



Universidad de las Ciencias Informáticas  
FACULTAD 5

# Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Título:**

“Agente conversacional con habilidades para la venta de  
productos en un mundo virtual en 3D”

**Autor:**

Nelson Manuel González Martínez.

**Tutor (es):**

Ing. Enelis Blanca Cuba Rondón.

Ing. Carlos Enrique de Armas Álvarez

**Co - Tutores:**

Ms C. Fermín Lorenzo Carvajal Rosabal.

Ms C. Zoraida Fernández Guevara.

## DECLARACIÓN DE AUTORÍA

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

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

Firma del Autor

Nelson Manuel González  
Martínez

Firma del Tutor

Ing. Enelis Blanca Cuba  
Rondón.

Firma del Tutor

Ing. Carlos Enrique de  
Armas Álvarez

### Síntesis de los tutores:

**Nombre y Apellidos:** Ing. Enelis Blanca Cuba Rondón.

**E-mail:** [ebcuba@uci.cu](mailto:ebcuba@uci.cu)

Graduada en el curso 2011-2012 como Ingeniera en Ciencias Informática. Especialista A, Jefa del grupo de inteligencia empresarial del Centro Entornos Interactivos 3D (VERTEX) y representante de los estudios que se realizan en la tecnología referente a los entornos virtuales en OpenSim.

**Nombre y Apellidos:** Ing. Carlos Enrique de Armas Álvarez.

**E-mail:** [cedearma@uci.cu](mailto:cedearma@uci.cu)

Ingeniero en Ciencias de la Informática. Profesor recién graduado del Centro VERTEX, con experiencias en la creación de agentes inteligentes en entornos virtuales 3D.

### Síntesis de los co-tutores:

**Nombre y Apellidos:** Fermín Lorenzo Carvajal Rosabal.

**E-mail:** [fcarvajal@uci.cu](mailto:fcarvajal@uci.cu)

El profesor es Licenciado en Educación en la especialidad de Lengua Inglesa y Máster en Teoría y Práctica del Inglés Contemporáneo. Posee 31 años de experiencia; 27 dellos en la enseñanza superior. Ha impartido asignaturas tales como Estilística Inglesa, Literatura de los Pueblos de Habla Inglesa, Historia de los Pueblos de Habla Inglesa, Lexicología y Semántica, Fonética Inglesa y Práctica Integral del Inglés. Ha impartido un gran número de cursos de postgrado, varios dellos preparados por él.

**Nombre y Apellidos:** Zoraida Fernández Guevara.

**E-mail:** [zorlis@uci.cu](mailto:zorlis@uci.cu)

La profesora es Filóloga en Lengua Rusa, licenciada en Educación en la especialidad de Lengua Inglesa y Máster en Tecnología de los procesos educativos. Posee 30 años de experiencia en la enseñanza superior. Ha impartido asignaturas tales como Idioma Ruso e Inglés, Comunicación Profesional, Formación Pedagógica y Multimedia para estudiantes no filólogos entre otras. Ha impartido un gran número de cursos de postgrado, varios dellos preparados por ella. Ha participado en investigaciones pedagógicas. Ha presentado trabajos en eventos nacionales e internacionales.

Una de las innovaciones de los últimos años han sido los metaversos; mundos virtuales que permiten que un número de personas interactúen y puedan realizar actividades sociales, educativas, científicas, entre otras. En la Universidad de Ciencias Informáticas, en el Centro Entornos Interactivos 3D se está desarrollando el entorno “*Virtual English*” sobre un metaverso llamado OpenSim. Este entorno está diseñado con áreas que contienen ejercicios definidos por roles, para fortalecer las habilidades comunicativas “desde el nivel uno hasta el nivel cuatro” de la asignatura de idioma extranjero. Una alternativa para el cumplimiento de este objetivo es el uso de agentes conversacionales. Por lo que el objetivo de este trabajo es desarrollar un agente conversacional que permita a los estudiantes intercambiar en el área de la venta de productos en un área del metaverso “*Virtual English*”, como un recurso educativo en apoyo a la materia Idioma Extranjero I. Para cumplir con el objetivo trazado, el agente conversacional o *chatbot* a desarrollar, permite la utilización de herramientas de procesamiento de lenguaje natural basado en texto, relacionado con el dominio de la venta de productos; además de la detección de errores lingüísticos que puedan presentar los usuarios.

**Palabras clave:** agente conversacional, chatbot, metaversos, opensim, procesamiento de lenguaje natural basado en texto, venta.

INTRODUCCIÓN .....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN .....	5
1.1 Introducción .....	5
1.2 Mundo Virtual (Metaversos) .....	5
1.2.1 OpenSimulator (OpenSim) .....	6
1.2.2 Second Life .....	6
1.2.3 Visores de navegación 3D .....	6
1.2.4 Valoración de la selección del servidor de aplicación 3D .....	7
1.3 Inteligencia Artificial (IA).....	8
1.3.1 Agentes Inteligentes.....	8
1.3.2 Agentes conversacionales (Bot o Chatbot) .....	10
Agentes conversacionales en el “Virtual English” .....	11
1.3.3 Agentes conversacionales de carácter comercial.....	11
1.3.4 Lingüística Computacional .....	12
Procesamiento de lenguaje natural (NLP por sus siglas en inglés) .....	13
1.3.5 Sistemas Basados en Reglas .....	13
1.3.6 Razonamiento Basado en Casos .....	13
1.4 Herramientas y lenguajes.....	14
1.4.1 Lenguaje AIML (del inglés Artificial Intelligence Mark-up Language) .....	14
1.4.2 Apache.....	16
1.4.3 MySQL.....	16
1.4.4 XAMPP .....	17
1.4.5 Sistemas Gestores de Base de Datos (SGBD) .....	17
1.4.6 EMS SQL Manager for MySQL .....	17
1.4.7 Entorno Integrado de Desarrollo (IDE por sus siglas en inglés).....	17
1.4.8 Microsoft Visual Studio.....	18
1.4.9 Lenguaje de Programación C# .....	18
1.5 SharpNLP .....	19

1.6 Metodologías de software .....	19
1.6.1 Programación Extrema (XP) .....	20
1.7 Ingeniería de Software Orientada hacia Agentes .....	20
1.8 Criterio de selección de la metodología de desarrollo .....	22
1.9 Conclusiones Parciales.....	22
CAPÍTULO 2: EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO DE LA SOLUCIÓN PROPUESTA .....	23
2.1 Introducción .....	23
2.2 Proceso de captura de requisitos .....	23
2.2.1 Requisitos funcionales .....	23
2.2.2 Requisitos no funcionales .....	24
2.3 Propuesta de solución.....	24
2.4 Funcionamiento de la propuesta de solución .....	25
2.4.1 Entrada de datos .....	25
2.4.2 Procesamiento de datos.....	26
2.4.3 Salida de datos .....	28
2.5 Usuarios relacionados con el sistema .....	28
2.6 Fase de exploración y definición del alcance del proyecto .....	29
2.6.1 Historias de Usuario (HU) .....	29
2.7 Fase de planificación de las iteraciones .....	34
2.7.1 Tareas de ingeniería .....	34
2.7.2 Plan de iteraciones .....	40
2.8 Plan de entregas .....	41
2.9 Fase de diseño .....	42
2.9.1 Patrones de Diseño .....	42
2.9.2 Diseño de la arquitectura del software .....	43
2.9.3 Diagrama de clases del sistema.....	45
2.9.4 Descripción de las tarjetas CRC.....	46
2.10 Consideraciones parciales .....	47

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS.....	48
3.1 Fase de implementación del sistema .....	48
3.1.1 Iteración 1 .....	48
3.1.2 Iteración 2 .....	51
3.1.3 Iteración 3 .....	52
3.1.4 Iteración 4 .....	53
3.1.5 Iteración 5 .....	54
3.2 Fase de Pruebas del Sistema .....	55
3.2.1 Pruebas de Aceptación .....	55
3.3 Consideraciones parciales .....	59
CONCLUSIONES .....	61
BIBLIOGRAFÍA.....	62

## ÍNDICE DE FIGURAS E IMÁGENES

Figura 1. Funcionamiento de un agente inteligente simple (11). .....	10
Figura 2. Proceso de la conversación entre humano-agente. ....	16
Figura 3. Secuencia de uso de los analizadores. ....	26
Figura 4. Flujo de actividades para el procesamiento del texto entrado por el usuario. ....	26
Figura 5. Arquitectura del sistema.....	45
Figura 6. Diagrama de clases del sistema.....	46
Figura 7. Iteraciones de las pruebas de aceptación. ....	60
Figura 8. Los Idiomas más hablados según. Babilon Language Coaching. 6-9-2013.....	66
Imagen 1. Ejemplo de uso del cliente de aplicaciones 3D Imprudence. ....	7
Imagen 2. Ubicación del supermercado en el metaverso “Virtual English”. ....	25
Imagen 3. Diseño y personalización del agente conversacional en el metaverso “Virtual English” .....	42

Tabla 1. Roles definidos para el acceso al entorno virtual.....	29
Tabla 2. HU1 Configuración del servidor de aplicaciones 3D, OpenSim .....	29
Tabla 3. HU2 Personalización del avatar en el metaverso .....	30
Tabla 4. HU3 Configuración del gestor de base de datos y el servidor MySQL.....	30
Tabla 5. HU4 Implementación de la conexión del agente conversacional al servidor OpenSim .....	30
Tabla 6. HU5 Diseño de las técnicas de mercado inherentes al agente conversacional .....	31
Tabla 7. HU6 Implementación de la funcionalidad de proponer productos y ofrecer alternativas .....	31
Tabla 8. HU7 Implementación de la funcionalidad de ofrecer información del precio de los productos .....	31
Tabla 9. HU8 Implementación de la funcionalidad de informar al cliente del precio total de la compra .....	32
Tabla 10. HU9 Diseño de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.....	32
Tabla 11. HU10 Implementación de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.....	32
Tabla 12. HU11 Diseño de la funcionalidad de ofertar descuento si el cliente compra una variedad de productos a la vez.....	33
Tabla 13. HU12 Implementación de la funcionalidad de ofertar descuento si el cliente compra una variedad de productos a la vez.....	33
Tabla 14. HU13 Diseño de las respuestas del agente conversacional .....	34
Tabla 15. HU14 Implementación del mecanismo de respuestas del agente conversacional con ayuda de la herramienta AIML .....	34
Tabla 16. Tarea de configuración del servidor de aplicaciones 3D, OpenSim .....	35
Tabla 17. Tarea de personalización del avatar en el metaverso.....	35
Tabla 18. Tarea de configuración del SGBD y el servidor MySQL .....	35
Tabla 19. Tarea de implementación de la conexión del agente conversacional al servidor OpenSim .....	36
Tabla 20. Tarea de definición de las técnicas de mercado inherentes al agente conversacional .....	36
Tabla 21. Tarea de implementación la funcionalidad de proponer productos y ofrecer alternativas .....	36
Tabla 22. Tarea de implementación de la funcionalidad de ofrecer información del precio de los productos .....	37

Tabla 23. Tarea de implementación de la funcionalidad de ofrecer información del precio total de la compra .....	37
Tabla 24. Tarea de diseño de la funcionalidad de ofertar descuento, por cantidad de un mismo producto comprado.....	38
Tabla 25. Tarea de Implementación de la funcionalidad de ofertar descuento, por cantidad de un mismo producto comprado .....	38
Tabla 26. Tarea de diseño de la funcionalidad de ofertar descuento, si el cliente compra una variedad de productos a la vez.....	38
Tabla 27. Tarea de Implementación de la funcionalidad de ofertar descuento, si el cliente compra una variedad de productos a la vez .....	39
Tabla 28. Tarea de diseño de las respuestas del agente conversacional.....	39
Tabla 29. Implementación del mecanismo de respuestas del agente conversacional con ayuda de la herramienta AIML .....	40
Tabla 30. Planificación de iteraciones por HU.....	40
Tabla 31. Planificación de entregas por iteraciones del producto.....	41
Tabla 32. Tarjeta CRC de la clase Program.....	46
Tabla 33. Tarjeta CRC de la clase Communicative_Function .....	46
Tabla 34. Tarjeta CRC de la clase Warehouse .....	47
Tabla 35. Tarea 1 Iteración 1 .....	49
Tabla 36. Tarea 2 Iteración 1 .....	49
Tabla 37. Tarea 3 Iteración 1 .....	50
Tabla 38. Tarea 4 Iteración 1 .....	50
Tabla 39. Tarea 5 Iteración 1 .....	50
Tabla 40. Tarea 6 Iteración 1 .....	50
Tabla 41. Tarea 7 Iteración 2 .....	51
Tabla 42. Tarea 8 Iteración 2 .....	51
Tabla 43. Tarea 9 Iteración 3 .....	52
Tabla 44. Tarea 10 Iteración 3 .....	52
Tabla 45. Tarea 11 Iteración 4 .....	53
Tabla 46. Tarea 12 Iteración 4 .....	54

Tabla 47. Tarea 13 Iteración 5 .....	54
Tabla 48. Tarea 14 Iteración 5 .....	55
Tabla 49. Caso de prueba HU6_P1 .....	55
Tabla 50. Caso de prueba HU7_P2 .....	56
Tabla 51. Caso de prueba HU8_P3 .....	56
Tabla 52. Caso de prueba HU10_P1 .....	57
Tabla 53. Caso de prueba HU12_P1 .....	58
Tabla 54. Caso de prueba HU14_P1 .....	58

## INTRODUCCIÓN

En la actualidad el aprendizaje a través de la red, (mayormente conocido como *e-learning*) y las Tecnologías de la Información y la Comunicación (TIC), han ganado espacios significativos para la reflexión y discusión permanente en el ámbito educativo. La educación a distancia ha evolucionado acompañada del desarrollo de los medios, siendo Internet y sus servicios en estos momentos, la compañía principal. De ahí la tendencia a este tipo de experiencias de aprendizaje que han servido como un acercamiento progresivo a posibles sistemas completamente a distancia (1). Esta situación ha permitido diseñar soluciones informáticas como los entornos de enseñanza y aprendizajes.

Debido a la necesidad de comunicarse, en el estudio de inglés (como segunda lengua o como lengua extranjera) se intercambian aspectos cognitivos y se establecen relaciones de trabajo (entre otras) como un factor importante para la formación de la cultura general integral de un profesional. Es además una de las tantas materias estudiadas mediante la Web, debido a la densidad de hablantes a nivel internacional (ver Anexo #1). Una de las problemáticas planteadas, por los instructores de idioma inglés, es lograr que los estudiantes se abstraigan en determinadas situaciones y desarrollen la imaginación para facilitar el aprendizaje de competencias comunicativas, siendo estas competencias los sistemas subyacentes de conocimiento y habilidad para el uso en la comunicación real y efectiva (2).

Muchas son las alternativas que se utilizan para lograr desarrollar este tipo de actividad con la mayor calidad posible, pero no siempre satisfacen las necesidades de los usuarios. Un ejemplo es el *software* educativo, el que juega un papel importante como apoyo al docente y al discente en la formación de este último. Si bien existen importantes y numerosos desarrollos de sistemas, en su mayoría corresponden a sistemas de práctica y ejercitación. Su principal característica es entregar al alumno la posibilidad de ejercitarse en una determinada tarea una vez que obtenga los conocimientos necesarios para su dominio (3).

Uno de los *software* que se usan para estos fines son los metaversos: mundos virtuales que recrean un entorno en 3D donde los usuarios interactúan social, cultural y económicamente a través de un soporte lógico, representados por un avatar (4).

El Centro Entornos Interactivos 3D (VERTEX) y el departamento de Humanidades de la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI), desarrollaron un metaverso al que se llamó “*Virtual English*” y sobre el servidor de aplicaciones 3D, OpenSim.

Este mundo simula un entorno imaginario donde se puede desarrollar el aprendizaje de habilidades comunicativas en la asignatura Idioma Extranjero I, a través de ejercicios definidos por roles, interacción con objetos y agentes conversacionales. Estos agentes son productos de software destinados a realizar actividades sociales, permitiendo a los usuarios interactuar y ejercitar, de manera inmersiva, algunas habilidades comunicativas mediante el diálogo escrito en distintas áreas del entorno. Además permiten verificar el resultado de las interacciones con los agentes en tiempo real.

En la asignatura Idioma Extranjero I, los estudiantes deben desarrollar habilidades comunicativas que les permitan realizar compras en un mercado. Por lo que se diseñó un área dentro del entorno “*Virtual English*” para este fin. En esta área se debe implementar un agente conversacional capaz de interactuar con los usuarios mediante una interfaz de texto, con la que se puede vender productos y responder preguntas de los usuarios relacionadas con este ámbito.

Debido a la situación problemática antes planteada, se propone el siguiente **problema de la investigación**: ¿Cómo implementar un agente conversacional capaz de comunicarse con los usuarios para realizar actividades de venta de productos en el entorno “*Virtual English*”?

Por lo que se define como **objeto de estudio**: agentes conversacionales, con el **objetivo general** de: desarrollar un agente conversacional para la venta de productos en el entorno *Virtual English* mediante una interfaz de entrada/salida basada en texto.

El trabajo se ha enmarcado en el **campo de acción**: los agentes conversacionales para la venta de productos, con interfaces basadas en texto.

Para solucionar el problema y alcanzar el objetivo de la investigación se definieron las siguientes **tareas investigativas**:

- Investigación de los agentes en el estado del arte.

- Investigación y recopilación de información referente a los elementos ya desarrollados sobre los agentes conversacionales.
- Selección de la herramienta que permita el desarrollo del diálogo del agente conversacional a través de texto.
- Realización del análisis y diseño para dar solución al problema planteado.
- Personalización de un agente conversacional que permita actividades de venta de productos.
- Implementación del proceso de detección de errores.
- Definición e implementación de las pruebas para validar la solución.

Para el desarrollo de “las tareas de investigación” se emplearon los métodos de investigación que se describen a continuación:

## **Métodos Teóricos:**

**Análisis-síntesis:** para la investigación de las tendencias del desarrollo de agentes y sus capacidades.

**Histórico-lógico:** para el estudio de la evolución de los agentes conversacionales.

## **Métodos Empíricos:**

**Revisión bibliográfica:** para realizar la investigación sobre los agentes conversacionales en fuentes de información como libros, artículos, revistas, publicaciones, entre otras; con el fin de desarrollar la base teórica del trabajo.

**Criterios de expertos:** para la comprobación funcional del agente conversacional como mecanismo de interacción.

**Estructura del contenido:** para un mayor entendimiento se estructuró el documento en tres capítulos:

**Capítulo 1: Fundamentación teórica de la investigación:** En este capítulo se realiza una búsqueda bibliográfica referente a los agentes inteligentes y los agentes conversacionales. También se investigan sus características, similitudes, carencias y su principio de funcionamiento, además de las tecnologías, lenguajes, bibliotecas y herramientas como servidores y metodologías de desarrollo para la confección de la solución.

**Capítulo 2: Solución propuesta para el desarrollo del agente conversacional:** En este capítulo se describe el principio de funcionamiento de la solución, principio de diseño como: arquitectura y patrones arquitectónicos. Se mostrará además una planificación de las Historias de Usuario, las tareas de ingeniería entre otros aspectos.

**Capítulo 3: Implementación y validación de los resultados:** En este capítulo se hace el modelado del diagrama de clases. Se describen las tareas por cada iteración para su posterior implementación. Se realizan las pruebas al agente conversacional implementado con el fin de medir su capacidad de interacción.

## CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

### 1.1 Introducción

Producto al desarrollo vertiginoso de la informática en muchas áreas del conocimiento, es inevitable que al paso del tiempo se tienda a la automatización de muchas de las actividades sociales, en las que normalmente su realización se haría a través del trabajo manual. Este es el caso de la educación, que por su necesidad ha incursionado en los métodos informáticos de interacción entre personas, para satisfacer determinadas situaciones en la enseñanza.

En el presente capítulo se realiza una investigación sobre agentes inteligentes y agentes conversacionales en mundos virtuales. Se analizan además las tecnologías, herramientas y mecanismos que pueden ser utilizadas en el desarrollo de la solución.

### 1.2 Mundo Virtual (Metaversos)

Son una recreación virtual detallada o parcialmente detallada de un entorno que puede o no ser real, en el que se interactúa a través de avatares o personajes y donde se pueden utilizar bienes y objetos del entorno. Muchos mundos virtuales son elaborados para satisfacer necesidades existentes, tanto profesionales como sociales y de ocio. Su origen se remonta a aplicaciones profesionales como simuladores de vuelo y simulación médica que dieron paso a actividades recreativas en forma de video juegos masivos MMOG<sup>1</sup>. Este es el caso de los sistemas MMORPG<sup>2</sup>, MMOFPS<sup>3</sup>, MMORLG<sup>4</sup>, Metaversos<sup>5</sup>, entre los más mencionados (5).

Algunos ejemplos de los más utilizados a nivel mundial son Opensim y *Second Life*. Su acceso es a través de aplicaciones conocidas como visores o clientes de navegación 3D, desarrollados para distintos sistemas operativos.

---

<sup>1</sup> Videojuego Multijugador Masivo en Línea (*Massively Multiplayer Online Game*).

<sup>2</sup> Videojuegos de Rol Multijugador Masivos en Línea (*Massively Multiplayer Online Role-Playing Game*).

<sup>3</sup> Multijugador Masivo en Primera Persona (*Massively Multiplayer First-Person Shooter*).

<sup>4</sup> Multijugador Masivo Online Basado en la Vida Real (*Massively Multiplayer Online Real-Life Games*).

<sup>5</sup> Es un concepto muy parecido a MMORPG: se trata de espacios 3D totalmente inmersivos.

### 1.2.1 OpenSimulator (OpenSim)

Es un servidor de aplicaciones 3D multiplataforma creado por Darren Guard y liberado bajo una licencia BSD (del inglés *Berkeley Software Distribution*), por lo que su código es abierto, tanto para uso comercial como doméstico. La última versión disponible es la 0.8.1. Se puede utilizar para crear un entorno virtual accesible a través de una variedad de clientes, en múltiples protocolos. Permite desarrollar entornos utilizando las tecnologías que mejor se ajusten al trabajo individual. Está diseñado para ser fácilmente ampliable a través de módulos cargables para construir completamente configuraciones personalizadas (6).

### 1.2.2 Second Life

Conocido por su abreviatura SL y cuya traducción al español sería "Segunda Vida", es un programa que simula un nuevo mundo virtual en 3D, que le permite a cualquier persona por medio de un *avatar* (personaje virtual), crear su propia comunidad en la red. Fue desarrollado por la empresa Linden Lab, en el año 2002, con el objetivo de establecer nuevas formas de convivir en los espacios virtuales.

Una vez ingresado en este mundo virtual el usuario puede realizar actividades políticas, culturales y económicas, aunque en este mundo virtual se convive y comparte con otras personas virtuales por medio de los avatares o personajes virtuales. Una particularidad de este entorno es que los usuarios eventualmente deben pagar dinero real por hacer uso de ciertas opciones, mediante una moneda canjeable llamada *Linden dollar* (7).

### 1.2.3 Visores de navegación 3D

También conocidos como clientes de servidores de aplicaciones 3D, son aplicaciones que permiten la conexión a un mundo virtual y además permiten la exploración e interactividad para los usuarios a través de avatares. Existen visores desarrollados por distintas compañías y con algunas ventajas entre ellos. Un ejemplo muy utilizado es Imprudence.

**Imprudence:** Es un visor de código abierto bajo los términos de GNU<sup>6</sup>, diseñado para su uso en OpenSim. Este está basado en el código del visor de *Second Life* y es compatible con el mismo, además está

---

<sup>6</sup> GNU: GNU is Not Unix. Movimiento y comunidad de Software y Conocimiento Libres.

desarrollado para Linux, MAC OSX y Windows (ver Imagen 1. *Ejemplo de uso del cliente de aplicaciones 3D Imprudence.*) (8).



Imagen 1. *Ejemplo de uso del cliente de aplicaciones 3D Imprudence.*<sup>7</sup>

#### 1.2.4 Valoración de la selección del servidor de aplicación 3D

Se selecciona como servidor el OpenSim pues es multiplataforma de código abierto, lo que permite ahorrar en la compra de una licencia para utilizar el metaverso Second Life, imposible para el país por ser un software privativo. Esta plataforma libre permite contar con el apoyo de una gran comunidad para las consultas durante el desarrollo del producto.

<sup>7</sup> Tomada de [www.updatemybathroom.com](http://www.updatemybathroom.com)

El desarrollo de los metaversos, en su gran totalidad, es debido a la inmersión que se les puede dar, haciéndose imprescindible un mundo simulado lo más real posible, basándose en personajes virtuales que tengan una inteligencia artificial intrínseca.

### 1.3 Inteligencia Artificial (IA)

Es un término introducido por John McCarthy en 1956, y se refiere a una rama de las ciencias de la computación que intenta encontrar esquemas generales de representación del conocimiento, y formalizar procesos de razonamiento coherentes, que permitan resolver problemas difíciles en dominios de aplicación concretos (9).

De esta rama de la computación se decidió estudiar materias como los agentes conversacionales e inteligentes, los sistemas basados en reglas y la lingüística computacional como marco de trabajo para darle solución al problema de investigación.

#### 1.3.1 Agentes Inteligentes

Son entes computarizados que interactúan en entornos virtualmente complejos donde perciben y actúan de forma autónoma, realizando un conjunto de tareas y cumpliendo objetivos para los cuales fueron diseñados y presentan, además, características humanas que facilitan el aprendizaje. Las características pueden expresarse desplegando texto, gráfico, iconos, voz, animación, multimedia o realidad virtual (10). En la Figura 1 se muestra el funcionamiento básico de un agente inteligente.

En la bibliografía estudiada, los agentes inteligentes tienen características distintivas como son (12):

- **Autonomía:** un agente es completamente autónomo si es capaz de actuar basándose en su experiencia. El agente es capaz de adaptarse aunque el entorno cambie drásticamente y puede brindar diferentes formatos (o estilos) de respuesta dependiendo del tipo de usuario, del tipo de pregunta realizada, etc.
- **Sociabilidad:** este atributo permite a un agente comunicarse con otros agentes, otras entidades e incluso con humanos.

- **Racionalidad:** el agente siempre realiza lo correcto a partir de los datos que percibe del entorno y sabe formular respuestas: analizando la frase introducida, es capaz de generar la respuesta más apropiada.
- **Reactividad:** un agente actúa como resultado ante los cambios de su entorno. En este caso, un agente percibe el entorno y esos cambios dirigen su comportamiento. Puede proporcionar respuestas de diversa naturaleza, es decir, no tiene que limitarse a brindar frases como respuestas, sino que pueden facilitar enlaces dentro de la Web, o incluso recursos electrónicos como respuesta.
- **Proactividad:** un agente es pro-activo cuando es capaz de controlar sus propios objetivos a pesar de los cambios en el entorno. Puede ser programado para comprender el contexto de la conversación y situar la pregunta en el mismo.
- **Adaptabilidad:** está relacionada con el aprendizaje que un agente es capaz de realizar y si puede cambiar su comportamiento basándose en ese aprendizaje. Se encuentra en fase de desarrollo, pero los avances son alentadores.
- **Movilidad:** capacidad de un agente de trasladarse a través de una red telemática.
- **Veracidad:** se asume que un agente no comunica información falsa a propósito.
- **Personalidad:** cada agente es único y depende del programador las características que quiera darle, como emociones o comportamiento no verbal.

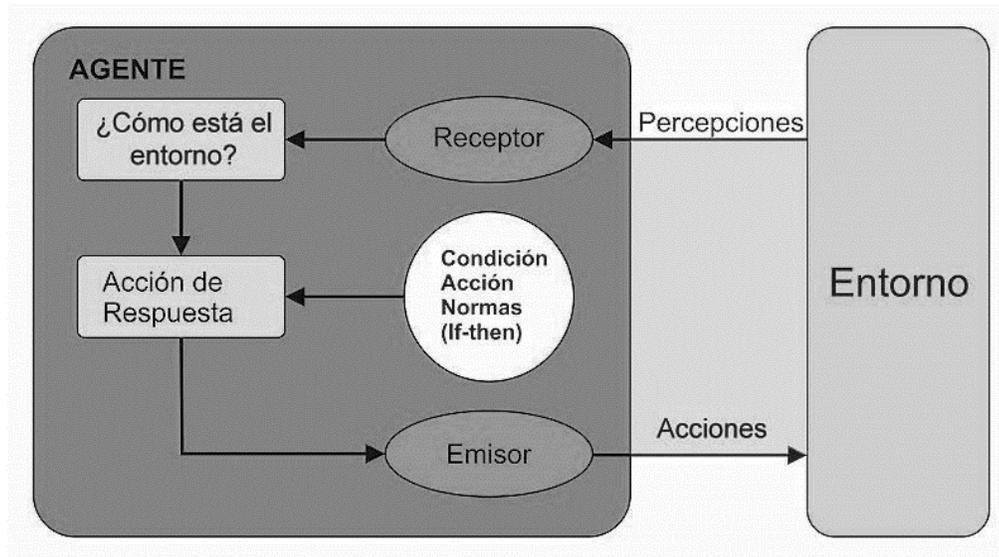


Figura 1. Funcionamiento de un agente inteligente simple (11).

### 1.3.2 Agentes conversacionales (*Bot o Chatbot*)

Son agentes virtuales que generalmente están diseñados para realizar actividades sociales de diferentes tipos, utilizando técnicas pertenecientes a la lingüística computacional (epígrafe 1.3.4) como el procesamiento de lenguaje natural. Para el análisis de los diálogos en un sistema de preguntas y respuestas que son manejadas en distintos casos por sistemas basados en reglas (epígrafe 1.3.5) o razonamiento basado en casos (epígrafe 1.3.6). Por ello estos pueden ser programados y diseñados mono o multitarea (13) (14).

Estos agentes conversacionales, en dependencia de su desarrollo, comparten un grupo de características con los agentes inteligentes como sociabilidad, personalidad y pro-actividad. Ello ha permitido que el desarrollo de estos tipos de agentes evolucione hasta tener nuevas características (en dependencia del entorno donde estén y su objetivo) como movilidad, reactividad e incluso autonomía; incrementando así su usabilidad tanto cultural, educativa como en el aspecto económico. Un ejemplo de esto se evidencia en los agentes conversacionales en el “*Virtual English*”.

Hay varios problemas pendientes en cuanto a la “inteligencia” del agente conversacional. Sin duda uno de los más evidentes es la interpretación pragmática de la conversación. Los chatbots no pueden seguir el hilo

de una conversación por mucho tiempo (dando respuestas que satisfagan a su interlocutor) ya que no logran resolver las referencias anafóricas<sup>8</sup> ya sean de tipo correferencial, de sentido o elípticas (14). (Para comprender los tipos anafóricos ver **GLOSARIO**).

### **Agentes conversacionales en el “*Virtual English*”**

En la UCI se han desarrollado agentes conversacionales con distintas funcionalidades dentro del entorno “*Virtual English*”, ubicados en lugares específicos del entorno con el fin de realizar distintas actividades como la corrección de errores ortográficos y gramaticales, en función de alcanzar un nivel A3 del idioma inglés (15) o elevar el grado de inmersión al orientar a los usuarios dentro del entorno (16).

Estos agentes conversacionales contienen características muy propias según su programación. Por ejemplo, en el caso del agente conversacional “*Virtual Guide*” (que tiene como objetivo elevar el grado de inmersión al orientar a los usuarios dentro del entorno, citado anteriormente), es un agente con la característica de autonomía, que le permite moverse por el entorno y ayudar a los estudiantes de inglés básico a orientarse en el mundo virtual para que desarrollen las funciones comunicativas correspondientes a preguntar y dar direcciones.

Para el desarrollo del presente trabajo se decidió confeccionar un agente conversacional, que brinde las características necesarias para dar cumplimiento al objetivo del presente trabajo.

#### **1.3.3 Agentes conversacionales de carácter comercial**

Si bien los *chatbots* aún tienen demasiados problemas para que sean adoptados como una tecnología sólida, hay casos puntuales en donde su utilización puede considerarse exitosa. Como ejemplos se pueden nombrar ciertos *chatbots* creados para el mensajero instantáneo de Microsoft (*Messenger* o *MSN*). Una empresa llamada IMI (*IM Interactive Corp*) desarrolló un intérprete de AIML<sup>9</sup> en C# para que sea posible crear en este lenguaje, *chatbots* para *Microsoft Messenger* y ya se dispone de una lista de *chatbots*

---

<sup>8</sup> **Anáfora:** Referencia a un término o a una parte anterior del discurso (48).

<sup>9</sup> Lenguaje de Marcado para la Inteligencia Artificial (*Artificial Intelligence Mark-Up Language*)

existentes para MSN, incluyendo algunos como Robin o Wilma creados para un propósito específico distinto del mero interés académico (14).

Algunos ejemplos de estos chatbots son mostrados a continuación:

- Robin es un *chatbot* creado por el Ministerio de Sanidad y Consumo de España. Su función es informar a los jóvenes a través de Messenger sobre enfermedades de transmisión sexual y consumo de alcohol.
- Microsoft creó su propio *chatbot*, *Encarta Instant Answer*, para promocionar su enciclopedia encarta, sin embargo con la finalización de dicho producto “por parte de Microsoft” el *chatbot* fue dado de baja.
- Algunos sistemas informáticos como *Remedy*, utilizan *chatbots* para que los usuarios puedan realizar consultas comunes de manera rápida, estos *chatbots* simulan un operario para la atención al usuario.

### 1.3.4 Lingüística Computacional

Es un campo científico que trabaja en la incorporación de la habilidad de manejo del lenguaje humano a los ordenadores. En la informática la lingüística computacional es asociada a la Inteligencia Artificial que procura la toma de decisiones utilizando el lenguaje natural.

“Desdel punto de vista de la vinculación a la lingüística, la lingüística computacional puede ser considerada una sub-disciplina de la lingüística teórica, en tanto uno de sus objetivos es la elaboración de modelos formales del lenguaje humano.

La lingüística computacional forma parte de las denominadas industrias de la lengua, un sector industrial que proporciona datos y programas informáticos aplicados al tratamiento del lenguaje. Datos tales como diccionarios electrónicos e impresos en bancos de datos terminológicos y diccionarios, y programas tales como sistemas de traducción automática, interfaces de consultas de bases de datos en lenguaje natural, correctores ortográficos y estilísticos.

Esta misma actividad, cuyos resultados se plasman en las aplicaciones lingüísticas de la informática, vincula la lingüística computacional con la lingüística aplicada, una rama dedicada a la aplicación de los resultados y métodos de la investigación lingüística a campos tales como la enseñanza de idiomas, la traducción e interpretación, o la logopedia” (17).

### **Procesamiento de lenguaje natural (NLP por sus siglas en inglés)**

Es una disciplina de la IA que se encarga de encontrar los mecanismos computacionales para la comunicación entre las personas y las máquinas utilizando el lenguaje natural.

Un ejemplo de esta aplicación es el sistema computacional llamado *Slot Unification Parser for Anaphora Resolution* (SUPAR). Este sistema resuelve la anáfora pronominal en español con 81%, e inglés con un 74% de éxito pudiéndose extender fácilmente a otros idiomas (17).

#### **1.3.5 Sistemas Basados en Reglas**

Es una técnica de IA que define una serie de acciones a ser ejecutadas para una situación particular donde la resolución se obtiene a partir del proceso de razonamiento de un motor de inferencia, el conocimiento de resolución de problemas está almacenado habitualmente como reglas de producción. Una regla es una sentencia de tipo: si – entonces (*if– then*). El “si” especifica el dato o serie de datos que hace que se aplique la regla. El “entonces” especifica las acciones que deben ser tomadas que hacen que la regla se aplique (18).

#### **1.3.6 Razonamiento Basado en Casos**

Es un paradigma de resolución de problemas que se caracteriza por no confiar únicamente en el conocimiento general del dominio del problema, o realizar asociaciones a lo largo de relaciones entre descripciones del problema y conclusiones.

Este paradigma es capaz de utilizar conocimientos específicos de experiencias previas, es decir, situaciones de un problema concreto (casos). Un problema nuevo es resuelto cuando se encuentra un caso similar y se reutiliza en la resolución del nuevo problema.

Otra de las características es que tiene un aprendizaje incremental, puesto que esta experiencia se guarda cada vez que se resuelve un problema, pasando a estar disponible para resolver futuros problemas (19).

## 1.4 Herramientas y lenguajes

Para todo trabajo de desarrollo informático es imprescindible el rol que juegan las herramientas y tecnologías. Estas varían en cuanto a entorno de trabajo, lenguaje de programación utilizado, bibliotecas utilizadas, objetivo de uso, entre otros aspectos. Pero más importante y a pesar de sus diferencias, existen herramientas y tecnologías que están diseñadas para posibilitar el trabajo en conjunto (como un equipo de desarrollo).

En este epígrafe daremos un repaso a las tecnologías de inteligencia artificial, herramientas, servidores, bibliotecas y lenguajes de programación utilizados para cumplir con el desarrollo de la aplicación a la que está orientado el trabajo.

### 1.4.1 Lenguaje AIML (*del inglés Artificial Intelligence Mark-up Language*)

Es un lenguaje etiquetado<sup>10</sup> de programación basado en XML<sup>11</sup> que utiliza la técnica de IA de los Sistemas Basados en Reglas. Fue diseñado específicamente para ayudar en la creación de la primera entidad *chatbot* informática de lenguaje artificial online o A.L.I.C.E. (del inglés *Artificial Linguistic Internet Computer Entity Chatterbot*). Está especializado en la creación de agentes software con lenguaje natural, conocidos como *Alicebots*. Fue desarrollado por el Dr. Richard Wallace y la comunidad de código abierto *Alicebot* entre los años 1995 y 2000 (20).

#### Componentes de AIML

Cuenta de tres componentes fundamentales para establecer las conversaciones entre el usuario y el agente conversacional, los cuales se muestran a continuación (20):

---

<sup>10</sup> Es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación (42).

<sup>11</sup> Lenguaje de Marcas Extensible (XML por sus siglas en inglés).

- **Categorías:** son las unidades fundamentales de conocimiento; agrupan dos componentes patrón (*pattern*) y plantilla (*template*) los cuales relacionan un mensaje de entrada del usuario y uno de salida del *chatbot* respectivamente y que normalmente se codifican en ese orden.
- **Patrones:** son las entradas del usuario mediante textos que serán reconocidas como patrones.
- **Plantillas:** si un patrón coincide con una categoría y es el que más concuerda, entonces la respuesta del *chatbot* será mostrada desde la plantilla, donde contienen las respuestas y otros componentes de AIML que permiten personalizar dicha respuesta.

### **Biblioteca AIMLbot.dll**

Es una biblioteca de código abierto que permite utilizar el lenguaje AIML desde código C# mediante interfaces que facilitan el procesamiento de mensajes mediante AIML (20).

Su principio de funcionamiento parte del conocimiento sobre las posibles preguntas y respuestas con las que cuenta para establecer una conversación (esto es, las rutas a los ficheros AIML). Una vez empieza la interacción con el usuario será el vínculo entre los patrones AIML (patrones de conversación almacenados) y el mecanismo que le permite la interacción al usuario.

Para una mejor comprensión del funcionamiento de la biblioteca *AIMLbot* para establecer las conversaciones ver la Figura. 2.

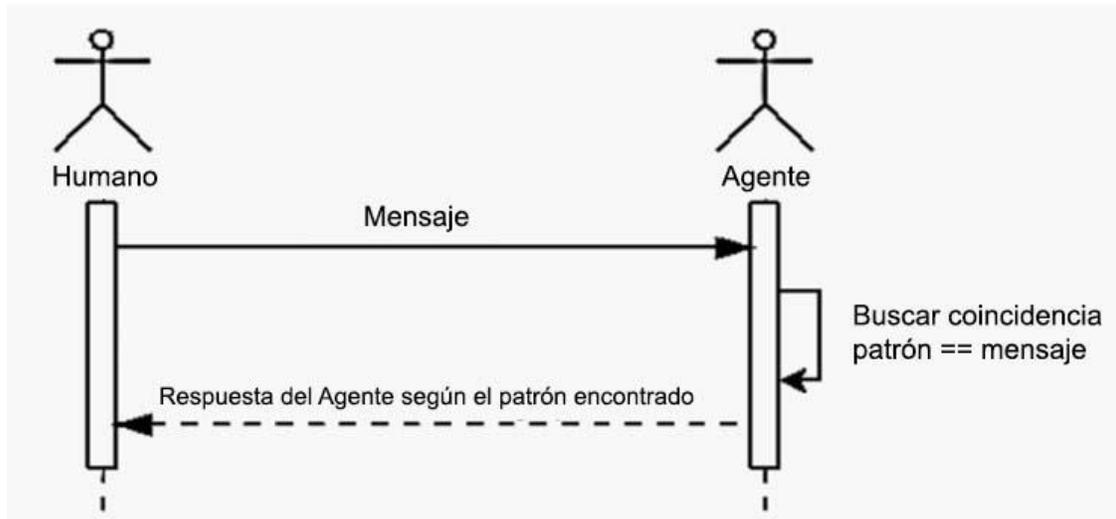


Figura 2. Proceso de la conversación entre humano-agente<sup>12</sup>.

#### 1.4.2 Apache

Es un servidor Web que implementa el protocolo HTTP y es de código abierto, ejecutable en sistemas operativos como Linux, MacOS X y Windows. Las capacidades de este servidor pueden ser ampliadas incorporándole nuevos módulos, pues su diseño modular es altamente configurable. Presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero ha sido criticado por la falta de una interfaz gráfica que ayude en su configuración (21).

#### 1.4.3 MySQL

Es un sistema de gestión de bases de datos relacionales liberado para la licencia GPL de GNU y es el sistema de *OpenSource* más popular debido a su rendimiento, alta fiabilidad y facilidad de uso. El sistema lo desarrolla, distribuye y soporta MySQL AB. Esta es una compañía *OpenSource* de segunda generación que une los valores y metodología *OpenSource* con un exitoso modelo de negocio.

<sup>12</sup> Estudio y mejora de la usabilidad de un mundo virtual híbrido: interacción humano-agente”, Figura 6, pág. 39 (43)

El *software* de bases de datos MySQL es un sistema cliente/servidor que consiste en un servidor SQL multi-hilo que trabaja con diferentes programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones (22).

### 1.4.4 XAMPP

Es un servidor independiente de plataforma, de *software* libre y que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X (23).

### 1.4.5 Sistemas Gestores de Base de Datos (SGBD)

Son un conjunto de programas que permiten crear y mantener una base de datos, asegurando la integridad, confidencialidad y seguridad (24). Un ejemplo de SGBD es el *EMS SQL Manager for MySQL* utilizado en el desarrollo de este trabajo para la gestión de la base de datos de la aplicación.

### 1.4.6 EMS SQL Manager for MySQL

Es una herramienta privativa desarrollada por *SQL technology* para la administración y desarrollo de bases de datos MySQL desde la versión 3.23. Este gestor, además, cuenta con funcionalidades como disparadores (*triggers*), vistas, procedimientos almacenados, funciones, entre otras (25).

### 1.4.7 Entorno Integrado de Desarrollo (IDE por sus siglas en inglés)

Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva sin necesidad de trabajo orientado a archivos de texto (26).

#### 1.4.8 Microsoft Visual Studio

Es un IDE para el sistema operativo *Windows* fabricado por la empresa de desarrollo de *software* Microsoft. Visual Studio soporta lenguajes como C#, C++ y Visual Basic. Net, además brinda el entorno ASP.NET para el desarrollo Web (27).

Para el desarrollo del presente trabajo se utilizaron como tecnologías análogas, el lenguaje de programación C#, la biblioteca. NET OpenMetaverse para el desarrollo con el gestor de aplicaciones 3D OpenSimulator y la biblioteca OpenNLP para el procesamiento de lenguaje natural.

#### 1.4.9 Lenguaje de Programación C#

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma. NET, que después fue aprobado como un estándar por la ECMA<sup>13</sup> (ECMA-334) e ISO<sup>14</sup> (ISO/IEC 23270). Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma. NET, aunque incluye mejoras derivadas de otros lenguajes (28).

Ha sido base de desarrollo para muchas bibliotecas y programas utilizados en el desarrollo del presente trabajo como la aplicación SharpNLP que será explicada más adelante.

#### Biblioteca LibOpenMetaverse

Es una biblioteca. Net basada en la arquitectura cliente/servidor, usada para acceder y crear mundos virtuales en 3D, es compatible con Microsoft. NET Framework 2.0 y versiones posteriores, así como Mono 1.9 y posterior.

La biblioteca consta de varias dependencias (29):

- **OpenMetaverse.dll:** Es el núcleo de la biblioteca OpenMetaverse, que puede ser usado para crear aplicaciones clientes y servidor.
- **OpenMetaverseTypes.dll:** Usada para funciones específicas 3D y bibliotecas matemáticas.

---

<sup>13</sup> Asociación Europea de Manufactura de Computadoras (*European Computer Manufacturers Association*)

<sup>14</sup> Organización Internacional de Normalización (*Normalitation International Organization*)

- **OpenMetaverse.StructuredData.dll:** Conjunto de librerías para dar soporte a *Linden Lab Structured Data* (LLSD) y *Java Script Object Notation* (JSON).
- **OpenJpeg:** Un envoltorio. NET para codificar y decodificar los datos JPEG2000.

## 1.5 SharpNLP

Es una colección de herramientas para el procesamiento del lenguaje natural, escritas en C#. Internamente utilizan la biblioteca OpenNLP que es el nombre de un grupo de proyectos de código abierto relacionados con el procesamiento del lenguaje natural (NLP) y el nombre de una biblioteca de herramientas de NLP escrito en Java (30).

### Biblioteca OpenNLP

Es el nombre de un grupo de proyectos de código abierto relacionados con el NLP, y el nombre de una biblioteca de herramientas de NLP escrita por *Jason Baldridge, Tom Morton, y Gann Bierner*. Contiene una biblioteca de acceso de base de datos *WordNet*<sup>15</sup> llamado *SharpWordNet* (30).

En adición, este es un conjunto de herramientas basadas en aprendizaje de máquinas para el procesamiento de texto en lenguaje natural. Es compatible con las tareas más comunes de NLP, como divisor frase, tokenizer (etiquetador), etiquetador de parte de discurso, *chunker* (utilizado para encontrar anotaciones sintácticas no recursivas), analizador, buscador de nombres, herramienta de correferencia e interfaz a la base de datos léxica *Wordnet*. Estas tareas suelen ser necesarias para crear servicios más avanzados de procesamiento de texto (31).

## 1.6 Metodologías de software

Las metodologías de software según la disciplina de ingeniería de software, es un marco de trabajo en el cual se aplican diferentes técnicas, procedimientos y artefactos ingenieriles con el objetivo de planificar, desarrollar y controlar la implementación exitosa de un software informático (32).

---

<sup>15</sup> Base de datos léxica del idioma inglés para aplicaciones de inteligencia artificial liberada bajo licencia BSD (44).

De estas técnicas de desarrollo, las llamadas metodologías ágiles están compuestas por un grupo de métodos basados en el progreso iterativo e incremental donde los requerimientos y soluciones evolucionan mediante la colaboración de grupos auto-organizados y multidisciplinarios. Estos métodos enfatizan las comunicaciones cara a cara en vez de la documentación. También destacan que el software funcional es la primera medida del progreso, combinado con la preferencia por las comunicaciones cara a cara (32).

Un ejemplo de estas técnicas es la metodología de desarrollo Programación Extrema, también conocida por sus siglas XP.

### 1.6.1 Programación Extrema (XP)

Es una metodología de desarrollo de la ingeniería de *software* formulada por Kent Beck. Es el más destacado de los procesos ágiles de desarrollo de software y al igual que éstos, XP se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Los valores originales de la programación extrema son: simplicidad, comunicación, retroalimentación (*feedback*) y coraje. Un quinto valor, respeto, fue añadido en la segunda edición de *Extreme Programming Explained* (32).

### 1.7 Ingeniería de Software Orientada hacia Agentes

Es una metodología de software que “realiza las tareas de la ingeniería de software usando como clave de abstracción el concepto de agente” (33). Surgido en respuesta a la demanda de los requerimientos de los sistemas multiagente. Estos sistemas son esencialmente distribuidos y están compuestos por un conjunto de agentes de software que interactúan a fin de satisfacer sus objetivos.

“La ingeniería de software orientada hacia agentes es una disciplina reciente que aplica técnicas de ingeniería del software e inteligencia artificial, busca la construcción del anhelado software que piensa o el software con inteligencia. Este paradigma ha tomado cada día más importancia, ya que posee unas arquitecturas y características propias que no encajan fácilmente en ninguno de los paradigmas de ingeniería del software clásicos. Esta proviene de la fusión entre la inteligencia artificial, la tecnología de objetos y de componentes, sistemas distribuidos, sociedad y la ciencia del comportamiento entre otros.

De la *tecnología de objetos* rescata: mensajes entre objetos, encapsulamiento, clases y herencia. De los *sistemas distribuidos* considera: la distribución de datos y procesos, conectividad, redes, protocolos, interoperabilidad e Internet. De la *sociedad y la ciencia del comportamiento*: interacciones entre humanos, roles, conductismo, cognitivismo. De la *inteligencia artificial*: representación del conocimiento, razonamiento, aprendizaje y el enfoque de agentes inteligentes.

Como casi todo enfoque informático para la solución de problemas, los sistemas multiagente proponen ayudas metodológicas de ingeniería de software. En este caso metodologías de ingeniería de software orientada a agentes y las notaciones respectivas” (34).

Algunas de las metodologías y notaciones de ingeniería de software orientada a agentes, son (34):

- Vocales de Yves Demazeau, es una de las primeras propuestas en el área, y considera la concepción de sistemas multiagente desde varios puntos de vista, correspondientes a las vocales: Agente, Entorno, Interacciones, y Organización.
- GAIA de Michael Wooldridge y Nick Jennings de la Universidad de Southampton, propone cómo realizar un análisis basado en roles del sistema multiagente.
- MASE de Scott A. Deloach, propone agentes como extensiones de objetos y proporciona la herramienta *AgentTool* para análisis, diseño e implementación.
- *AgentUML* de James Odell, propone una notación, extendiendo el Lenguaje de Modelado unificado, para especificar protocolos de comunicación entre agentes.
- MADKiT es una herramienta de desarrollo, propuesta por Jacques Ferber, basada en el paradigma Agente-Rol-Organización de la metodología *Aalaadin*.
- ADELFE del grupo IRIT de la Universidad de Toulouse, trata especialmente los temas de cooperación entre agentes.
- INGENIAS del grupo GRASIA de la UCM, extiende la metodología MESSAGE y proporciona un conjunto de herramientas para modelar y generar código de sistemas multiagente.
- Mas-CommonKADS de Carlos Iglesias en la UPM extiende la metodología *CommonKADS*, para sistemas expertos, a agentes, utilizando estructuración orientada a objetos y lenguajes de especificación de protocolos.

### 1.8 Criterio de selección de la metodología de desarrollo

Para el desarrollo del presente trabajo se utilizó la metodología de desarrollo ágil XP. Esta decisión fue tomada teniendo en cuenta los siguientes criterios como:

- Pequeño grupo de desarrollo.
- Alta interacción con el cliente.
- Basta experiencia en el desarrollo de agentes con esta metodología por parte del Centro Entornos Interactivos 3D (VERTEX) de la UCI.

### 1.9 Conclusiones Parciales

Es importante destacar que el agente conversacional o chatbot que se desea realizar, puede contener características pro-activas, personalidad, veracidad y sociabilidad. Además estará dentro de un entorno virtual realizando actividades sociales y tomando decisiones al nivel que su “inteligencia” lo permite y respondiendo cosas en las que disponga el conocimiento para hacerlo. Esto permite concluir que a pesar de no ser un agente inteligente el creado en este trabajo, si contiene características de este y puede serlo con un desarrollo más profundo.

## CAPÍTULO 2: EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO DE LA SOLUCIÓN PROPUESTA

### 2.1 Introducción

Para lograr la correcta elaboración de la aplicación, es importante un trabajo de investigación previo que propicie la descripción del proceso de desarrollo teniendo en cuenta sus funcionalidades. En este capítulo se describen las fases de exploración, planificación y diseño para la elaboración de este proyecto. Entre los elementos que se podrán observar están las historias de usuarios, plan de iteraciones, las tarjetas CRC, entre otros de igual importancia.

### 2.2 Proceso de captura de requisitos

La captura de requisitos es una fase importante para el desarrollo de la solución, donde son tomados de manera documentada y representan el contenido y las funcionalidades del producto.

En el sistema propuesto se definieron requisitos funcionales y no funcionales. Los requisitos funcionales responden a las funcionalidades que debe cumplir el agente conversacional implementado. Los requisitos no funcionales son los que no describen información a guardar, ni funciones a realizar (33).

#### 2.2.1 Requisitos funcionales

Son aquellas características requeridas para el funcionamiento de cada agente conversacional y se expresa (por lo general) en una declaración en forma verbal en los encuentros previos que se tendrán con el cliente (34).

A continuación se muestran los requisitos funcionales para el desarrollo del agente conversacional:

- **RF 1.** Saludar al posible cliente.
- **RF 2.** Preguntar por lo que desea el cliente.
- **RF 3.** Proponer productos y ofrecer alternativas si alguno de estos faltase.
- **RF 4.** Ofrecer información sobre el precio de los productos.

- **RF 5.** Informar el precio total de la compra del cliente.
- **RF 6.** Realizar oferta de reducción de precios si el cliente compra varios productos a la vez.
- **RF 7.** Realizar ofertas de reducción de precios si el cliente compra varios productos de un mismo tipo a la vez.
- **RF 8.** Restituir vuelto al pago dado por el usuario.

### 2.2.2 Requisitos no funcionales

Los requisitos no funcionales son atributos de calidad que especifican criterios que pueden usarse para juzgar la operación del sistema en lugar de sus comportamientos específicos (32).

A continuación se muestran los requisitos no funcionales:

- **RnF 1. Usabilidad:** Los usuarios que interactúen con el sistema deberán consultar previamente el manual de usuario del sistema para un mejor entendimiento del trabajo con OpenSim e *Imprudence Viewer*.
- **RnF 2. Hardware:** Los requerimientos mínimos de hardware que se necesitan son: 1Gb de memoria RAM o superior, una velocidad de Microprocesador de 2.5 GHz o superior, una tarjeta de video de 128 Mb y una capacidad de almacenamiento en disco de 5 Gb o superior.
- **RnF 3. Software:** Los usuarios del sistema deben contar con el sistema operativo Windows 7 o superior. Soporte para la plataforma. NET. Además de contar con un gestor de base de datos y servidor MySQL y el visor 3D *Imprudence* como cliente del servidor de aplicaciones 3D OpenSim.
- **RnF 4. Seguridad:** Los usuarios del sistema tendrán acceso solo a las funcionalidades que les corresponden en dependencia del tipo de usuario.

### 2.3 Propuesta de solución

Se propone el desarrollo de un agente conversacional en la zona del mercado del metaverso “*Virtual English*”, mostrada en la Imagen 2. A través de este agente los usuarios podrán dialogar y realizar actividades enmarcadas en el área de la compra y venta de productos.

Este agente realizará análisis lingüísticos basados en analizadores que serán utilizados en toda la conversación. Su funcionamiento se explica a continuación.



*Imagen 2. Ubicación del supermercado en el metaverso "Virtual English".*

## 2.4 Funcionamiento de la propuesta de solución

Para realizar el análisis lingüístico del texto introducido por el usuario, el agente conversacional dividirá su funcionamiento en tres partes fundamentales:

1. Entrada de datos
2. Procesamiento de datos
3. Salida de datos

### 2.4.1 Entrada de datos

La entrada de datos se realiza a través de una ventana de chat disponible en el metaverso para interactuar con el agente conversacional. Mediante esta ventana el usuario puede entrar cualquier texto (preferentemente sobre actividades de venta y adquisición de productos), siendo esta entrada inmediatamente procesada.

2.4.2 Procesamiento de datos

Una vez entrados los datos, el procesamiento se realiza a través de tres analizadores lingüísticos de una manera secuencial Figura 3: analizador léxico, analizador de funciones comunicativas y analizador gramatical respectivamente. Así se determinará si debe responder al texto entrado por el usuario según el conocimiento que posee el agente o enviar un mensaje correctivo en caso de encontrar algún error en la estructura del texto. Para una mayor comprensión de las variantes que toma el flujo de datos mirar la Figura 4.

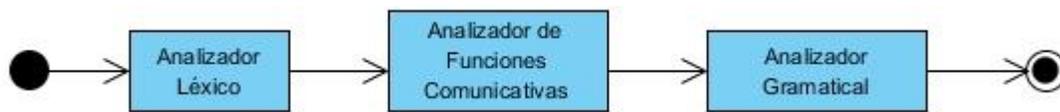


Figura 3. Secuencia de uso de los analizadores.

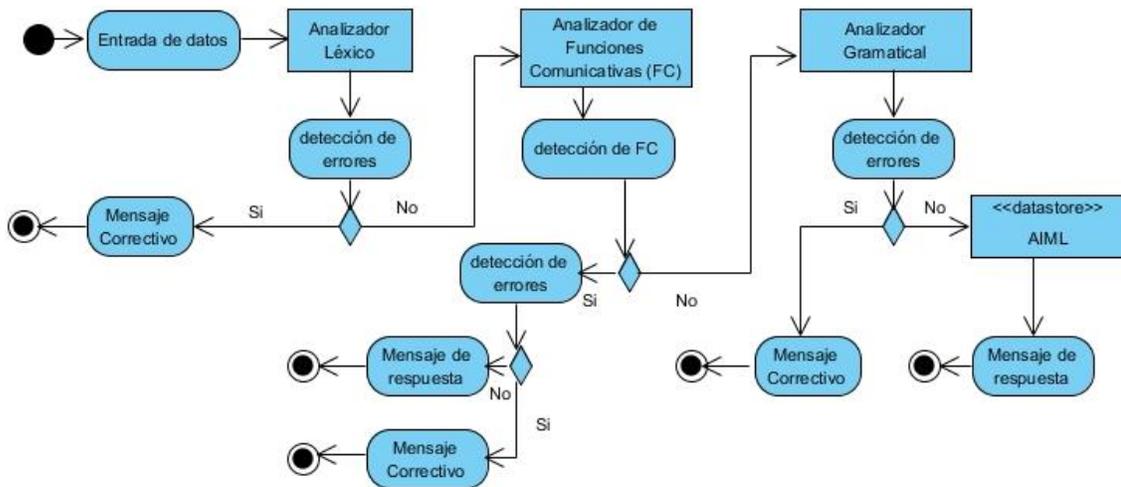


Figura 4. Flujo de actividades para el procesamiento del texto entrado por el usuario.

En el desarrollo de esta aplicación se reutilizaron dos de los analizadores gramaticales citados anteriormente (léxico y gramatical), puesto que ambos se han utilizado en la UCI para la confección de trabajos relacionados con este tema y se ha probado su correcto funcionamiento (epígrafe Agentes conversacionales en el “Virtual English”).

### **Analizador Léxico**

Este es un analizador que se encarga básicamente de comprobar si todas las palabras del texto entrado por el usuario pertenecen al alfabeto del idioma inglés. Utiliza un mecanismo intermedio mediante el cual se realiza conexión, desconexión y consultas a una base de datos MySQL que contiene palabras en inglés.

Tras una comparación entre las palabras entradas en el texto y las contenidas en la base de datos, se envía un mensaje correctivo - al usuario - de las palabras que no pertenecen al idioma. Es válido incluir en este criterio las palabras en inglés mal escritas, que por consiguiente conllevan a un error léxico.

De no encontrar ninguna falta léxica continuará el flujo del procesamiento hacia el analizador de funciones comunicativas.

### **Analizador de Funciones Comunicativas**

Este analizador se encarga de identificar en el texto proporcionado por el usuario, estructuras del lenguaje referentes a actividades de oferta y adquisición. Para identificar estas estructuras se utiliza un sistema basado en reglas del tipo si - entonces (*if – then*).

Pertenciente a cada texto introducido que coincide con alguna regla, existen variables que necesitan ser gestionadas porque de estas dependen las respuestas que se brindarán al usuario. Estas variables son producto y cantidad de productos a comprar, y pueden variar su presencia en dependencia del texto en procesamiento. La gestión de estas variables dependen de ciertos parámetros como existencia, costo o cantidad existente; utilizados de distintas maneras (o NO si no es necesario) en las variables.

Para poner un ejemplo, el procesamiento de datos a un posible texto entrado por el usuario como *“How many pounds of mangoes do you have?”*, comienza cuando el analizador busque coincidencias entre la estructura *“How many pounds”* y el conjunto de reglas definidas. Una vez encontrada una regla que coincida se analizará el o los productos a los que se hace mención en el texto (*mangoes*) con los parámetros correspondientes. Así se llegará a una respuesta del agente conversacional una vez se termine el análisis.

Para preguntas como *“I need to buy twenty pounds of tomatoes, nine pounds of mangoes and twenty five pounds of potatoes”*. El analizador después de encontrar la regla que coincida con la estructura *“I need to*

*buy*". Procederá a analizar el resto de variables en la pregunta tomando como punto de referencia la posición donde se encuentra la cadena "*pounds of*".

Hay que tener en cuenta que en dependencia de los valores que resulten del análisis pueden surgir nuevas variables como: la cantidad de compras de productos de diversos tipos y la cantidad de productos de un mismo tipo que se compran en una misma pregunta. Son estas nuevas variables las que permiten realizar ofertas de descuento.

Estas entradas antes mencionadas son solo algunos ejemplos de textos entrados a los que el analizador de funciones comunicativas hace un procesamiento en busca de la respuesta más adecuada. De no encontrar una regla que se relacione con el texto en procesamiento pasará a realizarse el análisis gramatical.

### **Análisis Gramatical**

Es la última parte del procesamiento de datos y se encarga de velar por que el texto cuente con una estructura gramatical correspondiente a las estructuras definidas en el idioma. Ejemplos de funciones realizadas para el análisis gramatical son: determinar si la oración tiene un sujeto, número y persona del sujeto, revisar si el sujeto coincide en número y persona con el verbo (o la forma verbal), revisar el tiempo del verbo (o la forma verbal), revisar si el verbo (o forma verbal) coincide con el tiempo de la oración, entre otros aspectos que se tienen en cuenta para un correcto análisis.

#### **2.4.3 Salida de datos**

El proceso de salida de datos es la última parte que describe el funcionamiento del sistema. Esta salida se hará a través de la ventana de chat disponible en el metaverso (la usada en la entrada de datos), permitiendo visualizar tanto los mensajes correctivos enviados por los analizadores como las respuestas a las preguntas correctamente validadas. Esta ventana de chat permite además continuar con la interacción del usuario con el agente conversacional.

### **2.5 Usuarios relacionados con el sistema**

En el sistema propuesto se establecieron dos roles para el acceso al entorno virtual y a cada agente conversacional: administrador y usuario registrado. En la tabla 1 se muestran los roles definidos con una breve descripción de cada rol:

Tabla 1. Roles definidos para el acceso al entorno Virtual.

Rol	Descripción
<b>Administrador</b>	El usuario administrador será la representación del avatar, este tiene el control total sobre todas las funciones del sistema y las animaciones, el estado y la información de venta brindada por el bot
<b>Usuario registrado</b>	Son los avatares que interactuarán con el agente conversacional y con todas las funcionalidades de este.

## 2.6 Fase de exploración y definición del alcance del proyecto

En la fase de exploración se define el alcance real del proyecto y las necesidades del cliente se establecen a partir de las diferentes historias de usuarios para su posterior implementación. Al mismo tiempo el equipo de desarrollo se estudia las herramientas, tecnologías y las prácticas a ser utilizadas durante el desarrollo. Todo el proceso antes descrito puede tomar un tiempo, que varía según se realicen cada una de las tareas planificadas (35).

### 2.6.1 Historias de Usuario (HU)

Son una representación de los requisitos del sistema y se utilizan para transmitirle estos requisitos al cuerpo de desarrolladores del software. Las HU utilizan el lenguaje común del usuario en su elaboración y son utilizadas para estimar el tiempo que el equipo de desarrollo tomará para realizar la entrega.

Las HU se utilizan en las metodologías de desarrollo ágiles para especificar los requisitos y se acompañan de las discusiones con los usuarios y las pruebas de validación (36).

Tabla 2. HU1 Configuración del servidor de aplicaciones 3D, OpenSim

Historias de Usuario # 1	
<b>Nombre:</b> Configuración del servidor OpenSim.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Baja
<b>Estimación de Tiempo:</b> 1 día	<b>Número de Iteración:</b> 1

<b>Descripción:</b> Configuración del servidor XAMPP para iniciar los servidores OpenSim, Apache y MySQL. Sobre estos se ejecutan el “ <i>Virtual English</i> ”, XoopsCube para configurar las cuentas del metaverso y la base de datos MySQL.
<b>Observaciones:</b>

Tabla 3. HU2 Personalización del avatar en el metaverso

Historias de Usuario # 2	
<b>Nombre:</b> Personalización del avatar en el metaverso.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Baja
<b>Estimación de Tiempo:</b> 1 día	<b>Número de Iteración:</b> 1
<b>Descripción:</b> Se hará uso del gestor de contenido XoopsCube para la creación del usuario que estará asignado al avatar en el mundo virtual. Una vez accedido al mundo virtual se personalizará de manera gráfica el agente conversacional.	
<b>Observaciones:</b>	

Tabla 4. HU3 Configuración del gestor de base de datos y el servidor MySQL

Historias de Usuario # 3	
<b>Nombre:</b> Configuración del gestor de base de datos y servidor MySQL	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Baja
<b>Estimación de Tiempo:</b> 1 día	<b>Número de Iteración:</b> 1
<b>Descripción:</b> Se configurará el gestor de base de datos <i>EMS SQL Manager para MySQL</i> y se conectará con el servidor MySQL del metaverso OpenSim. Este SGBD trabajará sobre la base de datos que contiene las palabras del idioma inglés.	
<b>Observaciones:</b>	

Tabla 5. HU4 Implementación de la conexión del agente conversacional al servidor OpenSim

Historias de Usuario # 4	
<b>Nombre:</b> Implementación de la conexión del agente conversacional al servidor OpenSim.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Media
<b>Estimación de Tiempo:</b> 2 días	<b>Número de Iteración:</b> 1

<b>Descripción:</b> Se implementará la conexión del agente conversacional, con el que se podrá establecer el flujo de información con el servidor de aplicación 3D, OpenSim.
<b>Observaciones:</b>

*Tabla 6. HU5 Diseño de las técnicas de mercado inherentes al agente conversacional*

Historias de Usuario # 5	
<b>Nombre:</b> Diseño de las técnicas de mercado inherentes al agente conversacional.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Media
<b>Estimación de Tiempo:</b> 5 días	<b>Número de Iteración:</b> 1
<b>Descripción:</b> Se diseñarán las técnicas de negocio que debe tener el agente a la hora del diálogo con el cliente como: proponer y ofrecer productos, ofrecer información e informar el precio de los productos y realizar ofertas de reducción de precios. Estas actividades se realizarán teniendo en cuenta un conjunto de preguntas y respuestas definidas.	
<b>Observaciones:</b>	

*Tabla 7. HU6 Implementación de la funcionalidad de proponer productos y ofrecer alternativas*

Historias de Usuario # 6	
<b>Nombre:</b> Implementación de la funcionalidad de proponer productos y ofrecer alternativas	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimación de Tiempo:</b> 10 días	<b>Número de Iteración:</b> 1
<b>Descripción:</b> Se implementará la funcionalidad de proponer productos y ofrecer alternativas si alguno de estos productos faltase. Determinar qué productos vender, dependerá de la pregunta entrada por el usuario. Si esta pregunta es referente a productos con los que no se cuenta o se han agotado, se modificará la respuesta a una alternativa para comprar otros.	
<b>Observaciones:</b>	

*Tabla 8. HU7 Implementación de la funcionalidad de ofrecer información del precio de los productos*

Historias de Usuario # 7	
<b>Nombre:</b> Implementación de la funcionalidad de ofrecer información del precio de los productos.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimación de Tiempo:</b> 10 días	<b>Número de Iteración:</b> 2

<b>Descripción:</b> Se implementará una funcionalidad para ofrecerle información del precio de los productos que el usuario solicita mediante una pregunta. Estos precios varían en dependencia del tipo de producto en cuestión. Si este producto no está disponible por alguna razón, se incorporará el análisis hecho en la HU6.
<b>Observaciones:</b>

*Tabla 9. HU8 Implementación de la funcionalidad de informar al cliente del precio total de la compra*

Historias de Usuario # 8	
<b>Nombre:</b> Implementación de la funcionalidad de informar al cliente el precio total de la compra.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimación de Tiempo:</b> 10 días	<b>Número de Iteración:</b> 2
<b>Descripción:</b> Se implementará la funcionalidad de ofrecer información del precio total de la compra realizada por parte del cliente. Este precio varía en dependencia de la cantidad de productos comprados. Si la compra realizada por el cliente cuenta con distintos tipos de productos, o varios productos comprados de un mismo tipo (inclusive los dos aspectos anteriormente mencionados) en una misma compra, se realizarán las ofertas de descuento de las HU9 y HU11.	
<b>Observaciones:</b>	

*Tabla 10. HU9 Diseño de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado*

Historias de Usuario # 9	
<b>Nombre:</b> Diseño de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Media
<b>Estimación de Tiempo:</b> 5 días	<b>Número de Iteración:</b> 3
<b>Descripción:</b> Se diseñará la funcionalidad que permitirá darle una oferta de descuento al usuario, en dependencia de la cantidad de un mismo producto comprado. Si la cantidad máxima de un producto pedido es mayor que 20, se hará el descuento restando el 5% del precio total de la cantidad solicitada del producto en cuestión.	
<b>Observaciones:</b>	

*Tabla 11. HU10 Implementación de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado*

Historias de Usuario # 10
---------------------------

<b>Nombre:</b> Implementación de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimación de Tiempo:</b> 15 días	<b>Número de Iteración:</b> 3
<b>Descripción:</b> Se implementará la funcionalidad de realizar descuento acorde a la cantidad de un mismo producto comprado, teniendo en cuenta lo descrito en la HU9.	
<b>Observaciones:</b>	

Tabla 12. HU11 Diseño de la funcionalidad de ofertar descuento si el cliente compra una variedad de productos a la vez

Historias de Usuario # 11	
<b>Nombre:</b> Diseño de la funcionalidad de ofertar descuento si el cliente compra varios productos a la vez.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Media
<b>Estimación de Tiempo:</b> 5 días	<b>Número de Iteración:</b> 4
<b>Descripción:</b> Se definirá la funcionalidad que brinda la posibilidad de ofertar una reducción de precios realizada, en dependencia de la variedad de productos que el usuario desee comprar. Si esta cantidad es mayor que uno, se le descontará al costo total de la compra un N% de la variedad de productos comprados.	
<b>Observaciones:</b>	

Tabla 13. HU12 Implementación de la funcionalidad de ofertar descuento si el cliente compra una variedad de productos a la vez

Historias de Usuario # 12	
<b>Nombre:</b> Implementación de la funcionalidad de ofertar descuento si el cliente compra varios productos a la vez.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimación de Tiempo:</b> 15 días	<b>Número de Iteración:</b> 4
<b>Descripción:</b> Se implementará la funcionalidad que le permite al agente conversacional realizar una oferta de reducción de precios, si el cliente compra una variedad de productos a la vez.	
<b>Observaciones:</b> Al igual que la HU6, el resultado se hará visible en la actividad de compra del cliente.	

Tabla 14. HU13 Diseño de las respuestas del agente conversacional

Historias de Usuario # 13	
<b>Nombre:</b> Diseño de respuestas del agente conversacional.	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Media
<b>Estimación de Tiempo:</b> 10 días	<b>Número de Iteración:</b> 5
<b>Descripción:</b> Se definirá parte del diálogo del agente conversacional, con la mayor cantidad de posibles respuestas dentro del contexto de la compra y venta de productos.	
<b>Observaciones:</b>	

Tabla 15. HU14 Implementación del mecanismo de respuestas del agente conversacional con ayuda de la herramienta AIML

Historias de Usuario # 14	
<b>Nombre:</b> Implementación del mecanismo para las respuestas del agente conversacional con ayuda de la herramienta AIML	
<b>Prioridad:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Estimación de Tiempo:</b> 15 días	<b>Número de Iteración:</b> 5
<b>Descripción:</b> Se Implementará el mecanismo de respuestas del agente conversacional con ayuda de la herramienta AIML, permitiendo que realice una conversación aceptable e instructiva.	
<b>Observaciones:</b>	

## 2.7 Fase de planificación de las iteraciones

Es una de las etapas más importantes durante el desarrollo de toda aplicación. Se basa en definir el orden a seguir de las iteraciones del proyecto, cada implementación y cada entrega que se vaya a realizar del producto en sí, es decir, cada parte entregable del sistema en general, para lo cual serán necesarios algunos encuentros con el cliente (37).

### 2.7.1 Tareas de ingeniería

También conocidas como tareas de implementación, son actividades en las que se descomponen las historias de usuario, y determinan en qué iteración se realizará a mano de un desarrollador específico (37). A continuación se muestran las tareas de ingeniería realizadas para el desarrollo del presente trabajo.

*Tabla 16. Tarea de configuración del servidor de aplicaciones 3D, OpenSim*

Tarea de Ingeniería	
<b>Número de Tarea:</b> 1	
<b>Nombre de Tarea:</b> Tarea de personalización del avatar en el metaverso.	
<b>Tipo de Tarea:</b> Configuración	<b>Puntos Estimados:</b> Baja
<b>Fecha de Inicio:</b> 1 / 3 / 2015	<b>Fecha Fin:</b> 1 / 3 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se configurará el servidor de aplicaciones 3D, y los servidores Apache, MySQL para el XoopsCube y la base de datos de palabras en inglés respectivamente.	

*Tabla 17. Tarea de personalización del avatar en el metaverso*

Tarea de Ingeniería	
<b>Número de Tarea:</b> 2	
<b>Nombre de Tarea:</b> Tarea de personalización del avatar en el metaverso.	
<b>Tipo de Tarea:</b> Diseño	<b>Puntos Estimados:</b> Baja
<b>Fecha de Inicio:</b> 2 / 3 / 2015	<b>Fecha Fin:</b> 2 / 3 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se hará uso del gestor de contenido XoopsCube para la creación del usuario que estará asignado al avatar en el mundo virtual. Una vez accedido al mundo virtual se personalizará el agente conversacional.	

*Tabla 18. Tarea de configuración del SGBD y el servidor MySQL*

Tarea de Ingeniería	
<b>Número de Tarea:</b> 3	
<b>Nombre de Tarea:</b> Tarea de configuración del SGBD y servidor MySQL.	
<b>Tipo de Tarea:</b> Configuración	<b>Puntos Estimados:</b> Medio
<b>Fecha de Inicio:</b> 3 / 3 / 2015	<b>Fecha Fin:</b> 3 / 3 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	

**Descripción:** Se configurará el gestor de base de datos EMS SQL Manager para MySQL y se conectará con el servidor MySQL del metaverso OpenSim. Este SGBD trabajará sobre la base de datos que contiene las palabras del idioma inglés.

*Tabla 19. Tarea de implementación de la conexión del agente conversacional al servidor OpenSim*

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 4	
<b>Nombre de Tarea:</b> Tarea para la Implementación de la conexión del agente conversacional al servidor OpenSim.	
<b>Tipo de Tarea:</b> Implementación	<b>Puntos Estimados:</b> Medio
<b>Fecha de Inicio:</b> 4 / 3 / 2015	<b>Fecha Fin:</b> 5 / 3 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se implementará la conexión del agente conversacional con el que se podrá realizar el flujo de información con el servidor de aplicación 3D, OpenSim.	

*Tabla 20. Tarea de definición de las técnicas de mercado inherentes al agente conversacional*

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 5	
<b>Nombre de Tarea:</b> Tarea de definición de las técnicas de mercado inherentes al agente conversacional.	
<b>Tipo de Tarea:</b> Diseño	<b>Puntos Estimados:</b> Media
<b>Fecha de Inicio:</b> 6 / 3 / 2015	<b>Fecha Fin:</b> 10 / 3 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Estas técnicas serán diseñadas teniendo en cuenta las actividades más comunes realizadas en un mercado como: comprar, preguntar por productos, preguntar precios, preguntar por ofertas, entre otras.	

*Tabla 21. Tarea de implementación la funcionalidad de proponer productos y ofrecer alternativas*

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 6	

<b>Nombre de Tarea:</b> Tarea para implementar la funcionalidad de proponer productos y ofrecer alternativas.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> Alta
<b>Fecha de Inicio:</b> 11 / 3 / 2015	<b>Fecha Fin:</b> 20 / 3 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Esta actividad consiste en proponer varios productos al cliente si alguno de estos se ha agotado o no existen. Se le brindará además, al usuario, la alternativa de escoger otros productos disponibles.	

*Tabla 22. Tarea de implementación de la funcionalidad de ofrecer información del precio de los productos*

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 7	
<b>Nombre de Tarea:</b> Tarea para implementar la funcionalidad de ofrecer información del precio de los productos.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> Alta
<b>Fecha de Inicio:</b> 23 / 3 / 2015	<b>Fecha Fin:</b> 1 / 4 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Esta actividad consiste en responder el precio de los productos que son gestionados por el agente.	

*Tabla 23. Tarea de implementación de la funcionalidad de ofrecer información del precio total de la compra*

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 8	
<b>Nombre de Tarea:</b> Tarea para implementar la funcionalidad de ofrecer información del precio total de la compra.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> Alta
<b>Fecha de Inicio:</b> 2 / 4 / 2015	<b>Fecha Fin:</b> 11 / 4 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Esta actividad consiste en responder el precio de la compra del o los productos que son vendidos por el agente conversacional.	

Tabla 24. Tarea de diseño de la funcionalidad de ofertar descuento, por cantidad de un mismo producto comprado

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 9	
<b>Nombre de Tarea:</b> Tarea de diseño de la funcionalidad de ofertar descuento en por ciento, por cantidad de un mismo producto comprado.	
<b>Tipo de Tarea:</b> Diseño	<b>Puntos Estimados:</b> Media
<b>Fecha de Inicio:</b> 14 / 4 / 2015	<b>Fecha Fin:</b> 18 / 4 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Esta actividad se diseñará para estar relacionada con el acto de venta. En dependencia de la cantidad en libras de productos de un mismo tipo que compre el usuario (más de 20), se le hará un descuento en por ciento.	

Tabla 25. Tarea de Implementación de la funcionalidad de ofertar descuento, por cantidad de un mismo producto comprado

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 10	
<b>Nombre de Tarea:</b> Tarea de Implementación de la funcionalidad de ofertar descuento en por ciento, por cantidad de un mismo producto comprado.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> Alta
<b>Fecha de Inicio:</b> 19 / 4 / 2015	<b>Fecha Fin:</b> 2 / 5 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> El resultado del descuento se hará visible en el precio que se pondrá en la ventana de chat del metaverso al solicitar un producto.	

Tabla 26. Tarea de diseño de la funcionalidad de ofertar descuento, si el cliente compra una variedad de productos a la vez

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 11	

<b>Nombre de Tarea:</b> Tarea de diseño de la funcionalidad de ofertar descuento, si el cliente compra una variedad de productos a la vez.	
<b>Tipo de Tarea:</b> Diseño	<b>Puntos Estimados:</b> Media
<b>Fecha de Inicio:</b> 5/ 5 / 2015	<b>Fecha Fin:</b> 9 / 5 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Esta actividad se diseñará para estar relacionada con el acto de venta, al igual que la tarea 5. En dependencia de la cantidad de productos de distintos tipos que compre el usuario (más de 1), se le hará un descuento en por ciento.	

*Tabla 27. Tarea de Implementación de la funcionalidad de ofertar descuento, si el cliente compra una variedad de productos a la vez*

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 12	
<b>Nombre de Tarea:</b> Tarea de Implementación de la funcionalidad de ofertar descuento, si el cliente compra una variedad de productos a la vez.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> Alta
<b>Fecha de Inicio:</b> 10 / 5 / 2015	<b>Fecha Fin:</b> 24 / 5 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> El resultado del descuento se hará visible en el precio que se pondrá en la ventana de chat del metaverso al solicitar un producto, de igual modo que en la tarea 6.	

*Tabla 28. Tarea de diseño de las respuestas del agente conversacional*

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 13	
<b>Nombre de Tarea:</b> Tarea de diseño de las respuestas del agente conversacional.	
<b>Tipo de Tarea:</b> Diseño	<b>Puntos Estimados:</b> Media
<b>Fecha de Inicio:</b> 27/ 5 / 2015	<b>Fecha Fin:</b> 5 / 6 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> En esta actividad se diseñarán los diálogos del agente conversacional, teniendo en cuenta los temas que tratará y cómo responderá en estos, procurando la mayor cantidad de posibles respuestas dentro del contexto de las actividades de compra y venta.	

Tabla 29. Implementación del mecanismo de respuestas del agente conversacional con ayuda de la herramienta AIML

<b>Tarea de Ingeniería</b>	
<b>Número de Tarea:</b> 14	
<b>Nombre de Tarea:</b> Implementación del mecanismo para las respuestas del agente conversacional con ayuda de la herramienta AIML.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> Alta
<b>Fecha de Inicio:</b> 6 / 65 / 2015	<b>Fecha Fin:</b> 20 / 6 / 2015
<b>Programador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Las respuestas del agente conversacional se realizarán a través de patrones de entradas y posibles respuestas a estas entradas en el AIML.	

### 2.7.2 Plan de iteraciones

Para un mejor entendimiento de cada una de las etapas de desarrollo del proyecto, se plantea a continuación un plan de iteraciones por cada una de las HU definidas. Estas, idealmente, deben durar más de una a tres semanas (36).

Tabla 30. Planificación de iteraciones por HU

No. Iteración	Historias de Usuario	Tiempo Estimado (días)
1	<ol style="list-style-type: none"> <li>1. Configuración del servidor de aplicaciones 3D, OpenSim.</li> <li>2. Personalización del avatar en el metaverso.</li> <li>3. Configuración del gestor de base de datos y servidor MySQL.</li> <li>4. Implementación de la conexión del agente conversacional al OpenSim.</li> <li>5. Definición de técnicas de mercado inherentes al agente conversacional.</li> <li>6. Implementación de la funcionalidad de proponer productos y ofrecer alternativas.</li> </ol>	20

2	<p>7. Implementación de la funcionalidad de ofrecer información del precio de los productos.</p> <p>8. Implementación de la funcionalidad de informar al cliente el precio total de la compra.</p>	20
3	<p>9. Definición de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.</p> <p>10. Implementación de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.</p>	20
4	<p>11. Definición de la funcionalidad de ofertar reducción de precios si el cliente compra una variedad de productos a la vez.</p> <p>12. Implementación de la funcionalidad de ofertar reducción de precios si el cliente compra una variedad de productos a la vez.</p>	20
5	<p>13. Diseño de respuestas del agente conversacional.</p> <p>14. Implementación del mecanismo de respuestas del agente conversacional con ayuda de la herramienta AIML.</p>	25

## 2.8 Plan de entregas

En esta fase se establece la prioridad de cada historia de usuario por parte del cliente y los programadores realizan una estimación del esfuerzo necesario para realizar cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma (36).

*Tabla 31. Planificación de entregas por iteraciones del producto*

Sistema	Agente conversacional con capacidad para la oferta y venta de productos en un mundo virtual 3D
Iteración 1	10%
Iteración 2	40%
Iteración 3	60%
Iteración 4	80%

Iteración 5	100%
-------------	------

## 2.9 Fase de diseño

XP establece estructuras de diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño entendible y fácil de implementar. Esto contribuye a un menor tiempo y esfuerzo en el desarrollo (37).

Partiendo del boceto de un agente conversacional, este tendrá un diseño sencillo y similar a los seres humanos en la vida real. La diferencia más notable es que su diseño será en un mundo en 3D, a diferencia de la mayoría de los *chatbots* modernos. Un ejemplo dello se muestra en la Imagen 3.



*Imagen 3. Diseño y personalización del agente conversacional en el metaverso "Virtual English".*

### 2.9.1 Patrones de Diseño

Los desarrolladores con experiencia que trabajan la programación orientada a objetos, acumulan un repertorio - tanto de principios generales como de soluciones - basadas en aplicar ciertos estilos que les guían en la creación de software. Estos principios y estilos, codificados con un formato estructurado que describa el problema y la solución, son llamados patrones (38).

### Patrones de GRASP

Patrones de software para asignar responsabilidades, son considerados como “Buenas Prácticas” para el diseño del software. El objetivo fundamental de su utilización es caracterizar objetos y clases. Conocer a quién asignar una responsabilidad, quién es el responsable y quién puede manejarlas (39).

1. **Creador:** Este patrón asigna a la clase B la responsabilidad de crear una instancia de clase A, si se cumple uno o más de los casos siguientes:

B agrega objetos de A.

B contiene objetos de A.

B registra instancias de objetos de A.

B utiliza más estrechamente objetos de A.

B tiene los datos de inicialización que se pasarán a un objeto de A cuando sea creado.

Este patrón de diseño se pone de manifiesto en la clase *Program*, donde se crean las instancias de los objetos indispensables para el desarrollo de la aplicación.

2. **Experto en información (o Experto):** Asignar la responsabilidad al experto en información. Es la clase que contiene la información necesaria para llevar a cabo la responsabilidad.

Un ejemplo del uso del patrón Experto es la clase *Warehouse*. Esta contiene los datos necesarios sobre los productos en venta por el agente conversacional. Entre los datos que maneja están: cantidad, costo, tipo, nombre, entre otros.

### 2.9.2 Diseño de la arquitectura del software

La distribución de la aplicación está diseñada con una arquitectura por capas, permitiendo que a cada nivel se le deleguen actividades propias y, a la vez, la posibilidad de agregar nuevas modificaciones por nivel, sin necesidad de afectar otros. Las capas con las que cuenta el sistema son:

- 1. Capa de presentación:** Conocida también como capa de usuario, es la capa en la que se encuentra la interfaz gráfica y el resto del entorno virtual. En la aplicación la representa el cliente de aplicación 3D *Imprudence*.
- 2. Capa de lógica del negocio:** Es la capa donde se genera el programa y sus funcionalidades, se reciben sus solicitudes, se validan y se vierten las respuestas. Esta capa la representan las bibliotecas utilizadas, el agente conversacional y su base de conocimiento.
- 3. Capa de datos:** Es donde residen los datos y la encargada de acoger las solicitudes de datos de la capa superior. En la aplicación en desarrollo esta capa está compuesta por servidores Apache, MySQL y OpenSIM, que son necesarios a lo largo del software como base de consultoría y desarrollo. También la compone un SGBD para la edición de la base de datos del alfabeto de palabras de idioma Inglés.

La representación gráfica de la arquitectura que presenta la aplicación se muestra en la Figura 6. *Diagrama de clases del sistema*

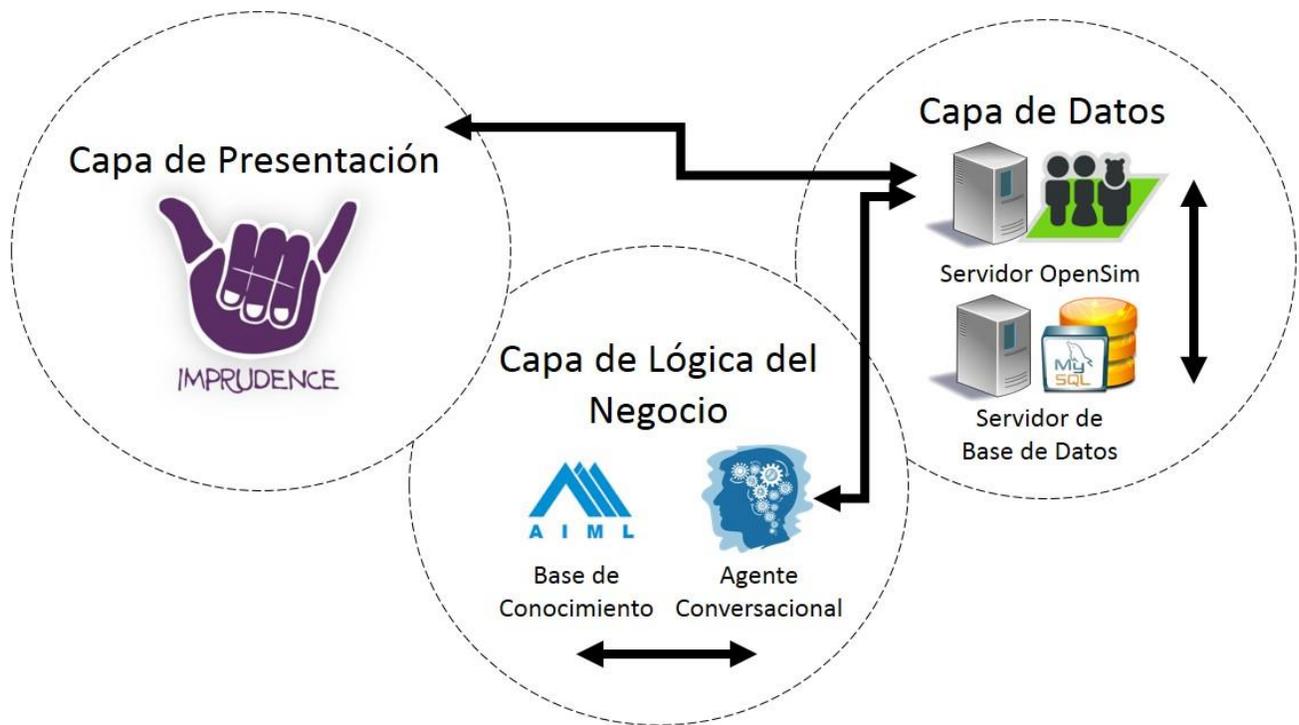


Figura 5. Arquitectura del sistema.

### 2.9.3 Diagrama de clases del sistema

Para una comprensión más acertada de la estructura del sistema en desarrollo y el conjunto de clases heredadas de otras aplicaciones con objetivos similares (epígrafe 1.3.2), se realizó el diagrama de clases mostrado en la Figura 6. *Diagrama de clases del sistema*

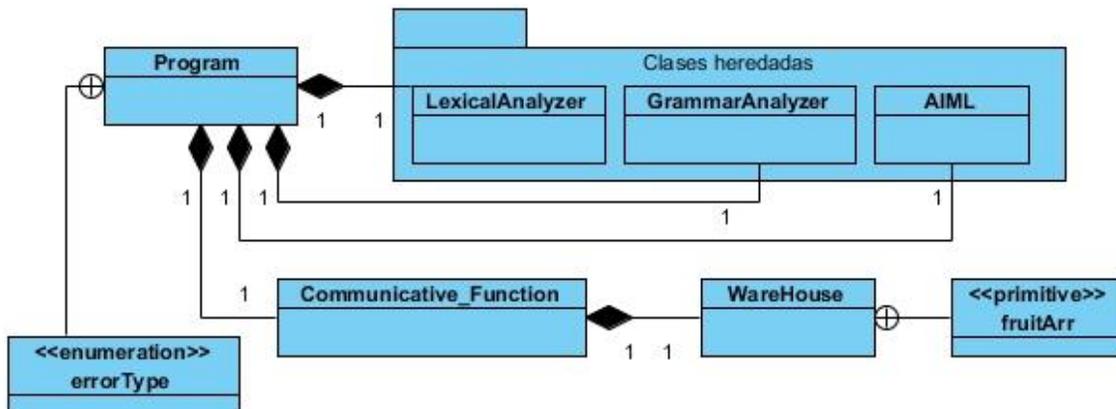


Figura 6. Diagrama de clases del sistema.

### 2.9.4 Descripción de las tarjetas CRC

El uso de las tarjetas CRC (del inglés *Class, Responsibilities and Collaboration*) permite al programador centrarse y apreciar el desarrollo orientado a objetos. La clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad (37). A continuación se ofrece una muestra de las tarjetas CRC de las clases implementadas en la aplicación.

Tabla 32. Tarjeta CRC de la clase Program

Program	
Funcionalidades:	Clases relacionadas:
Se encarga de controlar el flujo lógico del análisis lingüístico y el comportamiento del agente conversacional. Es la que recibe las señales de las actividades del agente.	<ul style="list-style-type: none"> <li>• LexicalAnalyzer</li> <li>• Grammar_Manager</li> <li>• Communicative_Function</li> <li>• AIML</li> </ul>

Tabla 33. Tarjeta CRC de la clase Communicative\_Function

Communicative_Function
------------------------

Funcionalidades:	Clases relacionadas:
Se encarga de realizar el análisis de funciones comunicativas del texto introducido por el usuario y responder preguntas o enviar mensajes correctivos en caso de encontrar error. Además, toma de Warehouse información de los productos.	<ul style="list-style-type: none"> <li>• Program</li> <li>• Warehouse</li> </ul>

Tabla 34. Tarjeta CRC de la clase Warehouse

Warehouse	
Funcionalidades:	Clases relacionadas:
Contiene la lista de productos que pueden ser vendidos por el agente conversacional. Además, contiene funcionalidades relacionadas con la gestión de estos productos.	<ul style="list-style-type: none"> <li>• Program</li> <li>• SemanticAnalyzer</li> </ul>

## 2.10 Consideraciones parciales

Mediante el desarrollo de este capítulo se concluyó que el análisis lingüístico (con el que cuenta el agente conversacional) contiene desventajas en el momento de seguir el flujo de la conversación establecida con el usuario. Debido a ello se concluyó que los textos entrados por el usuario deben tener una estructura completa, de manera que exprese claramente la intención.

## CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS

Luego de la definición del diseño y la planificación del proyecto, con el objetivo de darle una mejor organización al trabajo, es imprescindible plantear tareas de implementación por cada una de las iteraciones definidas. En este capítulo se plasma el proceso de implementación del sistema, por lo que se parte de la definición de las tareas por iteraciones. Además, se muestran las pruebas y la evaluación que se realizará en conjunto con el cliente y un grupo de expertos, para verificar si el sistema cumple o no con los objetivos definidos.

### 3.1 Fase de implementación del sistema

La implementación se refiere a la ejecución o puesta en marcha de la propuesta del proyecto. En esta etapa, al igual que en otras de la metodología XP, el usuario tiene un papel importante, pues es quién construye las historias de usuarios (40).

El sistema en desarrollo cuenta con un total de cinco iteraciones (epígrafe 2.7.2). A continuación se hace una descripción de cada una de estas iteraciones por las que transita el progreso de la aplicación.

#### 3.1.1 Iteración 1

Las tareas a cumplir para la primera fase de desarrollo, se basan en las siguientes HU:

- **HU 1** Configuración del servidor de aplicaciones 3D, OpenSim.
- **HU 2** Personalización del avatar en el metaverso.
- **HU 3** Configuración del gestor de base de datos y servidor MySQL.
- **HU 4** Implementación de la conexión del agente conversacional al servidor OpenSim
- **HU 5:** Definición de técnicas de mercado inherentes al agente conversacional.
- **HU 6:** Implementación de la funcionalidad de proponer productos y ofrecer alternativas.

Esta fase termina con la configuración del entorno de trabajo para el desarrollo del agente conversacional y la implementación de algunas funcionalidades que debe cumplir el agente, como ofrecer productos y alternativas si estos productos no se encuentran. Para cumplir con este objetivo se realizaron las siguientes tareas:

**Tarea 1** Configurar el servidor de aplicaciones 3D, OpenSim.

**Tarea 2** Personalizar el avatar en el metaverso.

**Tarea 3.**Configurar el SGBD y el servidor MySQL.

**Tarea 4:** Implementar la conexión del agente conversacional al servidor OpenSim.

**Tarea 5:** Definir las técnicas de mercado inherentes al agente conversacional.

**Tarea 6:** Implementar la funcionalidad de proponer productos y ofrecer alternativas.

*Tabla 35. Tarea 1 Iteración 1*

<b>Nombre de la tarea:</b> Configurar el servidor de aplicaciones 3D, OpenSim.	
<b>No. Tarea:</b> 1	<b>No. HU:</b> 1
<b>Tipo de Tarea:</b> Configuración	<b>Estimación:</b> 1 día
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se configurará el servidor de aplicaciones 3D, y los servidores Apache, MySQL que se utilizarán para el XoopsCube y la base de datos de palabras en inglés respectivamente.	

*Tabla 36. Tarea 2 Iteración 1*

<b>Nombre de la tarea:</b> Tarea de personalización del avatar en el metaverso.	
<b>No. Tarea:</b> 2	<b>No. HU:</b> 2
<b>Tipo de Tarea:</b> Configuración	<b>Estimación:</b> 1 día
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se hará uso del gestor de contenido XoopsCube para la creación del usuario que estará asignado al avatar en el mundo virtual. Una vez que el usuario haya accedido al mundo virtual se personalizará el agente conversacional.	

Tabla 37. Tarea 3 Iteración 1

<b>Nombre de la tarea:</b> Tarea de configuración del SGBD y servidor MySQL	
<b>No. Tarea:</b> 3	<b>No. HU:</b> 3
<b>Tipo de Tarea:</b> Configuración	<b>Estimación:</b> 1 día
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se configurará el gestor de base de datos EMS SQL Manager para MyQSL y se conectará con el servidor MySQL del metaverso OpenSim. Este SGBD trabajará sobre la base de datos que contiene las palabras del idioma inglés.	

Tabla 38. Tarea 4 Iteración 1

<b>Nombre de la tarea:</b> Tarea de implementación de la conexión del agente conversacional al servidor OpenSim	
<b>No. Tarea:</b> 4	<b>No. HU:</b> 4
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 2 días
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se implementará la conexión del agente conversacional, con el que se podrá realizar el flujo de información con el servidor de aplicación 3D, OpenSim.	

Tabla 39. Tarea 5 Iteración 1

<b>Nombre de la tarea:</b> Definición de técnicas de mercado inherentes al agente conversacional.	
<b>No. Tarea:</b> 5	<b>No. HU:</b> 5
<b>Tipo de Tarea:</b> Diseño	<b>Estimación:</b> 5 día
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se modelarán algunos mecanismos para realizar las actividades de mercado inherentes al agente conversacional. Estos son: proponer productos, informar de precios, informar costo total de compra y ofrecer alternativas si faltase algún producto.	

Tabla 40. Tarea 6 Iteración 1

<b>Nombre de la tarea:</b> Implementar la funcionalidad de proponer productos y ofrecer alternativas.	
<b>No. Tarea:</b> 6	<b>No. HU:</b> 6
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 10 días
<b>Desarrollador Responsable:</b> Nelson Manuel González Martínez	

**Descripción:** Se desarrollará la funcionalidad de proponer productos existentes en el mercado y alternativas para estos productos si alguno faltase.

### 3.1.2 Iteración 2

Las historias de usuario analizadas en la segunda fase del desarrollo son:

**HU 7:** Implementación de la funcionalidad de ofrecer información del precio de los productos.

**HU 8:** Implementación de la funcionalidad de informar al cliente el precio total de la compra.

Esta fase termina con la implementación de algunas funcionalidades que debe cumplir el avatar, como son vender u ofrecer otros productos e informar al cliente el precio de la compra y de los productos. Para cumplir con este objetivo se realizaron las siguientes tareas:

**Tarea 7:** Implementar la funcionalidad de ofrecer información del precio de los productos.

**Tarea 8:** Implementar la funcionalidad de ofrecer información del precio total de la compra.

*Tabla 41. Tarea 7 Iteración 2*

<b>Nombre de la tarea:</b> Implementar la funcionalidad de ofrecer información del precio de los productos.	
<b>No. Tarea:</b> 7	<b>No. HU:</b> 7
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 10 días
<b>Desarrollador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se desarrollará la funcionalidad de informar al cliente el precio los productos que desea comprar.	

*Tabla 42. Tarea 8 Iteración 2*

<b>Nombre de la tarea:</b> Implementar la funcionalidad de ofrecer información del precio total de la compra.	
<b>No. Tarea:</b> 8	<b>No. HU:</b> 8
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 10 días
<b>Desarrollador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se desarrollará la funcionalidad de informar al cliente el precio total de la compra y la vía mediante la cual se puede realizar.	

### 3.1.3 Iteración 3

Para el desarrollo de las tareas de la tercera iteración se analizaron las siguientes historias de usuarios:

**HU 9:** Definición de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.

**HU 10:** Implementación de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.

Esta fase termina con la implementación de algunas funcionalidades de oferta que debe cumplir el avatar, como el descuento en dependencia de la cantidad de productos de un mismo tipo. Para cumplir con este objetivo se realizaron las siguientes tareas:

**Tarea 9:** Diseñar el mecanismo para ofertar descuento, por cantidad de un mismo producto comprado.

**Tares 10:** Implementar el mecanismo para ofertar descuento, por cantidad de un mismo producto comprado.

*Tabla 43. Tarea 9 Iteración 3*

<b>Nombre de la tarea:</b> Diseño del mecanismo para ofertar descuento en por ciento, en función de la cantidad de frutas a comprar de un mismo tipo.	
<b>No. Tarea:</b> 9	<b>No. HU:</b> 9
<b>Tipo de Tarea:</b> Diseño	<b>Estimación:</b> 5 días
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se modelará el mecanismo para ofertar descuento en por ciento, en función de la cantidad de frutas a comprar de un mismo tipo. Un aspecto a tener en cuenta sobre este descuento es que la reducción empezará a partir de 20 productos comprados.	

*Tabla 44. Tarea 10 Iteración 3*

<b>Nombre de la tarea:</b> Implementación del mecanismo para ofertar descuento, en por ciento, en función de la cantidad de frutas a comprar de un mismo tipo.	
<b>No. Tarea:</b> 10	<b>No. HU:</b> 10
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 15 días
<b>Desarrollador Responsable:</b> Nelson Manuel González Martínez	

**Descripción:** Se implementará el mecanismo para ofertar descuento en por ciento, en función de la cantidad de frutas a comprar de un mismo tipo.

### 3.1.4 Iteración 4

Las historias de usuario analizadas en la cuarta fase del desarrollo fueron:

**HU 11:** Diseño de la funcionalidad de ofertar descuento si el cliente compra una variedad de productos a la vez.

**HU 12:** Implementación de la funcionalidad de ofertar descuento si el cliente compra una variedad de productos a la vez.

Esta fase termina con la implementación de algunas funcionalidades que debe cumplir el avatar, tales como el descuento en dependencia de la cantidad de productos de diferentes tipos. Para cumplir con este objetivo se realizaron las siguientes tareas:

**Tarea 11:** Diseño del mecanismo para ofertar descuento si el cliente compra una variedad de productos a la vez.

**Tares 12:** Implementación del mecanismo para ofertar descuento si el cliente compra una variedad de productos a la vez.

*Tabla 45. Tarea 11 Iteración 4*

<b>Nombre de la tarea:</b> Diseño del mecanismo para ofertar descuento si el cliente compra una variedad de productos a la vez.	
<b>No. Tarea:</b> 11	<b>No. HU:</b> 11
<b>Tipo de Tarea:</b> Diseño	<b>Estimación:</b> 5 días
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se modelará el mecanismo para ofertar reducción de precios si el cliente compra una variedad de productos a la vez. Esta reducción se notará a partir de que el cliente compre más de 2 productos.	

Tabla 46. Tarea 12 Iteración 4

<b>Nombre de la tarea:</b> Implementación del mecanismo para ofertar descuento si el cliente compra una variedad de productos a la vez.	
<b>No. Tarea:</b> 12	<b>No. HU:</b> 12
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 15 días
<b>Desarrollador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Se desarrollará el mecanismo para reducción de precios si el cliente compra una variedad de productos a la vez.	

### 3.1.5 Iteración 5

Las tareas de desarrollo para la última iteración se apoyaron en las siguientes historias de usuario:

**HU 13:** Diseño de respuestas del agente conversacional.

**HU 14:** Implementación del mecanismo de respuestas del agente conversacional a través de la herramienta AIML.

Esta fase termina con la implementación del resto de la funcionalidad de realizar una conversación fluida con el apoyo de la herramienta AIML. Para cumplir con este objetivo se realizaron las siguientes tareas:

**Tarea 13:** Diseñar respuestas del agente conversacional.

**Tarea 14:** Implementar el mecanismo para las respuestas del agente conversacional con ayuda de la herramienta AIML.

Tabla 47. Tarea 13 Iteración 5

<b>Nombre de la tarea:</b> Diseñar respuestas del agente conversacional.	
<b>No. Tarea:</b> 13	<b>No. HU:</b> 13
<b>Tipo de Tarea:</b> Diseño	<b>Estimación:</b> 10 días
<b>Diseñador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> En esta actividad se diseñarán los diálogos del agente conversacional, teniendo en cuenta los temas que tratará y cómo responderá en estos, procurando la mayor cantidad de posibles respuestas dentro del contexto de las actividades de venta.	

Tabla 48. Tarea 14 Iteración 5

<b>Nombre de la tarea:</b> Implementar el mecanismo para las respuestas del agente conversacional con ayuda de la herramienta AIML.	
<b>No. Tarea:</b> 2	<b>No. HU:</b> 14
<b>Tipo de Tarea:</b> Desarrollo	<b>Estimación:</b> 10 días
<b>Desarrollador Responsable:</b> Nelson Manuel González Martínez	
<b>Descripción:</b> Implementar el mecanismo para las respuestas del agente conversacional con ayuda de la herramienta AIML	

### 3.2 Fase de Pruebas del Sistema

La fase de pruebas al sistema es donde se hacen pruebas del código a diferentes niveles, tanto sus módulos como la integración de estos, por parte del equipo de desarrollo. Para esto se preparan pruebas de aceptación en donde se asegura que todo lo solicitado por el cliente cumple con los requisitos necesarios y no hace algo que provoque inquietudes (36).

#### 3.2.1 Pruebas de Aceptación

Son conocidas como pruebas de caja negra, ya que es el propio cliente quien la realiza en compañía de uno de los representantes del equipo de desarrollo y se orientan a las funcionalidades del sistema. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de las especificaciones de la lista de reservas del producto (36). A continuación, se muestran las pruebas de aceptación realizadas a la solución propuesta.

Tabla 49. Caso de prueba HU6\_P1

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU6_P1	<b>Historia de Usuario:</b> HU6
<b>Nombre:</b> Implementación de la funcionalidad de proponer productos y ofrecer alternativas.	
<b>Descripción:</b> El agente conversacional le propondrá una serie de productos, si el usuario le pregunta. Si no existe, o se ha agotado algún producto requerido por el usuario, el agente propondrá otras alternativa.	
<b>Condiciones de Ejecución:</b> El usuario entra un texto en inglés preguntando por uno o una variedad de productos que el agente conversacional pueda tener.	

<p><b>Pasos de Ejecución:</b></p> <ul style="list-style-type: none"> <li>- Escribir un texto en inglés, por la ventana del chat general. Una vez que el bot envíe un mensaje para establecer la conversación por el chat privado, escribir nuevamente el texto en la ventana de chat privado y presionar la tecla “Enter” o click en el botón “Enviar”.</li> <li>- Acceder a la ventana chat privado del bot dando “click derecho” sobre este y luego “click izquierdo” sobre “Enviar IM”, luego escribir un texto en inglés y presionar la tecla “Enter” o click en el botón “Enviar”.</li> </ul>
<p><b>Resultados Esperados:</b> La respuesta dependerá de la cantidad de productos que el usuario solicite y de la cantidad que exista de estos productos. Estas respuestas pueden ser:</p> <ol style="list-style-type: none"> <li>1. Sugerencia de otros productos en caso de tener agotados uno o más productos por los que pregunta el cliente.</li> </ol>

Tabla 50. Caso de prueba HU7\_P2

Caso de Prueba de Aceptación	
<b>Código:</b> HU7_P2	<b>Historia de Usuario:</b> HU7
<b>Nombre:</b> Implementación de la funcionalidad de ofrecer información del precio de los productos	
<b>Descripción:</b> Responderá el precio de los productos que el usuario gestione.	
<b>Condiciones de Ejecución:</b> Se ejecutará cuando el usuario pregunte el precio de algún producto.	
<p><b>Pasos de Ejecución:</b></p> <ul style="list-style-type: none"> <li>- Escribir un texto en inglés, por la ventana del chat general. Una vez que el bot envíe un mensaje para establecer la conversación por el chat privado, escribir nuevamente el texto en la ventana de chat privado y presionar la tecla “Enter” o click en el botón “Enviar”.</li> </ul> <p>Acceder a la ventana chat privado del bot dando “click derecho” sobre este y luego “click izquierdo” sobre “Enviar IM”, luego escribir un texto en inglés y presionar la tecla “Enter” o click en el botón “Enviar”.</p>	
<p><b>Resultados Esperados:</b> La respuesta será el precio de los productos en cuestión y una propuesta de servicio de rebaja de precios.</p> <p>Si el usuario además, pregunta el precio de un producto, el producto en cuestión será revisado por el caso de prueba HU7_P1.</p>	

Tabla 51. Caso de prueba HU8\_P3

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

<b>Código:</b> HU8_P3	<b>Historia de Usuario:</b> HU8
<b>Nombre:</b> Implementación de la funcionalidad de informar al cliente el precio total de la compra.	
<b>Descripción:</b> El agente conversacional responderá el precio total de una compra realizada por el usuario.	
<b>Condiciones de Ejecución:</b> Se ejecutará cuando el usuario entre un texto en inglés para realizar una compra de uno o varios de productos.	
<b>Pasos de Ejecución:</b> <ul style="list-style-type: none"> <li>- Escribir un texto en inglés, por la ventana del chat general. Una vez que el bot envíe un mensaje para establecer la conversación por el chat privado, escribir nuevamente el texto en la ventana de chat privado y presionar la tecla “Enter” o click en el botón “Enviar”.</li> <li>- Acceder a la ventana chat privado del bot dando “click derecho” sobre este y luego “click izquierdo” sobre “Enviar IM”, luego escribir un texto en inglés y presionar la tecla “Enter” o click en el botón “Enviar”.</li> </ul>	
<b>Resultados Esperados:</b> La respuesta será el precio de la compra y un comentario sobre cómo hacer el pago.	

Tabla 52. Caso de prueba HU10\_P4

Caso de Prueba de Aceptación	
<b>Código:</b> HU10_P4	<b>Historia de Usuario:</b> HU10
<b>Nombre:</b> Implementación de la funcionalidad de ofertar descuento por cantidad de un mismo producto comprado.	
<b>Descripción:</b> Se ofertará un descuento por cantidad de un mismo producto comprado.	
<b>Condiciones de Ejecución:</b> Cada vez que el usuario compre o pregunte por el costo de 20 o más libras de uno o varios productos.	
<b>Pasos de Ejecución:</b> <ul style="list-style-type: none"> <li>- Escribir un texto en inglés, por la ventana del chat general. Una vez que el bot envíe un mensaje para establecer la conversación por el chat privado, escribir nuevamente el texto en la ventana de chat privado y presionar la tecla “Enter” o click en el botón “Enviar”.</li> </ul>	

- Acceder a la ventana chat privado del bot dando “click derecho” sobre este y luego “click izquierdo” sobre “Enviar IM”, luego escribir un texto en inglés y presionar la tecla “Enter” o click en el botón “Enviar”.

**Resultados Esperados:** El agente conversacional responderá el precio de los productos gestionados, teniendo en cuenta que a este se le resta un por ciento del precio total de los producto por el que se pida una cantidad mayor a 20.

Tabla 53. Caso de prueba HU12\_P5

Caso de Prueba de Aceptación	
<b>Código:</b> HU12_P5	<b>Historia de Usuario:</b> HU12
<b>Nombre:</b> Implementación de la funcionalidad de ofertar reducción de precios si el cliente compra una variedad de productos a la vez	
<b>Descripción:</b> Se realizará una oferta de reducción de precios si el cliente compra una variedad de productos a la vez	
<b>Condiciones de Ejecución:</b> El usuario compra varios productos de diferentes tipos a la vez.	
<b>Pasos de Ejecución:</b>	
<ul style="list-style-type: none"> <li>- Escribir un texto en inglés por la ventana del chat general. Una vez que el bot envíe un mensaje para establecer la conversación por el chat privado. Escribir nuevamente el texto en la ventana de chat privado y presionar la tecla “Enter”, o click en el botón “Enviar”.</li> <li>- Acceder a la ventana chat privado del bot dando “click derecho” sobre este y luego “click izquierdo” sobre “Enviar IM”, luego escribir un texto en inglés y presionar la tecla “Enter” o click en el botón “Enviar”.</li> </ul>	
<b>Resultados Esperados:</b> Cada vez que el usuario compre más de una variedad de productos se reducirá el precio de esta compra en un por ciento por producto distinto comprado.	

Tabla 54. Caso de prueba HU14\_P6

Caso de Prueba de Aceptación	
<b>Código:</b> HU14_P6	<b>Historia de Usuario:</b> HU14
<b>Nombre:</b> Implementación del mecanismo para las respuestas del agente conversacional con ayuda de la herramienta AIML.	

<b>Descripción:</b> Se crearán un grupo de respuestas que el agente conversacional brinda en el proceso de interacción a través de la herramienta AIML.
<b>Condiciones de Ejecución:</b> El usuario entra un texto en inglés
<b>Pasos de Ejecución:</b> <ul style="list-style-type: none"> <li>- Escribir un texto en inglés por la ventana del chat general. Una vez que el bot envíe un mensaje para establecer la conversación por el chat privado. Escribir nuevamente el texto en la ventana de chat privado y presionar la tecla “Enter”, o click en el botón “Enviar”.</li> <li>- Acceder a la ventana chat privado del bot dando “click derecho” sobre este y luego “click izquierdo” sobre “Enviar IM”, luego escribir un texto en inglés y presionar la tecla “Enter” o click en el botón “Enviar”.</li> </ul>
<b>Resultados Esperados:</b> Si el agente conversacional tiene respuesta para este mensaje, responderá de acuerdo a la pregunta entrada y el patrón de respuesta definido. Hay que tener en cuenta que si el texto entrado por un usuario tiene algún error léxico o gramatical, el texto no será respondido por el AIML sino por el analizador lingüístico que encuentre el error. Al igual que si contiene alguna estructura referente a actividades de compra y venta, será respondido por el analizador correspondiente.

Las pruebas de aceptación realizadas, arrojaron algunas no conformidades en el texto de salida resultante de la interacción con el agente conversacional. Estas no conformidades fueron solucionadas en las iteraciones correspondientes. En estas iteraciones, además de la detección y corrección de las no conformidades, se procuró velar por el correcto funcionamiento de las HU implementadas y probadas en iteraciones anteriores por las pruebas de aceptación. Al final de la última iteración se corrigieron todas las no conformidades detectadas en el agente conversacional (Ver Figura 7).

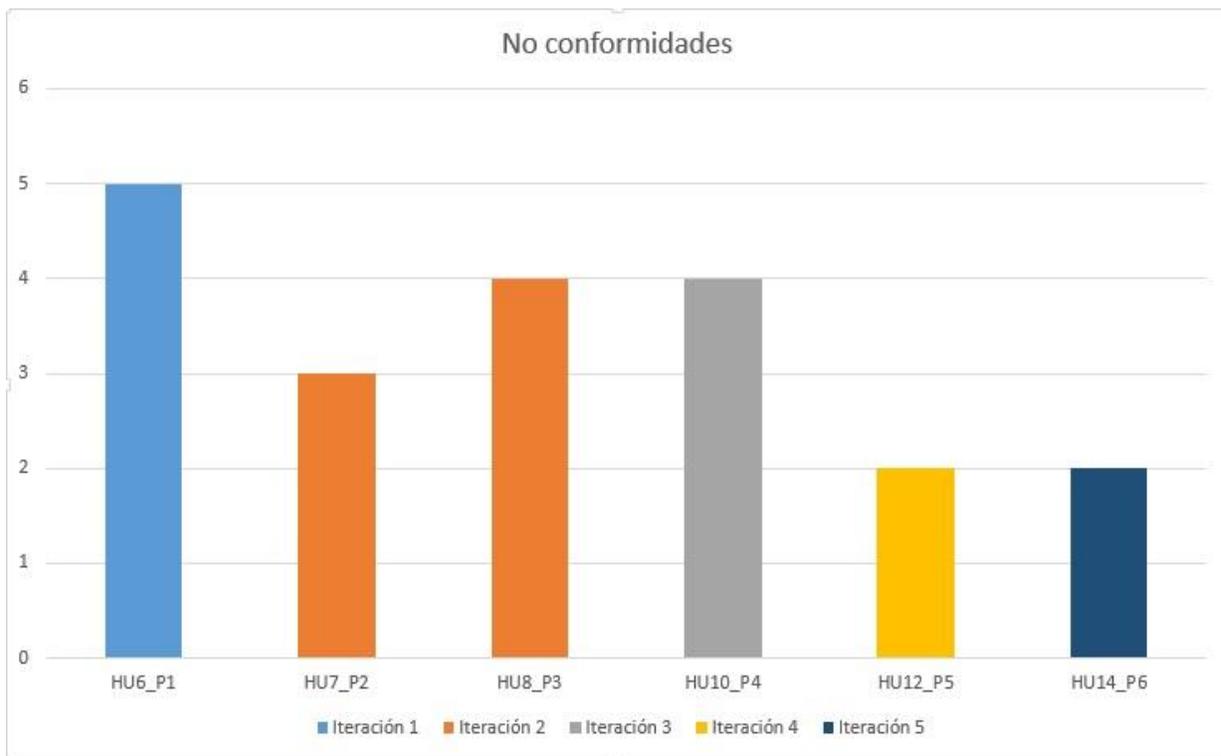


Figura 7. Iteraciones de las pruebas de aceptación.

### 3.3 Consideraciones parciales

En el presente trabajo, las reglas a partir de las que se rige el análisis del texto (en el analizador de funciones comunicativas) contienen algunas funcionalidades en común, resultando que algunas HU se relacionen entre sí para explicar estas funcionalidades. Por consiguiente, se concluyó que - en las pruebas de aceptación - esta relación también es tomada en cuenta tal y como se realiza en las HU.

### CONCLUSIONES

Con el desarrollo del presente trabajo se arribó a las siguientes conclusiones:

Los agentes conversacionales pueden tener características en común con un agente inteligente como son pro-actividad, personalidad, veracidad y sociabilidad. A pesar de no ser un agente inteligente el creado en este trabajo, podría serlo con un desarrollo dirigido en este sentido.

Dado que la naturaleza de este agente conversacional es realizar actividades sociales en cuanto a la venta de productos en el entorno "*Virtual English*", debido a las pruebas de aceptación realizadas, es el adecuado para ser destinado a la práctica de la comunicación de los usuarios en el área de la venta de productos.

A pesar de la detección de algunos errores en las estructuras gramaticales de las oraciones entradas por el usuario (referentes a las actividades de compra) hay que tener en cuenta que esta corrección no se ha podido realizar con el analizador gramatical "*GrammarAnalyzer*", debido a algunas faltas de este analizador. Este analizador puede ser fuente de investigación y desarrollo para futuros trabajos.

## BIBLIOGRAFÍA

1. *Diseño instruccional para la producción de cursos en línea y e-learning*. Alvarado, A. 2003, Vol. 3, pág. 9.
2. *Competencias Comunicativas*. Hymes, Dell. s.l. : Pride and Holmes, 1972.
3. Gross, B. *Diseño y programas educativos. Pautas pedagógicas para le elaboración de software*. Barcelona : Ariel Education, 1997.
4. Sanz, Cecilia, Zangara, Alejandra y Escobar G, Margda Lorena. *Posibilidades Educativas de Second Life*. Instituto de Investigación en Informática & Facultad de Informática Universidad de La Plata. La Plata: s.n.
5. Mundos Virtuales. [En línea] <http://www.mundosvirtuales.net>.
6. Morelo, Doris. URBE Mundos Virtuales y Educación. [En línea] 29 de 12 de 2012. <http://mundosvirtualesyeducacion.blogspot.com/2011/12/mundos-virtuales-OpenSimulator-open-sim.html>.
7. Second Life Data. [En línea] 7 de marzo de 2013. <https://secondlifedata.wordpress.com/tag/como-empezar-en-second-life/>.
8. Ayuda para Second Life. [En línea] <http://ayuda-secondlife.blogspot.com/imprudence-viewer.html>.
9. Moret, B. *Fundamentos de la inteligencia Artificial. 2da Edición*. 2004.
10. Choua, C, Chanb, T y Linc, C. *Redefining the learning companion: the past, present, future on educational agents*. 2002. págs. 255-265.
11. Angel. borrowbits.com. [En línea] 6 de 2 de 20132. <https://borrowbits.com/2013/02/agentes-inteligentes-en-dispositivos-moviles/>.
12. Torres, Juan Carlos Cobos. *Integración de un chatbot como habilidad de un robot social con gestor de diálogos*. Universidad de Carlos III. Madrid: s.n.
13. Dorfman, Mauricio, y otros. *Asistentes Virtuales de Clase como complemento a la educación universitaria oresencial*. Buenos Aires: s.n., 2011.

14. Rodríguez , Juan, Merlino, Hernán y Fernández, Enrique. *Comportamiento Adaptable de Chatbots Dependiente del Contexto*. Facultad de Ingeniería, Grupo de Investigación en Sistemas de Información (SIGI), Universidad Nacional de Lanús. Argentina: s.n., 2014.
15. Ibarra Monteagudo, Yeili y Quintanal Reytor, Rafael Arístides. *Bots conversacionales con capacidad para corregir errores ortográficos y gramaticales, en función de alcanzar un nivel A3 del idioma inglés en el metaverso "Virtual English"*. Universidad de las Ciencias Informáticas. La Habana: s.n.
16. Arma, José Carlos Enrique de. *Agente inteligente para elevar el grado de Inmersión al orientar a los usuarios dentro del entorno "Virtual English" de la asignatura Inglés Básico*. Universidad de las Ciencias Informáticas. La Habana: s.n., 2014.
17. Gómez, Javier Guinovart. *Fundamentos de la Lingüística Computacional: base teórica, líneas de investigación y aplicaciones*.
18. Ceccaroni, Luigi. *Arquitectura de los sistemas basados en el conocimiento*. Departamento de Ciencias de la Computación, Universidad Politécnica de Cataluña. 2009.
19. Lozano, Laura y Fernández, Javier. *Razonamiento basado en casos. Una visión general*. 2008.
20. A.L.I.C.E Artificial Intelligence Funtation. [En línea] 12 de 2 de 2014. <http://www.alicebot.org/aiml.html>.
21. Apache HTTP Server Project. *The Apache Software Foundation*. [En línea] [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html).
22. *Manual de Referencia de MySQL Capítulo 1*.
23. [Apachefriends.org](http://www.apachefriends.org/es/index.html). [En línea] <http://www.apachefriends.org/es/index.html>.
24. *Sistemas de Gestión de Bases de datos y SIG, Capítulo 9*. Universidad de Murcia.
25. SQL Manager.net. *EMS Datamanager Solution*. [En línea] <http://www.sqlmanager.net/en/products/mysql/manager/download/5/2177>.

26. fergarciac. [En línea] 2013. <http://fergarciaac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide>
27. Visual Studio.com. [En línea] <https://www.visualstudio.com/es-es>.
28. C con Clase. [En línea] <http://www.c.conclase.net/curso/index.php>.
29. Gomolan, Adamski. Ayuda par Second Life. [En línea] 2 de 12 de 2010. <http://ayuda-secondlife.blogspot.com/imprudence-viewer.html>.
30. SharpNLP-open source natural language processing tools. [En línea] 15 de 2 de 2014. <http://sharpnlp.codeplex.com>.
31. Apache OpenNLP Development. [En línea] 1 de 6 de 2014. <http://opennlp.apache.org/documentation/manual/opennlp.html>.
32. Canós, Joseph, Letelier, Patricio y Penadés M, Carmen. *Metodologías Ágiles en el desarrollo de Software*. Universidad Politécnica de Valencia. Valencia, España: s.n., 2003.
33. Wooldridge, M y Cincarani, P. *Agent-Oriented Software Engineering: The State of the Art*. Limerick: First International Workshop, 2000. págs. 1-28. Vol. 1957.
34. Cares, Carlos. *Ingeniería de Software Orientada a Agentes: Un ejemplo usando Tropos*. Departamento de Ingeniería de Sistemas y Departament de Llenguatges i Sistemes Informatics. Temuco, Chle: s.n., 2007. págs. 17-18.
35. Aspiazu, Guillermo Choque. Mente Errabunda. [En línea] 28 de 6 de 2008. <http://menteerrabunda.blogspot.com/2008/06/ingeniera-del-software-basada-en.html>.
36. A Sánchez, Emilio, Letelier, Patricio y H Canós, José. *Mejorando la gestión de historias de usuario en eXtreme Programming*. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. Valencia, España: s.n.
37. *Guía Práctica de Gestión de Requisitos*. Laboratorio Nacional de Calidad de Software, INTECO. 2008.

38. Calabria, Luis y Piriz, Pablo. *Metodología XP*. Facultad de informática, Universidad ORT Uruguay.
39. Echeverry Torbón, Luis Miguel y Delgado Carmona, Luz Elena. *CasosPrácticos de la metodología ágil XP al desarrollo de software*. Universidad Tecnológica de Pereira. 2007.
40. Ordeñez, Meylin y Pérez Rodríguez, Maité Corbea. *La metodología XP aplicada al Desarrollo de Software Educativo en Cuba*. 2007.
41. Larman, Craig. *UML y Patrones, Una introducción al análisis y diseño orientado a objetos y al proceso unificado*.
42. Garzás, J. JavierGarzas.com. [En línea] <http://www.javiergarzas.com/2014/08/los-patrones-grasp.html>.
43. Silveiro, Edgar Espinoza. *Maestría en Ingeniería de Sistemas con Mención en Gestión en Tecnología de la Información, Programación Extrema*. Lima, Perú: s.n., 2007.
44. Fernández Valmayor, A, y otros. *Lenguajes de programación, lenguajes de marcado y modelos hipermedia: una visión interesada de la evolución de los lenguajes informáticos*. Madrid: s.n.
45. Márquez, Eric Mayas. *Estudio y mejora de la usabilidad de un mundo virtual híbrido: iteración humano-agente*. Facultad de Matemática, Universidad de Barcelona. Barcelona : s.n., 2013. pág. 39.
46. G A, Miller, y otros. *WordNet an online lexical databases*. 1990. págs. 235-244.
47. Centro de Investigación Sociológicos (CIS). [En línea] <http://www.cis.es/>.
48. Cervantes.es. *Centro Virtual Cervantes*. [En línea] [http://cvc.cervantes.es/ensenanza/biblioteca\\_ele/diccio\\_ele/diccionario/referenciaanaforica.htm](http://cvc.cervantes.es/ensenanza/biblioteca_ele/diccio_ele/diccionario/referenciaanaforica.htm).

## ANEXO 1

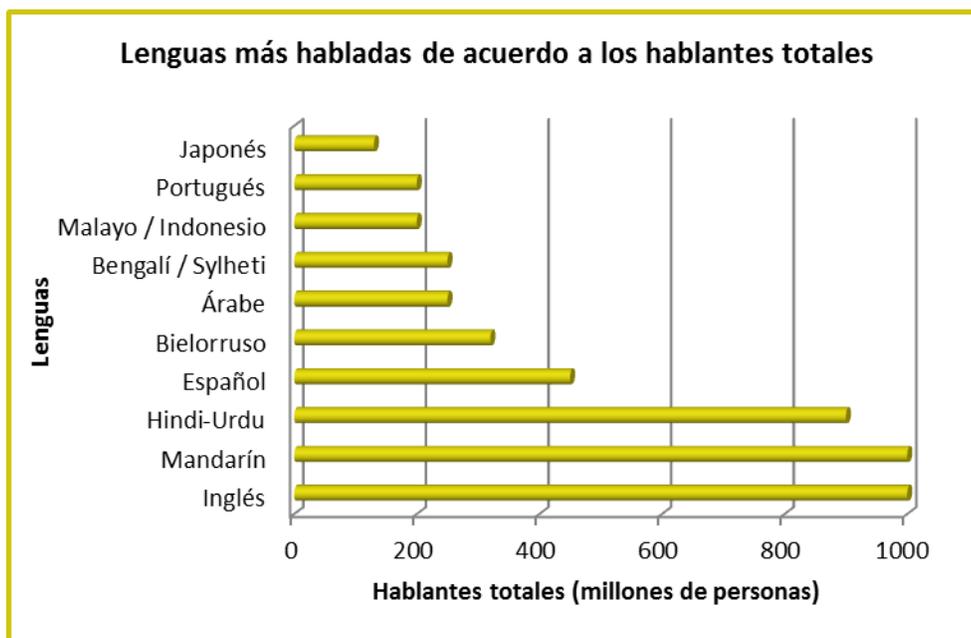


Figura 8. Los Idiomas más hablados según. Babilon Language Coaching. 6-9-2013<sup>16</sup>.

<sup>16</sup> Tomado de [http://blcmexico.blogspot.com/2013\\_09\\_01\\_archive.html](http://blcmexico.blogspot.com/2013_09_01_archive.html)

## GLOSARIO

**Docente:** *adjetivo/nombre común* [persona] Que se dedica a la enseñanza.

**Discente:** Formal [persona] Que recibe enseñanza.

**Intrínseca:** *adjetivo*. Que es propio o característico de la cosa que se expresa por sí misma y no depende de las circunstancias.

**Conductismo:** *nombre masculino*. Corriente de la psicología que se basa en la observación del comportamiento o conducta del ser que se estudia y que explica el mismo como un conjunto de relaciones entre estímulos y respuestas.

**Cognitivismo:** *nombre masculino*. Corriente de la psicología que explica el trabajo de la mente humana mediante un modelo hipotético de su funcionamiento.

**Anáfora:** Referencia a un término o a una parte anterior del discurso. "los pronombres, los adverbios y los verbos son frecuentemente anáforas que evitan la repetición de las palabras ya dichas en un discurso".

**Anáfora correferencial** o de referencia: los elementos relacionados tienen el mismo referente, es decir, remiten al mismo ente (animal, cosa o persona) de la realidad extralingüística; [He hablado con *María*, pero no *le* he dicho que venga]; [*María*] y [*le*] son elementos correferenciales.

**Anáfora elíptica:** un elemento presente y un elemento ausente en el discurso establecen una relación anafórica a causa de una identidad de referencia o de sentido; [Tus amigos han venido. \_\_ Me han dicho que \_\_ volverán pronto].

**Anáfora de sentido:** los elementos relacionados presentan el mismo sentido, aunque no son correferenciales; [Coge tú este *libro*; yo me llevaré *otro*.]; los elementos relacionados, aunque pertenecen a la misma clase, tienen distintos referentes.