

Universidad de Oriente

Sede “Julio Antonio Mella”

Facultad de Ingeniería en Telecomunicaciones, Informática y Biomédica

Trabajo de Diploma

En opción al título de Ingeniero en Informática

**Título:** “Herramienta Digital para la construcción de conocimiento para un Asistente Virtual”.

**Autor:** Jorge Ernesto Dubalón Hernández.

**Tutores:** Dr. Dionis López Ramos.

Ing. Yillian Herrera Rodríguez

Santiago de Cuba, 2022

“Año 64 de la Revolución”

**Resumen**

La población requiere respuestas inmediatas y acciones en tiempo real de diferentes servicios importantes (Ej.: Salud, Gobierno, Seguridad, entre otros). En esos momentos críticos, los canales tradicionales no pueden satisfacer la demanda pico y la búsqueda de información. Para resolver esta necesidad han sido creados los asistentes virtuales (Agentes Conversacionales) o chatbots. Programas que intenta imitar la conversación que puede proveer un ser humano. Estas son herramientas digitales que permiten la interacción hombre máquina. Son ampliamente utilizados en el sector empresarial, salud y gobierno porque garantizan una atención al usuario las 24 horas.

A pesar de los grandes beneficios que proporcionan los asistentes virtuales, la creación del conocimiento que usan para dar respuestas a las preguntas y la interacción con los usuarios es laboriosa y costosa. Esto debido a la necesidad de reunir especialista del tema objetivo del asistente virtual y reunir la información necesaria para estos asistentes virtuales. Este laborioso proceso puede detener la creación de los asistentes virtuales.

En esta investigación se propone el diseño e implementación de una herramienta para la creación, entrenamiento y despliegue de asistentes virtuales, reduciendo la necesidad de la interacción con especialistas. Para la creación de esta herramienta y el despliegue de los asistentes virtuales se emplea el lenguaje de programación Python y el marco de trabajo Rasa especializado en la creación de asistentes virtuales.

**Abstract**

Title: “Digital tool for the construction of knowledge for a Virtual Assistant”.

Índice

[**Introducción** 4](#_Toc96771073)

[**Capítulo 1. Marco Referencial** 6](#_Toc96771074)

[**1.1 Los marcos o frameworks para desarrollar chatbot** 6](#_Toc96771075)

[**1.2 Estado del arte de los frameworks para desarrollar chatbot** 6](#_Toc96771076)

[**Azure Bot Service** 6](#_Toc96771077)

[**DialogFlow** 7](#_Toc96771078)

[**Rasa** 8](#_Toc96771079)

[**BotPress** 10](#_Toc96771080)

[**1.3 Herramientas, Lenguajes de programación y Tecnologías** 12](#_Toc96771081)

[**Python** 12](#_Toc96771082)

[**PyCharComunity** 12](#_Toc96771083)

[**RASA** 12](#_Toc96771084)

[**MongoDB** 13](#_Toc96771085)

[**YAML** 13](#_Toc96771086)

[**GIT** 13](#_Toc96771087)

[**1.4 Metodología de desarrollo de software** 14](#_Toc96771088)

[**Capítulo 2. Planificación y Diseño** 16](#_Toc96771089)

[**2.1 Propuesta del Sistema** 16](#_Toc96771090)

[**2.2 Usuarios del Sistema** 17](#_Toc96771091)

[**2.3 Planificación del desarrollo del Sistema** 17](#_Toc96771092)

[**2.3.1. Historias de usuarios** 17](#_Toc96771093)

[**2.3.2. Historias técnicas** 21](#_Toc96771094)

[**2.3.3. Diseño de Base de Batos** 22](#_Toc96771095)

[**Capítulo 3. Implementación y Prueba** 24](#_Toc96771096)

# **Introducción**

Un Asistente virtual (AV) es un programa informático que permite a los seres humanos interactuar con la tecnología utilizando una variedad de métodos de entrada (voz, texto, gestos, tacto, etc.) y que suele estar disponible las 24 horas, los 7 días a la semana y los 365 días del año.

Durante muchos años, los chabots se utilizaron sólo en entornos de servicio al cliente, pero ahora se han añadido otros casos de uso principalmente dentro de las empresas para mejorar la experiencia del cliente y la eficiencia empresarial. Los chatbots son cada vez más populares y se conocen por una variedad de nombres diferentes: *bot conversacional de inteligencia artificial (IA), asistente virtual inteligente, asistente virtual para clientes, asistente digital*, entre otros. [16]

Así como los chatbots se conocen por una variedad de nombres diferentes, también suelen tener diferentes grados de inteligencia. Un chatbot básico es apenas un poco más avanzado que una solución de front-end[[1]](#footnote-1) para responder a preguntas frecuentes (FAQs).

Luego están los chatbots construidos en algunos de los frameworks de bot disponibles en el mercado. Estos pueden ofrecer características más avanzadas como el de recogida de datos u otras capacidades transaccionales simples, como por ejemplo tomar un pedido para una pizza. [16]

Sin embargo, sólo los chatbots con IA tienen la inteligencia y capacidad de ofrecer una experiencia conversacional sofisticada que la mayoría desea incorporar a su trabajo.

Los asistentes virtuales han venido cambiando el rostro de la asistencia virtual a nivel mundial, detonando en el paso un dinamismo que hoy se traduce en mayor competitividad, ahorro de tiempo y recursos, así como un mejor posicionamiento para los que usan esta tecnología en el mundo digital. [3]

Una atención al cliente pronta, específica y eficaz, servicio de venta, asistencia y acompañamiento puntual, así como ser una valiosa fuente de datos e información que nos pueden dar un panorama vital del negocio son otras de las ventajas que brinda el uso de los asistente virtuales. Además, también pueden identificar de manera eficiente el comportamiento y las necesidades de los clientes a partir de un conocimiento previamente dado e incluso llegar a entrenarse a partir de las conversaciones, pueden recopilar datos importantes para una mejor experiencia con el usuario. [3]

Con estos datos, los sistemas cognitivos que utilizan Inteligencia Artificial son capaces de depurar, ordenar, analizar y hacer referencias cruzadas de información, poniéndola disponible para diferentes usos: optimización de sitios web corporativos y redes sociales; redacción de mensajes, personalizados o destinados a un sector específico; llegando a mejorar el propio chatbot. [3]

En nuestro país actualmente esta tecnología no es muy explotada aunque han existido como es el ejemplo de Amanda (Usada para información sobre las elecciones en Cuba) [4]. Las personas necesitan satisfacer dudas sobre cualquier tema constantemente y las entidades han buscado formas para ello con el uso de personal especializado que debe estar disponibles las 24h. Muchas veces este personal no puede atenderlos por horario o problemas con el servicio, y este servicio genera altos costos. Las personas muchas veces tienen que moverse grandes distancias para acceder a la información que necesitan.

Los chatbots pueden resolver las necesidades antes expuestas, dar las ventajas que se han mencionado mejorando así las formas de trabajo con el cliente; pero necesitan el conocimiento necesario para ello.

**Problema de la investigación:**

Para la construcción del conocimiento que usa un asistente virtual se necesitan especialistas que puedan llevar el conocimiento especializado al formato que pueda comprender un asistente virtual.

Ese proceso de construcción es costoso, puede durar en el tiempo y contener el error humano.

**Objeto de Estudio:** Gestión del conocimiento de un Asistente Virtual.

**Campo de Estudio:** Aplicación Informática para la creación automática de conocimiento para un Asistente Virtual.

**Objetivo General:** Diseñar e implementar una aplicación informática que permita la creación del conocimiento, de manera automática, para un Asistente Virtual.

**Objetivo Específicos**

* Estudio del estado del arte de las herramientas para la creación de asistentes virtuales.
* Diseñar un prototipo de herramienta para la creación de conocimiento de un asistente virtual.
* Desplegar la herramienta junto a un sistema de gestión para la creación de asistentes virtuales.

**Hipótesis:**

Se implementará un algoritmo para la construcción del conocimiento de un asistente virtual de forma automática, contribuyendo a reducir el costo de creación de un Asistente Virtual y disminuir el posible error humano en la creación de asistentes virtuales.

# **Capítulo 1. Marco Referencial**

En este capítulo se explican los principales aspectos teóricos, los conceptos básicos de las tecnologías y la caracterización de las herramientas computacionales utilizadas.

## **1.1 Los marcos o frameworks para desarrollar chatbot**

Son espacios digitales donde los desarrolladores crean bots desde cero a partir de funciones y clases predefinidas. Los marcos hacen que sea más fácil crear un chatbot personalizado que comenzar de la nada, pero aún requieren conocimientos avanzados de codificación para su uso aunque algunos dan facilidad de arrastrar y soltar como forma de desarrollo sin mucho código o casi ninguno. [5]

## **1.2 Estado del arte de los frameworks para desarrollar chatbot**

### **Azure Bot Service**

**Azure Bot Service** es una herramienta en la nube que entra en la categoría Plataformas y herramientas de Chatbot de una pila tecnológica. [6] [11] [10]

Cuenta con la herramienta **Bot Framework Composer:**

Permite comenzar con una plantilla y un lienzo visual de creación de código abierto para ampliar y personalizar la conversación con código y componentes precompilados. [6] [11]

**Azure Bot Service** proporciona justo lo que necesita para crear y conectar bots inteligentes que interactúan de forma natural dondequiera que estén hablando sus usuarios, desde mensajes de texto / sms hasta Skype, Slack, correo de Office 365 y otros servicios populares.

Más de 18 empresas usan la tecnología de Microsoft en sus sistemas de trabajo como por ejemplo LUMENEO, Autonom8 y WBOT. [6] [11]

***Integraciones de Azure Bot Service***

Slack , Skype , API Telegram , son algunas de las herramientas populares que se integran con **Azure Bot Service**. [6] [11]

***Ventajas***

* IA y lenguaje natural.
* Código Abierto y extensible (Bot Framework Composer).
* Soluciones de nivel empresarial.
* Propiedad y control.

***Contras*** [6] [10]

* El SDK de su sistema tiene un nivel de complejidad relativamente alto, es avanzado, lo que requiere tiempo dominar su uso y más para personas que no cuentan con mucha experiencia usando esta plataforma o parecidas pertenecientes a Microsoft.
* Bot Framework Composer depende de los servicios en la nube de Azure.

***Azure Bot Service en cuanto a precios*** [6] [10]

**Gratis:** Este es un nivel gratuito sin costo con un total de **10 000 mensajes/mes**, herramientas de creación de bots y canales estándar gratuitos.

**Premium**: Este nivel es una versión de pago de**0.50 USD por cada 1000 Mensajes,** permiten que el chatbot se comunique con los usuarios dentro de su propia aplicación o en su aplicación web además de contener herramientas de creación de bots y canales estándar gratuitos y premium. Aparte de esto, también le cobran por los recursos consumidos en las funciones de Azure y la aplicación web de Azure.

### **DialogFlow**

Dialogflow desarrollada por Google, es una plataforma con comprensión del lenguaje natural que te facilita el diseño de una interfaz de usuario de conversación y su integración a tu aplicación para dispositivos móviles, aplicaciones web, dispositivos, bots, sistemas de respuesta de voz interactiva, etc. Proporciona nuevas y atractivas formas para que los usuarios interactúen con tu producto. [7] [12] [13] [10]

Dialogflow puede analizar múltiples tipos de entradas de tus clientes, incluidas entradas de texto o audio (como las de un teléfono o una grabación de voz). También puede responder a tus clientes de varias maneras, ya sea a través de texto o con voz sintética. [7] [12] [13]

***Integraciones de Dialogflow***

Dialogflow proporciona a los desarrolladores una variedad de plataformas para integrar sus agentes conversacionales. Esto incluye Facebook Messenger, Skype, Slack, Twilio, Viber, Twitter, iPhone, Google Assistant y otros. [7] [12] [13]

**Ventajas de Dialogflow**

* Agentes conversacionales integrados.
* Webhooks personalizados.
* Gran interfaz.
* Integraciones Out-Of-The-Box (OOTB), no requiere mucho desarrollo o personalización.
* Base de conocimientos.
* Multilingüe.
* Visualización rápida.

**Desventajas** **de Dialogflow** [7] [10]

* No es una plataforma tan flexible en algunas facetas. Por ejemplo, si decide que mover una respuesta de seguimiento bajo una intención diferente, no puede simplemente arrastrarla debajo de la intención deseada. En su lugar, deberá eliminar la intención existente, crear una nueva intención en una ubicación diferente y volver a escribir todas las frases de entrenamiento que ya haya creado, lo que desperdicia tiempo y resulta tedioso.
* Mucho trabajo manual, tiempo y capacitación: En muchos casos, Dialogflow hace que sea más difícil de lo que debería ser automatizar los procesos y ampliar el aprendizaje de su agente conversacional. Esto puede ser molesto porque tiene que ingresar muchas cosas manualmente, especialmente cuando considera la necesidad de entrenar a su bot con el tiempo.

**Precios de DialogFlow:** La edición estándar es gratuita, en caso de ser usado para muchas peticiones lo enviará a la versión de pago que cobra $0.002 por solicitud. Sin embargo, CX Agent Edition cobra $20 por cada 100 sesiones de chat y $45 por cada 100 sesiones de voz.

### **Rasa**

Rasa es una plataforma de código abierto para desarrollar IA conversacional (AVO). Las plataformas de código abierto son software con código fuente que cualquiera puede inspeccionar, modificar o mejorar. Al ser de código abierto, los desarrolladores podrán integrar características y funcionalidades adicionales según sus requisitos. La plataforma es fácil de personalizar y flexible, por lo tanto, se puede modificar según sus necesidades. [8] [13] [14]

Tiene un marco para la comprensión del lenguaje natural (NLU), la gestión del diálogo y las integraciones. Rasa X es un conjunto de herramientas gratuitas utilizadas para mejorar los asistentes contextuales creados con Rasa Open Source. Juntos, incluyen todas las características para crear excelentes chatbots basados en texto y voz. [8] [13] [14]

Hay varias compañías que usan Rasa y algunas de ellas son ERGO, Orange, Lemonade y T-Mobile. [8] [13] [14]

**Características** [8] [13] [14] [10]

* Integración en sistemas existentes. Al ser de código abierto se integra sin problemas aprovechando los beneficios de varios sistemas Back-end, APIs y Robotic Process Automation (RPA)
* Soporta varias intenciones (intent) únicas y múltiples para comprender lo que el usuario quiere, también admite entidades pre-entrenadas y personalizadas para ayudar a modificar la intención según la solicitud del usuario.
* Aprendizaje interactivo.
* Integración con aplicaciones de mensajería como Facebook messenger, Google Home, Rocket, Slack, Telegram y otras.
* Inteligencia multilingüe.
* Estructura y nivel de desarrollo sólido, avanzado.
* Se basa en el lenguaje de programación Python, uno de los más flexibles para desarrolladores.

**Ventajas de RASA:**

* Rasa tiene uno de los conjuntos de documentación y la comunidad de soporte en línea más completa. Esto es importante porque Rasa requiere una gran cantidad de conocimientos técnicos para su uso.
* Muchas opciones y posibilidades de personalización. Esto permite a los desarrolladores crear asistentes de texto y voz muy únicos impulsados por IA.
* Convierta el texto de forma libre en cualquier idioma en datos estructurados. Admite intenciones únicas y múltiples y entidades pre-entrenadas y personalizadas.

**Desventajas de RASA:**

* Requiere un nivel técnico considerable por su complejidad de desarrollo (comandos y trabajo con archivos) y su sofisticación.
* Compatibilidad entre su sistema y el de su Computadora Personal (PC), hay que mirar o elegir bien la versión de sus programas más adecuado para tu PC, y como tiene muchas versiones se dificulta muchas veces la compatibilidad.

### **BotPress**

Botpress es una plataforma de código abierto para construir asistentes virtuales de forma fácil, accesible y rápida. [9] [15] [10]

La creación de chatbots en esta plataforma es relativamente fácil para los desarrolladores. Tiene una unión de código repetitivo y la infraestructura que necesita para poner en marcha un chatbot. Es una plataforma completa para desarrolladores con todas las herramientas que necesita para construir, implementar y administrar chatbots de nivel de producción en un tiempo récord.

Botpress se ha implementado en producción miles de veces en 4 continentes. Una amplia gama de empresas, incluidas agencias digitales, organizaciones Fortune 500 (lista de Compañías), gobiernos y nuevas empresas están construyendo asistentes digitales con Botpress Platform. [9] [15]

**Características**

* Tareas integradas de procesamiento de lenguaje natural, como reconocimiento de intenciones, revisión ortográfica, extracción de entidades, etiquetado y otras.
* Un estudio de conversación visual para diseñar conversaciones y flujos de trabajo de varios turnos.
* Un emulador y un depurador para simular conversaciones y depurar tu chatbot.
* Soporte para canales de mensajería populares como Slack, Telegram, MS Teams, Facebook Messenger y un chat web incrustable.
* Un SDK y un editor de código para ampliar las capacidades. [9] [10]

**Ventajas de BotPress**

* Es independiente de la nube, por lo tanto, puede ejecutarlo en la plataforma en la nube o en la infraestructura local de su elección, lo que le brinda un control total sobre la privacidad de sus datos.
* El núcleo de la plataforma Botpress es de código abierto con miles de colaboradores de GitHub y observadores, y un próspero foro comunitario.
* La plataforma ofrece una gran experiencia de desarrollador al ser notablemente flexible, fácil de usar y rápida. Soporte para múltiples idiomas, incluyendo francés, árabe, español y más.
* Sin bloqueo de proveedores.

**Desventajas de BotPress** [9] [10]

* La aplicación se actualiza con bastante frecuencia y la documentación no va actualizada con estos cambios.
* Los videos tutoriales de la plataforma no son muchos y no abarcan todo, además no se han actualizado por lo que el aprendizaje se hace con dificultad.
* Difícil ejecutar varias instancias desde una instalación.
* En un inicio es un tanto complicado a nivel técnico.

De las plataformas estudiadas se puede decir que lo mejor es el uso de sistemas Open Source (Código Abierto) como lo son BotPress y RASA por su manejo y la posibilidad de modificar en aras de adaptar y dar mejores facilidades en comparación con las otras plataformas, permite una mayor colaboración entre los que usan este tipo de sistemas y por lo tanto tienen una gran comunidad de desarrolladores activos. En el caso de estas dos herramientas si se quiere un AV fiable, simple en funcionamiento y que esté disponible rápidamente, Botpress en la mejor opción. Si busca un AV fiable también, pero con algo más de complejidad en funcionamiento y desarrollo, Rasa es la mejor opción. Una relación entre estas dos plataformas que abarca casi cualquier proyecto dependiendo de su uso final, pero independientemente de su magnitud e impacto.

Las otras plataformas: Azure Bot Service, DialogFlow y Amazon Lex si bien son bastante completas y son las más usadas por las grandes empresas, estas son sistemas privados y hoy la industria del software libre es un gran paso de avance en la tecnología, una mejor alternativa, está en ascenso y es más flexible para los desarrolladores; aunque Azure Bot Service es la mejor alternativa entre estos sistemas privados si tiene ya experiencia y busca una buena estructura para un gran proyecto, pero si está comenzando y tiene poca experiencia la mejor es DialogFlow.

A partir del análisis realizado es mejor el empleo de BotPress y RASA, que brindan lo suficiente para una buena experiencia y resultados, su sistema de procesamiento de lenguaje natural tiene mejor desempeño que los otros sistemas, además es empleado por renombradas empresas en sus plataformas de servicios.

## **1.3 Herramientas, Lenguajes de programación y Tecnologías**

Las herramientas son objetos elaborados a fin de facilitar la realización de una tarea. Se diseñan y fabrican para cumplir uno o más propósitos específicos, por lo que son generalmente artefactos con una función técnica. Un lenguaje de programación es un lenguaje formal que especifica una serie de instrucciones para que una computadora produzca diversas clases de datos. Los lenguajes de programación pueden usarse para crear programas que pongan en práctica algoritmos específicos los cuales controlan el comportamiento físico y lógico de una computadora. La tecnología es la ciencia aplicada a la resolución de problemas concretos. Constituye un conjunto de conocimientos científicamente ordenados, que permiten diseñar y crear bienes o servicios que facilitan la adaptación al medio ambiente y la satisfacción de las necesidades humana.

### **Python**

Python 3.7 es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)) cuya filosofía hace hincapié en una [sintaxis](https://es.wikipedia.org/wiki/Sintaxis) que favorezca un código legible. Se trata de un lenguaje de programación [multiparadigma](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_multiparadigma), ya que soporta [orientación a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), [programación imperativa](https://es.wikipedia.org/wiki/Programaci%C3%B3n_imperativa) y, en menor medida, programación funcional. Es un [lenguaje interpretado](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_interpretado), usa [tipado dinámico](https://es.wikipedia.org/wiki/Tipado_din%C3%A1mico) y es [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma). Posee una licencia de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto), denominada [Python Software Fundation License](https://es.wikipedia.org/wiki/Python_Software_Foundation_License) que es compatible con la [Licencia pública general de GNU](https://es.wikipedia.org/wiki/GNU_General_Public_License) [17]. Se utilizó como lenguaje para la implementación de la Aplicación informática.

### **PyCharComunity**

PyCharComunity Edition 2021.1.2 es un entorno de desarrollo integrado (EDI) utilizado en la programación de computadoras, específicamente para el lenguaje Python. Proporciona análisis de código, un depurador gráfico, un comprobador de unidades integrado, integración con sistemas de control de versiones (VCSes) y es compatible con el desarrollo web con Django. PyCharm es multiplataforma, con versiones de Windows, MacOS y Linux [18]. La Edición comunitaria se publica bajo la Licencia de Apache.

### **RASA**

RASA es una empresa que desarrolla software para ofrecer soluciones conversacionales de forma Open Source en Python.

Es el framework para crear Asistentes Virtuales que se usará en el trabajo de investigación. (*Ver* [*Rasa*](#_Rasa) *en el Estado del Arte*).

### **MongoDB**

MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. [19]

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

MongoDB es una base de datos adecuada para su uso en producción y con múltiples funcionalidades. Esta base de datos se utiliza mucho en la industria.

### **YAML**

YAML es un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python, Perl, así como en el formato de los correos electrónicos. [20]

YAML fue creado bajo la creencia de que todos los datos pueden ser representados adecuadamente como combinaciones de listas, hashes (mapeos) y datos escalares (valores simples). La sintaxis es relativamente sencilla y fue diseñada teniendo en cuenta que fuera muy legible pero que a la vez fuese fácilmente mapeable a los tipos de datos más comunes en la mayoría de los lenguajes de alto nivel.

Es el formato de los archivos de entrenamiento que usan los chatbots creados en RASA.

### **GIT**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

GitHub es una Plataforma de Desarrollo Colaborativo para alojar proyectos en la nube de forma online (fuera de línea) utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

Específicamente para trabajar se usó la versión para escritorio de GitHub (GitHub Desktop).

## **1.4 Metodología de desarrollo de software**

**Programación Extrema (XP)**

XP es una metodología ágil de software que tienen como propósito satisfacer a los clientes mediante la entrega temprana y continua de un software funcional, cuando ello implica incluso apoyar el cambio de los requerimientos en cualquier etapa del desarrollo. Está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP [21] se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Para desarrollar un proyecto de software, la metodología propone cuatro fases, como se muestra a continuación:

**Figura 1.1:** Etapas de la Programación Extrema.

**Planificación**

En esta fase, se plantean a grandes rasgos las historias de usuarios (HU) que son de interés para la primera entrega del producto. Las historias de usuario son construidas con la información proporcionada por los clientes y el equipo de desarrollo comienza a familiarizarse con las herramientas, metodología y prácticas que serán usadas para realizar el proyecto. Las HU fueron diligenciadas por alguno de los desarrolladores, con el fin de que el cliente pudiera concentrar su atención en el análisis del requerimiento o en el caso de que se estuviera evaluando el diseño o una entrega de iteraciones. Pese a que el cliente no fue quien escribió y diligenció las HU, siempre se contó con su revisión previa antes de finalizar la reunión. Las HU representan los requerimientos de software, y son descritas bajo el lenguaje del cliente.

**Diseño**

En esta fase se establece la prioridad de cada HU y, correspondientemente, los programadores establecen una estimación de esfuerzo necesario para cada una de ellas. El orden de las historias implementadas en las iteraciones será determinado por el cliente.

**Codificación**

Prácticamente desde un principio se hacen pruebas en XP para favorecer entregas frecuentes al cliente que es el objetivo fundamental de la metodología, es imprescindible la participación del cliente como tal, o uno de los elementos fundamentales de la metodología el “Cliente In Situ” que es un representante del cliente.

**Prueba**

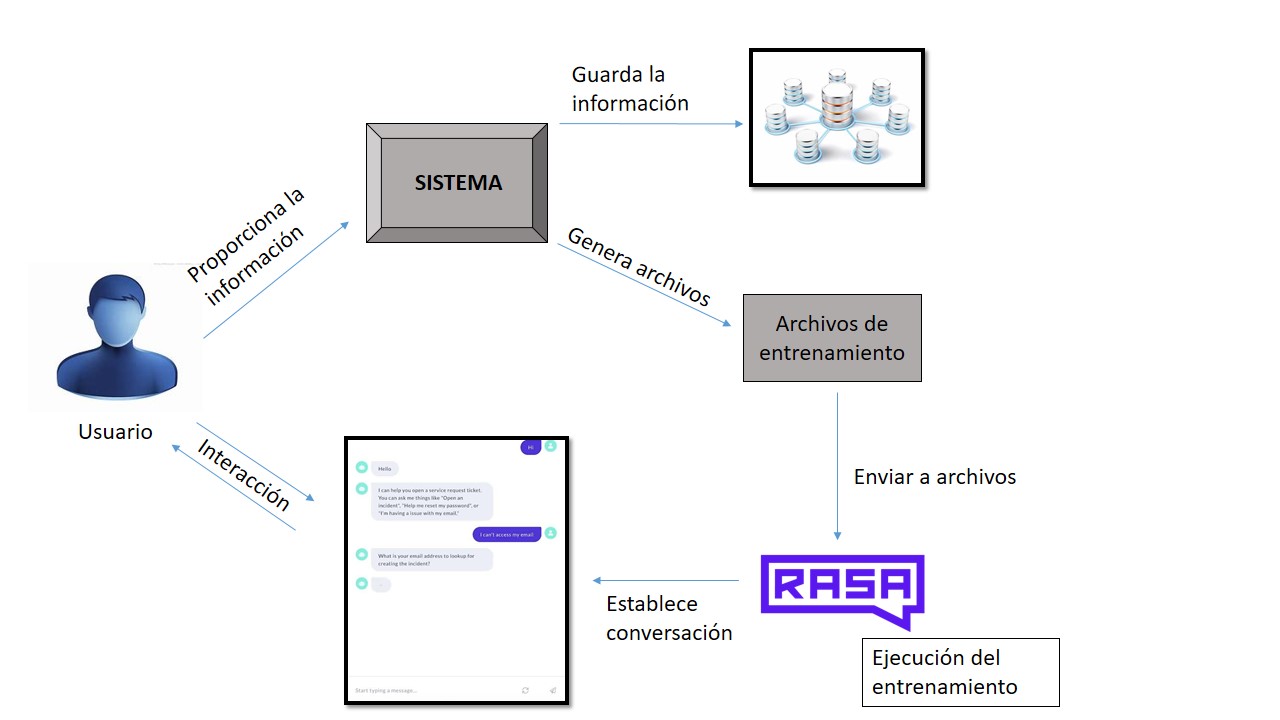
Se realizan testeos y se verifica que se han implementado todas las HU definidas en la fase de planificación o sus actualizaciones, es de vital importancia la participación del cliente en estas verificaciones.

# **Capítulo 2. Planificación y Diseño**

En este capítulo se documentan los aspectos más relevantes que fueron identificados durante el estudio del contexto del software y el sistema a partir de las necesidades encontradas. Se detallan los requerimientos del sistema mediante las Historias de Usuarios, así como aspectos relacionados con el diseño del *software*, tales como: modelo de clases, diagramas físicos de la base de datos y la arquitectura y los aspectos de la normalización.

## **2.1 Propuesta del Sistema**

Con el objetivo de dar solución al problema planteado anteriormente se propone desarrollar un Sistema de Generación de conocimiento para un Asistente Virtual hecho en Rasa que permita a partir de la entrada de datos *(Estos datos podrían guardarse en una base de datos para cargarlos nuevamente si es necesario)* poder generar preguntas y respuestas, para luego crear archivos de entrenamiento en el formato que Rasa entiende. Los asistentes virtuales podrán una vez entrenados con los archivos creados con el sistema saber responder dudas respecto al tema proporcionado.



**Figura 2.1.** **Estructura del Sistema**

## **2.2 Usuarios del Sistema**

Los usuarios son todas aquellas personas que interactúan de alguna forma con esta herramienta o que desempeñan algún rol específico. A continuación, se describen las actividades que realiza cada rol de usuario.

**Tabla 2.1 Usuarios del Software**

|  |  |
| --- | --- |
| **Usuario** | **Descripción** |
| Administrador | Acceso a funciones internas del sistema, base de datos y procesos en general del sistema. |
| Cliente | Acceso a realizar las operaciones comunes para lo que fue diseñada la herramienta, dichas operaciones se realizan desde la interfaz gráfica de la misma. |

## **2.3 Planificación del desarrollo del Sistema**

La planificación es la etapa inicial de todo proyecto. Es aquí donde se comienza a interactuar con el cliente para descubrir los requerimientos del sistema y realizar los ajustes a la metodología según las características del software.

## **2.3.1. Historias de usuarios**

Las Historias de Usuarios (HU) son la técnica utilizada en XP para especificar los requisitos de software, tanto funcionales como no funcionales y se descomponen en tareas de ingeniería asignadas a los programadores para ser implementadas durante las iteraciones. Están escritas con el vocabulario del cliente, no con vocabulario técnico. El tratamiento de las HU es muy dinámico y flexible, en cualquier momento las HU pueden romperse, emplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas.

A continuación, se hace una lista de las historias de usuario de la aplicación:

* Autenticación del Sistema.
* Crear Asistente Virtual
* Gestión de Conocimiento.

**Tabla 2.2 Historia de Usuario: Autenticación del Sistema**

|  |  |
| --- | --- |
| **Historia de usuario** | |
| **Número:1** | **Nombre de HU:** Autenticación del Sistema |
| **Usuario:** Cualquier usuario | |
| **Prioridad en negocio:** Alta | **Riesgo de desarrollo:** Alta |
| **Descripción: Donde el usuario inicia sesión para entrar al sistema, también en caso de no tener cuenta puede crear una.** | |
| **Observaciones:** Ninguna | |
| **Propuesta de interfaz:**  D:\DATOS ERNESTO\ING INFORMATICA\TESIS NETO\Documentos extras\Propuestas de interfaz\Login.png | |

**Tabla 2.3 Historia de Usuario: Crear Asistente Virtual**

|  |  |
| --- | --- |
| **Historia de usuario** | |
| **Número:2** | **Nombre de HU:** Crear Asistente Virtual |
| **Usuario:** Cualquier usuario | |
| **Prioridad en negocio:** Alta | **Riesgo de desarrollo:** Alta |
| **Descripción: Donde el usuario crea su Asistente Virtual con el que podrá interactuar.** | |
| **Observaciones:** Ninguna | |
| **Propuesta de interfaz:**  **D:\DATOS ERNESTO\ING INFORMATICA\TESIS NETO\Documentos extras\Propuestas de interfaz\Crear Asistente.png** | |

**Tabla 2.4 Historia de Usuario: Gestión de Conocimiento**

|  |  |
| --- | --- |
| **Historia de usuario** | |
| **Número:3** | **Nombre de HU:** Gestión de Conocimiento |
| **Usuario:** Cualquier usuario | |
| **Prioridad en negocio:** Alta | **Riesgo de desarrollo:** Alta |
| **Descripción: Donde se realizan todas las operaciones de generar conocimiento para el asistente virtual.**  Operaciones:   1. Cargar archivos con los datos (Archivo de texto). 2. Análisis de contenido para generar preguntas/respuestas. 3. Generar archivos de entrenamiento. 4. Guardar información y resultados. 5. Cargar información y resultados ya analizados. | |
| **Observaciones:** Ninguna | |
| **Propuesta de interfaz:**  **D:\DATOS ERNESTO\ING INFORMATICA\TESIS NETO\Documentos extras\Pagina principal.png** | |

### **2.3.2. Historias técnicas**

Las historias técnicas (HT) son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, confiable y rápido. Establecen condiciones mínimas para un correcto desempeño del sistema.

HT1: Usabilidad

El sistema debe brindar al usuario una clara navegabilidad de las funcionalidades del sistema.

HT2: Confiabilidad

El sistema debe ser capaz de funcionar correctamente cada vez que se utilice, las funciones para las que fue diseñado deben arrojar los resultados que el usuario espera, dichos resultados deben ser íntegros.

HT3: Seguridad

El espacio de trabajo de cada usuario será protegido mediante ***Autenticación personal*** para entrar y poder realizar acciones con la herramienta. La información proporcionada será almacenada en bases de datos, por tanto, el sistema debe asegurar la información que en la misma se manipulan.

HT4: Requisitos de Software

* Microsoft Windows 10 u 11
* Python 3.7

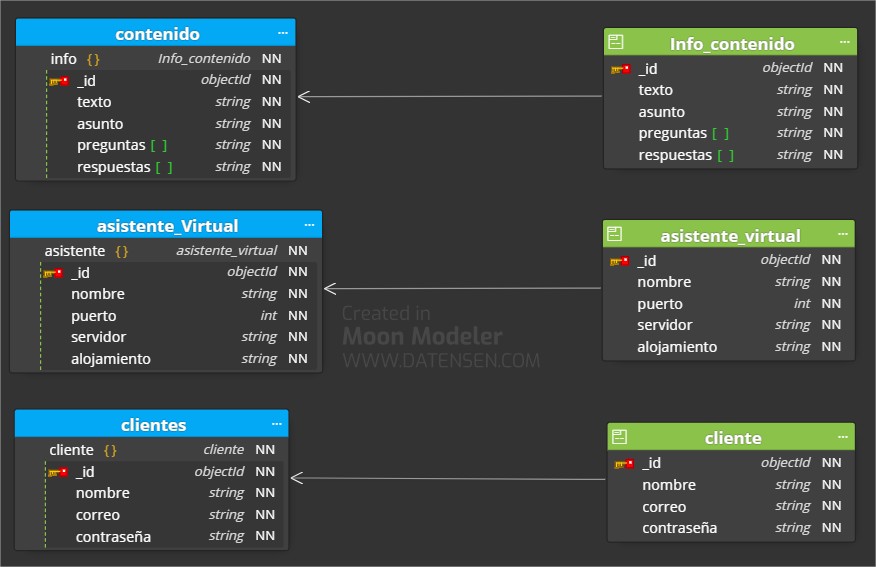
HT5: Requisitos de Hardware

Los requerimientos mínimos de hardware para correr la aplicación:

* Procesador Intel Corei3 a 2.3 GHz
* 4.0 GB de memoria RAM
* Disco duro de 500 GB o superior.

### **2.3.3. Diseño de Base de Batos**

El sistema cuenta con una base de datos no relacional (noSQL) que está basada en documentos ya que la herramienta de Gestión de base de Datos usada es MongoDB, sólo contamos con tres colecciones (como las tablas en es los sistemas SQL) las cuáles contienen documentos (como las filas de las tablas en es los sistemas SQL) y estos a su vez contienen los campos (clave-valor).



**Figura 2.2 Diagrama de Base de Datos**

**Colecciones Presentes**

* **Contenido:** Es donde se encuentran los documentos con la información y los resultados (texto con el contenido a analizar, las preguntas y respuestas derivadas del texto, etc)
* **Asistente\_Virtual**: Es donde se encuentra los documentos con la información respecto al chatbot (nombre, puerto, servidor, etc)
* **Clientes**: Es donde se encuentran los documentos con la información correspondiente a los usuarios que usan la herramienta. (nombre, correo, contraseña de inicio de sesión).

**Documentos Presentes**

* **Info\_contenido:** Es donde se encuentran los diferentes campos respecto a todo lo relacionado con la información (básicamente el texto con el contenido con el que se creará el conocimiento para el AV) que se le da a la herramienta y los resultados del análisis, uno de estos campos es el **\_id** que es el identificador de cada documento. Estos documentos pertenecen a la colección ***Contenido.***
* **Asistente:** Es donde se encuentran los diferentes campos respecto a todo lo relacionado con la información acerca de los asistentes que se hayan creado. Estos documentos pertenecen a la colección ***Asistente\_Virtual.***
* **Cliente:** Es donde se encuentran los diferentes campos respecto a todo lo relacionado con la información acerca de los usuarios que usen la herramienta. Estos documentos pertenecen a la colección ***Clientes.***

# **Capítulo 3. Implementación y Prueba**

Referencias Bibliográficas

[1] A. Augello, G. P., A. Machi, y S. Gaglio. (2012). An approach to enhance chatbot semantic power and maintainability: experiences within the FRASI project.

[2] Fourault, S. (2017). The Ultimate Guide to Designing A Chatbot Tech Stack. Retrieved from <https://chatbotsmagazine.com/the-ultimate-guide-to-designing-a-chatbot-tech-stack-333eceb431da> (20 de octubre de 2020)

[3] Arce, G. P., & Arce, R. P. (2021). Transform Your Business with AI Chatbots.

[4] Ernesto Guerra, Amanda, el chatbot que sabe de elecciones en Cuba. <https://planetachatbot.com/amanda-el-chatbot-que-sabe-de-elecciones-en-cuba-e2c3b5a7d835>

[5] <https://www.linkedin.com/pulse/dialogflow-vs-lex-watson-azure-bot-chatbot-quick-sherwin-fernandes> 30-10-2021

[6] <https://azure.microsoft.com/en-us/services/bot-services/> 4-11-2021

[6.1] <https://paginapropia.com/15-ventajas-y-desventajas-de-microsoft-azure/> 6-11-2021

[6.2] <https://stackshare.io/stackups/dialogflow-vs-microsoft-bot-framework/> 5-11-2021

[6.3] <https://rijsat.com/2021/11/03/create-azure-bot-service-from-azure-portal/> 6-11-2021

[7] <https://cloud.google.com/dialogflow/docs> 4-11-2021

[7.1] <https://cloud.google.com/dialogflow?hl=es> 5-11-2021

[7.2] <https://www.chatbots.org/dialogflow> 6-11-2021

[7.3] <https://www.capterra.es/reviews/180853/dialogflow?overall_rating_ge=5> 6-11-2021

[7.4] <https://www.linkedin.com/pulse/dialogflow-vs-lex-watson-azure-bot-chatbot-quick-sherwin-fernandes> 5-11-2021

[8] <https://rasa.com/> 3-11-2021

[8.1] <https://www.spaceo.ca/ai-chatbot-development-using-rasa-reasons/> 3-11-2021

[8.2] <https://www.chatbots.org/best-chatbot-builders#builder5> 6-11-2021

[9] <https://botpress.com/docs/introduction> 3-11-2021

[9.1] [https://www.getapp.es/reviews/2047766/botpress 4-11-2021](https://www.getapp.es/reviews/2047766/botpress%204-11-2021)

[10] <https://www.saashub.com/compare> 5-11-2021

[10.1] <https://www.spaceo.ca/top-ai-chatbot-frameworks/> 7-11-2021

[11] Waghmare, C. (2019). Introducing Azure Bot Service Building Bots for Business.

[12] Navin Sabharwal, A. A. (2020). Cognitive Virtual Assistants Using Google Dialogflow

Develop Complex Cognitive Bots Using the Google Dialogflow Platform.

[13] Raj, S. (2019). Building Chatbots with Python Using Natural Language Processing and Machine Learning.

[14] Álvaro Castillo Cabero, P. P. M., Joan Antoni Pastor Collado. (2020). Rasa Framework: Análisis e implementación de un ChatBot.

[15] Tri Chau Minh Tri, N. T. D., Koh Wee Lit (2020). Creating smart, human-like chatbot for businesses using BotPress platform.

[16] [Chatbots: La guía definitiva (2020) - IA Conversacional para Empresas | Artificial Solutions (artificial-solutions.com)](https://www.artificial-solutions.com/es/chatbots/) 11-2-2022

[17] [https://es.wikipedia.org/wiki/Python](https://es.wikipedia.org/wiki/Python%2011-2-2022)  11-2-2022

[18] <https://es.wikipedia.org/wiki/PyCharm> 11-2-2022

[19] [NoSQL - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/NoSQL) 11-2-2022

[20] [YAML - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/YAML) 11-2-2022

[21] «XP - Extreme Programing Ingenieria de Software». [XP - Extreme Programing Ingenieria de Software (mex.tl)](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html)

1. El Front-end es la parte de una web que conecta e interactúa con los usuarios que la visitan. Es la parte visible, la que muestra el diseño, los contenidos. [↑](#footnote-ref-1)