

**Universidad de las Ciencias Informáticas  
Facultad 10**



**Título:** “Aplicación para la configuración y administración de un entorno de alta disponibilidad basado en Cybercluster.”

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

**Autores:**

Magdiel Vega González.  
Victor Manuel García Mishekurin.

**Tutor:**

Ing. Omar Rey Lazarte.  
Ing. Reynier Pernia Rodríguez.

**Ciudad de la Habana, junio/2010  
“Año del 52 Aniversario del Triunfo de la Revolución”**

## Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_ días del mes de \_\_\_\_\_ del año 2010.

**Magdiel Vega González**

**Victor Manuel García Mishekurin.**

\_\_\_\_\_

\_\_\_\_\_

Firma de los Autores

**Ing. Omar Rey Lazarte**

**Ing. Reynier Pernia Rodríguez.**

\_\_\_\_\_

\_\_\_\_\_

Firma de los Tutores

### DATOS DE CONTACTO

**Autor:** Magdiel Vega González.

**Correo electrónico:** mvgonzalez@estudiantes.uci.cu

**Autor:** Victor Manuel García Mishekurin.

**Correo electrónico:** vmgarcia@estudiantes.uci.cu

**Tutor:** Omar Rey Lazarte

**Especialidad de graduación:** Ingeniería en Ciencias de la Informática.

**Categoría docente:** Instructor.

**Categoría Científica:-**

**Años de experiencia:** 3

**Años de graduado:** 3

**Correo electrónico:** orey@uci.cu

**Tutor:** Reynier Pernia Rodríguez.

**Especialidad de graduación:**

**Categoría docente:**

**Categoría Científica:**

**Años de experiencia:**

**Años de graduado:**

**Correo electrónico:**

**AGRADECIMIENTOS DE “VICTOR”**

*Agradezco:*

**AGRADECIMIENTOS DE “MAGDIEL”**

*Agradezco:*

## Dedicatoria

## Resumen

Debido al gran incremento de información almacenada en base de datos y la alta demanda de acceso a estos datos, se hace necesaria la implantación de tecnología capaz de garantizar una alta tolerancia a fallos así como un alto nivel de acceso a la información.

Cybercluster es una herramienta de replicación sincrónica multimaestro basado en PostgreSQL que proporciona un alto rendimiento a los servidores de base de datos con una alta demanda de lectura. El despliegue de esta herramienta requiere de un cierto nivel de conocimiento sobre los comandos de administración del sistema operativo GNU/Linux así como los pasos a seguir para la configuración y administración necesaria para el correcto funcionamiento del Cybercluster.

El presente trabajo de diploma propone el desarrollo de una herramienta que brinde una interfaz amigable que posibilite al administrador del sistema un fácil manejo de la configuración y administración de los servicios de la herramienta Cybercluster. Como lenguaje de programación se utilizó python que posee una gama de bibliotecas de las cuales se usó el Paramiko para establecer conexiones ssh entre estaciones de trabajo. Se aplicó la metodología de desarrollo de software Programación Extrema (XP), con el objetivo de construir técnicamente un software en el menor tiempo posible, económico, fiable y que funciones eficientemente.

Con la instalación y puesta en práctica de la herramienta, se podrá desplegar un sistema de replica basado en el Cybercluster sin necesidad de tener un alto nivel de conocimiento sobre el proceso de configuración y administración del mismo.

### Palabras Claves:

Cybercluster, Python, Paramiko, metodología.

Índice

**ÍNDICE DE TABLAS. .... IX**

**INTRODUCCIÓN. .... 1**

**CAPÍTULO 1: CARACTERÍSTICAS DEL SISTEMA. .... 4**

1.1. CONCEPTOS RELACIONADOS CON CYBERCLUSTER. .... 4

1.1.1. *Cybercluster.* ..... 4

1.1.2. *Composición de un entorno basado en Cybercluster.*..... 4

    Servidor balanceador de carga..... 4

    Servidor de replicación. .... 5

    Servidor de datos..... 5

1.1.3. *Proceso de Configuración.* ..... 5

    Servidores de datos: ..... 5

    Servidor balanceador de carga:..... 6

    Servidor de replicación: ..... 6

1.1.4. *Proceso de Administración.*..... 6

    Servidor de datos: ..... 6

    Servidor de replicación: ..... 6

    Servidor balanceador de carga:..... 7

1.2. CONCEPTOS RELACIONADOS CON BASES DE DATOS (BD)..... 7

1.2.1. *Sistemas Gestores de Base de Datos (SGBD)*..... 7

1.2.2. *Base de Datos.*..... 8

    Administrador de Datos..... 9

    Administrador de BD. .... 9

1.2.3. *Objetivos de la monitorización.*..... 10

1.3. HERRAMIENTAS, TECNOLOGÍAS Y METODOLOGÍAS UTILIZADAS. .... 10

1.3.1. *Sistemas Gestores de Base de Datos.* ..... 10

    PostgreSQL..... 10

1.3.2. *Lenguajes de programación.*..... 12

    Java. .... 12

    Python. .... 13

1.3.3. *IDE de programación para Python.* ..... 15

    Eclipse (PyDev). .... 15

    Integrar Eclipse con Python..... 16

    PyDev..... 16

    Wing IDE. .... 17

1.3.4. *Interfaz Gráfica de Python.* ..... 18

    QT..... 18

    Qt Designer. .... 18

    Qt Assistant..... 19

    Integrando Qt a Python con PyQt..... 19

1.3.5. *Metodología de desarrollo de software.* ..... 19

    XP..... 19

    Microsoft Solution Framework (MSF). .... 21

    Proceso Racional Unificado (RUP). .... 22



<b>CAPÍTULO 2: EXPLORACIÓN Y PLANIFICACIÓN.....</b>	<b>25</b>
2.1. FASE DE EXPLORACIÓN.....	25
2.1.1. <i>Flujo actual de los procesos</i> .....	25
2.1.2. <i>Objeto de automatización</i> .....	25
2.1.3. <i>Propuesta del sistema a desarrollar</i> .....	26
2.1.4. <i>Personal relacionado con el sistema</i> .....	26
2.1.5. <i>Historias de Usuario (HU)</i> .....	26
Relación de Historias de Usuarios.....	27
2.1.6. <i>Apariencia del sistema</i> .....	29
2.2. FASE DE PLANIFICACIÓN.....	30
2.2.1. <i>Estimación de esfuerzo por Historias de Usuario</i> .....	30
2.2.2. PLAN DE ITERACIONES.....	30
Iteración 1.....	31
Iteración 2.....	31
2.2.3. <i>Plan de duración de las iteraciones</i> .....	31
2.2.4. <i>Plan de entregas</i> .....	32
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.....</b>	<b>33</b>
3.1 TAREAS DE INGENIERÍA.....	33
3.1.1 <i>Tareas de las historias de usuario implementadas en la primera iteración</i> .....	33
3.1.2 <i>Tareas de las historias de usuario implementadas en la segunda iteración</i> .....	38
3.2 TARJETAS CRC (CLASE – RESPONSABILIDAD - COLABORADOR).....	41
3.3 PRUEBAS DE ACEPTACIÓN.....	42
<b>CONCLUSIONES.....</b>	<b>54</b>
<b>RECOMENDACIONES.....</b>	<b>55</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>56</b>
<b>BIBLIOGRAFÍA.....</b>	<b>57</b>
<b>GLOSARIO.....</b>	<b>59</b>

## Índice de Tablas.

Tabla # 1. Personal relacionado con el sistema.	26
Tabla # 2. HU Instalar Cybercluster.	27
Tabla # 3. HU Inicializar base de datos.	27
Tabla # 4. HU Gestionar Nodo.	28
Tabla # 5. HU Gestionar servicio del Nodo.	28
Tabla # 6. HU Monitorear sistema.	29
Tabla # 7. HU Copias de Seguridad de la base de datos.	29
Tabla # 8. HU Notificación de fallos.	29
Tabla # 9. Estimación de las HU.	30
Tabla # 10. Plan de duración de las iteraciones.	32
Tabla # 11. Plan de entregas.	32
Tabla # 12. HU en la primera iteración.	33
Tabla # 13. Tarea #1 de la HU Instalar Cybercluster.	33
Tabla # 14. Tarea #2 de la HU Instalar Cybercluster.	34
Tabla # 15. Tarea #3 de la HU Instalar Cybercluster.	34
Tabla # 16. Tarea #4 de la HU Instalar Cybercluster.	34
Tabla # 17. Tarea #1 de la HU Inicializar base de datos.	35
Tabla # 18. Tarea #2 de la HU Inicializar base de datos.	35
Tabla # 19. Tarea #3 de la HU Inicializar base de datos.	35
Tabla # 20. Tarea #4 de la HU Inicializar base de datos.	35
Tabla # 21. Tarea #1 de la HU Gestionar Nodo.	36
Tabla # 22. Tarea #2 de la HU Gestionar Nodo.	36
Tabla # 23. Tarea #3 de la HU Gestionar Nodo.	36
Tabla # 24. Tarea #4 de la HU Gestionar Nodo.	37
Tabla # 25. Tarea #5 de la HU Gestionar Nodo.	37
Tabla # 26. Tarea #6 de la HU Gestionar Nodo.	37
Tabla # 27. HU en la segunda iteración.	38
Tabla # 28. Tarea #1 de la HU Gestionar servicio del nodo.	38
Tabla # 29. Tarea #2 de la HU Gestionar servicio del nodo.	38
Tabla # 30. Tarea #3 de la HU Gestionar servicio del nodo.	39
Tabla # 31. Tarea #4 de la HU Gestionar servicio del nodo.	39
Tabla # 32. Tarea #5 de la HU Gestionar servicio del nodo.	39
Tabla # 33. Tarea #1 de la HU Monitorear sistema.	40
Tabla # 34. Tarea #2 de la HU Monitorear sistema.	40
Tabla # 35. Tarea #3 de la HU Monitorear sistema.	40
Tabla # 36. Tarea #1 de la HU Copias de Seguridad de la base de datos.	40
Tabla # 37. Tarea #2 de la HU Copias de Seguridad de la base de datos.	41
Tabla # 38. Tarea #1 de la HU Notificación de fallos.	41
Tabla # 39. Tarea #2 de la HU Notificación de fallos.	41
Tabla # 40. Planilla de Caso de Prueba.	43
Tabla # 41. CP #1 de la HU Instalar Cybercluster.	43
Tabla # 42. CP #2 de la HU Instalar Cybercluster.	44
Tabla # 43. CP #1 de la HU Inicializar base de datos.	44
Tabla # 44. CP #2 de la HU Inicializar base de datos.	45
Tabla # 45. CP #1 de la HU Gestionar Nodo.	45
Tabla # 46. CP #2 de la HU Gestionar Nodo.	45
Tabla # 47. CP #3 de la HU Gestionar Nodo.	46
Tabla # 48. CP #4 de la HU Gestionar Nodo.	46
Tabla # 49. CP #5 de la HU Gestionar Nodo.	47

Tabla # 50. CP #6 de la HU Gestionar Nodo.....	47
Tabla # 51. CP #1 de la HU Gestionar servicio del nodo.....	48
Tabla # 52. CP #2 de la HU Gestionar servicio del nodo.....	48
Tabla # 53. CP #3 de la HU Gestionar servicio del nodo.....	49
Tabla # 54. CP #4 de la HU Gestionar servicio del nodo.....	49
Tabla # 55. CP #5 de la HU Gestionar servicio del nodo.....	50
Tabla # 56. CP #6 de la HU Gestionar servicio del nodo.....	50
Tabla # 57. CP #1 de la HU Monitorear sistema.....	51
Tabla # 58. CP #2 de la HU Monitorear sistema.....	51
Tabla # 59. CP #1 de la HU Copia de seguridad a la base de datos.....	52
Tabla # 60. CP #2 de la HU Copia de seguridad a la base de datos.....	52
Tabla # 61. CP #1 de la HU Notificación de fallo.....	53
Tabla # 62. CP #2 de la HU Notificación de fallo.....	53

## Introducción.

El Archivo Nacional de la República de Cuba es la institución encargada de dirigir metodológicamente la política de gestión documental en el territorio nacional y de salvaguardar la memoria histórica de la Nación Cubana, para lo cual garantiza el procesamiento, conservación, uso, acceso y difusión de los fondos documentales que custodia.

En la Universidad de las Ciencias Informáticas está en proyecto la informatización de estos procesos y la digitalización de los archivos con el objetivo de lograr una mejor prestación de servicios además de brindarle protección y durabilidad a los documentos del Archivo Nacional.

Debido a la gran cantidad de información que deberá ser almacenada en base de datos y al flujo de peticiones constantes que se le realizarían a dicha base de datos, se hace necesario implantar un sistema que permita la disponibilidad de los datos y un elevado rendimiento en los servidores de base de datos.

En el trabajo de diploma “Sistema de réplica multimaestro para bases de datos distribuidas en PostgreSQL” se obtuvo una solución para este tipo de situación, la cual consiste en la implantación de un sistema de replica multimaestro para bases de datos distribuidas en PostgreSQL basado en la herramienta Cybercluster. Con este sistema se obtiene un mejor rendimiento de los servidores de base de datos y una alta disponibilidad de los datos, pero el mismo debe ser configurado y administrado de forma manual, lo cual constituye una demora para realizar cualquier tarea en estos servidores.

El presente trabajo propone diseñar e implementar una solución capaz de lograr la informatización de estos procesos de configuración y administración de una forma gráfica y sencilla para el usuario final.

Dada la situación problemática anteriormente planteada se formula el siguiente **problema científico**: ¿Cómo contribuir a facilitar el proceso de configuración y administración de un entorno de alta disponibilidad basado en Cybercluster (EADBC)?.

Enmarcando la investigación en el **objeto de estudio**: proceso de configuración y administración de un entorno de alta disponibilidad, delimitando el **campo de acción** en: un entorno de alta disponibilidad basado en Cybercluster.

El trabajo queda sustentado en la siguiente **idea a defender**: El desarrollo de una aplicación para la configuración y administración de la herramienta Cybercluster, facilitaría el trabajo a los administradores del sistema.

La siguiente investigación persigue cumplir el **objetivo general**: Desarrollar una aplicación de escritorio que facilite el proceso de configuración y administración de un EADBC.

Para darle cumplimiento al objetivo general se desglosan los siguientes **objetivos específicos**:

1. Efectuar una comparación de las herramientas y tecnologías para la implementación de la aplicación.
2. Caracterizar el proceso de configuración manual de un EADBC.
3. Elaborar una aplicación que permita la configuración y administración del Cybercluster.

Para dar cumplimiento a los objetivos enunciados y demostrar la validez de la solución propuesta se enumeran las siguientes **tareas de investigación**:

1. Investigación de las herramientas a utilizar para el desarrollo del software.
2. Estudio del proceso de configuración y administración de un EADBC.
3. Diseño de una aplicación para la configuración y administración de un EADBC.
4. Implementación de una interfaz visual para el manejo de la configuración y administración de un EADBC.
5. Ejecución de pruebas para garantizar la calidad de las funcionalidades de la aplicación.

Para lograr comprender la situación existente, se utilizarán los siguientes métodos científicos:

Los **métodos teóricos** utilizados fueron:

**Análítico Sintético**: Para la elaboración del presente trabajo de diploma se efectuará un análisis de las diferentes herramientas que son indispensables para la construcción de una aplicación de escritorio, teniendo en cuenta aquellas características que las definen y que las hacen diferentes. Además, se realizará un estudio del proceso de configuración y administración del Cybercluster, con el propósito de entenderlo y sintetizarlo en una aplicación sencilla para el administrador del sistema.

El presente trabajo está conformado por 3 capítulos, a continuación se presenta sus nombres y objetivos de forma general:

**Capítulo 1: Características del sistema:** El objetivo de este capítulo es la realización de un análisis de las herramientas, tecnologías y metodologías empleadas para el desarrollo de la aplicación.

**Capítulo 2: Exploración y Planificación del sistema propuesto:** Este capítulo tiene como principal objetivo establecer los diferentes requisitos del sistema basados en historias de usuario, así como el plan de entregas y de iteraciones.

**Capítulo 3: Implementación y Pruebas:** El objetivo de este capítulo es describir las tareas de ingeniería, de las historias de usuario establecidas en el capítulo anterior y los casos de pruebas de aceptación.

### **Capítulo 1: Características del Sistema.**

El presente capítulo contiene las definiciones y conceptos más generales, los cuales servirán de apoyo para la comprensión del trabajo. Además, facilitará una visión más amplia referente a un EADBC. Contiene también las características generales de las herramientas y tecnologías utilizadas en la solución del problema y el resultado de la investigación referida a la configuración y administración de un EADBC.

#### **1.1. Conceptos relacionados con Cybercluster.**

##### **1.1.1. Cybercluster.**

Es la primera solución de replicación sincrónica para PostgreSQL la cual es completamente basada en los WAL de PostgreSQL. El uso de esta herramienta es óptimo para aplicaciones web que requieren una alta demanda de lectura. Es software libre y puede usarse libremente bajo los términos de la licencia de BSD.

*Cybercluster sincroniza escrituras al banco de datos para asegurarse que todos los nodos del banco de datos dentro del racimo contengan los mismos datos. En adición al acceso de lectura se encuentra el balanceador de carga que asegura que todos los nodos de los bancos de datos pueden trabajar eficazmente al mismo tiempo. En la práctica esto significa que el acceso de lectura pueda ser casi infinito. [13]*

##### **1.1.2. Composición de un entorno basado en Cybercluster.**

Cybercluster está conformado por servidor(es) balanceadores de carga, servidor(es) de replicación y servidor(es) de datos.

- **Servidor balanceador de carga.**

Cybercluster tiene su propio balanceador de carga que cumple con la tarea de distribuir la carga dentro del racimo de servidores de datos. La carga en el sistema es determinada por el número de consultas activas. El servidor de datos que menos carga tenga será el seleccionado para realizar la nueva consulta.

Si el servidor balanceador de carga detecta alguna falla en un servidor de datos del racimo automáticamente retira este nodo del sistema activo asegurándose así que ninguna nueva consulta sea enviada al nodo con fallo.

- **Servidor de replicación.**

Es el elemento central del sistema. Solo recibe demandas de los servidores de datos del racimo y se encarga de copiar esos cambios a todos los servidores de datos del racimo. Si el servidor de replicación descubre un problema en un servidor de datos del racimo excluirá este nodo de la lista de activos y se escribirá en disco un log que contendrá la descripción del error.

- **Servidor de datos.**

El cluster (servidor de datos del racimo) son las máquinas que procesan las consultas reales que entran. Cuando un cluster recibe una consulta del cliente debe determinar si esta es en presencia de una demanda de escritura o lectura. En caso de escritura el cluster enviará un mensaje al servidor de replicación para asegurar que se realicen los cambios en el resto del racimo de nodos de datos. Si la demanda recibida es de lectura entonces es ejecutada directamente por el cluster y el resultado se envía al cliente.

### 1.1.3. Proceso de Configuración.

Para el correcto funcionamiento del Cybercluster es necesaria la configuración de varios ficheros en cada uno de los servidores que intervendrán en el sistema.

Este fichero es común en todos los servidores que conforman el sistema.

pg\_hba.conf: En este fichero se controla el acceso de los clientes, a que bases de datos pueden acceder, con qué usuario accederán y mediante que método de autenticación.

- **Servidores de datos:**

postgresql.conf: En este fichero se especificarán las características necesarias para que PostgreSQL pueda trabajar correctamente.



cluster.conf: Este archivo controla cual(es) nodo(s) es/son usado(s) como servidor de replicación y que puerto se usa para establecer la conexión con el/los mismo(s). Además de la información del servidor de datos y se establecen los datos que no se quieren replicar.

- **Servidor balanceador de carga:**

pglb.conf: Este archivo controla cuales nodos son usados como servidores de datos, que puerto se usa para establecer la conexión con los mismos y cuantas conexiones son permitidas para cada servidor de BD. Además, también registra la información del servidor balanceador de carga en cuestión y se establecen las opciones de log.

- **Servidor de replicación:**

pgreplicate.conf: Este archivo controla cuales nodos son usados como servidores de datos, servidor balanceador de carga y que puerto se usa para establecer la conexión con los mismos. Además, cual puerto es usado para la solicitud de replicación desde los nodos de dato. También se registra la información del servidor de replicación en cuestión y se establecen las opciones de log.

### 1.1.4. Proceso de Administración.

El proceso de administración del sistema es realizado mediante la ejecución de comandos sobre la consola de comandos Shell dependiendo de la acción que se desee ejecutar sobre el sistema.

- **Servidor de datos:**

Los comandos para gestión de servicios de los cluster son ejecutados por el usuario de acceso al PostgreSQL.

Iniciar servicio: postgres@cluster: \$/usr/local/pgsql/bin/pg\_ctl -D /usr/local/pgsql/data start

Detener servicio: postgres@cluster: \$/usr/local/pgsql/bin/pg\_ctl -D /usr/local/pgsql/data stop

Reiniciar servicio: postgres@cluster: \$/usr/local/pgsql/bin/pg\_ctl -D /usr/local/pgsql/data reload

- **Servidor de replicación:**

Los comandos para la gestión de servicios en este servidor deben ser ejecutados por el súper usuario **root**.

Iniciar servicio: `root@replicador:$ /usr/local/pgsql/bin/pgreplicate -D /usr/local/pgsql/share`

Detener servicio: `root@replicador:$ /usr/local/pgsql/bin/pgreplicate -D /usr/local/pgsql/share stop`

El servicio de monitoreo puede ser ejecutado tanto por el usuario de acceso a PostgreSQL como por el súper usuario root.

Monitoreo: `usuario@replicador:$ /usr/local/pgsql/bin/pgplmon <servidor_replicador> <puerto_replica>`

La ejecución de este comando retorna como resultado la cantidad de transacciones y consultas realizadas a cada Servidor de Datos activo en el racimo.

- **Servidor balanceador de carga:**

Los comandos para la gestión de servicios en este servidor deben ser ejecutados por el súper usuario **root**.

Iniciar servicio: `root@balanceador:$ /usr/local/pgsql/bin/pglb -D /usr/local/pgsql/share`

Detener servicio: `root@balanceador:$ /usr/local/pgsql/bin/pglb -D /usr/local/pgsql/share stop`

## **1.2. Conceptos relacionados con Bases de Datos (BD).**

### **1.2.1. Sistemas Gestores de Bases de Datos (SGBD).**

Los primeros sistemas manejadores de archivos se conocen como ISAM y VSAM estos usaban la técnica de archivos separados, convirtiéndose en los primeros sistemas de base de datos. Con el avance de las tecnologías los sistemas manejadores de archivos fueron integrados en una única colección de datos y a partir de ese momento la manipulación de éstos fue llevada a cabo por los sistemas gestores de bases de datos.

Algunos autores han expuesto su criterio sobre este término, a continuación se presentan algunas definiciones del tema:

*El sistema de gestión de la base de datos es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.*[4]

*Los SGBD son aplicaciones o Suite cuya finalidad es la de controlar la entrada y salida de datos de una base de datos, manteniendo la integridad de la misma.[5]*

*Conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra a los distintos tipos de usuarios los medios necesarios para describir y manipular los datos almacenados en la base de datos, garantizando su integridad, confidencialidad y disponibilidad.[6]*

Se denomina SGBD al software que permite la gestión de la información almacenada en una BD, su objetivo fundamental es brindar al usuario las herramientas que le permitan manipular los datos de manera abstracta, sin que este necesite conocer la dirección lógica o física de la información en la computadora.

Los SGBD facilitan una gestión conveniente de la información a almacenar o recuperar, según el caso, dándole facilidad, fiabilidad y rapidez al usuario a la hora de gestionar los datos almacenados en la BD. Los SGBD se han convertido en la actualidad en el instrumento o soporte básico más utilizado en la gestión de los sistemas informáticos.

## **1.2.2. Base de Datos.**

Con el desarrollo de las tecnologías algunos autores han expuesto sus criterios u opiniones sobre este término, algunas de estas definiciones quedan expuestas a continuación:

*Las base de datos son aplicaciones informáticas para manejar información. La mayoría de las bases de datos actuales permiten hacer listados, consultas, crear pantallas de visualización de datos, controlar el acceso de los usuarios, etc. También es cada vez más frecuente que las consultas se puedan hacer en un lenguaje estándar conocido como Structured Query Language (SQL).[1]*

*Una base de datos es el lugar donde se guardan los datos en reposo y al cual acceden las diferentes aplicaciones (sistemas o programas) de una organización dada.[2]*

Desde el punto de vista de la informática, *la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.[3]*

En resumen, las bases de datos facilitan el archivo de elevadas cantidades de información; son utilizadas para realizar actividades ejemplo: seleccionar los datos más importantes a la hora de

resolver un problema, estudiar y relacionar estos datos, facilitando de esta forma la obtención de conclusiones.

- **Administrador de Datos.**

El Administrador de Datos (AD) es la persona identificable que tendrá la responsabilidad central sobre los datos dentro de la empresa. Ya que los datos son uno de los activos más valiosos de la empresa, es imperativo que exista una persona que los entienda junto con las necesidades de la empresa con respecto a esos datos, a un nivel de administrador superior. Por lo tanto, es labor del AD decidir en primer lugar que datos deben ser almacenados en la BD y establecer políticas para mantener y manejar esos datos una vez almacenados.

- **Administrador de BD.**

El Administrador de Base de Datos (ABD) es el técnico responsable de implementar las decisiones del AD. El trabajo del ABD consiste en crear la BD real e implementar los controles técnicos necesarios para hacer cumplir las diversas decisiones de las políticas hechas por el AD. El administrador de base de datos también es responsable de asegurar que el sistema opere con el rendimiento adecuado y de proporcionar una variedad de otros servicios técnicos.

Funciones del Administrador de Base de Datos.

Principales responsabilidades de un ABD

- ✓ Definir procedimientos de respaldo y recuperación de las BD.
- ✓ Definir los esquemas de seguridad e integridad que tendrán las BD.
- ✓ Supervisar el desempeño de las aplicaciones que corren bajo la BD.
- ✓ Garantizar el correcto funcionamiento de la BD.
- ✓ Diseñar el sistema de backups de la BD.
- ✓ Establecer mecanismos para garantizar la seguridad de los datos almacenados en la BD.
- ✓ Definir perfiles de usuario, otorgar privilegios y gestionar usuarios.
- ✓ Optimizar el desempeño y tiempo de respuesta de los servidores de BD.

- ✓ Garantizar la disponibilidad de los datos almacenados en la BD.

### 1.2.3. Objetivos de la monitorización.

Los fines principales de monitorear un servidor de bases de datos son:

- ✓ Verificar el consumo de recursos de cada ordenador en tiempo real.
- ✓ Actividad positiva del mismo en las computadoras.
- ✓ Tiempo de respuesta del servidor.
- ✓ Recabar la mayor cantidad de información a fin de poder tener los suficientes datos para ubicar donde está el problema.
- ✓ Detección de problemas de hardware o red.
- ✓ Obtener información de una determinada tarea o consulta (especial para puesta a punto de sentencias o servidor).

## 1.3. Herramientas, Tecnologías y Metodologías utilizadas.

### 1.3.1. Sistemas Gestores de Bases de Datos.

- **PostgreSQL.**

*Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. El desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales, la que se denominada como PGDG. Está considerado como la más avanzada base de datos de código abierto en el mundo, tiene una gran cantidad de características que, normalmente, sólo se encuentran en las bases de datos comerciales.[7]*

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en lugar de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Sus características técnicas la hacen una de las BD más potentes y robustas del mercado. Su desarrollo comenzó hace ya más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL presenta un alto

funcionamiento ante grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

Características generales:

- ✓ Es una base de datos cien por ciento ACID (del acrónimo Atomicidad, Consistencia, Aislamiento y Durabilidad).
- ✓ Posee integridad referencial y espacio de tablas.
- ✓ Permite realizar transacciones anidadas (puntos de retorno).
- ✓ Contiene replicación asíncrona y envío en dos fases.
- ✓ Facilita un punto en tiempo de recuperación.
- ✓ Brinda la posibilidad de realizar copias de seguridad en caliente (Online/Hot Backup).
- ✓ Es UNICODE.
- ✓ Soporta juegos de caracteres internacionales.
- ✓ Permite el acceso encriptado vía SSL.
- ✓ Tiene una documentación completa.
- ✓ Está basado en la licencia BSD.
- ✓ Disponible para sistemas GNU/Linux en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows.
- ✓ Soporta funciones/procedimientos almacenados en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de Oracle), PL/Perl, PL/Python y PL/Tcl.
- ✓ Dispone de numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier BD, tiene disponible, entre otros, tipos genéricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.
- ✓ Permite el almacenamiento de objetos binarios grandes (gráficos, videos, sonido).

- ✓ Tiene una interfaz amigable para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lips, Scheme, QT y muchos otros.[8]

Ventajas:

- ✓ Instalación ilimitada debido a que no hay costo asociado a la licencia del software.
- ✓ Cuenta con un excelente soporte.
- ✓ Estabilidad y confiabilidad única.
- ✓ Extensible y Multiplataforma.
- ✓ Diseñado para ambientes de alto volumen.

### 1.3.2. Lenguajes de programación.

Un lenguaje de programación es una construcción mental del ser humano la cual contiene un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina, además permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias.

- **Java.**

Fue desarrollado por la compañía Sun Microsystems a principios de los 90's con gran dedicación y siempre enfocado en cubrir las necesidades tecnológicas más punteras. La idea original de Java fue utilizarse en cualquier tipo de electrodoméstico, pero el mismo fracaso. La idea fue rescatada por uno de los fundadores de Sun para utilizarla en Internet y convirtieron a Java en un lenguaje potente, seguro y universal pues lo puede utilizar todo el mundo y es gratuito.

Es un lenguaje de programación orientado a objetos con el cual podemos realizar cualquier tipo de aplicación. En la actualidad se ha extendido a muchos ámbitos de la informática. Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador, como móviles y agendas.

Cuenta con una Máquina de Java para cada sistema operativo que hace de puente entre el sistema operativo y el programa de Java convirtiéndolo así en un lenguaje multiplataforma. Lo que resulta una

ventaja significativa para los desarrolladores de software que antes tenían que hacer un programa para cada sistema operativo.

Algunas características:

✓ Orientado a Objetos:

Java trabaja con sus datos como objetos y con interfaces a esos objetos, soporta las características propias del paradigma orientado a objetos: abstracción, encapsulamiento, herencia y polimorfismo.

✓ Simple:

Posee una curva de aprendizaje muy rápida. Ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de estos.

✓ Robusto:

Realiza verificaciones en busca de problemas, tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores lo antes posible en el ciclo de desarrollo. Java obliga a la declaración explícita de los tipos de los ítems de información, reduciendo así, las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de la misma. [9]

○ **Python.**

Es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, es decir, no necesita compilar el código fuente para poder ejecutarlo, ofreciendo ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

Es un lenguaje de programación multiparadigma, lo cual obliga a que los programadores adopten un estilo de programación particular. Contiene gran cantidad de bibliotecas, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar tareas habituales sin necesidad de reprogramarlas desde cero.

Características de Python:

✓ Lenguaje Interpretado o de Script:



Es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje de máquina.

- ✓ Funciones y bibliotecas:

Dispone de un gran cúmulo de funciones incorporadas en el propio lenguaje, permitiendo el tratamiento de cadenas de caracteres, números, archivos, etc. Además, existen librerías que son importadas con facilidad en los programas para tratar temas específicos.

- ✓ Tipado Dinámico:

Se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo de valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.

- ✓ Fuertemente Tipado:

No permite tratar a una variable sin tener en cuenta el tipo de dato que almacena, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente. Es decir, si se tiene una variable que contiene un texto (variable de tipo cadena o string) se podrá tratar como un número (sumar "9" y el número 8).

- ✓ Sintaxis clara:

Posee una sintaxis muy visual gracias a una notación indentada (con márgenes) de obligado cumplimiento. Además, para distanciar las porciones de código se aconseja tabular, colocando un margen al código que iría dentro de una función o un bucle. Esto beneficia a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

- ✓ Multiplataforma:

El intérprete de Python está disponible en multitud de plataformas (Solaris, GNU/Linux, DOS, Window, OS/2, Mac OS) por lo que si no se utilizan bibliotecas específicas de cada plataforma el programa podrá correr en todos estos sistemas sin grandes cambios.

- ✓ Orientado a Objeto:

La orientación a objetos es un paradigma de programación en el que los conceptos relevantes del mundo real para los problemas se trasladan a clases y objetos del programa. La ejecución del programa consiste en una serie de interacciones entre los objetos.

Python también permite la programación imperativa, programación funcional y programación orientada a aspectos. [10]

### ❖ **Conexiones SSH en Python. Paramiko.**

Es un módulo para Python 2.2 (o superior) que implementa el protocolo SSH2 (cifrado y autenticado) para conexiones remotas seguras. A diferencia de SSL (TLS alias), el protocolo SSH2 no requiere certificados jerárquicos firmado por una autoridad central. Se puede conocer SSH2 como el protocolo que sustituirá telnet y RSH para el acceso remoto seguro a consolas, pero el protocolo también incluye la posibilidad de abrir canales arbitrarios a los servicios remotos a través del túnel encriptado, así es cómo funciona la SFTP por ejemplo.

Está escrito completamente en Python y está liberado bajo la licencia GNU LGPL (Lesser GPL). [11]

Se seleccionó **Python** como lenguaje de programación debido a que es multiplataforma, contiene un gran cúmulo de librerías que disponen de funciones para el tratamiento de cadenas de caracteres, números, archivos, etc. Facilita la programación de alto nivel sin necesidad de preocuparse por detalles de bajo nivel, como el manejo de la memoria empleada por el programa. Permite la programación orientada a procedimientos así como orientada a objetos. Además, ofrece una vía fácil y simple de emplear la programación orientada a objetos, especialmente, cuando se compara con lenguajes como C++ o Java.

### **1.3.3. IDE de programación para Python.**

#### ○ **Eclipse (PyDev).**

Es principalmente una plataforma de programación, usada para crear entornos integrados de desarrollo (del inglés IDE).

Fue desarrollado originalmente por IBM como sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto.

La arquitectura de plugin de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces, ayudas en línea para bibliotecas, etc.

Eclipse así como sus plugins está desarrollado por completo en Java. Un problema de las herramientas Java (como NetBeans) es que son demasiado pesadas. Es decir, necesitan una máquina muy potente para poder ejecutarse por el uso del API Swing para su interfaz gráfico.

Eclipse utiliza la biblioteca SWT en sustitución de Swing para un mayor aprovechamiento de los widgets nativos del sistema donde se ejecuta. La contrapartida de esta librería es que es nativa, es decir, es necesario disponer de una librería SWT específica para cada sistema operativo.

### ○ **Integrar Eclipse con Python.**

Eclipse es una herramienta Open-Source y PyDev se distribuye bajo la EPL (Eclipse Public License). Es un IDE completo que integra editor de código, depurador e intérprete. Al estar escrito en Java es multiplataforma, por lo que no hay problema con el sistema operativo en el que se instale.

Eclipse es un IDE escrito en java. En un principio fue pensado para el desarrollo de programas Java, pero mediante sus plugins, se puede extender su ámbito a otros lenguajes.

### ○ **PyDev.**

Es un plugin para Eclipse que entrega a esta herramienta la potencialidad de un editor Python, Jython y Iron Python de gran nivel. PyDev se distribuye bajo la Eclipse Public License. Es un IDE completo que integra editor de código, depurador e intérprete.

Pros:

- ✓ Incorpora un depurador gráfico de Python, resaltado de sintaxis, auto completado.
- ✓ Permite desarrollar módulos en Jython, que es la implementación de Java en Python y cuyo principal fuerte es el uso de la biblioteca estándar de Java en módulos Python.
- ✓ Estructura los proyectos en directorios, recursos, etc. lo que ayuda a organizarlos y establecer una jerarquía.

Contras:

- ✓ Es una herramienta pesada. Precisamente porque no está optimizada para el desarrollo de módulos Python, debido a la cantidad de opciones y extensiones que acepta puede resultar fácil perderse entre otras configuraciones que ofrece.
- ✓ Eclipse es lento comparado con otros IDEs y también falla bastante cuando tiene que hacer alguna tarea compleja.[12]
- **Wing IDE.**

Es un IDE comercial disponible tanto en Windows, GNU/Linux como en OS. Ocupa unos 50MB instalado en el sistema, e incluye todos los componentes de un IDE. Su aspecto visual es muy similar al de Eclipse. Agrupa los módulos en proyectos y muestra en el panel inferior todas las herramientas que incluye.

Pros:

- ✓ Es muy sencillo de usar. Tiene una interfaz muy intuitiva.
- ✓ Incluye un depurador gráfico muy útil, bastante parecido al de Eclipse.
- ✓ Incluye ayuda bastante extensa, incluyendo tutoriales y ejemplos.
- ✓ Se integra perfectamente con la versión de Python instalada y no hace falta configurar ninguna ruta porque las busca automáticamente.
- ✓ Tiene un sistema de indentación automática, muy útil en Python, sobre todo cuando copias código de algún otro sitio, o tienes que añadir una nueva estructura y por lo tanto mover las líneas.
- ✓ El auto completado de código es mucho mejor que el de Eclipse.[13]

Luego de la investigación realizada se decide seleccionar como IDE de programación a Eclipse. Posee una serie de características y facilidades para crear y manipular bloques de código de una forma fácil y cómoda, tales como:

- ✓ Resaltado de errores y posibles soluciones para los mismos.
- ✓ Auto completamiento de código.
- ✓ Define y utiliza plantillas de código que se escriben automáticamente.
- ✓ Ordenamiento de código, legible y fácil de entender.

### 1.3.4. Interfaz Gráfica de Python.

- **QT.**

Qt es un framework open source lanzado bajo licencia GPL concebido para el desarrollo de aplicaciones e interfaces multiplataforma. Fue creada por la compañía noruega Trolltech (una compañía de software fundada en el año 1994). Es desarrollada con el lenguaje de programación C++ de forma nativa pero existen módulos para otros lenguajes de programación como son: C, Python (PyQt), Java(Qt Jambi), Perl(PerlQt), Gambas(GbQt), Ruby(QtRuby), PHP(PHP-Qt), Mono (Qyoto) entre los más reconocidos. Es una biblioteca totalmente orientada a objetos, es por ello que las APIs (Application Programming Interface) cuentan con diferentes métodos, soporta el uso de diferentes motores de base de datos y el uso de archivos XML, además de otras estructuras de datos tradicionales.

Esta librería de desarrollo se encuentra disponible para las siguientes plataformas:

- ✓ X11 – Para X Windows System (GNU/Linux, UNIX, BSD).
- ✓ Mac – Para los sistemas Mac OS X.
- ✓ Windows – Para las diferentes versiones del Sistema Operativo Windows.
- ✓ PDA – Para los dispositivos empotrados.

El paquete Qt integra herramientas de desarrollo y soporte tales como Qt Designer y Qt Assistant, así como librerías de clases auxiliares a Qt.

- **Qt Designer.**

Es una herramienta para el desarrollo de formularios y presentaciones gráficas para las aplicaciones. Permite acelerar el desarrollo de interfaces de alto rendimiento, a la vez que proporciona una forma fácil de diseñar interfaces gráficas de usuario avanzadas generando el código fuente de las mismas, posibilitando al desarrollador ajustarlo a sus necesidades.

Este generador de interfaces fue creado inicialmente por la empresa Trolltech para trabajar en varias distribuciones de GNU/Linux, actualmente puede instalarse en otras plataformas como Windows y Mac OsX.

El Qt Designer utiliza como base la biblioteca gráfica Qt, que ha sido transportada a diversas plataformas, lo que permite que el código generado por él pueda ser utilizado en diversos sistemas operativos. Esta herramienta provee características muy poderosas como pre visualización de la interfaz, soporte para widgets y un editor de propiedades bastante poderoso.

- **Qt Assistant.**

Es un componente de Qt que permite navegar por la documentación de Qt, en forma de páginas HTML, e implementa opciones de búsqueda y de extensión. En ella podemos consultar documentación de cada una de las clases, así como ejemplos de muchas de ellas. Además, contiene manuales de otras utilidades de Qt (Qt Designer, Qt Liguist, qmake).

- **Integrando Qt a Python con PyQt.**

PyQt es un conjunto de herramientas para crear aplicaciones GUI, es una fusión entre Python y la biblioteca Qt. Está disponible para Windows, GNU/Linux y MacOS. Es software libre y esta licenciado bajo la licencia GNU/GPL.

PyQt cuenta con un conjunto de módulos con más de 300 clases y 6000 métodos que hacen más fácil el trabajo del desarrollador. Dota al desarrollador de instrumentos que facilitan el diseño de una interfaz gráfica que puede ir desde sencilla hasta avanzada, pero siempre manteniendo un entorno amigable y usable por el usuario.

### 1.3.5. Metodología de desarrollo de software.

Una metodología es una ciencia que analiza las vías de conocimiento, es la aplicación lógica de un método o un cúmulo de acciones efectuadas con el objetivo de obtener un resultado definido, es decir, representa la forma de integrar el proceso de la investigación, de registrar sus efectos y de demostrar viables resultados a un problema que conlleva a la toma de decisiones.

- **XP.**

*Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo, equipo de trabajo pequeño y cuyo plazo de entrega sea corto. La metodología 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. [14]*

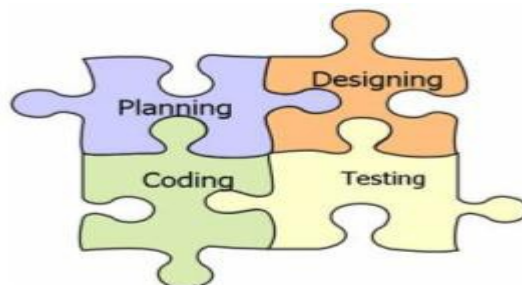


Figura #1 Partes de XP:

## Características de XP:

Esta metodología se basa en:

- ✓ Pruebas Unitarias: pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- ✓ Re fabricación: reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ✓ Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Esta metodología propone:

- ✓ Empezar en pequeño y añadir funcionalidades con retroalimentación continua.
- ✓ El manejo del cambio se convierte en parte sustantiva del proceso.
- ✓ El costo del cambio no depende de la fase o etapa.
- ✓ No introduce funcionalidades antes que sean necesarias.
- ✓ El cliente o usuario se convierte en miembro del equipo.

Derechos del cliente:

- ✓ Decidir que se implementa.
- ✓ Saber el estado real y el progreso del proyecto.
- ✓ Añadir, cambiar o quitar requerimientos en cualquier momento.
- ✓ Obtener lo máximo de cada semana de trabajo.

**Derechos del desarrollador:**

- ✓ Decidir cómo se implementará el proceso.
- ✓ Crear el sistema con la mejor calidad posible.
- ✓ Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- ✓ Estimar el esfuerzo para implementar el sistema.
- ✓ Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodologías es:

- ✓ La comunicación, entre los usuarios y los desarrolladores.
- ✓ La simplicidad, al desarrollar y codificar los módulos del sistema.
- ✓ La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

- **Microsoft Solution Framework (MSF).**

*Es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.*

Características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 ó 4 personas, así como también, proyectos que requieran 50 o más personas.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto.

- **Modelo de Arquitectura del Proyecto:** Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.



- **Modelo de Equipo:** Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.
- **Modelos de Proceso:** Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentando la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de Equipo.
- **Modelo de Gestión del Riesgo:** Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- **Modelo de Diseño del Proceso:** Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles los usuarios, el equipo y los desarrolladores.
- **Modelo de Aplicación:** Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones.

- **Proceso Racional Unificado (RUP).**

Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifican a este proceso de desarrollo.

RUP agrupa las actividades en grupos lógicos, definiendo 9 flujos de trabajo principales. Reconociendo los 6 primeros como flujos de ingeniería y el resto como flujo de apoyo.

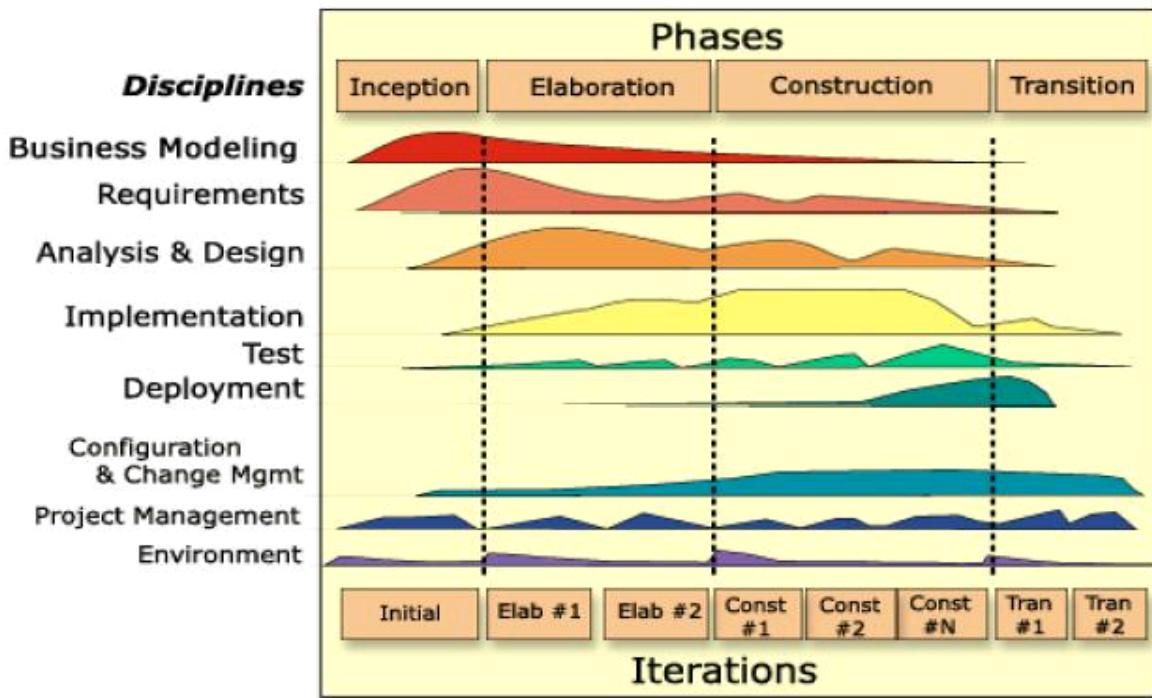


Figura # 2 Ciclo de vida de RUP.

El proceso de desarrollo de software está dividido en cuatro fases:

Inicio: Tiene como objetivo determinar la visión del proyecto.

Elaboración: El objetivo de esta fase es determinar la arquitectura óptima.

Construcción: Se concentra en la elaboración de un producto totalmente operativo y eficiente, y el manual de usuario.

Transición: Se instala el producto en el cliente y se entrena a los usuarios.

Características esenciales:

- ✓ Dirigido por Casos de Uso: Los casos de uso (CU) reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de usos guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los CU (cómo se llevan a cabo).

- ✓ **Centrado en la Arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- ✓ **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

Fue seleccionada **XP**, como la metodología de desarrollo debido a que se programa en parejas y el código generado puede ser modificado en cualquier momento. Contiene un diseño evolutivo el cual está centrado en las necesidades del momento. Permite reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. Además, el cliente forma parte activa del equipo de desarrollo, evitando con ello documentar todos los procesos que se realicen.

Teniendo en cuenta el estudio realizado anteriormente sobre las tendencias que existen actualmente en cuanto a los paradigmas de tecnologías y herramientas que más se utilizan en el campo de la informática, y además que cumplen con el Decreto Ley 3.390 al estar basada en tecnologías libres, se toma como decisión el desarrollo de una aplicación de escritorio utilizando Python como lenguaje de programación por los beneficios y ventajas que brinda en cuanto al trabajo con SSH2, utilizando Eclipse como IDE y la metodología XP para lograr la eficiencia del proyecto.

### **Capítulo 2: Exploración y Planificación.**

En la actualidad antes de iniciar un proyecto de desarrollo de software se desconoce de cierta forma los aspectos u objetivos que son de gran importancia para el mismo, entre estos cabe destacar el alcance que tendrá, el tiempo que demorará o el esfuerzo necesario para su evolución. La planificación contrarresta todos estos problemas, siendo un punto de partida y una alternativa eficiente para guiar el trabajo del equipo y obtener de esa forma la calidad deseada por el cliente. Para lograr estas metas la metodología XP propone las fases planificación y exploración, en las cuales se expondrán de forma breve el flujo actual de los procesos, la propuesta del sistema a desarrollar. Se identificará además el personal relacionado con el sistema y de forma más específica las historias de usuario.

#### **2.1. Fase de exploración.**

En esta fase se delimitará el alcance del proyecto, delimitando como fin la implementación y entrega del sistema propuesto. Para lograr este objetivo se hace ineludible concretar las necesidades existentes a través de las historias de usuario, las cuales son planteadas a grandes rasgos por los clientes. Partiendo de la información obtenida los programadores evaluarán de forma general el tiempo de desarrollo. Además, es necesario aclarar que tales estimaciones en esta etapa son primarias, ya que están basadas en datos de alto nivel los cuales pueden variar a medida que se analicen con mayor cuidado y detalle en las siguientes iteraciones.

##### **2.1.1. Flujo actual de los procesos.**

La configuración y administración de un sistema de réplica basado en Cybercluster se realiza hoy en día mediante la edición de ficheros en el caso de la configuración y la ejecución de comandos necesarios para iniciar, detener, reiniciar los servicios, entre otras operaciones de administración de los nodos que intervienen en el sistema de réplica. Estos procesos requieren del acceso físico a los diferentes ordenadores que formarán parte del sistema. El número de ordenadores que conforman el sistema no es limitado como tampoco la distancia que poseen entre ellos, las cuales pueden ser relativamente largas o cortas.

##### **2.1.2. Objeto de automatización.**

Para llevar a cabo la configuración y administración de un sistema de réplica basado en Cybercluster existen varios procesos que deben ser automatizados pues su ejecución de forma manual resulta tediosa y en muchas ocasiones algo complicada.

Entre los procesos que serán objeto de automatización se encuentran la instalación del sistema e inicialización de la base de datos. Además, la gestión, configuración y administración de los nodos, el monitoreo del sistema, la notificación de fallos y la realización de copias de seguridad de la base de datos.

### 2.1.3. Propuesta del sistema a desarrollar.

Para darle solución al problema planteado el presente trabajo de diploma propone desarrollar una herramienta administrativa que brinde un conjunto de funcionalidades acordes al flujo de trabajo actual. Dicho sistema debe establecer conexiones remotas entre ordenadores y la ejecución de comandos. Además, debe facilitar la configuración y administración de cada nodo desde un ordenador central y el monitoreo general del entorno desde una interfaz gráfica sencilla.

### 2.1.4. Personal relacionado con el sistema.

Una de las bases fundamentales que se deben tener en cuenta cuando se comienza el desarrollo de un sistema informático es la delimitación del usuario final; el cual en algunas ocasiones puede ser un personal relacionado con el sistema o alguien completamente ajeno a él.

Personal relacionado con el sistema.	Descripción.
Administrador del sistema.	Es el personal que posee los permisos necesarios para la gestión del sistema en general. También es el encargado de la gestión, configuración y administración de los nodos, el monitoreo del sistema y la realización de copias de seguridad de la base de datos.

Tabla # 1. Personal relacionado con el sistema.

### 2.1.5. Historias de Usuario (HU).

Las historias de usuario constituyen un aspecto fundamental en el desarrollo de todo software. Estas son escritas por el cliente o con ayuda del mismo, delimitando en su propio lenguaje lo que el sistema debe realizar. La discrepancia más visible entre estas historias de usuario utilizadas en la metodología XP y los documentos de especificación de requisitos usados por RUP, se hallan en el nivel de detalle requerido en la realización de los mismos.

Las historias de usuario son documentos que contienen exclusivamente los detalles mínimos y absolutamente necesarios para que los programadores puedan realizar una estimación lo más acertada posible y poco riesgosa del tiempo que llevará su desarrollo. Una vez llegado el período de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

Las historias de usuarios a implementar son:

- Instalar Cybercluster.
- Inicializar la base de datos.
- Gestionar Nodo.
- Gestionar servicio del Nodo.
- Monitorear sistema.
- Copias de Seguridad de la base de datos.
- Notificación de fallos.

### ○ Relación de Historias de Usuarios.

Historia de Usuario: Instalar Cybercluster.

Historia de Usuario	
Número : 1	Usuario : Administrador
Nombre: Instalar Cybercluster.	
Prioridad en el Negocio: Alta.	Riesgo de desarrollo: Medio.
Puntos de Estimación: 2	Iteración Asignada: 1
Programador responsable: Magdiel Vega González, Victor Manuel García Mishekurin.	
Descripción: Se realiza la conexión ssh al ordenador deseado, se copia en el mismo el instalador y se ejecuta de manera ordenada una serie de comandos para la instalación.	
Observaciones:	

Tabla # 2. HU Instalar Cybercluster.

Historia de Usuario: Inicializar base de datos.

Historia de Usuario.	
Número : 2	Usuario : Administrador
Nombre: Inicializar base de datos.	
Prioridad en el Negocio: Alta.	Riesgo de desarrollo: Bajo.
Puntos de Estimación: 2	Iteración Asignada: 1
Programador responsable: Magdiel Vega González, Victor Manuel García Mishekurin.	
Descripción: Se crea un usuario de acceso al servidor PostgreSQL. Además, se establece un directorio donde se inicializará la base de datos y se guardarán los ficheros de configuración del nodo.	
Observaciones:	

Tabla # 3. HU Inicializar base de datos.

Historia de Usuario: Gestionar Nodo.

Historia de Usuario.	
<b>Número:</b> 3	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Gestionar Nodo.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo de desarrollo:</b> Bajo.
<b>Puntos de Estimación:</b> 2	<b>Iteración Asignada:</b> 1
<b>Programador responsable:</b> Magdiel Vega González, Victor Manuel García Mishekurin.	
<b>Descripción:</b> Se realiza la acción de crear, actualizar y eliminar un nodo. El sistema debe dar la posibilidad de agregar un nodo, el cual puede ser servidor balanceador de carga, servidor de replicación o servidor de datos. Además, debe permitir tanto actualizar como eliminar un nodo existente en el sistema.	
<b>Observaciones:</b>	

**Tabla # 4. HU Gestionar Nodo.**

Historia de Usuario: Gestionar servicio del Nodo.

Historia de Usuario.	
<b>Número:</b> 4	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Gestionar servicio del Nodo.	
<b>Prioridad en el Negocio:</b> Media	<b>Riesgo de desarrollo:</b> Bajo.
<b>Puntos de Estimación:</b> 2	<b>Iteración Asignada:</b> 2
<b>Programador responsable:</b> Magdiel Vega González, Victor Manuel García Mishekurin.	
<b>Descripción:</b> Se gestiona el inicio, detención y reinicio de los servicio del nodo.	
<b>Observaciones:</b>	

**Tabla # 5. HU Gestionar servicio del Nodo.**

Historia de Usuario: Monitorear sistema.

Historia de Usuario.	
<b>Número:</b> 5	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Monitorear sistema.	
<b>Prioridad en el Negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Bajo.
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 2
<b>Programador responsable:</b> Magdiel Vega González, Victor Manuel García Mishekurin.	

**Descripción:** Debe permitir la ejecución de un comando al cual se le pasa como parámetro el IP del servidor de replicación y el puerto de monitoreo, devolviendo una lista de los nodos de base de datos activos con la información concerniente al rendimiento de cada uno.

**Observaciones:**

**Tabla # 6. HU Monitorear sistema.**

Historia de Usuario: Copias de Seguridad de la base de datos.

Historia de Usuario.	
<b>Número:</b> 6	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Copias de Seguridad de la base de datos.	
<b>Prioridad en el Negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 2
<b>Programador responsable:</b> Magdiel Vega González, Victor Manuel García Mishekurin.	
<b>Descripción:</b> Debe permitir la realización de las copias de seguridad a la base de datos de forma automática o manual.	
<b>Observaciones:</b>	

**Tabla # 7. HU Copias de Seguridad de la base de datos.**

Historia de Usuario: Notificación de fallos.

Historia de Usuario.	
<b>Número:</b> 7	<b>Usuario:</b> Administrador.
<b>Nombre:</b> Notificación de fallos	
<b>Prioridad en el Negocio:</b> Media.	<b>Riesgo de desarrollo:</b> Bajo.
<b>Puntos de Estimación:</b> 1	<b>Iteración Asignada:</b> 2
<b>Programador responsable:</b> Magdiel Vega González, Victor Manuel García Mishekurin.	
<b>Descripción:</b> El sistema debe permitir la notificación vía correo en caso de existir algún problema en algún servidor del sistema.	
<b>Observaciones:</b>	

**Tabla # 8. HU Notificación de fallos.**

### 2.1.6. Apariencia del sistema.

Teniendo en cuenta las diferentes características que definen la metodología XP, en cuanto a la apariencia e interfaz externa, el sistema deberá ser fácil de usar y poseer un entorno agradable al usuario final. Todas las funcionalidades que la aplicación brinda se encuentran organizadas y divididas por categorías, lo cual



permite una factible localización de la misma. La estructura ostentada por los formularios es sencilla contando para ello únicamente con los campos necesarios.

### 2.2. Fase de Planificación.

En la fase de Planificación el cliente con mucho cuidado define la prioridad que posee cada historia de usuario. En esta etapa los programadores efectúan una estimación del esfuerzo necesario para desarrollar cada una de estas historias de usuarios. De manera análoga realizan una toma de acuerdos sobre el contenido y plazo de la primera entrega plasmándose en un cronograma, ambas actividades son realizadas en conjunto con el cliente. Este proceso tiene una corta duración y puede realizarse teniendo en cuenta el alcance antes planteado. Si se escoge esta vía se puede determinar el número de iteraciones necesarias para su implementación con tan solo dividir la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto.

#### 2.2.1. Estimación de esfuerzo por Historias de Usuario.

Para el correcto desarrollo de la aplicación se realizó una estimación detallada de cada una de las historias de usuario identificadas, arrojando los resultados que se muestran a continuación:

Historias de usuario.	Puntos de estimación.
Instalar Cybercluster	2
Inicializar base de datos	2
Gestionar Nodo	2
Gestionar servicio del Nodo	2
Monitorear sistema	1
Copias de seguridad a la base de datos.	1
Notificación de fallos.	1

Tabla # 9. Estimación de las HU.

#### 2.2.2. Plan de iteraciones.

Puntualizadas las historias de usuario por parte del cliente y estimado el esfuerzo por los desarrolladores para la realización de las mismas, se procede a realizar la planificación de las etapas de implementación del sistema. Este plan concentra las historias de usuario por iteraciones, definiendo cuáles serán

desarrolladas en cada iteración del proceso de implementación. En relación con lo antes mencionado se decide implementar el sistema en dos iteraciones, las cuales se describen a continuación:

### ○ Iteración 1.

La iteración #1 tiene como objetivo central darle cumplimiento a las historias de usuario 1, 2 y 3. Dichas historias de usuario son de gran relevancia para la aplicación, ya que ellas constituyen el eje central de la estructura básica del sistema. Además del correcto funcionamiento de ellas dependen el resto de las funcionalidades. En esta primera etapa se llevará a cabo la instalación y configuración del entorno de réplica.

### ○ Iteración 2.

La iteración #2 tiene como objetivo central darle cumplimiento a las historias de usuario 4, 5, 6 y 7. Con la implementación de estas historias de usuarios se culmina con las funcionalidades secundarias del sistema propuesto, permitiendo la gestión de los servicios de los nodos, el monitoreo del entorno, las copias de seguridad de la base de datos y la notificación de fallos. Una vez terminada esta iteración se tendrá la aplicación lista para su entrega.

### 2.2.3. Plan de duración de las iteraciones.

El plan de iteraciones es un esquema que tiene como propósito describir la planificación temporal y de recursos, a nivel de tareas y actividades de las diferentes iteraciones. Su objetivo principal es seleccionar las historias de usuario, ubicándolas en la iteración en que se va a implementar.

No. Iteración.	Historias de Usuario.	Duración total de la Iteración.
1.	<ul style="list-style-type: none"><li>• Instalar Cybercluster.</li><li>• Inicializar base de datos.</li><li>• Gestionar nodo.</li></ul>	6 semanas.

2.	<ul style="list-style-type: none"> <li>• Gestionar servicio del nodo.</li> <li>• Monitorear sistema.</li> <li>• Copias de Seguridad de la base de datos.</li> <li>• Notificación de fallos.</li> </ul>	5 semanas
----	--	-----------

**Tabla # 10. Plan de duración de las iteraciones.**

### 2.2.4. Plan de entregas.

Es una planificación en la cual los desarrolladores y los clientes establecen los períodos de implementación a través de las historias de usuarios, clasificando las mismas según sus prioridades. En este plan se determina efectuar dos entregas, las cuales se manifiestan en la tabla que muestran a continuación:

No. Iteración.	Duración total de la iteración.	Fecha de inicio.	Fecha de fin.
1	6 semanas	18/01/10	01/03/10
2	5 semanas	15/03/10	19/04/10

**Tabla # 11. Plan de entregas.**

En el presente capítulo se realizó un análisis del flujo actual de los procesos, específicamente los que serán objeto de automatización para la configuración y administración de un sistema de réplica basado en Cybercluster. Además se identificó al administrador del sistema como el personal involucrado en el desarrollo del mismo, así como las historias de usuarios que cada desarrollador debe implementar en correspondencia con la iteración en que se encuentre.

## Capítulo 3: Implementación y Pruebas.

En el presente capítulo se describen las dos iteraciones a desarrollar durante la etapa de construcción del sistema propuesto. Además, se presentan de forma puntual cada una de las tareas de ingeniería generadas por cada historia de usuario, así como las pruebas de aceptación confeccionadas a las mismas.

### 3.1 Tareas de Ingeniería.

Las tareas de ingeniería se efectúan para detallar mejor las historias de usuario, facilitando con ello el entendimiento en el proceso de implementación. Cada HU puede contener una o más tareas de ingeniería en caso de necesitarla, explicando paso a paso las acciones que se realizan en la misma. Los puntos de estimación asignados a cada tarea son medidos por días según las características y la complejidad que posea la misma.

#### 3.1.1 Tareas de las historias de usuario implementadas en la primera iteración.

No. Iteración.	Historias de Usuario.
1.	<ul style="list-style-type: none"> <li>• Instalar Cybercluster.</li> <li>• Inicializar base de datos.</li> <li>• Gestionar nodo.</li> </ul>

Tabla # 12. HU en la primera iteración.

Instalar Cybercluster:

Tarea	
<b>Número tarea:</b> 1.	<b>Número historia:</b> 1.
<b>Nombre tarea:</b> Configuración de la interfaz para la conexión ssh.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.
<b>Fecha inicio:</b> 18/01/10.	<b>Fecha fin:</b> 18/01/10.
<b>Descripción:</b> Se configura la interfaz de usuario pedir los datos del servidor donde será instalado el Cybercluster.	

Tabla # 13. Tarea #1 de la HU Instalar Cybercluster.

Tarea	
<b>Número tarea:</b> 2.	<b>Número historia:</b> 1.

<b>Nombre tarea:</b> Comprobar los datos introducidos.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 2.
<b>Fecha inicio:</b> 19/01/10.	<b>Fecha fin:</b> 20/01/10.
<b>Descripción:</b> Se comprueban los datos del servidor verificando su validez.	

**Tabla # 14. Tarea #2 de la HU Instalar Cybercluster.**

Tarea	
<b>Número tarea:</b> 3.	<b>Número historia:</b> 1.
<b>Nombre tarea:</b> Copia del paquete instalador del Cybercluster.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 2.
<b>Fecha inicio:</b> 21/01/10.	<b>Fecha fin:</b> 22/01/10.
<b>Descripción:</b> Se realiza la copia del paquete instalador del Cybercluster en el servidor seleccionado.	

**Tabla # 15. Tarea #3 de la HU Instalar Cybercluster.**

Tarea	
<b>Número tarea:</b> 4.	<b>Número historia:</b> 1.
<b>Nombre tarea:</b> Instalación del Cybercluster.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 6.
<b>Fecha inicio:</b> 25/01/10.	<b>Fecha fin:</b> 1/02/10.
<b>Descripción:</b> Ejecución de una serie de comandos necesarios para la instalación del Cybercluster.	

**Tabla # 16. Tarea #4 de la HU Instalar Cybercluster.**

Inicializar base de datos.

Tarea	
<b>Número tarea:</b> 1.	<b>Número historia:</b> 2.
<b>Nombre tarea:</b> Configurar interfaz para autenticar al usuario de la base de datos.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0.3.
<b>Fecha inicio:</b> 2/02/10.	<b>Fecha fin:</b> 2/02/10.

**Descripción:** Se configura la interfaz de usuario que pedirá los datos necesarios para inicializar la base de datos.

**Tabla # 17. Tarea #1 de la HU Inicializar base de datos.**

Tarea	
<b>Número tarea:</b> 2.	<b>Número historia:</b> 2.
<b>Nombre tarea:</b> Comprobar si el usuario es válido.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0.3.
<b>Fecha inicio:</b> 3/02/10.	<b>Fecha fin:</b> 3/02/10.
<b>Descripción:</b> Se comprueban los datos introducidos verificando que se autenticó correctamente en caso de que existe el usuario de acceso a PostgreSQL.	

**Tabla # 18. Tarea #2 de la HU Inicializar base de datos.**

Tarea	
<b>Número tarea:</b> 3.	<b>Número historia:</b> 2.
<b>Nombre tarea:</b> Establecimiento del directorio.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 5.
<b>Fecha inicio:</b> 4/02/10.	<b>Fecha fin:</b> 10/02/10.
<b>Descripción:</b> Se establece un directorio donde se inicializará la base de datos y se guardarán los ficheros de configuración del nodo.	

**Tabla # 19. Tarea #3 de la HU Inicializar base de datos.**

Tarea	
<b>Número tarea:</b> 4.	<b>Número historia:</b> 2.
<b>Nombre tarea:</b> Inicializar la base de datos.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 4.
<b>Fecha inicio:</b> 11/02/10.	<b>Fecha fin:</b> 16/02/10.
<b>Descripción:</b> Se ejecutan los comandos necesarios para inicializar la base de datos.	

**Tabla # 20. Tarea #4 de la HU Inicializar base de datos.**

Gestionar Nodo.

Tarea	
<b>Número tarea:</b> 1.	<b>Número historia:</b> 3.
<b>Nombre tarea:</b> Configurar interfaz para crear un nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0.3.
<b>Fecha inicio:</b> 17/02/10.	<b>Fecha inicio:</b> 17/02/10.
<b>Descripción:</b> Se configura la interfaz que facilitará la introducción de los datos necesarios para la creación de un nodo.	

**Tabla # 21. Tarea #1 de la HU Gestionar Nodo.**

Tarea	
<b>Número tarea:</b> 2.	<b>Número historia:</b> 3.
<b>Nombre tarea:</b> Creación del nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 3.
<b>Fecha inicio:</b> 18/02/10.	<b>Fecha fin:</b> 22/02/10.
<b>Descripción:</b> Se realiza la copia de los ficheros de configuración necesarios para la creación y funcionamiento del nodo.	

**Tabla # 22. Tarea #2 de la HU Gestionar Nodo.**

Tarea	
<b>Número tarea:</b> 3.	<b>Número historia:</b> 3.
<b>Nombre tarea:</b> Actualizar nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 3.
<b>Fecha inicio:</b> 23/02/10.	<b>Fecha fin:</b> 25/02/10.
<b>Descripción:</b> Se realiza la actualización de los ficheros de configuración necesarios para el funcionamiento del nodo.	

**Tabla # 23. Tarea #3 de la HU Gestionar Nodo.**

Tarea	
<b>Número tarea:</b> 4.	<b>Número historia:</b> 3.

<b>Nombre tarea:</b> Configurar interfaz para eliminar un nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.
<b>Fecha inicio:</b> 26/02/10.	<b>Fecha fin:</b> 26/02/10.
<b>Descripción:</b> Se configura la interfaz que facilitará la eliminación de un nodo.	

**Tabla # 24. Tarea #4 de la HU Gestionar Nodo.**

Tarea	
<b>Número tarea:</b> 5.	<b>Número historia:</b> 3.
<b>Nombre tarea:</b> Comprobar si los datos de autenticación son correctos.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.
<b>Fecha inicio:</b> 1/03/10.	<b>Fecha fin:</b> 1/03/10.
<b>Descripción:</b> Se comprueba la validez de los datos introducidos para la autenticación con el servidor seleccionado para ser eliminado.	

**Tabla # 25. Tarea #5 de la HU Gestionar Nodo.**

Tarea	
<b>Número tarea:</b> 6.	<b>Número historia:</b> 3.
<b>Nombre tarea:</b> Eliminar del nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 3.
<b>Fecha inicio:</b> 2/03/10.	<b>Fecha fin:</b> 4/03/10.
<b>Descripción:</b> Se eliminan los ficheros de configuración en el servidor.	

**Tabla # 26. Tarea #6 de la HU Gestionar Nodo.**



3.1.2 Tareas de las historias de usuario implementadas en la segunda iteración.

No Iteración.	Historias de Usuario.
Iteración 2.	<ul style="list-style-type: none"> <li>• Gestionar servicio del nodo.</li> <li>• Monitorear sistema.</li> <li>• Copias de seguridad de la base de datos.</li> <li>• Notificación de fallos.</li> </ul>

Tabla # 27. HU en la segunda iteración.

Gestionar servicio del nodo.

Tarea	
<b>Número tarea:</b> 1.	<b>Número historia:</b> 4.
<b>Nombre tarea:</b> Configurar interfaz para inicializar el servicio del nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.
<b>Fecha inicio:</b> 5/03/10.	<b>Fecha fin:</b> 5/03/10.
<b>Descripción:</b> Se configura la interfaz que facilitará la inicialización de los servicios de un nodo.	

Tabla # 28. Tarea #1 de la HU Gestionar servicio del nodo.

Tarea	
<b>Número tarea:</b> 2.	<b>Número historia:</b> 4.
<b>Nombre tarea:</b> Configurar interfaz para la autenticación de la conexión con el nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.
<b>Fecha inicio:</b> 8/03/10.	<b>Fecha fin:</b> 8/03/10.
<b>Descripción:</b> Se comprueban si se estableció la conexión con el nodo.	

Tabla # 29. Tarea #2 de la HU Gestionar servicio del nodo.

Tarea	
<b>Número tarea:</b> 3.	<b>Número historia:</b> 4.

<b>Nombre tarea:</b> Iniciar servicio de nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 4.
<b>Fecha inicio:</b> 9/03/10.	<b>Fecha fin:</b> 12/03/10.
<b>Descripción:</b> Se ejecuta el comando necesario para iniciar el nodo. Se comprueban si se inició correctamente el servicio del nodo.	

Tabla # 30. Tarea #3 de la HU Gestionar servicio del nodo.

Tarea	
<b>Número tarea:</b> 4.	<b>Número historia:</b> 4
<b>Nombre tarea:</b> Configurar interfaz para detener el servicio del nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.
<b>Fecha inicio:</b> 15/03/10.	<b>Fecha fin:</b> 15/03/10.
<b>Descripción:</b> Se configura la interfaz que facilitará detener los servicios de un nodo.	

Tabla # 31. Tarea #4 de la HU Gestionar servicio del nodo.

Tarea	
<b>Número tarea:</b> 5.	<b>Número historia:</b> 4
<b>Nombre tarea:</b> Detener servicio del nodo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 4.
<b>Fecha inicio:</b> 16/03/10.	<b>Fecha fin:</b> 19/03/10.
<b>Descripción:</b> Se ejecuta el comando necesario para detener el nodo seleccionado. Se comprueba que se detuvo correctamente el servicio del nodo.	

Tabla # 32. Tarea #5 de la HU Gestionar servicio del nodo.

Monitorear sistema.

Tarea	
<b>Número tarea:</b> 1.	<b>Número historia:</b> 5.
<b>Nombre tarea:</b> Configurar interfaz para acceder al servidor replicador.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.

<b>Fecha inicio:</b> 22/03/10.	<b>Fecha fin:</b> 22/03/10.
<b>Descripción:</b> Se configura la interfaz para acceder al servidor de replicación.	

**Tabla # 33. Tarea #1 de la HU Monitorear sistema.**

Tarea	
<b>Número tarea:</b> 2.	<b>Número historia:</b> 5.
<b>Nombre tarea:</b> Comprobar si los datos de autenticación son correctos.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0,3.
<b>Fecha inicio:</b> 23/03/10.	<b>Fecha fin:</b> 23/03/10.
<b>Descripción:</b> Se comprueba si se estableció la conexión al servidor de replicación.	

**Tabla # 34. Tarea #2 de la HU Monitorear sistema.**

Tarea	
<b>Número tarea:</b> 3.	<b>Número historia:</b> 5.
<b>Nombre tarea:</b> Devolver resultados del monitoreo.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 4.
<b>Fecha inicio:</b> 24/03/10.	<b>Fecha fin:</b> 29/03/10.
<b>Descripción:</b> Se muestran los resultados obtenidos del monitoreo del sistema.	

**Tabla # 35. Tarea #3 de la HU Monitorear sistema.**

Copias de seguridad de la base de datos.

Tarea	
<b>Número tarea:</b> 1.	<b>Número historia:</b> 6.
<b>Nombre tarea:</b> Configurar la interfaz para la conexión a la base de datos.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 30/03/10	<b>Fecha fin:</b> 30/03/10
<b>Descripción:</b> Se configura la interfaz que facilitará la conexión con la base de datos.	

**Tabla # 36. Tarea #1 de la HU Copias de Seguridad de la base de datos.**

Tarea	
<b>Número tarea:</b> 2.	<b>Número historia:</b> 6.
<b>Nombre tarea:</b> Realizar copia de seguridad de la base de datos.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 5
<b>Fecha inicio:</b> 31/03/10	<b>Fecha fin:</b> 06/04/10
<b>Descripción:</b> Se realizará la copia de seguridad de la base de datos.	

**Tabla # 37. Tarea #2 de la HU Copias de Seguridad de la base de datos.**

Notificación de fallos.

Tarea	
<b>Número tarea:</b> 1.	<b>Número historia:</b> 7.
<b>Nombre tarea:</b> Verificar el funcionamiento del sistema.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 4
<b>Fecha inicio:</b> 07/04/10	<b>Fecha fin:</b> 12/04/10
<b>Descripción:</b> Se verifica el funcionamiento de cada servidor del sistema en busca de algún fallo.	

**Tabla # 38. Tarea #1 de la HU Notificación de fallos.**

Tarea	
<b>Número tarea:</b> 2.	<b>Número historia:</b> 7.
<b>Nombre tarea:</b> Enviar notificación.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 13/04/10	<b>Fecha fin:</b> 14/04/10
<b>Descripción:</b> Enviar notificación vía correo electrónico al administrador del sistema en cuanto ocurra algún fallo.	

**Tabla # 39. Tarea #2 de la HU Notificación de fallos.**

### 3.2 Tarjetas CRC (Clase – Responsabilidad - Colaborador).

Una clase es cualquier persona, objeto, evento, concepto, pantalla o reporte. Las responsabilidades de

una clase son los elementos que conoce y las acciones que realiza sobre ellos, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

### 3.3 Pruebas de aceptación.

Uno de los pilares de XP es el proceso de pruebas de aceptación. XP propone realizar pruebas constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También aumenta la seguridad de evitar efectos adyacentes no deseados a la hora de realizar modificaciones y refactorizaciones.

Las pruebas de aceptación se trazan para validar las necesidades de los usuarios y para dirigir la implementación. Se puede considerar que las historias de usuario y, por extensión, las pruebas asociadas juegan el papel de especificaciones del sistema. Cada prueba de aceptación trata de probar la funcionalidad de una historia de usuario.

Se redactan en colaboración con el cliente y el equipo de desarrollo. Nadie debe quedar aislado del proceso debido a la falta de entendimiento. Deben ser expresadas en un lenguaje sencillo que el cliente entienda en lugar de hablar en términos que solo comprendan los desarrolladores. Se escriben de manera concisa, precisa y sin ambigüedades para verificar un solo aspecto de un escenario relevante para la historia de usuario.

Para la realización de las pruebas de aceptación del sistema se utiliza la siguiente plantilla de caso de prueba:

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b>	<b>Historia de Usuario:</b>
<b>Nombre:</b>	
<b>Descripción:</b>	
<b>Condiciones de ejecución:</b>	
<b>Entrada/Pasos de ejecución:</b>	
<b>Resultados esperados:</b>	
<b>Evaluación de la prueba:</b>	

Tabla # 40. Planilla de Caso de Prueba.

Quedando conformadas así las pruebas de aceptación para cada historia de usuario en el orden a la iteración que pertenecen.

### Pruebas de aceptación para la iteración 1.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU1_IC_1	<b>Historia de Usuario:</b> Instalar Cybercluster
<b>Nombre:</b> Instalar Cybercluster	
<b>Descripción:</b> Probar que el Cybercluster se instala correctamente en el servidor especificado.	
<b>Condiciones de ejecución:</b> El usuario debe introducir una dirección IP válida y la contraseña de súper usuario correspondiente al mismo.	
<b>Entrada/Pasos de ejecución:</b> El usuario introduce los datos correctamente.	
<b>Resultados esperados:</b> Se crea una conexión ssh con el servidor y se instala el Cybercluster correctamente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla # 41. CP #1 de la HU Instalar Cybercluster.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU1_IC_2	<b>Historia de Usuario:</b> Instalar Cybercluster
<b>Nombre:</b> Instalar Cybercluster	
<b>Descripción:</b> Probar que el Cybercluster no se instala en el servidor especificado si el usuario introduce datos no válidos.	
<b>Condiciones de ejecución:</b> El cliente debe introducir la dirección IP del servidor y la contraseña de súper usuario correspondiente al mismo.	
<b>Entrada/Pasos de ejecución:</b> El cliente introduce la dirección IP del servidor incorrecta o la contraseña de súper usuario no se corresponde con la del servidor.	
<b>Resultados esperados:</b> No es posible crear la conexión ssh con el servidor y por tanto no se	

instala el Cybercluster.

**Evaluación de la prueba:** Prueba satisfactoria.

**Tabla # 42. CP #2 de la HU Instalar Cybercluster.**

### Caso de Prueba de Aceptación.

**Código:** HU2\_IBD\_1

**Historia de Usuario:** Inicializar la base de datos.

**Nombre:** Inicializar la base de datos.

**Descripción:** Probar que se inicie correctamente la base de datos.

**Condiciones de ejecución:** El cliente debe introducir un usuario y dos contraseñas iguales para crear el usuario de acceso a la base de datos.

**Entrada/Pasos de ejecución:** El cliente introduce los datos correctamente.

**Resultados esperados:** Se crea el usuario de acceso a la base de datos y se inicializa la misma.

**Evaluación de la prueba:** Prueba satisfactoria.

**Tabla # 43. CP #1 de la HU Inicializar base de datos.**

### Caso de Prueba de Aceptación.

**Código:** HU2\_IBD\_2

**Historia de Usuario:** Inicializar la base de datos.

**Nombre:** Inicializar la base de datos.

**Descripción:** Probar que se inicie correctamente la base de datos.

**Condiciones de ejecución:** El cliente debe introducir un usuario y dos contraseñas iguales para crear el usuario de acceso a la base de datos.

**Entrada/Pasos de ejecución:** El cliente introdujo dos contraseñas que no coinciden.

**Resultados esperados:** No se crea el usuario de acceso a la base de datos, ni se inicializa la misma.

**Evaluación de la prueba:** Prueba satisfactoria.

**Tabla # 44. CP #2 de la HU Inicializar base de datos.**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> HU3_GN_1	<b>Historia de Usuario:</b> Crear nodo.
<b>Nombre:</b> Crear un nuevo nodo del sistema.	
<b>Descripción:</b> Probar que se cree el nodo correctamente.	
<b>Condiciones de ejecución:</b> El cliente debe introducir la dirección IP del servidor donde se creará el nodo, el nombre del nodo y especificar quien(es) será(n) el/los servidor(es) de datos, servidor(es) de replicación y servidor(es) balanceador(es).	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos correctos en el formulario.	
<b>Resultados esperados:</b> El nodo se crea con los datos insertados por el cliente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

**Tabla # 45. CP #1 de la HU Gestionar Nodo.**

<b>Caso de Prueba de Aceptación.</b>	
<b>Código:</b> HU3_GN_2	<b>Historia de Usuario:</b> Crear nodo.
<b>Nombre:</b> Crear un nuevo nodo del sistema.	
<b>Descripción:</b> Probar que se cree el nodo correctamente.	
<b>Condiciones de ejecución:</b> El cliente debe introducir la dirección IP del servidor donde se creará el nodo, el nombre del nodo y especificar quien(es) será(n) el/los servidor(es) de datos, servidor(es) de replicación y servidor(es) balanceador(es).	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos en el formulario.	
<b>Resultados esperados:</b> Datos incorrectos, no se crea el nodo.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

**Tabla # 46. CP #2 de la HU Gestionar Nodo.**



Caso de Prueba de Aceptación.	
<b>Código:</b> HU3_GN_3	<b>Historia de Usuario:</b> Modificar nodo.
<b>Nombre:</b> Modificar los datos de un nodo del sistema.	
<b>Descripción:</b> Probar que se modifiquen los datos del nodo correctamente.	
<b>Condiciones de ejecución:</b> El cliente puede modificar tanto la dirección IP del nodo, el nombre y quien(es) será(n) el/los servidor(es) de datos, servidor(es) de replicación y servidor(es) balanceador(es). Cuando especifique los cambios en los datos debe introducir la contraseña de acceso al nodo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos correctos en el formulario.	
<b>Resultados esperados:</b> Datos correctos, se realizan las modificaciones del nodo.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla # 47. CP #3 de la HU Gestionar Nodo.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU3_GN_4	<b>Historia de Usuario:</b> Modificar nodo.
<b>Nombre:</b> Modificar los datos de un nodo del sistema.	
<b>Descripción:</b> Probar que se modifiquen los datos del nodo correctamente.	
<b>Condiciones de ejecución:</b> El cliente puede modificar tanto la dirección IP del nodo, el nombre y quien(es) será(n) el/los servidor(es) de datos, servidor(es) de replicación y servidor(es) balanceador(es). Cuando especifique los cambios en los datos debe introducir la contraseña de acceso al nodo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos en el formulario o la contraseña de acceso al nodo no es válida.	
<b>Resultados esperados:</b> Datos incorrectos, no se realizan las modificaciones del nodo.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla # 48. CP #4 de la HU Gestionar Nodo.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU3_GN_5	<b>Historia de Usuario:</b> Eliminar nodo.
<b>Nombre:</b> Eliminar un nodo del sistema.	
<b>Descripción:</b> Probar que se elimine el nodo correctamente.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que desea eliminar e introducir en el formulario la contraseña de acceso al nodo seleccionado.	
<b>Entrada/Pasos de ejecución:</b> Se selecciona el nodo y se introduce la contraseña de acceso al mismo.	
<b>Resultados esperados:</b> Datos correctos, el nodo es eliminado.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla # 49. CP #5 de la HU Gestionar Nodo.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU3_GN_6	<b>Historia de Usuario:</b> Eliminar nodo.
<b>Nombre:</b> Eliminar un nodo del sistema.	
<b>Descripción:</b> Probar que se elimine el nodo correctamente.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que desea eliminar e introducir en el formulario la contraseña de acceso al mismo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos en el formulario o la contraseña de acceso al nodo es inválida.	
<b>Resultados esperados:</b> Datos incorrectos, el nodo no es eliminado.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla # 50. CP #6 de la HU Gestionar Nodo.

Pruebas de aceptación para la iteración 2.

Caso de Prueba de Aceptación.
-------------------------------

<b>Código:</b> HU4_GSN_1	<b>Historia de Usuario:</b> Iniciar servicio del nodo
<b>Nombre:</b> Iniciar servicio del nodo.	
<b>Descripción:</b> Probar que el servicio del nodo se inicie.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que le desea iniciar el servicio e introducir en el formulario la contraseña de acceso al mismo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos correctos para acceso al nodo.	
<b>Resultados esperados:</b> El servicio del nodo es iniciado.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

**Tabla # 51. CP #1 de la HU Gestionar servicio del nodo.**

Caso de Prueba de Aceptación.	
<b>Código:</b> HU4_GSN_2	<b>Historia de Usuario:</b> Iniciar servicio del nodo
<b>Nombre:</b> Iniciar servicio del nodo.	
<b>Descripción:</b> Probar que el servicio del nodo se inicie.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que le desea iniciar el servicio e introducir en el formulario la contraseña de acceso al mismo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos para acceso al nodo.	
<b>Resultados esperados:</b> No se accede al nodo y no se puede iniciar el servicio.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

**Tabla # 52. CP #2 de la HU Gestionar servicio del nodo.**

Caso de Prueba de Aceptación.	
<b>Código:</b> HU4_GSN_3	<b>Historia de Usuario:</b> Detener servicio del nodo
<b>Nombre:</b> Detener servicio del nodo.	

<b>Descripción:</b> Probar que el servicio del nodo se detiene.
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que le desea detener el servicio e introducir en el formulario la contraseña de acceso al mismo.
<b>Entrada/Pasos de ejecución:</b> Se introducen datos correctos para acceso al nodo.
<b>Resultados esperados:</b> Se logra el acceso al nodo y se detiene su servicio.
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

Tabla # 53. CP #3 de la HU Gestionar servicio del nodo.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU4_GSN_4	<b>Historia de Usuario:</b> Detener servicio del nodo
<b>Nombre:</b> Detener servicio del nodo.	
<b>Descripción:</b> Probar que el servicio del nodo se detiene.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que le desea detener el servicio e introducir en el formulario la contraseña de acceso al mismo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos para acceso al nodo.	
<b>Resultados esperados:</b> No se logra el acceso al nodo y no se puede detener su servicio.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla # 54. CP #4 de la HU Gestionar servicio del nodo.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU4_GSN_5	<b>Historia de Usuario:</b> Reiniciar servicio del nodo
<b>Nombre:</b> Reiniciar servicio del nodo.	
<b>Descripción:</b> Probar que el servicio del nodo se reinició.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que le desea reiniciar el servicio e	

introducir en el formulario la contraseña de acceso al mismo.
<b>Entrada/Pasos de ejecución:</b> Se introducen datos correctos para acceso al nodo.
<b>Resultados esperados:</b> Se logra el acceso al nodo y se reinicia su servicio.
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

Tabla # 55. CP #5 de la HU Gestionar servicio del nodo.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU4_GSN_6	<b>Historia de Usuario:</b> Reiniciar servicio del nodo
<b>Nombre:</b> Reiniciar servicio del nodo.	
<b>Descripción:</b> Probar que el servicio del nodo se reinició.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el nodo que le desea reiniciar el servicio e introducir en el formulario la contraseña de acceso al mismo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos para acceso al nodo.	
<b>Resultados esperados:</b> No se logra el acceso al nodo y no se puede reiniciar su servicio.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla # 56. CP #6 de la HU Gestionar servicio del nodo.

Caso de Prueba de Aceptación.	
<b>Código:</b> HU5_MS_1	<b>Historia de Usuario:</b> Monitorear sistema
<b>Nombre:</b> Monitorear los nodos de base de datos del sistema.	
<b>Descripción:</b> Realizar un monitoreo de la carga de los nodos de base de datos del sistema.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el servidor replicador desde el que desea realizar el monitoreo e introducir la clave de acceso al mismo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos correctos para acceso al nodo.	

**Resultados esperados:** Se logra el acceso al nodo y se realiza el monitoreo de los nodos de base de datos del sistema.

**Evaluación de la prueba:** Prueba satisfactoria.

**Tabla # 57. CP #1 de la HU Monitorear sistema.**

Caso de Prueba de Aceptación.	
<b>Código:</b> HU5_MS_2	<b>Historia de Usuario:</b> Monitorear sistema
<b>Nombre:</b> Monitorear los nodos de base de datos del sistema.	
<b>Descripción:</b> Realizar un monitoreo de la carga de los nodos de base de datos del sistema.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar el servidor replicador desde el que desea realizar el monitoreo e introducir la clave de acceso al mismo.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos para acceso al nodo.	
<b>Resultados esperados:</b> No se logra el acceso al nodo y no se puede realizar el monitoreo de los nodos de base de datos del sistema.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

**Tabla # 58. CP #2 de la HU Monitorear sistema.**

Caso de Prueba de Aceptación.	
<b>Código:</b> HU6_CSBD_1	<b>Historia de Usuario:</b> Copia de seguridad de la base de datos
<b>Nombre:</b> Copia de seguridad de la base de datos.	
<b>Descripción:</b> Realizar copias de seguridad a la base de datos.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar la base de datos a la que desea realizar la copia de seguridad e introducir la clave de acceso al servidor de base de datos.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos correctos para acceso al servidor de base de datos.	

**Resultados esperados:** Se logra el acceso al servidor de base de datos y se realiza la copia de seguridad a la base de datos.

**Evaluación de la prueba:** Prueba satisfactoria.

**Tabla # 59. CP #1 de la HU Copia de seguridad a la base de datos.**

Caso de Prueba de Aceptación.	
<b>Código:</b> HU6_CSBD_2	<b>Historia de Usuario:</b> Copia de seguridad de la base de datos
<b>Nombre:</b> Copia de seguridad de la base de datos.	
<b>Descripción:</b> Realizar copias de seguridad a la base de datos.	
<b>Condiciones de ejecución:</b> El cliente debe seleccionar la base de datos a la que desea realizar la copia de seguridad e introducir la clave de acceso al servidor de base de datos.	
<b>Entrada/Pasos de ejecución:</b> Se introducen datos incorrectos para acceso al servidor de base de datos.	
<b>Resultados esperados:</b> No se logra el acceso al servidor de base de datos y no se puede realizar la copia de seguridad a la base de datos.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

**Tabla # 60. CP #2 de la HU Copia de seguridad a la base de datos.**

Caso de Prueba de Aceptación.	
<b>Código:</b> HU7_NF_1	<b>Historia de Usuario:</b> Notificación de fallo.
<b>Nombre:</b> Notificación de fallo.	
<b>Descripción:</b> Notificar vía correo electrónico en caso de fallo en algún nodo del sistema.	
<b>Condiciones de ejecución:</b> Acceso y comprobación del correcto funcionamiento de los nodos.	
<b>Entrada/Pasos de ejecución:</b> Se establece una conexión con los servidores y se comprueba que el servicio del nodo está iniciado correctamente.	

<b>Resultados esperados:</b> No se envía notificación por correo electrónico.
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

**Tabla # 61. CP #1 de la HU Notificación de fallo.**

Caso de Prueba de Aceptación.	
<b>Código:</b> HU7_NF_2	<b>Historia de Usuario:</b> Notificación de fallo.
<b>Nombre:</b> Notificación de fallo.	
<b>Descripción:</b> Notificar vía correo electrónico en caso de fallo en algún nodo del sistema.	
<b>Condiciones de ejecución:</b> Acceso y comprobación del correcto funcionamiento de los nodos.	
<b>Entrada/Pasos de ejecución:</b> No es posible establecer conexión con los servidores o los servicios del nodo no están funcionando correctamente.	
<b>Resultados esperados:</b> Se envía notificación por correo electrónico.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

**Tabla # 62. CP #2 de la HU Notificación de fallo.**

En el capítulo actual se realizaron las tareas de ingeniería necesarias para la implementación de las historias de usuario que permitirán la configuración y administración de un entorno de alta disponibilidad basado en Cybercluster. Además se efectuaron las pruebas de aceptación a las disímiles historias de usuario con las que cuenta el sistema, para comprobar su correcto funcionamiento.



### Conclusiones.

Con la culminación del presente trabajo de diploma se realizó el estudio del proceso de configuración y administración del Cybercluster con el fin de lograr entender su funcionamiento para facilitar el trabajo del usuario final garantizándole una mayor comodidad al administrar varios nodos desde una sola terminal, evitándole memorizar todos los comandos de configuración del sistema ahorrándole tiempo y recursos a la entidad a la que pertenezca. Para la realización de esta investigación se utilizaron métodos científicos que facilitaron:

- ❖ Seleccionar el lenguaje de programación Python, el IDE de desarrollo Eclipse y la metodología de desarrollo XP, luego de realizada una comparación entre las tecnologías necesarias para la implementación de la aplicación.
- ❖ Analizar el proceso de configuración manual de un EADBC, lográndose un sólido entendimiento del funcionamiento.
- ❖ Finalmente, obtener una aplicación de escritorio, sencilla y amigable, que permite al administrador del sistema un fácil manejo de la configuración y administración de la herramienta Cybercluster.

### **Recomendaciones.**

Con la realización del presente trabajo se desarrolla una aplicación para la configuración y administración de los servicios que brinda la herramienta Cybercluster, para mejor funcionamiento de la misma se recomienda:

- ❖ Desarrollar un entorno de instalación el cual permita ejecutar la herramienta correctamente en la estación de trabajo que cumpla con los requerimientos especificados.
- ❖ Adicionar un módulo que permita graficar la carga de transacciones de cada nodo para tener un mejor control sobre los mismos.
- ❖ Agregar un módulo que muestre el rendimiento de cada uno de los nodos para de esta forma puedan ser vigilados constantemente.

Hacer énfasis en la configuración de los nodos de bases de datos, en busca de mejorar así su eficiencia en dependencia de las máquinas donde son instalados.

---

## Referencias Bibliográficas.

1. Cameron, N. *PostgreSQL affiliates*.
2. *Diccionario básico de Informática*. (s.f.). Recuperado el 18 de 2 de 2010, de Diccionario básico de Informática: <http://usuarios.multimania.es/resve/diccioninform.htm>
3. *Java en castellano*. (s.f.). Recuperado el 20 de 2 de 2010, de Java en castellano: <http://www.programacion.com/java/>
4. Andrés, Maria Mercedes M. 2001. Citado; Disponible en: <http://www3.uji.es/~mmarques/f47/apun/node4.html>.
5. *Software Zone*. Citado; Disponible en: <http://www.softzone.es/glosario/s-t-y-u/>.
6. *LaBDA - Laboratorio de Bases de Datos Avanzadas* 31/10/2008 Citado; Disponible en: <http://basesdatos.uc3m.es/>.
7. *PostgreSQL*. (2010). Accedida el 20 de 2 de 2010, de PostgreSQL: <http://www.postgresql.org/about/>
8. Rozic, S. E. (Mayo 2004). *Bases de Datos y su aplicación con SQL*. Buenos Aires, Argentina: MP Ediciones.
9. *Slideshare.net*. (2008). Accedida el 18 de 2 de 2010, de Slideshare.net: <http://www.slideshare.net/rmonago/t1-gestion-de-datos-presentation>
10. Prado, T. J. (n.d.). Accedida el 2 18, 2010, en: [http://users.servicios.retecal.es/tjavier/python/Un\\_poco\\_de\\_Python.html](http://users.servicios.retecal.es/tjavier/python/Un_poco_de_Python.html)
11. *Paramiko*. (2009, noviembre 12). Accedida el 2, 19, 2010, en Paramiko: <http://www.lag.net/paramiko/>
12. *Wingware*. (n.d.). Accedida el 2, 20, 2010, en Wingware: <http://www.wingware.com/products>

## Bibliografía.

Zayas, Carlos Alberto de. Metodología de la Investigación Científica. Santiago de Cuba: s.n., 1995.

Camallea, Noel L. Núñez y Abalo, Ronald Coutin. Diccionario de Informática. Colombia: s.n., 2005.

León, Rolando Alfredo Hernández y González, Zayda Coello. EL Paradigma Cuantitativo de la Investigación Científica. Ciudad de la Habana: s.n., 2002.

Zadrozny, Fabio. Aptana Pydev. 2009. [En Línea] [Citado: 10 de Enero del 2010.] <http://pydev.org>.

PostgreSQL, Comunidad. PostgreSQL-es.org. 2009. [En Línea] [Citado: 20 de Enero del 2010.] <http://www.postgresql-es.org>.

Proyecto, Hermanos Carrero. Programación en Castellano. 2010. [En Línea] [Citado: 25 de Enero del 2010.] <http://www.programacion.com/noticia/1611/>.

PostgreSQL. PostgreSQL. 2010. [Citado: 18 Febrero 2010]. Disponible en: <http://www.postgresql.org/about/>.

Desarrollo Web. [Citado: 18 Febrero 2010]. Disponible en: <http://www.desarrolloweb.com/articulos/1325.php> >

Cortés, Esteban Leyva, y José Ignacio Prieto Tinoco. *Sistemas de Aplicaciones Informáticas*. Marzo, 2006 [España]: Mad, Marzo 2006.

Jiménez, Javier Alonso Albusac, y Fernando Arroba Rubio. Tecnologías Libres para Síntesis de... - Google Libros. 2006. [Citado: 18 Febrero 2010]. Disponible en: [http://books.google.com/cu/books?id=JSo9SOEy0SkC&pg=PA23&dq=ingenieria+de+software+es&hl=es&ei=IV99SbyEZPO8Qbt9L3IBQ&sa=X&oi=book\\_result&ct=result&resnum=6&ved=0CEwQ6AEwBQ#v=onepage&q=&f=false](http://books.google.com/cu/books?id=JSo9SOEy0SkC&pg=PA23&dq=ingenieria+de+software+es&hl=es&ei=IV99SbyEZPO8Qbt9L3IBQ&sa=X&oi=book_result&ct=result&resnum=6&ved=0CEwQ6AEwBQ#v=onepage&q=&f=false).

Sánchez, María A. Mendoza. Informatízate. 2004. [Citado: 18 Febrero 2010]. Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

Universidad EAFIT. (2009,11,14). Citado: 2 19, 2010, en Universidad EAFIT: <http://www.eafit.edu.co/eafitcn/centroinformatica/areas/baseDeDatos/index.shtm>

*Paramiko*. (2009, noviembre 12). Citado: 2 19, 2010, en Paramiko: <http://www.lag.net/paramiko/>

*Paramiko*. (2009, noviembre 12). Citado: 2 19, 2010, en Paramiko: <http://www.lag.net/paramiko/docs/>

**González Gutiérrez, David. 2009. Tutorial de Qt4 Designer y QDevelop.** Universidad Politècnica de Catalunya : s.n., 2009.

**Jelsoft Enterprises Ltd. 2010. Qt Center.** [En línea] 2010. [Citado el: 22 de 02 de 2010.] <http://www.qtcentre.org/>.

**Limited, Riverbank Computing. 2010.** Riverbank Computing Limited. [En línea] 2010. [Citado el: 20 de 02 de 2010.] <http://www.riverbankcomputing.co.uk/software/pyqt/intro>.

**Alvarez B., Alejandro y Valerio R., Esteban.** *Generadores de Interfaces de Usuario: QT Designer, NetBeans y Windows Forms Designer*. Universidad de Costa Rica, Escuela de Ciencias de la Computación e Informática : s.n.

## Glosario.

A continuación en orden alfabético, se muestra el significado de algunos términos usados en el documento cuyo uso no es común y puede dificultar la comprensión del mismo:

**ISAM:** Método de Acceso Secuencial Indexado. Método para almacenar información a la que se pueda acceder rápidamente.

**Log:** Registro oficial de eventos durante un rango de tiempo en particular.

**RSH:** Remote Shell. Programa de consola para ejecutar comandos en ordenadores remotos.

**SSL:** Secure Sockets Layer. Protocolo de Capa de Conexión Segura. Protocolos criptográficos que proporcionan comunicaciones seguras por una red.

**SSH:** Secure Shell. Protocolo informático que sirve para acceder a máquinas remotas.

**SFTP:** Secure File Transfer Protocol. Protocolo de red que proporciona la funcionalidad necesaria para la transferencia y manipulación de archivos sobre un flujo de datos fiable

**SWT:** *Standard Widget Toolkit*. Conjunto de componentes para construir interfaces gráficas en Java

**Swing:** Visual Swing para Eclipse. Herramienta de diseño gráfico, que consiste en un conjunto de plugins de Eclipse

**TLS:** Transport Layer Security. Seguridad de la Capa de Transporte. Protocolos criptográficos que proporcionan comunicaciones seguras por una red.

**VSAM:** Método de Acceso Virtual de Almacenamiento. Esquema de almacenamiento de IBM del sistema operativo OS/VS2

**Wal:** Write-Ahead Logging. Método estándar para el registro de transacciones.