Prototipo de un sistema de recomendación basado en Multiagentes para el análisis de memorias históricas y culturales



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Facultad de Ingeniería

Proyecto curricular Ingeniería de Sistemas

Nombre Estudiante	Código
Andrés David Ballén Duarte	20121020096

Bogotá D.C
2018

Prototipo de un sistema de recomendación basado en Multiagentes para el análisis de memorias históricas y culturales



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Tesis o trabajo de grado presentado como requisito para optar al título de:
Ingeniero de Sistemas

Nombre Estudiante	Codigo
Andrés David Ballén Duarte	20121020096

Director
Msc. Nancy Yaneth Gelvez García

Codirector
PhD. Helbert Eduardo Espitia Chuchango

Facultad de Ingeniería

Bogotá D.C

2018

Lema

Pero un último esfuerzo, uno más, tal vez sea el último, hay que proceder cada vez como si fuera la última, es el único medio de no retroceder.

"El innombrable" (1953), Samuel Beckett

Agradecimientos

Le agradezco a Dios por haberme dado la oportunidad de realizar la carrera de Ingeniería de Sistemas, por ser mi guia en los momentos mas duros y por brindarme una vida llena de aprendizajes, experiencias vividas en la universidad.

Le doy gracias a mis padres Lorenzo y Luz Dary por apoyarme en todo momento, por los valores que me han enseñado, ademas por la oportunidad que me dieron al momento de realizar mi carrera. Sobre todo por los consejos que me dieron los cuales me permitieron no desfallecer en la búsqueda del objetivo de finalizar la carrera.

Especial reconocimiento merece el interés mostrado por mi trabajo y las sugerencias recibidas de la profesora Nancy Yaneth Gelves, por la confianza en mí depositada. También me gustaría agradecer la ayuda recibida del profesor Helbert Eduardo Espitia por los consejos y guías para realizar este proyecto de grado. También a la profesora Lilian Bejarano Garzon por el apoyo del proyecto en los momentos precisos para la conclusión del Trabajo.

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia y amigos.

A todos ellos, muchas gracias.

Resumen

En este trabajo se realiza una propuesta de un sistema de recomendación basado en multiagentes, que muestra la arquitectura diseñada, el servidor utilizado para el desarrollo del sistema multiagente, así como la comunicación entre los agentes necesarios para realizar un recorrido recomendado. La propuesta implementada permite realizar recorrido de tipo recomendado o de favoritos a los usuarios sobre memorias históricas y culturales. Por medio de redes neuronales se determina si la ruta propuesta para el usuario es idónea, según sus gustos o preferencias o si es necesario realizar una retroalimentación de la información para mejorar los siguientes recorridos recomendados. Para generar las recomendaciones, se utilizó el conjunto de datos de un proyecto paralelo, alojado en Amazon Web Service (AWS). También se utilizo un api de base de datos de usuario en linea, el cual permitió el almacenamiento de la información. Para la visualización el proyecto se conecto a un prototipo de sistema de realidad aumentada, donde se ve la funcionalidad del sistema y los resultados obtenidos.

Palabras clave:.

Agente: Expresión gramatical que designa la persona, el animal o la cosa que realiza la acción del verbo.

Recomendación: Autoridad, representación o calidad por la que algo se hace más apreciable y digno de respeto.

SPADE: Es una plataforma Multiagent and Organizations basada en la tecnología XMPP / Jabber y escrita en el lenguaje de programación Python.

Json: Es un formato de texto ligero para el intercambio de datos.

Api: Es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Retroalimentación: El método de control de sistemas, en el cual los resultados obtenidos de una tarea o actividad son reintroducidos nuevamente en el sistema con el objeto de realizar las modificaciones necesarias.

Abstract

In this work a proposal of a recommendation system based on multiagent is made, which shows the architecture designed, the server used for the development of the multiagent system, as well as the communication between the necessary agents to carry out a recommended route. The implemented proposal allows to carry out the recommended type of route or bookmarks to users about historical and cultural memories. By means of neural networks it is determined if the proposed route for the user is suitable, according to their tastes or preferences or if it is necessary to perform a feedback of the information to improve the following recommended routes. To generate the recommendations, we used the data set of a parallel project, hosted in Amazon Web Service (AWS). An online user database API was also used, which allowed the storage of the information. For the visualization the project was connected to a prototype of an augmented reality system, where the functionality of the system and the obtained results are seen.

Keywords:

Agent: Grammatical expression that designates the person, the animal or the thing that performs the action of the verb.

Recomendation: Authority, representation or quality by which something becomes more appreciable and worthy of respect.

SPADE: Is a Multiagent and Organizations Platform based on the XMPP/Jabber technology and written in the Python programming language.

Json: It is a light text format for data exchange.

Api: It is a set of subroutines, functions and procedures (or methods, in object-oriented programming) that offers a certain library to be used by other software as an abstraction layer.

Feedback: The method of system control, in which the results obtained from a task or activity are reintroduced back into the system in order to make the necessary modifications.

Tabla de Contenido

1 Introducción **12**
- 1.1 Titulo y definición del Tema de investigación . . . 15
- 1.2 Planteamiento del problema . . . 15
  - 1.2.1 Descripción del problema . . . 15
  - 1.2.2 Formulación del problema . . . 16

2 Justificación de la investigación **17**

3 Objetivos **19**
- 3.1 Objetivo General . . . 19
- 3.2 Objetivos Específicos . . . 19

4 Delimitación **20**
- 4.1 Alcances . . . 20
- 4.2 Limitaciones . . . 20

5 Valor Diferencial **21**

6 Marco referencial **22**
- 6.1 Estado de Arte . . . 22
- 6.2 Marco Teórico . . . 26
  - 6.2.1 Matlab . . . 26
  - 6.2.2 Redes Neuronales . . . 27
  - 6.2.3 Scrum . . . 29
  - 6.2.4 Foundation for intelligent physical agents (FIPA) . . . 31
  - 6.2.5 Smart python multi-agent development environment (SPADE) . . . 32
  - 6.2.6 Extensible messaging and presence protocol (XMPP) . . . 33
  - 6.2.7 Sistema de recomendación . . . 35
- 6.3 Marco Conceptual . . . 37
  - 6.3.1 Agente . . . 37
  - 6.3.2 Red Neuronal . . . 37
  - 6.3.3 Multiagente . . . 37
  - 6.3.4 Arquitectura de Software . . . 38
  - 6.3.5 Neurolab . . . 38
  - 6.3.6 Numpy . . . 39
  - 6.3.7 Gunicorn . . . 39
  - 6.3.8 Flask . . . 39
- 6.4 Marco Metodológico . . . 40
  - 6.4.1 Tipo de estudio . . . 40
  - 6.4.2 Unidad de Análisis . . . 40
  - 6.4.3 Unidad de Estudio . . . 40
  - 6.4.4 Unidad de Tiempo . . . 40
  - 6.4.5 Unidad Geográfica . . . 40
  - 6.4.6 Metodología de la investigación . . . 40

6.4.7 Participantes . . . 41
6.4.8 Instrumentos y Equipos . . . 41
6.4.9 Metodología de Ingeniería . . . 41
6.4.10 Procedimiento de la Metodología . . . 41
6.4.11 Sprint de Inicio . . . 41
6.4.12 Sprint de Elaboración . . . 42
6.4.13 Sprint de Construcción . . . 42
6.4.14 Sprint de Transición . . . 43
6.4.15 Herramientas . . . 43

7 Desarrollo de los requerimientos 45
7.1 Requerimientos . . . 45
7.2 Casos de Uso . . . 46

8 Prototipo Sistema Multiagente 51
8.1 Estructura de los agentes inteligentes . . . 51
8.1.1 Agente Usuario . . . 51
8.1.2 Agente Sistema . . . 51
8.1.3 Agente Retroalimentación . . . 52
8.1.4 Agente Recomendación . . . 53
8.1.5 Agente Recorrido . . . 53
8.2 Arquitectura del Sistema Multiagente . . . 54

9 Diagrama de sistema 57
9.1 Diagrama de clases . . . 57
9.2 Diagrama de Actividades . . . 58
9.2.1 Registrar Usuario . . . 59
9.2.2 Registrar Preferencias . . . 60
9.2.3 Recorrido . . . 61
9.2.4 Valoración de Contenido . . . 62
9.3 Diagramas de Secuencia . . . 63
9.3.1 Valoración de Contenido . . . 63
9.3.2 Registrar Usuario . . . 64
9.3.3 Registrar Preferencias . . . 64
9.3.4 Recorrido . . . 65
9.4 Diagramas de Estados . . . 65
9.4.1 Valoración de Contenido . . . 65
9.4.2 Registrar Usuario . . . 66
9.4.3 Registrar Preferencias . . . 66
9.4.4 Recorrido . . . 66

10 Diagramas de Interacción de Agentes 67
10.1 Diagrama de Interacción de los agentes inteligentes . . . 67

11 Diagramas de la aplicación 67
11.1 Diagrama de Componentes de la aplicación . . . 68
11.2 Diagrama de infraestructura de la aplicación . . . 68
11.3 Diagrama de Nodos de la aplicación . . . 69
11.4 Diagrama punto de vista de cooperación . . . 69

12 Conexión Base de Datos **71**
12.1 Conexión Base de Datos Usuario 71
12.2 Conexión Base de Datos multimedial 73

13 Desarrollo del Prototipo **75**
13.1 Funcionamiento del Sistema 75
13.2 Desarrollo de los agentes inteligentes 76
13.3 Desarrollo de la Red Neuronal y recomendación 77
13.4 Visualización de preferencias y obras 79
13.5 Entradas y Salidas del Sistema 80
13.6 Ejecución del sistema 82
13.7 Resultados de Efectividad del sistema 85
13.8 Entregables 85

14 Conclusiones **87**

15 Recomendaciones **87**

16 Bibliografía **88**

Lista de Figuras

1 Ejemplos de implementación en Matlab [32]. 27
2 Soluciones de matlab para empresas [32]. 27
3 Plan de desarrollo de la metodología scrum [38]. 30
4 Diagrama de Sprint en scrum [39]. 31
5 Arquitectura abstracta FIPA [41]. 32
6 Modelo de la plataforma SPADE [42]. 33
7 Diagrama del agente en SPADE [42]. 34
8 Modelo de sistema de recomendación para libros electronicos inteligentes [27]. 36
9 Diagrama de referencia de un agente según FIPA [41]. 37
10 Ejemplo de punto de vista del modelado de un multiagente [45]. 38
11 Diagrama de Casos de Uso. Fuente Propia. 49
12 Arquitectura Agente Usuario. Fuente Propia. 51
13 Arquitectura Agente Sistema. Fuente Propia. 52
14 Arquitectura Agente Retroalimentación. Fuente Propia. 52
15 Estructura de la Red Neuronal. Fuente Propia. 53
16 Arquitectura Agente Recomendación. Fuente Propia. 53
17 Arquitectura Agente Recomendación. Fuente Propia. 54
18 Prototipo del sistema multiagente v1 [53]. 55
19 Prototipo del sistema multiagente. Fuente Propia. 55
20 Diagrama de Sistema. Fuente Propia. 57
21 Diagrama de Casos de Uso. Fuente Propia. 58
22 Diagrama de actividades Registrar Usuario. Fuente Propia. 59
23 Diagrama de actividades Registrar Preferencias. Fuente Propia. 60
24 Diagrama de actividades Recorrido. Fuente Propia. 61
25 Diagrama de actividades Valoración de Contenido. Fuente Propia. 62
26 Diagrama de Secuencia Valoración de Contenido. Fuente Propia. 63
27 Diagrama de Secuencia Registrar Usuario. Fuente Propia. 64
28 Diagrama de Secuencia Registrar Preferencias. Fuente Propia. 64
29 Diagrama de Secuencia Recorrido. Fuente Propia. 65
30 Diagrama de Estados Valoración de Contenido. Fuente Propia. 65
31 Diagrama de Estados Registrar Usuario. Fuente Propia. 66
32 Diagrama de Estados Registrar Preferencias. Fuente Propia. 66
33 Diagrama de Estados Recorrido. Fuente Propia. 66
34 Diagrama de Interacción de los agentes inteligentes. Fuente Propia. 67
35 Diagrama de Componentes Recorrido. Fuente Propia. 68
36 Diagrama de infraestructura. Fuente Propia. 68
37 Diagrama de Nodos. Fuente Propia. 69
38 Diagrama de Componentes Recorrido. Fuente Propia. 69
39 Modelo de la base de datos usuario [6]. 72
40 Respuesta de la petición Api Base de datos usuario [6]. 72
41 Respuesta de la petición Api Base de datos multimedial [6]. 73
42 Casos de Entrenamiento [53]. 78
43 Vista ejemplo del recorrido 13 79
44 Imagen de configuración preferencias usuario [5] 80
45 Ejemplo de muestra pintura [5] 80

46 Ejemplo de muestra escultura [5] 81
47 Pantalla Principal Servidor Web. Fuente Propia. 81
48 Respuesta de la búsqueda recorrido recomendado. Fuente Propia. . . . 82
49 Salida con recorrido pendiente. Fuente Propia. 82
50 Salida sin recorrido pendiente. Fuente Propia. 82
51 Comandos de instalación del servidor SPADE [54]. 82
52 Pagina principal del servidor SPADE. Fuente Propia. 82
53 Inicialización del servidor web. Fuente Propia. 83
54 Lista de Agentes del servidor SPADE. Fuente Propia. 83

Lista de Tablas

1 Matriz de Requerimientos Parte 1. 45
2 Matriz de Requerimientos Parte 2. 46
3 Casos de Uso 47
4 Descripción de Casos de Uso 48
5 Nombres y identificadores de los agentes. Fuente Propia. 76
6 Comportamientos Agente Usuario. Fuente Propia. 76
7 Comportamientos Agente Sistema. Fuente Propia. 77
8 Comportamientos Agente Recorrido. Fuente Propia. 77
9 Comportamientos Agente Recomendación. Fuente Propia. 77
10 Comportamientos Agente Retroalimentación. Fuente Propia. 77
11 Tabla de resultados de la red neuronal [53]. 78
12 Tabla de porcentajes recomendación. Fuente Propia. 79
13 Tabla de datos de calificación y tiempo por los usuarios. Fuente Propia. 86
14 Tabla de resultados obtenidos. Fuente Propia. 86

1 Introducción

En el transcurso de la historia se han generado hechos de transcendencia, los cuales han sido recopilados por medio de memorias artististicas y culturales, permitiendo a las personas conocer eventos de la historia humana. A través de los años los museos han recolectado estos elementos, tales como: documentos, revistas, imágenes, pinturas, etc. Según [1] Un museo es una institución permanente, sin fines de lucro, al servicio de la sociedad y abierta al público, que adquiere, conserva, estudia, expone y difunde el patrimonio material e inmaterial de la humanidad y su ambiente con fines de estudio, educación y recreo.

Hoy en día la Internet como fuente de búsqueda de las personas, ha generado, que muchos contenidos de los museos se puedan ver desde un dispositivo con acceso a la red, generando una disminución de las personas que pueden visitar estos lugares, con un porcentaje de inasistencia del 86.5% [2]. Estos eventos han generado una revolución en los museos, creando una transformación profunda a la hora de presentar las exposiciones a los visitantes, formando incentivos que permitan atraer a mas personas a ver estas colecciones de información.

El presente documento forma parte de la propuesta para realizar un prototipo de un sistema recomendador basado en multiagentes, que permita analizar la información de un usuario que acceda al sistema, creando un recorrido personalizado a través de las calificaciones, tiempo de visualización de las colecciones artísticas y culturales. Toda la información se recopila, permitiendo ver que tipo de colecciones y esculturas son idóneas para un determinado usuario. Todos los eventos que se han manejado en el proyecto se pueden observar en un trabajo similar de la Universidad de la Ciudad de Hong Kong, donde se muestra un monumento histórico virtual izado recreado para su estudio [3].

Al analizar documentos que se han realizado acerca de la relación usuario-obra-museo, se encuentra que los usuarios tiene una cantidad limitada para recorrer las colecciones que hay en un museo [7], permitiendo analizar la dificultad que se tiene al ver todas las obras de arte. Otro punto, son las obras de arte, donde el usuario puede ver colecciones que no le sean del agrado, llegando a caso de terminar el recorrido lo antes posible sin poder ver todo lo que puede llegar a ofrecer el museo. Por consiguiente, en este trabajo se quiere dar a conocer una propuesta que permita a las personas ver recorridos recomendados de colecciones de arte y esculturas que han sido vistas gratamente por usuarios que han recorrido el museo. Este desarrollo del sistema permite realizar un estudio de los gustos y preferencias de la persona para mejorar la relación entre el usuario y el museo otorgando un aprendizaje de la historia que sea beneficioso para el día a día.

Para el desarrollo del proyecto de tesis se realiza un estado de arte dividido en cuatro temas fundamentales: El uso de tecnologías por parte de los museos en las colecciones de arte, El uso de realidad aumentada en proyectos virtuales, Sistemas multiagentes para la revisión de documentos y Sistemas de recomendación de documentos y colecciones museísticas. Una vez analizados los documentos expuestos en el trabajo se obtienen las conclusiones que permiten identificar el hilo conductor para

realizar en el proyecto. Teniendo en cuenta el objetivo final, la creación del prototipo, desea recopilar información de los usuarios acerca de los gustos y preferencias, para analizar los datos obtenidos almacenados dentro de las bases de conocimiento, fomentando la conexión del aplicativo con una api con realidad aumentada, como medio de inmersión hacia un ambiente museístico.

Por medio del uso de agentes inteligentes, se desea crear un sistema multiagente que analice la información suministrada por parte de los usuarios para realizar de una manera auténtica, formativa y agradable el mejoramiento del conocimiento de la historia por parte de las personas para motivar a las personas a recorrer un museo [4]. También se muestra todo lo realizado en el proyecto, el desarrollo de la metodología Scrum, los datos arrojados por las pruebas realizadas a la red neuronal y al sistema multiagente, ademas de los resultados obtenidos al desarrollar el prototipo.

El desarrollo de este proyecto forma parte de un prototipo de sistema recomendador realizado con realidad aumentada [5]. Este proyecto usa el sistema recomendador basado en multiagentes permitiendo el manejo de los datos, utilizando los apis de usuarios y multimedial [6]. Este sistema comprende tres proyectos realizados paralelamente, el cual tiene como objetivo la inclusión de este tipo de tecnologías para crear un gran sistema conjunto. En este documento se mostrara el desarrollo del sistema recomendador basado en multiagentes, también la conexión con los otros proyectos anteriormente expuestos, así como las conclusiones que se obtuvieron.

Capítulo 1 : Contextualización de la Investigación

El capítulo primero de este proyecto muestra el análisis realizado para abarcar el proyecto. El planteamiento del problema en la búsqueda de dar una solución a las visitas de los usuarios en los museos. También se muestra los objetivos planteados para el desarrollo de la investigación así como el marco referencial, que contiene toda la información del proyecto y de investigaciones cercanas, con la finalización de la metodología de trabajo utilizada, que permitió el desarrollo del sistema recomendador.

1.1 Titulo y definición del Tema de investigación

Título: PROTOTIPO DE UN SISTEMA DE RECOMENDACIÓN BASADO EN MULTIAGENTES PARA EL ANÁLISIS DE MEMORIAS HISTÓRICAS Y CULTURALES.

Tema de Investigación: Integración de un prototipo de sistema recomendador basado en multiagentes para sugerir contenidos de memorias históricas y culturales a las personas.

1.2 Planteamiento del problema

1.2.1 Descripción del problema

En Colombia existen actualmente más de 362 museos, ubicados en los 24 de los 32 departamentos que conforman el país. Dependiendo de la ubicación geográfica cada museo tiene como objetivo formar conciencia de los acontecimientos sociales de la historia [8]. En la actualidad los museos tratan de ofrecer un nuevo paradigma de aprendizaje para las personas, como la muestra de exposiciones artísticas o exhibiciones de eventos culturales de tradiciones indígenas. Todos estos eventos crean un ambiente creciente sobre la importancia étnica y cultural que tiene el país, tanto del patrimonio histórico como de la memoria e identidad.

Una problemática que se tiene a nivel nacional es la falta de posibilidades que tienen los menores de edad para asistir a estos lugares, ya que por falta de recursos no pueden llegar a los museos, muchos de ellos con las ansias de conocer elementos importantes para su desarrollo cultural. En 2015 el DANE [2] realizo una encuesta de consumo cultural donde se encontró que el porcentaje de inasistencia a los museos durante el año 2014 fue de 84.5%, también dan a conocer las razones por las cuales las personas no asisten a los museos, una de ellas es el desinterés por asistir con un 39.1%; la falta de tiempo con 35.9% y la ubicación lejana con un 25.1%. Estos datos demuestran la poca importancia que tienen las personas por conocer la historia y acontecimientos en el país. Según [9] "Los museos no tienen más público porque no existe una formación cultural desde la escuela. En el currículo no hay una comprensión de la importancia del arte en la construcción del sujeto, lo que ocurre en otros países donde los niños, desde muy pequeños están en los museos". Esta afirmación denota la falta de formación cultural que tienen las personas para conocer y visitar estos centros de exposición en el país, creando una problemática social y cultural por parte de las personas hacia estos lugares. La falta de interés por visitar estos lugares es una problemática de cultura, donde las personas no les interesa el conocimiento y falta de aprendizaje de eventos pasados.

Cuando una persona asiste a un museo colombiano, esta puede observar las distintas obras de arte que hay en la exposición, pero no puede otorgar una calificación ni recomendar la obra que esta observando. Este síntoma de problemática a la interacción del usuario con el entorno, permite que a las obras de arte no se les de la importancia de visualización correspondiente.

Los museos dan a los visitantes una visita guiada por medio de un instructor, que presenta todo el museo de manera lineal por cada obra expuesta, donde los visitantes no tienen posibilidad de observar las colecciones que les llamen más la atención. Estos eventos generan discordia a la hora de ir de visita ya que el usuario no ve las obras que mas le llaman la atención, teniendo que observar todo el museo. Esto crea un síntoma de oportunidad perdida, ya que los usuarios van a un museo a ver obras por contenido, no por gusto, formando la ilusión de desagrado a ir a ver colecciones de arte.

En Bogotá los museos más importantes que se encuentran en la capital son el Museo Nacional de Colombia, El museo del Oro del Banco de la República, Museo de Arte Colonial entre otros. Estos museos se basan principalmente en la muestra de Arte, Historia, Arqueología y Etnografía del país, donde la mayoría muestran las colecciones de arte de manera simple y poco novedosa, es decir muestran un recuadro o arquitectura junto a un letrero que contiene una descripción de la colección. Esto genera que se necesite la ayuda de guías dentro de los museos.

Desde el enfoque interdisciplinario de la ingeniería de sistemas se observan las falencias que hay en los museos. No obstante, al analizar el uso de tecnologías que se usan en los recintos, se puede pre-veer la poca interacción de los visitantes con las colecciones de arte que están expuestas, también la poca emoción que se tiene al observar elementos que no son relevantes para los usuarios. Estos motivos permiten intuir parte de los problemas que tienen los usuario que visitan a los museos, con lo cual se formula las problemáticas que hay y el reto que se formulan para solucionarlo.

Con lo expuesto anteriormente, se fomenta la iniciativa de crear un sistema recomendador basado en multiagentes, que permita analizar los gustos y preferencias de los usuarios, por medio de las calificaciones, tiempo de visualización, configuración de movimientos y técnicas de obras de arte para crear un recorrido personalizado que recomiende memorias históricas y culturales a un usuario. El propósito del proyecto es la creación del prototipo multiagente, ademas también se utilizara un api de información de usuarios y de colecciones de arte, que permita toda la data dentro de un entorno controlado para su uso. Por otra parte por medio de la salida del sistema, se desea conectar con un aplicativo de realidad aumentada, que permita la visualización y manipulación de los datos por un usuario, donde se pueda mostrar el funcionamiento prototipo multiagente al publico.

1.2.2 Formulación del problema

¿Qué conocimientos, conceptos y saberes son necesarios para la implementación de una plataforma cognitiva de interacción que permita la recomendación de recorridos de colecciones históricas y culturales?

2 Justificación de la investigación

El impacto sociocultural que pueden generar los museos en las personas ha creado el interés de desarrollar este proyecto, donde la opinión que tiene el usuario de las colecciones históricas y culturales es importante para analizar los gustos y preferencias a la hora de visitar un centro cultural. La falta de asistencia de visitantes a los museos a creado que estos lugares evolucionen, con la creación de aplicaciones en páginas web, donde muestran las exposiciones de arte a los usuarios. En cambio en los museos colombianos, fomentar estos cambios puede ser mas complicado, debido a la falta de recursos económicos. Esto intuye en las falencias tecnológicas de los museos públicos nacionales, al momento de mostrar nuevos contenidos a los usuarios. Otro tema son los gustos de los visitantes de los museos, donde, no todas las obras expuestas son del agrado de ellos, es decir, el usuario que visita solo va a observar las obras que mas le llaman la atención.

Todos estos eventos crean la necesidad de optimizar el tiempo del usuario en el museo, mostrando los temas de importancia sin necesidad de recorrer todo el museo. Así, mediante la implementación de esta herramienta se podrá gestionar los gustos y preferencias en cuanto a obras de arte, donde por medio de la aplicación el usuario podrá agregar las preferencias que quiere en un recorrido, como el numero de obras a observar, movimiento y técnica, ademas de calificar las pinturas o esculturas que están en un recorrido. Estos datos permiten analizar el usuario que esta en el prototipo. Cada vez que el usuario utilice la plataforma, podrá crear un recorrido recomendado o favorito, que permita observar las obras de su preferencia. También con el uso de las preferencias del usuario, se formulan recorridos recomendados a otras personas que interactuen con la plataforma, y que tengan gustos similares, creando así una base de conocimiento que recolecte toda esta información para su uso correcto dentro de un museo virtual.

Los usos de los agentes inteligentes dentro del prototipo ayudaran a determinar, que tipo de información de los usuarios es relevante para crear un nuevo recorrido recomendado. Por medio del uso de redes neuronales se analiza la información obtenida por un usuario. Datos como la calificación y tiempo de visualización se obtienen como proporción de información dada por el usuario. Esto permite verificar, que falencias hubieron y que debe ser corregido para un próximo recorrido. Con este prototipo se ayudará a mejorar el comportamiento de un usuario dentro del museo, analizando sus interacciones dentro del sistema permitiendo dar una mejora continua a las obras que se puedan presentar en un recorrido.

Se pueden ver otros ejemplos de países que han implementado aplicaciones en línea, que muestran museos desde vistas virtuales como:

- La fundación Mapfre ha creado una aplicación en línea que permite conocer las actividades culturales que realiza el Instituto de Cultura en España [10].
- El museo de Louvre creo una aplicación para conocer más de 100 obras de arte que pueden verse en las exposiciones.

- El museo Estatal del Hermitage de San Petersburgo tiene una aplicación que puede realizar una visita virtual con modelos 3D de obras de arte virtualizadas [11].

Estas prácticas realizadas alrededor del mundo han hecho que los museos colombianos queden relegados a los cambios tecnológicos que han surgido. Por medio de aplicaciones en línea, se ha incentivado el desarrollo de herramientas que permiten la interacción de los usuarios con las obras en los museos, como lo son los códigos de escaneo Quick Response (QR), donde los centros históricos muestran información relacionada con las colecciones de arte, dando datos relacionados con la historia, edad y eventos importantes que han ocurrido. Todo lo anteriormente comentado se fundamenta en la importancia que tiene el uso de la información sobre las colecciones de arte.

El desarrollo del aplicativo sirve para dar a conocer elementos de historia y cultura en el país, porque por medio de la socialización y aprendizaje de estos temas, se da a conocer otro tipo de aprendizaje al publico. En la aplicación se tiene información relevante de una obra de arte o escultura, la cual es la descripción , historia, nombre de la obra, tamaño, dimensiones, etc.

La mayoría de proyectos tiene información, pero muy poco manejo e interacción del usuario. Este proyecto se creo con la importancia de dar otro tipo de solución a la problemática de visita en los museos, porque por medio de un aplicativo que muestre estas obras en cualquier lugar del mundo, permite a todas las personas acceden a un interesante contenido para visualizar. El nivel de efectividad del sistema recomendador dará a conocer el porcentaje de satisfacción de los usuarios con el aplicativo.

La importancia de desarrollar este proyecto en Colombia parte de la falta de proyectos que ayuden al conocimiento de elementos históricos y culturales; Por medio de la creación del prototipo de recomendación basado en Multiagentes, se fomente el conocimiento de memorias históricas y colecciones de arte que permitan a los usuarios conocer elementos relevantes de la historia humana. Por medio del prototipo se podrá configurar las preferencias, como movimiento y técnica de una obra de arte permitiendo al usuario calificar el contenido mostrado en los recorridos. El uso de dispositivos móviles, dará la posibilidad a todos los usuarios que tengan uno, el acceso a este aplicativo por medio de la Internet.

3 Objetivos

3.1 Objetivo General

Implementar un prototipo para la recomendación de recorridos de memorias históricas y culturales, por medio de la plataforma de desarrollo SPADE, con la creación de un sistema recomendador basado en multiagente que permita analizar las calificaciones y tiempos de un usuario en las colecciones de arte dentro de un entorno museístico, con el fin de mejorar el interés de los visitantes en los museos.

3.2 Objetivos Específicos

- Determinar el diseño del prototipo de la aplicación por medio de una arquitectura multiagente, que permita gestionar el desarrollo de recorridos personalizados para un usuario.

- Establecer un sistema de filtrado de información, que permita analizar las interacciones del usuario con la aplicación, ya sean calificaciones y tiempo de visualización de las obras, con el fin de formar un nuevo recorrido personalizado para un usuario.

- Estimar el nivel de valoración del sistema, por medio del análisis de las calificaciones dadas por el usuario a las obras de arte ofrecidas por el sistema durante la visualización de un recorrido recomendado.

- Establecer la conexión del sistema multiagente con los aplicativos de base de datos de usuario y multimedial, que permita el flujo de información necesarios para el crear un recorrido recomendado con colecciones de arte.

4 Delimitación

4.1 Alcances

El proyecto busca crear un prototipo de sistema recomendador basado en multiagentes que permita crear recorridos recomendados con colecciones de arte, utilizando la retroalimentación de tiempo y valoraciones que haya otorgado un usuario a las obras, creando así, una base de conocimiento para la gestión de recorridos para nuevos usuarios que utilicen el aplicativo.

Para el desarrollo de este proyecto se analiza el déficit que tienen las personas al momento de visitar los museos, donde se identifica la falta de fomento del aprendizaje de las personas acerca de los acontecimientos que han transcurrido en la historia en Colombia. También se analiza la dificultad de acceso a las colecciones de arte por parte de los estudiantes, creando una motivación para desarrollar el prototipo de sistema recomendador basado en multiagentes que permita relacionar las experiencias por otras personas dentro de un ambiente museístico, estableciendo el rol que tienen los usuarios al momento de otorgar una calificación a la obra de arte presentada en el recorrido. También por medio del uso de preferencias del usuario, se recopilar toda la información necesaria dentro de la base de datos. Esta recopilación de los datos del usuario permite analizarlos y almacenarlos utilizando un api de base de datos. De estos datos se gestiona una métrica que se establece dentro del sistema multiagente para procesar las calificaciones y tiempo de duración en una obra verificando que datos son esenciales para la creación de un nuevo recorrido.

Describiendo el prototipo del sistema, la principal función que tiene es la entrega de un recorrido, ya sea recomendado o favorito, donde por medio de los agentes se creara un análisis interno que permita entregar una respuesta. Para la creación del sistema se utilizara la plataforma de desarrollo de multiagente SPADE, donde por medio del estándar FIPA (Foundation for intelligent physical agents) estos agentes inteligentes coordinaran acciones que permitan llegar al objetivo del recorrido, utilizando la información del api multimedia de memorias históricas y culturales.

Para el desarrollo de el prototipo se espera tener un alcance de aceptación de mas del 60% de usuarios que puedan interactuar con el sistema, donde se busca tener un balance entre los datos que se puedan recoger de los usuarios de prueba y los usuarios finales para llegar a un prototipo estable. Con una tabla de seguimiento se desea analizar el manejo de los datos y de la información que se están utilizando para así identificar las calificaciones obtenidas por los usuarios durante la visita al recorrido.

4.2 Limitaciones

Las limitaciones de este proyecto están basadas en el manejo de la información dentro del sistema. Lo primero es la dificultad para acceder a los datos acerca de los gustos y preferencias del usuario debido a que se debe buscar y recopilar la información obtener y manejar la base de conocimiento. El siguiente inconveniente es acerca del uso de la información de las colecciones museísticas y memorias históricas, donde la creación de una base de datos con esa información puede llegar a tener inconvenientes,

uno de ellos es la fiabilidad y usabilidad de los datos y el otro es la homogeneidad de los datos. Otro problema que se tiene sobre la información histórica y cultural, es que debe ser licenciada o libre para poder presentarse en el proyecto, lo cual puede limitar y generar un costo más para completar el proyecto.

Siendo un prototipo de desarrollo de agentes, estos agentes se desean desarrollar con la plataforma SPADE implementada en la versión 2.7 de Python. El manejo y análisis de la red neuronal se utiliza por medio de Matlab ya que este elemento permite analizar, entrenar y ver gráficamente la red neuronal antes de introducirla en el sistema. Para el manejo del servidor web se utilizará por defecto el que viene el python, el cual permite la conexión y manejo de información del sistema con otras entidades. Para el manejo del sistema se desea buscar un servidor gratuito en la nube que proporcione el almacenamiento del sistema para que tenga una alta disponibilidad de ser buscado en cualquier lugar. El uso de computadores, celulares, tabletas son herramientas que se tienen por el estudiante a cargo.

5 Valor Diferencial

El desarrollo del proyecto permitirá dar a conocer a las personas, otra manera de interactuar con memorias históricas y culturales, donde por medio de configurar preferencias y dar una calificación a una obra de arte, se fomente la creación de un recorrido recomendado o favorito para un nuevo usuario. El uso de un sistema multiagente permitirá mejorar el uso y tratamiento de información de una base de datos, dando una respuesta rápida y eficiente en términos de solución al momento de entregar una nueva recomendación. Por medio de la red neuronal se posibilita crear un nuevo auto-aprendizaje de la información almacenada en el sistema para analizar los datos de un usuario.

Este proyecto permitirá gestionar datos de memorias históricas en cualquier lugar del país, donde la información puede utilizarse para crear una memoria museística de esos datos, ayudando a cuidar y conservar los eventos culturales del mundo utilizando esta tecnología. El uso de la información de los usuarios y colecciones proporcionaran datos que permiten su comprensión, para crear nuevas recomendaciones de colecciones de arte para las personas.

6 Marco referencial

6.1 Estado de Arte

Debido a la gran masa de información que tienen los museos y al poco uso de tecnologías de vanguardia que permitan una interacción del usuario con las colecciones, los museos han utilizado la técnica de museografía como el conjunto de técnicas y prácticas relativas al funcionamiento de un museo para atraer a las personas hacia los museos [12]. Un ejemplo claro de museografía es el trabajo titulado Animating narratives of remembrance que expone una proyección en 3D de historias, imágenes y metadatos recopilados de la Primera Guerra Mundial utilizando estrategias de visualización y museografía permiten dar a conocer a los visitantes los testimonios y eventos que sucedieron durante esa época redefiniendo las memorias culturales [13].

Al intentar plasmar el uso de plataformas digitales en los museos se encuentran trabajos como "The Virtual Museums of Caen: A case study on modes of representation of digital historical content [14]", que intentan plasmar la imagen de reconstruir un museo virtual de Caen por medio del uso del Google "cardboard" donde los usuarios puedan explorar el museo por medio de esta tecnología. También se observa el uso de museografía a través de la realidad aumentada el cual permite realizar una copia digital de un objeto real para llevarlo al plano digital, como se observa en el documento la virtualización de las grutas de Mogao, provincia de Gansu, China del noroeste. Este objeto digitalizado permite mostrar al público un contenido cultural de un sitio donde las personas no pueden tener acceso. Esto fomenta la utilización de una interfaz tecnológica con que el usuario pueda interactuar con la obra [3]. Para ilustrar las tecnologías de información en el aprendizaje del patrimonio cultural en los países, en 2015 se creó en el congreso internacional del Patrimonio una propuesta cuyo objetivo es promover el interés desde una perspectiva global a través de la virtualización de un museo virtual que permita introducir a las personas en la historia de la antigua Grecia utilizando la recolección de bodegas de información de la historia por medio de la tecnología [15]. Estos trabajos permiten observar la importancia de la museografía que se tiene al renovar y exponer eventos pasados de la historia por medio de herramientas tecnológica que mejoran la interacción entre el usuario y el producto.

En el uso de tecnologías puede observarse que los museos han adquirido programas que incentivan la interacción del usuario con las colecciones que se presentan, un ejemplo claro es el museo del MIT, donde al usar las Google Glass el usuario obtiene información detallada de los objetos que está observando, el usuario puede seleccionar varios tipos de opciones como: vista, visión de conjunto, código QR y empezar el tour. Estas opciones abarcan una gran variedad de posibilidades al momento de seleccionar una visita en el museo [16]. Un ejemplo del uso de las Google Cardboards es el utilizar una aplicación visual en el celular para poder observar el entorno creado a través de la realidad aumentada. Las Google Cardboard son gafas de cartón plegable que sirven para alojar un dispositivo móvil, este tiene como objetivo ofrecer a un coste mínimo una experiencia de realidad aumentada a los usuarios. En el documento de "Personalized Augmented Reality Experiences in Museums using Google Cardboards" [17] los integrantes del grupo usan las Google Cardboards para probar el contenido de una aplicación creada para proporcionar visitas al Museo Arqueológico de Trípolis exami-

nando las ventajas y desventajas que tuvieron al usar este elemento, debido al costo que tiene la tecnología de realidad aumentada al desarrollar un proyecto con bajos recursos.

Al revisar los proyectos de realidad aumentada se puede identificar una parte importante, que es el reconocimiento de objetos, donde existen trabajos que permiten identificar el entorno virtual que usan para manipular el entorno real a su conveniencia. Uno de ellos es el trabajo de [18] el cual trata de la identificación de marcadores de realidad aumentada para analizar los distintos ángulos que se pueden ver con una cámara. Este proyecto indica el ángulo y distancia correcta que debe tener un usuario a la hora de usar la aplicación para que no tenga inconvenientes en la detección de los marcadores de posición. Esta información es importante debido a que se debe conocer los alcances y desventajas que se tienen al probar este tipo de tecnología. Otro de los trabajos se observa en [19] donde se muestra la importancia del manejo de los objetos al momento de trabajar en la realidad aumentada utilizando el software de vuforia SDK, puesto que debido a la profundidad y ángulo de el objeto virtualizado en el mundo real este tendrá un comportamiento según lo indique el usuario. Estos trabajos ayudan a entender el funcionamiento de herramientas de realidad aumentada permitiendo identificar los hardware y software necesarios para el desarrollo de esta tecnología.

Adicionalmente en el tópico de los multiagentes se puede observar que los trabajos realizados en los últimos años tienen a analizar una gran cantidad de información, uno de ellos es un sistema multiagente integrador de bibliotecas digitales [20]. En este se expone un sistema multiagente que realiza la búsqueda de documentos de una biblioteca virtual donde permite al usuario realizar consultas de libros de una manera rápida y eficaz. En el artículo de Modelado de un sistema multiagente aplicado a la predicción de la personalidad en Twitter [21], los autores del documento realizan un sistema multiagente que permite a través del test de personalidad de los cinco grandes estudiar los usuarios que están en la red social, estos datos los clasifica a través de Naive Bayes y KNN, mostrando los resultados de los clasificadores en los escenarios marcados del proyecto. Estos documentos muestran el uso de los Multiagentes en distintas ramas y con un propósito como la agilización de información a la hora de encontrar una respuesta a lo planteado.

Un grupo de trabajo de la Universidad de Southeast China realizo un trabajo titulado Robust Neuro-Adaptive Containment of Multileader Multiagent Systems With Uncertain Dynamics [22], donde describen un sistema multiagente (MAS) que tiene de capacidad manifestar la solución a un problema de control de cooperación de los MAS frente a una directriz múltiple que está sujeta a una dinámica no lineal desconocida con perturbaciones externas. Este documento explica los problemas de contención asintótica que se tiene al hacer uso de redes neuronales, teoría de aproximación y la teoría de estabilidad de Lyapunov de los sistemas de non-smooth, mediante la introducción de una matriz de ponderación ideales para el aproximador de redes neuronales dentro de cada seguidor una nueva clase de controladores de contención están diseñados adicionalmente para alcanzar precisamente contención asintótica en la masa considerado donde la norma euclidiana de contención vector de error converge asintóticamente a cero. En el último, se muestran simulaciones numéricas para verificar la validez de estos resultados teóricos demostrando que el vector de error contención del bucle cer-

rado será uniformemente en última instancia acotado si los datos de las simulaciones son validados con los resultados teóricos. En la universidad de tecnología de Hefei en china Design of workshop production management control system based on Multi-Agent [23]. Este trabajo habla de un sistema de tecnología multiagente que se aplica en la fabricación avanzada de componentes electrónicos proponiendo un sistema flexible, integrado, inteligente y de gestión en red taller, donde el proceso de producción lo simulan mediante el uso de matlab.

En la séptima conferencia de inteligencia computacional, modelamiento y simulación de 2016 se presenta el artículo "Simulación del enfoque multiagente en la nube multi-Medio Ambiente Uso de Matlab [24]". Donde se habla de la importancia que tiene hoy en día la computación en la nube que permite a los usuarios realizar migración de tareas en los sistemas ubicuos. Se muestran las dificultades que se tiene al tener un gran número de proveedores y servicios disponibles para acceder al servidor. Este artículo presenta un enfoque multiagente para ayudar a la selección y provisión de los recursos en un entorno multi-nube para las solicitudes entrantes. Este marco permite la monitorización automatizada y aprovisionamiento de recursos. Esta investigación ha demostrado y puesto en práctica este enfoque en un entorno multi-nube simulada creada en Matlab para el aprovisionamiento de recursos informáticos se pueden beneficiar a los usuarios. En el congreso de [25] se presentó un sistema de combinación multiagente de algoritmos evolutivos y minería de datos para mejorar la búsqueda y problemas de optimización del mundo real, este documento muestra la importancia que tiene las funciones objetivo y restricciones para realizar evaluaciones de simulaciones matemáticas de los sistemas bajo análisis prescindiendo del uso de derivadas. Los autores proponen que hay algunos algoritmos que funcionan mejor en algunos problemas y mal en otros, donde los autores proponen enfoque basado en la hibridación de varias metaheurísticas que actúan sinérgicamente, al uso de parámetros auto-adaptativos y en la introducción de reglas obtenidas tanto del conocimiento del problema, como de derivaciones de conocimiento a partir de soluciones exploradas en las generaciones anteriores. El entorno general para estos componentes es el llamado paradigma multiagente. Con la combinación de estos elementos, se pueden desarrollar aplicaciones de apoyo a la toma de decisiones. Este trabajo busca ayudar en la formulación del problema y aumentar la eficiencia computacional para poder tratar problemas del mundo real. En concreto, aquí abordan el problema de diseño de una red de distribución de agua.

Otro trabajo titulado Intelligent Control and Asset Management: An Event-based Control Road Map muestra el desarrollo de un sistema de control inteligente y de gestión de archivos (ICAM) con enfoque específico en la industria del petróleo, donde han gestionado una arquitectura multiagente dividiéndose en diversos agentes para realizar tareas específicas. La mayoría de los agentes son, por naturaleza, computacionalmente intensivas; éstas fueron implementadas como rutinas de matlab. El Supervisor (o "agente maestro"), por otra parte, estaba destinado a incorporar la "inteligencia" de ICAM, por lo que se creó en el sistema experto G2. Este sistema experto tiene como objetivo manejar los niveles de eficacia y operación de los procesos de desarrollo permitiéndole ser más eficaz, eficiente y más fácil de extender y mantener [26].

Todos los sistemas multiagentes buscan resolver problemas de una manera clara y

concisa de ver los problemas de una manera general para así simplificarlos y gestionarlos para llegar a una solución, por último, se habla de los sistemas recomendadores, estos sistemas permiten aconsejar y mejorar la relación de los usuarios con los sistemas que estén usando. Estos sistemas gestionan la información que tienen en el entorno para así fomentar información clara y concisa de que temas están gustando a las personas y crear una recomendación personalizada de datos.

En [27] se habla de un sistema recomendador de libros basado en el comportamiento de usuarios donde se plantea la problemática al seleccionar libros de investigación que sean relevantes para una persona, donde se presenta las problemáticas que se tienen con el volumen de información, retroalimentación y capacidad de computo a la hora de procesar esos datos en tiempo real. Estos sistemas recomendadores ayudan a solucionar problemas de procesamiento de datos a la vez que mejoran los tiempos de entrega de resultados, otro proyecto con este tema se basa en un sistema recomendador de recursos educativos de nombre Recommend Me [28]. Este sistema desarrolla un algoritmo utilizando el framework Django que tiene como propósito observar las interacciones del usuario con el sistema identificando los cursos que ofrece el programa de educación, otorgando una nota para identificar la precisión del algoritmo al momento de ser probado. Otro documento llamado Intelligent Digital Learning: Agent-Based Recommender System [29] habla de un sistema de recomendación basado en agentes teniendo como objetivo ayudar a los estudiantes a superar sus deficiencias sugiriendo recursos de aprendizaje pertinentes. La principal idea es proporcionar el apoyo adecuado a fin de que su experiencia de aprendizaje eficaz. Los autores diseñan un sistema cooperativo basado en agentes autónomos donde los agentes son capaces de actualizar y mejorar el resultado de la recomendación en nombre de sus experiencias pasadas en la plataforma de aprendizaje.

En la conferencia internacional sobre tecnologías avanzadas de aprendizaje de la IEEE en 2017 se presentó [30], este documento muestra la tecnología mejorada para el aprendizaje, donde es un tema que ha recibido mucha preocupación por los investigadores. Al mismo tiempo, hay una cantidad cada vez mayor de los recursos educativos abiertos indexados de acuerdo con los estándares del World Wide Web Consortium (W3C). Los recursos educativos abiertos (REA) pueden ser un complemento útil de los contenidos ofrecidos a un alumno durante un curso en línea. El cálculo de la mejor REA para ofrecer al alumno en cada punto de su curso es un aspecto de personalización del desarrollo del proyecto. En el artículo muestra el diseño del sistema MORS para resolver este problema en el contexto de Massive Open Online Course (MOOC). Este sistema se basa en un aprendiz perfil, en los metadatos que describen el curso y en un proceso cuidadosamente diseñado para consultar los puntos finales para los REA.

Para la recomendación de aprendizaje de objetos el artículo Argumentation-based Hybrid Recommender System for Recommending Learning Objects [31], habla de un método hibrido basado en la teoría de la argumentación que combina elementos basados en contenido, técnicas de recomendación y de colaboración basadas en el conocimiento proporcionando a los estudiantes un sistema que es capaz de generar más argumentos para justificar su idoneidad. Este método ha sido probado mediante el uso de una base con datos reales sobre los estudiantes y objetos de aprendizaje, obteniendo resultados prometedores donde se presenta un método recomendación híbrido fundamentado en

la teoría de la argumentación que combina elementos basados en contenido, técnicas de recomendación y de colaboración basadas en el conocimiento y proporciona a los estudiantes con los objetos para los que el sistema es capaz de generar más argumentos para justificar su idoneidad.

En el tema de museos el trabajo de Beounaret y Lenne [7], muestra la implementación de sistemas recomendados en las colecciones de museos debido a la basta cantidad de información que se muestra a una persona al visitar un museo. Este trabajo muestra la importancia por encontrar obras de arte relevantes en un tiempo determinado al usar dispositivos móviles para guiar a las personas dentro de los museos. Abordando los temas en este estado de arte se puede observar la importancia de los museos en el presente donde por medio de técnicas como la museografía, permiten al usuario interactuar y tener una mejor experiencia al estar en un museo, también por medio del uso de multiagentes y sistemas recomendadores se analizaron documentos que permiten identificar la relación de los conjuntos de información que se tienen en las bases de datos, permitiendo hacer un uso de esa información para recomendar y obtener resultados a los problemas que se tengan. Por medio de las memorias históricas y culturales de los hechos se prevé usar esa información para gestionar y crear memorias de eventos que permitan interpretar y dar a conocer los datos de una manera práctica y eficiente.

6.2 Marco Teórico

6.2.1 Matlab

Matlab es una herramienta de desarrollo de software para el análisis iterativo y de procesos de diseño con un lenguaje de programación. Está orientado para llevar a cabo proyectos donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. Matlab integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo, donde los problemas y sus soluciones son expresados del mismo modo que se escriben tradicionalmente, sin necesidad de hacer uso de la programación tradicional.

El nombre de matlab proviene de la contracción de los términos Matrix Laboratory, y fue creado principalmente para proporcionar fácil acceso a las librerías LINPACK y EISPACK, siendo estas, las librerías de mayor importancia en la computación y el cálculo matricial. El programa permite realizar de un modo rápido la resolución numérica de problemas en un menor tiempo, que al resolverlos con otros lenguajes tradicionales como pueden ser Fortran, C o Java.

Este software goza en la actualidad de un alto nivel de implementación en centros universitarios y también en departamentos de investigación y desarrollo de compañías en el mundo. En entornos universitarios esta herramienta se usa para el cálculo de expresiones de grado superior. También se usa para la solución de ecuaciones no lineales y lineales.

Para el desarrollo de una producción matlab las implementa principalmente en tres puntos que se pueden observan en la figura 1.



Figura 1: Ejemplos de implementación en Matlab [32].

Las soluciones que nos ofrece esta aplicación, permiten resolver problemas de distintas maneras, estos pueden ser problemas de aprendizaje profundo, visión por computador, procesamiento de señal, finanzas, robótica o sistemas de control, todos estos pertenecen al gran abanico de posibilidades para realizar un proyecto con la plataforma.



Figura 2: Soluciones de matlab para empresas [32].

6.2.2 Redes Neuronales

Una de las ramas de la inteligencia artificial muy aplicada por la comunidad científica son las redes neuronales artificiales (RNAs), describiéndolas como las redes en las que existen elementos procesadores de información cuyas interacciones locales dependen del comportamiento del conjunto del sistema [9].

El principal propósito de los sistemas neuronales es el control centralizado de las funciones biológicas. Algunas de ellas se encargan del abastecimiento de energía; el sistema neuronal está conectado con el metabolismo, control del cardio y la respiración. No obstante, cuando se habla del tema de computación neuronal solo se toma en cuenta las funciones sensoriales y motoras, así como un proceso interno que se puede denominar como pensamiento [33].

El desarrollo de nuevas tecnologías de la información actualmente existe una tendencia que permite reproducir todas esas fases de entrenamiento de las redes neuronales artificiales. Sin embargo, la implementación de funciones sensoriales artificiales para hacer que las maquinas puedan ver y oír permitiendo crear una nueva extensión hacía una nueva técnica de aprendizaje en los computadores.

Las primeras explicaciones teóricas sobre el cerebro y el pensamiento ya fueron dadas por algunos antiguos filósofos griegos como Platón (427-347 a.C.) y Aristóteles (384-422 a.C.). Las mismas ideas sobre el proceso mental también las mantuvo Descartes (1596-1650) y los filósofos empiristas del siglo XVIII.

Alan Turing, en 1936, fue la primer persona en estudiar el cerebro como una nueva forma de ver el mundo de la computación; sin embargo, las primeras personas que dieron a conocer los fundamentos de la computación neuronal fueron Warren Mc-Culloch, un neuro fisiólogo y Walter Pitts, un matemático, quienes en 1943 establecieron una teoría acerca de cómo trabajaban las neuronas [34].

En el año 1957, Frank Rosenblatt comenzó el desarrollo de una neurona con el nombre de Perceptrón. El perceptrón es una de las redes neuronales más antigua. Hoy en día se usa para reconocer patrones de comportamiento. En su época este modelo era capaz de generalizar el aprendizaje de reconocer otros patrones de comportamiento similares, aunque no se le hubieran presentado anteriormente. Sin embargo, tenía una serie de limitaciones, la más conocida era la de la función OR-exclusiva y en general, no era capaz de clasificar clases no separables linealmente [35].

En 1959, Bernard Widrow y Marcial Hoff, de Stanford, desarrollaron el modelo ADALINE (ADAptative LINear Elements). Esta fue la primera red neuronal aplicada a un problema real que consistía en eliminar los ecos de las líneas telefónicas utilizando filtros adaptativos, donde se ha usado comercialmente durante varias décadas.

En 1969 las redes neuronales tuvieron numerosas críticas que frenaron su desarrollo hasta 1982. Marvin Minsky y Seymour Papert, del Instituto Tecnológico de Massachusetts (MIT) publicaron un libro de nombre perceptrones donde modificaban y mejoraban la idea de Rosenblat acerca de esta neurona.

En 1974 Paul Werbos desarrolló la idea básica del algoritmo de aprendizaje de propagación hacia atrás (backpropagation); cuyo significado quedó definitivamente aclarado en 1985. En 1977 Stephen Grossberg hablo de la Teoría de Resonancia Adaptada (TAR). La Teoría de Resonancia Adaptada es una arquitectura de red que se diferencia de todas las demás previamente inventadas. La misma simula otras habilidades del cerebro: memoria a largo y corto plazo.

En 1985, el Instituto Americano de Física comenzó lo que ha sido la reunión anual Neural Networks for Computing. En 1987, el IEEE celebró la primera conferencia internacional sobre redes neuronales con más de 1.800 asistentes y 19 nuevos productos mostrados. A partir de este momento, se creó un auge por el interés por esta área de la I+D, como lo demuestran tanto el número de congresos y reuniones científicas especializadas, como la aparición de revistas científicas de calidad dentro de la inteligencia artificial, así como el interés demostrado por diversos tipos de empresas en utilizar esta tecnología para desarrollar aplicaciones concretas [33].

En la actualidad se pueden encontrar varios proyectos de redes neuronales, uno de ellos es la memoria USB de nombre Movidius. Esta memoria diseñada por Intel tiene la capacidad de ejecutar una red neuronal con solo conectarla a un computador [36]. Por otra parte, otra empresa de software con inteligencia artificial Deep mind abrió una plataforma de entrenamiento que permite construir redes neuronales de una manera práctica y eficiente para el desarrollo de esta tecnología [37]. Estos proyectos se juntan además de las nuevas prácticas de las empresas para ofrecer conducción autónoma en los vehículos por medio del uso de las redes neuronales.

6.2.3 Scrum

Scrum es una metodología con enfoque ágil para el desarrollo de productos de software, que se aplica de manera regular para el trabajo en equipo obteniendo el mejor resultado posible de un proyecto. Esta metodología de trabajo permite el desarrollo rápido y sencillo del trabajo al desarrollo de software, donde se manejan prioridades del proyecto para que los equipos se auto-organicen, permitiendo determinar la manera de cumplir las expectativas al momento de emplear los esfuerzos en el desarrollo de un proyecto.

En Scrum se realizan entregas parciales y regulares del producto final, donde se prioriza el beneficio que pueden aportar al receptor del proyecto. Scrum está capacitado para trabajar en entornos completos, donde el tiempo para desarrollar el proyecto es corto, teniendo requisitos cambiantes o pocos definidos y donde la competividad, la innovación y la productividad son ejes fundamentales.

En Scrum se habla del enfoque de carrera de relevos, donde se permite el analizar el conflicto que puede haber al desarrollar los objetivos de máxima velocidad y flexibilidad de un proyecto; Creando en su lugar un enfoque holístico o estilo rugby, donde un equipo intenta ir a la distancia como una unidad, pasando la pelota hacia adelante y hacia atrás para tener requisitos competitivos al momento de desarrollar un sistema [38].

Uno de los puntos fuertes de esta metodología de desarrollo, es la facilidad para resolver situaciones con el cliente para evitar que el producto final se demore demasiado, los costes se disparen o la calidad no sea aceptable, también cuando la moral de los equipos de trabajo es baja ó se quiere identificar y solucionar ineficiencias sistemáticamente para desarrollar un producto.

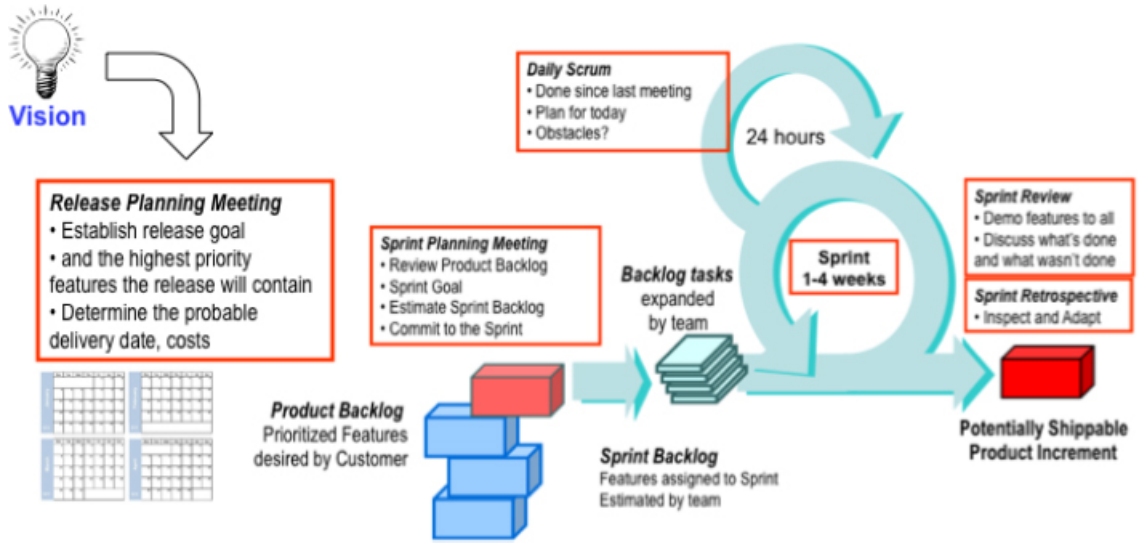


Figura 3: Plan de desarrollo de la metodología scrum [38].

Los principales Beneficios de Scrum son:

- Entrega de resultados: Al realizar entregas mensuales o quincenales de resultados se puede gestionar los avances con el cliente, también obtener resultados anticipados de avances del proyecto, tener una flexibilidad y adaptación a las necesidades del cliente o a los cambios en el mercado.
- Mitigación de Riesgos: Al tener reuniones de entregas se pretende gestionar los errores que pueda haber en el proyecto, corrigiéndolos y evitando fugas de mala producción.
- Relación entre el cliente y el equipo de desarrollo: Con esta metodología el desarrollo del proyecto va acorde a lo que quiere el cliente.
- Maximización del retorno de inversión (ROI): Se genera la producción de software que aportan un valor mayor a un negocio teniendo un retorno de inversión.
- Productividad y Calidad.

El desarrollo en Scrum se realiza de manera iterativa e incremental. Cada iteración se denomina Sprint, donde tiene una duración determinada de 2 y 4 semanas, obteniendo como resultado una nueva versión de software con prestaciones listas para su uso. Cada Sprint, se ajusta a la funcionalidad de la aplicación construida añadiendo nuevas prestaciones que aporten un valor mayor al negocio.

Los pasos en el desarrollo del producto son los siguientes:

- Product Backlog: Conjunto de requisitos que se usan para identificar la inversión que se va a realizar considerando los costos y beneficios.

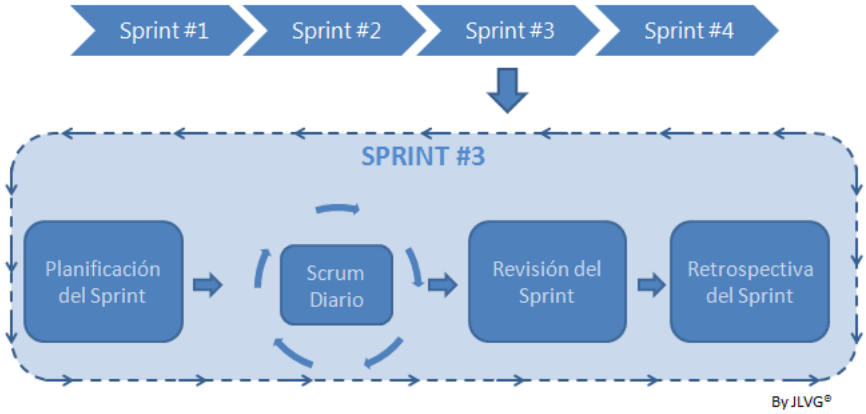


Figura 4: Diagrama de Sprint en scrum [39].

- Sprint Planning: Reunión con el Cliente el cual tiene como objetivo determinar los Sprint y reuniones con los desarrolladores.
- Sprint: Iteración de duración de una parte del proyecto para realizar cambios o adaptaciones de software.
- Sprint Backlog: Lista de las tareas necesarias para llevar a cabo el sprint.
- Daily Sprint Meeting: Reunión diaria de cómo máximo 15 minutos donde el equipo se sincroniza para trabajar de forma coordinada.
- Demo y retrospectiva: Reunión que se celebra al final del sprint para presentar el avance realizado durante un sprint.

Para el desarrollo de un proyecto el equipo de Scrum está formado por los siguientes roles:

- Scrum máster: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología.
- Product owner (PO): Representante de los clientes, el cual se enfoca en la parte del negocio y el responsable de entregar el valor al dinero invertido (ROI) del proyecto.
- Team: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta cada Sprint.

6.2.4 Foundation for intelligent physical agents (FIPA)

FIPA es una organización de estándares de la Sociedad de Computación IEEE que promueve el uso de agentes inteligentes y su uso con otras tecnologías.

FIPA, la organización de estándares para agentes y sistemas multi-agente, fue oficialmente aceptada por el IEEE como su undécimo comité de estándares el 8 de junio de 2005.

FIPA se formó originalmente como una organización suiza en 1996 para producir especificaciones de estándares de software para agentes heterogéneos e interactivos y sistemas basados en agentes. Desde sus inicios, FIPA ha jugado un papel crucial en el desarrollo de estándares de agentes y ha promovido una serie de iniciativas y eventos que contribuyeron al desarrollo y la aceptación de la tecnología de los agentes. Además, muchas de las ideas originadas y desarrolladas en FIPA ahora se están enfocando claramente en las nuevas generaciones de tecnología Web / Internet y especificaciones relacionadas [40].

En marzo de 2005, la Junta Directiva de FIPA presentó esta oportunidad a toda la membresía de FIPA, quienes votaron unánimemente para unirse a la Sociedad de Computación de IEEE. Ahora es el momento de mover los estándares para los agentes y los sistemas basados en agentes al contexto más amplio del desarrollo de software. En resumen, la tecnología del agente debe funcionar e integrarse con tecnologías que no sean agentes. Con este fin, la IEEE Computer Society ha aceptado formalmente a FIPA para formar parte de su familia de comités de estándares.

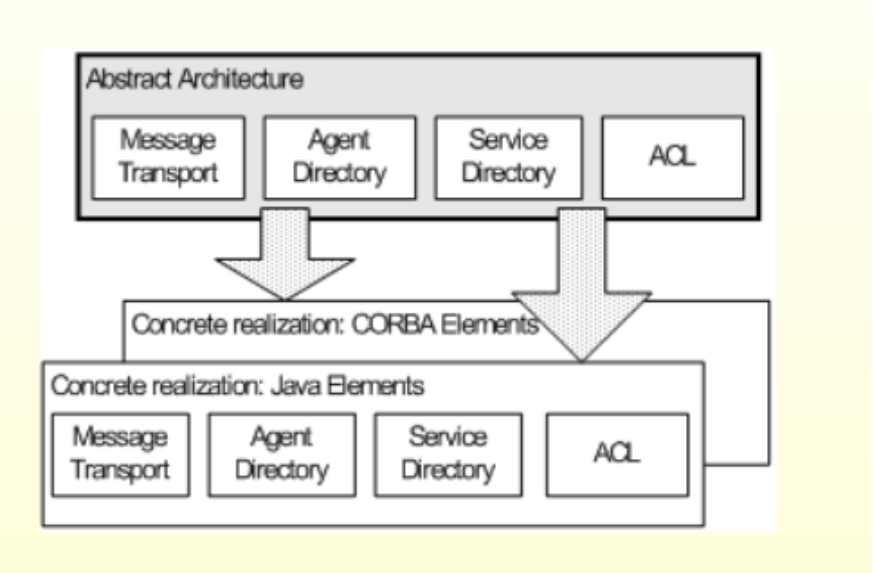


Figura 5: Arquitectura abstracta FIPA [41].

6.2.5 Smart python multi-agent development environment (SPADE)

SPADE es una plataforma libre de sistemas multi-agente desarrollada en Python y basada en la tecnología de mensajería instantánea XMPP desarrollada en 2005 en la Universidad Politécnica de Valencia. La plataforma nació como una prueba de concepto para probar la tecnología de la mensajería instantánea como protocolo de transporte para los agentes inteligentes. Desde ese momento el proyecto ha seguido creciendo y añadiendo nuevas características aprovechando la flexibilidad del protocolo de mensajería instantánea basado en XML y la cantidad de extensiones desarrolladas para el mismo que son aprovechables dentro del marco de los sistemas multi-agente. La plataforma SPADE está basada en un conjunto de estándares, siendo los más desta-

cables FIPA y XMPP/Jabber [42].

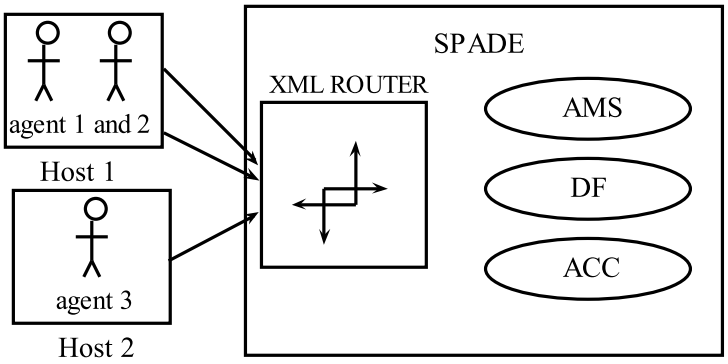


Figura 6: Modelo de la plataforma SPADE [42].

Las principales características de la plataforma SPADE son:

- Soporte del estándar FIPA mediante el protocolo de mensajería instantánea XMPP (Agentes AMS y DF incluidos).
- Notificación de presencia entre agentes.
- Organizaciones Virtuales basadas en el protocolo de multi-conferencia MUC.
- Comunicación P2P entre agentes.
- Invocación remota de servicios entre agentes usando el estándar XML-RPC.
- Procesadores de lenguajes de contenido.
- Modelo de agente BDI basado en Conocimiento, Deseos e Intenciones.
- Modelo de comportamientos: Cíclicos, Periódicos, Timeout, una ejecución, máquina de estados finita y basado en eventos.
- Soporte de comunicación con otras plataformas mediante diversos protocolos de transporte: JADE y SIMBA.
- Interfaz gráfica basada en web.

6.2.6 Extensible messaging and presence protocol (XMPP)

XMPP es el protocolo extensible de mensajería y presencia, un conjunto de tecnologías abiertas para mensajería instantánea, presencia, chat multiusuario, llamadas de voz y video, colaboración, middleware liviano, sindicación de contenido y enrutamiento generalizado de datos XML [43].

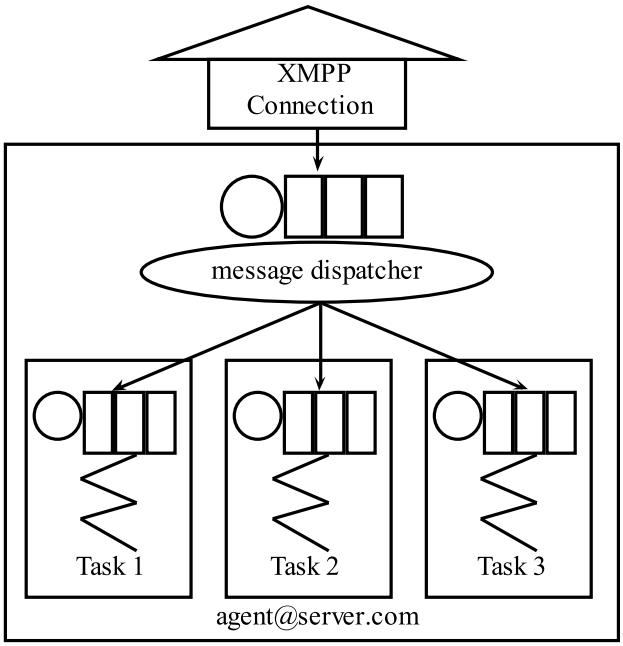


Figura 7: Diagrama del agente en SPADE [42].

XMPP se desarrolló originalmente en la comunidad de código abierto Jabber para proporcionar una alternativa abierta y descentralizada a los servicios cerrados de mensajería instantánea en ese momento. XMPP ofrece varias ventajas clave sobre tales servicios:

- Abierto: Los protocolos XMPP son gratuitos, abiertos, públicos y fácilmente comprensibles; Además, existen múltiples implementaciones en forma de clientes, servidores, componentes de servidor y bibliotecas de códigos.
- Estándar: El Grupo de trabajo de ingeniería de Internet (IETF) ha formalizado los principales protocolos de transmisión XML como una tecnología aprobada de mensajería instantánea y presencia. Las especificaciones XMPP se publicaron como RFC 3920 y RFC 3921 en 2004, y la XMPP Standards Foundation continúa publicando muchos Protocolos de Extensión XMPP. En 2011, se revisaron las RFC principales, lo que dio como resultado las especificaciones más recientes (RFC 6120, RFC 6121 Y RFC 7622).
- Comprobado: Las primeras tecnologías Jabber / XMPP fueron desarrolladas por Jeremie Miller en 1998 y ahora son bastante estables; cientos de desarrolladores están trabajando en estas tecnologías, hay decenas de miles de servidores XMPP corriendo en Internet hoy en día, y millones de personas usan XMPP para mensajería instantánea a través de servicios públicos tales como implementaciones de Google Talk y XMPP en organizaciones de todo el mundo.
- Descentralizado: La arquitectura de la red XMPP es similar al correo electrónico; como resultado, cualquiera puede ejecutar su propio servidor XMPP, permitiendo a las personas y organizaciones tomar el control de su experiencia de comunicación.

- Seguro: Cualquier servidor XMPP puede estar aislado de la red pública (por ejemplo, en una intranet de la compañía) y se ha integrado una seguridad robusta utilizando SASL y TLS en las especificaciones principales de XMPP. Además, la comunidad de desarrolladores de XMPP está trabajando activamente en el cifrado de extremo a extremo para elevar aún más la barra de seguridad.

- Extensible: Usando el poder de XML, cualquiera puede construir una funcionalidad personalizada sobre los protocolos centrales; para mantener la interoperabilidad, las extensiones comunes se publican en la serie XEP, pero dicha publicación no es necesaria y las organizaciones pueden mantener sus propias extensiones privadas si así lo desean.

- Flexible: Las aplicaciones XMPP más allá de la mensajería instantánea incluyen administración de red, sindicación de contenido, herramientas de colaboración, intercambio de archivos, juegos, monitoreo de sistemas remotos, servicios web, middleware liviano, computación en la nube y mucho más.

- Diverso: Una amplia gama de empresas y proyectos de código abierto utilizan XMPP para crear y desplegar aplicaciones y servicios en tiempo real; nunca se verá "bloqueado" cuando use tecnologías XMPP.

6.2.7 Sistema de recomendación

"Un sistema recomendador es una tecnología de filtrado de información personalizada, usada para predecir si a un usuario particular le gusta un ítem en particular (problema de predicción), o identificar un conjunto de N ítems que pueden interesarles a ciertos usuarios (problema de recomendación top-N.) [51]."

"Los sistemas de recomendación son una serie de mecanismos y técnicas aplicadas a la recuperación de información para intentar resolver el problema de sobrecarga de datos en Internet. Estos ayudan a los usuarios a elegir los objetos que pueden ser de utilidad o de su interés, pudiendo estos objetos ser cualquier tipo, tales como libros, películas, canciones, páginas Web, blogs [52]."

En estas definiciones puede observarse que un sistema de recomendación es un conjunto de mecanismos y técnicas que sirven para recuperar información para intentar predecir la solución a un problema de recomendación.

Los sistemas recomendadores se caracterizan por asemejar información relevante del usuario, pero se distinguen de la siguiente manera:

- Frecuencia de uso: Los sistemas son enfocados para un uso prolongado y de larga duración.

- Representación de necesidades: Estos sistemas se expresan en formas de pregunta para analizar la información de los perfiles del usuario.

- Objetivo: Los sistemas de recomendación eliminan la información que no es importante de acuerdo con los perfiles de usuario.

- Ámbito Social: La relación con el usuario es importante para analizar los gustos y preferencias en el sistema.

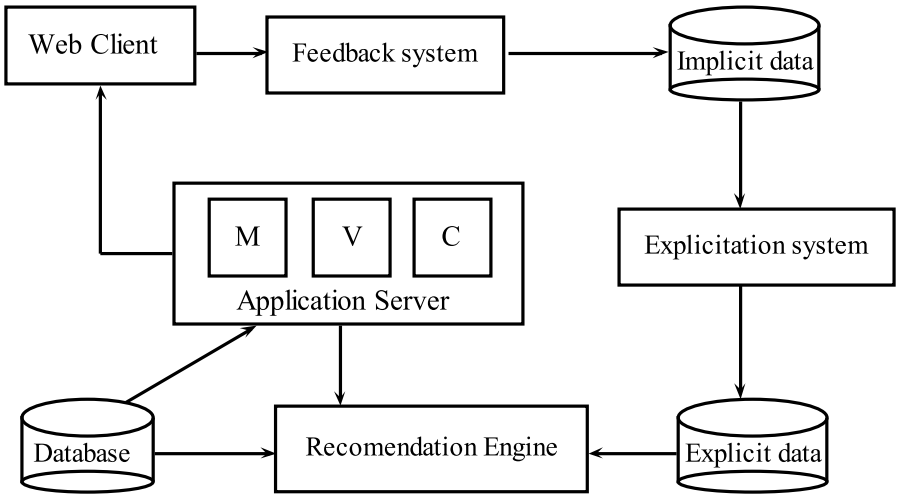


Figura 8: Modelo de sistema de recomendación para libros electronicos inteligentes [27].

Técnicas de recomendación

Los sistemas recomendador contrastan el perfil del usuario con las características necesarias para realizar una recomendación acertada en donde existen tres técnicas de filtrado que identifican el tipo de sistema que puede emplearse:

1. Retroalimentación explicita: La retroalimentación explícita consiste en que, a través del proceso de una encuesta, el usuario da una calificación de los contenidos. Esta retroalimentación permite conocer acertadamente los gustos del usuario permitiendo dar una calificación de los contenidos.

2. Retroalimentación implícita: La retroalimentación implícita evalúa sin intervención de los usuarios las posibles opciones de recomendación, ya sean por medio de películas, artículos web, libros, programas de televisión, y otros. Estos eventos permiten analizar y comprender los gustos y preferencias del usuario.

3. Retroalimentación mixta: La retroalimentación mixta incluye el balance de la retroalimentación explicita o implícita, donde se mezcla el uso de una encuesta para evaluar la calificación de un contenido principal, donde por medio de recomendaciones se fomenta la inclusión de contenidos que pueden ser del agrado para un usuario.

6.3 Marco Conceptual

6.3.1 Agente

Un agente es un sistema informático capaz de percibir su medioambiente con la ayuda de sensores y de actuar sobre dicho entorno mediante efectores (células nerviosas que ejecutan respuestas ante los estímulos que reciben). Un agente ideal racional debe hacer todo lo posible para maximizar el rendimiento de sus acciones, basándose en la evidencia proporcionada por sus sensores y en el conocimiento que disponga. Un agente es autónomo cuando sus acciones y elecciones dependen más de su propia experiencia que del conocimiento introducido sobre el entorno por el programador [44].

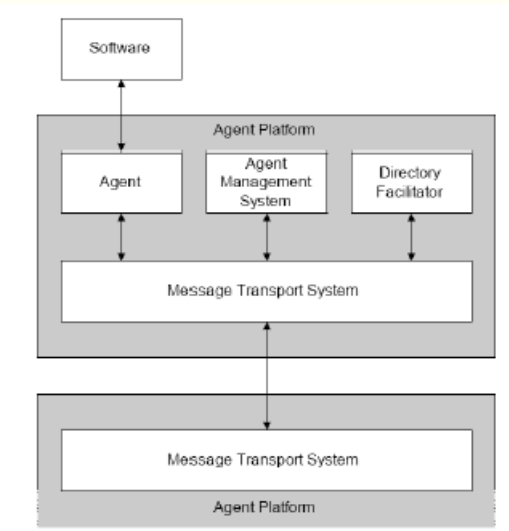


Figura 9: Diagrama de referencia de un agente según FIPA [41].

6.3.2 Red Neuronal

"Una red neuronal es un procesamiento distribuido masivamente paralelo que tiene una tendencia natural para almacenar conocimiento empírico y hacerlo disponible para el uso.

Esta definición de red neuronal del autor recuerda que en el cerebro hay dos aspectos:

1. El conocimiento se adquiere por la red a través de un proceso de aprendizaje.

2. Las conexiones interneuronicas, se conocen como pesos sinápticos y se usan para almacenar conocimiento" [45].

6.3.3 Multiagente

El sistema multiagente es una colección de múltiples agentes autónomos (inteligentes), cada uno actuando hacia sus objetivos mientras todos interactuan en un entorno com-

partido, pudiendo comunicar y posiblemente coordinar sus acciones.

Conjunto de agentes autónomos, generalmente heterogéneos y potencialmente independientes, que trabajan en común resolviendo un problema.

Un Sistema multiagente es un sistema distribuido de nodos o elementos de un sistema de inteligencia artificial, donde la conducta de esos resultados produce un resultado de un conjunto inteligente.

Un Sistema multiagente es un sistema computacional compuesto de múltiples agentes que interaccionan entre ellos donde cada agente tendrá sus propios objetivos y motivaciones y el éxito de la interacción requerirá de la cooperación, coordinación y negociación entre ellos [33].

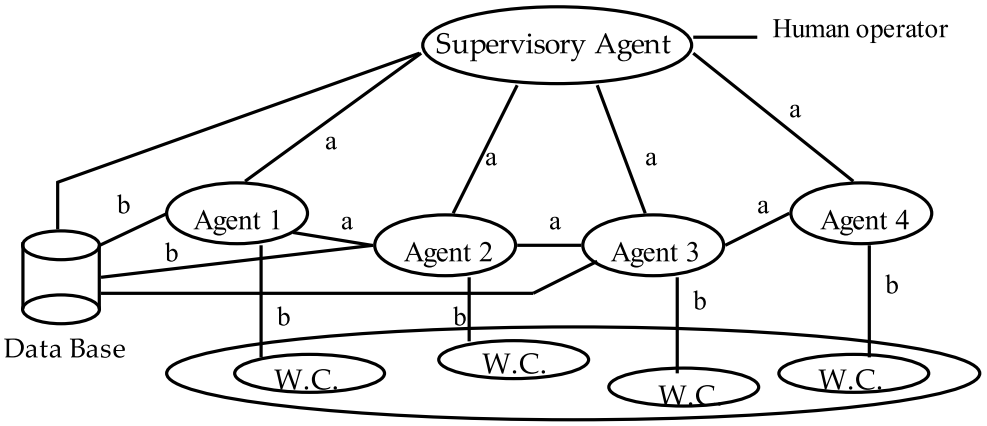


Figura 10: Ejemplo de punto de vista del modelado de un multiagente [45].

6.3.4 Arquitectura de Software

De acuerdo a Software Engineering Institute (SEI), La arquitectura de software se refiere a "las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos [47]."

6.3.5 Neurolab

Neurolab es una biblioteca para el desarrollo de redes neuronales en python. Contiene algoritmos de redes neuronales, algoritmos de entrenamiento de red y un framework que permite explorar y crear otro tipo de redes neuronales.

Este framework se caracteriza por utilizar python en estado puro, ademas del manejo de Numpy, también tiene un api de desarrollo idéntico al Neural Network Toolbox (NNT) de Matlab. Su manejo es flexible a la hora de configurar y entrenar los algoritmos, con un numero ilimitado de capas de red y neuronal que pueden uti-

lizarse en el desarrollo de algoritmo de aprendizaje.

Los tipos de redes neuronales que maneja son :

- Perceptrón de capa única
- Perceptrón de avance de alimentación multicapa
- Capa competidora (capa de Kohonen)
- Cuantización vectorial de aprendizaje (LVQ)
- Elman Recurrent network
- Red de Hopfield Recurrente
- Hemming Recurrent network

6.3.6 Numpy

NumPy es el paquete fundamental para la informática científica con Python. Contiene, entre otras cosas :

- Un poderoso objeto de matriz N-dimensional
- Funciones sofisticadas (difusión)
- Herramientas para integrar el código C / C ++ y Fortran
- Álgebra lineal útil, transformada de Fourier y capacidades de números aleatorios

Además de sus usos científicos , NumPy también se puede usar como un contenedor multidimensional eficiente de datos genéricos. Se pueden definir tipos de datos arbitrarios. Esto permite a NumPy integrarse de manera rápida y sin problemas con una amplia variedad de bases de datos. NumPy está licenciado bajo la licencia BSD , lo que permite su reutilización con pocas restricciones [48].

6.3.7 Gunicorn

Gunicorn 'Green Unicorn' es un servidor HTTP Python WSGI para UNIX. Es un modelo de trabajador pre-tenedor. El servidor de Gunicorn es ampliamente compatible con varios frameworks web, simplemente implementado, liviano en los recursos del servidor, y bastante rápido [49]

6.3.8 Flask

Flask es un micro framework web escrito en Python . Se clasifica como microframework porque no requiere herramientas o bibliotecas particulares. [50] No tiene una capa de abstracción de base de datos, validación de formularios ni ningún otro componente donde las bibliotecas de terceros preexistentes proporcionan funciones comunes. Sin embargo, Flask admite extensiones que pueden agregar características de la aplicación como si estuvieran implementadas en el Flask mismo. Existen extensiones para mapeadores relacionales de objetos, validación de formularios, manejo de cargas, varias tecnologías de autenticación abierta y varias herramientas relacionadas con marcos comunes.

6.4 Marco Metodológico

6.4.1 Tipo de estudio

El tipo de estudio del proyecto es de tipo descriptivo, donde se plantea el desarrollo del prototipo sistema recomendador basado en multiagentes, identificando las valoraciones y opiniones de los usuarios acerca de las colecciones de arte. Por medio del sistema multiagente se desea encontrar la recomendación "perfecta" que permita mostrar a los usuarios las obras de arte que sean gratificantes. También se analiza el comportamiento de los usuarios dentro del sistema que permita determinar la funcionalidad del sistema recomendador, es decir, que cumple con las exigencias para las cuales ha sido creado. El desarrollo de este proyecto accede a utilizar tecnologías que están a la vanguardia para representar eventos que pueden mostrarse al público de una manera diferente.

6.4.2 Unidad de Análisis

Personas que se encuentren entre los 17 y los 60 años.

6.4.3 Unidad de Estudio

Las variables del proyecto se manejan en rangos de calificación por parte del usuario, en este caso de 1 a 5 en la valoración a una obra de arte, en términos de tiempo se utilizo la duración en segundos. Estos parámetros manejados dentro del sistema recomendador se establecieron para posibilitar los resultados que arroje el sistema al momento de crear una nueva recomendación.

6.4.4 Unidad de Tiempo

El desarrollo de proyecto fue de 24 semanas.

6.4.5 Unidad Geográfica

La Ciudad de Bogotá.

6.4.6 Metodología de la investigación

En el proyecto se manejo una metodología cuantitativa que permita evaluar a través de los parámetros e indicadores del perfil del usuario. Todos estos datos se analizaron según la base de datos de usuario y base de datos de colecciones donde se genera una tabla y gráfica final que muestre la solución obtenida al momento de probar el sistema. También se utilizó una metodología cualitativa descriptiva para analizar la información y verificar la validación del resultado obtenido en la salida del programa.

6.4.7 Participantes

La población de las pruebas principalmente contará con estudiantes universitarios que cumplan los requisitos de la unidad de análisis en términos de edad, donde se estima el tiempo de dos semanas para la realización de las pruebas con los usuarios de la aplicación. El espacio de interés fue principalmente en la Universidad Distrital sede de ingeniería, también se analiza que la cantidad de personas de población es de aproximadamente 200 personas donde se tomo una muestra del 10% de la población de manera aleatoria que sirva para desarrollar la investigación.

6.4.8 Instrumentos y Equipos

Con el desarrollo de la aplicación, se realiza en una primera instancia una recolección de datos del usuario para el sistema recomendador por medio de una encuesta. Con una lista de todas las carreras de la Universidad de la Facultad de Ingeniería, se seleccionan los datos de mayor relevancia en la encuesta para crear la base de datos de conocimiento. Este proceso permitirá realizar pruebas iniciales al sistema recomendador para así gestionar ajustes para complementar el sistema multiagente.

6.4.9 Metodología de Ingeniería

El proyecto se desarrollará bajo la metodología de desarrollo de software SCRUM. Esta elección se basa en su gran aplicación y por su versatilidad, al ser una metodología ágil. Del mismo modo, por su simplicidad en implementación a través de la adición incremental de funcionalidades, reduce la complejidad del código, influyendo en una rápida corrección de los errores, un manejo de riesgos y, por ende, una reducción en los costos de desarrollo. A su vez, a través del desarrollo, se llevarán las inspecciones y adaptaciones que sean necesarias para llevar a cabo el proyecto.

6.4.10 Procedimiento de la Metodología

El proyecto se desarrollará en diez Sprint con un periodo de duración de 24 semanas con entregas de los avances del proyecto cada dos semana, hasta la finalización del Sprint de desarrollo.

6.4.11 Sprint de Inicio

El primer paso para el desarrollo del proyecto es la recolección de información de usuarios para la creación de la primera base de datos de conocimiento del sistema recomendador.

En este paso se realiza la búsqueda de datos a través de una encuesta, donde los primeros parámetros de recolección son los gustos, preferencias y tipos de coleccione preferidos en el momento de visitar un museo. Con el manejo de estos datos se fijan los parámetros que se pueden extraer para crear el sistema recomendador, donde posteriormente se genere un diagnóstico que permita identificar las entradas y salidas del prototipo. Por otra parte, se comienza a diseñar los agentes inteligentes del proyecto

los cuales tienen como propósito indagar y analizar la información de los usuarios y las colecciones para establecer la mejor recomendación de obras de arte en él museo.

Una vez realizado los pasos de los agentes se continua con la creación del diseño de la arquitectura del sistema multiagente la cual está compuesta por agentes inteligentes, estos agentes estarán desarrollados con SPADE, donde por medio del uso de redes neuronales se permitirá analizar la información que ha sido suministrada para encontrar una recomendación de colecciones para los usuarios.

6.4.12 Sprint de Elaboración

Este Sprint de elaboración se basa en los datos obtenidos durante el Sprint de inicio donde se analizará la primera arquitectura que se tiene del proyecto, se verificara la arquitectura y se elaborara un diseño general con su respectivo modelado UML. De la parte de almacenamiento se tiene una api la cual está configurada para la transaccionalidad de información de los datos del sistema. Esta base de información está preparada para el flujo de datos del sistema donde tendrá la capacidad de consumir los recursos informáticos necesarios para ejecutar las consultas necesarias del proyecto. Esta base de datos está gestionada a la medida para el manejo del sistema.

El Sprint de elaboración se basa principalmente en el diseño final de la arquitectura del software del sistema multiagente, también del diseño del servicio web y los recursos necesarios para acceder al api de la base de datos. El desarrollo de este Sprint es fundamental ya que de aquí se obtienen todos los diseños necesarios para construir el prototipo durante la etapa de construcción. En caso de tener un error o no completar el sprint se hará una reunión en donde se determine si se puede continuar a la siguiente fase del procedimiento de la metodología.

6.4.13 Sprint de Construcción

Esta etapa de desarrollo es fundamental para dar inicio a la construcción del prototipo, ya que es la etapa que surge de la creación y modelamiento de todo lo realizado para el sistema. El progreso de esta etapa contara de cinco Sprint de trabajo donde cada uno tiene un objetivo general para la creación del sistema. Al final de cada proceso de Sprint se realiza una reunión que permite identificar si el objetivo de la fase ha sido cumplido. Si el objetivo se cumple se procede a seguir con la siguiente etapa hasta finalizar el Sprint de Construcción.

Esta etapa de desarrollo se realizará según:

- Sprint 1: En esta primera parte de desarrollo se implementarán las conexiones necesarias para utilizar el api que contiene la base de datos del sistema, esta fue creada a partir de los datos obtenidos del tratamiento de información de la encuesta realizada durante el primer Sprint de trabajo.
- Sprint 2: En esta etapa se crean los agentes inteligentes del proyecto utilizando SPADE.

- Sprint 3: Después de crear los agentes inteligentes se comienza a gestionar la arquitectura multiagente, con sus propios estándares de comunicación y conexión para el desarrollo.

- Sprint 4: Se modifica los agentes del sistema recomendador del proyecto, para tener la interacción de las calificaciones de los usuarios donde se muestra las recomendaciones de contenido en el museo.

- Sprint 5: Esta etapa final se culmina con la unión del sistema recomendador y del sistema multiagente donde el objetivo es completar el diseño evaluado durante el Sprint de elaboración del proyecto.

Una vez finalizado los cinco sprint en la etapa de construcción se realiza una reunión final para verificar el objetivo de desarrollo de la fase de construcción. Se realizan las primeras pruebas del sistema , se analizan los resultados para verificar que la recomendación sea adecuada y se continua con el Sprint de Transición.

6.4.14 Sprint de Transición

Este Sprint tiene como objetivo realizar la documentación y la corrección de errores presentados por las pruebas e implementaciones del desarrollo del proyecto. También tiene como objetivo verificar y analizar los resultados obtenidos al probar el sistema recomendador y sistema multiagente.

Se estima que durante el desarrollo del segundo Sprint de trabajo se cree una retroalimentación del progreso de cada Sprint principal de la metodología donde se evalúen todos los eventos obtenidos durante las etapas. Estos procesos de retroalimentación permiten reconocer las posibles fallas que hallan el desarrollo creando así una opción de reparación de estos errores para la correcta continuación en el proyecto.

6.4.15 Herramientas

Este proyecto utilizo el paradigma orientado a objetos, con el lenguaje de programación Python, servidor web en heroku, la herramienta de desarrollo de agentes inteligentes SPADE, Neurolab y Matlab para el análisis y verificación de las redes neuronales.

Capítulo 2 : Análisis de Requerimientos

En el primer paso para desarrollar el proyecto, se definieron los requerimientos de uso del sistema, los casos de uso con su descripción y la visualización de estos. Todo como base para empezar el desarrollo de los agentes inteligentes y del sistema recomendador.

7 Desarrollo de los requerimientos

7.1 Requerimientos

En las reuniones dentro del sprint de inicio se tomaron en cuenta los actores que intervienen con el sistema, uno de ellos es el usuario. Este actor interactua con el Modelo de Visualización de la Aplicación gestionando el requerimiento de un recorrido recomendado. Otros actores son el sistema recomendador, el api de base de datos y el administrador.

En la tabla 1 se pueden ver los requerimientos encontrados para la aplicación, cada requerimiento tiene su nombre, descripción, prioridad y si es funcional o No.

ID	Nombre	Descripción	Prioridad	Funcional
1	Registrar puntuación del sistema	Este requerimiento nos permite otorgar la puntuación del sistema al recorrido visto por un usuario.	Medio	Si
2	Crear recorrido	Este requerimiento permite gestionar un recorrido de colecciones de arte para un usuario especifico.	Alto	Si
3	Registrar recorrido	Permite guardar el recorrido creado por el sistema recomendador.	Alto	Si
4	Obtener Recomendación	Permite crear la recomendación de obras de arte para luego crear el recorrido.	Alto	Si
5	Registrar Encuesta	Permite guardar los datos arrojados por la encuesta dentro de las preferencias del usuario.	Medio	Si
6	Buscar usuario	Este requerimiento permite obtener el usuario requerido para realizar una recomendación de obras de arte.	Medio	Si
7	Buscar Obras Arte	Este Requerimiento permite buscar las obras de arte seleccionadas para el recorrido.	Medio	Si
8	Buscar Favoritos	Permite buscar las obras favoritas que ha visto el usuario en los recorridos.	Bajo	Si

Tabla 1: Matriz de Requerimientos Parte 1.

.

Esta matriz de requerimientos permite realizar un análisis profundo de las necesidades del sistema, donde al hacer una comparación de los requerimientos entre ellos se puede analizar que requisitos son necesarios para la creación de los casos de uso. Para esta comparación se realizo una matriz donde se identificaron los requerimientos mas importantes. Estos son el de registrar puntuación del sistema, calificar recorrido,

9	Buscar Preferencias Usuario	Este requerimiento permite obtener las preferencias del usuario al momento de ver una obra de arte.	Medio	Si
10	Estado de la obra	Este requerimiento permite observar si la obra ha sido vista por el usuario.	Medio	Si
11	Registro Usuario	Este requerimiento permite agregar a un usuario dependiendo el tipo de logueo.	Medio	Si
12	Registro Obras Arte	Este requerimiento permite anexar las obras de arte necesarias para el desarrollo del proyecto, dentro de la base de datos.	Medio	Si
13	Obtener Obras más Vistas	Este requerimiento permite obtener las obras de arte mas vistas por los usuarios.	Medio	Si
14	Registrar Favoritos	Este requerimiento permite al usuario colocar obras favoritas que haya visto.	Medio	Si
15	Quitar Favoritos	Este requerimiento permite al usuario eliminar las obras favoritas que tenga seleccionadas.	Medio	Si
16	Autenticarse en el sistema	Este requerimiento permite al usuario loguearse dentro del sistema dependiendo del tipo de logueo.	Bajo	Si
17	Interfaz de Museo	Este requerimiento permite observar la interfaz grafica que se obtiene al mostrar el museo virtual al usuario.	Medio	No
18	Valorar Contenido	Este requerimiento permite dar una valoración por parte del usuario al contenido que ha observado.	Alto	Si
19	Tiempo en un contenido	Este requerimiento obtiene el tiempo del usuario al momento de observar una obra de arte.	Alto	Si
20	Calificar Recorrido	Este requerimiento permite dar una valoración del sistema al recorrido utilizado por un usuario.	Alto	Si
21	Registrar Preferencias	Este requerimiento permite al usuario anexar las preferencias para ver un recorrido.	Alto	Si

Tabla 2: Matriz de Requerimientos Parte 2.

valorar contenido y el tiempo en un recorrido. Estos datos dan a conocer los primeros casos de uso que se pueden obtener.

7.2 Casos de Uso

Al realizar un análisis exhaustivo de los requerimientos obtenidos para el sistema, se pudo evidenciar los requisitos que van a ser más utilizados. Por medio de este

análisis se determinaron los casos de uso de la aplicación, como se observan en la tabla 3, donde todos los actores del sistema mencionados anteriormente tienen un propósito para interactuar, por lo que los catorce casos de uso son importantes para ponerlos en practica en el desarrollo del prototipo. En la tabla 4 se muestra una breve descripción de cada caso de uso, con su respectivo identificador y actor que va a interactuar. Estos casos de uso son generales tanto para el sistema recomendador, como para el sistema mayor con realidad aumentada.

ID	Nombre
1	Registrar puntuación recorrido
2	Valorar un Contenido
3	Duración del usuario en Contenido
4	Ver Recorrido Recomendado
5	Ver obras más vistas
6	Ver obras Favoritas
7	Añadir obras favoritas
8	Eliminar obras favoritas
9	Registrar Usuario
10	Registrar Colección
11	Realizar Encuesta de Preferencias
12	Registrar Preferencias
13	Cambiar Preferencias
14	Estado de la obra

Tabla 3: Casos de Uso

ID	Nombre	Descripción	Actores
1	Registrar puntuación recorrido	El usuario sistema recomendador dará un resultado de 0 o 1 indicando si el recorrido realizado por un usuario es correspondiente a sus gustos y preferencias	Sistema Recomendador
2	Valorar un Contenido	El usuario a través de la interfaz del dispositivo da una calificación de 1 a 5 al contenido que está observando, la cual se guardara dentro del api de base de datos.	Usuario
3	Duración del usuario en Contenido	El modelo vista calcula el tiempo que ha durado un usuario observando la obra de arte.	Modelo vista, Usuario
4	Estado de la Obra	Este caso de uso permite identificar si el usuario ha visto una obra dentro del museo.	Usuario

5	Ver Recorrido Recomendado	Cuando el usuario desea realizar un recorrido en el museo el modelo vista realiza una consulta al sistema recomendador del recorrido correspondiente para un usuario, donde el sistema recomendador realiza una consulta a la base de datos para retornar las obras del usuario	Usuario, Api de Base de datos, Modelo vista, Sistema Recomendador
6	Ver obras más vistas	El modelo vista obtiene del api de Base de datos un numero limitado de las obras vistas por los usuarios.	Api de Base de datos, Modelo vista
7	Ver obras Favoritas	El sistema presenta al usuario las obras favoritas que han sido seleccionadas durante los recorridos en el museo.	Usuario, Api de Base de datos, Modelo vista
8	Añadir obras favoritas	El usuario añade a la colección las obras que le han sido de su agrado dentro del recorrido virtual.	Usuario, Api de Base de datos, Modelo vista
9	Eliminar obras favoritas	Este Requerimiento permite buscar las obras de arte seleccionadas para el recorrido.	Usuario, Api de Base de datos, Modelo vista
10	Registrar usuario	El usuario por medio de la interfaz grafica del modelo vista se registra dentro del sistema para poder realizar un recorrido virtual en el museo.	Api de Base de datos, Modelo vista
11	Registrar colección	Los administradores de la base de datos registran la información de las colecciones de arte dentro del servidor RDF.	Api de Base de datos
12	Realizar Encuesta de Preferencias	El usuario realiza la encuesta, la cual permite acceder a las elecciones de visualización que puede tener el usuario al interactuar con el sistema.	Usuario, Api de Base de datos, Modelo vista
13	Registrar Preferencias	Cuando el usuario realiza la encuesta el sistema registra los cambios realizados.	Usuario, Api de Base de datos, Modelo vista
14	Cambiar Preferencias	El usuario puede cambiar la configuración de las preferencias con el sistema.	Usuario, Api de Base de datos, Modelo vista

Tabla 4: Descripción de Casos de Uso

Estos casos de uso fueron los elegidos para representar el sistema, la figura 11 muestra el diagrama de casos de uso. En esta figura se pueden observar los actores usuario, sistema recomendador y administrador. Estos actores son los que intervienen en el desarrollo de la recomendación. El administrador es el encargado de suministrar las pinturas y esculturas a utilizar dentro de la base de datos, el sistema recomendador es el encargado de realizar el recorrido recomendado al usuario, también de manejar toda la información obtenida dentro del sistema. El usuario esta encargado de solicitar un nuevo recorrido, este puede ser un recorrido recomendado o favorito.

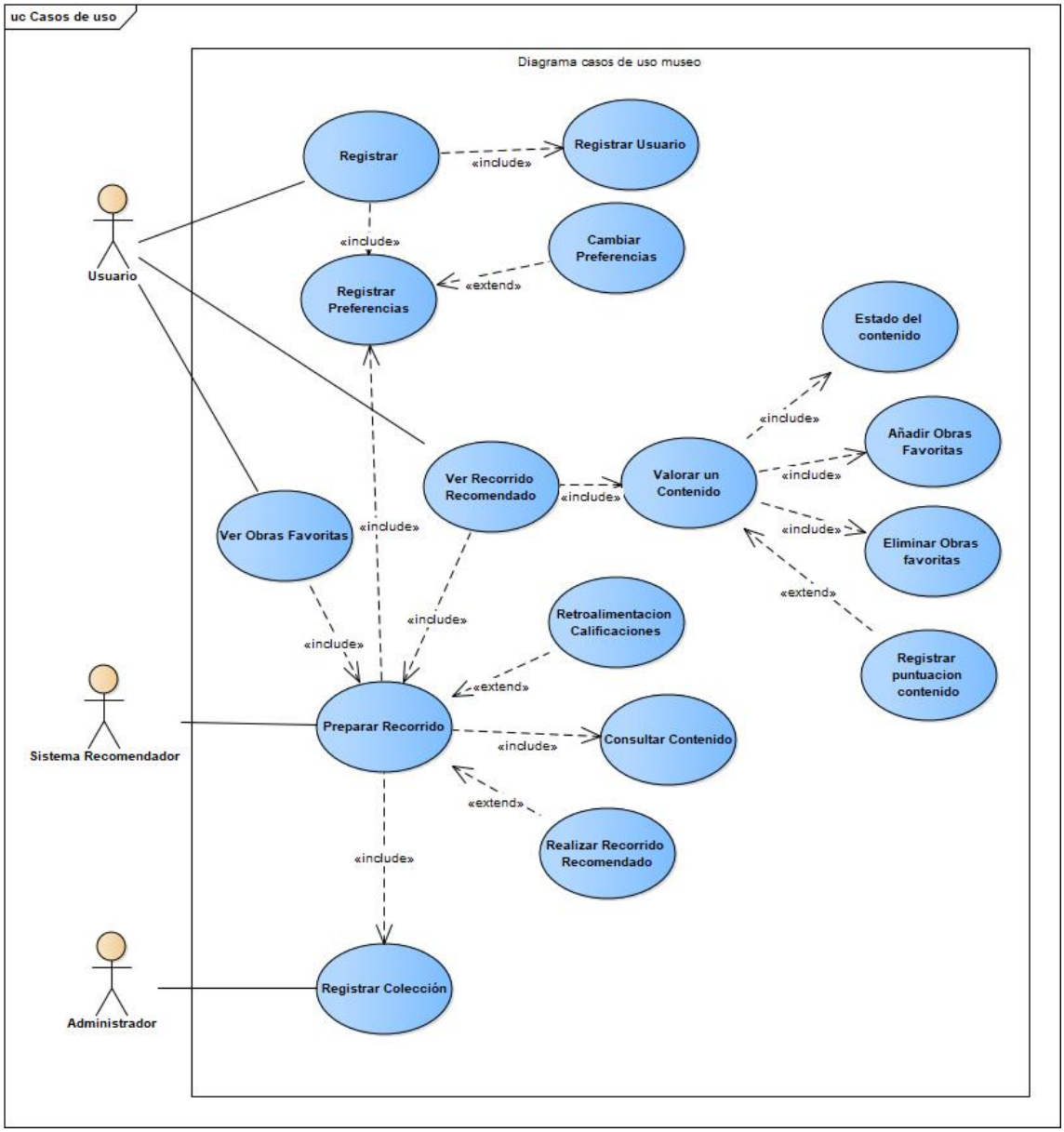


Figura 11: Diagrama de Casos de Uso. Fuente Propia.

Capítulo 3 : Definición de la arquitectura del sistema multiagente

En el desarrollo del prototipo del sistema multiagente, lo primero que se hizo fue determinar la arquitectura de los agentes inteligentes, para después crear la arquitectura general del sistema. Se crearon dos tipos de arquitectura multiagente, la primera sirvió como punto de partida del proyecto y base de un articulo científico, y la segunda como modelo fundamental como se determino que fuera el sistema. En este capitulo se muestra lo expuesto anteriormente y se formaliza el modelo final del prototipo.

8 Prototipo Sistema Multiagente

8.1 Estructura de los agentes inteligentes

Como primer paso de la metodología, es el desarrollo del modelo general del sistema multiagente. Lo primero que se plantea es la planificación de los agentes inteligentes que van a conformar el sistema. En este caso se determino utilizar un total de cinco agentes inteligentes, el agente de usuario, agente sistema, agente retroalimentacion, agente recorrido, agente recomendador. Cada agente tiene una función especifica para ayudar a crear un recorrido para un usuario, teniendo como resultado final su funcionalidad con el sistema multiagente.

8.1.1 Agente Usuario

El primer agente a desarrollar es el agente inteligente de usuario, este agente es el encargado de realizar la petición al sistema de un nuevo recorrido para un usuario, este puede ser de tipo recomendado o favorito. El agente crea la solicitud y espera a que el sistema entregue la respuesta. Internamente como se observa en la figura 12 este agente esta compuesto por un modulo de solicitud y respuesta el cual tienen los comportamientos predefinidos de recibir y entregar una solicitud.

El agente de usuario está a cargo de ingresar al sistema para solicitar una recomendación del usuario. En la plataforma de desarrollo de SPADE, el agente inteligente se designa con una ID y la dirección IP de la máquina que está trabajando, en este caso "usuario@127.0.0.1". Estos datos permiten conocer el usuario requerido para hacer el recorrido. Una vez que el agente de usuario de recomendación obtiene un formato de archivo JSON, muestra el resultado determinado por el sistema.

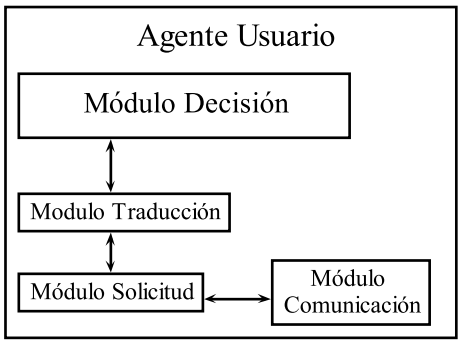


Figura 12: Arquitectura Agente Usuario. Fuente Propia.

8.1.2 Agente Sistema

El segundo agente es el agente de sistema, este agente tiene como responsabilidad el manejo de todos los agentes dentro del sistema, manejando las referencias de todos los agentes además del sistema de comunicación y el funcionamiento adecuado de como se esta creando un nuevo recorrido. El agente del sistema es un tipo de agente reactivo que tiene la función de obtener el identificador del usuario que requiere una nueva solicitud de recorrido, donde a través del estímulo este agente contribuye a la interacción de los otros agentes en su entorno.

El agente de sistema cuenta con un modulo de análisis de entorno, que evalúa el estado del usuario en el sistema, verificando si tiene un recorrido activo o necesita un nuevo recorrido. También tiene un modulo de decisión que permite interactuar con los agentes de retroalimentación y recorrido para obtener la información necesaria para realizarlo. Un modulo de comunicación necesario para llevar a cabo sus tareas de emprendimiento con otros agentes. La arquitectura de este agente expuesta anteriormente puede observarse en la figura 13.

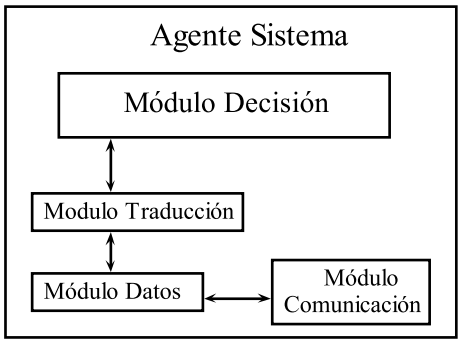


Figura 13: Arquitectura Agente Sistema. Fuente Propia.

8.1.3 Agente Retroalimentación

El agente de retroalimentación es el agente encargado de analizar la información de un recorrido realizado por un usuario, este es un agente de aprendizaje que se compone de un módulo de autoaprendizaje formado por la red neuronal, donde al tener la interacción de las calificaciones del usuario con las obras y el tiempo de visualización permitiendo realizar una retroalimentación con los datos para mejorar el aprendizaje del agente. Este agente analiza las calificaciones que da el usuario al sistema con una puntuación de 1 a 5, donde 1 representa el valor de puntuación más bajo y 5 el valor más alto. Estos resultados mas el tiempo de duración en la obra se analizan para verificar el progreso de la recomendación por parte del sistema, gestionando correcciones y reparaciones para mejorar los nuevos recorridos recomendados. El diagrama de arquitectura del agente se puede observar en la figura 14.

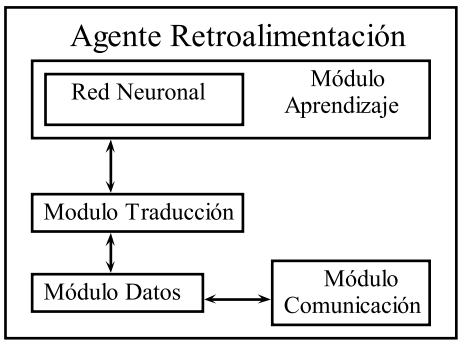


Figura 14: Arquitectura Agente Retroalimentación. Fuente Propia.

Para el desarrollo del agente inteligente de retroalimentación, se tuvo en cuenta el análisis del articulo [53] para crear la red neuronal, el diseño estructural de la red puede observase en la figura 15, donde la red neuronal esta compuesta por tres capas, con dos entradas y una salida. El análisis de las funciones obtenidas se mostraran en el capitulo de resultados.

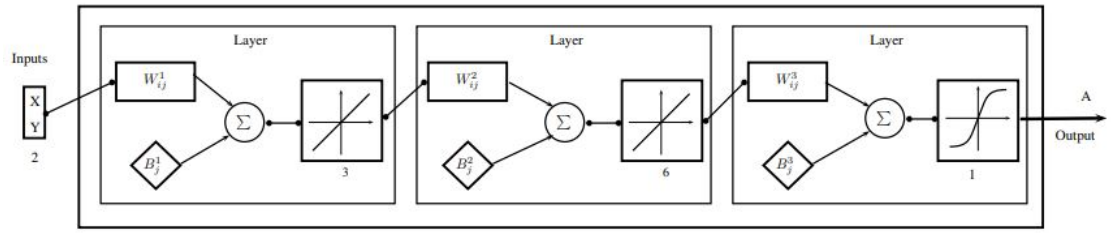


Figura 15: Estructura de la Red Neuronal. Fuente Propia.

8.1.4 Agente Recomendación

El agente de recomendación es un agente reactivo con un módulo de consultas que permite analizar los datos obtenidos por el agente de búsqueda, procesa la información que llega a la medición para concluir con la ruta recomendada para el usuario. Este agente procesa los datos del agente de usuario verificando nuevas coincidencias sobre los gustos y preferencias de otros usuarios, entregando una respuesta de la recomendación de un archivo del formato JSON.

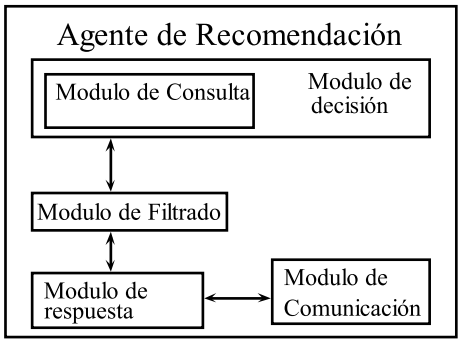


Figura 16: Arquitectura Agente Recomendación. Fuente Propia.

8.1.5 Agente Recorrido

El agente de recorrido es el agente inteligente encargado de unir los resultados obtenidos por los agentes de retroalimentación y recomendación a la hora de crear un nuevo recorrido recomendado para un usuario, donde por medio del modulo de consulta se gestiona y se almacena la información correspondiente a los datos generados para el recorrido del usuario, este agente se encarga de crear el recorrido favorito por medio de una consulta para después almacenarla. La estructura del agente se puede observar en la figura 17.

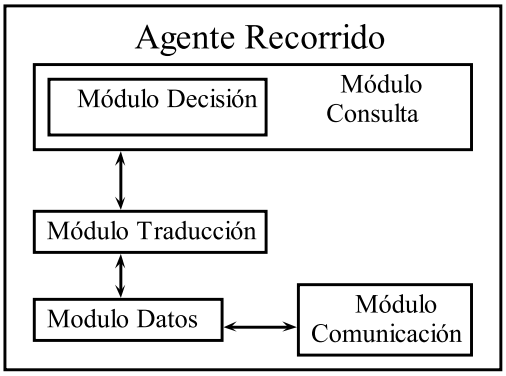


Figura 17: Arquitectura Agente Recomendación. Fuente Propia.

8.2 Arquitectura del Sistema Multiagente

Para el desarrollo de este prototipo, se utilizaron dos tipos de arquitecturas, la primera visualizada en la figura 18, se puede observar el primer modelamiento del sistema recomendador, con cinco agentes inteligentes, en este caso los agentes compuestos por, usuario, búsqueda, retroalimentación, recorrido y sistema. Esto se implemento como primer avance de desarrollo del sistema.

En la segunda etapa de desarrollo con la metodología scrum, se elaboro una mejora de la arquitectura, con base en el primer modelo realizado, este diseño se observa en la figura 19. Como no existe una guía determinada para realizar un tipo de arquitectura especifico, se realizo este diseño de multiagente con la necesidad de cubrir el proceso de recomendación de recorridos para una persona, por lo tanto, se crea este sistema que permite la elección de los diferentes agentes que existen en el prototipo. En este caso, el agente del sistema, el oficial de verificación, el agente de retroalimentación, que fue responsable de analizar las calificaciones de los usuarios para verificar si la ruta era del agrado, la recomendación y el agente de búsqueda, que tienen como objetivo la búsqueda de información precisa para llegar el objetivo de la recomendación. Esta fue la razón por la cual se desarrolló este tipo de sistema de agentes múltiples, debido a la necesidad de encontrar otra forma de realizar recorridos recomendados a los usuarios.

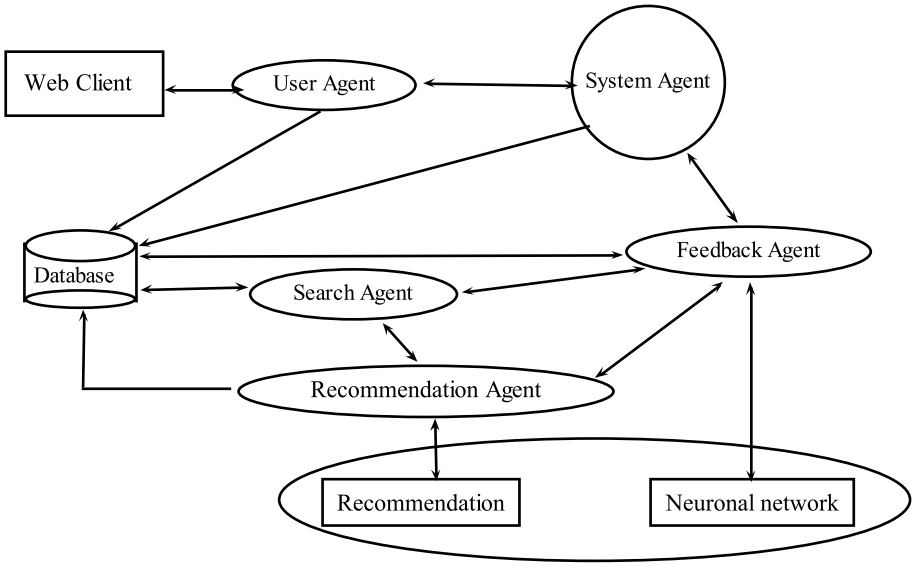


Figura 18: Prototipo del sistema multiagente v1 [53].

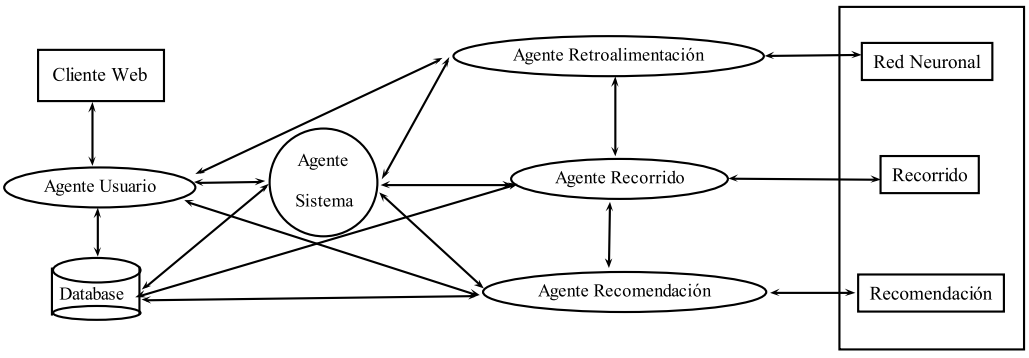


Figura 19: Prototipo del sistema multiagente. Fuente Propia.

Capítulo 4 : Modelamiento del sistema multiagente

El capítulo 4 del proyecto muestra el modelamiento del sistema recomendador basado en multiagente, donde se colocan los diagramas del sistema, diagramas de clases, actividades, secuencia, estados, interacción, infraestructura, componentes y nodos.

9 Diagrama de sistema

Una vez realizado el modelamiento arquitectónico del sistema, se procede a crear los diagramas del prototipo. El primer diagrama realizado es el de sistema, este se compone de un modulo de sistema multiagente, en el cual están alojados los agentes inteligentes de usuario, sistema, recorrido, retroalimentación y recomendación, comunicados con SPADE como servidor de aplicaciones de agentes inteligentes. Lo siguiente es el modelo de recomendación, este se compone de los dos tipos de conexiones a los apis de almacenamiento de datos de usuarios y obras de arte. En la figura 20 se muestra el diagrama de sistema elaborado, con los items comentados anteriormente.

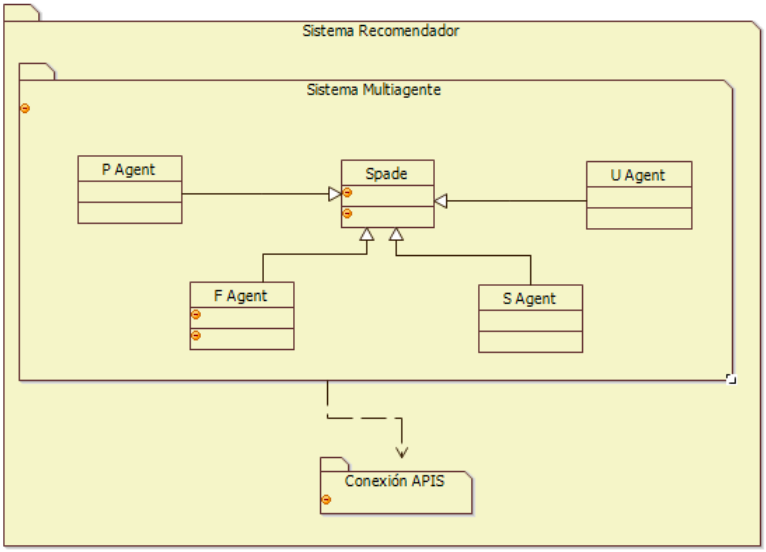


Figura 20: Diagrama de Sistema. Fuente Propia.

9.1 Diagrama de clases

Durante el desarrollo de los casos de uso se han analizado distintos tipos modelos que han sido candidatos para el proyecto. El diagrama de clases muestra todos los elementos colaborativos en el sistema recomendador, uno de ellos son los agentes que interactuan en el sistema, estos son los agentes de usuario, sistema, recorrido, recomendación y de retroalimentación. Estas clases permiten gestionar el recorrido recomendado para un usuario. Con cada agente se necesita un tipo de indicación que permita crear procesos para realizar una recomendación, por lo que existen las clases de recorrido, red neuronal, entre otros, los cuales permiten amortizar el funcionamiento del sistema para que cumpla el objetivo de crear un recorrido.

En la figura 21 se puede observar el diagrama de clases definido para el sistema con los elemento comentados anteriormente, también los tipos de conexiones, ademas de los elementos que hay internamente, como métodos y atributos.

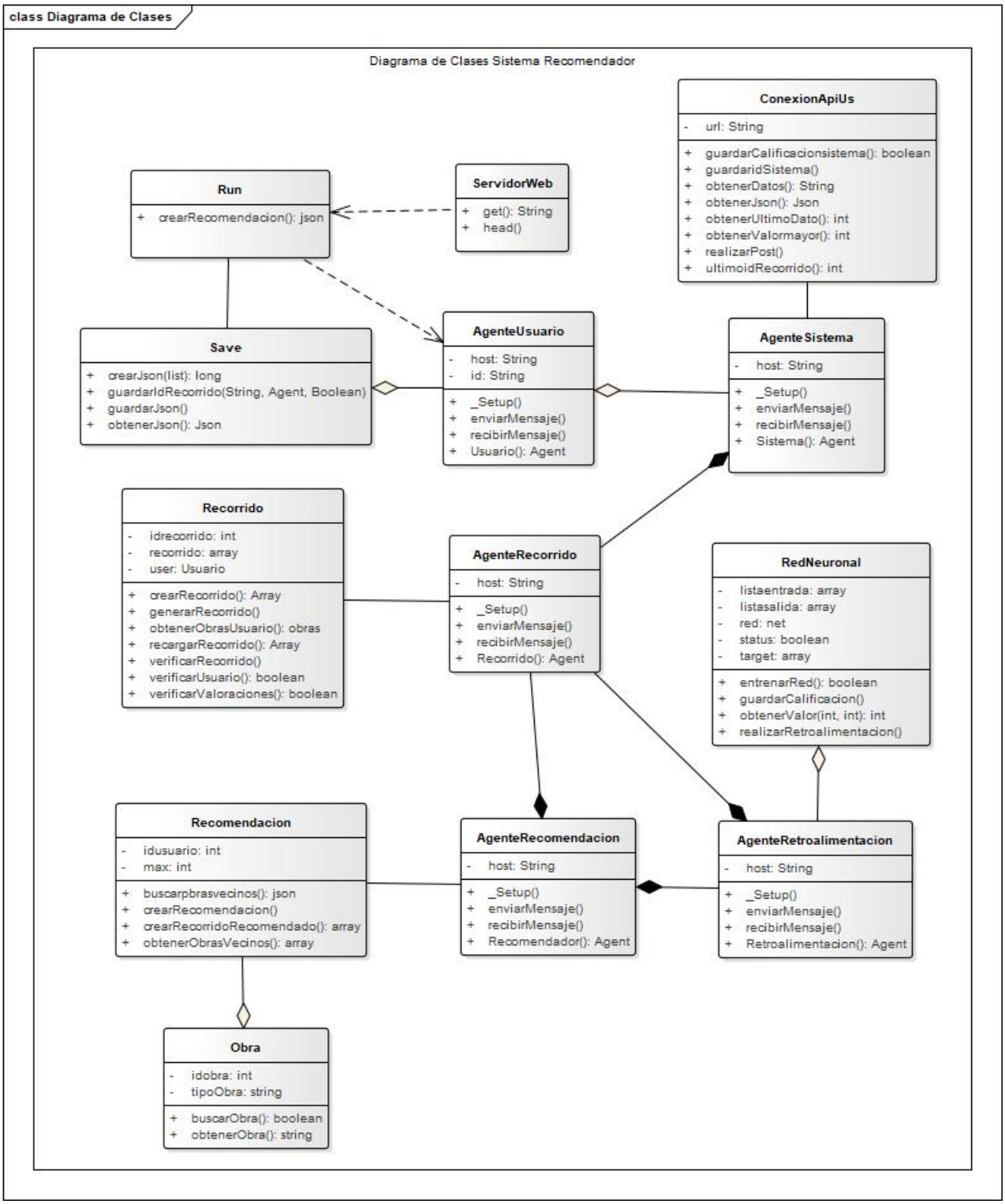


Figura 21: Diagrama de Casos de Uso. Fuente Propia.

9.2 Diagrama de Actividades

Cada clase tiene un Diagrama de actividad, este diagrama permite ver el comportamiento de los casos de uso, desde un punto inicial hasta el punto de finalización, en este caso se muestra los diagramas de actividad de registrar usuario, preferencias, valoración de recorrido y recorrido.

9.2.1 Registrar Usuario

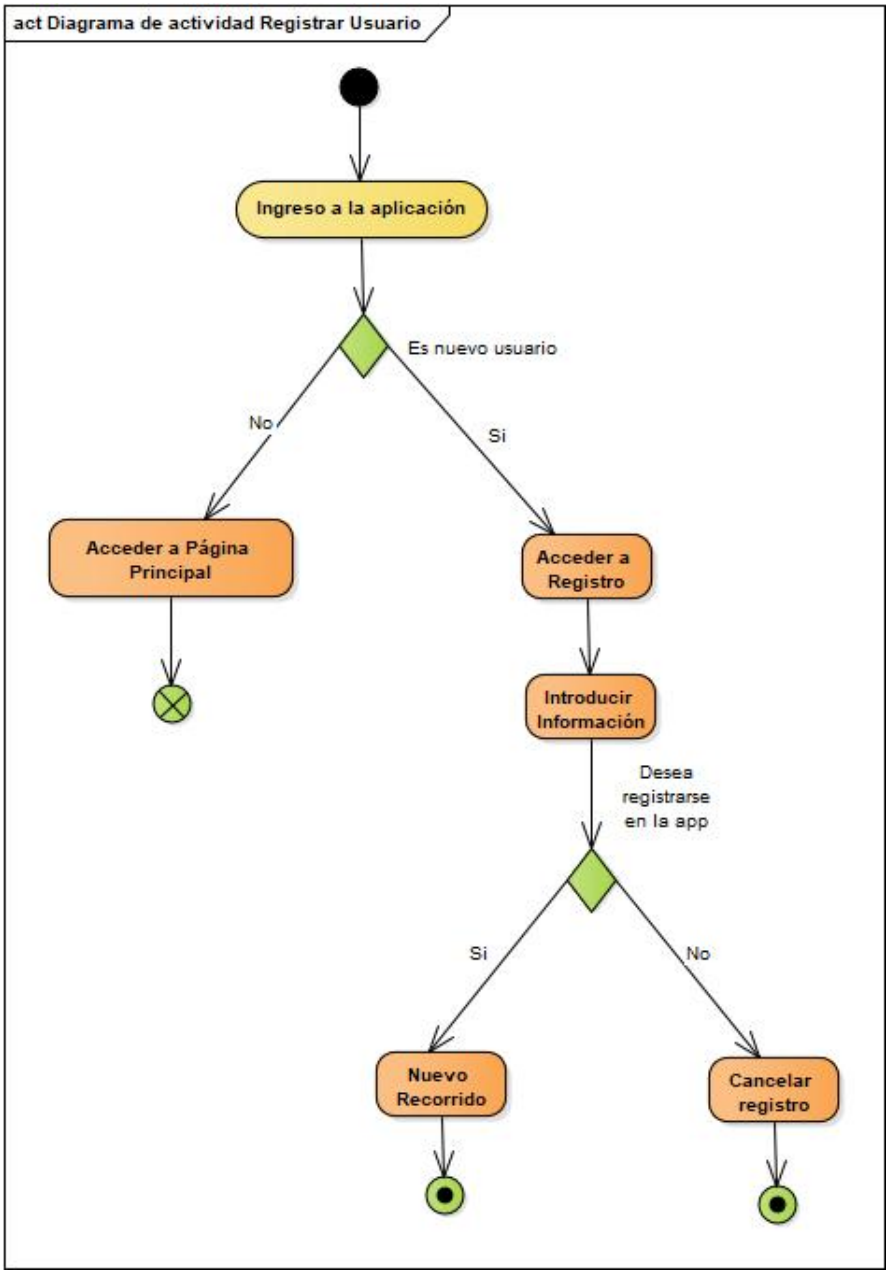


Figura 22: Diagrama de actividades Registrar Usuario. Fuente Propia.

9.2.2 Registrar Preferencias

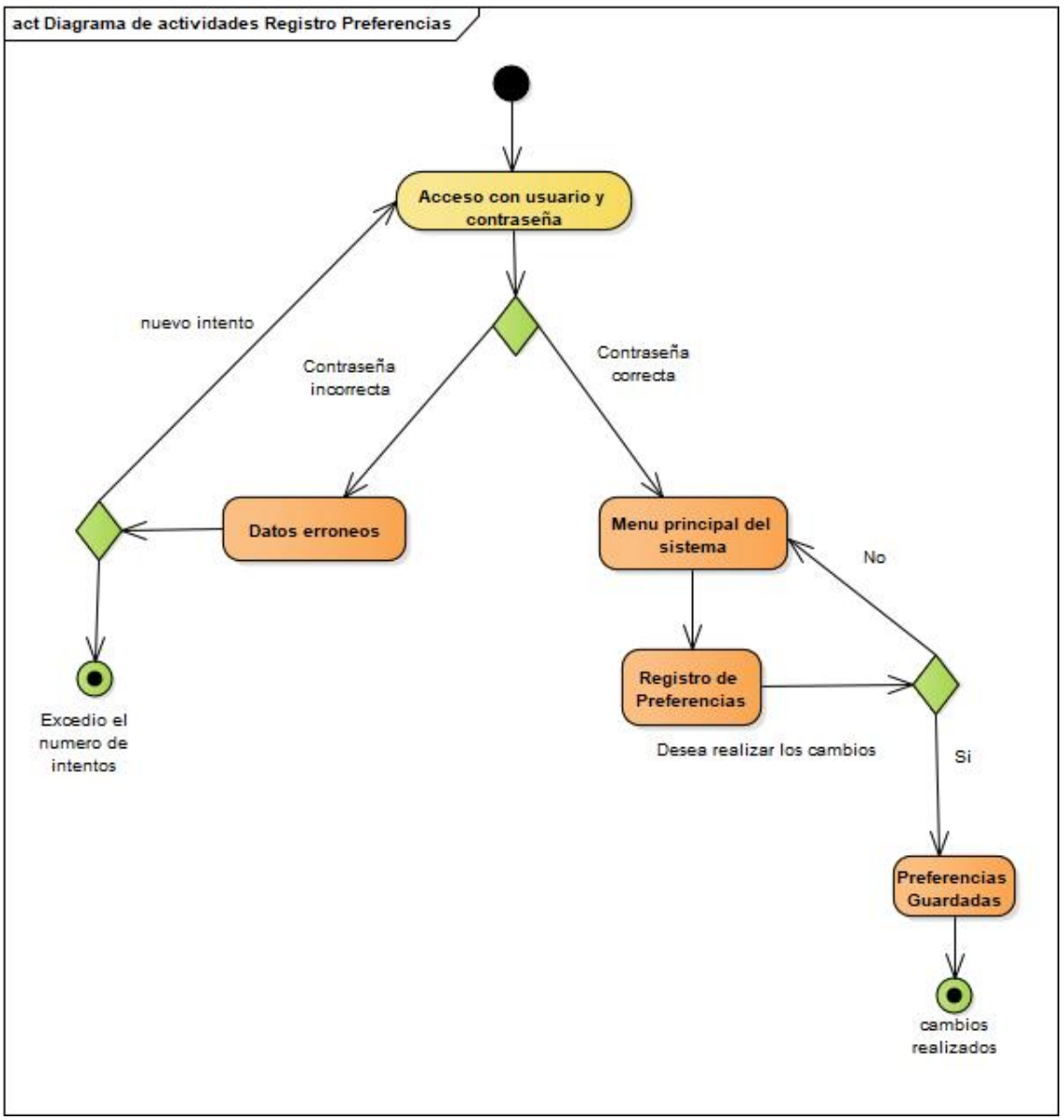


Figura 23: Diagrama de actividades Registrar Preferencias. Fuente Propia.

9.2.3 Recorrido

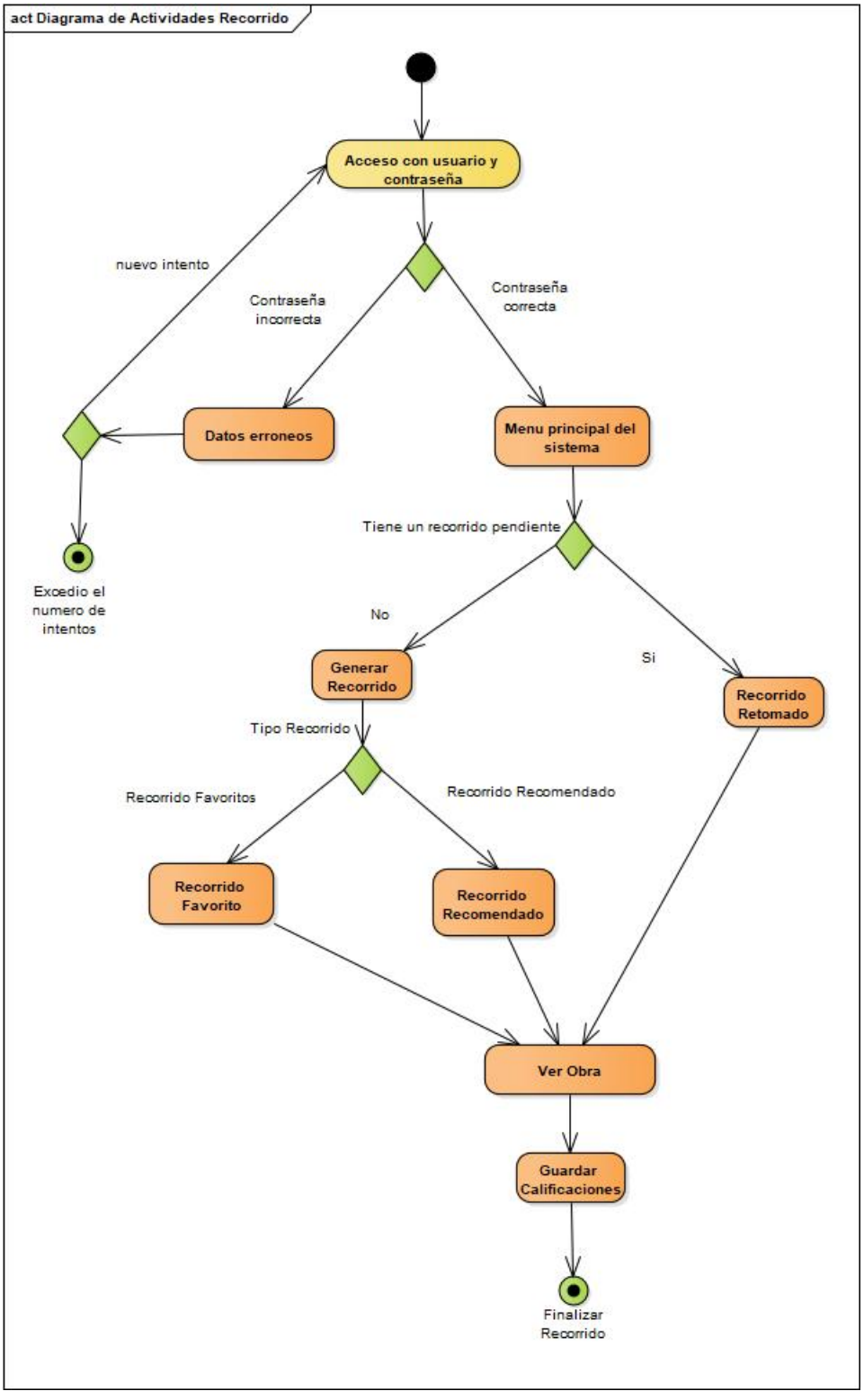


Figura 24: Diagrama de actividades Recorrido. Fuente Propia.

9.2.4 Valoración de Contenido

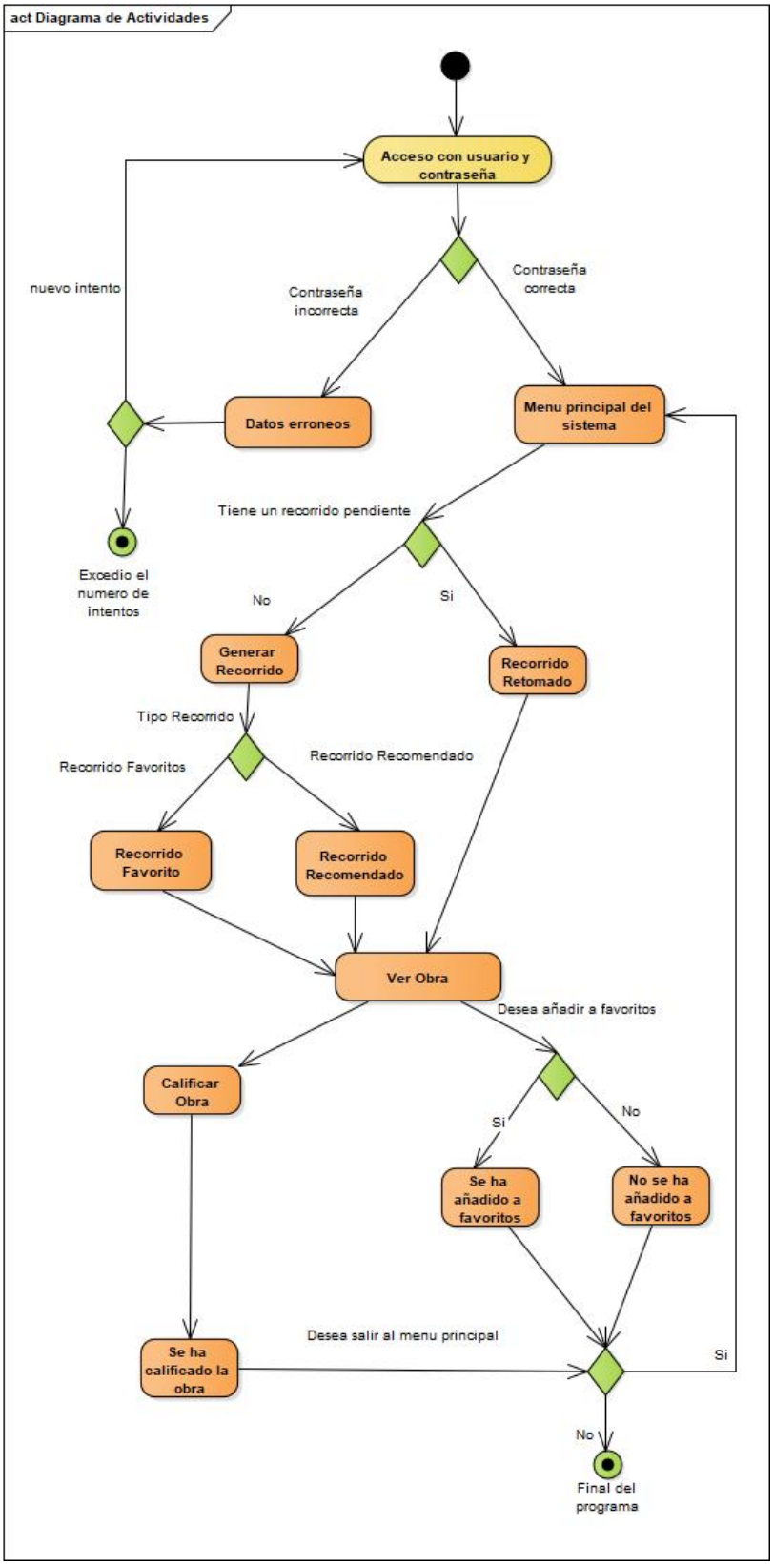


Figura 25: Diagrama de actividades Valoración de Contenido. Fuente Propia.

9.3 Diagramas de Secuencia

Los diagramas de secuencia muestran los mensajes entre las instancias de las clases, componentes, subsistemas y actores. En los siguientes diagramas se muestran las relaciones que hay entre los distintos casos de uso.

9.3.1 Valoración de Contenido

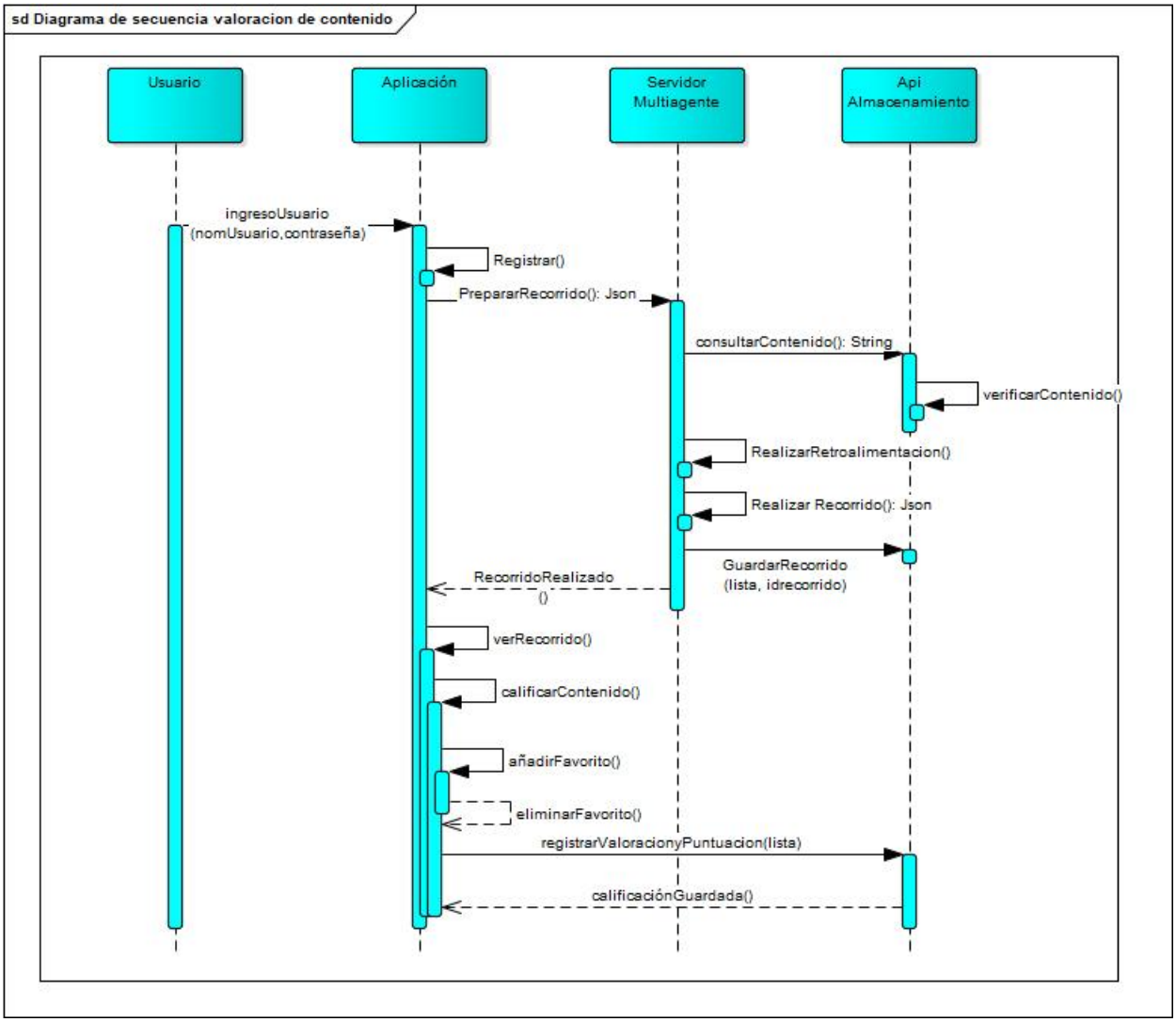


Figura 26: Diagrama de Secuencia Valoración de Contenido. Fuente Propia.

9.3.2 Registrar Usuario

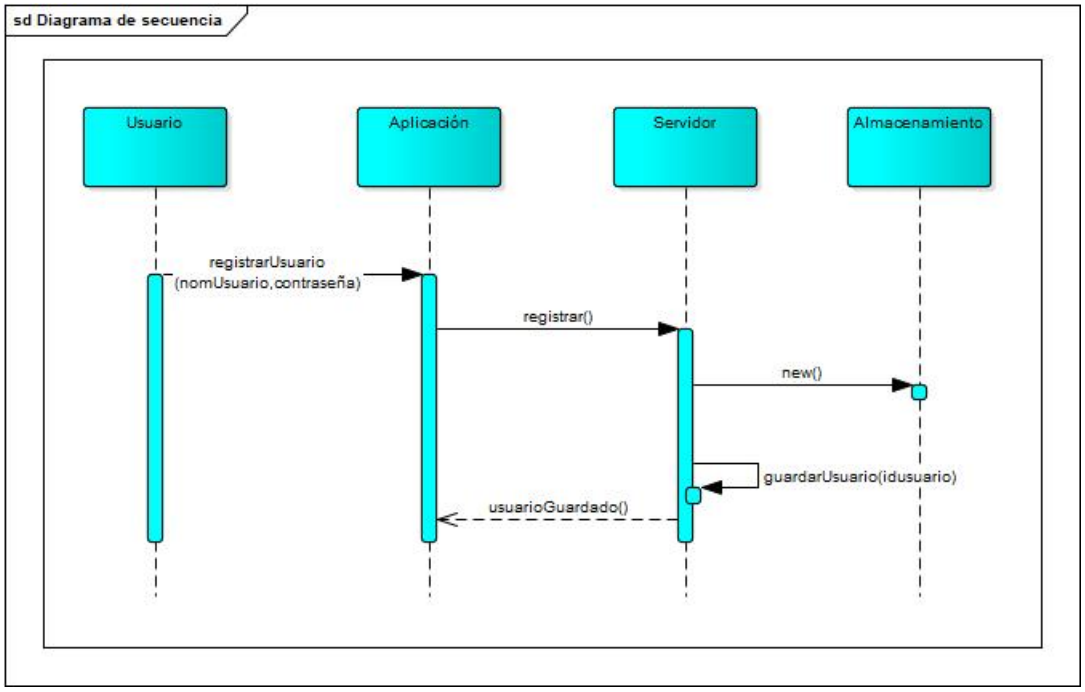


Figura 27: Diagrama de Secuencia Registrar Usuario. Fuente Propia.

9.3.3 Registrar Preferencias

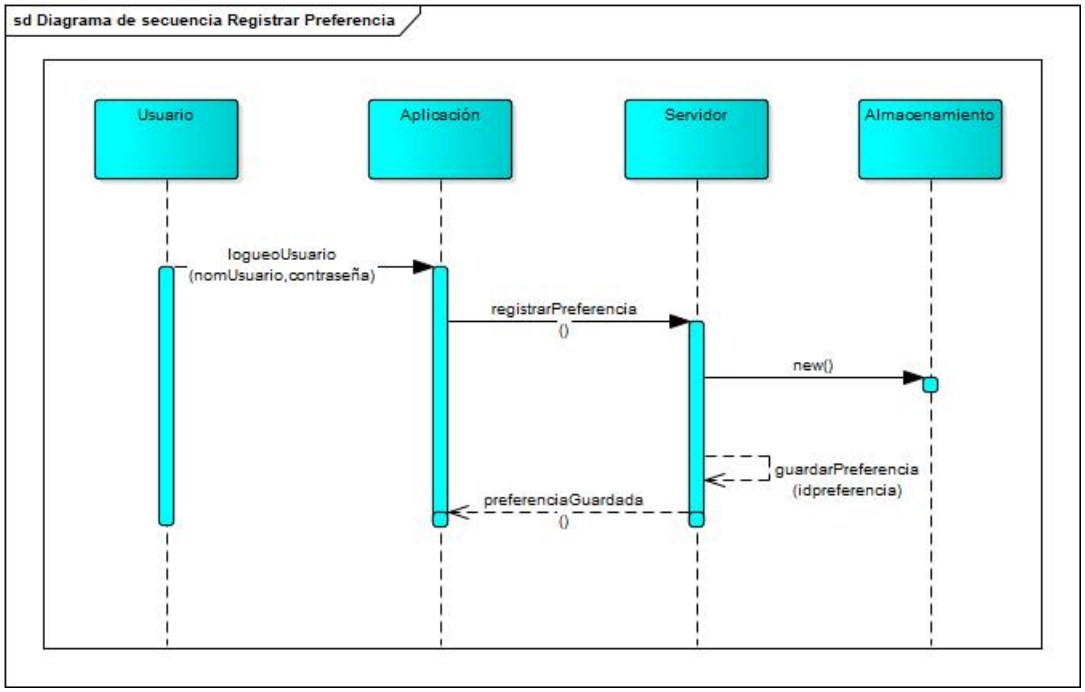


Figura 28: Diagrama de Secuencia Registrar Preferencias. Fuente Propia.

9.3.4 Recorrido

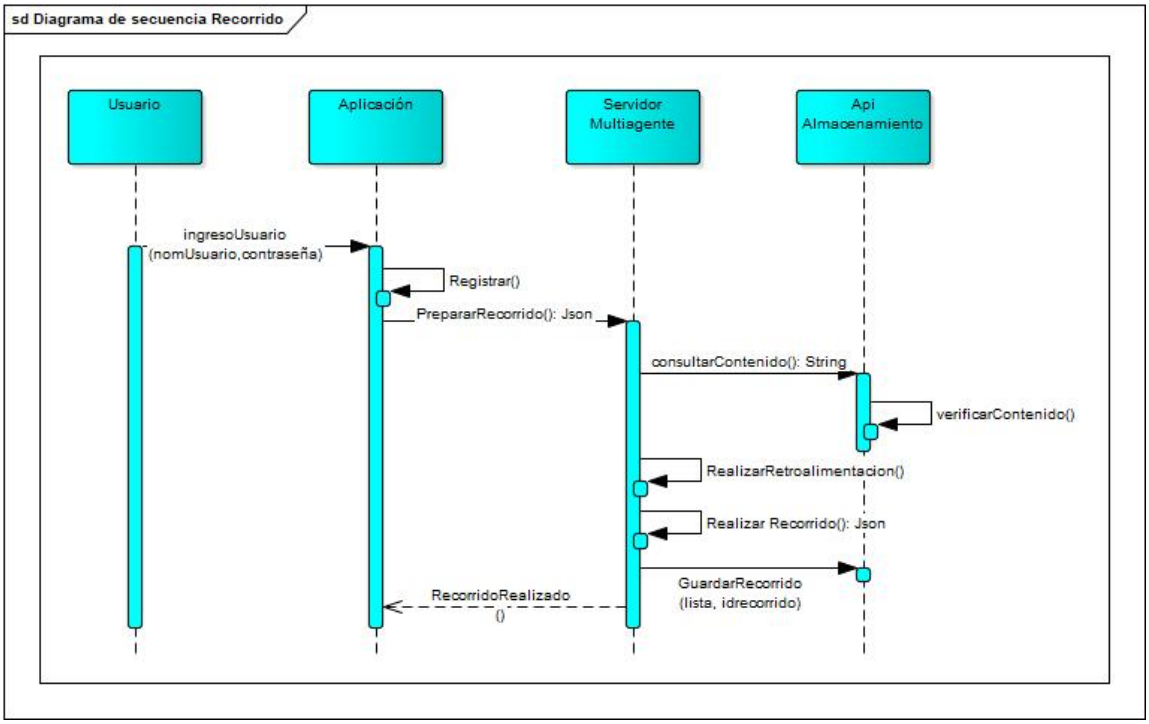


Figura 29: Diagrama de Secuencia Recorrido. Fuente Propia.

9.4 Diagramas de Estados

Los diagramas de estados muestran los comportamientos entre distintos objetos, estos diagramas se muestran posteriormente continuando con la progresión de los casos de uso utilizados en el proyecto.

9.4.1 Valoración de Contenido

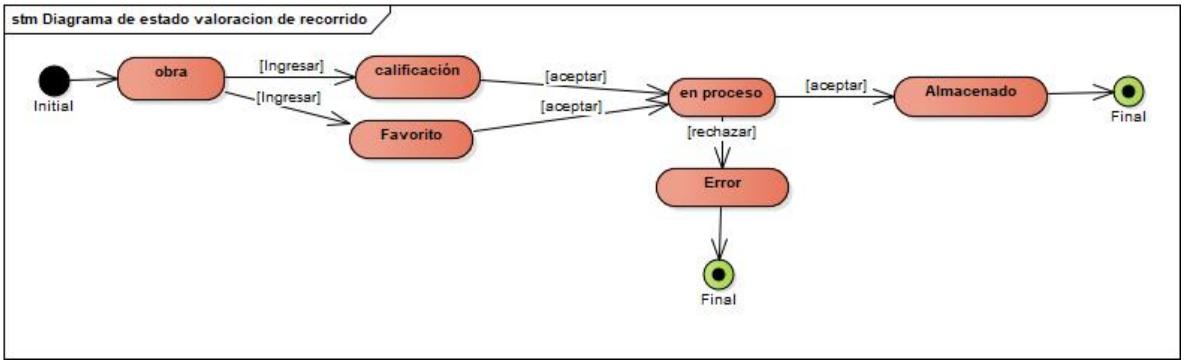


Figura 30: Diagrama de Estados Valoración de Contenido. Fuente Propia.

9.4.2 Registrar Usuario

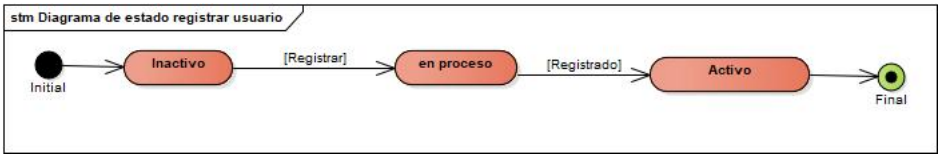


Figura 31: Diagrama de Estados Registrar Usuario. Fuente Propia.

9.4.3 Registrar Preferencias

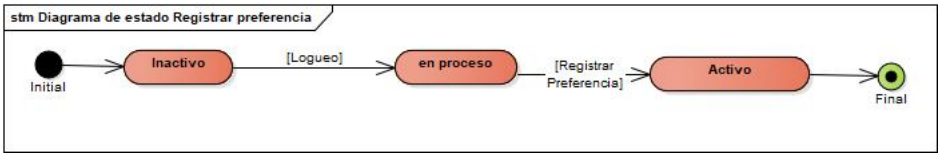


Figura 32: Diagrama de Estados Registrar Preferencias. Fuente Propia.

9.4.4 Recorrido

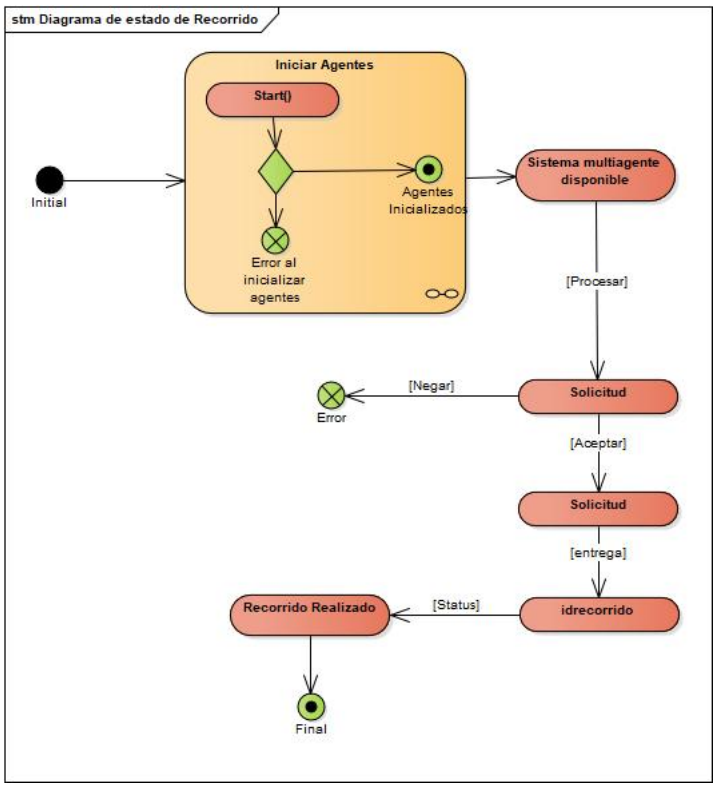


Figura 33: Diagrama de Estados Recorrido. Fuente Propia.

10 Diagramas de Interacción de Agentes

Los diagramas de interacción muestra las comunicación de los agentes inteligentes con su entorno. En esta sección se mostrara los cinco agentes inteligentes con sus respectivas llamadas y respuestas que pueden ser ejecutados en el sistema. Estos diagramas se muestran en las figura 34, donde se delimita el tipo de comunicación que tiene y las salidas hasta entregar el identificador de recorrido.

10.1 Diagrama de Interacción de los agentes inteligentes

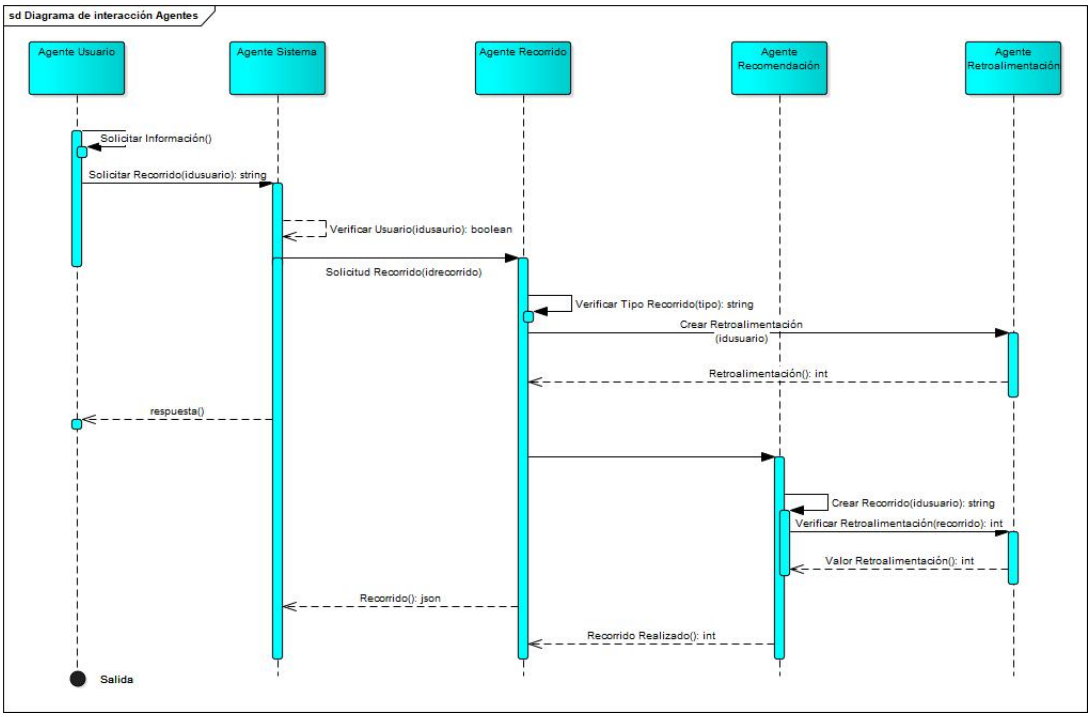


Figura 34: Diagrama de Interacción de los agentes inteligentes. Fuente Propia.

11 Diagramas de la aplicación

Los diagramas expuestos anteriormente muestran el desarrollo del prototipo multiagente. Como se ha comentado en la introducción, este trabajo forma parte de un prototipo general. En esta sección se mostrara el diagrama de nodos, infraestructura, componentes y punto de vista de la aplicación. En ellos se puede observar la conexión de los tres prototipos: El sistema de recomendación, apis de base de datos y la aplicación de realidad aumentada.

11.1 Diagrama de Componentes de la aplicación

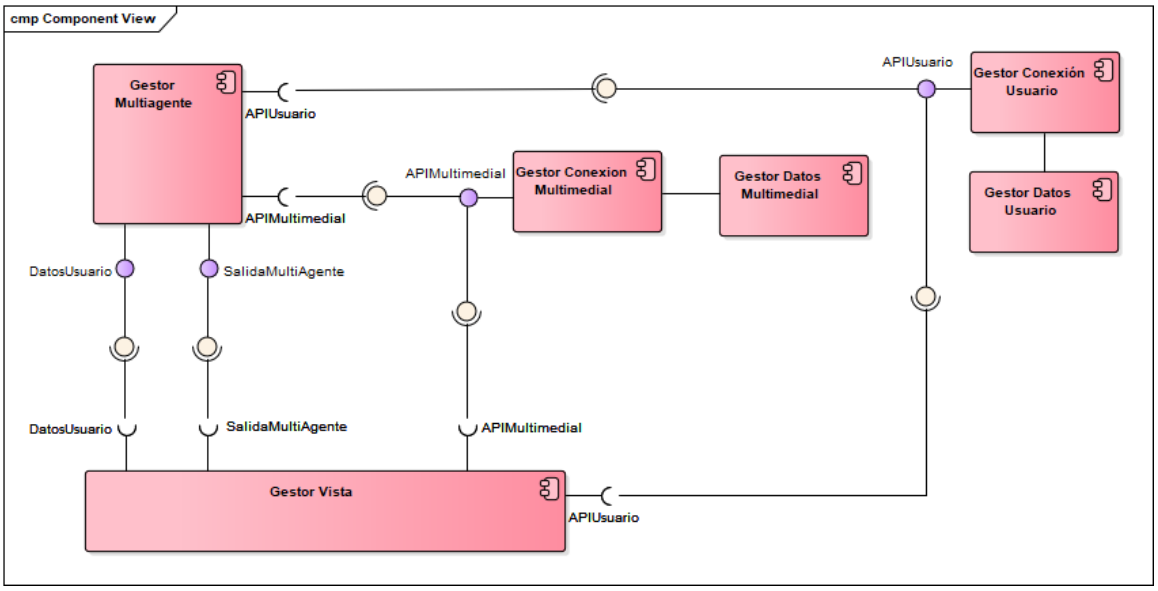


Figura 35: Diagrama de Componentes Recorrido. Fuente Propia.

11.2 Diagrama de infraestructura de la aplicación

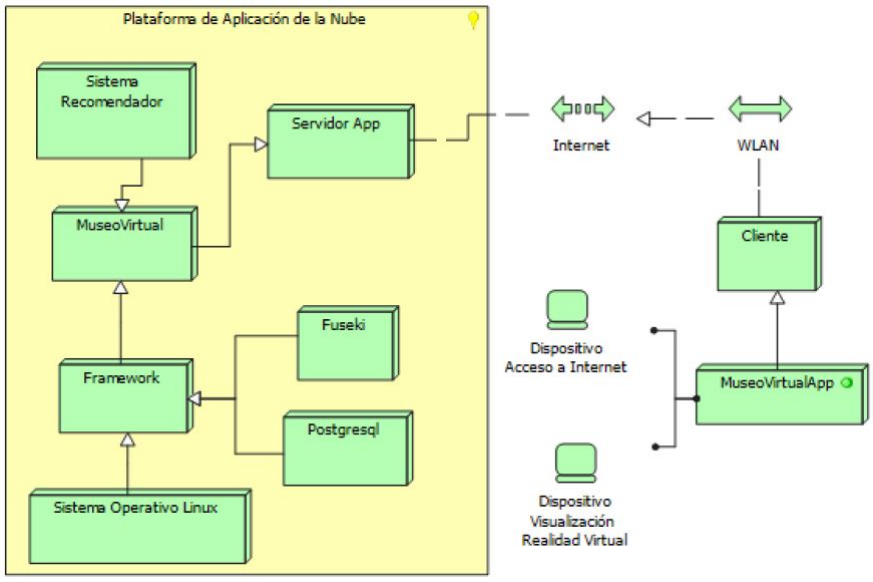


Figura 36: Diagrama de infraestructura. Fuente Propia.

11.3 Diagrama de Nodos de la aplicación

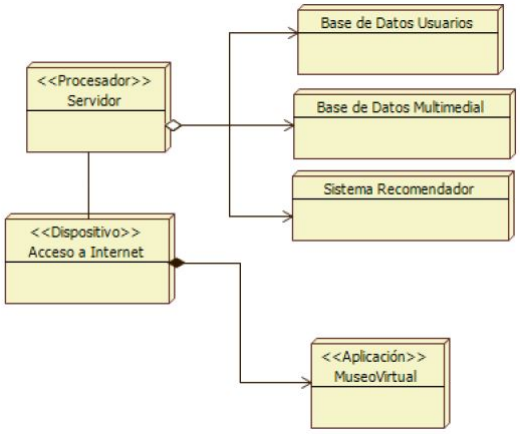


Figura 37: Diagrama de Nodos. Fuente Propia.

11.4 Diagrama punto de vista de cooperación

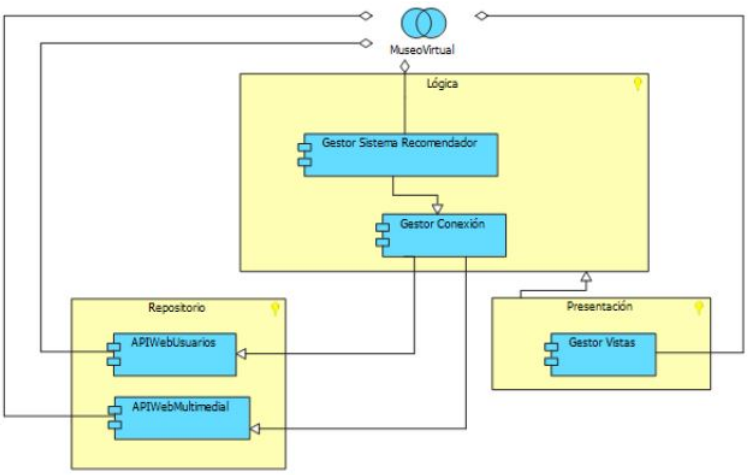


Figura 38: Diagrama de Componentes Recorrido. Fuente Propia.

Estos cuatro diagramas expuestos anteriormente dan una vista perspectiva del prototipo final del proyecto, la cooperación entre los sistemas y la visualización general a lo que se quiere llegar a mostrar con el proyecto.

Capítulo 5 : Conexión del Aplicativo

Este capítulo de conexiones, muestra los dos aplicativos de base de datos utilizados para el almacenamiento y uso de la información necesarios para el desarrollo del proyecto. También se muestra los links de conexión necesarios para conectar en el api de usuarios y api multimedial, la una alojada en herokuapp y la otra en aws (Amazon Web Service).

12 Conexión Base de Datos

12.1 Conexión Base de Datos Usuario

Una de las razones de un buen funcionamiento del sistema recomendador, es la utilización de una fuente de almacenamiento que permita la transaccionabilidad de peticiones en un tiempo determinado. Para el desarrollo de este proyecto se ha utilizado una base de datos, creada por estudiantes el cual permite realizar peticiones de tipo get para obtener información necesaria para gestionar una recomendación.

En la figura 39 se muestra la estructura de la base de datos de usuarios, esta es de tipo SQL con un total de seis tablas, la tabla de ObraVistaRecorrido es la tabla donde se almacena la información del usuario con la obra, contiene dos columnas fundamentales que son la de calificación y tiempo de visualización. Estos valores son lo que se utilizan dentro del agente de retroalimentación y que son fundamentales para la creación de un nuevo recorrido recomendado.

La tabla de RecorridoSistemaRecomendador es la encargada de almacenar la información del identificador de recorrido del sistema, donde por medio de la calificación del sistema se guarda un valor generado por el agente de retroalimentación, el cual sirve para mejorar un recorrido futuro para el usuario.

La Tabla de Preferencia es donde se almacena la información predeterminada de gustos de un usuario. Por medio de esta información el sistema recomendador gestiona el recorrido recomendado idóneo para el usuario que valla a usar el aplicativo.

La tabla tipo de Preferencia contiene las siete configuraciones del usuario dentro del sistema, ya sea el numero de obras, el tipo de movimiento o técnica de pintura o escultura, el valor de la precisión de la recomendación y la calificación de visita del usuario a los museos.

Para la conexión y comunicación de la base de datos, se encuentra el uso de heroku, donde por medio del link `http://api-usuarios-museal.herokuapp.com` se puede observar la pantalla de bienvenida de la aplicación. Una vez se tiene el link, se tienen preparadas las consultas necesarias para obtener la información del aplicativo por medio de una petición al servidor de tipo Get. Por medio del link `http://api-usuarios-museal.herokuapp.com/api/tipopreferencia` se puede observar en la figura 40 la salida del json de la aplicación de base de datos, en este caso se muestran las preferencias del sistema y su correspondiente identificador.

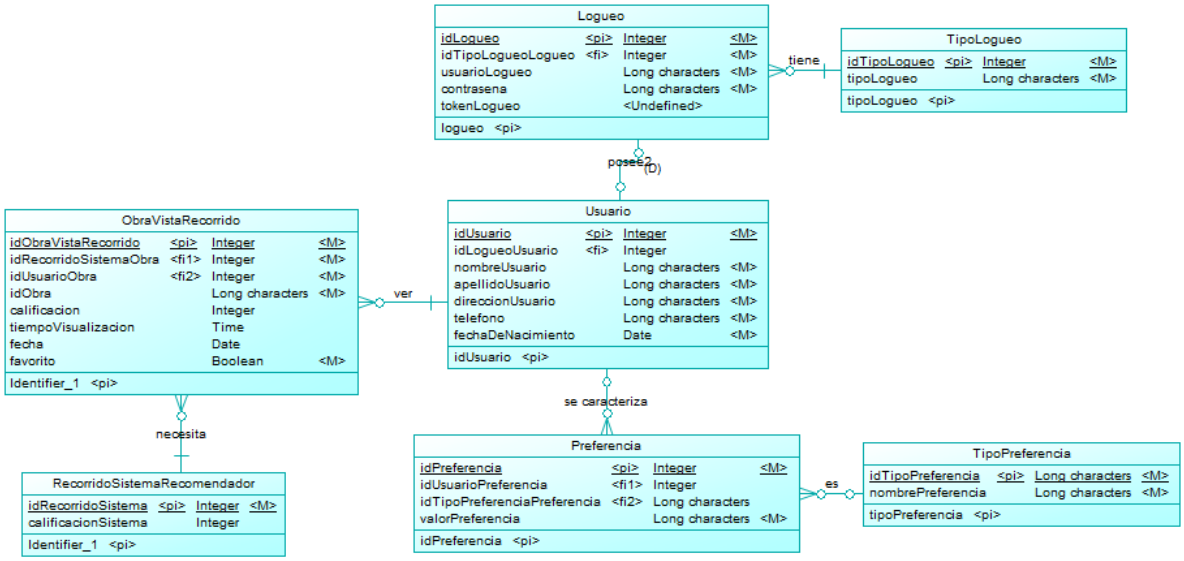


Figura 39: Modelo de la base de datos usuario [6].

```
{"status":"success","data[
{"idtipopreferencia":1,"nombrepreferencia":"Numero de Obras"},
{"idtipopreferencia":2,"nombrepreferencia":"Precision de la recomendacion"},
{"idtipopreferencia":3,"nombrepreferencia":"Frecuencia de visita al museo"},
{"idtipopreferencia":4,"nombrepreferencia":"Movimiento pintura"},
{"idtipopreferencia":5,"nombrepreferencia":"Tecnica pintura"},
{"idtipopreferencia":6,"nombrepreferencia":"Movimiento escultura"},
{"idtipopreferencia":7,"nombrepreferencia":"Tecnica escultura"}],
"message":"Retrieved ALL Tipo de Preferencia"}
```

Figura 40: Respuesta de la petición Api Base de datos usuario [6].

12.2 Conexión Base de Datos multimedial

Existe el api de la base de datos multimedial, que permite el almacenamiento de las obras de arte. En el desarrollo del prototipo se utiliza este api para acceder y manejar las pinturas y esculturas. Para la conexión y comunicación con la base de datos se realiza de una manera idéntica a la anteriormente expuesta, con un servidor web que entrega un json con la descripción de una obra de arte. Por medio del link http://ec2-18-222-128-51.us-east-2.compute.amazonaws.com se puede acceder a la página principal del servidor, también se pueden ver las pinturas y esculturas solo con /api/pinturas o esculturas.

Al ser una base de datos con ontologias, es complicado obtener un diagrama sencillo de la aplicación, lo que si se puede observar en la figura 41 un ejemplo de salida de una pintura utilizando el enlace http://ec2-18-222-128-51.us-east-2.compute.amazonaws .com/api/pinturas. Otros detalles que pueden observase en la figura, son el uso de ontologias para el almacenamiento de la información, para muestra de ello los datos de titulo, fuente , nombre, técnica entre otros, son útiles para ver la información de las obras. Todos los datos utilizados se almacenan en la base de datos de usuario permitiendo la unión de las obras con los datos de usuario, permitiendo su manejo dentro del sistema recomendador.


```
1  {"fuente":"https://www.metmuseum.org/art/collection/
     search/435683?rndkey=20180705&offset=0&rpp=20&pg=1
     &pos=4&ft=*&ao=on&high=on"},
2  {"title":"Island_of_the_Dead","lenguaje":"Ingles"},
3  {"nombre":"Arnold_Bocklin","fechaNacimiento":null,"
     pais":null},{"nombrePintura":"Island_of_the_Dead",
     "tecnicaPintura":"Oleo_sobre_madera"},
4  {"periodoHistorico":"null","movimiento":"Simbolismo"}
```


Figura 41: Respuesta de la petición Api Base de datos multimedial [6].

Capítulo 6 : Desarrollo de la aplicación multiagente

Una vez realizados los procesos de diseños y conexión del prototipo, se procede a crear los agentes "inteligentes", las clases necesarias y los requisitos para crear una recomendación. Lo primero a desarrollar es el funcionamiento del sistema, después los agentes inteligentes para la creación del sistema multiagente, también de ejecutar y realizar pruebas de la red neuronal. Este capitulo dará a conocer todos estos pasos en el desarrollo del sistema, como método propuesto en el desarrollo de este prototipo.

13 Desarrollo del Prototipo

13.1 Funcionamiento del Sistema

El desarrollo del prototipo para la recomendación de memorias históricas y culturales tiene como propósito crear recorridos recomendados para los usuarios que quieran ver obras de arte. En el desarrollo se determino el uso de preferencias de usuario, las cuales tienen como propósito obtener información de usuario, recreando condiciones iniciales de elementos que pueden llamarle la atención.

Los tipos de preferencia del proyecto son siete, estos se crearon con base en análisis realizado en el articulo [53], de este se obtuvieron los siguientes tipos de preferencias :

- Numero de Obras : Este tipo de preferencia permite identificar el numero de obras de un usuario en un recorrido, el rango de obras de obtiene de un mínimo de cinco a un máximo de diez.
- Precisión de la recomendación : Este item de calificación entre uno a cinco permite forzar la puntualidad del sistema recomendador a la hora de crear un nuevo recorrido recomendado con uno como menor a cinco como mayor nivel de precisión.
- Frecuencia de visita al museo : Este item de calificación entre uno (menor) a cinco (mayor) da a conocer la cantidad de visitas que realiza el usuario al museo.
- Movimiento pintura : Este item permite al usuario seleccionar un movimiento histórico de las pinturas que se tienen en la base de datos multimedial.
- Técnica pintura : Este item permite al usuario seleccionar la técnica de pintura dentro de la base de datos multimedial.
- Movimiento escultura : Este item permite al usuario seleccionar un movimiento histórico de las esculturas que se tienen en la base de datos multimedial.
- Técnica escultura : Este item permite al usuario seleccionar la técnica de escultura dentro de la base de datos multimedial.

Como medida de interacción se plantearon estas preferencias, que permiten al usuario modelar el museo a los gustos de un usuario una vez realizado el primer recorrido recomendado en el museo, después de esto se pueden cambiar estos ajustes para ver nuevas obras dentro de un recorrido. Otro recorrido que se puede realizar dentro del sistema es el recorrido de favoritos, este recorrido se basa en información del usuario, donde por medio de las obras favoritas que este obtiene, se crea un recorrido personalizado con esas obras que han sido del agrado. Una vez se este dentro del recorrido, al momento de observar una obra de arte, como medida de anexo de información el usuario califica la obra de 1 a 5, con uno como menor puntuación y cinco como mayor, indicando el valor de agrado del usuario. Estos datos se analizan cuando el usuario termina un recorrido creando una retroalimentación de datos dentro del prototipo, mejorando el porcentaje de acierto dentro del sistema.

13.2 Desarrollo de los agentes inteligentes

En la creación de los agentes inteligentes, hay varios elementos que se tienen en cuenta, uno de ellos es el comportamiento que tiene, otro el objetivo y la salida que tiene como satisfacción de llegar a cumplir el propósito con el cual han sido creados.

Los agentes inteligentes desarrollados para el prototipo de sistema recomendador se caracteriza por tener ciertas propiedades, una de ellas es su identificador. En el proyecto los agentes tienen un nombre de usuario y un dominio de servidor, el cual permite conectarse a la plataforma de comunicaciones en SPADE por medio del servidor SPADE, este proceso permite agregar agentes al servidor permitiendo una corriente de comunicación abierta con cualquier agente que se encuentre disponible en el sistema. Para la comunicación del sistema SPADE tiene un componente interno del despachador de mensajes, permitiendo el flujo de comunicación entre agentes realizando envíos y recibimientos de mensajes por medio de la biblioteca del agente.

Cada agente tiene un comportamiento especifico para cumplir una función dentro del prototipo, en este caso los cinco agentes del sistema tiene un comportamiento One-Shot (Un Disparo), este comportamiento es definido para realizar tareas que solo se ejecutan una vez cuando esta ejecutado, determinando esa actividad cuando el despachador de mensajes le indique que tiene que ser ejecutado.

Num	Nobre Agente	Identificador Agente
1	Agente Usuario	NumUsuario@127.0.0.1
2	Agente Sistema	asistema@127.0.0.1
3	Agente Recorrido	arecorrido@127.0.0.1
4	Agente Recomendación	arecomendacion@127.0.0.1
5	Agente Retroalimentación	aretroalimentacion@127.0.0.1

Tabla 5: Nombres y identificadores de los agentes. Fuente Propia.

En la tabla 5 se observa el nombre de usuario del agente con su respectivo dominio de servidor con el que fue construido para el prototipo, en la tabla 6 se muestra el agente usuario , en la tabla 7 se muestra el agente sistema , en la tabla 8 se muestra el agente recorrido , en la tabla 9 se muestra el agente recomendación , en la tabla 10 se muestra el agente retroalimentación, cada uno con sus comportamientos, directorio de comunicación, con su ontología , performativa (Acción de realizar un pedido) y identificador de conversación necesarios para comunicarse con los demás agentes del servidor.

Num	Comportamiento	Ontología	Performativa	IdConversación
1	OneShotBehaviour	Recomendador	request	Inicio
2	OneShotBehaviour	Respuesta	request	Res

Tabla 6: Comportamientos Agente Usuario. Fuente Propia.

Dentro de los agentes existen clases que permiten realizar su funcionamiento correcto, una de ellas es la clase de recorrido la cual se divide en recorrido recomendado y recorrido favorito. Cada recorrido es diferente y tiene parámetros opuestos. La clase

Num	Comportamiento	Ontología	Performativa	IdConversación
1	OneShotBehaviour	sistema	request	Iniciar
2	OneShotBehaviour	recorrido	request	Rec
3	OneShotBehaviour	recorrido	request	Fav

Tabla 7: Comportamientos Agente Sistema. Fuente Propia.

Num	Comportamiento	Ontología	Performativa	IdConversación
1	OneShotBehaviour	busqueda	request	Iniciar
2	OneShotBehaviour	inicio	request	Recomendado
3	OneShotBehaviour	inicio	request	Favorito
4	OneShotBehaviour	recorrido	request	Recomendado
5	OneShotBehaviour	recorrido	request	Favorito

Tabla 8: Comportamientos Agente Recorrido. Fuente Propia.

Num	Comportamiento	Ontología	Performativa	IdConversación
1	OneShotBehaviour	Recomendador	request	Inicio
2	OneShotBehaviour	Recomendacion	request	Recomendar

Tabla 9: Comportamientos Agente Recomendación. Fuente Propia.

Num	Comportamiento	Ontología	Performativa	IdConversación
1	OneShotBehaviour	retroalimentacion	request	Iniciar
2	OneShotBehaviour	retroalimentacion	request	Rec
3	OneShotBehaviour	retroalimentacion	request	Fav

Tabla 10: Comportamientos Agente Retroalimentación. Fuente Propia.

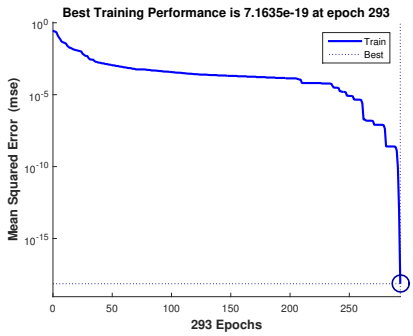
de red neuronal permite al agente obtener todas las propiedades de la red multicapa, desde su entrenamiento hasta la toma de decisiones. Esto comentado anteriormente puede observarse en el diagrama de clases, donde se puede ver detalladamente las relaciones y contenido de cada clase utilizada dentro del prototipo.

13.3 Desarrollo de la Red Neuronal y recomendación

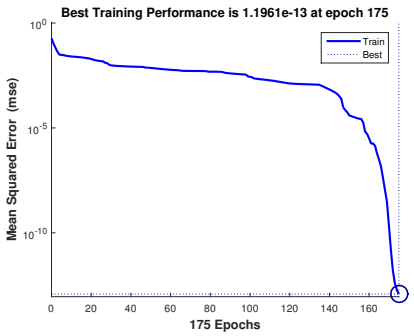
Una parte fundamental para el análisis de la información del usuario dentro del sistema, es la red neuronal. Esta red permite interpretar la información del usuario, dándole una puntuación retroactiva para crear un nuevo recorrido recomendado. El diseño de la red se puede observar en la figura 15. Esta red esta compuesta por dos entradas, seis neuronas en la segunda capa y una salida.

En el desarrollo de la red neuronal se realizaron distintas pruebas que permitieron analizar el margen de error de la red que da el entrenamiento de la red con distintas funciones de activación. Estos casos se pueden observar en las figuras 42(a), 42(b), 42(c), 42(d).

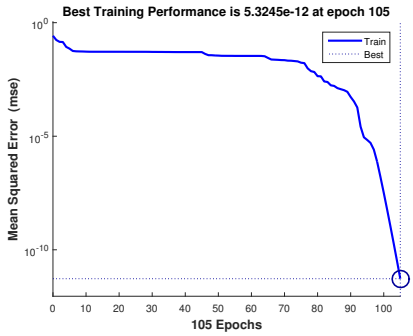
En la red neuronal se realizaron distintas configuraciones que permitieron analizar el rendimiento de todas las neuronas. En todos los casos la red neuronal tiene tres diferentes capas de personalización. En el primer caso la configuración de la red con las dos primeras capas de logsig y la tercera de satlin. En el segundo caso se realizó la configuración de dos capas con logsig y la tercera con purelin como función de



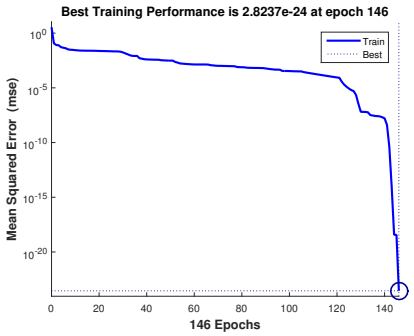
(a) Mejor entrenamiento del caso 1.



(b) Mejor entrenamiento del caso 2.



(c) Mejor entrenamiento del caso 3.



(d) Mejor entrenamiento del caso 4.

Figura 42: Casos de Entrenamiento [53].

Caso	Max	Min	Media	Desviación Estandar
1	1,8205E-06	7,8464E-17	4,52E-07	7,62215E-07
2	3,8011	1,1961E-09	5,53E-01	1,33040547
3	0,64175	5,3245E-08	1,60E-01	0,32087485
4	0,0010988	2,8237E-20	3,67E-04	0,00063346

Tabla 11: Tabla de resultados de la red neuronal [53].

activación. En el tercer caso se realizó la configuración de una capa con purelin, seguidas de dos capas de logsig. En el caso final se realiza una configuración diferente a los tres casos iniciales, donde la primera capa hay una función de activación logsig, la segunda tiene la función satlin y la tercera tiene la función purelin.

Los datos obtenidos durante el desarrollo del entrenamiento de la red neuronal se colocan en la tabla 11, donde se especifica el valor máximo, mínimo obtenido, además del promedio y derivación estándar del análisis de la información. En esta tabla se puede observar el menor valor de rendimiento que se obtuvo al realizar varios casos de entrenamiento, el cual es el caso 4, donde se determina que es una opción de configuración valida para realizar el manejo de la red neuronal para el sistema.

Un punto de vital importancia del prototipo es el desarrollo del recorrido recomendado. Por medio de la salida de la base de datos se genera un recorrido. Con la red neuronal se obtiene una calificación validada por las puntuaciones dadas al usuario y tiempo de duración por obra de arte. Con estos datos y los valores generados por las preferencias se crea un valor de persistencia para la búsqueda dentro de la base de datos.

Caso	Nombre	Porcentaje
1	Frecuencia Visita	10%
2	Presición Recomendación	60%
3	Valor de Retroalimentación	30%

Tabla 12: Tabla de porcentajes recomendación. Fuente Propia.

En la tabla 12, se muestra el porcentaje dado por el sistema de recomendación. Este valor toma un porcentaje de la puntuación del recorrido realizado por el usuario, el valor de visita al museo por el usuario y la precisión de la recomendación. Estos porcentajes suman el 100% y el promedio redondeado muestra el valor final para buscar obras para un usuario.

Para buscar la recomendación se introduce en la consulta de base de datos el identificador de usuario, para encontrar usuario que tengan obras similares a las que se están buscando, después se procede con los usuarios vecinos, a buscar las obras que tengan un valor mayor o igual al promedio de la media de calificaciones dadas en el sistema, con el tiempo se ejecuta el proceso anterior. Después de realizar este proceso se realiza un filtrado de información de las obras dependiendo el movimiento y técnica de preferencia por el usuario. Una vez realizado esto se obtiene un recorrido recomendado, este ejemplo de salida puede verse en la figura 43. Las obras almacenadas dentro de la base de datos multimedial pueden ser de tipo pintura (p) o escultura (e).

```
{"status":"success",
"data":[{"idobravistarecorrido":10,
  "idrecorridosistemaobra":13,
  "idobra":"e20"},
  {"idobravistarecorrido":11,
  "idrecorridosistemaobra":13,
  "idobra":"e1"},
  {"idobravistarecorrido":12,
  "idrecorridosistemaobra":13,
  "idobra":"p59"},
  {"idobravistarecorrido":13,
  "idrecorridosistemaobra":13,
  "idobra":"p26"}}
```

Figura 43: Vista ejemplo del recorrido 13

13.4 Visualización de preferencias y obras

Una entrada que tiene el sistema son las preferencias, en la figura 44 se observa la pantalla de configuración del sistema de realidad aumentada donde el usuario guarda los cambios que sean de su agrado, estos pueden ser el numero de obras a observar, la valoración de visitas al museo o el tipo de precisión del sistema recomendador, también se puede seleccionar el tipo de movimiento o técnica de pintura o escultura, que puede observarse en un recorrido.

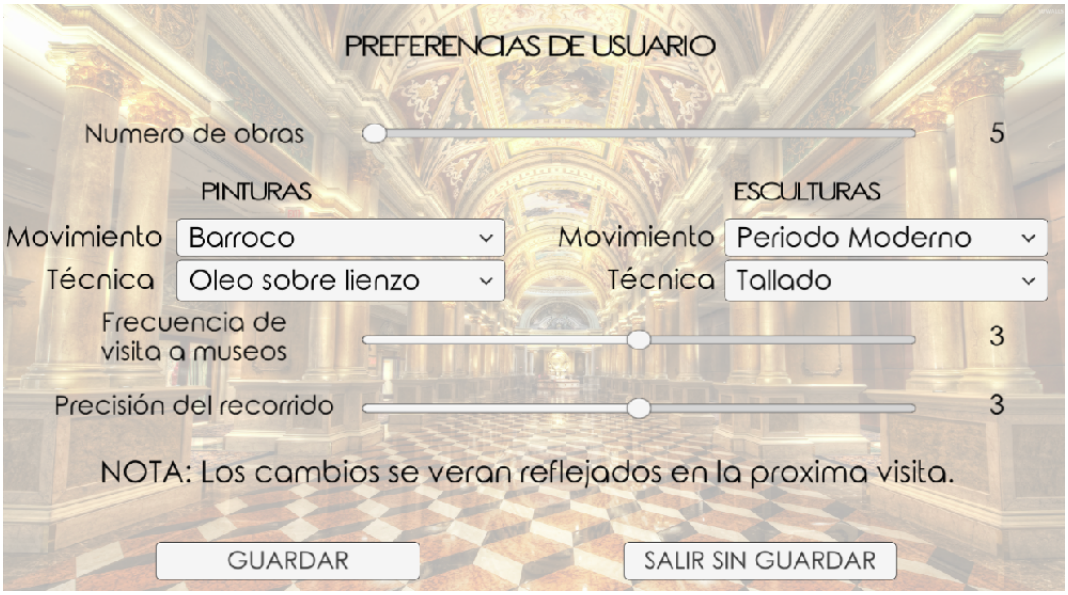


Figura 44: Imagen de configuración preferencias usuario [5]



Figura 45: Ejemplo de muestra pintura [5]

Las figuras 45, 46 dan a conocer la valoración que puede dar el usuario a la pintura, esta puede ser de 1 a 5 estrellas, y si es favorita, aportando con un "like". En las figuras se muestra respectivamente una pintura y una escultura.

13.5 Entradas y Salidas del Sistema

Para la comunicación del sistema con otra aplicación, se creo un servidor web que permite entregar una petición de tipo Get. El sistema recomendador recibe como entrada el identificador de usuario; y si el recorrido es recomendado, favorito



Figura 46: Ejemplo de muestra escultura [5]

o conocer si tiene un recorrido pendiente. El link de acceso al servicio es http://servidor-multiagente.herokuapp.com/

```
{"status": 200,
"data": "Ingrese el la ruta especifica para ver o
   realizar un tipo de recorrido nuevo",
"favorito": "/recorrido/favorito/<idusuario>",
"recomendado": "/recorrido/recomendado/<idusuario>",
"usuario": "/recorrido/<idusuario"}
```

Figura 47: Pantalla Principal Servidor Web. Fuente Propia.

En la figura 47 se muestra la interfaz principal del servidor web donde se ve la salida de la pagina web. En este caso muestra los 3 tipos de interacción que se pueden realizar. En la figura 48 se muestra la salida al realizar una petición de un recorrido recomendado. El link de este recorrido es http://servidor-multiagente.herokuapp.com/recorrido/recomendado/idusuario, donde al obtener respuesta se puede ver que el json de salida muestra un estatus, un "data" y el usuario que se ha solicitado. El recorrido favorito muestra una salida parecida al recorrido recomendado, la única diferencia es el contenido de las obras a las que esta dirigida el recorrido. Por ultimo la verificación del usuario, donde por medio del identificador, se puede mostrar si el usuario tiene un recorrido pendiente (figura 49) o si el usuario no tiene recorrido para mostrar (figura 50).

Toda la información almacenada dentro de la base de datos de usuarios puede mostrarse por medio del sistema recomendador, donde por medio del identificador da a conocer, que recorrido tiene un usuario, el idrecorrido y las obras de arte asignadas a este recorrido. Esta salida se observa en la figura 43.

```
salida : {"status": "success",
"data": {"idrecorridosistema": "13"},
"usuario": "8@127.0.0.1"}
```

Figura 48: Respuesta de la búsqueda recorrido recomendado. Fuente Propia.

```
{"data": {"status": true,
"idrecorridosistema": 13,
"usuario": "8127.0.0.1"}
```

Figura 49: Salida con recorrido pendiente. Fuente Propia.

```
{"data": {"status": false,
"idrecorridosistema": null},
"usuario": "8@127.0.0.1"}
```

Figura 50: Salida sin recorrido pendiente. Fuente Propia.

13.6 Ejecución del sistema

Una vez terminado el proyecto, lo primero que se debe realizar es la instalación del servidor multiagente SPADE. Según [54], en un entorno linux se procede a correr los comandos visualizados en la figura 51.

```
$ pip install SPADE
$ configure.py 127.0.0.1
$ runspade.py
```

Figura 51: Comandos de instalación del servidor SPADE [54].

Después de haber realizado el proceso de instalación del servidor multiagente, lo que se realiza primero es la verificación de que el sistema este corriendo. En la figura 52 se muestra la interfaz principal del servidor multiagente cuando esta en ejecución en el sistema.

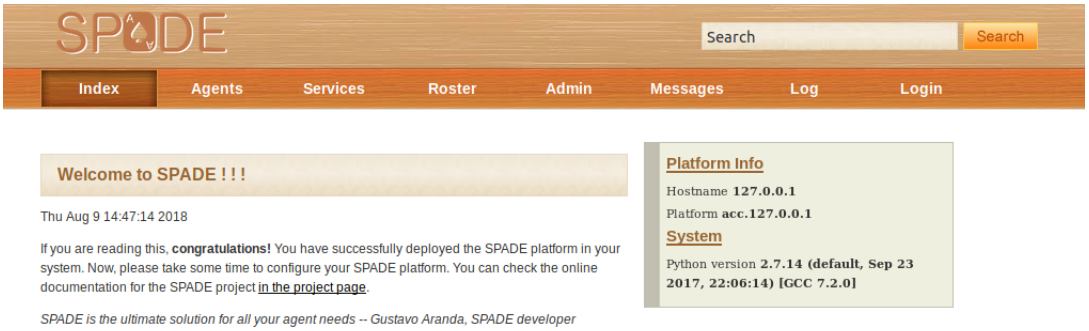


Figura 52: Pagina principal del servidor SPADE. Fuente Propia.

Luego de tener corriendo lo anterior dentro de una terminal, se procede a ejecutar

el servicio del sistema recomendador. Dentro de la carpeta se puede encontrar un archivo de python con el nombre de servidor web.py. Lo único que se debe realizar es correr el comando visualizado en la figura 53; eso si, se deben instalar los programas necesarios para la ejecución del sistema.

```
$ python servidorweb.py
agente iniciado : asistema@127.0.0.1
agente iniciado : arecorrido@127.0.0.1
agente iniciado : arecomendacion@127.0.0.1
 Entrenando Red Neuronal
Thu Aug  9 14:45:57 2018 Server Starts - 127.0.0.1:8080
```

Figura 53: Inicialización del servidor web. Fuente Propia.

Cuando ya este corriendo el servidor web del sistema recomendador basado en multiagentes, se procede a verificar que agentes están corriendo dentro del sistema. En la figura 54, se puede visualizar la lista interna de agentes inteligentes del sistema, donde se muestra los agentes de recorrido, retroalimentación, sistema y 10-1, el cual fue el agente de prueba para tomar esta figura.

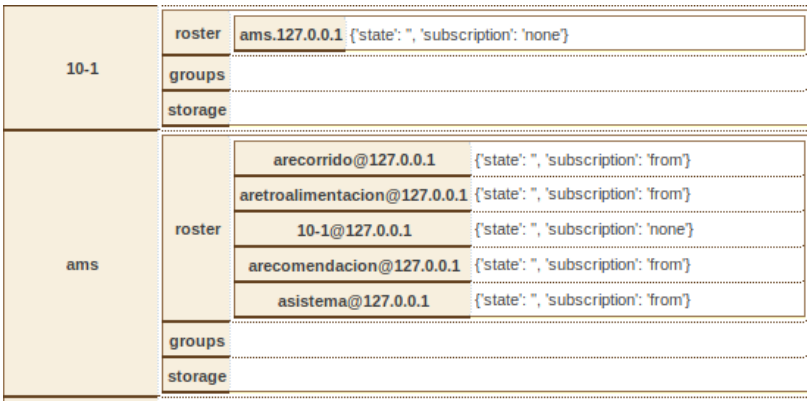


Figura 54: Lista de Agentes del servidor SPADE. Fuente Propia.

Capítulo 7 : Resultados del Prototipo

En este capítulo del documento se trata de los resultados obtenidos durante las pruebas realizadas con el aplicativo de realidad aumentada. Se obtienen las conclusiones de los usuarios que han utilizado la plataforma, así como las obras de arte que más han sido expuestas en el aplicativo. También se obtienen los promedios de tiempo y calificación por obra, ademas del promedio general de calificación obtenido por todos los usuarios que utilizaron el aplicativo.

13.7 Resultados de Efectividad del sistema

Una vez terminado el prototipo, se realizaron pruebas a mas de 20 usuarios, estos interactuaron con la aplicación, realizando un mínimo de dos recorridos recomendados. Las calificaciones que han otorgado y el tiempo empleado por las personas, es analizado por la red neuronal, donde estos datos son tomados para verificar la efectividad del sistema recomendador.

El número de participantes que han utilizado el prototipo han sido de más de 35 personas (se siguen realizando pruebas), con un total de más de 129 recorridos realizados por el sistema recomendador, con quinientas obras de arte observadas.

Todos estos datos han sido satisfactorios, al momento de ver lo que se ha logrado con el desarrollo del proyecto. En la tabla 13 se muestra el promedio de calificación y duración de visualización de las 15 primeras obras de arte mas vistas en el prototipo. En el id obra se puede observar un indicativo p o e antes del número, esto quiere decir que la p significa pintura y la e escultura.

Al revisar los resultados obtenidos en la tabla 14, se puede analizar el buen rendimiento que ha tenido el sistema recomendador, ya que la mayoría de recorridos de obras de arte, han tenido calificaciones entre 3 y 4, con tiempos de duración de los 36 segundos hasta más de 80. Estas valoraciones dan a entender que el objetivo de las recomendaciones es bueno, ya que los usuarios han reaccionado de manera agradable a las obras que le han recomendado.

13.8 Entregables

Se hace entrega de los siguientes elementos:

- Copia del documento de presentación del proyecto de grado.
- Copia del código fuente del proyecto.
- Copia del servidor para multiagentes SPADE.
- Copia del script de ejecución del sistema.

IdObra	Promedio Calificación	Promedio Tiempo (seg)	Número de Vistas
p61	3 / 5	55 seg	20
p31	3 / 5	71 seg	18
e5	3 / 5	42 seg	14
p52	4 / 5	73 seg	14
p45	3 / 5	89 seg	14
e14	4 / 5	76 seg	13
p26	3 / 5	48 seg	12
p62	3 / 5	74 seg	12
e21	3 / 5	55 seg	10
e18	3 / 5	71 seg	10
e15	3 / 5	50 seg	10
e1	4 / 5	36 seg	9
e4	4 / 5	56 seg	9
e13	3 / 5	80 seg	9
e23	4 / 5	44 seg	9

Tabla 13: Tabla de datos de calificación y tiempo por los usuarios. Fuente Propia.

Número	Promedio Calificación	Promedio Tiempo (seg)
1	3.644 / 5	54.53 segundos

Tabla 14: Tabla de resultados obtenidos. Fuente Propia.

14 Conclusiones

Se concluye la realización del proyecto del prototipo de sistema recomendador basado en multiagentes para la recomendación de memorias históricas y culturales, donde se determina una arquitectura de desarrollo para agentes inteligentes que permitió gestionar los recorridos recomendados para los usuarios. También se desarrollo los agentes inteligentes por medio del uso de la plataforma de multiagente SPADE. Por medio de la gestión calificaciones y tiempo de duración en una obra de arte (pintura o escultura) se utilizo las redes neuronales, que permitieron analizar los datos obtenidos por un usuario permitiendo crear recorridos de memorias históricas idóneos para las personas que utilizan el sistema.

El desarrollo de la red neuronal y los agentes inteligentes permitió establecer el sistema de filtrado de información de los datos del usuario, ya que por medio de consultas preestablecidas y datos obtenidos con la retroalimentación de datos, se establecieron preferencias que permitieron la creación de un nuevo recorrido personalizado para un usuario, creando así, un modelo de recomendación donde se estableció los datos necesarios para determinar los gustos del usuario.

El nivel de preferencia del prototipo se determino con el rango de calificación dado de 1 a 5, donde al verificar todas las valoraciones dadas por los usuarios que utilizaron el sistema, se obtuvo un valor de 3.6 sobre 5. Este valor determina un buen rendimiento del sistema recomendador, para la creación de recorridos recomendados y favoritos. Esto otorga un buen impulso al desarrollo de este prototipo recomendado, que con la ayuda de un aplicativo de realidad aumentada permitió dar un buen balance del proyecto.

Se concluye que el uso de servidores web utilizado en el aplicativo de base de datos de usuario y multimedial, permitió entablar conexión entre el sistema recomendador y la base de datos, ya que por medio de links otorgados por el grupo de almacenamiento se gestionaron las consultas requeridas para poder obtener toda la información tanto de los usuario como de las obras de arte.

15 Recomendaciones

Teniendo el cuenta el alcance y limitaciones de este proyecto, se dejan las siguiente recomendaciones para el futuro de este:

- Es necesario correr el sistema recomendador basado en multiagentes en un entorno de trabajo linux, con las recomendaciones dadas en el script de ejecución.
- Es necesario tener disponible los servidores de aplicaciones de base de datos que se utilizaron en el proyecto, es decir tener en funcionamiento el api de usuario y el api de base de datos.
- El sistema de recomendación queda detallado para realizar cambios a futuro, ya sea para anexar nuevas funcionalidades o complementar lo entregado.

16 Bibliografía

[1] ICOM (Consejo Internacional de Museos), "Definición del Museo- ICOM," 2007. [Online]. Available: http://icom.museum/la-vision/definicion-del-museo/L/1/. [Accessed: 20-Sep-2017].

[2] C. Oquendo, "Dia Internacional de los Museos - Archivo Digital de Noticias de Colombia y el Mundo desde 1.990 - eltiempo.com," 2015. [Online]. Available: http://www.eltiempo.com/archivo/documento/CMS-15773375. [Accessed: 21-Sep-2017].

[3] S. Kenderdine, L. K. Y. Chan, and J. Shaw, "Pure Land: Futures for Embodied Museography," J. Comput. Cult. Herit., vol. 7, no. 2, pp. 1–15, 2014.

[4] L. F. Zapata, "Sistemas Multiagente," Foundations, pp. 1–42, 2010.

[5] Monje, K.C.D., Cruz, A.F.S.: ANÁLISIS E IMPLEMENTACIÓN CULTURALES Y DE VIRTUALIDAD EN CONTEXTOS CULTURALES Y DE CONTENIDO HISTÓRICO, (2017)

[6] Camilo, E., Aldana, V., Augusto, J., Torres, C.: Tratamiento de datos para la aplicación de las TICS en escenarios de postconflicto: Elaboración de centros de memoria Histórica, cultural, museos y Senderos ecoturisticos de zonas afectadas por el conflicto con elementos de Realidad Mixta y Sistemas Recom, (2018)

[7] I. Benouaret and D. Lenne, "Personalizing the Museum Experience through Context-Aware Recommendations," Proc. - 2015 IEEE Int. Conf. Syst. Man, Cybern. SMC 2015, pp. 743–748, 2016.

[8] OEI, "Informe de Colombia - Sistemas Nacionales de Cultura," 2017. [Online]. Available: http://www.oei.es/historico/cultura2/colombia/08c.htm. [Accessed: 30-Sep-2017].

[9] J. Hilera and V. José Martínez Hernando, Redes neuronales artificiales: fundamentos, modelos y aplicaciones J.R. Hilera González, V.J. Martínez Hernando. (PDF Download Available), Primera Edición. Madrid, 1995.

[10] MAPFRE, "Nuestras aplicaciones - Fundación MAPFRE," 2017. [Online]. Available: https://www.fundacionmapfre.org/fundacion/es_es/apps/. [Accessed: 30-Sep-2017].

[11] M. Hermitage, "COMUNICADO DE PRENSA - The State Hermitage - aplicaciones móviles" 2017. [Online]. Available: http://www.hermitageapp.com/en/museum/press.html. [Accessed: 30-Sep-2017].

[12] RAE, "Diccionario de la Lengua Española," Edición del Tricentenario, 2017. [Online]. Available: http://dle.rae.es/?id=Q8nqRs0. [Accessed: 20-Sep-2017].

[13] H. Kong, "Animating narratives of remembrance," 2013 Digit. Herit. Int. Congr., pp. 315–322, 2013.

[14] J. McCaffery, A. Miller, A. Vermehren, and A. Fabola, "The Virtual Museums of Caen: A case study on modes of representation of digital historical content," 2015 Digit. Herit. Int. Congr. Digit. Herit. 2015, pp. 541–548, 2015.

[15] S. Giglio, L. Gabriele, A. Tavernise, P. Pantano, E. Bilotta, and F. Bertacchini, "Virtual museums and Calabrian cultural heritage: Projects and challenges," 2015 Digit. Herit. Int. Congr. Digit. Herit. 2015, pp. 703–708, 2015.

[16] M. Mason, "The MIT Museum Glassware Prototype: Visitor Experience Exploration for Designing Smart Glasses," J. Comput. Cult. Herit., vol. 9, no. 3, pp. 1–28, 2016.

[17] M. . Theodorakopoulos et al., "Personalized Augmented Reality Experiences in Museums using Google Cardboards," 12th Int. Work. Semant. Soc. Media Adapt. Pers., 2017.

[18] D. Horie and R. Katsuma, "Recognizing AR markers using optical illusion markers and a low-cost camera," 2017 IEEE 7th Annu. Comput. Commun. Work. Conf. CCWC 2017, 2017.

[19] J. G. Grandi, I. Berndt, H. G. Debarba, L. Nedel, and A. Maciel, "Collaborative manipulation of 3D virtual objects in augmented reality scenarios using mobile devices," 2017 IEEE Symp. 3D User Interfaces, pp. 264–265, 2017.

[20] C. Sánchez-Sánchez, H. Jiménez-Salazar, C. Rodriguez-Lucatero, E. Villatoro-Tello, and G. Ramírez-De-La-Rosa, "Sistema multiagente integrador de bibliotecas digitales," Res. Comput. Sci., vol. 93, pp. 45–56, 2015.

[21] C. Padilla-navarro and M. R. Carlos, "Modelado de un sistema multi-agente aplicado a la predicci ´ on de la personalidad en Twitter Multi-agent system applied to prediction of personality on Twitter," vol. 115, pp. 147–156, 2016.

[22] G. Wen, P. Wang, and S. Member, "Robust Neuro-Adaptive Containment of Multileader Multiagent Systems With Uncertain Dynamics," pp. 1–12, 2017.

[23] C. Wei, J. Jing, C. Mei, and G. Shaoting, "Design of workshop production management control system based on Multi- Agent," IEEE, pp. 427–430, 2016.

[24] N. T. Yazdi and C. H. Yong, "Simulation of Multi-Agent Approach in Multi-Cloud Environment Using Matlab," Proc. Int. Conf. Comput. Intell. Model. Simul., vol. 2016–Septe, pp. 77–79, 2016.

[25] J. Izquierdo, E. Campbell, and I. Montalvo, "Combinación multi-agente de algoritmos evolutivos y minería de datos para mejorar la búsqueda en problemas de optimización del mundo real," Congr. Métodos, 2015.

[26] J. H. Taylor, "Intelligent Control and Asset Management: An Event-based Control Road Map," IEEE, 2016.

[27] E. R. Núnez-valdéz et al., "Plataforma de recomendación de contenidos para libros electrónicos inteligentes basada en el comportamiento de los usuarios," Vent. Informática, vol. 14, pp. 25–40, 2012.

[28] A. Collado Sánchez, "Sistema de recomendación de recursos basado en filtrado colaborativo para la plataforma edX," 2016, p. 171, 2014.

[29] I. Brigui-Chtioui, P. Caillou, and E. Negre, "Intelligent Digital Learning: Agent-Based Recommender System," Proc. 9th Int. Conf. Mach. Learn. Comput., pp. 71–76, 2017.

[30] H. Hajri, Y. Bourda, and F. Popineau, "MORS: A System for Recommending OERs in a MOOC," 2017 IEEE 17th Int. Conf. Adv. Learn. Technol., pp. 50–52, 2017.

[31] M. Rovatsos, G. Vouros, and V. Julian, "Multi-agent systems and agreement technologies: 13th European Conference, EUMAS 2015 and third international conference, AT 2015 Athens, Greece, December 17, 2015 revised selected papers," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9571, no. January, 2016.

[32] Matlab, "MATLAB - MathWorks," 2017. [Online]. Available: https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab. [Accessed: 21-Oct-2017].

[33] C. A. Ruiz, M. Susana, B. Autor, : Damián, and J. Matich, "Universidad Tecnológica Nacional – Facultad Regional Rosario Departamento de Ingeniería Química Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ) Redes Neuronales: Conceptos Básicos y Aplicaciones," 2001.

[34] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophys., vol. 5, no. 4, pp. 115–133, 1943.

[35] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in . . . ," Psychol. Rev., vol. 65, no. 6, pp. 386–408, 1958.

[36] A. Raya, "Intel presenta la Inteligencia Artificial en una memoria USB," 2017. [Online]. Available: http://omicrono.elespanol.com/2017/07/inteligencia-artificial-en-una-memoria-usb/. [Accessed: 20-Oct-2017].

[37] Deepmind, "Open sourcing Sonnet - a new library for constructing neural networks," https://deepmind.com, 2017. [Online]. Available: https://deepmind.com/blog/open-sourcing-sonnet/. [Accessed: 20-Oct-2017].

[38] M. Sliger, "Agile Project Management with Scrum," PMI® Glob. Congr. 2011, 2011.

[39] Management Plaza, "¿Como se desarrolla un proyecto con scrum? - Management Plaza," 2017. [Online]. Available: http://managementplaza.es/blog/desarrollar_un_proyecto_con_scrum/. [Accessed: 20-Oct-2017].

[40] FIPA, "Bienvenido a la Fundación para Agentes Físicos Inteligentes." [Online]. Available: http://www.fipa.org/. [Accessed: 17-Nov-2017].

[41] J. A. B. Bañares, "Comunicación entre agentes."

[42] M. Gregori, J. Cámara, and G. Bada, "A jabber-based multi-agent system platform," Proc. fifth Int. Jt. Conf. Auton. agents multiagent Syst., pp. 1282–1284, 2006.

[43] Github, "XMPP — Una descripción general de XMPP," 2017. [Online]. Available: https://xmpp.org/about/technology-overview.html. [Accessed: 17-Jan-2018].

[44] S. Russel and P. Norvig, Inteligencia Artificial, Segunda Ed. Madrid: Prentice hall, 2004.

[45] M. Kubat, "Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7.," The Knowledge Engineering Review, vol. 13, no. 4. p. S0269888998214044, 1999.

[46] K. KOUISS, H. PIERREVAL, and N. MEBARKI, "Using multi-agent architecture in FMS for dynamic scheduling," Chapman Hall, no. 0956–5515, p. 7, 1997.

[47] L. Bass, P. Clements, and R. Kazman, "Software Architecture in Practice," Addison Wesley, vol. 2nd Edition, 2003.

[48] . © Copyright 2017 NumPy developers.: NumPy — NumPy, http://www.numpy.org/, (2017)

[49] Gunicorn - Servidor HTTP de Python WSGI para UNIX, http://gunicorn.org/

[50] Bienvenido — Flask (Python Microframework), http://flask.pocoo.org/

[51] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95, 1995, pp. 194–201.

[52] Universitat Pompeu Fabra., Hipertext.net. Universidad Pompeu Fabra.

[53] Ball, A., Gelvez, N., Espitia, H., Distrital, U., Jos, F.: Prototype of a recommendation system based on multi-agents in the analysis of movies dataset. WEA 2018. 12 (2018)

[54] SPADE: Installation, https://pythonhosted.org/SPADE/spade.quickstart. installation.html