UNIVERSIDAD PRIVADA TELESUP

FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

TESIS

SISTEMA EXPERTO COMO SOPORTE A LA PROGRAMACIÓN CON LENGUAJE PYTHON 3.X BASADO EN MICROSOFT BOT FRAMEWORK

PARA OBTENER EL TÍTULO DE:
INGENIERO DE SISTEMAS E INFORMÁTICA

AUTOR:

Bach. DANIEL ALEJANDRO VEGA VILLAMONTE

LIMA – PERÚ

2018

ASESOR DE TESIS

……

Dr. Auccahuasi Aiquipa Wilver

JURADO EXAMINADOR

……

DR. VÁSQUEZ ROMERO, ISSAAK RAFAEL

Presidente

……

DR. NELSON MARCOS, RICHARDSON PORLLES

Secretario

……

ING. OVALLE PAULINO, DENIS CHRISTIAN

Vocal

DEDICATORIA

Este trabajo de investigación está dedicado a Dios, que tantas fuerzas me ha dado siempre y a mi amada familia, con muy especial mención a mi Madre por su apoyo y confianza.

AGRADECIMIENTO

El presente trabajo de investigación ha sido posible gracias a la Universidad Privada Telesup, gestora de mis competencias profesionales adquiridas, por los años que he transcurrido por sus aulas. Asimismo, quiero dar las gracias a todas las personas que han contribuido al desarrollo de esta tesis, por ejemplo, mi asesor el Dr. Wilver Auccahausi, el Lic. Pedro Purilla, el Ing. Bernardo Hermitaño, el Ing. Raúl Quispe, los estudiantes que me apoyaron en la encuesta, y todas aquellas personas que de diversas maneras contribuyeron a la culminación de este trabajo.

RESUMEN

El presente trabajo de investigación se enfocó en demostrar que el uso de un sistema experto basado en Microsoft Bot Framework sirve como soporte en la programación con lenguaje Python 3.x.

El problema general refiere a la interrogante ¿Cómo el sistema experto sirve como soporte a la programación con lenguaje Python 3.x basado en Microsoft Bot Framework?, cuya hipótesis principal señala que el sistema experto basado en Microsoft Bot Framework sirve como soporte en la programación con Python 3.x.

Como técnica de recolección de datos se aplicó una encuesta a una muestra conformada por 32 estudiantes de la carrera de Ingeniería de Sistemas de la Universidad Privada Telesup. En cuanto a la hipótesis específica, se comprobó que el uso del sistema experto se relacionaba significativamente con el número de errores de sintaxis cometidos en lenguaje Python 3.x.

Finalmente se recomienda desarrollar sistemas expertos como soporte a la programación en lenguajes como C#, Java, Ruby, etc.; también se recomienda para futuros desarrollos que el sistema experto guíe al usuario en la solución de errores lógicos y errores en tiempo de ejecución.

Palabras claves: Sistema experto, Microsoft Bot Framework, Python 3.x, programación, errores de sintaxis.

ABSTRACT

The present research work focused on demonstrating that the use of an expert system based on Microsoft Bot Framework serves as a support in the programming with Python 3.x language.

The general problem refers to the question: How does the expert system support programming with Python 3.x language based on the Microsoft Bot Framework? whose main hypothesis is that the expert system based on the Microsoft Bot Framework serves as a support for programming with Python 3.x.

As a data collection technique, a survey was applied to a sample made up of 32 students from the Systems Engineering program at the Telesup Private University. Regarding the specific hypothesis, it was found that the use of the expert system was significantly related to the number of syntax errors committed in Python 3.x language.

Finally, it is recommended to develop expert systems to support programming in languages such as C #, Java, Ruby, etc.; It is also recommended for future developments that the expert system guides the user in the solution of logical errors and errors in runtime.

Keywords: Expert system, Microsoft Bot Framework, Python 3.x, programming, syntax errors.

ÍNDICE DE CONTENIDOS

Página

CARÁTULA.....I

ASESOR DE TESIS II

JURADO EXAMINADOR III

DEDICATORIA IV

AGRADECIMIENTO V

RESUMEN VI

ABSTRACT VII

ÍNDICE DE CONTENIDOS VIII

ÍNDICE DE TABLAS XI

ÍNDICE DE FIGURAS XII

INTRODUCCIÓN 13

I. PROBLEMA DE INVESTIGACIÓN 15

1.1. Planteamiento del problema 15

1.2. Formulación del problema 16
1.2.1. Problema General 16
1.2.2. Problema específico 16

1.3. Justificación y aportes 16
1.3.1. Justificación Práctica 16
1.3.2. Justificación de la Viabilidad de la Investigación 18

1.4. Objetivos de la investigación 18
1.4.1. Objetivo General 18
1.4.2. Objetivo Específico 18

II. MARCO TEÓRICO 19

2.1. Antecedentes 19
2.1.1. Antecedentes Nacionales 19
2.1.2. Antecedentes internacionales 27

2.2. Bases teóricas de las variables 38
2.2.1. Bases teóricas de la variable independiente: Sistema Experto 38
2.2.1.1. Definición de Sistema Experto 38
2.2.1.2. Componentes de un sistema experto 40
2.2.1.3. Procesamiento del Lenguaje Natural (PLN) 42
2.2.1.4. Ventajas y desventajas de los sistemas expertos 43
2.2.1.5. Microsoft Bot Framework 44
2.2.1.6. Dimensiones del Sistema Experto 52
2.2.2. Bases teóricas de la variable dependiente: Lenguaje de Programación Python 3.x 53
2.2.2.1. Definición de Python 53
2.2.2.2. Características del lenguaje Python 3.x 54
2.2.2.3. Filosofía de Python 55
2.2.2.4. Elementos y estructuras básicas del lenguaje Python 56
2.2.2.5. Diferencias entre Python 2.x y 3.x 60
2.2.2.6. Definición de la dimensión de Python 60

2.3. Definición de Términos Básicos 61

III. MÉTODOS Y MATERIALES 64

3.1. Hipótesis 64
3.1.1. Hipótesis General 64
3.1.2. Hipótesis Específica 64

3.2. Variables de estudio 64
3.2.1. Definición Conceptual 64
3.2.2. Definición Operacional 65

3.3. Tipo y nivel de investigación 65
3.3.1. Tipo de investigación 65
3.3.2. Nivel de la investigación 65

3.4. Diseño de la investigación 66

3.5. Población y muestra de estudio 66
3.5.1. Población 66
3.5.2. Muestra 66

3.6. Técnicas e Instrumentos de Recolección de Datos 67
3.6.1. Técnica de recolección de datos: 67
3.6.2. Instrumento de recolección de datos: El Cuestionario 67
3.6.2.1. Validez del Instrumento 68

3.7. Métodos de Análisis de Datos 68

IV. RESULTADOS 69

4.1. Resultados Descriptivos 69

4.2. Prueba de Hipótesis 70
4.2.1. Hipótesis Principal 70
4.2.2. Contrastación de la Hipótesis Específica 70

V. DISCUSIÓN 74

VI. CONCLUSIONES 76

VII. RECOMENDACIONES 77

REFERENCIAS BIBLIOGRÁFICAS 78

ANEXOS 82

Anexo 1: Matriz de Consistencia 82

Anexo 2: Matriz de Operacionalidad 83

Anexo 3: Instrumento 84

Anexo 4: Validación de Instrumentos 85

Anexo 5: Matriz de Datos 89

Anexo 6: Propuesta de Valor 90

ÍNDICE DE TABLAS

Tabla 1 *Algunas definiciones de IA, organizadas en cuatro categorías* 38
Tabla 2 *Tipos de Activity en Microsoft Bot Framework* 49
Tabla 3 *Definición Operacional de las variables* 65
Tabla 4 *Validación de Expertos* 68
Tabla 5 *Resumen del procesamiento de datos del cuestionario* 69
Tabla 6 *Resumen Estadístico del cuestionario* 69
Tabla 7 *Comparación de desviaciones estándar* 72

ÍNDICE DE FIGURAS

Figura 1 Ilustración de montos de financiamiento en Chat Bots. 17
Figura 2 Ventana de Nuevo Proyecto de Microsoft Bot Framework 45
Figura 3 Tipos de comunicación virtual 46
Figura 4 Estructura del código de MessageController.cs 48
Figura 5 Código del método Post dentro de MessageController.cs 50
Figura 6 HandleSystemMessage dentro de MessageController.cs 51
Figura 7 Código de la clase RootDialog 52
Figura 8 Ejemplo con tratamiento de listas 54
Figura 9 Ejemplo de variables en Python 56
Figura 10 Comentarios en Python 57
Figura 11 Ejemplo de IF-ELSE en Python 57
Figura 12 Ejemplo de Bucle FOR 58
Figura 13 Ejemplo de Bucle While 58
Figura 14 Ejemplo de una clase y su correspondiente instancia. 59
Figura 15 Ejemplo de llamada de atributos y métodos 59
Figura 16 Árbol de excepciones en Python 60
Figura 17 Densidades suavizadas 71
Figura 18 Gráfico de Caja y Bigotes 72
Figura 19 Gráfico de Cuantiles 73
Figura 20 Comprobación de compatibilidad con navegadores 90
Figura 21 Foto 1 en la que se observa a los estudiantes resolviendo la prueba 91
Figura 22 Foto 2 en la que se observa a los estudiantes resolviendo la prueba 91
Figura 23 Foto 3 en la que se observa a los estudiantes resolviendo la prueba 91
Figura 24 Primer ejercicio del Pre test 92
Figura 25 Primer ejercicio del Pos test 92

INTRODUCCIÓN

El presente trabajo de investigación ha sido realizado con el propósito de contribuir a la educación y formación de los estudiantes de los cursos básicos de algorítmica, estructura de datos y lenguaje de programación, introduciéndolos en un lenguaje que cada día es más utilizado por diversas empresas y centros académicos en áreas como la inteligencia artificial y la computación cuántica. Este trabajo se encuentra estructurado en siete capítulos, a continuación, se detalla en qué consiste cada uno:

El capítulo I trata sobre el planteamiento del problema, en este apartado se describe el tema a investigar, se formula el problema general y los problemas específicos y finalmente se brinda la justificación correspondiente, a nivel práctico y a nivel de la viabilidad.

El capítulo II corresponde al marco teórico, el cual se divide en dos subcapítulos: el primero dedicado a los antecedentes (nacionales e internacionales) cada uno con su resumen y conclusiones; el segundo subcapítulo está dedicado al desarrollo teorético de la variable independiente (sistema experto) y de la variable dependiente (lenguaje de programación Python 3.x); en ambos casos se brinda la definición y dimensiones correspondientes.

El capítulo III se centra en los materiales y métodos utilizados en la investigación, es decir en la hipótesis general y las hipótesis específicas, las variables de estudio (las cuales se definen a nivel conceptual y operativo), el tipo, nivel y diseño de la investigación, la población y muestra del estudio y finalmente las técnicas, instrumentos y métodos de recolección y análisis de datos.

En el capítulo IV se presentan los resultados descriptivos de la investigación y la contrastación de las hipótesis.

Los capítulos V, VI y VII están centrados en la discusión, conclusiones y recomendaciones respectivamente.

Finalmente se presenta en los anexos la matriz de consistencia, la matriz de operacionalidad, el instrumento utilizado, el certificado de validez del instrumento mediante jucio de expertos, la matriz de datos y la propuesta de valor

Este trabajo constituye un antecedente para futuros desarrollos en cuanto a inteligencia artificial por parte de los estudiantes y futuros profesionales de nuestra Alma Mater.

I. PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento del problema

En los últimos años la inteligencia artificial y la computación cuántica se han desarrollado muy rápidamente debido en primer lugar a la necesidad de “identificar una explicación científica que diera cuenta en una actividad totalmente ordinaria de cuáles son las operaciones de cálculo que debemos realizar a diario.” (Annovi, 2016)

De esta forma desde la década del 50 del s. XX se han ido gestando diversos modelos teóricos y matemáticos (máquinas de Turing, redes neuronales, teoría general de sistemas, etc.) que si bien es cierto tenían como precursores a trabajos del s. XIX (por ejemplo, los de Charles Babbage y Ada Lovelance) encontraron mayor interés en aquellos años.

Gracias al incremento en la producción de sofisticados elementos electrónicos es que se ha podido llevar muchos de estos modelos al mundo real siendo uno de los de mayor interés el computador cuántico, por su aplicación a los mercados financieros y sobre todo a la criptografía.

Uno de los lenguajes que actualmente está siendo muy aplicado en campos como la inteligencia artificial y la computación cuántica por su fácil curva de aprendizaje y su gran potencia es Python, el cual cuenta con múltiples librerías como numpy y scipy para procesamiento científico de datos, scikit-learn y TensorFlow para inteligencia artificial y QISKit para computación cuántica, las cuales son constantemente usadas, revisadas y mejoradas por una comunidad grande de programadores.

De manera que cuantas más personas aprendan a programar mejor para su propia economía, la de sus familias y la de sus localidades caso contrario será la misma tecnología la que continué generando despidos laborales por la desaparición de trabajos que una máquina puede desarrollar mejor que una persona.

Con respecto a este último punto, debemos considerar lo siguiente:

En 1949 Norbert Wiener publicó un artículo sobre la inteligencia artificial en el que al exponer sus exitosas experiencias en la construcción de robots predijo que en el futuro todo el trabajo humano directo iría siendo sustituido por los equipos automatizados. Tal predicción la sustentó muy sólidamente a tal punto que generó una lógica preocupación en el mundo académico respecto del futuro. Poco después esas predicciones habrían de convertirse en una realidad lacerante (Roel Pineda, 2005).

1.2. Formulación del problema

1.2.1. Problema General

¿Cómo el sistema experto sirve como soporte a la programación con lenguaje Python 3.x basado en Microsoft Bot Framework?

1.2.2. Problema específico

¿Cómo usar el sistema experto basado en Microsoft Bot Framework para corregir errores de sintaxis relacionados al lenguaje de programación Python 3.x?

1.3. Justificación y aportes

1.3.1. Justificación Práctica

Este trabajo consistió en crear un chatbot para guiar al usuario en la programación con Python. El interés en cuanto a los chatbots ha crecido de forma exponencial, como afirma Hosseini (2017):

> Casi el 50 por ciento de las personas se contactan con un negocio a través de mensajes que a través una llamada telefónica. [Como se observa en la Figura 1] la cantidad de fondos ha aumentado de manera constante si tenemos en cuenta el 2012 y 2013. El aumento dramático en la financiación que ocurrió en 2016 podría explicarse por el hecho de que, en 2016, las empresas de tecnología ampliamente conocidas como Google Facebook,

Microsoft y Apple anunciaron la apertura de sus plataformas Chat Bot para que los desarrolladores construyan Chat Bots.

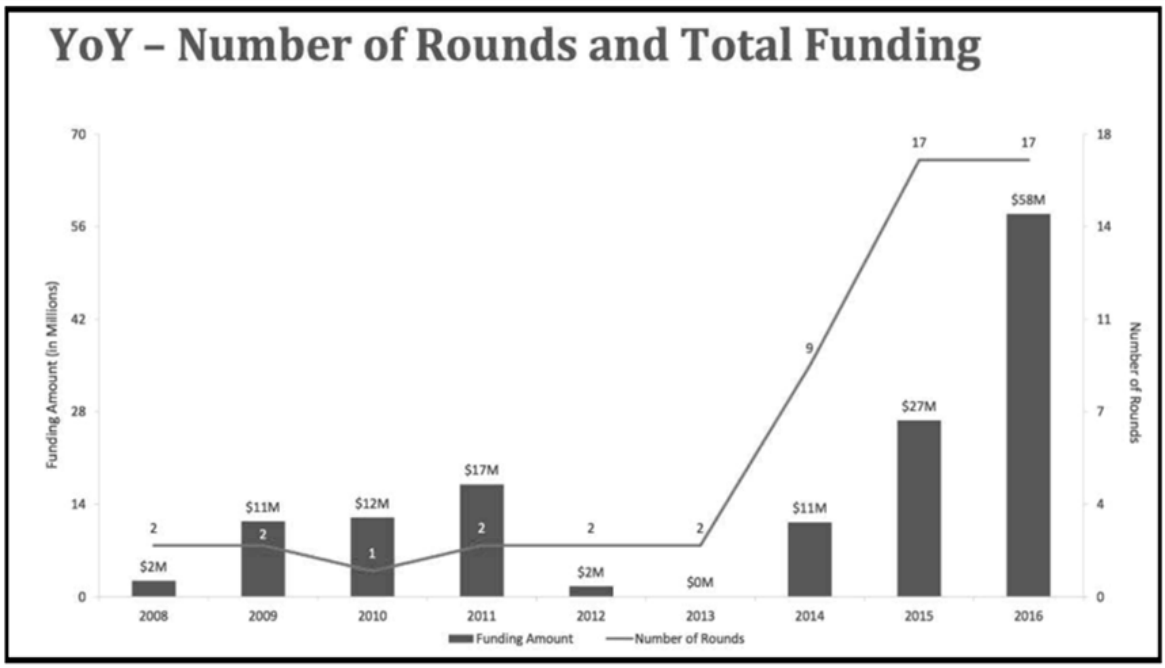

Figura 1: Ilustración de montos de financiamiento en Chat Bots.

Fuente: Sajjad Hosseini (2017)

Además es necesario el uso de un ChatBot para que el usuario reciba instrucción sobre la sintaxis del lenguaje Python. En cuanto a este punto, el chatbot puede indicar una dirección url en la cual el usuario tiene la posibilidad de encontrar más información para solucionar su problema.

Para la realización del ChatBot se utilizó el Microsoft Bot Framework el cual facilita ampliamente el desarrollo de bots, porque permite al programador modificar el código fuente. Este framework posee amplia documentación en inglés y español.

También es importante mencionar que se optó por utilizar Python 3 puesto que es la última versión y el soporte de Python 2.7 finalizará en 2020. En el marco teórico se especifican las diferencias entre Python 3 y Python 2.

1.3.2. Justificación de la Viabilidad de la Investigación

Se demuestra que esta investigación es viable por cuanto se encontró como antecedentes que se han realizado sistemas de instrucción exitosos (como en el caso de Cortana, Siri y el asistente de Whatsapp). Por obvias razones Cortana y Siri han sido desarrollados por un grupo importante de ingenieros bien preparados, lo cual generó como resultado en ambos casos un sistema que permite reconocer la voz humana, pasar la información de voz a texto, realizar el análisis semántico y sintáctico de ese texto, procesarlo, emitir una respuesta y nuevamente convertirla a formato de voz para comunicar. Incluso en caso no encontrar una respuesta en su base de datos realizan una consulta en internet y muestran información por ejemplo de Wikipedia.

Otros ejemplos son Eliza y Alizia, dos bots que permiten simular (pero nunca reemplazar) el comportamiento de una terapista, CleverBot que cada vez responde con más coherencia, diversos framework para crear chatbots, de los cuales uno de los más sofisticados es MoMo 1.0 el cual permite el uso de scripts avanzados. También se han realizado varios chatbots para la plataforma de celulares Android, IOS y Windows Phone.

1.4. Objetivos de la investigación

1.4.1. Objetivo General

Demostrar que el sistema experto sirve como soporte en la programación con lenguaje Python 3.x basado en Microsoft Bot Framework.

1.4.2. Objetivo Específico

Demostrar que el sistema experto basado en Microsoft Bot Framework sirve para corregir errores de sintaxis relacionados al lenguaje de programación Python 3.x

II. MARCO TEÓRICO

2.1. Antecedentes

2.1.1. Antecedentes Nacionales

En el campo de la inteligencia artificial se han desarrollado interesantes investigaciones, por ejemplo, Mendoza Cárdenas & Cáceres Zárate (2016) realizaron una tesina titulada “Sistema experto para la selección de postulantes en puestos de una agencia bancaria usando la metodología CommonKads” en la Universidad Nacional Mayor de San Marcos.

Resumen: El presente trabajo plantea el desarrollo de un Sistema Experto para la selección de personal en puestos de agencias bancarias. El objetivo principal es obtener al personal mejor capacitado para el puesto de trabajo. Los resultados obtenidos son reflejados en le eficiencia del proceso: reducción del tiempo de selección y de los costos del mismo, y del resultado final del proceso de selección.

Conclusiones: No se puede automatizar al 100% el proceso de selección de personal. Existen 2 tipos de Test usados en la mayoría de los procesos de selección: Cualitativos y Cuantitativos. Los test cuantitativos nos dan como resultado un valor numérico, el cual puede ubicarse en un rango de valores que nos permitan saber la competencia o no del candidato; por otro lado, los test cualitativos requieren la corrección de un profesional y el resultado de la evaluación dependerá en menor o mayor medida de la experiencia del evaluador. En este trabajo hemos usado ambos tipos de Test: Cualitativo (Test de Competencias) y cuantitativo (Test de Dominó). Se han revisado trabajos en los cuales la falta de una metodología para el desarrollo de SE, reduce el nivel de confianza de los resultados. Existen diversas metodologías para el desarrollo de sistemas expertos, en este trabajo hemos seguido la metodología CommonKads reconocida internacionalmente. Esta metodología nos brinda las pautas para definir adecuadamente nuestra base de conocimientos y motor de inferencia como base fundamental del desarrollo de estos Sistemas basados en conocimientos.

Mientras más específico sea el campo de aplicación del SE, mejores serán los resultados. Delimitar adecuadamente el uso del SE, permitirá tener una claridad de base de conocimiento y reglas necesarias para obtener las conclusiones correctas.

El uso de Sistemas Expertos de apoyo a la toma de decisiones, y en específico para este trabajo, ayudó a reducir el tiempo, costos y que los resultados sean más exactos. El tiempo empleado en los procesos de selección serán menores, ya que el tiempo empleado para calificar y obtener un resultado producto de las evaluaciones psicológicas será inmediato. Los costos, permitirá la reducción de personal evaluador, menor espacio físico (test online), y de cara a los proyectos, se evitarán reprocesos en selección de personal por incapacidad y falta de respuesta del personal seleccionado. Se evitará el error Humano inmerso en toda actividad en la cual intervenga el hombre, ya que la evaluación será automatizada según los parámetros determinados para las calificaciones de estos Test.

Se realizaron las pruebas necesarias para determinar el grado de confiabilidad del sistema y se obtuvieron resultados alentadores: sensibilidad 77.78%, especificidad 98.59%, valor predictivo positivo de 87.50% y factor de aceptación de 96.25%. Con estos resultados se garantiza la obtención de resultados confiables en la implementación del sistema.

Benites Llerena & Cahuata Peralta (2016) realizaron una tesis titulada: “Sistema experto para la detección de operaciones sospechosas de lavado de activos en entidades financieras utilizando CommonKADS” en la Universidad Nacional Mayor de San Marcos.

Resumen: El lavado de activos es un problema mundial capaz de corroer los sistemas financieros internacionales ocultando sus procedencias ilícitas en un vasto volumen de transacciones del mercado financiero oficial, siendo la detección de lavado de activos un proceso complejo que consume tiempo valorable a los oficiales de cumplimiento por la gran cantidad de operaciones bancarias realizadas por los clientes que deben ser analizadas y evaluadas para determinar si son operaciones bancarias sospechosas de lavado de activos. Las investigaciones realizadas se han centrado principalmente en técnicas de minería de datos y

técnicas de aprendizaje debido a las grandes cantidades de información que se maneja, reduciendo la precisión del resultado otorgado.

El presente trabajo tiene como objetivo presentar el diseño e implementación de un sistema experto con el fin de proporcionar apoyo a la detección de lavado de activos basándose en la determinación del riesgo de la transacción financiera inusual y el conocimiento del cliente. Para el desarrollo del sistema experto se usó la metodología CommonKADS por ser un estándar de facto en la construcción de sistemas basados en conocimiento. El sistema toma como entrada los datos de las operaciones bancarias contenidas en un caso inusual reportado y los datos del cliente obteniendo, mediante el motor de inferencia, el riesgo final del caso, que permitirá concluir si se debe reportar o no como sospechosa. Las pruebas realizadas han consistido en la evaluación de 251 casos que contienen el resultado del experto, obteniendo como resultado un porcentaje de acierto del 98.81% a la cuarta iteración.

Conclusiones: Existen diversos métodos relacionados con la detección del lavado de activos, pero no existe un caso de estudio para la detección del nivel de riesgo de casos de operaciones inusuales sospechosas de lavado de activos.

Se ha construido una base de conocimiento sobre la detección del nivel de riesgo de casos de operaciones inusuales sospechosas de lavado de activos en entidades financieras, a partir del conocimiento de expertos.

Se ha desarrollado el sistema experto usando la metodología CommonKADS cuya base teórica ha sido desarrollada en el capítulo 3 y la descripción del sistema, en el capítulo 4. El sistema experto ha sido desarrollado en el lenguaje de programación Java para el desarrollo y el lenguaje Clips para la elaboración de la base del conocimiento

La metodología CommonKADS aplicada cubrió todas las etapas del proyecto desde el estudio en organización, modelado del conocimiento, análisis y diseño, selección de la arquitectura, implementación y gestión del proyecto, demostrando ser una buena elección para el desarrollo de sistemas expertos.

A partir de 251 instancias de pruebas se muestra un índice de acuerdo de 0.99 respecto al conocimiento estándar. Además, muestra una sensibilidad de 0.97 y una especificidad de 0.99, los cuales indican que el sistema propuesto detecta de manera acertada los casos positivos de la referencia estándar, por lo que se recomienda su uso.

Espinoza Cubas (2015) realizó una tesis titulada: *"Análisis, diseño e implementación de un sistema experto para la evaluación de la calidad de tanques de almacenamiento de combustibles",* en la Pontificia Universidad Católica del Perú.

Resumen: Hoy en día la extracción de recursos naturales es una de las principales actividades en nuestro país. Gracias a la riqueza de nuestro territorio, múltiples empresas invierten en este sector, siendo las industrias mineras y de hidrocarburos las que generan mayores ingresos a los inversionistas y al país.

El incremento de estas actividades ha generado un incremento en la demanda de diferentes tipos de estructuras metálicas que den soporte a las diferentes industrias, como la minera, pesquera o petrolera. Una de las estructuras con mayor requerimiento son los tanques de almacenamiento, los cuales son usados para conservar distintos tipos de líquidos, minerales o combustibles que serán utilizados para la extracción, tratamiento, transporte o algún otro proceso que maneje la organización.

Los tanques de almacenamiento luego de ser diseñados e instalados en las plantas o puntos de extracción almacenan distintos tipos de líquidos y son expuestos a distintos tipos de condiciones climáticas. Esto a lo largo del tiempo hacen que se puedan presentar fallas o averías, las cuales podrían perjudicar la conservación de los líquidos al interior de estos.

El presente proyecto plantea la implementación de un sistema experto que pueda evaluar la condición y calidad de un tanque de almacenamiento de combustibles, utilizando las reglas ingresadas por un usuario experto, y en base a éstos poder determinar cuál es el estado y nivel de calidad actual de un tanque de almacenamiento.

Para esto se desarrollarán los principales componentes del sistema experto, los cuales consisten en: base de hechos, motor de inferencia, interfaz gráfica y una base de conocimientos. Por medio de la interfaz gráfica el usuario interactuará con el sistema experto e introducirá la información a la base de hechos, la cual en conjunto con las reglas almacenadas en la base de conocimientos será evaluada por el motor de inferencia, el cual realizará un análisis y brindará un resultado sobre la calidad del tanque a los especialistas, agilizando las labores dentro del proceso de calidad.

Conclusiones: Luego de realizar el presente trabajo se pudo llegar a las siguientes conclusiones:

Se logró realizar el análisis, diseño e implementación de un sistema para la evaluación de la calidad de tanques de almacenamiento, utilizando un sistema experto.

Se logró construir una base de conocimientos, en la cual se plasmaron las reglas de la norma API-650, para la evaluación de la calidad de los tanques, establecida por la American Petroleum Institute.

Se logró construir una base de hechos en la cual se almacenan todos los datos relacionados a un tanque de almacenamiento, esta base de hechos luego será utilizada junto con la base de conocimientos por el motor de inferencia.

Se logró construir un motor de inferencia utilizando el lenguaje de programación Java, el cual se encarga de realizar la evaluación de los tanques.

Se debe contar con una persona experta en el tema para poder realizar un proyecto de sistemas expertos, ya que sobre su conocimiento se construirá la base de la aplicación.

Los sistemas expertos son herramientas ideales para la evaluación de la calidad, no solo de tanques de almacenamiento o estructuras metálicas, sino para cualquier tipo de producto.

Los sistemas expertos no pueden reemplazar a un experto humano en la labor que este realiza, pero sí puede apoyarlo en su labor, realizando un rápido análisis del producto, el cual luego puede ser revisado por el experto humano.

Vásquez Ruiz (2013) realizó la tesis titulada *"Desarrollo de un sistema experto para evaluar las competencias en investigación de los docentes de la Universidad Nacional de San Martín"* en la Universidad Nacional de San Martín.

Resumen: En esta investigación se ha desarrollado un Sistema Experto para evaluar las competencias en investigación de los docentes de la Universidad Nacional de San Martín. Para lograr establecer procedimientos adecuados que permitan evaluar competencias en investigación. Se ha elaborado reglas y controles condicionados, que han juntado elementos evaluativos del ámbito cognitivo y correspondientes a los estados conductuales de un profesional. Todo esto ha sido posible gracias a los fundamentos metodológicos de la ingeniería del conocimiento y la inteligencia artificial que nos facilitan desarrollar aplicaciones para realizar tareas expertas.

Este estudio aplicativo busca aproximarse al rigor de una evaluación experta, aunque se ha abordado el problema desde la perspectiva básica de un test que ayudaría a identificar el nivel de competencia en investigación universitaria de un docente cualesquiera.

Las estrategias de la solución del Sistema Experto desarrollado subyacen en sus reglas lógicas que sacan conclusiones por cada respuesta contestada por el usuario evaluado. La medición del nivel de competencias en investigación de una persona es sumamente compleja, por la gran cantidad de factores que intervienen; sin embargo, un test automatizado, es una aproximación válida y de mucha importancia para la toma de decisiones de una organización o institución.

Finalmente, el trabajo de investigación es una solución tecnológica que integra un enfoque de la actitud y aptitud del evaluado en un software Web, que puede ampliarse y mejorarse, si fuera necesario dar mayor precisión a su funcionalidad experta.

Conclusiones: Al analizar la situación actual de la investigación en la Universidad se logró establecer algunos indicadores básicos en investigación, respecto al panorama internacional en I+D; también, se observó que la UNSM-T ha logrado avanzar en aspectos de infraestructura, sin embargo, carece de una producción en investigación científica con un estándar aceptado mundialmente, teniendo la principal dificultad la selección de personal altamente competitiva en investigación.

Se diseñó un prototipo mediante la metodología de Buchanan y en el esquema de conocimientos se integró dos enfoques evaluativos, de aptitud y actitud para medir el nivel de competencia en investigación, en el primero se incluyen cuarenta preguntas tipo test; y en el segundo, se incluye veintidós preguntas tipo test. Esto permitió tener una eficiencia muy aproximada a la evaluación de expertos humanos.

Se implementó el Sistema Experto con encadenamiento de reglas y controles condicionados en un entorno Web, bajo un nivel de performance de usabilidad adecuado.

Se evaluó la aplicación del Sistema Experto, al margen del test de consistencia del software, determinando su eficiencia y potencialidad en la medición de los niveles de competencias en investigación en la Universidad Nacional de San Martín.

Marín Méndez, Gutiérrez Lozano, & Hernández Valderrama (2015) realizaron una tesis titulada *"Implementación de un sistema Experto para el diagnóstico de desnutrición en niños menores de 5 años utilizando lógica fuzzy"* en la Universidad Nacional de Trujillo.

Resumen: La presente tesis plantea un modelo de Sistema Experto a través de la aplicación de las técnicas de la Inteligencia Artificial, el cual capta el conocimiento básico que permite a una persona desempeñarse como un experto frente a problemas complicados, en este caso el Sistema Experto se encargara de diagnosticar mediante la Lógica Difusa el grado de desnutrición del paciente.

Se diseña el Sistema Experto con las siguientes características: los síntomas y signos del paciente considerados como variables de entrada, una Base de Hechos que almacena los síntomas particulares del paciente, una Base de Conocimientos representado por las Reglas de Producción formalizados por la lógica de

predicados que contienen todo el conocimiento del experto humano, un motor de inferencia que utiliza la búsqueda en un árbol And/Or y finalmente las variables de salida.

Posteriormente se realiza la simulación del Sistema Experto a través de un prototipo desarrollado en el IDE Eclipse, así mismo, se diseña la base conocimientos con la ayuda de la librería “jFuzzyLogic” que permitirá verificar los resultados obtenidos y compararla con los resultados reales. De los resultados obtenidos en las pruebas efectuadas se evalúa el Sistema Experto llegándose a la conclusión de que los diagnósticos obtenidos tienen un grado de confiabilidad que es aceptado por el experto humano.

Conclusiones: Al concluir el presente trabajo se pretende alcanzar el objetivo general planteado, mediante la construcción de un sistema experto que pueda diagnosticar el grado de desnutrición en niños menores a 5 años y de esta manera ayude al tratamiento a seguir para prevenir futuras enfermedades.

Una de las herramientas que nos permite manejar de manera adecuada el lenguaje natural y ambiguo utilizado tanto por el médico como el paciente, es la lógica difusa, que nos permite evaluar síntomas difusos que el paciente presenta y proporciona un determinado grado de certeza.

Se diseñó la base de conocimiento, que abarca un 60% del conocimiento y experiencia del experto, esto hace que el sistema sea confiable. La base de conocimiento está construida en base a reglas, se trata de representar el conocimiento mediante reglas de reproducción, puesto que ofrecen una gran facilidad para la creación y la modificación de la base de conocimiento.

Las conclusiones presentadas se basan en la discusión de esta investigación, las cuales al concretarse enmarcará la solución de los problemas descritos, logrando mejorar el diagnóstico de desnutrición de un niño menor a 5 años, de acuerdo a los síntomas que presenta este, y de esta manera evitar mayores complicaciones, tomando decisiones oportunas tempranas.

Diremos también que la utilización de la lógica difusa para el control de sistemas tiene sus ventajas y desventajas, y por tanto hay que conocerlas y analizarlas, entre otras plantearemos las siguientes:

Con los sistemas basados en lógica difusa se puede evaluar mayor cantidad de variables, entre otras, variables lingüísticas, no numéricas, simulando el conocimiento humano.

Se relaciona entradas y salidas, sin tener que entender todas las variables, permitiendo que el sistema pueda ser más confiable y estable que uno con un sistema de control convencional.

Es posible obtener prototipos rápidamente, ya que no requiere conocer todas las variables acerca del sistema antes de empezar a trabajar, siendo su desarrollo más económico que el de sistemas convencionales, porque son más fáciles de designar. Se simplifica también la adquisición y representación del conocimiento y unas pocas reglas abarcan gran cantidad de complejidades.

Por todo lo anterior, que por un lado puede ser una ventaja y por otro un posible riesgo, los sistemas en lógica difusa requieren más simulación y una excelente depuración y pruebas antes de pasar a ser operacionales.

Podríamos resumir que la utilización de la lógica difusa es aconsejable para procesos muy complejos, es decir cuando se carece de un modelo matemático simple o para procesos altamente no lineales.

Un sistema de lógica difusa se basa más en la intuición del diseñador que en la precisión y rigor matemático. Debido a esto, los controladores con lógica difusa pueden ser más sencillos, económicos y flexibles que los tradicionales.

2.1.2. Antecedentes internacionales

Reyes Macías (2013) realizó una tesis titulada: *"Diseño conceptual de un Sistema Experto Informático, como herramienta de apoyo en el proceso de elaboración de nuevas leyes, procedimientos, normas y reglamentos en el Ecuador",* en la Universidad Andina Simón Bolívar, Sede Ecuador.

Resumen: La presente tesis se compone de 5 capítulos, en cada uno de ellos se realizó la correspondiente investigación, estudio y análisis.

Es importante mencionar que para el desarrollo de la misma se hizo uso de la metodología de recopilación, análisis y aplicación de la información más relevante concerniente al tema investigado.

En el primer capítulo se realiza el planteamiento del problema, objetivo general y objetivos específicos, la hipótesis, justificación y alcance de la tesis.

El segundo capítulo comprende el marco conceptual, que contempla algunas definiciones relativas a campos como el Desarrollo, la Transparencia, Gobierno Electrónico y TIC's, sin profundizar en las conceptualizaciones teóricas de dichos temas, sino abordándolos desde el punto de vista del desarrollo.

En el tercer capítulo se realiza un estudio sobre los sistemas de información, analizando los conceptos generales, fundamentos, componentes, ciclos de vida y aplicaciones en el gobierno electrónico.

En el cuarto capítulo se estudian los sistemas expertos, se analizan sus características generales, los tipos existentes y los usos actuales.

En el capítulo quinto se desarrolla la propuesta del Diseño conceptual de un Sistema Experto como herramienta de apoyo para la elaboración de nuevas leyes en el Ecuador.

Finalmente se establecen las principales conclusiones y recomendaciones del trabajo.

Conclusiones: La propuesta presentada, abre una serie de usos directos del sistema planteado, su ámbito de aplicación puede ir desde la elaboración de nuevas normativas en la Asamblea Nacional, pasando por su empleo en los GAD´s y juntas parroquiales, hasta llegar al uso por parte del resto de la ciudadanía interesada en algún tema en particular, contribuyendo a democratizar el acceso a la información.

El uso correcto de una herramienta de esta naturaleza se constituirá en un soporte para lograr una mayor consistencia interna, eliminar los vacíos, fomentar la transparencia y en algunos casos evitar las contradicciones de la nueva normativa

con la existente, además de disminuir los tiempos de creación de las mismas y por lo tanto apoyando al desarrollo, por el soporte brindado en la elaboración de mejores normativas.

La aplicación debe ser publicada en un sitio Web de acceso público a partir de perfiles de usuarios, de esta manera se estará fomentado la participación ciudadana al brindar una herramienta que facilite entender las diversas interrelaciones existentes entre la normativa vigente, democratizando el acceso a la información y dando impulso al gobierno electrónico.

Cabe recalcar que el éxito de la implementación estará dado por dos características básicas que debe tener el sistema, la más importante es la calidad de la información con la cual esté alimentado el mismo y la facilidad de uso de la herramienta. Si una de ellas no considera lo recomendado en esta tesis, es casi seguro que el SE esté condenado al fracaso.

La implementación del SE planteado, debe realizarlo un grupo de personas conformado por: expertos en el ámbito constitucional, ingenieros del conocimiento, ingenieros de sistemas con experiencia en desarrollo de sistemas expertos y sistemas multicapas, un diseñador gráfico para la parte de las interfaces de usuarios y un grupo de usuarios diverso para la validación del sistema.

Una de las partes que demandará un enorme esfuerzo, es la clasificación e ingreso de los diferentes documentos existentes, que son necesarios para que el sistema pueda realizar un análisis exhaustivo de lo que se desee validar o normar. Existe una forma de disminuir este esfuerzo y es a través de convenios con empresas existentes en el mercado ecuatoriano como Lexis y Fiel para incorporar su base de documentos al sistema, cabe notar que el sistema propuesto puede complementar de forma extraordinaria a estas herramientas, ya que realiza un salto funcional con respecto a ellas, al contar con una base de conocimiento que le permite aprender y dar un apoyo mayor a la persona que utiliza las aplicaciones.

Dada la gran complejidad del SE y los altos costos en los que se debe incurrir para su implementación, el sistema inicial debería ser auspiciado por uno de los organismos del estado dedicados a la elaboración de normativas como la Asamblea Nacional. Para mantener el sistema actualizado es necesario que de forma

periódica se distribuyan las nuevas versiones del mismo, que contengan los nuevos documentos incorporados a la base de conocimiento, de esa manera se aprovecharía todo el esfuerzo de implementar una aplicación de este tipo.

Los Sistemas de Información son herramientas que impulsan la participación ciudadana, principalmente de los segmentos de población que tienen acceso al uso de TIC's y que las emplean como sus herramientas de trabajo diarias. Se puede complementar la Ley Orgánica de Transparencia y Acceso a la Información Pública, si también se comienzan a brindar nuevos servicios adicionales a los establecidos en la misma, democratizando el acceso a la información.

Se recomienda que el sistema posea la capacidad de conectarse a través de una interface a otras aplicaciones que manejen búsquedas por proximidad y contexto como NVivo, de esta manera el sistema tendrá una mayor versatilidad, además de disminuir el tiempo y costo de implementación a la vez que aumenta su versatilidad.

Cabe destacar que esta tesis brinda los lineamientos generales que debe tener el sistema experto planteado, sin llegar a profundizar en los detalles técnicos para la implementación como código de programación y diseño detallado. Debido a que esto va más allá del alcance del proyecto planteado.

Este documento recoge los elementos más relevantes para la implementación de la herramienta propuesta, por lo tanto, se recomienda utilizar en el desarrollo de la aplicación los lineamientos expuestos en la presente tesis. Aquí se establecen de forma sencilla y de fácil comprensión los principales criterios a considerar para el desarrollo de la aplicación.

Hernández Sánchez & Lozada Cortés (2017) realizaron una tesis titulada: *"Aplicación para dispositivos móviles con sistema operativo Android para el aprendizaje de conceptos básicos de lógica de programación",* en la Universidad Distrital Francisco José de Caldas, Argentina.

Resumen: El proyecto "Aplicación para dispositivos móviles con sistema operativo Android para el aprendizaje de conceptos básicos de lógica de programación", es una aplicación enfocada a dispositivos móviles con S.O. Android desarrollada con la principal motivación de ser una herramienta de ayuda para el proceso de

aprendizaje de los conceptos básicos de la lógica de programación tales como: variables, constantes, tipos de datos, operadores de datos, estructuras de control, estructuras repetitivas y funciones.

Tras el levantamiento de la información, se establecieron los mecanismos para permitir al estudiante visualizar, afianzar y medir sus conocimientos en la lógica de programación de una manera didáctica mediante el modelo educativo constructivista y por medio de cuatro módulos: módulo de usuario, módulo de registro, módulo de enseñanza y módulo de evaluación.

Para la realización de este proyecto se utilizó el entorno de desarrollo Android Studio 2.3.3 empleando el lenguaje de programación Java y el metalenguaje de marcas extensibles XML; así mismo, se empleó el motor de base de datos SQLite el cual cumple la función de alojar la información básica del estudiante, los temas o conceptos que se abordan en la aplicación, su progresión en estos temas, las notas de los ejercicios y las evaluaciones realizadas.

Conclusiones: Según las pruebas de integración, y del sistema junto con la aceptación de la aplicación evidenciada en las pruebas de validación de juego, se logra que la aplicación sea una herramienta de apoyo en el aprendizaje de conceptos básico de la lógica de programación para los estudiantes de primeros semestres que cursan carreras afines al desarrollo de software.

Se debe aprovechar el incremento en las nuevas tecnologías para adaptar la enseñanza de la lógica de programación hacia los dispositivos móviles, dado que esto facilita a las personas interesadas a aprender los fundamentos del desarrollo de software.

Aunque no fue fácil, dado al gran trabajo que supone y a la gran complejidad que supone la metodología RUP, la sensación final fue satisfactoria, en el momento en que se eligió y aplicó al desarrollo del proyecto.

La aplicación cuenta con un módulo de puntos, el cual puede ser utilizado como recurso de evaluación por los maestros o por los mismos estudiantes para revisar los avances constantes de estos últimos.

Para el modelamiento de la base de datos fue fundamental realizar el proceso de normalización de ésta, ya que éste permite eliminar la redundancia de datos y las relaciones varios a varios entre las tablas. Como resultado se logró garantizar la integridad de los datos.

Fernández García (2017) realizó una tesis titulada: *"Diseño y desarrollo de una plataforma de soporte inteligente virtual para aprendizaje de inglés"* en la Universidad Pontificia Comillas, España.

Resumen: La inteligencia empresarial es un campo que está en crecimiento en este momento. Se ha explotado en muchas disciplinas, pero en educación todavía quedan muchos avances por hacer. Este proyecto ha servido para desarrollar una plataforma de aprendizaje del inglés, que sirve de soporte a alumnos y profesores tanto dentro como fuera del aula. El resultado es una aplicación web capaz de generar ejercicios adaptados a las carencias del alumno y de presentar su evolución.

Conclusiones: A lo largo de este documento se han ido presentando el diseño, desarrollo y resultados finales de la tesis de máster. Tal y como se ha explicado al principio del documento, este proyecto se ha desarrollado con la idea de proporcionar un soporte para alumnos y profesores.

Tanto el objetivo principal como los secundarios, presentados en el Capítulo 4. se han cumplido. Se ha desarrollado una plataforma web capaz de generar ejercicios de manera aleatoria y adaptada a cada alumno. Esta plataforma realiza un análisis de las respuestas dadas por el alumno para obtener información sobre los errores que se han cometido. Al conocer los fallos que un alumno ha tenido, se le puede ofrecer información para que estudie los conceptos se han detectado como no entendidos y además se le pueden mostrar las preguntas que ha acertado y cuáles de ellas ha fallado.

Uno de los puntos más interesantes del sistema, es la posibilidad de seguir la evolución del alumno y compararla a la media de la clase. De esta manera se puede saber si el alumno va con retraso respecto al resto de los alumnos y se puede tratar antes de que el problema vaya a más. Además, pasando el ratón por encima de cada punto de la gráfica se puede obtener información sobre que fallos ha cometido

concretamente el alumno en cada uno de los intentos que ha realizado. Se ha añadido una funcionalidad que consiste en mostrar al alumno frases que le motiven y le ayuden a seguir practicando y aprendiendo.

También se ha creado una interfaz para que los profesores puedan acceder a la aplicación. De esta manera podrán observar la evolución de cada uno de los alumnos de su clase y tomar las medidas que ellos consideren necesarias.

Corral Díaz & Carrillo Medina (2015) realizaron una tesis de Máster titulada: *"Creación de un sistema experto para asistir al ingeniero en software en la elaboración de documentos de requerimientos"* en la Universidad de las Fuerzas Armadas, Ecuador.

Resumen: La complejidad de los proyectos de software, así como el carácter multidisciplinar de ingeniería de requisitos obliga a los desarrolladores a seleccionar cuidadosamente las técnicas y prácticas de especificación de requerimientos durante el proceso de desarrollo de software. Uno de los mayores problemas en la especificación de requerimientos es describir de un lenguaje natural a una especificación, en la que el requerimiento tenga aplicación en el dominio, esta descripción se ve comúnmente afectado por la mala comprensión del dominio de la aplicación, el mayor trabajo para el ingeniero de software es explorar sobre ámbitos desconocidos y trabajar sobre un lenguaje en el que, tanto el usuario como el ingeniero, mantienen un grado de abstracción diferente. En este trabajo nos hemos centrado en los problemas de la especificación de requerimientos asistidos y/o apoyados por un sistema experto: indicando lo que se debe especificar. Una de las técnicas utilizadas para la especificación de requerimientos es determinar los tipos de marcos de problemas con la finalidad de obtener la tipificación de requerimientos. Hemos implementado un sistema experto el cual permite obtener un documento de especificación de requerimientos basado en el estándar IEEE 830, documento entendible y legible para las fases iniciales del desarrollo de un producto software.

Conclusiones: Uno de los principales problemas en la construcción de un producto de software es entender y analizar las necesidades del usuario. Para poder especificar los requerimientos de un sistema es necesario hacer una aproximación

al análisis del problema, en esta tesis se planteó realizar este análisis mediante la aplicación de Marcos de Problemas, sugeridos por Michael Jackson que es una técnica que permite mantener la integridad de los datos, a través del análisis de dominios, tipificación de requerimientos.

En esta tesis se desarrolló un sistema experto que de soporte al ingeniero de conocimiento para la especificación de requerimientos basados en el Estándar IEEE 830 y Marcos de Problemas, utilizando la metodología ideal para la construcción del sistema. De la evaluación del sistema podemos indicar que éste es un soporte para la generación de un documento legible y útil para el análisis y diseño de proyectos de software que puede ser utilizado como un referente de negociación.

El sistema experto permite inferir requerimientos funcionales utilizando la técnica interior-exterior propuesta por Michael Jackson, en la que los requerimientos son identificados por los expertos para deducir a un tipo de marco de problema, permitiendo al sistema dar recomendaciones sobre requerimientos funcionales indicando el porqué de las mismas, lo que asegura la confiabilidad de los requisitos.

En lo que respecta a los requerimientos no funcionales fueron inferidos utilizando características propias de todos los sistemas software.

De las evaluaciones y de los resultados obtenidos se puede concluir que éstos fueron satisfactorios para el Ingeniero de Conocimiento y el Experto, siendo un sistema funcional.

La herramienta desarrollada genera el documento de especificación de requisitos de software basada en el estándar IEEE 830, artefacto que resulta de gran utilidad y apoyo para el análisis y diseño de proyectos de software, el cual ha sido utilizado como referente de negociación para el desarrollo de un producto de software.

Iñiguez Loján & Luna Mejía (2014) realizaron una tesis de grado titulada: *"Desarrollo e implementación de un sistema experto para la evaluación del aprendizaje receptivo del idioma inglés: caso: estudiantes de la academia particular Fine Tuned English (Zamora) de edades entre 3 a 18 años"* en la Universidad Nacional de Loja, Ecuador.

Resumen: Para el presente trabajo investigativo se ha analizado a fondo el proceso de evaluación que sigue la academia Fine Tuned English Zamora, como: crear, corregir, calificar una evaluación según un formato establecido, a este proceso se lo ha automatizado con la finalidad de agilizarlo y llevar un mejor control de la información generada por cada interacción con los usuarios. Para llegar a esta automatización se ha desarrollado un software denominado MACOOL. Este software está desarrollado con la metodología Iconix, fundamentada en 4 fases: análisis de requerimientos, análisis y diseño preliminar, diseño e implantación y pruebas.

MACOOL es un sistema web desarrollado en Ruby onRails, utilizando como lenguaje de programación RUBY, MySql como repositorio de datos, Sublime Text como editor de Código, pudiendo ser utilizado en cualquier explorador WEB.

MACOOL registra y maneja todo el proceso de gestión de los usuarios del sistema: Administrador, Docente, Estudiantes; los cuales pueden utilizar las funciones según su rol.

Se han elaborado 2 manuales que explican paso a paso como se realizan cada una de las tareas en el sistema: manual de usuario y programador.

Conclusiones: La ingeniería de requerimientos permitió determinar las necesidades del usuario, constituyéndose en la base fundamental para el proceso de desarrollo del Sistema, cumpliendo con las expectativas y ofreciendo un producto de calidad.

La utilización de formatos de evaluación establecidos en la Academia permite a los docentes y director mejorar la administración en la creación de evaluaciones.

La metodología Iconix, es la más apropiada y la que mejor se acopla al desarrollo del Sistema Web, ya que con las fases de Análisis, Diseño, Implementación, Pruebas y Mantenimiento permiten obtener una mejor funcionalidad en poco tiempo y de manera sólida.

El plan de validación de los usuarios permitió detectar errores durante el desarrollo del Sistema, brindando seguridad y reduciendo a mayor contraste los tiempos e informes físicos que se maneja en el centro educativo.

Cabrera Espinoza (2016) realizó una tesis de grado titulada: *“Desarrollo de un software educativo para la enseñanza aprendizaje del lenguaje de programación JAVA para los alumnos del segundo año de bachillerato figura profesional aplicaciones informáticas del colegio de bachillerato "Marta Bucaram de Roldós" del Cantón Yantzaza, provincia de Zamora Chinchipe periodo lectivo 2014-2015”* en la Universidad Nacional de Loja, Ecuador.

Resumen: La presente investigación se deduce al observar que existe una dificultad de aprendizaje en el lenguaje de programación JAVA, por tal motivo se establece cómo proyecto de tesis: “DESARROLLO DE UN SOFTWARE EDUCATIVO PARA LA ENSEÑANZA-APRENDIZAJE DEL LENGUAJE DE PROGRAMACIÓN JAVA PARA LOS ALUMNOS DEL SEGUNDO AÑO DE BACHILLERATO FIGURA PROFESIONAL APLICACIONES INFORMÁTICAS DEL COLEGIO DE BACHILLERATO “MARTA BUCARAM DE ROLDÓS” DEL CANTÓN YANTZAZA, PROVINCIA DE ZAMORA CHINCHIPE PERIODO LECTIVO 2014-2015”.

Al ver la necesidad de docentes, para comenzar a programar en JAVA con alumnos que no poseen los conocimientos suficientes en la programación en el lenguaje de programación JAVA, le resulta muy difícil dictar sus clases, si la mayoría de los alumnos no logran captar con claridad las enseñanzas dictadas por el docente, por tal motivo el docente no logra avanzar con los temas establecidos.

Al observar eso se propone como objetivo general realizar un software educativo que servirá de herramienta para la adaptación a la programación en el lenguaje de programación JAVA, para los estudiantes del segundo año de bachillerato figura profesional aplicaciones informáticas del colegio de bachillerato “Marta Bucaram de Roldós” del Cantón Yantzaza de la provincia de Zamora Chinchipe.

Los métodos utilizados son el método aplicado, científico, los cuales nos sirvieron para analizar los conocimientos sobre el JAVA, que poseen los estudiantes del segundo año de bachillerato figura profesional aplicaciones informáticas del colegio de bachillerato “Marta Bucaram de Roldós” del cantón Yantzaza de la provincia de Zamora Chinchipe y analizar el contenido que será agregado al software educativo.

De los resultados de las encuestas aplicadas a los estudiantes y docentes se puede deducir que un 75% de los docentes indican que los estudiantes poseen dificultad para programar en JAVA. Mientras que el 78,26% de los estudiantes afirman que tienen dificultad para programar en JAVA, por cuanto se puede deducir la incidencia de la problemática expuesta en el proyecto de tesis planeado.

En los resultados de las encuestas aplicadas para la validación del software se puede deducir que los estudiantes se encuentran conformes con el trabajo realizado exclusivamente para ayudarles a mejorar el proceso de aprendizaje del lenguaje de programación JAVA.

Conclusiones: Al culminar la investigación de la problemática y el desarrollo del software educativo para dar solución a ese inconveniente se concluye:

1. El proceso de observación y recolección de datos que se realizó a los alumnos del segundo año de bachillerato figura profesional aplicaciones informáticas del Colegio de Bachillerato “Marta Bucaram de Roldós”, definió lo difícil que es programar por primera vez en un lenguaje de programación.

2. El análisis de la planificación curricular propuesta por el docente de los estudiantes del segundo año de bachillerato aplicaciones informáticas del Colegio de Bachillerato “Marta Bucaram de Roldós”, de la materia desarrollo de aplicaciones, permitió constatar que no estaba avanzando correctamente con el plan establecido.

3. Se inició el diseño de imágenes en Adobe Photoshop Cs6 y Adobe Illustrator Cs6, videos en Camtasia, sonido en Adobe Audition Cs6 y animaciones en Adobe Flash Cs6, tratando de adaptar las especificaciones establecidas por docentes y alumnos, con el objetivo de darle el diseño más llamativo al software educativo.

4. Se empezó a desarrollar el software en Adobe Flash Cs6 importando lo creado en los otros programas, manteniendo el nivel y calidad académica que debe poseer.
5. Validación y socialización del software con los estudiantes del segundo año de bachillerato figura profesional aplicaciones informáticas del Colegio de Bachillerato “Marta Bucaram de Roldós”. Dando por culminada su creación.

2.2. Bases teóricas de las variables

2.2.1. Bases teóricas de la variable independiente: Sistema Experto

2.2.1.1. Definición de Sistema Experto

Introducción: Inteligencia artificial

Para comprender que es un sistema experto y todos los aspectos que giran alrededor de este concepto es necesario entender que es la inteligencia artificial.

Según McCarthy (2007) profesor emérito de la Universidad de Stanford y que acuñó el término inteligencia artificial en la Conferencia de Dartmouth en el año 1956, define la IA como "la ciencia y la ingeniería de crear máquinas inteligentes, especialmente programas de computación inteligentes. Está relacionada con la tarea similar de utilizar ordenadores para comprender la inteligencia humana, pero la IA no se limita a métodos que sean observables biológicamente".

Esta definición abarca cuatro grandes enfoques dentro de la IA, dos de ellos centrados en los humanos y los otros centrados en la racionalidad según se puede apreciar en la tabla 1:

Tabla 1

Algunas definiciones de IA, organizadas en cuatro categorías

Sistemas que piensan como humanos	Sistemas que piensan racionalmente
«El nuevo y excitante esfuerzo de hacer que los computadores piensen… máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985) «[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje…» (Bellman, 1978)	«El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985) «El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)
Sistemas que actúan como humanos	**Sistemas que actúan racionalmente**
«El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990) «El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)	«La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole *et al.*, 1998) «IA… está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)

Fuente: Russell & Norvig (2004)

Russell & Norvig (2004) enfatizan su preferencia por el enfoque basado en el diseño de agentes racionales porque:

> Ofrece al menos dos ventajas. La primera es más general que el enfoque que proporcionan las «leyes del pensamiento», dado que el efectuar inferencias correctas es sólo uno de los mecanismos existentes para garantizar la racionalidad. La segunda es más afín a la forma en la que se ha producido el avance científico que los enfoques basados en la conducta o pensamiento humano, porque la norma de la racionalidad está claramente definida y es de aplicación general.

Otro punto importante es la diferencia entre la IA fuerte y la IA débil, lo cual ha sido y es materia de muchos debates. Los defensores de la IA fuerte sostienen que una máquina que emula inteligencia puede llegar a desarrollar conciencia y a pensar, esto porque la conciencia y el pensamiento (tomando como ejemplo al ser humano) no son más que una manifestación del cerebro, el cual opera bajo un conjunto de reglas. Estos pensadores consideran que cualquier manifestación espiritual, sentimental o emocional se reduce al sistema nervioso, por lo tanto, no existe sujeto al que analizar, solo existe una máquina que opera bajo ciertas leyes físicas, químicas y biológicas. (Urbina Padilla, 2013)

Por su parte los defensores de la IA débil sostienen que, si existe un sujeto que analizar porque existe un espíritu en los humanos, cuya manifestación y operacionalización se da a través del pensamiento, el cual podemos percibir midiendo la actividad encefálica. Luego, una máquina que opera bajo los principios de inteligencia artificial no puede pensar, ni ser consciente, solo sigue un conjunto de reglas o fórmulas matemáticas de manera automatizada. (Urbina Padilla, 2013)

El Sistema Experto

Gutiérrez Lozano & Hernández Valderrama (2015)**,** afirman que un sistema experto “simula la habilidad de un experto humano a la hora de resolver un determinado tipo de problema, mediante la aplicación específica de conocimientos y de procedimientos de inferencia, ya que no se cuenta con una solución algorítmica práctica”.

Entonces, un sistema experto es aquel sistema informático que permite simular el razonamiento de un experto humano en una determinada área del conocimiento. A continuación, se detallan los componentes de un sistema experto:

2.2.1.2. Componentes de un sistema experto

Los sistemas expertos utilizan un conjunto de proposiciones en una base de hechos y un conjunto de reglas especificadas en una base de conocimientos; además utilizan un motor de inferencia que selecciona las reglas necesarias para cumplir con un objetivo, por ejemplo, responder a las consultas de un usuario.

Base de hechos

Según Gutiérrez Lozano & Hernández Valderrama (2015) , la base de hechos "contiene los datos del problema, así como los elementos y hechos relativos a la solución de un problema en particular. A su vez almacena la información dada por el usuario en respuesta a las preguntas del sistema."

En otras palabras, la base de hechos es aquel repositorio que contiene el conjunto de sucesos y afirmaciones verificables acerca de una rama de la ciencia.

Base de conocimientos

Según postula (Feigenbaum, 1979) la base de conocimientos de un sistema experto se define como:

> Un conjunto de operadores que se utilizan para manipular la base de datos. Algunos ejemplos de operadores incluyen:

- En la demostración de teoremas, reglas de inferencia tales como modus ponens y resolución;
- En ajedrez, reglas para mover piezas de ajedrez;

- En Integración simbólica, reglas para simplificar los formularios a integrar, como la integración por partes o la sustitución trigonométrica.

Asimismo, Vásquez Ruiz (2013) define la base de conocimientos como:

> Una base de datos que almacena todo el conocimiento del sistema experto en forma de reglas. Este conocimiento comprende los datos que describen el problema, las reglas utilizadas, la forma de combinar estas reglas, los nuevos datos deducidos y las propuestas de solución. Se caracteriza porque los conocimientos son descritos de manera declarativa, almacenados en pequeños fragmentos y no existe jerarquía entre los mismos. En la creación de una base de conocimientos se debe de tener en cuenta qué objetos serán definidos, cómo son las relaciones entre estos objetos, cómo se formularán y procesarán las reglas.

Motor de inferencia

Badaró, Javier, & Jorge Agüero (2013) definen el motor de inferencia:

> Un programa de computadora que provee metodologías para razonamiento de información en la base de conocimiento. Este componente provee direcciones sobre cómo usar el conocimiento del sistema para armar la agenda que organiza y controla los pasos para resolver el problema cuando se realiza una consulta. Tiene tres elementos principales: (1) Intérprete, ejecuta la agenda seleccionada; (2) programador, mantiene el control sobre la agenda; (3) control de consistencia, intenta mantener una representación consistente de las soluciones encontradas.

El motor de inferencia es la parte más importante de un sistema experto, porque permite la toma de decisiones y la elaboración de conclusiones en base a razonamientos; para ello utiliza un sistema de control, el cual puede ser de dos tipos: sistema de control sistémico o sistema de control heurístico.

Dentro del control sistémico se encuentra el encadenamiento hacia adelante y hacia atrás; y también el control irrevocable o por tentativas. Para el ámbito de esta tesis

nos interesa analizar el encadenamiento hacia adelante y el encadenamiento hacia atrás.

Con respecto al encadenamiento hacia adelante Feigenbaum (1979) sostiene que: “El objetivo es hacer avanzar la situación, o el estado del problema, desde su configuración inicial a una que satisfaga una condición de objetivo.” Es decir que el motor de inferencia parte de hechos y reglas que se encuentran en la base de datos para elaborar conclusiones:

En cuanto al encadenamiento regresivo, el motor de inferencia sigue el procedimiento inverso, parte de una hipótesis u objetivo a probar y se buscan hechos y reglas que actúen como evidencia. Feigenbaum (1979) afirma:

> Una estrategia alternativa, el razonamiento hacia atrás, implica el uso de otro tipo de operador, que se aplica no a una situación de dominio de tarea actual sino a la meta. La declaración de objetivo, o la declaración del problema, se convierte en uno o más objetivos secundarios que son (uno espera) más fáciles de resolver y cuyas soluciones son suficientes para resolver el problema original. Estos subobjetivos pueden a su vez reducirse a sub-objetivos secundarios, y así sucesivamente, hasta que cada uno de ellos sea aceptado como un problema trivial o su solución se logre mediante la solución de su sub-problema.

2.2.1.3. Procesamiento del Lenguaje Natural (PLN)

Como afirma Dale R. citado por Martínez Cámara (2015) el PLN es una “disciplina centrada en el diseño e implementación de aplicaciones informáticas que se comunican con personas mediante uso de lenguaje natural.”

A partir de lo citado, mencionaré que el PLN es una rama de la inteligencia artificial dedicada a lograr la comunicación entre humanos y máquinas que emulan inteligencia. Los sistemas de PLN identifican la estructura de las oraciones, separan las palabras clave, analizan su semántica y su sintaxis, luego convierten las

estructuras naturales en estructuras formales, las procesan, las vuelven a convertir a lenguaje natural y las muestran a los usuarios correspondientes.

Abarca básicamente tres campos, la recuperación de la información, la traducción automática y los sistemas de diálogo. La recuperación de la información implica que el sistema lee un conjunto de datos no estructurados, busca las entidades correspondientes, las agrupa mediante relaciones para clasificarlas, organizarlas en función a un propósito y luego presentarlas a los usuarios interesados; esto se aplica cuando se quiere realizar el análisis de sentimientos y opiniones de los tweets de una cierta cantidad de personas de alguna localidad con respecto a algún representante político o a un producto o servicio.

En la traducción automática, por ejemplo, si se desea traducir de árabe a español, el sistema de PLN puede trabajar de la siguiente manera: traducir primero internamente de árabe a inglés, y luego de inglés al español, y mostrar la traducción final al usuario.

Para efectos de este proyecto, me centraré en el campo de los sistemas de dialogo. El usuario escribe una pregunta o consulta y el sistema busca una palabra clave, si la detecta entonces responde de acuerdo con la palabra clave seleccionada, en caso no detecte ninguna palabra clave, el sistema responde indicando que no entiende la consulta.

2.2.1.4. Ventajas y desventajas de los sistemas expertos

Entre las principales ventajas tenemos la que menciona Santacreu Ríos (2014)**;** el afirma que un sistema experto puede:

> Superar las capacidades humanas, por ejemplo, cuando *se requiere analizar un gran volumen de datos* en un corto espacio de tiempo, como ocurre en los Servicios de Seguridad y Emergencias. El experto humano se ve obligado a despreciar información no considerada importante, en cambio el Sistema Experto tiene mayor capacidad para el análisis sin que aparezca el cansancio u otros efectos propios de la condición humana que empeoran los resultados

Entonces un sistema experto puede trabajar horas, días o hasta meses seguidos sin sufrir agotamiento o cansancio.

También se recomienda usarlos cuando los expertos humanos en una determinada área del conocimiento son escasos, en situaciones difíciles donde la subjetividad humana genera riesgos, por ejemplo, tomar malas decisiones o elaborar conclusiones incorrectas. Otra ventaja es que, a comparación de un sistema clásico, un sistema experto puede funcionar con una pequeña cantidad de reglas (no es necesario mayor cantidad de código fuente).

Entre las desventajas, se tiene que la recursividad es muy baja, por lo que, si se quiere hacer un sistema experto para otro objetivo diferente, hay que empezar de nuevo.

2.2.1.5. Microsoft Bot Framework

Es un marco de trabajo que permite el desarrollo de bots, por ejemplo, aplicaciones que implementan visión artificial para procesamiento de imágenes, procesamiento de lenguaje natural (chatbot de texto y de voz), traducción automática, etc.

En este framework se puede programar utilizando C#, Node.js, o cualquier lenguaje de programación mediante la utilización de las APIs REST que proporciona el Framework. Para la fase de programación mucho se recomienda el uso de Virtual Studio en su versión gratuita (Visual Studio Community Manager o bien Visual Studio Code). Para hacer las pruebas correspondientes se recomienda el botframework-emulator y para subir la aplicación a la nube el portal Azure Services.

El Bot framework permite la implementación de aplicaciones en plataformas como Skype, Facebook, Slack, Telegram, o en una web particular (gracias al uso de iframes); cuando se implementa en una web cualquiera es posible la modificación de los estilos mediante CSS o BootStrap.

Virtual Studio permite ejecutar la aplicación de manera local y también cuenta con el modo de depuración, el cual se utiliza cuando se quiere observar como interactúan los componentes internos del sistema para detectar errores de programación en tiempo de ejecución.

Instalación de Microsoft Bot Framework

Para instalar el framework primero hay que tener instalado Virtual Studio 2017 (también se puede utilizar la versión 2015) el cual se puede descargar desde la página oficial. Luego es necesario descargar la plantilla BotApplication, lo cual se puede hacer desde el siguiente enlace: http://aka.ms/bf-bc-vstemplate

Dicha plantilla debe ser ubicada en la carpeta "Visual C#" la cual (salvo se haya configurado una dirección distinta en la instalación) se encuentra en:

C:\Users\{NombreDeUsuario}\Documents\VisualStudio2017\Templates\ProjectTemplates\Visual C#

A continuación, iniciamos el Visual Studio, hacemos click en File>New>Project y se despliega la ventana mostrada en la figura 2.

El nombre por defecto es "Bot Aplication1", y se puede cambiar según se considere (por ejemplo "Prueba1", "WebBot1", etc.); luego hacemos click en OK y ya tenemos el entorno listo para trabajar.

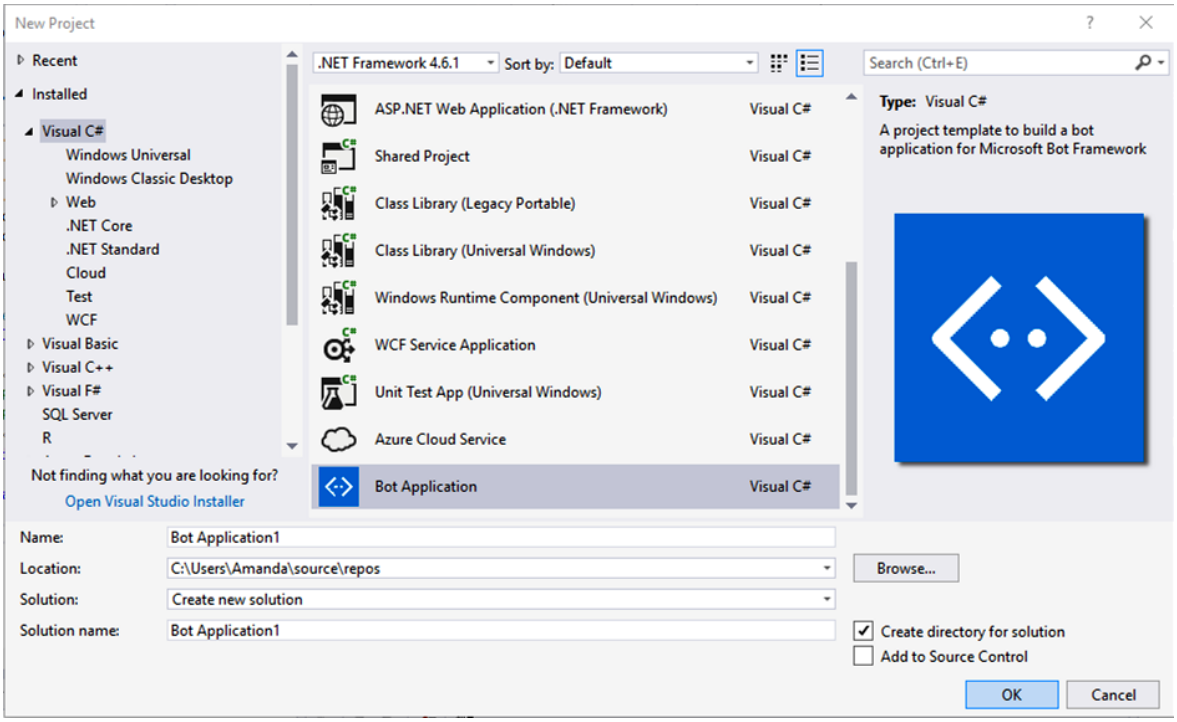

Figura 2: Ventana de Nuevo Proyecto de Microsoft Bot Framework

Fuente: Elaboración Propia

Definición de Chatbot

Es un sistema conversacional que permite simular la comunicación con un ser humano. Existen múltiples herramientas que permiten el desarrollo de este tipo de sistemas, por ejemplo: Microsoft Bot Framework, Watson IBM, Api.ai, Motion.ai, Smooch.io, PandoraBots, ClaudiaBot, y muchos más. La lista de herramientas y de chatbots listos para ser usados por el público crece de manera exponencial.

Como afirma Hosseini (2017) “Dado que los ChatBots utilizan de manera inherente los mismos protocolos de comunicación que las aplicaciones de mensajería instantánea, se deben categorizar en Mensajería instantánea.” Por ejemplo, una aplicación de mensajería instantánea debe ser capaz de enviar y recibir mensajes de manera asíncrona, es decir, sin necesidad de recargar toda la página de forma que un usuario pueda enviar un mensaje mientras el chatbot está enviando una respuesta y viceversa, sin tener que volver a cargar todo el historial de mensajes anterior.

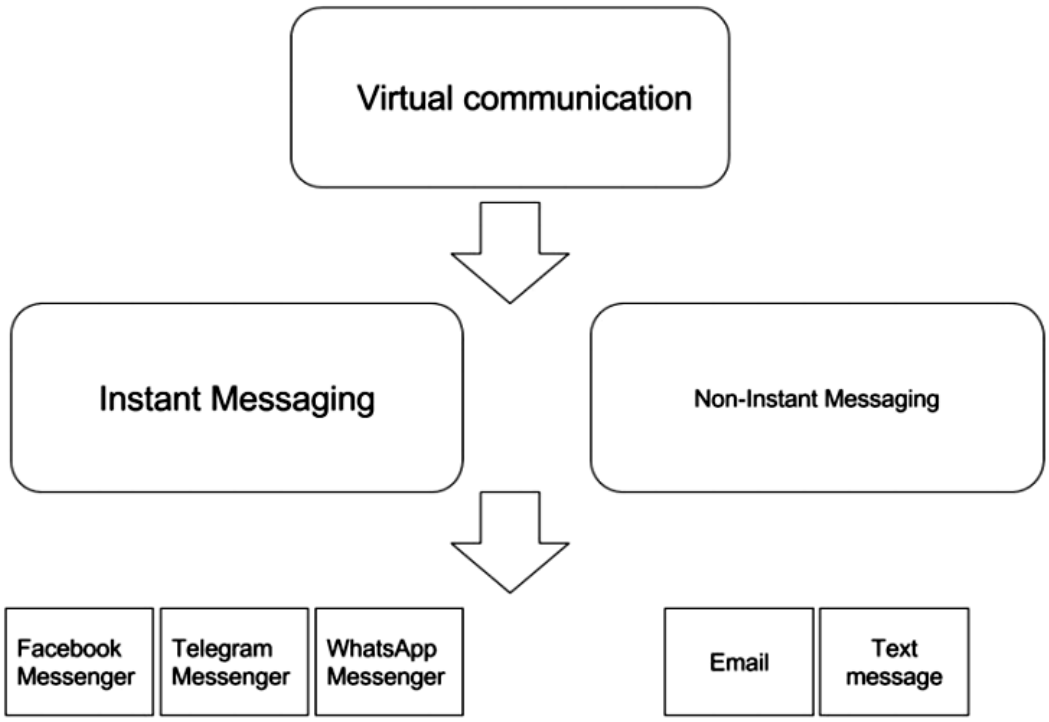

Figura 3: Tipos de comunicación virtual

Fuente: Hosseini *(2017)*

Tipos de Chat-Bot

Existen múltiples maneras de clasificar los chat-bots, en el curso de bots de TI Capacitación se proponen 3 categorías: Utilidades, Sociables y Asistentes.

- ChatBot de Utilidades: El ChatBot que forma parte de este proyecto es de utilidades puesto que cumple una función específica que consiste en ayudar a los usuarios en la programación con Python, tiene objetivos específicos, por ejemplo, mostrar la sintaxis de un bucle for o while, o indicar como resolver un error de sangría (indentation en inglés) y el éxito de este chatbot será medido tomando como base el buen cumplimiento de sus objetivos. Los chatbots de utilidades también son utilizados en entornos empresariales cuando se quiere conocer cómo funciona una nueva plataforma, los usuarios pueden realizar consultas al chatbot sobre cómo hacer reportes o interactuar con dicha plataforma en algún aspecto. Su nivel de personalidad no es muy alto.
- ChatBot Sociable: Estos chatbots están destinados para la tratar con usuarios externos a una empresa, por ejemplo, clientes que desean conocer las ofertas disponibles o información sobre un nuevo producto. La plataforma CLIENGO, el chatbot de megacursos.com, son ejemplos de chatbots sociables. Poseen un nivel de interacción más alto.

- ChatBot Asistente: Son aquellos que interactúan con el sistema operativo, por ejemplo, Cortana de Microsoft, Siri de Apple, SimSimi de Android. Estos bots son capaces de conversar con el usuario gracias a su alto nivel de personalidad (igual que el ChatBot Sociable), pero además son capaces de ejecutar tareas por sí mismos, por ejemplo, abrir el e-mail, redes sociales, ejecutar un aplicativo del sistema, etc.

Funcionamiento de un Chat-Bot en Microsoft Bot Framework

A continuación, explicaré el código que viene por defecto al crear el proyecto. Para la implementación se utilizó el lenguaje C#, aunque como mencionamos anteriormente también se puede desarrollar en Node.js o en cualquier lenguaje de programación mediante la utilización de APIs REST proporcionadas por el Bot Framework.

En la aplicación de nombre “Prueba1” centraremos nuestra atención en MessagesController.cs ubicado en la carpeta Controllers. Todo empieza cuando el usuario envía un mensaje (o realiza otras actividades como las que se detallan en la tabla 2) el cual es enviado a la clase Messages Controller. En la figura 4 podemos observar el código de esta clase, desde la primera a la sexta línea encontramos las referencias a todas las bibliotecas necesarias, la primera vez que hayamos creado el proyecto será necesario ir a Build>BuildSolution para importarlas y tenerlas listas para ser usadas.

En la línea 8 de la figura 4 se define el nombre del espacio de trabajo que es igual al nombre del proyecto. En la línea 11 encontramos la clase MessagesController la cual posee dos métodos cuyo detalle se pueden observar en la figura 5.

```
Prueba 1                                   Prueba_1.MessagesController
 1  using System.Net;
 2  using System.Net.Http;
 3  using System.Threading.Tasks;
 4  using System.Web.Http;
 5  using Microsoft.Bot.Builder.Dialogs;
 6  using Microsoft.Bot.Connector;
 7
 8  namespace Prueba_1
 9  {
10      [BotAuthentication]
11      public class MessagesController : ApiController
12      {
13          /// <summary>
14          /// POST: api/Messages
15          /// Receive a message from a user and reply to it
16          /// </summary>
17          public async Task<HttpResponseMessage> Post([FromBody]Activity activity)...
30
31          private Activity HandleSystemMessage(Activity message)...
59      }
60  }
```

Figura 4: Estructura del código de MessageController.cs

Fuente: Elaboración Propia

El primer método de Message Controller, ubicado desde la línea 17 hasta la 29, es un método público de tipo asíncrono que recibe un objeto denominado “Activity” mediante el método post. Si este “activity” es el mensaje de un usuario entonces lo procesa, caso contrario lo deriva al segundo método, ubicado desde la línea 31 hasta la 58. Existen muchos tipos de actividades, la tabla 2 muestra una lista de las principales actividades.

Tabla 2

Tipos de Activity en Microsoft Bot Framework

Tipo Activity	Interfaz	Descripción
Message	IMessageActivity	Representa una comunicación entre bot y usuario.
ConversationUpdate	IConversationUpdateActivity	Indica que el bot fue agregado a una conversación, otros miembros fueron agregados o eliminados de la conversación o que los metadatos de la conversación han cambiado.
ContactRelationUpdate	IContactRelationUpdateActivity	Indica que el bot fue agregado o eliminado de la lista de contactos del usuario.
Typing	ITypingActivity	Indica que el usuario o bot en el otro extremo de la conversación está escribiendo una respuesta.
Ping	n/a	Representa un intento de determinar si el endpoint de un bot es accesible.
DeleteUserData	n/a	Indica a un bot que un usuario ha solicitado que el bot elimine cualquier dato del usuario que pueda haber almacenado.
EndOfConversation	IEndOfConversationActivity	Indica el final de una conversación.
Event	IEventActivity	Representa una comunicación enviada a un bot que no es visible para el usuario.
Invoke	IInvokeActivity	Representa una comunicación enviada a un bot para solicitar que realice una operación específica. Este tipo de actividad está reservado para uso interno por el Microsoft Bot Framework.
MessageReaction	IMessageReactionActivity	Indica que un usuario ha reaccionado a una actividad existente. Por ejemplo, un usuario hizo clic en el botón “Me gusta” de un mensaje.

Fuente: Microsoft *(2017)*

```
8   namespace Prueba_1
9   {
10      [BotAuthentication]
11      public class MessagesController : ApiController
12      {
13          /// <summary>
14          /// POST: api/Messages
15          /// Receive a message from a user and reply to it
16          /// </summary>
17          public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
18          {
19              if (activity.Type == ActivityTypes.Message)
20              {
21                  await Conversation.SendAsync(activity, () => new Dialogs.RootDialog());
22              }
23              else
24              {
25                  HandleSystemMessage(activity);
26              }
27              var response = Request.CreateResponse(HttpStatusCode.OK);
28              return response;
29          }
```

Figura 5: Código del método Post dentro de MessageController.cs

Fuente: Elaboración Propia

En la figura 5, analizaremos el método de nombre Post, de tipo Task<HttpResponseMessage>. Este método recibe una actividad que ha sido enviada a través , luego, si el tipo de actividad es un mensaje escrito por un usuario entonces ejecuta el método RootDialog de la clase Dialogs, en el cual se encuentran todos los algoritmos relacionados a las respuestas del robot; si la actividad es de otro tipo entonces realiza una llamada al segundo método de la clase MessagesController, el cual se denomina “HandleSystemMessage”, si no se han presentado errores, el sistema retorna un código de estado Http 200 que significa que todo se ejecutó exitosamente y que la aplicación puede proseguir a la espera del siguiente mensaje del usuario.

El método privado HandleSystemMessage permite implementar todo lo que se debe realizar en actividades que no sean un mensaje, inicialmente el código aparece como se observa en la figura 6.

```
31        private Activity HandleSystemMessage(Activity message)
32        {
33            if (message.Type == ActivityTypes.DeleteUserData)
34            {
35                // Implement user deletion here
36                // If we handle user deletion, return a real message
37            }
38            else if (message.Type == ActivityTypes.ConversationUpdate)
39            {
40                // Handle conversation state changes, like members being added and removed
41                // Use Activity.MembersAdded and Activity.MembersRemoved and Activity.Action for info
42                // Not available in all channels
43            }
44            else if (message.Type == ActivityTypes.ContactRelationUpdate)
45            {
46                // Handle add/remove from contact lists
47                // Activity.From + Activity.Action represent what happened
48            }
49            else if (message.Type == ActivityTypes.Typing)
50            {
51                // Handle knowing tha the user is typing
52            }
53            else if (message.Type == ActivityTypes.Ping)
54            {
55            }
56
57            return null;
58        }
```

Figura 6: HandleSystemMessage dentro de MessageController.cs

Fuente: Elaboración Propia

Por ejemplo, si el usuario solicita eliminar un dato de sí mismo que el sistema haya almacenado (DeleteUserData), como información de contacto que el usuario le haya brindado anteriormente, el sistema le mostrará un mensaje con la información respectiva (la cual puede ser el nombre completo, el celular, el correo, etc.) y un mensaje a modo de confirmación de la acción eliminar.

Para propósitos del sistema de la presente tesis no es necesario implementar el método HandleSystemMessage.

Finalmente, en la figura 7 se observa en la línea 9 la clase RootDialog, esta clase también posee dos métodos, el primero espera a que el usuario envíe un mensaje y luego realiza una llamada al segundo método; el segundo método recibe ese mensaje en un objeto de tipo activity. En la línea 23 se cuenta el número de caracteres de ese mensaje y en la línea 26 el sistema envía una respuesta al usuario, la respuesta es de la siguiente manera: “You sent {activity.text} which was”, finalmente espera una nueva respuesta del usuario y se repite el proceso.

```
1   using System;
2   using System.Threading.Tasks;
3   using Microsoft.Bot.Builder.Dialogs;
4   using Microsoft.Bot.Connector;
5
6   namespace Prueba_1.Dialogs
7   {
8       [Serializable]
9       public class RootDialog : IDialog<object>
10      {
11          public Task StartAsync(IDialogContext context)
12          {
13              context.Wait(MessageReceivedAsync);
14
15              return Task.CompletedTask;
16          }
17
18          private async Task MessageReceivedAsync(IDialogContext context, IAwaitable<object> result)
19          {
20              var activity = await result as Activity;
21
22              // calculate something for us to return
23              int length = (activity.Text ?? string.Empty).Length;
24
25              // return our reply to the user
26              await context.PostAsync($"You sent {activity.Text} which was {length} characters");
27
28              context.Wait(MessageReceivedAsync);
29          }
30      }
31  }
```

Figura 7: Código de la clase RootDialog

Fuente: Elaboración Propia

2.2.1.6. Dimensiones del Sistema Experto

Base de hechos y conocimientos

En los sistemas basados en reglas intervienen dos elementos importantes: la base de conocimiento y los datos. (...) En situaciones deterministas, las relaciones entre un conjunto de objetos pueden ser representadas mediante un conjunto de reglas. El conocimiento se almacena en la base de conocimiento y consiste en un conjunto de objetos y un conjunto de reglas que gobiernan las relaciones entre esos objetos. La información almacenada en la base de conocimiento es de naturaleza permanente y estática, es decir, no cambia de una aplicación a otra, a menos que se incorporen al sistema experto elementos de aprendizaje (Castillo, Gutiérrez, & Hadi, 1998, p. 24).

Motor de inferencia

Hace referencia al rendimiento, la usabilidad, la navegabilidad, la adaptabilidad y la rapidez del sistema para responder a las diversas preguntas que los usuarios le presenten Tal como se ha mencionado en la sección anterior, hay dos tipos de elementos: los datos (hechos o evidencia) y el conocimiento (el conjunto de reglas almacenado en la base de conocimiento). El motor de inferencia usa ambos para obtener nuevas conclusiones o hechos. Por ejemplo, si la premisa de una regla es cierta, entonces la conclusión de la regla debe ser también cierta. Los datos iniciales se incrementan incorporando las nuevas conclusiones. Por ello, tanto los hechos iniciales o datos de partida como las conclusiones derivadas de ellos forman parte de los hechos o datos de que se dispone en un instante dado (Castillo, Gutiérrez, & Hadi, 1998, p. 30).

2.2.2. Bases teóricas de la variable dependiente: Lenguaje de Programación Python 3.x

2.2.2.1. Definición de Python

Podemos considerar como mejor concepto el brindado por el propio creador del lenguaje, Van Rossum (2017), el cual afirma que:

> Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y sus tipos dinámicos, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, https://www.python.org/, y puede distribuirse libremente.

2.2.2.2. Características del lenguaje Python 3.x

A continuación, se listan las principales características que han permitido que este lenguaje sea hoy en día bastante utilizado (una buena evidencia de esto es que el portal StackOverFlow tiene más de 962,000 preguntas relacionadas a Python a la fecha de redacción de esta tesis, mientras que c#, php, javascript y java tienen entre 1, 200,000 y 1, 625,000 preguntas).

A continuación, describiremos brevemente las principales características de este lenguaje:

Facilidad de aprendizaje

La sintaxis de Python es muy sencilla y puesto que usa sangría (y no llaves) para separar los bloques de código, permite a los programadores entender el código rápidamente, dado que no sucederá como en otros lenguajes que en una sola línea hay múltiples instrucciones que deberían estar colocadas en varias líneas.

Orientado a programación funcional

Python tiene soporte para realizar programación funcional o estructural al estilo de lenguajes como LISP. Por ejemplo, en Python se pueden hacer funciones de tratamiento de listas (por ejemplo, append, insert, remove, count, etc.) o se puede trabajar con listas de comprensión (que son usadas para mejorar la rapidez de una aplicación).

A continuación, un ejemplo tomado del manual de documentación de Python, en este ejemplo se muestra los valores de PI, al principio con un decimal, luego dos, y así hasta mostrar 6 decimales, todo esto en una sola línea.

```
>>> from math import pi
>>> [str(round(pi, i)) for i in range(1, 6)]
['3.1', '3.14', '3.142', '3.1416', '3.14159']
```

Figura 8: Ejemplo con tratamiento de listas

Fuente: Van Rossum (2017)

Orientado a objetos:

Python tiene soporte para trabajar con clases y objetos. Las clases se definen con la palabra reservada “class” seguida del nombre de dicha clase. Los métodos se definen con la palabra reservada “def”. Detallaremos este punto más adelante.

Múltiples librerías:

Como afirma Coutinho Menezes (2016) “Existen múltiples librerías que pueden ser utilizadas para diversos propósitos como son: Minería de datos, machine learning, deep learning, computación cuántica, procesamiento científico de datos, edición de código python, web-scraping, animación 3d, etc.” También tenemos las librerías Scipy, Numpy, Scikit-learn, Pandas, qiskit, beautiful-soup, jupyter, entre otras.

2.2.2.3. Filosofía de Python

También conocido como el Zen de Python gracias a Tim Peters, se constituye como una referencia y guía sobre la forma en que se debe programar en lenguaje Python.

- Bello es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- La legibilidad cuenta.
- Los casos especiales no son tan especiales como para quebrantar las reglas.
- Lo práctico gana a lo puro.
- Los errores nunca deberían dejarse pasar silenciosamente.
- A menos que hayan sido silenciados explícitamente.
- Frente a la ambigüedad, rechaza la tentación de adivinar.
- Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.

- Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.
- Ahora es mejor que nunca.
- Aunque nunca es a menudo mejor que ya mismo.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, puede que sea una buena idea.
- Los espacios de nombres (namespaces) son una gran idea ¡Hagamos más de esas cosas!

2.2.2.4. Elementos y estructuras básicas del lenguaje Python

En python encontramos elementos básicos como son las variables, comentarios, tipos de datos (cadenas, números y boleanos); en cuanto a las estructuras tenemos las condicionales, los bucles, las funciones y la programación orientada a objetos.

Variables

Una variable es aquel elemento de programación que guarda un valor que puede cambiar durante la ejecución de un programa. Para declarar un variable en python se deben seguir ciertas reglas, las cuales son:

- Las variables deben empezar con una letra, nunca con números y caracteres especiales.
- Las variables no deben tener espacios en blanco, pero pueden tener guiones intermedios y guiones bajos.
- Las variables que empiezan y terminan con doble guión están reservadas como variables especiales de python.

```
"""Datos numéricos"""
var1 = 340

"""Datos decimales"""
var_3_ = 76.2

"""Cadena de texto (string)"""
var_2 = "Soy un string"
```

Figura 9: Ejemplo de variables en Python

Fuente: Elaboración Propia

El tipo de variable depende del tipo de dato que está guardando, un tipo de dato numérico hará que la variable sea numérica, por ejemplo. Una de las ventajas de Python es su tipado dinámico lo cual permite que no sea necesario declarar el tipo de una variable, solamente asignarle un valor. En el apartado de condicionales analizaremos las variables con valores booleanos.

Comentarios

En Python es posible hacer comentarios de una sola línea y comentarios multilínea, en la siguiente imagen se explica la manera:

```
1  #Este es un comentario de una sola línea
2
3  """
4  Este es un comentario de múltiples líneas
5  Línea 1 del comentario
6  Línea 2 del comentario
7  Línea 3 del comentario
8
9
10 Y mucho más
11 """
```

Figura 10: Comentarios en Python

Fuente: Elaboración Propia

Condicionales

En python para las estructuras condicionales se tiene el if-else, y no hay soporte para switch. En la figura 11, la expresión ‘’n == 1’’ de evaluarse como verdadero devolverá un valor boleano “True” y entonces se ejecutará el bloque dentro de if, caso contrario pasará al “elif” y luego al “else”.

```
n = 2
if (n == 1):
    print("Hola")
elif (n == 2):
    print("Hola x2")
else:
    print("Adios")
```

```
Hola x2
```

Figura 11: Ejemplo de IF-ELSE en Python

Fuente: Elaboración Propia

Bucles FOR y WHILE

Para declarar un bucle for se utiliza la función range, al ser python un lenguaje que para separar el código no utiliza llaves sino solo sangría todo lo que este sangrado hacia la izquierda de la palabra reservada for será ejecutado por este bucle.

```
for i in range (3):
    print("Número: {}".format(i))
```

```
Número: 0
Número: 1
Número: 2
```

Figura 12: Ejemplo de Bucle FOR

Fuente: Elaboración Propia

Con respecto al bucle while la sintaxis no varía mucho con respecto a otros lenguajes, simplemente no se utilizan llaves, las sentencias no se cierran con puntos y comas y después de colocar la condición del while se colocan dos puntos, para indicar que todo lo que viene debajo es parte del código que será ejecutado por ese bucle.

```
a = 1
while(a < 5):
    print ("Iteración Nº",format(a))
    a = a + 1
```

```
Iteración Nº 1
Iteración Nº 2
Iteración Nº 3
Iteración Nº 4
```

Figura 13: Ejemplo de Bucle While

Fuente: Elaboración Propia

Programación orientada a objetos

Para declarar una clase se usa la palabra reservada “class”. Una clase consta de atributos y métodos. Para llamar a un método se utiliza la palabra reservada “def”. En este punto sería importante mencionar que, si por el momento no se requiere implementar una clase, se puede utilizar la palabra reservada “pass”.

```
class Perro:
    #ATRIBUTOS
    nombre = "Fido"
    color="marron"

    #MÉTODOS
    def ladrar(self):
        return "Guau guau!!"

#Instanciar la clase
perro1 = Perro()
```

Figura 14: Ejemplo de una clase y su correspondiente instancia.

Fuente: Elaboración Propia

Para llamar a los atributos simplemente se coloca el nombre del objeto, luego un punto y finalmente el nombre del atributo; se procede de manera muy parecida para la llamada a los métodos con la diferencia de que se agregan paréntesis. Luego se puede imprimir en pantalla con la función print y darle un formato con la función ‘’format”.

```
#Llamar a los atributos
nom = perro1.nombre
col = perro1.color

#Llamar al método
lad = perro1.ladrar()

#Imprimir
print("{} es {}. {}"
      .format(nom,col,lad))
```

```
Fido es marron. Guau guau!!
```

Figura 15: Ejemplo de llamada de atributos y métodos

Fuente: Elaboración Propia

2.2.2.5. Diferencias entre Python 2.x y 3.x

Las diferencias entre Python 2 y Python 3 no son abismales, como afirma Van Rossum **(2018)** “después de estudiar los cambios, descubrirá que Python realmente no ha cambiado tanto; en general, estamos reparando errores y molestias bien conocidas, y eliminando una gran cantidad de elementos antiguos.”

Un de las diferencias es el cambio de la sentencia print por la función print, lo cual permite mayores opciones al momento de imprimir un resultado o mensaje, el mejor procesamiento de caracteres como la letra ñ y otros caracteres latinos gracias a la codificación Unicode, el cambio de la función raw_input() por la función input(), el mejor tratamiento de número decimales (en Python 2 una división inexacta entre números enteros daba como resultado solo la parte entera del cociente, en Python 3, se da como respuesta la parte entera seguida de varios decimales), el tratamiento de variables mediante listas de compresión, entre otras.

A la fecha de redacción de este informe todavía existen librerías que no están soportadas por Python 3, en dichos casos se recomienda usar Python 2, pero en general, se debe recordar que, como se mencionó en la justificación del proyecto, el soporte para Python 2.7 finalizará en 2020, por lo cual es necesario que la comunidad de programadores realice la migración respectiva de sus proyectos.

2.2.2.6. Definición de la dimensión de Python

Errores de sintaxis

```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- SyntaxError
      |    +-- IndentationError
      |         +-- TabError
```

Figura 16: Árbol de excepciones en Python

Fuente: Elaboración Propia

Syntax Error: Se da cuando el intérprete no logra entender lo escrito por el programador, por ejemplo, cuando falta uno de los paréntesis en una expresión o se utilizan llaves en una sentencia if o bucle (for y while).

Indentation Error: Se da cuando no se alinea correctamente una línea de código con las líneas cercanas. Es una subclase de Syntax Error.

Tab Error: Es una subclase de Indentation Error. Este error aparece cuando en la sangría no se hace un uso adecuado de los tabs y los espacios.

2.3. Definición de Términos Básicos

Activity:

Según la documentación de Microsoft Bot Framework, el conector usa un objeto Activity para pasar información entre el bot y el usuario. El tipo más común de actividad es el mensaje, pero hay otros tipos de actividad que se pueden usar para comunicar varios tipos de información a un bot o canal. (Microsoft, 2018)

Asíncrono:

Es un tipo de comunicación en el cual un usuario puede seguir desarrollando funciones en un sistema mientras este se encuentra ejecutando un proceso que será terminado en un futuro. Para el caso del chatbot, mientras el sistema envía una respuesta a una pregunta anterior, el usuario puede estar preparando una nueva pregunta.

Bot:

Según Microsoft (2017) “es una abreviación de Robot. Un Bot es un programa de computadora autónomo que es capaz de llevar a cabo tareas concretas e imitar el comportamiento humano. Tradicionalmente un bot efectúa sus tareas a través de Internet.” Aunque también existen aquellos que actúan desde el escritorio, por ejemplo, los bots que sean desarrollados con el framework MoMo – Versión 1.00 o anterior de Antakira Software son aplicaciones de escritorio que funcionan en Windows, Mac y Linux.

Bot framework emulator:

Según Microsoft (2018): “El Bot Framework Emulator es una aplicación de escritorio que permite a los desarrolladores probar y depurar sus bots, ya sea de forma local o remota. Usando el emulador, usted puede chatear con su bot e inspeccionar los mensajes que su bot envía y recibe. El emulador muestra los mensajes tal como aparecerían en una interfaz de usuario de chat web y registra las solicitudes y respuestas de JSON al intercambiar mensajes con su bot.”.

Canal:

Es el medio por el cual es transmitido el mensaje o solicitud del usuario. El canal puede ser web (mediante el uso de iframes), a través de Skype, Telegram, Facebook, etc.

Chatbot:

Según Microsoft (2017) “es un programa de computadora autónomo que imita una conversación con personas y que utiliza inteligencia artificial. Un Chatbot puede transformar la manera en que un usuario interactúa con Internet, convirtiendo una serie de tareas en una simple conversación.”

Conector:

Según la documentación respectiva, el Bot Framework Connector proporciona una API REST única que permite que un bot se comunique a través de múltiples canales como Skype, Email, Slack y más. Facilita la comunicación entre bot y usuario transmitiendo mensajes del bot al canal y del canal a bot. En el SDK de BotBuilder para .NET, la biblioteca de Connector habilita el acceso al Connector (Microsoft, 2018).

Diálogo:

Un diálogo permite maquetar y administrar los flujos de una conversación. Además, según la documentación del framework, un diálogo puede estar compuesto por otros cuadros de diálogo para maximizar la reutilización, y un contexto de diálogo mantiene la pila de cuadros de diálogo que están activos en la conversación en

cualquier momento. Una conversación que comprende diálogos es portátil en todas las computadoras, lo que hace posible que la implementación de su bot aumente (Microsoft, 2018).

Inferencia:

En lógica, una inferencia es toda conclusión que se obtiene a partir de una o más premisas. En cuanto a los agentes lógicos (por ejemplo, los sistemas de diálogo), según afirman Russell & Norvig (2004) “la inferencia debe cumplir con el requisito esencial de que cuando se PREGUNTA a la base de conocimiento, la respuesta debe seguirse de lo que se HA DICHO a la base de conocimiento previamente.”

Inteligencia:

Como afirma McCarthy (2007) “La inteligencia es la parte computacional de la capacidad de alcanzar metas en el mundo. Diversos tipos y grados de inteligencia ocurren en las personas, muchos animales y algunas máquinas.”

III. MÉTODOS Y MATERIALES

3.1. Hipótesis

3.1.1. Hipótesis General

El sistema experto basado en Microsoft Bot Framework sirve como soporte en la programación con el lenguaje Python 3.x.

3.1.2. Hipótesis Específica

El sistema experto basado en Microsoft Bot Framework permite corregir errores de sintaxis relacionados a la programación con lenguaje Python 3.x

3.2. Variables de estudio

3.2.1. Definición Conceptual

Variable independiente (VI): Sistema experto
Un sistema experto puede definirse como un sistema informático (hardware y software) que simula a los expertos humanos en un área de especialización dada. Como tal, un sistema experto debería ser capaz de procesar y memorizar información, aprender y razonar en situaciones deterministas e inciertas, comunicar con los hombres y/u otros sistemas expertos, tomar decisiones apropiadas, y explicar por qué se han tomado tales decisiones. Se puede pensar también en un sistema experto como un consultor que puede suministrar ayuda a (o en algunos casos sustituir completamente) los expertos humanos con un grado razonable de fiabilidad (Castillo, Gutiérrez, & Hadi, 1998, p. 15).

Variable dependiente (VD): Lenguaje de programación Python 3.x
El lenguaje de programación Python es muy interesante (...) debido a su simpleza y claridad. Aunque simple, es también un lenguaje poderoso que puede ser usado para administrar sistemas y desarrollar grandes proyectos. Es un lenguaje claro y objetivo, pues va directamente al punto, sin rodeos (Coutinho Menezes, 2017).

3.2.2. Definición Operacional

La variable dependiente será medida considerando cuántos errores de sintaxis cometen los usuarios en el pre test (en donde no usar el sistema experto) y comparando estos resultados con el número de errores de sintaxis que los usuarios cometan en el pos test (en donde si usarán el sistema experto).

Tabla 3

Definición Operacional de las variables

Variables	Dimensiones	Indicadores	ítems	Escala de medición
Sistema Experto	Base de hechos y de conocimientos	Datos y reglas	Número de datos Númerode reglas	KPI
	Motor de inferencia	Mecanismos de resolución	Número de mecanismos de resolución	KPI
Programación en Python 3.x	Errores	Errores de sintaxis	Número de errores de sintaxis corregidos por prueba	KPI

Fuente: Elaboración Propia

3.3. Tipo y nivel de investigación

3.3.1. Tipo de investigación

La presente investigación es de tipo aplicada porque busca la solución de un problema práctico (Hernández Sampieri, 2014). Además, según Castillero Mimenza (2018) este tipo de investigación busca: “encontrar mecanismos o estrategias que permitan lograr un objetivo concreto […]. Por consiguiente, el tipo de ámbito al que se aplica es muy específico y bien delimitado, ya que no se trata de explicar una amplia variedad de situaciones, sino que más bien se intenta abordar un problema específico.”

3.3.2. Nivel de la investigación

La presente investigación es explicativa porque según Hernández Sampieri (2014), “[las investigaciones explicativas] son más estructuradas que los otros tipos de

investigación, y que abarca en su propósito la exploración, la descripción y correlación con lo cual permite generar un sentido de entendimiento más completo.”

3.4. Diseño de la investigación

La investigación será cuasiexperimental porque como afirma Hernández Sampieri (2014):

> En los diseños cuasiexperimentales, los sujetos no se asignan al azar a los grupos ni se emparejan, sino que dichos grupos ya están conformados antes del experimento: son grupos intactos (la razón por la que surgen y la manera como se integraron es independiente o aparte del experimento).

En la presente investigación se realizará un pre test y un pos test porque se evaluará la calidad del sistema mediante la formación 1 grupo de muestra al que se le aplicará un test o prueba sin usar el sistema (pre test) y un test con el uso del sistema (pos test), con lo cual se verificará si el sistema realmente es útil para los estudiantes de ingeniería de sistemas de la Universidad Privada Telesup.

3.5. Población y muestra de estudio

3.5.1. Población

En la presente investigación la población que se tomará como objeto de estudio estará conformada por todos los estudiantes de la carrera de ingeniería de sistemas de la Universidad Privada Telesup, que suman un total de 430 alumnos aproximadamente.

3.5.2. Muestra

Para Balestrini Acuña (2006), la muestra “es obtenida con el fin de investigar, a partir del conocimiento de sus características particulares, las propiedades de una población”

El muestreo fue por conveniencia, de manera que la muestra estuvo conformada por un conjunto 32 alumnos de la carrera de ingeniería de sistemas de la Universidad Privada Telesup, de los cuales 25 cursaban el V ciclo 2018-I turno mañana, 6 cursaban el V ciclo 2018-I turno tarde y 1 cursaba el I ciclo 2018-I.

3.6. Técnicas e Instrumentos de Recolección de Datos

3.6.1. Técnica de recolección de datos:

La técnica que se utilizó fue la encuesta que consistió en recopilar la información en la muestra de estudio.

Se optó por la técnica de encuestas porque es masiva y rápida, menos costosa y permite de forma relativamente rápida, obtener la información.

La encuesta constó de 2 preguntas en donde se especificaron los tópicos básicos, esta encuesta estuvo disponible en el sistema y fue resuelta por un grupo de alumnos sin apoyo del sistema experto. Luego se realizó una segunda encuesta con preguntas sobre los mismos tópicos al mismo grupo, pero con la diferencia de que los estudiantes utilizaron el sistema experto.

3.6.2. Instrumento de recolección de datos: El Cuestionario

Según afirma López-Roldán & Fachelli (2015) “El cuestionario constituye el instrumento de recogida de los datos donde aparecen enunciadas las preguntas de forma sistemática y ordenada, y en donde se consignan las respuestas mediante un sistema establecido de registro sencillo.”

La investigación estuvo dada por el instrumento llamado cuestionario, al cual se pudo acceder de manera online. En la parte superior del cuestionario se identificó el propósito.

Tiempo de duración: 75 minutos aproximadamente.

Contenido: El cuestionario estuvo conformado por 4 preguntas las cuales cubrieron 2 tópicos principales dentro del lenguaje de programación Python (condicionales y bucles). Esta prueba se resolvió en 2 fases de 2 ejercicios cada una. La primera fase tomó un tiempo aproximado de 45 minutos y la segunda en promedio 30 minutos.

3.6.2.1. Validez del Instrumento

Tabla 4

Validación de Expertos

Mgtr. Ing. Barrantes Ríos Edmundo José	Experto Metodológico
Mgtr. Ovalle Paulino, Christian	Experto Temático

Fuente: Elaboración Propia

3.7. Métodos de Análisis de Datos

El análisis de datos se llevó a cabo con los valores que se obtuvieron mediante la aplicación del instrumento elegido para la recolección de datos como es el cuestionario.

Para los análisis estadísticos correspondientes, es decir el análisis descriptivo e inferencial, se utilizó el software estadístico STATGRAPHICS. La estadística descriptiva implica las estimaciones de las medidas de tendencia central y en cuanto a la estadística inferencial, esta implica la comparación de las muestras de los resultados de los instrumentos a fin de verificar las hipótesis planteadas en la investigación.

Se utilizaron los siguientes estadígrafos:

- Estadística descriptiva: Promedio, desviación estándar, coeficiente de variación, curtosis estandarizada, entre otras.
- Estadística inferencial: Para la prueba de hipótesis se utilizó la prueba t de student para la comparación de medias y la prueba f para la comparación de la desviación estándar.

Los análisis se realizaron con un nivel de significancia estadística del 95%.

IV. RESULTADOS

4.1. Resultados Descriptivos

Se realizó una comparación de dos muestras - Pre Test & Pos test

Muestra 1: Pre Test

Muestra 2: Pos test

Selección de la Variable: Pos test

Muestra 1: 4 valores en el rango de 15.0 a 26.0

Muestra 2: 4 valores en el rango de 1.0 a 4.0

Tabla 5

Resumen del procesamiento de datos del cuestionario

	NUMERO ERRORES DE SINTAXIS		ERRORES
ALUMNOS	PRE TEST	POS TEST	RESTA
ALUMNO 1	25	0	25
ALUMNO 2	31	0	31
ALUMNO 3	21	2	19
ALUMNO 4	7	0	7
...	...	...	...
ALUMNO 32	8	0	8

Fuente: Elaboración Propia

Tabla 6

Resumen Estadístico del cuestionario

	Pre Test	*Pos test*
Recuento	4	4
Promedio	21.75	2.25
Desviación Estándar	4.99166	1.25831
Coeficiente de Variación	22.9502%	55.9247%
Mínimo	15.0	1.0
Máximo	26.0	4.0
Rango	11.0	3.0
Sesgo Estandarizado	-0.861626	0.922101
Curtosis Estandarizada	-0.000712372	0.909229

Fuente: Elaboración Propia

Esta tabla anterior contiene el resumen estadístico para las dos muestras de datos a fin de evaluar las medidas de tendencia central del diseño cuasi experimental utilizándose como opción las diferencias entre los estadísticos de las dos muestras siendo esta estadísticamente significativas; de particular interés son el sesgo estandarizado y la curtosis estandarizada que pueden usarse para comparar si las muestras provienen de distribuciones normales.

4.2. Prueba de Hipótesis

4.2.1. Hipótesis Principal

La contrastación de la hipótesis principal se realiza mediante la contrastación de la hipótesis específica. Si se rechaza la hipótesis nula correspondiente a la hipótesis específica, entonces necesariamente se debe aceptar la hipótesis principal.

4.2.2. Contrastación de la Hipótesis Específica

a) El planteo de las hipótesis

H0: El sistema experto basado en Microsoft Bot Framework no permite corregir errores de sintaxis relacionados a la programación con lenguaje Python 3.x

H1: El sistema experto basado en Microsoft Bot Framework si permite corregir errores de sintaxis relacionados a la programación con lenguaje Python 3.x

b) Comparación de medias:

Intervalos de confianza del 95.0% para la media de Pre Test: 21.75 +/- 7.94285 [13.8071; 29.6929]
Intervalos de confianza del 95.0% para la media de Pos test: 2.25 +/- 2.00225 [0.247753; 4.25225]
Intervalos de confianza del 95.0% intervalo de confianza para la diferencia de medias:
Suponiendo varianzas iguales: 19.5 +/- 6.29814 [13.2019; 25.7981]

Prueba t para comparar medias

Hipótesis nula: media1 = media2
Hipótesis Alternativa.: media1 <> media2
Suponiendo varianzas iguales: t = 7.57603 valor-P = 0.000274893
Se rechaza la hipótesis nula para alfa = 0.05.

La prueba t permite comparar las medias de las dos muestras, para los intervalos, o cotas, de confianza para cada media y para la diferencia entre las medias.
Se nota que es particular interés el intervalo de confianza para la diferencia entre las medias, el cual se extiende desde 13.2019 hasta 25.7981. Puesto que el intervalo no contiene el valor 0, existe una diferencia estadísticamente significativa entre las medias de las dos muestras, con un nivel de confianza del 95.0%.

Al evaluar la prueba t para evaluar hipótesis específicas acerca de la diferencia entre las medias de las poblaciones de las cuales provienen las dos muestras; en este caso, la prueba se ha construido para determinar si la diferencia entre las dos medias es igual a 0.0 versus la hipótesis alterna de que la diferencia no es igual a 0.0. Puesto que el valor-P calculado es menor que 0.05, se puede rechazar la hipótesis nula en favor de la alterna.

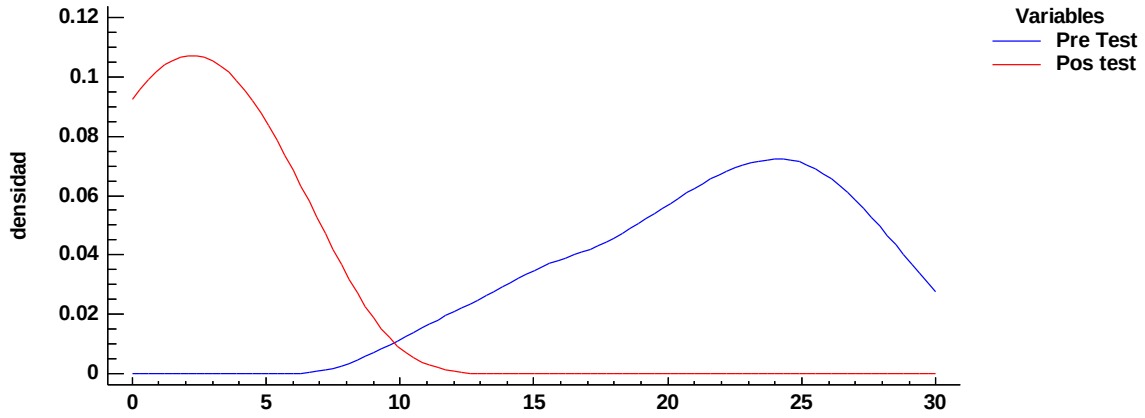

Figura 17: *Densidades suavizadas*

Fuente: Elaboración Propia

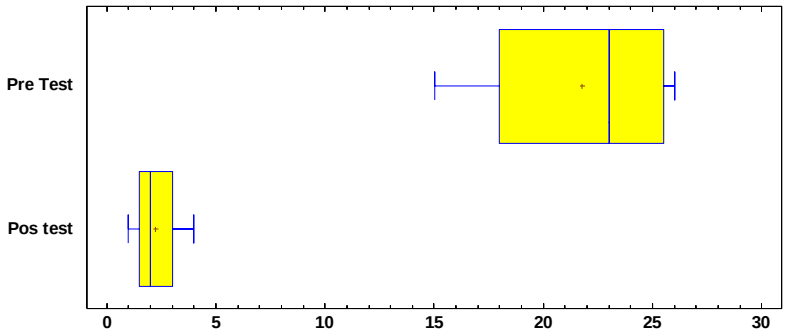

Figura 18: *Gráfico de Caja y Bigotes*

Fuente: Elaboración Propia

c) Comparación de desviaciones estándar

Tabla 7

Comparación de desviaciones estándar

	Pre Test	*Pos test*
Desviación Estándar	4.99166	1.25831
Varianza	24.9167	1.58333
Gl	3	3

Razón de Varianzas= 15.7368

Fuente: Elaboración Propia

Intervalos de confianza del 95.0%

Desviación Estándar de Pre Test: [2.82773; 18.6117]

Desviación Estándar de Pos test: [0.71282; 4.69167]

Razones de Varianzas: [1.01928; 242.964]

Prueba-F para comparar Desviaciones Estándar

Hipótesis Nula: sigma1 = sigma2

Hipótesis Alternativa.: sigma1 <> sigma2

F = 15.7368 valor-P = 0.0486887

Se rechaza la hipótesis nula para alfa = 0.05.

Para asegurar el manejo de las medidas de tendencia central y comprobar su distribución normal se ejecutó la prueba-F para comparar las varianzas de las dos muestras. También construye intervalos o cotas de confianza para cada desviación

estándar y para la razón de varianzas. De particular interés es el intervalo de confianza para la razón de varianzas, el cual se extiende desde 1.01928 hasta 242.964. Puesto que el intervalo no contiene el valor de 1, existe diferencia estadísticamente significativa entre las desviaciones estándar de las dos muestras con un 95.0%.

También puede ejecutarse una prueba-F para evaluar una hipótesis específica acerca de las desviaciones estándar de las poblaciones de las cuales provienen las dos muestras. En este caso, la prueba se ha construido para determinar si el cociente de las desviaciones estándar es igual a 1.0 versus la hipótesis alternativa de que el cociente no es igual a 1.0. Puesto que el valor-P calculado es menor que 0.05, se puede rechazar la hipótesis nula en favor de la alterna.

d) Prueba de Kolmogórov-Smirnov

Estadístico DN estimado = 1.0

Estadístico DN estimado = 1.0

Estadístico K-S bilateral para muestras grandes = 1.41421

Valor P aproximado = 0.0366314

Se realiza la prueba de Kolmogórov-Smirnov para comparar las distribuciones de las dos muestras y reafirmar que los datos están dentro de la distribución normal porque la distancia máxima entre las distribuciones acumuladas de las dos muestras, en este caso, la distancia máxima es 1.0, que puede verse gráficamente seleccionando Gráfica de Cuantiles de particular interés es el valor-P aproximado para la prueba. Debido a que el valor-P es menor que 0.05, existe una diferencia estadísticamente significativa entre las dos distribuciones normales con un nivel de confianza del 95.0%.

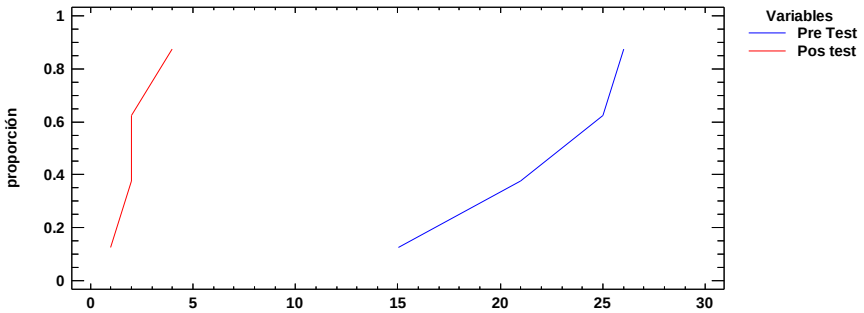

Figura 19: *Gráfico de Cuantiles*

Fuente: Elaboración Propia

V. DISCUSIÓN

Considerando los resultados del estudio obtenidos mediante la estadística descriptiva e inferencial observamos que existe un grado de correlación directo entre el uso de un sistema experto basado en Microsoft Bot Framework y el aprendizaje del lenguaje de programación python, a continuación, se procede a compararlos resultados de esta investigación y discutir los principales hallazgos con las conclusiones de los antecedentes y teorías referenciados en el Capítulo II.

En la tesis de Vásquez Ruiz (2013) titulada *"Desarrollo de un sistema experto para evaluar las competencias en investigación de los docentes de la Universidad Nacional de San Martín"*; se concluyó que "se evaluó la aplicación del Sistema Experto, al margen del test de consistencia del software, determinando su eficiencia y potencialidad en la medición de los niveles de competencias en investigación en la Universidad Nacional de San Martín."

En la presente tesis se evaluó el sistema mediante un pre test y un pos test y se consideró como indicador el número de errores de sintaxis en cada prueba. Luego de observar que el número de errores disminuyó significativamente se concluyó que el sistema experto basado en Microsoft Bot Framework sí permite guiar a los estudiantes de la Carrera de Ingeniería de Sistemas de la Universidad Privada Telesup en la solución de errores de sintaxis relacionados al lenguaje Python 3.x.

Hernández Sánchez & Lozada Cortés (2017) realizaron una tesis titulada: *"Aplicación para dispositivos móviles con sistema operativo Android para el aprendizaje de conceptos básicos de lógica de programación",* en la Universidad Distrital Francisco José de Caldas y concluyeron que "según las pruebas de integración, y del sistema junto con la aceptación de la aplicación evidenciada en las pruebas de validación de juego, se logra que la aplicación sea una herramienta de apoyo en el aprendizaje de conceptos básicos de lógica de programación para los estudiantes de primeros semestres que cursan carreras afines al desarrollo de software".

Asimismo, Mendoza & Cáceres, (2016), realizaron una tesina titulada “Sistema experto para la selección de postulantes en puestos de una agencia bancaria usando la metodología CommonKads” y concluye que “El uso de Sistemas Expertos de apoyo a la toma de decisiones, y en específico para este trabajo, ayudó a reducir el tiempo, costos y que los resultados sean más exactos. El tiempo empleado en los procesos de selección serán menores, ya que el tiempo empleado para calificar y obtener un resultado producto de las evaluaciones psicológicas será inmediato. Los costos, permitirá la reducción de personal evaluador, menor espacio físico (test online), y de cara a los proyectos, se evitarán reprocesos en selección de personal por incapacidad y falta de respuesta del personal seleccionado.”

En cuanto a la presente investigación, de los resultados de las pruebas realizadas se colige que el sistema experto basado Microsoft Bot Framework sirve de apoyo y refuerzo en cuanto al aprendizaje de conceptos básicos del lenguaje de programación Python 3.x en los estudiantes de la Carrera de Ingeniería de Sistemas de la Universidad Privada Telesup.

VI. CONCLUSIONES

Al finalizar la investigación se llegó a las siguientes conclusiones:

Se concluye que el sistema experto basado en Microsoft Bot Framework si sirve como soporte en la programación con el lenguaje Python 3.x en los estudiantes de la Carrera de Ingeniería de Sistemas de la Universidad Privada Telesup.

Producto de la observación de los resultados de contrastar la hipótesis específica, se rechaza la hipótesis nula y se colige que el sistema experto basado en Microsoft Bot Framework sí permite corregir errores de sintaxis relacionados a la programación con lenguaje Python 3.x en los estudiantes de la Carrera de Ingeniería de Sistemas de la Universidad Privada Telesup a un nivel de significancia estadística del 95%. Se demostró mediante la prueba de Kolmogórov-Smirnov que la distribución de los datos era normal y que, por tanto, el contraste de la hipótesis específica se debía realizar mediante la prueba t de student para la comparación de medias y la prueba f para la comparación de las desviaciones estándar. Para la prueba de t de student se consideró como hipótesis nula que la media aritmética del pre test era igual a la media del pos test, luego de realizar esta prueba se concluyó que dicha hipótesis debía ser rechazada en favor de la hipótesis del investigador, la cual afirma que la media del pos test es menor que la del pre test. Finalmente, con respecto a la prueba de f se consideró como hipótesis de trabajo que el coeficiente sigma del pre test era igual al del pos test, luego de realizar esta prueba se demostró que la hipótesis alternativa, la cual afirma que el coeficiente sigma del pos test es menor que la del pre test, debía ser aceptada.

VII. RECOMENDACIONES

Para futuros trabajos se recomienda la realización de un sistema experto como soporte a la programación en lenguajes como C#, Java, PHP, Ruby; y otros lenguajes que tengan alta demanda o alto potencial en el mercado laboral.

También se recomienda hacer que el sistema pueda ayudar a corregir errores en tiempo de ejecución y errores lógicos. En cuanto a los errores en tiempo de ejecución tenemos los Arithmetic Errors (Zero Division Error), Index Error, etc. En cuanto a los errores lógicos, por ejemplo, si se pide hacer un programa que calcule la suma de los “n” primeros números naturales, se podría crear una regla que verifique que el usuario realmente está sumando esos números, porque puede suceder que el usuario haya cometido el error de hacer un programa que sume los n+1 primeros números naturales, o que el programa solo sume números pares.

Otro aspecto importante es que en futuras investigaciones se puede medir el tiempo que demoran los programadores resolviendo un conjunto de ejercicios en el pre test y en el pos test.

También se pueden realizar preguntas considerando otros tópicos los cuales son: uso de funciones, listas, diccionarios, tuplas, programación orientada a objetos, etc.

Se recomienda implementar este sistema en concursos como los realizados en la Jornada de Integración de Estudiantes de Ingeniería de Sistemas (JINESIS), o en los salones de clases para la evaluación de los estudiantes.

REFERENCIAS BIBLIOGRÁFICAS

Annovi, F. (2016). *La aventura de contar: de la máquina de Turing a la computación cuántica.* Barcelona: Batiscafo.

Badaró, S., Javier, L., & Jorge Agüero, M. (2013). Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones. *Ciencia y Tecnología*, 349-354.

Balestrini Acuña, M. (2006). *COMO SE ELABORA EL PROYECTO DE INVESTIGACION.* Caracas, República Bolivariana de Venezuela: BL Consultores Asociados, Servicio Editorial.

Benites Llerena, E. O., & Cahuata Peralta, J. R. (2016). *Sistema experto para la detección de operaciones sospechosas de lavado de activos en entidades financieras utilizando CommonKADS.* Tesis de grado, Universidad Nacional Mayor de San Marcos, Lima, Perú.

Cabrera Espinoza, J. O. (2016). *Desarrollo de un software educativo para la enseñanza aprendizaje del lenguaje de programación JAVA para los alumnos del segundo año de bachillerato figura profesional aplicaciones informáticas del colegio de bachillerato "Marta Bucaram de Roldós".* Tesis de grado, Universidad Nacional de Loja, Loja. Obtenido de http://dspace.unl.edu.ec/handle/123456789/14556

Castillero Mimenza, O. (2018). *Los 15 tipos de investigación (y características).* Obtenido de sitio web de Psicología y Mente: https://psicologiaymente.net/miscelanea/tipos-de-investigacion

Castillo, E., Gutiérrez, J. M., & Hadi, A. S. (1998). *Sistemas Expertos y Modelos de Redes Probabilísticas.* Madrid: Academia Española de Ingeniería.

Corral Díaz, M. A., & Carrillo Medina, J. L. (2015). *Creación de un sistema experto para asistir al ingeniero en software en la elaboración de documentos de requerimientos.* Tesis de Máster, Universidad de las Fuerzas Armadas, Investigación, innovación y transferencia tecnológica, Latacunga, Ecuador.

Coutinho Menezes, N. N. (2016). *Introducción a la Programación con Python.* Sao Paulo: Novatec Editora Ltda.

Espinoza Cubas, D. (2015). *Análisis, diseño e implementación de un sistema experto para la evaluación de la calidad de tanques de almacenamiento de combustibles.* Tesis de Grado, Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería, Lima, Perú.

Feigenbaum, E. A. (1979). *Handbook of Artificial Intelligence.* (A. Ban, & E. A. Feigenbaum, Edits.) Stanford: Stanford University.

Fernández García, E. (2017). *Diseño y desarrollo de una plataforma de soporte inteligente virtual para aprendizaje de inglés.* Tesis de Máster, Universidad Pontificia Comillas, Madrid, España.

Gutiérrez Lozano, F., & Hernández Valderrama, S. (2015). *Implementación de un sistema Experto para el diagnóstico de desnutrición en niños menores de 5 años utilizando lógica fuzzy.* Tesis de Grado, Universidad Nacional de Trujillo, Facultad de Ciencias Físicas y Matemáticas, Trujillo, Perú.

Hernández Sampieri, R. (2014). *METODOLOGÍA DE LA INVESTIGACIÓN* (Sexta ed.). México D.F.: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.

Hernández Sánchez, L. S., & Lozada Cortés, M. L. (2017). *Aplicación para dispositivos móviles con sistema operativo Android para el aprendizaje de conceptos básicos de lógica de programación.* Tesis de Grado, Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Bogotá, Colombia.

Hosseini, S. (2 de Abril de 2017). *Chat Bots Implementation and User Engagement.* Finlandia: Helsinki Metropolia University of Applied Sciences.

Iñiguez Loján, C. A., & Luna Mejía, A. P. (2014). *Desarrollo e implementación de un sistema experto para la evaluación del aprendizaje receptivo del idioma inglés: caso: estudiantes de la academia particular fine tuned english (zamora) de edades entre 3 a 18 años.* Tesis de Grado, Universidad

Nacional de la Loja, Área de la energía, las industrias y los recursos naturales no renovables., Loja, Ecuador.

López-Roldán, P., & Fachelli, S. (2015). *METODOLOGÍA DE LA INVESTIGACIÓN SOCIAL CUANTITATIVA.* Barcelona, España: Universitat Autònoma de Barcelona.

Martínez Cámara, E. (Marzo de 2015). *Análisis de Opiniones en Español.* Sociedad Española para el Procesamiento del Lenguaje Natural.

McCarthy, J. (2007). *What is artificial intelligence?* Stanford University, Computer Science Department, California.

Mendoza Cárdenas, S. R., & Cáceres Zárate, J. N. (2016). *Sistema experto para la selección de postulantes en puestos de una agencia bancaria usando la metodología CommonKads.* Tesina, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas, Lima, Perú.

Microsoft. (2017). *TI Capacitación: Diplomado de Bots*. Obtenido de sitio web de TI Capacitación: https://ticapacitacion.com/curso/botses

Microsoft. (2018). *Azure Bot Service Documentation*. Obtenido de sitio web de Microsoft: https://docs.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-3.0

Reyes Macías, J. R. (2013). *Diseño conceptual de un Sistema Experto Informático, como herramienta de apoyo en el proceso de elaboración de nuevas leyes, procedimientos, normas y reglamentos en el Ecuador (Tesis Maestría).* Universidad Andina Simón Bolívar, Quito, Ecuador.

Roel Pineda, V. (2005). *La crisis general de la globalización. Crítica la fundamentalismo neoliberal.* Lima: Cartolan.

Russell, S. J., & Norvig, P. (2004). *INTELIGENCIA ARTIFICIAL. UN ENFOQUE MODERNO* (Segunda ed.). (D. Fayerman Aragón, Ed., & J. M. Corchado Rodríguez, Trad.) Madrid, España: PEARSON EDUCACIÓN, S.A.

Santacreu Ríos, L. J. (2014). *Implementación de reglas en la toma de decisiones en incendio forestal: Reglas básicas de razonamiento que imitan el*

comportamiento de los Expertos, motor de inferencia basado en Lógica Difusa. España: Editorial Académica Española.

Urbina Padilla, D. A. (20 de Noviembre de 2013). Las neurociencias y el espíritu: Una visión teísta. Lima, Perú. Obtenido de https://www.youtube.com/watch?v=Y1153hHRsd8

Van Rossum, G. (2017). *El tutorial de Python.* (F. L. Drake, Ed.) Argentina. Obtenido de http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf

Van Rossum, G. (4 de Junio de 2018). *What's New In Python 3.0*. Obtenido de Python Software Foundation Web site: https://docs.python.org/3/whatsnew/3.0.html

Vásquez Ruiz, J. J. (2013). *Desarrollo de un sistema experto para evaluar las competencias en investigación de los docentes de la Universidad Nacional de San Martín.* Universidad Nacional de San Martín, Tarapoto.

ANEXOS

Anexo 1: Matriz de Consistencia

TÍTULO: Sistema experto como soporte a la programación con lenguaje Python 3.x basado en Microsoft Bot Framework								
PROBLEMA PRINCIPAL	**OBJETIVO GENERAL**	**HIPÓTESIS GENERAL**	**VARIABLES**	**DEFINICIÓN CONCEPTUAL**	**DEFINICIÓN OPERACIONAL**	**DIMENSIONES**	**INDICADORES**	**ESCALA DE INDICADORES**
¿Cómo el sistema experto sirve como soporte a la programación con lenguaje Python 3.x basado en Microsoft Bot Framework?	Demostrar que el sistema experto sirve como soporte en la programación con lenguaje Python 3.x basado en Microsoft Bot Framework.	El sistema experto basado en Microsoft Bot Framework sirve como soporte en la programación con el lenguaje Python 3.x.	VI: Sistema experto	La variable independiente es el sistema experto, el cual está conformado por la base de hechos, la base de conocimientos y el motor de inferencia.	La variable independiente se mide mediante casos de prueba, por ejemplo, la prueba de compatibilidad, entre otras.	Base de hecho y de conocimientos	Datos y reglas	Número de datos Número de reglas (KPI)
						Motor de inferencia	Mecanismos de resolución	Número de mecanismos de resolución (KPI)
SECUNDARIOS	**SECUNDARIOS**	**SECUNDARIOS**	**VARIABLES**	**DEFINICIÓN CONCEPTUAL**	**DEFINICIÓN OPERACIONAL**	**DIMENSIONES**	**INDICADORES**	**ESCALA DE INDICADORES**
¿Cómo usar el sistema experto basado en Microsoft Bot Framework para corregir errores de sintaxis relacionados al lenguaje de programación Python 3.x?	Demostrar que usar el sistema experto basado en Microsoft Bot Framework sirve para corregir errores de sintaxis relacionados al lenguaje de programación Python 3.x	El sistema experto basado en Microsoft Bot Framework permite corregir errores de sintaxis relacionados a la programación con lenguaje Python 3.x	**VD:** Programación en lenguaje Python	Python es un lenguaje de programación interpretado, estructurado y orientado a objetos. Permite el scripting y el desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.	La variable dependiente se mide a través del número de errores que el sistema haya corregido. No se consideran errores de sintaxis, solo excepciones.	ERRORES	Errores de sintaxis	KPI

Anexo 2: Matriz de Operacionalidad

Variables	Dimensiones	Indicadores	ítems	Escala de medición
Sistema Experto	Base de hechos y de conocimientos	Datos y reglas	Número de datos Númerode reglas	KPI
	Motor de inferencia	Mecanismos de resolución	Número de mecanismos de resolución	KPI
Programación en Python 3.x	Errores	Errores de sintaxis	Número de errores de sintaxis corregidos por prueba	KPI

Anexo 3: Instrumento

UNIVERSIDAD PRIVADA TELESUP
CARRERA DE: INGENIERÍA DE SISTEMAS
ENCUESTA 1: ENCUESTA PARA EVALUAR CONOCIMIENTOS INICIALES SOBRE EL LENGUAJE DE PROGRAMACIÓN PYTHON 3.X.

Estimado estudiante:

La presente encuesta tiene como objetivo recopilar información sobre el grado de conocimientos que usted tiene sobre las estructuras básicas del lenguaje de programación Python 3.x antes de usar el sistema experto.

1. PREGUNTAS SOBRE CONDICIONALES: Escriba un programa que reciba como entrada un año cualquiera y que luego indique cuantos años faltan para llegar al año actual o bien, cuantos años han pasado desde el año actual.

2. PREGUNTAS SOBRE BUCLES: Evalúe el factorial de un número utilizando el ciclo WHILE.

ENCUESTA 2: ENCUESTA PARA EVALUAR CONOCIMIENTOS SOBRE EL LENGUAJE DE PROGRAMACIÓN PYTHON 3.X

Estimado estudiante:

La presente encuesta tiene como objetivo recopilar información sobre el grado de conocimientos que usted tiene sobre las estructuras básicas del lenguaje de programación Python 3.x mientras utiliza el sistema experto.

1. PREGUNTA SOBRE CONDICIONALES: Escriba un programa que permita dividir dos números y que imprima un mensaje indicando si la división es exacta o inexacta.

2. PREGUNTA SOBRE BUCLES: Imprima los 10 primeros números de la sucesión de Fibonacci.

Anexo 4: Validación de Instrumentos

CERTIFICADO DE VALIDEZ DE CONTENIDO DE LOS INSTRUMENTOS

Nº	Dimensiones / ítems	Pertinencia[1]		Relevancia[2]		Claridad[3]		Sugerencias
	I.ENCUESTA 1	Sí	No	Sí	No	Sí	No	
1	PREGUNTA SOBRE CONDICIONALES: Escriba un programa que reciba como entrada un año cualquiera y que luego indique cuantos años faltan para llegar al año actual o bien, cuantos años han pasado desde el año actual.	✓		✓		✓		
2	PREGUNTA SOBRE BUCLES: Evalúe el factorial de un número utilizando el ciclo WHILE.	✓		✓		✓		
Nº	Dimensiones / ítems	Pertinencia[1]		Relevancia[2]		Claridad[3]		Sugerencias
	II.ENCUESTA 2	Sí	No	Sí	No	Sí	No	
1	PREGUNTA SOBRE CONDICIONALES: Escriba un programa que permita dividir dos números y que imprima un mensaje indicando si la división es exacta o inexacta.	✓		✓		✓		
2	PREGUNTA SOBRE BUCLES: Imprima los 10 primeros números de la sucesión de Fibonacci.	✓		✓		✓		

Observaciones (precisar si hay suficiencia): Si Hay Suficiencia

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [ ] No aplicable [ ]

Apellidos y nombres del juez validador. Dr/ Mg:

Ovalle Paulino Christian

DNI: 40234321

Especialidad del validador: Docente Temático

07/04/2018.

[1]**Pertinencia:** El ítem corresponde al concepto teórico formulado.
[2]**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo
[3]**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

Firma del Experto Informante.
Especialidad

Nº	Dimensiones / items	Pertinencia[1]		Relevancia[2]		Claridad[3]		Sugerencias
	I.ENCUESTA 1	Si	No	Si	No	Si	No	
1	PREGUNTA SOBRE CONDICIONALES: Escriba un programa que reciba como entrada un año cualquiera y que luego indique cuantos años faltan para llegar al año actual o bien, cuantos años han pasado desde el año actual.	√		√		√		
2	PREGUNTA SOBRE BUCLES: Evalúe el factorial de un número utilizando el ciclo WHILE.	√		√		√		

Nº	Dimensiones / items	Pertinencia[1]		Relevancia[2]		Claridad[3]		Sugerencias
	II.ENCUESTA 2	Si	No	Si	No	Si	No	
1	PREGUNTA SOBRE CONDICIONALES: Escriba un programa que permita dividir dos números y que imprima un mensaje indicando si la división es exacta o inexacta.	√		√		√		
2	PREGUNTA SOBRE BUCLES: Imprima los 10 primeros números de la sucesión de Fibonacci.	√		√		√		

Observaciones (precisar si hay suficiencia): SI EXISTE SUFICIENCIA

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [ ] No aplicable [ ]

Apellidos y nombres del juez validador. Dr/ Mg:

BARRANTES RIOS EDMUNDO JOSÉ

DNI: 25651955

Especialidad del validador: Docente Metodologo

[1]**Pertinencia:** El ítem corresponde al concepto teórico formulado.
[2]**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo
[3]**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

Firma del Experto Informante.
Especialidad

Anexo 5: Matriz de Datos

	NÚMERO ERRORES DE SINTAXIS		ERRORES
ALUMNOS	PRE TEST	POS TEST	RESTA
ALUMNO 1	25	0	25
ALUMNO 2	31	0	31
ALUMNO 3	21	2	19
ALUMNO 4	7	0	7
ALUMNO 5	28	0	28
ALUMNO 6	27	0	27
ALUMNO 7	14	0	14
ALUMNO 8	26	4	22
ALUMNO 9	25	0	25
ALUMNO 10	79	0	79
ALUMNO 11	15	0	15
ALUMNO 12	18	0	18
ALUMNO 13	18	0	18
ALUMNO 14	36	0	36
ALUMNO 15	26	0	26
ALUMNO 16	7	0	7
ALUMNO 17	9	0	9
ALUMNO 18	9	0	9
ALUMNO 19	10	0	10
ALUMNO 20	25	2	23
ALUMNO 21	8	0	8
ALUMNO 22	17	0	17
ALUMNO 23	22	0	22
ALUMNO 24	0	0	0
ALUMNO 25	0	0	0
ALUMNO 26	15	1	14
ALUMNO 27	13	0	13
ALUMNO 28	18	0	18
ALUMNO 29	18	0	18
ALUMNO 30	16	0	16
ALUMNO 31	16	0	16
ALUMNO 32	8	0	8
	PROMEDIO		**18.69**
	DESV. ESTÁNDAR		**13.90**

Anexo 6: Propuesta de Valor

El sistema experto está conformado por dos elementos, el primero es la plataforma web REPL.IT, la cual brinda un entorno de programación online en diferentes lenguajes, en este proyecto se utilizó la consola online de Python 3. El segundo elemento se refiere al bot conversacional, el cual respondió a las preguntas del usuario.

El bot conversacional fue desarrollado utilizando Microsoft Bot Framework para C# y fue desplegado utilizando el portal Azure.

Cuando se realizó el pre test y el pos test, la plataforma REPL.IT y el bot conversacional estuvieron disponibles mediante iframes en una página web alojada en GALEON.COM.

Se utilizó la plataforma web browsershots.org para verificar que el sistema experto funciona en la mayoría de los navegadores.

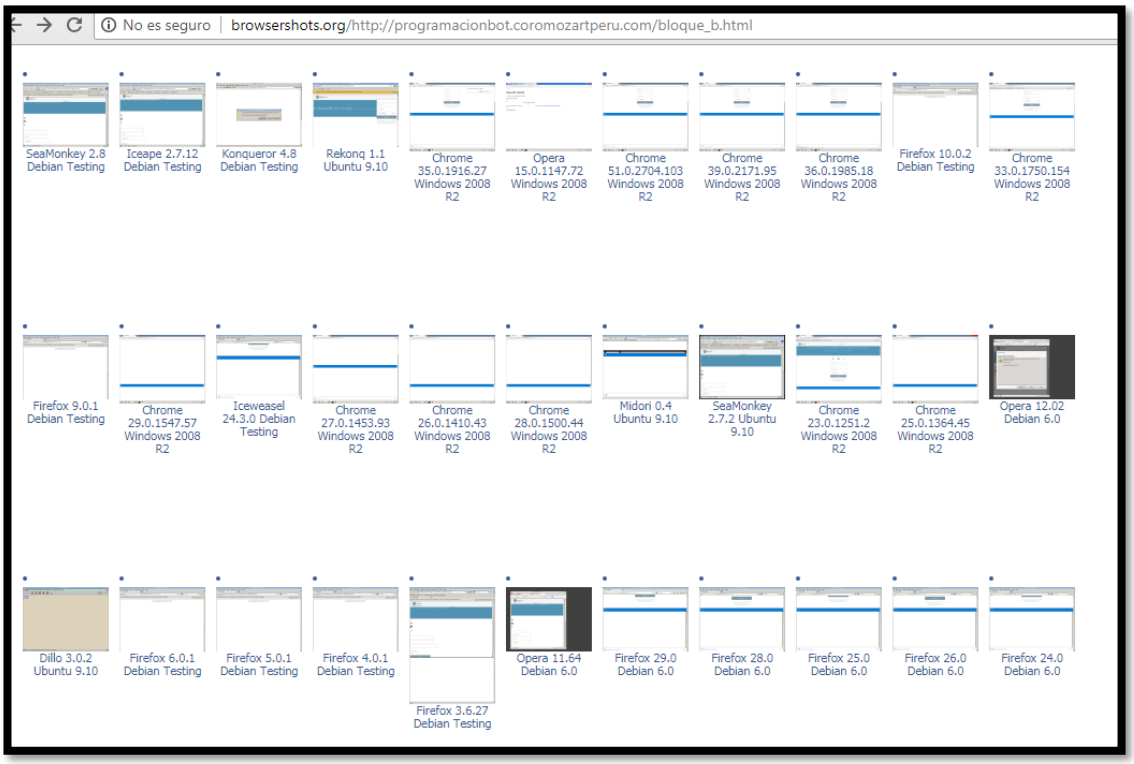

Figura 20: Comprobación de compatibilidad con navegadores

Fuente: Elaboración Propia

Figura 21: Foto 1 en la que se observa a los estudiantes resolviendo la prueba

Fuente: Elaboración Propia

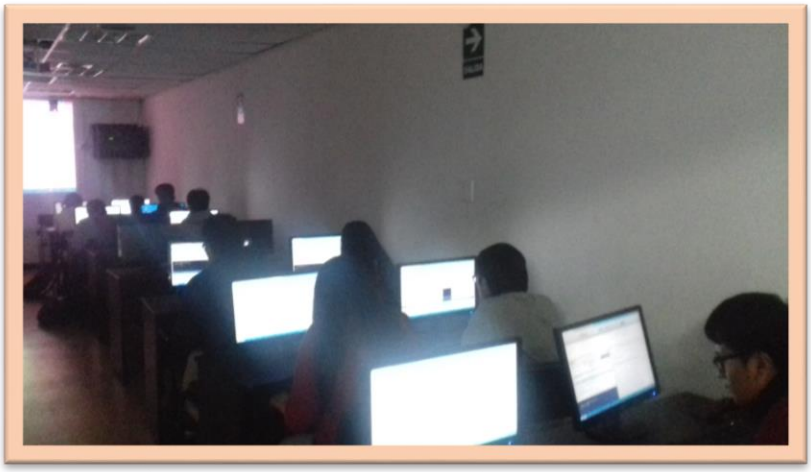

Figura 22: Foto 2 en la que se observa a los estudiantes resolviendo la prueba

Fuente: Elaboración Propia

Figura 23: Foto 3 en la que se observa a los estudiantes resolviendo la prueba

Fuente: Elaboración Propia

A continuación, se muestran dos capturas de pantalla, la primera corresponde al pre test y la segunda al pos test del primer ejercicio, el cual está relacionado al uso de condicionales. Se puede apreciar la mejora en cuanto a los errores de sintaxis y la respuesta del bot conversacional frente a una pregunta del usuario.

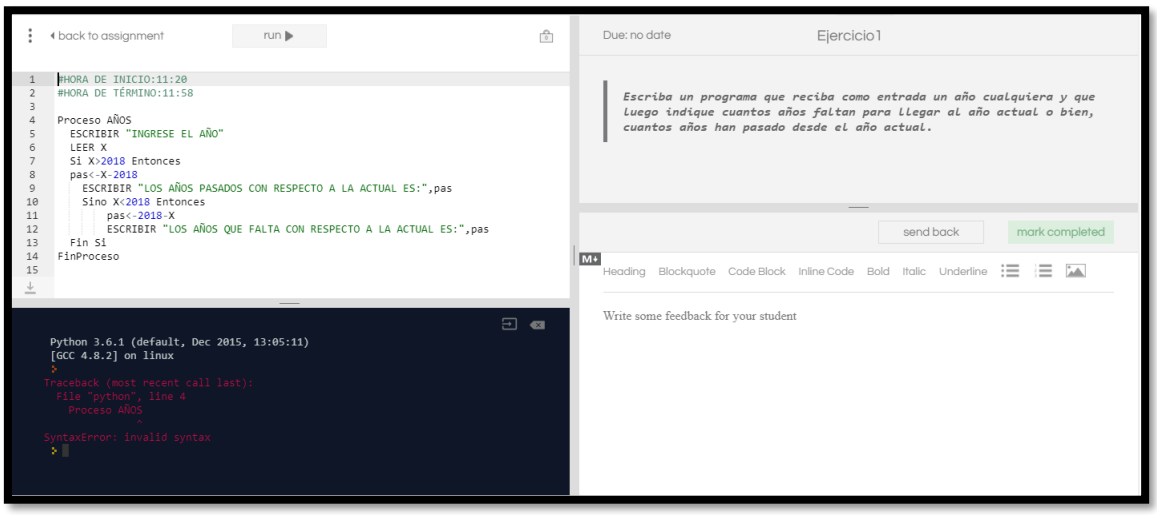

Figura 25: Primer ejercicio del Pre test

Fuente: Elaboración Propia

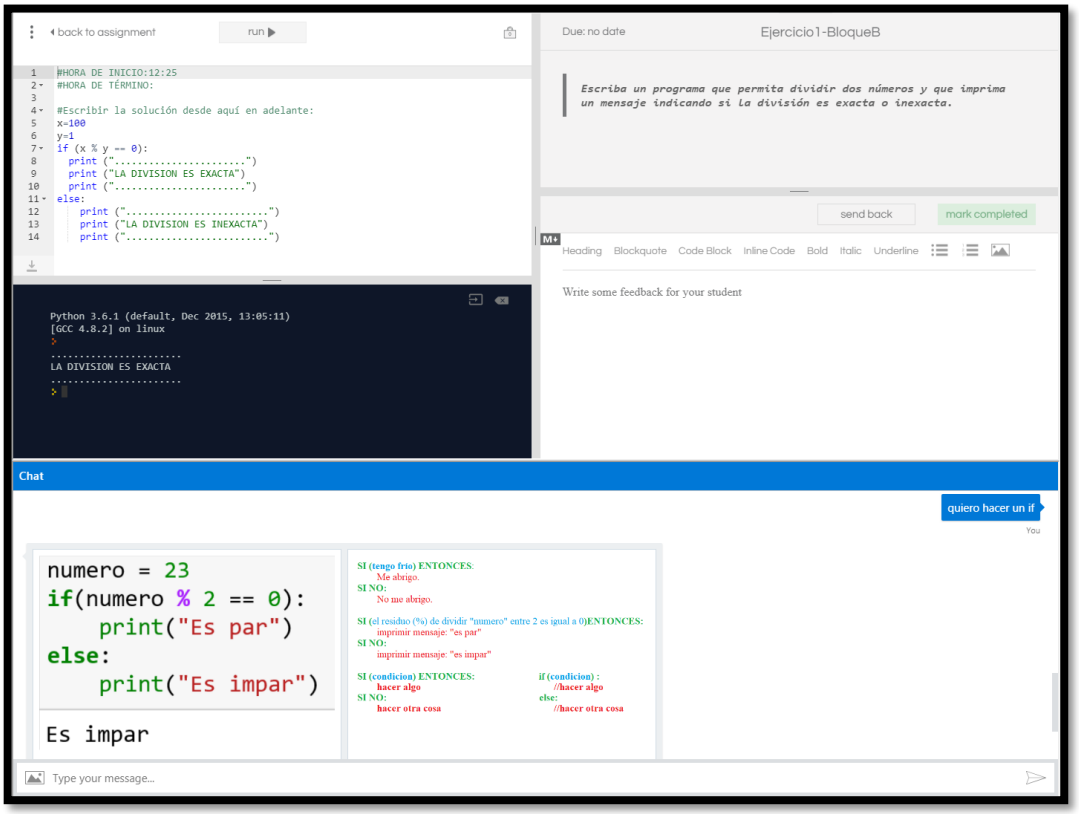

Figura 24: Primer ejercicio del Pos test

Fuente: Elaboración Propia