

$$E_{11} = \frac{(a + b)x(a + c)}{n} = 18.5$$

$$E_{12} = \frac{(a + b)x(b + d)}{n} = 10.5$$

$$E_{21} = \frac{(c + d)x(a + c)}{n} = 18.5$$

$$E_{22} = \frac{(c + d)x(b + d)}{n} = 10.5$$

Calcular el valor del estadístico χ^2 :

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^k \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{(30 - 18.5)^2}{18.5} + \frac{(0 - 10.5)^2}{10.5} + \frac{(7 - 18.5)^2}{18.5} + \frac{(21 - 10.5)^2}{10.5}$$

$$\chi^2 = 7.15 + 10.5 + 7.15 + 10.5 = 35.3$$

Calculamos los grados de libertad, región crítica para un nivel de confianza del 95%:

$$gl = (2 - 1)x(2 - 1) = 1 \text{ Y el nivel de confianza } \alpha = 0.05$$

ji-cuadrado para la región crítica es 3.84

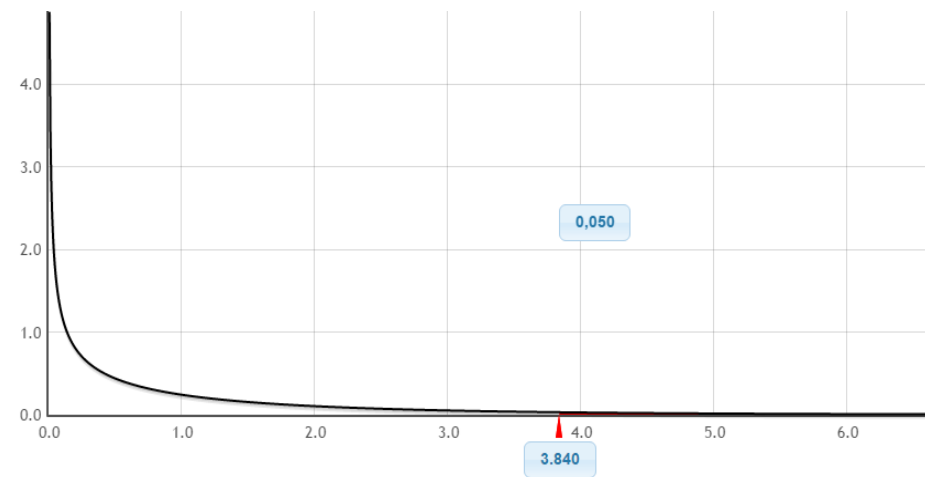


Figura N° 4.40: Curva ji – cuadrado con 1 grado de libertad para OV.

Como se observa $\chi_{tabla}^2 < \chi_{obs}^2$: $3.840 < 35.30$ se rechaza la H_0 y se acepta H_1 .

∴ Según la muestra se usa χ^2 , ji-cuadrada, con un grado de libertad, los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor carga innecesaria en el acceso a datos que con la técnica de Framework Hibernate, por consiguiente tendremos código más limpio y necesario.

H. PRUEBA DE HIPÓTESIS PARA INCOMPATIBILIDAD DE TIPOS DE DATO (ID)

Para las muestras del indicador incompatibilidad de tipo de dato en el acceso a datos del componente software denominado Middleware SQL y el Framework Hibernate. Se utiliza el estadístico ji-cuadrado, puesto que las muestras son menores o iguales 30, no paramétricas y cualitativas.

Datos tomados de la tabla N° 4.21 y 4.22.

Incompatibilidad de Tipo de Dato			
Componente Software	Si	No	Total
Middleware SQL	a=30	b=0	a+b=30
Framework Hibernate	c=7	d=21	c+d=30
Total	a+c=37	b+d=21	n=60

Tabla N°4.26: Tabla de consistencia para prueba de hipótesis de ID.

Planteamos las hipótesis:

H_0 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente igual o mayor incompatibilidad de tipos de dato en el acceso a los datos que con la técnica de Framework Hibernate.

H_1 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor incompatibilidad de tipos de dato en el acceso a los datos que con la técnica de Framework Hibernate.

Calculamos los valores estimados:

$$E_{11} = \frac{(a+b)x(a+c)}{n} = 18.5$$

$$E_{12} = \frac{(a+b)x(b+d)}{n} = 10.5$$

$$E_{21} = \frac{(c+d)x(a+c)}{n} = 18.5$$

$$E_{11} = \frac{(a+b)x(a+c)}{n} = 10.5$$

Calcular el valor del estadístico χ^2 :

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^k \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{(30 - 18.5)^2}{18.5} + \frac{(0 - 10.5)^2}{10.5} + \frac{(7 - 18.5)^2}{18.5} + \frac{(21 - 10.5)^2}{10.5}$$

$$\chi^2 = 7.15 + 10.5 + 7.15 + 10.5 = 35.3$$

Calculamos los grados de libertad, región crítica para un nivel de confianza del 95%:

$$gl = (2 - 1) \times (2 - 1) = 1 \text{ Y el nivel de confianza } \alpha = 0.05$$

ji-cuadrado para la región crítica es 3.84

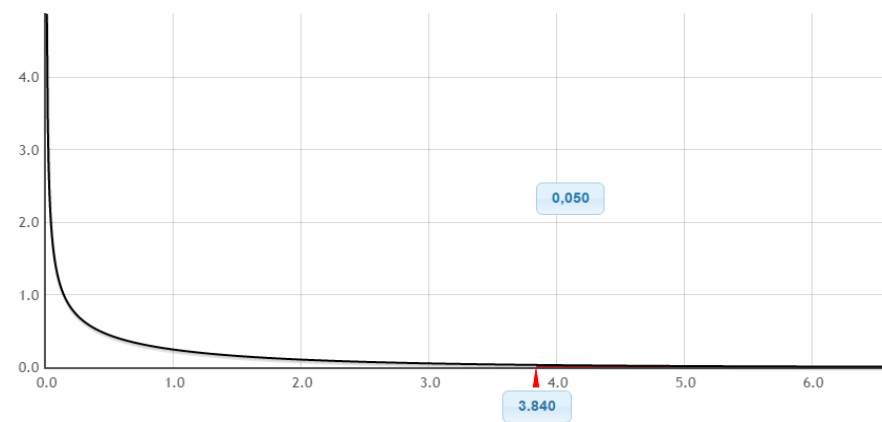


Figura N° 4.41: Curva ji – cuadrado con 1 grado de libertad para ID.

Como se observa $\chi_{tabla}^2 < \chi_{obs}^2 : 3.840 < 35.30$ se rechaza la H_0 y se acepta H_1 .

∴ Según la muestra se usa χ^2 , ji-cuadrada, con un grado de libertad, los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor incompatibilidad de tipos de dato en el acceso a los datos que con la técnica de Framework Hibernate.

I. PRUEBA DE HIPÓTESIS MULTIPLATAFORMA Y TRANSPARENCIA DEL COMPONENTE (MT)

Para las muestras del indicador multiplataforma y transparencia del componente en el acceso a datos del componente software desarrollado denominada Middleware SQL y el Framework Hibernate. Se utiliza el estadístico ji-cuadrado, puesto que las muestras son menores o iguales 30, no paramétricas y cualitativas.

Datos tomados de la tabla N° 4.21 y 4.22.

Multiplataforma y Transparencia del Componente			
Componente Software	Si	No	Total
Middleware SQL	a=30	b=0	a+b=30
Framework Hibernate	c=0	d=30	c+d=30
Total	a+c=30	b+d=30	n=60

Tabla N°4.28: Tabla de consistencia para prueba de hipótesis de MT.

Planteamos las hipótesis:

H_0 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente igual o menor adaptabilidad al entorno multiplataforma y transparencia en el acceso a los datos que con la técnica de Framework Hibernate.

H_1 : Los softwares desarrollados con la técnica de Middleware SQL tienen significativamente mayor adaptabilidad al entorno multiplataforma y transparencia en el acceso a los datos que con la técnica de Framework Hibernate.

Calculamos los valores estimados:

$$E_{11} = \frac{(a+b)x(a+c)}{n} = 15$$

$$E_{12} = \frac{(a+b)x(b+d)}{n} = 15$$

$$E_{21} = \frac{(c+d)x(a+c)}{n} = 15$$

$$E_{22} = \frac{(c+d)x(b+d)}{n} = 15$$

Calcular el valor del estadístico χ^2 :

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^k \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{(30 - 15)^2}{15} + \frac{(0 - 15)^2}{15} + \frac{(0 - 15)^2}{15} + \frac{(30 - 15)^2}{15}$$

$$\chi^2 = 15 + 15 + 15 + 15 = 60$$

Calculamos los grados de libertad, región crítica para un nivel de confianza del 95%:

$$gl = (2 - 1)x(2 - 1) = 1 \text{ Y el nivel de confianza } \alpha = 0.05$$

ji-cuadrado para la región crítica es 3.84

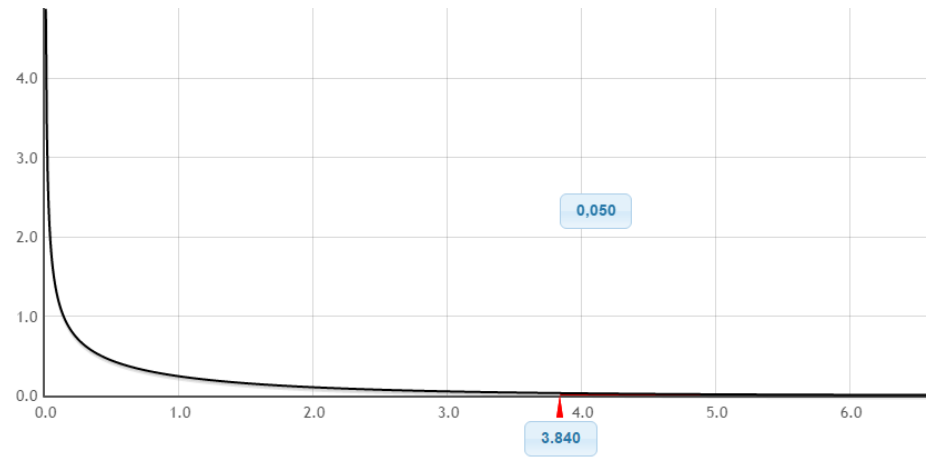


Figura N° 4.42: Curva ji – cuadrado con 1 grado de libertad para MT.

Como se observa $x_{tabla}^2 < x_{obs}^2 : 3.840 < 60$ se rechaza la H_0 y se acepta H_1 .

∴ Según la muestra se usa χ^2 , ji-cuadrada, con un grado de libertad, los softwares desarrollados con la técnica de Middleware SQL tienen significativamente mayor adaptabilidad al entorno multiplataforma y transparencia en el acceso a los datos que con la técnica de Framework Hibernate.

4.3 DISCUSIÓN

Acorde al análisis del resultado y el análisis estadístico t con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, los tiempos de acceso a datos de los softwares que utilizan la técnica de Middleware SQL es significativamente inferior a los que utilizan la técnica de Framework Hibernate, sin embargo Callejas et al. (2011), afirman que el framework Hibernate es robusto, capaz de soportar cantidades inmensas de instrucciones SQL, sin afectar su rendimiento frente a otros frameworks de persistencia. De la misma forma Tuya et al. (2007), afirman que las pruebas para determinar si los tiempos de respuesta del sistema tanto en condiciones normales como en condiciones especiales usando el frameworks, se encuentran dentro de los límites predefinidos. Y según Throughput (2010), afirma que el volumen de trabajo o de información que fluye a través del sistema, así como el tiempo de respuesta, el costo por transacción y medidas de utilización de recursos, para evaluar el rendimiento de una aplicación. Por consiguiente bajo este sustento afirmamos que el tiempo de acceso a las fuentes de datos distribuidos con el uso del Middleware SQL es inferior, de esta manera mejoramos el acceso a la base de datos distribuida.

De la misma forma el análisis del resultado y el análisis estadístico t student con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, las líneas de código implementado en el acceso a datos de los softwares que utiliza la técnica de Middleware SQL son significativamente menor a los que utilizan la técnica de Framework Hibernate, sin embargo O'Neil (2008), sostiene que los ORMs se han convirtiéndose en una práctica necesaria en la manipulación de objetos entre la memoria y su ubicación real en la base de datos, permitiendo al desarrollador definir de forma clara la gestión entre el modelo de objetos y el esquema de base de datos, además de expresar las operaciones de esta última en términos de objetos. Según Benjelloun (2006), estos frameworks de persistencia buscan simplificar el proceso de desarrollo y las líneas de código implementado. Por consiguiente bajo este sustento contrastado, afirmamos que la cantidad de código implementado para el acceso a base de datos distribuida es significativamente menor con la técnica de Middleware SQL que con el Framework Hibernate.

De la misma forma el análisis del resultado y el análisis estadístico t student con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, la memoria heap utilizada en el acceso a datos de los softwares que utiliza la técnica de Middleware SQL es significativamente menor a los que utilizan la técnica de Framework Hibernate, así mismo Nolte (1995), sostiene que en términos generales de sistemas de información, el rendimiento es considerado como la reducción de uso de la CPU, Memoria Heap y la búsqueda de métodos más óptimos para el sistema. Sin embargo Callejas et al. (2011), concluye que Hibernate se podría considerar un ganador y de mayor performance dentro de la evaluación de eficiencia, sin embargo las consultas realizadas con HQL tienen un desempeño bajo, aunque brinda facilidad al programador al momento de realizar consultas, le resta eficiencia al framework puesto que internamente debe realizar la traducción a lenguaje SQL. Por consiguiente estos sustentos nos ayudan a confirmar con el uso de la técnica de Middleware SQL tienen un uso considerablemente menor de la memoria heap.

De la misma forma el análisis del resultado y el análisis estadístico t student con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, la CPU utilizada en el acceso a datos de los softwares que utiliza la técnica de Middleware SQL es

significativamente menor a los que utilizan la técnica de Framework Hibernate, así mismo Nolte (1995), sostiene que en términos generales de sistemas de información, el rendimiento es considerado como la reducción de uso de la CPU, Memoria Heap y la búsqueda de métodos más óptimos para el sistema. Por consiguiente estos sustentos y las pruebas realizadas nos ayudan a confirmar que con el uso de la técnica de Middleware SQL tienen un uso considerablemente menor de la CPU.

De la misma forma el análisis del resultado y el análisis estadístico t student con 58 grados de libertad y un nivel de confianza del 95%, $u_m < u_h$, la cantidad de nodos recorridos por los servicios para el acceso a los datos de los softwares que utiliza la técnica de Middleware SQL es significativamente menor de acuerdo a la teorías de optimización de los web services y los middlewares, como lo menciona Erl(2009), que la solución lógica diseñada acorde con la orientación a servicios puede ser calificado como orientación a servicios optimizados basado en el menor recorrido de nodos entre los servicios.

Del mismo modo, si comparamos los resultados y el análisis estadístico del indicador cualitativo con el estadístico χ^2 , ji-cuadrada, con un grado de libertad y un nivel de confianza del 95%, indicamos que los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor acoplamiento de modelo, lógica y acceso a datos que con la técnica de Framework Hibernate. Sin embargo Richardson (2009), sostiene que los ORMs se han convirtiéndose en una práctica necesaria en la manipulación de objetos entre la memoria y su ubicación real en la base de datos, permitiendo al desarrollador definir de forma clara la gestión entre el modelo de objetos y el esquema de base de datos pero incrementando su acoplamiento al modelo y lógica, además de expresar las operaciones de esta última en términos de objetos. Bajo este sustento afirmamos que con el uso componente software denominado Middleware SQL reducimos el acoplamiento de los objetos, la lógica con el acceso a los datos.

Del mismo modo, si comparamos los resultados y el análisis estadístico del indicador cualitativo con el estadístico χ^2 , ji-cuadrada, con un grado de libertad y un nivel de confianza del 95%, indicamos que los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor carga innecesaria en el acceso a datos que con la técnica de Framework Hibernate. Así como afirman Callejas et al. (2011),

que Hibernate carga los objetos relacionados reduciendo rendimiento en su carga inicial. Bajo este sustento, concluíamos que con el uso componente software denominado Middleware SQL solo extraemos datos necesarios, obtener un código limpio y menor sobrecarga innecesaria.

De la misma forma, si comparamos los resultados y el análisis estadístico del indicador cualitativo con el estadístico χ^2 , ji-cuadrada, con un grado de libertad y un nivel de confianza del 95%, indicamos que los softwares desarrollados con la técnica de Middleware SQL tienen significativamente menor incompatibilidad de tipos de dato en el acceso a los datos que con la técnica de Framework Hibernate. Así como afirma que existen inconvenientes en los tipos de datos de los diferentes DBMS a pesar de manejar el estándar SQL-92 como los campos numéricos fechas generando incompatibilidad de los sistemas (Velasco, 2010). Bajo este sustento concluíamos que con el uso componente software denominado Middleware SQL ayudamos a mejorar que el acceso a los datos sea de forma transparente a los diversos tipos de datos en los DBMS.

Finalmente si comparamos los resultados y el análisis estadístico del indicador cualitativo con el estadístico χ^2 , ji-cuadrada, con un grado de libertad y un nivel de confianza del 95%, indicamos que, los softwares desarrollados con la técnica de Middleware SQL tienen significativamente mayor adaptabilidad al entorno multiplataforma y transparencia en el acceso a los datos que con la técnica de Framework Hibernate. Como concluye Velasco (2010), que con una interfaz especializada es posible asegurar que la transparencia y adaptabilidad a un entorno de sistemas de base de datos distribuida y heterogénea. Con este sustento terminamos afirmado que con el Middleware SQL aseguramos esta transparencia y adaptabilidad a este entorno distribuido y heterogéneo.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- a.** Según el marco teórico desarrollado en el capítulo II, sección 2.2; las técnicas para aplicar el marco de trabajo Scrum y la notación UML, desarrollado en el capítulo III, sección 3.5.3, sustentados en el capítulo II, las secciones 2.2.7 y 2.2.8, los artefactos Scrum obtenidos en el capítulo IV; la pila del proyecto en la tabla N° 4.1, la pila de la primera iteración en la tabla N° 4.2 y 4.3, las tareas desarrollados en la iteración I y principalmente en el análisis de tipos de datos en las tablas N° 4.5 y 4.6, N° 4.69, análisis y elección de patrones de diseño e implementación en la figura N° 4.1 y 4.2, diseño del modelo arquitectónico en la figura N° 4.7 y 4.8, diagrama arquitectónico diseñado y formalizado. Se concluye que con el modelo arquitectónico analizado y diseñado, mejoramos la eficiencia de acceso a las bases de datos distribuidas basado en técnicas de optimización y el principio del software middleware, logrando delegar la responsabilidad a un componente software el acceso de forma transparente y homogénea a las fuentes de datos heterogéneos y distribuidos.
- b.** Según el marco teórico desarrollado en el capítulo II, sección 2.2; las técnicas para aplicar el marco de trabajo Scrum y la notación UML, desarrollado en el capítulo III, sección 3.5.3, sustentados en el capítulo II, las secciones 2.2.7 y 2.2.8, los artefactos Scrum obtenidos en el capítulo IV; la iteración II desarrollo del componente software de la sección 4.3.1; la pila de la segunda iteración en la tabla N° 4.7 y 4.8, las tareas desarrolladas en la iteración II; los requisitos funcionales de la tabla N° 4.9 y 4.18, el diagrama de casos de uso de la figura N° 4.10 y 4.16, los diagramas de clases de diseño en la figura N° 4.11 y 4.24, el diagrama de base de datos relacional de la figura N° 4.12. Desarrollado y probado el componente software denominado middleware SQL y sus contratos Web Services, denominado Middleware WSSQL. Entonces estamos cumpliendo

con la premisa de que la organización, el nivel de abstracción del componente, permite mantener la autonomía, transparencia, ayudar aminorar el inconveniente de las equivalencias de los tipos de datos y la adaptación al entorno multiplataforma para la manipulación de los datos en diferentes motores relacionales logrando mejorar el acceso a las bases de datos distribuía.

- c.** Según el marco teórico desarrollado en el capítulo II, sección 2.2; las técnicas para aplicar el marco de trabajo Scrum y la notación UML, desarrollado en el capítulo III, sección 3.5.3, sustentados en el capítulo II, las secciones 2.2.7 y 2.2.8, los artefactos Scrum obtenidos en el capítulo IV; la iteración III medición y formalización de tablas de eficiencia de la sección 4.4.1; la tercera pila en la tabla N° 4.19, las tareas desarrolladas en la iteración III; el software cliente denominado ScpMid con la técnica de Middleware SQL de la figura N° 4.28, el software cliente con la técnica de Framework Hibernate , las mediciones de los indicadores de eficiencia en la tabla N° 4.20, 4.21 y 4.22. Medidas y formalizadas en las tablas de eficiencia. Se concluye que las interfaces del Middleware SQL, es un componente sofisticado, sistema cuyo controlador de conexión especializada de tal manera que no necesite tener una para cada motor relacional y así poder ofrecer mayor eficiencia y transparencia frente a otros componentes como frameworks en el acceso a las bases de datos distribuida.
- d.** Según el capítulo IV, la sección 2.3, discusión de la investigación podemos concluir que hemos mejorado el tiempo de acceso a los datos; menor líneas de código implementado en el acceso a datos; mejor y mejor volumen de datos cargados en la memoria heap; menor uso considerable de la CPU; mejor acoplamiento de las entidades y de la lógica a la capa de acceso a datos; evitar la carga innecesaria de datos; mejorar la transparencia en la incompatibilidad de tipos de dato; especializar, reusar y tener control de los planes de ejecución y finalmente tener la transparencia y adaptabilidad a entornos distribuidos y heterogéneos. Dichas premisas nos permite concluir y corroborar que si desarrollamos software usando el componente software denominado Middleware SQL entonces mejoramos la eficiencia de acceso a las bases de datos distribuida.

5.2 RECOMENDACIONES

- a.** Se debe investigar para que el Middleware SQL pueda generar clases pojos para facilitar y agilizar en desarrollo de nuevos proyectos.
- b.** Se debe investigar y desarrollar un adaptar para que el Middleware SQL pueda soportar de forma transparente las fuentes de datos basados en archivos.
- c.** Se debe investigar sobre la carga de los drivers jdbc en tiempo de ejecución para automatizar la carga manual y sin detener el funcionamiento Middleware SQL.

REFERENCIA BIBLIOGRÁFICA

1. Alonso, A. et al. (2005). *Introducción a la ingeniería del software*. Madrid, España: Delta Publicaciones. 542p.
2. Álvarez, G. (2012). *Integración de esquemas en bases de datos heterogéneas fuertemente acopladas*. Pachuca, México: Segundo encuentro nacional de computación.
3. Anguera, M. T. (1990). *Metodología observacional. Metodología de la investigación en ciencias del comportamiento*. España: Universidad de Murcia.
4. Antonio, J. (2001). *El gran libro del protocolo*. Madrid, España: Autor.
5. Aratoma, S. (2007). *Tesis de Grado y Metodología de Investigación en Organizaciones, Mercado y Sociedad*. Lima, Perú: DSG Vargas S.R.L.
6. Balek, D. (2002). *Software Connectors and Their Role in Component Deployment*. Recuperado de <http://www.omg.org/docs/formal/02-06-65.pdf>.
7. Bavaresco, A. (2006). *Proceso Metodológico en la Investigación. (Cómo hacer un diseño de investigación)*. Maracaibo: La Universidad de Zulia.
8. Beneken et al. (2003). *See several advantages of using components: they can run Object Management Group, CORBA Componentes*. USA: Object Management Group, CORBA Componentes.
9. Benjelloun, R. (2006). *Archimède: A Canadian solution for institutional repository*. (En: Library Hi Tech. Vol. 23. No. 4). Québec, Canada: Université Laval.
10. Bieberstein (2008). *A Practical Guide for the Service-Oriented Architect*. Boston, USA: Pearson education.
11. Bougettaya, A, Malik, Z, Rezgui, A, y Korff, L. (2006). *A Scalable Middleware for Web Databases*. Virginia Tech, USA: Journal of Database Management.
12. Cafassi, E. (1998). *Internet: Políticas y comunicaciones*. Buenos Aires, Argentina: Biblos.

13. Callejas, M., Peñalosa, D. I. y Alacón, A. C. (2011). *Evaluación y análisis de rendimiento de los frameworks de persistencia*. (Tesis de Maestría). Universidad de Manizales. Colombia.
14. Cardelli, L. (1985). *On Understanding Types, Data Abstraction, and Polymorphism*. Estados Unidos: AT&T Bell Laboratories, Murray Hill.
15. Chávez, N. (2007). *Introducción a la Investigación Educativa*. Maracaibo, Venezuela.
16. Cheng, S. (2010). *Microsoft Windows Communication Foundation 4.0 Cookbook for Developing SOA Applications*. Birmingham, Reino Unido: Packt Publishing.
17. Cobo, A. (s.f.). *Base de datos relacionales: Teoría y práctica* (1ª ed.). Madrid, España: Vision Libros.
18. Cockburn, A (2006) “*Use Case fundamentals*”. (1ª ed.) Addison-Wesley Professional.
19. Colección Esencial (2011). *Esencial Internet Explorer 9*. Cataluña, España: Editions ENI.
20. Councill y Heineman (2001). *Overview of COM+*. In *Component-Based Software Engineering*. Salzburg, Austria: Addison-Wesley Professional.
21. Craing, I. (2002). *The Interpretation of object-oriented programming languages*. (2ª ed.). Gran Bretaña: Springer-Verlag London.
22. Dabous F. (2004). *Using Software Architectures and Design Patterns for Developing Distributed Applications*. S65.Publication Date: 2004.
23. Date, C. J. (2001). *Introducción a los sistemas de bases de datos*. (7ª ed.). México: Editorial Pearson Educación de México, S. A. de C. V.
24. Debrauwer, L. y Van, F. (2005). *UML2: Iniciación, ejemplos y ejercicios corregidos*. Cataluña, España: Editions ENI.
25. Dikmans, L., Luttikhuizen, R. (2012). *SOA Made Simple*. Birmingham, Reino Unido: Packt Publishing Ltd.
26. Eisenberg et al. (2003). *Introduce algunas características de XML, cambios en las funciones, estandarización del objeto sequence y de las columnas autonumericas*. SQL: 2003 Has Been Published.
27. Elmasri, R; Navathe, S. B. (2000). *Sistemas de bases de datos*. (7ª ed.). México: Editorial Addison Wesley Longman de México, S. A. de C. V. México.

28. Erl, T. (2009). *SOA Design Patterns*. Boston, United States of America: Prentice Hall.
29. Fowler, M. (1999). *UML gota a gota*. Juárez, México: Addison Wesley Longman.
30. Gao, T. (2007). *The complete Reference to Professional SOA with Visual Studio 2005 (C# & VB 2005) .NET 3.0*. United States of America: Addison Wesley Longman.
31. Marcos, E., Vela, B. y Vara, J. (2005). *Diseño de base de datos objetos-relacionales con UML*. Madrid, España: DYKINSON.
32. Gosling, J.; Joy, H. y Steele, G. (1996). *The Java Language Specification*. Recuperado de <https://docs.oracle.com/javase/specs/jls/se7/jls7.pdf>.
33. Groth, D.; Skandier, T. (2005). *Guía del estudio de redes*. (4ª ed.). Sybex.
34. Gutiérrez, J. y Tena, J. (2003). *Protocolos criptográficos y seguridad en redes*. Cantabria, España: Gráficas Calima.
35. Hernandez S. y Bautista P. (2007). *Metodología de la Investigación Científica – Proyectos de investigación Científica*. Cantabria, España: Gráficas Calima.
36. Hernandez, M. (2000). *Metodología de la investigación – Tipos y Niveles de Investigación*. Maracaibo, Venezuela: Departamento de investigación.
37. IEFT (2006). *Middleware Disaing Architecture*. United States of America.
38. ISO/IEC (2005). *Base de datos distribudas. Information technology - Security techniques - Code of practice for information security managemen*. United States of America.
39. Kendall, K. E. y Kendall, J. E. (1997). *Análisis y diseño de sistemas*. (7ª ed.). México: Editorial Prentice Hall Iberoamericana.
40. Kent, B. et al. (2001), *Principios de manifiesto ágil*. Recuperado de <http://www.agilemanifesto.org/iso/es/manifesto.html>.
41. Kroenke, D. (2003). *Procesamiento de base de datos: fundamentos, diseño e implementación*. (8ª ed.). Juárez, México: Pearson.
42. Kruchten, Philippe (1995). *Diseño conceptual del modelo entidad relación*. Baltimore, Venezuela.
43. Larman, C. (1999). *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México, D.F., México: Prentice Hall.
44. Licesio J. y Rodríguez A. (20013). *Tema 4: Internet y Teleinformática. Consultado*. Madrid, Espana: Universidad Rey Juan Carlos.

45. López V. (2010), *Fragmentación de tablas en bases de datos distribuidas, implementación de una base de datos distribuida*. (Tesis de Grado). Universidad Técnica del Norte. Ecuador.
46. Lucas, A., Romera, P., Fraile, M., Argente, F. y Alfaro, A. (1993). *Diseño y Gestión de Sistema de Base de Datos*. Madrid, España: Editorial Paraninfo.
47. Luján, S. (2001). *Programación en internet: Clientes web*. Alicante. España: Club Universitario.
48. Luza (2010). *Modelado de UML*. Primera edición. Madrid, España: Editorial Paraninfo.
49. Münch J. (2010). *New Modelling Concepts for Today's Software Processes*. Paderborn: Proceedings.
50. Nevado, V. (s.f.). *Introducción a las bases de datos relacionales*. Madrid, España: Vision Libros.
51. Nolte, J. (1995). *Towards a Persistence Framework for High Performance Computing Systems*. Pennsylvania, USA.
52. Orfali, R; Harkey, D; Edwards, J. (1998). *Cliente/Servidor. Guía de supervivencia*. (2ª ed.). México: Editorial McGraw-Hill Interamericana Editores S. A. de C. V.
53. Osorio, F. (2008). *Base de datos relacionales: Teoría y práctica*. (1ª ed.). Madrid, España: Thomson.
54. Pagani, J. B. (2012). *Investigación en tecnologías de la información- Optimización de bases de datos*. Recuperado de <http://investigacionit.com.ar/es/optimizacion-de-bases-de-datos-mysql/>.
55. Palacio, J. (2008). *Flexibilidad con Scrum*. Recuperado de <http://www.safecreative.org/work/0710210187520>.
56. Pham, A. y Pham, P. (2010). *Scrum Rin Action: Agile Software Project Management and Development*.
57. Pons, O., Marín, N., Medina, J., Acid, S. y Vila, A (2005). *Introducción a las bases de datos: El modelo relacional*. Madrid, España: Thomson.
58. Pressman, R. S. (2005). *Ingeniería de Software. Un enfoque práctico*. (5ª ed). España: Editorial McGraw-Hill/Interamericana de España, S. A. U.
59. Pressman, R. S. (2011). *Software Engineering, a practitioner's approach* España: Editorial McGraw-Hill/Interamericana de España, S. A. U.

60. Ramírez, M. G. (2010). *Sistema de acceso distribuido remoto con entidades informáticas*. (Tesis de Grado). Universidad Autónoma Metropolitana Unidad Azcapotzalco. México.
61. Romero, L. (1997). *Publicar en Internet: guía práctica para la creación de documentos HTML*. Cantabria, España: Universidad de Cantabria.
62. Rosenberg, D. y Stephens, M. (2007). *Use Case Driven Object Modeling with UML: Theory and Practice*. Washington, Estados Unidos de América: Apress.
63. Rosenberg, D. y Scott, K. (2003). *Use Case Driven Object Modeling with UML*. Madrid: Pearson Educación.
64. Rumbaugh, J., Jacobson, I. y Booch, G. (2006). *El lenguaje unificado de modelado, manual de referencia*. Madrid: Pearson Educación.
65. Richardson, C. (2009). *ORM in Dynamic Languages*. In: *Communications of the ACM – A Direct Path to Dependable*. New York, USA: ACM. p. 48-55. ISSN: 0001-0782
66. Schwaber y Sutherland (2011). *Marco de trabajo Scrum*. España: Editorial McGraw-Hill Interamericana de España.
67. Silberschatz, A., Korth, H. F. y Sudarshan S. (1998). *Fundamentos de bases de datos*. España: Editorial McGraw-Hill Interamericana de España.
68. Sommerville I. (2005). *Ingeniería de software*. . (7ª ed.). España: PEARSON ADDISON WESLEY.
69. Stair, R. y Reynolds, G. (1999). *Principios de Sistemas de Información: Enfoque administrativo*. (4ª ed.). Madrid, España: Thomson.
70. Stallings, W. (2004), *Comunicaciones y Redes de Computadores*. (7º ed.). España: Editorial Prentice Hall Iberia.
71. Sun Microsystems (2007). *Core J2EE Patterns - Data Access Object*. USA.
72. Szyperski, C. (2002). *Component Software: Beyond Object-oriented Programming*. Boston, USA: Addison-Wesley Longman Publishing.
73. Tamayo, M. y Tamayo, A. (1997). *El Proceso de la Investigación Científica*. (3ª ed.). México: LIMUSA.
74. Tanenbaum, A. S. (1997). *Redes de computadoras*. (Tercera edición). México: Editorial Prentice Hall Hispanoamericana, S. A.
75. Throughput IBM. (2010). *Measurements of performance*. USA: Armonk .IBM Corporation.

76. Tuya, G. J. et al. (2007). *Técnicas cuantitativas para la gestión en la ingeniería el software: Técnicas y prácticas en las pruebas de software*. Madrid, España.
77. Velasco, M. (2008). *Especificación de una interfaz común SQL para la comunicación de base de datos heterogéneos en diferentes sistemas operativos*. (Tesis de Maestría). Universidad Autónoma de Querétaro. México.
78. Vértice (s. f.). *Cuidados enfermeros en la Unidad de Cuidados Intensivos (UCI)*. Madrid, España: Publicaciones Vértice.
79. Weitzenfeld, A. (s.f.). *Ingeniería de software orientada a objetos: con UML, Java e internet*. Madrid, España: Thomson.
80. Zyl, P. V. et al. (2006). *Comparing the performance of object databases and ORM tools*. In: *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. South African, South African.
81. Zorrilla (1993). *Investigación científica- Tipo de Investigación Aplicada*. México: Océano.

ANEXO A

MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES

VARIABLES	DIMENSIONES	INDICADORES	ITEMS	INSTRUMENTO
COMPONENTE SOFTWARE (CS)	Modelo Arquitectónico	Métodos de abstracción	¿Cuáles son los métodos utilizados para una abstracción objetiva en el acceso de base de datos distribuida?	Ficha de análisis documental
			¿Cómo los métodos de abstracción estandarizada ayudan un patrón arquitectónico que permite el acceso a base de datos distribuida?	Ficha de análisis documental
		Patrón de diseño	¿Cuáles son los patrones de diseño que permiten el acceso a base de datos distribuida?	Cuestionario
			¿Los patrones de diseño de las arquitecturas actuales solucionan el problema de acceso a base de datos distribuida?	Cuestionario
			¿Cree que el diseño un patrón basado en software middleware posibilita solucionar las deficiencias que se tienen en el acceso a base datos distribuida?	Cuestionario
			¿Es posible los patrones de diseño nativas ofrecen mayor performance que otros patrones como los framework en el acceso a base de datos distribuida?	Ficha de análisis documental
	Capa de Abstracción	Desacoplamiento	¿Cómo se desacopla las funcionalidades mediante la capa de abstracción el acceso a base de datos distribuidos?	Cuestionario
		Ocultamiento	¿Qué técnicas de ocultamiento en implementación utiliza se usa para configurar un componente software que accede a base de datos distribuida?	Ficha de análisis documental
		Especialización	¿De qué manera se especializa clases o subcomponentes para el componente software mantenga en el acceso a base de datos distribuida?	Ficha de análisis documental
			¿Cómo se separa la especialización de clases de configuración y servicios de acceso en sistemas informáticos que utilizan bases de datos distribuidos?	Ficha de análisis documental
	Empaquetamiento	¿Qué estándares se utilizan para empaquetamiento jerárquico de clases de configuración y servicios en los sistemas informáticos con acceso a base de datos distribuida?	Cuestionario	
	Interfaz	Comportamiento	¿Cuáles son los métodos utilizados para identificar los comportamientos de una interfaz intermedia para acceder de base de datos distribuida y heterogénea?	Cuestionario
		Propiedades	¿Cómo identificar las propiedades de una interfaz de un componente software que permite acceder a base de datos distribuida?	Cuestionario
			¿En qué medida facilita la identificación de propiedades de una interfaz la creación de contratos en el acceso a base de datos distribuida?	Cuestionario
		Escalable	¿Cuáles son los indicadores que permiten identificar que una interfaz sea altamente escalable en el acceso a fuentes de datos distribuido y heterogéneo?	Ficha de análisis documental
	Servicio	¿De qué manera utilizar las interfaces para crear servicios web que faciliten el acceso a las fuentes de datos distribuidas y heterogéneas?	Cuestionario	

		Multiplataforma	¿Es posible mantener un entorno multiplataforma haciendo uso de interfaces en el acceso a fuentes de datos distribuidos y heterogéneos? y ¿De qué manera transparente?	Cuestionario
BASE DE DATOS DISTRIBUDA (BDD)	Sistema Distribuido	Sistema Operativo	¿En qué medida los sistemas operativos actuales gestionan entornos distribuidos?	Ficha de análisis documental
			¿Qué sistemas operativos se utiliza para implementar sistemas distribuidos para acceso a base de datos heterogéneos?	Cuestionario
		Protocolo de comunicación	¿Cuáles son los protocolos que permiten crear un ambiente distribuido en el acceso a base de datos heterogéneos?	Ficha de análisis documental
			¿Qué función cumplen los protocolos de comunicación en sistemas distribuidos de acceso a datos?	Ficha de análisis documental
		Red de computadoras	¿De qué manera la red de computadoras ayuda a un sistema distribuido mantener su autónomo en el acceso a base de datos distribuido?	Cuestionario
		Aplicación cliente/servidor (Nodos)	¿Es posible contar con varias aplicaciones C/S como uno solo sistema homogéneo en el acceso a base de datos distribuida?	Cuestionario
	¿Cómo funciona las aplicaciones cliente/servidor en un sistema distribuido?		Ficha de análisis documental	
	Base de datos relacional	Uso de CPU en base de datos	¿Cómo funcionan los servidores de base de datos distribuidos y cuál es el uso real de la CPU?	Ficha de análisis documental
			¿Culés son los servidores de base de datos relacional más usados en los sistemas distribuidos?	Cuestionario
		Plan de Ejecución	¿Cómo se gestiona los planes de ejecución en los sistemas de gestor de base de datos?	Cuestionario
		Procesamiento de consulta SQL	¿Cómo gestiona un sistema de base de datos distribuida y heterogénea el procesamiento de consulta SQL?	Ficha de análisis documental
			¿Es más eficiente el procesamiento de consulta SQL en las SGBD o utilizando componentes software como frameworks?	Ficha de análisis documental
			¿Cuáles son los métodos de optimización, técnicas para el procesamiento de una consulta SQL en una base de datos heterogénea?	Cuestionario
		Tipo de dato	¿Es posible mantener homogeneidad de tipo de dato en diferentes sistemas gestores de base de datos?	Cuestionario
			¿En qué medida afecta la heterogeneidad de los tipos de datos de la base de datos en la construcción de sistemas informáticos?	Cuestionario
		Tiempo de acceso	¿Es posible mantener pool de sesiones de conexiones en una base de datos distribuida? y ¿Es posible reducir el tiempo de acceso con el uso de pool de conexiones?	Cuestionario
			¿En qué medida mejoran el performance la gestión de las sesiones de conexión el acceso a las bases de datos distribuida?	Cuestionario
		Memoria	¿De qué manera se incrementa el consumo de memoria heap en las bases de datos distribuida?	Ficha de análisis documental

Tabla N° A.1: Matriz de operacionalización de las variables

ANEXO B ANÁLISIS DOCUMENTAL

FECHA														
TIEMPO														
HORAS	10													
	APELLIDO Y NOMBRES :													

Tabla N° B.1: Patronos de diseño e implementación, modelos arquitectónicos en el de acceso a datos distribuidos.

ANEXO C

CUESTIONARIO

1. ¿Cuáles son los patrones de diseño que permiten el acceso a base de datos distribuida?

.....
.....

2. ¿Los patrones de diseño de las arquitecturas actuales solucionan el problema de acceso a base de datos distribuida?

.....
.....

3. ¿Cree que el diseño un patrón basado en software middleware posibilita solucionar las deficiencias que se tienen en el acceso a base datos distribuida?

.....
.....

4. ¿Cómo se desacopla las funcionalidades mediante la capa de abstracción el acceso a base de datos distribuidos?

.....
.....

5. ¿Qué estándares se utilizan para empaquetamiento jerárquico de clases de configuración y servicios en los sistemas informáticos con acceso a base de datos distribuida?

.....
.....

6. ¿Cuáles son los métodos utilizados para identificar los comportamientos de una interfaz intermedia para acceder de base de datos distribuida y heterogénea?

.....
.....

7. ¿Cómo identificar las propiedades de una interfaz de un componente software que permite acceder a base de datos distribuida?

.....
.....

8. ¿En qué medida facilita la identificación de propiedades de una interfaz la creación de contratos en el acceso a base de datos distribuida?

.....
.....

9. ¿De qué manera utilizar las interfaces para crear servicios web que faciliten el acceso a las fuentes de datos distribuidas y heterogéneas?

.....
.....
10. ¿Es posible mantener un entorno multiplataforma haciendo uso de interfaces en el acceso a fuentes de datos distribuidos y heterogéneos? y ¿De qué manera transparente?

.....
.....
11. ¿Qué sistemas operativos se utiliza para implementar sistemas distribuidos para acceso a base de datos heterogéneos?

.....
.....
12. ¿De qué manera la red de computadoras ayuda a un sistema distribuido mantener su autónomo en el acceso a base de datos distribuido?

.....
.....
13. ¿Es posible contar con varias aplicaciones C/S como uno solo sistema homogéneo en el acceso a base de datos distribuida?

.....
.....
14. ¿Culés son los servidores de base de datos relacional más usados en los sistemas distribuidos?

.....
.....
15. ¿Cómo se gestiona los planes de ejecución en los sistemas de gestor de base de datos?

.....
.....
16. ¿Cuáles son los métodos de optimización, técnicas para el procesamiento de una consulta SQL en una base de datos heterogénea?

.....
.....
17. ¿Es posible mantener homogeneidad de tipo de dato en diferentes sistemas gestores de base de datos?

18. ¿En qué medida afecta la heterogeneidad de los tipos de datos de la base de datos en la construcción de sistemas informáticos?

.....

.....

19. ¿Es posible mantener pool de sesiones de conexiones en una base de datos distribuida? y ¿cómo gestionarlos?

.....

.....

20. ¿En qué medida mejoran el performance la gestión de las sesiones de conexión el acceso a las bases de datos distribuida??

.....

.....

Tabla N° C.1: Cuestionario para recolectar información de los arquitectos de software.

ANEXO D

IMPLEMENTACIÓN Y PRUEBA DE LA INTERFAZ DEL MIDDLEWARE SQL

```

* @return=
*/
private WsSoapConexionResponse obtenerSoapConexionDataRequest (WsSoapConexionRequest peticion ) throws GenericWebServiceClientException{
    try {
        // Convertimos la peticion en un String XML
        String xmlRequest = XmlConverterUtil.objectToXmlString(WsSoapConexionRequest.class, peticion, Boolean.FALSE);

        // Se crea el mensaje que va ser enviado al servicio web
        Element requestElement = DomUtil.createElementMessageByNamespace(NAMESPACE_URI, OBTENER_SOAP_CONEXION_DATA_REQUEST, DATA, xmlRequest);

        // Objeto fuente que contiene el mensaje a enviarse
        DOMSource source = new DOMSource(requestElement);

        // Objeto con la respuesta del servicio web
        DOMResult result = new DOMResult();

        GnlParametroSimple URLWebService = parametroProvider.obtenerParametroSimplePorIdParametro(GeneralConstantes.URL_WEB_SERVICE_MID);
        String urlWebService = URLWebService.getValor();

        // Se envia el mensaje, consume el servicio y recibe la respuesta
        getWebServiceTemplate().sendSourceAndReceiveToResult(urlWebService, source, result);

        // Se recupera la respuesta del web service, que es un String XML
        String xmlResponse = DomUtil.getFirstChildValueFromResultElementByTagName(result, DATA);

        // Se obtiene y retorna el objeto
        // WsObtenerGruposDiagnosticosPorCodigosGruposRespuesta a partir del String XML
        WsSoapConexionResponse respuesta = (WsSoapConexionResponse) XmlConverterUtil.xmlStringToObject(WsSoapConexionResponse.class, xmlResponse);

        return respuesta;
    }
}

```

Figura N° D.1: Interfaz que permite acceder a las fuentes de base de datos distribuida.

```

package com.pe.ssit.acc.spcmid.wsdao.impl;
import java.util.ArrayList;
@Component("clienteDaoWs")
public class ClienteDaoWs extends GenericDaoWs implements ClienteDao {
    @Autowired
    private SoapConexionClientWebService soapConexionClientWebService;

    @Override
    public void guardar(ClienteImpl obj) throws GenericDaoException {
        try {
            /*inicio query registro de cliente*/
            soapConexionClientWebService.executeQuery(QueryDefinition.SPCMID_QUERY_1, obj.getNombres()+" "+obj.getTelefono()+
                "+" +obj.getDireccion()+" "+obj.getUsuarioCreacion()+" "+obj.getEstado());
            /*fin query registro de cliente*/
        } catch (Exception e) {
            ErrorPrinterUtil.printErrorInClass(log, getClass(), e, "guardar",
                "obj", obj, SecurityDataProviderUtil.getIdSesion());
            throw new GenericDaoException(getText(SpcMidConstantes.KEY_MENSAJE_ERROR_GENERICO), e);
        }
    }
}

```

Figura N° D.2: Capa de acceso a las fuentes de información por la aplicación denominada SPCMID.

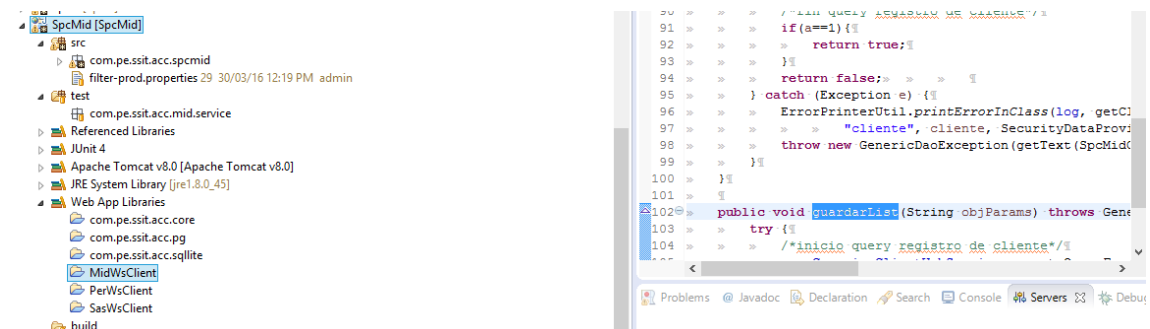


Figura N° D.3: Inyección de los servicios del Middleware SQL a la aplicación denominada SPCMID en tiempo de ejecución.

	id_cliente [PK] serial	nombres character vai	telefono character vai	direccion character vai	usuario_crea character vai	fecha_creaci timestamp w	estado character vai
1	309	ClienteSpch_1	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
2	310	ClienteSpch_2	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
3	311	ClienteSpch_3	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
4	312	ClienteSpch_4	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
5	313	ClienteSpch_5	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
6	314	ClienteSpch_6	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
7	315	ClienteSpch_7	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
8	316	ClienteSpch_8	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
9	317	ClienteSpch_9	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
10	318	ClienteSpch_10	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
11	319	ClienteSpch_11	Tef_2	Direccion real di	UADMIN	2016-09-08 04:	DES
*							

Figura N° D.4: Datos almacenados del cliente en una base de datos relacional.

Tiempo Busqueda : 91 millis - 00:00:00

🗑 Eliminar
🔍 Buscar

Código	Nombre	Teléfono	Estado
309	ClienteSpch_1	Tef_2	DES
310	ClienteSpch_2	Tef_2	DES
311	ClienteSpch_3	Tef_2	DES
312	ClienteSpch_4	Tef_2	DES
313	ClienteSpch_5	Tef_2	DES
314	ClienteSpch_6	Tef_2	DES
315	ClienteSpch_7	Tef_2	DES
316	ClienteSpch_8	Tef_2	DES
317	ClienteSpch_9	Tef_2	DES
318	ClienteSpch_10	Tef_2	DES

i

Gestionar

Los(as) (Cliente), ha sido Buscado con éxito.

Figura N° D.4: Prueba del servicio listar clientes con el SPCMID desde la base de datos de clientes.

ANEXO E

ESQUEMA DE BASE DE DATOS UTILIZADOS PARA LA PRUEBA DEL MIDDLEARE SQL

detalle_venta			
	Column Name	Data Type	Allow Nulls
🔑	id_cliente	int	<input type="checkbox"/>
🔑	id_producto	int	<input type="checkbox"/>
	cantidad	decimal(20, 0)	<input type="checkbox"/>
			<input type="checkbox"/>

producto			
	Column Name	Data Type	Allow Nulls
🔑	id_producto	int	<input type="checkbox"/>
	nombres	nvarchar(100)	<input type="checkbox"/>
	telefono	nvarchar(20)	<input type="checkbox"/>
	direccion	nvarchar(200)	<input type="checkbox"/>
	usuario_creacion	nvarchar(80)	<input type="checkbox"/>
	fecha_creacion	datetime	<input type="checkbox"/>
	estado	nvarchar(3)	<input type="checkbox"/>
			<input type="checkbox"/>

Figura E.1: Estructuras de producto y detalle venta en SQL SERVER 2012 para prueba del Middleware SQL.

The screenshot displays the pgAdmin III interface. On the left, the 'Object browser' shows a tree view of the database structure. The 'cliente' table is highlighted under the 'public' schema. The right pane shows the 'Properties' tab for the 'cliente' table, listing various attributes such as Name, OID, Owner, and Primary key. The bottom pane shows the SQL script for creating the table, including the table name, column definitions, and a primary key constraint.

```

-- Table: cliente
-- DROP TABLE cliente;
CREATE TABLE cliente
(
  id_cliente serial NOT NULL,
  nombres character varying(100) NOT NULL,
  telefono character varying(20) NOT NULL,
  direccion character varying(200) NOT NULL,
  usuario_creacion character varying(80) NOT NULL,
  fecha_creacion timestamp without time zone NOT NULL,
  estado character varying(3) NOT NULL,
  CONSTRAINT cliente_pkey PRIMARY KEY (id_cliente)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE cliente OWNER TO postgres;

```

Figura E.2: Estructura de clientes en POSTGRES para prueba del Middleware SQL

