

Universidad de las Ciencias Informáticas

FACULTAD 10

O2PMigration:

Herramienta para la migración de bases de datos Oracle a PostgreSQL

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autores

Marcos Ortíz Valmaseda

Susany Rodríguez Cervantes

Tutor

Ing. Santiago Almira Ramírez

Ciudad de La Habana

Marzo del 2009

Año del 50 Aniversario del Triunfo de la Revolución

“No apoyo solo el código abierto, o sea que la gente nada más conozca cómo funciona un programa, sino el software libre. Y la diferencia es que para nosotros es un asunto ético. El software privativo no es ético porque no respeta la libertad, y el código abierto no se interesa por esta cuestión. Si yo soy el “padre del código abierto” se hizo con “esperma” que me robaron sin mi consentimiento“

Richard Stallman

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter no exclusivo.

Para que así conste firmo la presente a los ____ días del mes de Marzo del año 2009

Marcos Ortíz Valmaseda

Firma del autor

Susany Rodríguez Cervantes

Firma del autor

Ing. Santiago Almira Ramírez

Firma del tutor

Datos de Contacto

Nombre y Apellidos del Tutor: Santiago Almira Ramírez.

Email: almira@uci.cu

Currículum Vitae:

Graduado de Ingeniería en Tecnologías Nucleares y Energéticas en el año 2005. Se incorporó a la Universidad de las Ciencias Informáticas desde ese mismo año impartiendo las asignaturas Física I y Física II. Ha tutorado varias tesis de grado que han tenido excelentes resultados. Está matriculado en la Maestría de Tecnologías Nucleares y Energéticas donde ha cursado los cursos correspondientes. Otros cursos de Postgrado que ha cursado son:

- Introducción al Matlab.
- Introducción al Latex para la tipografía de textos científicos

Agradecimientos

Marcos:

Agradezco a mi madre por tanta paciencia que ha tenido en estos años conmigo y haberme hecho un hombre de bien; agradezco a todas las personas que han confiado en mí; y me han apoyado en el desarrollo del presente trabajo.

Susany:

Agradezco a mi bebe por existir; a mi madre y a mi padre por tanto apoyo cuando más los necesité; a mi hermana del alma, a mi tity por tanto cariño y comprensión, a mi abuelita chichía por ser la abuela más linda y más buena del mundo, a toda mi familia por su apoyo incondicional, a mis profesores Loania y Susel por su ayuda en los momentos más difíciles en mi etapa de estudiante y a todos los que de una forma u otra contribuyeron en mi formación.

Dedicatoria

Marcos

A mi madre y a mi abuela, motores impulsores de mi vida

A mi padre por quererme

A Luisito por ayudarme a ser quien soy

A todo mis amigos y a los que han contribuido con mi formación en lo personal y en lo profesional

A Gladys por estar siempre a mi lado

A Santiago por indicarnos el camino

A Jorge Luis, Raciél y Abel por ser mis baluartes

A Maritza y a Leonardo por darme el apoyo que siempre me han dado

Susany

A mi bebe sobre todas las cosas, que ha sido desde el momento empunto que nació mi razón de vivir

A mi madre y a mi padre que han sido mi faro y guía

A mi hermana

A mi tity

A mi abuelita chichía

A toda mi familia y a los que han contribuido de una forma u otra con mi formación.

RESUMEN

La revolución informática de estos tiempos se desarrolla vertiginosamente; muchas veces tiende a sorprender su rápida y constante evolución. Cada nuevo día se necesitan aplicaciones; software en general, que hagan las acciones más fáciles al usuario; y hardware más potente que permita realizarlas con el menor costo de tiempo y esfuerzo. Pero a la vez que surgen estos nuevos componentes; las grandes empresas transnacionales productoras de software y hardware aumentan su fortuna y monopolio por segundos, debido a las patentes y licencias que protegen los mismos.

La presente investigación llevará a cabo el desarrollo de una herramienta bajo licencia GPL que se pueda utilizar en la migración de bases de datos Oracle de ETECSA a bases de datos PostgreSQL; implementada con el lenguaje Python con el framework de desarrollo gráfico multiplataforma Qt 4.

Palabras claves: Desarrollo, migración, Software Libre, Framework

Índice

| | |
|---|-----------|
| <u>Capitulo 1. Fundamentación teórica</u> | <u>5</u> |
| <u>Base de datos.....</u> | <u>6</u> |
| <u>Sistema Gestor de Base de Datos (SGBD).</u> | <u>7</u> |
| <u>Ventajas y desventajas de los SGBD</u> | <u>8</u> |
| <u>Definición de Software Libre.....</u> | <u>10</u> |
| <u>Beneficios del software libre.....</u> | <u>11</u> |
| <u>Licencia Pública General (GPL)</u> | <u>12</u> |
| <u>Licencia de Distribución de Software Berkeley (BSD)</u> | <u>12</u> |
| <u>Herramientas, lenguajes y tecnologías presentes y a usar.....</u> | <u>13</u> |
| <u>Eclipse IDE.....</u> | <u>13</u> |
| <u>Oracle Instant Client.....</u> | <u>13</u> |
| <u>PostgreSQL 8.3.....</u> | <u>14</u> |
| <u>Python.....</u> | <u>14</u> |
| <u>cx_Oracle.....</u> | <u>15</u> |
| <u>PyGreSQL.....</u> | <u>15</u> |
| <u>Qt 4.4.....</u> | <u>15</u> |
| <u>Ventajas de la utilización de Qt.....</u> | <u>16</u> |
| <u>Metodología a utilizar.....</u> | <u>16</u> |
| <u>SXP.....</u> | <u>17</u> |
| <u>Tendencias actuales de las aplicaciones de migración de Oracle a PostgreSQL.....</u> | <u>18</u> |

| | |
|--|----|
| Herramientas comerciales de migración de bases de datos Oracle a PostgreSQL..... | 18 |
| Herramientas libres de migración de bases de datos Oracle a PostgreSQL..... | 21 |
| Capítulo 2.Descripción y Análisis de la solución propuesta..... | 24 |
| Necesidad de migración a PostgreSQL 8.3..... | 24 |
| Orafce..... | 25 |
| Python 2.5..... | 26 |
| Necesidad del uso de Python..... | 27 |
| Módulos de Python a usar..... | 27 |
| cx_Oracle..... | 27 |
| PyGreSQL..... | 28 |
| cx_OracleTools..... | 28 |
| cx_Logging..... | 29 |
| Arquitectura base del sistema..... | 29 |
| Capítulo 3. Desarrollo ágil del sistema de migración de bases de datos Oracle a PostgreSQL | 30 |
| Planificación del proyecto por roles..... | 30 |
| Modelo de Dominio | 31 |
| Lista de Reserva del Producto(LRP)..... | 32 |
| Historias de usuarios y tareas de ingeniería..... | 33 |
| Plan de Releases..... | 45 |
| Diseño con metáforas..... | 45 |
| Capítulo 4. Validación de la solución propuesta..... | 48 |

| | |
|---|-----------|
| <u>Casos de Pruebas.....</u> | <u>48</u> |
| <u>Casos de pruebas para la Historia de usuario: U_O2P_0.1.....</u> | <u>48</u> |
| <u>Casos de pruebas para la Historia de usuario: U_O2P_0.2.....</u> | <u>49</u> |
| <u>Casos de pruebas para la Historia de usuario: U_O2P_0.3.....</u> | <u>50</u> |
| <u>Casos de pruebas para la Historia de usuario: U_O2P_0.4.....</u> | <u>51</u> |
| <u>Casos de pruebas para la Historia de usuario: U_O2P_0.6.....</u> | <u>53</u> |
| <u>Resultados Obtenidos.....</u> | <u>53</u> |
| <u>Acerca del tiempo de desarrollo</u> | <u>54</u> |
| <u>Acerca de las funcionalidades obtenidas.....</u> | <u>54</u> |
| <u>Conclusiones.....</u> | <u>56</u> |
| <u>Recomendaciones.....</u> | <u>57</u> |
| <u>Referencias Bibliográficas.....</u> | <u>58</u> |
| <u>Bibliografía.....</u> | <u>61</u> |
| <u>Anexos.....</u> | <u>64</u> |
| <u>Glosario de términos.....</u> | <u>78</u> |

INTRODUCCIÓN

El mundo de la informática en la actualidad se desarrolla a un ritmo bastante acelerado que a veces llega a sorprender a muchos desarrolladores, ingenieros de software, analistas, diseñadores; y tanto es así que cada día salen a la luz nuevos procesadores, software con nuevas características, entre otros componentes.

En este proceso vertiginoso y evolutivo de software y hardware, las grandes transnacionales productoras de estos componentes, se enriquecen por segundos; aprovechándose de la dependencia que han implantado a lo largo de estos años sobre las empresas o usuarios que utilizan dichos componentes, aplicándole las más restrictivas licencias de uso.

Un excelente ejemplo de estas grandes compañías es ORACLE¹, la cual constituye el líder del mercado en soluciones empresariales que necesiten de una base de datos; con su producto insignia: Oracle Enterprise System.

En los últimos años, esta compañía ha estado perdiendo terreno frente a varios archirrivalés que están bajo licencias libres como son PostgreSQL y MySQL.

PostgreSQL es un sistema gestor de bases de datos libre del tipo Objeto-Relacional, liberado bajo la licencia BSD. El desarrollo de PostgreSQL está dirigido por una comunidad de desarrolladores y organizaciones comerciales (como US.Army, la Biblioteca del Congreso de los Estados Unidos, el Laboratorio Nacional de Sandia, las compañías Loma Negra y Quilmas de Argentina, las compañías de Entel y la Superintendencia de AFPs en Chile, y una de las corporaciones financieras más grandes del mundo: Deutsche Bank), la cual es denominada como el PGDG (PostgreSQL Global Development Group). Está considerado como el mejor O-RDBMS (Object-Relational Database Management System) libre hasta el momento (según las mismas empresas que le dan soporte además de las que lo usan como gestor de bases de datos como el mismo Deutsche Bank). Actualmente soporta 34 plataformas. La última versión estable es la 8.3.6.

MySQL es un sistema de gestión de base de datos relacional, multihilo, multiusuario con más de seis millones de instalaciones en todo el mundo, o sea, grandes empresas o las mismas PYMES lo usan mucho en sus aplicaciones de almacenamiento de datos, como por ejemplo: Motorola, la cual tiene una de

las bases de datos más grandes del mundo. Actualmente es desarrollado por la SUN Microsystems, la cual lo compró por el precio de 1 millón de millones de dólares y su última versión es la 5.1 GA (Generally Available) [1].

En Cuba y específicamente la Universidad de las Ciencias Informáticas se ha insertado en la necesidad que existe de llevar a cabo la migración paulatina a software libre de las aplicaciones empresariales que todavía usan software propietario en nuestro entorno.

Particularmente en la facultad 10, el Proyecto UNICORNIOS² se dedica al desarrollo y producción de software libre para aportar una pequeña pero necesaria contribución a este proceso global de migración en todo el país.

Una de las empresas que utiliza Oracle en Cuba como sistema gestor de bases de datos es la Empresa de Telecomunicaciones de Cuba SA. (ETECSA), la cual tiene que pagar anualmente a Oracle Corporation por concepto de utilización de dicho software la suma de 3 millones de dólares anuales, lo cual se podría ahorrar con la utilización de un sistema gestor de bases de datos libre como PostgreSQL.

En la actualidad existen muchas herramientas con las cuales se puede llevar a cabo una migración de base de datos Oracle a bases de datos PostgreSQL, pero la mayoría son propietarias como por ejemplo: EnterpriseDB, líder de las empresas consultoras que dan soporte a PostgreSQL, ganadora del premio a la mejor solución de bases de datos en los LinuxWorld Excellence Awards del año 2008 con su producto Postgres Plus; o libres como por ejemplo: Ora2Pg, el cual es un script desarrollado en Perl ,pero se hace muy difícil su utilización y configuración; además de tener muy clara su documentación. Actualmente dicha herramienta está en su versión 4.9, la cual presenta nuevas características pero todavía se hace muy engorroso el trabajo con la misma, además de ser totalmente en modo texto.

A partir de los elementos expuestos se plantea como **problema**: ¿Cómo migrar las aplicaciones empresariales de ETECSA desarrolladas bajo el sistema gestor de bases de datos Oracle al sistema de bases de datos PostgreSQL de manera fácil, con el menor costo de tiempo y sin que se pierda calidad y fiabilidad en las mismas?

Si se desarrolla una herramienta con la calidad requerida, esta garantizará la calidad y fiabilidad de la

² Proyecto UNICORNIOS: Proyecto productivo de la Facultad 10 especializado en servicios de migración a software libre

migración de las bases de datos Oracle a PostgreSQL.

Por tanto, el **objeto de estudio** es, el estudio de las herramientas para la migración de bases de datos.

Derivado de esto, el **campo de acción** son las herramientas libres para la migración de las bases de datos Oracle a bases de datos PostgreSQL.

El **objetivo general** es: Desarrollar una herramienta basada en software libre que permita la migración de bases de datos Oracle a sistemas gestores de bases de datos PostgreSQL que garantice la calidad y fiabilidad de las aplicaciones empresariales de la Empresa de Telecomunicaciones de Cuba SA (ETECSA).

Como **objetivos específicos** se definieron los siguientes:

- Analizar las aplicaciones utilizadas para la migración de bases de datos Oracle.
- Diseñar la aplicación que permita la correcta y eficiente migración de las bases de datos de las aplicaciones empresariales de ETECSA implementadas con Oracle hacia PostgreSQL.
- Implementar la herramienta diseñada.

Para la realización de los objetivos específicos se proponen las siguientes **tareas**:

- Estudiar y analizar las herramientas para efectuar la migración satisfactoria de bases de datos Oracle a PostgreSQL
- Estudiar las últimas tecnologías y tendencias mundiales que se utilizan para la construcción de una herramienta como la que se pretende desarrollar.
- Seleccionar la metodología de desarrollo a utilizar.
- Seleccionar las herramientas para el desarrollo de la aplicación.
- Realizar el diseño de la aplicación usando la metodología de desarrollo seleccionada.
- Implementar una herramienta que facilite y agilice la migración de bases de datos Oracle de ETECSA a sistemas de bases de datos PostgreSQL.

En la presente investigación se utilizan varios **métodos científicos** para poder darle cumplimiento a las tareas de la misma. Uno de estos métodos a utilizar es el teórico donde se ponen de manifiesto el análisis y la síntesis, esto se ve presente en la interpretación del estado del arte y las tendencias actuales de las diferentes herramientas en el mundo, y a partir de dicho análisis poder dar conclusiones preliminares de por qué desarrollar otra herramienta con dichas características.

Otro de los métodos usados es la entrevista para la recogida de la información acerca de las herramientas a utilizar para el desarrollo de la aplicación; y para evaluar el comportamiento correcto o no de la aplicación.

Este trabajo está estructurado en 4 capítulos y anexos, que incluye todo lo relacionado con el trabajo investigativo sobre la migración de bases de datos Oracle a PostgreSQL, así como la implementación y pruebas del sistema en cuestión. A continuación se muestra una breve descripción de cada uno de los capítulos.

Capítulo 1: Fundamentación teórica. En este capítulo se hace un análisis del tema a tratar, de las tecnologías actuales relacionadas con él. Además se hace una breve descripción de las técnicas y tecnologías utilizadas para la implementación del sistema.

Capítulo 2: Descripción y análisis de la solución propuesta. Se realiza un análisis de las herramientas utilizadas, así como los lenguajes y tecnologías escogidas para darle solución al problema planteado; y la arquitectura base del sistema.

Capítulo 3: Desarrollo ágil del sistema de migración de bases de datos Oracle a PostgreSQL. En este capítulo se realiza el desarrollo ágil de O2PMigration. Se explica toda la dinámica del proyecto en forma de historias de usuarios, prototipos de interfaz de usuario y algunos modelos auxiliares además del plan de releases para las entregas intermedias.

Capítulo 4: Validación de la solución propuesta. En este capítulo se plasman los casos de pruebas o test de aceptación a las que fue sometida la aplicación en cada una de las iteraciones. Se exponen los resultados obtenidos y se muestran las funcionalidades alcanzadas en el período de desarrollo.

Capítulo 1. Fundamentación teórica

Cuba está inmersa en un proceso paulatino de cambios en el desarrollo en los medios informáticos apostando por las aplicaciones de código y estándares abiertos, lo cual se podrá lograr en un futuro. Esta premisa puede tocarse de cerca con las diferentes ediciones de las convenciones de la Ferias de Informática Habana, en las que se expone cada vez con mayor hincapié el concepto que sigue la filosofía del software libre y lo que beneficiaría al país la utilización de este tipo de software.

Hay muchos pasos dados en este sentido; la red INFOMED de Ministerio del Salud Pública, las amplias listas de discusión referentes a estos temas, la creación de la distribución cubana de GNU/Linux llamada Nova Linux, la Comunidad de Software Libre Cubana y la Facultad 10 de la Universidad de las Ciencias Informáticas pueden ser excelentes referencias sobre lo que se ha estado haciendo en esta materia en el país.

Otra de las empresas que se ha enfrascado en este proceso es la Empresa de Telecomunicaciones de Cuba S.A (ETECSA), la cual para la administración de sus disímiles servicios con que cuentan se han replanteado la necesidad de migrar todos sus servidores a software libre usando específicamente la distribución de GNU/Linux CentOS 5.0, la cual está basada en Red Hat Enterprise Linux, una de las más usadas en el mundo informático para la realización de tareas tan difíciles.

Esta empresa cuenta con una gran gama de servicios ya montados sobre esta plataforma, pero se les presenta la problemática del necesario cambio del sistema gestor de bases de datos por excelencia que es Oracle en su versión 10g, sin verse en la necesidad de perder o corromper todos los datos que tienen en las mismas, lo cual constituye el objetivo principal de esta investigación.

Base de datos

El término de Base de Datos fue escuchado por primera vez en 1963, en un simposio celebrado en California EUA; con los progresos en la adquisición y almacenamiento de datos en diferentes ambientes como gobierno, industria y aplicaciones científicas, arrojan como resultado el surgimiento de enormes bases de datos, cuyo tamaño se incrementa rápidamente, tanto en número de registros como en la dimensión de los mismos, llegando a alcanzar proporciones de almacenamiento del orden de los

Terabytes (Tb). La necesidad de explorar estas bases de datos y extraer información y conocimiento que sea de interés para los propietarios de las mismas, se ha incrementado en la actualidad.

Una Base de Datos o Banco de Datos se puede definir como un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso[2]. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. La utilización de bases de datos como plataforma para el desarrollo de Sistemas de Aplicación en las Organizaciones se ha incrementado notablemente en los últimos años, se debe a las ventajas que ofrece su utilización, algunas de las cuales se comentan a continuación:

- **Globalización de la información:** permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- **Eliminación de información inconsistente:** Si el sistema está desarrollado a través de archivos convencionales, una cancelación de compra por ejemplo deberá operarse tanto en el archivo de facturas del Sistema de Control de Cobranza como en el archivo de facturas del Sistema de Comisiones.
- Permite mantener la integridad de la información como cualidad altamente deseable y tiene por objetivo almacenar correctamente la información.
- Independencia de datos como factor esencial en la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de los datos implica un divorcio entre programas y datos.

Sistema Gestor de Base de Datos (SGBD).

Los Sistemas Gestores de Base de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos Sistema Gestor de Base de Datos (SGBD) y Sistema Manejador de Base de Datos (DBMS). [3]

El propósito general de los Sistemas de Gestión de Base de Datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información. Estos sistemas de Gestión de Base de Datos están compuestos de un lenguaje de definición de datos (DDL:

Data Definition Language), de un lenguaje de manipulación de datos (DML: Data Manipulation Language) y de un lenguaje de consulta (SQL: Structured Query Language).

Este tipo de software tienen características comunes que los definen tales como:

Índices: El empleo adecuado de índices en una relación acelera el acceso a la información, pero consume espacio considerable, es por esto que vale la pena hacer un análisis cuidadoso de cuáles atributos requieren ser indexados.

Niveles

Interno: cómo se almacenan y recuperan los datos (único).

Externo: cómo perciben los datos los usuarios (muchos).

Conceptual: enlace entre los anteriores. Existen distintos objetivos que deben cumplir los SGBD.

Abstracción de la información

Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos, por lo que se definen varios niveles de abstracción.

Independencia

La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Redundancia mínima

Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

Consistencia

En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

Seguridad

Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

Integridad

Un SGBD trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

Respaldo y recuperación

Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Control de la concurrencia

Un SGBD debe tener un mecanismo para el control del acceso de los usuarios de manera concurrente a la información, que podría derivar en inconsistencias.

Tiempo de respuesta

Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Ventajas y desventajas de los SGBD

Los SGBD tienen como desventajas los siguientes aspectos:

- El costo de actualización del hardware y software son muy elevados (Dependiendo de que tipo de SGBD se use, ya sea propietario o libre).
- El costo (salario o remuneración) del administrador de la base de datos es grande.
- El mal diseño de esta puede originar problemas futuros.
- Un mal adiestramiento a los usuarios puede originar problemas en el futuro.
- Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
- Generan campos vacíos en exceso debido a las constantes operaciones de tipo DML (INSERT/UPDATE/DELETE).
- El mal diseño de seguridad genera problemas en esta.

A pesar de todos estos inconvenientes los SGBD tienen una serie de ventajas como son:

- Facilidad de manejo de grandes volúmenes de información.
- Gran velocidad en muy poco tiempo.
- Independencia del tratamiento de información.

- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- Comprobación de información en el momento de introducir la misma.
- Integridad referencial al terminar los registros.

Los SGBD se desglosan en:

- SGBD Libres.
- SGBD Propietarias.

Dentro de estos dos grupos se analizan brevemente:

SGBD Propietarias:

- Oracle (Oracle Corp.)
- MySQL (SUN Microsystems)
- DB2 (IBM)
- InterBase (CodeGear)
- MaxDB (SAP)
- JavaDB (SUN Microsystems)
- SQL Server (Microsoft Corp.)
- Infomix (IBM)

SGBD Libres:

- PostgreSQL
- FireBird
- SQLite

Hay que destacar que hasta el año pasado MySQL formaba parte de los Sistemas Gestores de Base de Datos Libres pero actualmente forma parte de la lista de Gestores de Base de Datos Privativos, aunque él mismo tiene doble licenciamiento.

Esto estuvo dado debido que Sun Microsystems invirtió un total de 1 millón de millones de dólares, los cuales se distribuyeron en 800 millones en efectivo y otros 200 millones en la adquisición de acciones de la compañía MySQL AB.

Definición de Software Libre

La Free Software Foundation (Fundación de Software Libre) es una organización creada en Octubre de 1984 a partir del esfuerzo de Richard Matthew Stallman y otros entusiastas del software libre con el propósito de difundir este movimiento. El "Software Libre" es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en "libre" como en "libertad de expresión", lo que ha dado lugar a cierta confusión. "Software Libre" se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. [4]

De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y adaptarlo a necesidades específicas. El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Beneficios del software libre

Razones económicas

Ahorros importantes al liberarse del pago de licencias y especialmente por la replicación casi gratuita de aplicaciones comunes a toda la administración pública.

Independencia tecnológica

Se deja de depender de terceros (a menudo transnacionales) para el diseño, desarrollo y mantenimiento de sus sistemas de información, retomando el control total de sus procesos.

Control de la información

El acceso al código fuente, la libertad de inspeccionar el funcionamiento del software, la libertad de decidir la manera en que almacenan los datos y la posibilidad de modificar cualquiera de estos aspectos queda en manos del Estado, lo cual le permite el control total de la información y por consiguiente el ejercicio de la Soberanía Nacional.

Confiabilidad y estabilidad

El software libre realizado por comunidades está sometido a la inspección de un importante número de personas, este número de verificadores es mucho mayor que el del software propietario. Estas personas identifican los problemas, los resuelven, y comparten las soluciones con los demás.

Seguridad

La información que el Estado maneja generalmente es importante y/o confidencial, puede ser muy peligroso que esta información llegue a manos incorrectas.

Por esta razón es imprescindible que el Estado pueda verificar que su software no tenga puertas de entrada traseras, voluntarias o accidentales, y que pueda cerrarlas en caso de encontrarlas. Esto sólo es posible con software libre.

Desarrollo del País

Se genera Transferencia Tecnológica hacia los actores nacionales productores de software, acelerando el desarrollo endógeno y reforzando la soberanía nacional.

Licencia Pública General (GPL)

GNU GPL (Licencia Pública General) es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios [5] .

Existen varias licencias "hermanas" de la GPL, como la licencia de documentación libre GNU (GFDL) que cubre los artículos de la Wikipedia, la Open Audio License, para trabajos musicales, y otras menos restrictivas, como la LGPL, o la LGPL (Lesser General Public License o Library General Public License), que permiten el enlace dinámico de aplicaciones libres a aplicaciones no libres. La licencia GPL, al ser un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se le denomina contrato de licencia o acuerdo de licencia.

Licencia de Distribución de Software Berkeley (BSD)

La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. [6]

Es muy similar en efectos a la licencia MIT. Las versiones más antiguas de la licencia incluían una cuarta cláusula, llamada la cláusula de publicidad. En 1999 esta cláusula fue revocada con efecto retroactivo de las distribuciones BSD de la Universidad de California, Berkeley.

El autor, bajo esta licencia, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación.

Herramientas, lenguajes y tecnologías presentes y a usar.

Con la evolución existente en el mundo, y más marcado en las aplicaciones libres; se hace necesario realizar la selección de las herramientas, tecnologías y lenguajes para el desarrollo óptimo de la misma. Teniendo en cuenta las características del entorno en el que se trabaja, se realizó la debida selección con los especialistas involucrados en el caso.

Eclipse IDE

Eclipse IDE³ está considerado como uno de los mejores entornos de programación del momento desarrollado por la Fundación Eclipse, una organización sin ánimos de lucro que publica oficialmente todos los proyectos relacionados con Eclipse y su objetivo principal es el desarrollo de una plataforma libre de desarrollo que contenga todas las herramientas necesarios para el ciclo completo de un software determinado. [7]

3 Integrated Development Environment en inglés (Entorno Integrado de Desarrollo)

Actualmente la versión estable es la 3.4 con nombre de código: Ganymede. Muchas compañías le dan soporte a Eclipse, algunas de estas son: IBM, NASA, Intel, HP, AMD, entre otras.

Oracle Instant Client

Oracle es básicamente una herramienta cliente-servidor para la gestión de bases de datos, es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que solo se vea en empresas muy grandes y multinacionales, por norma general. Es un sistema manejador de Bases de Datos Relacional que hace uso de los recursos de los sistemas informáticos en todas las arquitecturas de hardware, lo que permite garantizar su aprovechamiento en ambientes cargados de información, por su capacidad de almacenar y acudir a los datos de forma recurrente. [8]

El manejador de Bases de Datos Oracle, surgió a final de los años 70 y principio de los años 80. George Koch y su equipo fueron los primeros en desembarcar en el terreno de Oracle en 1982, durante un proceso de evaluación de sistema de gestión de bases de datos para una importante aplicación comercial. La última versión es la 11g, pero la aplicación se desarrolla para la versión 10g, ya que es la versión usada en la Empresa de Telecomunicaciones de Cuba (ETECSA). Oracle Instant Client es la librería que permite el establecimiento de las conexiones con Oracle, el cual tiene varias opciones en dependencia de la versión de Oracle que se esté usando (8,9i,10g y 11g) y de la arquitectura. (ya sea de 32 o 64 bits)

PostgreSQL 8.3

PostgreSQL es un sistema gestor de base de datos objeto-relacional libre, liberado bajo la licencia BSD. Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PostgreSQL Grupo Global de Desarrollo (PGDG), sus siglas en inglés se definen como: PostgreSQL Global Development Group. PostgreSQL ha tenido una larga evolución, comenzando con el proyecto Ingres en la Universidad de Berkeley. La última versión estable es la 8.3.6. [9]

Este proyecto, liderado por Michael Stonebraker, fue uno de los primeros intentos en implementar un

motor de base de datos relacional.

Python

Para la desarrollo del sistema propuesto se propone emplear el lenguaje Python, ya que en la actualidad se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation.

Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. [10]

Es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación estructurada y programación funcional. Otros muchos paradigmas más están soportados mediante el uso de extensiones.

Usa tipos de dato dinámico y conteo referencia⁴ para el manejo de memoria. La última versión estable del lenguaje es actualmente la 2.6.1.

cx_Oracle

Es el módulo escrito en Python por la compañía Computronix que permite el acceso a la API de bases de datos desde el propio lenguaje a Oracle. El módulo está escrito para ser utilizado en las versiones oficiales de Oracle 8, 9i, 10g y 11g. La versión actual es la 5.0. [11]

PyGreSQL

Es el módulo que permite desde Python la conexión a PostgreSQL permitiendo la ejecución embebida de

4 El término conteo referencial se centra en el algoritmo de colección de basura en memoria implementado en Python

Véase <http://www.python.org/doc/2.5.2/ext/refcounts.html>

consultas desde el propio lenguaje. Se distribuye como un paquete dentro de los repositorios de la mayoría de las distribuciones GNU/Linux. La versión actual es la 4.0. [12]

Qt 4.4

Qt 4.4 ⁵es un framework para el desarrollo de aplicaciones multiplataforma perteneciente a la compañía Trolltech, la cual a su vez es propiedad de NOKIA [13]. La versión actual de Qt es la 4.5, la presenta características como:

- Compatibilidad multiplataforma con un solo código fuente.
- Excelente documentación (Qt Assistant).
- Internacionalización de aplicaciones (Qt Linguist).
- Arquitectura MV(Modelo-Vista).
- Soporte para XML, conexión a bases de datos, SVG, OpenGL, programación para redes.
- Soporte para aplicaciones 64-bits en la plataforma Mac con la API Cocoa.
- Librerías para analizar la carga de las aplicaciones haciendo benchmark de las mismas.

Ventajas de la utilización de Qt

Las ventajas de usar este framework gráfico son disímiles:

- Se distribuye con una documentación excelente además de ejemplos claros y concisos que aclaran cualquier duda que tengan los programadores.
- Es usado en disímiles tecnologías como la telefonía celular (Symbian S60⁶), aplicaciones de tipo SCADA, dispositivos multipropósito, y otras.
- Tiene licenciamiento dual en este momento, pero para el mes de junio saldrá la versión 4.5 bajo la licencia GPL.
- Es multiplataforma. El mismo código que se escribe en GNU/Linux, puede ser ejecutado fácilmente en Windows XP o en Mac OSX.

5 Qt es una marca registrada de Nokia

6 Symbian S60 es uno de los nuevos productos de Nokia con el sistema operativo del mismo nombre

- Por las amplias características con que cuenta, se hace muy fácil construir aplicaciones para distintos usos.
- Cuenta con un excelente diseñador gráfico de aplicaciones llamado Qt Designer, el cual hace más fácil todavía el trabajo del programador.
- Tiene un sistema integrado de internacionalización llamado Qt Linguist lo que permite portar las aplicaciones al lenguaje que se desee.

Metodología a utilizar

Las metodologías ágiles intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados. Se basan en promover iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto, logrando que se minimicen los riesgos desarrollando software en cortos tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar poco tiempo. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación.

Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Debido a las grandes ventajas que proporcionan estas metodologías se propone para el desarrollo de este trabajo el uso de la metodología ágil SXP. [14]

SXP

SXP es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos.

XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Consta de 4 fases principales:

- **Planificación-Definición:** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto;
- **Desarrollo:** es donde se realiza la implementación del sistema hasta que este listo para ser entregado;
- **Entrega:** es la puesta en marcha; y por último
- **Mantenimiento:** es la fase donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, Diseño, Implementación, Pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad.

Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo. [15]

Tendencias actuales de las aplicaciones de migración de Oracle a PostgreSQL.

Las herramientas existentes en el mundo se pueden clasificar en:

- **Propietarias – Comerciales:**
 - EnterpriseDB con su PostgreSQL Plus (USA)
 - CommandPrompt, Inc. con su Mammoth Postgresql (USA)
 - Fujitsu con sus servicios de consulta, entrenamiento y migración (Japón)
 - Hub.org con sus servicios de alojamiento Web (Canadá)
- **Libres**
 - Ora2Pg
 - Ora_Pg
 - OraLink

Herramientas comerciales de migración de bases de datos Oracle a PostgreSQL.

Como ya se ha expuesto anteriormente, PostgreSQL está considerada como la mejor base de datos Open Source del mundo. La última versión estable son: 8.3.6 y la 8.2.7. La versión 8.3, era muy esperada por la comunidad de usuarios de PostgreSQL del mundo, por la gran cantidad de mejoras que presenta la misma. [16]

Existen disimiles de empresas que le dan soporte a la aplicación. Una de las compañías líderes en este aspecto es: EnterpriseDB, la cual está considerada como la mejor solución para PostgreSQL en el mundo, además de que dispone de herramientas para realizar una migración desde Oracle 8,9 y 10g, aspecto por el cual ha ganado tanta popularidad.

Los productos insignias de la compañía son:

- **PostgreSQL Plus:** es una base de datos de código abierto fácil de utilizar y masivamente para la próxima generación y aplicaciones de la Web 2,0 de hoy en día. Construido sobre PostgreSQL, PostgreSQL Plus es una distribución binaria de fuente abierta, preconfigurada que simplifica el

despliegue de empresa, eliminando la necesidad de montar manualmente e integrar componentes de software de una variedad de lugares web.

➤ **PostgreSQL Plus Advanced Server:**

Es una base de datos de código abierto fácil de utilizar excelente para la próxima generación de aplicaciones llamadas Web 2,0 de hoy en día.

Construido sobre PostgreSQL, Postgres Plus Advanced Server es un conjunto de licencia comercial de Postgres Plus que añade muchas capacidades avanzadas.

Postgres Plus Advanced Server incluye todas las prestaciones y características de Postgres Plus, junto con las siguientes:

- Ejecuta aplicaciones escritas para correr contra una base de datos Oracle, a una fracción del precio de Oracle.
- Proporciona herramientas automatizadas para migrar de Oracle y otras bases de datos comerciales.
- Cloud Edition ofrece la máxima flexibilidad y escalabilidad para aplicaciones con rapidez o que cambian los requisitos de volumen incierto.
- Avanzada dinámica de ajuste (DynaTune™) con perfiles de trabajo asegura el rendimiento óptimo de bases de datos, incluso como las configuraciones de los patrones de uso o cambio.
- Gestión avanzada y herramientas de monitorización de proporcionar la máxima visibilidad y control sobre las bases de datos y aplicaciones.
- El registro de auditoría ayuda a abordar la ley Sarbanes-Oxley y otros requisitos de cumplimiento.
- Garantía Comercial y de indemnización.

Otros productos de **EnterpriseDB** son:

- **Database Server:** es la base de datos relacional, es el núcleo de la suite. (Dígase PostgreSQL Plus y PostgreSQL Plus Advanced Server).
- **Replication Server:** permite replicar datos a lo largo de toda la organización, casi en tiempo real.
- **EnterpriseDB Migration Toolset:** un punto clave, permite migrar todos los datos y la lógica de negocio de la Base de datos a EnterpriseDB.
- **EnterpriseDB Developer Studio:** es una consola de desarrollo y DBA que los usuarios de Oracle

pueden usar, prácticamente sin formación adicional.

- **EnterpriseDB DBA Management Server:** es una herramienta web para administrar, monitorizar, sacar registros de varias bases de datos EnterpriseDB y PostgreSQL.[17]

EnterpriseDB se inserta como una buena opción para la migración de bases de datos Oracle a PostgreSQL, pero al ser una empresa que ofrece soporte comercial, estos productos no son gratis ni nada por el estilo: la implantación de PostgreSQL Plus en una empresa cuesta \$ 95 por desarrollador y en la versión básica cuesta \$ 995 por socket(o sea por núcleos del procesador), y en la versión Premium cuesta \$2,995; y la implantación de PostgreSQL Plus Advanced Server los mismo que cuesta la de PostgreSQL Plus en la versión básica, y en la versión Premium cuesta \$4,495; y al tener esos precios no es viable para el país, ya que se seguiría con la misma dependencia de una empresa comercial para la puesta en marcha y mantenimiento de las bases de datos PostgreSQL. [18]

Lo mismo sucede con todos los productos y servicios que ofertan las otras empresas como Fujitsu y Hub.org; todos son caros e imposibles de implantar para la economía de Cuba; el cual está inmerso en el impulso y utilización del Software Libre por los organismos empresariales. Uno de los que ha dado el paso hacia la utilización del software libre es la Empresa de Telecomunicaciones de Cuba S.A (ETECSA), la cual se ha planteado la elaboración por los especialistas del Proyecto UNICORNIOS de la Facultad 10 de la Universidad de las Ciencias Informáticas de un Plan de Migración para efectuar la misma en dicho organismo; y uno de los pilares para efectuar dicha migración es que se debe contar con una aplicación libre que tenga como objetivo principal la migración de bases de datos Oracle a PostgreSQL, ya que ETECSA utiliza para sus grandes bases de datos Oracle 10g como sistema gestor.

Esto trajo como consecuencia hacer un estudio del estado del arte de las aplicaciones libres en el mundo con este objetivo en específico.

Herramientas libres de migración de bases de datos Oracle a PostgreSQL.

Las herramientas principales que se pueden utilizar con dichos objetivos son:

- Ora_Pg
- OraLink

- Ora2Pg

Ora_Pg

Básicamente, Ora_Pg es un script escrito en Perl encargado de exportar el esquema de una base de datos Oracle a un esquema compatible con PostgreSQL.

Es un proyecto que ya está desaparecido, no existe documentación de ningún tipo en Internet sobre el mismo y el sitio oficial de proyectos relativos a PostgreSQL (<http://pgfoundry.org>) tenía un apartado para este proyecto, pero como no se hacía más nada en el mismo, fue eliminado.

OraLink

OraLink es una pasarela para acceder a Oracle directamente de bases de datos PostgreSQL. Puede ser utilizado para la integración, la replicación, o simplemente para acceder a Oracle utilizando datos PostgreSQL. Esta herramienta está escrita en C, se acerca un poco más a lo que necesita pero debido que se ha descontinuado su desarrollo no resulta factible el uso de la misma.

Ora2Pg

Ora2Pg es un módulo de Perl para exportar una base de datos Oracle a un esquema de PostgreSQL. Esta aplicación se conecta a su base de datos Oracle, extrae su estructura, y genera un script SQL que puede cargar en su base de datos PostgreSQL.

La misma hace un backup de todo el esquema (tablas, vistas, secuencias, índices, privilegios) con todas las llaves ya sean primarias, foráneas y únicas exportadas a código PostgreSQL sin necesidad de editar el SQL generado⁷. Asimismo, se puede efectuar el volcado de datos Oracle en la base de datos PostgreSQL como proceso en línea o en un archivo. Puede elegir qué columnas se pueden exportar para cada tabla. Esta herramienta está desarrollada por Gilles Darold (gilles@darold.net) y es la que más promete a la hora de hacer efectiva la migración. Presenta disímiles ventajas con respecto a las anteriormente expuestas como:

- La división de columnas para cada tabla a exportar a PostgreSQL.
- Trabaja con versiones distintas de Oracle (8,9i y 10g).
- Posee una extensa guía de migración muy intuitiva en su sitio oficial.
- Existen varios desarrolladores trabajando en la aplicación.

⁷ En teoría, debe ser así, pero siempre hay que editar el código SQL generado

Pero Ora2Pg presenta varias desventajas por las cuales surge la necesidad de desarrollar una nueva herramienta libre como son:

- Al haber sido desarrollado en Perl, el mantenimiento del paquete es algo engorroso (se han establecido contactos con el desarrollador principal y las ideas futuras es migrarlo a Python).
- Al ser desarrollado en Perl, se está atado al largo ciclo de desarrollo propio del lenguaje a expensas de una nueva versión del mismo; además de que se hace un poco difícil también incorporarle una interfaz gráfica para una mejor interacción con el usuario.
- Existe muy poca documentación de la aplicación, y la disponible y considerada la más actualizada está en el idioma francés.
- La configuración de la herramienta se hace complicada por el hecho de que hay que conectarse constantemente al sitio oficial de módulos de Perl (<http://cpan.perl.org>) lo cual en Cuba, representa un problema crítico, por el reducido ancho de banda con que se cuenta y que todos no tienen acceso a Internet.
- La aplicación no está disponible en la mayoría de distribuciones de GNU/Linux, sólo existen paquetes para Debian GNU/Linux, Ubuntu, Red Hat y Fedora.
- Los algoritmos de conversión con que cuenta pueden mejorarse a la hora de realizar la conversión a código SQL compatible con PostgreSQL aprovechando al máximo las nuevas características del gestor como el mejor soporte para las vistas materializadas, particionamiento de tablas, espacios de tablas, entre otras.

Por todo lo anteriormente expuesto, se prosigue a desarrollar una nueva herramienta, la cual será escrita en Python 2.5 utilizando el framework de desarrollo multiplataforma Qt4 liberado bajo la licencia GPL, que tenga como base a la aplicación Ora2Pg por las excelentes ideas que promueve y por la experiencia de sus desarrolladores, los cuales ya llevan más de 4 años trabajando en la misma. Dicha herramienta trata de darle solución a la totalidad o a la mayoría de los problemas con que cuenta Ora2Pg:

- Para la conexión a la base de datos Oracle se cuenta con el módulo escrito en Python bajo licencia Open Source llamado **cx_Oracle**, el cual está disponible con su documentación y fuentes en SourceForge.net, además de sus ejemplos.

- La interfaz gráfica de la misma se desarrollaría con la combinación de **Python** y **Qt4** por las excelentes ventajas que presenta expuestas anteriormente; aunque el módulo principal de la misma se escribirá independiente de cualquier Interfaz Gráfica de Usuario⁸.
- El módulo **Python-pygresql** para la conexión a la base de datos PostgreSQL, el cual garantiza la fiabilidad a la hora de realizar esta tarea.
- El paquete **Postgresql-orafce** para la simulación de muchas de las funciones propias de Oracle.

Capítulo 2.Descripción y Análisis de la solución propuesta.

En el presente capítulo se llevara a cabo el análisis de las herramientas a utilizar, así como los lenguajes y tecnologías escogidas para darle solución al problema planteado.

Necesidad de migración a PostgreSQL 8.3

PostgreSQL es la primera base de datos de código abierto para la aplicación sincronizada de exploración, publicada bajo la licencia BSD; ésta ofrece a los usuarios la opción de desplegar una base de datos no privativa, lo que les permite ahorrar dinero, mejorar el rendimiento y aumentar la productividad. [19]

PostgreSQL ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos

⁸ Más conocida en el idioma inglés como Graffic User Interface(GUI)

adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema:

- clases
- herencia
- tipos
- funciones

Existen otras características que aportan potencia y flexibilidad adicional como son:

- Restricciones (Constraints)
- Disparadores (Triggers)
- Reglas (Rules)
- Integridad transaccional

Estas características colocan a PostgreSQL en la categoría de las Bases de Datos identificadas como **Objeto-Relacionales**. Nótese que éstas son diferentes de las referidas como orientadas a objetos, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. PostgreSQL tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en las que PostgreSQL fue pionera. [20]

PostgreSQL cuenta con varios años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de los datos.

Funciona en los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Es totalmente compatible con ACID⁸, tiene pleno soporte para las claves foráneas, procedimientos almacenados (en varios idiomas), soporte SQL92 y SQL99, tipos de datos incluyendo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. PostgreSQL también soporta el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o videos, tiene su propio lenguaje procedural llamado PL/pgSQL, muy parecido en su sintaxis al PL/SQL de Oracle; pero además cuenta con interfaces de programación nativas en lenguajes

8 Véase Glosario de términos

populares como C/C++, Java,.Net, Perl, Python, Ruby, Tcl, ODBC⁹, entre otros.

Orafce

Orafce es un paquete que se distribuye normalmente en las distribuciones de GNU/Linux junto con los paquetes contrib de la instalación de PostgreSQL, el cual como su nombre lo describe son una serie de aplicaciones de contribución que ayudan a la mejora del gestor. Orafce es desarrollado y mantenido por Pavel Stehule (stehule@kix.fsv.cvut.cz) y está liberado bajo la licencia BSD.

Orafce constituye una capa de emulación de funciones propias de Oracle, para que tengan más o menos el mismo comportamiento en PostgreSQL.

Algunas de estas funciones son:

- **Simulación de la tabla “dual”:** PostgreSQL no necesita una tabla “dual”, pero la misma es intensamente usada por los usuarios de Oracle para construcción de funciones, disparadores, reglas, secuencias, entre otras.
- **Simulación del módulo DBMS_OUTPUT:** El mecanismo que sigue PostgreSQL para el envío de alguna notificación al cliente es con la función RAISE NOTICE, pero Oracle lo hace de otra forma con dbms_output.put_line(). Esta simulación permite traducir menos código en PL/pgSQL. Este paquete incluye las funciones enable(), disable(), serveroutput(), put(), put_line() y otras, todas compatibles con su semejante en Oracle.
- **Simulación del módulo utl_file:** Permite a los programas construidos con el lenguaje PL/pgSQL leer y escribir ficheros que sean accesibles desde el servidor. Para el uso de este paquete se debe modificar la tabla utl_file.file_dir, especificando el o los directorios en los cuales se pueden realizar estas operaciones.
- **Simulación del módulo PLVdate:** Permite el trabajo con las fechas de días de negocio. Es multicultural, pero las configuraciones por defecto son sólo para los países europeos.
- **Simulación de los módulos PLVsrt y PLVchr:** Contiene útiles funciones para el trabajo con cadenas y caracteres.

⁹ Las interfaces de programación de PostgreSQL son PL/Python, PL/Perl, PL/Ruby, PL/Java, C entre otros

Para el uso en la base de datos de Orafce, solo se debe ejecutar el script orafce.sql situado en el directorio contrib de PostgreSQL de la siguiente forma:

```
psql -U postgres postgres < orafunc.sql
```

Python 2.5

Python es un lenguaje que es comparado con Tcl, Perl, Scheme o Java, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de un lenguaje interpretado o de script, con un tipado dinámico muy bien definido, multiplataforma y orientado a objetos¹⁰. Cuenta con módulos propios para la realización de pruebas sobre una amplia gama de plataformas. Esto nos permite detectar problemas durante el desarrollo más rápido, y como resultado de una forma mucho más sólida de liberación.

Python no es un lenguaje con ejecución rápido como los lenguajes compilados, pero sin embargo es más flexible y más portable. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi-interpretado.

Necesidad del uso de Python

Python es un lenguaje de sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros; hacen posible que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido. Éste es distribuido bajo OSI aprobada por la licencia de código abierto que hace que sea libre de usar, incluso para productos comerciales.

La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en Python parecen pseudocódigo. Por este motivo se trata además de uno de los mejores lenguajes para comenzar a programar.

Python no es adecuado sin embargo para la programación de bajo nivel o para aplicaciones en las que el

1 0 Python es tan flexible que puede adaptarse a cualquier paradigma de programación.

rendimiento sea crítico. Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles. [21]

Python posee muchas cualidades para seguir escalando dentro de los lenguajes de programación más usados en el desarrollo de Software Libre.

El entorno de ejecución de Python detecta muchos de los errores de programación que escapan al control de los compiladores y proporciona información muy rica para detectarlos y corregirlos. Es un lenguaje potente, seguro, flexible, pero con una gran cantidad de módulos para todas las necesidades que pueden ser útiles a la hora de programar con Python. [22]

Módulos de Python a usar

cx_Oracle

cx_Oracle es un módulo de extensión que permite el acceso a Oracle y se ajusta a las especificaciones de la API de base de datos en Python.

Éste utiliza siempre el setup.py para construir e instalar el módulo que hace uso de la DistUtils que es el módulo disponible en Python 2.0.

Este módulo se ha construido con Oracle 9.2.0, 10.2.0, 11.1.0 en Linux, Solaris y Windows. Otros han informado de éxito en otras plataformas tales como el Mac OS X.

Cx:Oracle es desarrollado por la compañía Computronix donde se puede encontrar toda la información acerca del módulo además de su sitio oficial en SourceForge.net.

PyGreSQL

PyGreSQL es un módulo de Python con interfaces de una base de datos PostgreSQL. PostgreSQL se incrusta la biblioteca de consulta para facilitar la utilización de las poderosas características de PostgreSQL desde un script de Python. La última versión estable es la 4.0.

PyGreSQL es desarrollado y probado en un sistema NetBSD, pero también funciona en la mayoría de las otras plataformas donde PostgreSQL y Python está funcionando.

Se basa en el código escrito PyGres95 por Pascal Andre. D'Arcy a PyGreSQL le cambió el nombre a partir de la versión 2.0 y sirve como el "BDFL" de PyGreSQL.. [23]

cx_OracleTools

cx_OracleTools son una serie de útiles scripts en Python que facilitan el trabajo del programador con bases de datos Oracle. Este módulo es desarrollado por la misma compañía que desarrolla cx:Oracle, por lo cual también se puede encontrar toda su documentación en el sitio de la compañía además del SourceForge.net.

Algunas de las funciones básicas de cx:OracleTools son:

- **DBDebugger**: Permite un simple debugging de PL/SQL.
- **DumpData**: Guarda toda la información de los SELECT como INSERT en un fichero.
- **ExportData**: Exporta todos los datos de la base de datos a un archivo portable.
- **ExportColumn**: Exporta todos los datos de una columna específica.
- **ExportXML**: Exporta los datos de una tabla en un fichero XML simple.

Para poder usar este módulo deber ser instalado en el huésped donde se encuentra el servidor Oracle.

cx_Logging

Es un módulo de Python que se comporta similarmente al módulo logging encontrado en la librería estándar de Python, pero con la diferencia de que éste tiene una interfaz del lenguaje C. Es usado por los

módulos cx:Oracle y cx:OracleTools. Es desarrollado y mantenido por Anthony Tuininga (anthony.tuininga@gmail.com).

Arquitectura base del sistema

Debido al estudio que se realizó en conjunto con los administradores de bases de datos de Oracle de ETECSA se llegó a la conclusión de que la arquitectura base del sistema estaría conformada por una arquitectura de 3 capas:

- **Capa de presentación:** la cual estaría centrada en la presentación al usuario que va a usar la aplicación, conformada por una interfaz de usuario desarrollada con Qt4.
- **Capa de lógica de negocio:** contendría el núcleo de la aplicación con sus principales módulos y funcionalidades.
- **Capa de Abstracción de datos:** la cual estaría conformada por los módulos de acceso a las determinadas bases de datos con las cuales interactúa. ([Ver Anexo # 1](#))

Capítulo 3. Desarrollo ágil del sistema de migración de bases de datos Oracle a PostgreSQL

En el desarrollo de este capítulo se realiza el desarrollo ágil de O2PMigration utilizando la metodología SXP la cual fue propuesta en el 2008 por la Ingeniera Gladys Marsi Peñalver Romero. Se explica la dinámica del proyecto a través de historias de usuarios, prototipos de interfaz de usuario y otros modelos auxiliares.

Planificación del proyecto por roles

| Rol | Responsabilidad | Nombre |
|------------------------------------|--|---|
| Gerente (Manager) | Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto. | Marcos Ortíz Valmaseda |
| Cliente (Customer) | El cliente participa en las tareas que involucran la lista de reserva del producto. | Empresa de Telecomunicaciones de Cuba SA. |
| Programadores (Programmers) | Es el encargado de producir el código y escribir las pruebas unitarias. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo. | Marcos Ortíz Valmaseda |
| Analista (Analyst) | Es el encargado de escribir las historias de usuario y las pruebas funcionales para validar su implementación. | Susany Rodríguez Cervantes |
| Diseñadores (Designers) | Encargados del diseño del sistema; así como el de los prototipos de interfaces, máximos responsables de la realización del diseño de | David Padrón Álvarez |

| | | |
|--------------------------------------|--|----------------------------|
| | las metáforas y supervisan el proceso de construcción. | |
| Encargado de Pruebas (Tester) | Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas. | Marcos Ortíz Valmaseda |
| Arquitecto (Architect) | Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema. Ayuda en el diseño de las metáforas. | Susany Rodríguez Cervantes |

Modelo de Dominio

Dentro de las actividades más importantes definidas en la metodología SXP se encuentra la definición del Modelo de Historias de Usuario del Negocio, en el cual se hace una detallada descripción del negocio en cuestión. Pero si dicho negocio no está bien definido entre los clientes y los ejecutores del proyecto; entonces es generado el llamado Modelo de Dominio:

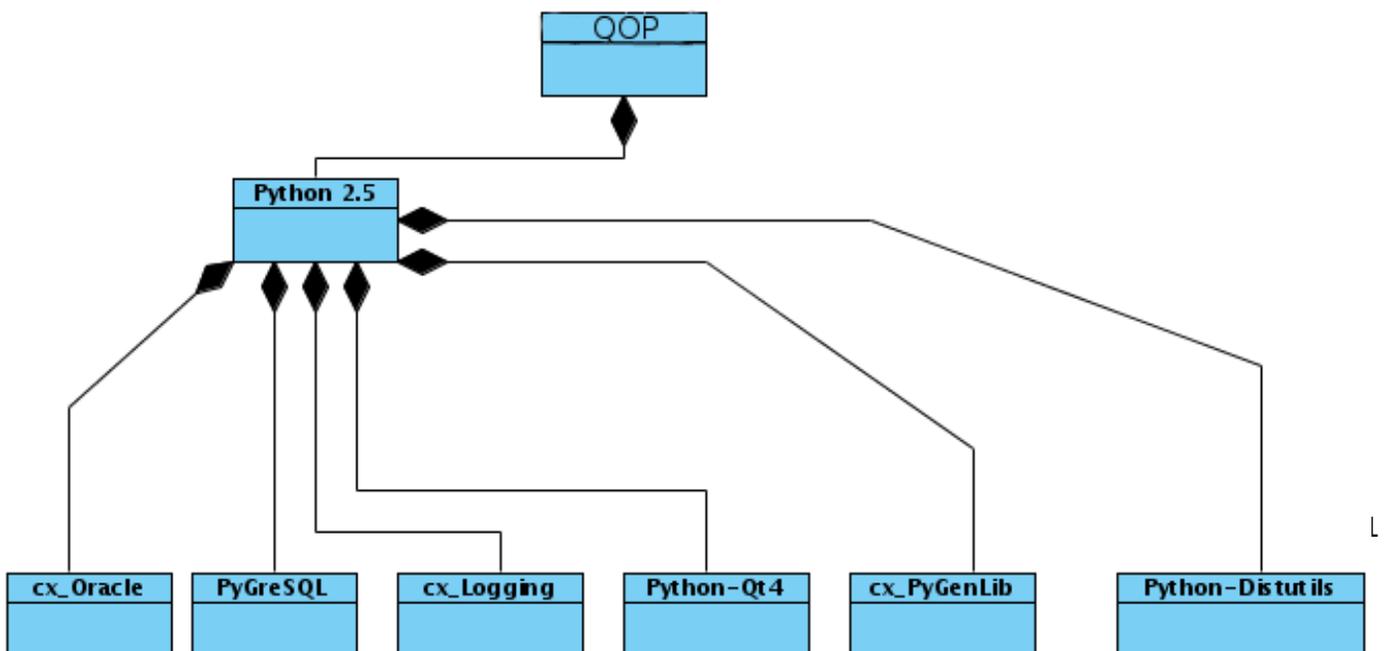


Figura.1 Modelo de Dominio de O2PMigration

Lista de Reserva del Producto(LRP)

Otra de las actividades más importantes definidas en la metodología SXP es la Lista de Reserva del Producto(LRP) , en la cual se recoge en una lista priorizada todo el trabajo a desarrollar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para una iteración.

Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Esta lista puede estar conformada por requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas. ([Ver Anexo 5](#))

Historias de usuarios y tareas de ingeniería.

A continuación se dan a conocer las distintas historias de usuarios que están presentes en el sistema; así como su relación con los prototipos de interfaz usuario y las tareas asociadas a cada historia de usuario; estas se desarrollan por la prioridad que tienen y por los usuarios encargados de las mismas. Ésta es solo una planificación inicial, el proceso es cambiante para ir adecuándolo a las necesidades y nuevas propuestas. Todas las decisiones se toman de conjunto con el cliente que es parte del equipo de desarrollo.

| Historia de Usuario | |
|----------------------------|---|
| Número: U_O2P_0.1 | Nombre Historia de Usuario: Conectar BD Oracle |

| | |
|--|------------------------------------|
| Modificación de Historia de Usuario Número: 1.0 | |
| Usuario: Marcos Ortíz Valmaseda | Iteración Asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 2 semanas |
| Riesgo en Desarrollo: Alto | Puntos Reales: 3 semanas |
| Descripción: La presente historia de usuario tiene como objetivo conectarse a la base de datos Oracle para poder lograr las tareas futuras. | |
| Observaciones: Debe llenar todos los campos, es decir entrar usuario, contraseña, id de la conexión, puerto y huésped, además de escribirlos correctamente. La PC donde se este trabajando debe tener instalado el cx_Oracle. | |
| Prototipos: Para ver este prototipo por favor (Ver anexo 2) | |

| | |
|---|--|
| Tarea de Ingeniería | |
| Número Tarea: 1 | Número Historia de Usuario: U_O2P_0.1 |
| Nombre Tarea: Estudio del módulo cx_Oracle | |

| | |
|--|-----------------------------------|
| Tipo de Tarea : Estudio | Puntos Estimados: 1 semana |
| Fecha Inicio: 20/10/08 | Fecha Fin: 27/10/08 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se profundizó en el estudio del módulo cx_Oracle para su utilización. | |
| Observación: Éste es el módulo que se utiliza en Python que permite el acceso a Oracle. | |

| Tarea de Ingeniería | |
|--|--|
| Número Tarea: 2 | Número Historia de Usuario: U_O2P_0.1 |
| Nombre Tarea: Desarrollo del módulo de conexión para Oracle | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |
| Fecha Inicio: 27/10/08 | Fecha Fin: 03/11/08 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se desarrollo el módulo de conexión a la base de dato Oracle utilizando cx_Oracle. | |

| Historia de Usuario | |
|--|--|
| Número: U_O2P_0.2 | Nombre Historia de Usuario: Conectar Base de Datos PostgreSQL |
| Modificación de Historia de Usuario Número: 1.0 | |
| Usuario: Marcos Ortíz Valmaseda | Iteración Asignada: 1 |
| Prioridad en Negocio: | Puntos Estimados: 1 semana |

| | |
|---|---------------------------------|
| Alta | |
| Riesgo en Desarrollo: | Puntos Reales: 2 semanas |
| Alto | |
| Descripción: La presente historia de usuario tiene como objetivo conectarse a la BD PostgreSQL para luego realizar las tareas futuras. | |
| Observaciones: Debe llenar todos los campos, es decir entrar usuario, contraseña, id de la conexión, puerto y huésped, además de escribirlos correctamente. Que esté instalado el PostgreSQL Server 8.2 o 8.3 o el PostgreSQL Client de las versiones anteriores | |
| Prototipos: Para ver este prototipo por favor(Ver anexo #3) | |

| | |
|--|--|
| Tarea de Ingeniería | |
| Número Tarea: 1 | Número Historia de Usuario: U_O2P_0.2 |
| Nombre Tarea: Estudio del módulo de conexión PostgreSQL | |
| Tipo de Tarea : Estudio | Puntos Estimados: 1/3 de semana |
| Fecha Inicio: 05/11/08 | Fecha Fin: 08/11/08 |
| Programador Responsable: Marcos Ortíz Valmaseda | |

| |
|--|
| Descripción: Estudio del módulo de conexión de Python para el desarrollo de la herramienta. |
| Observación: Este es el módulo que se utiliza en Python para el acceso a las base de datos de PostgreSQL. |

| Tarea de Ingeniería | |
|---|--|
| Número Tarea: 2 | Número Historia de Usuario: U_O2P_0.2 |
| Nombre Tarea: Desarrollar el módulo de conexión para PostgreSQL | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |
| Fecha Inicio: 08/11/08 | Fecha Fin: 15/11/08 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Desarrollo del módulo de conexión de Python PyGreSQL para el desarrollo de la herramienta O2PMigration. | |

| Historia de Usuario | |
|--|--|
| Número: U_O2P_0.3 | Nombre Historia de Usuario: Visualizar_datos_Oracle |
| Modificación de Historia de Usuario Número: 1.0 | |
| Usuario: Marcos Ortíz Valmaseda | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 semana |
| Riesgo en Desarrollo: | Puntos Reales: 2 semanas |

| | |
|---|--|
| Alto | |
| Descripción: La presente historia de usuario tiene como objetivo visualizar todos los datos de la BD Oracle. | |
| Observaciones: Debe mostrar todos los campos, es decir funciones, tablas, reglas, disparadores, procedimientos, grupos de usuarios y vistas. La PC donde se este trabajando debe tener instalado el cx_Oracle. | |
| Prototipos: Para ver este prototipo por favor (Ver Anexo # 4) | |

| | |
|---|--|
| Tarea de Ingeniería | |
| Número Tarea: 1 | Número Historia de Usuario: U_O2P_0.3 |
| Nombre Tarea: Desarrollo de las funcionalidades | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 de semana |
| Fecha Inicio: 06/01/09 | Fecha Fin: 13/01/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Desarrollo de las funcionalidades para la visualización de todos los datos de Oracle. | |

| |
|----------------------------|
| Tarea de Ingeniería |
|----------------------------|

| | |
|---|--|
| Número Tarea: 2 | Número Historia de Usuario: U_O2P_0.3 |
| Nombre Tarea: Mejoramiento de las funcionalidades | |
| Tipo de Tarea : Mejora | Puntos Estimados: ½ de semana |
| Fecha Inicio: 08/03/09 | Fecha Fin: 10/03/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se refinaron las funcionalidades de visualización de los datos de Oracle. | |

| Historia de Usuario | |
|--|--|
| Número: U_O2P_0.5 | Nombre Historia de Usuario: Exportar datos a PostgreSQL |
| Modificación de Historia de Usuario Número: 1.0 | |
| Usuario: Marcos Ortíz Valmaseda | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 2 semanas |
| Riesgo en Desarrollo: Alto | Puntos Reales: 3 semanas |
| Descripción: La presente historia de usuario guardar los datos Oracle en un archivo .sql. | |
| Observaciones: Debe guardar los datos como .sql para luego poder cargarlos en la base de datos PostgreSQL. | |

Que el directorio donde se exportaran los datos tengan los permisos pertinentes.

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 1 | Número Historia de Usuario: U_SD_0.5 |
| Nombre Tarea: Estudio del paquete Orafce | |
| Tipo de Tarea : Estudio | Puntos Estimados: 1 semana |
| Fecha Inicio: 01/12/08 | Fecha Fin: 08/12/08 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se estudió el funcionamiento del paquete Orafce el cual emula la mayoría de las funciones propias de Oracle. | |

| Tarea de Ingeniería | |
|--|---|
| Número Tarea: 2 | Número Historia de Usuario: U_SD_0.5 |
| Nombre Tarea: Desarrollo de las funcionalidades de Exportar a SQL | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |
| Fecha Inicio: 08/12/08 | Fecha Fin: 15/12/08 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se desarrolló las funcionalidades de exportar a SQL utilizando módulos propios de Python para la gestión de ficheros y comprensión de los mismos a un fichero .gzip. | |

| Historia de Usuario | |
|---|--|
| Número: U_O2P_0.6 | Nombre Historia de Usuario: Cargar datos a PostgreSQL |
| Modificación de Historia de Usuario Número: 1.0 | |
| Usuario: Marcos Ortíz Valmaseda | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 3 semanas |
| Riesgo en Desarrollo: Alto | Puntos Reales: ¾ de semana |
| Descripción: La presente historia de usuario toma los datos que se guardaron de la BD Oracle. | |
| Observaciones: Debe guardar los datos como un archivo .sql para luego poder cargarlos en la base de datos PostgreSQL. Que el directorio donde se exportarán los datos tengan los permisos pertinentes. | |
| Prototipos: Para ver este prototipo por favor (Ver Anexo # 4) | |

| Tarea de Ingeniería | |
|--|--|
| Número Tarea: 1 | Número Historia de Usuario: U_O2P_0.6 |
| Nombre Tarea: Estudiar métodos de conexión | |
| Tipo de Tarea : Estudio | Puntos Estimados: 1 semana |
| Fecha Inicio: 11/03/09 | Fecha Fin: 18/03/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se estudiaron todos los métodos de autenticación de PostgreSQL | |

| Tarea de Ingeniería | |
|--|--|
| Número Tarea: 2 | Número Historia de Usuario: U_O2P_0.6 |
| Nombre Tarea: Estudiar el funcionamiento del cliente de PostgreSQL | |
| Tipo de Tarea : Estudio | Puntos Estimados: 1 semana |
| Fecha Inicio: 18/03/09 | Fecha Fin: 25/03/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se estudió como trabaja de forma interna el cliente de PostgreSQL. | |

| Tarea de Ingeniería | |
|--|--|
| Número Tarea: 3 | Número Historia de Usuario: U_O2P_0.6 |
| Nombre Tarea: Desarrollo de las librerías correspondientes para la conexión a PostgreSQL. | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |

| | |
|---|----------------------------|
| Fecha Inicio: 25/03/09 | Fecha Fin: 02/04/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se desarrolló el módulo de conexión usando el módulo PyGreSQL bajo el estándar DATA BASE API 2.0 de Python. | |

| Historia de Usuario | |
|---|---|
| Número: U_O2P_0.4 | Nombre Historia de Usuario: Visualizar datos PostgreSQL. |
| Modificación de Historia de Usuario Número: 1.0 | |
| Usuario: Marcos Ortíz Valmaseda | Iteración Asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 3 semanas |
| Riesgo en Desarrollo: Alto | Puntos Reales: ¾ de semana |
| Descripción: La presente historia de usuario tiene como objetivo visualizar todos los datos de la BD PostgreSQL. | |
| Observaciones: Debe mostrar todos los campos, es decir funciones, tablas, reglas, disparadores, procedimientos, grupos de usuarios y vistas. La PC donde se vaya a llevar a cabo la migración debe tener instalado el PostgreSQL Server 8.2 o 8.3 o si se accede de forma remota el PostgreSQL Client 8.2 o 8.3. | |

Prototipos:

Para ver este prototipo por favor ([Ver Anexo #4](#))

| Tarea de Ingeniería | |
|--|--|
| Número Tarea: 1 | Número Historia de Usuario: U_O2P_0.4 |
| Nombre Tarea: Estudiar componentes de Python | |
| Tipo de Tarea : Estudio | Puntos Estimados: 1 semana |
| Fecha Inicio: 16/02/09 | Fecha Fin: 23/02/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Estudio del componente QTabWidget para poder cargar los datos en su contenido. | |

| Tarea de Ingeniería | |
|---|--|
| Número Tarea: 2 | Número Historia de Usuario: U_O2P_0.4 |
| Nombre Tarea: Desarrollo de las funcionalidades | |
| Tipo de Tarea : Desarrollo | Puntos Estimados: 1 semana |
| Fecha Inicio: 23/02/09 | Fecha Fin: 29/02/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Desarrollo de las funcionalidades para la visualización de todos estos datos. | |

| Tarea de Ingeniería | |
|---|--|
| Número Tarea: 3 | Número Historia de Usuario: U_O2P_0.4 |
| Nombre Tarea: Refinamiento de las funcionalidades. | |
| Tipo de Tarea : Mejora | Puntos Estimados: 1 semana |
| Fecha Inicio: 29/02/09 | Fecha Fin: 06/03/09 |
| Programador Responsable: Marcos Ortíz Valmaseda | |
| Descripción: Se refinan todas las funcionalidades de visualización de los datos de PostgreSQL. | |

Plan de Releases

| Release | Descripción de la iteración | Orden de la HU a implementar | Duración total |
|-------------------|--|--|---------------------|
| Iteración1 | En esta iteración se desarrollan las condiciones necesarias para Conectarse a las BD Oracle y PostgreSQL. | U_O2P_0.1 U_O2P_0.2 | 20/10/08 - 20/11/08 |
| Iteración2 | En esta iteración el sistema será capaz de Visualizar los datos de la BD de Oracle, además de exportar los datos a PostgreSQL. También se refinan las conexiones a las Bases de Datos. | U_O2P_0.1 U_O2P_0.2 U_O2P_0.4 U_O2P_0.5 | 20/11/08 - 15/02/09 |
| Iteración3 | En esta iteración se refinan lo que se | U_O2P_0.3 | 16/02/09 – 02/04/09 |

| | | | |
|-------------------|---|-------------------------------------|---------------------|
| | hizo en la segunda iteración, además de desarrollar como cargar los datos a PostgreSQL y visualizarlos. | U_O2P_0.5 U_O2P_0.6 U_O2P_0.4 | |
| Iteración4 | | | 04/04/09 - 20/05/09 |

Diseño con metáforas

Debido a que SXP está basada en XP, y dicha metodología define un término llamado metáfora, lo cual según Martin Fowler es una historia compartida que describe como debería funcionar el sistema y define que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema.

El Diseño con metáforas es sencillamente el diseño de la solución más simple que pueda funcionar y ser implementado en un momento dado del proyecto; lo cual genera el artefacto conocido como Modelo de Diseño, que a su vez está compuesto por un diagrama de paquetes, el cual expone dicho diseño.

Los diagramas de paquetes describen los elementos físicos del sistema y sus relaciones. Muestra las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados. Estos muestran además la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, librerías, binarios o ejecutables.

A continuación se representa el diagrama de paquetes para el sistema que se propone.

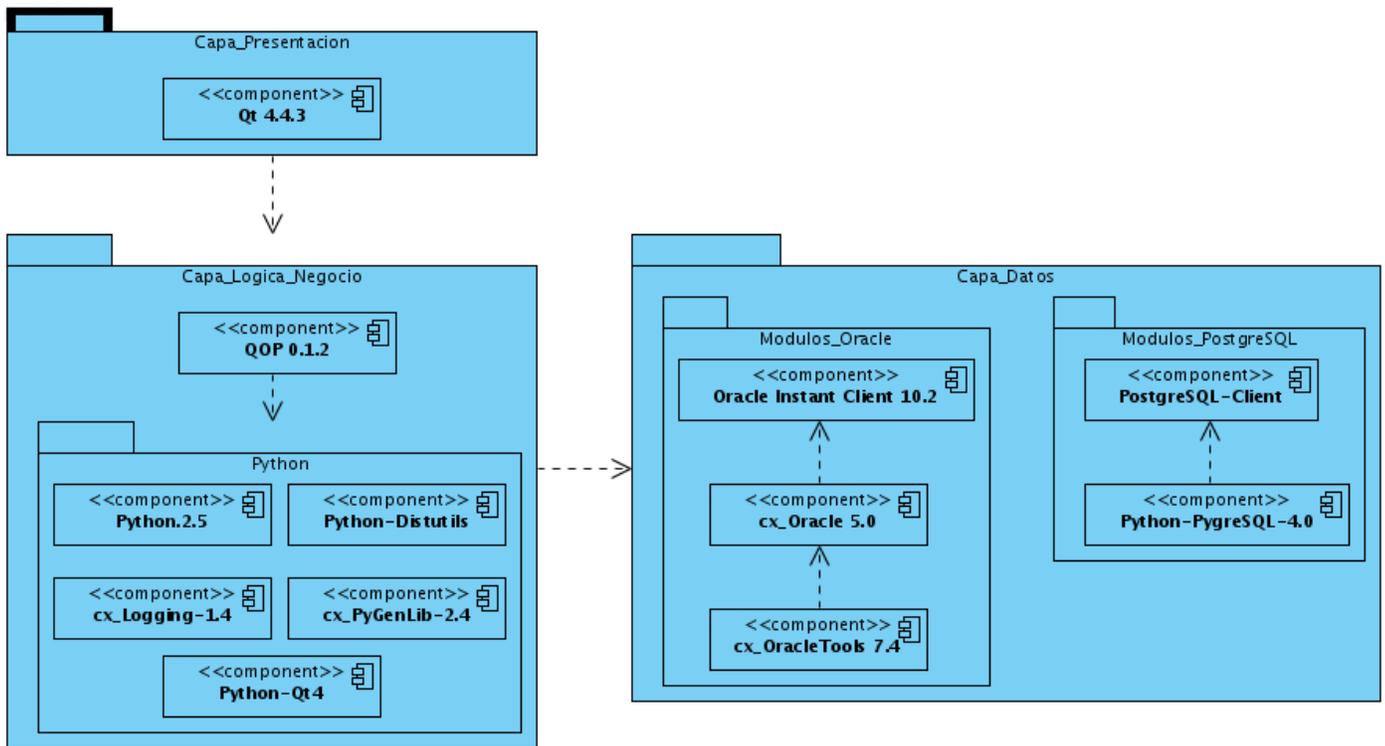


Figura 2 Diagrama de paquetes de O2PMigration

Descripción:

En la **capa de presentación** se tiene el componente Qt 4.4.3, el cual es el framework gráfico que se utiliza en el desarrollo de la interfaz visual de la aplicación. Esta capa de presentación depende de la capa de lógica de negocio para mostrar los datos.

En la **capa de lógica de negocio** se encuentra el componente QOP 0.1.2, que no es más que el módulo principal de la aplicación, el cual engloba todo el funcionamiento del negocio; éste componente necesita del paquete de Python para su desarrollo.

Dentro del paquete de Python encontramos el componente Python 2.5 que es el lenguaje sobre el que se desarrolla QOP 0.1.2.

También se encuentra el componente Python- Distutils, el cual es el módulo que se utiliza para la

construcción del instalador de QOP y para instalar el módulo cx_Logging y la librería cx-PyGebLib. Además se puede encontrar el componente Python-Qt 4 que no es más que el módulo que me permite utilizar el framework Qt 4.4.3 desde Python. Esta capa depende de la capa de Datos.

La **capa de Datos** está constituida por los paquetes Módulos de Oracle y Módulos de PostgreSQL. Dentro del paquete de Módulos de Oracle se tiene el módulo cx_OracleTools 7.4, que depende del módulo de extensión cx_Oracle que permite el acceso a Oracle y se ajusta a las especificaciones de la API de base de datos en Python, este depende de la librería Oracle Instant Client 10.2 que permite el establecimiento de las conexiones con Oracle.

Dentro del paquete Módulos de PostgreSQL se encuentra Python-PyGreSQL-4.0 que es el módulo de Python utilizado para la conexión a la base de datos PostgreSQL, el mismo depende del componente PostgreSQL-Client que no es más que el cliente del gestor de bases de datos PostgreSQL.

Capítulo 4. Validación de la solución propuesta

En el presente capítulo se exponen los casos de pruebas o test de aceptación que se le realizaron al sistema; éstas fueron realizadas en cada una de las iteraciones, las cuales fueron necesarias para avanzar hacia la próxima iteración ya que para lograr un producto con calidad es necesario implementar un plan de pruebas desde el principio, y así darle seguimiento a los cambios y desarrollar iterativamente. En este capítulo además de las pruebas se dan a conocer los resultados obtenidos hasta el momento.

Casos de Pruebas

La programación extrema define entre iteración e iteración un conjunto de casos de pruebas o test de aceptación para poder avanzar a una iteración superior. Durante el desarrollo de O2PMigration se realizaron un conjunto de prueba a las que fue sometido el sistema para comprobar el funcionamiento de acuerdo a las Historias de Usuario. Se confeccionó un Plan de Prueba para organizar el desarrollo de las mismas; para más información ([Ver Anexo # 7](#)).

Se definieron casos de prueba para todas las historias de usuario, a continuación se dan a conocer las pruebas que se realizaron a cada una de las historias de usuario con las que cuanta el sistema.

Casos de pruebas para la Historia de usuario: U_O2P_0.1

Esta sección cubre el conjunto de pruebas funcionales que se le realizan a la historia de usuario: Conectar_BD_Oracle.

En esta historia de usuarios se intenta probar que los datos entrados estén correctamente y que el usuario tenga los permisos requerido para conectarse a la Base de datos Oracle solicitada.

Además se verificará que el huésped de Oracle esté funcionando correctamente.

Caso de Prueba de Aceptación

| | |
|--|---|
| Código Caso de Prueba: U_O2P_1_1 | Nombre Historia de Usuario: Conectar BD Oracle |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Al iniciar cada cliente debe llenar los campos correspondientes para la conexión a la base de dato Oracle para esto se realizarán las siguientes pruebas: Se entrarán datos incorrectos para comprobar que no se conecten usuarios no deseados. Se entrarán los datos correctamente para comprobar que se conecte a la base de datos deseada. | |
| Condiciones de Ejecución: Debe tener instalado el CX-Oracle. | |
| Entrada / Pasos de ejecución: Se entra usuario, contraseña, id de la conexión, puerto y huésped de la Base de Datos que se desea entrar y se pulsa el botón Conectar para realizar la conexión. | |
| Resultado Esperado: Que se pueda tener acceso a la Base de Datos deseada. | |
| Evaluación de la Prueba: Prueba Satisfactoria | |

Casos de pruebas para la Historia de usuario: U_O2P_0.2

Esta sección cubre el conjunto de pruebas funcionales que se le realizan a la historia de usuario: Conectar_BD_PostgreSQL.

En esta historia de usuario se intenta probar que los datos entrados estén correctamente y que el usuario tenga los permisos requeridos para conectarse a la base de datos solicitada.

Además se verificará si el huésped de PostgreSQL está funcionando correctamente.

| Caso de Prueba de Aceptación | |
|--|---|
| Código Caso de Prueba: U_O2P_2_1 | Nombre Historia de Usuario: Conectar BD PostgreSQL |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Al iniciar cada cliente debe llenar los campos correspondientes para la conexión a la base de dato PostgreSQL para esto se realizarán las siguientes pruebas: Se entrarán datos incorrectos para comprobar que no se conecten usuarios no deseados. Se entrarán los datos correctamente para comprobar que se conecte a la base de datos deseada. | |

| |
|--|
| Condiciones de Ejecución: Debe tener instalado del PostgreSQL 8.2 en adelante. |
| Entrada / Pasos de ejecución: Se entra usuario, contraseña, id de la conexión, puerto y huésped de la Base de Datos que se desea entrar y se pulsa el botón conectar para realizar la conexión. |
| Resultado Esperado: Que se pueda tener acceso a la Base de Datos deseada. |
| Evaluación de la Prueba: Prueba Satisfactoria |

| Caso de Prueba de Aceptación | |
|---|---|
| Código Caso de Prueba: U_O2P_2_2 | Nombre Historia de Usuario: Conectar BD PostgreSQL |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Después de comprobar que se realice una buena conexión se procederá a comprobar que el huésped de PostgreSQL esté funcionando correctamente. | |
| Condiciones de Ejecución: Se necesita estar conectado a la base de dato deseada. | |
| Entrada / Pasos de ejecución: Para esto es necesario verificar que el atributo connection de los parámetros de conexión este activado. | |
| Resultado Esperado: Que se conozca el verdadero estado del huésped al que se esté conectando. | |
| Evaluación de la Prueba: Prueba Satisfactoria | |

Casos de pruebas para la Historia de usuario: U_O2P_0.3

Esta sección cubre el conjunto de pruebas funcionales que se le realizan a la historia de usuario: Visualizar_datos_Oracle.

En esta historia de usuario se intenta probar que se muestren todos los campos correctamente.

| Caso de Prueba de Aceptación | |
|--|--|
| Código Caso de Prueba: U_O2P_3_1 | Nombre Historia de Usuario: Visualizar datos Oracle |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Se comprobará que se muestren en el QListView los datos de la base de datos Oracle que se está conectado. | |

| |
|---|
| Condiciones de Ejecución: Se necesita estar conectado a la base de dato deseada y que la misma tenga el atributo connection activo. |
| Entrada / Pasos de ejecución: Para esto es necesario verificar que se muestren correctamente las funciones, tablas y paquetes, reglas, procedimientos, grupos de usuarios, índices, disparadores y las vistas de la base de datos que se está conectado. |
| Resultado Esperado: Que se muestren correctamente los datos de la base de datos a la que se está conectado. |
| Evaluación de la Prueba: Prueba Satisfactoria |

Casos de pruebas para la Historia de usuario: U_O2P_0.4

Esta sección cubre el conjunto de pruebas funcionales que se le realizan a la historia de usuario: Visualizar_datos_PostgreSQL.

En esta historia de usuario se intenta probar que se muestren todos los campos correctamente.

| Caso de Prueba de Aceptación | |
|--|--|
| Código Caso de Prueba: U_O2P_4_1 | Nombre Historia de Usuario: Visualizar datos PostgreSQL |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Se comprobará que se muestra en el QListView los datos de la base de datos PostgreSQL que se está conectado. | |
| Condiciones de Ejecución: Se necesita estar conectado a la base de dato deseada y que la misma tenga el atributo connection activo. | |
| Entrada / Pasos de ejecución: Para esto es necesario verificar que se muestren correctamente las funciones, tablas y paquetes, reglas, procedimientos, grupos de usuarios, índices, disparadores y las vistas de la base de dato que se está conectado. | |
| Resultado Esperado: Que se muestren correctamente los datos de la base de dato a la que se está conectado. | |
| Evaluación de la Prueba: Prueba Satisfactoria | |

Casos de pruebas para la Historia de usuario: U_O2P_0.5

Esta sección cubre el conjunto de pruebas funcionales que se le realizan a la historia de usuario: Exportar_datos_a_PostgreSQL.

En esta historia de usuario se intenta probar que el archivo que se intenta exportar se guarde con la extensión .sql.

También se comprobará la codificación de la Base de Datos PostgreSQL.

| Caso de Prueba de Aceptación | |
|---|--|
| Código Caso de Prueba: U_O2P_5_1 | Nombre Historia de Usuario: Exportar datos a PostgreSQL |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Se comprobará que el archivo que se esté exportando se guarde en .sql. | |
| Condiciones de Ejecución: Se necesita estar conectado a la base de datos deseada y que la misma tenga el atributo connection activo. | |
| Entrada / Pasos de ejecución: Para esto es necesario que se busque el archivo que se guardó y luego se verificará que la extensión sea .sql. | |
| Resultado Esperado: Que el archivo exportado a PostgreSQL se guarde con la extensión solicitada. | |
| Evaluación de la Prueba: Prueba Satisfactoria | |

| Caso de Prueba de Aceptación | |
|--|--|
| Código Caso de Prueba: U_O2P_5_2 | Nombre Historia de Usuario: Exportar datos a PostgreSQL |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Después de verificar que el archivo exportado se guarde con la extensión requerida se procederá a verificar el parámetro encoding de la base de datos PostgreSQL. | |
| Condiciones de Ejecución: Se necesita estar conectado a la base de datos deseada y que la misma | |

| |
|---|
| tenga el atributo connection activado. |
| Entrada / Pasos de ejecución: Para esto es necesario verificar que el encoding de la base de datos PostgreSQL sea UTF8, en caso de no estarlo se cambiará el mismo a UTF8. |
| Resultado Esperado: Que el encoding de la Base de datos PostgreSQL sea compatible con el encoding de la Base de Datos Oracle que se exportó. |
| Evaluación de la Prueba: Prueba Satisfactoria |

Casos de pruebas para la Historia de usuario: U_O2P_0.6

Esta sección cubre el conjunto de pruebas funcionales que se le realizan a la historia de usuario: Cargar_datos_a_PostgreSQL.

En esta historia de usuario se intenta probar que la conexión a la Base de Datos PostgreSQL este funcionando.

| Caso de Prueba de Aceptación | |
|--|--|
| Código Caso de Prueba: U_O2P_6_1 | Nombre Historia de Usuario: Cargar datos a PostgreSQL |
| Nombre de la persona que realiza la prueba: Marcos Ortíz Valmaseda | |
| Descripción de la Prueba: Verificar la conexión a la base de datos PostgreSQL a la que se desean cargar los datos. | |
| Condiciones de Ejecución: Se necesita estar conectado a la base de dato deseada y que la misma tenga el atributo connection activo. | |
| Entrada / Pasos de ejecución: Para esto es necesario verificar que el atributo connection de la base de datos PostgreSQL este activo. | |
| Resultado Esperado: Que se carguen los datos la Base de Datos PostgreSQL correctamente. | |
| Evaluación de la Prueba: Prueba Satisfactoria | |

Resultados Obtenidos

En este epígrafe se dan a conocer los resultados obtenidos hasta el momento por el equipo de desarrollo de O2PMigration. Resaltar que como resultado de este trabajo O2PMigration está disponible en su versión 0.1.4, lo cual significa que se pueden esperar resultados en versiones posteriores.

Acerca del tiempo de desarrollo

La migración de las aplicaciones que usan Oracle en ETECSA es una problemática que venía dada desde enero del año 2008, por la necesidad de una efectiva migración a PostgreSQL, por las razones ya expuestas anteriormente; por lo que se comenzó a dar los primeros pasos en febrero de este mismo año. Se comenzó realizando labores de investigación para ir tomando conocimientos sobre este tema.

En abril del 2008 se comenzaron a estudiar las herramientas, lenguajes y tecnologías necesarias para el desarrollo de esta aplicación. Aprender a trabajar con estas herramientas, aprender PostgreSQL y Python tomó aproximadamente tres meses. Luego se instaló lo necesario para empezar a desarrollar a finales del mes de mayo. Posteriormente se comenzó a realizar el análisis y diseño del sistema, donde se hizo el levantamiento de requisitos y se sacaron las historias de usuario, esta tarea demoró aproximadamente 2 meses.

En septiembre del 2008 se empezaron a dar los primeros pasos en el desarrollo de O2PMigration ya que se comienzan a implementar un grupo de tareas y en febrero del 2009 se obtiene la primera versión de O2Pmigration. El 19 de marzo del 2009 se realizó el lanzamiento de la primera versión estable la versión 0.1.2.

Si se tiene en cuenta que el equipo de desarrollo no tenía experiencia previa de los elementos y conceptos más importantes para desarrollar un sistema para la migración de Bases de Datos de Oracle a PostgreSQL, el período de tiempo empleado fue relativamente corto y se superaron las expectativas de la planificación inicial.

Acerca de las funcionalidades obtenidas.

O2PMigration con varias funcionalidades que constituyen la esencia de la misma, con las cuales se hace un pequeño resumen atendiendo a la necesidad que existe de conocer el funcionamiento interno de la aplicación:

| Nombre | Descripción |
|---------------------|---|
| get_data() | Devuelve todos los datos de la base de datos Oracle en un diccionario simple dependiendo del tipo que se escoja(COPY o INSERT) |
| get_sql_data() | Devuelve la definición de la base de datos en una cadena (TABLES, PACKAGES, TABLESPACES, FUNTIONS, PROCEDURES, TRIGGERS, VIEWS) |
| export_data() | Exporta los datos a un archivo .sql dando la posibilidad de comprimirlo o no |
| oracle_connection() | Establece la conexión con la base de datos Oracle a través de cx_Oracle |
| pg_connection() | Establece la conexión con la base de datos PostgreSQL a través de PyGreSQL |
| view_all() | Muestra toda la información de las bases de datos en la interfaz de usuario (GUI) |

Todas las funcionalidades están documentadas con el estilo propio del lenguaje Python. Dicho estilo sigue el siguiente formato:

```
""" Aquí va el comentario acerca de la función siguiendo el estándar de Python
"""
```

@summary: Resumen de la funcionalidad (¿Qué hace?)

@author: Autor de la misma

@params: Argumentos de la función

@require: Módulos o funcionalidades requeridas para el correcto funcionamiento de la misma

@return: Valores que retorna la función

Conclusiones

A partir de los objetivos planteados y el trabajo realizado en esta investigación donde se desarrolló una herramienta basada en la migración de las bases de datos Oracle se arribó a los siguientes resultados:

- Se realizó un estudio detallado de las aplicaciones utilizadas para la migración de bases de datos Oracle existentes, así como sus características y funcionamiento.
- Se diseñó una herramienta que permite la correcta y eficiente migración de las bases de datos de las aplicaciones empresariales de ETECSA implementadas con Oracle hacia PostgreSQL.
- Se implementó dicha herramienta.

Por todo lo anteriormente expuesto se concluye que los objetivos trazados para el presente trabajo se han cumplido satisfactoriamente.

Recomendaciones

- Implementar nuevas funcionalidades para mejorar la calidad y robustez del producto.
- Validar el funcionamiento de la herramienta a gran escala.
- Implementar nuevas funcionalidades que permitan la migración no solo de bases de datos Oracle a PostgreSQL, sino que se pueda construir la base de una suite completa de migración a PostgreSQL.
- Llevar el producto en futuras versiones a modo texto.
- Que la herramienta sea plurilingüe.

Referencias Bibliográficas

- [1] MySQL, *MySQL Server 5.1 Generally Available (GA)* [citado el: 18 February 2009] Available from world wide web : <http://www.mysql.com/dev/>
- [2] Wikipedia.org, 1997. Wikipedia. Wikipedia.org [En línea] [Citado el: 20 de mayo del 2008] Available from world wide: http://es.wikipedia.org/wiki/Base_de_datos
- [3] Monografias.com S.A, 1997. Monografías. Monografias.com [En línea] [Citado el: 9 de mayo de 2008] Available from world wide: <http://www.monografias.com/trabajos35/comparativa-bases-datos/comparativa-bases-datos.shtml>
- [4] Free Software Foundation, 2001, Gnu, Gnu.org [En línea] [Citado el: 9 de mayo del 2008] Available from world wide: <http://www.gnu.org/philosophy/free-sw.es.html>
- [5] Free Software Foundation, 2001, Gnu, *El sistema operativo GNU* [En línea] [Citado el: 9 de mayo del 2008] Available from world wide: <http://www.gnu.org/philosophy/free-sw.es.html>
- [6] Ídem [5]
- [7] Sitio oficial de *Eclipse* [En línea] Available from world wide: <http://www.eclipse.org/>
- [8] Daniel F.A., Junio 2007, *Administración, configuración y optimización de un Sistema de Bases de Datos Descentralizado en Oracle Database 10g Release 2* [citado el: 19 November 2008] Available from world wide: http://bibliodoc.uci.cu/TD/TD_0289_07.pdf
- [9] The PostgreSQL Global Development Group, 2005, *PostgreSQL*, PostgreSQL.org [En línea] [Citado el:

9 de mayo del 2008] Available from world wide: <http://www.postgresql.org/docs/8.0/interactive/index.html>

[10] The Python Organization ,2008, *Python*. Python Org [En línea] [Citado el 4 de mayo de 2008.] Available from world wide web: <http://www.python.org>

[11] Sitio de *cx_Oracle* [En línea] [Citado el 6 de enero del 2009] Available from world wide web: <http://www.cxtools.net/default.aspx?nav=cxorlb>

[12] Sitio Oficial de *PyGreSQL* [En línea] [Citado el: 9 de mayo del 2008] Available from world wide: <http://www.pygresql.org/>

[13] *Tutorial de Qt 4.3* [En línea] Available from world wide web: <http://doc.trolltech.com/4.3/tutorial.html>

[14] Gladys Marsi, P. R., 2008, *MA-GMPUR2 Metodología ágil para proyectos de software libre*, Universidad de las Ciencias Informáticas, Facultad 10 [citado el: 21 February 2009] Available from world wide: http://bibliodoc.uci.cu/TD/TD_0693_07.pdf

[15] Ídem [14]

[16] PostgreSQL Global Development Group, 2008, *PostgreSQL 8.3.6 Release Notes* [citado el: 20 November 2008] Available from world wide web: <http://www.postgresql.org/docs/8.3/static/release-8-3.html>

[17] EnterpriseDB Corporation, *Open Source Database Products* [citado el: 24 February 2009] Available from world wide web: <http://www.enterprisedb.com/products>

[18] EnterpriseDB Corporation, *Database Subscription Pricing* [citado el: 24 February 2009] Available from world wide web: <http://www.enterprisedb.com/products/purchase.do>

[19] PostgreSQL Global Development Group, 2008, *PostgreSQL 8.3 Release Notes* [citado el: 17 February 2009] Available from world wide web: <http://www.postgresql.org/about/press/presskit83>

[20] PostgreSQL Global Development Group, 2 de abril del 2008, *PostgreSQL 8.3.0 Documentation.PGDG*. actualizado: 2 de febrero del 2009 [citado el: 17 February 2009] Available from world wide web: <http://www.postgresql.org/files/documentation/pdf/8.3/postgresql-8.3-US.pdf>

[21] Raúl, G.D., 2008, *Python para todos* [citado el: 18 February 2009] Available from world wide web: <http://edge.launchpad.net/improve-python-spanish-doc/0.4/0.4.0/+download/Python%20para%20todos.pdf>

[22] Daniel, B.F., *Revista Python 2. Google App Engine. El servidor de aplicaciones que utiliza PYTHON. MUNDO PYTHONM* [citado el: 18 February 2009] Available from world wide web: <http://www.scribd.com/doc/8690133/Revista-Python-2/>

[23] Guido,V.R., PEP. 2008, 249 -- *Python Database API Specification v2.0* [citado el: 16 February 2009] Available from world wide web: <http://www.python.org/dev/peps/pep-0249>

Bibliografía

DML, SQL y los comandos DML [en línea] [Citado el 5 de junio de 2008] Available from world wide web: http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sql_y_los_comandos_dml.php

Documentación Oficial de Qt [citado el: 17 February 2009] Available from world wide web: <http://doc.trolltech.com>

Ercoli, Jorge Desarrollador 5 Estrellas .Net "Qué es un ORM". ORM [En línea] Available from world wide web: <http://metodologiasdesistemas.blogspot.com/2007/10/que-es-un-orm-object-relational-mapping.html>

Free Software Foundation, Inc. *El sistema Operativo GNU* [citado el: 9 May 2008.] Available from world wide web: <http://www.gnu.org/philosophy/philosophy.es.html#LicensingFreeSoftware>

Free Software Foundation, Inc 2001, Gnu. *Gnu.org* [En Línea] [Citado el: 9 de mayo de 2008.] Available from world wide web: <http://www.gnu.org/philosophy/free-sw.es.html>

James Shannon, Ben Adida, and Don Baccus, *Oracle to Postgres Conversion*. openacs.org [En línea] [Citado el: 9 de mayo de 2008.] Available from world wide web: <http://openacs.org/doc/openacs-3/html/oracle-to-pg-porting.html>.

Monografias.com S.A. , 1997. Monografías. Monografias.com [En línea] [Citado el: 9 de mayo de 2008.] Available from world wide web: <http://www.monografias.com/trabajos35/comparativa-bases-datos/comparativa-bases-datos.shtml>

MySQL, 16 de octubre del 2006, MySQL Hispano. *mysql-hispano.org* [En línea] [Citado el: 4 de mayo de

2008.] Available from world wide web: <http://www.mysql-hispano.org/index.php?m=read&id=974>

MySQL AB., 2008, MySQL. MySQL.com [En línea] [Citado el: 4 de mayo de 2008.] Available from world wide web: <http://dev.mysql.com/doc/refman/5.0/es/index.html>

Ora2Pg [En línea] [Citado el 6 de junio de 2008] Available from world wide web: <http://pgfoundry.org/projects/ora2pg/>

Oracle Developer Community, 2008, Oracle Corporation. *Oracle* [En línea] [Citado el: 3 de mayo de 2008.] Available from world wide web: <http://www.oracle.com/global/es/index.html>

PgFoundry is the PostgreSQL Development Group's, *PostgreSQL PgFoundry* [En Línea] [Citado el 6 de junio de 2008] Available from world wide web: <http://www.pgfoundry.org>

PostgreSQL Fujitsu Support, *Fujitsu PostgreSQL* [En línea] [Citado el 4 de junio de 2008]

Qt 4.5 Preview Release [En línea] Available from world wide web: <http://trolltech.com/about/news/qt-4.5-tech-preview-released>

Sitio oficial de *Eclipse* [En línea] Available from world wide web: <http://www.eclipse.org>

Sitio de *cx_Oracle* [En línea] [Citado el 6 de enero del 2009] Available from world wide web: <http://www.cxtools.net/default.aspx?nav=cxorb>

Sitio de *Python* [En línea] [Citado el 17 de febrero del 2009] Available from world wide web: <http://www.python.org/download/releases/2.5/highlights/>

The PostgreSQL Company since 1997, *Command Prompt Mammoth PostgreSQL, PostgreSQL Support,*

Consulting and Development [citado el: 4 June 2008] Available from world wide web:

<http://commandprompt.com/>

The PostgreSQL Global Development Group, 2005, *PostgreSQL. PostgreSQL.org* [En línea] [Citado el: 9 de mayo de 2008.] Available from world wide web:

<http://www.postgresql.org/docs/8.0/interactive/index.html>

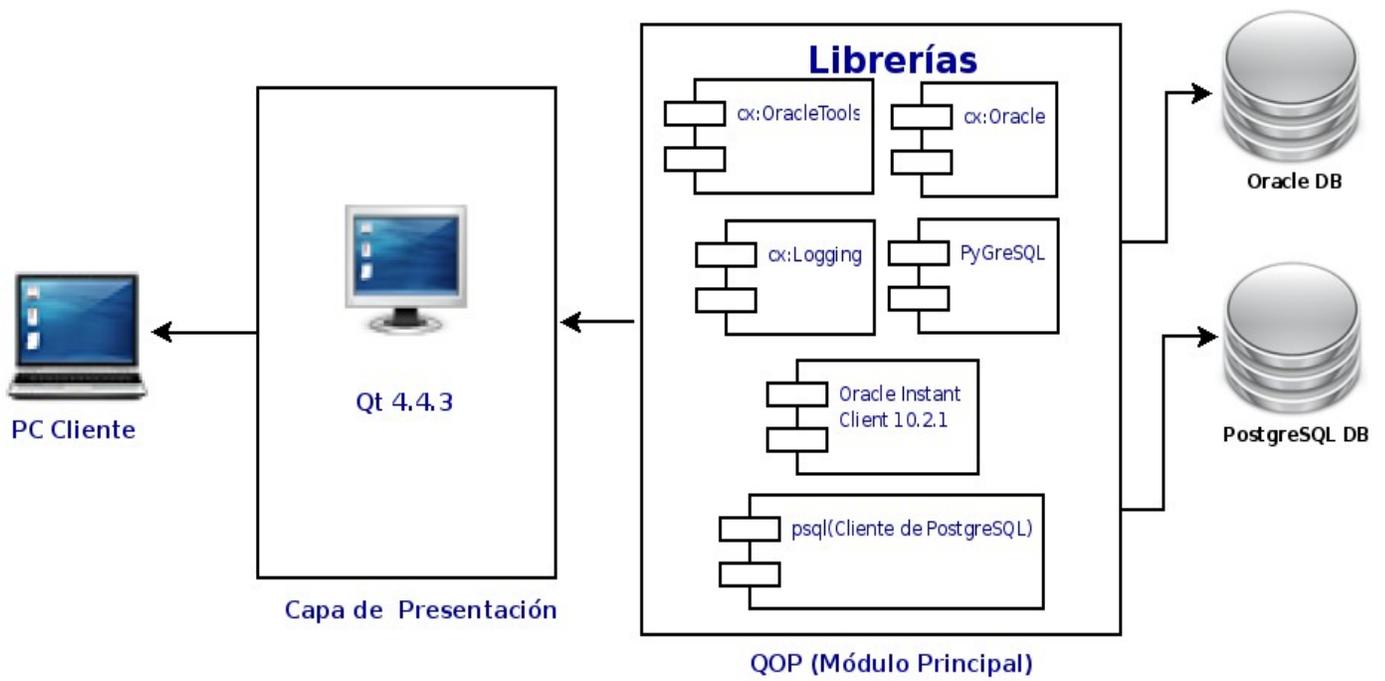
The Python Organization ,2008, *Python Org. Python* [En línea] [Citado el 4 de mayo de 2008.] Available from world wide web: <http://www.python.org>

The Python SQL Toolkit and Object Relational Mapper, 2008, SQLAlchemy Org. SQLAlchemy [En línea] [Citado el 8 de junio de 2008] Available from world wide web: <http://www.sqlalchemy.org/>

Tutorial de Qt 4.3 [En línea] Available from world wide web: <http://doc.trolltech.com/4.3/tutorial.html>

Anexos

Anexo # 1 Arquitectura base de O2PMigration



Anexo #2 Prototipo de la Historia de usuario Conectar_BD_Oracle

Conexion a Oracle - oracle_con.ui

Conexion Opciones



Autenticación

Login

Contraseña

Guardar Contraseña

Huésped

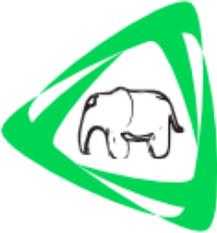
Puerto

SID

OK Cancel

Anexo #3 Prototipo de la Historia de usuario Conectar_BD_PostgreSQL

Conexión a PostgreSQL - pg_conn.ui



Autenticación

Login

Contraseña

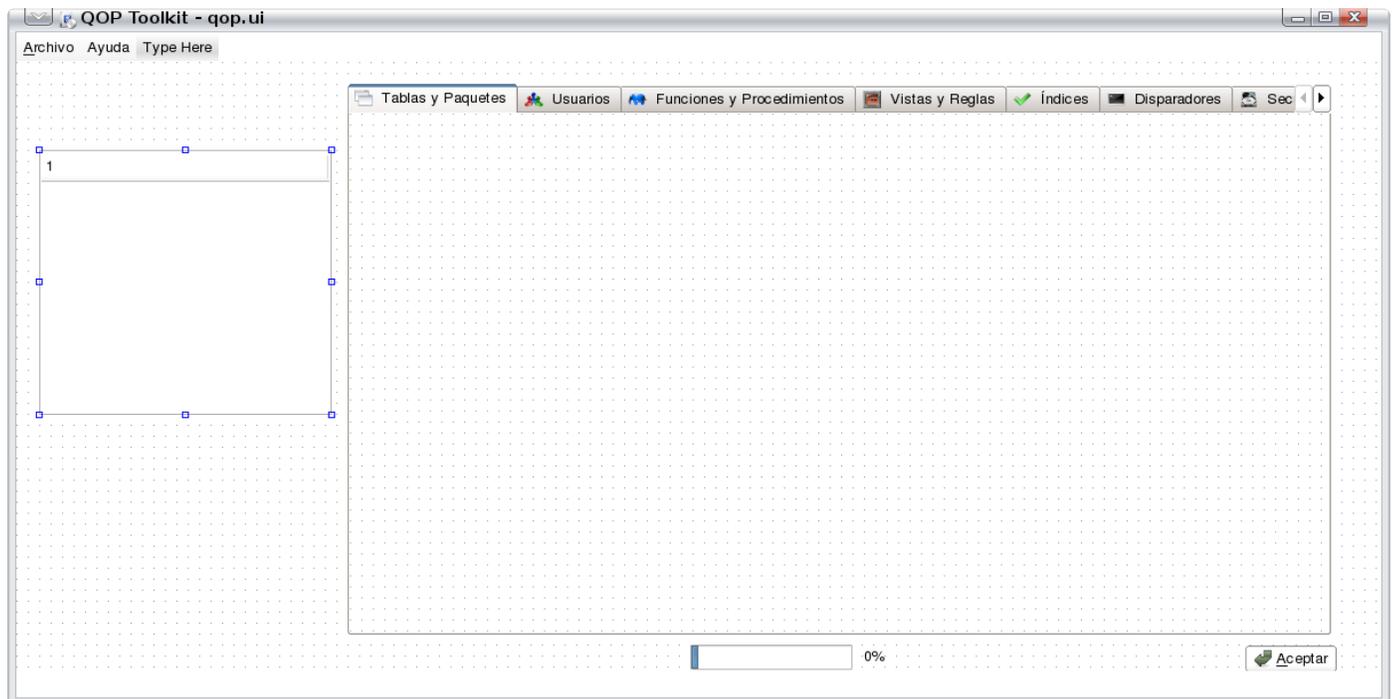
Guardar Contraseña

Huésped

Puerto

OK Cancel

Anexo #4 Prototipo de las Historias de usuario Visualizar



Anexo #5 Lista de Reserva del Producto(LRP)

| Asignado a | Ítem * | Descripción | Estimación | Estimado por |
|------------------|--------|---|----------------------|--------------|
| Prioridad | | Muy Alta | | |
| Marcos | X | <i>Estudio de herramientas basadas en software libre para la migración de bases de datos de Oracle a PostgreSQL</i> | 08/12/08 19/12/08 | al Marcos |
| Susany | 1 | <i>Estudio de arte del proyecto</i> | 11/10/08 19/12/08 | al Susany |
| Marcos | 2 | <i>Estudio de la API de Python 2.5 (Módulos re,sys,os y gzip)</i> | 4/01/09 15/02/09 | al Marcos |
| Marcos | 3 | <i>Introducción del trabajo de diploma</i> | 08/12/08 19/12/08 | al Marcos |
| Susany | 4 | <i>Diseño teórico-metodológico del Trabajo de Diploma</i> | 11/10/08 19/12/08 | al Susany |
| Prioridad | | Alta | | |
| Marcos | 5 | <i>Conectar a Oracle</i> | 9/10/08 20/01/09 | al Marcos |
| Marcos | 6 | <i>Conectar a PostgreSQL</i> | 9/10/08 20/01/09 | al Marcos |
| Marcos | 7 | <i>Definición de la interfaz de la aplicación en Qt</i> | 9/10/08 20/01/09 | al Marcos |
| Marcos | 8 | <i>Visualizar datos de PostgreSQL</i> | 9/10/08 20/01/09 | al Marcos |
| Marcos | 9 | <i>Visualizar datos de Oracle</i> | 9/10/08 20/01/09 | al Marcos |
| Prioridad | | Media | | |
| Susany | 10 | <i>Requerimientos del sistema</i> | 11/10/08 | al Marcos |

| | | | | |
|--|-----------|---|-----------------------------|---------------|
| | | | 19/12/08 | |
| Prioridad | | | Baja | |
| Marcos-Susany | 11 | Manual de Usuario de O2PMigration | 20/12/08 al 20/05/09 | |
| Susany | 12 | Elaborar LRP | 11/10/08 al 19/12/08 | Marcos |
| Susany | 13 | Manual de Identidad | | Marcos |
| RNF (Requisitos No Funcionales) | | | | |
| Marcos | 14 | Instalación de las herramientas necesarias para O2PMigration | 11/10/08 al 19/12/08 | Marcos |

Anexo # 6 Instalación del módulo cx_Oracle

El módulo cx_Oracle es necesario para el funcionamiento de la aplicación por lo que se debe explicar como se instala desde las fuentes, las cuales son distribuidas con el paquete en sí.

Para realizar la instalación efectiva del módulo se deben hacer algunas configuraciones previas que se irán explicando detalladamente. Este módulo necesita como se ha explicado anteriormente, depende del Oracle Instant Client, el cual hay que descargarlo de su URL¹ oficial del sitio de la compañía dada, ya que no se puede distribuir con el paquete de O2PMigration por las restricciones impuestas en la licencia de uso del mismo.

Para la compilación del módulo se necesitan dos paquetes: Oracle Instant Client Basic Lite y el Oracle Instant Client SDK, los cuales necesitan que se sea un usuario registrado de Oracle para efectuar su

¹ La URL de descarga es <http://www.oracle.com/technology/tech/instantclient/index.html>

descarga.

Luego de que se cuente con dichos paquetes, se procede a la instalación de los mismos en el sistema.

Lo primero que hay que hacer es descomprimir los mismos en un directorio dado:

```
root@Debian-Server:~# unzip basiclite-10.2.0.4.0-linux-x86.zip
```

```
Archive: basiclite-10.2.0.4.0-linux-x86.zip
```

```
inflating: instantclient_10_2/BASIC_LITE_README
```

```
[...]
```

```
inflating: instantclient_10_2/libocijdbc10.so
```

```
inflating: instantclient_10_2/ojdbc14.jar
```

```
root@Debian-Server:~# unzip sdk-10.2.0.4.0-linux-x86.zip
```

```
Archive: sdk-10.2.0.4.0-linux-x86.zip
```

```
creating: instantclient_10_2/sdk/
```

```
creating: instantclient_10_2/sdk/include/
```

```
inflating: instantclient_10_2/sdk/include/occi.h
```

```
inflating: instantclient_10_2/sdk/include/occiCommon.h
```

```
[...]
```

```
inflating: instantclient_10_2/sdk/SDK_README
```

```
extracting: instantclient_10_2/sdk/ottclasses.zip
```

```
inflating: instantclient_10_2/sdk/ott
```

Al ser descomprimidos estos paquetes se crea un directorio instantclient_10_2 con el siguiente contenido:

```
root@Debian-Server:~/instantclient_10_2# ls -l
```

```
total 33504
```

```
-rw-rw-r-- 1 root root 238 2008-03-12 04:37 BASIC_LITE_README
```

```
-r--r--r-- 1 root root 1609607 2008-03-12 04:37 classes12.jar
```

```
-rwxrwxr-x 1 root root 67542 2008-03-12 04:37 genezi
```

```
-rwxrwxr-x 1 root root 21038613 2008-03-12 04:37 libclntsh.so.10.1
```

```
-r-xr-xr-x 1 root root 3796601 2008-03-12 04:37 libnnz10.so
```

```
-rwxrwxr-x 1 root root 1664116 2008-03-12 04:37 libocci.so.10.1
-rwxrwxr-x 1 root root 4351321 2008-03-12 04:37 libociicus.so
-r-xr-xr-x 1 root root 138033 2008-03-12 04:37 libocijdbc10.so
-r--r--r-- 1 root root 1555682 2008-03-12 04:37 ojdbc14.jar
drwxrwxr-x 4 root root 4096 2008-03-12 04:37 sdk
```

Luego se crea el directorio `/opt/oracle/instantclient` al cual se copiará todo el contenido del directorio `instantclient_10_2`

```
root@Debian-Server:~# mkdir -p /opt/oracle/instantclient
```

```
root@Debian-Server:~# mv instantclient_10_2 /opt/oracle/instantclient
```

Ahora se debe especificarle al sistema de linkeo de librerías dinámicas (ld^2) la ubicación de un nuevo directorio donde podrá encontrar más bibliotecas software para el programa que lo requiera; en este caso ese "programa" será el módulo de python para acceder a Oracle, `cx_Oracle`. Es decir, que además de las bibliotecas disponibles en, por ejemplo, `/usr/lib` y `/usr/local/lib`, también se pondrá a disposición de los programas del usuario las bibliotecas del directorio `/opt/oracle/instantclient`.

Esto se hace con el programa `ldconfig`³:

```
root@Debian-Server:~# echo "/opt/oracle/instantclient"
```

```
> /etc/ld.so.conf.d/oracle.conf
```

```
root@Debian-Server:~# ldconfig
```

```
root@Debian-Server:~# ldconfig --print | grep /opt/oracle
```

```
libocijdbc10.so (libc6,x86) => /opt/oracle/instantclient/libocijdbc10.so
```

```
libociicus.so (libc6,x86) => /opt/oracle/instantclient/libociicus.so
```

```
libocci.so.10.1 (libc6,x86) => /opt/oracle/instantclient/libocci.so.10.1
```

```
libnnz10.so (libc6,x86) => /opt/oracle/instantclient/libnnz10.so
```

```
libclntsh.so.10.1 (libc6,x86) => /opt/oracle/instantclient/libclntsh.so.10.1
```

² Véase http://math.utah.edu/docs/info/ld_toc.html

³ Véase <http://linux.die.net/man/8/ldconfig>

```
root@Debian-Server:~# cd /opt/oracle/instantclient
```

```
root@Debian-Server:/opt/oracle/instantclient# ln -s libclntsh.so.10.1
```

```
libclntsh.so
```

```
root@Debian-Server:/opt/oracle/instantclient# ln -s libocci.so.10.1 libocci.so
```

La idea es crear un archivo con extensión .conf en el directorio /etc/ld.so.conf.d/ , con un nombre relacionado con nuestro propósito, que sólo indica un nuevo directorio de bibliotecas de ubicación "no estándar".Luego hay que refrescar la información de bibliotecas, con el comando "ldconfig". Después, "ldconfig --print" muestra todas las bibliotecas disponibles en el sistema para enlazar dinámicamente, pero como ahora sólo son de nuestro interés las del directorio /opt/oracle, se filtra la salida con un grep acorde. También se hacen dos enlaces simbólicos a las bibliotecas principales del Instantclient, cuyo número de versión es "genérico", a diferencia de los archivos originales, cuya versión es 10.1. Al momento de compilar el cx_Oracle, éste buscará la versión genérica de estas librerías, no una versión específica (por ende, serán los archivos libclntsh.so y libocci.so).

Ya se tiene todo lo necesario para la instalación del módulo cx_Oracle,pero todavía se necesitan algunas dependencias las cuales se resolverán en el instante que se ejecute el siguiente comando:

```
root@Debian-Server:~# apt-get install python-dev python-setuptools build-essential
```

```
Leyendo lista de paquetes... Hecho
```

```
Creando árbol de dependencias
```

```
Leyendo la información de estado... Hecho
```

```
Se instalarán los siguientes paquetes extras:binutils dpkg-dev g++ g++-4.2 gcc gcc-4.2 libc6-dev libgomp1  
libstdc++6-4.2-dev libtimedate-perl linux-libc-dev make patch python-pkg-resources python2.5-dev
```

```
Paquetes sugeridos:binutils-doc debian-keyring g++-multilib g++-4.2-multilib gcc-4.2-doc libstdc++6-4.2-  
dbg autoconf automake1.9 bison flex gcc-doc gcc-multilib gdb libtool manpages-devgcc-4.2-locales gcc-  
4.2-multilib libgcc1-dbg libgomp1-dbg libmudflap0-4.2-dbg libmudflap0-4.2-dev glibc-doc libstdc++6-4.2-  
doc make-doc diff-doc
```

```
Se instalarán los siguientes paquetes NUEVOS:
```

```
binutils build-essential dpkg-dev g++ g++-4.2 gcc gcc-4.2 libc6-dev libgomp1 libstdc++6-4.2-dev
```

```
libtimedate-perl linux-libc-dev make patch python-dev python-pkg-resourcespython-setuptools python2.5-dev
```

0 actualizados, 18 se instalarán, 0 para eliminar y 0 no actualizados.

Necesito descargar 12,8MB de archivos.

After this operation, 52,9MB of additional disk space will be used.

¿Desea continuar [S/n]?

[...]

Cuando se compilan módulos de python que acceden a librerías nativas (generalmente en C), debe instalarse el paquete "python-dev", que contiene los headers de las estructuras, funciones y demás símbolos del lenguaje. El metapaquete "build-essential" nos provee todo lo básico necesario para compilar programas en C/C++.

Luego se descomprime el módulo el cual se distribuye con el paquete. El archivo es cx_Oracle-5.0.tar.gz. La descompresión puede ser en un directorio cualquiera.

```
root@Debian-Server:~# tar xvzf cx_Oracle-5.0.tar.gz
```

```
cx_Oracle-5.0/
```

```
cx_Oracle-5.0/test/
```

```
cx_Oracle-5.0/test/test_dbapi20.py
```

```
cx_Oracle-5.0/test/LongVar.py
```

```
cx_Oracle-5.0/test/LobVar.py
```

```
cx_Oracle-5.0/test/DateTimeVar.py
```

```
cx_Oracle-5.0/test/test.py
```

```
cx_Oracle-5.0/test/NumberVar.py
```

[...]

Luego, para poder compilar/construir primero e instalar después el módulo, hay que establecer la variable de entorno ORACLE_HOME:

```
root@Debian-Server:~/cx_Oracle-5.0# export ORACLE_HOME=/opt/oracle/instantclient/
```

Luego se debe construir el módulo:

[root@Debian-Server](#):~/cx_Oracle-5.0# python setup.py build

running build

running build_ext

building 'cx_Oracle' extension

gcc -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions build/temp.linux-x86-2.5/c

[...]

Ahora ya se puede instalar el módulo:

[root@Debian-Server](#):~/cx_Oracle-5.0#python setup.py install

running install

running build

running build_ext

running install_lib

copying build/lib.linux-x86-2.5/cx_Oracle.so -> /usr/lib/python2.5/site-packages

running install_egg_info

Writing /usr/lib/python2.5/site-packages/cx_Oracle-5.0.egg-info

Ya se puede probar el módulo desde el intérprete de Python para comprobar que se instaló con éxito:

[root@Debian-Server](#):~# python

Python 2.5.2 (r252:60911, Jan 4 2009, 17:40:26)

[GCC 4.3.2] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>> import cx_Oracle

>>>

Nota: La instalación del módulo fue probada en el siguiente sistema:

Debian GNU/Linux 5.0 Lenny

Intel(R) Pentium(R) 4 CPU 3.00GHz

1 GB de RAM

Anexo #7 Plantilla de Plan de Prueba para O2PMigration

1. HU a probar

- Conectar_BD_Oracle
- Conectar_DB_PostgreSQL
- Visualizar_datos_Oracle
- Visualizar_datos_PostgreSQL
- Exportar_datos_a_PostgreSQL
- Cargar_datos_a_PostgreSQL

2. Cronograma

A continuación se muestra el cronograma de prueba para la herramienta O2PMigration

| No. | Tarea | Fecha | Responsable | Participantes | Observaciones |
|-----|---|----------|------------------------|--|--|
| 1. | Comprobar Usuario y contraseña para conectarse a Oracle. | 9/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se comprobará que los datos entrados estén correctamente y que el usuario tenga los permisos requeridos para conectarse a esta BD. |
| 2. | Comprobar que el huésped de Oracle este funcionando correctamente | 9/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se comprobará que la máquina a la que se esté conectando funcione correctamente. |
| 3. | Comprobar la | 10/03/09 | Marcos Ortíz | Marcos Ortíz | En esta tarea se comprobará |

| | | | | | |
|----|---|----------|---------------------------|--|---|
| | conexión a PostgreSQL | | Valmaseda | Valmaseda Susany Rodríguez Cervantes | que los datos entrados estén correctamente y que el usuario tenga los permisos requeridos para conectarse a esta BD. |
| 4. | Comprobar que el huésped de PostgreSQL funcione correctamente | 10/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se comprobará que la máquina a la que se esté conectando funcione correctamente. |
| 5. | Comprobar la extensión del archivo exportado | 11/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se verificará que el archivo exportado esté en .sql |
| 6. | Verificar que esté instalado el Oracle Instant Client | 14/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se verificará que la herramienta cliente-servidor para la gestión de bases de datos este instalada y funcionando correctamente. |
| 7. | Verificar el encoding de la Base de Datos PostgreSQL | 14/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se verificará el encoding de la Base de Datos PostgreSQL para que se pueda exportar los datos sin problema |
| 8. | Comprobar datos de Oracle mostrados | 19/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se verificará que los datos de Oracle que se muestran estén correctamente. |

| | | | | | |
|-----|---|----------|------------------------|--|--|
| 9. | Comprobar los datos de PostgreSQL mostrados | 19/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se verificará que los datos de PostgreSQL que se muestran estén correctamente. |
| 10. | Comprobar la conexión a PostgreSQL | 19/03/09 | Marcos Ortíz Valmaseda | Marcos Ortíz Valmaseda Susany Rodríguez Cervantes | En esta tarea se comprobará la conexión a PostgreSQL para poder exportar los datos hacia ella. |

3. Evaluación de las pruebas

El proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Cuando se realiza una prueba se dan ciertos criterios de evaluación como son:

Criterio de No conformidad, Pedidos de Cambios y Lista de Chequeo.

A continuación se describen estos criterios:

No conformidad: La no conformidad vendrá dada si se viola algún requisito funcional definido por el cliente, es decir, que lo que se le brinde no sea lo que el cliente solicitó.

Pedidos de Cambios: El pedido de cambio vendrá dado por cambios en requerimientos, funcionalidad, diseño o por errores y/o defectos.

Listas de Chequeos: Proporciona un apoyo mayor mediante preguntas que los probadores deben de responder mientras leen el artefacto. Esta técnica proporciona listas que ayudan al probador a saber qué tipo de faltas buscar.

Glosario de términos

ACID: (Atomicity, Consistency, Isolation, Durability)

Atomicity (Atomicidad): La atomicidad de una transacción garantiza que todas sus acciones sean realizadas o ninguna sea ejecutada.

Consistency (Consistencia): Muy relacionada con la "atomicidad". Las transacciones han de cumplir las restricciones definidas dentro la base de datos. Si no las pueden cumplir, se evita su ejecución. De esta forma se conserva la integridad y coherencia de los datos.

Isolation (Aislamiento): Esto garantiza que las transacciones que se estén realizando en el sistema sean invisibles al resto del sistema (usuarios, otras transacciones...) hasta que estas hayan sido finalizadas.

Durability (Durabilidad): La durabilidad de una transacción garantiza que al instante en el que se finaliza la transacción esta perdure a pesar de otras consecuencias, como caídas del sistema o de la base de datos

DBA: es la persona encargada de definir y controlar las bases de datos corporativas, además proporciona asesoría a los usuarios y ejecutivos que la requieran.

DML: son las siglas de Data Manipulation Language y se refiere a los comandos que permiten a un usuario manipular los datos en un repositorio, es decir, añadir, consultar, borrar o actualizar. En SQL los comandos SELECT, INSERT, UPDATE y DELETE son comandos DML.

Framework: término usado en programación orientada a objetos para definir un conjunto de clases que definen un diseño abstracto para solucionar un conjunto de problemas relacionados.

Historia de usuario: término en el cual se resume todo lo referente a las especificaciones de los requerimientos, así como lo relacionado con los casos de uso, conocidos del proceso unificado.

Migración: Término usado en informática para describir el proceso de pasar de una plataforma específica a otra; por ejemplo , el proceso de paso de Windows XP a Ubuntu GNU/Linux en un determinado organismo.

ORM: es un componente de software que me permite trabajar con los datos persistidos como si ellos fueran parte de una base de datos orientada a objetos (en este caso virtual). Debido a que lo estándar es trabajar con BD relacionales, se deben realizar operaciones que permitan transformar un registro en objeto y viceversa. A esta funcionalidad se la llama Mapeo objeto-relacional (ORM).

Perl: Lenguaje de programación creado por Larry Wall en 1987. Sus siglas quieren decir : Lenguaje Práctico para la Extracción e Informe. La última versión estable de perl es la 5.8. Es considerado el mejor lenguaje para tratamiento de cadenas por el excelente trabajo con las expresiones regulares. Es un lenguaje del tipo imperativo con variables, expresiones, asignaciones y bloques de código delimitado por llaves. Su logo es un camello.

Ruby: Lenguaje de programación creado por Yukihiro Matsumoto (Matt) en 1991 derivado de C. Totalmente orientado a objetos (Todo en Ruby es un objeto), hace que el trabajo del programador sea más fácil y placentero.

URL(Uniform Resources Locator): Localizador uniforme de recursos. Es la dirección de Internet incluye: "http" que indica el nombre del protocolo usado, "www" que es el nombre del servidor, "dir" es un directorio, "subdir" un subdirectorío y "file" el nombre de un archivo. Es la manera estándar de asignar direcciones de cualquier recurso en Internet que forma parte del WWW.

Schema: Término dado a la definición de la base de datos, incluyendo los datos, las funciones, las vistas, secuencias; todo menos los datos.

SDK (Software Development Kit): es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, framework, plataformas de hardware, ordenadores, videoconsolas, sistemas operativos.

SQL: es un lenguaje de base de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos. Pero como sucede con cualquier sistema de normalización hay excepciones para casi todo; de hecho, cada motor de bases de datos tiene sus peculiaridades y lo hace diferente de otro motor, por lo tanto, el lenguaje SQL normalizado (ANSI) no nos servirá para resolver todos los problemas, aunque si se puede asegurar que cualquier sentencia escrita en ANSI será interpretable por cualquier motor de datos.

SUN Microsystems: Las siglas SUN se derivan de «Stanford University Network», proyecto que se había creado para interconectar en red las bibliotecas de la Universidad de Stanford. Se hizo famosa por el eslogan «*The network is the computer*» («La red es la computadora»). Algunos de sus productos han sido servidores y estaciones de trabajo para procesadores SPARC, los sistemas operativos SunOS y Solaris, el sistema de archivos de red (NFS), la plataforma de programación Java y conjuntamente con la Corporación American Telephone and Telegraph (AT&T), la estandarización del UNIX System V Release 4.