



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE TECNOLOGÍAS DE LA INFORMACIÓN,
TELECOMUNICACIONES E INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS

TEMA:

SISTEMA DE TRANSFERENCIA DE DATOS DE ORIGEN MÚLTIPLE A ORACLE PARA LA EMPRESA SOLINFO.

Proyecto de Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

SUBLÍNEA DE INVESTIGACIÓN: Intercambio de Información

AUTOR: Adrián Darío Moyano González

TUTOR: Ing. Mg. Clay Aldas Flores

Ambato – Ecuador

Abril -2019

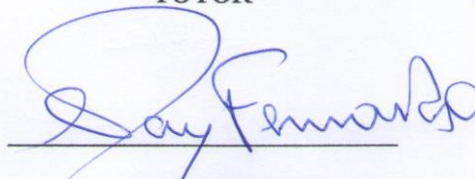
APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

“SISTEMA DE TRANSFERENCIA DE DATOS DE ORIGEN MÚLTIPLE A ORACLE PARA LA EMPRESA SOLINFO.”, de señor Adrián Darío Moyano González, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, abril de 2019

TUTOR

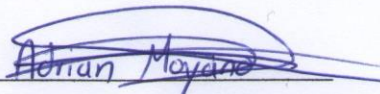


Ing. Clay Aldas. Mg.

AUTORÍA

El presente Proyecto de Investigación titulado: "SISTEMA DE TRANSFERENCIA DE DATOS DE ORIGEN MÚLTIPLE A ORACLE PARA LA EMPRESA SOLINFO.", es absolutamente original, autentico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprendan del mismo son de exclusiva responsabilidad del autor.

Ambato abril, 2019



Adrián Darío Moyano González


CC: 1804411971

DERECHOS DEL AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, abril de 2019



Adrián Darío Moyano González

CC: 1804411971

APROBACIÓN COMISIÓN CALIFICADORES

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Hernán Naranjo e Ing. Hernando Buenaño Mg., revisó y aprobó el Informe Final del trabajo de graduación titulado "SISTEMA DE TRANSFERENCIA DE DATOS DE ORIGEN MÚLTIPLE A ORACLE PARA LA EMPRESA SOLINFO.", presentado por el señor Adrián Darío Moyano González de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Elsa Pilar Urrutia. Mg.



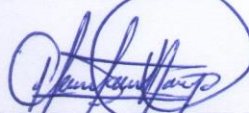
PRESIDENTE DEL TRIBUNAL

Ing. Hernando Buenaño. Mg.



DOCENTE CALIFICADOR

Ing. Hernán Naranjo



DOCENTE CALIFICADOR

DEDICATORIA

A mi madre por el apoyo incondicional que me ha brindado a lo largo de mi vida. Por el cuidado, cariño, educación y valores impartidos que han marcado mi vida. A mis hermanos por su apoyo, compañía y por estar presentes en todos los momentos importantes de mi vida, a mis amigos y compañeros por el apoyo brindado en el transcurso de tiempo que compartimos juntos, a todos con los que compartí momentos alegres y, a mis profesores que con su consejo y apoyo me motivaron a seguir adelante, todos ustedes me han impulsado a ser posible el cumplimiento de mis metas y objetivos.

Adrián Darío Moyano González

AGRADECIMIENTO

A Dios por guiar mis pasos, haber cuidado de mí y de mi familia, por su provisión y amor del cual he sido testigo durante mi vida. A mi familia en especial a mi madre Ana María González Ruiz por todo su esfuerzo, amor y confianza que depositó en mí, a todos mis amigos y a todos mis profesores que me guiaron durante mi educación en la primaria, secundaria y universidad, porque han marcado mi vida y me han llenado de muchos momentos felices, porque me han ayudado a lograr mis metas, a todos quienes han sido compañeros en el aprendizaje de la vida.

Muchas gracias a todos.

Adrián Darío Moyano González

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DEL AUTOR	iv
APROBACIÓN COMISIÓN CALIFICADORES	v
DEDICATORIA	vi
AGRADECIMIENTO.....	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS	xiii
RESUMEN EJECUTIVO	xv
ABSTRACT	xvi
INTRODUCCIÓN	xvii
CAPÍTULO I EL PROBLEMA	1
1.1 Tema de investigación	1
1.2 Planteamiento del problema.....	1
1.3 Delimitación.....	2
1.4 Justificación	2
1.5 Objetivos.....	3
1.5.1 Objetivo General	3
1.5.2 Objetivos Específicos	3
CAPÍTULO II MARCO TEÓRICO	4
2.1 Antecedentes investigativos.....	4
2.2 Fundamentación teórica.....	8
2.2.1 Historia de las bases de datos	8
2.2.2 ¿Qué es el copyleft?.....	9
2.2.3 ¿Qué significa «GPL»?.....	10
2.2.4 Software libre, ¿Significa que se está empleando la GPL?	10
2.2.5 ¿Qué es el software libre?.....	10
2.2.6 El sistema de gestión de bases de datos.....	10
2.2.7 Migración de datos	11
2.2.8 Casos en los que es necesaria la migración de datos.....	11

2.2.9 Metodología de migración de base de datos.....	12
2.2.10 Los fallos más comunes en la migración de datos	22
2.2.11 Transferencia de datos.....	23
2.2.12 Diseño centrado en el usuario.....	24
2.2.13 Usabilidad.....	24
2.2.14 Base de datos relacional	24
2.2.15 Oracle	25
2.2.16 PostgreSQL.....	28
2.2.17 MySQL.....	38
2.2.18 Microsoft SQL Server	44
2.2.19 NetBeans.....	46
2.2.20 Microsoft Visual Studio	46
2.2.21 Cadena de conexión.....	47
2.2.22 Base de datos no relacional	47
2.2.23 Microsoft Visual FoxPro	51
2.2.24 Los gestores de bases de datos más usados	53
2.2.25 Comprobante electrónico.....	55
2.2.26 Metodologías Ágiles.....	57
2.3 Propuesta de solución	64
CAPÍTULO III METODOLOGÍA	65
3.1 Modalidad de la aplicación.....	65
3.1.1 Investigación documental – bibliográfica	65
3.1.2 Investigación aplicada	65
3.2 Población y muestra.....	65
3.3 Recolección de la información	65
3.4 Procesamiento y análisis de datos.....	66
3.5 Desarrollo del proyecto.....	66
CAPÍTULO IV DESARROLLO DE LA PROPUESTA.....	67
4.1 Análisis de la situación actual.....	67
4.2 Requerimientos del sistema de transferencia de datos de origen múltiple a Oracle.....	70
4.2.1 Selección de las bases de datos y sus versiones	70
4.2.2 Selección del lenguaje de programación	73
4.2.3 Selección de las herramientas y recursos adecuados.....	74

4.3 Metodologías	75
4.3.1 Metodologías ágiles.....	75
4.4 Metodología aplicada.....	76
4.4.1 Fase: Planificación del proyecto.....	77
4.4.2 Fase: Diseño	79
4.4.3 Fase: Codificación.....	80
4.4.4 Fase: Pruebas	101
4.5 Análisis de resultados	101
4.6 Implementación del sistema de transferencia de datos.....	107
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES	112
5.1 Conclusiones.....	112
5.2 Recomendaciones	113
Material de referencia	114
1) Bibliografía.....	114
2) Anexos y Apéndices	117
Anexo A: Historias de usuarios	117
Anexo B: Manual de usuario	122
Anexo C: Manual de instalación.....	133
3) Glosario Técnico y Acrónimos.....	137

ÍNDICE DE FIGURAS

Fig. 2.1 Esquema de migración.....	22
Fig. 2.2 Ranking de las bases de datos.....	53
Fig. 2.3 Tendencia de las bases de datos.....	53
Fig. 2.4 Sector privado obligados a emitir comprobantes electrónicos	56
Fig. 2.5 Sector público obligados a emitir comprobantes electrónicos.....	57
Fig. 4.6 Fases de la metodología XP.....	76
Fig. 4.7 Mapeo de historias de usuario iteración 1 y 2	78
Fig. 4.8 Mapeo de historias de usuario iteración 3 y 4	78
Fig. 4.9 Contenido de la iteración 1	81
Fig. 4.10 Interfaz principal del sistema de transferencia	81
Fig. 4.11 Interfaz de nueva conexión.....	82
Fig. 4.12 Selección de tablas del origen.....	84
Fig. 4.13 Selección de columnas del origen.....	84
Fig. 4.14 Conexión destino	85
Fig. 4.15 Selección tablas del destino	85
Fig. 4.16 Vincular columnas origen con destino.....	86
Fig. 4.17 Interfaz de transferir.....	88
Fig. 4.18 Iteración 2	88
Fig. 4.19 Conexión a bases de datos no relacionales	91
Fig. 4.20 Comprobación de datos	92
Fig. 4.21 Iteración 3	95
Fig. 4.22 Clases del sistema	97
Fig. 4.23 Iteración 4	97
Fig. 4.24 Interfaz principal mensajes de ayuda.....	98

Fig. 4.25 Mensaje personalizado.....	99
Fig. 4.26 Comprobación de conexión	99
Fig. 4.27 Agregar columna.....	100
Fig. 4.28 Fin del proceso.....	100
Fig. 4.29 Mensaje personalizado.....	102
Fig. 4.30 Error por el conjunto de caracteres	103
Fig. 4.31 Gráfico de resultados de las pruebas realizadas.....	105
Fig. 4.32 Gráfico de frecuencia de los errores	107
Fig. 4.33 Generación del jar.....	108
Fig. 4.34 Generación del ejecutable.....	108
Fig. 4.35 Generación del instalador del sistema	109
Fig. 4.36 Selección de idioma.....	109
Fig. 4.37 Selección de la carpeta de destino	110
Fig. 4.38 Tareas adicionales.....	110
Fig. 4.39 Resumen de la instalación.....	111
Fig. 4.40 Instalación completada	111

ÍNDICE DE TABLAS

Tabla 2.1 Fases de la metodología Strategy 7 Corporation	13
Tabla 2.2 Características de la metodología migración Oracle-sqlServer-SSMA.....	14
Tabla 2.3 Características de la metodología de migración Álvaro Herrera	15
Tabla 2.4 Características de la metodología de migración de datos de Hudicka	19
Tabla 2.5 Tipos de datos en Oracle	26
Tabla 2.6 Tipos de datos en PostgreSQL	31
Tabla 2.7 Operadores en PostgreSQL	32
Tabla 2.8 Tipos numéricos en MySQL	38
Tabla 2.9 Formato binario para dígitos sobrantes	39
Tabla 2.10 Tipos de fecha en MySQL	39
Tabla 2.11 Precisión fraccional en MySQL	40
Tabla 2.12 Tipo de datos de cadena	40
Tabla 2.13 Categorías NoSQL	48
Tabla 2.14 Tipos de datos en Visual FoxPro	52
Tabla 2.15 Diferencias entre las metodologías ágiles y no ágiles.....	58
Tabla 4.16 Características del servidor de Solinfo.....	68
Tabla 4.17 Características de Oracle.....	69
Tabla 4.18 Características del computador del administrador	69
Tabla 4.19 Versiones consideradas de PostgreSQL.....	70
Tabla 4.20 Versiones consideradas de MySQL	71
Tabla 4.21 Versiones consideradas de Oracle.....	71
Tabla 4.22 Versiones consideradas de Microsoft SQL Server	72
Tabla 4.23 Comparación entre Netbeans y Visual Studio	73
Tabla 4.24 Características del computador del administrador de Solinfo	74

Tabla 4.25 Conectores de las bases de datos.....	75
Tabla 4.26 Problemas en Oracle	89
Tabla 4.27 Problemas en MySQL.....	89
Tabla 4.28 Problemas en SQLServer	90
Tabla 4.29 Problemas en PostgreSQL	90
Tabla 4.30 Pruebas realizadas	104
Tabla 4.31 Problemas representativos entre bases de datos.....	105
Tabla 4.32 Frecuencia de los errores.....	106

TEMA:

Sistema de transferencia de datos de origen múltiple a Oracle para la empresa Solinfo

RESUMEN EJECUTIVO

Las herramientas tecnológicas que se enfocan en la administración de la información están en constante evolución, con mecanismos que utiliza los datos como base para realizar análisis de mercados, tendencias, proyecciones entre otros servicios que permite mejorar diferentes áreas de una empresa.

En el Ecuador el SRI (Servicio de Rentas Internas) clasifica a los contribuyentes que están obligados a emitir comprobantes electrónicos desde el 2014 hasta el 2024, las instituciones que se encuentran dentro de la clasificación deben dejar de emitir comprobantes físicos y ajustarse a esta obligación.

Solinfo brinda soluciones informáticas como: comprobantes electrónicos, Slego ERP cloud, Servicios Oracle entre otros, con especialización en Finanzas, Producción y Comercio, el servicio de comprobantes electrónicos funciona a partir de los datos de la empresa solicitante del servicio, donde el proceso de obtención, transformación y carga de los datos se realiza de manera manual, para que se ajuste al esquema de la base de datos de la empresa, lo que provoca deficiencia en este proceso previo al servicio de comprobantes electrónicos, por esto, el automatizar este proceso previo al servicio es necesario para reducir tiempo y esfuerzo.

El presente proyecto propone la implementación de un Sistema de Transferencia de Datos de Origen Múltiple a Oracle para la empresa Solinfo, que se adapte a las necesidades independientemente si se trata de una base de datos relacional o no relacional y al sistema de la empresa, para minimizar el impacto que requiere los procesos de obtención, transformación y carga de los datos, con la finalidad de mejorar el servicio de comprobantes electrónicos.

Descriptor: bases de datos, información, conexiones, esquema de datos, transformación de datos, tipos de datos, conjunto de caracteres, equivalencias de datos.

THEME:

System of transfer of data of multiple origin to Oracle for the company Solinfo

ABSTRACT

The technological tools that focus on information management are constantly evolving, with mechanisms that use the data as a basis to perform market analysis, trends, projections among other services that allow improving different areas of a company.

In Ecuador, the SRI (Internal rents service) classifies taxpayers who are required to issue electronic receipts from 2014 to 2024, institutions that are within the classification must stop issuing physical receipts and comply with this obligation.

Solinfo provides IT solutions such as: electronic vouchers, Slego ERP cloud, Oracle Services among others, specializing in Finance, Production and Commerce, the electronic voucher service works from the data of the company requesting the service, where the process of obtaining , transformation and loading of the data is done manually, so that it fits the scheme of the company database, which causes deficiency in this process prior to the service of electronic receipts, for this the automation of this process prior to Service is necessary to reduce time and effort.

This project proposes the implementation of a Multiple Origin Data Transfer System to Oracle for the company Solinfo, which adapts to the needs independently if it is a relational or non-relational database and the company system, to minimize the impact that the processes of obtaining, transforming and loading the data requires, in order to improve the electronic voucher service.

Descriptors: databases, information, connections, data schema, data transformation, data types, character set, data equivalences.

INTRODUCCIÓN

Capítulo I, denominado EL PROBLEMA, describe el problema que será el motivo de la investigación, con su respectiva justificación y objetivos a cumplir a lo largo del proyecto dando solución al problema.

Capítulo II, denominado MARCO TEÓRICO, Manifiesta los antecedentes investigativos sobre temas similares que servirán como sustento para el desarrollo del proyecto y el planteamiento de una propuesta para dar solución al problema.

Capítulo III, denominado METODOLOGÍA, Presenta los diferentes tipos de modalidades de investigación a utilizarse, plantea los lineamientos en los cuales se basará la investigación, por último, se presenta un listado de las diferentes actividades necesarias para cumplir con los objetivos planteados.

Capítulo IV, denominado DESARROLLO DE LA PROPUESTA, describe el desarrollo de la propuesta llevando a cabo las actividades detalladas en el capítulo anterior.

Capítulo V, denominado CONCLUSIONES Y RECOMENDACIONES, presenta las conclusiones y se formulan las recomendaciones, respecto al problema planteado.

CAPÍTULO I

EL PROBLEMA

1.1 Tema de investigación

Sistema de transferencia de datos de origen múltiple a Oracle para la empresa Solinfo.

1.2 Planteamiento del problema

Actualmente la información constituye un activo de suma importancia para las empresas de todo el mundo, esta información es usualmente gestionada por uno de los diferentes medios tecnológicos que se encuentran disponibles en el mercado. [1]

La información es utilizada por las empresas para diferentes propósitos, el gestionar la para realizar un estudio de mercado, ejecutar algún plan de mejora, construir un big data que ayude a la toma de decisiones, mejorar procesos internos de la empresa, para automatizar procesos u obtener un mejor servicio al cliente entre otros, en cualquier caso, lo fundamental con que una empresa cuenta es una base de datos relacional o no relacional, archivos digitales, archivos físicos o algún medio tecnológico que le permita administrar la información.

En el Ecuador las empresas están conscientes de la importancia de la información que manejan por lo cual utilizan diversas formas para gestionar la desde bases de datos, archivos digitales o físicos, estos últimos son más accesibles tanto económicamente como a nivel de capacitación del personal, para este u otros casos surge la necesidad de cambiar de una plataforma a otra, ya sea por condiciones económicas, por dificultad de administración, por crecimiento de la empresa, por cambios de política de uso de la

herramienta en cuestión, porque los servicios de la empresa así lo requiere o para mejorar la administración de la información.

En Ambato existen empresas que brindan soluciones informáticas de diferente tipo, que inician su trabajo a partir de los datos de la empresa solicitante, Solinfo brinda el servicio de comprobantes electrónicos que requiere de los datos de la empresa solicitante del servicio, es en esta parte donde se presenta el problema de manipular distintas bases de datos ya sean relacionales o no relacionales para la transferencia de los datos necesarios para el servicio de comprobantes electrónicos, el transferir los datos entre distintas bases de datos, con la consideración que se cuenta con esquemas de bases de datos ya establecidas, que no pueden ser modificados porque son utilizadas para realizar procesos propios de la empresa que brinda el servicio y la solicitante, se realiza a través de los procesos manuales de selección, transformación y carga de datos. Motivo por el cual se requiere de un sistema que actúe como intermediario para realizar la transferencia de datos.

1.3 Delimitación

Área Académica: Software.

Línea de Investigación: Desarrollo de software.

Sublínea de Investigación: Intercambio de Información.

Delimitación Espacial: El proyecto de investigación se realizará en Solinfo, en la provincia Tungurahua, cantón Ambato, ubicado en la Avenida Rafael Darquea y Pedro Vascones.

Delimitación Temporal: La presente investigación se desarrollará durante el periodo febrero 2018 hasta febrero 2019.

1.4 Justificación

Se plantea la realización de la presente investigación considerando la importancia que tiene una transferencia de datos entre esquemas de bases de datos ya establecidas, esta transferencia se puede generar por diferentes escenarios tales como: la necesidad de

una empresa de trasladar sus datos temporalmente por cuestiones de mantenimiento, el utilizar servicios de terceros, por cambio de la base de datos, entre otros, el escenario en el que se enfoca la investigación es el utilizar servicios tecnológicos de terceros, específicamente el servicio de comprobantes electrónicos que actualmente brinda la empresa Solinfo, previo a dicho servicio Solinfo realiza los procesos de selección, transformación y carga de datos de manera manual o con una plantilla que sirve únicamente para empresas que utilizan la base de datos Oracle.

El desarrollo del sistema de transferencia de datos dentro de la empresa Solinfo, aportará a mejorar los procesos de selección, transformación y carga de datos de diverso origen con el fin de proporcionar una mejora al servicio de los comprobantes electrónicos, con el cual se pretende aumentar su competitividad realizando la transferencia de datos directamente entre las bases de datos para su tratamiento correspondiente, sin la necesidad que el usuario requiera de conocimientos profundos de las bases de datos en cuestión. Para garantizar el correcto desarrollo e implementación del sistema de transferencia de datos la empresa Solinfo prestará todas las facilidades del caso.

1.5 Objetivos

1.5.1 Objetivo General

- Implantar un sistema de transferencia de datos de origen múltiple a Oracle para la empresa Solinfo.

1.5.2 Objetivos Específicos

- Analizar los procesos internos realizados por la empresa y realizar el levantamiento de los requerimientos necesarios para el desarrollo del sistema.
- Seleccionar las bases de datos y sus versiones para las conexiones.
- Seleccionar las herramientas y recursos adecuados para la construcción del sistema propuesto.
- Implantar un sistema de transferencia de datos de origen múltiple a Oracle para la empresa Solinfo.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes investigativos

Para la presente investigación se han considerado proyectos similares relacionadas con la migración de base de datos, las definiciones, factores críticos, inconvenientes y problemas que intervienen en este proceso o investigaciones que abarcan desde la importancia que ha obtenido el uso de base de datos hasta su incidencia en los modelos de negocios actuales, los mismos que se describen a continuación:

Susana Beatriz Caraguay Martínez realizado en el 2012 con el título “METODOLOGÍA PARA MIGRACIÓN DE DATOS QUE PERMITA ASEGURAR Y CONSERVAR LA INTEGRIDAD Y CONSISTENCIA DE LA INFORMACIÓN ADMINISTRADA POR LA EMPRESA VSYSTEMS”, [2]. De acuerdo a la investigadora analiza los factores que provocan una migración de base de datos, los aspectos a considerar en el proceso y los pasos que debe tener todo proceso de migración de datos con la finalidad de proponer una metodología adecuada para la migración datos de acuerdo a las especificaciones propias del trabajo.

Marcia Meza Bermeo, en su trabajo de investigación realizado en el 2010 titulado “MOVIMIENTOS DE DATOS – RELACION CON OTRAS BASES.” [3], plantea y analiza los requerimientos tecnológicos, además el costo beneficio entre otros aspectos, este trabajo tiene un enfoque con las bases de datos Oracle y SqlServer considerando el interés que existe para que los sistemas contemplen cambios de plataformas desde su planificación.

Al final de la investigación se concluye, una herramienta que ofrezca armar parámetros de ejecución para extracción o carga de datos de forma ágil y que a la vez hace reporte a través de informes o estadísticas representa minimizar riesgos de transferencia de datos y una inversión mínima de tiempo-recurso.

Se reafirma la búsqueda de optimizar tiempo, recursos y minimizar el riesgo dentro de una transferencia de datos, considerando los aspectos propios de cada caso para ajustarse a la realidad de una empresa.

Dentro del trabajo elaborado por Gerardo Villagómez, Harold Álvarez, Danny Vivanco y Fabricio Echeverría, en el 2017 denominado “IMPLEMENTACIÓN DE LA MIGRACIÓN DE LA BASE DE DATOS DEL SISTEMA CANOPUS DE INFORMIX 9.4C A ORACLE 10G.” [4], los investigadores consideran varios aspectos para la migración de datos como que la base sea más liviana, segura, que soporte grandes cantidades de tamaño en byte y el mantenimiento de grupos de usuarios, algunos de los problemas que se presentaron en el trabajo fueron el rendimiento del CPU (Unidad Central de Proceso) y de los discos duros, además analizan los beneficios cualitativa y cuantitativamente, el trabajo entre otras cosas concluye que es de gran importancia el apoyo por parte de la alta dirección para afrontar problemas, que una correcta migración debe seguir una serie de pasos para asegurar su éxito y se determina que por los cambios de requerimientos es necesario una revisión completa de su arquitectura con el fin de garantizar que el cambio se adapte y no cause problemas en otros procesos.

El trabajo de Segundo Leopoldo Pauta Ayabaca realizado en el 2016 denominado “LAS BASES DE DATOS HETEROGÉNEAS Y SU INCIDENCIA EN LA GENERACIÓN DE REPORTE ACADÉMICOS DEL PERSONAL DOCENTE DE LA UNIVERSIDAD CATÓLICA DE CUENCA” [5], plantea el problema de como un sistema o arquitectura que funciona correctamente en el entorno de una empresa puede generar complicaciones para funcionar correctamente con un sistema externo, analizando los diferentes factores que se involucran.

Al finalizar la investigación se estableció como conclusión que una consideración de suma importancia en el análisis de las bases de datos heterogéneas en los distintos

sistemas de información de la Universidad Católica de Cuenca es la incompatibilidad semántica y sintáctica de sus bases de datos, lo que dificultó en gran medida la integración en un nuevo sistema.

La complejidad que existen en adaptar una base de datos a otra base de datos independientemente del tipo que sea es la incompatibilidad que existe entre la terminología que utilizan las distintas bases de datos.

Paul Mendoza del Carpio, realizado en el 2010 en su investigación “UN FRAMEWORK PARA LA MIGRACIÓN DE DATOS HACIA BASES DE DATOS ORIENTADAS A OBJETOS.” [6], considera la incompatibilidad que se genera entre modelos conceptual y relacional ocasionan gastos de tiempo y esfuerzo en un alto nivel, lo que demuestra la problemática que existe entre arquitecturas de bases de datos y entre sistemas, por lo cual este enfoque ayuda a fortalecer la propuesta de centrarse en los datos para su transferencia y no a todo un esquema.

El investigador al final llegó a la conclusión que el diseño del framework permite al desarrollador una más fácil y no exhaustiva implementación de aplicaciones para migración de datos hacia BDOO.

El diseñar una herramienta capaz de ajustarse a especificaciones propias, sin recurrir a métodos generales que requieren modificar la metodología interna de una empresa, es una solución que cada vez más es acogida por las empresas.

En el trabajo realizado por Cesar Alejandro Castillo Alvarado en el 2015 titulado “UNA GESTIÓN DE DATOS PARA MEJORAR Y DAR SOPORTE A LA TOMA DE DECISIONES EN LOS NEGOCIOS” [7], demuestra la importancia que tiene la correcta gestión de la información para las empresas y las diferentes formas en que la información se maneja y se obtiene para poder focalizarla a propósitos específicos.

Al término de la investigación se llegó a la conclusión que para llevar a cabo un proyecto de Gestión de Datos con una implementación exitosa en cualquier negocio u organización interesada en mejorar la integridad de su información para tomar decisiones basados en datos correctos y oportunos para el negocio.

El análisis realizado por Sergio Eduardo Mancilla Escobar, en el 2013 en su trabajo “USO DE BASES DE DATOS NOSQL DOCUMENTALES PARA CREAR SITIOS WEB DE ALTO RENDIMIENTO.” [8], sobre las bases de datos NoSQL aportan para el presente trabajo con información de manejo de dichas bases de datos en comparación con las bases sql, con esto se busca la manera de tener estas opciones contempladas para el sistema.

En la investigación se concluye que las bases de datos NoSQL documentales, como es el caso de MongoDB 10gen, no poseen seguridad o integridad en los datos, es por esta razón que no se deben utilizar en transacciones importantes.

Las bases de datos tienen diferente funcionalidad, esto acorde a la finalidad por la que fueron creadas e incluso bases de datos que tienen la misma finalidad cuentan con su propia forma de utilización por lo que tienen cierto grado de dificultad para llevar a cabo una integración entre ellas.

Sergio Fernández Díez y Jesús Madero de la Fuente en su trabajo realizado en el 2015 titulado “AUDITORÍA DE USABILIDAD Y ACCESIBILIDAD DE APLICACIONES WEB” [9]. Presentan los diversos referentes a considerar para realizar una auditoría sobre la usabilidad y la accesibilidad en aplicaciones web, estos referentes tienen diferentes aspectos a considerar dentro de una auditoría de usabilidad, estos aspectos son analizados y comparados con la finalidad de realizar una auditoría con aspectos generales agrupando los diferentes referentes basados en varios estándares internacionales, para establecer indicadores de medición de usabilidad y accesibilidad.

El investigador concluye que no existe un proceso específico para auditar la usabilidad y la accesibilidad de una aplicación web, aun cuando la norma UNE-EN ISO 19011 proporciona una guía general.

Los estándares internacionales no siempre se ajustan a la realidad de una empresa, por lo que se generalmente se cambia alguna pauta o se implementa parcialmente un estándar.

2.2 Fundamentación teórica

2.2.1 Historia de las bases de datos

El uso de las bases de datos se desarrolló a partir de las necesidades de almacenar grandes cantidades de información o datos. Sobre todo, desde la aparición de las primeras computadoras, el concepto de bases de datos ha estado siempre ligado a la informática. [10]

En 1884 Herman Hollerith creó la máquina automática de tarjetas perforadas, siendo nombrado así el primer ingeniero estadístico de la historia. En esta época, los censos se realizaban de forma manual. En la época de los sesenta, las computadoras bajaron los precios para que las compañías privadas las pudiesen adquirir; dando paso a que se popularizara el uso de los discos, debido a que a partir de este soporte se podía consultar la información directamente, sin tener que saber la ubicación exacta de los datos. [10]

La década de los setenta, Edgar Frank Codd, científico informático inglés conocido por sus aportaciones a la teoría de bases de datos relacionales, definió el modelo relacional “Un modelo relacional de datos para grandes bancos de datos compartidos”, en esta década también se creó Oracle Corporation a partir del trabajo de Edgar F. Codd. A principios de los 70, cuando empiezan a hacerse accesibles en línea distintas bases de datos, en su mayoría de organizaciones del gobierno de los Estados Unidos de América. Y, al mismo tiempo, también algunas instituciones sin ánimo de lucro — como por ejemplo la American Chemical Society— reciben ayudas de la National Science Foundation, que les permiten pasar de la información en papel al formato digital. [10]

A principios de los años ochenta comenzó el auge de la comercialización de los sistemas relacionales, y SQL comenzó a ser el estándar de la industria, ya que las bases de datos relacionales con su sistema de tablas (compuesta por filas y columnas) pudieron competir con las bases jerárquicas y de red, como consecuencia de que su nivel de programación era sencillo y su nivel de programación era relativamente bajo.

En la década de 1990 la investigación en bases de datos giró en torno a las bases de datos orientadas a objetos. Las cuales han tenido bastante éxito a la hora de gestionar datos complejos en los campos donde las bases de datos relacionales no han podido desarrollarse de forma eficiente. Así se desarrollaron herramientas como Excel y Access del paquete de Microsoft Office que marcan el inicio de las bases de datos orientadas a objetos. [10]

En la actualidad, las tres grandes compañías que dominan el mercado de las bases de datos son IBM (International Business Machines), Microsoft y Oracle. Por su parte, en el campo de internet, la compañía que genera gran cantidad de información es Google.

Aunque existe una gran variedad de software que permiten crear y manejar bases de datos con gran facilidad, como por ejemplo LINQ (Lenguaje Integrado de consultas), que es un proyecto de Microsoft que agrega consultas nativas semejantes a las de SQL a los lenguajes de la plataforma .NET. El objetivo de este proyecto es permitir que todo el código hecho en Visual Studio sea también orientado a objetos; ya que antes de LINQ la manipulación de datos externos tenía un concepto más estructurado que orientado a objetos; y es por eso que trata de facilitar y estandarizar el acceso a dichos objetos. [10]

2.2.2 ¿Qué es el copyleft?

El copyleft es un método general para liberar un programa u otro tipo de trabajo (en el sentido de libertad, no de gratuidad), que requiere que todas las versiones modificadas y extendidas sean también libres. La manera más simple de hacer que un programa sea software libre consiste en ponerlo en el dominio público, sin copyright. Esto permite compartir el programa y sus mejoras a quienes así lo deseen.

Sin embargo, también posibilita que otra gente sin interés cooperativo convierta el programa en software privativo. Pueden hacer cambios, muchos o pocos, y distribuir el resultado como un producto privativo. Quienes reciban el programa modificado en esas condiciones no podrán disfrutar de la libertad que el autor original les dio. El intermediario se la ha arrebatado. [11]

2.2.3 ¿Qué significa «GPL»?

«GPL» significa General Public License («Licencia Pública General»). La más difundida de tales licencias es la Licencia Pública General de GNU, o «GPL de GNU», para abreviar. Puede reducirse aún más, a «GPL», cuando se sobreentienda que nos estamos refiriendo a la «GPL de GNU». [11]

2.2.4 Software libre, ¿Significa que se está empleando la GPL?

No, en absoluto; hay muchas otras licencias de programas libres. Cualquier licencia que conceda al usuario ciertas libertades específicas es una licencia de software libre. [11]

2.2.5 ¿Qué es el software libre?

La definición de software libre estipula los criterios que se tienen que cumplir para que un programa sea considerado libre. De vez en cuando modificamos esta definición para clarificarla o para resolver problemas sobre cuestiones delicadas. [11]

Software libre es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

Es decir, el «software libre» es una cuestión de libertad, no de precio. Para entender el concepto, piense en «libre» como en «libre expresión», no como en «barra libre». [11]

2.2.6 El sistema de gestión de bases de datos

Un SGBD (Sistema de Gestión de Bases de Datos) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una BD (Base de Datos) nunca se accede o manipula directamente sino a través del SGBD. Se puede considerar al SGBD como el interfaz entre el usuario y la BD. [12]

2.2.7 Migración de datos

Se denomina migración a el proceso de pasar de un entorno operativo a otro entorno operativo, cambiarse a un entorno diferente implica tomar medidas para garantizar que las aplicaciones actuales continúen trabajando en el nuevo entorno y asegurarse de que las nuevas características puedan ser explotadas. [13]

Una migración de BD es el proceso que se realiza con la finalidad de trasladar los datos almacenados de un origen de datos a otro, para dicho proceso se debe tener clara la razón por la cual se desea migrar. Una migración puede llevarse a cabo para trasladar un esquema dentro de un mismo servidor, o de un servidor a otro, cambiar de fabricante o de plataforma de cómputo. [13]

Es un proceso que se realiza para mover o trasladar los datos almacenados de un origen de datos a otro, para lo cual es indispensable que antes de empezar cualquier proceso de esta naturaleza, se tenga clara y documentada la razón por la cual se está migrando, además de elaborarse la planeación detallada de las actividades contempladas. Dicha migración se requiere llevar a cabo cuando es necesario mover un esquema dentro del mismo servidor, o de un servidor a otro, así como para actualizar la versión del software, y hacer un cambio de manejador de bases de datos por el de otro fabricante o para cambiarlo a una plataforma de cómputo distinta. [14]

Para hacer una buena migración de datos tenemos que tener en cuenta diferentes aspectos para que durante el proceso de migración no tenga ninguna afectación o se dañe la base, para ello es necesario tener presente una planificación, una analítica de la base de datos, una aplicación, testing, la migración, evaluación, contador de registros, mapeador de tipos de datos, restricciones y triggers, codificación de caracteres. [15]

2.2.8 Casos en los que es necesaria la migración de datos

Existen casos particulares en las que es necesaria realizar una migración de datos dentro de una empresa, algunos casos en los que es necesario la migración son: [16]

- Se ha vuelto muy lenta la carga y consulta de datos.

- Se debe mantener compatibilidad hacia arriba con otros productos.
- El volumen de información que se maneja necesita mayor robustez de la base de datos.
- Se desea mantener actualizados los productos.
- Es necesario cambiar hacia un producto de base de datos sin licencia.
- El actual Sistema Operativo -S.O.- esta desactualizado.
- No es posible usar los programas nuevos.
- Se necesita bajar los gastos en tecnología.
- El Sistema Operativo actual no es confiable desde que se utiliza Internet.
- Se desea armar una red.
- Se han comprado nuevos equipos.

2.2.9 Metodología de migración de base de datos

Actualmente no existe un estándar específico para procesos de migración, pero existen algunas metodologías de migración que han sido diseñadas a medida para algunas empresas para ejecutar migración de datos. [2]

Algunas de las metodologías de migración existentes son:

- Strategy 7 Corporation: Migración y transformación de datos
- Metodología de migración de base de datos de oracle a sql-server mediante ssma de Docserver
- Migración de datos (Álvaro Alejandro Herrera Jaque)
- La Metodología Completa de Migración de Datos (Joseph R. Hudicka, Dulcian Inc)

Metodología de migración Strategy 7 Corporation

Strategy 7 Corporation es una empresa dedicada a ayudar a organizaciones a adaptarse a los cambios tecnológicos. Para cumplir con los objetivos de innovación y cambio tecnológico cuenta con una metodología de migración de datos compuesta por 6 fases.

Las fases que forman parte de ésta metodología son: Análisis, Mapeo, Diseño, Diseño detallado, Construcción, Test & Deploy.

Tabla 2.1 Fases de la metodología Strategy 7 Corporation

Fases	Actividades
Análisis	Reunir reglas de sistema y negocio Reunir métricas de datos del origen Almacenar información de otros datos (reglas de negocio y origen)
Mapeo	Análisis de mapeo de datos Análisis de integridad Diseño detallado, descripción vías de extracción Determinar medios de transformación Determinar reglas de validación Documentos de requerimientos
Diseño	Estrategia para ejecutar la migración Determinar configuración de software Estrategias para pruebas Estrategias de aseguramiento de calidad Estrategias para certificación
Diseño detallado	Crear proyecto de migración detallado Determinar estándares a usarse Desarrollar especificación de programas Desarrollar planes y benchmark Diseñar funciones, excepciones, manejo, logeos, etc.
Construcción	Comunicar estándares de programación Ejecutar motor de migración (herramienta de migración) Extracción, transformación, validación y carga Pruebas de unidad
Test y deploy	Ejecutar plan de test Realizar migraciones simuladas Mejorar reglas de negocio

Fuente: S. B. Caraguay [2]

Metodología de migración de base de datos de oracle a sql-server mediante ssma de Docserver

La metodología para migración desde Oracle a SQLServer usando SSMA es una guía para cambio de datos entre gestores Oracle y SQLServer usando herramientas propias de SQLserver y Microsoft.

Esta metodología reside en el sitio web DocServer y consta de 4 procesos principales los cuales son: Estrategia de definición con 4 subprocesos, Análisis con 4 subprocesos, Diseño con 2 subprocesos y Construcción con 1 subproceso. [2]

Tabla 2.2 Características de la metodología migración Oracle-sqlServer-SSMA

Proceso	Subproceso	Descripción
Estrategia de definición	Inventario	Consolidación de inventario sobre el origen y la base de datos destino, modelos de datos, volúmenes de datos disponibles. Uso de Analyzer Migration que es parte de microsoft SSMA toolset
	Estrategia de mapeo	Definir una estrategia de mapeo, de la fuente al origen. Mediante la herramienta SSMA se puede automatizar el mapeo, pero se debe plantear estrategias para aquellos tipos de datos que no necesitan un manejo distinto
	Selección de herramienta	Propone el uso de SSMA para la migración
	Opciones de puesta en marcha	Opciones para la puesta en marcha en producción, mediante migración incremental o total
Análisis	Análisis de modelos de datos	Analizar la cardinalidad y opcionalidad de los modelos de datos del origen y el destino.
	Análisis de datos	Derivación de los dominios, rangos y valores válidos para los dominios
	Evaluación calidad de datos	Determinación de registros incorrectos en relación a términos de la lógica del negocio
	Extracción, auditoría y criterios de evaluación	Obtener criterios de extracción basados en el análisis de resultados y opciones de implementación. Los criterios de auditoría y validación son basados en los requerimientos para evaluar la correcta y completa migración de datos
Diseño	Mapeo	Reglas de mapeo y especificación basada en la estructura de origen y destino
	Diseño de scripts de migración	Secuencia de ejecución de scripts basado en dependencia (técnicas y reglas de negocio)
Construcción	Development script	Ejecución de la migración y pruebas de scripts de migración de base de datos, scripts de extracción y scripts de transformación. Auditar y validar la ejecución de scripts

Fuente: S. B. Caraguay [2]

Metodología de Migración de datos (Álvaro Alejandro Herrera)

Álvaro Herrera en un curso de migración de datos en web denominado Migración de datos, propone una metodología para la migración de datos basada en 9 fases. [2]

Tabla 2.3 Características de la metodología de migración Álvaro Herrera

Fase	Actividad
Levantamiento de información	<ol style="list-style-type: none"> 1. Conocer el nuevo modelo de datos. 2. Establecer los conceptos que se deben migrar al nuevo modelo y clasificarlos. 3. Identificar fuentes de los datos. 4. Identificar responsables de los datos. 5. Definición de las tablas críticas a ser migradas. 6. Definición de campos críticos. 7. Identificar ciclos de actualización de la información. 8. Establecer volumen de los datos. 9. Clasificación de datos: grande/mediano/pequeño, o se puede clasificar por rango de datos: 1-30, 31-100, 100-500, etc. 10. Establecer datos históricos que serán traspasados al nuevo sistema: Tratar que se queden en los actuales sistemas. 11. Establecer los flujos de información en el antiguo sistema. 12. Establecer los flujos de información en el nuevo sistema. 13. Establecer la estrategia de migración de datos: <ol style="list-style-type: none"> 14. Desarrollo de una carga inicial. 15. Desarrollo de una carga incremental. 16. Identificar migraciones de datos que van a ser constantes en el tiempo. 17. Establecer las brechas de datos que existirá entre las fuentes de datos v/s el nuevo modelo. Se deberá definir si el gap (brecha) es cubierto por una digitación manual de los datos o un poblamiento automático. 18. Establecer esfuerzo en la corrección de datos antes y después de la limpieza. 19. El entregable de ésta etapa es una planilla con información de objetos.

Tabla 2.3 (Cont)

	<p>20. Establecer equipos de trabajo de la próxima etapa si es necesario.</p>
Análisis de calidad	<p>1. Establecer una declaración de buenas intenciones con los participantes del proyecto, explicando la metodología a utilizar.</p> <p>2. Explicar las fuentes de información detectadas en la etapa anterior.</p> <p>3. Establecer calidad de información en:</p> <p>4. Integridad</p> <p>5. Consistencia</p> <p>6. Nulidad</p> <p>7. Establecer Herramienta de migración.</p> <p>8. Establecer forma de migración de cada dato y clasificar en: Manual o Automática.</p> <p>9. Establecer criterios de calidad de los datos a migrar.</p> <p>10. Establecer métricas de calidad.</p> <p>11. Establecer ambientes de desarrollo y pruebas para la construcción de ETL.</p> <p>12. Coordinar con producción impacto en el desempeño en los actuales productos, al momento de realizar las cargas de datos. 13. Definir y construir herramientas de limpieza automática. 14. Informar calidad de los datos a migrar, completando la siguiente información de la planilla establecida en la etapa de levantamiento de información.</p> <p>15. Forma de Migración: Manual/Automática 16. Calidad del dato: Buena/Regular/Mala</p>
Transformación / Mapeo	<p>1. Establecer reglas de transformaciones de los datos</p> <p>2. Establecer secuencia de las migraciones.</p> <p>3. Definir qué cosas se cambiarán en el origen y que cosas son candidatas a transformaciones.</p> <p>4. Establecer las relaciones entre las fuentes de datos de Origen y de Destino.</p>

Tabla 2.3 (Cont)

	<ol style="list-style-type: none"> 5. Establecer esfuerzos de construcción de Extracción de datos, Transformación de datos e Inyección de datos. 6. Establecer si existen agrupaciones de datos desde las fuentes de origen a la de destino o viceversa. <ol style="list-style-type: none"> 7. Establecer los controles adicionales 8. Definir la administración de errores. 9. Poblar los diccionarios. 10. Tener presente que las reglas pueden sufrir cambios a medida que el proyecto avanza. 11. Diseño del plan de pruebas
Limpieza de datos	<ol style="list-style-type: none"> 1. Establecer criterios de limpieza de los datos. 2. Establecer el esfuerzo real del trabajo de limpieza. 3. Establecer planificación de limpieza de información. 4. Realizar limpieza de datos. 5. Definir herramientas que ayuden a la revisión de la información.
Construcción ETL	<p>Esta etapa, se realiza la construcción de los extractores, transformadores e inyectores de datos.</p> <ol style="list-style-type: none"> 1. Establecer el modelo técnico que se utilizará en los Extractores, Transformadores e Inyectores de datos. 2. Construir herramientas que ayuden a la revisión de información. 3. Construir Extractores de datos. 4. Construir Transformadores de datos. 5. Construir Inyectores de datos al sistema nuevo. 6. Realizar pruebas técnicas unitarias de: Extractores, Transformación, Inyección.
Pruebas	<ol style="list-style-type: none"> 1. Establecer un plan de pruebas para revisar los datos. 2. Por cada regla, se debe definir un escenario. 3. Definir estadísticas de cuadratura. Si existen diferencias establecer claramente los por qué. 4. Establecer criterios de calidad de pruebas.

Tabla 2.3 (Cont)

	<p>5. Realizar pruebas funcionales unitarias de: Extractores, Transformación, Inyección.</p> <p>6. Realizar pruebas de volumen de: Extractores, Transformación, Inyección.</p> <p>7. Revisar datos cargados.</p> <p>8. Análisis y clasificación de errores.</p> <p>9. Revisar los controles adicionales que se deben administrar por restricciones propias del software.</p>
	<p>1. Realizar simulación de cargas reales.</p> <p>2. Establecer calendario de cargas reales.</p>
Cargas Reales	1. Realizar cargas reales.
Revisión	1. Revisar los datos en el sistema nuevo

Fuente: S. B. Caraguay [2]

La Metodología Completa de Migración de Datos

Dulcian Inc, es una empresa que provee servicios de consultoría, desarrollo de aplicaciones web, datawarehousing y migración de datos específicamente en entorno Oracle.

Joseph Judicka miembro de la empresa Dulcian Inc. propuso una metodología que consta de 7 fases, en base a su experiencia profesional en proyectos de migración.

La migración de datos, entendida como un proceso de transferencia de datos de un sistema a otro, a menudo más reciente, parece simple. No obstante, aún requiere mucho tiempo para prepararse, así como algo de tiempo para finalizar con la validación de migración adecuada.

Es por eso que la migración completa de datos puede tomar notablemente más tiempo que el tiempo necesario para extraer, transformar y cargar datos en nuevas bases de datos. Cada instrucción metodológica dada apunta a considerar los siguientes pasos en la migración de datos. [2]

Tabla 2.4 Características de la metodología de migración de datos de Hudicka

Fases	Actividades
Pre estrategia	<p>Se determina el enfoque del proyecto general</p> <p>Permite determinar el alcance de la migración. Identificar el número de requerimientos de migración de sistemas heredados</p> <p>Conteo de la estructura de datos</p>
Estrategia	<p>Examinar el plan de migración</p> <p>Evaluar la cantidad de información existente para determinar plazos</p> <p>Elaborar el documento de estrategia de migración, que contiene los objetivos de la migración</p> <p>Calidad de datos de alto nivel</p>
Pre análisis	<p>Se realiza un detalle de los objetivos</p> <p>Se determina responsables de ejecutar las tareas</p> <p>Evaluación de datos de sistemas heredados para determinar si serán trasladados</p>
Análisis	<p>Se obtiene un check list de los datos que serán migrados, esta información tiene tres fuentes principales: Análisis de datos heredados descritos en la fase de pre-análisis</p> <p>Reportes de auditorías. Examinar el uso de los campos</p>
Pre Diseño	<p>Se completa el modelo de datos y se genera el esquema físico de la base de datos</p>
Diseño	<p>Se realiza el mapeo de datos sobre la estructura física creada</p> <p>Generalmente deben participar en esta fase un analista de negocio, un analista de sistemas que genera las rutinas</p>
Pre-Test / test/ implementación	<p>Permite identificar errores físicos y lógicos</p> <p>Se realiza correcciones de errores físicos</p> <p>Se realiza correcciones de mapeo</p> <p>Ejecutar cargas y evaluar si el número de registros que se carga es el correcto</p> <p>Evaluar si los campos se cargaron correctamente</p> <p>Evaluar si el formato cargado fue el correcto</p>
Revisión	<p>Todas las modificaciones al modelo de datos, cambios de reglas y scripts modificados son combinados. Se revisa coherencia entre modelo lógico y física.</p>

Fuente: S. B. Caraguay [2]

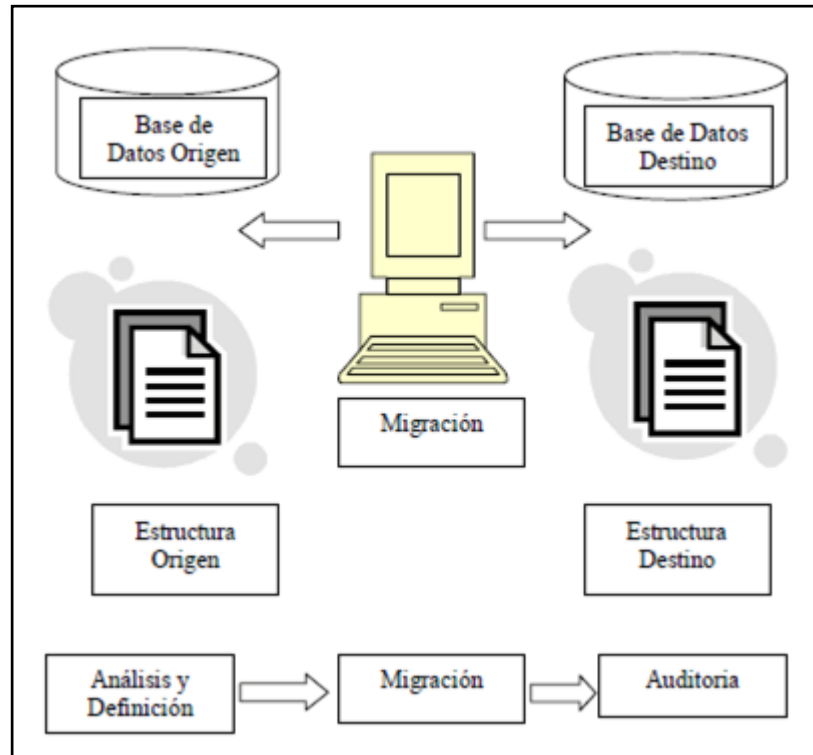
La utilización de las fases permite realizar un trabajo ordenado, conocer las estructuras de las bases de datos involucradas, familiarizarse con los datos y su distribución, considerar aquellos datos que requerirán atención especial, aquellos que en adelante no serán indispensables, entre otros. A continuación, se listan las fases en común a tomar en cuenta antes, durante y posterior a la migración de las metodologías mencionadas anteriormente. [13]

- **Planificación.** Lo más importante al migrar una base de datos es llevar a cabo un proceso de planificación y análisis del trabajo. Debe considerarse el análisis de la estructura de la base de datos origen y la planificación, es decir que la planificación de bases de datos es la verificación, a ver si resulta conveniente o no un traslado parcial o total, dejar la aplicación donde se encuentra o reemplazar un elemento antiguo por uno nuevo o saber si es necesario mover una arquitectura. En esta parte se revisará todas las tareas previas a la migración de las bases, los requisitos previos que tenemos que tener para la migración, las tareas antes de la migración y sobre todo las tareas posteriores de la migración. Lo importante de la planificación es que esto sea aplicado cuando la base de datos destino este en producción. Para hacer esta planeación tenemos que tener en cuenta el procedimiento y sobretodo unas rutinas en las cuales encontramos requisitos previos si cumple con todos los requisitos previos para tareas y sub tareas de migración, en especial la obtención de tareas que sirven a varias plataformas.
- **Contador de registros.** Se debe hacer un conteo de los registros existentes, los registros migrados exitosamente y conservar los registros que no se logren migrar, esto dará a los usuarios la certeza de que su información es coherente y no ha sufrido pérdidas. En este punto puede tomarse en cuenta la opinión de los administradores de base de datos para tomar acciones correctivas acerca de los datos que no hayan podido migrarse, si la migración se hace manual mediante consultas de inserción se deberá o se recomendará hacer mediante la inicialización de un contador para cada registro insertado con éxito y otro para los que no serán insertados, y con la suma de estos dos será igual a los registros originales.

- **Mapeador de tipos de datos.** Algunas plataformas no soportan algunos tipos de datos, así que es necesario planificar el mapeo de los campos en la nueva base de datos, son las plataformas que al ser diferentes algunas no son soportadas o no nos soportaran algunos datos debido a esto se deberá hacer un mapeo de los campos en la nueva base de datos
- **Restricciones y triggers.** Antes de iniciar la migración de la base de datos, es recomendable deshabilitar los Triggers y/o restricciones que nos puedan generar error al momento que el SMBD (Sistema Manejador de Base de Datos) ejecute el proceso de escritura de los datos.
- **Codificación de caracteres.** Cuando el copiado se realiza de forma automática, es necesario identificar la codificación de caracteres que la base de datos destino espera, pues así evitaremos el reemplazo automático de caracteres o en su caso, pérdida de los mismos, es decir encontramos una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados, algunas requerirán alguna pequeña manipulación de los datos, pero cuando es necesaria aplicaran las siguientes transformaciones
- **Pruebas.** Toda buena implementación requiere una fase de pruebas, podría pensarse en presentación de los reportes utilizados por los usuarios de la aplicación para tener la certeza de que no existirá ausencia de información al momento de poner en marcha la migración. Este paso da la oportunidad de observar y corregir las excepciones no controladas, es donde se miden los resultados y se analizan para determinar los ajustes necesarios y así dar el valor si es viable o no viable la migración de datos.
- **Implementación.** Como su nombre lo dice, es la fase que implica poner en marcha la migración física de los datos. Según los requerimientos debe identificarse el momento apropiado y el tiempo estimado que puede tardarse en llevarse a cabo. Existen sistemas que pueden ser pausados por minutos, horas e incluso días enteros, por el contrario, existen otros tantos que no pueden detener su marcha pues representaría pérdidas económicas para la empresa que lo requiera.
- **Monitoreo.** Una vez finalizada la fase de implementación es recomendable realizar observaciones muy de cerca para asegurarse de que la información que

se ha migrado es consistente y fiable, en caso de no ser así puede requerirse corrección de errores. [13]

Fig. 2.1 Esquema de migración



Fuente: J. Fernández [13]

2.2.10 Los fallos más comunes en la migración de datos

Prevenir estos riesgos y mitigar sus consecuencias es la clave para superar cualquier obstáculo que impida el correcto desenlace de la migración de datos. A la hora de proceder, el desarrollador deberá tener especial cuidado con:

- Especificaciones incompletas o inexactas: para hacer frente a este desafío, es necesario que los proyectos cuenten con especificaciones de mapeo columna a columna, deben incluir las reglas del negocio, de limpieza de datos, las validaciones, etc. Lo más importante es recordar que para realizar esta tarea se necesita tiempo y que también hace falta contar con los perfiles idóneos. Sin un nivel detallado de estas especificaciones es muy probable que el proyecto no llegue a realizarse de la forma esperada o a término.

- Problemas de calidad de datos que afectan a los plazos establecidos para la conclusión del proyecto: para evitar que un problema de este tipo tenga sus repercusiones en fases más avanzadas del proceso de migración de datos, es muy recomendable realizar siempre las actividades del perfilado de datos en etapas tempranas del proyecto.
- Dificultades en la gestión del proyecto de ejecución del proyecto de migración de datos: este riesgo puede provenir de distintos orígenes y en cada caso habrá que atender a sus causas para paliar sus efectos negativos y superarlo. [17]

2.2.11 Transferencia de datos

La transferencia de datos es una parte dentro de la migración de base datos, siendo uno de los aspectos más importante al igual que la estructura de la base de datos, el tratamiento que se da a los datos del origen de la base de datos es la obtención, transformación y carga hacia la base de datos destino, donde se debe contemplar los tipos de datos y sus equivalentes entre las diferentes bases de datos para su transformación de esta manera evitar problemas durante el proceso y garantizar que todos los datos sean transferidos.

Ahora bien, una migración de bases de datos realiza una serie de etapas para completar la migración, pero nos enfocaremos en tres etapas que involucra a la transferencia de datos las cuales son: [17]

- Extracción: es el proceso mediante el que se toman los datos del sistema o los sistemas de origen.
- Transformación: en esta etapa al dato le son aplicadas las reglas del negocio para convertirlo, adecuarlo y prepararlo para el siguiente paso.
- Carga: una vez que los datos se han transformado y convertido se cargan al nuevo sistema, de forma directa o por etapas, dependiendo del tipo de conexión disponible entre ambos.

Estas etapas son las que realizan en esencia la transferencia de datos por lo tanto son las que sirven para comparar los procesos identificados.

2.2.12 Diseño centrado en el usuario

El concepto de DCU (Diseño Centrado en el Usuario o UCD del inglés User-Centered Design) ha ganado popularidad en los últimos años, como proceso encaminado al diseño de productos (generalmente software) que respondan a las necesidades reales de sus usuarios finales. [18]

2.2.13 Usabilidad

La usabilidad es la cualidad de los productos que se pretende obtener mediante el DCU; dicho de otro modo, el objetivo principal del DCU es obtener productos más usables. En ese sentido, consideraremos también que la ingeniería de usabilidad, que tiene el mismo propósito, es equivalente en la práctica al Diseño Centrado en el Usuario. [18]

La usabilidad usualmente es disminuida por las actuales medidas de seguridad que se adoptan en los diferentes sistemas ya sean de escritorio o web, ya que estas medidas de seguridad pueden llegar a tener un gran nivel de complejidad únicamente para el acceso a un sistema, es aquí donde el debate de usabilidad contra seguridad toma relevancia y requiere de un planteamiento técnico para inclinarse por alguno de estos o establecer un equilibrio adecuado dependiendo de las características del sistema y de las observaciones de los interesados.

2.2.14 Base de datos relacional

Es una base de datos que se ajusta al modelo relacional. El modelo relacional tiene los siguientes aspectos principales: [19]

- Estructuras
Los objetos bien definidos almacenan o acceden a los datos de una base de datos.
- Operaciones
Las acciones claramente definidas permiten a las aplicaciones manipular los datos y las estructuras de una base de datos.
- Reglas de integridad

Las reglas de integridad rigen las operaciones sobre los datos y las estructuras de una base de datos.

2.2.15 Oracle

Oracle es una empresa que se inicia en los años 70 por un estudio realizado por George Koch sobre los sistemas gestores de datos siendo una de las bases de datos más completas en soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma, basándose en una arquitectura cliente servidor para la gestión de bases de datos relacionales desarrollada por Oracle Corporación, con una interfaz comprensible, siendo capaz de administrar bases de datos, crear tablas, vistas y otros objetos de bases de datos, importar, exportar y visualizar datos de tablas, ejecutar scripts de SQL y generar informes. [20]

Oracle tiene una arquitectura que se compone de dos partes, la base de datos que consta de las estructuras físicas y las estructuras lógicas y la otra parte de la instancia de la base de datos que se compone de estructuras de memoria y de procesos asociados con la instancia. Una Base de Datos Oracle es un conjunto de datos almacenado y accesible según el formato de tablas relacionales. Una tabla relacional tiene un nombre y unas columnas, su definición. Los datos están almacenados en las filas. Las tablas pueden estar relacionadas con otras, está almacenada físicamente en ficheros, y la correspondencia entre los ficheros y las tablas es posible gracias a las estructuras internas de la Base de Datos, que permiten que diferentes tipos de datos estén almacenados físicamente separados. Esta división lógica se hace gracias a los espacios de tablas, tablespaces. [20]

La base de datos de Oracle consta de una capa física que consiste en un conjunto de archivos que se encuentran en el disco y una capa lógica son las estructuras que mapean los datos hacia los componentes físicos. [20]

Tipos de datos en Oracle

El Tipo de datos determina el carácter de la información que contiene el campo. No es lo mismo almacenar valores numéricos con los que posteriormente podremos realizar operaciones matemáticas, que guardar textos como los nombres de los alumnos. [21]

Entre los más utilizados están:

- **Número:** Es el tipo numérico más flexible. Permite determinar su tamaño y su precisión en cuanto al número de valores decimales que puede representar.
- **Texto [VARCHAR]:** Tipo de campo destinado a cadenas de caracteres cuya longitud no es fija. En la propiedad Longitud se determina su longitud máxima.
- **Sí/No:** Este tipo de datos, conocido como “booleano”, se utiliza para representar datos Verdadero/Falso o Sí/No o 1/0.
- **Fecha, Hora y Fecha/Hora:** Estos tipos de datos se utilizan para almacenar información referida a horas, fechas o su combinación. El formato de presentación lo podemos seleccionar en la propiedad Formato. [21]

Propiedades de los tipos de datos

Cada tipo de dato tiene una serie de opciones que permiten ajustar el comportamiento del campo a las necesidades de cada situación. [21]

Los tipos de datos se agrupan en los siguientes conjuntos:

Tabla 2.5 Tipos de datos en Oracle

Alfanuméricos	Numéricos	Fecha	Binarios	Otros
Char	Number	date	raw	rowid
varchar (2)	Float	timestamp	long raw	urowid
Varchar		timestamp with time zone	blob	
Nchar		interval	clob	
nvarchar (2)			nlob	
Long			bfile	

Fuente: M. de Educación [21]

Versiones de Oracle

Versión 10.2

La Base de datos Oracle 10g Release 2 Standard Edition esta optimizada para su despliegue en medianas industrias. Esta soportada en un único servidor soportando hasta un máximo de 4 CPUs (Unidad Central de Procesamiento), o en un ambiente de servidores en clúster, con un máximo de 4 CPUs en el clúster. La base de datos Oracle 10g Release 2 Standard Edition está disponible en todos los sistemas operativos soportados por Oracle entre los cuales se incluye Windows, Linux y Unix.

La base de datos Oracle 10g Standard Edition proporciona una rápida instalación tanto en un único servidor como en un ambiente de clúster. La base de datos va a estar preconfigurada lista para ser usada en producción, completa con espacio automatizado, administración de almacenamiento y de memoria, back up y recuperación automatizada y administrador de estadísticas automatizado. [34]

Versión 11.1

Oracle Database 11g fue desarrollada con la máxima capacidad de recursos de ingeniería para un solo producto. Oracle Database 11g es el resultado de un proceso de desarrollo que incorpora las opiniones de grupos de usuarios y partners para brindar un mejor desempeño, seguridad y administración automatizada. Como base de datos de próxima generación, el producto presenta las siguientes características: [35]

- SecureFiles
- Real Application Testing
- Advanced Compression
- Total Recall

Versión 11.2

Las nuevas capacidades disponibles con estas soluciones en Oracle Database 11g versión 2 o 11.2.

- Reducción de los tiempos de inactividad no programados

Los errores de hardware, que provocan que falle el servidor, son en esencia impredecibles y, cuando ocurren, se traducen en tiempos de inactividad de las aplicaciones. Del mismo modo, una gama de funciones de disponibilidad de datos.

- Disponibilidad del servidor

La disponibilidad del servidor está relacionada con garantizar el acceso ininterrumpido a los servicios de las bases de datos a pesar de las posibles fallas inesperadas de una o más máquinas que alojan el servidor de la base de datos, lo cual puede producirse debido a fallas de hardware o de software.

- Oracle Real Application Clusters

Oracle Real Application Clusters (RAC) es la primera tecnología de agrupación en clústeres de bases de datos en permitir que dos o más equipos (nodos) de un grupo de servidores accedan al mismo tiempo a una sola base de datos compartida. [35]

2.2.16 PostgreSQL

A fines de 1996, el nombre del servidor de base de datos de Postgres95 se cambió por POSTGRESQL, para honra tanto el nombre de Berkeley como sus capacidades SQL. POSTGRESQL es un software de código abierto. El término "software de código abierto" a menudo confunde a las personas. Con el software comercial, una empresa contrata programadores, desarrolla un producto y lo vende a los usuarios. Con la comunicación por internet, sin embargo, existen nuevas posibilidades. El software de código abierto no tiene compañía. En su lugar, programadores capaces con interés y algo de tiempo libre se reúnen a través de Internet e intercambian ideas. Alguien escribe un programa y lo coloca en un lugar al que todos pueden acceder. Otros programadores se unen y hacen cambios. Cuando el programa es lo suficientemente funcional, los desarrolladores anuncian la disponibilidad del programa a otros usuarios de Internet. Los usuarios encuentran errores y características faltantes y los reportan a los desarrolladores, quienes, a su vez, mejoran el programa. [29]

Suena como un ciclo inviable, pero en realidad tiene varias ventajas:

- No se requiere una estructura de la empresa, por lo que no hay gastos generales ni restricciones económicas.
- El desarrollo del programa no se limita a un personal de programación contratado, sino que aprovecha las capacidades y la experiencia de un gran grupo de programadores de Internet.
- Se facilita la retroalimentación de los usuarios, lo que permite la prueba del programa por parte de un gran número de usuarios en un corto período de tiempo. [29]

Tipos de datos en PostgreSQL

Es tentador pensar que las bases de datos serían más fáciles de usar si solo existiera un tipo de datos, un tipo que podría contener cualquier tipo de información, como números, cadenas de caracteres o fechas. [29]

- Resultados consistentes

Las columnas de un tipo uniforme producen resultados consistentes. La visualización, clasificación, agregados y uniones entregan resultados consistentes. No surge ningún conflicto sobre cómo se comparan o muestran los diferentes tipos. Por ejemplo, al seleccionar de una columna INTEGER siempre se obtienen valores INTEGER. [29]

- Validación de datos

Las columnas de un tipo uniforme solo aceptan datos debidamente formados; Los datos inválidos son rechazados. Por ejemplo, una columna de tipo INTEGER rechazará un valor DATE. [29]

- Almacenamiento compacto

Las columnas de un tipo uniforme se almacenan de forma más compacta.

- Actuación

Las columnas de un tipo uniforme se procesan más rápidamente.

Por estas razones, cada columna en una base de datos relacional puede contener solo un tipo de datos. No puede mezclar tipos de datos dentro de una columna. Esta limitación puede causar algunas dificultades. Por ejemplo, nuestra tabla de amigos incluye una columna de edad del tipo INTEGER. Sólo los números enteros se pueden colocar en esa columna. [29]

- Cadena de caracteres

Los tipos de cadena de caracteres son los tipos de datos más utilizados. Pueden contener cualquier secuencia de letras, dígitos, puntuación y otros caracteres válidos. Las cadenas de caracteres típicas son nombres, descripciones y direcciones de correo. Puede almacenar cualquier valor en una cadena de caracteres. [29]

- Número

Los tipos de números permiten el almacenamiento de números. Los tipos de números son INTEGER, INT2, INT8, OID, NUMERIC (), FLOAT y FLOAT4. INTEGER, INT2 e INT8 almacenan números enteros de varios rangos. Los rangos más grandes requieren más almacenamiento. [29]

- Lógico

El único tipo lógico es BOOLEAN. Un campo BOOLEAN puede almacenar solo verdadero o falso, y por supuesto NULL. [29]

- Operadores de soporte

Los operadores son similares a las funciones, son divididos por los tipos de datos y su funcionalidad y la diferencia que existe entre operadores iguales pero que están orientados a ciertas especificaciones propias de los tipos de datos.

Una función de operador implementa un determinado símbolo de operador. El servidor de bases de datos proporciona funciones especiales invocadas por SQL, denominadas funciones de operador, que implementan operadores. Una función de operador procesa entre uno y tres argumentos y devuelve un valor. [29]

Tabla 2.6 Tipos de datos en PostgreSQL

Categoría	Tipo	Descripción
cadena de caracteres	text	variable longitud de almacenamiento
	varchar(length)	variable longitud de almacenamiento con longitud máxima.
	char(length)	longitud de almacenamiento fija
Numero	integer	entero, rango ± 2 billones
	int2	entero, rango ± 32 mil
	int8	entero, rango $\pm 4 \cdot 10^{18}$
	oid	identificador de objeto
	numeric	número, definido por el usuario
	float	número flotante, precisión de 15 dígitos.
	float4	número flotante, precisión de 6 dígitos.
Temporal	date	fecha
	time	hora
	timestamp	fecha y hora
	interval	intervalo de tiempo
Logical	boolean	boolean, verdadero or falso
Geometric	point	punto
	lseg	segmento de línea
	path	Lista de puntos
	box	rectángulo
	circle	circulo
	polygon	polígono
Network	inet	dirección ip con máscara de red opcional
	cidr	dirección de red ip
	macaddr	ethernet mac address

Fuente: B. Momjian [29]

Los operadores más comunes son.

Tabla 2.7 Operadores en PostgreSQL

Tipo	Función	Ejemplo	Valor retornado
Carácter		Col1 col2	Añadir col2 al final de col1
String	~	Col ~ pattern	Boolean, col coincide con el patrón de expresión regular
	!~	Col !~ pattern	Boolean, col no coincide con el patrón de expresión regular
Number	!	!col	Factorial
	+	Col1 + col2	Adición
	-	Col1- col2	Sustracción
	*	Col1 * col2	Multiplicación
	/	Col1 / col2	División
temporal	+	Col1 + col2	Suma de valores temporales.
	-	Col1 – col2	Resta de valores temporales.

Fuente: B. Momjian [29]

- Objetos grandes (BLOBs)

PostgreSQL no puede almacenar valores de más de varios miles de bytes con los tipos de datos analizados hasta ahora, ni los datos binarios se pueden ingresar fácilmente entre comillas simples. En su lugar, los objetos grandes, también llamados objetos grandes binarios o BLOBS, se utilizan para almacenar valores muy grandes y datos binarios. [29]

Características NoSQL implementadas en PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional que, gracias a su extensibilidad, ha permitido la incorporación de nuevas funcionalidades encaminadas a agilizar y flexibilizar la manipulación de los datos, entre las que destacan el almacenamiento efímero y los tipos de datos HSTORE y JSON. [30]

Los tipos de datos HSTORE y JSON proveen opciones de gestión de datos sin esquema con la ventaja de cumplir con las propiedades ACID, permitiéndole al gestor dar soporte a aplicaciones que requieran flexibilidad en el modelo de datos (EnterpriseDB,

2014). HSTORE fue añadido en la versión 8.2 como un módulo que implementa un tipo de datos, con el mismo nombre, para almacenar pares de clave-valor dentro de un único valor de PostgreSQL; útil en situaciones como filas con muchos atributos raramente examinados o semiestructurados. [30]

Con estas características PostgreSQL permite:

- El almacenamiento de datos de forma ágil.
- El almacenamiento de datos libres de esquema.
- La lectura de datos relacionales de una tabla y su retorno como JSON y viceversa, haciendo uso de sus operadores y funciones.
- La integración fácil de sentencias convencionales SQL con tipos de datos JSON y HSTORE y basados en el mismo planificador de consultas, optimizador y tecnologías de indexado. [30]

Con estas características PostgreSQL mejora considerablemente su rendimiento, sin embargo, requiere de conocimientos en conceptos y práctica de no relacionales para evitar inconvenientes en el manejo de los datos al poder permitir el almacenamiento libre de esquemas.

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central. [31]

Versiones de PostgreSQL

Versión 9.2

Esta versión se ha centrado en gran medida en las mejoras de rendimiento, aunque no faltan las nuevas características de SQL. El trabajo también continúa en el área de soporte de replicación. Las mejoras principales incluyen:

- Permitir que las consultas recuperen datos solo de los índices, evitando el acceso al almacenamiento dinámico (exploraciones de solo índice)

- Permitir que el planificador genere planes personalizados para valores de parámetros específicos incluso cuando se usan declaraciones preparadas
- Mejore la capacidad del planificador para usar bucles anidados con exploraciones de índice interno
- Permitir que los esclavos de replicación de transmisión reenvíen datos a otros esclavos (replicación en cascada)
- Permitir que pg_basebackup realice copias de seguridad básicas desde servidores en espera
- Agregue una herramienta pg_receivexlog para archivar los cambios del archivo WAL a medida que se escriben
- Agregue el método de acceso al índice SP-GiST (espacio-particionado GiST)
- Añadir soporte para tipos de datos de rango
- Añadir un tipo de datos JSON
- Agregue una opción de barrera de seguridad para las vistas
- Permitir que las cadenas de conexión libpq tengan el formato de un URI

Esta versión ya no cuenta con soporte disponible por ende no revive actualizaciones ni correcciones de errores.

Versión 9.3

Las mejoras principales en PostgreSQL 9.3 incluyen:

- Añadir vistas materializadas
- Hacer vistas simples auto-actualizables
- Agregue muchas funciones para el tipo de datos JSON, incluidos operadores y funciones para extraer elementos de los valores JSON
- Implemente la opción LATERAL estándar de SQL para las subconsultas y las llamadas de función FROM
- Permitir que los contenedores de datos externos admitan escrituras (inserciones / actualizaciones / eliminaciones) en tablas externas
- Agregue un contenedor de datos foráneos de Postgres para permitir el acceso a otros servidores de Postgres

- Añadir soporte para desencadenadores de eventos
- Agregue capacidad opcional a las páginas de datos de suma de comprobación e informe daños
- Evitar que las actualizaciones de filas de campos no clave bloqueen las comprobaciones de claves externas
- Reducir considerablemente los requisitos de memoria compartida de System V

PostgreSQL 9.3 actualmente no cuenta con soporte disponible.

Versión 9.4

Las mejoras principales en PostgreSQL 9.4 incluyen:

- Agregue json, un tipo de datos más capaz y eficiente para almacenar datos JSON
- Agregue un nuevo comando de SQL ALTER SYSTEM para cambiar las entradas del archivo de configuración postgresql.conf
- Reducir la fuerza de bloqueo para algunos comandos ALTER TABLE
- Permitir que las vistas materializadas se actualicen sin bloquear las lecturas simultáneas
- Agregue soporte para la decodificación lógica de datos WAL, para permitir que los cambios en la base de datos se distribuyan en un formato personalizable
- Permitir que los procesos de trabajo en segundo plano se registren, inicien y terminen dinámicamente

Versión 9.5

Las mejoras más importantes en PostgreSQL 9.5 incluyen:

- Permitir que los INSERT que generen conflictos de restricción se conviertan en ACTUALIZAR o se ignoren
- Agregue las características de análisis de GROUP BY GROUPING SETS, CUBE y ROLLUP
- Agregar control de seguridad a nivel de fila

- Cree mecanismos para rastrear el progreso de la replicación, incluidos los métodos para identificar el origen de los cambios individuales durante la replicación lógica.
- Agregar índices de rango de bloque (BRIN)
- Mejoras sustanciales de rendimiento para la clasificación
- Mejoras sustanciales de rendimiento para máquinas con múltiples CPU

Versión 9.6

Las mejoras principales en PostgreSQL 9.6 incluyen:

- Evite escanear páginas innecesariamente durante las operaciones de congelación al vacío
- La replicación sincrónica ahora permite múltiples servidores en espera para una mayor confiabilidad
- La búsqueda de texto completo ahora puede buscar frases (varias palabras adyacentes)
- postgres_fdw ahora soporta remota una, tipo, ACTUALIZACIÓN s, y BORRAR s
- Mejoras sustanciales en el rendimiento, especialmente en el área de escalabilidad en servidores de múltiples CPU – zócalos

Versión 10

Las mejoras principales en PostgreSQL 10 incluyen:

- Replicación lógica mediante publicación / suscripción.
- Particionamiento de tabla declarativa.
- Consultas mejoradas de paralelismo.
- Mejoras significativas en el rendimiento general.
- Autenticación de contraseña más fuerte basada en SCRAM-SHA-256.
- Monitorización y control mejorados.

El software MySQL ofrece un servidor de bases de datos SQL muy rápido, multiproceso, multiusuario y robusto. MySQL Server está diseñado para sistemas de

producción de carga pesada de misión crítica, así como para la integración en software de implementación masiva. Oracle es una marca registrada de Oracle Corporation y / o sus filiales. MySQL es una marca comercial de Oracle Corporation y / o sus filiales, y no será utilizada por el Cliente sin la autorización expresa por escrito de Oracle. Otros nombres pueden ser marcas registradas de sus respectivos dueños. [32]

El software MySQL tiene doble licencia. Los usuarios pueden elegir utilizar el software MySQL como un producto de código abierto según los términos de la Licencia Pública General de GNU o pueden comprar una licencia comercial estándar de Oracle. [32]

Las características principales de MySQL

Internos y Portabilidad

- Escrito en C y C ++.
- Probado con una amplia gama de compiladores diferentes.
- Funciona en muchas plataformas diferentes.
- Para portabilidad, usa CMake en MySQL 5.5 y superior. Las series anteriores utilizan GNU Automake, Autoconf y Libtool.
- Probado con Purify (un detector de fugas de memoria comercial), así como con Valgrind, una herramienta GPL. [32]

Seguridad

- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite la verificación basada en host.
- Seguridad de contraseña mediante el cifrado de todo el tráfico de contraseña cuando se conecta a un servidor. [32]

Escalabilidad y Límites

- Soporte para grandes bases de datos. Utilizamos MySQL Server con bases de datos que contienen 50 millones de registros. También conocemos a usuarios que usan MySQL Server con 200,000 tablas y aproximadamente 5,000,000,000 de filas.

- Soporte para hasta 64 índices por tabla. Cada índice puede constar de 1 a 16 columnas o partes de columnas. El ancho máximo del índice para las InnoDBtablas es de 767 bytes o 3072 bytes. [32]

Conectividad

- Los clientes pueden conectarse a MySQL Server utilizando varios protocolos:
- Los programas cliente de MySQL se pueden escribir en muchos idiomas. Una biblioteca de cliente escrita en C está disponible para clientes escritos en C o C ++, o para cualquier idioma que proporcione enlaces de C.
- Las API para C, C ++, Eiffel, Java, Perl, PHP, Python, Ruby y Tcl están disponibles, lo que permite que los clientes MySQL se escriban en muchos idiomas. [32]

2.2.17 MySQL

Admite varios tipos de datos SQL en varias categorías: tipos numéricos, tipos de fecha y hora, tipos de cadena, tipos espaciales y el JSON tipo de datos.

Tipo Numérico

Tabla 2.8 Tipos numéricos en MySQL

Tipo de datos	Almacenamiento requerido
Tinyint	1 bytes
Smallint	2 bytes
Mediumint	3 bytes
int, integer	4 bytes
Bigint	8 bytes
Float	4 bytes si $0 \leq p \leq 24$, 8 bytes si $25 \leq p \leq 53$
Float	4 bytes
double (precision), real	8 bytes
decimal (m, d), numeric (m, d)	varia
bit (m)	aproximadamente $(m+7) / 8$ bytes

Fuente: MySQL Data Types [32]

Los valores para DECIMAL (y NUMERIC) columnas se representan mediante un formato binario que contiene nueve dígitos decimales (base 10) en cuatro bytes. El almacenamiento para las partes enteras y fraccionarias de cada valor se determina por separado. [32]

Tabla 2.9 Formato binario para dígitos sobrantes

Dígitos sobrantes	Número de bytes
0	0
1	1
2	1
3	2
4	2
5	3
6	3

Fuente: MySQL Data Types [32]

Tipo de fecha y hora

Para TIME, DATETIME y TIMESTAMP columnas, el almacenamiento requerido para las tablas creadas antes de MySQL 5.6.4 difiere de las tablas creadas desde 5.6.4 en adelante. Esto se debe a un cambio en 5.6.4 que permite que estos tipos tengan una parte fraccionaria, que requiere de 0 a 3 bytes. [32]

Tabla 2.10 Tipos de fecha en MySQL

Tipo de datos	Almacenamiento antes de MySQL 5.6.4	Almacenamiento a partir de MySQL 5.6.4
Year	1 byte	1 byte
Date	3 bytes	3 bytes
Time	3 bytes	3 bytes + segundos de almacenamiento fraccional
Datetime	8 bytes	5 bytes + segundos de almacenamiento fraccional
Timestamp	4 bytes	4 bytes + segundos de almacenamiento fraccional

Fuente: MySQL Data Types [32]

A partir de MySQL 5.6.4, el almacenamiento para YEAR y DATE permanece sin cambios. Sin embargo, TIME, DATETIME, y TIMESTAMP están representados de manera diferente. DATETIME se empaqueta de manera más eficiente, requiriendo 5 en lugar de 8 bytes para la parte no fraccional, y las tres partes tienen una parte fraccionaria que requiere de 0 a 3 bytes, dependiendo de la precisión fraccional de segundos de los valores almacenados. [32]

Tabla 2.11 Precisión fraccional en MySQL

Precisión de segundos fraccional	Almacenamiento requerido
0	0 bytes
1, 2	1 byte
3, 4	2 bytes
5, 6	3 bytes

Fuente: MySQL Data Types [32]

Tipo cadena

M representa la longitud de columna declarada en caracteres para tipos de cadenas no binarios y bytes para tipos de cadenas binarias. L representa la longitud real en bytes de un valor de cadena dado. [32]

Tabla 2.12 Tipo de datos de cadena

Tipo de datos	Almacenamiento requerido
char (m)	M^x bytes, 255
binary	M bytes, $0 \leq M \leq 255$
varchar(m), varbinary(m)	L+ 1 bytes si los valores de columna requieren 0 - 255 bytes, L+ 2 bytes si los valores requieren más de 255
tinyblob, tinytext	L+ 1 bytes, donde $L < 2^8$
blob text	L+ 2 bytes, donde $L < 2^{16}$
mediumblob, mediumtext	L+ 3 bytes, donde $L < 2^{24}$
longblob, longtext	L+ 4 bytes, donde $L < 2^{32}$

Fuente: MySQL Data Types [32]

Los tipos de cadena de longitud variable se almacenan utilizando un prefijo de longitud más datos. El prefijo de longitud requiere de uno a cuatro bytes dependiendo del tipo de datos, y el valor del prefijo es L (la longitud del byte de la cadena). Por ejemplo, el

almacenamiento para un MEDIUMTEXT valor requiere L bytes para almacenar el valor más tres bytes para almacenar la longitud del valor. [32]

VARCHAR, VARBINARY y los tipos BLOB y TEXT son tipos de longitud variable. Para cada uno, los requisitos de almacenamiento dependen de estos factores:

- La longitud real del valor de la columna
- La longitud máxima posible de la columna.
- El conjunto de caracteres utilizado para la columna, porque algunos conjuntos de caracteres contienen caracteres de varios bytes.

JSON

En general, el requisito de almacenamiento para una JSON columna es aproximadamente el mismo que para una columna LONGBLOB o LONGTEXT; es decir, el espacio consumido por un documento JSON es aproximadamente el mismo que sería para la representación de cadena del documento almacenada en una columna de uno de estos tipos. Sin embargo, existe una sobrecarga impuesta por la codificación binaria, incluidos los metadatos y los diccionarios necesarios para la búsqueda, de los valores individuales almacenados en el documento JSON. Por ejemplo, una cadena almacenada en un documento JSON requiere de 4 a 10 bytes de almacenamiento adicional, según la longitud de la cadena y el tamaño del objeto o la matriz en la que se almacena. [32]

El desarrollo de MySQL ahora usa un modelo de hito. El cambio a este modelo proporciona lanzamientos de hitos más frecuentes, en los que cada hito introduce un pequeño subconjunto de funciones probadas a fondo. Después de los lanzamientos para un hito, el desarrollo continúa con otro pequeño número de lanzamientos que se centra en el siguiente conjunto de características. [32]

Versiones de MySQL

Versión 5.5

Características añadidas en MySQL 5.5

- Autenticación conectable. La autenticación MySQL admite dos nuevas capacidades, la autenticación conectable y los usuarios proxy. Con la

autenticación conectable, el servidor puede usar complementos para autenticar las conexiones entrantes del cliente, y los clientes pueden cargar un complemento de autenticación que interactúe correctamente con el complemento del servidor correspondiente. Esta capacidad permite a los clientes conectarse al servidor MySQL con las credenciales adecuadas para los métodos de autenticación que no sean la autenticación MySQL incorporada basada en las contraseñas nativas de MySQL almacenadas en el servidor MySQL.mysql.user mesa.

- Unicode. Soporte para caracteres Unicode suplementarios; es decir, caracteres fuera del plano multilingüe básico (BMP). Estos nuevos conjuntos de caracteres Unicode incluyen caracteres complementarios: utf16, utf32, y utf8mb4.
- Compatibilidad con IPv6. MySQL Server puede aceptar conexiones TCP / IP de clientes que se conectan a través de IPv6. [33]

Características eliminadas en MySQL 5.5

- La languagevariable del sistema (uso lc_messages_dir y lc_messages).
- El log_bin_trust_routine_creatorsvariable del sistema (uso log_bin_trust_function_creators).
- La myisam_max_extra_sort_file_size variable del sistema.
- La record_buffervariable del sistema (uso read_buffer_size).
- La sql_log_updatevariable del sistema.
- La table_lock_wait_timeoutvariable del sistema.[33]

Versión 5.6

Características añadidas en MySQL 5.6

Tipos de datos. Estos cambios de tipo de datos han sido implementados:

- MySQL ahora permite segundos fraccionarios para TIME, DATETIMEy TIMESTAMP valores, con una precisión de hasta microsegundos (6 dígitos).

- Anteriormente, como máximo, una `TIMESTAMP` columna por tabla podría inicializarse o actualizarse automáticamente a la fecha y hora actuales. Esta restricción ha sido levantada. Cualquier `TIMESTAMP` definición de columna puede tener cualquier combinación de `DEFAULT CURRENT_TIMESTAMP` y `ON UPDATE CURRENT_TIMESTAMP` cláusulas. Además, estas cláusulas ahora se pueden utilizar con `DATETIME` definiciones de columna.
- En MySQL, el `TIMESTAMP` tipo de datos se diferencia en formas no estándar de otros tipos de datos en términos de valor predeterminado y asignación de inicialización automática y atributos de actualización. Estos comportamientos siguen siendo los predeterminados. [33]

Mejoras en el esquema de rendimiento. El esquema de rendimiento incluye varias características nuevas:

- Instrumentación para la entrada y salida de tablas. Las operaciones instrumentadas incluyen accesos de nivel de fila a tablas base persistente o tablas temporales. Las operaciones que afectan a las filas son buscar, insertar, actualizar y eliminar.
- Filtrado de eventos por tabla, basado en el esquema y / o nombres de tablas.
- Filtrado de eventos por hilo. Se recopila más información para los hilos.
- Tablas de resumen para tablas e índices de E / S, y para bloqueos de tablas.
- Instrumentación para declaraciones y etapas dentro de declaraciones.
- Configuración de instrumentos y consumidores al inicio del servidor, que anteriormente solo era posible en tiempo de ejecución. [33]

Características eliminadas en MySQL 5.6

- Se prohíbe explícitamente asignar el valor `DEFAULT` a parámetros de función o función almacenados o variables locales de programa almacenadas
- La `--safe-mode` opción de servidor.
- La `--skip-thread-priority` opción de servidor. [33]

Versión 5.7

Características añadidas en MySQL 5.7

- Sistema y variables de estado. La información de la variable de sistema y estado ahora está disponible en las tablas de Esquema de rendimiento.
- Dos funciones de agregación JSON `JSON_ARRAYAGG ()` y `JSON_OBJECTAGG ()`. `JSON_ARRAYAGG ()` toma una columna o expresión como su argumento y agrega el resultado como una única JSON matriz. La expresión puede evaluar a cualquier tipo de datos MySQL.
- Soporte JSON. A partir de MySQL 5.7.8, MySQL es compatible con un JSON tipo nativo. Los valores JSON no se almacenan como cadenas, sino que utilizan un formato binario interno que permite el acceso de lectura rápida a los elementos del documento. [33]

Características eliminadas en MySQL 5.7

- Se elimina la compatibilidad con las contraseñas que utilizan el formato de hashing de contraseña anterior a 4.1, lo que implica los siguientes cambios. Las aplicaciones que utilizan alguna característica que ya no es compatible deben modificarse.
- En MySQL 5.6.6, el `YEAR (2)` tipo de datos estaba en desuso. El soporte para `YEAR (2)` ahora se ha eliminado. Una vez que actualice a MySQL 5.7.5 o superior, las `YEAR (2)` columnas restantes se deben convertir `YEAR (4)` para volver a ser utilizables.
- Las tablas con nombres especiales que se usan para habilitar y deshabilitar el `InnoDBMonitor` estándar y el `Monitor de InnoDBbloqueo` (`innodb_monitory innodb_lock_monitor`) se eliminan y reemplazan por dos variables dinámicas del sistema. [33]

2.2.18 Microsoft SQL Server

Es un sistema de gestión de bases de datos relacionales, o RDBMS, que admite una amplia variedad de aplicaciones de procesamiento de transacciones, inteligencia empresarial y análisis en entornos corporativos de TI. [36]

En SQL Server, cada columna, variable local, expresión y parámetro tiene un tipo de datos relacionado. Un tipo de datos es un atributo que especifica el tipo de datos que el objeto puede contener: datos enteros, datos de caracteres, datos monetarios, datos de fecha y hora, cadenas binarias, etc. [37]

SQL Server proporciona un conjunto de tipos de datos del sistema que definen todos los tipos de datos que se pueden usar con SQL Server. También puede definir sus propios tipos de datos en Transact-SQL o Microsoft .NET Framework. Los tipos de datos de alias se basan en los tipos de datos suministrados por el sistema. [37]

Los tipos de datos en SQL Server están organizados en las siguientes categorías:

- Numéricos exactos
- Cadenas de caracteres Unicode
- Numéricas aproximadas
- Cadenas de caracteres
- Fecha y hora
- Cuerdas binarias
- Otros tipos de datos [37]

En SQL Server, según sus características de almacenamiento, algunos tipos de datos se designan como pertenecientes a los siguientes grupos:

- Tipos de datos de gran valor: varchar (max) y nvarchar (max)
- Tipos de datos de objetos grandes: texto, ntext, imagen, varbinary (max) y xml

Versiones de Microsoft SQL Server

SQL Server 2008

Windows Server 2008 no se admite para servidores del sitio o roles del sistema de sitio excepto para el punto de distribución y el punto de distribución de extracción.

Características en desuso

- Implementación del servicio clásico en Azure para la puerta de enlace de administración y el punto de distribución en la nube.

- La experiencia de usuario de Silverlight del punto de sitios web del catálogo de aplicaciones ya no se admite. Los usuarios deben utilizar el nuevo Centro de software. [37]

Características eliminadas

- System Center Configuration Manager ha eliminado la compatibilidad con Protección de acceso a redes.
- La administración fuera de banda en System Center 2012 Configuration Manager no se ve afectada por este cambio. [37]

SQL Server 2012, 2014, 2016 y 2017

Estas versiones presentan mejoras en características, rendimiento, disponibilidad entre otros aspectos sin embargo para la finalidad de este análisis no presentan cambios significativos por lo cual se resume las versiones como las cuales puede ser usadas sin una versión de actualización acumulativa mínima para los sitios siguientes:

- Un sitio de administración central
- Un sitio primario
- Un sitio secundario [37]

2.2.19 NetBeans

Es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. [38]

2.2.20 Microsoft Visual Studio

Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic, permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así, se

pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros. [39]

2.2.21 Cadena de conexión

Es una cadena que contiene información acerca de una fuente de datos (generalmente un motor de base de datos), además de incluir la información necesaria para conectarse a la misma. Si bien las sintaxis exactas de dichas cadenas varían en función de la fuente de datos, generalmente deberá especificar lo siguiente: [22]

- Dirección del servidor de bases de datos.
- Nombre de la base de datos.
- Nombre de usuario y contraseña del usuario de la base de datos.

2.2.22 Base de datos no relacional

La respuesta a la necesidad de gestionar volúmenes masivos de información surge de la base de datos no relacionales, término acuñado a finales de los 90 y que engloba todas las tecnologías de almacenamiento estructurado que no cumplen el esquema relacional. [23]

La cantidad de información manejada por comunidades, redes sociales, buscadores, y muchos otros proyectos en el ámbito de la Web 2.0 es abrumadora, lo que ha hecho que surjan nuevas arquitecturas de almacenamiento de información, que deben ser de alto rendimiento, escalables y distribuidas. [23]

Aunque esta tecnología surgió de unas necesidades muy concretas, su difusión y algunos proyectos para encapsular sus funcionalidades y hacerlas más amigables a desarrolladores acostumbrados a SQL está provocando que también se usen en proyectos de pequeño tamaño, con lo que todo indica que a medio plazo convivirán con las bases de datos tradicionales independientemente del volumen de datos a gestionar. Las bases de datos no relacionales son aquellas que no poseen un esquema entidad relación y no utilizan SQL como el lenguaje principal para la realización de consultas. [23]

Entre las características más importantes de las bases de datos no relacionales se encuentran:

- No-relacionales
- Distribuidas
- Código abierto
- Escalabilidad horizontal

Las bases de datos NoSQL no poseen esquemas de usuarios, estructuras como tablas, no permiten la realización de uniones en las consultas, cuentan con una cantidad considerable de categorías, las cuales poseen características específicas que proporcionan soluciones óptimas y eficientes ante diversos problemas. [8]

Las características de las distintas categorías de bases de datos NoSQL existentes en la actualidad y las principales bases de datos que se encuentran en cada categoría.

Tabla 2.13 Categorías NoSQL

Categoría	Características	Ejemplos
Familia de columnas	Clave por cada columna. Una tupla es equivalente a una columna.	Hadoop Cassandra Stratosphere
Clave-valor	Datos almacenados en hashmap. Cada elemento posee llave única.	DynamoDB Redis Scalaris
Documentales	Información almacenada en documentos. Escalabilidad y alto rendimiento.	MongoDB CouchDB RavenDB
Orientadas a grafos	La información se almacena en nodos. Las relaciones son aristas y vértices	Neo4J InfoGrid DEX
Orientada a objetos	Integra lenguajes de programación y no solamente SQL.	Db4o Versant Perst
Bases XML (lenguaje de marcas extensible)	Los datos se almacenan en formato XML y se complementan con las documentales.	eXist Sedna Qizx

Fuente: S. E. Mancilla [8]

Las bases de datos no relacionales realmente es una categoría muy amplia para un grupo de soluciones de persistencia que no siguen el modelo de datos relacional, y que no utilizan SQL como lenguaje de consulta; pero, en resumen, las bases de datos no

relacionales pueden clasificarse en función de su modelo de datos en las siguientes cuatro categorías:

- Orientadas a clave - valor (Key - Value stores)
- Orientadas a columnas (Wide Column stores)
- Orientadas a documentos (Document stores)
- Orientadas a grafos (Graph databases) [23]

Orientadas a clave - valor

Los sistemas de clave-valor prometen un rendimiento excelente para volúmenes de datos muy grandes, a cambio de ser muy simples y renunciar a funcionalidades que tenemos en otros sistemas como la verificación intrínseca de la integridad de datos, llaves extranjeras y disparadores. Las validaciones de los datos se delegan completamente en la aplicación cliente, siendo la base de datos, simplemente el lugar donde se guardan los datos. No se verifican integridades, no se comprueban referencias cruzadas, todo esto se ha de implementar a nivel de aplicación, en el código del cliente. [23]

En un sistema relacional existen bases de datos y dentro de cada base de datos tenemos tablas formadas por filas y columnas. En un sistema clave-valor existen contenedores, también se les llama cabinets, en cada contenedor podemos tener tantas parejas de clave-valor como queramos. Hay sistemas que permiten tener claves duplicadas y hay otros que no, o que se puede indicar que no queremos que se dupliquen. En cada contenedor es posible tener datos de la misma naturaleza (por ejemplo, productos, pedidos, clientes, etc.) o totalmente diferentes (puede haber un contenedor por cliente), todo depende de los desarrolladores de la aplicación. [23]

Orientadas a columnas

Las bases de datos orientadas a columnas son probablemente más conocidas por la aplicación BigTable de Google o por la implementación Cassandra de Apache. A primera vista son muy similares a las bases de datos relacionales, pero en realidad son muy diferentes. Una de las principales diferencias radica en el almacenamiento de datos por filas (sistema relacional) versus el almacenamiento de datos por columnas

(sistema orientado a columnas) y otra la optimización de consultas para mejorar los tiempos de respuesta en comparación con los sistemas relacionales. [23]

Las bases de datos orientadas a columnas son en realidad lo que se podría suponer, tablas de datos donde las columnas de valores de datos representan el almacenamiento estructural. Los datos son almacenados como secciones de las columnas de datos en lugar de filas de datos, como en la mayoría de los gestores relacionales. Esto tiene ventajas para los almacenes de datos, sistemas de gestión de relaciones con clientes, catálogos de bibliotecas de tarjetas y otros sistemas ad-hoc de consulta donde los agregados se calculan a través de un gran número de elementos de datos similares. [23]

Orientadas a documentos

Una base de datos orientada a documentos está diseñada para gestionar información orientada a documentos o datos semi-estructurados. Este tipo de bases de datos constituye una de las principales categorías de las llamadas bases de datos NoSQL. La popularidad del término "base de datos orientada a documentos" o "almacén de documentos" ha crecido a la par con el uso del término NoSQL en sí. A diferencia de las conocidas bases de datos relacionales con su definición de "tabla", los sistemas documentales están diseñado entorno a la definición abstracta de un "documento". Almacenar y recuperar todos los datos relacionados como una sola unidad puede entregar ventajas enormes en el rendimiento y la escalabilidad. De este modo, los gestores de datos no tienen que hacer operaciones complejas como las uniones para encontrar los datos que normalmente están relacionados, ya que todo se encuentra en un mismo lugar. [23]

Orientadas a grafos

Las bases de datos orientadas a grafos representan la información como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar teoría de grafos para recorrer la base de datos ya que esta puede describir atributos de los nodos (entidades) y las aristas (relaciones). [23]

Una base de datos orientada a grafos debe estar absolutamente normalizada, esto quiere decir que cada tabla tendría una sola columna y cada relación tan solo dos, con esto se consigue que cualquier cambio en la estructura de la información tenga un efecto tan solo local. [23]

Este tipo de base de datos está diseñada para los datos cuyas relaciones son bien representadas en forma de grafo, o sea, los datos son elementos interconectados con un número no determinado de relaciones entre ellos. La información a gestionar por este tipo de almacenamiento pudiera ser las relaciones sociales, el transporte público, mapas de carreteras o topologías de red, entre otros ejemplos. [23]

Por definición, una base de datos orientada a grafos es cualquier sistema de almacenamiento que permite la adyacencia libre de índice. Esto quiere decir que cada elemento contiene un puntero directo a sus elementos adyacentes por lo cual no es necesario realizar consultas por índices. [23]

2.2.23 Microsoft Visual FoxPro

El sistema de desarrollo de bases de datos Microsoft® Visual FoxPro® es una herramienta poderosa para crear rápidamente aplicaciones de bases de datos de escritorio, cliente enriquecido, cliente distribuido, cliente / servidor y web de alto rendimiento. Utilice su potente motor de datos para administrar grandes volúmenes de datos, su programación orientada a objetos para reutilizar componentes en todas las aplicaciones, sus funciones de servicios web XML para aplicaciones distribuidas y su soporte XML integrado para manipular rápidamente los datos. [24]

Los desarrolladores tendrán las herramientas necesarias para administrar datos, desde la organización de tablas de información, la ejecución de consultas y la creación de un DBMS (Sistema integrado de administración de bases de datos) hasta la programación de una aplicación de administración de datos completamente desarrollada para los usuarios finales. [24]

- Manejo de datos e interoperabilidad. Cree soluciones compatibles con .NET con XML jerárquico y servicios web XML. Intercambie datos con SQL Server a través de capacidades de lenguaje SQL mejoradas y tipos de datos recientemente admitidos.
- Desarrollador extensible herramientas de productividad. Mejore sus interfaces de usuario con formularios de usuario acoplables, anclaje automático de controles y soporte de imagen mejorado. Personalice la

ventana de propiedades con sus propiedades favoritas, editores personalizados, fuentes y configuraciones de color.

- Flexibilidad para construir todo tipo de soluciones de base de datos. Cree e implemente aplicaciones remotas e independientes para Tablet PC basadas en Windows. Cree y acceda a componentes y servicios web XML compatibles con la tecnología Microsoft .NET.
- Características del sistema de informes. La nueva arquitectura de salida extensible proporciona un control preciso de la salida y el formato de los datos del informe. Diseño con múltiples bandas de detalle, rotación de texto y encadenamiento de informes. Los informes de salida compatibles incluyen en XML, HTML, formatos de imagen y una ventana de vista previa de impresión personalizable de varias páginas. Compatible con versiones anteriores de los informes de Visual FoxPro existentes.

Tabla 2.14 Tipos de datos en Visual FoxPro

Tipo	Byte	Ejemplos
Numérico	1 á 20	-.9999999999E+19 a .9999999999E+20
Punto Flotante	1 á 20	-.9999999999E+19 a .9999999999E+20
Memo	4	Texto longitud variable - Máximo según memoria
Doble Precisión	8	+/-4.94065645841247E-324 a +/-8.9884656743115E307
Character	1 á 254	“Prueba” “123” “01/01/95” Cualquier carácter
Logical	1	.T.(verdadero) .F. (falso)
Integer (Entero)	4	-2147483647 a 2147483646
Carácter (Binario)	10	Sin tabla de códigos
Memo (Binario)	10	Sin tabla de códigos
General (Obj.OLE)	4	Limitado por Memoria disponible
Fecha - Fecha y hora (Date - DateTime)	8	{01/01/95} {01/01/95 12:30:00 pm}

Elaborado por: El Investigador

2.2.24 Los gestores de bases de datos más usados

El Ranking BD-Engines clasifica los sistemas de administración de bases de datos según su popularidad, 341 sistemas en ranking.

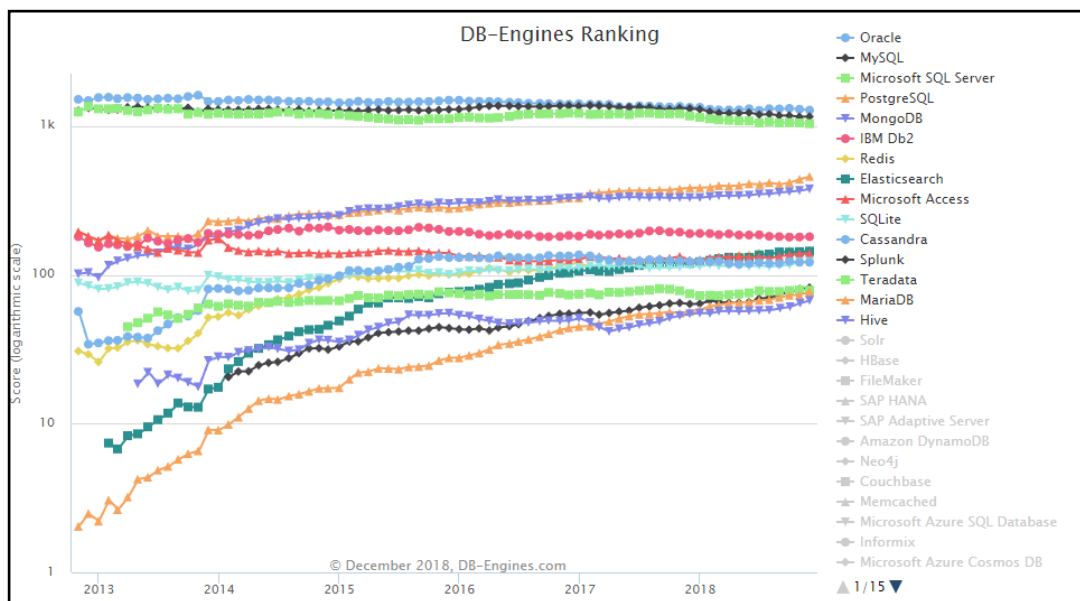
Fig. 2.2 Ranking de las bases de datos

Rank			DBMS	Database Model	Score		
Dec 2018	Nov 2018	Dec 2017			Dec 2018	Nov 2018	Dec 2017
1.	1.	1.	Oracle +	Relational DBMS	1283.22	-17.89	-58.32
2.	2.	2.	MySQL +	Relational DBMS	1161.25	+1.36	-156.82
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1040.34	-11.21	-132.14
4.	4.	4.	PostgreSQL +	Relational DBMS	460.64	+20.39	+75.21
5.	5.	5.	MongoDB +	Document store	378.62	+9.14	+47.85
6.	6.	6.	IBM Db2 +	Relational DBMS	180.75	+0.87	-8.83
7.	7.	↑8.	Redis +	Key-value store	146.83	+2.66	+23.59
8.	8.	↑10.	Elasticsearch +	Search engine	144.70	+1.24	+24.92
9.	9.	↓7.	Microsoft Access	Relational DBMS	139.51	+1.08	+13.63
10.	10.	↑11.	SQLite +	Relational DBMS	123.02	+0.31	+7.82
11.	11.	↓9.	Cassandra +	Wide column store	121.81	+0.07	-1.40
12.	12.	↑15.	Splunk	Search engine	82.18	+1.81	+18.39
13.	13.	↓12.	Teradata +	Relational DBMS	79.16	-0.14	+4.42
14.	14.	↑17.	MariaDB +	Relational DBMS	77.26	+4.01	+20.52
15.	15.	↑19.	Hive +	Relational DBMS	67.38	+2.81	+12.71

Fuente: Tendencia de popularidad [25]

En esta figura se presenta las bases de datos más populares y el modelo que estas utilizan con esta información se establece una idea de cuáles son las bases de datos más utilizadas.

Fig. 2.3 Tendencia de las bases de datos



Fuente: Tendencia de popularidad [25]

Apreciando las tendencias de las bases de datos se evidencia que Oracle, MySQL y Microsoft SQL Server son las que se mantienen estables como líderes de bases de datos.

El Ranking DB-Engines es una lista de los sistemas de gestión de bases de datos clasificados por su popularidad actual. Se mide la popularidad de un sistema utilizando los siguientes parámetros:

- Número de menciones del sistema en sitios web, medido como número de resultados en consultas de motores de búsqueda. En este momento, se utiliza Google, Bing y Yandex para esta medición. Para contar solo los resultados relevantes, se busca <nombre del sistema> junto con el término base de datos, por ejemplo, "Oracle" y "base de datos".
- Interés general en el sistema. Para esta medida, se utiliza la frecuencia de búsquedas en Google Trends.
- Frecuencia de las discusiones técnicas sobre el sistema. Se utiliza la cantidad de preguntas relacionadas y la cantidad de usuarios interesados en los conocidos sitios de preguntas y respuestas relacionados con TI Stack Overflow y DBA Stack Exchange.
- Número de ofertas de trabajo, en las que se menciona el sistema. Se utiliza la cantidad de ofertas en los principales motores de búsqueda de empleo, Indeed y Simply Hired.
- Número de perfiles en redes profesionales, en los que se menciona el sistema. Se utiliza las redes profesionales más populares internacionalmente LinkedIn y Upwork.
- Relevancia en las redes sociales. Se cuenta el número de tweets de Twitter, en los que se menciona el sistema. [26]

Se calcula el valor de popularidad de un sistema al estandarizar y promediar los parámetros individuales. Estas transformaciones matemáticas se realizan de manera que se mantenga la distancia de los sistemas individuales. Eso significa que, cuando el sistema A tiene un valor dos veces mayor en la clasificación de los motores de base de datos que el sistema B, entonces es dos veces más popular cuando se promedia sobre los criterios de evaluación individuales. [26]

Para eliminar los efectos causados por las cantidades cambiantes de las fuentes de datos, el puntaje de popularidad es siempre un valor relativo, que debe interpretarse en comparación con otros sistemas solamente. [26]

Con esta información se establece a las siguientes bases de datos como las más usadas

- MySQL
- Microsoft SQL Server
- Oracle
- Microsoft Access
- PostgreSQL

2.2.25 Comprobante electrónico

Es un documento que cumple con los requisitos legales y reglamentarios exigibles para todos los comprobantes de venta, garantizando la autenticidad de su origen y la integridad de su contenido y que se emite a través de una nueva modalidad electrónica autorizada por el Servicio de Rentas Internas. [27]

Beneficios

- Tiene la misma validez que los documentos físicos.
- Reducción de tiempos de envío de comprobantes.
- Ahorro en el gasto de papelería física y su archivo.
- Contribuye al medio ambiente, debido al ahorro de papel y tintas de impresión.
- Mayor seguridad en el resguardo de los documentos.
- Menor probabilidad de falsificación.
- Procesos administrativos más rápidos y eficientes.

¿Qué documentos pueden ser emitidos electrónicamente?

- Facturas
- Notas de crédito
- Notas de débito
- Comprobantes de retención

- Guías de remisión [28]

¿Quiénes están obligados a emitir comprobantes electrónicos?

Sector privado

Las empresas del sector privado tienen obligación de emitir comprobantes si cumplen con los siguientes requisitos.

Fig. 2.4 Sector privado obligados a emitir comprobantes electrónicos

Grupo	Fecha de inicio	Sujetos pasivos
1	A partir de 01/08/2014	Sociedades emisoras y administradoras de tarjetas de crédito
2	A partir de 01/10/2014	Instituciones financieras bajo el control de la Superintendencia de Bancos y Seguros excepto mutualistas de ahorro y crédito para la vivienda y sociedades emisoras y administradoras de tarjetas de crédito. Contribuyentes especiales que realicen, según su inscripción en el RUC (Registro único de contribuyentes) actividades económicas correspondientes al sector telecomunicaciones y subsectores; televisión pagada. Exportadores calificados por el SRI como contribuyentes especiales.
3	A partir de 01/01/2015	Los demás contribuyentes especiales señalados en los grupos anteriores. Contribuyentes que posean autorización de impresión de comprobantes de venta, retención y documentación complementaria, a través de sistemas computarizados (autoimpresores). Contribuyentes que realicen ventas a través de internet. Los sujetos pasivos que realicen actividades económicas de exportación.

Fuente: Comprobante electrónico - SRI [28]

Sector público

Las empresas del sector público tienen obligación de emitir comprobantes si cumplen con los siguientes requisitos.

Fig. 2.5 Sector público obligados a emitir comprobantes electrónicos

Grupo	Fecha de inicio	Sujetos pasivos
1	01/01/2015	Empresas públicas y empresas de servicios públicos. Entidades del sector público financiero. Empresas de economía mixta.
2	01/04/2015	Organismos y entidades de la Función Ejecutiva. La Asamblea Nacional. Organismos y entidades de la función judicial, con excepción de sus organismos auxiliares mencionados en el Art. 178 de la Constitución de la República del Ecuador. Los organismos y entidades de la Función de Transparencia y Control Social. Los organismos y entidades de la Función Electoral. Universidades y Escuelas Politécnicas públicas.
3	01/07/2015	Los organismos y entidades de los Gobiernos Autónomos Descentralizados, incluidas las mancomunidades conformadas por los mismos. Los organismos y entidades públicas no descritas en ninguno de los grupos señalados anteriormente.

Fuente: Comprobante electrónico - SRI [28]

2.2.26 Metodologías Ágiles

Las metodologías en general se clasifican según su enfoque y características esenciales, las más recientes, que se fueron gestando a finales del siglo pasado y que se han comenzado a manifestar desde principios del actual, se han denominado “metodologías ágiles” y surgen como una alternativa a las tradicionales, estas metodologías se derivan de la lista de los principios que se encuentran en el “Manifiesto Ágil”. [40]

La aparición de las metodologías ágiles no puede ser asociada a una única causa, sino a todo un conjunto de ellas, si bien es cierto que la mayoría de autores lo relacionan con una reacción a las metodologías tradicionales, ¿cuáles fueron las causas de esta reacción?, los factores que comúnmente se mencionan son la pesadez, lentitud de reacción y exceso de documentación, en definitiva, falta de agilidad de los modelos de desarrollo formales; otro punto importante sería la explosión de la red, las aplicaciones Web y las aplicaciones móviles, así como el crecimiento notorio del movimiento open source. [40]

Tabla 2.15 Diferencias entre las metodologías ágiles y no ágiles

Metodologías ágiles	Metodologías tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Fuente: M. Rouse [36]

Entre las metodologías ágiles más destacadas están:

- XP (Extreme Programming)
- Scrum
- Crystal Clear
- DSDM (Dynamic Systems Development Method)
- FDD (Feature Driven Development)

XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [41]

Historias de usuario

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. [43]

Independiente una historia debería ser independiente de otras (lo más posible). La dependencia entre las historias hace que sea más difícil planificar, priorizar y estimar. A menudo, se puede reducir las dependencias haciendo una combinación de historias, o partiendo historias de forma diferente. [43]

Pequeña una buena historia debe ser pequeña en esfuerzo, generalmente representando no más de 2-3 personas/semana de trabajo. Una historia que es más grande va a tener más errores asociados a la estimación y alcance. [43]

Planeación del lanzamiento.

El cliente define, utilizando Historias de Usuario, los requerimientos de manera general, y precisa su importancia; Con base en ellas, los desarrolladores estiman el costo de implementarlas y se definen las características de una entrega y el número de iteraciones que se necesitarán para terminarla. [41]

Iteraciones

Entregas Pequeñas se refiere al uso de ciclos cortos de desarrollo (iteraciones) que le muestran software terminado al cliente y obtienen retroalimentación de él. La

definición de terminado está relacionada con las pruebas de aceptación. Diseño Simple indica que el sistema debe ser tan simple como sea posible, en un momento determinado, lo cual implica que los desarrolladores deben preocuparse únicamente por las historias de usuario planeadas para la iteración actual, sin importar cuanto pueden cambiar por funciones futuras. [41]

Velocidad del proyecto

Es una medida de la capacidad que tiene el equipo de desarrollo para evacuar las historias de usuarios en una determinada iteración. Esta medida se calcula totalizando el número de historias de usuario realizadas en una iteración. Para la iteración siguiente se podrá (teóricamente) implementar el mismo número de historias de usuario que en la iteración anterior. [44]

Programación en pareja

XP promueve que todo el código sea escrito en parejas trabajando en el mismo ordenador. La programación en parejas incrementa la calidad del código sin impactar en la fecha de entrega. En contra de lo que parece, dos personas que trabajan en un mismo equipo añadirán la misma funcionalidad que dos personas trabajando por separado, excepto que el código será de mucha mayor calidad. [45]

Reuniones diarias

El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie. [45]

Simplicidad

Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando. Características fundamentales del código: Testeable, legible, comprensible y explicable. [45]

Metáforas

Una “metáfora” es algo que todos entienden, sin necesidad de mayores explicaciones. La metodología XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo. [45]

Solución “spike”

Una solución “spike”, es una solución muy simple para plantear posibles soluciones, de manera, que solamente se aborda el problema en concreto y se aísla de otro tipo de preocupaciones. [45]

Refactorización

La recodificación consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible. [45]

Glosario de términos

Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código. [42]

Riesgos

Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema. [42]

Funcionalidad extra

Nunca se debe añadir funcionalidad extra al programa, aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos. [42]

Refactorizar

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. [42]

Codificación

A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada. [42]

Pruebas

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos. El uso de las pruebas en XP es el siguiente:

- Se deben crear las aplicaciones que realizarán las pruebas con un entorno de desarrollo específico para test.
- Hay que someter a test las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.
- Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican.
- Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.

Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos. [42]

SCRUM

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está

especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. [41]

Crystal Methodologies

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo, Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). [41]

DSDM

Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases. [41]

FDD

Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad. [41]

2.3 Propuesta de solución

El presente trabajo tiene como beneficiario a la empresa Solinfo, con el desarrollo de un sistema de transferencia de datos que facilite el proceso de transferencia de datos de múltiple origen hacia Oracle, el sistema abarcará el proceso de obtención de datos, este proceso se encargará de extraer los datos seleccionados desde una base de datos origen, el proceso de transformación de datos que compara los tipos de datos y su codificación para adaptarlos o convertirlos a su equivalente en la base de datos destino y el proceso de carga de datos el cual realizará la carga de los datos ya transformados al esquema de la base de datos destino, estos procesos son esenciales para llevar a cabo una transferencia de datos, el sistema de transferencia de datos realizará los procedimientos previo al servicio de comprobantes electrónicos que actualmente brinda la empresa Solinfo.

CAPÍTULO III

METODOLOGÍA

3.1 Modalidad de la aplicación

Esta investigación utilizará la modalidad de investigación de campo ya que para obtener la información necesaria se acudirá a Solinfo y observar el proceso que lleva acabo para la obtención de la información, necesario para el servicio de comprobantes electrónicos; además se analizará la plantilla que utiliza la empresa para parametrizar la información y así se ajuste a la arquitectura que maneja la Solinfo.

3.1.1 Investigación documental – bibliográfica

Se considera esta modalidad ya que se adquirirá definiciones y análisis de trabajos con temas relacionados y la recopilación de información necesaria para el desarrollo del sistema que se propone en el presente trabajo.

3.1.2 Investigación aplicada

Esta modalidad se aplicó por la utilización de los conocimientos adquiridos a lo largo de la carrera universitaria.

3.2 Población y muestra

La presente investigación por su característica no requiere población ni muestra.

3.3 Recolección de la información

Para la obtención de información que conlleve a cabo los procesos ligados a la recolección de datos necesarios para el servicio de comprobantes electrónicos se optará por la investigación de campo, analizando el proceso actual que se utiliza para la parametrización de la información desde el servidor de la empresa y el análisis del

esquema de la base de datos y sus procedimientos almacenados que inician con el servicio de comprobantes electrónicos.

3.4 Procesamiento y análisis de datos

Los datos que se recolectará con los métodos antes mencionados se utilizarán para su respectivo análisis con el fin de sustentar cuantitativamente el aporte que generará la propuesta planteada contra en proceso que se lleva a cabo actualmente, con la ayuda de graficas que faciliten la comprensión cuantitativa de los resultados.

De la información recogida por la observación de campo se realizará una revisión y análisis, que sustente los cambios que se propone en la selección y obtención de información para el servicio de comprobantes electrónicos. Con la finalidad de una mejor comprensión del análisis de los datos obtenidos se utilizará cuadros comparativos.

3.5 Desarrollo del proyecto

Las siguientes actividades se realizarán para cumplir los objetivos establecidos dentro del presente trabajo de investigación.

- Análisis de la situación actual
- Requerimientos del sistema de transferencia de datos de origen múltiple a Oracle
- Selección de las bases de datos y sus versiones
- Selección del lenguaje de programación
- Selección de las herramientas y recursos adecuados
- Metodologías
- Metodologías tradicionales
- Metodologías ágiles
- Metodología aplicada
- Análisis de resultados
- Implementación del sistema de transferencia de datos

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1 Análisis de la situación actual

Actualmente la empresa Solinfo realiza los procesos de obtención, transformación y carga de datos de manera manual, en estos procesos se encuentran involucrados la empresa Solinfo y la empresa solicitante del servicio de comprobantes electrónicos, las empresas que buscan el servicio de comprobantes electrónicos deben proporcionar los datos necesarios para dicho servicio, estos datos pueden estar almacenados en bases de datos relacionales, no relacionales, hojas digitales, en archivos planos, entre otras formas de almacenamiento de datos.

Los datos pasan por un proceso de transformación para que se acoplen a la arquitectura de la empresa Solinfo la cual trabaja con la base de datos Oracle, y posteriormente realizar la carga de datos a la arquitectura ya establecida en la base de datos de la empresa sin crear una nueva base de datos sino que únicamente se cargan los datos que proporciona la empresa solicitante, este proceso se lleva a cabo con el personal de la empresa solicitante y Solinfo con el fin de proporcionar únicamente los datos necesarios, garantizando la integridad y confidencialidad de los datos.

Los procesos de selección, transformación y carga de datos requiere de personal con conocimientos en diversas bases de datos por parte de las empresas involucradas para garantizar que los datos son los indicados, así evitar pérdida de tiempo y esfuerzo, además se requiere un tiempo adicional en el proceso de transformación de datos en el caso que la arquitectura origen y destino sean diferentes, los aspectos más relevantes son el tipo de conexión, los diferentes tipos de datos y sus equivalentes, diferentes conjuntos de caracteres (codificaciones distintas en cada columna para una misma

tabla), condiciones establecidas (claves primarias, foráneas, auto numéricos entre otros) y la lógica empresarial.

La empresa actualmente utiliza un servidor IBM System x3100 M4 con las siguientes características.

Tabla 4.16 Características del servidor de Solinfo

Factor forma/altura	Torre
Procesador	SerienIntel Xeon E3-1200
Número de procesadores	Uno/uno
Caché	8MB
Memoria	1x2 ó 1x4 GB estándar, máximo 32 GB
Ranuras de expansión	4 ranuras PCIe Gen 2
Bahías de discos	4 serial ATA (SATA)
Almacenamiento interno	12 TB SATA
Interfaz de red	Gigabit Ethernet dual
Fuente de alimentación	350 W fijos o 80-Plus
Soporte RAID	ServeRAID – C100
Gestión de sistemas	IMM2
Sistemas operativos compatibles	Microsoft Windows Server 2008 R2 / Microsoft Windoes Server 2008, Red Hat Linux, Suse Linux
Garantía limitada	Un año de unidad sustituible por el cliente (CRU)

Fuente: Solinfo

La cual contiene una base de datos Oracle 11g Enterprise Edition Release 11.2.0.3.0 – 64 bits Pro, con las siguientes características.

Tabla 4.17 Características de Oracle

	Banner
1	Oracle Database 11g Enterprise Edition Release
2	PL/SQL Release 11.2.0.3.0 - Production
3	Core 11.2.0.3.0 Production
4	TNS for Linux: Version 11.2.0.3.0 - Production
5	NLSRTL Version 11.2.0.3.0 - Production

Fuente: Solinfo

En dicha base de datos se almacenan los datos relacionados con la empresa y sus clientes y se realizan los procesos de carga de datos de fuentes externas como archivos planos o formatos predefinidos que se acogen a las necesidades de la empresa. Para acceder al servidor se utiliza un computador de escritorio con las siguientes características.

Tabla 4.18 Características del computador del administrador

Características	
Procesador	Intel® Core™ i7-4770 CPU 3.40 GHz
Memoria Instalada (RAM)	12 GB
Tipo de Sistema	Sistema Operativo de 64 bits, procesador x 64
Sistema Operativo	Windows 8.1 Pro

Elaborado por: El Investigador

Este computador realiza la conexión al servidor a través de la red de la empresa, las características de este computador se consideraron como requerimientos, ya que este computador será en el que se realice la implementación del sistema de transferencia de datos.

Con el análisis de la situación actual del proceso de transferencia de datos (obtención, transformación y carga de datos), a continuación, se determina los beneficios que brindará el Sistema de Transferencia de Datos de Origen Múltiple a Oracle:

- Ahorrar de tiempo y esfuerzo.
- Vincular directamente los datos desde un múltiple origen hacia un destino.

- Automatizar procesos de obtención, transformación y carga de datos de un esquema de base de datos a otro esquema.
- Acceder a diferentes bases de datos relacionales y no relacionales desde un mismo sistema.
- Establecer conexiones remotas.

4.2 Requerimientos del sistema de transferencia de datos de origen múltiple a Oracle

Para el desarrollo del sistema de transferencia de datos, se realiza la selección de las bases de datos y sus versiones, del lenguaje de programación, las herramientas y recursos adecuados.

- Selección de las bases de datos y sus versiones
- Selección del lenguaje de programación
- Selección de las herramientas y recursos adecuados

4.2.1 Selección de las bases de datos y sus versiones

Para la selección de las bases de datos se consideraron las bases más populares de acuerdo a el Ranking BD-Engines y el método que emplea, acorde a esto se analizan las versiones de las bases de datos PostgreSQL, MySQL, Oracle, Microsoft SQLServer para las bases de datos relacionales y para las bases de datos no relacionales se centró en trabajar directamente con los archivos que este tipo de base de datos generan.

PostgreSQL

Las versiones consideradas son las siguientes.

Tabla 4.19 Versiones consideradas de PostgreSQL

Liberación	Fecha Liberación	Tiene Soporte
9.2	10/09/2012	No
9.3	09/09/2013	No
9.4	18/12/2014	Si
9.5	07/01/2016	Si
9.6	29/09/2016	Si
10	05/10/2017	Si

Elaborado por: El Investigador

Con lo investigado sobre PostgreSQL la versión 9.5 ha realizado el ajuste de la prioridad del operador para que coincida con el estándar SQL y tiene soporte disponible, su actualización 9.5.15 es la más reciente fue lanzada el 08-11-2018, esta versión cuenta con alrededor de 2 años en el mercado y con soporte disponible por lo cual se encuentra establecida en el mercado, por estos motivos esta versión es la seleccionada para agregarse al sistema.

MySQL

Las versiones consideradas son las siguientes.

Tabla 4.20 Versiones consideradas de MySQL

Versión	Fecha Inicio / Fin	Desarrollo está activo
5	2004-2012	No
5.1	2006 - 2014	No
5.5	2010 - 2015	Si
5.6	2011 - 2015	Si
5.7	2013 - actualmente	Si

Elaborado por: El Investigador

La versión 5.5 de MySQL es la seleccionada para agregarse al sistema con su última actualización 5.5.62 en su edición community ya que no presenta cambios con los tipos de datos o funcionalidad de estos, además que ya se encuentra establecida en el mercado.

Oracle

Como ya lo mencionamos este gestor de base de datos puede ser tanto el origen como el destino siendo la única base de datos que tiene esta característica en el sistema, las versiones consideradas son las siguientes.

Tabla 4.21 Versiones consideradas de Oracle

Versión	Fecha Inicio / Fin	Desarrollo está activo
10.2	2006 - 2013	No
11.1	2007 - 2015	Si
11.2	2009 - 2018	Si
12.1	2013 - actualmente	Si

Elaborado por: El Investigador

Para la base de datos Oracle se seleccionó la versión 11.2 ya que cuenta con soporte y es una de las más utilizadas de las versiones mostradas.

Microsoft SQL Server

Las versiones consideradas son las siguientes.

Tabla 4.22 Versiones consideradas de Microsoft SQL Server

Versión	Año	Nombre de la versión	Nombre clave
10.50	2010	SQL Server 2008 R2	Kilimanjaro
11.0	2012	SQL Server 2012	Denali
12.0	2014	SQL Server 2014	SQL 14
13.0	2016	SQL Server 2016	-
14.0	2017	SQL Server 2017	Next 2017

Elaborado por: El Investigador

La versión SQL Server 2012 es la seleccionada para agregarse como base al sistema de transferencia de datos, ya que es de las versiones más utilizadas al igual que SQL Server 2008, esto se debe a que tienen tiempo en el mercado y cuentan con soporte necesario para su correcto funcionamiento, además de que es una de las bases de datos más utilizadas.

Bases de datos no relacionales

Para las bases de datos no relacionales se planteó establecer la conexión directamente con los archivos con la extensión dbf que estas bases de datos generan, esto con la finalidad de abarcar de una mejor manera las bases de datos no relacionales, esta propuesta es posible porque el sistema se orienta a los datos y no al esquema o estructura de las bases de datos.

Las versiones de las bases de datos seleccionadas como base no implican que el sistema no funcionará con otras versiones, sino que los plugin que se utilizan para realizar la conexión con dichas versiones de las bases de datos son específicos para esas versiones y pueden presentar inconvenientes con otras versiones.

Por petición de Solinfo el sistema no contara con seguridades adicional a las seguridades propia de cada una de las bases de datos que contiene el sistema, por motivos de proporcionar un mayor grado de usabilidad al sistema, es decir el sistema no cuenta con un módulo de inicio de sesión o autenticación de usuarios, que dificulte el acceso al sistema.

4.2.2 Selección del lenguaje de programación

A continuación, se presenta una comparación de las características antes mencionadas

Tabla 4.23 Comparación entre Netbeans y Visual Studio

	NetBeans	Visual Studio
Desarrollador	Apache software Foundation Oracle Corporation	Microsoft
Soporta S.O.	Windows	Windows
	MacOS	
	Linux	MacOS
	Solaris	
Lanzamiento estable	3 de octubre de 2016	8 de mayo de 2018
RAM	512 MB	2 GB
Procesador	Intel Pentium III de 800MHz o equivalente	1.8 GHz
Espacio en disco duro	750 MB	20 - 50 GB
Idioma disponible	28 idiomas	12 idiomas
Website	netbeans.apache.org	visualstudio.com
Ediciones	-	Enterprise
		Professional
		Community

Elaborado por: El Investigador

Conocimiento del desarrollador

La empresa Solinfo no estableció un lenguaje de programación como requerimiento por lo tanto el análisis para la selección del lenguaje de programación que mejor se adapte al caso está a cargo del desarrollador. Por parte del desarrollador cuenta con la experiencia en los dos lenguajes de programación con una inclinación hacia netbeans al ser el lenguaje de programación que más ha utilizado por ende está más familiarizado con este y cuenta con mayor experiencia en dicho lenguaje de programación.

Finalmente, netbeans proporciona la ventaja que intenta ser un solo lenguaje, es decir una vez compilado el sistema se ejecutará en cualquier hardware y bajo cualquier sistema operativo, mientras que visual studio requiere de un framework y se ejecutarán en Windows, esta ventaja es importante ya que el sistema deberá ser escalable.

Las características del computador en el que se realiza el desarrollo del sistema son las siguientes.

Tabla 4.24 Características del computador del administrador de Solinfo

Características	
Procesador	AMD A9-9420 RADEON R5
Memoria Instalada (RAM)	6 GB
Tipo de Sistema	Sistema Operativo de 64 bits, procesador x 64
Sistema Operativo	Windows 10 Pro

Elaborado por: El Investigador

El lenguaje de programación java en la API Netbeans brinda mayor facilidad de manipulación de controles, para personalizarlos y al estar más familiarizado, se optó por este lenguaje, otro aspecto relevante será las características del computador del desarrollador, ya que se necesita instalar las bases de datos para realizar pruebas al sistema.

4.2.3 Selección de las herramientas y recursos adecuados

Una vez seleccionadas las bases de datos, sus versiones y el lenguaje de programación, es necesario establecer si se requiere de componentes adicionales para llevar a cabo el sistema propuesto. Como el sistema de transferencia de datos trabajará con las bases de datos mencionadas anteriormente, se requiere tener acceso a dichas bases de datos para llevar a cabo las pruebas necesarias, para este fin se formuló crear una máquina virtual que contenga las bases de datos, además servirá para probar la funcionalidad de conexiones remotas a las bases de datos.

El Sistema de Transferencia de datos tiene como objetivo proporcionar un único sistema para la obtención, transformación y carga de datos que permite conectarse a diferentes bases de datos relacionales y no relacionales como origen y a Oracle como destino con el fin de mejorar un proceso anterior al servicio de comprobantes electrónicos.

Para la creación de la máquina virtual se utilizará virtual box en su versión 5.1.34 porque esta herramienta ha sido utilizada anteriormente, se está familiarizado con ella y no presentó ningún problema con las características del computador del investigador anteriormente mencionadas.

Los componentes adicionales para realizar las conexiones con las bases de datos son los siguientes:

Tabla 4.25 Conectores de las bases de datos

Base de datos	Componente	Fuente
PostgreSQL	postgresql-9.4.1212	https://jdbc.postgresql.org/download.html#others
MySQL	mysql-connector-java-5.1.46	https://dev.mysql.com/downloads/connector/j/5.1.html
Oracle	ojdbc7	https://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html
SqlServer	sqljdbc42	https://mvnrepository.com/artifact/com.microsoft.sqlserver/sqljdbc42?repo=cvut-ida
Bases de datos no relacionales	DBF_JDBC30	http://freshmeat.sourceforge.net/projects/javadbfs/

Elaborado por: El Investigador

4.3 Metodologías

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.

4.3.1 Metodologías ágiles

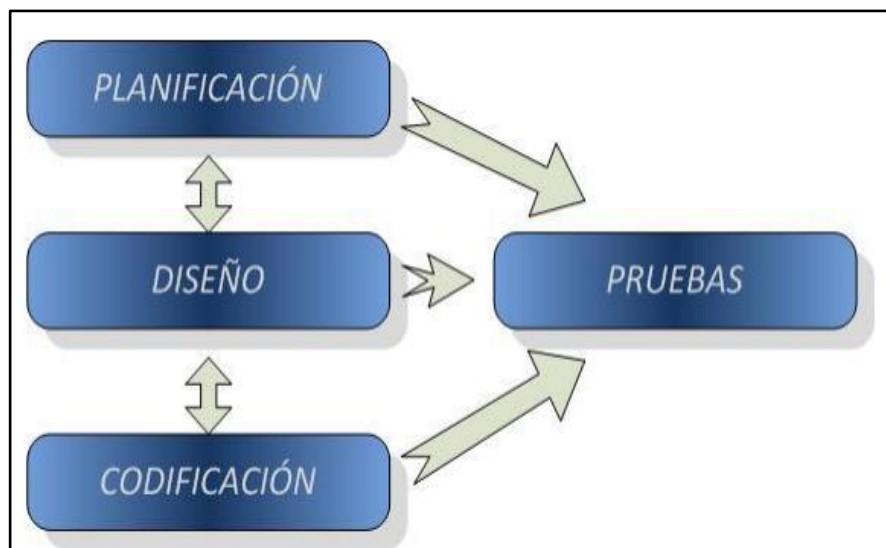
De las metodologías ágiles mencionadas la metodología XP fomenta la comunicación entre el equipo de desarrollo y el cliente, cuenta con facilidad para cubrir cambios de requisitos y establece iteraciones funcionales con el fin de realizar retroalimentación entre el cliente y el equipo de desarrollo, con estas características la metodología XP se adecua a las necesidades que plantea el presente trabajo, proporcionando los mecanismos necesarios para afrontar cambios en los requisitos y mitigar los riesgos técnicos que existen por la naturaleza del problema, como la compatibilidad entre los tipos de datos, la codificación de caracteres, las conexiones a las bases de datos, el tratamiento de los datos dentro del lenguaje como tipo objeto, entre otros.

4.4 Metodología aplicada

Por las razones antes mencionadas el presente trabajo se desarrollará bajo la metodología XP la misma que cuenta con las siguientes fases:

- **Fase: Planificación del proyecto.**
 - Historias de usuario.
 - Planeación del lanzamiento.
 - Iteraciones.
 - Velocidad del proyecto.
 - Programación en pareja.
 - Reuniones diarias.
- **Fase: Diseño.**
 - Diseños simples.
 - Glosarios de términos.
 - Riesgos.
 - Funcionalidad extra.
 - Refactorizar.
- **Fase: Codificación.**
- **Fase: Pruebas.** [42]

Fig. 4.6 Fases de la metodología XP



Fuente: A. Fallis [39]

4.4.1 Fase: Planificación del proyecto

Historias de usuario

Las historias de usuarios se desarrollaron entre el investigador con el representante y tutor de la empresa Solinfo en las cuales se abordó el proceso de obtención, transformación y carga de datos que especificaciones utiliza qué aspectos se busca añadir bajo que parámetros trabajara entre otras características.

Para establecer las historias de usuario para cada iteración se optó por definir las historias de usuario por nivel de funcionalidad en cada proceso del sistema de este modo las historias se dividieron en cuatro actividades fundamentales que representan procesos por los cuales el sistema debe pasar para completar su objetivo, estos procesos son:

- Conexiones
- Selección de datos
- Vinculación de datos
- Transformación de datos
- Carga de datos

Las historias de usuarios se realizaron teniendo en cuenta validaciones y escenarios como se muestra en el Anexo A.

En las historias de usuario anteriormente presentadas muestran requerimientos y especificaciones determinadas para el sistema en base a esos y al uso de la técnica de mapeo de historias de usuarios, se procede a construir el plan de versiones con las actividades, tareas e interacciones.

Mapeo de historias de usuario

El mapeo de historias es una técnica defendida por Jeff Patton, proporciona una forma de concebir todo el producto o servicio como una serie de tareas que el usuario completa. implica la construcción de una cuadrícula de historias de usuario que se presentan bajo encabezados que representan la experiencia del usuario moviéndose a través de su producto.

Iteración 1 y 2

Fig. 4.7 Mapeo de historias de usuario iteración 1 y 2



Elaborado por: El Investigador

Iteración 3 y 4

Fig. 4.8 Mapeo de historias de usuario iteración 3 y 4



Elaborado por: El Investigador

Iteraciones

Para el desarrollo del sistema en base a las historias de usuarios y en reuniones con el representante de Solinfo se llegó al conceso de realizar cuatro iteraciones funcionales

que serán revisadas por el representante de Solinfo junto con el investigador para realizar la retroalimentación y preparar la siguiente iteración.

Programación en pareja

Para este caso la programación en pares se ajustó a la realidad de la empresa, con el apoyo del representante de Solinfo y del encargo del servidor el mismo que se encargará de la administración del sistema de transferencia de datos se acordó que aportarán con ideas y opiniones durante el tiempo de desarrollo del sistema a nivel de diseño y codificación.

Reuniones diarias

En este aspecto al establecer que el desarrollo del sistema de transferencia de datos se realizará en las instalaciones de Solinfo se concretó reuniones informales los días miércoles y jueves con el representante de Solinfo y los días lunes, miércoles y viernes con el encargado del servidor de la empresa con el fin de ajustarse a los horarios de la empresa y al mismo tiempo mantener la comunicación constante.

4.4.2 Fase: Diseño

Diseños simples

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

Simplicidad

Este aspecto se cumplió con el diseño de interfaces simple que presenta únicamente la información que requiere durante el proceso en el que se encuentra, durante el proceso de selección únicamente presenta el esquema y los datos de la base de datos origen.

Metáforas

Durante el proceso de transformación de datos, el sistema realiza internamente una comparación de los tipos y codificación de los datos entre el origen que pueden ser SQLServer, MySQL, Oracle, Postgresql o archivos de bases de datos no relaciones y el destino Oracle, para presentar al usuario a través de la interfaz la cantidad de caracteres de cada campo.

Refactorización

Para los diseños se establecieron aspectos generales para todo el sistema entre los cuales constan, que procesos similares mantengan una misma estética, que sea usable y mantenga una estética general, evitar colores o fondos muy llamativos, que los diseños no cambien drásticamente durante los procesos internos y que sean intuitivos para el usuario final.

Glosario de términos

Para este punto se especificó que se debe mantener una correcta nomenclatura y un código limpio para garantizar la reutilización del código, también para facilitar la compresión del código.

Riesgos

Con el apoyo del encargado del servidor de la empresa en las reuniones establecidas se procura cubrir los posibles riesgos en el diseño ya que se cuenta con el apoyo de la persona que será la encargada de administrar el sistema de transferencia de datos, la misma que revisará el diseño en las reuniones establecidas.

Funcionalidad extra

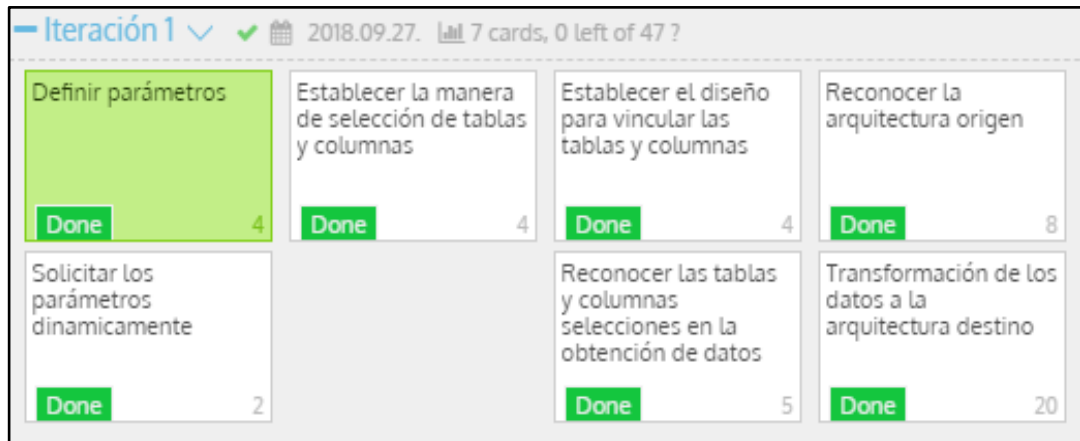
Las funcionales fueron establecidas desde la planificación y constan en las historias de usuario por lo mismo el desarrollo del sistema está basado en estos aspectos por lo que funcionalidades extras solo se contemplan con la garantía de reutilización de código, y establecer dichas funcionalidades extra como recomendaciones para futuras mejoras al sistema.

4.4.3 Fase: Codificación

Esta fase involucra la unión de todas las anteriores para realizar la codificación de acuerdo con las especificaciones, diseño, funcionalidades y características para el sistema de transferencia de datos, además se muestra partes de código como ejemplo de algunos métodos del sistema.

Iteración 1:

Fig. 4.9 Contenido de la iteración 1



Elaborado por: El Investigador

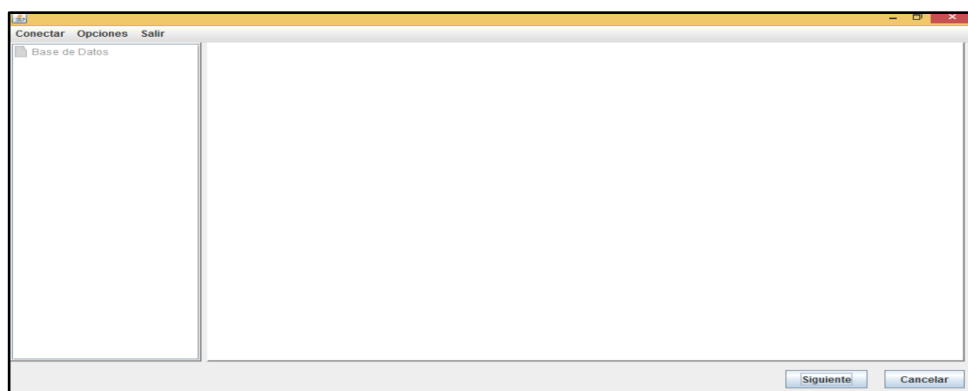
En esta iteración se presentó los diseños, conceptos y funcionalidad básica del sistema, como retroalimentación se obtuvo el ajuste del diseño de conexión principal y ajuste de parámetros de las bases que maneja el sistema.

Diseños de Interfaces

El diseño ofrece usabilidad en cada proceso que el sistema realiza.

Interfaz Principal

Fig. 4.10 Interfaz principal del sistema de transferencia

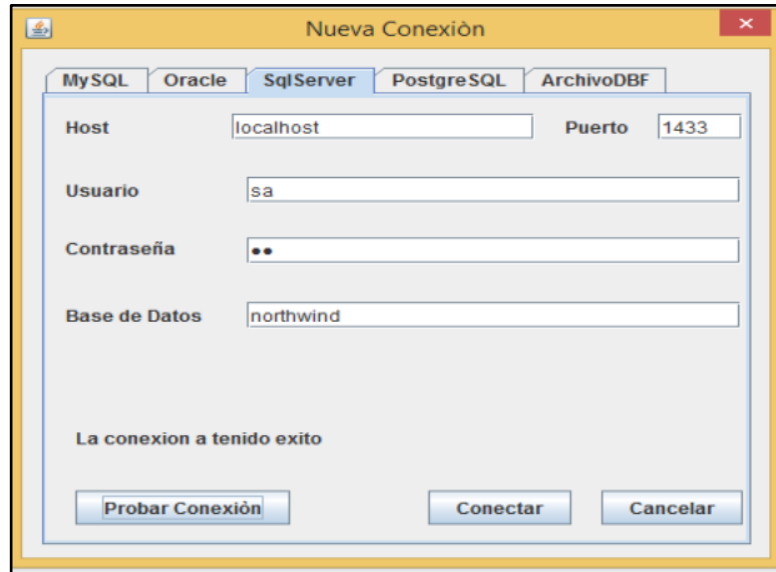


Elaborado por: El Investigador

Es la interfaz principal del sistema de transferencia de datos, la cual muestra un menú con las opciones principales y el diseño de la ventana principal que el sistema mantiene durante los procesos de obtención, transformación, carga de datos y procesos internos de vinculación y adaptación de datos.

Conexiones

Fig. 4.11 Interfaz de nueva conexión



Elaborado por: El Investigador

La interfaz de nueva conexión permite ingresar los parametros necesarios de acuerdo al escenario de la base de datos para establecer una conexión con dicha base de datos, en caso de ocurrir algun tipo de error presentara un mensaje con el posible problema.

La clase conexiones

Esta clase permite establecer la conexión con las bases de datos, reconociendo cual es la base de datos, de esta forma acorde al escenario presenta los parámetros necesarios y un control de errores para cada tipo de conexión.

```
1. Este método identifica la base de datos con la que se esté
   trabajando y dirigir al proceso adecuado para el proceso
2. public Connection
   conectar(String host, int puerto, String BaseDatos, String usu
   ario, String contra,String servidor, int op){
3.     String forName,lineaConexion="";
4.     Connection cn=null;
5.     //System.out.println("contraseña "+contra+" usuario
   "+usuario);
6.     switch (op){
7.         case 1: //Mysql
8.             forName = "com.mysql.jdbc.Driver";
9.             lineaConexion = "jdbc:mysql://" +host+": "+puert
   o+"/" +BaseDatos;
10.                break;
11.                //System.out.println("Conexion Oracle");
12.                case 2: //Oracle
```

```

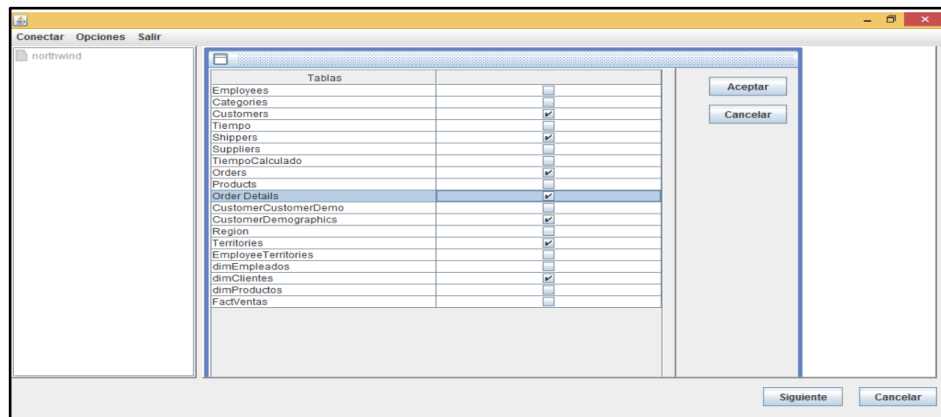
13.         forName = "oracle.jdbc.driver.OracleDriv
er";
14.         lineaConexion = "jdbc:oracle:thin:@"+hos
t+":"+puerto+": "+BaseDatos+"";
15.         //System.out.println("Conexion
SQLServer");
16.
17.         break;
18.         case 3: //SQLServer
19.             forName = "com.microsoft.sqlserver.jdbc.
SQLServerDriver";
20.             lineaConexion = "jdbc:sqlserver://" + host
+":"+puerto+":databaseName="+BaseDatos+";User =
"+usuario+";Password="+contra;
21.             break;
22.         case 4: //Postgresql
23.             forName = "org.postgresql.Driver";
24.             lineaConexion = "jdbc:postgresql://" + hos
t+":"+puerto+"/ "+BaseDatos+"";
25.             break;
26.         default:
27.             forName="";
28.             lineaConexion="";
29.             break;
30.     }
31.     try
32.     {
33.         Class.forName(forName);
34.         String cadenaConexion=lineaConexion;
35.         //System.out.println("linea
"+lineaConexion);
36.         cn=DriverManager.getConnection(lineaConexion
,usuario,contra);
37.     } catch (SQLException ex) {
38.         if (op==1)
39.             errores.GestionarMysql(ex);
40.         if (op==2)
41.             errores.GestionarOracle(ex);
42.         if (op==3)
43.             errores.GestionarSqlserver(ex);
44.         if (op==4)
45.             errores.GestionarPostgresql(ex);
46.         errores.mensaje();
47.     } catch (Exception ex) {
48.         errores.Gestionar(ex);
49.         errores.mensaje();
50.     }
51.     return cn;
52. }

```

Obtención de datos

Para la obtención de datos se cuenta con las siguientes interfaces la primera sirve para la selección de tablas de la base de datos origen.

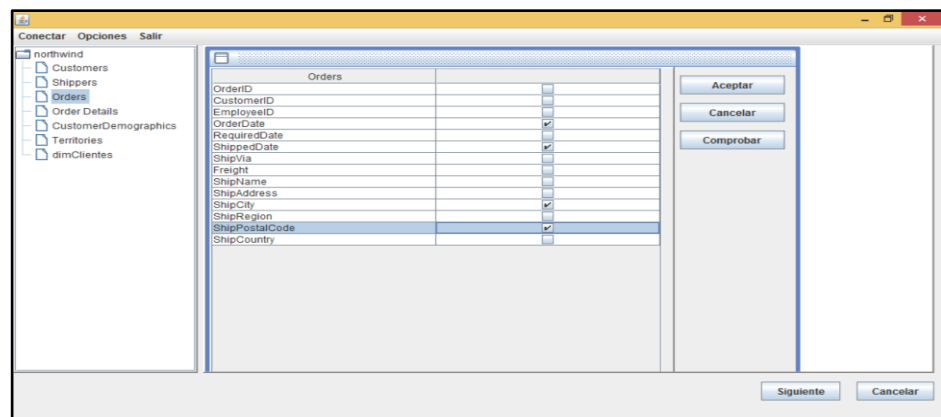
Fig. 4.12 Selección de tablas del origen



Elaborado por: El Investigador

En la siguiente figura se muestra como se realiza la selección de columnas de las tablas anteriormente seleccionadas que contiene la base de datos de origen.

Fig. 4.13 Selección de columnas del origen



Elaborado por: El Investigador

Vinculación de datos

Para el proceso de vincular los datos del origen con la base de datos del destino, primero se debe realizar una conexión al destino.

En la figura 4.23 se presenta la interfaz que permite la conexión a Oracle al ser el único destino, aunque se estable de forma dinámica por lo mismo puede ser adaptado para más bases de datos destino.

Fig. 4.14 Conexión destino

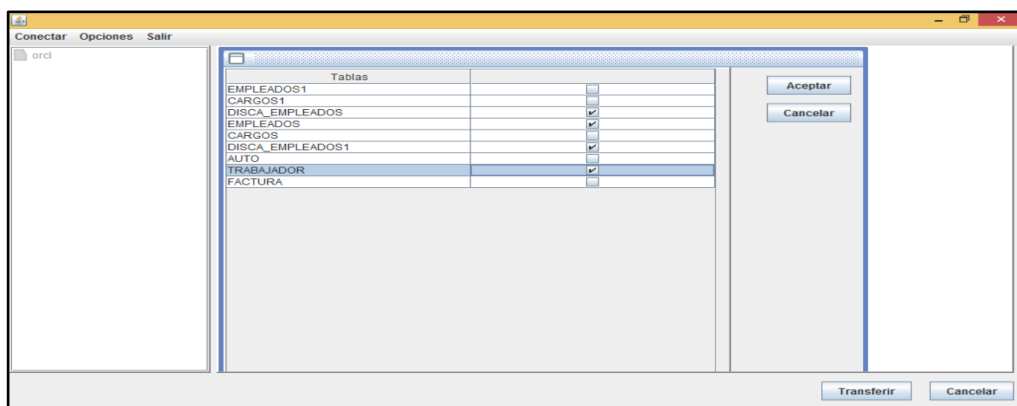


Elaborado por: El Investigador

Al realizar la conexión con éxito se repite el proceso de selección de tablas y columnas esta vez con la base de datos del destino.

La selección de tablas es idéntica al proceso con la base origen.

Fig. 4.15 Selección tablas del destino

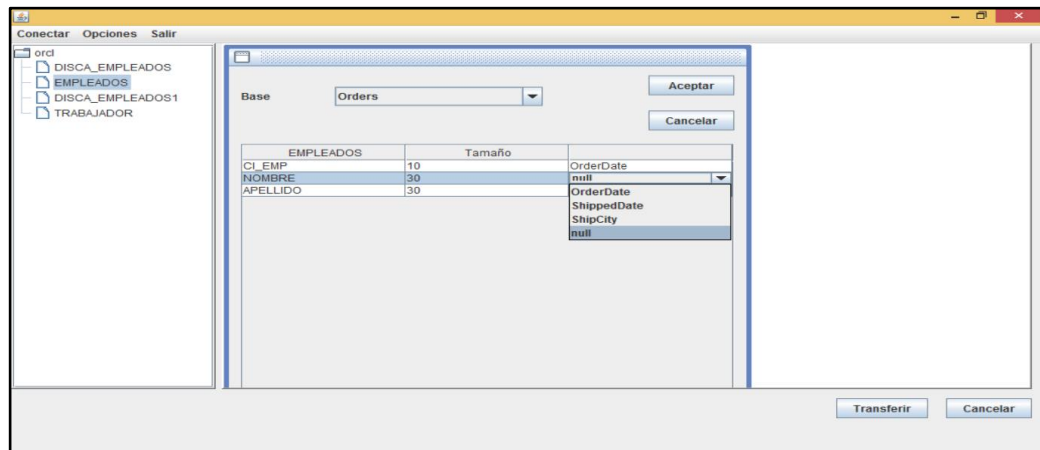


Elaborado por: El Investigador

Sin embargo, la vinculación de las columnas tiene una variante al escoger la tabla destino se presenta la interfaz con la tabla y columnas del origen para seleccionar y vincular con las columnas del destino.

La figura 4.25 muestra la tabla destino con sus columnas correspondientes y la tabla origen con sus columnas, para vincular los datos entre el origen y destino, este proceso es idéntico para cada tabla del destino que previamente fue seleccionada.

Fig. 4.16 Vincular columnas origen con destino



Elaborado por: El Investigador

Transformación de datos

El proceso de transformación de datos se realiza a la par que el proceso de vinculación de datos para esto se establece una estructura de listas que almacenan los valores de los datos para la comparación de las estructuras seleccionadas, con los valores de la base de datos para establecer las equivalencias entre los esquemas de bases de datos, este proceso se ejecuta al realizar la transferencia.

Carga de datos

El proceso de carga de datos requiere ejecutar los procesos anteriores en cadena para establecer si existen problemas en la adaptación de los datos, esto proceso evita que se realice una transferencia incompleta de los datos.

```

1. Este método permite identificar con que bases de datos se está
   trabajando y dirige el proceso acorde a la base que
   identifique.
2. void transferencia(int x){
3.     ArrayList<String> columinsert = new ArrayList<String>(
   );
4.     SelecEspecifico(ptablacolumnas.get(x));
5.     String sql="", nomCol="", values="";
6.     int n =0;
7.     System.out.println("columnasinsertar.size()+"columnas
   insertar.size());
8.     for(int j=1;j<columnasinsertar.get(x).length;j++){
9.         System.out.println("columnasinsertar.get(j)
   "+j+" => "+columnasinsertar.get(x)[j]+" tam
   "+columnasinsertar.size());
10.        sql="insert into "+columnasinsertar.get(x)[0]+"
   ("+nomCol+") values ("+values+")";

```

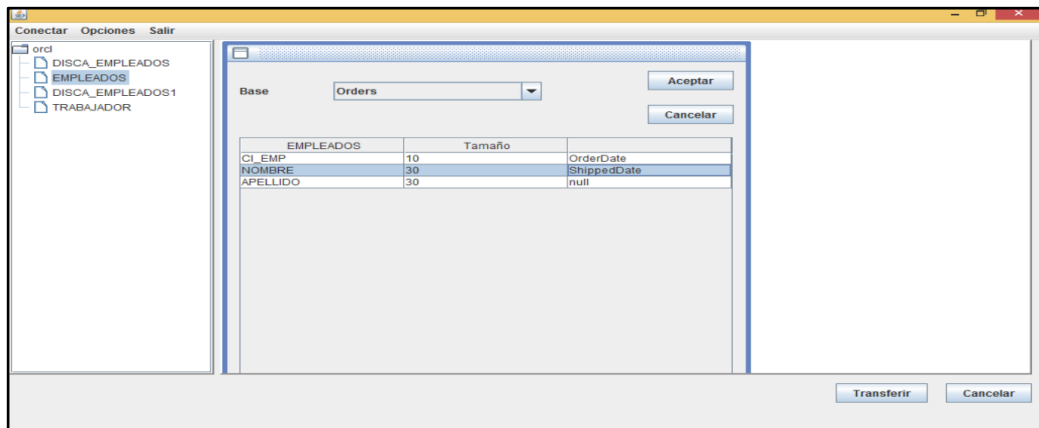
```

11.         //          System.out.println("");
12.         try{
13.             PreparedStatement
14.             psd = cn.prepareStatement(sql);
15.             //System.out.println("resultadosdatos.size()
16.             "+resultadosdatos.size()+" ptablacolumnas
17.             "+ptablacolumnas.size());
18.             //          for(int z=0;z<ptablacolumnas.size();z++){
19.                 for(int y=0;y<resultadosdatos.size();y++){
20.                     //System.out.println("x "+x+" y "+y+" if
21.                     "+resultadosdatos.get(y)[0].toString()+" ==
22.                     "+ptablacolumnas.get(x)[1]+" len
23.                     "+resultadosdatos.get(y).length);
24.                     if(resultadosdatos.get(y)[0].toString()
25.                     .equals(ptablacolumnas.get(x)[1])){
26.                         for(int j=1;j<resultadosdatos.get(y)
27.                         ).length ;j++){
28.                             //resultadosdatos.remove(y);
29.                             //if(ptablacolumnas.get(y)[j].eq
30.                             uals(resultadosdatos.get(y)[j])){
31.                                 psd.setString(j,
32.                                 resultadosdatos.get(y)[j].toString());
33.                                 //System.out.println("x "+x+" y
34.                                 "+y+" j "+j+" valor "+resultadosdatos.get(y)[j].toString()+"
35.                                 tamlen "+resultadosdatos.get(y).length);
36.                                 //}
37.                             }
38.                             n=psd.executeUpdate();
39.                         }
40.                     }
41.                 //          }
42.                 //          psd.executeUpdate();
43.                 if(n>0){
44.                     controlTransferencia++;
45.                     //JOptionPane.showMessageDialog(null,
46.                     "Transferencia realizada correctamente");
47.                 //          CargarTabla(servidor1);
48.                 }
49.                 } catch (SQLException ex){
50.                     //JOptionPane.showMessageDialog(null,"SQLExc
51.                     eption tranferir "+ex);
52.                     errores.GestionarOracle(ex);
53.                     errores.mensaje();
54.                 } catch (Exception ex){
55.                     //JOptionPane.showMessageDialog(null,"Except
56.                     ion tranferir "+ex);
57.                     errores.Gestionar(ex);
58.                     errores.mensaje();
59.                 }
60.             }
61.         }

```

El sistema al terminar con los procesos de selección y transformación de datos, permite realizar la transferencia con el botón “Transferir” el mismo que en los procesos anteriores mantenía el nombre de “Siguiente”.

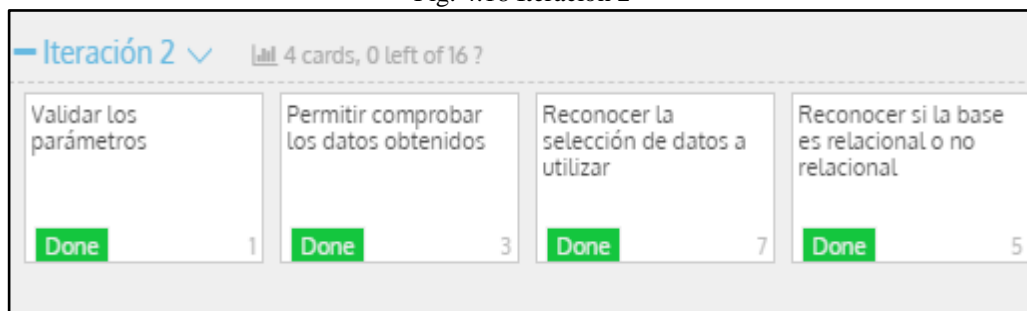
Fig. 4.17 Interfaz de transferir



Elaborado por: El Investigador

Iteración 2:

Fig. 4.18 Iteración 2



Elaborado por: El Investigador

En la iteración 2 se estableció controles de validación, función de comprobar datos, la cual permitir que el sistema reconozca los datos seleccionados y pueda ser visualizado por el usuario si así lo cree necesario y el tipo de base si es relacional o no relacional.

Validación de parámetros para la conexión al origen y destino para conseguir validar los parámetros se crearon tablas con los errores que se presentaron con la finalidad de garantizar un mensaje claro del problema y posible solución a este, fue necesario crear una tabla para cada base de datos ya que cada base utiliza un identificador diferente para cada error.

A continuación, se muestran los problemas presentados en cada base de datos.

Problemas en Oracle

Tabla 4.26 Problemas en Oracle

Número de error	Descripción	Mensaje Personalizado	Posible Solución
17002	Got minus one from a read call	El host o el Puerto son incorrectos	Verifique que el nombre del host y del puerto estén correctos
12505	TNS; listener does not currently know of SID given in connect descriptor	El nombre del servidor SID es incorrecto o el listener no está escuchando	Verifique el nombre del servidor SID este correcto o que el listener este escuchando
1017	Invalid username/password; login denied	El usuario o contraseña es incorrecto	Verifique que el nombre del usuario y contraseña estén correctos
12899	Value too large for column	El valor es demasiado grande para la columna	Modifique el tamaño del campo o asigne otro dato al campo específico

Elaborado por: Investigador

Problemas en MySQL

Tabla 4.27 Problemas en MySQL

Número de error	Descripción	Mensaje Personalizado	Posible Solución
0	Communication link failure	El host o el puerto son incorrectos	Verifique que el nombre del host y del puerto estén correctos
1049	Unknown database	El nombre de la base de datos esta incorrecto o no tiene permisos sobre la base	Verifique el nombre de la base, verifique que la base de datos existe
1045	Access denied for user	El usuario o contraseña es incorrecto	Verifique que el nombre del usuario y contraseña estén correctos

Elaborado por: Investigador

Problemas en SqlServer

Tabla 4.28 Problemas en SQLServer

Número de error	Descripción	Mensaje Personalizado	Posible Solución
0	No se puede realizar la conexión TCP/IP al host "Nombre del host", puerto "número del puerto"	El host o puerto son incorrectos	Verifique que el nombre del host y del puerto estén correctos
18456	Unknown database	El nombre de la base de datos esta incorrecto o no tiene permisos sobre la base	Verifique el nombre de la base, verifique que la base de datos existe
4060	No se puede abrir la base de datos "nombre de la base" solicitada con el inicio de sesión	El usuario o contraseña están incorrectos	Verifique que el nombre del usuario y contraseña estén correctos
207	Invalid column name	Nombre de la columna esta repetida	Verifique que la misma columna no esté en diferentes campos a transferir

Elaborado por: Investigador

Problemas en PostgreSQL

Tabla 4.29 Problemas en PostgreSQL

Número de error	Descripción	Mensaje Personalizado	Posible Solución
0	El intento de conexión fallo	Los datos ingresados están incorrectos	Verifique que los parámetros están correctos
0	No existe la base de datos "nombre de la base"	Los datos ingresados están incorrectos	Verifique que los parámetros estén correctos
0	La autenticación password fallo para el usuario "nombre del usuario"	Los datos ingresados están incorrectos	Verifique que los parámetros estén correctos

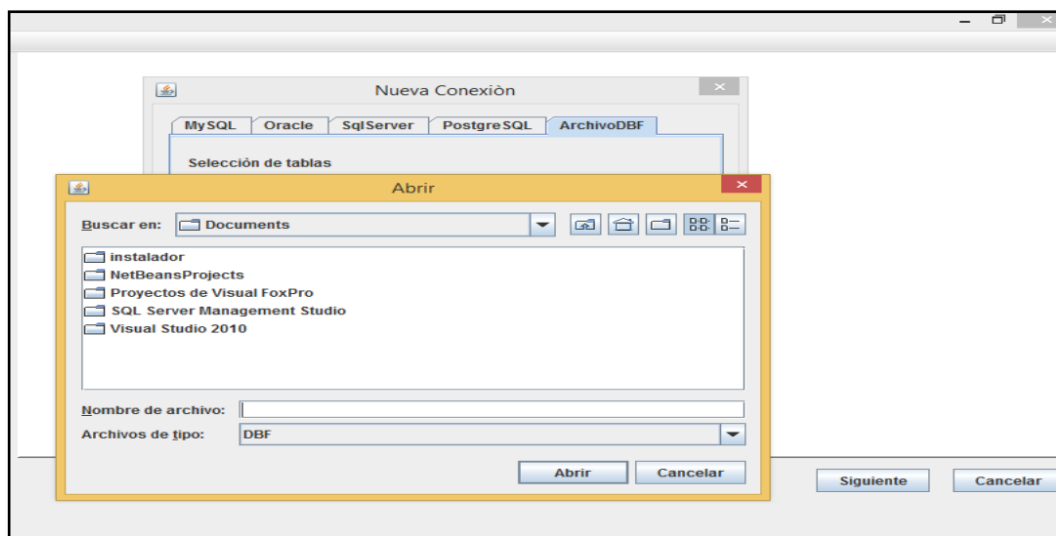
Elaborado por: Investigador

En el caso particular de PostgreSQL todos los mensajes de error propios de la base de datos, utilizan el mismo número de error por lo que una distinción del mensaje no se

pudo establecer, por este motivo se estableció un mensaje general para los errores que se presenten al momento de conectarse a esta base de datos.

Para el caso de las bases de datos no relacionales no se presenta los problemas de conexión debido a que la conexión se establece directamente con los archivos creados por este tipo de bases de datos, de esta manera el usuario no ingresa parámetros para la conexión en su lugar debe buscar y abrir los archivos que necesite, este proceso se realiza desde el mismo sistema siendo una variante al proceso de obtención de datos que se realiza en las bases de datos relacionales.

Fig. 4.19 Conexión a bases de datos no relacionales

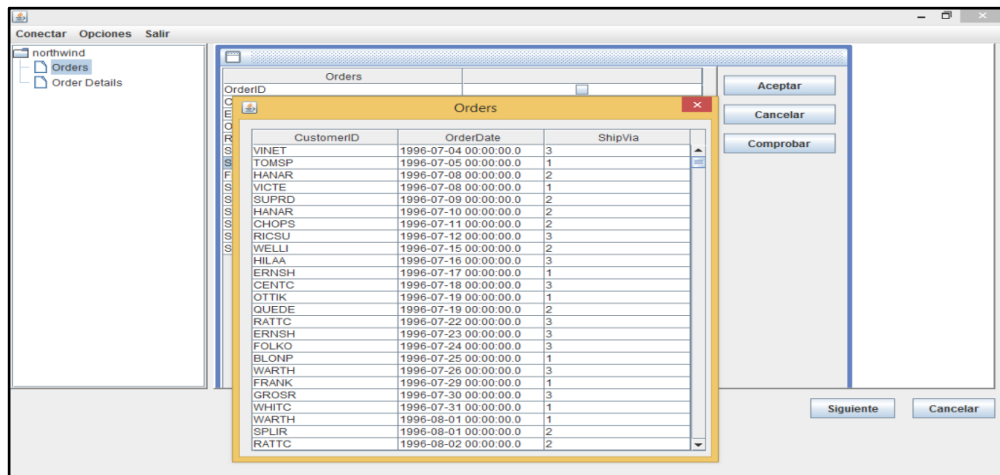


Elaborado por: El Investigador

Para permitir comprobar los datos seleccionados se cuenta con el botón “Comprobar” el mismo que es visible únicamente en el proceso de selección de datos de la base de datos origen, con el fin de evitar un error por parte del usuario al intentar utilizar esta función durante un proceso que no sea posible.

La funcionalidad del botón “Comprobar” es visualizar en una ventana emergente los campos seleccionados por el usuario para garantizar si son los datos que desea transferir, esta funcionalidad únicamente visualiza los 10 primeros campos seleccionados con el fin de no saturar la base con una consulta innecesaria de todos los datos almacenados.

Fig. 4.20 Comprobación de datos



Elaborado por: El Investigador

Para reconocer la selección de datos a utilizar por el proceso de vincular los datos se creó una variable con el nombre “repetidos” que se encarga de recorrer la estructura `ArrayList<String[]>` para comprobar que no existan variables repetidas y en el caso de existir las remueva.

```

1. Este método acorde al caso que identifica recorre las columnas
   seleccionadas y verifica si existe valores repetidos de
   encontrarlos los elimina.
2. public void seleccionTablasColumnas(int controltablasColumnas)
   {
3.     int controlseleccioncolumnas=0;
4.     switch(controltablasColumnas){
5.         case 1:
6.             //System.out.println("case 1 aceptar intra");
7.             if(jtContenidoBase.getRowCount()<=1){
8.                 if(columnas.size()>0){
9.                     jtContenidoBase.setEnabled(true);
10.                    DefaultMutableTreeNode
   raiz= new DefaultMutableTreeNode(b);
11.                    for(int x=0;x<columnas.size();x
   ++) {
12.                        DefaultMutableTreeNode
   nodo= new DefaultMutableTreeNode();
13.                        nodo.setUserObject(columna
   s.get(x));
14.                        raiz.add(nodo);
15.                    }
16.                    DefaultTreeModel
   dt= new DefaultTreeModel(raiz);
17.                    jtContenidoBase.setModel(dt);
18.                    lblMensaje.setText("Eliga las
   columnas de cada tabla a transferir");
19.                    this.dispose();
20.                }else{

```

```

21.                                     if(controlrepetido==1){
22.                                         Principal.ptablacolumnas.
remove(valorborrar);
23.                                         Principal.ptablacolumnas.
add(tablacolumnas(jtContenidoBase.getLastSelectedPathComponent().toString()));
24.                                     }else{
25.                                         if(columnas.size()>0)
26.                                             Principal.ptablacolumnas.add(tablacolumnas(jtContenidoBase.getLastSelectedPathComponent().toString()));
27.                                     }else{
28.                                         JOptionPane.showMessageDialog(this, "Debe elegir al menos una columna");
29.                                         controlseleccioncolumnas++;
30.                                     }
31.                                     }

```

El proceso de transformación de los datos al esquema de la base de datos destino, se lleva a cabo con el método “transferir”, el mismo que hace uso de otros métodos para reconocer los tipos de datos que fueron seleccionados por el usuario y los transforma para poder transferirlos al destino.

```

1. El método realiza todos los procesos anteriores para llevar a
cabo la transferencia con la aplicación de métodos anidados y
verificación de los datos y su estructura utilizando
comparación de valores.
2.
3. void transferencia(int x){
4.     ArrayList<String> columinsert = new ArrayList<String>(
);
5.     SelecEspecifico(ptablacolumnas.get(x));
6.     String sql="", nomCol="", values="";
7.     int n =0;
8.     System.out.println("columnasinsertar.size()+"columnas
insertar.size());
9.     for(int j=1;j<columnasinsertar.get(x).length;j++){
10.         System.out.println("columnasinsertar.get
(j) "+j+" => "+columnasinsertar.get(x)[j]+" tam
"+columnasinsertar.size());
11.         if(j<columnasinsertar.size()||columnasins
ertar.size()==1){
12.             nomCol = nomCol+ columnasinsertar.get
(x)[j]+", ";
13.             values=values+"?, ";
14.         }
15.         else{
16.             nomCol=nomCol+columnasinsertar.get(x
)[j];
17.             values=values+"?";
18.         }

```

```

19.         }
20.         try{
21.             PreparedStatement
psd = cn.prepareStatement(sql);
22.             //System.out.println("resultadosdatos.size()
"+resultadosdatos.size()+" ptablacolumnas
"+ptablacolumnas.size());
23.             //         for(int z=0;z<ptablacolumnas.size();z++){
24.                 for(int y=0;y<resultadosdatos.size();y++){
25.                     //System.out.println("x "+x+" y "+y+" if
"+resultadosdatos.get(y)[0].toString()+" ==
"+ptablacolumnas.get(x)[1]+" len
"+resultadosdatos.get(y).length);
26.                 } catch (SQLException ex){
27.                     //JOptionPane.showMessageDialog(null,"SQLExc
eption transferir "+ex);
28.                     errores.GestionarOracle(ex);
29.                     errores.mensaje();
30.                 } catch (Exception ex){
31.                     errores.Gestionar(ex);
32.                     errores.mensaje();
33.                 }
34.             }

```

El sistema reconoce si se está trabajando con una base de datos relacional o no relacional gracias a la variable “ControlDBF” que identifica el tipo de base de datos, esto se debe a que cada tipo de base de datos tiene sus propios métodos, es fácilmente identificable los pasos que sigue cada tipo de base de datos gracias a la nomenclatura presente que diferencia los métodos de las base de datos relacionales y no relacionales con las letras DBF (es la extensión que contiene los archivos que generan las bases no relaciones), así los nombres de los métodos para las bases de datos no relacionales son los que contienen este acrónimo al final.

Como se mencionó para las bases de datos no relacionales se establece una conexión directamente con los archivos que contienen los datos, para este proceso se utiliza un método de lectura de archivos que se presenta a continuación.

```

1. Este método se utiliza en caso de ser una base de datos no
relacional para realizar la lectura de los datos.
2. public void lectura() throws IOException {
3.     File fichero=null;
4.     DefaultListModel model = new DefaultListModel<>();
5.     JFileChooser se = new JFileChooser();
6.     se.setMultiSelectionEnabled(true);
7.     FileNameExtensionFilter
filtroImagen=new FileNameExtensionFilter("DBF","dbf");
8.     se.setFileFilter(filtroImagen);
9.     se.setSelectionMode(JFileChooser.FILES_ONLY);
10.     int estado = se.showOpenDialog(null);

```

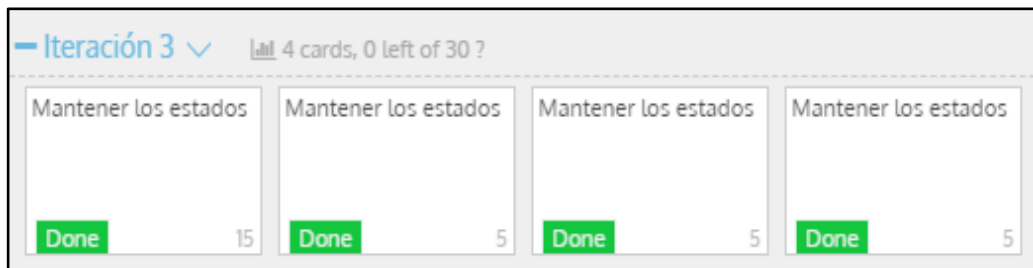
```

11.         if (estado == JFileChooser.APPROVE_OPTION) {
12.             for(File item : se.getSelectedFiles()){
13.                 String [] valores = new String[2];
14.                 valores[0]=item.getName().toString().sub
string(0, item.getName().toString().length()-4);
15.                 valores[1]=item.getAbsolutePath().toStri
ng();
16.                 model.addElement(item.getName().toString
().substring(0, item.getName().toString().length()-4));
17.                 pathDBF.add(valores);
18.             }
19.         }
20.         jListArchivosDBF.setModel(model);
21.     }
22.     for(File item : se.getSelectedFiles()){
23.         String [] valores = new String[2];
24.         valores[0]=item.getName().toString().sub
string(0, item.getName().toString().length()-4);
25.         valores[1]=item.getAbsolutePath().toStri
ng();
26.         model.addElement(item.getName().toString
().substring(0, item.getName().toString().length()-4));
27.         pathDBF.add(valores);
28.     }

```

Iteración 3:

Fig. 4.21 Iteración 3



Elaborado por: El Investigador

En la iteración 3 se crearon métodos que permitan mantener los estados en cada uno de los procesos de la transferencia de datos con esto se proporciona la funcionalidad de modificar una selección ya sea por ser incorrecta o incompleta, así el usuario puede regresar al paso anterior y modificar su selección en el caso de requerirlo sin tener que volver a ejecutar todo el proceso nuevamente.

Para esto se amplió la funcionalidad de la variable con el nombre “repetidos” mencionada anteriormente, que se encarga de recorrer la estructura `ArrayList<String[]>` para comprobar que no existan variables repetidas y en el caso de existir las remueva, ahora cuenta con un ciclo que le permite recorrer todas las

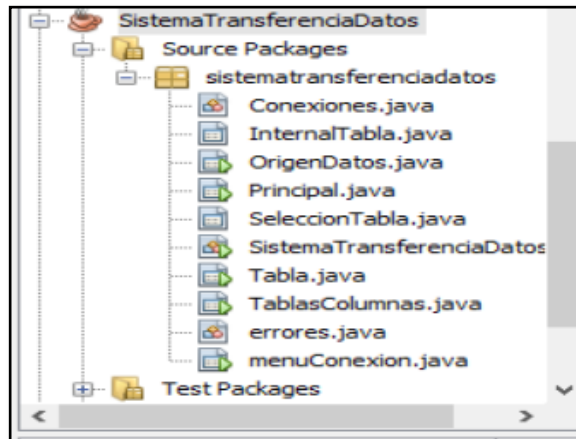
estructuras utilizadas en las diferentes interfaces y clases para garantizar que al necesitar modificar una selección previa el usuario visualice su selección anterior y la modifique guardando los cambios realizados.

A continuación, algunas de las estructuras y clases que recorre la variable para mantener los estados.

```
1. Las variables se utilizan acrónimos con la finalidad
   simplificar la nomenclatura.
2.  h = host, b = base, u = usuario, p = puerto
3. String h,b,u,con,servidor="";
4.     int p,tipobase=0,controlTransferencia=0;
5.
6. Variables staticas utilizadas en diferentes clases
7. Con la finalidad de mantener estados y reconocer valores de
   las bases de datos.
8. Variable para distinguir el tipo de base de datos que se
   utiliza dentro del sistema
9.     static int destino=0;
10.         static int controlDBF=1, controlDBF1=1;
11.         ArrayList<String> tablas = new ArrayList<String>();
12.
13.         //Valores de la tabla y columnas de la ventana
   InternalTabla
14.         Variables staticas que se mantiene en su clase pero que
   son cambiantes en las otras clases que llevan a cabo los
   diferentes procesos
15.         Variable de control para las columnas
16.         static ArrayList<String[]> columinsert = new ArrayL
ist<String[]>();
17.         Variable de control para las tablas de las columnas
18.         static ArrayList<String[]> ptablacolumnas = new Arr
ayList<String[]>();
19.         Variable de control para los estados de las columnas
20.         static ArrayList<String[]> columnasinsertar = new A
rrayList<String[]>();
21.         Variable de control para comprobar los datos a traferir
22.         static ArrayList<String[]> datosTrasferir = new Arr
ayList<String[]>();
23.         Variable de control para la ubicación de los archivos
   DBF de las bases de datos no relacionales
24.         static ArrayList<String[]> pathDBF = new ArrayList<
String[]>();
25.         //Almacena los resultados de las consultas
26.         static ArrayList<String[]> resultadosdatos = new Ar
rrayList<String[]>();
```

Las clases para el sistema de transferencia de datos se crearon acorde a la funcionalidad que estas aportan así cada una intercambia la información requerida por los procesos que se llevan a cabo por el sistema.

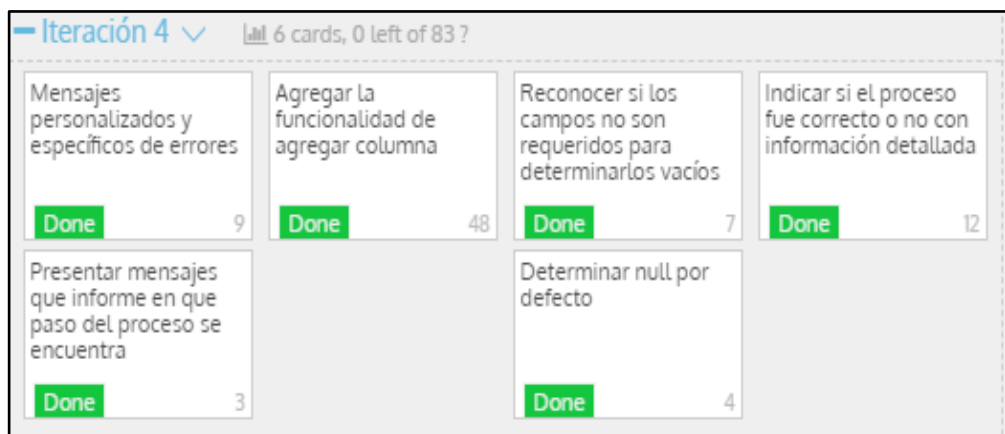
Fig. 4.22 Clases del sistema



Elaborado por: El Investigador

Iteración 4:

Fig. 4.23 Iteración 4

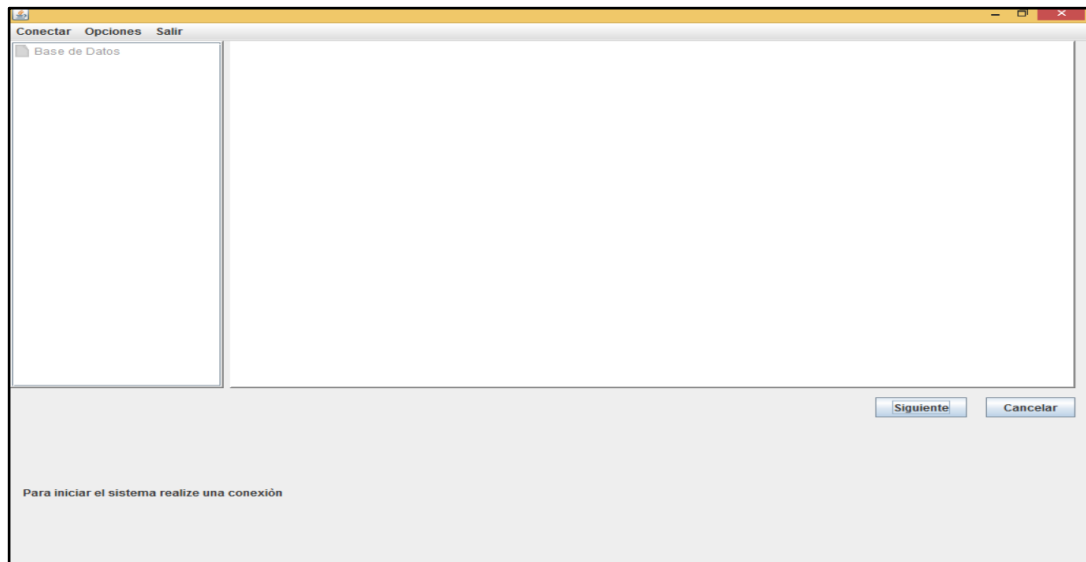


Elaborado por: El Investigador

En la iteración 4 es necesario realizar una leve modificación al diseño para contar con un espacio destinado a mostrar mensajes que brindan ayuda al usuario a comprender en que paso del proceso se encuentra y que debe hacer el usuario, los estados de los procesos determinan valores por defecto en determinados casos para evitar error por parte del usuario.

Para ayudar al usuario a guiarse existen mensajes de ayuda ubicados en la parte inferior de la interfaz, los mensajes van cambiando durante los procesos para ayudar como guía al usuario.

Fig. 4.24 Interfaz principal mensajes de ayuda



Elaborado por: El Investigador

En base a los tipos de errores generados y documentados se creó una clase de errores que contiene dichos errores y como solucionarlos, por lo general son errores por parte del usuario al momento de ingresar los parámetros de conexión e intentar establecer la conexión.

```
1. La clase de errores se encarga de dirigir y gestionar un error
   ocurrido de acuerdo a la base de datos y el identificador del
   error.
2. public class errores {
3. public static void GestionarOracle (SQLException exception){
4.     String saltoLinea = "\n";
5.     //Mensajes personalizados para el usuario
6.     String problema = "EL PROBLEMA GENERADO PUEDE
   DEBERSE A LOS SIGUIENTES FACTORES:";
7.     String solucion = "POR FAVOR, PRUEBE LAS
   SIGUIENTES SOLUCIONES";
8.     String mensajeFinal = "NOTA: En caso de persistir
   el problema, comuníquese con Soporte Técnico \n      o con el
   Administrador del Sistema";
9.     //contiene la cadena con el mensaje de error a
   mostrar al usuario
10.     String mensaje = null;
11.     // Verificamos el numero de error y
   personalizar el mensaje.
12.     switch (exception.getErrorCode())
13.     {
14.         case 1017:
15.             // //Error personalizado
16.             mensaje = problema +
17.                 saltoLinea +
18.                 "1.- El usuario es incorrecto" +
19.                 saltoLinea +
```

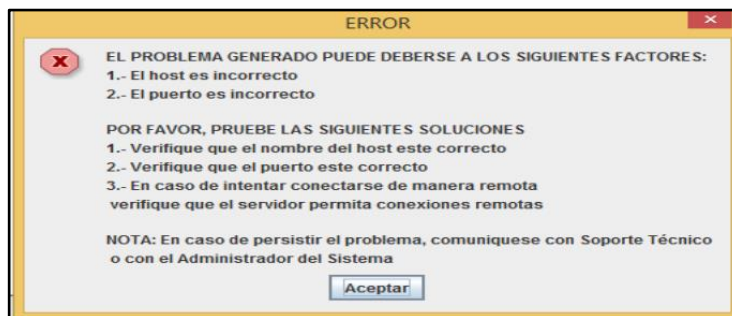
```

20.         incorrecta" +
21.         saltoLinea +
22.         saltoLinea +
23.         solucion +
24.         saltoLinea +
25.         "1.- Verifique que el nombre del
26.         usuario este correcto" +
27.         saltoLinea +
28.         "2.- Verifique que la contraseña
29.         este correcta" +
30.         saltoLinea +
31.         saltoLinea +
32.         mensajeFinal;
33.     break;

```

Esta clase genera los mensajes personalizados de acuerdo al error que se presente con el siguiente formato determinado.

Fig. 4.25 Mensaje personalizado



Elaborado por: El Investigador

El mensaje determina si fue posible o no la conexión utilizando el botón “Probar Conexión”, los mensajes están estandarizados, es decir muestra la información acorde al escenario que se presente.

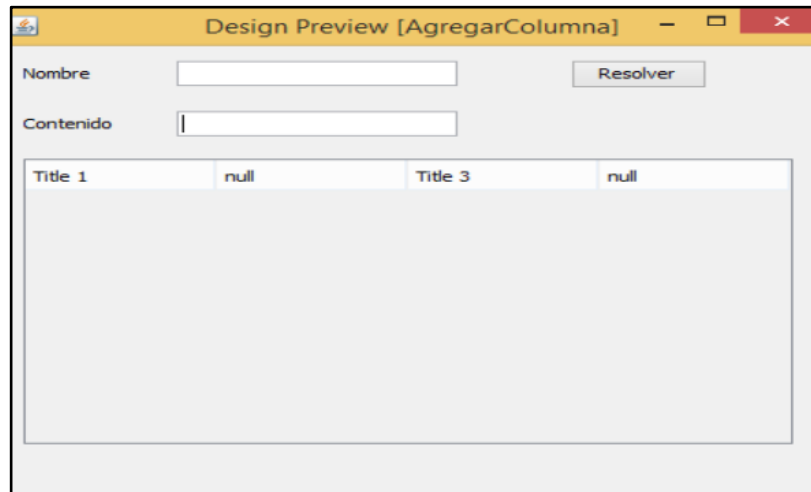
Fig. 4.26 Comprobación de conexión



Elaborado por: El Investigador

Finalmente, la funcionalidad agregar columna que permite al usuario insertar una nueva columna inexistente en la base de datos origen pero que es necesaria en la base de datos destino sin que esto afecte de ningún modo a los datos o esquema de la base de datos origen, esto con el fin de insertar una columna necesario en el destino y evitar hacerlo manualmente para esto se cuenta con la siguiente interfaz.

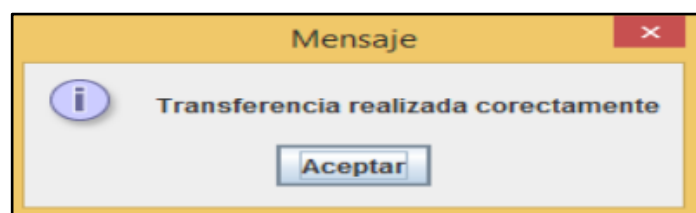
Fig. 4.27 Agregar columna



Elaborado por: El Investigador

Esta interfaz permite crear la nueva columna y ubicar restricciones basadas en los valores de la base de datos origen, así se puede asegurar que el contenido de la nueva columna será ingresada con determinadas condiciones, es decir si requiero ingresar una columna de activos pero solo a las personas registradas después del 2018 es posible siempre y cuando en la base de datos origen exista una columna con la fecha de ingreso de la persona, de esta manera el sistema cuenta con un generador de columnas que se ingresa automáticamente para su vinculación y con los datos que el usuario requiere. Si la transferencia es finalizada correctamente, sin ningún tipo de problema el sistema presenta el siguiente mensaje y se restablece.

Fig. 4.28 Fin del proceso



Elaborado por: El Investigador

En el caso de presentar algún problema mostrará un mensaje personalizado indicando el problema, su posible solución y regresará a su último estado consistente del cual después de solucionar el problema puede volverse a intentar transferir los datos.

4.4.4 Fase: Pruebas

Las fases de pruebas se llevaron a cabo con cada una de las iteraciones y con la presencia del representante de Solinfo el mismo que realizaba la retroalimentación necesaria que fue reflejada en las iteraciones siguientes, además se tomó en cuenta las observaciones y opiniones del encargado del servidor de la empresa. Estas pruebas fueron llevadas a cabo en la empresa, pero bajo la infraestructura del investigador esto durante las tres iteraciones, posterior a esto las pruebas se realizaron bajo la infraestructura propia de la empresa.

Las bases de datos utilizadas son:

SQLServer

- AdventureWorksDW, Northwind, pubs

PostgreSQL

- Practica, postgres

MySQL

- farmacia, mysql, productos

Oracle

- orcl, farmacia

4.5 Análisis de resultados

Pruebas de la iteración 1

Durante las pruebas de la iteración 1 en presencia del representante de Solinfo se estableció la necesidad de mantener los estados durante los diferentes procesos que lleva a cabo el sistema, un registro con los errores más comunes que se pueden

presentar a modo de FAQ (Frequently Asked Questions / Preguntas frecuentes), sin ninguna observación sobre los diseños de las interfaces.

Esta prueba presento el diseño de las interfaces y la interacción entre ellas para demostrar cómo se enlazan los procesos del sistema de transferencia de datos.

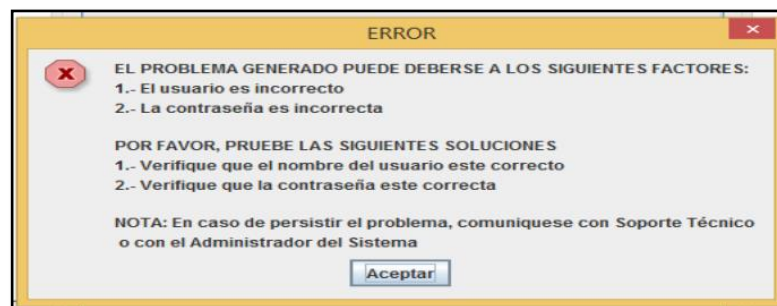
Pruebas de la iteración 2

Como retroalimentación en estas pruebas el representante de Solinfo realizó la observación que se debe pulir el proceso que mantiene los estados en cada uno de los procesos para permitir al usuario regresar a un punto anterior sin tener la necesidad de realizar todo el proceso nuevamente.

Las observaciones que se realizaron en las pruebas de la iteración 1 se presentaron en estas pruebas de la siguiente manera

Para el registro de errores la siguiente interfaz muestra la información con la misma plantilla que exhibe el problema y las posibles soluciones de acuerdo al caso que se presente, la información se obtiene del FAQ que se encuentra en la clase de errores.

Fig. 4.29 Mensaje personalizado



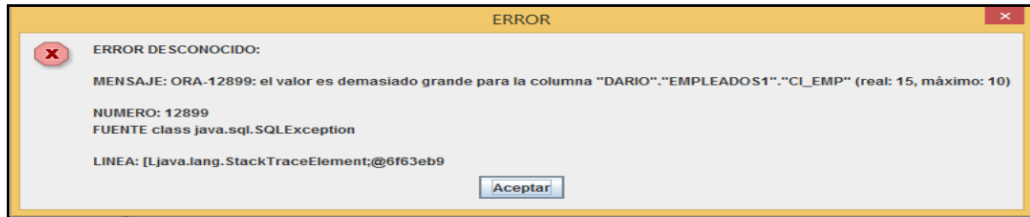
Elaborado por: El Investigador

Pruebas de la iteración 3

Las pruebas presentaron errores regulares durante la transferencia de datos sobretodo en los conjuntos de caracteres el cual es el error más recurrente en todas las bases de datos, este problema se presenta en el momento de vincular las columnas, durante este proceso se puede cometer errores por parte del usuario al vincular una columna con un espacio de 10 caracteres a otra con un espacio de 5 caracteres, siendo la proporción de un espacio mayor a uno menor, este problema se presenta sin importar los tipos de

datos, para esto se acordó agregar una columna con el tamaño de cada una de las columnas a vincular.

Fig. 4.30 Error por el conjunto de caracteres



Elaborado por: El Investigador

Pruebas de la iteración 4

Estas pruebas se llevaron a cabo bajo la infraestructura de la empresa Solinfo implementando el sistema para llevar a cabo dichas pruebas, se observó la necesidad de pulir el contenido de los mensajes de ayuda.

En estas pruebas se planteó la necesidad de realizar cambios al diseño con la finalidad de añadir un espacio que contenga mensajes de ayuda al usuario de esta manera permita guiar al usuario sobre los diferentes procesos que lleva a cabo el sistema, entonces se reajustaron los componentes de la interfaz principal ya que es el lugar idóneo para añadir los mensajes de ayuda porque esta interfaz se mantiene visible durante todos los procesos del sistema.

Al realizar las pruebas al sistema final se encontraron los siguientes resultados:

- Conexión a las diferentes bases de datos relaciones y no relaciones

Al realizar las conexiones con las diferentes bases de datos relacionales de forma local y de forma remota con los servidores de la empresa Solinfo no se encontró ningún tipo de inconvenientes ya que las conexiones fueron exitosas, además se comprobó el correcto funcionamiento de los mensajes personalizados, forzando errores al ingresar mal los diferentes parámetros de conexión se obtuvo los mensajes correctos acorde a los errores, excepto en la conexión de Oracle en el cual los mensajes no estaban acorde a los errores este problema se debió a la asignación incorrecta de los números de errores mismos que fueron corregidos, el tiempo para realizar una conexión es aceptable considerando que existen otros aspectos fuera del sistema que pueden influir en el cálculo como es el porcentaje de uso que mantiene.

Mientras que la conexión a una base de datos no relacional se realiza directamente con el archivo físico al leerlo internamente, así que el tiempo depende exclusivamente de las capacidades del servidor, el inconveniente existente con una base no relacional es que al momento de realizar la lectura de los archivos cualquier cambio durante el tiempo que tarda el sistema en completar sus procesos no son reflejados en la transferencia, es decir si un cambio en la base no relacional se dio durante el proceso del sistema de transferencia de datos este cambio no se reflejara en la transferencia, esto ocurre porque no se establece una conexión constante con la base de datos no relacional, lo cual no ocurre con el proceso si se trata de una base de datos relacional.

- Transferencia de datos

Al transferir los datos de una base a Oracle en el caso de una base de datos relacional se requiere mantener una conexión con el origen hasta el final del proceso para garantizar que los cambios que se realizaron durante el proceso de selección de ser el caso se reflejen en la transferencia, del número de pruebas realizadas con diferentes bases de datos de origen el 78% se realizaron sin ningún inconveniente mientras que el 22% presentaron diferentes inconvenientes.

Tabla 4.30 Pruebas realizadas

Base Origen	Base Destino	Presentaron problemas	%	No presentaron problemas	%	Número de pruebas
MySQL	Oracle	4	0,07	6	0,10	10
Oracle	Oracle	2	0,03	10	0,17	12
SQLServer	Oracle	3	0,05	15	0,25	18
PostgreSQL	Oracle	1	0,02	10	0,17	11
Archivos DBF	Oracle	3	0,05	5	0,08	8
Total			0,22	46,00	0,78	59

Elaborado por: Investigador

Fig. 4.31 Gráfico de resultados de las pruebas realizadas



Elaborado por: El Investigador

Los problemas que presentaron durante las pruebas se utilizaron en el FAQ con el fin de proporcionar soluciones a dichos problemas de una manera que sea comprensible para el usuario, evitando utilizar lenguaje técnico, identificadores de errores como son códigos o números propios de cada base de datos.

Los problemas presentados durante las pruebas son los siguientes:

Tabla 4.31 Problemas representativos entre bases de datos

Base Origen	Base Destino	Número de problemas	Problema
MySQL	Oracle	4	1. Los mensajes no visualizaba los problemas de los parámetros
			2. El formato de tipo de dato date no se ajustaba al destino
			3. El límite de los 10 datos de comprobación no funcionó
			4. Diferencia en los conjuntos de caracteres
Oracle	Oracle	2	1. El formato del tipo de datos date se invertía al realizar la transferencia
			2. La cadena de conexión no diferenciaba entre origen y destino
SQLServer	Oracle	3	1. El método comprobar no funcionaba
			2. La sintaxis de conexión fallaba en ocasiones concretas
			3. Diferencia en los conjuntos de caracteres
PostgreSQL	Oracle	1	1. Diferencia en los conjuntos de caracteres
Archivos DBF	Oracle	3	1. Los datos se cruzaban entre ellos
			2. Permitía abrir el mismo archivo lo que puede provocar duplicación de datos
			3. Diferencia en los conjuntos de caracteres

Elaborado por: Investigador

Frecuencia de los problemas presentados

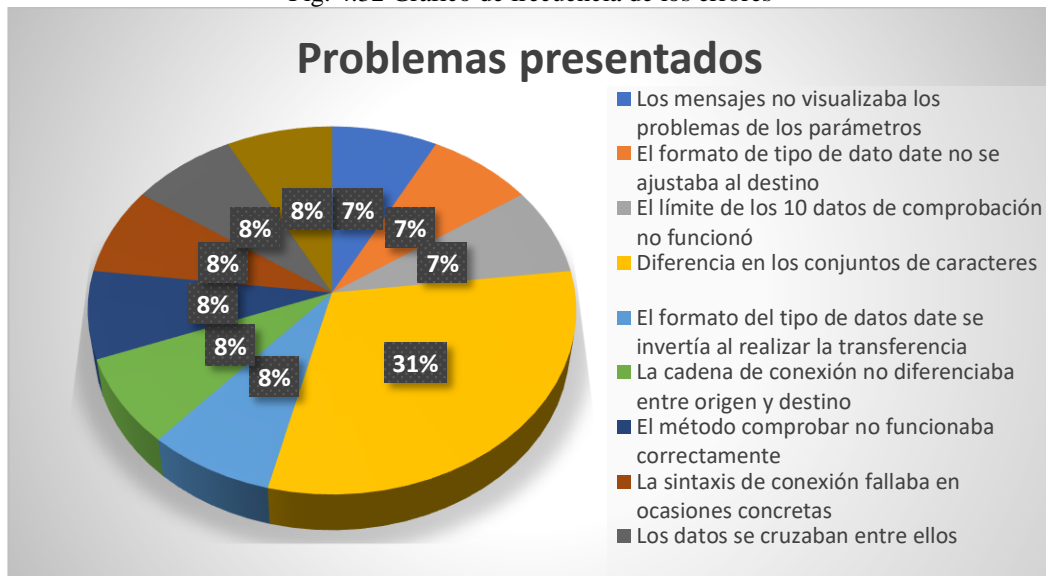
Tabla 4.32 Frecuencia de los errores

Problema	Frecuencia	Porcentaje
Los mensajes no visualizaban los problemas de los parámetros	1	0,08
El formato de tipo de dato date no se ajustaba al destino	1	0,08
El límite de los 10 datos de comprobación no funcionó	1	0,08
Diferencia en los conjuntos de caracteres	4	0,31
El formato del tipo de datos date se invertía al realizar la transferencia	1	0,08
La cadena de conexión no diferenciaba entre origen y destino	1	0,08
El método comprobar no funcionaba correctamente	1	0,08
La sintaxis de conexión fallaba en ocasiones concretas	1	0,08
Los datos se cruzaban entre ellos	1	0,08
Permitía abrir el mismo archivo lo que puede provocar duplicación de datos	1	0,08
Total	13	100%

Elaborado por: Investigador

En la figura 4.51 se demuestra que el problema más recurrente es la diferencia en los conjuntos de caracteres este problema se presenta con frecuencia por la naturaleza del sistema que trabaja con dos arquitecturas de datos ya establecidas y con la limitante de no poder modificar la base de datos origen fuera del proceso de transferencia, debido a esta limitante la propuesta de solución a este problema es proporcionar información específica del o los campos que se vean afectados para solucionarlo en la arquitectura de la base de datos destino la cual es de propiedad de Solinfo y tiene la capacidad de modificarla.

Fig. 4.32 Gráfico de frecuencia de los errores



Elaborado por: El Investigador

- Tiempos de espera del sistema

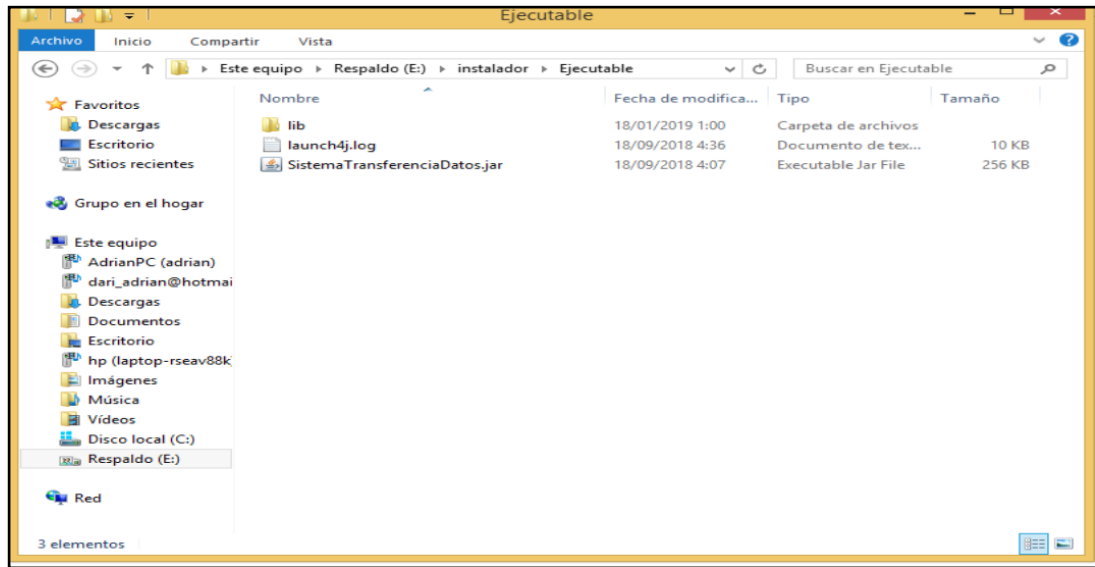
Este aspecto depende de varios factores para ser medido la cantidad de datos que se van a transferir, la carga del servidor que mantiene la base origen y el servidor de la base destino en caso de tratarse de una conexión remota influye la cantidad de tráfico de la red a la cual se conectara, estos aspectos son muy variables por lo cual no se puede realizar una medición exacta desde el propio sistema, sin embargo, Oracle la cual es la base de datos destino y se encuentra en la empresa Solinfo cuenta con la herramienta necesaria para visualizar las modificaciones en la base con esto se puede comprobar fácilmente los resultados de la transferencia aunque el establecer un tiempo estimado depende de los aspectos anteriormente mencionados.

4.6 Implementación del sistema de transferencia de datos

Para realizar la implementación del sistema de transferencia de datos se debe crear un instalador, para lo cual se generó un jar del proyecto desde Netbeans hacia una ubicación específica al igual que las librerías utilizadas para realizar las conexiones a las bases de datos relacionales y no relacionales.

A continuación, se demuestra los pasos necesarios para la creación del instalador del sistema de transferencia de datos.

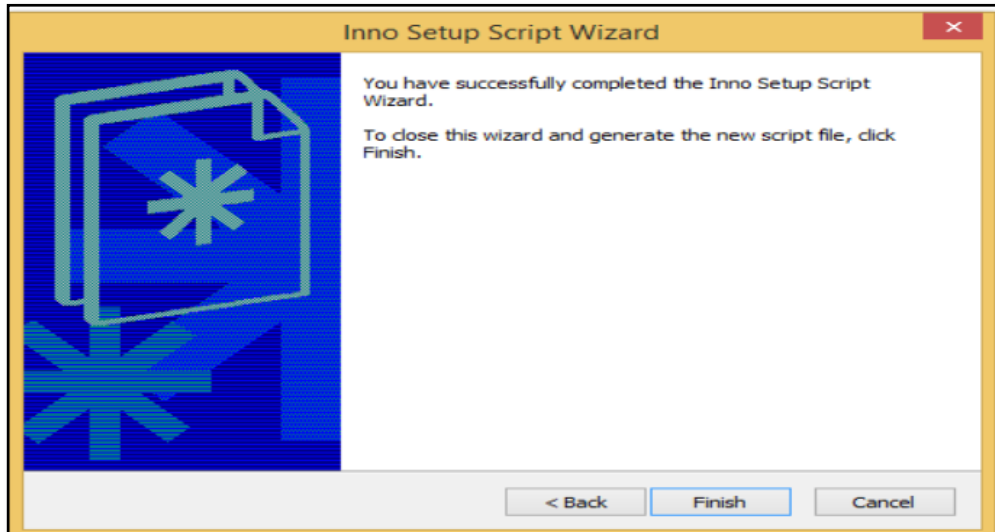
Fig. 4.33 Generación del jar



Elaborado por: Investigador

Con el jar del proyecto y las librerías utilizadas se generó el ejecutable del sistema de transferencia de datos con la herramienta Inno Setup versión 5.6.1

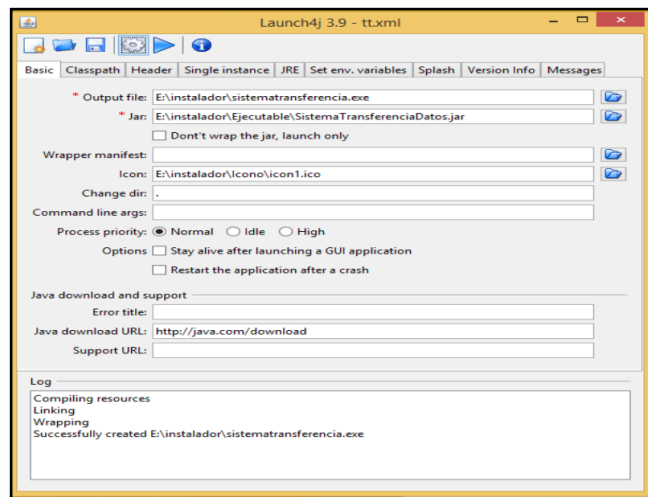
Fig. 4.34 Generación del ejecutable



Elaborado por: Investigador

Finalmente, con el ejecutable del sistema, la ubicación del jre y un icono, se crea el instalador con la herramienta launch4j 3.9.

Fig. 4.35 Generación del instalador del sistema



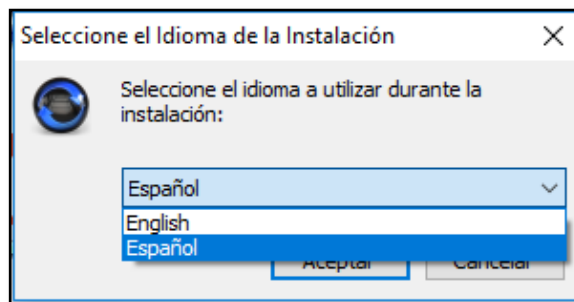
Elaborado por: Investigador

Instalación

Para realizar la instalación del sistema de transferencia de datos se ejecuta el archivo setup que se generó del paso anterior.

A continuación, aparecerá la siguiente ventana.

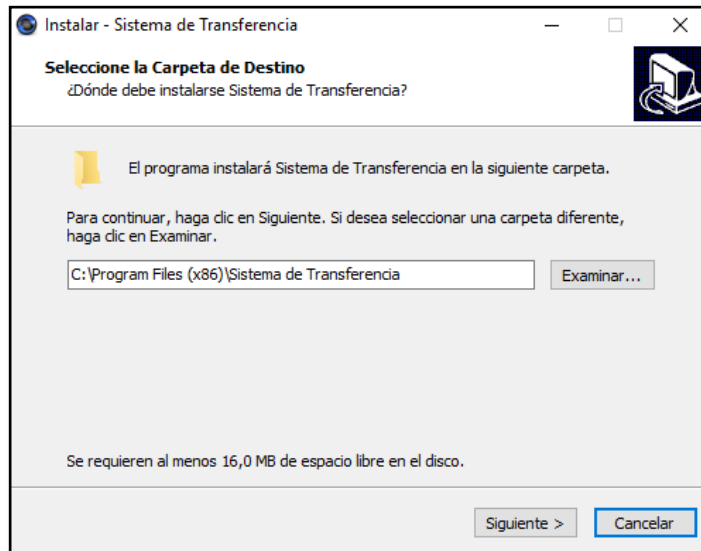
Fig. 4.36 Selección de idioma



Elaborado por: Investigador

En esta ventana se puede elegir el idioma en el que se desea realizar el proceso de instalación. Seleccionado el idioma se presiona el botón aceptar y se deberá definir la ubicación de instalación del sistema.

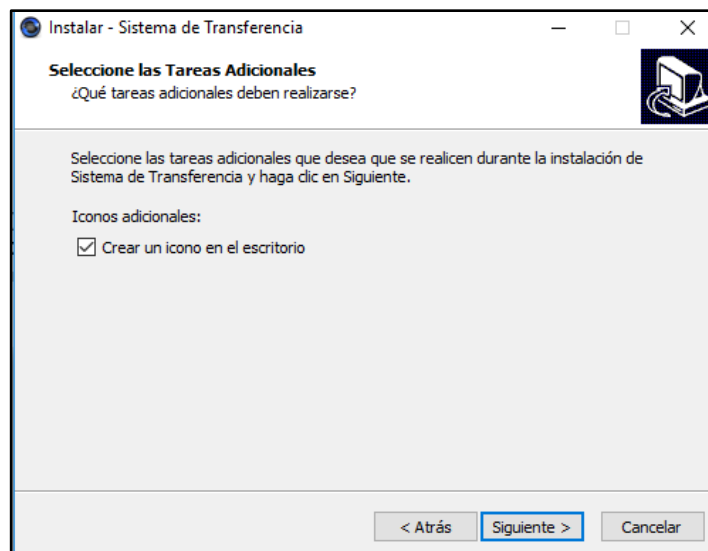
Fig. 4.37 Selección de la carpeta de destino



Elaborado por: Investigador

En la siguiente ventana permite crear un icono de acceso directo en el escritorio esto si el usuario así lo desea caso contrario desmarcar dicha opción.

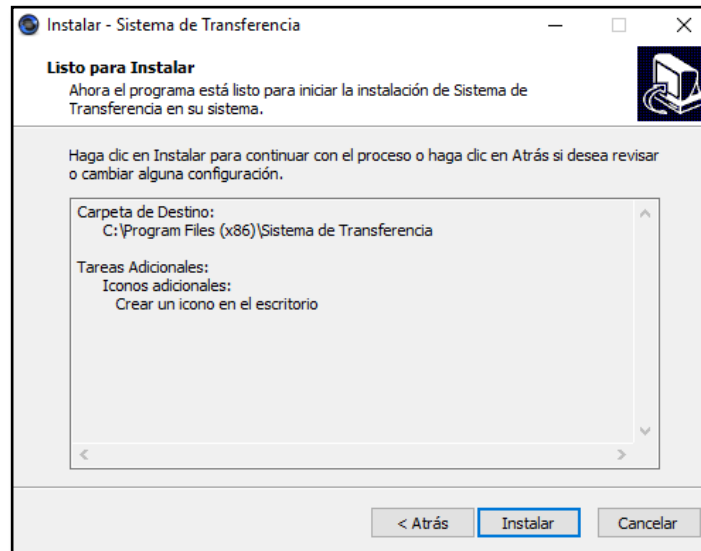
Fig. 4.38 Tareas adicionales



Elaborado por: Investigador

A continuación, se presenta un resumen de las opciones que seleccionamos con anterioridad.

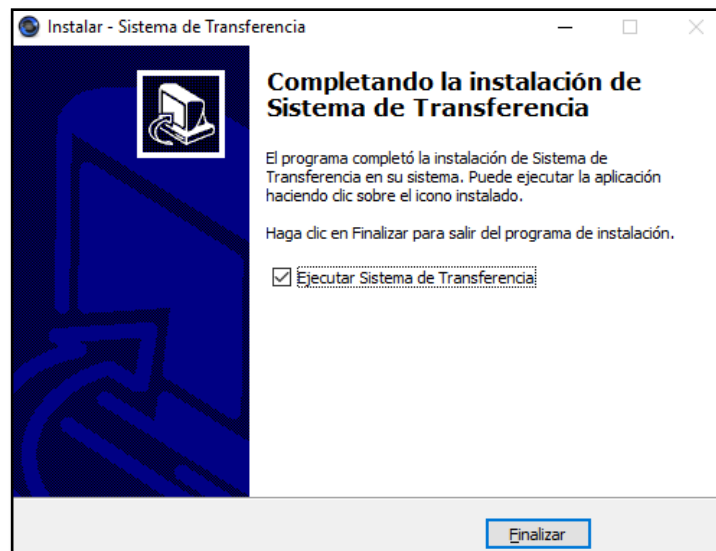
Fig. 4.39 Resumen de la instalación



Elaborado por: Investigador

Finalmente, la instalación finalizará permitiendo ejecutar el sistema si así el usuario lo desea.

Fig. 4.40 Instalación completada



Elaborado por: Investigador

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Los procesos de selección, transformación y carga de datos se realizan con el uso de procedimientos internos en Oracle, lo que permite el traspaso de datos de Oracle a Oracle, este esquema de trabajo limita el servicio de comprobantes electrónicos a clientes que utilizan Oracle.
- Existe una gran variedad de bases de datos, que se utilizan de acuerdo a la necesidad de una empresa, las bases de datos cuentan con diversas versiones que ajustan o cambian funcionalidades que puede afectar de forma positiva o negativa al funcionamiento de una empresa, por este motivo se seleccionaron las bases de datos más utilizadas con versiones que se encuentran ya establecidas en el mercado.
- Durante el desarrollo del sistema de transferencia de datos se identificó al proceso de transformación de datos como el proceso más crítico del sistema, la complejidad que envuelve el reconocer y transformar un formato definido por una base de datos a su similar o equivalente en otra base de datos represento un gran reto ya que no todos los datos cuentan con su equivalente directo y se requirió adaptar los formatos para realizar este proceso.
- Debido a la naturaleza del sistema se puede generar errores en la vinculación de datos por la codificación con la que cuente el esquema de datos, con el fin de mitigar el grado de error que se puede generar se amplió la información de los datos que el sistema proporciona al usuario.

5.2 Recomendaciones

- Al administrador del sistema se recomienda realizar un análisis de nuevas bases de datos que pueden añadirse al sistema ya sea como origen o destino, de esta forma abarcar más arquitecturas para establecer generalidades y mejorar las funcionalidades propias del sistema.
- Se recomienda al administrador que realice mantenimiento al FAQ de ayuda si se conecta con una versión de base de datos diferente a las presentadas, ya que el identificar de error puede haber cambio.
- Durante el transcurso del desarrollo del sistema se presentaron requerimientos para incrementar las funcionalidades de este, tales como agregar filtros que permitan limitar los datos a obtener en base a condiciones de los mismos datos o la funcionalidad de ejecutar código SQL directamente desde el sistema.

Material de referencia

1) Bibliografía

- [1] Á. L. Toctaquisa Raza, “SISTEMA MULTIEMPRESARIAL CON ARQUITECTURA DE BASES DE DATOS DISTRIBUIDA PARA ADMINISTRACIÓN DE PROCESOS OPERACIONALES – CASO DE APLICACIÓN EMPRESA ENMARSICIA LTDA.,” p. 131, 2017.
- [2] S. B. Caraguay Martínez, “Metodología para migración de datos de permita asegurar y conservar la integridad y consistencia de la información administrada por la empresa VSYSTEMS,” *Uma ética para quantos?*, vol. XXXIII, no. 2, p. 168, 2012.
- [3] M. M. BERMEO, “MOVIMIENTOS DE DATOS – RELACION CON OTRAS BASES,” p. 69, 2010.
- [4] G. Villagomez, H. Álvarez, D. Vivanco, and F. Echeverría, “Implementación De La Migración De La Base De Datos Del Sistema Canopus De Informix 9 . 4c A Oracle 10g Implementación de la Migración de la Base de Datos del Sistema Canopus de Informix 9 . 4C a Oracle 10g Resumen,” no. May, 2017.
- [5] S. L. Pauta Ayabaca, “LAS BASES DE DATOS HETEROGÉNEAS Y SU INCIDENCIA EN LA GENERACIÓN DE REPORTES ACADÉMICOS DEL PERSONAL DOCENTE DE LA UNIVERSIDAD CATÓLICA DE CUENCA,” 2016.
- [6] P. Mendoza del Carpio, “MITOO : un framework para la migración de datos hacia bases de datos orientadas a objetos MITOO : A Framework for the Migration of Data to Databases Object-Oriented,” pp. 57–67, 2010.
- [7] C. A. CASTILLO ALVARADO, “UNA GESTIÓN DE DATOS PARA MEJORAR Y DAR SOPORTE A LA TOMA DE DECISIONES EN LOS NEGOCIOS,” *Univ. Politec. Val.*, vol. 18, no. 2, pp. 1–4, 2015.
- [8] S. E. Mancilla Escobar, “Uso De Bases De Datos Nosql Documentales Para Crear Sitios Web De Alto Rendimiento,” 2013.
- [9] S. Fernández Díez and J. Madero de la Fuente, “Auditoría de usabilidad y accesibilidad de aplicaciones web,” 2015.
- [10] A. d’Alòs-Moner and N. Ferran, “Del elefante a Internet: breve historia de las bases de datos y tendencias de futuro,” *El Prof. la Inf.*, vol. 10, no. 3, pp. 22–26, 2001.
- [11] O. de L. y C. de la FSF, “Lista de licencias con comentarios,” 2018. [Online]. Available: <https://www.gnu.org/licenses/license-list.html>. [Accessed: 19-Oct-2018].
- [12] INEI, “Herramientas para el Desarrollo de Sistemas de Información,” INEI, 1997. [Online]. Available: <http://www.gobiernodigital.gob.pe/publica/metodologias/Lib5083/INDEX.HTM>.
- [13] J. FERNANDEZ and J. C. GIRALDO, “METODOLÓGIA PARA LA CONSTRUCCIÓN DE UN MIGRADOR UNIVERSAL DE BASES DE DATOS,” *DYNA*, 2010. [Online]. Available: <https://revistas.unal.edu.co/index.php/dyna/article/view/25668/39195>.
- [14] S. L. Corona-Correa, “Factores críticos de éxito en el proceso de migración de bases de datos relacionales,” p. 10.
- [15] A. Y. LEGUIZAMON TARAZONA, “PAUTAS PARA UNA CORRECTA

- MIGRACION DE BASES DE DATOS,” 2017.
- [16] C. P. ORTEGA VELA, “PROCESO DE MIGRACIÓN DE DATOS PARA EL MÓDULO DE PAGOS DE LA FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS PARA LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES & INGENIERÍA EN NETWORKING PARA EL PROTOTIPO DEL SISTEMA ACADÉMICO PARA LA UNIVERSIDAD DE GUAY,” 2015.
- [17] PowerData, “Introducción a la Migración de Datos,” p. 26, 2015.
- [18] J. Sánchez, “En busca del Diseño Centrado en el Usuario (DCU): definiciones, técnicas y una propuesta,” 2011. [Online]. Available: http://www.nosolousabilidad.com/articulos/dcu.htm?utm_source=iNeZha.com&utm_medium=im_robot&utm_campaign=iNezha.
- [19] F. Cordova and E. Cuzco, “Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales,” *Tesis previa a la obtención del Título Ing. en Sist.*, vol. 1, p. 88, 2013.
- [20] J. Vegas, “ORACLE: Arquitectura,” 1998. [Online]. Available: <https://www.infor.uva.es/~jvegas/cursos/bd/orarq/orarq.html>.
- [21] M. de Educación, “Tipos de datos.” [Online]. Available: http://www.ite.educacion.es/formacion/materiales/93/cd/m2_1/tipos_de_datos.html.
- [22] Plesk, “Acceso a bases de datos con cadenas de conexión.”
- [23] I. Hansel, G. Del Busto, I. Osmel, and Y. Enríquez, “Bases de datos NoSQL,” *Rev. Telemática*, vol. 11, no. 3, pp. 21–33, 2012.
- [24] E. De Desarrollo and D. V. Foxpro, *Curso de Visual Foxpro*, no. I. .
- [25] “Tendencia de popularidad,” 2018. [Online]. Available: https://db-engines.com/en/ranking_trend.
- [26] “ranking definition,” 2018. [Online]. Available: https://db-engines.com/en/ranking_definition.
- [27] R. Internas and F. Comprobantes, “Comprobantes Beneficios Electrónicos Información básica Qué son ? comprobantes electrónicos ?”
- [28] “Comprobante electronico,” 2018. [Online]. Available: <https://edocs.ec/ques-un-comprobante-electronico/>. [Accessed: 06-Jun-2018].
- [29] B. Momjian, “PostgreSQL Introduction and Concepts,” *Read*, p. 462, 2002.
- [30] A. Enterprisedb, “Using the NoSQL Capabilities in Postgres.”
- [31] “PostgreSQL.” [Online]. Available: <https://www.postgresql.org/about/>.
- [32] “MySQL Data Types,” 2018.
- [33] O. Corporation, “MySQL,” 2018.
- [34] O. D. Sheet, “ORACLE DATABASE 10G,” pp. 1–2.
- [35] I. D. E. L. Producto, “Informe del producto,” pp. 2006–2007, 2011.
- [36] M. Rouse, “Microsoft SQL Server,” 2017.
- [37] “SQL Server Data types (Transact-SQL),” 2018.
- [38] Oracle, “Información de NetBeans IDE 8.1,” 2018. [Online]. Available: <https://netbeans.org/community/releases/81/>.
- [39] Ecu RED, “Microsoft Visual Studio.”
- [40] Y. Daniel and A. Balaguera, “Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles . Estado actual for mobile devices . present state,” 2013.
- [41] A. Navarro Cadavid, J. D. Fernández Martínez, and J. Morales Vélez,

- “Revisión de metodologías ágiles para el desarrollo de software,” *Prospectiva*, vol. 11, no. 2, pp. 30–39, 2013.
- [42] A. . Fallis, “Metodologia Actual Metodologia XP,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [43] P. Letelier and C. Penadés, “Métodologías ágiles para el desarrollo de software.”
- [44] L. M. Echeverry and L. E. Delgado Carmona, “Caso Practico De La Metodologia Agil Xp Al Desarrollo De Software,” p. 110, 2007.
- [45] A. C. Garcia, “La Programación Extrema Autores : Marycarmen Díaz Labrador,” no. November 2013, 2017.

2) Anexos y Apéndices

Anexo A: Historias de usuarios

HISTORIA DE USUARIO N° 1

YO como administrador del sistema.

NECESITO acceder a la base de datos.

PARA obtener los datos.

Escenario N° 1: Validación de parámetros correcta

DADO: que se necesita establecer una conexión a la base de datos.

CUANDO: ingrese los parámetros de la base de datos

ENTONCES: el sistema verifica los parámetros si son válidos visualizará la página principal del sistema.

Escenario N° 2 Validación de parámetros incorrecta

DADO: que se necesita establecer una conexión a la base de datos.

CUANDO: ingrese los parámetros de la base de datos

ENTONCES: el sistema verifica los parámetros sino son válidos visualizará un mensaje mostrando el posible error y posible solución.

HISTORIA DE USUARIO N° 2

YO como administrador del sistema

NECESITO seleccionar los datos de una base origen.

PARA transferirlos a una base destino.

Escenario N° 1

DADO: que establezco una conexión a la base origen.

CUANDO: seleccione las tablas y columnas de la base origen.

ENTONCES: establezco una conexión con la base de datos destino (Oracle).

Escenario N° 2

DADO: que establezco una conexión a la base origen.

CUANDO: no seleccione las tablas o no seleccione las columnas de la base origen.

ENTONCES: el sistema visualizará un mensaje especificando si falta seleccionar las tablas o las columnas dependiendo del caso.

HISTORIA DE USUARIO N° 3

YO como administrador del sistema.

NECESITO conectarme a la base destino.

PARA vincular las tablas y columnas del origen con las tablas y columnas correspondientes en el destino.

Escenario N° 1 Validación de parámetros correcta

DADO: que se necesita establecer una conexión a la base de datos destino.

CUANDO: ingrese los parámetros de la base de datos

ENTONCES: el sistema verifica los parámetros si son válidos visualizará las tablas de la base de datos destino.

Escenario N° 2 Validación de parámetros incorrecta

DADO: que se necesita establecer una conexión a la base de datos.

CUANDO: ingrese los parámetros de la base de datos destino

ENTONCES: el sistema verifica los parámetros sino son válidos visualizará un mensaje mostrando el posible error y posible solución, manteniendo el estado de los procesos anteriores.

HISTORIA DE USUARIO N° 4

YO como administrador del proceso

NECESITO que los valores en cada estado no se pierdan

PARA poder regresar a un estado anterior.

Escenario N° 1: Acepta un estado por primera ocasión

DADO: acepto la selección de datos.

CUANDO: presione el botón de aceptar.

ENTONCES: el sistema guardara el estado actual de la selección.

Escenario N° 2: Acepta un estado por segunda ocasión

DADO: acepto la selección de datos.

CUANDO: presione el botón de cancelar.

ENTONCES: el sistema guardara el estado actual de la selección y eliminara el estado anterior.

Escenario N° 3: Cancela un estado por primera ocasión

DADO: acepto la selección de datos.

CUANDO: presione el botón de cancelar.

ENTONCES: el sistema no guardara el estado actual de la selección.

Escenario N° 4: Cancela un estado por segunda ocasión

DADO: acepto la selección de datos.

CUANDO: presione el botón de cancelar.

ENTONCES: el sistema no guardará el estado actual de la selección y mantendrá el estado anterior.

HISTORIA DE USUARIO N° 5

YO como administrador del proceso

NECESITO manejar datos sin importar el origen.

PARA manipular los datos de una misma manera

Criterios de aceptación

- El manejo de los datos debe ser totalmente transparentes sin importar su origen ya sea relacional o no relacional.
- Únicamente se manejan datos para su transferencia.
- Debe contar con mensajes personalizados de errores o inconsistencias con lenguaje no técnico.
- Únicamente debe modificar la base destino solo con la transferencia de datos no debe permitir ningún tipo modificación a la base origen fuera del contexto de la transferencia.
- Debe proporcionar una manera de comprobar la selección de datos, con el fin de evitar una mala selección de datos.
- Debe ser capaz de permitir modificaciones durante la selección de datos.
- Debe permitir conexiones remotas.
- Debe mantener los parámetros de conexión con el fin de agilizar una nueva transferencia.
- Las bases de datos que se deben incluir son Oracle y PostgreSQL como base de datos relacionales y para las no relacionales se sugiere VisualFoxPro.

**SISTEMA DE TRANSFERENCIA DE
DATOS DE ORIGEN MÚLTIPLE A
ORACLE PARA LA EMPRESA
SOLINFO**

Manual de Usuario

2019

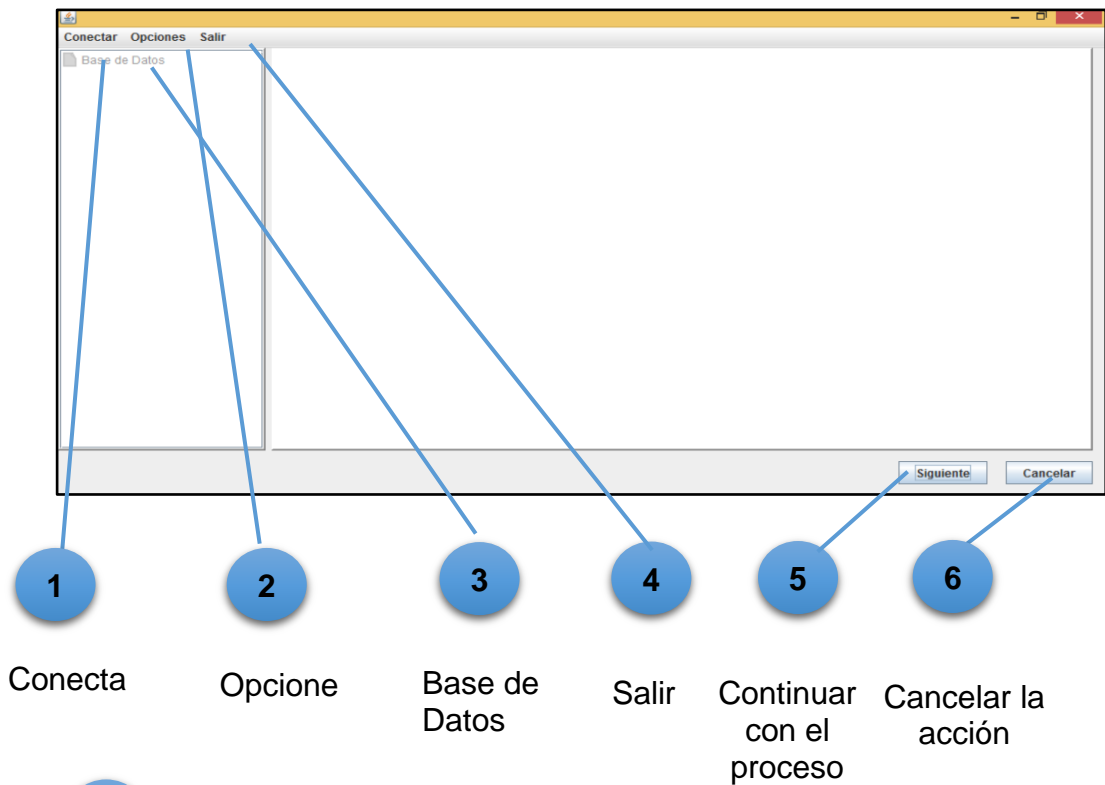
INTRODUCCIÓN

El presente sistema fue realizado como Proyecto de Titulación de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

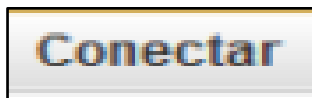
El propósito de este Manual es facilitar al usuario la operación de las diferentes pantallas de captura y consulta de la información que se administra en el SISTEMA DE TRANSFERENCIA DE DATOS DE ORIGEN MÚLTIPLE A ORACLE PARA LA EMPRESA SOLINFO.

PAGINA PRINCIPAL DEL SISTEMA

Esta ventana se muestra al ejecutar el software.



1 Conectar



Al presionar este menu permite realizar una conexión nueva a una base de datos entre las cuales pueden ser:

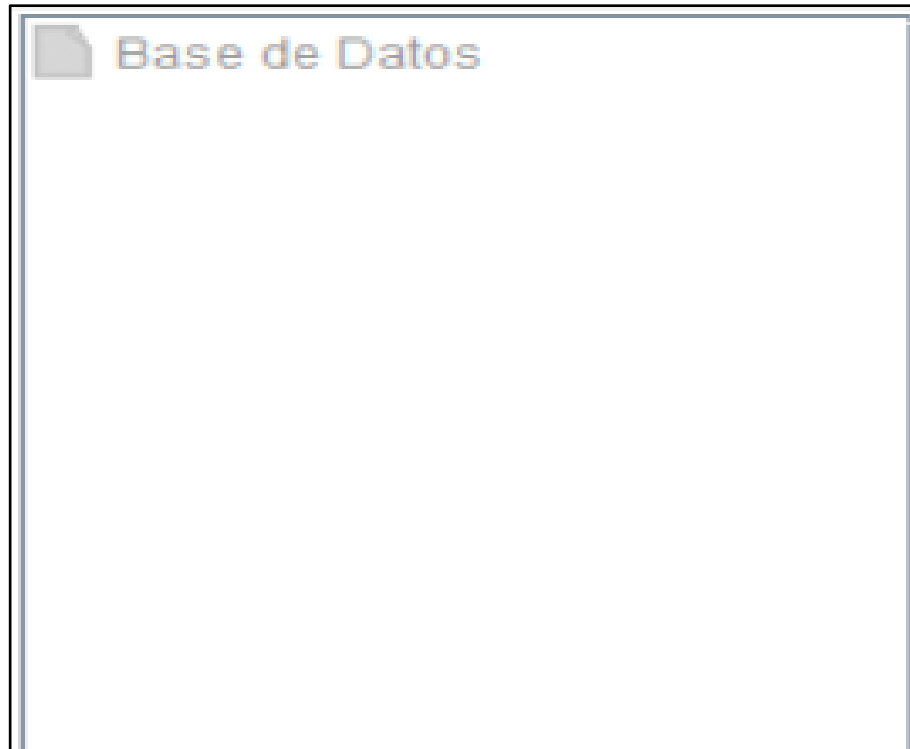
- Mysql.
- Oracle.
- Sqlserver.
- Postgresql.
- ArchivosDBF.

2 Opciones



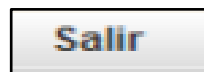
Menu desplegable que presenta las base de datos que contiene el sistema

3 Base de Datos



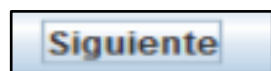
Espacio designado al contenido de la base de datos tablas y columnas cuando se realice una conexión estableciendo los parametros necesarios, este campo se modificará acorde a las acciones del usuario presentando información necesaria para cada uno de los procesos que el sistema necesita para realizar correctamente una transferencia de datos..

4 Salir



Este boton cierra totalmente el sistema

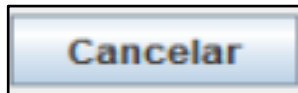
5 Continuar con el proceso



Este boton permite continuar los procesos acorde se vayan realizando correctamente, cuando se termine el proceso de selección de datos y se haya realizado la conexión a la base de datos destino el boton cambiara de nombre

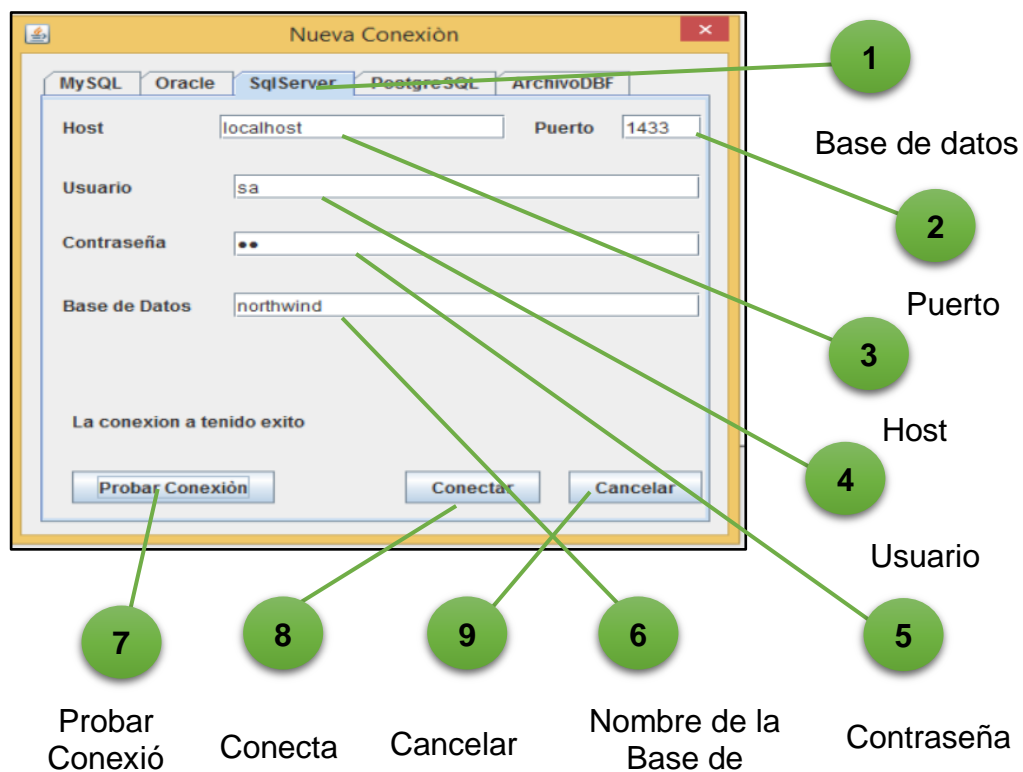
al de “Transferencia” este es el indicador de que se encuentra en el proceso de transformación y carga de datos.

- 6 Cancelar la acción



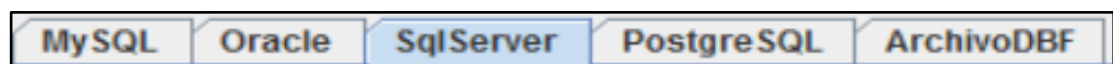
Este botón cancela la última acción realizada del usuario, en el caso de encontrarse en un paso de guardado de estado este botón restablecerá el ultimo estado consistete en el caso de no existir un estado se definirá con valores por defecto.

Página de nueva conexión



Esta ventana permite realizar una nueva conexión a una de las bases de datos existente en el sistema a través de los parámetros necesarios para establecer **1** conexión.

Base de datos



Son los tipos de base de datos con los que sistema puede conectarse de acuerdo con la base seleccionada aparecerán los parámetros necesarios para realizar la conexión.

2 Puerto

Este campo es para el ingreso del puerto de la base de datos.

3 Host

Este campo es para el ingreso del host en el cual permite conexiones locales o remotas.

4 Usuario

Este campo es para el ingreso del usuario de la base de datos.

5 Contraseña

Este campo es para el ingreso de la contraseña de la base de datos.

6 Nombre de la base de datos

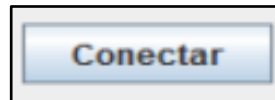
El campo es para el ingreso del nombre de la base de datos.

7 Probar Conexión

El botón sirve para comprobar si la conexión a la base de datos se estableció correctamente, a diferencia del botón conectar el proceso con este botón es más sencillo y rápido.

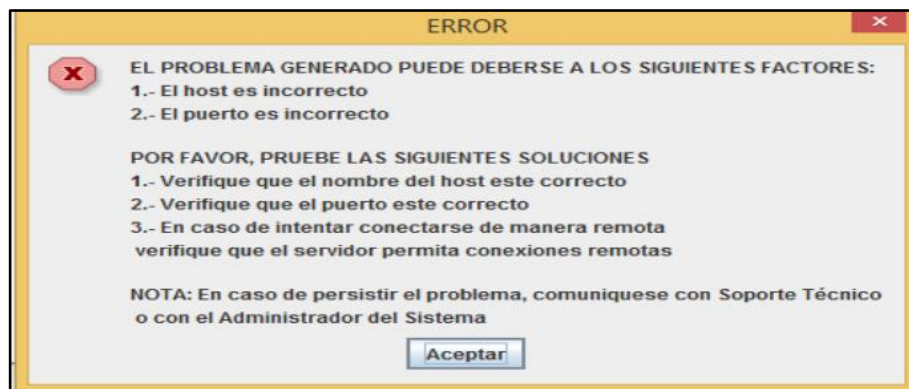
8

Conectar



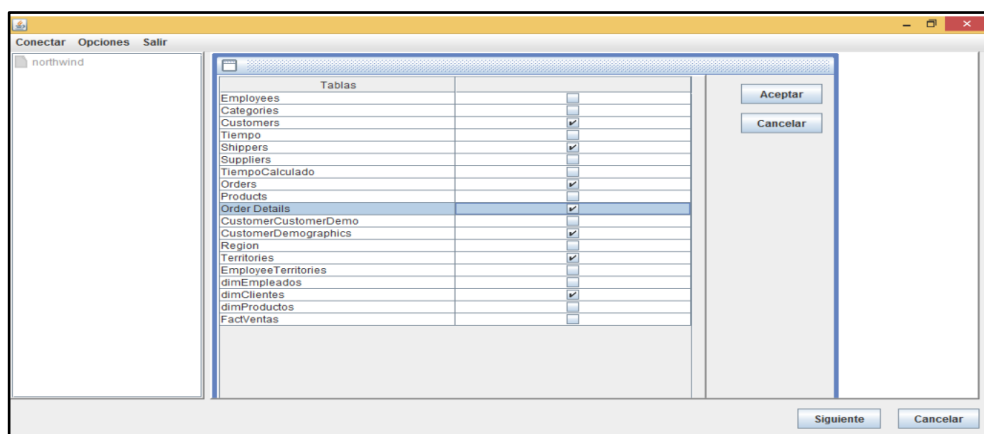
Este botón sirve para realizar la conexión a la base de datos, en caso de no presentar ningún problema cargara las tablas de la base de datos.

Caso de presentar algún problema.



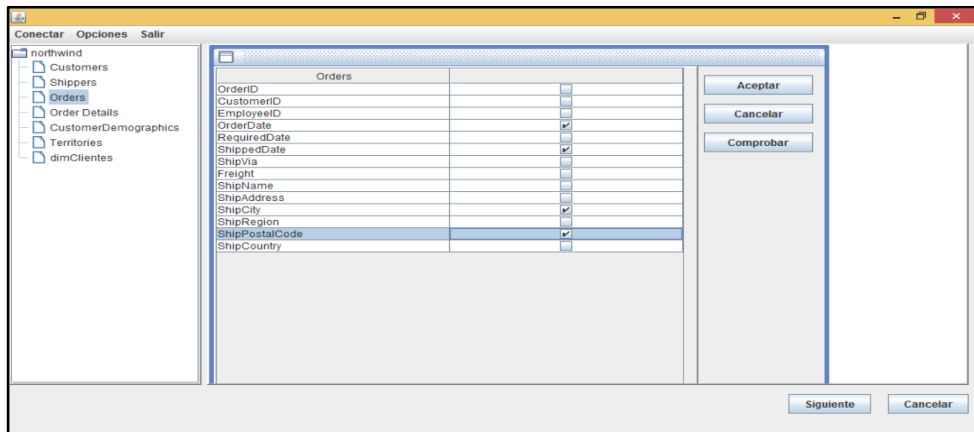
El sistema visualizará un mensaje personalizado con los posibles problemas y posibles soluciones las cuales el sistema ha identificado.

Caso de no presentar algún problema.

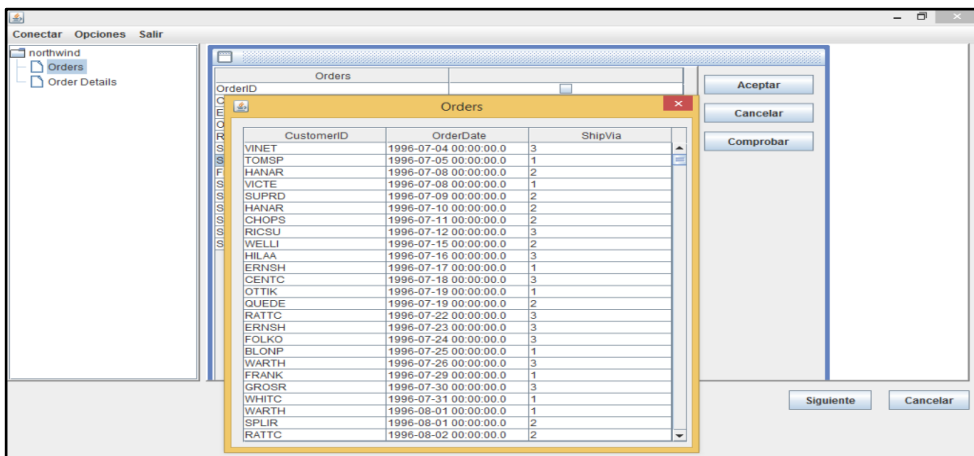


El sistema continuará con el proceso de selección de datos como primer paso la selección de tablas a utilizar en el proceso.

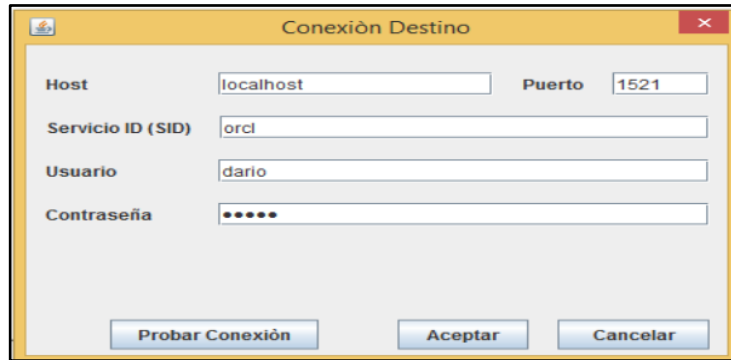
Una vez seleccionadas las tablas se procede con el botón siguiente el cual filtrará y cargará las tablas y continuará al paso de selección de columnas.



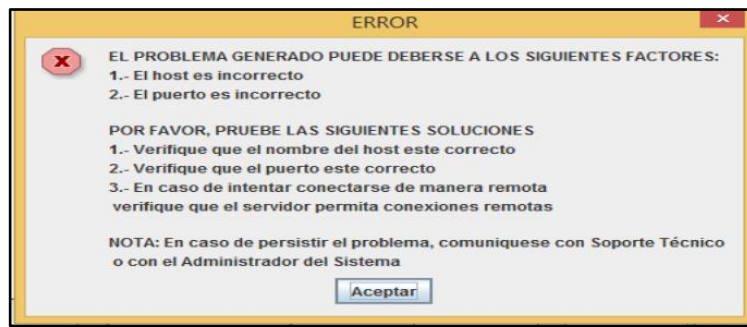
Durante este proceso es posible realizar una comprobación de los datos seleccionados con el fin de verificar si son los datos que se desea transferir.



Con la selección de las columnas el proceso de selección de datos en el origen terminará y el sistema continuará con la conexión a la base de datos destino ilustrada en la siguiente figura.

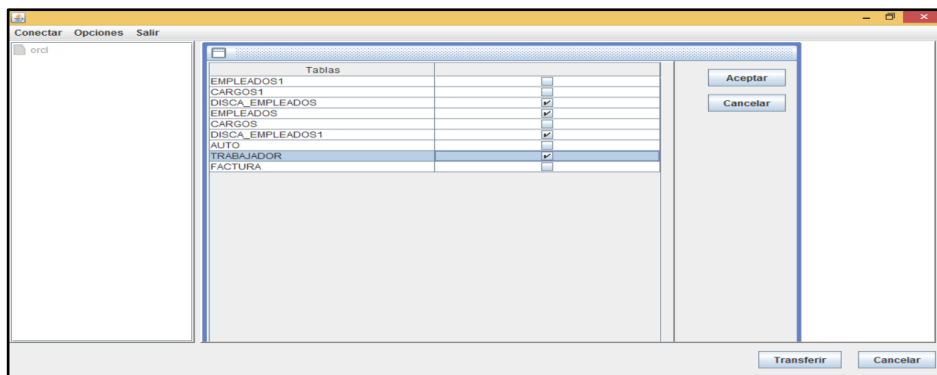


En esta ventana solicitará los parámetros de la base del mismo modo que se estableció la conexión de la base de datos origen, con un apartado de mensajes personalizados que sirven de guía durante este paso.

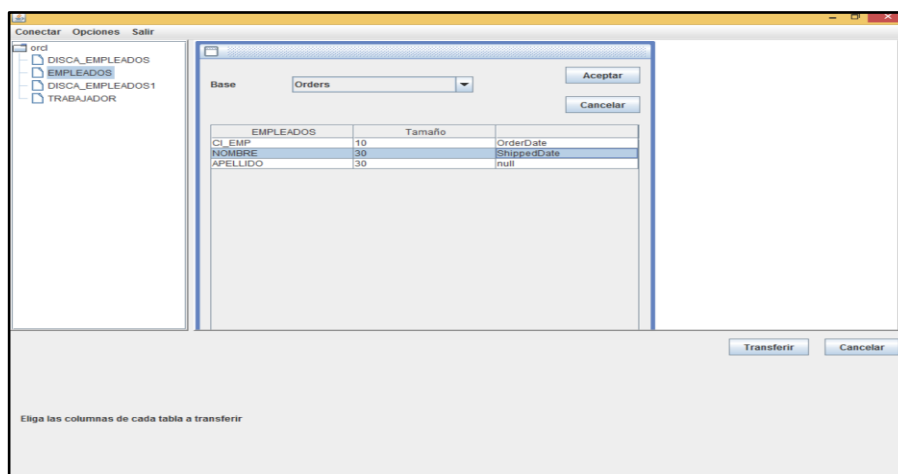


Esta ventana aparecerá cuando ocurra un error inesperado en el sistema, presentando el posible problema y su respectiva solución.

Una vez conectado a la base de datos destino se procede a realizar la selección de tablas a las que se desea insertar los datos.



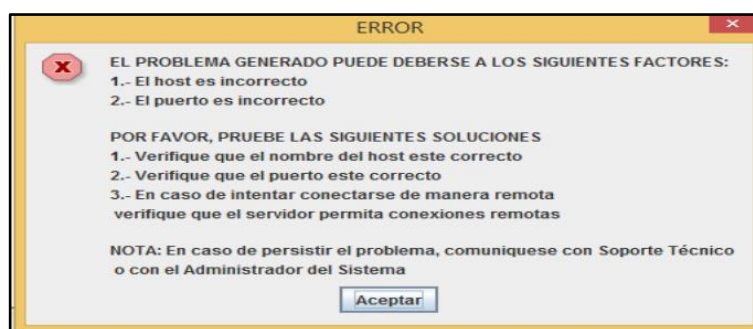
A continuación, se muestra la vinculación de datos de origen con el destino.



En este paso el botón de siguiente reconocerá los datos y cambiará su nombre al de “Transferir” es el indicador de que ya se encuentra en el proceso de transformación de datos, adecuando el formato del origen al destino.

Al presionar el botón “Transferir” se lleva a cabo dos procesos el de transformación de datos y el proceso de carga de datos. Con esto los procesos para la transferencia de datos terminan con un mensaje informativo de que sucedió durante este último proceso.

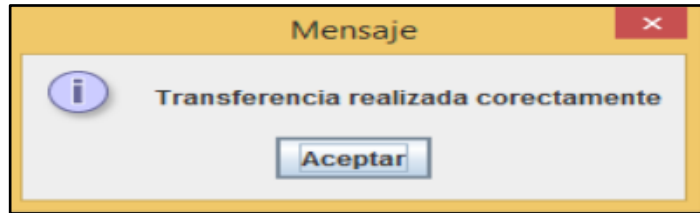
Caso de presentar algún problema.



El sistema visualizará un mensaje personalizado con los posibles problemas y posibles soluciones las cuales el sistema ha identificado y regresará a su último estado consistente del cual después de solucionar el problema puede volverse a intentar transferir los datos.

Caso de no presentar algún problema.

Si la transferencia es finalizada correctamente, sin ningún tipo de problema el sistema presenta el siguiente mensaje y se restablece.



Y todos los valores se restablecerán dejando al sistema preparado para realizar una nueva transferencia de datos.

Anexo C: Manual de instalación

MANUAL DE INSTALACIÓN DEL SISTEMA DE TRANSFERENCIA DE DATOS

Este manual está dirigido al personal técnico responsable de instalar el sistema de transferencia de datos por tanto se asume que el lector está familiarizado con conceptos básicos de administración bajo Windows y para la puesta en práctica de los conocimientos explicados aquí, cuenta con un usuario Windows que posee los permisos adecuados (normalmente un usuario Administrador de Windows).

Sistema de transferencia de datos es un software diseñado para establecer conexión con determinadas bases de datos y realizar la transferencia de datos, por lo que el software como tal se entrega separado de las Bases de Datos. Por este motivo se requiere de su independiente de las bases de datos.

Requerimientos del sistema

a) Requerimientos de hardware

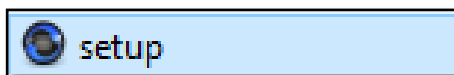
- Computadora

b) Requerimientos de software

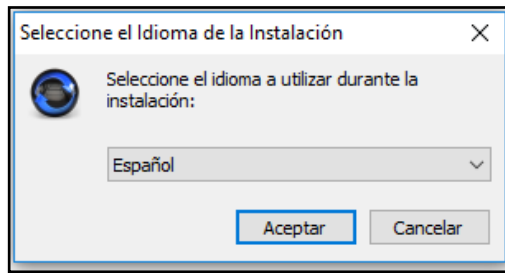
- Sistema operativo Windows
- Instalador del Sistema de transferencia de datos
- Contar con un jre mínimo la versión 1.7.0
- Se requiere al menos 16 MB libres en el disco

Instalación

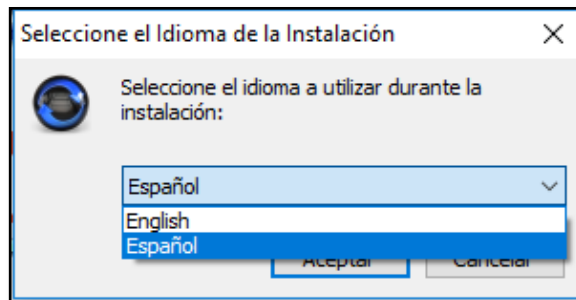
Para realizar la instalación del sistema de transferencia de datos se ejecuta el archivo setup.



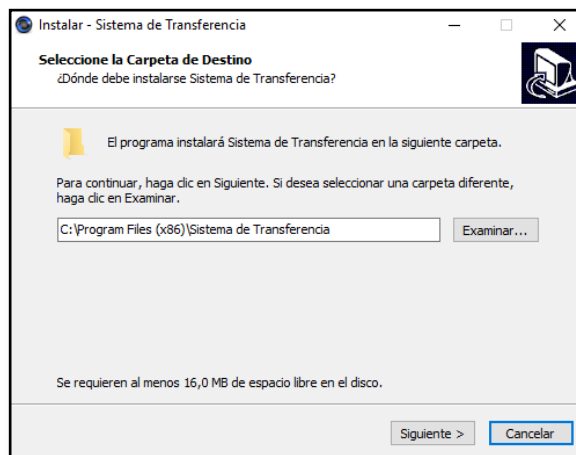
A continuación, aparecerá la siguiente ventana.



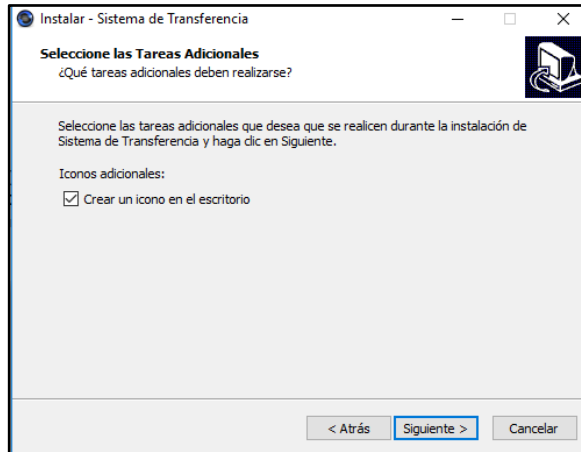
En esta ventana se puede elegir el idioma en el que se desea realizar el proceso de instalación.



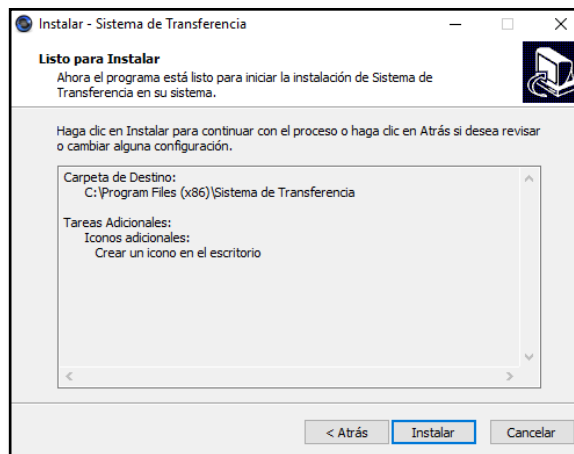
Seleccionado el idioma se presiona el botón aceptar y se deberá definir la ubicación de instalación del sistema.



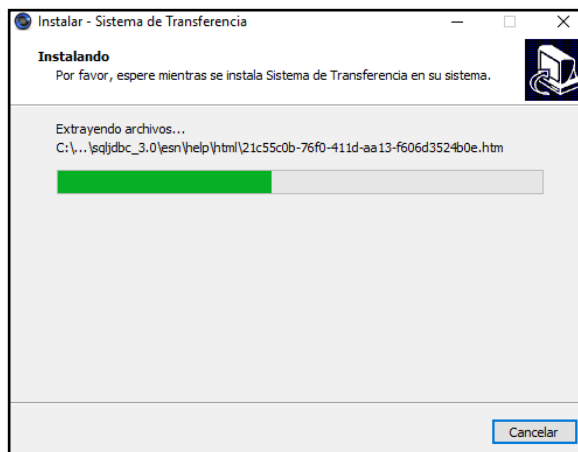
En la siguiente ventana permite crear un icono de acceso directo en el escritorio esto si el usuario así lo desea caso contrario desmarcar dicha opción.



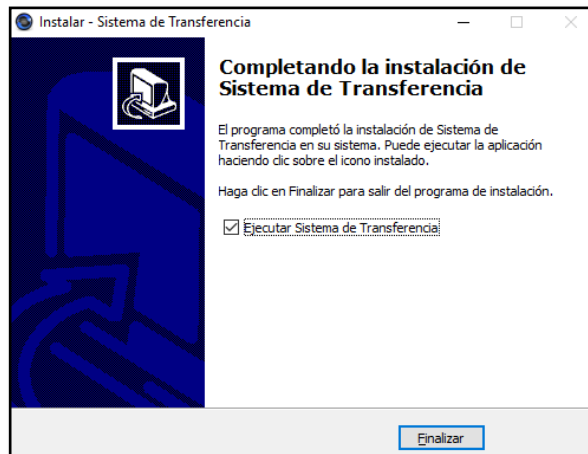
A continuación, se presenta un resumen de las opciones que seleccionamos con anterioridad.



Al presionar el botón Instalar comenzará la instalación deberemos esperar que dicha instalación termine.



Finalmente, la instalación finalizará permitiendo ejecutar el sistema si así el usuario lo desea.



3) Glosario Técnico y Acrónimos

BD: Base de datos.

SGBD: Sistema de Gestión de Bases de Datos.

COPYLEFT: Es un método general para liberar un programa u otro tipo de trabajo.

IDE: Integrated Development Environment. Es un ambiente integrado para el desarrollo de aplicaciones o software, consiste en un editor de código, herramientas para la construcción y depuración de código.

BLOB: Binary Large Objects / Objetos binarios grandes.

CLOB: Almacenan texto que contienen gran cantidad de bytes. Reemplaza al tipo de dato LONG.

SRI: Servicio de rentas Internas.

RAD: Desarrollo Rápido de Aplicaciones.

SQL: Lenguaje de Consulta Estructurada.

NOSQL: Es un enfoque hacia la gestión de datos y el diseño de base de datos que es útil para grandes conjuntos de datos distribuidos.

JRE: Incluye el JVM bibliotecas principales y otros componentes adicionales para ejecutar aplicaciones y applets escritos en Java.

USABILIDAD: Podemos definir la usabilidad como la medida en la cual un producto puede ser usado por usuarios específicos para conseguir objetivos específicos.

XML: Es un formato universal para datos y documentos estructurados.