

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMATICA**



**TESIS DE GRADO**

**TITULO: “SEGURIDAD EN BASES DE DATOS DISTRIBUIDAS  
UTILIZANDO AGENTES INTELIGENTES”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

**POSTULANTE: GUERY SANZ GUERRERO SELAEZ**

**TUTOR METODOLÓGICO: M.SC. ALDO VALDEZ ALVARADO**

**ASESOR: LIC. BRIGIDA CARVAJAL BLANCO**

**LA PAZ – BOLIVIA**

**2013**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

*Dedicado a mi familia y amigos; quienes me apoyaron en toda mi formación profesional.*

## **AGRADECIMIENTOS**

Al personal docente de la carrera de Informática de la Universidad Mayor de San Andrés, en especial a los docentes Lic. Brígida Carvajal Blanco y M. Sc. Aldo Ramiro Valdez Alvarado quienes me guiaron en el desarrollo de esta investigación.

# ÍNDICE

|   |    |
|---|----|
| CAPITULO I .....                        | 1  |
| MARCO INTRODUCTORIO.....                | 1  |
| 1.1    INTRODUCCIÓN.....                | 1  |
| 1.2    ANTECEDENTES .....               | 2  |
| 1.3    PLANTEAMIENTO DEL PROBLEMA ..... | 6  |
| 1.3.1. PROBLEMA CENTRAL.....            | 6  |
| 1.3.2. PROBLEMAS SECUNDARIOS.....       | 7  |
| 1.4    OBJETIVOS .....                  | 7  |
| 1.4.1. OBJETIVO GENERAL.....            | 7  |
| 1.4.2. OBJETIVOS ESPECÍFICOS .....      | 8  |
| 1.5    HIPÓTESIS .....                  | 8  |
| 1.5.1. DESCRIPCIÓN DE VARIABLES.....    | 8  |
| 1.6    JUSTIFICACIÓN DEL TEMA .....     | 9  |
| 1.6.1. ECONÓMICA .....                  | 9  |
| 1.6.2. SOCIAL .....                     | 10 |
| 1.6.3. CIENTÍFICA .....                 | 10 |
| 1.7    ALCANCE Y LÍMITES .....          | 11 |
| 1.7.1. ALCANCE .....                    | 11 |
| 1.7.2. LÍMITES .....                    | 11 |
| 1.8    APORTES .....                    | 12 |
| 1.8.1. PRÁCTICO.....                    | 12 |
| 1.8.2. TEÓRICO.....                     | 12 |
| 1.9. METODOLOGÍA.....                   | 13 |
| CAPÍTULO II .....                       | 16 |

|  |    |
|--|----|
| MARCO TEÓRICO .....  | 16 |
| 2.1 SEGURIDAD .....  | 16 |
| 2.1.1. METODOLOGÍAS DE TESTEO DE SEGURIDAD .....                     | 16 |
| 2.2 BASES DE DATOS DISTRIBUIDAS .....                                | 25 |
| 2.2.1. ARQUITECTURA DE UN SISTEMA DE BASE DE DATOS DISTRIBUIDAS..... | 26 |
| 2.2.2. TIPOS DE ALMACENAMIENTO .....                                 | 30 |
| 2.2.3. NIVELES DE TRANSPARENCIA.....                                 | 32 |
| 2.2.4. PROCESAMIENTO DISTRIBUIDO DE CONSULTAS .....                  | 33 |
| 2.2.5. RECUPERACIÓN .....  | 33 |
| 2.3 INTELIGENCIA ARTIFICIAL Y AGENTES INTELIGENTES .....             | 34 |
| 2.3.1. INTELIGENCIA ARTIFICIAL .....                                 | 34 |
| 2.3.2. AGENTES INTELIGENTES .....                                    | 40 |
| 2.4 METODOLOGÍA PROMETHEUS .....                                     | 53 |
| 2.4.1. ETAPAS BÁSICAS.....   | 53 |
| 2.5 MODELO VISTA CONTROLADOR.....                                    | 58 |
| CAPITULO III .....   | 60 |
| MARCO APLICATIVO.....  | 60 |
| 3.1 INTRODUCCIÓN .....   | 60 |
| 3.1.1. PRINCIPALES AMENAZAS DE SEGURIDAD.....                        | 61 |
| 3.2 ESPECIFICACIÓN DEL SISTEMA .....                                 | 62 |
| 3.2.1. IDENTIFICACIÓN DE ACTORES .....                               | 62 |
| 3.2.2. ESCENARIOS .....  | 64 |
| 3.2.3. DEFINICIÓN DE METAS .....                                     | 65 |
| 3.3 ARQUITECTURA DEL SISTEMA .....                                   | 67 |
| 3.3.1. DETERMINACIÓN DE TIPOS DE AGENTES.....                        | 67 |

|   |    |
|---|----|
| 3.3.2. ANÁLISIS DESCRIPTIVO GENERAL .....                               | 69 |
| 3.4 DISEÑO DETALLADO DEL SISTEMA.....                                   | 71 |
| 3.4.1. DEFINICIÓN DETALLADA DE LAS ACTIVIDADES DEL AGENTE INTELIGENTE . | 71 |
| 3.4.2. DETALLE DE LAS RELACIONES ENTRE AGENTES, DATOS Y PROTOCOLOS..    | 72 |
| 3.5 DESARROLLO DEL SISTEMA DE PRUEBA .....                              | 74 |
| 3.5.1. MODELO VISTA CONTROLADOR.....                                    | 74 |
| 3.5.2. BASE DE DATOS DISTRIBUIDA.....                                   | 75 |
| CAPITULO IV .....   | 79 |
| PRUEBA DE HIPÓTESIS .....   | 79 |
| 4.1 PRUEBA DE FUNCIONALIDAD DEL AGENTE INTELIGENTE .....                | 80 |
| 4.1.1. PRUEBA DE SEGURIDAD SIN EL AGENTE .....                          | 80 |
| 4.1.2. PRUEBA DE SEGURIDAD CON EL AGENTE EN EL SISTEMA “A” .....        | 84 |
| 4.1.2. PRUEBA DE SEGURIDAD CON EL AGENTE EN EL SISTEMA “B” .....        | 87 |
| 4.1.3. GRAFICA RESULTANTE COMPARATIVA DE TIEMPOS DE EJECUCIÓN .....     | 90 |
| 4.1.4. TABLA COMPARATIVA DE ATAQUES .....                               | 91 |
| CAPITULO V .....  | 93 |
| CONCLUSIONES Y RECOMENDACIONES .....                                    | 93 |
| 5.1 CONCLUSIONES.....   | 93 |
| 5.2 RECOMENDACIONES .....   | 94 |
| ÍNDICE DE FIGURAS .....   | 96 |
| CAPITULO 2 .....  | 96 |
| CAPITULO 3 .....  | 96 |
| CAPITULO 4 .....  | 97 |
| BIBLIOGRAFÍA .....  | 98 |

## RESUMEN

La investigación se centra en el estudio de la seguridad de sistemas con bases de datos distribuidas, los principales riesgos que presentan dichos sistemas y el uso de agentes inteligentes para el control de la seguridad de los mismos. Las bases de datos distribuidas son un método de almacenamiento de información descentralizada la cual se apoya en dividir la información del sistema entre los diferentes nodos del sistema dando a cada uno de ellos autonomía con relación de los otros; por otro lado también es necesario que dichos nodos del sistema trabajen juntos como si fueran uno; pues son parte del sistema en sí. Los agentes inteligentes son una rama de la Inteligencia Artificial, la cual propone la elaboración de un programa que emule la inteligencia de un ser humano pues sería capaz de tomar decisiones racionales y además poder aprender a través de los sucesos que ocurran en el sistema. Por lo cual se tomó en cuenta la posibilidad de que la seguridad para un sistema de bases de datos distribuida pueda ser monitoreada y controlada por un agente inteligente teniendo en cuenta las grandes capacidades de esta metodología dentro de la Inteligencia Artificial. Al final de la investigación se presentó un resultado favorable en la implementación del agente en sistemas de bases de datos distribuidas, ya que los resultados que presentaron las pruebas en los sistemas donde se implementó el agente para el control en la seguridad de las bases de datos, comprobaron la hipótesis de la investigación. El agente incrementaba considerablemente la seguridad de la información en dichos sistemas; por otro lado también se notó un incremento en el tiempo de respuesta de estos sistemas; a pesar de que el agente consume bastantes recursos del sistema, es eficiente y eficaz incluso en sistemas que no cuenten con servidores modernos.

*Palabras claves: Base de datos distribuidas, agentes inteligentes, seguridad, inteligencia artificial, PROMETHEUS.*

## ABSTRACT

The research focuses on the study of the security of systems with distributed databases, the main risks of such systems and the use of intelligent agents to control the safety thereof. The distributed database is a storage method of decentralized information which is based on dividing the system information between different nodes giving each autonomy from the other system, on the other hand it is also necessary that such system nodes work together as one, they are part of the system itself. Intelligent agents are a branch of Artificial Intelligence, which proposes the development of a program that emulates the intelligence of a human being it would be able to make rational decisions and also to learn through the events that occur in the system. By which took into account the possibility that the security system distributed databases can be monitored and controlled by an intelligent agent considering the great capabilities of this methodology in Artificial Intelligence. At the end of the investigation provided a favorable result in the implementation of agent systems distributed databases as the test results presented in systems where the agent was implemented to control the security of databases, proved the hypothesis of the research. The agent significantly enhanced the information security in such systems , on the other hand an increase is also noted in the response time of these systems , although the agent consumes many system resources is efficient and effective even in systems not have modern servers.

*Keywords: Database distributed, intelligent agents, security, artificial intelligence, PROMETHEUS.*

# CAPITULO I

## MARCO INTRODUCTORIO

### 1.1 INTRODUCCIÓN

La importancia de los datos en la vida cotidiana se remonta desde el inicio mismo de la historia, porque a través de ellos las personas tomaron, toman y tomarán decisiones importantes en cuanto a sus vidas e incluso aquellas que pueden marcar hasta el destino de un pueblo, ciudad o nación.

Por este hecho las personas han ido innovando en el almacenamiento de dichos datos, guardándolos primero en papel, los cuales agruparon en libros y estos en bibliotecas y/o almacenes de información.

Hoy en día gracias a la tecnología, el almacenamiento de la información ha evolucionado mucho a lo largo del tiempo, hasta llegar a las bases de datos o bancos de datos los cuales son un conjunto de datos relacionados a un mismo contexto y organizados sistemáticamente. A esta tecnología se le sumaron varias metodologías, pero en la que se concentrará esta investigación será en la de bases de datos distribuidas (BDD) y agentes inteligentes para verificar la integridad y seguridad en las bases de datos distribuidas.

Las bases de Datos Distribuidas son un conjunto de bases de datos relacionadas lógicamente y situadas en diferentes espacios lógicos e interconectadas por una red de comunicación, lo que se trata de conseguir al distribuir las bases de datos es que cada una de ellas tenga un funcionamiento autónomo y también funcionen correctamente en conjunto, para el correcto

funcionamiento e integridad de los datos en este conjunto de bases de datos utilizaremos agentes inteligentes los cuales estarán encargados de revisar los accesos sin autorización (Anonimo, Linux Máxima Seguridad – Edición especial, 2011), para ello tendremos en cuenta las principales amenazas a los sistemas con bases de datos (SQL-injection, SCRIP-injection, PHP-injection e Ingresos sin autorización). (Herzog, OSSTMM3. Manual de la metodología abierta de testeo de seguridad, 2010)

Los agentes inteligentes son programas autónomos o semiautónomos, los cuales están compuestos por tres componentes: perceptores, procesador y efectores. Los perceptores son todas las partes o módulos que reciben la información o los estímulos del ambiente, el procesador se encarga de revisar la información recogida por los perceptores y determinar la acción racional correspondiente a este estímulo y sus efectores son las partes o módulos que realizan las acciones que el procesador evalúa pertinente según el estímulo.

## **1.2 ANTECEDENTES**

Todo se remonta a la antigüedad donde ya existían bibliotecas y muchas clases de registros de datos, en esos tiempos los datos eran sobre cosechas, censos, propiedades, entre otros, pero dicho almacenamiento de datos era ineficiente y poco confiable ya que no se podía contar con máquinas que automaticen el proceso. Sin embargo hasta 1884 con la máquina creada por el estadista Herman Hollerith que constaba de tarjetas perforadas, se pudo automatizar ciertas tareas y así fue como se vio la necesidad de crear máquinas que automaticen los procesos repetitivos.

El concepto en sí de una base de datos empezó a partir de la necesidad de almacenar gran cantidad de datos, frente a esta gran necesidad y con la creación de los computadores se vio evidente que este problema de

almacenamiento se podría solucionar puesto que esta cantidad de datos se los podría almacenar lógicamente en esta nueva máquina y con esto solucionar muchos problemas. Es por esto que la Informática siempre ha estado ligada a la información y/o datos que los computadores almacenan y/o procesan.

En 1960 y los años posteriores a la época de los 60, los computadores bajaron de precio para posibilitar a que las empresas privadas y las personas particulares pudieran adquirir dichos computadores. Debido a esto se popularizaron los discos que permitían mover información de un lugar a otro (claro que dicha información era relativamente pequeña), en esta misma época se dio inicio a las bases de datos de red y jerárquicas las cuales guardaban los datos en estructuras de árboles y listas.

Después de esto se empezó a estudiar la manera de mejorar este nuevo método de almacenamiento de datos por lo que se empezó a desarrollar más y más técnicas innovadoras, hasta que en 1970 Edgar Frank Codd escribió un artículo con nombre: "A Relational Model of Data for Large Shared Data Banks" ("Un modelo relacional para grandes bancos de datos compartidos"), a partir de esta publicación y otros estudios y artículos relacionados se llegó a definir el modelo de base de datos relacional así como las reglas que se necesitan para el uso correcto de dicho modelo y la administración del mismo para los encargados de estas bases de datos. (Valencia, 2011)

Como consecuencia de este nuevo método poco después en la misma época, Lawrence J. Ellison viendo el nuevo modelo creado por Edfar Frank Codd decidió crear una nueva empresa de desarrolló "Relational Software System" mejor conocida actualmente como "Oracle Corporation", esta empresa desarrolló un sistema de gestión de bases (gestor de base de datos) al cual le dieron el nombre de la misma compañía ORACLE.

Posterior a esto en los años 80 se desarrolló el lenguaje SQL (Structured Query Language) o lo que es lo mismo un lenguaje de consultas o lenguaje declarativo de acceso a bases de datos relacionales, dicho lenguaje permite realizar consultas en las bases de datos con el fin de extraer solo los datos necesarios para el funcionamiento de los programas. Al ver estos avances en la nueva tecnología de bases de datos en los años 90 se fueron creando poco a poco nuevos gestores de bases de datos, como los de la empresa Microsoft (SQL server, Access) y también se crearon algunos para software libre (postgreSQL, MySQL). (Herzog, OSSTMM2.1. Manual de la metodología abierta de testeo de seguridad, 2000)

Después de observar que la cantidad de los datos a almacenar se incrementaba cada vez más se necesitaba mejorar esta metodología hasta ese entonces funcional de las bases de datos y así se fueron aumentando otras tecnologías como el Paradigma Orientado a Objetos y la programación modular.

Por lo cual nació la nueva forma de almacenamiento de datos: Las Bases de datos Orientados a Objetos (BDOO) y Bases de Datos Distribuidas (BDD).

La investigación se centrará en las Bases de Datos Distribuidas (BDD), las cuales son un conjunto de Bases de Datos relacionadas lógicamente y distribuidas en diferentes espacios lógicos e interconectados entre sí a través de una red de comunicación. Estas bases de datos pueden funcionar autónomamente y aun así permiten realizar operaciones locales (en cada base de datos) y distribuidas (en otra base de datos de la red), lo importante en estas bases de datos es que la relación lógica entre ellas debe ser clara y bien estructurada para no producir errores en la topología de la base de datos. En esta investigación para conservar la seguridad de los datos en el sistema de Bases de Datos Distribuidas se utilizará a los Agentes Inteligentes los cuales regularán el flujo de los datos, controlarán la seguridad en la base de datos

implementando controles de restricción a los datos entrantes para prevenir inyección de código, interrupción en el servicio, controlando el ingreso de usuarios no autorizados y restringir el acceso a los datos por los usuarios según nivel de acceso. (Camfar, 2010)

El estudio de agentes tuvo sus comienzos en trabajos de conductistas psicológicos como Skinner en 1953 los cuales trataban de reducir la psicología a correlaciones de Entrada y Salida. Con lo cual se llegó a la metáfora de la computadora de los agentes de Putnam en 1960 y Lewis en 1966, así es como poco a poco se fue avanzando en el desarrollo de esta metodología hasta que en 1983 Jon Doyle propuso que el diseño de agentes racionales sea la meta o uno de los propósitos principales a conseguir en la Inteligencia Artificial. Esta propuesta fue tomada en consideración pero no consolidada, poco después en 1988 Horvitz fue más específico al sugerir que el empleo de la racionalidad brindaría la máxima utilidad como fundamento de la Inteligencia Artificial. (Choque, 2013)

Así fue como se declaró los parámetros mínimos de racionalidad en que una entidad sea considerada un agente, en 1991 Russell y Wefald abordaron explícitamente las posibles arquitecturas de los agentes.

Finalmente se llega a la conclusión o definición que un agente es una entidad capaz de percibir su entorno, de procesar la información (percepciones) recogida y determina la salida o respuesta racional que efectuará, es decir que analiza cual es la mejor acción, teniendo en cuenta la maximización del resultado, la eficiencia y la eficacia.

Existen muchas clasificaciones entre los agentes inteligentes según su arquitectura y según su comportamiento, pero en la que se basará esta

investigación será en los agentes reactivos de software. La arquitectura básica de este agente se resume en las siguientes partes: receptores, efectores y el núcleo o centro de procesamiento. Teniendo en cuenta todo esto, el agente reactivo debe basarse en las percepciones del ambiente para tomar las decisiones con respecto a las acciones que intervendrán en el ambiente.

Los receptores se encargan de percibir todos los estímulos del ambiente y transmitirlos al módulo de procesamiento del agente el cual evaluará la situación y buscará la respuesta o solución más óptima o racional una vez que se encuentre la respuesta más óptima, es turno de los efectores los cuales realizarán las acciones necesarias en el ambiente para cumplir con la respuesta o solución seleccionada.

### **1.3 PLANTEAMIENTO DEL PROBLEMA**

Las bases de datos distribuidas son una arquitectura eficiente de almacenamiento de datos, pero existen problemas inherentes a esta arquitectura, el principal es la conexión entre las bases de datos propias del sistema, ya que para poder tener una interconexión entre las bases de datos las restricciones de acceso no pueden ser tan estrictas como en un sistema centralizado. De esta forma se logra que el flujo de datos entre ellas no se vea afectado por las restricciones, la red que conecta estas bases de datos pueden dañar la información que ingresa al sistema (aumentando ruido a la información).

#### **1.3.1. PROBLEMA CENTRAL**

Existen muchos problemas en cuanto a la seguridad de sistemas informáticos. Para el desarrollo de esta investigación se centrará en sistemas con bases de datos distribuidas y teniendo como problemática principal la siguiente pregunta:

*¿Cómo se puede incrementar el control de la seguridad en la conexión y cualquier proceso de consulta o extracción de información concerniente o almacenada en las bases de datos distribuidas?*

### **1.3.2. PROBLEMAS SECUNDARIOS**

- Múltiples conexiones entre bases de datos del sistema causa insuficiente seguridad en el acceso a dichas base de datos.
- Los usuarios no autorizados, hackers o cualquier usuario que entra ilícitamente al sistema e ingresen códigos de programación no autorizados, para sustraer información registrada en las bases de datos.
- Existen muchas metodologías para el testeado de seguridad lo cual hace difícil elegir una de ellas y darle prioridad en la investigación.
- La red que conecta las bases de datos puede dañar los datos (aumentar ruido).
- La topología de la red para que las bases de datos del sistema se comuniquen elevan la complejidad en la estructuración del sistema.
- La topología de la red provoca dificultad en el manejo de datos entre bases de datos del sistema.

## **1.4 OBJETIVOS**

### **1.4.1. OBJETIVO GENERAL**

Plantear un agente inteligente que incremente el control de la seguridad en la conexión y cualquier proceso de consulta o extracción de información concerniente o almacenada en las bases de datos distribuidas.

### **1.4.2. OBJETIVOS ESPECÍFICOS**

- Establecer cuáles son las principales amenazas que afectan o comprometen a la seguridad en bases de datos distribuidas.
- Proporcionar al agente de un módulo que se encargue de controlar los accesos al sistema.
- Analizar los posibles casos en los que los agentes inteligentes deberán interactuar con otros agentes o con un usuario registrado del sistema para resolver algún problema.
- Analizar todas las metodologías para testeos de seguridad que puedan ser automatizadas con el agente inteligente.

### **1.5 HIPÓTESIS**

*El agente inteligente aumenta el control de la seguridad en la conexión y cualquier proceso de consulta o extracción de información que involucren a las bases de datos distribuidas, centrándose en el control de SQL-injection, ataque por Fuerza bruta y la verificación en la identificación de los usuarios incrementando el reconocimiento de la IP o MAC*

#### **1.5.1. DESCRIPCIÓN DE VARIABLES**

- Seguridad en el acceso a bases de datos distribuidas: Se debe tomar en cuenta que la seguridad de bases de datos distribuidas es afectada por las siguientes amenazas: "SQL-Injection, uso de caracteres especiales, ruido, errores de conectividad, caídas del sistema y replicación de la información".
- Agente Inteligente: Esta tecnología es muy buena y versátil ya que puede ser usada en cualquier problema que tenga el mismo patrón o recurrencia. Es por eso que usar a los agentes inteligentes para el

control de las bases de datos distribuidas ayudará mucho a mejorar la integridad de datos y su seguridad.

## **1.6 JUSTIFICACIÓN DEL TEMA**

### **1.6.1. ECONÓMICA**

La funcionalidad de un sistema con base de datos distribuida es eficiente, ya que cada módulo descentralizado funciona independientemente de los otros o en conjunto y su funcionalidad es mayor por el hecho de que el sistema descentralizado se encuentra en la red local; lo cual hará que los computadores conectados a esta red trabajen con mayor fluidez (tiempo de procesamiento del sistema) con el sistema. Las empresas y/o instituciones que implementen esta metodología de bases de datos distribuidas tendrían una significativa reducción en los costos, ya que esta metodología aprovecha o incrementa el uso de las redes locales reduciendo considerablemente el uso o consumo de internet para sistemas informáticos, esto reducirá la necesidad de un ancho de banda considerablemente alto, para un flujo de datos masivo que existe en sistemas informáticos (transacciones de información, subida o descarga de archivos, etc.). Otro punto importante es la estabilidad en la conexión para un buen flujo de información y teniendo en cuenta que los precios y la estabilidad del internet en el país no son de mala o pésima calidad y que dicho uso implicara un gran presupuesto para el servicio de internet por parte de las empresas.

Por lo cual cuando se usa un sistema de bases de datos distribuidas el sistema trabajará casi en su totalidad en una red local o en su defecto en internet con un uso limitado de usuarios, teniendo una mayor estabilidad en cuanto a su funcionamiento y limitará el uso de internet para transacción de datos entre los

nodos del sistema, teniendo como resultado una reducción en el presupuesto de dicho servicio.

### **1.6.2. SOCIAL**

En Bolivia un sistema con esta característica aumentaría la productividad de las empresas pues reduciría el consumo de ancho de banda de internet, teniendo en cuenta que el servicio de internet actualmente es muy deficiente en nuestro país y con un costo relativamente elevado, el reducir el uso del mismo reduciría costos. Al trabajar localmente con una parte de la base de datos el procesamiento de los sistemas incrementaría y no dependería del internet para su funcionamiento, con lo cual empresas con sucursales o divisiones en los distintos departamentos o municipios de Bolivia podrían tener un funcionamiento normal incluso con un ancho de banda relativamente pequeño, lo cual beneficiaría considerablemente el desarrollo de estas empresas en el país.

### **1.6.3. CIENTÍFICA**

El desarrollo de estas dos tecnologías -bases de datos distribuidas y agentes inteligentes- fue de mucha ayuda para el desarrollo global de la informática, pero lamentablemente dichas tecnologías no fueron utilizadas en conjunto para incrementar la seguridad, debido a que los agentes inteligentes son mayormente usados para ayudar en tareas repetitivas o morosas, brindando alertas tempranas a los usuarios. Por otra parte las bases de datos distribuidas fueron diseñadas para sistemas que necesitan ser ubicados en lugares o espacios lógicos relativamente alejados, los cuales pueden funcionar solos o al unísono con mucha eficiencia. Por esta razón los sistemas que cuentan con

bases de datos distribuidas y agentes inteligentes solo usan a los agentes para tareas de alerta o de apoyo para los usuarios.

Para este proyecto de investigación se usaran estas dos tecnologías para incrementar la funcionalidad de las bases de datos distribuidas, haciéndolas más seguras en dos puntos fundamentales: controlando el acceso a ellas y verificando los datos ingresado al sistema.

## **1.7 ALCANCE Y LÍMITES**

### **1.7.1. ALCANCE**

Acciones que el agente inteligente realizará en el sistema para incrementar la seguridad en las bases de datos distribuidas:

- Control de conexión a las bases de datos.
- Control de identificación de usuarios.
- Verificación de los datos que se ingresarán al sistema
- Control de datos o caracteres extraños.
- Verificación de SQL-injection
- Verificación de duplicidad
- Validación de duplicidad

### **1.7.2. LÍMITES**

Restricciones que el Agente Inteligente tendrá en el monitoreo del sistema de bases de datos distribuidos:

- El agente no controlara otros tipos de injection (php, java script) que no sea el que se menciona en el subtítulo anterior.

- No se tendrá un control minucioso en los protocolos que se use en la red del sistema, por lo que no se podrá comprobar o prever el uso de tecnologías de estenografía, encriptado u otros métodos que se pueden utilizar para ocultar información en los protocolos de la red.
- Restricciones en el control de otros tipos de monitoreo para el aumento de la seguridad del sistema, estas otras restricciones las cuales no fueron tomadas en cuenta debido a no ser de gran relevancia o abarcar un nivel de seguridad con mayor detalle en la parte física de los sistemas.
- El agente se limitará a trabajar en un entorno web, orientado a objetos y por ende con sistemas basados en el paradigma orientado a objetos dentro de la codificación del sistema.

## **1.8 APORTES**

### **1.8.1. PRÁCTICO**

El aporte práctico de la investigación estará dado en el desarrollo de un módulo o clase para un sistema que cumpla con las especificaciones (Desarrollo en PHP y codificado en base al paradigma orientado a objetos), esto permitirá ver la funcionalidad de esta investigación y como el desarrollo de este módulo o clase se lo realizará en PHP, el código podrá subirse a la nube. Para fines prácticos en la comprobación o denegación de la hipótesis que se propone en esta tesis el sistema estará en una red local.

### **1.8.2. TEÓRICO**

El aporte teórico de la investigación será toda la investigación y recopilación acerca de las metodologías abiertas de testeo de seguridad que se podrán

cumplir o implementar utilizando a los agentes para este propósito. Por otra parte el análisis para el desarrollo de la base de conocimiento del agente también será un gran aporte, pues dichas reglas estarán muy asociadas a los procesos que son factibles de automatizar y hasta qué punto es posible esto por lo cual será un aporte teórico para otras personas que se interesen en este tema.

## 1.9. METODOLOGÍA

Para el desarrollo de la investigación se utilizará el método cuantitativo, ya que esta metodología se adaptará muy bien al desarrollo que se realizará en esta investigación debido a los datos generados, los cuales serán analizados para verificar o refutar la hipótesis planteada en esta investigación.

Las fases que se utilizarán en esta investigación se describen a continuación:

**Idea:** La idea principal de esta investigación se expresa en la siguiente frase: “EL INCREMENTO EN EL CONTROL DE LA SEGURIDAD, LA CONEXIÓN Y CUALQUIER PROCESO QUE INVOLUCRE A LAS BASES DE DATOS”

**Planteamiento del Problema:** El problema planteado en la investigación es realizar un análisis de los datos o metodologías abiertas existentes para el testeado de la seguridad en bases de datos distribuidas, así como el uso de agentes para su aplicación.

**Planteamiento de Hipótesis:** La hipótesis planteada en el primer capítulo servirá también para el desarrollo de la metodología y será la siguiente: “El

*agente inteligente aumenta el control de la seguridad en la conexión y cualquier proceso de consulta o extracción de información que involucren a las bases de datos distribuidas, centrándose en el control de SQL-injection, ataque por Fuerza bruta y la verificación en la identificación de los usuarios incrementando el reconocimiento de la IP o MAC”.*

**Inmersión Inicial en el Campo:** En esta fase de la metodología se analizará la idea y el planteamiento del problema anteriormente especificados, para tener una vista más clara de lo que se espera conseguir en esta investigación.

**Concepción del diseño de Estudio:** Esta fase nos ayudara a programar a grandes rasgos cómo se llevara a cabo las siguientes etapas de la investigación, entre las cuales se definirá la manera de abordar el estudio de la problemática.

**Recolección de Datos:** Esta fase es en la que se definirá cual de toda la información recolectada será estudiada en esta investigación, esto nos ayudará a centrarnos en la problemática principal y delimitar el alcance de la investigación.

**Análisis de Datos:** Una vez establecida y recopilada toda la información necesaria para esta investigación, se pasará al proceso de análisis de dicha información donde se estudiarán cuáles son los tópicos o métodos que podrán ser automatizados mediante los agentes inteligentes.

Para el desarrollo del agente inteligente utilizaremos la metodología “prometheus” la cual fue creada por Lin Padgham y Michael Winikoff. Dicha metodología fue desarrollada conjuntamente con la empresa “Agent Oriented

Software” en su ambiente de desarrollo TDT (Prometheus Design Tool), en este ambiente actualmente la metodología sigue en desarrollo y esta metodología tiene tres etapas básicas las cuales son:

1. Especificación del sistema.
2. Arquitectura del sistema.
3. Diseño detallado del sistema.

Dicha metodología fue creada a partir de la técnica de desarrollo de AUML se la utiliza como técnica para el desarrollo de los agentes, por lo que también será utilizada con el mismo propósito para representar los protocolos de interacción del agente inteligente. (Anonimo, Catarina.udlap.mx, 2012)

Posteriormente para la implementación del agente, utilizaremos como herramienta de desarrollo el lenguaje PHP conjuntamente con el paradigma orientado a objetos.

Se utilizará dicho lenguaje de programación web para desarrollar un sistema con bases de datos distribuidas. Este sistema servirá para la comprobación o denegación de la hipótesis planteada, también se utilizará algunas herramientas relativamente modernas entre ellas HTML 5, JQUERY, AJAX; asegurando que el agente diseñado en esta investigación pueda ser implementado en sistemas web de última generación con el único requisito de que dichos sistemas web sean compatibles con el lenguaje de desarrollo web PHP en el que será desarrollado el agente.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1 SEGURIDAD**

##### **2.1.1. METODOLOGÍAS DE TESTEO DE SEGURIDAD**

Las metodologías de testeo de seguridad son un conglomerado de test para la comprobación de la seguridad en un sistema informático, dichas metodologías realizan un análisis de los sistemas atacando a los mismos desde el exterior, este ataque sirve al desarrollador o equipo de desarrolladores para verificar los puntos débiles de sus sistemas, así también brinda una información muy importante al desarrollador ya que le da un estimado del punto de colapso del sistema en el cual el sistema se colgara por la presión aplicada al mismo.

Estas metodologías de teste son en realidad un conjunto de reglas y/o lineamientos para ver “Cuándo, Qué y Cuáles” elementos son testeados para la comprobación de la seguridad de un sistema. Por otra parte las limitaciones para estas metodologías de testeo de seguridad en la cual se basara esta tesis reside o radica en la diferencia considerable de los dos tipos de testeo “Testeo Externo a Interno” y “Testeo Interno a Interno”, esta diferencia principalmente se encuentra en los privilegios de acceso, los objetivos y resultados de cada uno de estos dos métodos de testeo, ya que estos dos tipos de testeo divergen mucho entre sí. Para el Desarrollo y la comprobación o negación de la hipótesis presentada en esta investigación se utilizará la metodología de “Testeo Externo a Interno” ya que satisface tanto al desarrollo de los objetivos planteados, la

resolución de los problemas presentados y a la comprobación o refutación de la hipótesis de la presente tesis.

Por otra parte para que cualquiera de estas normas o formas de testeo sea considerado un test OSSTMM (Open Source de Testeo de Seguridad Manual de Metodología) por la ISECOM (Institute For Security And Open Methodologies) debe cumplir con las siguientes condiciones para que sea considerado una metodología de testeo certificada por la ISECOM:

- Resultado cuantificable.
- El proceso debe ser consistente y repetible según condiciones.
- Valido sin importar el tiempo en el que se lo realice.
- Basado en el mérito del desarrollador que lo elaboró, no así en marcas comerciales y/o empresas.
- Exhaustivo
- Concordante con las leyes individuales y locales del derecho humano de la privacidad

Por otra parte ISECOM no asegura que al aplicar las OSSTMM se brinda una protección infalible por lo cual tampoco estas normas se podrán constituir como un apoyo legal en caso de cualquier problema jurídico. (Herzog, OSSTMM3. Manual de la metodología abierta de testeo de seguridad, 2010)

#### **2.1.1.1. PROCESO DE ANÁLISIS DE SEGURIDAD**

El proceso del análisis de seguridad a aplicarse se evaluará o se centrará en evaluar las siguientes áreas las cuales reflejaran el nivel de seguridad presente en el sistema. Dichas áreas son las siguientes:

| <b>Áreas de seguridad</b> | <b>Descripción</b>  |
|---------------------------|---|
| Visibilidad               | Hace referencia a lo que puede verse, ingresarse o modificarse, así como las ondas de radio, luz por encima del espectro visible, dispositivos de comunicación, entre otros.  |
| Acceso                    | Punto de entrada de seguridad o acceso al nivel de seguridad, lo cual no puede ser una barrera física, esto puede incluir pero no limitar una ventana, pagina web de acceso a la red, protocolos de comunicación para la conexión de computadoras en una red.     |
| Confianza                 | Este punto se especializa al nivel de seguridad y confianza que la empresa tiene a sus empleados, en este punto se incluye la clase, control de acceso, no-repudio, confidencialidad e integridad, y otros que provean a los usuarios de confianza en el sistema. |
| Autenticación             | El usuario que quiera ingresar al sistema deberá autenticarse primero para poder acceder a los datos almacenados en el sistema, por lo cual debe existir un módulo en el sistema que se especialice en la autenticación de los usuarios.                          |
| No-repudio                | Provee una garantía que ninguna persona o institución bloqueará el acceso o el movimiento en la navegación del sistema a ningún usuario que requiera el servicio del sistema.   |

|                  |   |
|------------------|---|
| Confidencialidad | Esta es la cláusula del sistema que provee a los usuarios la seguridad de que cualquier interacción con el sistema, por parte de los usuarios o cualquier información privilegiada del mismo sea vista por las partes involucradas en dicha transacción, protegiendo la información privilegiada de usuarios invasores o personas mal intencionadas que quieran acceder a esta información. |
| Privacidad       | Solo el sistema y sus partes involucradas tienen acceso a sus detalles y a la información que se almacena en este sistema.  |
| Autorización     | Los procesos tienen una justificación de negocio el cual define los niveles de autorización de cada usuario inscrito en el sistema y le da los permisos correspondientes para extraer los datos.  |
| Integridad       | Es la seguridad de que el proceso realizado por el sistema tiene una finalidad y no puede ser cambiado, redirigido, modificado o alterado por ningún usuario que no tenga la autorización correspondiente.  |
| Seguridad        | Es el elemento que verifica que un proceso no puede dañar a otros sistemas o procesos que no estén relacionados al proceso original que se modificó o se está modificando.  |
| Alarma           | Actividad o proceso independiente que sirve para informar al usuario que las actividades no siguen las  |

|  |   |
|--|---|
|  | normas o violan los procesos propios del sistema. |
|--|---|

**Tabla 1 Áreas de la seguridad**  
**Fuente: Herzog, OSSTMM3. 2010**

### **2.1.1.2. MAPA DE TESTEO DE SEGURIDAD**

El mapa de seguridad abarca los seis campos principales para el testeado de la seguridad, si se cubren cada una de estas posibles irrupciones en la seguridad se podrá incrementar la seguridad de cualquier sistema, no obstante dicho incremento no simboliza que se el sistema será infalible contra el o los ataques de usuarios maliciosos.



**Figura 2. 1 Mapa de Seguridad**  
**Fuente: OSSTMM v2.1**

Estas seis áreas o campos principales mostrados en la figura 2.1 son base para el control de la seguridad y serán estudiadas a continuación:

- a) Seguridad de la Información.

Este es uno de los puntos principales en la tesis ya que en este punto se evaluará toda la información ingresada en los sistemas, esto implica la revisión

que se hará mediante el agente controlando caracteres especiales, SQL-injection o cualquier información ingresada en el sistema que dañe la integridad de las bases de datos pertenecientes al sistema.

La revisión de privacidad de dicha información donde la seguridad estará presente para defender el acceso a la información solo por usuarios autorizados y con el nivel necesario de autorización; así como la recolección de documentos implica la verificación de la información testeada que pertenece a los diferentes niveles del sistema. El tiempo dedicado para la búsqueda de esta información y la verificación de la misma dependerá del tamaño del sistema y organización que alimente dicha información. (Herzog, OSSTMM3. Manual de la metodología abierta de testeo de seguridad, 2010)

b) Seguridad de los procesos.

En esta área se debe realizar la prevención de la interferencia o modificación de los procesos que se generan en los sistemas,

El testeo de solicitud es el método que verifica la seguridad en el acceso o los privilegios de acceso a los módulos del sistema y/o información privilegiada de la organización.

El testeo de sugerencia dirigida es el que enumera los puntos de acceso privilegiados para comprobar el acceso a dichos módulos y acceso físico a la organización (email, teléfono, correo, entre otros).

EL testeo de personas confiables se lo hace a las personas que trabajan en la organización para ver el nivel de confianza que se les puede dar a dichas personas así como el nivel de privilegios de acceso.

### c) Seguridad en las Tecnologías de Internet

En esta área se evaluará la seguridad que se puede implementar en el acceso que se puede tener por medio de las tecnologías de Internet.

La logística y control reduce la información de los faso positivos y de los negativos revisando los protocolos para el envió de datos como el TCP, UDP, ICMP y otros protocolos de este tipo, con este control se lo debe hacer para encontrar la IP o MAC del computador con el cual se trató de ingresar o irrumpir al sistema.

El sondeo de la Red sirve como introducción para los sistemas a ser analizados, ya que su labor es recolectar los datos, obtener la información y políticas de control del sistema. Esta fase sirve para hacer un reconocimiento a toda la red detectar el IP central junto con todos los asociados a los mismos para dibujar un mapa del sistema.

La Identificación de los servicios del sistema depende del escaneo de los puerto asociados al mismo para realizar una prueba invasiva del sistema donde se verificará la recepción de los protocolos de envió de datos tunelizados, encapsulados o de enrutamiento. Así como la enumeración de los servicios de internet activos o accesibles y la seguridad programada en los cortafuegos (Firewall).

La búsqueda de información competitiva es la información generada por la compañía la cual debe tener un alto nivel de seguridad para que no pueda ser accedida por personas sin en nivel de seguridad requerido. La revisión de privacidad está centrada en los niveles de permisos de acceso de cada usuario dentro del sistema así como de la accesibilidad a los datos.

La obtención de documentos, al igual que en uno de los anteriores puntos verifica el tiempo de acceso a la información así como la validación de la misma y al igual que en el punto anterior el tiempo que dura dicho test dependerá mucho de la cantidad de información almacenada por la compañía y/o empresa.

La búsqueda y verificación de vulnerabilidades consta en la revisión de cada una de las posibles fallas o vulnerabilidades que podría tener un sistema o aplicación que este siendo testeada para la comprobación de la seguridad. Este proceso se lo realizará imitando ser un atacante o hacker que trata de irrumpir en el sistema, o se lo podrá realizar utilizando software invasivo los cuales tratan de irrumpir en el sistema utilizando fuerza bruta o SQL-injection, este tipo de software es muy útil ya que también se puede realizar una prueba de estrés en la cual se verificará a partir de cuántos proceso simultáneos el sistema empieza a presentar errores o lentitud al ejecutar sus procesos.

El testeado de las aplicaciones de internet se lo realizará para encontrar los fallos de seguridad en el acceso y/o identificación de los usuarios y enumerar los accesos a de aplicaciones cliente/servidor de internet al sistema; esto también podrá utilizarse para la auditoria del sistema ya que dicha funcionalidad es requerida en cualquier sistema grande para realizar los controles pertinentes del mismo, por lo cual este tipo de testeado serán desarrollados por administradores del sistema.

El enrutamiento es la comprobación de sistemas de routers y las defensas programadas en las mismas, donde se restringirá el acceso y flujo del tráfico en la red e internet. El testeado de sistemas confiados está basado en afectar la presencia de internet en el sistema para delimitar el test de vulnerabilidades y el test de cortafuegos.

El test de control de acceso se lo realizará tanto en los métodos de protección del computador como el Firewall, protocolos de red y otros de seguridad. Así también se tomara en cuenta las barreras que tiene el sistema en sí, estas son los controles de acceso por medio del logueo o identificación de los usuarios, es en este proceso de identificación por el cual los intrusos o hackers tratan de irrumpir en el sistema utilizando injection (SQL, PHP, SCRIPT), por fuerza bruta o colgando al servidor en el que se encuentra la aplicación.

El testeo de sistema de detección de intrusos está enfocado al rendimiento y susceptibilidad de un ID los cuales son centrados en el control de Accesos según el ID, en el que se necesitara un control de los ID. El testeo de las medidas de contingencia se realiza a los planes y/o políticas que se deben seguir cuando hay un ataque de un programa malicioso, una emergencia, error en el sistema, u otros que puedan dañar al sistema.

Descifrado de contraseñas este proceso es importante ya que valida la robustez de las contraseñas, esto se mide según el número de caracteres de la contraseña, si dicha contraseña es alfanumérica, esta también puede tener caracteres especiales para darle solides a la seguridad en el sistema, para mejorar la seguridad o robustez de la contraseña se deberá verificar que dicha contraseña tenga varios de los caracteres descritos anteriormente y que la cantidad de dichos caracteres sea mayor o igual a seis u ocho dígitos.

El testeo de denegación de servicios está dado por la comprobación de acciones ya sean fortuitas o intencionales por las cuales se imposibilite el acceso a algún tipo de servicio del sistema, mayormente esta denegación de

servicios sirve para restringir las puertas de acceso que tiene los servidores, restringir la instalación de software por parte de los usuarios finales, etc.

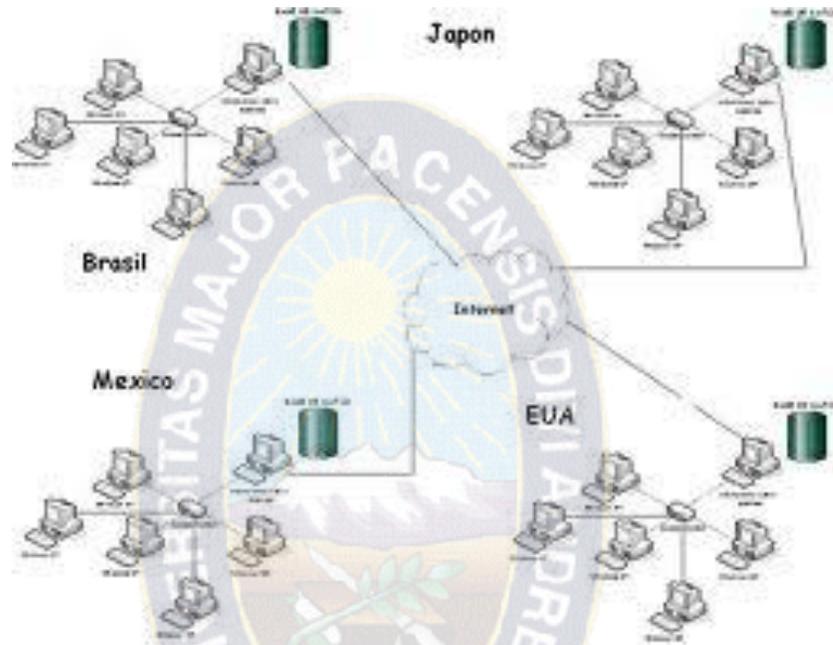
Evaluación de políticas de seguridad se basa en la verificar de cuan eficientes son las políticas de seguridad implementadas para el control del acceso y extracción de los datos almacenados en el sistema, todas estas políticas de seguridad deben estar bien especificadas, y ser de conocimiento de todos los usuarios para su uso correcto uso y un mejor control en el manejo de dichas políticas. (Herzog, OSSTMM3. Manual de la metodología abierta de testeo de seguridad, 2010)

## **2.2 BASES DE DATOS DISTRIBUIDAS**

Las bases de datos distribuidas son un grupo de bases de datos interconectados por un canal (Red local, internet comunicación con una máquina virtual, u otros), cada base de datos del sistema debe cumplir las formas normales y Boyce-codd para dar una mejor integridad y estabilidad al sistema. Cada base de datos en los nodos del sistema, debe ser un sistema sustentable por sí solo, y a su vez también deberán funcionar como si fuera una sola con las demás bases de datos del sistema, de tal manera que cualquier usuario que ingrese al sistema tenga a disposición la información de cualquier base de datos de la red emulando que la información de todo el sistema se encuentran en la base de datos local o nodo del sistema.

Otra definición de este modelo de base de datos es la definición de un tipo de base de datos virtual, en la cual la información esta almacenada en varias bases de datos reales las cuales se encuentran en distintos espacios lógicos,

las cuales tienen una conexión lógica entre ellas (red local, red wifi, internet, entre otros). (Proal, 2013)



**Figura 2. 2 Modelado de un sistema de Bases de Datos Distribuidas**

Fuente: <http://kevinmartell91.blogspot.com/2012/11/7-bases-de-datos-distribuidas.html>

Como se puede ver en la Figura 2.2 un sistema de bases de datos distribuidas tiene esta estructura con los nodos del sistema en redes locales y una conexión entre dichos nodos vía internet. (Martell, 2012)

### **2.2.1. ARQUITECTURA DE UN SISTEMA DE BASE DE DATOS DISTRIBUIDAS**

La arquitectura de una base de datos distribuidos es muy compleja y se puede dividir en dos tipos de estructuras según su orientación al ser diseñada para la integración lógica.

### 2.2.1.1. Integración lógica por medio de diseño top-down (DistDB)

Esta arquitectura para la integración lógica de arriba hacia abajo nos muestra que desde un esquema global se puede dividir para la distribución del sistema.

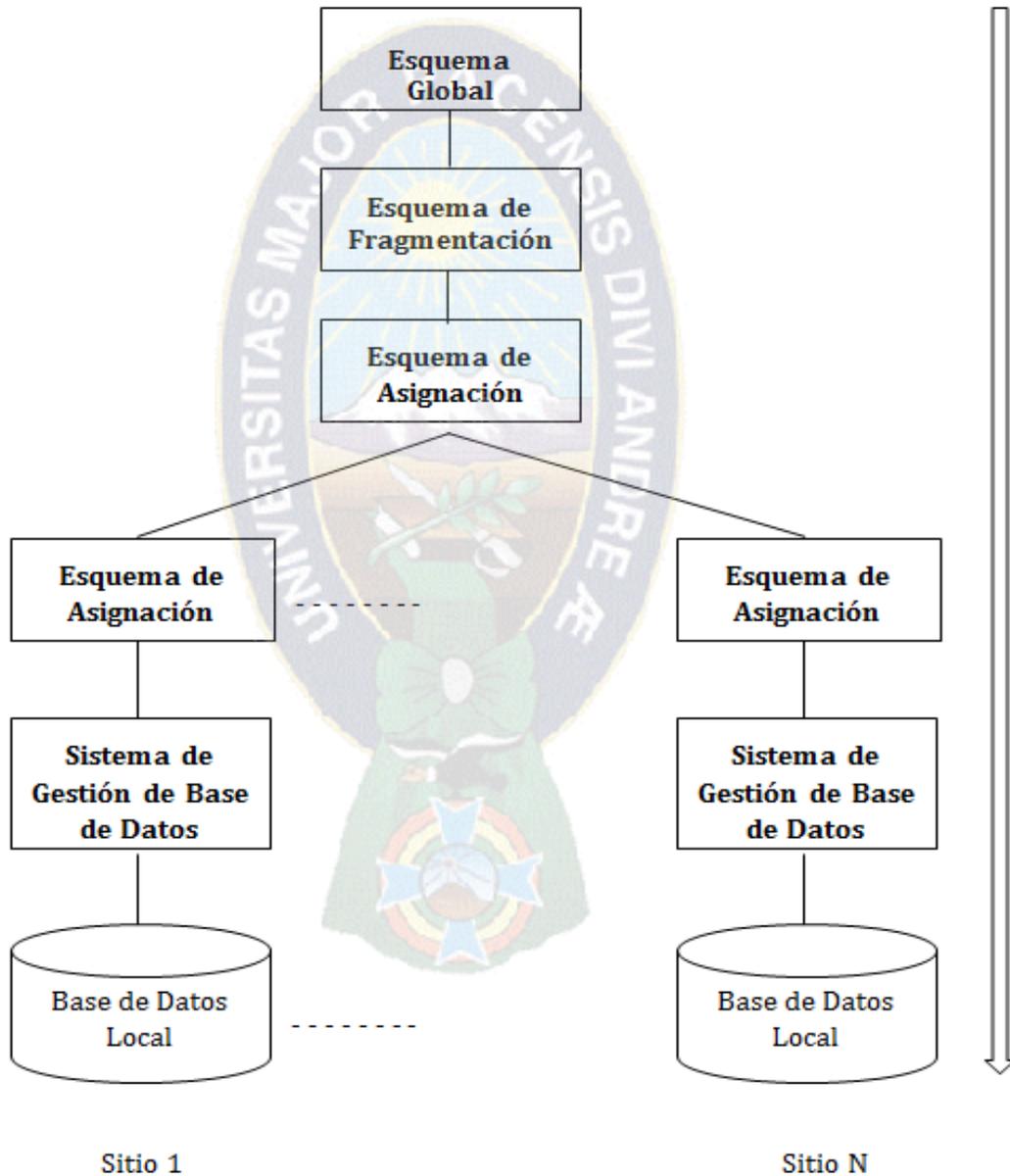


Figura 2. 3 Integración lógica top-down  
Fuente: <https://iessanvicente.com/colaboraciones/BBDDdistribuidas.pdf>

### 2.2.1.2. Integración lógica por medio de bottom-up (Multidatabase)

Esta arquitectura para la integración lógica de abajo hacia arriba ensambla el sistema desde los fragmentos o nodos del mismo, teniendo como resultado el esquema de fragmentación que indica las relaciones entre los fragmentos.

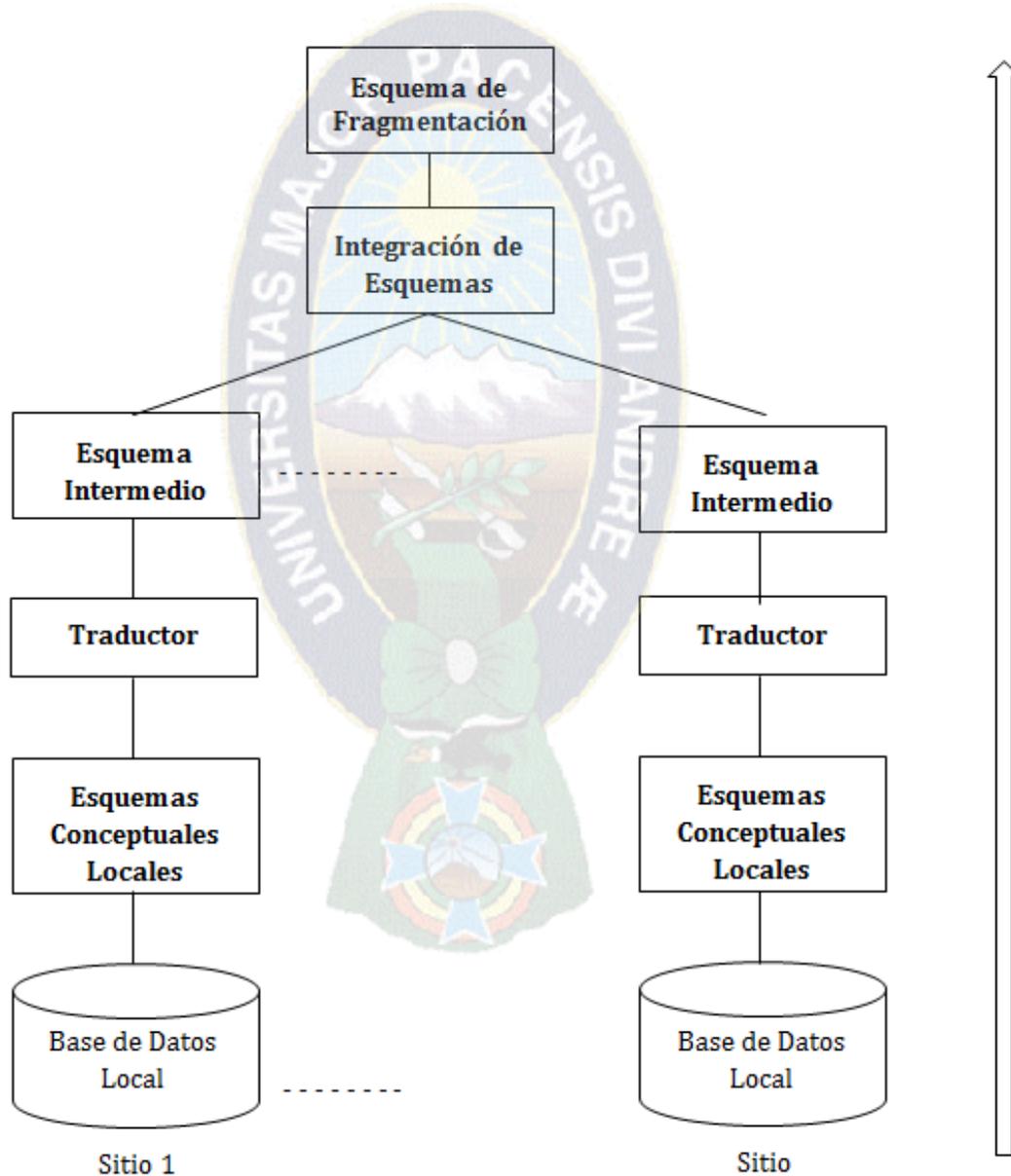


Figura 2. 4 Integración lógica bottom-up  
Fuente: <https://iessanvicente.com/colaboraciones/BDDdistribuidas.pdf>

| Arquitectura | Componentes  |
|--------------|--|
| top-down     | <p>En el <b>Esquema Global</b> se define los datos que estarán incluidos en la base de datos distribuida tal como si fuera una base de datos no distribuida.</p> <p>El <b>Esquema de Fragmentación</b> es la relación entre lo global y lo fragmentado (un esquema global está compuesta de varios fragmento del esquema pero un fragmento del esquema solo le pertenece a un esquema global).</p> <p>EL <b>Esquema de Asignación</b> define el sitio donde el o los fragmentos están localizados.</p> <p>Un <b>Esquema de Asignación Local</b> traduce los fragmentos locales a objetos, los cuales son administrados por el sistema de manejo de las bases de datos.</p> <p>El <b>Sistema de Manejo de las Bases de Datos</b> en si es el gestor de base de datos que se utilizara en cada nodo del sistema.</p> |
| bottom-up    | <p>En el <b>Esquema de Fragmentación</b> se representará las relaciones entre los nodos del sistema de bases de datos distribuidas.</p> <p>La <b>Integración de Esquemas</b> mostrará en detalle cual es la relación existente entre los elementos pertenecientes a los nodos del sistema.</p> <p>En el <b>Esquema Interno</b> se mostrará las relaciones entre las tablas y datos de cada nodo del sistema de bases de datos distribuidas.</p> <p>Los <b>Esquemas Conceptuales Locales</b> mostrarán el esquema o diagrama ER de cada nodo del sistema por separado.</p>  |

Tabla 2 Descripción de Componentes por Arquitectura  
Fuente: Vicente Toledo, 2008

## 2.2.2. TIPOS DE ALMACENAMIENTO

Existen tres tipos de almacenamiento en las bases de datos distribuidas, esta división es muy evidente ya que se puede dividir las tablas pertenecientes a las bases de datos por sus registros o tuplas, así también se puede realizar dicha división por sus atributos en es te último caso se necesitara un atributo que sirva de nexo entre las divisiones de esta tabla.

| Tipo      | Descripción  |
|-----------|--|
| Replica   | <p>Este tipo de almacenamiento conserva varias copias o replicas idénticas de una tabla en varios de los nodos pertenecientes al sistema, para ello la tabla que se replica en cada nodo debe tener los mismos atributos, así como las mismas tuplas de la tabla y dichas tablas deberán estar normalizadas.</p> <p>Por este motivo se puede llegar a tener un gran conflicto en la actualización de todas las tablas que se replicaron en el sistema ya que si en una de ellas se inserta, modifica o elimina algún dato se deberá efectuar la misma acción en las réplicas para conservar la integridad y el funcionamiento autónomo de cada nodo.</p> |
| Fragmento | <p>Este tipo de almacenamiento en sistemas distribuidos hace referencia a las divisiones que se pueden efectuar en una o varias tablas del sistema, estos tres tipos de fragmentación serán descritos a continuación:</p> <p>a) Horizontal</p>   |

|  |  |
|--|--|
|  | <p>Este tipo de fragmentación divide la tabla por sus tuplas o datos que se encuentran en la o las tablas del sistema, con la condición que en dichas tablas no exista tuplas o datos repetidos para prever o evitar duplicidad de los datos almacenados en el sistema, lo cual será beneficioso para reducir el tiempo de respuesta e incrementar la funcionalidad de los sistemas con bases de datos distribuidas.</p> <p>b) Vertical</p> <p>En esta fragmentación la tabla será dividida por los atributos los cuales tendrán una unión entre ellos mayormente se trata de la clave primaria o el identificador de la tupla, esto servirá para relacionar los datos de la o las tablas fragmentadas del sistema.</p> <p>c) Mixto</p> <p>Es una combinación entre los dos tipos anteriores de fragmentación en el cual la tabla no solo será dividida por sus atributos sino también por sus tuplas.</p> |
|--|--|

|                     |   |
|---------------------|---|
| Replica y Fragmento | Este tipo de almacenamiento es una conjunción o combinación entre las dos divisiones anteriormente mencionadas, se tendrá una o varias tablas del sistema que acepten la replicación de sus datos y a su vez puedan estar divididas por sus atributos utilizando un atributo de conexión entre ellas, el cual será uno de los datos que serán replicados en las tablas. |
|---------------------|---|

**Tabla 3 Tipos de almacenamiento**  
**Fuente: Vicente Toledo, 2008**

### **2.2.3. NIVELES DE TRANSPARENCIA**

El nivel de transparencia hace referencia a la separación de la semántica de alto nivel de un sistema de los aspectos de bajo nivel relacionados, si el tratamiento o separación de estos niveles es adecuado, esto ayudara a ocultar detalles de implementación a las capas de alto nivel o a otros usuarios.

La independencia de datos en cada una de las bases de datos del sistema se puede dar al controlar dos aspectos para los niveles de transparencia, dichos niveles serán descritos a continuación:

#### **2.2.3.1. Independencia Lógica**

Hace referencia a que si existiera algún tipo de cambio en la estructura lógica de la base de datos, dichos cambios no deberían afectar a las aplicaciones que deben ser ejecutadas o usadas por los usuarios finales.

### **2.2.3.2. Independencia Física**

Esto hace referencia a ocultar los detalles sobre las estructuras de almacenamiento a las aplicaciones de usuario, por lo cual se afirma que si la estructura de almacenamiento es modificada, esto no afecta a las aplicaciones de usuario.

### **2.2.4. PROCESAMIENTO DISTRIBUIDO DE CONSULTAS**

Este proceso es muy monótono en bases de datos centralizadas; por otro lado, es muy importante en bases de datos distribuidas ya que el objetivo es convertir transacciones de usuario en instrucciones para manipular datos; por esta razón, la transacción afectará la velocidad de respuesta del sistema.

### **2.2.5. RECUPERACIÓN**

Para recuperar la información de este tipo de sistemas con bases de datos distribuidas es frecuente encontrar los siguientes problemas en dichos procesos, los cuales serán descritos a continuación:

#### **2.2.5.1. Fallo de los Nodos**

Si un nodo falla, el sistema deberá continuar trabajando con los nodos que aún funcionan; si el nodo que falló es una base de datos local, se deberá separar los datos entre los miembros restantes del sistema, esta operación en general no es implementada debido a la complejidad que representa dividir la información que un nodo almacena en los otros nodos del sistema.

### **2.2.5.2. Copias Múltiples de Fragmentos de Datos**

Es necesario un subsistema o encargado de control de la replicación para mantener la consistencia de la base de datos distribuida, así como la actualización de los datos registrado en cada uno de los nodos ya que cabe la posibilidad de que en la replicación de dichos datos se produzcan errores o se agregue ruido en la transacción lo cual dañaría la integridad de los datos.

### **2.2.5.3. Transacción Distribuida Correcta**

Se pueden producir fallos durante las transacciones, incluso si estas son correctas si se plantea el caso de que alguno de los nodos que fallaron intervino en la transacción, se deberá revisar si la transacción se la realizó correctamente. Para tratar de prevenir este tipo de fallos se puede utilizar las funciones propias de las redes, las cuales tiene algunos instrumentos especializados para la comprobación de una correcta transacción de por la red.

## **2.3 INTELIGENCIA ARTIFICIAL Y AGENTES INTELIGENTES**

### **2.3.1. INTELIGENCIA ARTIFICIAL**

La inteligencia artificial es una rama de la informática, la cual fue inicialmente mencionada en la década de los 40, pero no tuvo gran impacto debido a que en ese tiempo no se le dio gran importancia; por otra parte en la era de 1950 el matemático británico Alan Turing realizó un trabajo de investigación en esta área, en la cual planteo la pregunta “¿Puede pensar una maquina?” y propuso un juego para comprobar la inteligencia de una máquina, donde dos personas, una de ellas responderá las preguntas determinadas en el juego junto con la

máquina que se desea probar, la otra persona estará en un cuarto separado y definirá cuales de las respuestas son de la máquina y cuales son hechas por la persona, si la persona que analiza las respuestas elige las respuestas de la maquina como si fueran las de un ser humano se dirá que esta máquina es inteligente, a este juego se lo llamo “Maquina de Turing”.

Este enfoque del juego de la máquina de Turing ve a la inteligencia artificial como una imitación del comportamiento humano, dicho enfoque no fue muy convincente a lo largo del tiempo ya que ahora la inteligencia se la denomina o categoriza según el comportamiento racional, a su vez la inteligencia artificial I.A. Se apoya en dos ramas de las ciencias exactas, una de ellas es la lógica matemática y la otra es la creciente ciencia de la computación con sus grandes avances en la electrónica. (Choque, 2013, págs. 2, IAp01\_InteligenciaArtificial.pdf)

El propósito principal de esta ciencia es el estudio de la replicación o automatización de la inteligencia humana, para poder diseñar un sistema que pueda discernir la información, compararla con la información anterior o almacenada, aprender y tomar sus propias decisiones racionales con la intervención mínima o nula de un ser humano.

Esta ciencia se ha beneficiado mucho del crecimiento del hardware ya que los procesos que se desarrollan para la imitación de la inteligencia humana tienden a ser complejos y a necesitar una gran velocidad de procesamiento.

### **2.3.1.1. DEFINICIONES DE LA INTELIGENCIA ARTIFICIAL**

Existen muchas definiciones para la inteligencia artificial debido a que la inteligencia como tal es un concepto difícil de expresar y entender, por lo cual entre todas las definiciones existentes o dadas para definir esta ciencia las de mayor aceptación son las siguientes:

| <b>Autor</b>  | <b>Definición</b>   |
|---------------|---|
| Elaine Rich.  | “Estudio de cómo hacer que las computadoras hagan cosas que por el momento las personas las realizan de mejor manera” |
| Marvin Minsky | “Ciencia de cómo hacer cosas que requerirían inteligencia si fueran hechas por seres humanos”                         |

**Tabla 4 Definiciones Aceptadas de la I.A.**

**Fuente: Choque, 2013, págs. 5, IAp04\_AgentesInteligentes.pdf**

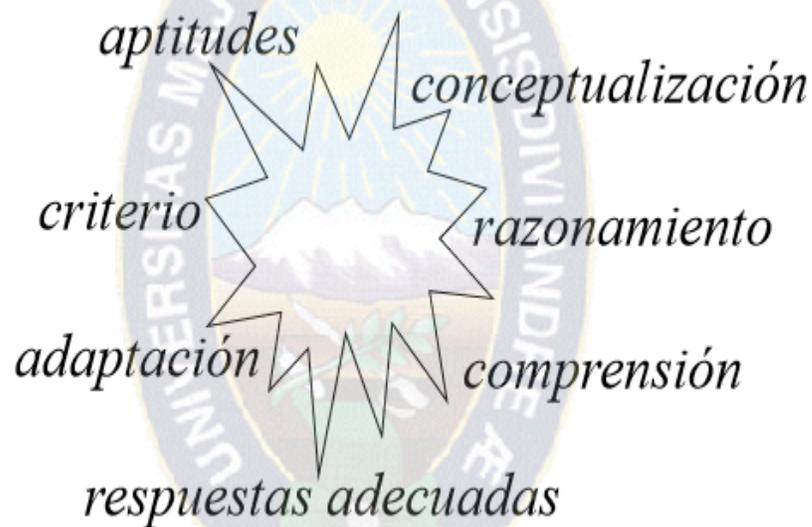
### **2.3.1.2. INTELIGENCIA**

La inteligencia es un concepto muy difícil de expresar ya que engloba distintos tipos de acciones, es por esto que es un proceso propio del ser humano y es muy distintivo del mismo ya que es gracias a esto que el ser humano puede procesar a gran velocidad los datos recibidos, compararlos o combinarlos entre sí y gracias a todo esto convertir dichos datos en información.

Es por esto que el ser humano es capaz de discernir entre respuestas correctas e incorrectas, o definir lo bueno de lo malo utilizando la información que es recibida o percibida por la multiplicidad de sensores que este posee y también

el conocimiento que ha almacenado a lo largo de sus experiencias pasadas y/o información recopilada a lo largo de su vida. (Choque, 2013, págs. 2-3, IAp01\_InteligenciaArtificial.pdf)

Esta característica principal del ser humano es polifacética y engloba los siguientes campos definidos en la figura 2.5 a continuación.



**Figura 2. 5 Naturaleza polifacética**  
**Fuente: Choque, 2013, págs. 2, IAp01\_InteligenciaArtificial.pdf**

Por otro lado la jerarquía en la pirámide del conocimiento brinda los niveles en los que es representado el conocimiento o dividido según el nivel de procesamiento al cual son sometidos los datos.

El nivel de procesamiento puede cambiar según la importancia que se le dará a cada dato en el instante en el que debe ser procesado, ya que este nivel mientras más alto sea es más productivo en el desarrollo del proceso y se lo representa en una pirámide como se lo muestra en la figura 2.6.



Figura 2. 6 Pirámide del conocimiento  
Fuente: Choque, 2013, págs. 2, IAp01\_InteligenciaArtificial.pdf

### **2.3.1.3. FUNDAMENTOS DE LA INTELIGENCIA ARTIFICIAL**

Al igual que en muchas ramas de las ciencias, la Inteligencia Artificial necesita ser comprobada o solventada por otras ciencias para su comprobación o verificación y ser reconocida como una ciencia verificable y aceptada.

Por este principio fundamental la Inteligencia Artificial se apoya en algunas ramas de las ciencias para que reconocida una ciencia, dichas ramas deben ser y estar reconocidas a su vez como ciencias probadas o aun con la verificación no se podrá reconocer como una ciencia.

Algunas de las ciencias que sirven de apoyo o solventa a la Inteligencia Artificial para su comprobación serán descritas a continuación, esta descripción mostrará los fundamentos de cada ciencia que apoya a la comprobación de la inteligencia Artificial como ciencia probada.

| Fundamento                                      | Características  |
|---|--|
| Fundamento Filosófico                           | <ul style="list-style-type: none"> <li>• Alan Turing usa la computación con la máquina de Turing (1936).</li> <li>• Aristóteles 430 AC y George Boole en 1847 usan la lógica.</li> <li>• Gerolamo Cardano en 1530 uso la probabilidad.</li> <li>• John Von Neumann en 1944 uso la teoría de las decisiones.</li> </ul> |
| Fundamento Matemático                           | <ul style="list-style-type: none"> <li>• Alan Turing usa la computación con la máquina de Turing (1936).</li> <li>• Aristóteles 430 AC y George Boole en 1847 usan la lógica.</li> <li>• Gerolamo Cardano en 1530 uso la probabilidad.</li> <li>• John Von Neumann en 1944 uso la teoría de las decisiones.</li> </ul> |
| Fundamento Psicológico con el Proceso Cognitivo | <ul style="list-style-type: none"> <li>• Memoria.</li> <li>• Atención.</li> <li>• Conocimiento.</li> <li>• Percepción.</li> <li>• Razonamiento.</li> <li>• Asociación.</li> <li>• Interpretación.</li> </ul>   |

|  |   |
|--|---|
| Fundamento de ingeniería computacional ramas | <ul style="list-style-type: none"> <li>• Ramas</li> <li>• Maquinas Inteligentes.</li> <li>• Herramientas de Simulación.</li> <li>• Actividad racional SE.</li> <li>• Modelos computacionales RNA.</li> <li>• Sistemas Expertos</li> <li>• Resultados.</li> <li>• Gestión del conocimiento.</li> </ul> |
| Lingüística                                  | <ul style="list-style-type: none"> <li>• Aprendizaje del lenguaje.</li> <li>• Lingüística ocupacional o procesamiento del lenguaje natural.</li> <li>• Teoría conductista no abordada</li> </ul>  |

**Tabla 5 Descripción de los parámetros de las ciencias que apoyan a la I.A.**  
**Fuente: Miqueleiz, 2003**

### **2.3.2. AGENTES INTELIGENTES**

Los agentes inteligentes son cualquier tipo de programa, robot o humano que tenga sensores para percibir el mundo exterior y también tenga efectores para responder a dichos estímulos.

Los agentes inteligentes son parte de la Inteligencia Artificial y a su vez una rama de la informática que generó gran expectativa cuando fue presentada ya que en ese tiempo veían a la Inteligencia Artificial como la manera de darle a la maquina un pensamiento racional, teniendo en cuenta que la inteligencia es un rasgo característico de los seres humanos. Lo que se intenta en la Inteligencia Artificial es modelar dicha inteligencia, ya que lo que se busca al crear este tipo de programas inteligentes es dar a los usuarios un ayudante que se pueda

amoldar a las necesidades del usuario, por lo que estos programas pueden ser de gran ayuda para cualquier usuario.

Estos agentes inteligentes en definición son agentes que siempre realizan o toman la decisión correcta para que esto sea posible es necesario que el agente posea una base de conocimiento la cual pueda ser incrementada automáticamente o manualmente por algún usuario. (Ferré, 2010)

### **2.3.2.1. TIPOS DE AGENTES SEGÚN SU CATEGORÍA**

Existen muchos tipos de agentes inteligentes los cuales pueden ser agrupados en las siguientes categorías:

| <b>Categorización</b>   | <b>Descripción</b>   |
|-------------------------|--|
| Agentes de Colaboración | Estos agentes interactúan entre si y se colaboran para llegar a una meta mayor.  |
| Agentes de Interfaz     | Este tipo de agentes son como asistentes personales que colaboran con el usuario fina, observan e interpretan las acciones del usuario y aprenden de dichas acciones para sugerir mejoras.                     |
| Agentes Móviles         | Estos agentes se desplazan por el sistema una o más veces según la necesidad del proceso que requiere el usuario, el agente pasara por los distintos sitios extrayendo la información que el usuario requiere. |

|                                   |   |
|-----------------------------------|---|
| Agentes de Información o Internet | Este agente de información o internet se moverá por internet aprovechando los enlaces que existen en los portales web, con el fin de verificar o buscar alguna información en concreto. |
| Agentes Reactivos                 | Estos agentes son los que reaccionan a estímulos, por cada estímulo el agente tiene relacionada una acción adecuada y racional.   |
| Agentes Híbridos                  | Estos agentes nacen por la necesidad de maximizar la habilidad de los otros tipos de agentes y reducir las debilidades que estos otros agentes presentan.                               |
| Agentes Inteligentes              | Este tipo de agentes son denominados así por la habilidad de tomar decisiones racionales y aprender ya sea dependiendo de un usuario o autónomamente.                                   |

**Tabla 6 Tipos de agente según su categoría**  
**Fuente: Montero, 2007**

### **2.3.2.2. ATRIBUTOS DE LOS AGENTES INTELIGENTES**

Algunos de los atributos serán descritos a continuación:

| Atributos | Descripción   |
|-----------|---|
| Autonomía | <ul style="list-style-type: none"> <li data-bbox="532 296 870 327">• <i>Totalmente Autónomo</i></li> </ul> <p data-bbox="597 401 1433 804">La autonomía del agente radica en que el agente pueda aprender por si solo sin intervención de ningún usuario, esto se puede lograr haciendo que el agente valla aprendiendo de cada decisión que vea tomar a un usuario, para que posteriormente viendo las decisiones del usuario vaya aprendiendo y tomando sus propias decisiones en casos de conflicto, esto iguala a la forma de aprendizaje en la que un ser humano incrementa su conocimiento con el paso del tiempo.</p> <ul style="list-style-type: none"> <li data-bbox="532 877 789 909">• <i>Semi Autónomo</i></li> </ul> <p data-bbox="597 982 1433 1386">A diferencia del otro tipo de agentes este deberá pedir la intervención de algún usuario del sistema en caso de que no pueda encontrar una respuesta racional a algún problema encontrado en el sistema, por lo que el agente preguntara al usuario sobre la mejor decisión para la resolución del problema, así también este agente deberá aprender sobre la decisión que el usuario tomo y posteriormente poder tomar la decisión basándose en dichas respuestas.</p> <ul style="list-style-type: none"> <li data-bbox="532 1459 748 1491">• <i>Dependiente</i></li> </ul> <p data-bbox="597 1564 1433 1749">Este agente no tiene ningún tipo de inteligencia y es totalmente dependiente de los usuarios para la toma de cualquier decisión, siempre se tiene una comunicación directa con el usuario administrador debido a que el agente inteligente no puede tomar</p> |

|                                  |  |
|----------------------------------|--|
|                                  | una decisión directa sin consultarla al usuario administrador.   |
| Movilidad                        | Es la capacidad del agente de poder migrar de un ambiente en caso de ser necesario, este atributo es mayormente usado si el agente tiene acceso a internet o a distintos ambientes del sistema.  |
| Inteligencia                     | Capacidad del agente de reconocer los eventos de su entorno, determinar su trascendencia o significado y tomar una decisión o llevar a cabo una acción para influir en este evento.  |
| BDI (belief, desire, intentions) | Creencia, deseo e intención del agente, la creencia es toda la información que el agente tiene de su entorno, el deseo es el objetivo o meta del agente y la intención es el método o decisión que el agente aplicara según el estímulo.                                       |
| Reactividad                      | Este atributo se muestra en la capacidad del agente de responder a los estímulos de su entorno, teniendo en cuenta que este tipo de agentes hace una relación entre los estímulos y las acciones, las cuales están relacionadas en la base del conocimiento.                   |
| Pro actividad                    | La pro actividad de un agente se muestra cuando dicho agente puede intervenir su entorno sin haber tenido antes un estímulo de este, esto ocurre cuando el agente ve que el entorno no va de acuerdo a la meta que persigue o para la cual fue diseñado el agente inteligente. |

|              |   |
|--------------|---|
| Sociabilidad | El agente es sociable al interactuar con otro agente o ser humano, esto asegurara que el agente trabaje correctamente o racionalmente en la búsqueda del objetivo cuando interactúe con otro agente o ser humano. |
|--------------|---|

**Tabla 7 Atributos de los agentes inteligentes.**

Fuente: Miqueleiz, 2003

### **2.3.2.3. ESTRUCTURA DE AGENTES**

La arquitectura general de un agente inteligente está definida por la siguiente formula:

$$\text{Agente} = \text{Programa Agente} + \text{Arquitectura}$$

#### a) Programa Agente

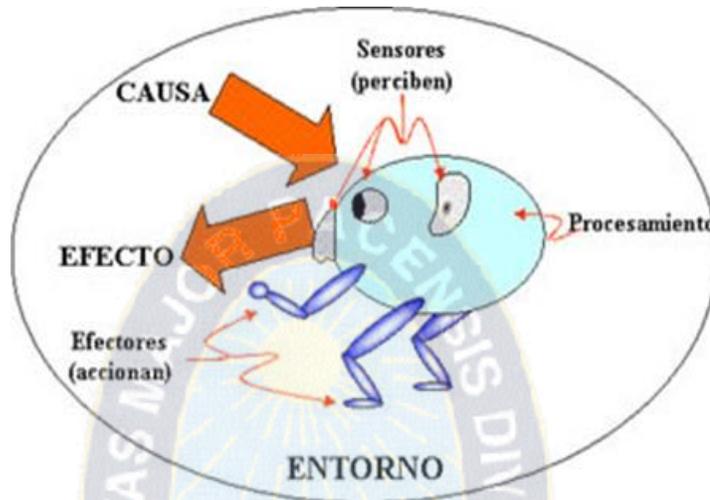
El programa agente es el que se encargara del procesamiento de todas las entradas o estímulos provenientes del entorno, es decir es el que elige o relaciona el estímulo con la respuesta racional posible para el problema que se presenta en el entorno.

#### b) Arquitectura de un Agente

La arquitectura básica de un agente no es tan compleja como se podría llegar a pensar, ya que este tipo de programas tiene pocas partes distintivas.

En la Figura 2.7 se muestra gráficamente la arquitectura básica de agente inteligente, especificando las partes básicas de este agente como los sensores,

efectores y procesamiento de información.



**Figura 2. 7 Arquitectura básica de un agente**  
 Fuente: [wikipedia.org/wiki/Agente\\_inteligente](http://wikipedia.org/wiki/Agente_inteligente)

Como se ve en la imagen los agentes tiene tres componentes principales los cuales se describirán a continuación:

| Componente | Descripción  |
|------------|--|
| Sensores   | <p>Cualquier tipo de objeto o cosa que sirva para la recolección de datos del entorno los cuales serán procesados por el agente para la elección de la acción racional que se debe seguir.</p> <p>Cada elemento que pueda percibir datos del entorno es tomado como un sensor, teniendo en cuenta que los agentes procesaran los datos que sean recolectados por los sensores es necesario discernir entre los datos recolectados.</p> |

|               |  |
|---------------|--|
| Efectores     | Los efectores son objetos o cosas que sirven para intervenir en el ambiente, eso hace que cualquier parte del agente que intervenga con el ambiente para modificarlo, cambiarlo o alterarlo de cualquier forma se puede considerar un efector.             |
| Procesamiento | Este tipo de elemento puede tener un proceso de caja negra o un proceso claro; este proceso cambia los datos recolectados por los sensores y los convierte en salidas o tomas de decisiones racionales para ser procesadas o ejecutadas por los efectores. |

**Tabla 8 Descripción de componentes.**  
**Fuente: Ferré, 2010**

#### **2.3.2.4. CLASIFICACIÓN DE AGENTES POR SUS CARACTERÍSTICAS**

Partiendo de la arquitectura básica de los agente inteligentes y aumentando algunas características especiales o modificando algunas de las características principales de los agentes inteligentes, estas variaciones pueden diferenciarlos y/o definirlos en varios conjuntos a los agentes de la siguiente manera:

a) Agente basado en una tabla

Este tipo de agentes inteligente tiene una tabla que almacena la información que relaciona los estímulos con las acciones que el agente realizará en su entorno para lograr cambiarlo de manera que influya positivamente a lograr el objetivo o meta que dicho agente sigue.

En esta tabla se almacenará las relaciones entre los estímulos, las acciones y la meta principal junto con las sub metas, todos estos datos serán procesados y

con ellos se escogerá una acción acorde a la necesidad.

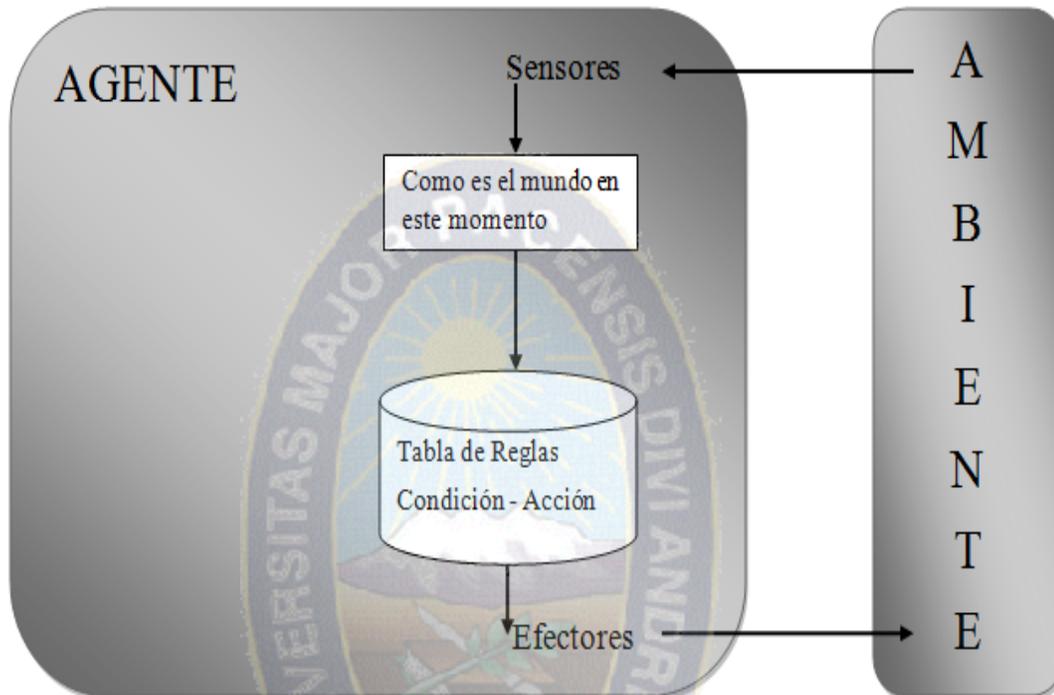


Figura 2. 8 Agente basado en tabla

Fuente: <http://www.robot.uji.es/docencia/II28/teoria/transparencias-tema02.pdf> año: 2010

#### b) Agente de Reflejo Simple

Este agente inteligente simula o se asemeja a un camino ya que dicho agente solo conecta a los estímulos con las acciones que este ejecutara, por lo cual no procesa las entradas para convertirlas en salidas, solo asocia las entradas con las salidas predeterminadas que se tienen registradas.

En conclusión este agente solo toma o elige la acción correspondiente al estímulo dado por el ambiente según las reglas previas que tenga, dichas relaciones o reglas están definidas previamente o serán almacenadas en la base de conocimiento del agente por un usuario administrador del sistema o

alguna persona que tenga conocimiento de programación y/o de la estructura y codificación del agente.

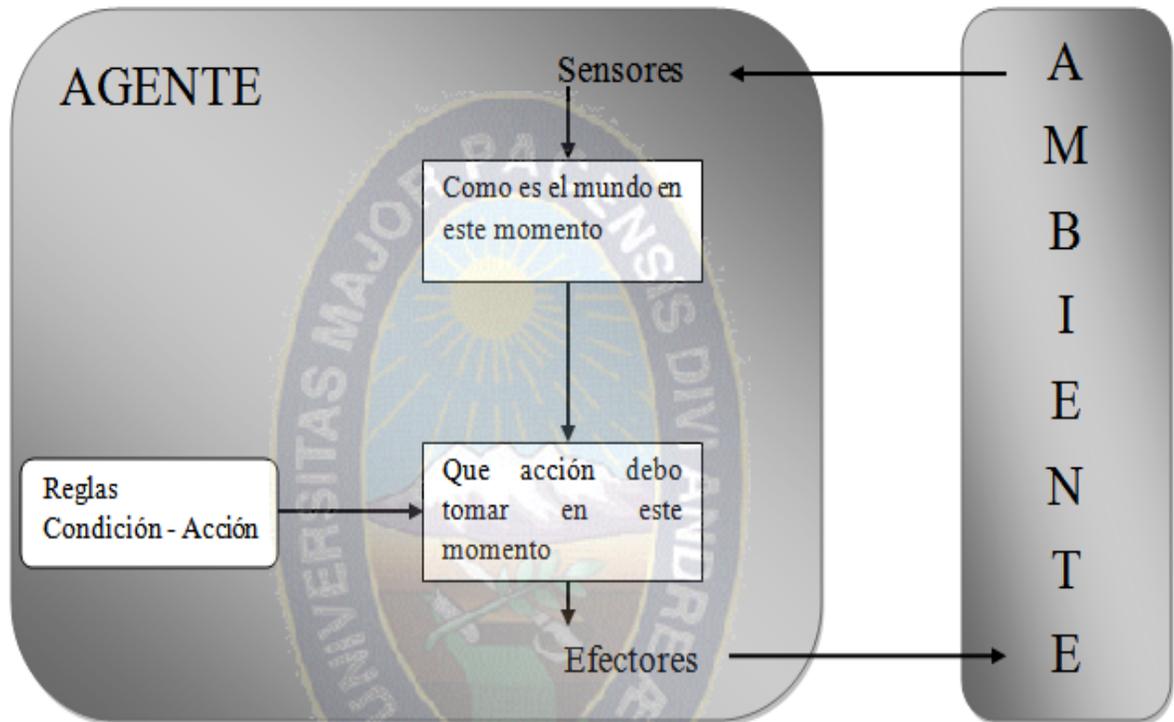


Figura 2. 9 Modelo de agente reactivo

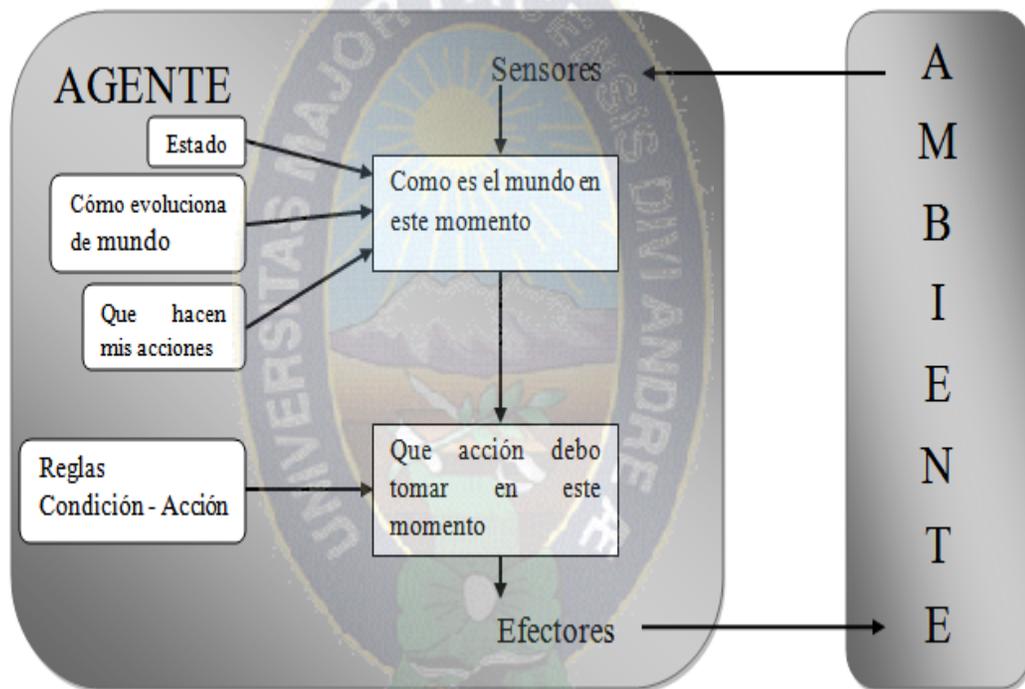
Fuente: <http://www.robot.uji.es/docencia/II28/teoria/transparencias-tema02.pdf> año: 2010

### c) Agente de Reflejo con Estado Interno

Estos agentes inteligentes son similares a los agentes de reflejo simple que se explicaron anteriormente en esta investigación, la diferencia fundamental de este tipo de agentes es que dicho agente tiene una bandera o estado interno del agente, ya que dicho estado influenciará en la toma de decisiones del agente.

Este estado interno cambiara según el ambiente y la meta que persiga el agente inteligente, este estado o bandera servirá para indicar al agente la

decisión racional que deberá tomar para cumplir con la meta esto es útil cuando no es posible tomar una decisión a partir de una sola percepción, esto puede ser porque los sensores no proporcionan la información suficiente del ambiente, teniendo esto en cuenta el estado interno del agente completara la información necesaria para la toma de decisiones racionales.



**Figura 2. 10 Modelo de agente reactivo con estado interno**

Fuente: <http://www.robot.uji.es/docencia/II28/teoria/transparencias-tema02.pdf> año: 2010

d) Agente Basado en Objetivos

Este agente inteligente basa sus decisiones en el objetivo que el sistema persigue, se centra directamente en el objetivo u objetivos para los que fue diseñado, por lo que la elección de las salidas se determinara según la conveniencia del objetivo que se persigue.

Incluso si este agente inteligente cuenta con el estado interno que proporciona mayor información del ambiente, esto puede no ser suficiente si se tiene dos o más opciones para seleccionar que sean igualmente racionales.

Por lo que si llegara a darse el caso de tener dos o más opciones que sean igualmente correctas el agente deberá tener en cuenta el objetivo o meta para el que fue diseñado e implementado en el sistema y de esta forma escoger la acción racional más acertada con relación al objetivo perseguido por el sistema.

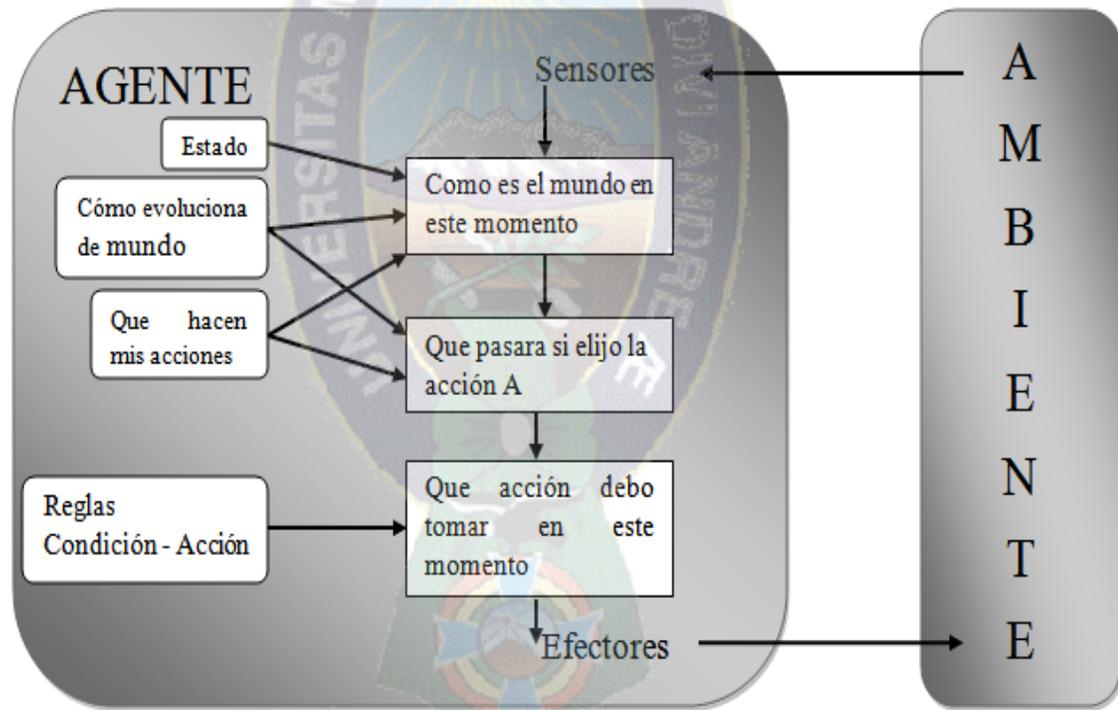


Figura 2. 11 Modelo de agente basado en objetivos

Fuente: <http://www.robot.uji.es/docencia/II28/teoria/transparencias-tema02.pdf> año: 2010

e) Agente Basado en Utilidades

Este agente inteligente posee las características anteriormente mencionadas, el estado interno que brinda más información del ambiente, también toma en



## 2.4 METODOLOGÍA PROMETHEUS

Es una metodología de desarrollo para agentes inteligentes, la cual fue creada por “Lin Padgham y Michael Winikoff”, la cual se basa en definir un lenguaje de modelado, dicho lenguaje a su vez forma parte de los fundamentos de AUML.

En la actualidad la metodología prometheus se encuentra en desarrollo conjunto con Agent Oriented Software, con el cual sacaron la herramienta para el modelado de agentes inteligentes PDT o Prometheus Desing Tool. (Lin Padgham, 2009)

Entre las características principales de Prometheus podemos mencionar:

- Brinda soporte al desarrollo de agentes inteligentes que contengan creencias, objetivos, planes y eventos.
- Permite representar una estructura jerárquica para la representación de múltiples niveles de abstracción.
- Se tiene acceso a varios artefactos de diseño que permiten representar el esquema de los agentes.
- Utiliza procesos iterativos para el desarrollo de la ingeniería de software.
- Metodología ampliamente usada tanto en lo académico como en lo profesional.

### 2.4.1. ETAPAS BÁSICAS

Existen tres etapas básicas para el desarrollo correcto de esta metodología y la buena utilización de las herramientas que nos facilita la creación de los

diagramas y partes del código de sistemas inteligente, estas etapas serán desglosadas a continuación. (Montero, 2007)

#### **2.4.1.1. Especificación del Sistema**

En esta etapa se encuentra el detalle superficial de cómo el agente trabajara con el ambiente teniendo en cuenta, así como la funcionalidad que dicho agente deberá cumplir y para sustentar el desarrollo se elaboraran los casos de uso correspondientes a la ingeniería de software.

#### **2.4.1.2. Arquitectura del Sistema**

En esta etapa se detallara el funcionamiento del sistema y como el o los agentes se acoplaran a dicho funcionamiento, se definirá las acciones, los eventos en los que el agente apoyara al sistema así como la información que se compartirá entre los agentes y el sistema y en esta etapa también se podrá definir el tipo de agentes que se necesitara desarrollar para cumplir con todo lo anteriormente especificada.

Esta etapa tiene gran relevancia en el desarrollo de la metodología de desarrollo de sistemas inteligentes, ya que teniendo en cuenta que la arquitectura definirá el tipo de agente que será desarrollado.

#### **2.4.1.3. Diseño Detallado del Sistema**

En esta etapa se hará un detalle o descripción más explícita sobre los agentes, la capacidad de dichos agentes, los eventos que estarán presentes en el sistema y el plan a seguir para la interacción del agente y el sistema.

En la figura 2.13 se muestran las tres etapas descritas anteriormente, dichas etapas de la metodología estarán enlazadas según el funcionamiento del sistema que se está desarrollando, aun así los elementos representados como los enlaces entre dichos elementos que se muestran en la figura son los elementos principales u obligatorios para el desarrollo de esta metodología para la creación de sistemas inteligentes.

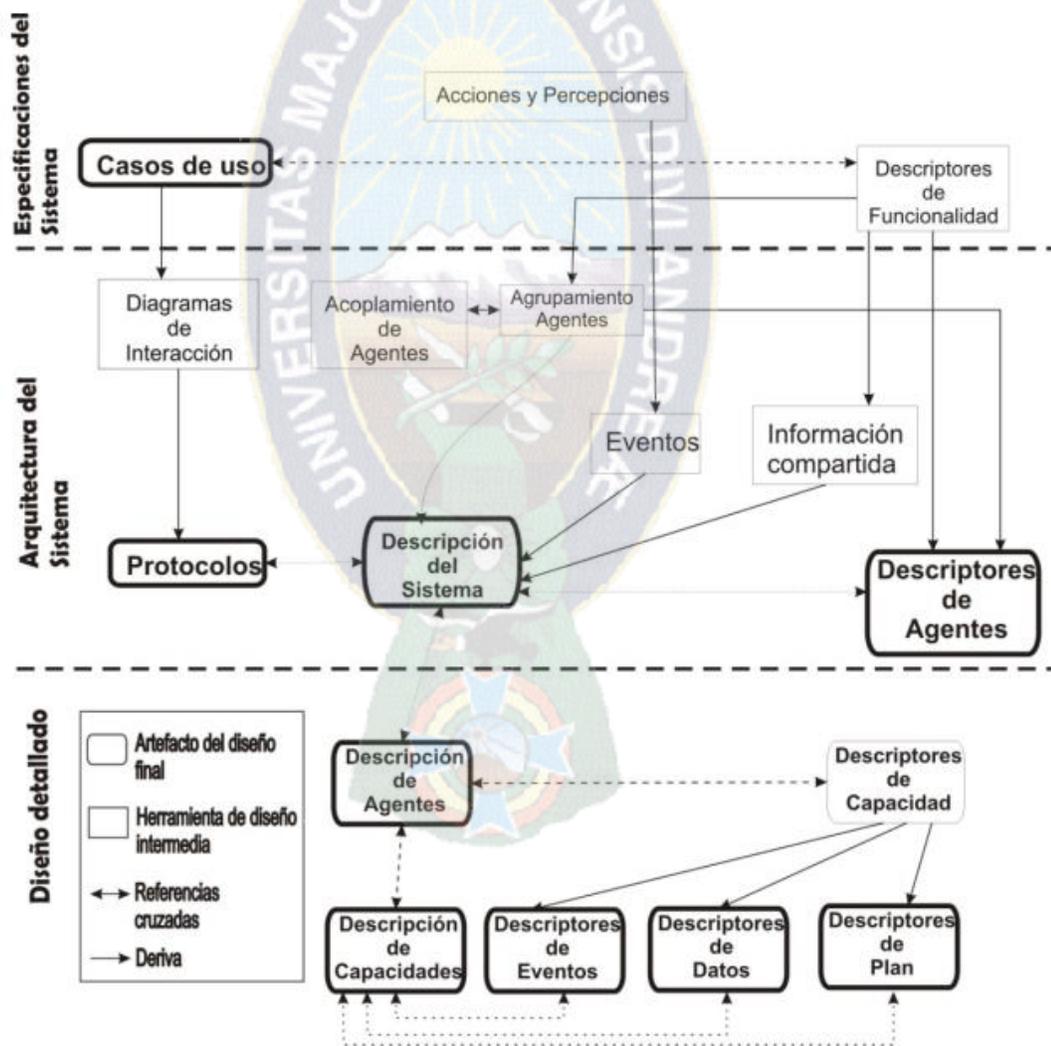


Figura 2. 13 Metodología de diseño Prometheus  
Fuente: Padgham 2002

Para el desarrollo de la etapa de especificación y arquitectura del sistema es necesitaria conocer los siguientes conceptos utilizados en la metodología:

| <b>Conceptos</b> | <b>Descripción</b>  |
|------------------|---|
| Percepción       | Toda información provista por el entorno por medio de los sensores.   |
| Evento           | Es la información dada por la percepción pero con mayor relevancia.   |
| Acción           | Interacción del agente con su entorno.  |
| Actores          | Persona o rol que interactúe con el sistema, estas entidades pueden ser humanos, entidades del sistema u otros sistemas.  |
| Escenarios       | Son una secuencia de pasos que describen una operación en el sistema.   |
| Protocolo        | Funciones o interacciones bien definidas.   |
| BDI              | Este término hace referencia a las creencias (conocimiento que tiene el agente de su entorno), deseos (Objetivos) e intenciones (representa lo que el agente decide llevar a cabo). |

**Tabla 9 Conceptos utilizados en la metodología**  
**Fuente: Lin Padgham, 2009**

Una vez identificados y definidos todos los conceptos utilizados en el diseño de la metodología se deberá definir en detalle las etapas que sigue la metodología prometheus.

| Etapas                           | Detalle  |
|----------------------------------|--|
| Especificación del sistema       | <ul style="list-style-type: none"> <li>• Identificación de actores y la relación con el sistema.</li> <li>• Desarrollo de escenarios.</li> <li>• Definición de Metas.</li> <li>• Agrupación de metas y otros elementos para formar roles y objetivos básicos.</li> </ul> |
| Diseño de la Arquitectura        | <ul style="list-style-type: none"> <li>• Tipos de Agentes.</li> <li>• Protocolos de integración.</li> <li>• Diagrama general del sistema</li> </ul>  |
| Diseño detallado                 | <ul style="list-style-type: none"> <li>• Definición detallada de cada Agente</li> <li>• Refinación de Agentes (opcional)</li> <li>• Definición de relaciones o diagramas de procesos</li> <li>• Diagrama general de capacidad</li> </ul>                                 |
| Desarrollo del sistema de prueba | <ul style="list-style-type: none"> <li>• Mvs</li> <li>•</li> </ul>   |

**Tabla 10 Etapas de la metodología Prometheus**  
**Fuente: Hernández, 2011**

## 2.5 MODELO VISTA CONTROLADOR

La programación en M.V.C. o Modelo Vista Controlador es muy popular entre los programadores ya que brinda una forma de separar capas diferenciadas el desarrollo de cualquier sistema, esta diferenciación hace que los programas que utilicen el M.V.C. sean mucho más fáciles de comprender o entender y dar soporte a estos sistemas ya que en caso de alguna modificación o error el sistema el programador podrá encontrar fácilmente el error o la parte que se necesita modificar, es por este motivo que este tipo de desarrollo o modelo de programación es muy utilizado para el desarrollo de sistemas informáticos.

A continuación describiremos los elementos de este modelo.

| Elemento    | Descripción   |
|-------------|---|
| Modelo      | En esta etapa se programara todo lo relacionado con las bases de datos, las relacionadas existentes entre las tablas (1-1, 1-n, n-n ), claves primarias, etc.   |
| Vista       | Esta es la etapa donde se programara la parte visual de software, estos archivos son los que interactuaran directamente con los usuarios finales.   |
| Controlador | Es la lógica del programa, la cual interactúa con las dos etapas anteriores ya que extrae los datos de las bases de datos por medio del modelo y la muestra en la vista o viceversa desde la vista recaba los datos a ser registrados en la base de datos por medio del modelo. |

**Tabla 11 Elementos del M.V.C.**  
**Fuente: Villamarín, 2013**

Si se siguen las reglas de este tipo de programación se conseguirá un mejor tiempo de reacción por parte del programa, también será mucho más fácil el mantenimiento del software ya que en caso de cualquier error el programador sabrá exactamente dónde encontrar la causa del error.



## CAPITULO III

### MARCO APLICATIVO

#### 3.1 INTRODUCCIÓN

Este capítulo servirá para explicar todo el desarrollo necesario para la comprobación o refutación de la hipótesis planteada en esta investigación, a su vez dicha especificación y desarrollo se apoyara en el “Marco Teórico” (Capitulo II) en el cual se expusieron todos los puntos necesarios para el desarrollo de la investigación.

Para detallar el desarrollo del agente inteligente, el cual se encargara de la seguridad de las bases de datos distribuidas se necesitara alguna metodología de desarrollo cumpliendo con las normas de la ingeniería de software, por lo cual la metodología que se utilizara o se escogió será la metodología para desarrollo de agentes inteligentes llamada “Prometheus”, dicha metodología tiene desarrollada una herramienta propia la cual se llama “Prometheus Design Tool (PDT)” la cual nos ayudara en todo el proceso que se sigue en esta metodología, ya que nos servirá como guía para el desarrollo de las siguientes fases: (Lin Padgham, 2009)

- Especificación del Sistema.
- Diseño Arquitectónico.
- Diseño Tallado.

Esta metodología fue elegida para el desarrollo del agente debido a que es una metodología creada especialmente para el desarrollo de sistemas inteligentes, y teniendo en cuenta que será necesario el desarrollo de un agente inteligente se adaptara la metodología PROMETHEUS, debido a que dicha metodología tiene una herramienta (Prometheus desing tools) muy útil para el desarrollo de cualquier sistema inteligente, es por esto que se escogió esta metodología de desarrollo la cual se adaptada para el desarrollo del agente inteligente que se diseñara y desarrollara en esta investigación.

### 3.1.1. PRINCIPALES AMENAZAS DE SEGURIDAD

Después del análisis de las metodologías de testeo que serán necesarias para la comprobación del incremento de la seguridad en sistemas de bases de datos distribuidas, se llegó a concluir cuales son las principales amenazas para este tipo de sistemas, las cuales se especificaran a continuación:

Las principales amenazas en la seguridad de la información, la seguridad en los procesos y la seguridad en las tecnologías de internet son:

| Principales Amenazas | Descripción  |
|----------------------|--|
| SQL-injection.       | Esta amenaza se debe a que existen personas inescrupulosas que tratan de irrumpir en los sistemas por medio del ingreso de códigos SQL para saltar las barreras del sistema. |

|                               |   |
|-------------------------------|---|
| Fuerza bruta.                 | Al igual que en la anterior amenaza, se tratara de averiguar la contraseña de algún usuario registrado por medio de sucesivos intentos hasta dar con la combinación correcta. |
| Errores de conexión           | Los errores de conexión se deben a las caídas que pueden sufrir las redes que conectan los nodos de estos sistemas  |
| Caídas del sistema            | Esta amenaza depende tanto del mantenimiento que se debe realizar a cada nodo del sistema.  |
| Replicación de la información | Esta amenaza se produce cuando los nodos comparten la información registrada en ellos, con lo cual se puede producir una duplicidad en los datos almacenados.                 |

**Tabla 12 Descripción de las amenazas**  
**Fuente: Miqueleiz, 2003 y Montero, 2007**

## **3.2 ESPECIFICACIÓN DEL SISTEMA**

### **3.2.1. IDENTIFICACIÓN DE ACTORES**

Para el desarrollo de la investigación y el desarrollo del agente inteligente es necesario la identificación de los posibles usuarios o actores que interactuaran con el sistema inteligente y la especificación de los mismos estos actores deben ser posibles de identificar en cualquier sistema que quiere utilizar al agente. Al agente inteligente que se desarrollará se lo dividió en las dos funciones

principales que realizara, la primera será la interacción con los agentes y la segunda será la interacción entre los agentes, la identificación de los actores del sistema será descrita a continuación:

| Actor               | Características   |
|---------------------|---|
| Actor usuario       | <ul style="list-style-type: none"> <li>• No se comunicara directamente con el agente, solo tendrá comunicación directa con el sistema.</li> <li>• Este actor será el que alimente al sistema con información.</li> <li>• En caso de algún tipo de intento de irrupción, SQL-injection o cualquier acción que parezca peligroso para el sistema, el agente mostrara un aviso que indique esta falta.</li> </ul>  |
| Actor administrador | <ul style="list-style-type: none"> <li>• Este actor es el que se encargara de lidiar con los errores que existan en el sistema.</li> <li>• Verificara en caso de irrupciones en el sistema las direcciones IP o MAC registradas por el agente.</li> <li>• Verificara el correcto funcionamiento de los agentes en los nodos.</li> <li>• Tomará las decisiones para la resolución de los conflictos, con lo cual incrementara la inteligencia del agente.</li> </ul> |
| Agente nodo         | <ul style="list-style-type: none"> <li>• Este agente interactuara como intermediario entre el sistema y la base de datos ya que es en este punto donde realizara el control.</li> <li>• Controlará las conexiones a las bases de datos distribuidas.</li> </ul>   |

|                      |  |
|----------------------|--|
|                      | <ul style="list-style-type: none"> <li>• Verificara la integridad de los datos, controlando el uso de caracteres especiales, palabras reservadas usadas para SQL-injection y otros datos que afecten la integridad del sistema.</li> </ul>   |
| Agente administrador | <ul style="list-style-type: none"> <li>• Verificara el buen funcionamiento del sistema, previniendo que cada nodo posea la información suficiente para que pueda trabajar autónomamente al sistema.</li> <li>• Verificación de conflictos entre agentes nodo, esto puede ocurrir si uno de los agentes nodo se encuentra clonando o replicando la información de una tabla necesaria para cada nodo.</li> <li>• Verificar duplicidad de la información generada en el sistema viendo si las tablas de réplica en cada nodo no tiene duplicidad en su información.</li> </ul> |

**Tabla 13** Tabla de actores con sus respectivas características.  
Fuente: Propia.

### 3.2.2. ESCENARIOS

En este proceso de la metodología de desarrollo para agentes inteligentes se detallaran los escenarios que participaran en el sistema, por lo cual se mostrara cada escenario del sistema, con una nota que detallara cada escenario del sistema en los cuales el usuario final ingresara los datos al sistema.

En la Figura 3.2 se muestra los esenarios encontrados en el sistema, con una brebe explicacion de los misos en el rectangulo superior, dicha imagen fue generada con el software proporcionado por la metodologia de desarrollo PROMETHEUS.

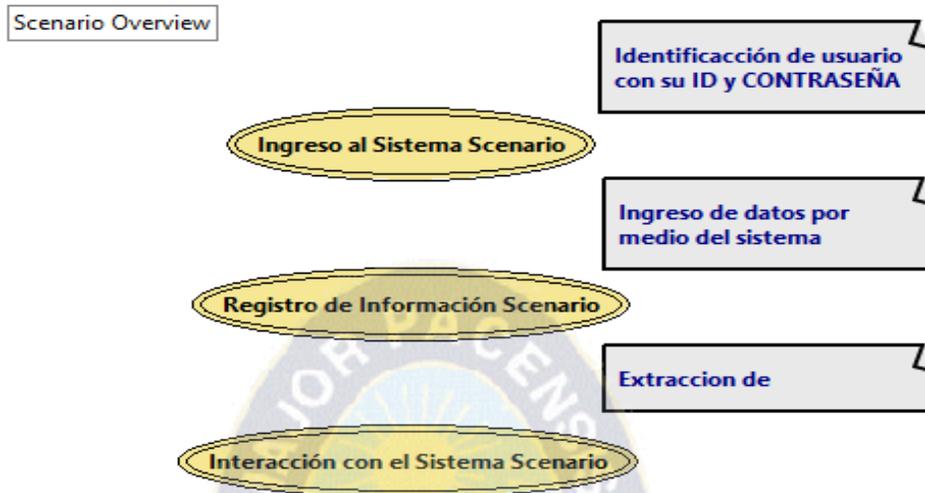


Figura 3. 1 Diagrama de descripción general de escenarios  
Fuente: Propia

### 3.2.3. DEFINICIÓN DE METAS

En este proceso de la metodología de desarrollo de agentes inteligentes se definirá la meta para la cual se desarrollara el agente, dicha meta del sistema serán coadyuvadas por las submetas o escenarios relacionados al sistema en desarrollo, con estos escenarios o submetas se activaran los agentes lo cual ayudara al incremento en la seguridad de las bases de datos o nodos del sistema.

Estas metas y submetas que perseguirán los agentes inteligentes para poder comprobar o refutar la hipótesis planteada en esta investigación serán descritas a continuación según su relevancia o jerarquía en el sistema:

| Tipos de Meta | Descripción |
|---------------|-------------|
|---------------|-------------|

|                |  |
|----------------|--|
| Meta principal | Incrementar la seguridad en los sistemas con bases de datos distribuidos.  |
| Sub-Metas      | <ul style="list-style-type: none"> <li>• Control en las conexiones a cada nodo perteneciente a la base de datos distribuida.</li> <li>• Control de irrupciones al sistema.</li> <li>• Verificación de caracteres especiales.</li> <li>• Verificación de palabras reservadas utilizadas para el SQL-injection.</li> <li>• Control en la identificación de usuarios.</li> <li>• Registro de IP o MAC en el acceso al sistema.</li> </ul> |

**Tabla 14 Listado de metas u objetivos del agente**  
Fuente: Propia

En la Figura 3.3 se muestra la descripción general de los objetivos o metas del sistema.



**Figura 3. 2 Diagrama de descripción general de objetivos o metas del sistema**  
Fuente: Propia

### 3.3 ARQUITECTURA DEL SISTEMA

#### 3.3.1. DETERMINACIÓN DE TIPOS DE AGENTES

Para la determinación de los tipos de agentes se tendrá que verificar los dos roles principales en cada rol se agrupara los agentes a los cuales se podrán asociar los roles identificados en el sistema.

Para ello cada rol debe especificarse según el tipo de acción que realizará; esto es necesario para determinar cuántos tipos de agentes son necesarios en el sistema a desarrollar.

Agrupación por roles de agente para la definición del tipo de agente inteligente que se necesitara desarrollar.

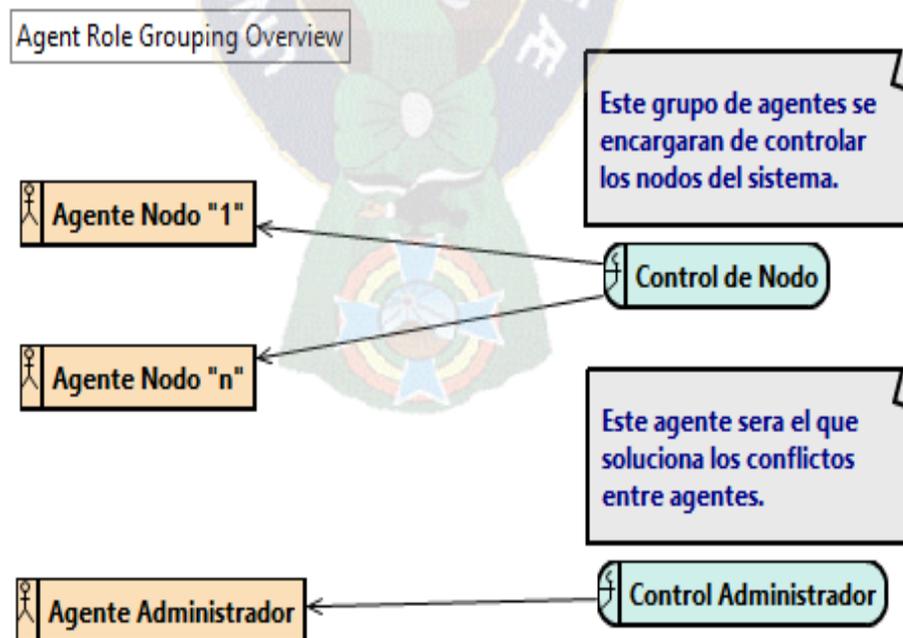


Figura 3. 3 Diagrama general de agrupación por rol  
Fuente: Propia

Posteriormente de la anterior agrupación se necesitara realizar otro tipo de agrupación la cual se basa en una agrupación por los datos o como en este caso por las bases de datos que se involucran en el sistema.

Una vez analizando los tipos de datos o las bases de datos se podrán agrupar estos en los roles a los cuales pertenecen los agentes y a su vez en esta etapa del proceso de la metodología de desarrollo se definirá al rol según el tipo de dato o base de datos.

Información general de acoplamiento de datos según el rol previamente establecido por cada agente inteligente para la asociación a un nodo o conjunto de datos que el supervisara.

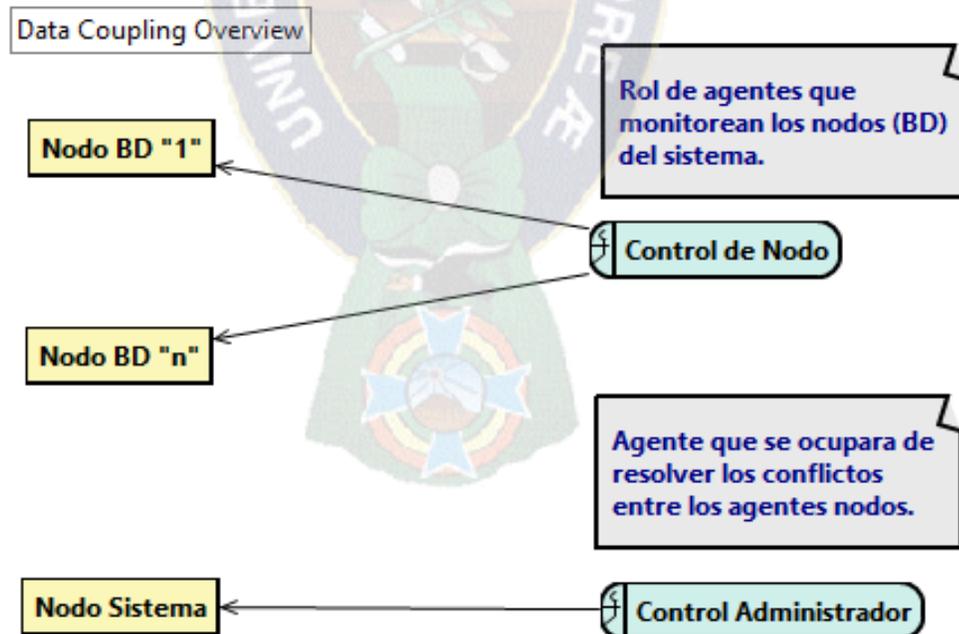


Figura 3. 4 Diagrama general de acoplamiento con los datos  
Fuente: Propia

### 3.3.2. ANÁLISIS DESCRIPTIVO GENERAL

En esta etapa de la metodología se realizara un análisis general de cómo será nuestro sistema y la interacción del mismo con los agentes necesarios para el incremento en la seguridad en la base de datos distribuidas, para la realización de este análisis se hará una descripción minucioso de las iteraciones y relaciones entre los componentes del sistema.

Para las relaciones existentes en el diagrama generado en este punto o paso de la metodología Prometheus, se tiene que describir cada elemento y las relaciones entre los mismos, la relación entre los usuarios y los escenarios del sistema por los cuales se tendrán acceso a las bases de datos distribuidas. Desde dichos escenarios los usuarios accederán al sistema y/o a los datos que el mismo almacena en las bases de datos distribuidas, lo cual generara estímulos que el agente utilizara para relacionarse con el sistema y realizar las acciones acordes a los estímulos que reciba.

Una vez procesados dichos estímulos y tomando las acciones correspondientes a cada uno de ellos, el agente se comunicara con el sistema (o base de datos nodo del sistema) mediante los protocolos establecidos para realizar la acción correspondiente al estímulo y comunicarse con las bases de datos distribuidas.

La Figura 3.5 es la grafica que representa las relaciones existentes en todo el sistema.

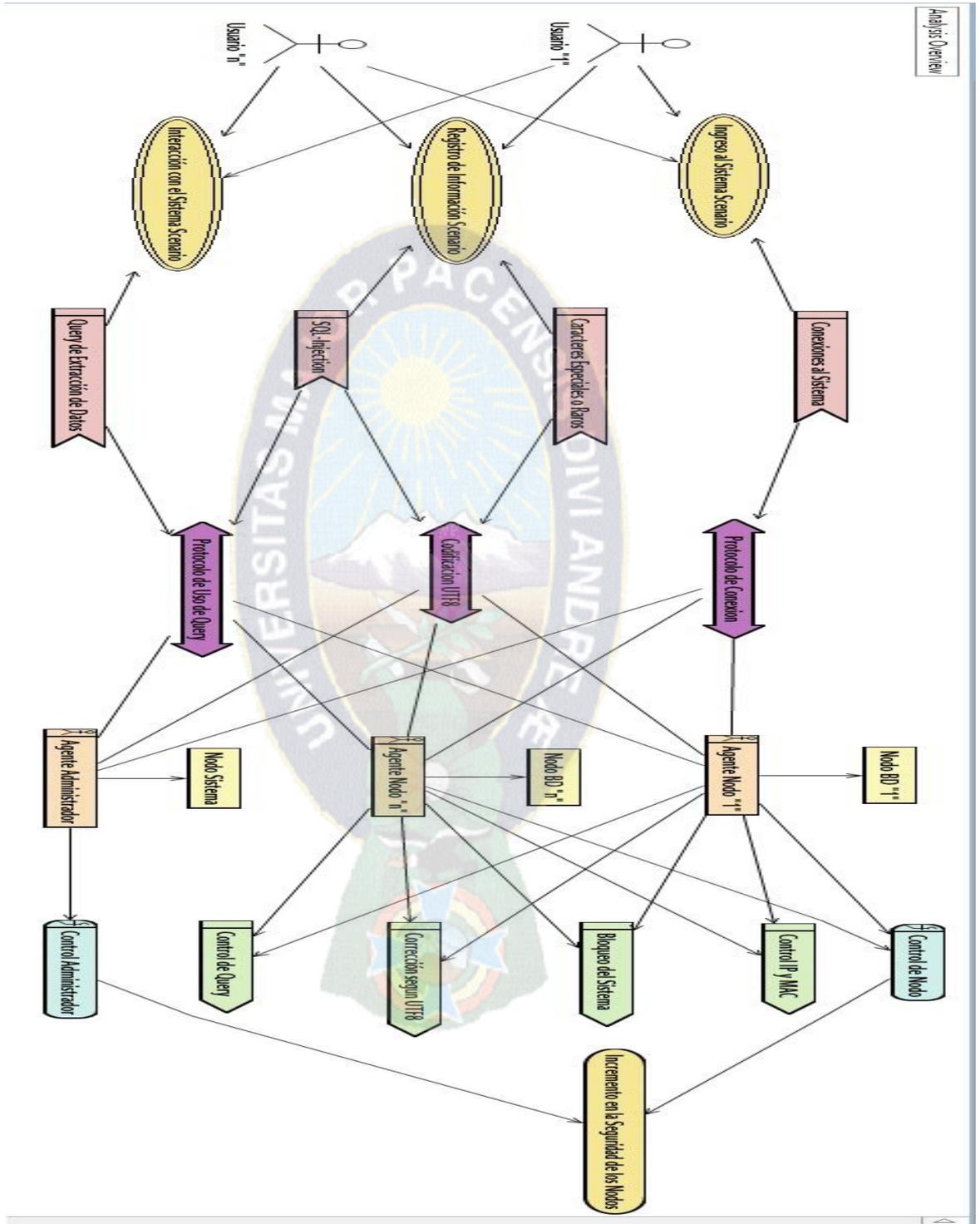


Figura 3. 5 Diagrama de análisis descriptivo general  
Fuente: Propia

### 3.4 DISEÑO DETALLADO DEL SISTEMA

#### 3.4.1. DEFINICIÓN DETALLADA DE LAS ACTIVIDADES DEL AGENTE INTELIGENTE

En esta etapa de la metodología de desarrollo de agentes inteligentes se realizara una descripción detallada de las percepciones y acciones vinculadas a cada agente inteligente para poder cumplir con los objetivos o metas para los que es diseñado.

Por lo cual se asocia a cada agente con las percepciones que este recibirá y también con las acciones que dicho agente podrá ejecutar en premura de especificar a qué percepción le corresponde cada acción.

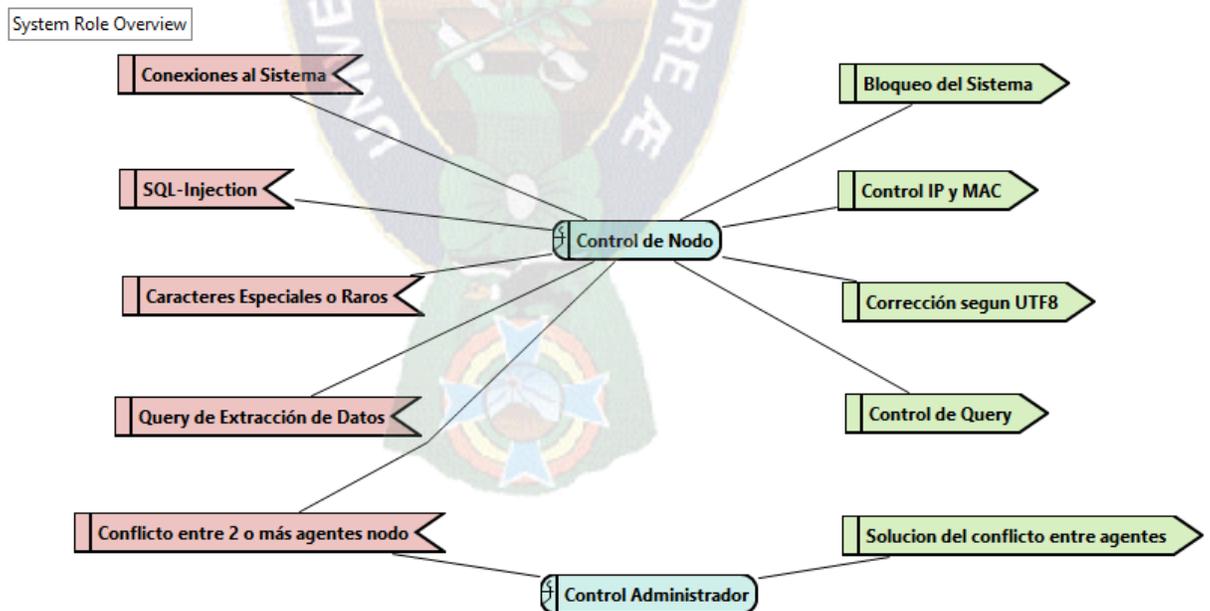


Figura 3. 6 Diagrama de especificación de actividades según rol  
Fuente: Propia

Otros puntos a ser especificados en el desarrollo de esta etapa es:

- Determinación de relaciones entre percepción y acción, las cuales serán almacenadas en la base del conocimiento del agente inteligente que se selecció conveniente para su desarrollo.
  - Revisión de las relaciones establecidas entre percepciones y acciones que afectaran al agente.
  - Depuración de estímulos
  - Depuración de acciones
  - Revisión de la integración
- Definir los detalles internos de cada agente y especificar cómo los agentes realizarán sus tareas, teniendo en cuenta el entorno en el cual se desarrollaran dichas tareas.
  - Definición de estado interno
  - Verificación de formas de conexión
  - Depuración de estímulos
- Cada agente es refinado en términos de sus:
  - Capacidades
  - Eventos internos
  - Planes
  - Estructuras de datos
  - Reacción a estímulos
  - Jerarquía de estímulos

### **3.4.2. DETALLE DE LAS RELACIONES ENTRE AGENTES, DATOS Y PROTOCOLOS**

Esta etapa servirá para ver las relaciones existentes entre los agentes y los protocolos que utilizaran para conectarse con las bases de datos o nodos del sistema, teniendo en cuenta los estímulos por los cuales el agente deberá realizar todas esta acción.

Los protocolos son las especificaciones de la forma de en qué el agente inteligente llevara a cabo las acciones necesarias para el cumplimiento de su objetivo y buen funcionamiento del sistema, en otras palabras es el detalle de como el agente realizara las acciones predeterminas en su base de conocimiento y también de las nuevas acciones que el agente aprenda con el paso del tiempo.

El diagrama especificara las conexiones entre las percepciones, los agentes, las acciones, las bases de datos o nodos del sistema y los protocolos necesarios para el correcto funcionamiento del sistema.

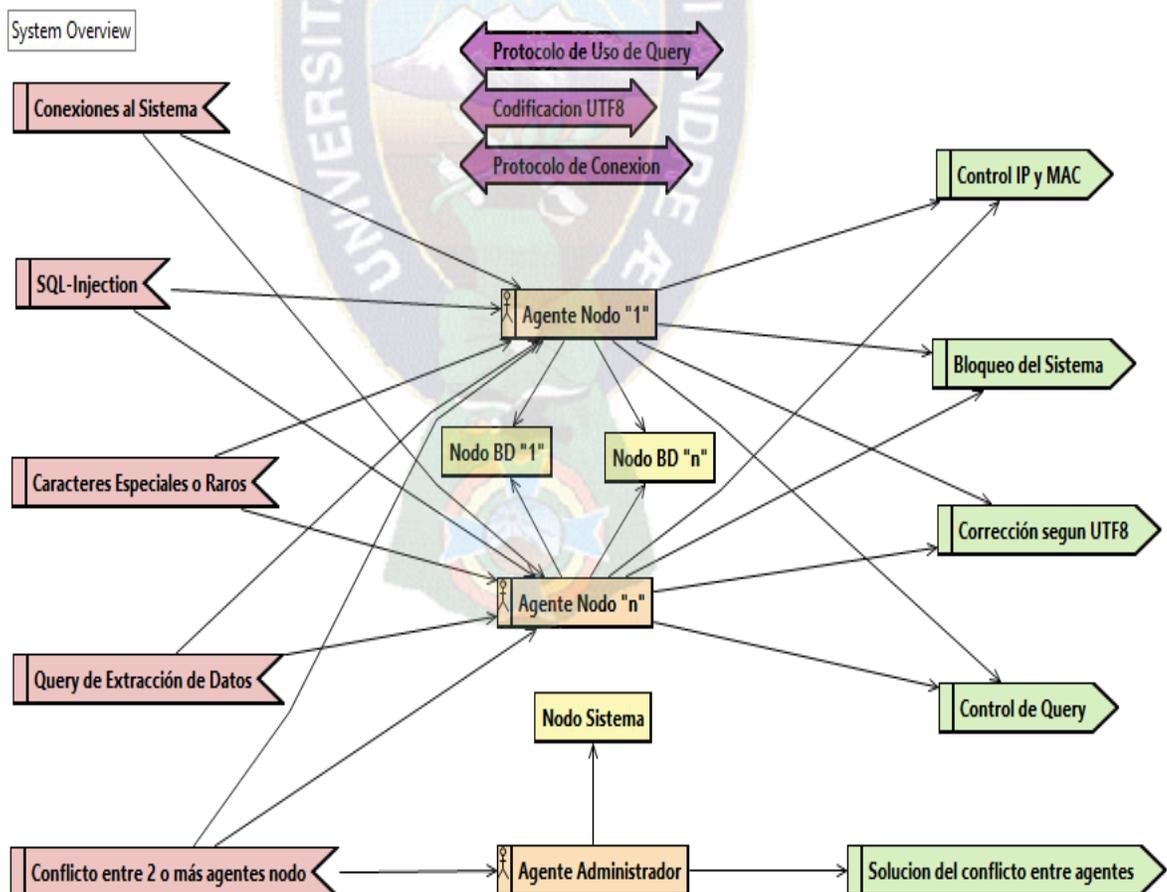


Figura 3. 7 Diagrama de relaciones del sistema

Fuente: Propia

- El diagrama general de capacidad, captura la estructura de los planes para la capacidad y los eventos asociados con estos planes para verificar el correcto o coherente funcionamiento de los protocolos asociados a los estímulos y las acciones que afectan o ejecuta el agente.
- El comportamiento dinámico se describe mediante los diagramas de procesos basados en los protocolos de interacción.

### 3.5 DESARROLLO DEL SISTEMA DE PRUEBA

#### 3.5.1. MODELO VISTA CONTROLADOR

Este modelo trata de dividir las tres principales acciones de un sistema informático, como se muestra en la figura:

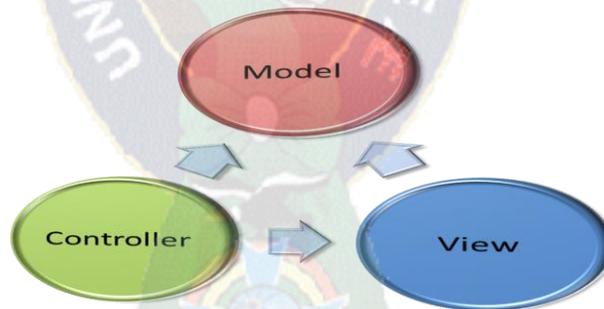


Figura 3. 8 Modelo vista controlador  
Fuente: fernando-gaitan.com.ar

**Vista.-** Esta parte del sistema se desarrolló con los lenguajes HTML 5, JQUERY y AJAX. Estos lenguajes serán usados para crear las interfaces que estarán en contacto directo con los usuarios finales, es por esto que dichas interfaces o vistas fueron desarrolladas de una manera amigable para el usuario.

**Controlador.-** Esta parte del sistema la cual se desarrolló en el lenguaje PHP, se encargara de dar a dicho sistema toda la funcionalidad necesaria del mismo y la conexión y extracción de información de las bases de datos la cual será provista por el modelo de datos.

**Modelo.-** El modelo se encargara de determinar las reglas o relaciones que existen entre los datos almacenados en los nodos (bases de datos) del sistema, esta capa es muy importante ya que es la que nos da las reglas de la extracción de datos o inserción de datos a los nodos del sistema.

El sistema desarrollado tiene esta arquitectura y adicionalmente tiene al agente que se encargara de hacer el nexo entre los nodos, el modelo y su respectivo controlador, la adición del agente propuesta en esta investigación se la realizara para que el agente controle las conexiones a los nodos y el flujo de datos de los nodos al sistema.

### **3.5.2. BASE DE DATOS DISTRIBUIDA**

Para el desarrollo de la base de datos distribuida se utilizara la arquitectura **top-down** la cual consiste en que partiendo de un esquema de base de datos centralizada será dividirá o fragmentara la misma para crear los nodos del sistema, esta fragmentación se la realizara de tal forma que cada nodo del sistema pueda trabajar independientemente de los otros.

Esta independencia de los nodos se debe a que un sistema de este tipo debe poder funcionar en su red local sin la interferencia de los demás nodos, la comunicación entre los nodos solo debe darse cuando uno de ellos necesita

información almacenada en otro y para la replicación de los datos que se encuentran en una o varias tablas, las cuales serán repartidas entre los nodos para que estos sean independientes, como se muestra en la figura 3.9.

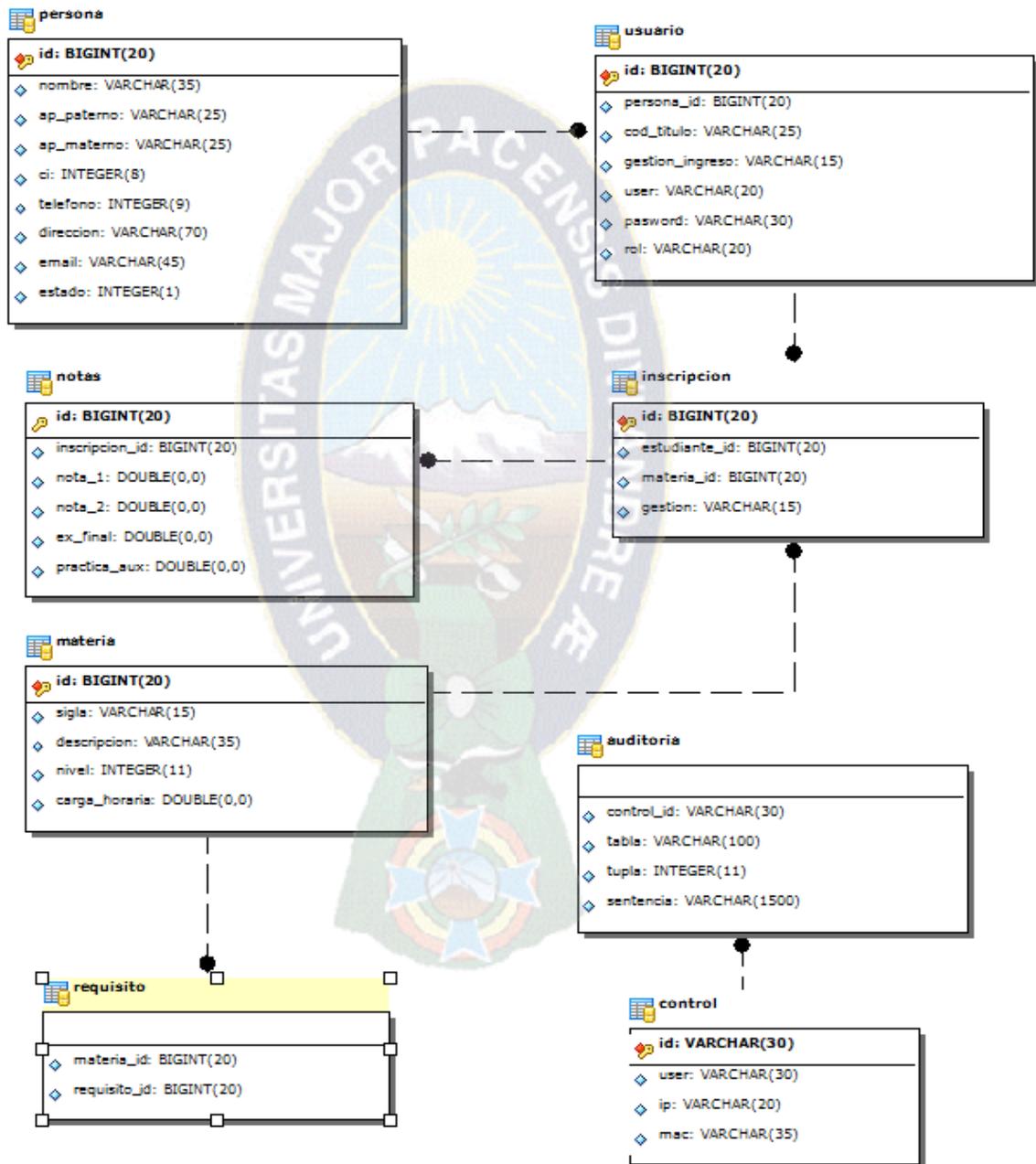


Figura 3. 9 Diagrama de Nodo 1 para estudiante  
Fuente: Propia

En la Figura 3.10 se muestra la estructura del nodo perteneciente a la parte administrativa y docente.

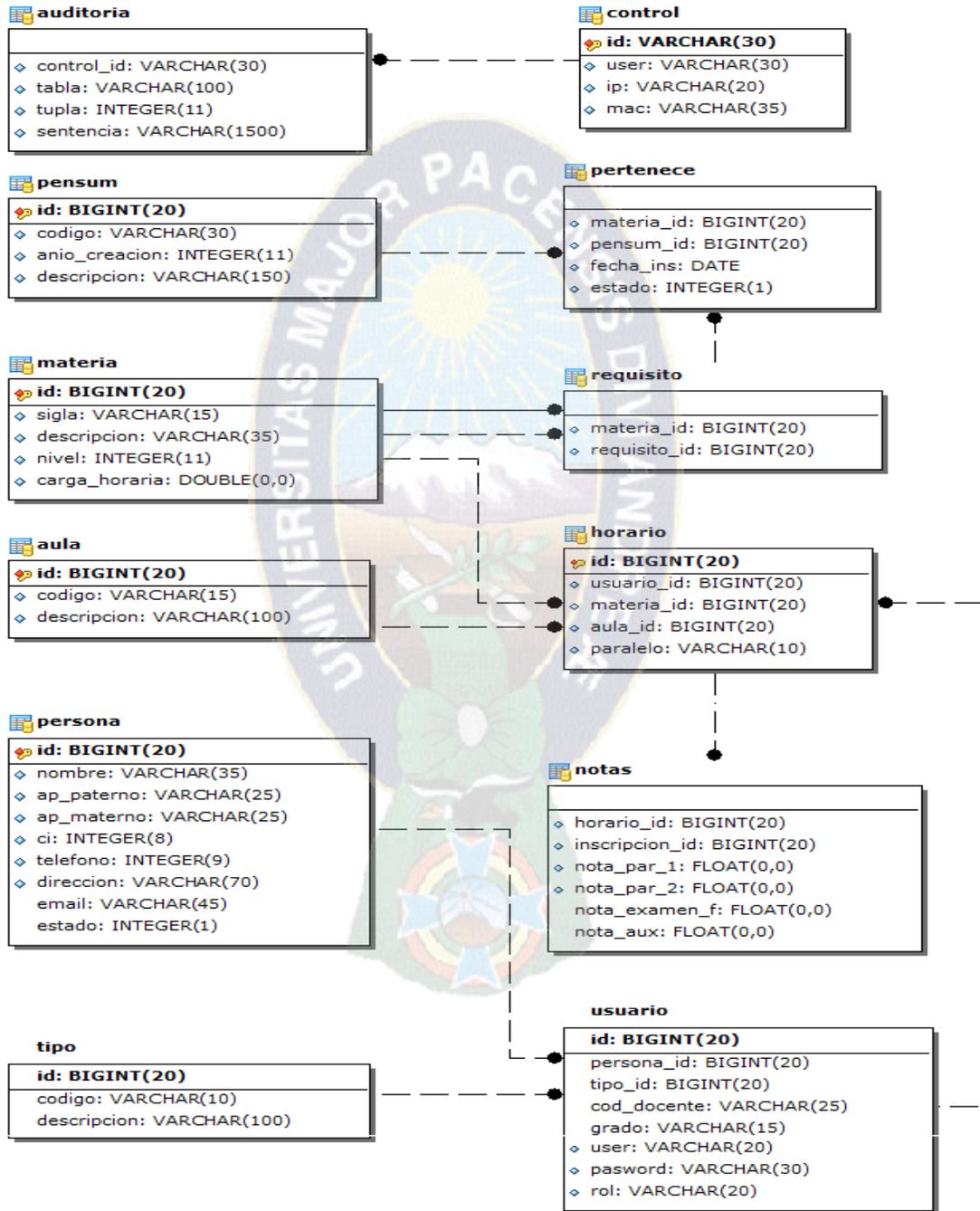


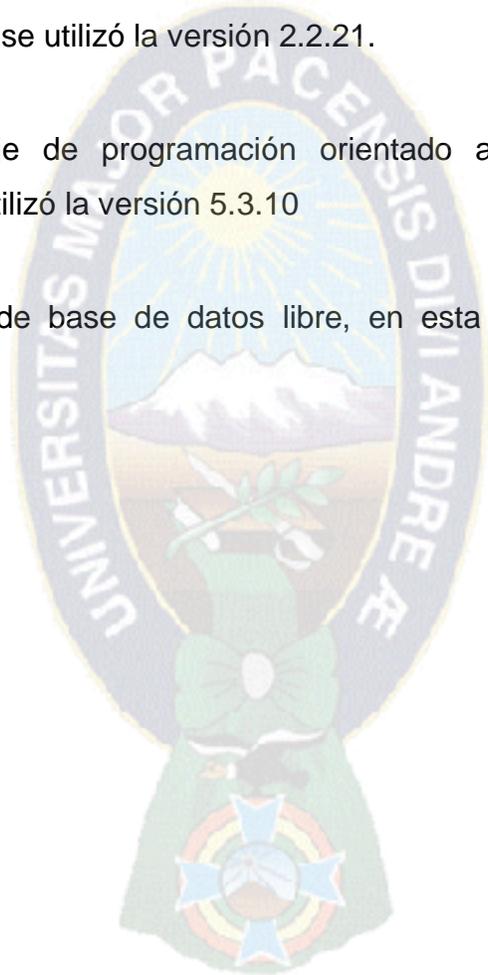
Figura 3. 10 Diagrama Nodo 2 para Docentes y Administrativos  
Fuente: Propia

Todo el sistema desarrollado será ejecutado en el servidor de aplicaciones web “WAMP server”, este servidor viene con tres elementos para el correcto funcionamiento de sistemas web lo cuales son:

Apache.- Es el encargado de conectar la base de datos y el lenguaje PHP, en esta investigación se utilizó la versión 2.2.21.

PHP 5.- Lenguaje de programación orientado a sistemas web, en esta investigación se utilizó la versión 5.3.10

MySQL.- Gestor de base de datos libre, en esta investigación se utilizó la versión 5.5.20



## CAPITULO IV

### PRUEBA DE HIPÓTESIS

Para la comprobación o refutación de la hipótesis planteada en esta investigación utilizaremos el software intrusivo “BADBOY” el cual cuenta con varias herramientas para medir cuantitativamente la funcionalidad del agente inteligente planteado para el incremento en el control de la seguridad de sistemas con bases de datos distribuidas.

Algunas de las características más importantes de este software son:

- Controles de tiempos de respuesta.
- Detección de errores.
- Pruebas de funcionamiento a los sistemas informáticos.
- Pruebas de Ataques (Injection, Sobrecarga, Fuerza bruta.)
- Pruebas de estrés.
- Pruebas de complejidad.
- Revisión en la carga de librerías.
- Expresar resultados en gráficas.
- Verificación de rutas de acceso.
- Pruebas con múltiples hilos

Es debido a estas características se eligió esta herramienta de testeado de sistemas para realizar la prueba de hipótesis en esta investigación.

## **4.1 PRUEBA DE FUNCIONALIDAD DEL AGENTE INTELIGENTE**

Para realizar las pruebas al agente con resultados que nos sirvan para llegar a una conclusión lógica y cuantitativa se necesitara realizar por lo menos dos pruebas en este caso se realizara la prueba con tres diferentes sistemas, la primera prueba a realizarse será sobre un sistema sin la implementación del agente, por otro lado la segunda y tercera prueba se la realizara en el sistemas en los que estará implementado el agente.

Las características de los sistemas deberán satisfacer los requerimientos de funcionamiento del agente los cuales están descritos en el capítulo tres.

### **4.1.1. PRUEBA DE SEGURIDAD SIN EL AGENTE**

Las prueba realizada a este sistema, mostraron los siguientes resultados descritos a continuación.

- Informe de Tiempo

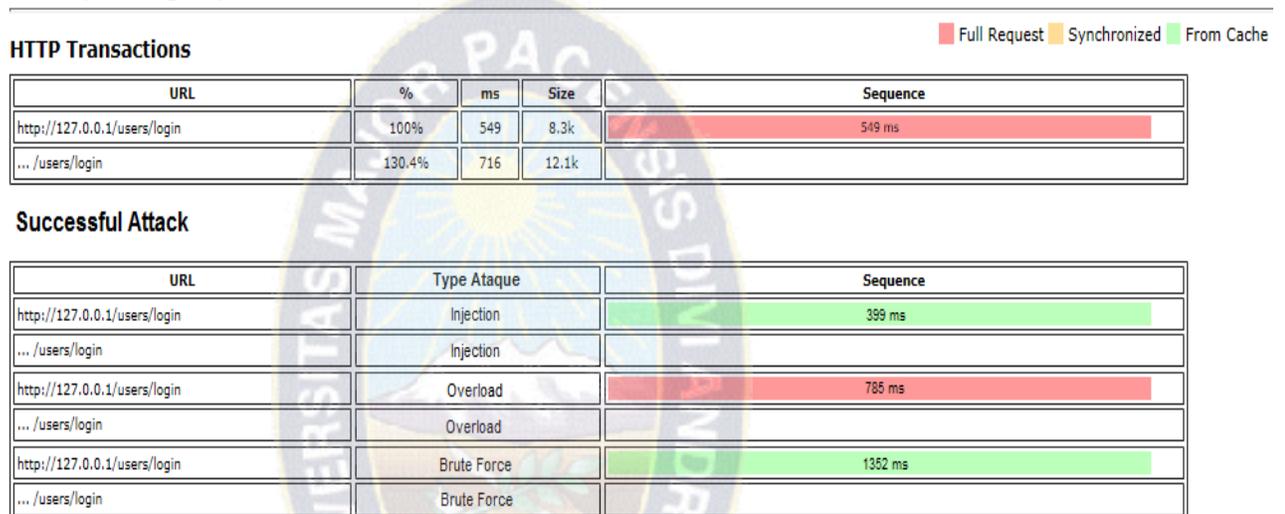
Este informe es en el que se reportara el tiempo en el que el software intrusivo pudo acceder o completar la evaluación o ataques al sistema.

Es por esto que esta prueba determinara el nivel de seguridad del sistema teniendo en cuenta el tiempo en el que el software logro acceder al sistema o completo el ataque sin éxito (no accedió al sistema).

Viendo los resultados de esta prueba vemos que en los tres ataques realizados por el software se logró el acceso al sistema utilizado "Injection" en un tiempo

de 339 ms y por “Fuerza bruta” en 1352 ms, por otro lado en cuanto a “Sobrecarga” el sistema completo el testeó en 785 ms como se muestra en la imagen.

### Badboy Timing Report



**Figura 4. 1 Informe de tiempos de respuesta sin agente**  
Fuente: Propia

Por lo que podemos concluir que este sistema es poco seguro ya que el tiempo en el cual el software logro irrumpir en el sistema por dos métodos de ataque y viendo los tiempos en el que se logró ingresar al sistema se ve la deficiencia en los ataques por inyección y una falta de control en el número de intentos de identificación fallidos

- Resultados General de la Prueba

Esta prueba muestra los tiempos de respuesta del sistema en las diferentes interfaces o funcionalidades del sistema y muestra la cantidad de errores producidos en la navegación por todo el sistema, estos errores se pueden deber

a fallos en las direcciones o errores en la carga de archivos de estilos o funciones en java script.

## Badboy Test Results

Report created 22:05 02 Dec 2013

### Overview

| Total Played | Succeeded | Failed | Warnings | Assertions | Average Time | Max Time |
|--------------|-----------|--------|----------|------------|--------------|----------|
| 28           | 26        | 2      | 0        | 0          | 858          | 1316     |

### Test Failed

[Expand All](#)

|                     | Total | Succ | Fail | Wrn | Asst | Avg | Max  |
|---------------------|-------|------|------|-----|------|-----|------|
| ☐ Coprobacion_tesis | 28    | 26   | 2    | 0   | 0    | 858 | 1316 |
| ☐ Test              | 28    | 26   | 2    | 0   | 0    | 858 | 1316 |
| ● Sistema_prueba    | 28    | 26   | 2    | 0   | 0    | 858 | 1316 |

### All Responses

| Id/Label | URL / Reference                               | Count | Status (Success or Failed) | Avg Time | Max Time |
|----------|---|-------|----------------------------|----------|----------|
| 20       | http://127.0.0.1/users/login                  | 1     | Success                    | 1316     | 1316     |
| 41       | http://127.0.0.1/users/login                  | 1     | Success                    | 1311     | 1311     |
| 54       | http://127.0.0.1/SpaMaterials/index           | 1     | Success                    | 822      | 822      |
| 58       | http://127.0.0.1/SpaMaterials/index/4         | 1     | Success                    | 806      | 806      |
| 62       | http://127.0.0.1/SpaMaterials/edit/4          | 1     | Success                    | 780      | 780      |
| 75       | http://127.0.0.1/spa_materials/edit/4         | 1     | Success                    | 1058     | 1058     |
| 88       | http://127.0.0.1/SpaTratamientos/index        | 1     | Success                    | 874      | 874      |
| 92       | http://127.0.0.1/SpaTratamientos/edit/1       | 1     | Success                    | 785      | 785      |
| 186      | http://127.0.0.1/SpaTratamientos/index        | 1     | Success                    | 835      | 835      |
| 195      | http://127.0.0.1/SpaTratamientoSesions/index  | 1     | Success                    | 826      | 826      |
| 199      | http://127.0.0.1/SpaTratamientoSesions/view/1 | 1     | Success                    | 741      | 741      |
| 205      | http://127.0.0.1/SpaTratamientoSesions/index  | 1     | Success                    | 815      | 815      |
| 209      | http://127.0.0.1/SpaTratamientoSesions/add/4  | 1     | Success                    | 797      | 797      |
| 257      | http://127.0.0.1/SpaTratamientoSesions/add/4  | 1     | Success                    | 1118     | 1118     |
| 268      | http://127.0.0.1/SpaTratamientoSesions/view/4 | 1     | Success                    | 700      | 700      |
| 277      | http://127.0.0.1/SpaMaterialSesions/index     | 1     | Success                    | 880      | 880      |
| 281      | http://127.0.0.1/SpaMaterialSesions/view/1    | 1     | Success                    | 792      | 792      |
| 287      | http://127.0.0.1/SpaMaterialSesions/index     | 1     | Success                    | 741      | 741      |
| 291      | http://127.0.0.1/SpaEquipoSesions/add/2       | 1     | Success                    | 799      | 799      |
| 304      | http://127.0.0.1/SpaEquipoSesions/add/2       | 1     | Success                    | 1027     | 1027     |
| 314      | http://127.0.0.1/SpaMaterialSesions/view/2    | 1     | Success                    | 710      | 710      |
| 323      | http://127.0.0.1/SpaPromocions/index          | 1     | Success                    | 931      | 931      |
| 327      | http://127.0.0.1/SpaPromocions/view/18        | 1     | Failed - Error 404         | 364      | 364      |
| 336      | http://127.0.0.1/SpaPromocions/index          | 1     | Success                    | 824      | 824      |

Figura 4. 2 Resultado general de la prueba sin agente  
Fuente: Propia

- Control de Tiempos por Hilos (Thread)

Esta prueba se la realizara para ver la cantidad de procesos simultáneos que puede soportar el sistema sin afectar su tiempo de respuesta.

| Url  | Responses | Errors | Avg Tim ^ |
|--|-----------|--------|-----------|
| 127.0.0.1/Users/logout (370)                 | 12        | 0      | 713       |
| 127.0.0.1/SpaReservaTratamientos/index (352) | 12        | 0      | 448       |
| 127.0.0.1/SpaPromocions/view/18 (327)        | 12        | 0      | 344       |
| 127.0.0.1/SpaPromocions/index (336)          | 12        | 0      | 342       |
| 127.0.0.1/SpaPromocions/index (323)          | 13        | 0      | 351       |
| 127.0.0.1/SpaPromocions/edit/18 (340)        | 12        | 0      | 436       |
| 127.0.0.1/SpaMaterials/index (361)           | 12        | 0      | 346       |
| 127.0.0.1/SpaMaterialSesions/view/2 (314)    | 13        | 0      | 336       |
| 127.0.0.1/SpaTratamientos/index (186)        | 17        | 0      | 346       |
| 127.0.0.1/SpaTratamientos/edit/1 (92)        | 18        | 0      | 373       |
| 127.0.0.1/SpaTratamientoSesions/view/4 (268) | 16        | 0      | 353       |
| 127.0.0.1/SpaTratamientoSesions/view/1 (199) | 16        | 0      | 334       |
| 127.0.0.1/SpaTratamientoSesions/index (205)  | 16        | 0      | 334       |
| 127.0.0.1/SpaTratamientoSesions/index (195)  | 17        | 0      | 344       |
| 127.0.0.1/SpaTratamientoSesions/add/4 (257)  | 16        | 0      | 728       |
| 127.0.0.1/SpaTratamientoSesions/add/4 (209)  | 16        | 0      | 366       |
| 127.0.0.1/SpaMaterialSesions/view/1 (281)    | 15        | 0      | 346       |
| 127.0.0.1/SpaMaterialSesions/index (287)     | 15        | 0      | 344       |
| 127.0.0.1/SpaMaterialSesions/index (277)     | 16        | 0      | 356       |
| 127.0.0.1/SpaEquipoSesions/add/2 (304)       | 15        | 0      | 648       |
| 127.0.0.1/SpaEquipoSesions/add/2 (291)       | 15        | 0      | 360       |
| 127.0.0.1/spa_materials/edit/4 (75)          | 20        | 0      | 659       |
| 127.0.0.1/SpaTratamientos/index (88)         | 18        | 0      | 338       |
| 127.0.0.1/SpaMaterials/edit/4 (62)           | 20        | 0      | 364       |
| 127.0.0.1/users/login (41)                   | 42        | 0      | 945       |

**Figura 4. 3 Control de hilos sin agente**  
Fuente: Propia

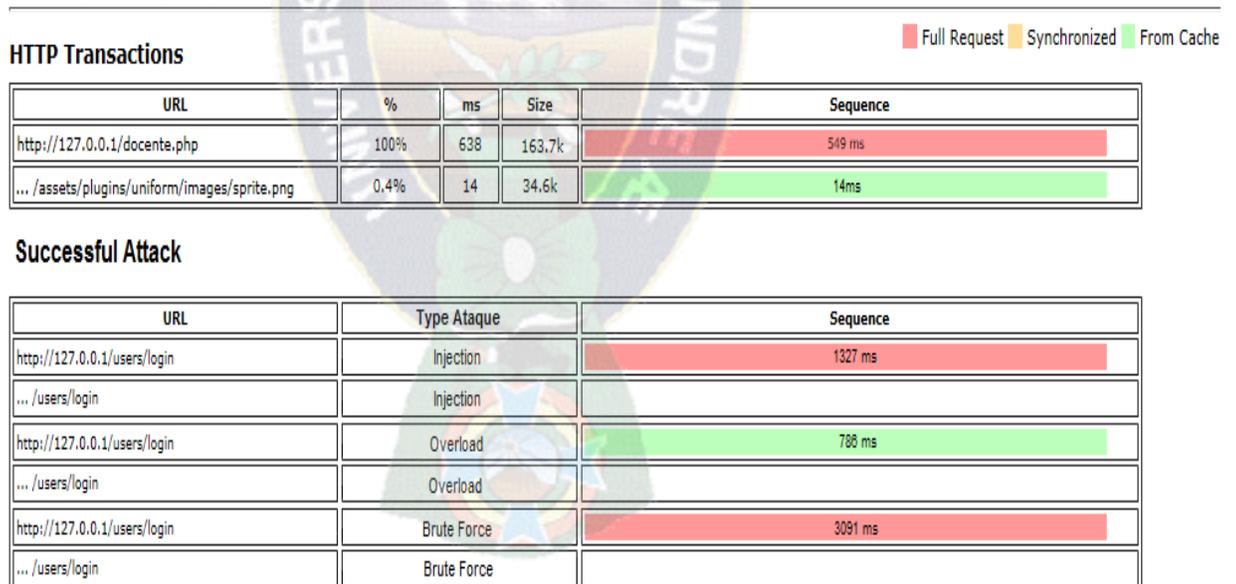
#### 4.1.2. PRUEBA DE SEGURIDAD CON EL AGENTE EN EL SISTEMA “A”

La prueba realizada al sistema “A”, mostro los siguientes resultados.

- Informe de Tiempo

Como se puede apreciar en la Figura 4.4 el sistema “A” soporta los ataques de “Injection” en un tiempo de 1327 ms y “Fuerza bruta” en 3091 ms, por otro lado el ataque con “Sobrecarga” cuelga o inhabilita el sistema en un tiempo de 786 ms

#### Badboy Timing Report



**Figura 4. 4 Informe de tiempos de respuesta con agente**  
Fuente: Propia

Con estos resultados podemos concluir que el incremento en la seguridad en el sistema “A” fe exitoso ya que el software no pudo irrumpir en este sistema, como nos muestran las pruebas, por otro lado se ve que el sistema es más

pesado o lento en el procesamiento de las solicitudes debido a la revisión efectuada por el agente y esto derivó en que con el ataque de sobrecarga el sistema “A” colapsara.

- Resultados General de la Prueba

Esta prueba se la realizara a todo el sistema viendo los tiempos de respuesta de cada una de las interfaces que posee el sistema de prueba

### Badboy Test Results

Report created 23:17 02 Dec 2013

#### Overview

| Total Played | Succeeded | Failed | Warnings | Assertions | Average Time | Max Time |
|--------------|-----------|--------|----------|------------|--------------|----------|
| 13           | 12        | 1      | 0        | 0          | 1192         | 3363     |

#### Test Failed

[Expand All](#)

- [-] Test Suite 1
  - [-] Test 1
    - Step 1

| Total | Succ | Fail | Wrn | Asst | Avg  | Max  |
|-------|------|------|-----|------|------|------|
| 13    | 12   | 1    | 0   | 0    | 1192 | 3363 |
| 13    | 12   | 1    | 0   | 0    | 1192 | 3363 |
| 13    | 12   | 1    | 0   | 0    | 1192 | 3363 |

#### All Responses

| Id/Label | URL / Reference                 | Count | Status (Success or Failed) | Avg Time | Max Time |
|----------|---------------------------------|-------|----------------------------|----------|----------|
| 5        | http://127.0.0.1/               | 1     | Success                    | 650      | 650      |
| 15       | http://127.0.0.1/login.php      | 1     | Success                    | 668      | 668      |
| 24       | http://127.0.0.1/index.php      | 1     | Success                    | 522      | 522      |
| 33       | http://127.0.0.1/docente.php    | 1     | Success                    | 783      | 783      |
| 42       | http://127.0.0.1/pensum.php     | 1     | Failed - Error 404         | 28       | 28       |
| 45       | http://127.0.0.1/docente.php    | 1     | Success                    | 673      | 673      |
| 54       | http://127.0.0.1/plan.php       | 1     | Success                    | 822      | 822      |
| 63       | http://127.0.0.1/_docentes.php  | 1     | Success                    | 2971     | 2971     |
| 99       | http://127.0.0.1/index.php      | 1     | Success                    | 617      | 617      |
| 108      | http://127.0.0.1/estudiante.php | 1     | Success                    | 704      | 704      |
| 207      | http://127.0.0.1/estudiante.php | 1     | Success                    | 2906     | 2906     |
| 654      | http://127.0.0.1/docente.php    | 1     | Success                    | 786      | 786      |
| 810      | http://127.0.0.1/docente.php    | 1     | Success                    | 3363     | 3363     |

Figura 4. 5 Resultado general de la prueba con agente  
Fuente: Propia

En esta prueba se muestra los tiempos de respuesta y errores que se pueden producir al navegar por el sistema.

- Control de tiempos por Hilos (Thread)

Al igual que en la prueba realizada al sistema sin el agente esta prueba sirve para medir el tiempo de respuesta del sistema con varios procesos simultáneos.

Como se puede apreciar en la Figura 4.6 el sistema no presenta problemas de tiempo de respuesta en cuanto a interfaces que solo muestran la información almacenada en los nodos, pero cuando se realiza algún tipo de registro o replicación de datos entre los nodos, el tiempo de respuesta se incrementa abruptamente.

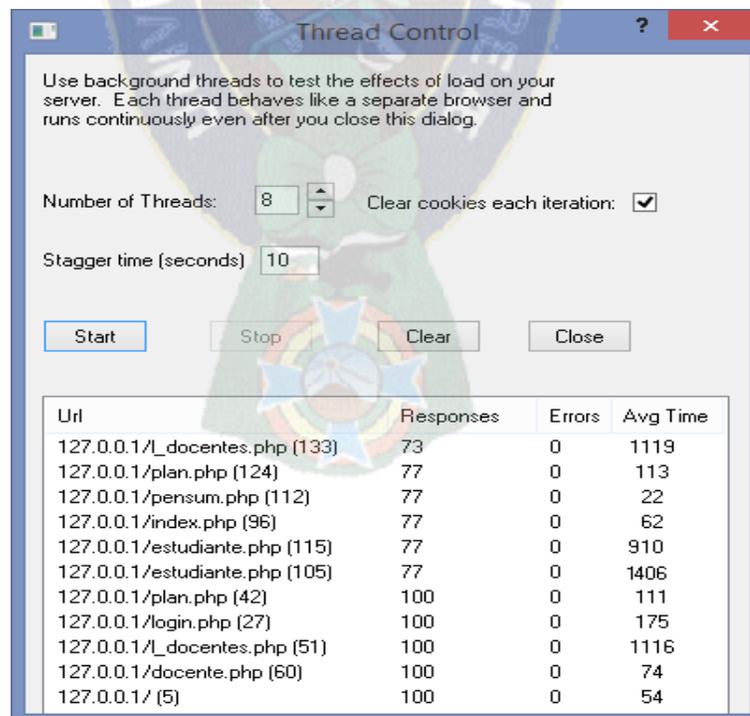


Figura 4. 6 Control de hilos con agente  
Fuente: Propia

Este control nos sirve como una prueba de estrés del sistema “A” y verifica las variaciones en los tiempos de respuesta del sistema cuando se ejecutan múltiples acciones a la vez.

#### 4.1.2. PRUEBA DE SEGURIDAD CON EL AGENTE EN EL SISTEMA “B”

La prueba realizada al sistema “B”, mostro los siguientes resultados.

- Informe de Tiempo

Como se muestra en la Figura 4.7 los ataques al sistema no fueron exitosos ya que el sistema “B” reporto los siguientes resultados: el ataque de Injection se completó en 971 ms, se completó el test de Sobrecarga en 1829 ms y el test de Fuerza bruta se lo completo en 1096 ms.

##### Badboy Timing Report

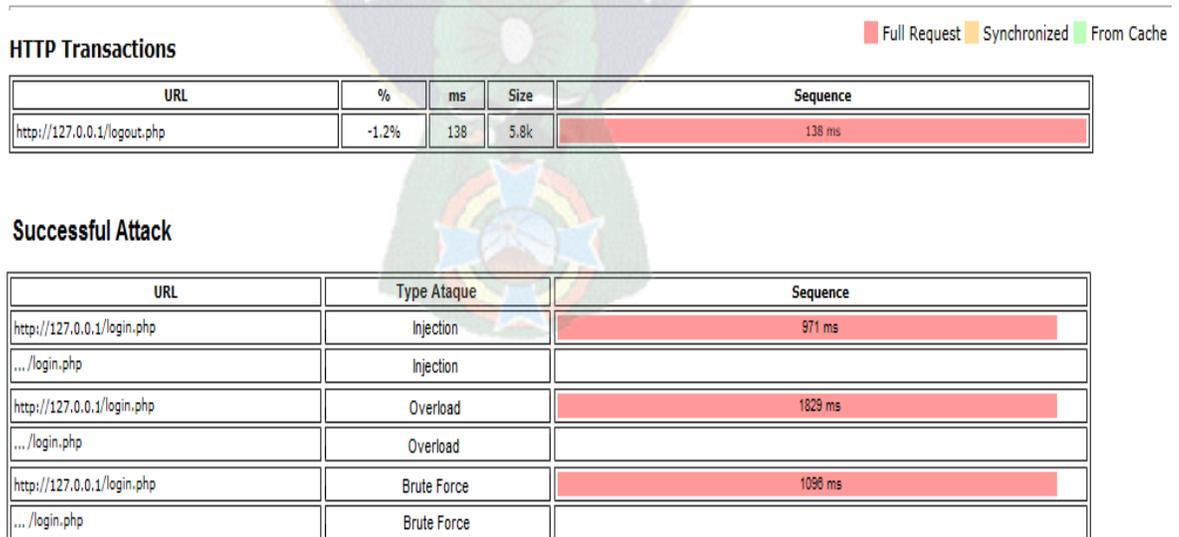


Figura 4. 7 Informe de tiempos de respuesta con agente en el sistema B  
Fuente: Propia

Los resultados obtenidos en esta prueba apoyan a los que se obtuvieron en la prueba que se realizó al sistema “A”, por lo cual podemos concluir que el agente mejoro el control en la seguridad de los sistemas.

- Resultados General de la Prueba con el sistema “B”

Esta prueba se la realizara a todo el sistema “B” viendo los tiempos de respuesta de cada una de las interfaces que posee este sistema, este tiempo de respuesta servirán para ver cuáles son las interfaces del sistema que pueden afectar el rendimiento del mismo.

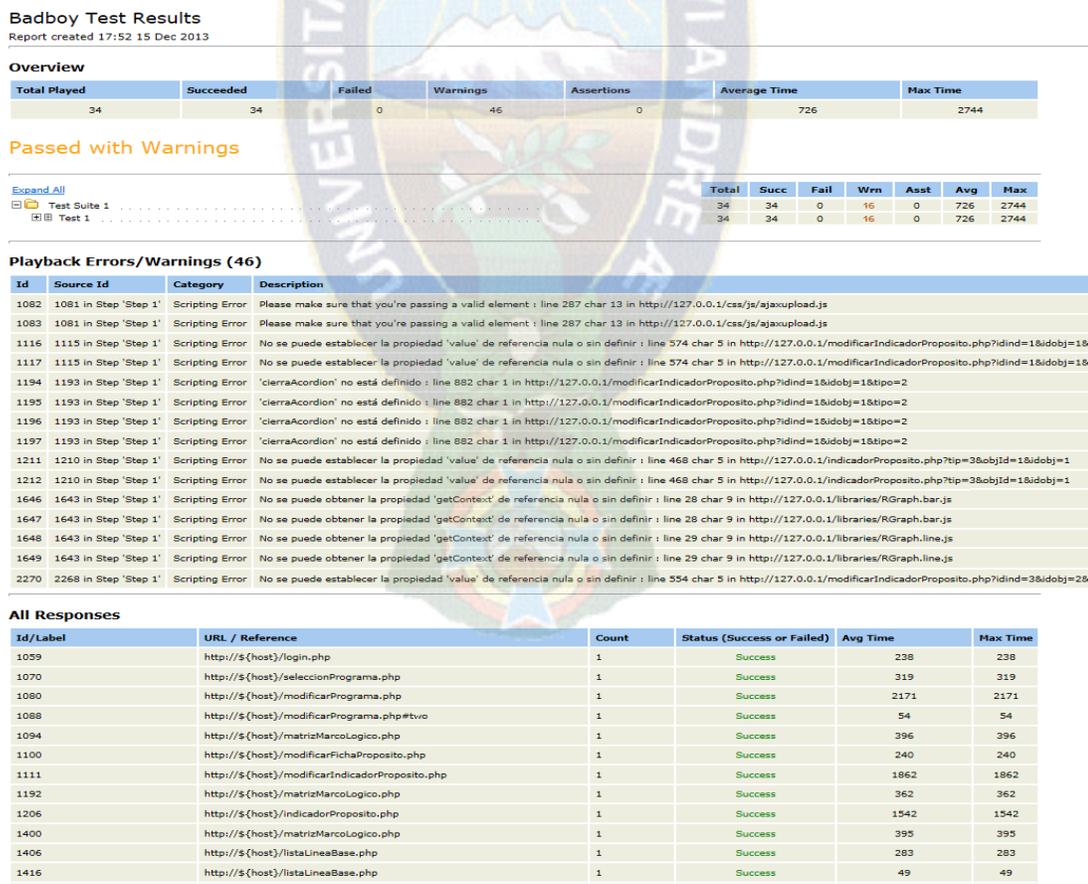


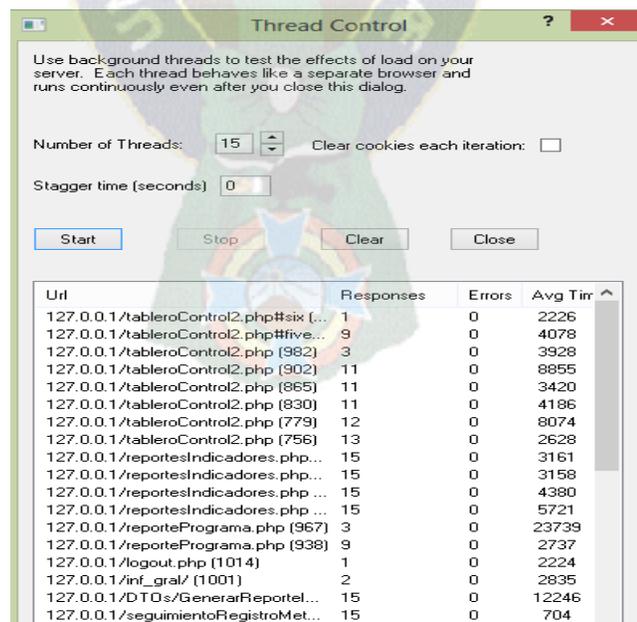
Figura 4. 8 Resultado general de la prueba con agente en el sistema “B”  
Fuente: Propia

Como se puede ver en la Figura 4.8 se tiene algunas alertas correspondientes al uso de algunas funciones y estilos utilizados en el sistema “B”.

- Control de tiempos por Hilos (Thread) en el sistema “B”

Al igual que en la prueba realizada al sistema “A” esta prueba sirve para medir el tiempo de respuesta del sistema con varios procesos simultáneos.

Como se puede apreciar en la Figura 4.9 el sistema no presenta problemas de tiempo de respuesta en cuanto a interfaces que solo muestran la información almacenada en los nodos, pero cuando se realiza algún tipo de registro o replicación de datos entre los nodos, el tiempo de respuesta se incrementa abruptamente.



The screenshot shows a window titled "Thread Control" with a help icon and a close button. The window contains the following text and controls:

Use background threads to test the effects of load on your server. Each thread behaves like a separate browser and runs continuously even after you close this dialog.

Number of Threads: 15 Clear cookies each iteration:

Stagger time (seconds) 0

Start Stop Clear Close

| Url                                     | Responses | Errors | Avg Tirm |
|---|-----------|--------|----------|
| 127.0.0.1/tableroControl2.php#six (...) | 1         | 0      | 2226     |
| 127.0.0.1/tableroControl2.php#five...   | 9         | 0      | 4078     |
| 127.0.0.1/tableroControl2.php (982)     | 3         | 0      | 3928     |
| 127.0.0.1/tableroControl2.php (902)     | 11        | 0      | 8855     |
| 127.0.0.1/tableroControl2.php (865)     | 11        | 0      | 3420     |
| 127.0.0.1/tableroControl2.php (830)     | 11        | 0      | 4186     |
| 127.0.0.1/tableroControl2.php (779)     | 12        | 0      | 8074     |
| 127.0.0.1/tableroControl2.php (756)     | 13        | 0      | 2628     |
| 127.0.0.1/reportesIndicadores.php...    | 15        | 0      | 3161     |
| 127.0.0.1/reportesIndicadores.php...    | 15        | 0      | 3158     |
| 127.0.0.1/reportesIndicadores.php ...   | 15        | 0      | 4380     |
| 127.0.0.1/reportesIndicadores.php ...   | 15        | 0      | 5721     |
| 127.0.0.1/reportePrograma.php (967)     | 3         | 0      | 23739    |
| 127.0.0.1/reportePrograma.php (938)     | 9         | 0      | 2737     |
| 127.0.0.1/logout.php (1014)             | 1         | 0      | 2224     |
| 127.0.0.1/inf_gral/ (1001)              | 2         | 0      | 2835     |
| 127.0.0.1/DTOs/GenerarReportel...       | 15        | 0      | 12246    |
| 127.0.0.1/seguimientoRegistroMet...     | 15        | 0      | 704      |

**Figura 4. 9 Control de hilos con agente en el sistema "B"**  
Fuente: Propia

Este control nos sirve como una prueba de estrés del sistema “B” y verifica las variaciones en los tiempos de respuesta del sistema cuando se ejecutan múltiples acciones a la vez.

#### **4.1.3. GRAFICA RESULTANTE COMPARATIVA DE TIEMPOS DE EJECUCIÓN**

Esta grafica resultante realiza una comparación de los tiempos de respuesta entre las dos pruebas realizadas al sistema sin el agente y con el agente.

Como podemos apreciar en la figura 4.10 la diferencia en tiempos de respuesta se hace evidente en cuanto a las dos pruebas, la primera prueba del sistema sin agente muestra un tiempo de respuesta estable con una diferencia mínima entre sus ocurrencias (Denotado por el círculo azul).

A diferencia de los tiempos de respuesta del sistema con el agente inteligente que mostro un tiempo de respuesta muy bajo en las interfaces que solo muestran datos o informativas (denotado por el rectángulo rojo inferior), pero en las interfaces que necesitan de la revisión de los datos ingresados al sistema por el agente se muestra un incremento en el tiempo de respuesta del sistema (denotado por el rectángulo rojo superior).

Por el tamaño de la figura 4.10 se la dejara en la página siguiente.

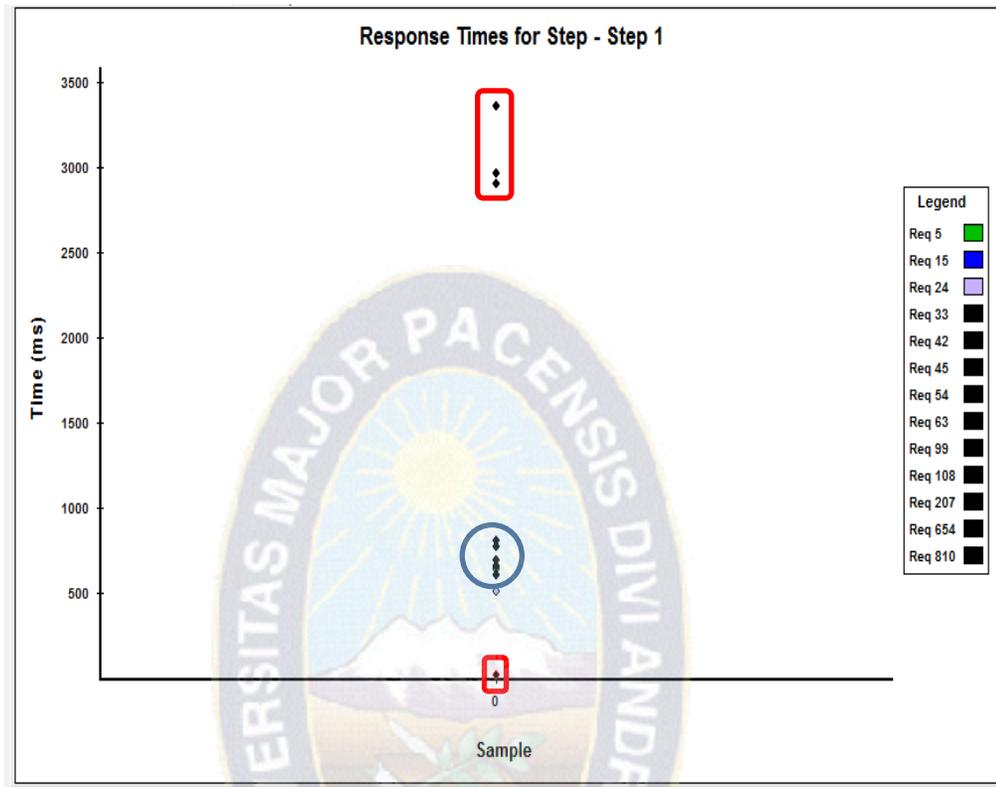


Figura 4. 10 Comparación grafica de tiempos de respuesta  
Fuente: Propia

#### 4.1.4. TABLA COMPARATIVA DE ATAQUES

Describe el resultado de los ataques realizados por el software BADBOY.

| Tipo de Ataque | Sistema sin el agente              | Sistema "A" con el agente                              | Sistema "B" con el agente                             |
|----------------|------------------------------------|--|---|
| Injection      | Irrumpió en el sistema en 339 (ms) | Completo la revisión en 1327 (ms) sin lograr irrumpir. | Completo la revisión en 971 (ms) sin lograr irrumpir. |

|              |  |   |   |
|--------------|--|---|---|
| Sobrecarga   | Completo la revisión en 785 (ms) sin sobrecargar el sistema. | Sobrecargo el sistema en 786 (ms) haciendo que colapsara. | Completo la revisión en 1829 (ms) sin sobrecargar el sistema. |
| Fuerza Bruta | Irrumpió en el sistema en 1352 (ms)                          | Completo la revisión en 3091 (ms) sin lograr irrumpir.    | Completo la revisión en 1096 (ms) sin lograr irrumpir.        |

**Tabla 15 Tabla comparativa de los resultados del ataque**  
Fuente: Propia

En esta tabla se resume el resultado de los ataques realizados por el software intrusivo “BADBOY”, con el fin de comprobar el incremento en el control de la seguridad al utilizar es agente inteligente.

Como se puede apreciar en la tabla los sistemas “A y B” en los que se implementó el agente inteligente repelieron exitosamente los ataques del software intrusivo, por otro lado el sistema que no contaba con el agente mostraron debilidades en los ataques de Injection y Fuerza bruta ya que el software pudo acceder a este sistema por estos dos métodos.

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

La implementación del agente inteligente desarrollado en esta investigación para el control de la seguridad en sistemas web desarrollados en el lenguaje PHP mejoró el control en la seguridad de los sistemas de información con bases de datos distribuidas. Este incremento en el control de la seguridad se debe a que el agente cuenta con funciones para la detección de SQL-injection, la identificación de los usuarios y la detección de la IP y Mac de cualquier persona que trata de ingresar al sistema, con lo cual podemos prevenir el ataque por fuerza bruta.

Por medio del estudio a las metodologías de testeado de seguridad se determinaron que las principales amenazas a la seguridad en sistemas con bases de datos distribuidas son: “SQL-injection, Fuerza bruta, Errores de conexión, Caídas del sistema, Replicación de la información”.

Se proporcionó al agente inteligente de un módulo que permite el control de la autenticación de los usuarios, registrando el IP y MAC del usuario que desea identificarse y verificando el id y password del mismo.

A través de la metodología Prometheus se pudo apreciar que los posibles conflictos se darán en el momento en que se replica la información de un nodo a los otros nodos pertenecientes al sistema.

Para la aplicación de las metodologías de testeo y la comprobación de la hipótesis planteada en esta investigación, se utilizó el software intrusivo “BADBOY” que dio resultados favorables sobre el incremento del control de la seguridad con el uso del agente inteligente ya que el sistema con el agente pudo soportar el ataque de inyección y fuerza bruta efectuado por el software intrusivo “BADBOY”.

Por otro lado aunque el control de la seguridad en los sistemas de prueba mejoraron considerablemente, el tiempo de respuesta -tanto en la identificación como en cualquier proceso de extracción, registro o modificación de los datos registrados en estos sistemas- sufrió un incremento abrupto en el tiempo de procesamiento.

Este incremento en el tiempo de procesamiento del sistema se debe a las comprobaciones que efectúa el agente a cualquier dato que se desea ingresar o registrar en el sistema, también dicha demora se debe a las transacciones de datos de nodo a nodo del sistema.

## **5.2 RECOMENDACIONES**

La seguridad es muy importante dentro del procesamiento de datos, pues como se dijo al principio de la investigación, los datos son importantes para todas las personas interesadas, pues ayudan a la toma de decisiones. Desde este punto de vista se recomienda la implementación del agente en sistemas que cuenten con servidores altamente modernos pues tienen una gran capacidad de procesamiento de datos, para que el problema sobre el tiempo de procesamiento disminuya y la seguridad de los datos mejore.

Se recomienda que el agente inteligente sea implementado en sistemas basados en el Modelo Vista Controlador, en lenguaje php y con una programación orientada a objetos; pues son las condiciones básicas para el correcto funcionamiento de los agentes de control.



# ÍNDICE DE FIGURAS

## CAPITULO 2

|  |    |
|--|----|
| <i>Figura 2. 1 Mapa de Seguridad</i> .....                                     | 20 |
| <i>Figura 2. 2 Modelado de un sistema de Bases de Datos Distribuidas</i> ..... | 26 |
| <i>Figura 2. 3 Integración lógica top-down</i> .....                           | 27 |
| <i>Figura 2. 4 Integración lógica botton-up</i> .....                          | 28 |
| <i>Figura 2. 5 Naturaleza polifacética</i> .....                               | 37 |
| <i>Figura 2. 6 Pirámide del conocimiento</i> .....                             | 38 |
| <i>Figura 2. 7 Arquitectura básica de un agente</i> .....                      | 46 |
| <i>Figura 2. 8 Agente basado en tabla</i> .....                                | 48 |
| <i>Figura 2. 9 Modelo de agente reactivo</i> .....                             | 49 |
| <i>Figura 2. 10 Modelo de agente reactivo con estado interno</i> .....         | 50 |
| <i>Figura 2. 11 Modelo de agente basado en objetivos</i> .....                 | 51 |
| <i>Figura 2. 12 Modelo de agente basado en la utilidad</i> .....               | 52 |
| <i>Figura 2. 13 Metodología de diseño Prometheus</i> .....                     | 55 |

## CAPITULO 3

|   |    |
|---|----|
| <i>Figura 3. 1 Diagrama de descripción general de escenarios</i> .....                    | 65 |
| <i>Figura 3. 2 Diagrama de descripción general de objetivos o metas del sistema</i> ..... | 66 |
| <i>Figura 3. 3 Diagrama general de agrupación por rol</i> .....                           | 67 |
| <i>Figura 3. 4 Diagrama general de acoplamiento con los datos</i> .....                   | 68 |
| <i>Figura 3. 5 Diagrama de análisis descriptivo general</i> .....                         | 70 |

|  |           |
|--|-----------|
| <i>Figura 3. 6 Diagrama de especificación de actividades según rol .....</i> | <i>71</i> |
| <i>Figura 3. 7 Diagrama de relaciones del sistema .....</i>                  | <i>73</i> |
| <i>Figura 3. 8 Modelo vista controlador .....</i>                            | <i>74</i> |
| <i>Figura 3. 9 Diagrama de Nodo 1 para estudiante.....</i>                   | <i>76</i> |
| <i>Figura 3. 10 Diagrama Nodo 2 para Docentes y Administrativos.....</i>     | <i>77</i> |

## **CAPITULO 4**

|  |           |
|--|-----------|
| <i>Figura 4. 1 Informe de tiempos de respuesta sin agente .....</i>                  | <i>81</i> |
| <i>Figura 4. 2 Resultado general de la prueba sin agente .....</i>                   | <i>82</i> |
| <i>Figura 4. 3 Control de hilos sin agente .....</i>                                 | <i>83</i> |
| <i>Figura 4. 4 Informe de tiempos de respuesta con agente .....</i>                  | <i>84</i> |
| <i>Figura 4. 5 Resultado general de la prueba con agente.....</i>                    | <i>85</i> |
| <i>Figura 4. 6 Control de hilos con agente .....</i>                                 | <i>86</i> |
| <i>Figura 4. 7 Informe de tiempos de respuesta con agente en el sistema B .....</i>  | <i>87</i> |
| <i>Figura 4. 8 Resultado general de la prueba con agente en el sistema "B" .....</i> | <i>88</i> |
| <i>Figura 4. 9 Control de hilos con agente en el sistema "B".....</i>                | <i>89</i> |
| <i>Figura 4. 10 Comparación grafica de tiempos de respuesta .....</i>                | <i>91</i> |

## BIBLIOGRAFÍA

Anonimo. (2011). *Linux Máxima Seguridad – Edición especial*. California: Prentice Hall.

Anonimo. (2012). *Catarina.udlap.mx*. Recuperado el 20 de 06 de 2013, de *Catarina.udlap.mx*:  
[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/camargo\\_m\\_d/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/camargo_m_d/capitulo3.pdf)

Camfar. (08 de 2010). *Bases de datos distribuidos*. Recuperado el 02 de 06 de 2013, de *Bases de datos distribuidos*:  
<http://www.buenastareas.com/ensayos/Base-De-Datos-Distribuidos/152710.html>

Choque, D. G. (12 de 04 de 2013). *Gnomio.com*. Recuperado el 15 de 04 de 2013, de *Gnomio.com*:  
[http://saguicas.gnomio.com/file.php/2/IAp04\\_AgentesInteligentes.pdf](http://saguicas.gnomio.com/file.php/2/IAp04_AgentesInteligentes.pdf)

Ferré, A. P. (25 de 2 de 2010). *Inteligencia Artificial*. Recuperado el 19 de Septiembre de 2013, de *Inteligencia Artificial*:  
<http://www.robot.uji.es/docencia/II28/teoria/transparencias-tema02.pdf>

Hernández, A. G. (2011). *Sistemas Multiagente*. Recuperado el 28 de 11 de 2013, de *Metodologia Prometheus y PDT*:  
<http://www.uv.mx/aguerra/documents/2011-sma-slides-08.pdf>

Herzog, P. (20 de 10 de 2000). *OSSTMM2.1. Manual de la metodología abierta de testeo de seguridad*. Recuperado el 16 de 06 de 2013, de OSSTMM2.1. Manual de la metodología abierta de testeo de seguridad.: <http://isecom.securenetltd.com/OSSTMM.es.2.1.pdf>

Herzog, P. (14 de 12 de 2010). *OSSTMM3. Manual de la metodología abierta de testeo de seguridad*. Recuperado el 18 de 06 de 2013, de OSSTMM3. Manual de la metodología abierta de testeo de seguridad: <http://www.isecom.org/mirror/OSSTMM.3.pdf>

Lin Padgham, J. T. (2009). *Prometheus Design Tool*. Recuperado el 13 de agosto de 2013, de Prometheus Design Tool: <http://www.cs.rmit.edu.au/agents/pdt/docs/PDT-Manual.pdf>

Luya, k. M. (2012). *Bases de Datos*. Recuperado el 19 de 11 de 2013, de Bses de datos: <http://kevinmartell91.blogspot.com/2012/11/7-bases-de-datos-distribuidas.html>

Martell, K. (7 de 11 de 2012). *Bases de Datos Distribuidas: soluciones homogéneas y heterogéneas, integradas, federadas o multibase*. Recuperado el 07 de 11 de 2013, de <http://kevinmartell91.blogspot.com/2012/11/7-bases-de-datos-distribuidas.html>

Miqueleiz, F. (13 de 5 de 2003). *Los fundamentos de la inteligencia artificial*. Recuperado el 28 de 10 de 2013, de Los fundamentos de la inteligencia

artificial:

<http://www.angelfire.com/falcon/miqueleiz/03FundamentosdelalA.htm>

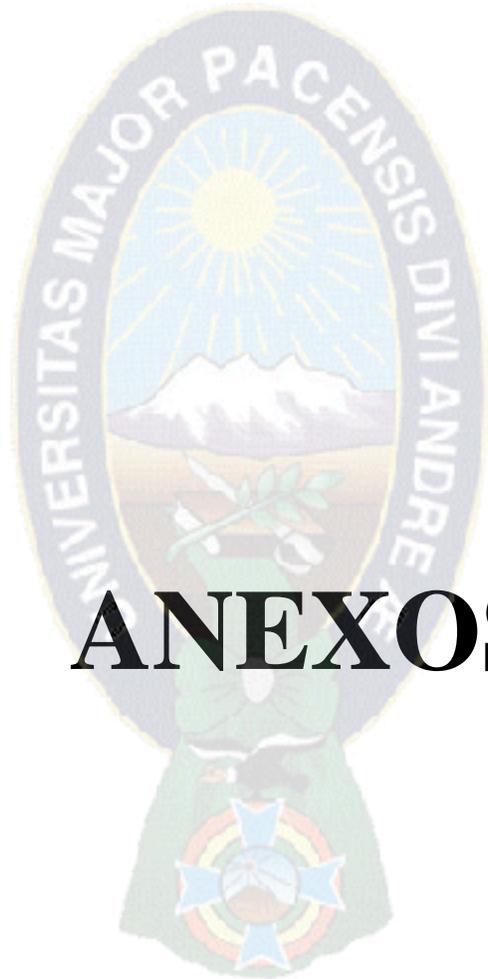
Montero, D. C. (2007). *Universidad de las Américas Puebla*. Recuperado el 27 de Agosto de 2013, de Universidad de las Américas Puebla: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/camargo\\_m\\_d/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/camargo_m_d/capitulo3.pdf)

Proal, C. (2013). *5. BASES DE DATOS DISTRIBUIDAS*. Recuperado el 17 de 10 de 2013, de 5. BASES DE DATOS DISTRIBUIDAS: <http://carlosproal.com/bda/bda05.html>

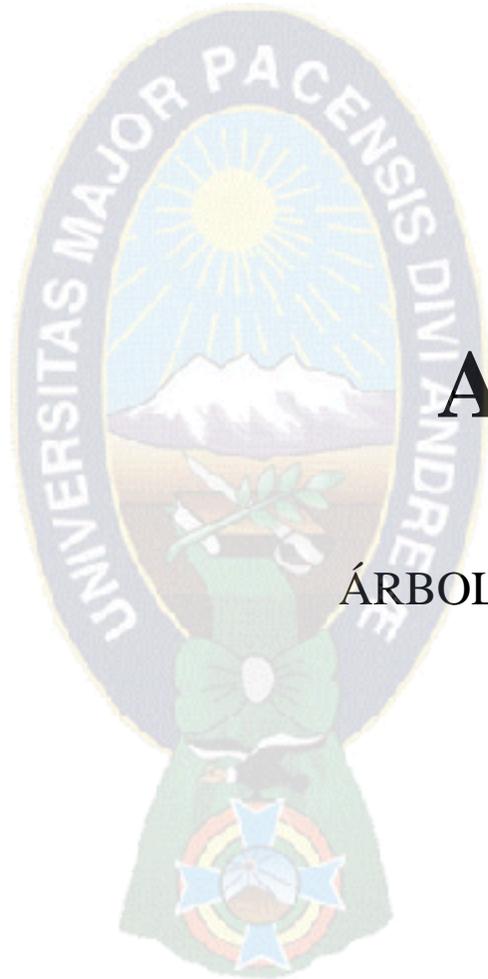
Valencia, U. P. (09 de 2011). *Historia de las bases de Datos*. Recuperado el 07 de 05 de 2013, de Historia de las bases de Datos: <http://histinf.blogs.upv.es/2011/01/04/historia-de-las-bases-de-datos/>

Vicente Toledo, I. M. (15 de 7 de 2008). *Bases de Datos Distribuidas*. Recuperado el 17 de 10 de 2013, de Bases de Datos Distribuidas: <https://iessanvicente.com/colaboraciones/BBDDdistribuidas.pdf>

Villamarín, A. (11 de 04 de 2013). *Modelo Vista Controlador adaptado a la web*. Recuperado el 06 de 12 de 2013, de Modelo Vista Controlador adaptado a la web: <http://ant.onio.org/2013/04/10/modelo-vista-controlador-adaptado-a-la-web.html>



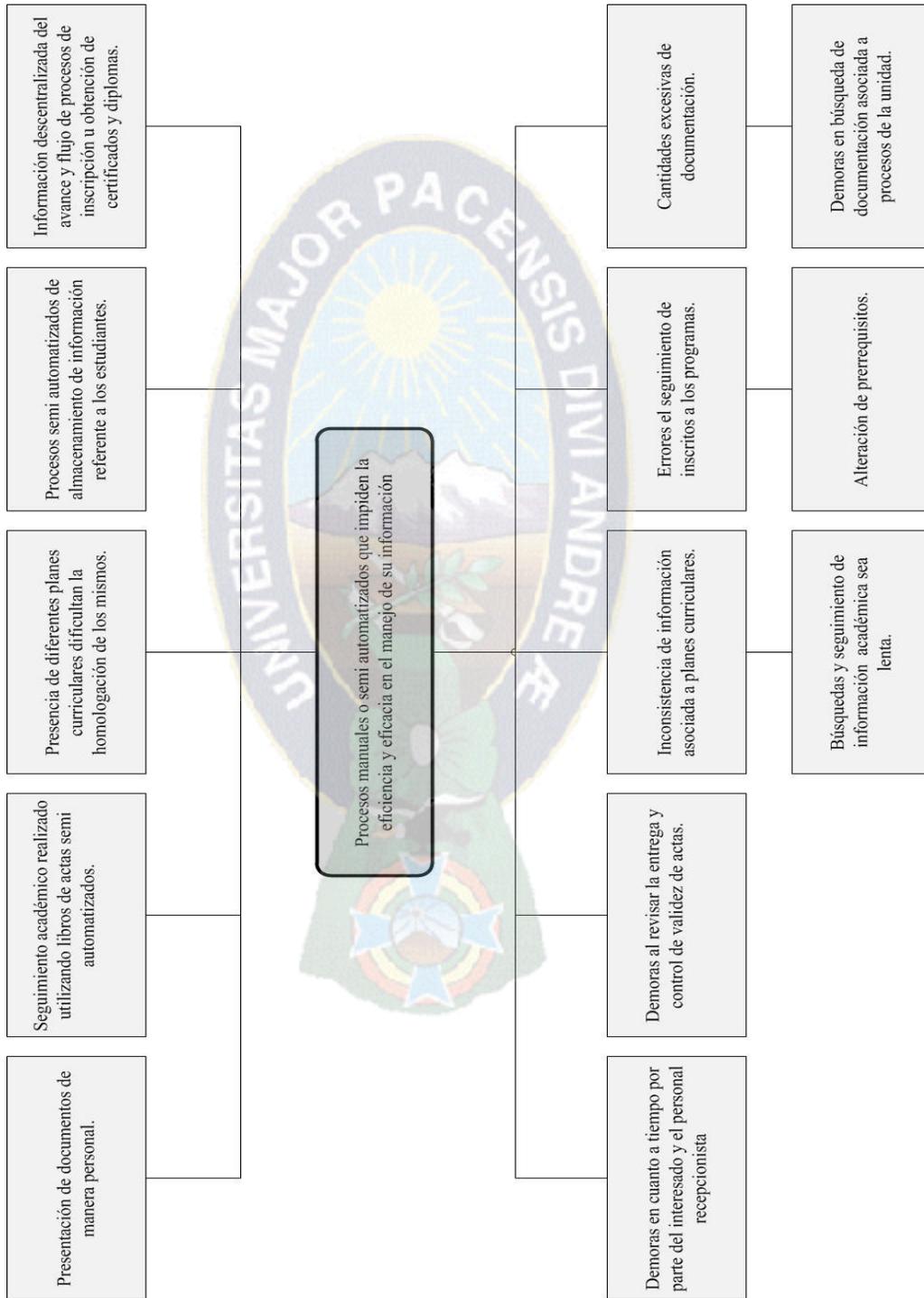
# ANEXOS



# ANEXO A

## ÁRBOL DE PROBLEMAS

## Árbol de Problemas

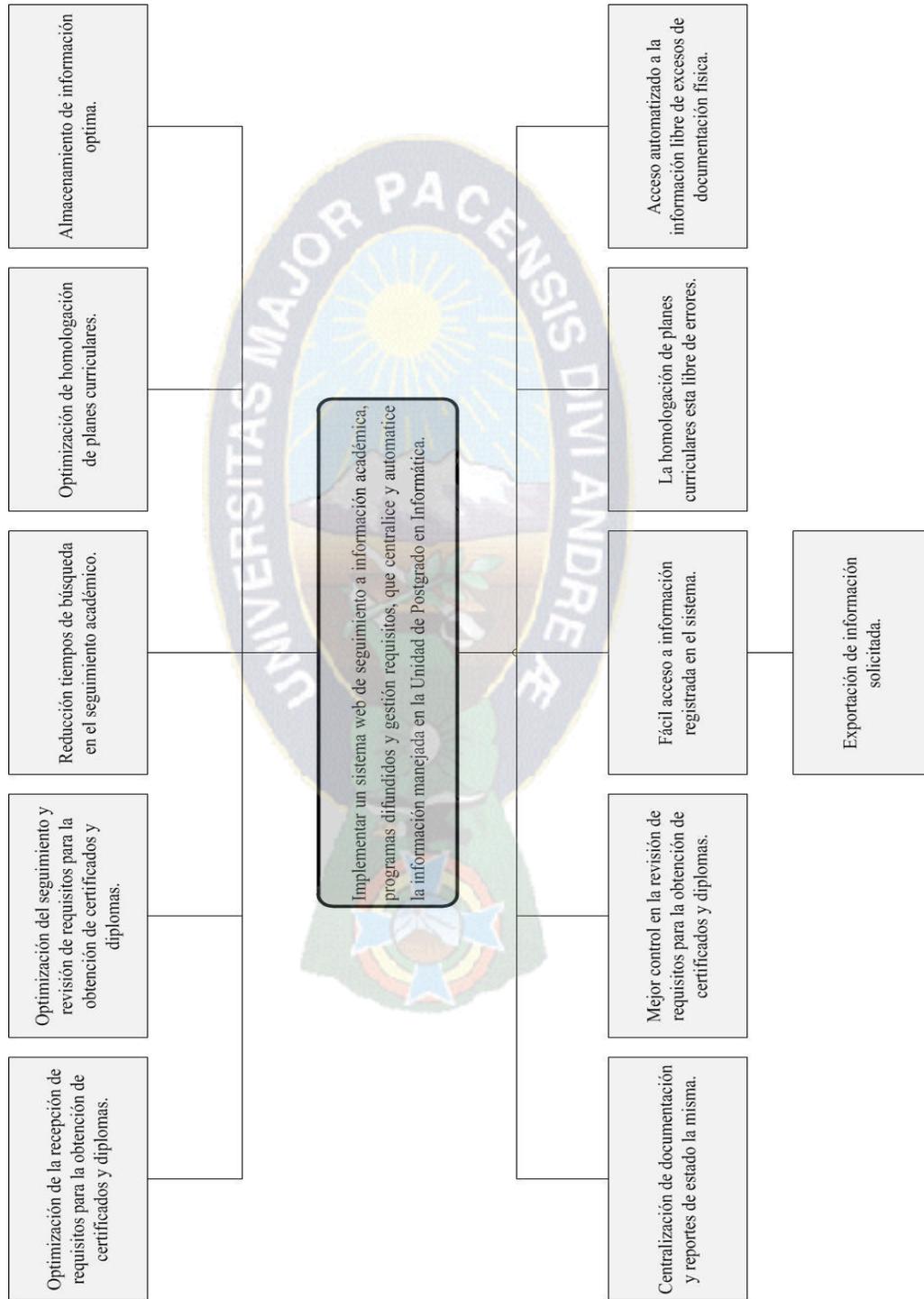




# **ANEXO B**

## **ÁRBOL DE OBJETIVOS**

## Árbol de Objetivos





# **ANEXO C**

## **MATRÍZ DE MARCO LÓGICO**

**Tabla 2. Matriz de Marco Lógico**

| RESUMEN NARRATIVO  | INDICADORES   | MEDIOS DE VERIFICACIÓN   | SUPUESTOS |
|--|---|--|-----------|
| <p><b>FIN</b></p> <p>Aumentar el control en las conexiones y accesos a las bases de datos distribuidas</p> | <p>Monitoreo de acceso al sistema con la implementación del agente.</p> <p>Monitoreo de acceso al sistema sin la implementación del agente.</p> | <p>Comparación de resultados en pruebas del sistema con y sin el agente inteligente.</p> |           |
| <p><b>PROPÓSITO</b></p>  |   |  |           |

|   |  |  |  |
|---|--|--|--|
| <p>Construir un sistema que implemente al agente inteligente para automatizar las normas o métodos de seguridad para bases de datos distribuidas.</p> | <p>Arquitectura del agente.</p> <p>Perceptores y efectores de los agentes.</p>                   | <p>Detalle de la arquitectura del agente.</p> <p>Detalle de los perceptores que recogerán los estímulos del sistema.</p> <p>Detalle de los efectores que interactuarán con el sistema.</p> | <p>Estructura del agente soportada o compatible con el framework.</p> <p>Disponibilidad de datos de acuerdo al cronograma.</p> |
| <p><b>PRODUCTO</b></p> <ul style="list-style-type: none"> <li>• Normas de seguridad que puedan ser</li> </ul>   | <ul style="list-style-type: none"> <li>• Normas o métodos de seguridad que pueden ser</li> </ul> | <p>Número de normas o métodos de</p>   | <p>Ningún cambio en las normas o</p>   |

|   |  |   |   |
|---|--|---|---|
| <p>automatizadas.</p> <ul style="list-style-type: none"> <li>• Información resultante concreta.</li> <li>• Diseño de agente inteligente.</li> </ul> | <p>automatizados.</p> <ul style="list-style-type: none"> <li>• Estructura programática del proceso de automatización de las normas o métodos de seguridad.</li> <li>• Información procesada y necesaria para el desarrollo de la investigación.</li> <li>• Estructura detallada del agente inteligente que se desarrollara.</li> <li>• Cronograma de desarrollo del agente inteligente.</li> </ul> | <p>seguridad a ser automatizados.</p> <p>Orden de las normas o métodos de seguridad por importancia.</p> <p>Cantidad de información recolectada y procesada.</p> <p>Detalle del cronograma y la estructura de los agentes inteligentes.</p> | <p>metodologías de seguridad a ser aplicadas.</p> |
|---|--|---|---|

|   |   |   |   |
|---|---|---|---|
| <p><b>ACTIVIDAD</b></p> <ul style="list-style-type: none"> <li>• Recopilación de Información</li> <li>• Delimitación de la investigación</li> </ul> | <ul style="list-style-type: none"> <li>• Cantidad de información recolectada</li> <li>• Calidad de información recolectada</li> <li>• Definición de límites o temas de estudio</li> </ul> | <p>Número de páginas web, libros, pdf o archivos recolectados acerca de la investigación.</p> <p>Detalle de temas que se tomaran en cuenta para la investigación.</p> | <p>Acceso a internet, bibliotecas y todo lugar o sitio donde se pueda recopilar la información.</p> |
|---|---|---|---|

**Fuente: Propia**