

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS



TEMA:

DISEÑO Y DESARROLLO DE UN ASISTENTE VIRTUAL PARA REDUCIR LA
DESINFORMACIÓN SOBRE LOS TRÁMITES DE LA FACULTAD DE INGENIERÍA
(PUCE).

AUTOR:

ESPINOZA CHICAIZA ISAAC ALEJANDRO

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TECNOLOGÍAS DE LA INFORMACIÓN

QUITO, Julio - 2022

DEDICATORIA

Realizar la dedicatoria personal máximo párrafo de 15 líneas

AGRADECIMIENTO

Realizar el agradecimiento personal máximo párrafo de 15 líneas

RESUMEN

En la actualidad, las herramientas tecnológicas son utilizadas cada vez más por las empresas para generar beneficios como la automatización de sus procesos o mejorar su competitividad. Un asistente virtual es ese tipo de herramienta que cada vez tiene mayor impacto en el ambiente laboral ya que las empresas necesitan eficiencia en sus procesos y un agente virtual puede servir de solución digital para esta problemática.

El presente proyecto está enfocado en el diseño y desarrollo de un prototipo de un asistente virtual para gestionar la información sobre trámites de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador. El agente virtual se llevó a cabo con la metodología de investigación aplicada con métodos de muestreo selectivo fue la base para obtener la información necesaria que formaría parte de la base de conocimiento del agente virtual. Para desarrollar el sistema experto, se utilizó un flujo de proceso para el desarrollo de inteligencia artificial y una hoja de ruta para el desarrollo del modelo de IA.

En conclusión, el presente proyecto ofrece un asistente virtual para mejorar y facilitar el modo en que se entrega la información de trámites de la facultad de ingeniería de la PUCE y reducir una posible desinformación entre los estudiantes.

ABSTRACT

Nowadays, technological tools are more and more used by companies to generate benefits such as the automation of their processes or improve their competitiveness. A virtual assistant is that kind of tool that has had an increasingly impact on the work environment because companies need efficiency in their processes and a virtual agent can be useful as a digital solution for this problem.

This project is focused on the design and development of a prototype of a virtual assistant to manage information on procedures of the engineering faculty of the “Pontificia Universidad Católica del Ecuador”. The virtual agent was made with the applied research methodology with selective sampling methods, which was the basis for obtaining the necessary information that would be part of the virtual agent's knowledge base. To develop the expert system, a process flow for the development of artificial intelligence and a roadmap for the development of the AI model were used.

In conclusion, this project offers a virtual assistant to improve and ease the way in which the information on procedures of the PUCE engineering faculty is delivered and reduce possible misinformation among students.

ÍNDICE

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS.....	IV
ÍNDICE DE FIGURAS.....	IV
ÍNDICE DE GRÁFICOS	V
ÍNDICE DE CUADROS.....	V
CAPÍTULO I: INTRODUCCIÓN	1
1. INTRODUCCIÓN	1
1.1. Planteamiento del problema.....	2
1.2. Objetivos	3
1.2.1. Objetivo General.....	3
1.2.2. Objetivos Específicos	3
1.3. Justificación.....	4
1.4. Alcance.....	5
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA	6
2. MARCO DE REFERENCIA	6
2.1. Antecedentes	6
2.2. Marco Teórico.....	7
2.3. Marco Conceptual	13

2.4. Marco Espacial	20
CAPÍTULO III: METODOLOGÍA	22
3. METODOLOGÍA	22
3.1. Metodología de Investigación	22
3.2. Metodología Informática.....	23
CAPÍTULO IV: DESARROLLO.....	26
4. DESARROLLO	26
4.1. Definición de Procesos.....	26
4.2. Definición de Reglas de Negocio.....	40
4.3. Desarrollo del Asistente Virtual.....	52
Flujo del Proceso para el Desarrollo de Inteligencia Artificial	52
Backend	63
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	73
5. CONCLUSIONES Y RECOMENDACIONES.....	73
5.1. CONCLUSIONES	73
5.2. RECOMENDACIONES	74
BIBLIOGRFÍA	75
GLOSARIO DE TÉRMINOS.....	79
ANEXOS.....	80
Anexo A: Formato para la solicitud de trámites	80

Anexo B: Archivo de Información del Sistema Experto Milu..... 81

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

ÍNDICE DE FIGURAS

Figura 1 Elementos de Prolog.....	16
Figura 2 Flujo del Proceso para el Desarrollo de Inteligencia Artificial	18
Figura 3 Roadmap para el desarrollo del modelo de Inteligencia Artificial.....	20
Figura 4 Archivo simplification.rules - Formato de las reglas	60
Figura 5 Entorno de trabajo de SWI-Prolog	62
Figura 6 Archivo milu.pl – Sección de código: Interfaz entrada/salida	64
Figura 7 Archivo milu.pl – Sección de código: Funciones básicas	65
Figura 8 Archivo milu.pl – Sección de código: Funciones básicas	66
Figura 9 Archivo milu.pl – Sección de código: Funciones básicas	66
Figura 10 Archivo milu.pl – Sección de código: Función Milu	67
Figura 11 Archivo milu.pl – Sección de código: Función Milu	68
Figura 12 Archivo milu.pl – Sección de código: Función Milu	68
Figura 13 Archivo milu.pl – Sección de código: Función Milu	69
Figura 14 Archivo milu.pl – Sección de código: Función Principal	70
Figura 15 Archivo reply.rules - Reglas.....	71
Figura 16 Archivo reply.rules - Reglas.....	71
Figura 17 Archivo reply.rules - Reglas.....	72
Figura 18 Archivo simplification.rules – Reglas de simplificación	72

ÍNDICE DE GRÁFICOS

Gráfico 1 Diagrama de Flujo – Certificado de Notas	27
Gráfico 2 Diagrama de Flujo – Certificado Estudiante Regular	28
Gráfico 3 Diagrama de Flujo – Rendir Examen Atrasado	29
Gráfico 4 Diagrama de Flujo – Recalificación	30
Gráfico 5 Diagrama de Flujo – Cambio de Carrera	31
Gráfico 6 Diagrama de Flujo – Cambio Modalidad de Titulación	32
Gráfico 7 Diagrama de Flujo – Copia Acta de Grado.....	33
Gráfico 8 Diagrama de Flujo – Syllabus de Materias Aprobadas.....	34
Gráfico 9 Diagrama de Flujo – Solicitud de Información sobre Titulación	35
Gráfico 10 Diagrama de Flujo – Reingreso a la Carrera.....	36
Gráfico 11 Diagrama de Flujo – Retiro de Materia	37
Gráfico 12 Diagrama de Flujo – Toma de Materia en otra Facultad	38
Gráfico 13 Diagrama de Flujo – Proceso Secretaría de la Facultad	39

ÍNDICE DE CUADROS

Cuadro 1 Cédula de la Entrevista	53
---	----

CAPÍTULO I: INTRODUCCIÓN

1. INTRODUCCIÓN

Los estudiantes dentro de las universidades del Ecuador deben realizar diferentes tipos de gestiones para que se lleven a cabo procesos administrativos necesarios dentro de la institución. Para conocer el tipo de gestión y cómo se realizarla, solicitan información en las oficinas de la facultad correspondiente. Sin embargo, esta información no suele ser fácil de conseguir ya que se encuentra dispersa y la gestión de trámites resulta ser un proceso de tiempo prolongado. La Pontificia Universidad del Ecuador se encuentra inmersa en este contexto donde las soluciones tecnológicas pueden generar beneficios para la institución. Precisamente mejorar la eficiencia de procesos y mejorar la competitividad, son algunos beneficios que caracterizan a este tipo de soluciones y son la razón por la cual las empresas e instituciones apuestan para reducir la brecha tecnológica que existe en el país.

Una de las áreas donde se han producido más avances tecnológicos ha sido la inteligencia artificial y cada vez va adquiriendo mayor relevancia debido a la variedad de campos en los que está siendo aplicada. Los asistentes virtuales son una solución que está basada en inteligencia artificial y llegará a tener un mayor impacto en el ámbito laboral, ya que, las empresas e instituciones conocen los beneficios de automatización y eficiencia que estos pueden ofrecer.

Entonces, el objetivo de este trabajo es precisamente proponer una solución con un asistente virtual para la problemática de la gestión de trámites. Esta propuesta permitirá disponer a los estudiantes de la información necesaria sobre los trámites de manera eficiente y

centralizada para así ofrecer un mejor servicio a los estudiantes de la facultad de ingeniería que significará un beneficio a los intereses de la Pontificia Universidad Católica del Ecuador.

1.1. Planteamiento del problema

La información de los trámites dentro de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador no suele ser sencilla de conseguir y muchas veces se encuentra dispersa en varias oficinas dentro del campus de la institución. Empezar a generar soluciones a este problema resultará en una mejor atención al estudiante que significará un beneficio a los intereses de la institución educativa. Los estudiantes serían los principales beneficiarios de tener la información de trámites centralizada, ya que, podrían agilizar sus pendientes con los trámites y evitar que el tiempo para realizar estas gestiones se prolongue.

La dificultad de conseguir información sobre trámites provoca un retraso para que los estudiantes de la facultad de ingeniería puedan realizar esta gestión y en algunos casos este problema puede llegar a la desinformación. El asistente virtual propuesto en el presente proyecto funcionará como solución para disponer a los estudiantes de la información necesaria sobre los trámites de manera eficiente y centralizada. El sistema permitirá a los estudiantes realizar preguntas sobre lo que se necesite saber sobre los trámites de la facultad. El agente proporcionará una respuesta óptima, generando de esta manera un mejor servicio para los estudiantes.

Con respecto a la problemática se expone la siguiente pregunta principal:

- ¿De qué manera un asistente virtual puede reducir la desinformación sobre los trámites de la facultad de ingeniería de la PUCE?

Y las siguientes preguntas secundarias:

- ¿Puede una solución digital solventar el problema de desinformación de trámites?
- ¿Es un asistente virtual la manera más óptima de informar a los estudiantes de los trámites?
- ¿Es PROLOG el lenguaje de programación más adecuado para el desarrollo de un asistente virtual?

1.2. Objetivos

1.2.1. Objetivo General

Diseñar y desarrollar un asistente virtual para reducir la desinformación sobre los trámites de la facultad de ingeniería (PUCE).

1.2.2. Objetivos Específicos

- Analizar lenguajes o herramientas de desarrollo para Inteligencia Artificial.
- Documentar la información de los trámites que se requieran de la facultad de ingeniería (PUCE).
- Desarrollar un asistente virtual que facilite la información de trámites de la facultad de ingeniería (PUCE).
- Documentar el proceso metodológico para el desarrollo de un asistente virtual con Inteligencia Artificial.

1.3. Justificación

Desde la antigüedad ha estado presente la importancia de darle valor y gestionar la información correctamente. Hoy en día, las instituciones y empresas utilizan herramientas de software para la gestión de la información y sus procesos, sin embargo, en algunos casos no se ha hecho uso de soluciones digitales. Entonces, existen procesos internos y externos que, enfocados a los intereses de las empresas e instituciones, se pueden generar este tipo de soluciones (García-Holgado & García-Peñalvo, 2015).

La Inteligencia Artificial (IA) cada vez tiene mayor relevancia dentro de numerosos campos como las finanzas, domótica, distribución de recursos, reconocimiento facial o mercadotecnia. Cada vez se realizan más aplicaciones de IA en estos y muchos otros campos debido a que se automatizan procesos que suelen llevarse a cabo de forma manual y se consigue mejor eficiencia en estos. Un ejemplo de ello, son los asistentes virtuales utilizados para la gestión. Esta forma de uso de la IA permite clasificar documentos o información de una manera más eficiente lo que se puede utilizar para generar soluciones digitales (Mayer-Pujadas, Ávila-Tomás, & Quesada-Varela, 2021).

Los asistentes virtuales cada vez se han ido utilizando más ya que utilizan agentes inteligentes fáciles de usar cuando el usuario y el asistente interactúan. La tecnología ha ido avanzando, lo que ha permitido el desarrollar asistentes virtuales con mayor compatibilidad y variedad de funcionalidades. Es muy utilizado el método de interacción por diálogo en tiempo real ya que se provee de información al usuario de manera rápida y comprensible. Sin embargo, el método de interacción puede variar y los asistentes virtuales continúan siendo un beneficio por la manera en la que interactúan con los usuarios. En el escenario de un asistente virtual en el hogar, el método de interacción más asertivo sería por diálogo en tiempo real. Por otro lado,

en el área de trabajo se podría apostar por un método de interacción por texto (González Ávila, López Martínez, & Hernández García, 2020).

La automatización del proceso de obtener información sobre trámites de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador es uno de los principales beneficios que puede proporcionar un agente virtual. Se puede obtener un control sobre la información brindada a los estudiantes, además de que esta tecnología significará un ahorro de tiempo y aumento de productividad en las actividades que los estudiantes necesiten realizar dentro de la facultad. De esta manera, la institución ofrecerá una mejor atención a sus estudiantes y satisfacer sus necesidades.

1.4. Alcance

El presente proyecto tiene como alcance el **Diseño y Desarrollo de un asistente virtual para reducir la desinformación sobre los trámites de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador** porque no se ha encontrado propuesta de solución similar dentro de la facultad de la institución. Se realizará este trabajo de tipo práctico para solucionar la dificultad de conseguir información sobre trámites de la facultad de ingeniería que provoca un retraso para que los estudiantes puedan realizar esta gestión. El agente podrá generar respuestas óptimas de las preguntas que realicen los estudiantes con relación a los trámites de la facultad.

Como resultado de este trabajo se entregará el asistente virtual desarrollado listo para ser usado por usuarios finales de la facultad de ingeniería de la PUCE. El sistema será intuitivo para que los usuarios puedan hacer uso del agente.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. MARCO DE REFERENCIA

2.1. Antecedentes

La inteligencia artificial es una de las áreas donde se han producido más avances en los últimos tiempos. Su relación con los asistentes virtuales empezó porque se requería mejorar la eficiencia de tareas de manera autónoma para que pueda el usuario realizar otras. No se conoce el verdadero impacto que han tenido la sociedad debido a que es una tecnología creciente y no se conoce de suficientes estudios. Se sabe que el uso de los asistentes virtuales irá incrementando con los años y llegará a tener un mayor impacto en el ámbito laboral (Jiménez Álvarez, 2019).

Se tomará en consideración investigaciones donde se ha aplicado un asistente virtual para la resolución de una problemática:

La tesis realizada por (Martínez Carpio, 2019) está enfocada en el diseño de un asistente virtual con machine learning que automatice los canales de atención, mejorando así la atención al cliente en el apartado comercial. Este asistente virtual optimiza los tiempos de respuesta para mejorar el servicio al cliente y así aumentar las ventas y su rentabilidad. Se utilizó la metodología ágil *Scrum* para atender los requerimientos del cliente mientras se esté desarrollando el proyecto, proporcionando así flexibilidad al proceso. Este trabajo concluyó en que la automatización de las respuestas para la interacción con el cliente generó mejoras en el proceso de comunicación y ventas.

En la siguiente tesis, (Abata Quinchuqui & Ramírez Soria, 2018) se planteó el desarrollo de un asistente virtual para la gestión del conocimiento de una organización donde la información no se encuentra estructurada adecuadamente, lo que impide ofrecer la información importante para la toma de decisiones. Con la metodología XP las iteraciones se trabajaron junto al cliente, es decir, para validar cada prototipo se probó con un usuario que se encuentre dentro de cada escenario. Además, dentro del desarrollo se evidenció la simplicidad, pruebas y corrección de errores con la metodología utilizada. Se concluyó que con una precisión del 86 y 88%, se consiguió generar respuestas oportunas para el usuario. Una manera pertinente de presentar la información es manteniendo el hilo de la conversación con el usuario.

2.2. Marco Teórico

(Rouhiainen, 2018) define a la Inteligencia Artificial (IA) como la capacidad de los ordenadores para realizar actividades que generalmente necesitan la inteligencia humana. Es la capacidad de los ordenadores para aprender a tomar decisiones haciendo uso de los datos de los algoritmos. En la actualidad la IA está siendo aplicada en diversas situaciones como para procesar eficiente y escalablemente los datos en la atención médica, mantenimiento predictivo en sectores industriales, para distribuir el contenido en las redes sociales y también es utilizada para seguridad contra amenazas cibernéticas. Además, la IA tiene la capacidad de ofrecer sugerencias y también predicciones por lo que tendrá más impacto en áreas de la salud, la educación, el bienestar, las relaciones interpersonales y en el ámbito laboral.

Dentro del campo de la Inteligencia Artificial existen diversas técnicas, pero de las más importantes encontramos a los sistemas expertos. Estos sistemas capturan el conocimiento de un área específica o de gran complejidad. Otra técnica de la IA son las de aprendizaje

automático (Machine Learning) donde se intenta deducir conocimiento a partir de datos simulando el razonamiento lógico. Las redes neuronales también son una técnica y rama de investigación del aprendizaje automático que tienen una gran capacidad de predecir y de adaptarse a ambientes cambiantes (Suárez, 2000). En la actualidad, estas y muchas otras técnicas son usadas para dar apoyo a funciones básicas de manejo de información (Badaró, Agüero, & Agüero, 2013).

Los asistentes virtuales son sistemas que están basados en el conjunto de la inteligencia humana y la artificial con los que los usuarios interactúan en lenguaje natural. El beneficio de estos asistentes es que son considerados una herramienta de mucha utilidad para cualquier tarea asignada (González Ávila, López Martínez, & Hernández García, 2020). Además, tienen cierta habilidad para aprender en relación con las consultas que los usuarios realizan con el fin de que las respuestas ofrecidas se adapten a sus necesidades (Jiménez Álvarez, 2019).

En los últimos años los asistentes virtuales han ido tomando mayor relevancia porque permiten la interacción con el usuario con un lenguaje natural. Con el simple hecho de mantener una conversación, los asistentes virtuales tienen la capacidad de ofrecer información de una manera muy eficiente. Las empresas más grandes del mercado están apostando por implementar la tecnología de un asistente virtual y llevarla a un nivel más avanzado, desarrollando nuevas funcionalidades que perfeccionen las respuestas que el asistente puede generar (Guijarro, 2020).

(Medina, Eisman, & Castro, 2013) definen los siguientes tipos de asistentes virtuales. Primero, los de propósito general que generan expectativa gracias a grandes proyectos como *Google Now* (Google) o *Siri* (Apple); poseen utilidades para conocer ubicaciones, envío de mensajes de voz o conocer el tiempo. Segundo, los asistentes virtuales de propósitos específicos se encargan de la resolución de problemas sobre una propiedad en específico como puede ser

una institución, un museo o una tienda, donde el asistente es capaz de dar solución a problemas in situ.

Scrum es una forma para la gestión de proyectos en donde se destaca la flexibilidad y la agilidad en el proceso. Se caracteriza por tener incertidumbre, es decir, se plantea un objetivo a donde dirigir el proyecto sin necesidad de tener un plan detallado. Los equipos se caracterizan por su autonomía, autosuperación y auto-enriquecimiento donde también se producirá la transmisión de conocimiento. Las fases para el desarrollo de Scrum están basadas en el ciclo de vida de desarrollo ágil:

- **Planificación de un Sprint (Sprint Planning):** El equipo realiza la planificación donde se definen de tareas que se deben realizar y los objetivos.
- **Reunión de Equipo (Sprint team meeting):** El equipo comparte información con relación al desarrollo y se realizan las adaptaciones necesarias. Se responden las preguntas:
 - ¿Qué sucedió en última reunión?
 - ¿Qué sucederá el día de hoy?
 - ¿Qué inconvenientes pueden interferir con las tareas?
- **Refinamiento (Backlog Refinement):** Estimación del tiempo dedicado a cada objetivo.
- **Revisión (Sprint Review):** El equipo y los stakeholders (interesados) se reúnen para presentar los avances y tomar decisiones alineadas a los objetivos.
- **Retrospectiva:** Se reúne el equipo para debatir lo ocurrido en el Sprint. Se analizan los errores cometidos, los aciertos y los inconvenientes que impidieron avanzar con lo planificado (Trigas Gallego, 2012).

El flujo de proceso para el desarrollo de Inteligencia Artificial está basado en el flujo del Big Data. Se debe definir el objetivo de negocio, seleccionar la fuente de datos para poder gestionarlos con el proceso ETC (Extracción, Transformación, Carga), almacenar los datos para posteriormente analizarlos y aplicar la inteligencia artificial con sus respectivas técnicas, se aplica Business Intelligence para darle utilidad a los datos y finalmente poder visualizarla en la última fase del flujo de proceso. Este se detalla a continuación:

1. Objetivo de Negocio

Identificar la mejor opción orientada al negocio de la empresa. La importancia final del proceso radica en el uso que se dará a la información para ser considerada valiosa para los objetivos del negocio y generar beneficios y ventajas para la empresa.

2. Elección de Fuente de Datos

Recopilar los datos que proceden de orígenes diferentes. Los datos se pueden generar por distintos individuos, empresas y máquinas. Se puede distinguir 5 tipos de fuentes de manera genérica:

- Datos generados en la web y medios sociales.
- Información transmitida en Machine to machine (M2M).
- Datos de transacciones bancarias, registros de teléfono, etc.
- Datos biométricos que comprende a datos generados por personas como escáner de retina, huellas dactilares, etc.
- Datos generados por la gente en el uso de las tecnologías.

3. Gestión de Datos (ETC, Extracción, Transformación, Carga)

Organizar los datos según el proceso estandarizado ETC. En la primera tarea se recopila datos que proceden de distintos orígenes y unificar esta información para ser usada en la siguiente tarea. La fase de transformación se llevará a cabo según los requerimientos de la empresa, donde se corrigen y detectan inconsistencias como errores de sintaxis, semánticos o por información incompleta para asegurar la calidad de los datos. En la fase de carga de datos se definen aspectos como el dónde serán almacenados o el modelo de integración de los datos.

4. Almacenaje

Carga de datos a repositorios como data warehouse o almacenaje en la nube. La información se encuentra representada en una inmensa cantidad de datos que una vez organizada se almacenan y depende del volumen para decidir el tamaño del almacén con el que se va a trabajar.

5. Analítica de Datos

Análisis de la información generada y generación de reglas para la Inteligencia Artificial. La importancia de esta fase tiene que ver con que se puede convertir la información en estrategias de negocio que se pueden realizar o tomar decisiones importantes para el futuro de la empresa.

6. Capa de Business Intelligence

Toma de decisiones en base a la información orientada a los objetivos de la empresa. Se tienen que trabajar los datos y se generan consultas para obtener diferentes escenarios buscando el más favorable y así poder tomar decisiones de negocio. Además de escenarios y decisiones también se pueden generar predicciones o planificar estrategias.

7. Visualización

Representar gráficamente los datos para que sean entendibles por el usuario. Esta fase cada vez va tomando mayor relevancia para poder entender procesos complejos, es decir, hacer entendibles unos datos trabajados con estadística y matemática desde un punto de vista empresarial (Fernández, 2016).

Un Roadmap es una sucesión de pasos a seguir para conseguir un objetivo. Este proporciona una visión general del desarrollo y de esta manera poder informar a los interesados del progreso de un proyecto y su alcance. Funciona de herramienta útil para las empresas ya que conocen en qué parte del proyecto se encuentran y hacia dónde se quiere dirigir el proyecto. Un Roadmap para el desarrollo de un modelo de Inteligencia Artificial será de ayuda para dar seguimiento al ciclo de desarrollo de un asistente virtual (da Silva, 2021).

El lenguaje *Prolog* nació en los años 70 en la Universidad de Edinburgh. Es utilizado para la implementación de sistemas expertos e inteligencia artificial en diversos campos como telecomunicaciones, finanzas, medicina, ingeniería, agricultura, educación y manufactura. *Prolog* está orientado a la programación lógica, donde se realiza un enfoque en qué hacer para resolver el problema, más no en el cómo se resuelve. La popularidad de este lenguaje se debe a

su capacidad de deducción y su sencillez en cuanto a sintaxis y semántica. Como es un lenguaje interpretado y declarativo, busca relación entre los objetos, variables, listas y reglas creadas. Al momento de realizar una pregunta, el programa deduce la solución en la base datos gracias a las reglas previamente establecidas. Cuando *Prolog* es ejecutado, el comportamiento puede asimilarse a una búsqueda dentro de un árbol para obtener las posibles soluciones (Jones Pérez & Yong Morales, 2007).

2.3. Marco Conceptual

Asistente virtual

Los asistentes virtuales generan soporte a múltiples clientes en una variedad de industrias que proveen servicios técnicos, administrativos o creativos; desde una ubicación remota (International Virtual Assistants Association, 2022).

Un asistente en informática es una “aplicación informática que guía al usuario inexperto en el manejo de un programa”. Es decir, puede generar una solución a un problema que el usuario solicite (Real Academia Española, 2022).

Un asistente virtual (digital) es una aplicación que es accionada por inteligencia artificial y lleva a cabo tareas para los usuarios. Uno de sus usos es para buscar información, por ejemplo, los asistentes de voz que ayudan a la búsqueda de información en el Internet (Brita Inteligencia Artificial, 2020).

El agente virtual propuesto en este proyecto ofrecerá información sobre los trámites de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador de manera eficiente.

En conjunto con la Inteligencia Artificial, se le designará la tarea de ofrecer soluciones a las preguntas que el usuario final le realizará.

Lenguajes de programación

Un lenguaje de programación es un sistema notacional utilizado para la descripción de cálculos de una manera legible para el ser humano y para la máquina. Pueden asimilarse a una interfaz entre el usuario y la máquina, por lo que se maneja una sintaxis y semántica de las declaraciones que permite el lenguaje (Gortázar Bellas, Martínez Unanue, & Fresno Fernández, 2016).

Los lenguajes de programación han ido evolucionando y hoy en día se pueden clasificar en cinco generaciones:

- **Primera Generación:** Se le atribuye al lenguaje de máquina, este reside en secuencias de unos y ceros; las computadoras modernas entienden este tipo de lenguaje.
- **Segunda Generación:** Pertenecen los lenguajes ensambladores. Estos emplean reglas para facilitar la escritura y lectura de programas. Se utilizan para la programación de drivers o ciertos fragmentos de sistemas operativos.
- **Tercera Generación:** Se le atribuye los lenguajes de alto nivel (Java, C o FORTRAN). Son más distantes de un lenguaje para máquina, pero más entendibles para el hombre. Utilizados para la programación de grandes sistemas software.

- **Cuarta Generación:** Son lenguajes que tienen un motivo específico (ABAP o SQL). Están diseñados para dar solución a problemas concretos. Como muestra de, SQL describe consultas, modificaciones o inserciones en bases de datos.
- **Quinta Generación:** Su principal uso yace en el campo de la IA (Inteligencia Artificial). Permiten la especificación de restricciones que se le demuestra a un sistema para resolver un problema (Mercury o PROLOG) (Gortázar Bellas, Martínez Unanue, & Fresno Fernández, 2016).

El lenguaje de programación que se utilizará para el desarrollo del presente proyecto es PROLOG. Este lenguaje pertenece a la Quinta Generación para el desarrollo de Inteligencia Artificial.

Lenguajes para el desarrollo de inteligencia artificial

La Quinta Generación de los lenguajes de programación se desempeña en el campo de la inteligencia artificial. Las funciones que cumplen estos lenguajes están basadas en tres núcleos como son la inteligencia, la facilidad de empleo y la fiabilidad-seguridad. La inteligencia se refiere al conocimiento y acceso a los datos conceptuales. La facilidad de empleo tiene que ver con la capacidad que tiene el sistema para entenderse con el usuario y las facilidades que le puede generar en su uso. La fiabilidad y seguridad son el mejoramiento de los métodos de verificación de ejecución automática de programas, mecanismos de seguridad y las funciones de decisión (Sánchez, 2009).

Como lenguajes básicos de Inteligencia Artificial se puede mencionar a Prolog (declarativo), Lisp (funcional) y SmallTalk (orientado a objetos). Prolog es parte de los

lenguajes declarativos y lógicos. La principal diferencia entre estos lenguajes y los más populares para la programación como pueden ser Java, C, Fortran o Pascal; es que los lenguajes declarativos son razonados de manera matemática, esto le permite usar mecanismos matemáticos que pueden optimizar el rendimiento de los programas (Gil Suárez & Martínez Buitrago, 2013).

Prolog (Programmation en Logique) es un lenguaje muy útil para el proceso de sistemas expertos ya que es descriptivo. Se posee una base específica de conocimiento que puede manipular para generar conclusiones. La base de conocimiento se compone de reglas y hechos; para dar solución al problema se pueden realizar preguntas y su respuesta se encuentra en dicha base. Si un nuevo programador desea hacer uso de Prolog, podrá interpretar la lógica de manera clara y comprender el entorno de un nuevo programa de manera sencilla ya que la estructura de este lenguaje es directa (Gil Suárez & Martínez Buitrago, 2013).

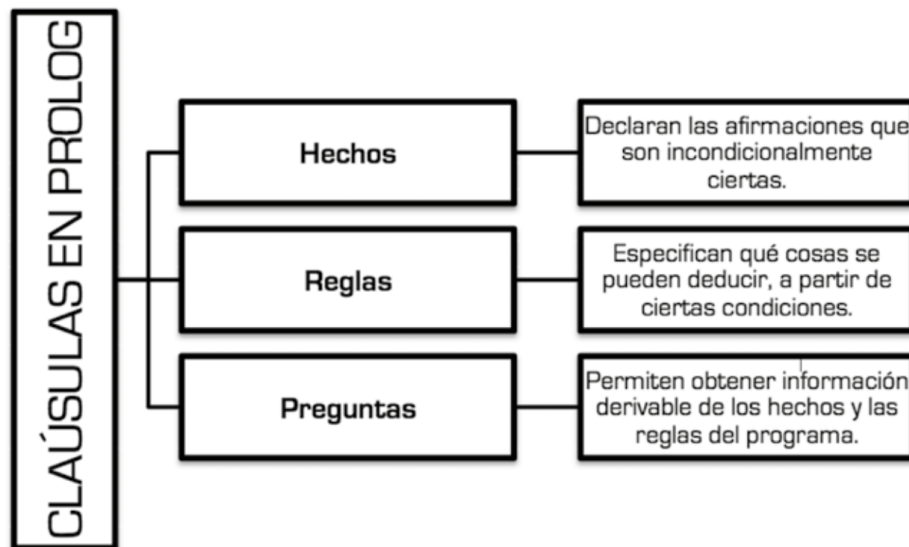


Figura 1 Elementos de Prolog (Gil Suárez & Martínez Buitrago, 2013).

En el desarrollo del asistente virtual se utilizará el lenguaje de programación PROLOG. Se considera este lenguaje debido a que se pueden generar programas legibles. También facilita el desarrollo al programador porque, frente a lenguajes populares como C++ o Java, Prolog infiere sus respuestas con respecto a las reglas que se declaran en el programa y el programador puede encontrar una solución de manera más sencilla (Jones Pérez & Yong Morales, 2007).

Para la gestión del proceso se utilizará Scrum que aportará la flexibilidad y agilidad necesaria al proyecto. Se aplicará las fases de Scrum que se encuentran basadas en el ciclo de vida de desarrollo ágil:

- **Planificación de un Sprint (Sprint Planning):** Se realizará una planificación para definir las tareas a realizar y los objetivos que guiarán el desarrollo del proyecto.
- **Reunión de Equipo (Sprint team meeting):** Se realizarán las adaptaciones necesarias en el desarrollo para cumplir con los objetivos planteados. Se responderán las preguntas:
 - ¿Qué se sucedió en última reunión?
 - ¿Qué se sucederá el día de hoy?
 - ¿Qué inconvenientes pueden interferir con las tareas?
- **Refinamiento (Backlog Refinement):** Se estimará el tiempo que se le ha dedicado a cada objetivo para alinear los tiempos reales con los tiempos proyectados.
- **Revisión (Sprint Review):** Se reunirá con los interesados o stakeholders para presentar los avances del desarrollo, alinear el proceso a los objetivos y cambiar el rumbo del proyecto en caso de ser necesario.

- **Retrospectiva:** Se analizarán los errores cometidos, los aciertos y los inconvenientes que impidieron avanzar con lo planificado (Trigas Gallego, 2012).

Se realizará el proceso de desarrollo de Inteligencia Artificial según el flujo de proceso:

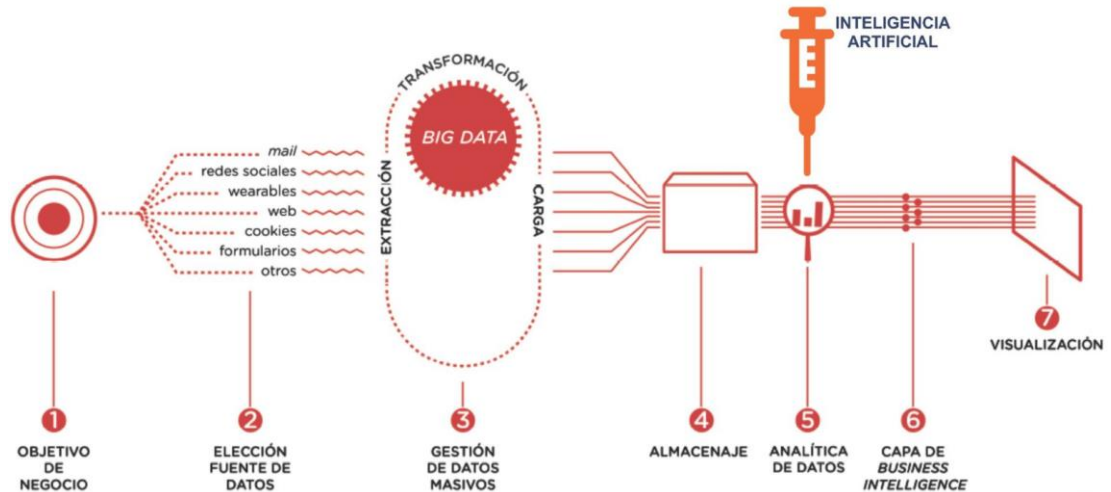


Figura 2 Flujo del Proceso para el Desarrollo de Inteligencia Artificial (Fernández, 2016).

1. **Objetivo de Negocio:** Se identificarán los intereses de la Pontificia Universidad Católica del Ecuador para establecer el objetivo al que se orientará el desarrollo del asistente virtual.
2. **Elección de Fuente de Datos:** La Fuente de los Datos es la información sobre trámites de facultad de ingeniería de la PUCE, donde se podrá definir cada proceso necesario para preparar los datos que serán utilizados en el desarrollo del agente virtual.

3. **Gestión de Datos (ETC, Extracción, Transformación, Carga):** Para el proceso de Extracción se definirán cada uno de los procesos, de esta manera la información estará unificada y lista para la siguiente fase. En la fase de transformación se analizará la información documentada. Se detectará y corregirá los errores en los procesos para generar información lo más completa posible y de calidad. Para la fase de carga se definirá el almacén de acuerdo con el volumen de la información documentada.
4. **Almacenaje:** Se cargarán los datos al repositorio o almacén correspondiente.
5. **Analítica de Datos:** Se definirán las reglas de negocio que mandarían en la inteligencia artificial del agente virtual.
6. **Capa de Business Intelligence:** Para esta fase el agente virtual podrá tomar decisiones orientadas a los objetivos del negocio, en este caso la institución PUCE.
7. **Visualización:** Los datos se representarán gráficamente para que los usuarios/estudiantes puedan empezar a hacer preguntas sobre los trámites de la facultad de ingeniería (Fernández, 2016).

A continuación, se describe cómo se aplicará el Roadmap (Hoja de Ruta) para el desarrollo de modelo de Inteligencia Artificial:

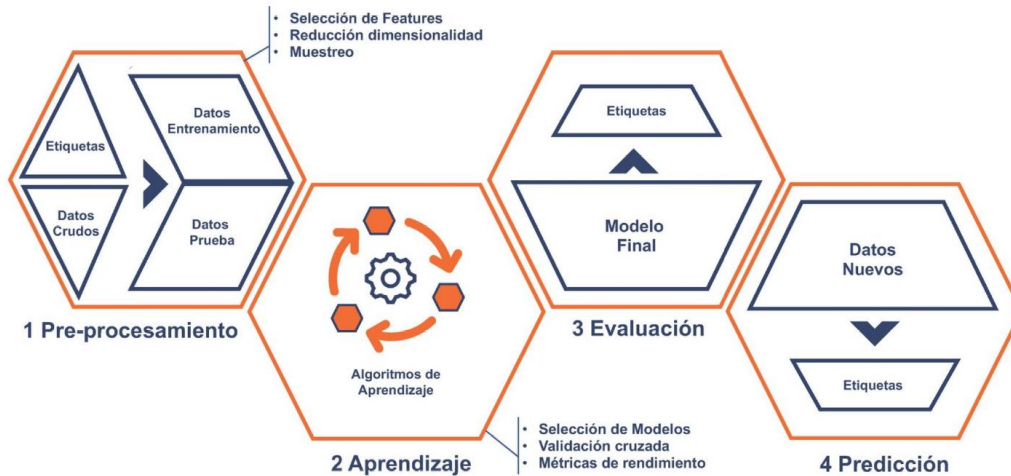


Figura 3 Roadmap para el desarrollo del modelo de Inteligencia Artificial.

- 1. Pre-procesamiento:** Se analizarán los datos para verificar su calidad y estructura antes de incluirlos en el desarrollo.
- 2. Aprendizaje:** Se establecerán las reglas de negocio que aplicará el agente virtual y podrá aprender de ellas.
- 3. Evaluación:** Se evaluará que el agente virtual pueda identificar los patrones en base a lo aprendido.
- 4. Predicción:** El agente virtual podrá predecir datos nuevos, en este caso las respuestas para las consultas que los usuarios realizarán.

2.4. Marco Espacial

El presente proyecto se desarrollará dentro de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador, sede Quito en la provincia de Pichincha. La toma de

información se llevará a cabo dentro de esta facultad para documentarla y poder realizar la base de conocimiento del asistente virtual propuesto.

CAPÍTULO III: METODOLOGÍA

3. METODOLOGÍA

El presente proyecto será desarrollado en base a la metodología de investigación aplicada, marco de referencia Scrum y técnicas de sistemas expertos.

3.1. Metodología de Investigación

La investigación que se llevará a cabo para el desarrollo del asistente virtual es de tipo aplicada. Esta metodología se orienta a la resolución de problemas encontrados en los procesos de distribución, producción y servicios de algún tipo de actividad humana. Este tipo de metodología utiliza como base a la investigación en las ciencias formales o fácticas para formular hipótesis de trabajo que ayuden a la resolución de problemas de la sociedad. El enfoque principal de este tipo de investigación se orienta a la mejora, perfección u optimización del funcionamiento de sistemas, normas, procedimientos, reglas tecnológicas en base a los avances de la tecnología y la ciencia (Nicomedes Nieto, 2018).

Como método se utilizará el muestreo selectivo de informantes clave que facilitará la obtención de información necesaria para el desarrollo del asistente virtual. En este caso los informantes nos proporcionarán la información sobre los trámites de la facultad para poder documentarla.

Se va a emplear la técnica de la entrevista. Esta conversación está fundamentada en una serie de preguntas o afirmaciones para obtener la información de los informantes clave.

El instrumento que se va a utilizar es la cédula de entrevista. Este utiliza preguntas abiertas para que el entrevistado pueda ampliar toda la información que se va a recopilar.

3.2. Metodología Informática

Scrum está basada en la idea de crear ciclos rápidos para el desarrollo llamadas iteraciones que dentro de Scrum se las denomina como “sprints” (Trigas Gallego, 2012). Scrum tiene la característica de modificar funciones con base en una funcionalidad que ya existe. El manejo de tiempos cortos dentro de la metodología permite la mejora continua, además de que durante el desarrollo se escuchan los requerimientos del cliente. La presentación de avances permitirá conseguir un mejor control y desarrollo de la evolución del trabajo para conseguir el mejor posible resultado del proyecto (Martínez Carpio, 2019).

Los sistemas expertos manejan un ambiente de desarrollo y uno de consulta. En el ambiente de desarrollo se crean los componentes y se introduce el conocimiento a la base de conocimientos. En el de consulta se obtiene conocimiento de expertos como pueden ser consejos. Como sus componentes principales se puede considerar a la base de conocimiento que comprende, formula y resuelve problemas; el motor de inferencia que se encarga de interpretar las reglas; el subsistema de justificación que permite al usuario realizar preguntas al sistema, es decir, la interfaz de usuario con la que el sistema interactúa con el usuario (Badaró, Agüero, & Agüero, 2013).

Para el proceso de desarrollo de Inteligencia Artificial, se llevará a cabo según el flujo de proceso ya descrito en la Figura 2:

1. **Objetivo de Negocio:** Identificar la mejor opción orientada al negocio de la empresa.
2. **Elección de Fuente de Datos:** Recopilar los datos que proceden de orígenes diferentes.
3. **Gestión de Datos (ETC, Extracción, Transformación, Carga):** Organizar los datos según el proceso estandarizado ETC.
4. **Almacenaje:** Carga de datos a repositorios como data warehouse o almacenaje en la nube.
5. **Analítica de Datos:** Análisis de la información generada y generación de reglas para la Inteligencia Artificial.
6. **Capa de Business Intelligence:** Toma de decisiones en base a la información orientada a los objetivos de la empresa.
7. **Visualización:** Representar gráficamente los datos para que sean entendibles por el usuario (Fernández, 2016).

Se tomará en consideración el Roadmap (Hoja de Ruta) para el desarrollo de modelo de Inteligencia Artificial ya descrito en la Figura 3. Esta herramienta será de gran utilidad para conocer cómo ayudará la IA al desarrollo del asistente virtual y dar seguimiento al ciclo de vida del proyecto:

1. **Pre-procesamiento:** Se analiza la estructura y la calidad de los datos.
2. **Aprendizaje:** Se establece las reglas que permitirá al asistente virtual realizar el proceso de aprendizaje.
3. **Evaluación:** Interpretación de los patrones.

4. Predicción: Predicción de nuevos datos.

De esta manera, se llevará a cabo el desarrollo del asistente virtual para que los usuarios finales consigan información sobre los trámites de la facultad de ingeniería.

CAPÍTULO IV: DESARROLLO

4. DESARROLLO

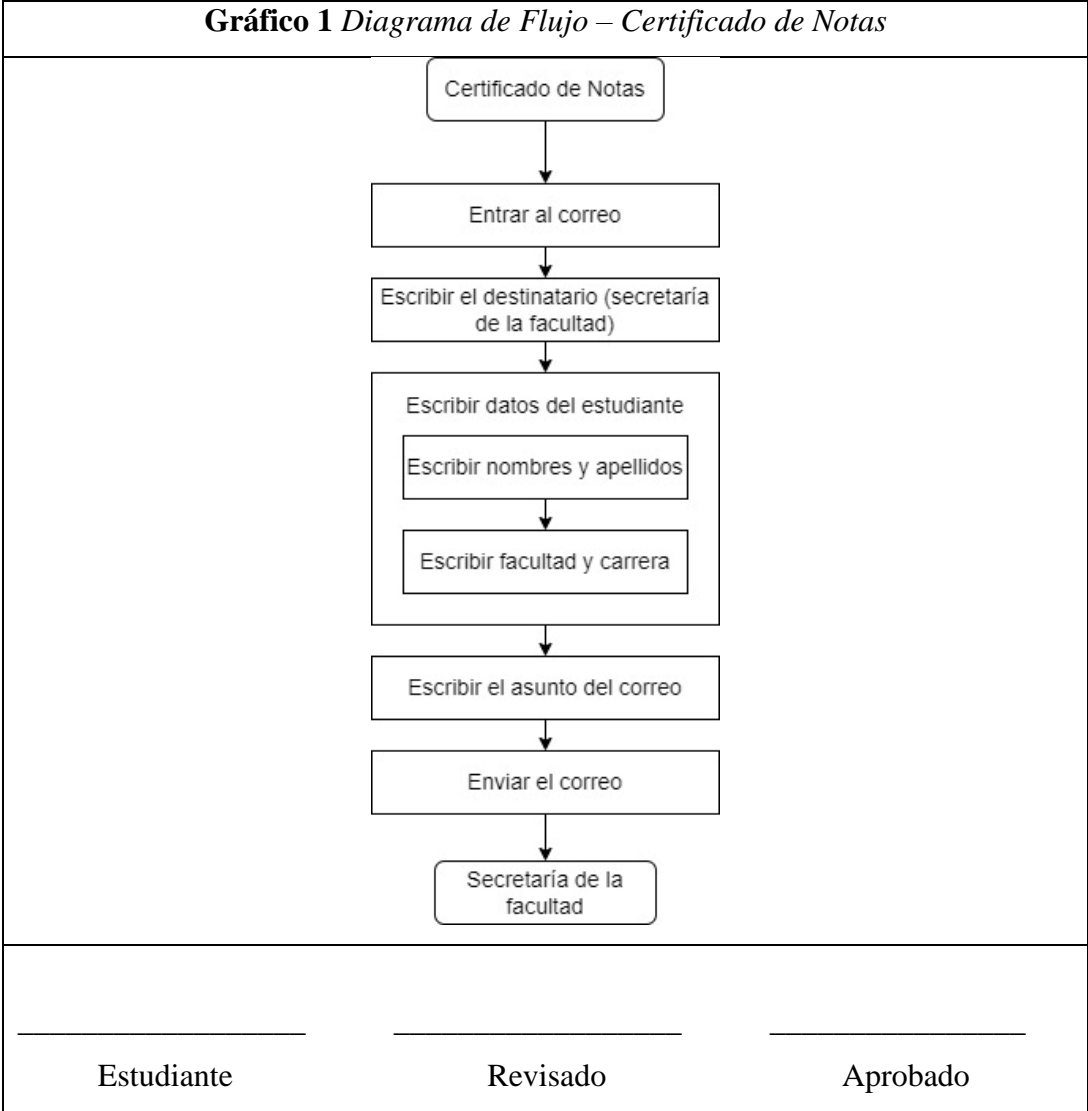
4.1. Definición de Procesos

A continuación, por medio de diagramas de flujo se definirán los procesos de trámites que realizan los estudiantes en la facultad de ingeniería de la PUCE.

1. Certificado de Notas

Código: CN-001

Responsable: Daniel Paredes



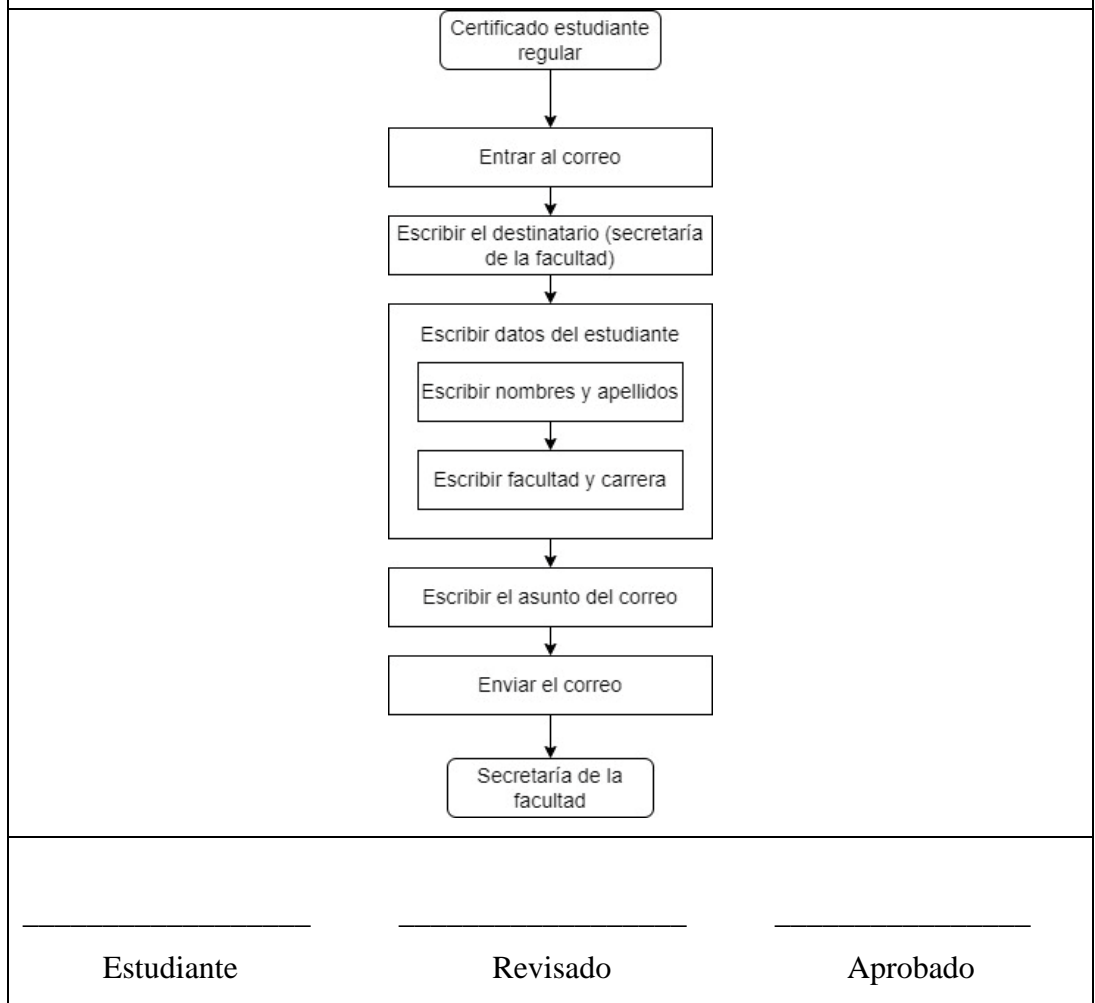
Elaborado por: Isaac Espinoza

2. Certificado Estudiante Regular

Código: CER-001

Responsable: Daniel Paredes

Gráfico 2 Diagrama de Flujo – Certificado Estudiante Regular



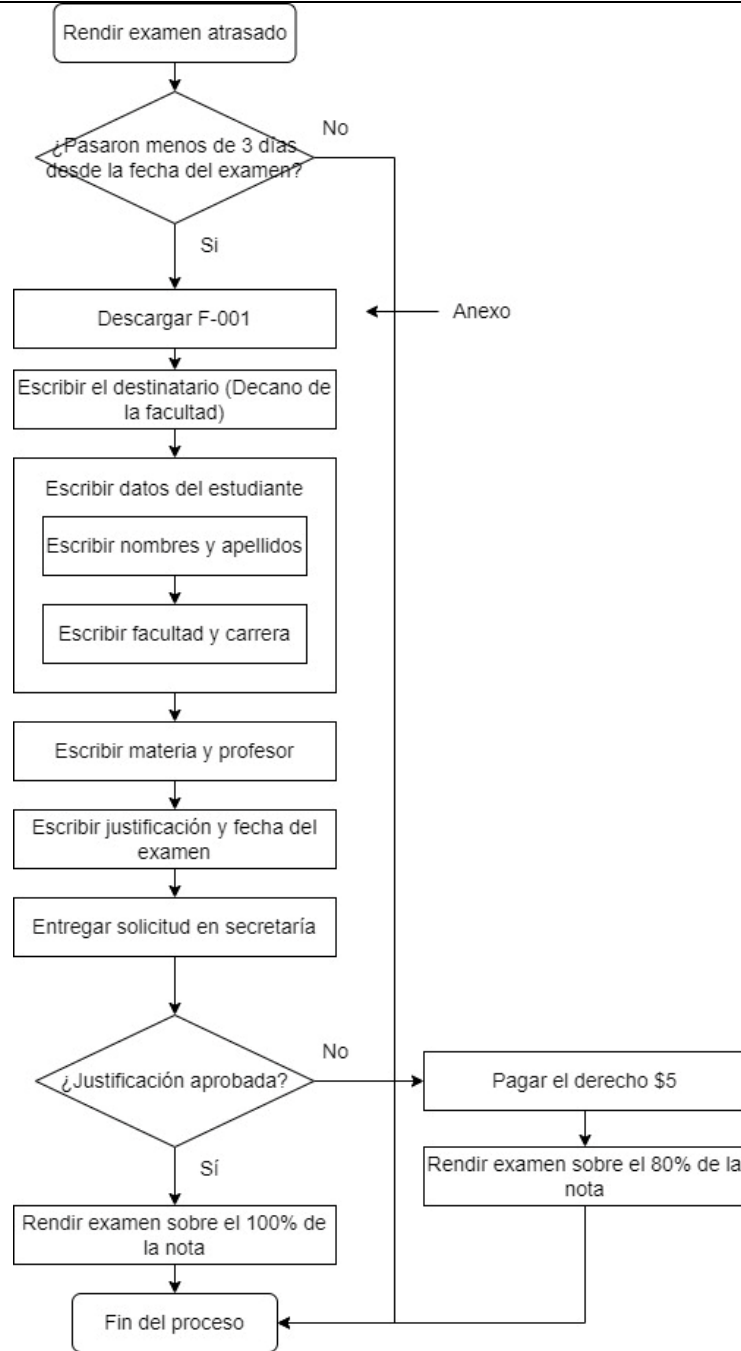
Elaborado por: Isaac Espinoza

3. Rendir Examen Atrasado

Código: REA-001

Responsable: Daniel Paredes

Gráfico 3 Diagrama de Flujo – Rendir Examen Atrasado



Estudiante

Revisado

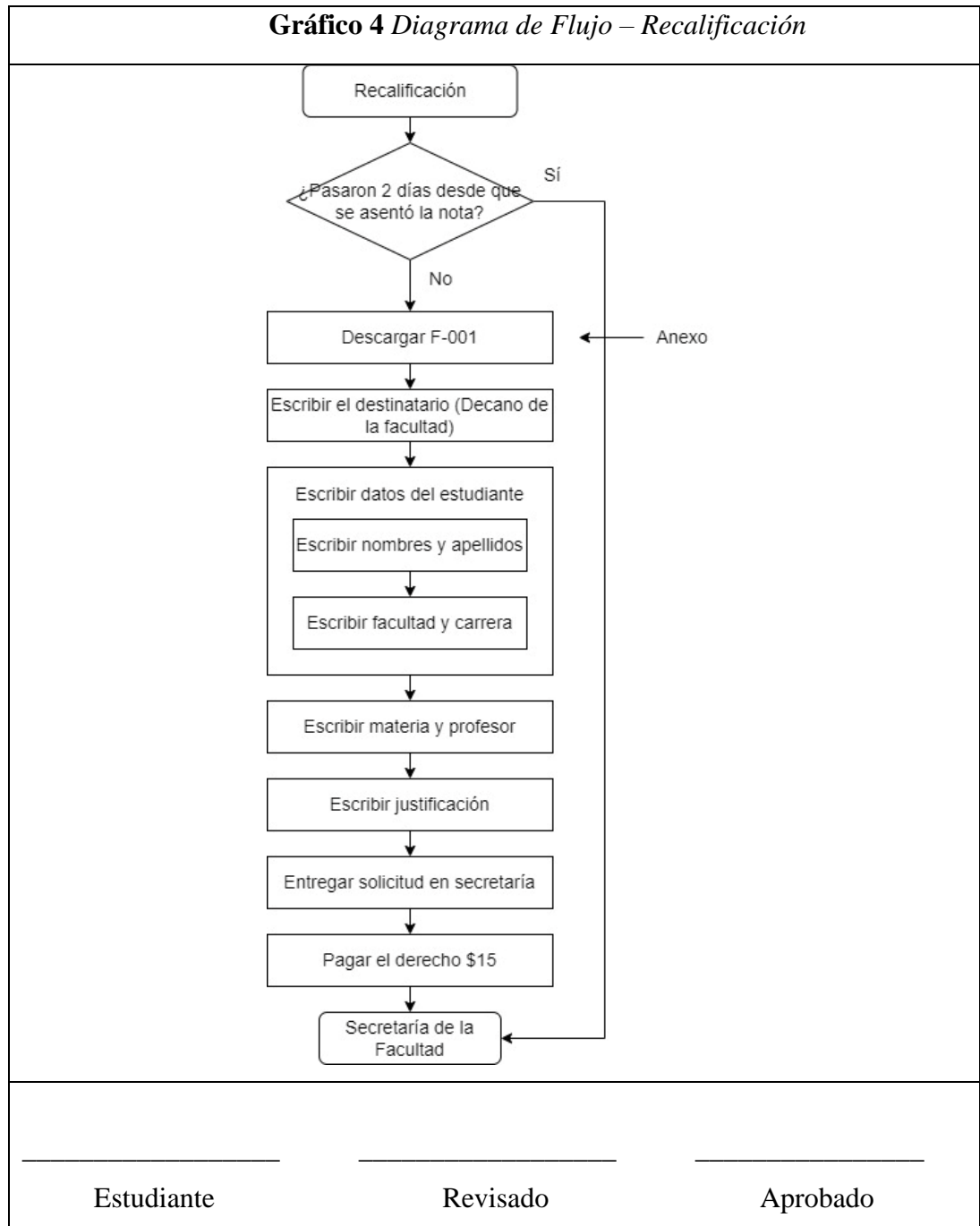
Aprobado

Elaborado por: Isaac Espinoza

4. Recalificación

Código: R-001

Responsable: Daniel Paredes

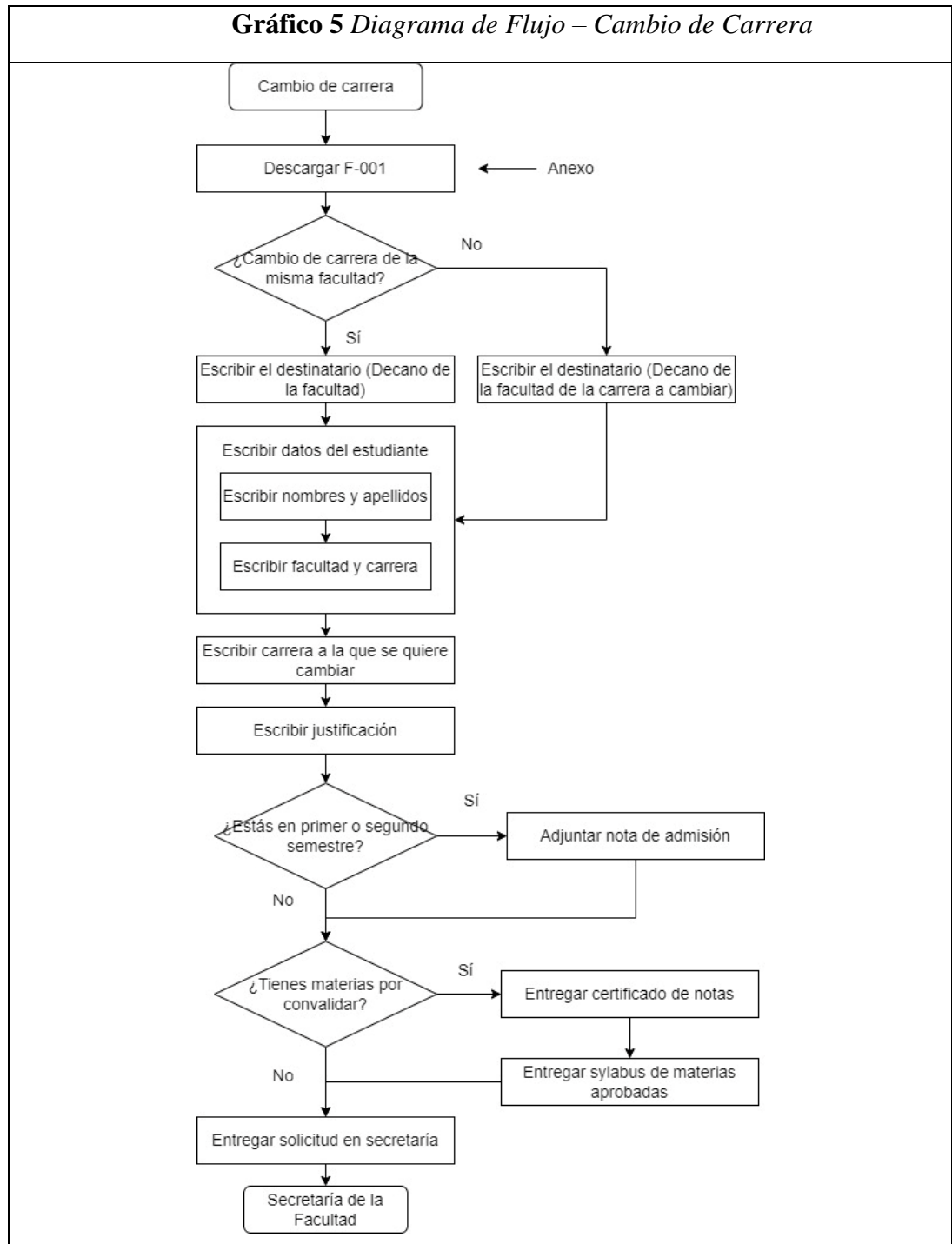


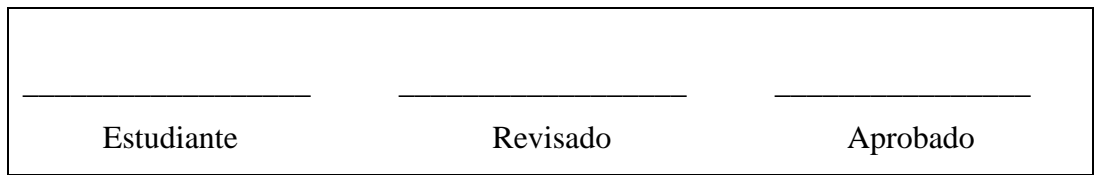
Elaborado por: Isaac Espinoza

5. Cambio de Carrera

Código: CC-001

Responsable: Daniel Paredes



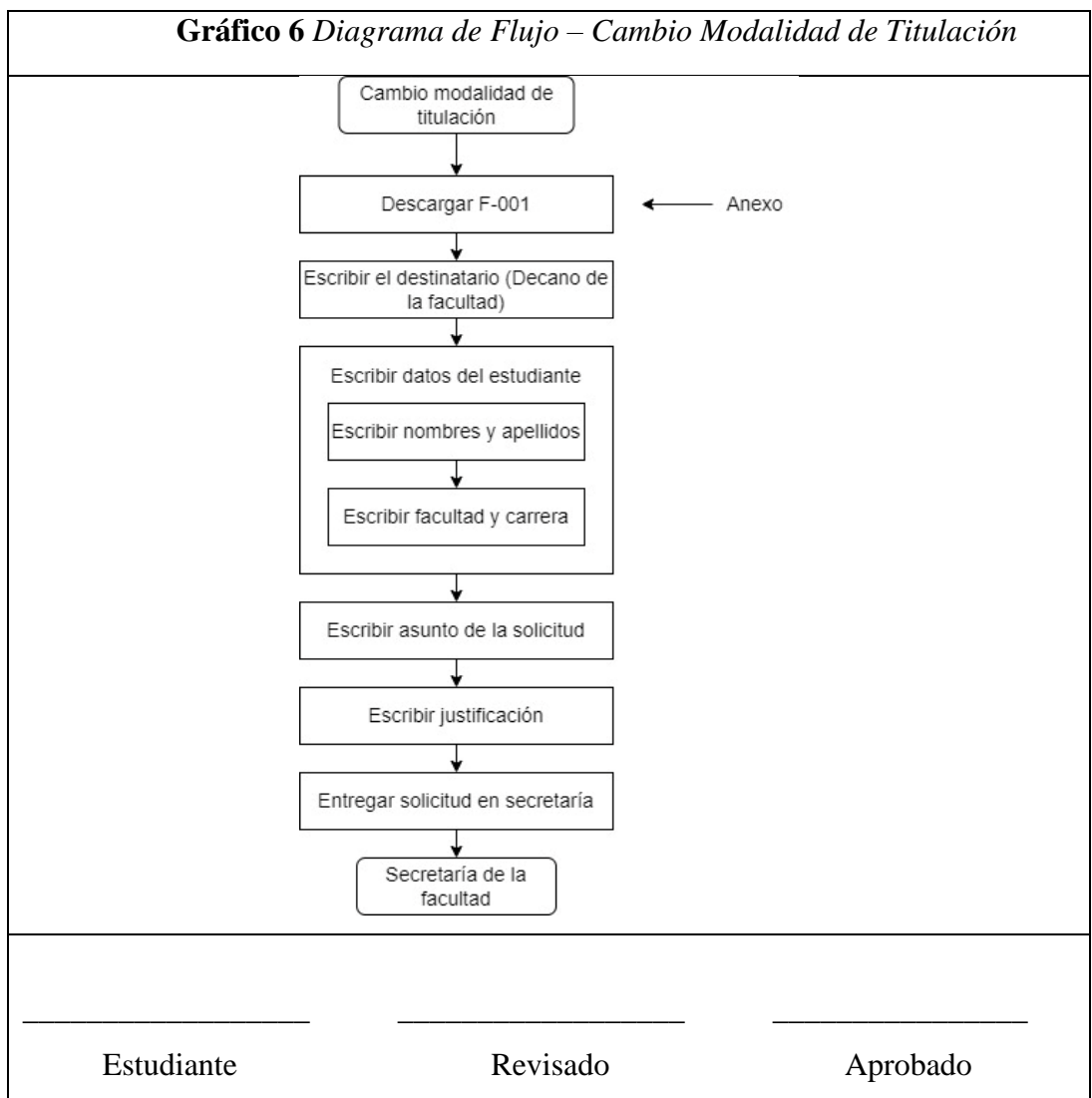


Elaborado por: Isaac Espinoza

6. Cambio Modalidad de Titulación

Código: CMT-001

Responsable: Daniel Paredes

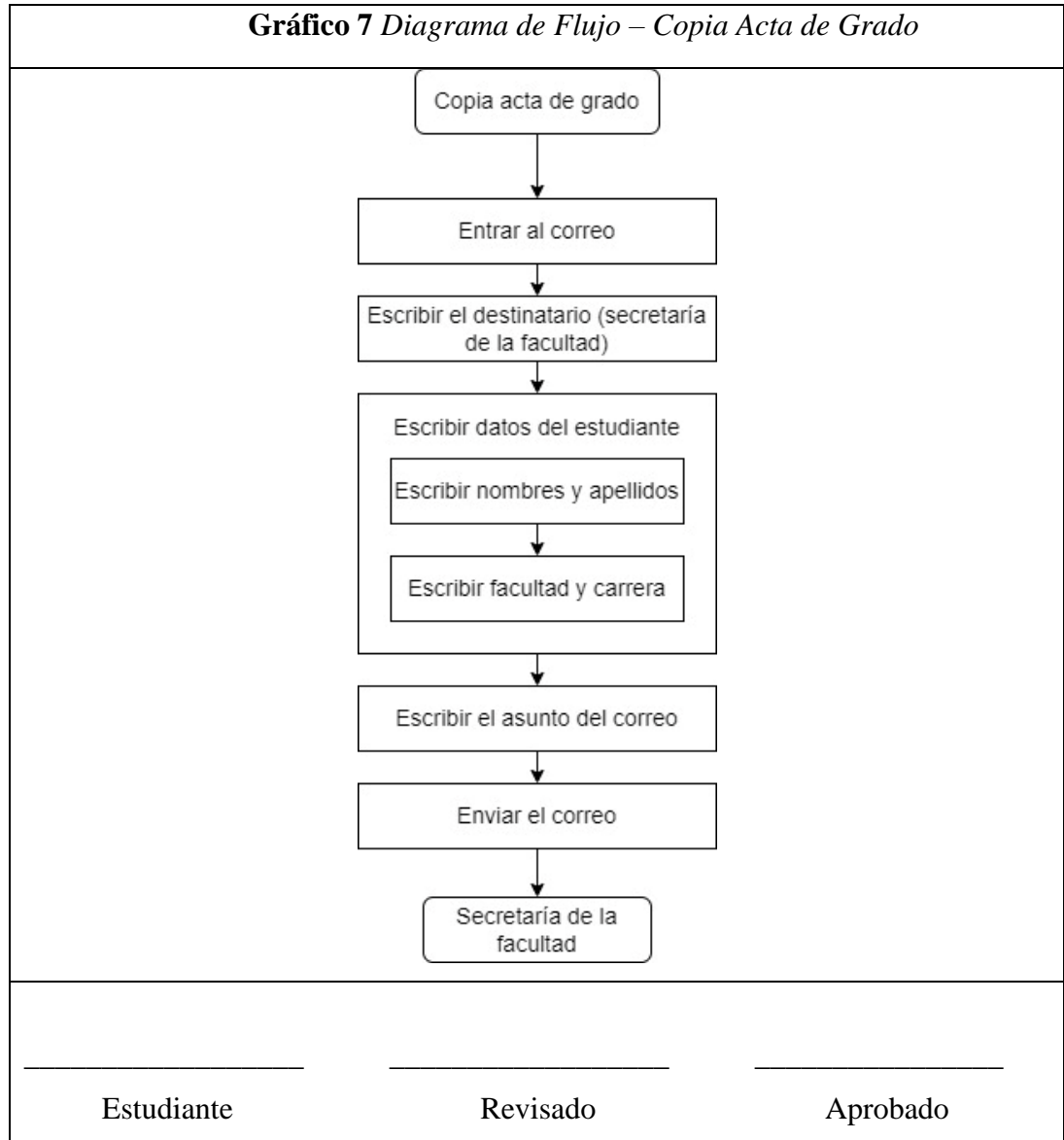


Elaborado por: Isaac Espinoza

7. Copia Acta de Grado

Código: CAG-001

Responsable: Daniel Paredes

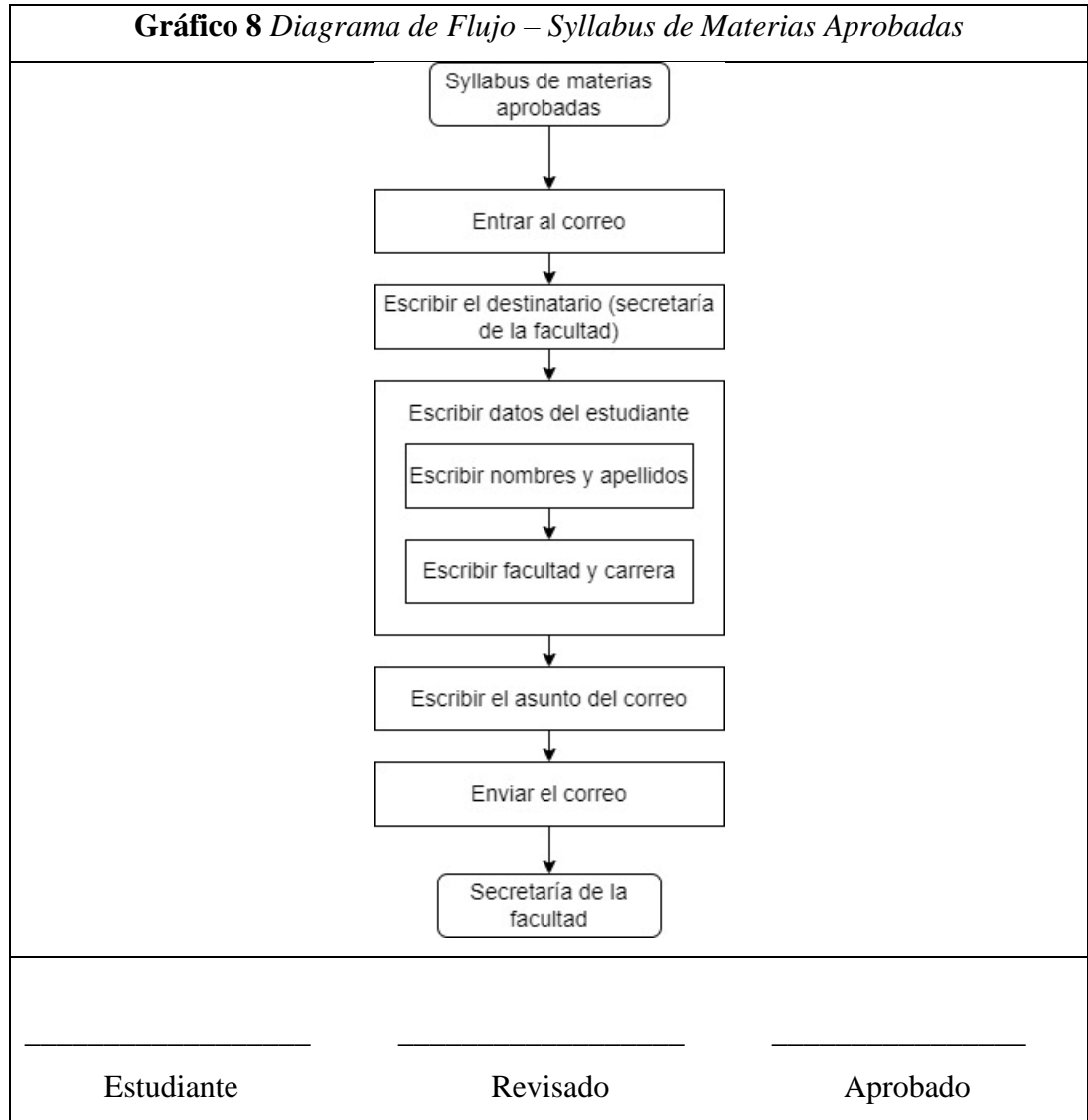


Elaborado por: Isaac Espinoza

8. Syllabus de Materias Aprobadas

Código: SMA-001

Responsable: Daniel Paredes

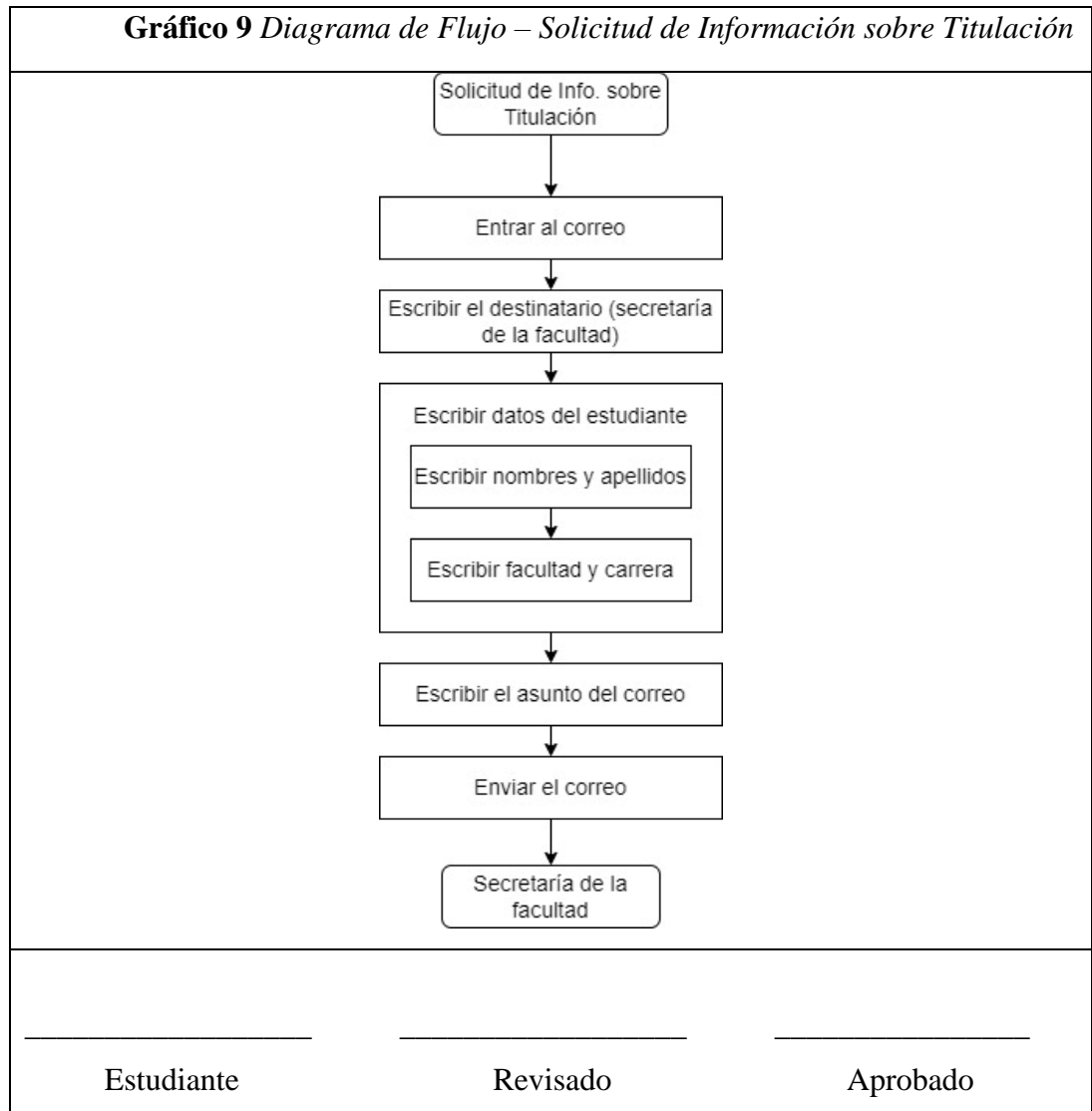


Elaborado por: Isaac Espinoza

9. Solicitud de Información sobre Titulación

Código: SIT-001

Responsable: Daniel Paredes

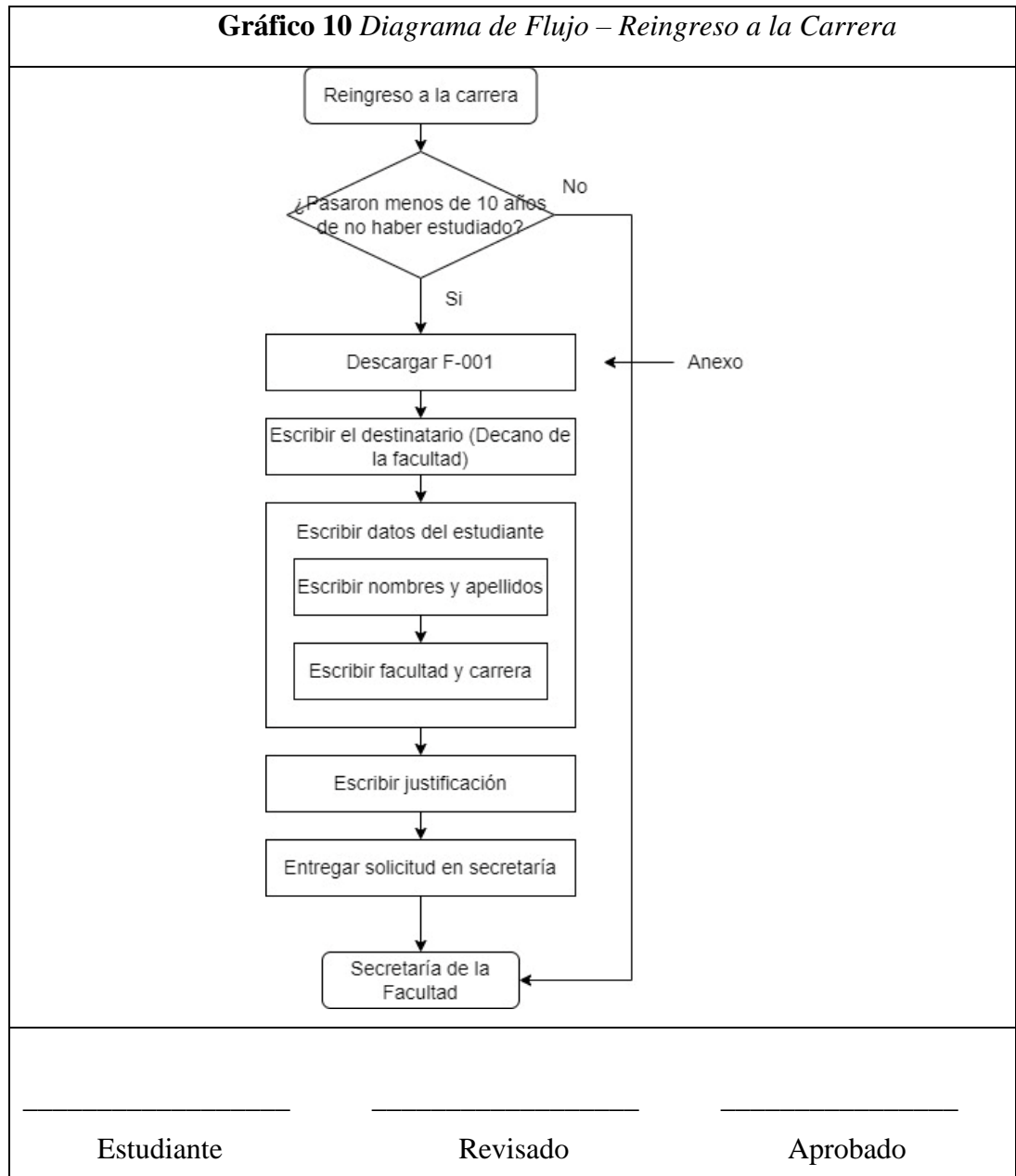


Elaborado por: Isaac Espinoza

10. Reingreso a la Carrera

Código: RC-001

Responsable: Daniel Paredes

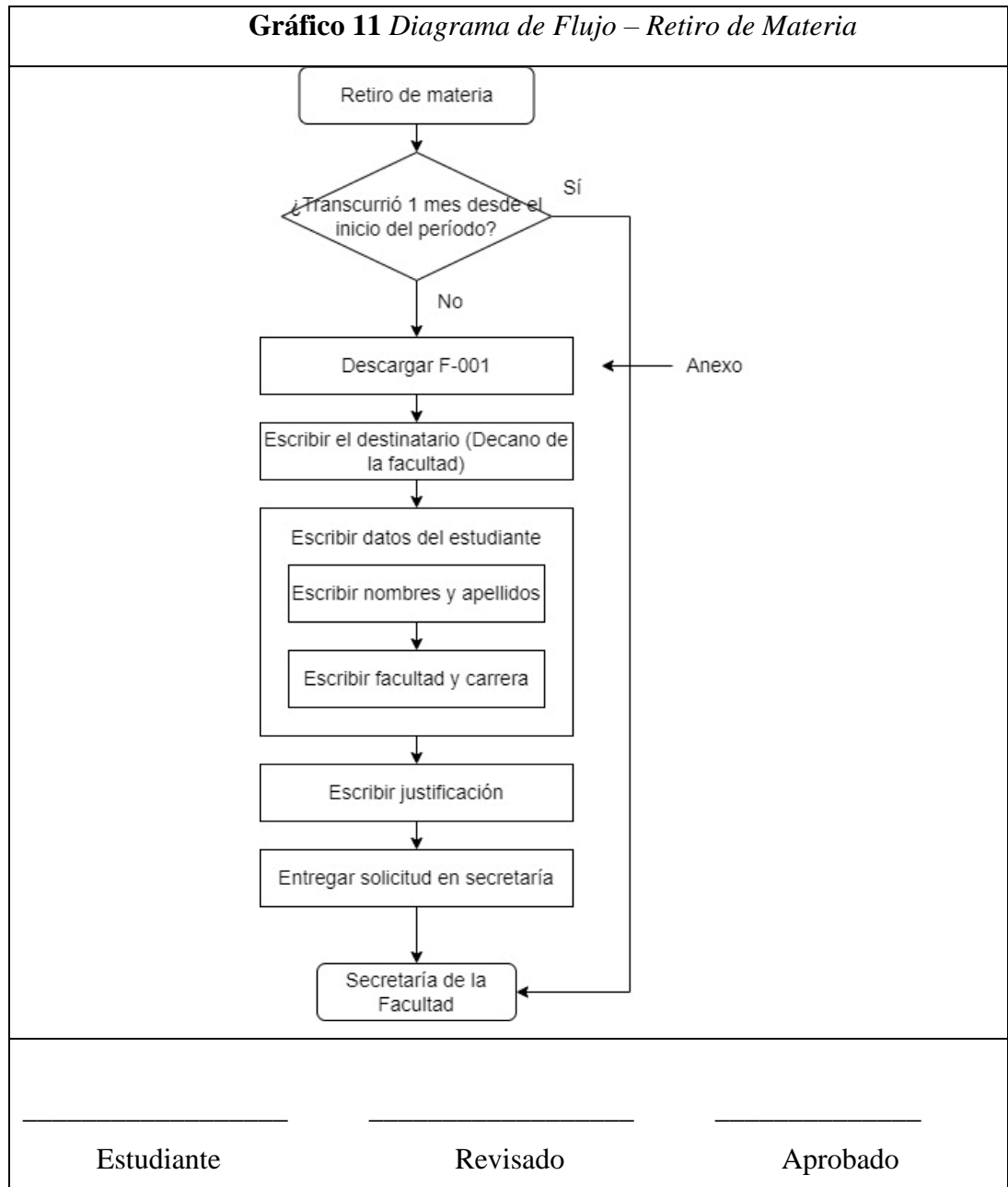


Elaborado por: Isaac Espinoza

11. Retiro de Materia

Código: RM-001

Responsable: Daniel Paredes

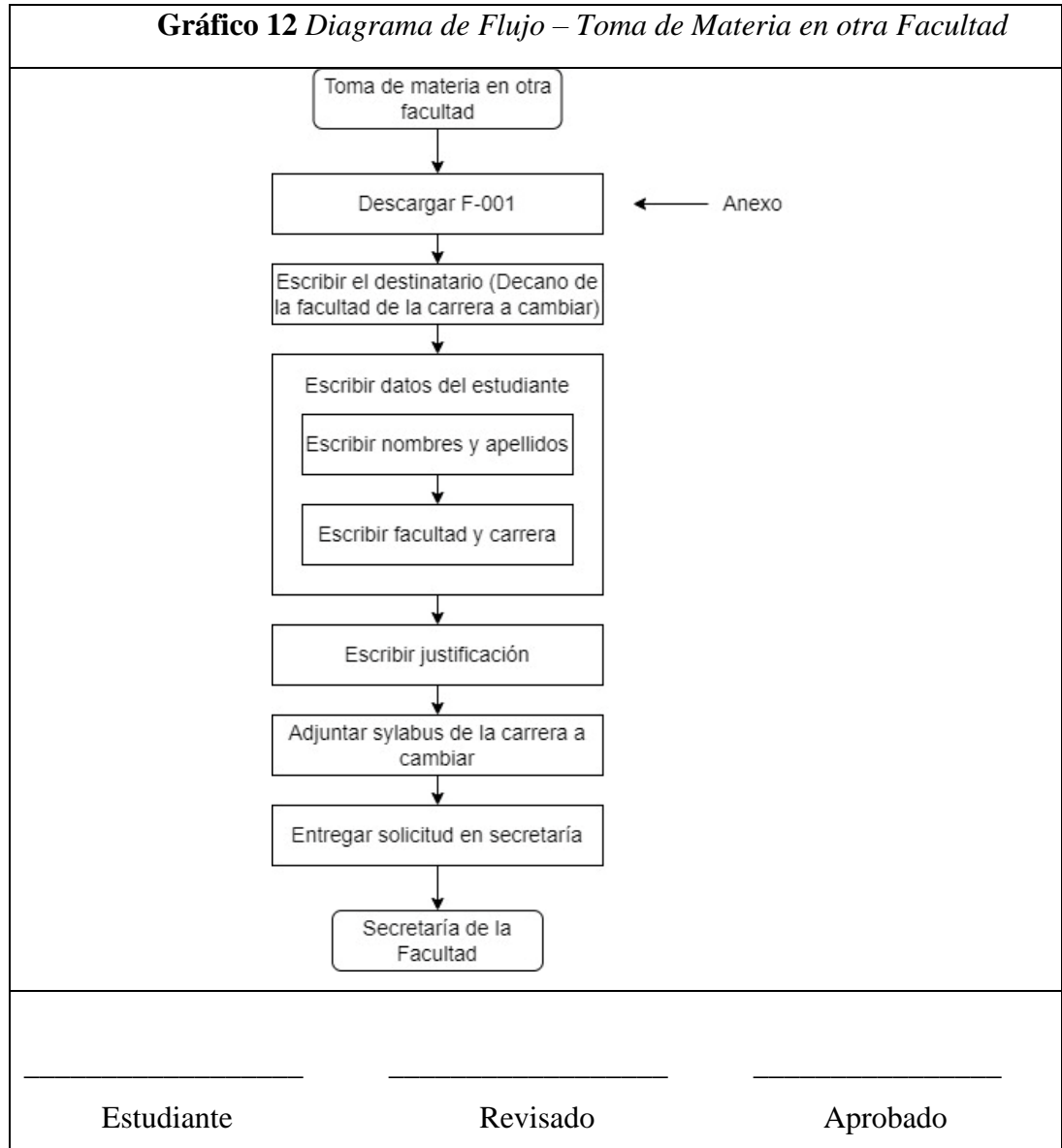


Elaborado por: Isaac Espinoza

12. Tomar Materia en otra Facultad

Código: TMF-001

Responsable: Daniel Paredes

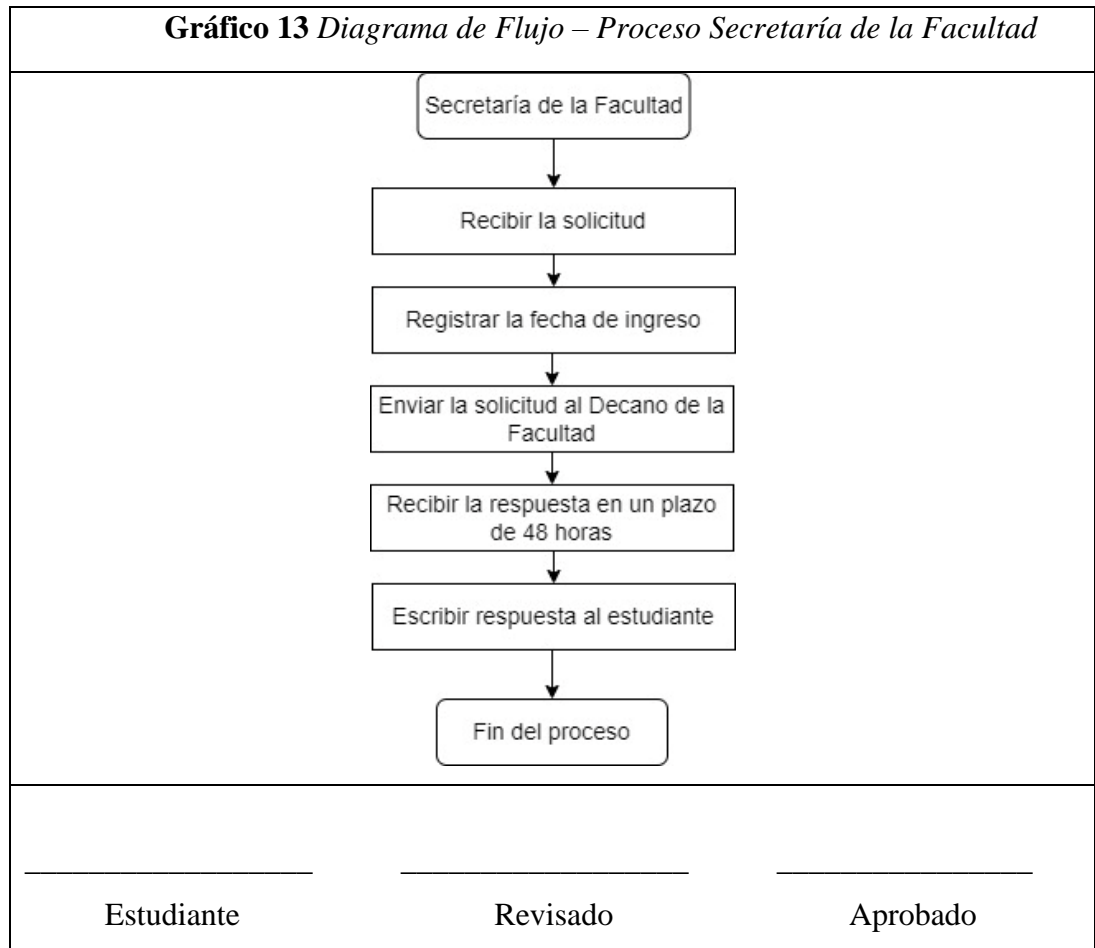


Elaborado por: Isaac Espinoza

13. Proceso Secretaría de la Facultad

Código: PSF-001

Responsable: Daniel Paredes



Elaborado por: Isaac Espinoza

4.2. Definición de Reglas de Negocio

En este apartado se definirán las reglas de negocio de cada proceso para programar la base de conocimiento del asistente virtual. También, se escribirá cada regla en lenguaje Prolog para entender la lógica de este lenguaje de programación.

1. Certificado de Notas

- Todo estudiante puede solicitar el certificado de notas del último semestre cursado.

Prolog:

% 1. Certificado de Notas

% Hechos

para(entrar_correo, escribir_destinatario).

para(escribir_destinatario, escribir_datos).

para(escribir_datos, escribir_asunto).

para(escribir_asunto, enviar_correo).

% Reglas

puede_solicitar(estudiante, certificado_notas):-

 realiza(estudiante, solicitud_correo).

realiza(estudiante, solicitud_correo):-

 para(entrar_correo, escribir_destinatario),

 para(escribir_destinatario, escribir_datos),

 para(escribir_datos, escribir_asunto),

 para(escribir_asunto, enviar_correo).

2. Certificado Estudiante Regular

- Todo estudiante puede solicitar un certificado de estudiante regular.

Prolog:

%2. Certificado Estudiante Regular

%Hechos

para(entrar_correo,escribir_destinatario).

para(escribir_destinatario,escribir_datos).

para(escribir_datos,escribir_asunto).

para(escribir_asunto,enviar_correo).

%Reglas

puede_solicitar(estudiante, certificado_notas):-

 realiza(estudiante, solicitud_correo).

realiza(estudiante, solicitud_correo):-

 para(entrar_correo,escribir_destinatario),

 para(escribir_destinatario,escribir_datos),

 para(escribir_datos,escribir_asunto),

 para(escribir_asunto,enviar_correo).

3. Rendir Examen Atrasado

- El estudiante puede solicitar rendir un examen atrasado si no han pasado 3 días hábiles desde la fecha del examen.
- El estudiante debe escribir o adjuntar una justificación válida para poder rendir el examen sobre el 100% de la nota.
- Si la justificación que entrega el estudiante no es válida, debe pagar el derecho para dar el examen (valor de \$5) y rendirá el examen sobre el 80% de la nota.

Prolog:

%3. Rendir Examen Atrasado

%Hechos

es_plazo(pasaron_3_dias_habiles_desde_fecha_examen).

para(descargar_f-001,escribir_destinatario_decano).

para(escribir_destinatario_decano,escribir_datos).

para(escribir_datos,escribir_materiayprofesor).

para(escribir_materiayprofesor,escribir_justificacionyfecha_examen).

para(escribir_justificacionyfecha_examen,entregar_solicitud).

aprobado(rendir_examen_100%).

no_aprobado(pagar_\$5_rendir_examen_80%).

%Reglas

puede_solicitar(estudiante, rendir_examen_atrasado):-

 cumple(estudiante, plazo),

 realiza(estudiante, solicitud),

 recibe(estudiante,aprobacion).

cumple(estudiante, plazo):-

 es_plazo(pasaron_3_dias_habiles_desde_fecha_examen).

realiza(estudiante, solicitud):-

 para(descargar_f-001,escribir_destinatario_decano),

 para(escribir_destinatario_decano,escribir_datos),

 para(escribir_datos,escribir_materiayprofesor),

 para(escribir_materiayprofesor,escribir_justificacionyfecha_examen),

 para(escribir_justificacionyfecha_examen,entregar_solicitud).

recibe(estudiante,aprobacion):-

aprobado(rendir_examen_100%).
no_aprobado(pagar_\$5_rendir_examen_80%).

4. Recalificación

- El estudiante puede solicitar recalificación de un examen si no transcurrieron 2 días desde que se asentó la nota.
- El estudiante debe pagar el derecho (valor de \$15) para la recalificación.

Prolog:

%4. Recalificación

%Hechos

es_plazo(pasaron_2_dias_habiles_desde_nota_acentada).

para(descargar_f-001,escribir_destinatario_decano).

para(escribir_destinatario_decano,escribir_datos).

para(escribir_datos,escribir_materiayprofesor).

para(escribir_materiayprofesor,escribir_justificacion).

para(escribir_justificacion,entregar_solicitud).

paga(estudiante,\$15).

%Reglas

puede_solicitar(estudiante, recalificacion):-

 cumple(estudiante, plazo),

 realiza(estudiante, solicitud),

 paga(estudiante,\$15).

cumple(estudiante, plazo):-

 es_plazo(pasaron_2_dias_habiles_desde_nota_acentada).

realiza(estudiante, solicitud):-

```
para(descargar_f-001,escribir_destinatario_decano),
para(escribir_destinatario_decano,escribir_datos),
para(escribir_datos,escribir_materiayprofesor),
para(escribir_materiayprofesor,escribir_justificacion),
para(escribir_justificacion,entregar_solicitud).
```

5. Cambio de Carrera

- El estudiante puede solicitar el cambio de carrera a una de la misma facultad o a una de otra facultad.
- Si el estudiante se encuentra en primer o segundo semestre debe adjuntar la nota de admisión.
- El estudiante puede adjuntar el certificado de notas del semestre y el syllabus de las materias aprobadas para convalidar materias.

Prolog:

%5. Cambio de Carrera

%Hechos

```
para(descargar_f-001,escribir_destinatario_decano).
para(escribir_destinatario_decano,escribir_datos).
para(escribir_datos,escribir_carrera_a_cambiar).
para(escribir_carrera_a_cambiar,escribir_justificacion).
para(escribir_justificacion,entregar_solicitud).
adjuntar(primer_o_segundo_semestre, nota_admision),
adjuntar(convalidar_materias, syllabus_materias_aprobadas).
```

%Reglas

puede_solicitar(estudiante, cambio_carrera):-
 realiza(estudiante, solicitud),
 adjunta(estudiante, documentos).

realiza(estudiante, solicitud):-
 para(descargar_f-001,escribir_destinatario_decano),
 para(escribir_destinatario_decano,escribir_datos),
 para(escribir_datos,escribir_carrera_a_cambiar),
 para(escribir_carrera_a_cambiar,escribir_justificacion),
 para(escribir_justificacion,entregar_solicitud).

adjunta(estudiante, documentos):-
 adjuntar(primer_o_segundo_semestre, nota_admision),
 adjuntar(convalidar_materias, syllabus_materias_aprobadas).

6. Cambio Modalidad de Titulación

- Todo estudiante puede solicitar el cambio de modalidad de titulación con su respectiva justificación.

Prolog:

```
%6.        Cambio Modalidad de Titulación
%Hechos
para(descargar_f-001,escribir_destinatario_decano).
para(escribir_destinatario_decano,escribir_datos).
para(escribir_datos,escribir_asunto).
para(escribir_asunto,escribir_justificacion).
para(escribir_justificacion,entregar_solicitud).
```

%Reglas

puede_solicitar(estudiante, cambio_carrera):-

realiza(estudiante, solicitud).

realiza(estudiante, solicitud):-

para(descargar_f-001,escribir_destinatario_decano),

para(escribir_destinatario_decano,escribir_datos),

para(escribir_datos,escribir_asunto),

para(escribir_asunto,escribir_justificacion),

para(escribir_justificacion,entregar_solicitud).

7. Copia Acta de Grado

- Todo estudiante puede solicitar una copia del acta de grado.

Prolog:

%7. Copia Acta de Grado

%Hechos

para(entrar_correo,escribir_destinatario).

para(escribir_destinatario,escribir_datos).

para(escribir_datos,escribir_asunto).

para(escribir_asunto,enviar_correo).

%Reglas

puede_solicitar(estudiante, certificado_notas):-

realiza(estudiante, solicitud_correo).

realiza(estudiante, solicitud_correo):-

para(entrar_correo,escribir_destinatario),

```
para(escribir_destinatario,escribir_datos),
para(escribir_datos,escribir_asunto),
para(escribir_asunto,enviar_correo).
```

8. Syllabus de Materias Aprobadas

- Todo estudiante puede solicitar el syllabus de las materias aprobadas de la carrera.

Prolog:

```
%8. Syllabus de Materias Aprobadas
```

```
%Hechos
```

```
para(entrar_correo,escribir_destinatario).
```

```
para(escribir_destinatario,escribir_datos).
```

```
para(escribir_datos,escribir_asunto).
```

```
para(escribir_asunto,enviar_correo).
```

```
%Reglas
```

```
puede_solicitar(estudiante, certificado_notas):-
```

```
    realiza(estudiante, solicitud_correo).
```

```
realiza(estudiante, solicitud_correo):-
```

```
    para(entrar_correo,escribir_destinatario),
```

```
    para(escribir_destinatario,escribir_datos),
```

```
    para(escribir_datos,escribir_asunto),
```

```
    para(escribir_asunto,enviar_correo).
```

9. Solicitud de Información sobre Titulación

- Todo estudiante puede solicitar información sobre el proceso de Titulación.

Prolog:

%9. Solicitud de Información sobre Titulación

%Hechos

para(entrar_correo,escribir_destinatario).

para(escribir_destinatario,escribir_datos).

para(escribir_datos,escribir_asunto).

para(escribir_asunto,enviar_correo).

%Reglas

puede_solicitar(estudiante, certificado_notas):-

realiza(estudiante, solicitud_correo).

realiza(estudiante, solicitud_correo):-

para(entrar_correo,escribir_destinatario),

para(escribir_destinatario,escribir_datos),

para(escribir_datos,escribir_asunto),

para(escribir_asunto,enviar_correo).

10. Reingreso a la Carrera

- El estudiante puede solicitar el reingreso a la carrera en caso de que no hayan transcurrido 10 años de no haber estudiado y adjuntar su respectiva justificación.

Prolog:

%10. Reingreso a la Carrera

%Hechos

es_plazo(pasaron_menos_10_anos_sin_estudiar).

para(descargar_f-001,escribir_destinatario_decanao).

```
para(escribir_destinatario_decano,escribir_datos).
para(escribir_datos,escribir_justificacion).
para(escribir_justificacion,entregar_solicitud).
```

```
%Reglas
```

```
puede_solicitar(estudiante, recalificacion):-
    cumple(estudiante, plazo),
    realiza(estudiante, solicitud).
```

```
cumple(estudiante, plazo):-
    es_plazo(pasaron_menos_10_anos_sin_estudiar).
```

```
realiza(estudiante, solicitud):-
    para(descargar_f-001,escribir_destinatario_decano),
    para(escribir_destinatario_decano,escribir_datos),
    para(escribir_datos,escribir_justificacion),
    para(escribir_justificacion,entregar_solicitud).
```

11. Retiro de Materia

- El estudiante puede solicitar el retiro de la materia si no ha transcurrido 1 mes desde el inicio del período y escribir su respectiva justificación.

Prolog:

```
% 11. Retiro de Materia
%Hechos
es_plazo(paso_1_mes_desde_inicio_periodo).
para(descargar_f-001,escribir_destinatario_decano).
para(escribir_destinatario_decano,escribir_datos).
```

```
para(escribir_datos,escribir_justificacion).
para(escribir_justificacion,entregar_solicitud).
```

```
%Reglas
```

```
puede_solicitar(estudiante, recalificacion):-
    cumple(estudiante, plazo),
    realiza(estudiante, solicitud).
```

```
cumple(estudiante, plazo):-
    es_plazo(paso_1_mes_desde_inicio_periodo).
```

```
realiza(estudiante, solicitud):-
    para(descargar_f-001,escribir_destinatario_deciano_carrera_a_cambiar),
    para(escribir_destinatario_deciano_carrera_a_cambiar,escribir_datos),
    para(escribir_datos,escribir_justificacion),
    para(escribir_justificacion,entregar_solicitud).
```

12. Tomar Materia en otra Facultad

- El estudiante puede solicitar tomar una materia en otra facultad adjuntando el syllabus de la carrera que tenga la materia solicitada.
- La secretaría de la facultad se encargará de verificar si se puede realizar el cambio.

Prolog:

```
%12. Tomar Materia en otra Facultad
%Hechos
para(descargar_f-001,escribir_destinatario_deciano_carrera_a_cambiar).
para(escribir_destinatario_deciano_carrera_a_cambiar,escribir_datos).
para(escribir_datos,escribir_justificacion).
```

```
para(escribir_justificacion,entregar_solicitud).
adjunta(estudiante, syllabus_carrera_a_cambiar).
```

```
%Reglas
```

```
puede_solicitar(estudiante, recalificacion):-
```

```
    realiza(estudiante, solicitud),
    adjunta(estudiante, syllabus_carrera_a_cambiar).
```

```
realiza(estudiante, solicitud):-
```

```
    para(descargar_f-001,escribir_destinatario_decano),
    para(escribir_destinatario_decano,escribir_datos),
    para(escribir_datos,escribir_justificacion),
    para(escribir_justificacion,entregar_solicitud).
```

13. Proceso Secretaría de la Facultad

- La Secretaría de la Facultad registra la fecha de ingreso de las solicitudes de los estudiantes.
- La Secretaría se encarga de enviar las solicitudes al Decano de la facultad, en caso de que se necesite que las apruebe.
- La Secretaría enviará una respuesta a los estudiantes en un plazo de 48 horas.

Prolog:

```
% 13. Proceso Secretaría de la Facultad
```

```
%Hechos
```

```
recibe_solicitud(secretaria).
```

```
para(recibe_solicitud, registrar_fecha_ingreso).
```

```
para(registrar_fecha_ingreso, enviar_solicitud_decano).
```

para(enviar_solicitud_decano, recibir_respuesta_dentro_48_horas).
para(recibir_respuesta_dentro_48_horas, escribir_respuesta_estudiante).

4.3. Desarrollo del Asistente Virtual

Flujo del Proceso para el Desarrollo de Inteligencia Artificial

1. Objetivo de Negocio

La innovación académica y organizacional es uno de los objetivos de la Pontificia Universidad Católica del Ecuador. Este se encuentra descrito en la visión de la institución:

“Nos identificamos como un proyecto inclusivo, eficiente, en diálogo con su entorno y en constante innovación académica y organizacional” (PUCE, 2022).

Entonces, el presente proyecto se encuentra alineado al objetivo de negocio de la PUCE ya que el asistente virtual busca innovación en la parte organizacional de los trámites de la facultad de ingeniería.

2. Elección de Fuente de Datos

La fuente de datos se determinó que sería la secretaría de la facultad de ingeniería porque desde este lugar se maneja toda la información sobre trámites necesaria para la generación de la base de conocimiento del asistente virtual.

El personal encargado de la parte administrativa de la secretaría tiene conocimiento sobre los trámites de la facultad de ingeniería, motivo por el que reciben las solicitudes y las procesan de acuerdo con los requerimientos de los estudiantes.

La entrevista se realizó a un miembro del personal con ayuda del siguiente instrumento:

Cuadro 1 *Cédula de la Entrevista*

Fecha: 12/05/2022	Lugar: Secretaría de la facultad de ingeniería de la PUCE.
Entrevistador: Isaac Espinoza	Entrevistado: Daniel Paredes
Tema: Trámites principales que realizan los estudiantes en la secretaría de la facultad de ingeniería.	
<ol style="list-style-type: none"> 1. ¿Qué certificados suelen solicitar un estudiante en la secretaría? 2. ¿Qué tipo de solicitudes suelen solicitar los estudiantes con respecto a las notas? 3. ¿Se realiza solicitud para cambio de carrera? ¿Qué otro tipo de solicitudes de este tipo se suele solicitar? 4. ¿Qué tipo de solicitud suelen realizar los estudiantes en relación con las materias? 5. ¿Qué otro tipo de solicitudes suelen realizar los estudiantes? 6. ¿Qué formato se maneja dentro de la secretaría para que los estudiantes puedan realizar solicitudes? 7. ¿Cómo maneja la secretaría de la facultad las solicitudes? ¿Qué tiempo tiene que esperar el estudiante para recibir una respuesta? 	

Elaborado por: Isaac Espinoza

Con los datos recogidos de la entrevista se analizaron los procesos:

- Certificados de notas, certificados de estudiante regular y cuánto tiempo tiene que esperar el estudiante para recibir una respuesta.
- Rendir un examen atrasado, qué datos tiene que incluir el estudiante en la solicitud y si se tiene que realizar un pago para realizar este trámite.
- Recalificación de un examen, el plazo que debe cumplir el estudiante para realizar esta solicitud y si se tiene que realizar algún tipo de pago.
- Cambio de carrera, cómo realizarlo de manera interna y externa a la facultad y qué documentos se deben adjuntar con la solicitud.
- Cambio de modalidad de titulación y qué tipo de razón debe incluir el estudiante.
- Copia Acta de Grado y si se debe cumplir algún requisito previo.
- Syllabus de materias aprobadas y qué tiempo debe esperar el estudiante para recibir una respuesta.
- Información sobre titulación que se maneja en la secretaría de acuerdo con la vigencia de la malla.
- Reingreso a la carrera y qué requisitos debe cumplir el estudiante para realizar este trámite.
- Retiro de una materia, el plazo que debe cumplir el estudiante y las razones que debe escribir el estudiante en la solicitud.
- Tomar una materia en otra facultad y qué tipo de datos se tiene que incluir en la solicitud.

A parte de estas solicitudes también se analizó el proceso que realiza la secretaría con las solicitudes de los estudiantes. Estos procesos son las principales solicitudes que suelen requerir los estudiantes de la facultad de ingeniería.

3. Gestión de Datos (ETC: Extracción, Transformación)

Extracción

La extracción de datos se realizó por medio de una entrevista al personal de la secretaría de la facultad de ingeniería. De esta manera se obtuvo a detalle la información sobre los trámites principales que realizan los estudiantes en la facultad.

Se definieron los procesos con diagramas de flujo, los cuales fueron corroborados posteriormente por el personal de la secretaría. Una conversión de datos no fue necesaria ya que los datos se llevaban directamente a la codificación del asistente virtual.

Los procesos diseñados para trabajar dentro del sistema experto son los siguientes:

- Certificados de notas
- Certificado de estudiante regular
- Rendir un examen atrasado
- Recalificación de un examen
- Cambio de carrera
- Cambio de modalidad de titulación
- Copia Acta de Grado
- Syllabus de materias aprobadas

- Información sobre titulación
- Reingreso a la carrera
- Retiro de una materia
- Tomar una materia en otra facultad
- Proceso que realiza la secretaría con las solicitudes

Transformación

Los datos no necesarios para la base de conocimiento del asistente virtual fueron descartados. Entre estos, los datos que no tenían mayor nivel de relevancia, datos duplicados o incorrectos obtenidos de la entrevista al personal de la secretaría de la facultad.

En cuanto al rediseño de procesos, se rediseñaron los procesos de *Rendir un examen atrasado* y *Recalificación de un examen*. Dentro del flujo de estos procesos se tenía que considerar un componente de pago y este se procedía de manera diferente en los dos procesos.

Para el proceso de *Cambio de carrera* y *Tomar una materia en otra facultad*, también se tuvo que rediseñar debido a que se necesita adjuntar diferentes documentos para poder finalizar el proceso.

Se rediseñó el proceso de *Información sobre titulación*. En un inicio se pensaba ofrecer una información precisa, sin embargo, estos datos pueden variar debido al rediseño de las mallas de las carreras. Entonces, la forma en que actúa este proceso consiste en que

el estudiante realice la solicitud a la secretaría y esta entregará la información que constantemente se encontraría actualizada.

En cuanto a las **reglas de negocio**, se obtuvo las siguientes:

- **Certificado de Notas**
 - Todo estudiante puede solicitar el certificado de notas del último semestre cursado.
- **Certificado Estudiante Regular**
 - Todo estudiante puede solicitar un certificado de estudiante regular.
- **Rendir Examen Atrasado**
 - El estudiante puede solicitar rendir un examen atrasado si no han pasado 3 días hábiles desde la fecha del examen.
 - El estudiante debe escribir o adjuntar una justificación válida para poder rendir el examen sobre el 100% de la nota.
 - Si la justificación que entrega el estudiante no es válida, debe pagar el derecho para dar el examen (valor de \$5) y rendirá el examen sobre el 80% de la nota.
- **Recalificación**
 - El estudiante puede solicitar recalificación de un examen si no transcurrieron 2 días desde que se asentó la nota.
 - El estudiante debe pagar el derecho (valor de \$15) para la recalificación.
- **Cambio de Carrera**
 - El estudiante puede solicitar el cambio de carrera a una de la misma facultad o a una de otra facultad.

- Si el estudiante se encuentra en primer o segundo semestre debe adjuntar la nota de admisión.
- El estudiante puede adjuntar el certificado de notas del semestre y el syllabus de las materias aprobadas para convalidar materias.
- **Cambio Modalidad de Titulación**
 - Todo estudiante puede solicitar el cambio de modalidad de titulación con su respectiva justificación.
- **Copia Acta de Grado**
 - Todo estudiante puede solicitar una copia del acta de grado.
- **Syllabus Materias Aprobadas**
 - Todo estudiante puede solicitar el syllabus de las materias aprobadas de la carrera.
- **Solicitud de Información sobre Titulación**
 - Todo estudiante puede solicitar información sobre el proceso de Titulación.
- **Reingreso a la Carrera**
 - El estudiante puede solicitar el reingreso a la carrera en caso de que no hayan transcurrido 10 años de no haber estudiado y adjuntar su respectiva justificación.
- **Retiro de Materia**
 - El estudiante puede solicitar el retiro de la materia si no ha transcurrido 1 mes desde el inicio del período y escribir su respectiva justificación.
- **Tomar Materia en otra Facultad**

- El estudiante puede solicitar tomar una materia en otra facultad adjuntando el syllabus de la carrera que tenga la materia solicitada.
- La secretaría de la facultad se encargará de verificar si se puede realizar el cambio.
- **Proceso Secretaría de la Facultad**
 - La Secretaría de la Facultad registra la fecha de ingreso de las solicitudes de los estudiantes.
 - La Secretaría se encarga de enviar las solicitudes al Decano de la facultad, en caso de que se requiera su aprobación.
 - La Secretaría enviará una respuesta a los estudiantes en un plazo de 48 horas.

4. Almacenaje (ETC: Carga)

Carga

Para el almacenaje de los datos no es necesario un repositorio de la magnitud de un data warehouse ya que el límite del tamaño de los datos es el tamaño del código fuente del asistente virtual.

En caso de que se necesite implementar el asistente virtual, este se puede alojar en un servidor de la facultad de ingeniería. El acceso al agente virtual podría estar disponible desde el dispositivo móvil de los estudiantes de la facultad.

En PROLOG las reglas de se definieron con un formato para manejarlas a manera de listas. Para poder identificarlas se les asignó una numeración, una prioridad que permitirá al sistema experto reconocer las reglas más importantes (las reglas con mayor prioridad se

ejecutarán primero), una palabra clave que servirá de patrón para que el asistente pueda reconocer la regla y por último las respuestas que el agente virtual puede devolver. La funcionalidad del código del sistema experto se explicará en la sección de *Backend*.

Figura 4 Archivo *simplification.rules* - Formato de las reglas

```
1 % El formato de las reglas es:
2 % rules(ID , importancia de la regla , [patrón], [
3 %           [respuesta 1],
4 %           [respuesta 2],
5 %           ...
6 %           [respuesta n]]).
```

Elaborado por: Isaac Espinoza

5. Analítica de Datos

Del análisis de los datos finales se pudo generar las reglas del negocio que formarían parte de la base de conocimiento del asistente virtual. La generación de las reglas para la Inteligencia Artificial permitirá al sistema experto simular el proceso de inteligencia de un humano para dar solución a un problema en específico que en esta ocasión es brindar información sobre trámites de la facultad de ingeniería.

La estructura de los datos finales se estableció con relación a los diagramas de flujo. Se entendió el flujo de cada proceso y se empezó a redactar cada uno de estos para que funcionen a manera de reglas dentro de la base de conocimiento del asistente virtual.

El archivo que contiene las funciones principales es *milu.pl*. El código está dividido en 4 secciones: Interfaz entrada/salida, Funciones básicas, Función Milu y Función Principal. La sección que en donde se lleva a cabo la respuesta del sistema experto es *Función Milu*.

Función Milu contiene las funciones primero para recorrer el archivo de respuestas *reply.rules* e imprimirlas de acuerdo con el texto que el usuario ingresa, tratar el texto del usuario para reducirlo a un patrón que el sistema experto identifica dentro de las reglas y por último identificar los sinónimos o palabras similares en el archivo de reglas de simplificación. Con estas funciones el sistema experto cumple su función de aprendizaje a manera de reconocer diferentes patrones en el texto que el usuario ingresa para realizar una consulta.

6. Capa de Business Intelligence

Roadmap para el desarrollo de Inteligencia Artificial

- 1. Pre-procesamiento:** El código fuente se encuentra distribuido en tres archivos: *milu.pl*, *reply.rules* y *simplificacion.rules*. El primero funciona como archivo principal del agente virtual, el segundo tiene toda la base de conocimiento necesaria para brindar las respuestas al usuario y el último se encarga de simplificar algunas palabras y que puedan encajar en un mayor número de reglas.
- 2. Aprendizaje:** La toma de decisiones podrá realizar gracias a las reglas de negocio previamente orientadas a los objetivos de la empresa. El manejo de los datos ya se ha detallado con las etapas del proceso ETC en la Gestión de Datos.
- 3. Evaluación:** Este sistema experto imita el proceso de razonamiento humano para encontrar similitudes entre las palabras que ingresa el usuario y las que encuentran en la base de conocimiento y así brindar una respuesta oportuna (Badaró, Agüero, & Agüero, 2013).

4. **Predicción:** De acuerdo con los sinónimos que pueda encontrar el asistente virtual en el archivo de simplificación, el sistema experto puede ofrecer soluciones con las palabras que tengan mayor similitud con las reglas.

7. Visualización

La representación gráfica de los datos se realizará desde el entorno de trabajo de SWI-Prolog donde se ejecuta el lenguaje de programación de los archivos .pl (extensión para ejecución de archivos con lenguaje Prolog).

Figura 5 Entorno de trabajo de SWI-Prolog

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('D:/PUCE/Carrera/8vo/Integracion/Tesis/milu/milu.pl').
true.

?- milu.
Llamando a Milu...
Hola, soy Milu, Con que informacion te puedo ayudar?

> como puedo solicitar el syllabus de las materias aprobadas?
milu: Para syllabus de materias aprobadas tienes que realizar una solicitud por
medio del correo institucional de la siguiente manera:
1. Entrar al correo y escribir destinatario (secretaria)
2. Escribir datos: nombres, apellidos, facultad y carrera
3. Escribir asunto del correo solicitando el syllabus de las materias aprobadas
4. Enviar el correo

> █
```

Elaborado por: Isaac Espinoza

Backend

Primero, se necesitó instalar el software SWI-Prolog para poder desarrollar el proyecto. El código se escribió a manera de texto plano con ayuda de herramientas como *Bloc de Notas* de Windows y Notepad++. Para la estructura del código se tomó como referencia el proyecto Eliza de (Mares, 2017), un sistema experto que habla con los usuarios sobre problemas, dificultades y sentimientos.

La codificación empieza en el archivo *milu.pl*, este se encuentra dividido en las secciones: Interfaz entrada/salida, Funciones básicas, Función Milu y Función Principal. Estas secciones serán explicadas con más detalle a continuación.

En esta primera parte, se encuentra el apartado de la interfaz de salida y entrada. Esto se relaciona con el usuario cuando ingresa el texto en la interfaz y el asistente virtual le retorna una respuesta. Se imprime la salida guardada en caso de que exista y luego se encarga de dar lectura la posible pregunta del usuario.

Figura 6 Archivo *milu.pl* – Sección de código: Interfaz entrada/salida

```
1 % Milu - archivo principal
2
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 %%%%%%%%% Interfaz entrada/salida %%%%%%%%%
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 % imprimir salida
8 out(Text):-write(Text).
9
10 % leer la pregunta
11 in(Text):-
12     nl,
13     write("> "),
14     readLine(Text),
15     write("milu: ").
16
17 readLine(Text):-
18     get_char(Char),
19     toLowerCase(Char,LChar),
20     readLine2(LChar,Text).
21 readLine2('\n',[]):-!.
22 readLine2(LChar,[LChar|T]):-readLine(T).
```

Elaborado por: Isaac Espinoza

En el apartado de funciones básicas, primero se declaran los signos de puntuación para que el asistente virtual pueda identificarlos en las entradas que ingresa el usuario. Esto será de utilidad para luego extraer el texto importante de cualquier entrada que pueda ingresar el usuario.

Figura 7 Archivo *milu.pl* – Sección de código: Funciones básicas

```
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 %%%%%%%%%% Funciones basicas %%%%%%%%%%
26 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27
28 % declaracion de signos de puntuación
29 charType('!', punctuation).
30 charType('?', punctuation).
31 charType('.', punctuation).
32 charType(',', punctuation).
33 charType('\', punctuation).
34 charType(' ', whitespace).
35 charType('\t', whitespace).
36
37 % minúsculas usando ASCII
38 toLowerCase(Char, LChar):-
39     char_code(Char, Code),
40     Code >= "A",
41     Code =< "Z",
42     NewCode is Code + 32,
43     char_code(LChar, NewCode), !.
44 toLowerCase(Char, Char).
45
```

Elaborado por: Isaac Espinoza

El código reconoce las minúsculas y mayúsculas tomando como referencia el código ASCII. Se eliminan los caracteres (signos de puntuación) que antes se habían declarado para finalmente transferir la salida a el archivo de respuestas *reply.rules*.

Figura 8 Archivo *milu.pl* – Sección de código: Funciones básicas

```
46 % mayúsculas usando ASCII
47 toUpperCase(Char, UChar):-
48     char_code(Char, Code),
49     Code >= "a",
50     Code =< "z",
51     NewCode is Code - 32,
52     char_code(UChar, NewCode), !.
53 toUpperCase(Char, Char).
54
55 % eliminar caracteres específicos
56 deleteChars([Char|Rest],Type,Out):-
57     charType(Char, Type),
58     deleteChars(Rest,Type,Out),!.
59 deleteChars([Char|Rest],Type,[Char|Out]):-
60     deleteChars(Rest,Type,Out),!.
61 deleteChars([],_, []).
62
63 % transferir salida a la lista de palabras
64 toWords([],[]):-!.
65 toWords(Line, [Word|ResWords]):-
66     readWord(Line, Word, ResLine),
67     toWords(ResLine, ResWords).
```

Elaborado por: Isaac Espinoza

Figura 9 Archivo *milu.pl* – Sección de código: Funciones básicas

```
69 % leer una palabra
70 readWord([], '', []).
71 readWord([Char|Res], '', Res) :- charType(Char, whitespace),!.
72 readWord([Char|ResLine], Word, Res) :-
73     readWord(ResLine, ResWord, Res),
74     atom_concat(Char, ResWord, Word).
```

Elaborado por: Isaac Espinoza

En la función Milu realizará todo el proceso de texto del usuario. Se encuentra la el cambio en los predicados que tiene que ver con las distintas opciones que están definidas en las reglas. Se ejecutan los archivos de reglas y simplificación, entonces se remueven los caracteres que no son necesarios y también se buscan sinónimos de las palabras en el archivo de simplificación para encontrar más coincidencias en las reglas.

Figura 10 Archivo *milu.pl* – Sección de código: Función Milu

```
76 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
77 %%%%%%%%%%%%%%% Función Milu %%%%%%%%%%%%%%%
78 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
79
80 % cambios en los predicados
81 :- dynamic resID/2.
82 resID(_,0).
83
84 % compilación de las reglas y reglas de simplificación
85 init:-
86     consult('D:/PUCE/Carrera/8vo/Integracion/Tesis/milu/simplificacion.rules'),
87     consult('D:/PUCE/Carrera/8vo/Integracion/Tesis/milu/reply.rules').
88
89 % remueve caracteres innecesarios
90 simplify(In, Out):-
91     deleteChars(In, punctuation, Out1),
92     toWords(Out1,Out2),
93     findSynonyms(Out2,Out3),
94     Out = Out3.
95
96 % encuentra sinonimos en el archivo de simplificacion
97 findSynonyms(Words, Syn) :-
98     sr(Words, Syn, RestWords, ResOutput),!,
99     findSynonyms(RestWords, ResOutput).
100 findSynonyms([Word| ResWords], [Word| ResSyn]):-
101     findSynonyms(ResWords, ResSyn),!.
102 findSynonyms([], []).
```

Elaborado por: Isaac Espinoza

Entonces, se busca en el archivo de reglas *reply.rules* la respuesta a la pregunta del usuario. Se compara el texto ingresado por el usuario con las reglas y entonces se retorna una respuesta. En caso de que la regla tenga más de una respuesta, se recorre en la lista de respuestas para llegar a la indicada.

Figura 11 Archivo *milu.pl* – Sección de código: Función *Milu*

```
104 % encuentra una respuesta
105 findReply(Words, Reply) :-
106     findReply2(Words, -2, 0, [], ID, Reply),
107     ID \= 0,
108     updateResID(ID) .
109
110 findReply2([H|T], ActScore, _, _, ID, Res):-
111     findall(Score, rules(_, Score, [H|T], _) , Rules),
112     Rules \= [], % bagof doesn't work as I except
113     max_list(Rules, NewScore),
114     ActScore < NewScore,
115     rules(NewID, NewScore, [H|T], Replies),
116     resID(NewID, ResID),
117     nth0(ResID, Replies, NewReply),
118     findReply2(T, NewScore, NewID, NewReply, ID, Res),!.
119 findReply2([_|T], ActScore, ActID, ActRes, ID, Res):-
120     findReply2(T, ActScore, ActID, ActRes, ID, Res).
121 findReply2([], _, ID, Res, ID, Res).
```

Elaborado por: Isaac Espinoza

Figura 12 Archivo *milu.pl* – Sección de código: Función *Milu*

```
123 % se mueve a la siguiente respuesta
124 updateResID(ID) :-
125     resID(ID, RID),
126     once(rules(ID, _, _, Replies)),
127     length(Replies, Len),
128     NRID is (RID + 1) mod Len,
129     retract((resID(ID, RID) :- !)),
130     asserta(resID(ID, NRID) :- !),!.
131 updateResID(ID) :-
132     resID(ID, RID),
133     once(rules(ID, _, _, Replies)),
134     length(Replies, Len),
135     NRID is (RID + 1) mod Len,
136     asserta(resID(ID, NRID) :- !).
137
```

Elaborado por: Isaac Espinoza

Para la respuesta en la salida, se establece que la primera letra se imprima en mayúscula y el resto del texto normalmente. Se imprime la respuesta y puntuación en caso de existir.

Figura 13 Archivo *milu.pl* – Sección de código: *Función Milu*

```
138 % En mayúscula la primera letra e imprime en la salida
139 writeWords([Word|Res]):-
140     string_chars(Word,[Char|RChar]),
141     toUpperCase(Char,UChar),
142     readWord([UChar|RChar],Out,_),
143     out(Out),
144     writeWords2(Res).
145 % escribe lista interna
146 writeWords2([Word|Res]):-
147     is_list(Word),
148     writeWords2(Word),
149     writeWords2(Res),!.
150 % escribe puntuación
151 writeWords2([Word|Res]):-
152     charType(Word,punctuation),
153     out(Word),
154     writeWords2(Res),!.
155 % escribe letra estándar
156 writeWords2([Word|Res]):-
157     out(" "),
158     out(Word),
159     writeWords2(Res),!.
160 writeWords2([]).
```

Elaborado por: Isaac Espinoza

En la función principal se encuentra la ejecución del asistente virtual. Ejecutando el comando *milu*. en la interfaz, el asistente enviará un saludo y ya estará preparada para recibir las preguntas del usuario. También se encuentra la regla para poder salir del programa y la ejecución de las demás funciones ya antes explicadas.

Figura 14 Archivo *milu.pl* – Sección de código: *Función Principal*

```
162 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
163 %%%%%%%%%% Función Principal %%%%%%%%%%
164 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
165
166 milu:-
167     out("Llamando a Milu...\n"),
168     init,
169     out("Hola, soy Milu, Con que informacion te puedo ayudar?\n"),
170     milu([hi]).
171
172 milu([quit|_]):-!.
173 milu(_):-
174     in(Line),
175     simplify(Line, Words),
176     findReply(Words,Reply),
177     writeWords(Reply),nl,
178     milu(Words).
```

Elaborado por: Isaac Espinoza

La base de conocimiento del asistente virtual se encuentra guardado en el archivo *reply.rules* donde cada regla seguirá un formato. En el formato se contempla: primero una identificación para enumerar las reglas (ID), la importancia de la regla que será de mucha utilidad para dar prioridad a una regla sobre otra, el patrón el cual contendrá la palabra o palabras clave para que el asistente virtual reconozca el texto del usuario dentro de las reglas; finalmente la respuesta o respuestas que puede devolver el asistente virtual:

Figura 15 Archivo *reply.rules* - Reglas

```
1 % El formato de las reglas es:
2 % rules(ID ,importancia de la regla , [patrón], [
3 %           [respuesta 1],
4 %           [respuesta 2],
5 %           ...
6 %           [respuesta n]]).
7
8 % Reglas primera opción
9 rules(1, 10,[notas|_],[
10      [para,certificado,de,notas,tienes,que,realizar,una,solicitud,por,medio,del,
11      correo,institucional,de,la,siguiente,'manera:','\n',
12      1,.,'Entrar',al,correo,y,escribir,destinatario,'(secretaria)','\n',
13      2,.,'Escribir','datos:',nombres,',',apellidos,',',facultad,y,carrera,'\n',
14      3,.,'Escribir',asunto,del,correo,solicitando,el,certificado,de,notas,'\n',
15      4,.,'Enviar',el,correo]]).
16
17 rules(2, 10,[regular|_],[
18      [para,certificado,de,estudiante,regular,tienes,que,realizar,una,solicitud,por,
19      medio,del,correo,institucional,de,la,siguiente,'manera:','\n',
20      1,.,'Entrar',al,correo,y,escribir,destinatario,'(secretaria)','\n',
21      2,.,'Escribir','datos:',nombres,',',apellidos,',',facultad,y,carrera,'\n',
22      3,.,'Escribir',asunto,del,correo,solicitando,el,certificado,de,estudiante,regular,'\n',
23      4,.,'Enviar',el,correo]]).
```

Elaborado por: Isaac Espinoza

También se hizo uso de reglas secundarias y variables para poder reconocer palabras independientes que el usuario pueda preguntar al asistente virtual y guiar a las respuestas que la base de conocimiento puede devolver. Se encuentra además configurado el mensaje que retornará el asistente virtual cuando se finalice el programa.

Figura 16 Archivo *reply.rules* - Reglas

```
40 rules(5, 0, [M|_],[
41      [te,puedo,ayudar,con,informacion,'sobre:','\n',
42      -, 'Solicitar',un,examen,atrasado,'\n',
43      -, 'Solicitar',una,recalificacion]]):-
44      member(M, [examen,rendir]).
45
```

Elaborado por: Isaac Espinoza

Figura 17 Archivo *reply.rules* - Reglas

```
223 rules(29, 0, [M|_], [  
224     [porque,preguntas,eso,?],  
225     [que,realmete,quieres,saber,?],  
226     [cual,seria,la,respuesta,?],  
227     [que,piensas,sobre,ello,?],  
228     [que,piensas,cuando,preguntas,eso,?],  
229     [le,haz,preguntado,a,alguien,mas,?]]) :-  
230     member(M, [porque,que]).  
231  
232 rules(30, 100, [quit|_], [  
233     [hasta,luego,.,'Nos',vemos,la,proxima,.]]) .
```

Elaborado por: Isaac Espinoza

El asistente virtual recorre el archivo *simplificacion.rules* donde se encuentran sinónimos de las palabras clave de las reglas para ampliar las posibilidades de brindar una respuesta oportuna al usuario:

Figura 18 Archivo *simplificacion.rules* – Reglas de simplificación

```
1 % Milu - reglas de simplificación  
2  
3 sr([nota|X],[notas|Y],X,Y).           % notas = nota = calificacion = calificaciones  
4 sr([calificacion|X],[notas|Y],X,Y).  
5 sr([calificaciones|X],[notas|Y],X,Y).  
6 sr([constante|X],[regular|Y],X,Y).  
7 sr([estudiantes|X],[estudiante|Y],X,Y).  
8 sr([alumno|X],[estudiante|Y],X,Y).  
9 sr([alumnos|X],[estudiante|Y],X,Y).  
10 sr([atraso|X],[atrasado|Y],X,Y).  
11 sr([recalificar|X],[recalificacion|Y],X,Y).  
12 sr([cambios|X],[cambio|Y],X,Y).  
13 sr([cambiar|X],[cambio|Y],X,Y).  
14 sr([intercambio|X],[cambio|Y],X,Y).  
15 sr([modo|X],[modalidad|Y],X,Y).  
16 sr([carreras|X],[carrera|Y],X,Y).  
17 sr([syllabus|X],[syllabus|Y],X,Y).  
18 sr([compendio|X],[syllabus|Y],X,Y).  
19 sr([sumario|X],[syllabus|Y],X,Y).  
20 sr([silabo|X],[syllabus|Y],X,Y).
```

Elaborado por: Isaac Espinoza

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- El asistente virtual propuesto cumple con el objetivo de reducir la desinformación sobre los trámites de la facultad de ingeniería de la PUCE porque puede brindar una mejor atención a los estudiantes en cuanto a la dudas que se les puedan generar.
- El sistema experto también genera otros beneficios para la institución (PUCE), como servir de apoyo para que el personal de la secretaría de la facultad de ingeniería pueda procesar las solicitudes y trabajar en otros aspectos administrativos con mayor eficiencia, puesto que, responder las consultas sobre información de trámites estaría solventado con el sistema experto.
- Después de analizar lenguajes de desarrollo para inteligencia artificial, se concluye que el lenguaje PROLOG facilita la programación debido a la sencillez de su sintaxis. De esta manera, el programador puede cometer menos errores en comparación con otros lenguajes como C++ o Java. Además, es clara la lectura que se puede dar a los programas aún si no se había utilizado PROLOG anteriormente.
- El flujo de proceso para el desarrollo de inteligencia artificial, como metodología informática, fue la base para el desarrollo del sistema experto. Este flujo fue principalmente diseñado para tratar Big Data, por lo que, se manejó la base de conocimiento del asistente virtual con estas bases y el componente de inteligencia artificial para el desarrollo de la lógica del sistema experto.

- En el proyecto, la entrevista como técnica de muestreo selectivo fue fundamental para la recolección de los datos. Estos se prepararon con el proceso Extracción, Transformación y Carga de Datos; dentro el flujo de proceso para el desarrollo de inteligencia artificial.

5.2. RECOMENDACIONES

- Se recomienda considerar implementar asistentes virtuales en empresas educativas, principalmente para facilitar la manera en la que se puede informar. Siempre se debe recordar que los beneficios que pueda generar un asistente virtual tienen que estar orientados a los objetivos de la empresa.
- Es recomendable documentar correctamente la información y tratarla con procesos como ETL para generar una base de conocimiento bien trabajada. Entonces, un asistente virtual es capaz de generar una amplia variedad de respuestas con el fin de satisfacer todos los requerimientos posibles de los usuarios.
- Para el desarrollo de la base de conocimiento de un agente virtual, la metodología informática utilizada debe enfocarse en la calidad del manejo de los datos. De esta manera, los datos obtenidos podrán ser analizados de una manera sencilla y entonces generar las reglas de negocio que pondrán al sistema experto en marcha.
- Se recomienda seguir apostando por soluciones digitales que mejoran la eficiencia de los procesos y la competitividad de la empresa. De esta forma, en el país se reducirá la latente brecha tecnológica.

BIBLIOGRFÍA

- Abata Quinchuqui, C. P., & Ramírez Soria, P. D. (07 de 2018). *Repositorio Institucional de la Universidad Politécnica Salesiana*. Obtenido de Desarrollo de un prototipo de asistente virtual para la gestión del conocimiento de una organización.: <https://dspace.ups.edu.ec/handle/123456789/15875>
- Badaró, S., Agüero, M. J., & Agüero, M. J. (2013). *Sistemas Expertos*. Obtenido de Dialnet: <https://dialnet.unirioja.es/servlet/articulo?codigo=4843871>
- Brita Inteligencia Artificial. (24 de 11 de 2020). *¿Qué es un Asistente virtual?* Obtenido de <https://brita.mx/que-es-un-asistente-virtual>
- da Silva, D. (17 de 05 de 2021). *¿Qué es un roadmap?: guía para hacer tu hoja de ruta*. Obtenido de <https://www.zendesk.com.mx/blog/roadmap-que-es/>
- Fernández, E. P. (2016). *BIG DATA EJE ESTRATÉGICO EN LA INDUSTRIA AUDIOVISUAL*. Barcelona: UOC.
- García-Holgado, A., & García-Peñalvo, F. J. (04 de 2015). *Repositorio Documental GREDOS*. Obtenido de Estudio sobre la evolución de las soluciones tecnológicas para dar soporte a la gestión de la información: <https://gredos.usal.es/handle/10366/125415>
- Gil Suárez, S. M., & Martínez Buitrago, D. I. (15 de 06 de 2013). *El uso de Prolog en el aula: de lógica a inteligencia artificial*. Obtenido de <https://revistas.udistrital.edu.co/index.php/vinculos/article/view/6516>

- González Ávila, N., López Martínez, I., & Hernández García, N. (09 de 03 de 2020). *Repositorio Tecnm Orizaba* . Obtenido de Aproximación al Análisis de Benchmark sobre Asistentes Virtuales: <http://repositorios.orizaba.tecnm.mx:8080/xmlui/handle/123456789/506>
- Gortázar Bellas, F., Martínez Unanue, R., & Fresno Fernández, V. (2016). *Lenguajes de Programación y Procesadores*. Madrid: Centro de Estudios Ramon Areces SA.
- Guijarro, P. M. (28 de 01 de 2020). *RUA*. Obtenido de Asistente virtual para un sistema de información: <http://rua.ua.es/dspace/handle/10045/101934>
- International Virtual Assistants Association. (Consultado: 1 de 05 de 2022). *LA ÚNICA ASOCIACIÓN COMERCIAL MUNDIAL PARA ASISTENTES VIRTUALES*. Obtenido de <https://ivaa.org/>
- Jiménez Álvarez, C. J. (2019). *Biblioteca Digital Universidad de Alcalá*. Obtenido de Desarrollo de servicios para asistentes virtuales: <https://ebuah.uah.es/xmlui/handle/10017/38511>
- Jones Pérez, K. M., & Yong Morales, G. A. (2007). *Introducción al lenguaje de programación lógica Prolog*. Obtenido de <http://www.di-mare.com/adolfo/cursos/2007-2/pp-prolog.pdf>
- Mares, M. (02 de 05 de 2017). *Prolog Eliza - psychotherapist chatbot*. Obtenido de <https://github.com/maresmar/eliza>
- Martínez Carpio, J. A. (09 de 2019). *Repositorio Institucional de la Universidad de Guayaquil* . Obtenido de Desarrollo de un Asistente Virtual (Chatbot) para la automatización de la atención al Cliente.: <http://repositorio.ug.edu.ec/handle/redug/45108>
- Mayer-Pujadas, M. A., Ávila-Tomás, J. F., & Quesada-Varela, V. J. (01 de 2021). *Science Direct*. Obtenido de La inteligencia artificial y sus aplicaciones en medicina II: importancia actual

y

aplicaciones

prácticas:

<https://www.sciencedirect.com/science/article/pii/S0212656720301463>

Medina, J., Eisman, E. M., & Castro, J. L. (2013). *redined*. Obtenido de Asistentes virtuales en plataformas 3.0: <https://redined.educacion.gob.es/xmlui/handle/11162/98052>

Nicomedes Nieto, E. T. (2018). *TIPOS DE INVESTIGACIÓN*. Obtenido de <https://core.ac.uk/download/pdf/250080756.pdf>

PUCE. (Consultado: 12 de 06 de 2022). *IGNATIUS*. Obtenido de Sistema de Información y Estadísticas: <https://www.puce.edu.ec/ignatius/mision-vision.php>

Real Academia Española. (Consultado: 1 de 05 de 2022). *asistente, ta*. Obtenido de <https://dle.rae.es/asistente>

Rouhiainen, L. (2018). *INTELIGENCIA ARTIFICIAL*. Obtenido de https://static0planetadelibroscom.cdnstatics.com/libros_contenido_extra/40/39308_Inteligencia_artificial.pdf

Sánchez, S. R. (2009). *El Nacimiento de una nueva Generación de Ordenadores: La Quinta*. Obtenido de <http://rabida.uhu.es/dspace/bitstream/handle/10272/3230/b15141524.pdf?sequence=1>

Suárez, J. D. (2000). *Técnicas de inteligencia artificial aplicadas al análisis de la solvencia empresarial*. Obtenido de Repositorio Institucional de la Universidad de Oviedo: <https://digibuo.uniovi.es/dspace/handle/10651/45810>

Trigas Gallego, M. (2012). *Metodología Scrum*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

GLOSARIO DE TÉRMINOS

Inteligencia Artificial (IA): Capacidad de los computadores para aprender a tomar una decisión mediante el uso de sus algoritmos (Rouhiainen, 2018).

Machine Learning: Disciplina del campo de la inteligencia artificial donde se intenta deducir conocimiento con razonamiento lógico partiendo de datos (Suárez, 2000).

Asistente Virtual: Sistemas basados en la unión de la inteligencia artificial y humana con la interacción en lenguaje natural con los usuarios (González Ávila, López Martínez, & Hernández García, 2020).

Usuario: Se refiere a una persona que tiene el derecho de usar algo con cierto tipo de limitación (RAE, 2022).

Prolog: Lenguaje orientado a la programación lógica generalmente utilizado para implementar sistemas expertos e inteligencia artificial (Jones Pérez & Yong Morales, 2007).

ID: Es usado como membrete o abreviación de la palabra identificación o identificador.

Soluciones digitales: Forma de dar solución a un problema mediante el uso de herramientas de software (García-Holgado & García-Peñalvo, 2015).

Anexo B: Archivo de Información del Sistema Experto Milu

Figura 20 Archivo *README.md* – Información del Funcionamiento de Milu

```
1 # milu
2 Prolog Milu - asistente virtual para trámites de la facultad de ingeniería de la PUCE.
3
4 ## Descripción del programa
5 Milu es una agente virtual que se comunica con lenguaje natural humano.
6 Puedes consular información sobre trámites de la facultad de ingeniería de la PUCE.
7 El asistente virtual se comunica en castellano.
8
9 El aplicativo se desarrollo en Prolog. Este grupo de comandos ejecutará Milu:
10
11 ~~~
12 ?- consult('directorio/milu.pl').
13 ?- milu.
14 ~~~
15
16 Después de iniciar, milu envía un saludo y le preguntará al usuario por alguna pregunta...
17
18 ~~~
19 Hola, soy Milu, Con que informacion te puedo ayudar?
20 > [pregunta del usuario]
21 ~~~
22
23 El programa puede finalizar cuando el usuario se despide....
24
25 ~~~
26 > chao.
27 milu: Hasta luego. Nos vemos la proxima.
28 ?- true.
29 ~~~
```

Elaborado por: Isaac Espinoza