



PRESENTACIÓN PROPUESTA

TESIS DE DOCTORADO: TESIS DE MAESTRÍA:
TRABAJO FINAL DE MAESTRÍA: TRABAJO FINAL DE ESPECIALIZACIÓN

1. PROPOSITOR: Angela Consuelo Checa Hurtado CEDULA: 34.319.575

2. PROGRAMA: Doctorado en Ingeniería - Sistemas Y Computación

3. DIRECTOR PROPOSTO: Jorge Eduardo Ortiz Triviño

DEPARTAMENTO: Ingeniería de Sistemas e Industrial

ASEORES:

TÍTULO: **ANGELA PROPONE=** Una arquitectura de servicios distribuidos basados en agentes para el sistema social inspirado TLÖN

TITULO QUE DEFINIMOS ANTES (Implementación del concepto de servicio distribuido basado en comunidades de agentes en el sistema social inspirado TLÖN)

4. ÁREA: Sistemas de Computación e Informática

LÍNEA DE INVESTIGACIÓN: Computación Aplicada, Sistemas Inteligentes

5. ANTECEDENTES Y JUSTIFICACIÓN: (Indicar los desarrollos previos, circunstancias y condiciones que llevaron a la conclusión de la necesidad y conveniencia del proyecto)

De acuerdo con las tendencias en la investigación de los sistemas de computación distribuida y de las redes de comunicaciones, se identifica que las ramas de las ciencias de la computación más relevantes para el desarrollo de esta tesis son las siguientes: los sistemas distribuidos especialmente la computación en la nube, las arquitecturas de servicios y microservicios, además de los modelos para sistemas de agentes, la social inspiración e inteligencia artificial y las redes inalámbricas.

6.1. Sistemas distribuidos

La computación en la nube es un término para referirse a un modelo donde el almacenamiento de información y el procesamiento se realizan en algún lugar lejano al cliente o usuario, ese lugar es denominado nube [1], que es un centro de datos dedicado, por lo general arrendado por terceros a empresas que no cuentan con infraestructura necesaria para la administración de los recursos informáticos. Una forma de entenderla es como una abstracción de un sistema que tiene sus recursos distribuidos interconectados y que son compartidos para lograr un propósito [2] como prestar servicios especializados. Por tanto, la computación en la nube se refiere a un conjunto de hardware, sistemas, software y aplicaciones que son entregadas al cliente a través de una red como internet.

Dependiendo de la naturaleza privada o pública de la nube que provee los servicios pueden ser de dos tipos: nubes públicas o privadas. En las primeras su uso es para todo público y por lo general son gratis o tienen una tarifa según la naturaleza del servicio solicitado. En las nubes privadas la infraestructura es de propiedad privada y sirve únicamente a los objetivos de una entidad o empresa. También existen las nubes híbridas que serían una combinación de nube privada con nube pública[3].

Algunos de los más conocidos casos de uso de computación en la nube son:

IaaS (*Infrastructure as a Service*) o Infraestructura como servicio: en esta se proveen servicios de alquiler de hardware, o almacenamiento de datos, con acceso directo a hardware virtualizado o dentro de un container. Algunos ejemplos de estos servicios son AWS EC2, OpenStack y AppNexus entre otros.

PaaS (*Platform as a Service*) o Plataforma como servicio: se enfoca en las necesidades cambiantes de los desarrolladores de aplicaciones, en estos los servidores pueden variar en caliente la cantidad de núcleos de CPU, de memoria o almacenamiento incluso ejecutan múltiples sistemas operativos[4]. La escalabilidad

automática de los servicios y balanceo de carga son unas de las ventajas de estos servidores Elásticos. Estos servicios proveen una plataforma que incluye todos los sistemas y ambientes necesarios para todo el ciclo de desarrollo, pruebas, despliegue y hosting, facilitando el desarrollo de aplicaciones empresariales[1]. Algunos referentes de este tipo de servicios son Google AppEngine, Microsoft Azure, Red Hat Makara y Amazon Web Service Elastic Beanstalk.

SaaS (Software as a Service) o software como servicio: se centra en proveer a los usuarios finales acceso a aplicaciones, permitiendo independencia de la infraestructura requerida para la ejecución de sesiones de aplicación ya que el usuario no necesita instalar el software localmente [5]. Algunos ejemplos de proveedores de este tipo incluyen SalesForce.com, NetSuite, Oracle, IBM, y Microsoft con Office 365.

6.2. Arquitecturas de servicios

Un servicio se puede entender como un mecanismo que habilita el acceso a una o más capacidades de negocio donde acceso es proporcionado por unas interfaces bien definidas y es ejercido de acuerdo con unas restricciones y políticas especificadas en la descripción del servicio [6].

Entre las arquitecturas de servicios mas relevantes se encuentran SOA (Service Oriented Architecture) y los microservicios. En el estilo SOA las aplicaciones tienen componentes independientes, con bajo acoplamiento, esos son los servicios, que se comunican entre si mediante protocolos de red. En contraste con las aplicaciones monolíticas donde los componentes están estrechamente acoplados y se comunican mediante llamados a funciones en el mismo espacio de procesos. En SOA la aplicación se divide en servicios más pequeños que aún están acoplados de forma moderada; sin embargo, podrían seguir formando parte de la misma aplicación monolito.

Las tecnologías que impulsaron el uso de SOA fueron HTTP, XML y SOAP que hoy son consideradas lentas y no tan eficientes comparadas con las modernas JSON, REST [7]. Por otra parte, en el estilo de arquitectura de microservicios una aplicación es un conjunto de pequeños servicios, cada uno ejecutando sus propios procesos y comunicaciones mediante mecanismos livianos como HTTP o una API (Application Protocol Interface), estos servicios tienen sus propias capacidades de negocio y se despliegan de forma automática e independiente. Estos requieren una mínima o nula administración centralizada, pueden ser escritos en diferentes lenguajes de programación y usar diferentes tecnologías de almacenamiento. La Figura 1 ilustra la diferencia entre el clásico estilo monolítico y las arquitecturas de servicios en términos de granularidad.

En Las aplicaciones monolíticas tanto la base de datos como la lógica de negocio y la interfaz son indivisibles, dependientes entre sí y se encuentran en un mismo espacio (Ver Figura 2a). En SOA los servicios se acceden mediante las interfaces publicadas usando una capa middleware, y en algunos abordajes esa capa se reemplaza por un bus de servicios equivalente denominado ESB (Enterprise Service Bus). Este sirve para proveer un mayor grado de interoperabilidad y reduce el número de canales de comunicación necesarios. Ese bus maneja los protocolos de red y los mensajes de enruteamiento. A pesar de eso, sus mecanismos de despliegue son largos y complejos (Ver Figura 2c). En los microservicios en cambio, la aplicación se descompone en muchos servicios (procesos) independientes con mecanismos que permiten a cada uno desarrollarse y desplegarse de forma rápida y continua sin afectarse entre sí directamente.

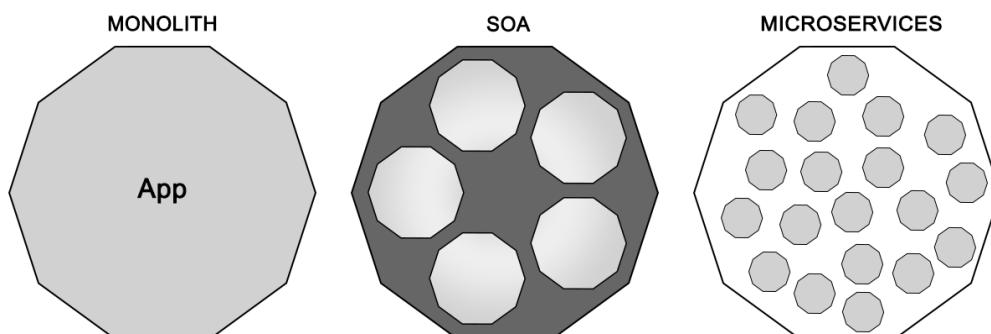


Figura 1 Comparación entre arquitecturas en términos de granularidad

También puede existir una la API Gateway que es la responsable de recibir las peticiones del cliente y enrutarlas hacia el conjunto de servicios correcto, haciendo conversión de protocolos si es necesario, construyendo las respuestas y retonándolas al cliente o capa de presentación[8] (Ver figura 2c).

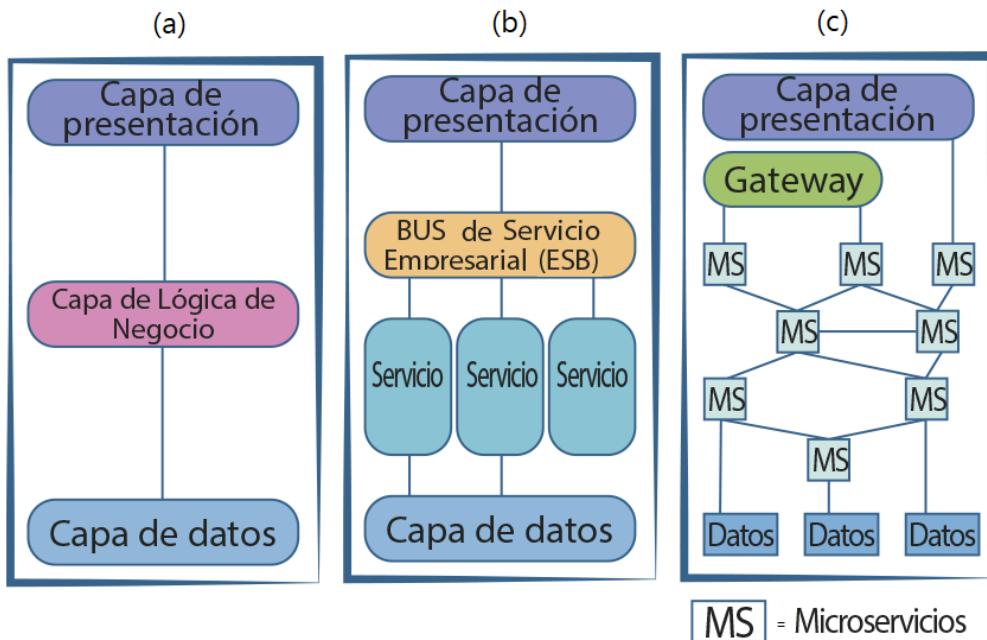


Figura 2 Representación de arquitecturas monolítica (a), orientada a servicios (b) y microservicios (c)

Los microservicios proveen las ventajas de escalabilidad y tolerancia a fallas, derivadas de las siguientes características:

Tamaño y enfoque: cada microservicio sirve a un único propósito, por ejemplo, un sistema distribuido puede contener un microservicio para cada funcionalidad, uno para cuentas, otro para pedidos, otro para manejo de inventario etc. Eso facilita la escalabilidad de la aplicación y el rastreo de cambios o adición de nuevas funcionalidades de forma rápida.

Independencia: Debido que están separados, cada uno es autónomo y ejecuta sus propios procesos, pro lo general en espacios virtuales o contenedores separados. Se pueden desarrollar o programar de forma independiente y por diferentes grupos de persona. Pueden ser desplegados de forma separada en los ambientes de producción De esta forma cuando falla uno de los microservicios no se afecta todo el sistema y las fallas pueden ser localizados y administrados fácilmente.

Adicionalmente otra ventaja es que son **reusables** gracias que múltiples microservicios actúan en conjunto para construir una aplicación grande, estos servicios pueden ser reutilizados en diferentes aplicaciones y en diferentes áreas de la misma aplicación. Por su pequeño tamaño pueden ser fácilmente reemplazados, por ejemplo, si se actualiza a una nueva tecnología, estas mejoras pueden ser efectuadas sin tener que reemplazar todos los componentes o funcionalidades de la aplicación [7].

6.3. inteligencia artificial

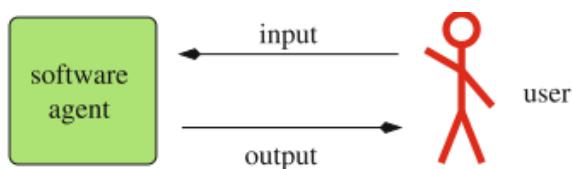
La tecnología asociada a los agentes ha evolucionado a partir de dos áreas de investigación: la Inteligencia Artificial (IA) y la computación distribuida [9]. En consecuencia, se aborda el tema IA porque de ésta se derivan importantes conceptos y tecnologías que pueden proveer a los sistemas como distribuidos como computación en la nube o el Internet de las cosas (IoT) de una conciencia de usuario, de ambiente y conciencia social que habilitan un amplio rango de aplicaciones innovadoras, además de proporcionar capacidades de autonomía y comportamiento propias [10]. Capacidades que se correlacionan con los objetivos buscados en sistemas de cómputo autónomos distribuidos, heterogéneos y altamente dinámicos.

La IA surgió a partir de la sinergia de diversas ciencias y áreas como la filosofía, matemáticas, informática, cibernetica, lingüística, economía, psicología, neurociencia y biología entre otras[11]. En la actualidad la Inteligencia Artificial es un término que se refiere a un conjunto de técnicas, algoritmos y paradigmas de cómputo cuyo propósito es hacer que los sistemas se comporten como si fueran inteligentes, aunque en realidad se considera que “simulan” la inteligencia.

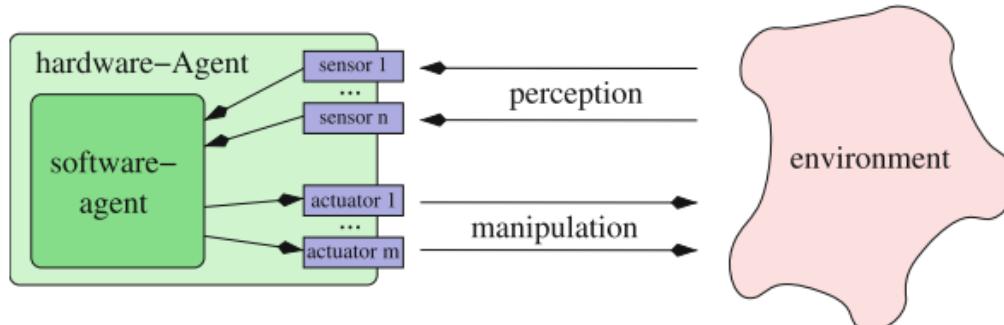
Teniendo en cuenta que los sistemas inteligentes operarán en entornos de altas demandas, deben enfrentar problemas no triviales y complejos [10] y requieren de mecanismos para hacer uso efectivo de los recursos disponibles para proveer servicios de forma exitosa, conforme a esto los agentes inteligentes son considerados como una tecnología apropiada y prometedora para hacer frente a esos desafíos [12]. En contraste con esto, el uso de estos agentes inteligentes se ha incrementado en diversas áreas incluyendo las tecnologías de interconexión y computación móvil para lograr comportamientos dinámicos, alta escalabilidad, auto recuperación y reconfiguración de redes.

6.3.1. Agentes

De forma general el término agente denota sistemas que procesan información y producen una salida. Estos pueden ser clasificados en diferentes formas. Una de ellas los clasifica en agentes software y agentes hardware [13]. Los agentes software consisten en programas que calculan un resultado a partir de una entrada, por ejemplo, recibe entradas de teclado, archivos o paquetes de red como si fueran entradas y actúan en el ambiente desplegando una salida en la pantalla, escribiendo archivos o enviando paquetes de red (Ver Figura 3a). Los agentes hardware o (robos autónomos) pueden ser vistos como entidades que perciben su entorno a través de sensores y mediante actuadores llevar a cabo acciones y cambios en su ambiente (Ver Figura 3b).



(a) Agente software con interacción de usuario



(b) Agente hardware y su interacción con el ambiente

Figura 3 (a) Agente software y (b) agente hardware. Modificado de [13]

Cuando se habla de percepción se refiere las entradas percibida por el agente en un instante dado. una secuencia de percepción de un agente es el historial completo de todo lo que el agente ha percibido. Por tanto, la elección de una acción en un instante dado depende enteramente de las secuencias de percepciones observada hasta el momento y no en algo que nunca haya percibido. Se dice que el comportamiento de un agente se describe por una función agente que mapea cualquier secuencia de percepción a una acción. Esta función es una abstracción matemática, pero el programa agente es una implementación concreta de esa función agente, que se ejecuta dentro de un sistema físico [11].

6.3.2. Sistemas multi-agente (MAS)

En los sistemas inteligentes cuando una tarea u objetivo se realiza mediante muchos agentes de forma coordinada se convierte en un sistema multi-agente. En este contexto, los sistemas multi-agente (MAS por las siglas en inglés para Multi Agent System) son plataformas software donde los agentes autónomos e inteligentes interactúan unos con otros para lograr un objetivo computacional[14]. Los MAS han sido usados en un gran rango de aplicaciones, las más comunes son el análisis de imágenes, procesamiento de lenguaje natural y sistemas robóticos. Recientemente han sido considerados en aplicaciones de blockchain, sistemas de recomendación, realidad aumentada y vehículos eléctricos [15].

En lo que respecta a las redes de comunicaciones sus usos más conocidos son en la gestión de redes cableadas y en predicciones al servicio de los protocolos de enruteamiento. Sin embargo, gracias a su potencial para solucionar problemas complejos, también se están considerando como herramienta para abordar diferentes retos de la computación en la nube. Por ejemplo, en De la Prieta, Rodríguez-González, Chamoso, Corchado y otros (2019) se encuentra un detallado estudio donde clasifican las aplicaciones que combinan los MAS con la computación en la nube, también describen cómo los sistemas multi-agente se desempeñan en este tipo de ambientes distribuidos e identifican un conjunto de campos de aplicación en los que los MAS pueden contribuir al desarrollo de los sistemas de cómputo en la nube. Estos campos a su vez ordenados a en categorías de monitoreo, configuración, seguridad y privacidad, asignación de recursos, negociación de los acuerdos de nivel de servicios, composición de servicios y auditoría de la calidad del servicio [16].

Por otra parte, en Sidhanta y Mukhopadhyay (2012) se presenta una propuesta de ambiente de ejecución que puede proveer el poder de cómputo de la nube a los sistemas multi-agente en redes conectadas de forma intermitente o redes peer-to-peer [14]. Otro abordaje que se encuentra en Al-Sayed, Hassan y Omara (2020) donde se propone una ontología de nube basada en una plataforma multi agente para descubrir los servicios de nube, debido a que se ha identificado una carencia de una descripción estandarizada para la publicación de servicios [17], cada vez más numerosos y diversos, que al no contar con descriptores complica cada vez más el proceso de descubrimiento de servicios de nube. Otros autores como en Gaglio y Lo Re (2014) habían definido desde hace algunos años estas aproximaciones que proveen soluciones basadas en agentes para mejorar los recursos y servicios de nube como una nueva disciplina llamada *Agent-based Cloud Computing* (computación de nube basada en agentes) [18], bajo la promesa que recae sobre los agentes autónomos de hacer que las nubes sean más inteligentes en su interacción con los usuarios y más eficientes en la asignación de recursos de procesamiento y almacenamiento a las aplicaciones.

Para concluir esta sección, es importante resaltar que los sistemas modernos altamente distribuidos revelan una demanda comportamientos grupales de los agentes, por tanto, es deseable que éstos actúen como si hicieran parte de una comunidad, es por esa razón que entre los temas de investigación se considera importante dirigir la mirada hacia los modelos social inspirados que puedan aportar con artefactos para construcción de un sistema de comunidades de agentes para servicios de nube sobre redes inestables como las redes inalámbricas. La social inspiración tiene relevancia porque de ella podrían derivarse propuestas de arquitecturas para ese tipo de sistemas. Por consiguiente, las arquitecturas para la construcción de un agente deberían incluir modelos para sí mismo y para otros agentes, dicho de otra forma, cualquier componente con razonamiento debería incluir mecanismos para razonar acerca de otros y de su interacción con ellos [19]. Todas esas capacidades deseadas requerirán la implementación de comportamientos sociales. Por ejemplo, de algoritmos de colaboración para conformación de equipos o alianzas y de comportamientos estratégicos en escenarios competitivos con recursos restringidos, en esta dirección es donde convergen las áreas temáticas de inteligencia artificial, computación en la nube y la social inspiración.

6.4. Redes inalámbricas

Las redes de comunicaciones que usan el aire como medios de transmisión para llegar a los usuarios son las denominadas redes inalámbricas, estas en comparación con las redes cableadas presentan desafíos para su funcionamiento y despliegue, por ejemplo, la seguridad, eficiencia energética y el acceso al espectro electromagnético. Una de las aplicaciones más populares es la telefonía móvil celular. A lo largo de las últimas décadas las tecnologías inalámbricas se han extendido desde los teléfonos inalámbricos, controles remotos, micrófonos inalámbricos hasta las modernas aplicaciones del internet de las cosas e internet inalámbrico de alta velocidad, entre otras. De forma que esta tecnología se ha instalado en un gran número de áreas de aplicación.

Las redes inalámbricas que conciernen a esta propuesta son las redes WiFi y las redes ad hoc. Las redes WiFi se refieren a una tecnología que permite a los dispositivos intercambiar datos o conectarse a través de señales de comunicación inalámbrica usando las bandas de frecuencia de 2,4 Ghz o 5Ghz [20]. Sus características y normas para su implementación se definen en el estándar IEEE 802.11 [21]. La principal y más evidente ventaja de estas redes es la movilidad, porque el usuario puede trasladarse libremente mientras accede a la red sin cables que lo limiten. Otra ventaja es la flexibilidad porque se pueden desplegar más rápidamente que las redes cableadas, de modo que se pueden aumentar, disminuir sin requerir grandes cambios en la infraestructura de edificios o causar sobrecargas en la infraestructura existente [22], esto es una ventaja principalmente para los operadores, porque disminuyen los costos de despliegue.

Aprovechando el avance en la computación en la nube y el auge de las soluciones de virtualización y containerización, ha emergido el enfoque de las Redes Definidas por Software (SDN) como una nueva propuesta para tratar de solucionar los desafíos de las redes inalámbricas. La principal motivación surge desde la naturaleza propietaria de los controladores usados por las arquitecturas de redes inalámbricas actuales. Estos controladores solo funcionan en dispositivos compatibles y dependen de interfaces propietarias [23]. Entonces, SDN aparece como una alternativa para implementar redes construyendo sus componentes de forma virtual mediante piezas de software. En esa área se identifica una oportunidad potencial para desarrollar proyectos de investigación como Ethanol descrito en artículo de Moura et al. (2020) donde se presenta una arquitectura de red definida por software para las redes inalámbricas de área local 802.11, o como el estudio de Moorthy, Venkataraman y Rama Rao (2020) donde analizan las amenazas de seguridad y posibles ataques a la privacidad de los datos en las nubes móviles definidas por software [24].

6.4.1 Redes ad hoc



Ad hoc: Es una locución latina que se refiere a una solución específica elaborada para un problema particular y por lo tanto no es generalizable ni utilizable para otros propósitos. Es usada para referirse a algo que es adecuado para un fin determinado o para una situación particular. Por consiguiente, una red ad hoc es un tipo de red que se arma para un propósito específico, es un sistema distribuido de comunicaciones autoorganizado y su estructura es creada y mantenida de forma dinámica [25], es por eso que también se conocen como redes sin infraestructura o *infrastructure-less networks*. Las Redes Ad hoc se establecen de forma temporal para ofrecer servicios o aplicaciones específicas, especialmente donde una infraestructura de red es imposible de establecer en un tiempo corto o es muy costosa. De forma ideal se desea que las redes ad hoc presenten las siguientes características [26]:

- **Topología Dinámica:** Al no tener una infraestructura definida, los nodos no están atados un medio físico y pueden moverse dentro de la topología de forma aleatoria e impredecible, pueden ingresar a la red o retirarse en cualquier momento. De esa cualidad surgen los desafíos en cuanto a adaptabilidad que le permitirían acomodarse rápidamente a los cambios en la red.
- **Multisaltos:** La información se transmite a través de un camino de múltiples saltos entre la fuente y el destino, por lo tanto, cada nodo actúa simultáneamente como cliente o servidor de la red, es decir, debe realizar operaciones de enrutamiento de la información y operaciones de nodo destino. La característica de dotar a los nodos de capacidades de enrutamiento es muy importante en las redes ad hoc porque de esta emerge el comportamiento colaborativo de los nodos que la componen.
- **Auto-organización:** En las redes ad-hoc los nodos deben determinar de manera autónoma sus parámetros de operación, es decir son capaces de auto-configurarse de acuerdo a la situación de la red en un momento dado. No requieren un punto de acceso (Access Point) que coordine su funcionamiento, entonces su organización es independiente de un control central.
- **Auto-recuperación:** Estas redes son capaces de auto-reparación de los defectos encontrados en sí misma, por ejemplo: si un nodo falla, la red debe ser capaz de ejecutar acciones necesarias para mantener la conectividad.

Todos los aspectos mencionados anteriormente representan retos de investigación para lograr un despliegue eficiente de este tipo de redes. Los principales desafíos se derivan de la naturaleza de las comunicaciones inalámbricas. Por otra parte, al permitir que los nodos se conecten o desconecten de la red, en este sentido la red se enfrenta al problema de soportar un alto dinamismo. Al igual que la seguridad de la red, porque puede ser propensa a ataques, que pueden inhabilitar algunos nodos e incluso toda la red. Por último, uno de los más grandes problemas es que los nodos o dispositivos móviles que la conforman por lo general son dispositivos con unos limitados recursos de memoria, capacidad de procesamiento y almacenamiento y restricciones en la velocidad y rango de transmisión, eso sumado a la dependencia de las

baterías que fácilmente agotan el suministro de energía al dispositivo cuando este realiza esfuerzos de computación.

En el desarrollo de los protocolos para las redes ad hoc se debe considerar su dinamismo en la topología de red junto con las restricciones de energía, movilidad, la incorporación de nuevos nodos si se quiere un lograr un diseño de red con comunicaciones robustas [20].

6.5. Sistema social inspirado TLÖN

El sistema TLÖN es una propuesta de sistema computacional distribuido, que se inspira en los modelos sociales humanos, para abstraer conceptos y comportamientos que pretenden ser implementados de forma artificial en todos los componentes de sistema. Se considera que la social inspiración conduciría a una solución que haga posible definir las características deseadas del todo y cada una de sus partes, y encaminaría hacia la implementación de una “pseudo sociedad ideal” en un ambiente computacional como las redes ad hoc inalámbricas, que sea capaz de manejar efectivamente problemas como ubicuidad, movilidad, dinamismo, comportamiento estocástico, coordinación, virtualización y auto-configuración.

La construcción del sistema TLÖN es el proyecto principal del grupo de investigación en Redes de Telecomunicaciones Dinámicas y Lenguajes de Programación Distribuidos (TLÖN). La forma en que se explica este sistema es mediante un modelo de capas como el de la Figura 4, planteado de manera que estas capas cumplan tareas análogas a la estructura de un estado, constituyendo un *pseudo-estado* TLÖN. En este el territorio es equivalente a la red ad hoc o ambiente de red, las instituciones estarían conformadas por el módulo de virtualización, las personas o actores de la sociedad serían los agentes que ejecutan tareas o prestan servicios y, al igual que en el plano real, se pueden agrupar de diferentes formas para conformar una comunidad de agentes. Por último, en el nivel más alto se encuentra una capa de aplicaciones específicas del sistema de cómputo que se soportan en las capas inferiores. En TLÖN las aplicaciones podrían entenderse como implementaciones de un nivel más alto de abstracción de la sociedad, los conceptos de cultura, justicia, costumbres o incluso la economía.

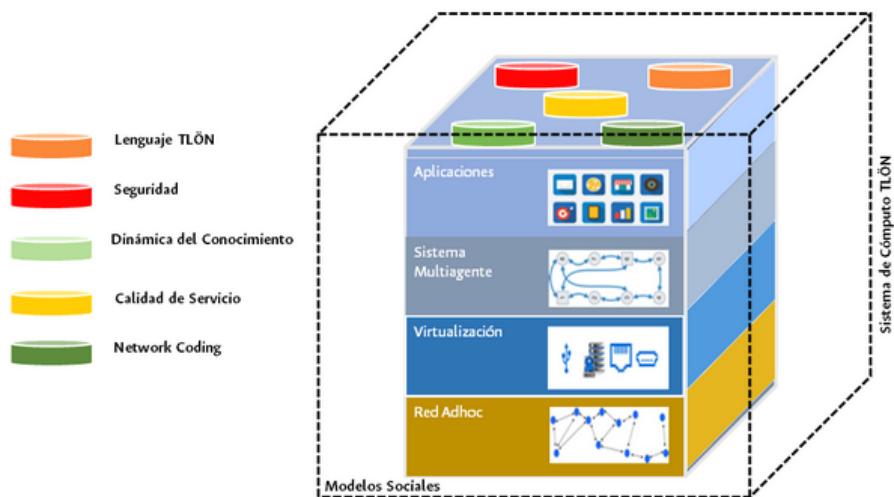


Figura 4 Modelo de capas del Sistema TLÖN

Dicho de otra forma, para TLÖN el objetivo es encontrar una forma de implementar artificialmente los modelos sociales en un sistema de cómputo distribuido, con infraestructura dinámica, que permita la prestación de múltiples servicios especializados de computación distribuida sin ser afectados por la característica ad hoc de sus nodos, tampoco por sus recursos limitados de procesamiento o almacenamiento ni por las características heterogéneas de los mismos.

Este enfoque social inspirado tiene precedentes como el descrito en Demazeau, An y Bajo (2018) en donde se presenta un modelo de sistema multi-agente organizado como una sociedad de agentes con diferentes roles usando un tipo de agentes agregadores que tienen la tarea de conectar a los otros agentes entre si conformando una red. En esa capa multi-agente se realizan interacciones entre agentes, de esas emerge la

organización en sociedades, ahí los agentes se influencian unos a otros y dependiendo de la frecuencia de sus interacciones podrán tener valores para representar sus fortalezas asignadas a medida que el tiempo pasa. Partiendo de eso el objetivo del sistema es analizar información adquirida dinámicamente de portales y exponer una representación de los eventos que han tenido lugar en cada portal [27]. En consecuencia, es necesario que los agentes cooperen de forma constructiva para lograr el objetivo global de recolección de información de diferentes sitios.

La característica de cooperación puede ser brindada por las capacidades de razonamiento individuales, es por esto que en TLÖN se intenta modelar el comportamiento los agentes a partir de las concepciones de existencia y esencia de Jean Paul Sartre, apropiando el modelo sartreano del ser [28], que enmarca tres dimensiones del individuo: el *En sí*, que se refiere a lo que soy (pasado y presente), el *Para sí*, en cuanto a lo que deseo ser en el futuro y el *Para el otro*, que representa el ser para el prójimo.

6.6. Desarrollos previos

Entre Los trabajos de investigación que han dado forma al actual estado del arte del sistema social inspirado TLÖN se encuentran: Latorre (2015) donde se realizó la implementación de un clúster sobre una red ad hoc y el análisis de la comunicación entre nodos [29]. Zárate (2016) que presenta un diseño de la primera versión del Sistema Operativo Orientado a Redes Ad hoc (S.O.V.O.R.A) y describe la implementación del orquestador de TLÖN [30], encargado de iniciar la red, gestionar recursos de cómputo y ser la base para la prestación de servicios dentro de la misma. Álvarez Rodríguez (2017) diseñó e implementó un modelo de agentes móviles TLÖN en una red ad hoc, con la capacidad de desplegarse y realizar tareas de acuerdo con roles definidos [31]. En Molina (2019) se construyó un módulo de seguridad informática que permite dar características de disponibilidad al sistema [32]. Por otra parte, respecto a la social inspiración se ha avanzado mediante estudios como el de Ospina (2020) que presentó un modelo de justicia computacional para la distribución de recursos en redes ad hoc y se logró mediante la formulación de un método de distribución utilizando juegos estocásticos, instituciones, justicia distributiva y computación adaptativa [33]. Los experimentos de simulación demostraron que es posible resolver el problema de la distribución en sistemas auto-organizantes abiertos usando una combinación de computación social inspirada y modelamiento basado en agentes.

6.7. Motivación

Uno de los factores que motiva la realización de esta investigación es que implicará esfuerzos en el estudio de las tecnologías para sistemas de cómputo distribuido, el internet de cosas, conceptos como Edge computing o Computación de Frontera, y las diferentes categorías de servicios de nube como Infraestructura como servicio, Contenedor como servicio, Plataforma como servicio, Funciones como servicio y Software como servicio para aplicarlos al ambiente dinámico de las redes inalámbricas heterogéneas. En consecuencia, también requiere el estudio de tecnologías de redes inalámbricas como WiFi y estándares para comunicaciones en IoT. Por otra parte, es necesario articular los conocimientos sobre sistemas multi-agente social inspirados. Todos los anteriores son temas de interés en los que trabajan comunidades de científicos de la computación. En esta perspectiva, se considera que este proyecto representa el reto de contribuir con una propuesta de arquitectura que permita avanzar en la implementación de servicios soportados en la capa de agentes del sistema TLÖN.

6.8. Identificación de necesidades

Para la construcción del sistema TLÖN es necesario abordar todas las capas especificadas en su diseño básico. Una de las capas que requiere mayor atención es la capa de sistema multi-agente en donde se conformaran las instituciones de TLÖN, que a su vez se supone posibilitará el despliegue aplicaciones para la capa más alta de TLÖN; poco explorada hasta el momento. Por tanto, una de las necesidades identificadas en el sistema es la implementación de comunidades de agentes que tengan comportamientos pseudo sociales. A partir de esto podemos decir que las comunidades de agentes deben tener un objetivo, una “razón de existencia” y para el sistema TLÖN esa razón sería la prestación de servicios. En consecuencia, la definición de las categorías de esos servicios para el sistema TLÖN se convierte en un requisito previo a su construcción. Por otra parte, también es importante contribuir con la clasificación de los posibles servicios distribuidos para TLÖN. Ante las anteriores necesidades, este proyecto pretende aportar con nuevo conocimiento y dar una solución para la capa 3 del sistema. Además, pretende definir nuevos

requerimientos en cuanto a servicios y aplicaciones para sistemas altamente distribuidos que usan redes inalámbricas.

6. IDENTIFICACIÓN DEL PROBLEMA:

El eje principal de problema es encontrar una forma para implementar servicios distribuidos en ambientes dinámicos como nubles móviles o redes IoT. En esta dirección se plantea la siguiente pregunta de investigación ¿Cómo implementar comunidades de agentes social inspirados sobre sistemas distribuidos altamente dinámicos?, que se convierte en el punto de partida hacia la búsqueda de una solución basada en modelos social inspirados que permita construir comunidades de agentes que sean capaces de comportarse como comunidad para proveer servicios de computación distribuida y servicios de nube inteligentes sobre redes ad hoc inalámbricas y que permitan posteriormente el despliegue de aplicaciones distribuidas en el sistema TLÖN.

7. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS:

Objetivo general:

Implementar el concepto de servicio distribuido basado en comunidades de agentes en el sistema social inspirado TLÖN.

Objetivos específicos:

1. Determinar las categorías de servicios para el sistema social inspirado TLÖN
2. Diseñar la estructura básica de un servicio distribuido para el sistema social inspirado TLÖN
3. Construir un ambiente para soportar la institucionalidad basada en comunidades de agentes para un sistema social inspirado.
4. Diseñar un sistema multi agentes con comportamiento social inspirado
5. Implementar la comunidad de agentes para la prestación de un servicio distribuido
6. Implementar dos escenarios de validación de los servicios distribuidos social inspirados

8. METODOLOGÍA:

Para este trabajo se seguirá los lineamientos de la Metodología de Investigación en Diseño (DRM por su nombre en inglés *Design Research Methodology*). Esta metodología de investigación fue propuesta por la profesora Lucienne Blessing de la Facultad de ciencias, tecnología y comunicación de la Universidad Luxemburgo, se orienta a enlazar las preguntas de investigación y brinda herramientas necesarias para abordar dichas preguntas de manera sistemática.

Está basada en la premisa de que el diseño es una actividad compleja que involucra artefactos, personas, herramientas, procesos, organizaciones y un entorno en el cual se desarrolla, por eso la metodología de investigación es muy importante para incrementar el entendimiento de los complejos fenómenos que intervienen en la tarea del diseño. Está orientado especialmente a ayudar en la investigación en ingeniería y diseño industrial.

Esta metodología se basa en cuatro etapas, a cada una se asignan unos objetivos y unos entregables. La Figura 5 muestra un resumen general de las etapas y el ciclo de ejecución.

Explicación de la Investigación (Research Clarification RC): pretende ayudar a los investigadores a formular un plan completo, claro y realista. En esta etapa se debe identificar los objetivos que se espera cumplir, enfocándose en los principales problemas de investigación, las preguntas y las hipótesis. También se identifican las áreas en las que se espera contribuir. Luego se debe realizar un modelo de impacto y modelo de referencia iniciales. Por otra parte, aquí también se determina el conjunto de criterios que miden el éxito y los factores que contribuirían o afectarían el logro de los objetivos o el fracaso.

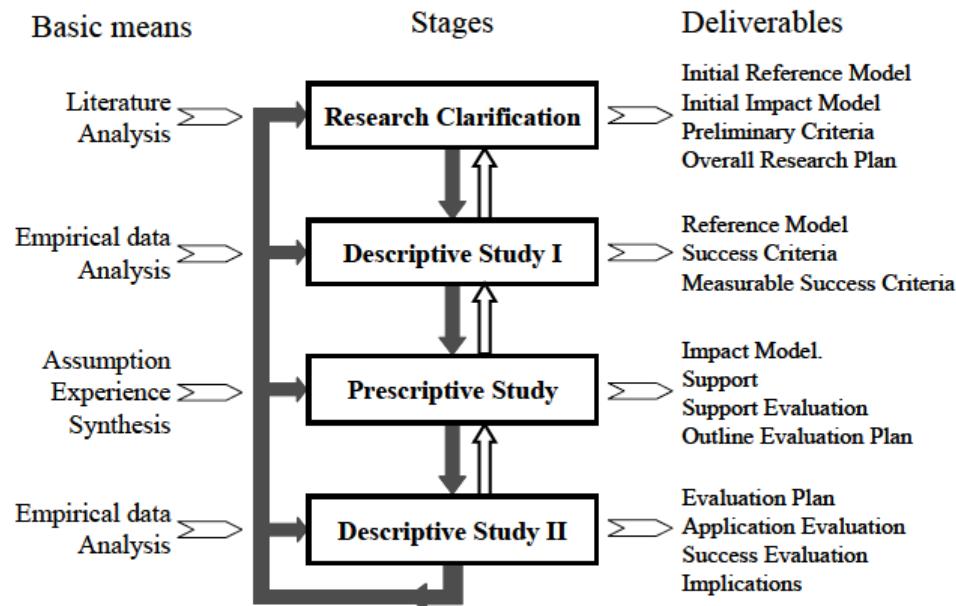


Figura 5 Etapas de DRM, sus recursos básicos y entregables. Modificado de [34]

Estudio Descriptivo I (Descriptive Study I DS-I): en esta etapa se parte del Modelo de referencia inicial creado en la etapa RC. La meta es incrementar el entendimiento del diseño y sus factores de éxito mediante la investigación del fenómeno de diseño, realizar revisión de la literatura de investigaciones empíricas, llevar a cabo una investigación científica y realizar ejercicios de razonamiento. El punto de partida es el Modelo de Referencia inicial y el estudio preliminar de criterios creado durante la etapa RC y en esta etapa deben completarse añadiendo aclaraciones sobre la línea principal de argumentación y se deben sugerir factores que conlleven al mejoramiento de la situación existente. Como resultado se debe entregar una actualización del modelo de Impactos iniciales.

Estudio Prescriptivo (Prescriptive Study PS): en esta etapa el objetivo es desarrollar el soporte previsto, que responda al qué es y cómo funciona. Que aborde los factores clave de forma sistemática, y descubrir que un gran nivel de detalle de la evaluación de los efectos puede tener lugar a partir de los criterios de éxito medibles. Además, puede ayudar a determinar la forma cómo proceder en la futura etapa DS-II mediante el desarrollo de un esquema del plan de evaluación que se usará como punto de partida en la etapa DS-II. Uno de los entregables más importantes es el modelo de las repercusiones previstas. Y por otro lado la documentación de apoyo actual.

Estudio Descriptivo II (Descriptive Study II DS-II): esta se enfoca en la evaluación de apoyo, los objetivos son entender el impacto de un apoyo o soporte. Identificar si puede ser usado para la tarea para la cual fue diseñado y si tiene el efecto esperado. Si contribuye a la evaluación de los éxitos o aciertos. Si el impacto esperado es representado en el modelo de impactos. Asimismo identificar las mejoras necesarias a realizar en el concepto, elaboración, introducción y contexto de los soportes. Por último, también pretende evaluar las hipótesis detrás de la situación actual representada en el modelo de referencia y si también está representada la situación ideal en el modelo de impactos. Como entregables se debe cumplir con los resultados de la evaluación de la aplicación, de la evaluación del éxito y las implicaciones y sugerencias para los entregables obtenidos hasta el momento en el transcurso de todas las etapas.

Las ventajas de usar DRM para realizar este proyecto pueden sintetizarse en que proporciona:

- Un marco de referencia para el diseño de una investigación,
- Ayuda para identificar áreas de investigación y desarrollar la argumentación necesaria,
- Lineamientos para la planeación de la investigación,
- Lineamientos para una investigación rigurosa y definición de un contexto,
- Ayuda para seleccionar métodos de investigación y
- Estrategias para posicionar proyectos y programas de investigación.

9. ACTIVIDADES:

Actividad	Producto
A.1 Revisión Literaria	Estado del arte servicios distribuidos y agentes social inspirados
A.2 Establecer los requerimientos del diseño de servicios para el sistema TLÖN	Documento captura y análisis de requerimientos
A.3 Definición de las categorías de servicios para un sistema distribuido social inspirado	Documento de descripción de categorías de servicios
A.4 Diseño de un servicio distribuido para el sistema.	Documento descriptivo del servicio distribuido para el sistema TLÖN
A.4 Definición de los requerimientos para la construcción de un ambiente	Documento de captura y análisis de requerimientos
A.5 Definición de los requerimientos para la construcción de una comunidad de agentes	Documento de captura y análisis de requerimientos
A.6 Diseño de un ambiente que soporta institucionalidad basado en agentes social inspirados.	Documento de diseño del prototipo inicial de un ambiente para agentes social inspirados
A.7 Diseño de una comunidad de agentes social inspirados para el sistema distribuido TLÖN	Documento descriptivo
A.8 Implementación de un ambiente para agentes social inspirados	Código fuente y pruebas de exploración
A.9 Diseño de escenarios de prueba de servicios distribuidos con agentes social inspirados	Documento descriptivo
A.10 Implementación de los servicios distribuidos del sistema TLON con agentes social inspirados.	Pruebas de exploración y código fuente
A.11 Desarrollo de pruebas y documentación	Documento descriptivo, de análisis cualitativo y cuantitativo
A.12 Implementación de escenarios de prueba	Pruebas de exploración y código fuente
A.13 Documentación detallada	Tesis

10. CRONOGRAMA:

Actividad	Año			
	1	2	3	4
Iteraciones OE 1				
A1				
A2				
A3				
Iteraciones OE 2				
Diseño del Modelo de Referencia				
A4				
A5				
A6				
A7				

Iteraciones OE 3									
Modelo de Impacto									
A8									
A9									
Modelo de Evaluación									
Iteraciones OE 4									
A10									
A11									
A12									
Modelo de Evaluación									
A13									
Elaboración de informe parcial									
Entrega de informe parcial									
Elaboración de informe final									
Entrega de informe final									

11. BIBLIOGRAFÍA BÁSICA:

- [1] N. Antonopoulos and L. Gillam, *Cloud Computing*, vol. 17, no. 1. Cham: Springer International Publishing, 2017.
- [2] F. Fitzek and M. Katz, *Mobile Clouds*. 2014.
- [3] B. Mahalakshmi and G. Suseendran, *An Analysis of Cloud Computing Issues on Data Integrity, Privacy and Its Current Solutions*, vol. 839, no. January. Springer Singapore, 2019.
- [4] N. K. Sehgal and J. M. Bhatt, Pramod Chandra P., Acken, *Cloud Computing with Security*, 2nd ed. Springer International Publishing, 2020.
- [5] S. Murugesan and I. Bojanova, Eds., *Encyclopedia of Cloud Computing*. 2016.
- [6] M. Richards, *Microservices vs. Service- Oriented Architecture*. O'Reilly Media, Inc., 2016.
- [7] M. Krämer, "A Microservice Architecture for the Processing of Large Geospatial Data in the Cloud," Technische Universität Darmstadt, 2018.
- [8] J. Bonér, *Reactive Microservices ArchitectureDesign Principles for Distributed Systems*. 2016.
- [9] J. Cao and S. K. Das, *Mobile Agents and Applications in Networking and Distributed Computing*. 2012.
- [10] W. Lawless, R. Mittu, D. Sofge, I. S. Moskowitz, and S. Russel, *Artificial Inteligence for the Internet Of Everything*. Elsevier, 2019.
- [11] S. J. Russell and P. Norvig, *Artificial Intelligence: A modern approach*. Prentice Hall, 2010.
- [12] K. Kravari and N. Bassiliades, "StoRM: A social agent-based trust model for the internet of things adopting microservice architecture," *Simul. Model. Pract. Theory*, vol. 94, no. March, pp. 286–302, 2019.
- [13] W. Ertel, *Introduction to Artificial Intelligence (Undergraduate Topics in Computer Science)*. 2017.
- [14] S. Sidhanta and S. Mukhopadhyay, "Managing a cloud for multi-agent systems on ad-hoc networks," *Proc. - 2012 IEEE 5th Int. Conf. Cloud Comput. CLOUD 2012*, pp. 996–997, 2012.
- [15] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," *J. Cloud Comput.*, vol. 6, no. 1, 2017.

- [16] F. De la Prieta, S. Rodríguez-González, P. Chamoso, J. M. Corchado, and J. Bajo, "Survey of agent-based cloud computing applications," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 223–236, 2019.
- [17] M. M. Al-Sayed, H. A. Hassan, and F. A. Omara, "An intelligent cloud service discovery framework," *Futur. Gener. Comput. Syst.*, vol. 106, pp. 438–466, 2020.
- [18] S. Gaglio and G. Lo Re, *Advances in Intelligent Systems and Computing* 260. Springer, 2014.
- [19] O. Shehory and A. Sturm, *Agent-oriented software engineering: Reflections on architectures, methodologies, languages, and frameworks*, vol. 9783642544. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.
- [20] K. Sinha, S. C. Ghosh, and B. P. Sinha, *Wireless Networks and Mobile Computing*. 2016.
- [21] IEEE Computer Society, *IEEE Std 802.11-2016 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, vol. 2016. 2012.
- [22] M. Gast, "802.11 Wireless Networks: The Definitive Guide," no. April. p. 630, 2005.
- [23] H. Moura, A. R. Alves, J. R. A. Borges, D. F. Macedo, and M. A. M. Vieira, "Ethanol : A Software-Defined Wireless Networking architecture for IEEE 802 . 11 networks," *Comput. Commun.*, vol. 149, no. October 2019, pp. 176–188, 2020.
- [24] V. Moorthy, R. Venkataraman, and T. Rama Rao, "Security and privacy attacks during data communication in Software Defined Mobile Clouds," *Comput. Commun.*, vol. 153, no. November 2019, pp. 515–526, 2020.
- [25] M. Barbeau, E. Kranakis, and I. Lambadaris, "Establishing a Communication Infrastructure in Ad Hoc Networks."
- [26] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *Comput. Networks*, vol. 56, no. 2, pp. 940–965, Feb. 2012.
- [27] Y. Demazeau, B. An, J. Bajo, A. F. Eds, and R. Goebel, *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity*. 2018.
- [28] J.-P. Sartre, *El ser y la nada*. 1943.
- [29] J. F. Latorre, "Implementación de un clúster de red Ad-Hoc para la adquisición de datos en la coordinación de redes logísticas," Universidad Nacional de Colombia, 2015.
- [30] H. Zárate, "Diseño de un Sub-Sistema de Cómputo Distribuido que permita implementar virtualización inalámbrica para gestionar recursos (Procesamiento, memoria, almacenamiento y dispositivos E/S) distribuidos en una Red Ad Hoc, mediante el modelo de pseudo Estado," Universidad Nacional de Colombia - Sede Bogotá, 2016.
- [31] G. D. Alvarez, "Construcción un sub-sistema de movilidad de agentes artificiales de software que permita generar desempeñan sobre el Sistema TLÖN acuerdo con el rol que ellos cuyos recursos se comparten a través de una red Ad hoc.," Unviersidad Nacional de Colombia, 2017.
- [32] S. J. Molina Sanchez, "Construcción de un módulo de seguridad Informática en el sistema TLÖN que permita tener características de disponibilidad," 2019.
- [33] J. Ospina López, "A computational justice model for resources distribution in Ad Hoc Networks," 2019.
- [34] L. T. M. Blessing and A. Chakrabarti, *DRM, a design research methodology*. 2009.
- [35] M. Becker, B. Schütt, and S. Amini, "Proposal Writing for International Research Projects," p. 98, 2014.

12. COSTOS DEL TRABAJO Y FUENTES DE FINANCIACIÓN:

Se estima unos costos aproximados basados en los ejemplos presentados en el documento guía para la presentación de proyectos internacionales de investigación elaborado por el Servicio de Intercambio Académico de Alemania (*German Academic Exchange Service*) [35]. Se adaptaron algunos valores al contexto de Colombia. La fuente de financiación principal son los fondos de ahorro personales. Sin embargo, como medida alternativa se planea aplicar a las convocatorias internas de la Universidad y a las de Colciencias.

Costos de personal (Recurso Humano):

Recurso Humano	Valor por 1 mes	Valor por 48 meses
Estudiante Doctorado	\$ 4.230.000	\$ 203.040.000
Director de tesis	\$ 6.230.000	\$ 299.040.000
TOTAL	\$ 10.460.000	\$ 502.080.000

Costos asociados a los materiales y herramientas:

DETALLE DE MATERIALES						
Descripción	Cantidad	Valor unitario			TOTAL	
			Estudiante	Universidad		
Computador	1	\$ 3.500.000	\$ 3.500.000		\$ 3.500.000	
RaspberryPi	8	\$ 170.000	\$ 1.360.000		\$ 1.360.000	
Cables y accesorios			\$ 370.000		\$ 370.000	
Papel	3	\$ 15.000	\$ 45.000		\$ 45.000	
Tinta Impresora	6	\$ 50.000	\$ 300.000		\$ 300.000	
Software			\$ 1.000.000	\$ 5.000.000	\$ 6.000.000	
Material Bibliográfico				\$ 120.000.000	\$ 120.000.000	
TOTAL			\$ 6.575.000	\$ 125.000.000	\$ 131.575.000	

Costos de viajes:

Costos de viajes	
Viajes Nacionales	\$ 15.000.000
Viajes Internacionales	\$ 38.000.000
Total viajes	\$ 53.000.000

Costos totales (incluyendo conferencias, capacitaciones y publicaciones):

Costos totales	
Recursos Humanos	\$ 502.080.000
Matriculas Doctorado	\$ 56.387.200
Materiales	\$ 131.575.000
Viajes	\$ 53.000.000
Conferencias, Capacitaciones y Publicaciones	\$ 85.000.000
TOTAL	\$ 828.042.200

13. COMENTARIO CON VISTO BUENO DEL DIRECTOR: (calificar los siguientes aspectos: organización, pertinencia, relevancia y originalidad).

14. FIRMA DEL PROPONENTE

X

Angela Consuelo Checa Hurtado

15. FIRMA DEL DIRECTOR

X

Jorge Eduardo Ortiz Triviño

16. FECHA: 27 de Octubre de 2020