



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO DE FIN DE MÁSTER

Máster en Ingeniería Informática

Propuesta Metodológica para el análisis y diseño de Chatbots basados en texto

Autor:

Juan Alonso Astruga

Tutores:

Alejandra Martínez Monés

Eduardo Gómez Sánchez

Titulo:

Propuesta metodológica para el análisis y diseño de chatbots basados en texto

Autor:

Juan Alonso Astruga

Tutores:

Alejandra Martínez Monés

Eduardo Gómez Sánchez

Agradecimientos

Antes de comenzar el trabajo, quería dar las gracias a todos los que me han ayudado a terminar esta etapa de mis estudios. En lo académico, gracias a todos los profesionales de la ETSII que se han esforzado para completar nuestra formación como ingenieros. Especialmente quería dar las gracias a los tutores de este Trabajo Fin de Máster, Alejandra y Eduardo, por todas las dudas resueltas, el apoyo y el feedback en la realización de este trabajo. En lo personal, quería agradecer a mi familia por tener la comprensión y paciencia para apoyarme durante el máster.

Gracias a todos.

Resumen

Los *chatbot* basados en texto son agentes conversacionales aplicados ampliamente en la interacción con el usuario, sobre todo para la respuesta a preguntas en torno a productos o servicios. En la actualidad existen muchas propuestas tecnológicas que apoyan el desarrollo de estos sistemas, aportando servicios basados en inteligencia artificial, etc. Sin embargo no hay metodologías usadas en la actualidad realmente para llevar a cabo su análisis y diseño. Por ello se propone una metodología para la elicitación de requisitos y diseño de chatbots de preguntas y respuestas. En este trabajo se presenta IMADTC (Iterative Methodology for the Analysis and Development of Textual Chatbots) que pretende dar respuesta a los problemas relacionados con el desarrollo de un *chatbot*, de forma que a través de la selección de técnicas necesarias de elicitación de requisitos se pueda llevar a cabo la creación de un *chatbot* textual orientado a preguntas y respuestas. La propuesta es validada mediante su uso en el diseño de un *chatbot* de este tipo y sobre la que se realizan recomendaciones de aplicación y mejoras a la propuesta.

Índice general

1. Introducción	1
1.1. Presentación	1
1.2. Motivación	1
1.3. Objetivos	2
1.4. Metodología	2
1.5. Contenido del documento	3
2. Chatbots: Concepto y retos de desarrollo	5
2.1. Agentes conversacionales y chatbots: concepto	5
2.2. Uso de chatbots basados en texto en distintos contextos	5
2.3. Chatbots en el contexto educativo	6
2.4. Retos para el desarrollo de chatbots	7
2.4.1. Retos Tecnológicos	7
2.4.2. Retos Conceptuales	7
2.4.3. Retos específicos al caso VirtualForest	9
2.5. Análisis de alternativas tecnológicas al uso de Chatbots	9
2.5.1. Criterios del análisis	10
2.5.2. Sistemas considerados y análisis realizado	10
2.5.3. Conclusión del análisis tecnológico	20
2.6. Metodologías de desarrollo de chatbots	20
2.6.1. Metodologías orientadas hacia el desarrollo del sistema	20
2.6.2. Conversation Driven Development	20
2.7. Conclusión del capítulo	23
3. Análisis de técnicas de elicitación de requisitos en chatbots	25
3.1. Clasificación de técnicas de elicitación de requisitos	25
3.2. Selección de técnicas	26
3.2.1. Técnicas de modelado	27
3.2.2. Técnicas de prototipado	28
3.2.3. Técnicas Agile	28
3.3. Selección de técnicas de análisis de requisitos para chatbots	28
3.3.1. Técnicas a aplicar en la variabilidad del lenguaje	29
3.3.2. Técnicas a aplicar en el control del diálogo	29
3.4. Criterios de validación de las técnicas seleccionadas	30
3.4.1. User stories	30
3.4.2. Escenarios	30
3.4.3. Uso de Prototipos	31
3.4.4. Laddering	31
3.5. Conclusión de la selección de técnicas	32

4. Propuesta Metodológica	33
4.1. Objetivos a cumplir con la propuesta	33
4.2. Estructura de la Propuesta IMADTC	33
4.3. Validación de la propuesta	36
4.3.1. Validación de las técnicas incorporadas en la metodología	36
4.3.2. Validación de la información obtenida por la aplicación de la metodología	38
4.4. Conclusión de la metodología	38
5. Aplicación y Evaluación de la propuesta	39
5.1. Organización del trabajo	39
5.2. Contextualización de los participantes en los test de usuario	39
5.3. Artefactos elaborados por la aplicación de la propuesta	40
5.3.1. Contextualización del sistema	40
5.3.2. Selección de objetivos a través de user stories	40
5.3.3. Diseño de escenarios	44
5.3.4. Diseño del Flujo Conversacional	48
5.4. Análisis del sistema	61
5.4.1. Requisitos funcionales del sistema	61
5.4.2. Requisitos No Funcionales	64
5.4.3. Requisitos de Información	65
5.4.4. Actores del sistema	65
5.4.5. Casos de Uso	66
5.4.6. Diagrama de clases	71
5.5. Desarrollo del sistema	71
5.6. Pruebas del sistema	72
5.6.1. Pruebas de Caja Blanca	72
5.6.2. Pruebas de Caja Negra	72
5.7. Evaluación de la propuesta	82
5.7.1. Evaluación de las técnicas de elicitación de requisitos utilizadas	82
5.7.2. Evaluación general de la aplicación de la propuesta	85
5.8. Discusión	87
5.9. Conclusión del capítulo	89
6. Conclusiones y trabajo futuro	91
Bibliografía	92
Apéndice A: Acrónimos	100
Apéndice B: Código del backend del prototipo de VirtualForest	100

Índice de figuras

4.1. Esquema del flujo de fases a llevar a cabo en la metodología y los artefactos producidos en cada una	34
5.1. Diagrama de Flujo Conversacional 1: Listar la oferta de másteres	48
5.2. Diagrama de Flujo Conversacional 2: Consultar condiciones acceso a un máster	49
5.3. Diagrama de Flujo Conversacional 3: Consultar cronograma de un máster	50
5.4. Diagrama de Flujo Conversacional 4: Consultar información sobre créditos ECTS	51
5.5. Diagrama de Flujo Conversacional 5: Consultar información sobre requisitos de idioma de un máster	52
5.6. Diagrama de Flujo Conversacional 6: Consultar información sobre requisitos de lenguajes de programación de un máster	53
5.7. Diagrama de Flujo Conversacional 7: Consultar información sobre modalidad de presencialidad de un máster	54
5.8. Diagrama de Flujo Conversacional 8: Consultar información sobre precios orientativos de un máster	55
5.9. Diagrama de Flujo Conversacional 9: Consultar contenido de un máster	56
5.10. Diagrama de Flujo Conversacional 10: Consultar colaboraciones de un máster	57
5.11. Diagrama de Flujo Conversacional 11: Solicitar reconocimiento de asignaturas	58
5.12. Diagrama de Flujo Conversacional 12: Solicitar información de prácticas en empresa	59
5.13. Diagrama de Flujo Conversacional 13: Solicitar información de becas para un máster	60
5.14. Diagrama de Flujo Conversacional 14: Solicitar información de contacto para un máster	61
5.15. Diagrama de casos de uso	70
5.16. Diagrama de clases	71
5.17. Estructura del sistema	71
5.18. Informe Resumen de SonarQube	73
5.19. Esquema del flujo de fases a llevar a cabo en la propuesta de mejora de metodología final. En azul se muestra la ampliación propuesta.	88

Índice de tablas

1.1. Tabla análisis bibliográfico	4
2.1. Tabla análisis de Rasa	11
2.2. Tabla análisis de ChatScript	12
2.3. Tabla análisis de Chaterbot	13
2.4. Tabla análisis de Pandorabots	14
2.5. Tabla análisis de Wit.ai	15
2.6. Tabla análisis de IBM Watson Assistant	16
2.7. Tabla análisis de DialogFlow	17
2.8. Tabla análisis de Azure Bot Services	18
2.9. Tabla análisis de Amazon Lex	19
5.1. Tabla de evaluación de user stories	83
5.2. Tabla de evaluación de escenarios	84
5.3. Tabla de evaluación de funcionalidades prototipadas	84
5.4. Tabla de evaluación de aplicación de laddering	85

Capítulo 1

Introducción

En este capítulo se da una explicación de los conceptos introductorios necesarios para entender el por qué del trabajo realizado, en qué consiste el trabajo, los objetivos marcados como parte de la investigación y la metodología de investigación seguida para ello.

1.1. Presentación

Un *chatbot* textual es un agente conversacional que intenta simular a través de distintos medios una conversación, en lenguaje natural, con un usuario. Los *chatbot* no son una tecnología nueva, son una tecnología que está resurgiendo últimamente, tanto por el servicio de preguntas y respuestas en las aplicaciones de mensajería instantánea, como por los avances en precisión que ofrecen las tecnologías de inteligencia artificial y reconocimiento del habla. En estos sistemas en que se da una interacción constante con el usuario final, un diseño de interacción es de vital importancia. Se requiere llevar la imagen que tiene el usuario del funcionamiento del *chatbot* a cómo funciona el *chatbot* en realidad. Para ello, y con el objetivo de garantizar un resultado viable, será necesario seguir una metodología, es decir un proceso probado y refinado que nos asegure tal resultado. Actualmente no hay un consenso por parte de la comunidad sobre qué proceso, independiente de la tecnología, se debe llevar a cabo para la construcción de un sistema de diálogo basado en *chatbots*. En respuesta a esta carencia, este trabajo presenta una propuesta metodológica para el análisis y diseño de *chatbots*, y más en concreto, para *chatbots* orientados a preguntas y respuestas. En el presente documento se explica el proceso de investigación seguido, desde la identificación de los problemas a los que debe dar solución, las técnicas existentes para resolver estos problemas y finalmente la propuesta metodológica a seguir.

1.2. Motivación

El trabajo del TFM se ha realizado en el ámbito del proyecto VirtualForest [1]. VirtualForest es un proyecto europeo financiado por el programa Erasmus+, dentro de una llamada específica para promover la virtualización de estudios superiores¹, en el que participa un consorcio internacional liderado por la Universidad de Valladolid. Por ésta, a su vez, participan el GSIC/EMIC (Grupo de Sistemas Inteligentes y Cooperativos / Educación, Medios, Informática y Cultura)², al que pertenecen los tutores de este trabajo, y el IUFOR (Instituto Universitario de Investigación y Gestión Forestal Sostenible)³. El objetivo que tiene el proyecto VirtualForest es contribuir a la virtualización de las actividades relacionadas con Ciencias Forestales a una escala internacional. Para ello, uno de los paquetes de trabajo del proyecto (en esta convocatoria denominados “resultados intelectuales”– *intellectual outputs*) incluye el desarrollo de un *chatbot* basado en texto que ayude al alumnado en tareas administrativas y relacionadas con las asignaturas de los programas universitarios de máster cuyo contenido tiene relación con Ciencias Forestales. En concreto, uno que ayude en cuestiones administrativas a alumnos de grado interesados en los programas de máster del ámbito forestal de la Escuela Técnica Superior de Ingenierías Agrarias de Palencia. Por

tanto el *chatbot* planteado ya se encuadra en la definición del proyecto dentro de la categoría de *chatbots* de preguntas y respuestas. De el paquete de trabajo del proyecto VirtualForest descrito es responsable el grupo GSIC/EMIC, arriba mencionado. Éste es un grupo interdisciplinar que está compuesto por profesores, investigadores y personal laboral de la Escuela Técnica Superior de Ingenieros de Telecomunicación, la Escuela de Ingeniería Informática y la Facultad de Educación y Trabajo Social de la Universidad de Valladolid. Es un Grupo de Investigación Reconocido (GIR) por la Universidad de Valladolid y está reconocido como Unidad de Investigación Consolidada (UIC) por la Junta de Castilla y León. El GSIC/EMIC tiene una trayectoria de más de 25 años en investigación en tecnología para el apoyo a la educación, y recientemente han iniciado una línea relacionada con agentes conversacionales, dentro de la que se encuadra el contenido de esta actividad.

El diseño de un *chatbot* depende mucho del dominio de conocimiento, y el contexto de comunicación que se quiera implementar. Es, como se ha dicho, muy importante traducir las necesidades del usuario al funcionamiento del sistema a través de la aplicación de una metodología que nos ayude a llegar al resultado deseado. Sin embargo, en el momento de realizar el análisis de requisitos en el proyecto VirtualForest, nos encontramos con dificultades como consecuencia, por un lado, de que el objetivo del sistema no estaba del todo definido en el propio proyecto, y por otro, que los clientes no tenían claro qué contenido querían darle al *chatbot* ni entendían las demandas de definición de éste que se les transmitieron. Debido a que no hay actualmente consenso sobre la propuesta metodológica a aplicar a la hora de llevar a cabo este proceso para este tipo de sistema basado en preguntas y respuestas, se propuso el desarrollo de una propuesta que permita alcanzar el objetivo de obtener un *chatbot* con las características descritas.

1.3. Objetivos

Teniendo en cuenta la motivación anteriormente presentada, el principal objetivo planteado de este trabajo es: *Realizar una propuesta metodológica para el análisis de requisitos y diseño de chatbots de preguntas y respuestas.* Este objetivo se desgrana en las siguientes actividades específicas:

- Identificar el papel de un *chatbot* en un contexto educativo.
- Identificar y analizar los retos existentes en el diseño de *chatbots* y sus consecuencias en relación a la especificación de requisitos.
- Identificar y analizar técnicas de análisis y de diseño más apropiadas para afrontar los retos de diseño de chatbots detectados.
- Interpretar la forma de articular las técnicas seleccionadas en una metodología.
- Validar la propuesta a través de la puesta en práctica de la misma en un caso concreto. Este caso será el de Virtual Forest descrito anteriormente.

1.4. Metodología

La metodología utilizada para el desarrollo del trabajo se corresponde con lo descrito en el método de investigación en la ingeniería [2]. En este método se itera obteniendo resultados cada vez más correctos que resuelven diferentes problemas. El método se divide en las siguientes etapas por las que se itera repetidamente hasta obtener una solución aceptable :

- Observación de las soluciones ya existentes.

¹Call 2020 Round 1 KA2 - Cooperation for innovation and the exchange of good practices KA226 - Partnerships for Digital Education Readiness.

²<https://www.gsic.uva.es/index.php?lang=es>

³iuFOR: <http://sostenible.palencia.uva.es/>

- Propuesta de soluciones que mejoran las anteriores.
- Desarrollo de la propuesta.
- Medición y análisis de la solución obtenida.

En el caso de este trabajo, para la etapa de observación de las soluciones existentes, se ha seguido un proceso de análisis bibliográfico (ver caps. 1, 2 y 3). Partiendo del análisis bibliográfico se ha realizado la identificación del problema a tratar y la propuesta de técnicas que puedan darle solución. Con esta información, se ha desarrollado la propuesta. La etapa de propuesta de soluciones ha consistido en una propuesta de metodología, para el análisis y diseño de *chatbots* de preguntas y respuestas, en base a los problemas identificados (ver cap. 4). Esta propuesta se ha aplicado en el desarrollo de un *chatbot*, para el caso Virtual Forest, como parte de la fase de desarrollo de la propuesta (ver cap. 5) y finalmente se ha obtenido la medición de su eficacia, analizando sus puntos fuertes y débiles a través de la evaluación de dicha experiencia de aplicación en Virtual Forest (ver caps. 5 y 6).

El análisis bibliográfico se ha llevado a cabo en varias ocasiones para los diferentes temas estudiados en el trabajo. El análisis se ha realizado a partir de búsquedas en las siguientes bases de datos bibliográficas: ACM Digital Library, Almena UVa, ResearchGate, Google Scholar, Microsoft Academic, IEEE Xplore. Esta combinación de fuentes se ha seleccionado por permitir abarcar tanto, desde un punto de vista centrado en la informática (ACM Digital Library, IEEE Xplore), como desde una perspectiva general (ResearchGate, Google Scholar, Microsoft Academic, Almena), los temas a investigar en cada caso.

Para cada una de las temáticas que se tratan en el trabajo se ha aplicado, a cada una de las bases de datos bibliográficas anteriores, una serie de cadenas y términos específicos para dicha temática. Se han filtrado los 20 resultados más relevantes de cada búsqueda en cada una de las bases de datos mencionadas anteriormente, eliminando duplicados que hubieran aparecido en búsquedas anteriores de otra base de datos. A estos documentos los denominamos “*inspeccionados*” en la Tabla 1.1. A continuación se ha valorado en cada documento considerado, a través de una breve lectura del *abstract*, introducción y resultados de investigación (si había dudas de su relación con la temática tratada), su incorporación al conjunto de información que compone el estudio de cada una de estas temáticas. Para determinar si un documento ofrecía información suficientemente relevante se llevó a cabo la lectura del resto del documento seleccionándolo si se consideraba que era finalmente relevante a lo expuesto en el trabajo. Estos documentos son los que se denomina “*seleccionados*” en la Tabla 1.1. Para llevar a cabo este proceso se ha utilizado el gestor de bibliografía Zotero. En determinadas ocasiones se ha necesitado buscar en específico fuera de este método la existencia de algún documento que pudiese aportar información sobre algún tema concreto.

1.5. Contenido del documento

El contenido del resto del documento es la siguiente:

- En el capítulo 2 se presenta el concepto de *chatbot*, su contextualización en dominios de uso y los roles que toma, así como los problemas tecnológicos y conceptuales existentes a la hora de desarrollar un *chatbot* así como los específicos al caso VirtualForest, las alternativas tecnológicas de implementación y las metodologías existentes para el diseño de *chatbots*.
- En el capítulo 3 se explica la clasificación de técnicas de elicitación de requisitos existentes, la selección adecuada para el problema de desarrollo de *chatbots* de preguntas y respuesta, y los criterios de validación de las técnicas seleccionadas.
- En el capítulo 4 se presenta la propuesta metodológica desarrollada y se da una explicación de los objetivos a cumplir con la propuesta, la estructura de la misma y los criterios de validación para la propuesta.
- En el capítulo 5 se da una explicación de los artefactos producidos, las características resultantes de la aplicación de la propuesta en el caso VirtualForest y la discusión del resultado obtenido.
- En el capítulo 6 se da una conclusión al trabajo realizado con los puntos más importantes del mismo.

<p>Tema: Contextualización de chatbots</p>	<p>Términos de búsqueda: <i>chatbot, application, domain, education</i>. Cadenas de búsqueda: <i>(chatbot AND domain), (chatbot AND application), (chatbot AND education)</i>.</p> <p>Inspeccionados: 30 ([3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28],[29], [30], [31], [32]).</p> <p>Selección: 12 ([5], [17], [19], [22], [23], [25], [26], [27], [28], [30], [31], [32]).</p>
<p>Tema: Metodologías de desarrollo de chatbots</p>	<p>Términos de búsqueda: <i>methodology, dialog, system, agent</i>. Cadenas de búsqueda: <i>(methodology AND (dialog AND (system OR agent))) OR (dialog AND (system OR agent))</i>.</p> <p>Inspeccionados: 18 ([33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51]).</p> <p>Selección: 5 ([41], [51], [50], [34], [49]).</p>
<p>Tema: Problemas asociados al desarrollo de los chatbots</p>	<p>Términos de búsqueda: <i>problem, difficulty, error, chatbot, development, dialog, structure, control, expresion, context, validation</i>. Cadenas de búsqueda: <i>((problem OR difficulty OR error) AND chatbot AND development),(dialog AND structure AND control),(context AND (dialog OR expression) AND validation)</i>.</p> <p>Inspeccionados: 20 ([52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70]).</p> <p>Selección: 16 ([52], [54], [55], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [69], [70].)</p>
<p>Tema: Técnicas de elicitación de requisitos</p>	<p>Términos de búsqueda: <i>software, requirements, elicitation, techniques, review, user story, prototyping, laddering, escenario</i>. Cadenas de búsqueda: <i>(software AND requirements AND elicitation) AND (techniques OR review)</i>.</p> <p>Inspeccionados: 22 ([71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92]).</p> <p>Selección: 12 ([71], [75], [77], [80], [81], [84], [85], [87], [88], [89], [90], [91]).</p>

Tabla 1.1: Tabla análisis bibliográfico

Capítulo 2

Chatbots: Concepto y retos de desarrollo

Para poder proponer una metodología efectiva para la especificación de requisitos de un *chatbot* textual se debe conocer previamente qué papeles va a desarrollar un *chatbot* como sistema, qué problemas han de abordarse en esa metodología y cómo, si es que existe solución, en la bibliografía disponible se han abordado estos problemas en posibles alternativas metodológicas. Por ello en este capítulo se expone el papel de un *chatbot* basado en texto dentro del contexto de tareas relacionadas con preguntas y respuestas en el dominio de la educación, así como los retos existentes a la hora de llevar a cabo el desarrollo de un *chatbot*, las alternativas tecnológicas que se han propuesto para afrontar dichos retos, y las metodologías existentes para el desarrollo de un sistema de este tipo.

2.1. Agentes conversacionales y chatbots: concepto

Un agente conversacional es un programa de software que interpreta y responde a frases enunciadas por los usuarios en lenguaje natural [93]. Los agentes conversacionales son un sistema de diálogo que lleva a cabo procesamiento del lenguaje natural (Natural Language Processing) y responde a la solicitud en un lenguaje natural para un humano. Estos se dividen generalmente en dos tipos: agentes de diálogo orientados a la tarea y *chatbots* [94].

Los agentes de diálogo orientados a la tarea [94] son sistemas programados para completar una tarea en específico que utilizan conversaciones cortas con el usuario para ayudar a completar esta tarea. Estos sistemas suelen seguir un flujo de diálogo, basado en una máquina de estados finita, con una estructura programada de antemano para llevar a cabo la conversación, y donde el sistema determina la dirección de la conversación, hace preguntas a los usuarios y, por lo general, malinterpreta o ignora una entrada que no sea una respuesta directa a la pregunta previamente realizada.

Los *chatbot* [94] son sistemas conversacionales que, a través de técnicas basadas en inteligencia artificial, intentan simular e imitar el flujo no estructurado de la conversación humana. Los *chatbots* son diferentes a los agentes conversacionales dirigidos a la tarea, ya que aprovechan las posibilidades de las técnicas de inteligencia artificial para responder a las conversaciones humanas con un conjunto específico de respuestas predefinidas, coherentes con la situación, para simular comprender la intención del usuario y analizar el contexto del discurso.

2.2. Uso de chatbots basados en texto en distintos contextos

Dada la versatilidad de los *chatbots* a la hora de mantener una conversación y su proliferación durante los últimos años, ha favorecido su aplicación a numerosos contextos de uso. Entre estos contextos se puede citar algunos de los más importantes como servicio al cliente y salud y bienestar físico.

Para el uso de un chatbot dedicado a dar soporte al servicio a clientes, muchas empresas utilizan *chatbots* para

la ayuda a sus consumidores. El uso de un *chatbot* como parte del servicio al cliente permite prestar atención las 24 horas del día a través del *chatbot*, lo que permite a los consumidores realizar solicitudes independientemente del horario de atención estándar, mejorando la satisfacción del usuario. La incorporación de un *chatbot* [31] a un modelo tradicional de atención al cliente puede mejorar la experiencia de éste, principalmente en la medida de la capacidad de respuesta, mientras mantiene un nivel similar [30] en las dimensiones de calidad de la información, calidad del sistema y satisfacción del usuario.

En el caso del contexto de salud y bienestar físico, la inteligencia artificial (IA) se utiliza cada vez más en la asistencia sanitaria. Aquí, los sistemas de chatbot basados en inteligencia artificial pueden actuar como agentes conversacionales automatizados, capaces de promover la salud, e incluso [28] diagnosticar enfermedades en una interacción con los usuarios a través de una interfaz de texto. Otras aplicaciones dentro de este campo permiten la prestación de atención sanitaria rápida [27] en caso de accidentes que pueden ocurrir en la vida cotidiana, y también en respuesta a los cambios de las condiciones de los pacientes con enfermedades crónicas.

Otro ámbito en el que han despertado interés es el de la educación, que se desarrolla en la siguiente sección.

2.3. Chatbots en el contexto educativo

Los chatbots se pueden utilizar en entornos de aprendizaje formales e informales y pueden proporcionar a los alumnos un apoyo adaptado a sus necesidades. Se han propuesto diferentes usos potenciales para los chatbots en contextos educativos. Por ejemplo, los *chatbots* para el apoyo al aprendizaje pueden preservar la información repitiendo lecciones pasadas cuando los estudiantes no pueden asistir a ellas. También pueden recopilar información durante un curso para la mejora del proceso de aprendizaje y la enseñanza. Estos *chatbots* pueden permitir a los estudiantes el acceso a las lecciones en el momento que mejor les convenga, estando disponibles en cualquier horario de estudio y pudiendo responder preguntas relacionadas con el material educativo. Un *chatbot* orientado a la educación puede ayudar a los estudiantes con problemas administrativos relacionados con la inscripción y administración de asignaturas de un curso. Estos ejemplos muestran algunos de los papeles que puede adoptar un *chatbot* en este contexto. De acuerdo a la investigación realizada por Sofie Roos [26], existen roles típicos de los *chatbots* en el contexto educacional: «conversación natural», «comunicación con el profesor o el entorno de aprendizaje», «preguntas y respuestas», y «evaluación y corrección del estudiante».

En el caso de Chatbots con el rol de Conversación Natural, el *chatbot* mantiene una conversación en lenguaje natural con el estudiante con el objetivo de mejorar las habilidades del estudiante en torno a escritura y lectura comprensiva de textos. Es decir, el objetivo es simplemente el forzar al estudiante a mantener una conversación simple con el *chatbot*. Casi siempre esto está, en esta temática, asociado al aprendizaje de un lenguaje distinto a la lengua materna del estudiante. Un ejemplo de esto es el sistema interactivo desarrollado por la Universidad Carlos III de Madrid [25] para crear un mundo 3D de interacción multiusuario en el que las interacciones con el *chatbot* a través de personajes virtuales fuerzan a los estudiantes a mantener conversaciones en una lengua distinta de la propia.

En el rol de Comunicación con el Profesorado, el *chatbot* se usa como medio de interacción de los estudiantes tanto con el medio de aprendizaje como con el profesor. El *chatbot* tiene dos funciones [23]: La primera es recopilar datos de las conversaciones mantenidas con el estudiante sobre el entorno educativo, para mejorar este, y la segunda es intentar por su propia cuenta contestar en primera instancia las preguntas que le dirige el alumno y si no lo realiza con suficiente confianza, trasladar la pregunta o preguntas al profesor.

Bajo el rol de Preguntas y Respuestas el *chatbot* es preparado para contestar una serie de preguntas sobre una temática concreta que el alumno estudiará. En ocasiones también se utiliza para resolver preguntas en torno a cuestiones administrativas [22]. El objetivo aquí no es evaluar el conocimiento del alumno, si no resolver las dudas que les surjan durante su proceso de aprendizaje [5] [19]. Éste es el rol más documentado, sobre el que más se ha trabajado y con más implementaciones de todos los descritos. El trabajo de estos *chatbots* suele ser simple porque los conocimientos ya están programados con anticipación. Los métodos de implementación utilizados en esta aplicación suelen incluir la coincidencia de patrones, el procesamiento del lenguaje natural y la minería de

datos. El *chatbot* haría coincidir la oración de entrada del hablante o usuario con ese patrón existente en la base de conocimientos. Luego, cada patrón se compara con el conocimiento del *chatbot*.

Para el caso de Evaluación del estudiante [17], el objetivo a perseguir es que se evalúe el progreso del estudiante en el aprendizaje de una materia en concreto al mismo tiempo que se corrigen los errores cometidos por el alumno en el aprendizaje de la tarea y se da *feedback* al profesor de los progresos de cada estudiante en la misma.

Dado que el trabajo a realizar como parte del TFM está relacionado con un *chatbot* en el contexto educativo, comprender éste supone un punto importante en el trabajo a realizar. Esta información sobre el rol que puede desempeñar el *chatbot* en el contexto de educación será tenida en cuenta en una parte posterior de este documento sobre la contextualización del *chatbot* desarrollado. Nos centraremos en el rol de los *chatbots* de preguntas y respuestas, por ser el rol del *chatbot* del caso que ha motivado el trabajo y porque son el tipo de *chatbot* que más se llega a implementar dentro de los roles que puede llevar a cabo un *chatbot* como sistema conversacional.

2.4. Retos para el desarrollo de chatbots

El desarrollo de *chatbots* requiere experiencia en áreas especializadas, como el aprendizaje automático y el procesamiento del lenguaje natural, lo que lo distingue del desarrollo de software tradicional. Es por tanto importante el comprender los problemas a los que se enfrenta un desarrollador de este tipo de sistemas antes de hacer una propuesta para su diseño, que permita incrementar la calidad del chatbot y su adopción por usuarios finales. Para ilustrar estos problemas nos vamos a basar en el trabajo presentado por Abdellatif y cols. [52], donde se estudió la cantidad de preguntas formuladas por desarrolladores en foros de ayuda en línea como *StackOverflow* en torno a los problemas más comunes y de mayor dificultad de resolución en el diseño de *chatbots*. Los problemas encontrados se han dividido en dos grupos según su causa: retos tecnológicos y retos conceptuales. Los retos tecnológicos son causados por el uso de la tecnología seleccionada para el desarrollo del *chatbot*. Los retos conceptuales por el contrario se derivan de una mala interpretación del contexto y la información relevantes para el *chatbot*. A continuación se exponen estos dos grupos de problemas y finalmente se exponen los problemas específicos del caso de VirtualForest.

2.4.1. Retos Tecnológicos

Los retos tecnológicos abarcan las categorías de integración, desarrollo y NLU. Estos incluyen: Problemas derivados de la integración con plataformas tecnológicas, problemas derivados con la programación del desarrollo del propio *chatbot* y problemas derivados del NLU. La categoría de **integración** recoge los temas de las integraciones entre distintos servicios y la plataforma de *chatbot*, así como la integración del *chatbot* desplegado y los entornos web. Dentro de esta categoría están los problemas relacionados con integración de tecnologías específicas y con el uso de API para realizar la comunicación entre el *chatbot* y canales externos al mismo. La categoría de **desarrollo** abarca problemas relacionados con la construcción de *chatbots* usando diferentes plataformas y *frameworks* de desarrollo, problemas sobre configuraciones y características determinadas de la plataforma escogida, e implementaciones específicas usando esas plataformas. En la categoría de **NLU** se agrupan los problemas relacionados con la especificación de *intents* y *entities*, así como su uso y configuración. También abarca la personalización y configuración de técnicas de *NLU* en las plataformas de desarrollo, y de mejora del rendimiento en el entrenamiento de los modelos para *NLU*. Viendo la naturaleza de los problemas tecnológicos se decidió que la forma de evitarlos en la medida de lo posible a la hora de realizar un desarrollo de *chatbots* basado en estos servicios sería el realizar un análisis de las alternativas para el desarrollo de *chatbots* actualmente existentes (ver sección 2.5).

2.4.2. Retos Conceptuales

De los cinco problemas principales identificados, los tres primeros problemas han sido de ámbito tecnológico, basándose su solución en escoger la tecnología más adecuada para cada situación. Los dos últimos problemas identificados (la variabilidad de las entradas proporcionadas por el usuario y el control del diálogo en la conversación con el usuario), por el contrario, basan su dificultad no en un problema de la plataforma, como en el caso anterior,

sino en la potencial mala comprensión de la intención del usuario al interactuar con el sistema, un problema que se debe tratar en el proceso de análisis del mismo con técnicas de elicitación de requisitos y con una metodología de diseño adecuada.

Variabilidad del lenguaje

El primer problema detectado que puede ocasionar errores en un proceso de elicitación de requisitos es la variabilidad de las formas de expresarse que puede usar el usuario para comunicar una acción al *chatbot*, especialmente en el caso de los sistemas basados en el NLU como método de aprendizaje y procesamiento de entradas del usuario. A la hora de controlar la variabilidad del lenguaje hay que tener en cuenta la influencia del contexto en la educación lingüística del usuario. La naturaleza y magnitud de los efectos ambientales, del contexto, tiene diferentes grados de influencia en el desarrollo del lenguaje que adquiere una persona [59]. Al igual que otros aspectos del comportamiento interpersonal, el uso del lenguaje se socializa para que coincida con las expectativas de la comunidad en cada momento. No solo eso, sino que a lo largo de la vida, características personales, por ejemplo, edad, alfabetización, habilidades del lenguaje, etc. modulan los efectos del contexto en la comprensión y uso del lenguaje por parte de una persona [60] y por tanto influyen en la selección de expresiones, palabras y la forma de articular una frase para expresar una idea. La variabilidad del tono adecuado con que el *chatbot* se comunica con el usuario está relacionado con la variabilidad del lenguaje, ya que la introducción de más de un tono y de formas de expresarse dentro de una misma conversación con el usuario puede conducir a que este responda también en múltiples tonos incrementando no solo la variabilidad de las frases de entrada que el *chatbot* reciba sino la confusión del propio usuario [61]. Esto implica que el uso de un cierto registro [62] en un cierto contexto de conversación pueden aumentar la comprensión y satisfacción del usuario al usar el *chatbot*.

La variabilidad de las expresiones usadas para llevar a cabo la comunicación con el *chatbot* suponen un problema a la hora de realizar una especificación de requisitos. Formas de expresarse en lenguaje natural que son equivalentes para el usuario pueden provocar resultados diferentes bajo la interpretación del *chatbot*, dependiendo de su preparación previa, que puede causar una especificación de requisitos sobre declaraciones ambiguas por parte del usuario de qué es lo que quiere conseguir en su interacción con el sistema [63]. Dado que en la mayoría de los casos no es posible llevar a cabo una especificación de todas las posibilidades de expresión por parte del usuario de una intención al interactuar con el *chatbot*, una posible solución [64] consiste en asignar una probabilidad como requisito no funcional bajo la que se reconocería un número de veces determinado la intención del usuario de forma correcta a la hora de comunicarse con el *chatbot*. Aunque técnicamente este es un requisito medible y verificable, puede distar del resultado esperado por el cliente según el conjunto de frases de prueba y las variaciones introducidas en ellas para demostrarlo, por lo que no se puede considerar como una solución completa.

Control del diálogo

El segundo problema a solventar es cómo controlar la variabilidad en la estructura que se da en un diálogo si se está utilizando éste como herramienta para conseguir un objetivo concreto. La mayoría de las veces los participantes de un diálogo lo inician con el objetivo de lograr algún objetivo subyacente. Tales metas son de naturaleza no comunicativa, y lograrlas requiere más que utilizar solo la comunicación. También requiere sopesar y comparar información y decidir cómo se va a guiar la conversación para obtener la meta, en función de la información proporcionada y de las sub-metas que se vayan generando en el desarrollo del diálogo [65].

Al llevar a cabo un proceso de elicitación de requisitos es difícil describir de forma clara la necesidad de mantener un contexto de diálogo con el sistema que determine como evoluciona la conversación. Desde el punto de vista de análisis bajo UML esto se suele reservar a diagramas de actividad o del estilo de máquinas de estado [66], o a representaciones donde el diálogo se desarrolla sobre un contexto específico y el número de transiciones de información entre distintos posibles estados es conocido y por tanto controlable con mayor o menor esfuerzo. En estas formas para especificar al completo el diálogo a llevarse a cabo para conseguir un objetivo concreto, si el dominio del diálogo es cambiante, la completitud será un problema añadido a resolver. El diálogo además será específico del idioma, dependiente de la cultura, y de otros aspectos contextuales, en particular en aspectos sociales

como el marco institucional en el que ocurre el diálogo, y el tipo de evento comunicativo que representa el diálogo. Para evitar métodos en los que haya que encontrar todas las posibilidades de diálogo, una opción consiste en utilizar métodos de predicción como los basadas en Markov [67] donde dependiendo del estado anterior del sistema y un conjunto de probabilidades obtenidas que determinan la acción a realizar es posible determinar el estado subsiguiente en el diálogo [69]. Por tanto, el reconocimiento del contexto, cambiante o no, como punto de influencia en la comprensión del diálogo [70] es un factor que se debe tener en cuenta a la hora de resolver este problema.

Los dos problemas expuestos a la hora de analizar los requisitos para el desarrollo de un *chatbot* comparten la característica de depender de las circunstancias que influyen en el usuario en el momento determinado de interactuar con el *chatbot*. Esto es así tanto para expresarse de una forma específica frente a otras opciones, como para en una conversación decantarse por una opción que guíe el diálogo frente a otras. Es por eso importante que se preste atención al contexto conversacional en el que se encuentra inmerso el usuario para, de esta forma, poder limitar las opciones posibles que baraje a la hora de comunicarse con el *chatbot*. Esto tendrá que ser tenido en cuenta a la hora de seleccionar técnicas de elicitación de requisitos apropiadas para nuestro problema.

2.4.3. Retos específicos al caso VirtualForest

Además de los problemas conceptuales anteriormente descritos, el trabajo a realizar en un desarrollo de sistemas basados en *chatbots* textuales orientados a una tarea de pregunta y respuesta tiene, en nuestro caso, estos problemas por parte de los clientes que las técnicas deben tener en cuenta:

- Poca claridad a la hora de delimitar las tareas que debe y no debe realizar el *chatbot* proyectado.
- Poca claridad a la hora de delimitar qué información debe o no conocer el *chatbot* con respecto a las tareas a realizar.
- Claridad en cuanto a la especificación del problema a resolver pero no tanto en el cómo se debe resolver.
- Poca confianza en cuanto al resultado a obtener mediante el sistema proyectado.
- Poca disposición en cuanto a tiempo para interactuar resolviendo los problemas anteriores.

Una vez hemos examinado los problemas existentes, es conveniente revisar, por un lado, las propuestas tecnológicas existentes para el desarrollo de *chatbots*, que serán necesarias para realizar la implementación de nuestro caso de prueba; y por otro, las metodologías propuestas para el diseño de *chatbots*, que necesitamos analizar para valorar y construir sobre ellas nuestra propuesta. Las alternativas tecnológicas se revisan en la sección 2.5. y las metodologías existentes, en la sección 2.6.

2.5. Análisis de alternativas tecnológicas al uso de Chatbots

Se presenta en esta sección un análisis de las alternativas tecnológicas para el desarrollo de *chatbots*. Se ha partido de las alternativas presentadas en el documento correspondiente al proyecto ColMOOC [95] de implementación de *chatbots* para fines de apoyo en la educación. La selección del estudio de ColMOOC para determinar qué alternativas se consideran en el análisis se decidió debido a que este estudio tiene una temática similar a la de este trabajo, por lo que tiene sentido usar la selección de alternativas tecnológicas consideradas en él. El método seguido para realizar este análisis ha consistido en seleccionar (de acuerdo a criterios relacionados con los problemas de carácter tecnológico especificados como relevantes para la problemática presentada en este TFM) aquellas alternativas que sean adecuadas para el trabajo a realizar. Se presentan a continuación los criterios utilizados y el resultado y conclusiones del análisis.

2.5.1. Criterios del análisis

Los criterios utilizados a la hora de analizar las tecnologías han sido los siguientes:

- Disponibilidad: Licencia bajo la que opera el sistema y precios de concesión de licencia para uso, desarrollo y comercialización de derivados del producto.
- Definición del modelo del agente en el sistema: Qué elementos constituye una definición del agente, si dispone de los elementos típicos en un *chatbot*, es decir, *intents*, *entities* y contextos del diálogo y cómo es su uso en el sistema.
- Cómo se realiza el procesamiento del lenguaje natural en el sistema: Qué se necesita para que el sistema lleve a cabo el procesamiento del lenguaje natural en el sistema y cómo se mejora la identificación de secciones clave dentro de una oración como el *intent* o las *entities*.
- Idiomas del sistema: En qué idiomas está disponible el sistema para llevar a cabo el procesamiento del lenguaje natural, si se requiere entrenar el sistema para que sea capaz de reconocer oraciones en español e inglés o si requiere software adicional.
- Interfaces del sistema: interfaces de interacción con el sistema. Tanto interacción a la hora de realizar la programación y creación del agente conversacional como a la hora de interactuar con el sistema a partir de otros sistemas clientes.
- Tratamiento de datos aportados por los usuarios: Cómo es el tratamiento que se hace de los datos de los usuarios al hacer estos uso del sistema, se necesita saber si se mantiene la confidencialidad de los datos aportados, si estos datos son usados para la mejora del sistema o si estos datos son cedidos para uso de terceros.

2.5.2. Sistemas considerados y análisis realizado

Sistemas considerados

El listado completo de alternativas examinadas para llevar a cabo el análisis es el siguiente: Rasa, ChatScript, Chatterbot, PandoraBots, WIT.ai, IBM Watson Conversation Service, DialogFlow, Azure Bot Service, Amazon Lex. Además, del listado del documento original, se han descartado las siguientes alternativas:

- ChatFuel (<https://chatfuel.com/>): Descartada debido a que requiere de forma obligatoria el uso de Messenger u otra red social donde se despliega el bot.
- Octane.ai (<https://www.octaneai.com/>): Descartada debido a que tiene una orientación de desarrollo demasiado centrada a la venta de productos y atención al cliente.
- Agentbot (<https://es.aivo.co/chatbot-agentbot>): Descartada por la misma razón que la anterior, demasiado especializada para el servicio al cliente en entornos comerciales.
- Reply.ai (<https://www.reply.ai/>): Descartada ya que no parece permitir el desarrollo por parte de developers, no permite la edición de código y su funcionalidad está muy limitada.

Análisis Tecnológico

Se procede a continuación a mostrar el análisis realizado:

Rasa (https://github.com/RasaHQ/rasa)
Presentación
Rasa es una alternativa tecnológica que proporciona un framework de aprendizaje basado en técnicas de Machine Learning para la construcción de sistemas conversacionales sobre voz y texto. Rasa utiliza el contenido de las conversaciones para entrenar el sistema de forma que se vuelva más inteligente y responda con mayor precisión en las conversaciones con los usuarios.
Disponibilidad
Rasa es un producto Open Source bajo la licencia Apache 2.0 ⁴ , permite el uso comercial del producto, su modificación, la distribución del producto o de sus modificaciones, así como el uso privado y patentado de productos derivados. Solo exige que se mantenga un aviso que informe a los receptores que en la distribución se ha usado código con la licencia Apache además de la propia licencia. No se ofrecen garantías, ni derechos sobre la marca Rasa ni los autores se hacen responsables de perjuicios ocasionados por su uso.
Definición del modelo del agente en el sistema
La definición del agente conversacional para Rasa consiste en la definición de lo que se llama dominios. Un dominio contiene los <i>intent</i> , <i>entities</i> , <i>actions</i> , <i>slots</i> y <i>forms</i> a los que el agente debe reaccionar y puede estar especificado entre uno o más archivos YAML normalmente utilizando la notación NLU. A mayores existen recursos como las <i>stories</i> y las <i>rules</i> que determinan la generalización a la hora de encontrar patrones en las conversaciones aportadas por los usuarios y permitir, a través del Machine Learning, reconocer la intención del usuario al hacer preguntas al bot, así como permiten definir escenarios en los que se deben ejecutar una serie de pasos en un orden específico.
Procesamiento del lenguaje natural
Rasa utiliza el NLU Training Data. Esto quiere decir que el procesamiento del lenguaje natural se basa en técnicas de Machine Learning sobre frases de ejemplo proporcionadas previamente. Los datos de entrenamiento de NLU consisten en expresiones aportadas comúnmente por los usuarios, que se utilizan como ejemplo de aprendizaje y están categorizadas en los diferentes <i>intent</i> disponibles. Estos ejemplos pueden contener el uso de <i>entities</i> previamente definidas, sinónimos de términos, expresiones regulares y tablas de referencia de términos asociados. Para el entrenamiento de un <i>intent</i> se deben dar una serie de sentencias de ejemplo en texto plano anotadas con la información a reconocer sobre ellas y que se asociará al <i>intent</i> y posibilitando el incluir metadatos a aportar al <i>intent</i> una vez el agente reconozca sentencias similares.
Idiomas en que está disponible
Permite el uso de múltiples idiomas (entre ellos el español) pero requiere de la aportación de diccionarios de palabras para esos idiomas. Desde la propia página de Rasa se sugiere el uso de spaCy ⁵ o MITIE ⁶ para el uso de lenguajes pre-entrenados evitando el comenzar desde cero su entrenamiento.
Interfaces de programación disponibles
La interfaz disponible para desarrolladores es la consola de comandos del sistema en el que se ha instalado. No existe otra forma de interactuar con el agente que no sea mediante edición de sus ficheros y la ejecución de scripts a través de la consola del mismo.
Interfaces de comunicación disponibles
A la hora de comunicarse con el agente conversacional está disponible una API accesible mediante peticiones HTTP que permite, a aplicaciones cliente, interactuar con el sistema.
Tratamiento de datos aportados por los usuarios
Dado que los datos pasan a ser parte del modelo de entrenamiento del agente. Ha de tenerse cuidado con el tratamiento de los mismos para evitar problemas de privacidad. Sin embargo, dado que el servidor de Rasa es una instalación Open Source bajo nuestro control no es necesario realizar ocultación de datos para terceros.

Tabla 2.1: Tabla análisis de Rasa

⁴<https://opensource.org/licenses/Apache-2.0>

⁵<https://spacy.io/>

⁶<https://github.com/mit-nlp/MITIE>

ChatScript (https://github.com/ChatScript/ChatScript)
Presentación
ChatScript es un framework para la creación de chatbots basados en reglas. Presenta un motor basado en reglas, donde las reglas son creadas por programadores, y se deben ejecutar a través de secuencias de comandos de programas en lo que constituye un flujo de diálogo. Utiliza un metalenguaje de scripting como código de programación.
Disponibilidad
ChatScript es un producto Open Source bajo la licencia MIT ⁷ , permite el uso comercial del producto, su modificación, la distribución del producto o de sus modificaciones así como el uso privado y patentado de productos derivados. Solo exige que se mantenga un aviso que informe a los receptores que en la distribución se ha usado código con la licencia MIT además de la propia licencia. No se ofrecen garantías, ni derechos sobre la marca ChatScript, ni los autores se hacen responsables de perjuicios ocasionados por su uso.
Definición del modelo del agente en el sistema
No hay un agente como tal en el sistema. Lo que permite crear este sistema es una colección de lo que llama <i>topics</i> que son una colección de reglas que determinan una serie de temas de conversación y sentencias asociadas a estos temas. Las reglas se componen de un tipo, una etiqueta, un patrón y una respuesta a la regla. El tipo de la regla determina si la regla hace referencia a una pregunta, una sentencia o lo que llaman un <i>gambit</i> que es la situación en la que el bot toma la iniciativa en la conversación con el usuario. El sistema busca una secuencia de palabras en la sentencia proporcionada por el usuario para detectar si, en el caso de que estén, se debe activar la regla asociada. La etiqueta tiene una función más pensada para el debugging y permite la manipulación de una regla para, en función de su etiqueta, ser manipulada por otras reglas. La respuesta no es solo el texto que se pretende responder al usuario ya que una vez se produce este texto se permite la activación de sentencias condicionales, bucles y llamadas a otras funciones que implican la ejecución de subsiguientes reglas.
Procesamiento del lenguaje natural
El procesamiento del lenguaje natural se basa principalmente en detectar cuándo hay coincidencias dentro de una oración introducida por el usuario y el patrón de una regla pre-programada. El sistema normalmente ejecuta las reglas en un orden específico, haciendo que cada regla no solo pase las restricciones de tipo y patrón, sino que también genere una salida destinada a llegar al usuario. Una vez que se tiene salida, el sistema concluye su ejecución (a menos que se indique explícitamente que se deben realizar más acciones). El sistema dispone de reconocimiento de usuarios y puede recordar el contexto de conversaciones mediante una combinación de memoria a corto plazo (solo disponible inmediatamente después de activar una regla) y de largo plazo (solo disponible dentro del mismo diálogo).
Idiomas en que está disponible
Está disponible en inglés nativo, pero es posible el introducir otros idiomas dentro de su funcionamiento. El inconveniente de esto es que básicamente hay que programar el “paquete” de lenguaje desde cero, incluyendo un diccionario de palabras y la gramática del lenguaje a aceptar.
Interfaces de programación disponibles
La interfaz disponible para desarrolladores es la consola de comandos del sistema en el que se ha instalado. No existe otra forma de interactuar con el agente que no sea mediante edición de sus ficheros y la ejecución de scripts a través de la consola del mismo.
Interfaces de comunicación disponibles
El chatbot se instala como un servidor y es capaz de ser integrado dentro de otros programas, reaccionando a las rutinas que estos activen en el, o también es posible levantar un websocket para que aplicaciones externas se comuniquen con el.
Tratamiento de datos aportados por los usuarios
Los datos aportados por los usuarios no son utilizados en el proceso de mejora y entrenamiento del chatbot, sin embargo si que se guardan datos de usuario durante la ejecución normal del sistema. Dado que es un sistema de instalación por servidor en nuestras máquinas privadas, no se envían datos de usuarios a terceros.

Tabla 2.2: Tabla análisis de ChatScript

⁷<https://opensource.org/licenses/MIT>

Chaterbot (https://github.com/gunthercox/ChatterBot)
Presentación
Chatterbot es una librería Python que permite la creación de software destinado a convertirse en un chatbot, a través del aprendizaje basado en técnicas de Machine Learning, capaz de hablar cualquier idioma. Una instancia no entrenada de ChatterBot comienza sin saber cómo comunicarse. Cada vez que un usuario se comunica con una sentencia, la biblioteca guarda el texto que ingresó y el contexto de la respuesta. A medida que ChatterBot recibe más entradas, aumenta la cantidad de respuestas que puede ofrecer y la precisión de cada respuesta en relación con la declaración de entrada.
Disponibilidad
Chatterbot es un software Open Source bajo la licencia BSD 3 ⁸ . Esta versión de la licencia BSD permite la redistribución ilimitada para cualquier propósito siempre que se mantengan sus avisos de derechos de autor y las renunciaciones de garantía de la licencia. La licencia también contiene una cláusula que restringe el uso de los nombres de los contribuyentes para la aprobación de un trabajo derivado sin un permiso específico.
Definición del modelo del agente en el sistema
El modelo de agente conversacional se construye continuamente con el entrenamiento del chatbot. El sistema de selección de información y de preprocesamiento de texto es lo que está disponible a los programadores. Estos dos elementos son los llamados preprocesadores y adaptadores lógicos. Los preprocesadores de ChatterBot son funciones simples que modifican la declaración de entrada que recibe un bot de chat antes de que la declaración sea procesada por el adaptador lógico. Por otro lado los adaptadores lógicos determinan la lógica de cómo ChatterBot selecciona una respuesta a una declaración de entrada determinada. Es posible seleccionar la cantidad de adaptadores lógicos que examinarán la declaración de entrada y en qué orden la examinará cada adaptador de los seleccionados. Si se utilizan varios adaptadores, el bot devolverá la respuesta de aquel con el valor de confianza calculado más alto. Si varios adaptadores devuelven la misma confianza, entonces el adaptador que se ingrese en la lista primero tendrá prioridad. Cada frase utilizada con el chatbot durante el entrenamiento es guardada en la base de datos asociada al chatbot junto con la respuesta esperada para ese tipo de sentencia. No hay por tanto un recurso que permita determinar un <i>intent</i> directamente ni <i>entities</i> . En cuanto al diálogo, el chatbot creado con ChatterBot es teóricamente capaz de aprender información en la sesión que recuerde basándose en su almacenamiento llevado a cabo en la base de datos. El adaptador lógico correspondiente debería dar la información aprendida en base a un criterio de mayor coincidencia encontrada con la información previamente introducida.
Procesamiento del lenguaje natural
La forma en la que funciona el procesamiento del lenguaje natural en este sistema consiste en que, tras pre procesar textualmente la cadena de entrada, enfrentarla a un conjunto de adaptadores establecidos por el programador (donde cada adaptador realiza distintos pasos de búsqueda en la base de datos de frases entrenadas para encontrar la que mejor coincidencia tenga con esa entrada), localiza la que mejor coincidencia tiene con la entrada, selecciona la respuesta asociada a esa entrada de mayor coincidencia y devuelve el resultado al usuario.
Idiomas en que está disponible
Inicialmente dispone de un corpus en inglés, pero están a disposición de los desarrolladores corpus pre entrenados en otros idiomas incluido el español.
Interfaces de programación disponibles
La única interfaz disponible para la configuración del sistema es en base a edición de ficheros y mediante línea de comandos.
Interfaces de comunicación disponibles
Es posible desarrollar un adaptador que realice peticiones get y post a una API externa.
Tratamiento de datos aportados por los usuarios
Aunque no envía datos a terceros, no queda claro si en base al aprendizaje por una base de datos común, lo aprendido durante una sesión con un usuario no se pueda usar en otra sesión de usuario con otro distinto pudiéndose potencialmente filtrar información confidencial.

Tabla 2.3: Tabla análisis de Chaterbot

⁸<https://opensource.org/licenses/BSD-3-Clause>

Pandorabots (https://home.pandorabots.com/home.html)
Presentación
Pandorabots es una plataforma de desarrollo de chatbots que utiliza un lenguaje de programación estándar abierto llamado Lenguaje de Marcado de Inteligencia Artificial (AIML) para la definición de las funcionalidades de los chatbots creados bajo ella.
Disponibilidad
Está disponible mediante licencias de uso. Existen distintas tarifas disponibles dependiendo de los servicios requeridos. El uso del sistema de creación de bots está disponible de forma gratuita, pero la comunicación con el bot a través de API requiere mínimo una tarifa de 9 \$ mes con posibilidad de 5000 mensajes mensuales y 3 \$ por cada 500 mensajes sobre el límite.
Definición del modelo del agente en el sistema
La definición del agente conversacional se hace a través de la definición de objetos AIML. Los objetos AIML pueden ser <i>topics</i> o <i>categories</i> . Una categoría está formada por una pregunta de entrada, una respuesta de salida y un contexto opcional. La pregunta o estímulo se denomina patrón, la respuesta se llama <i>template</i> . Los dos tipos principales de contexto opcional se denominan «that» y «topic». Las etiquetas AIML transforman la respuesta en un mini-programa que puede guardar datos, activar otros programas, dar respuestas condicionales y llamar de forma recursiva al comparador de patrones para insertar las respuestas de otras categorías. La parte de contexto opcional de la categoría consta de dos variantes, llamadas «that» y «topic». La etiqueta «that» aparece dentro de la categoría y su patrón debe coincidir con la última expresión del bot. La etiqueta «topic» aparece fuera de la categoría y recopila un grupo de categorías. El tema se puede establecer dentro de cualquier plantilla.
Procesamiento del lenguaje natural
El procesamiento del lenguaje natural se basa en la aplicación de técnicas de pattern matching y de ejecución de reglas y expresiones regulares para hacer que expresiones similares a las programadas inciten una respuesta en el bot.
Idiomas en que está disponible
Técnicamente soporta todos los lenguajes por tener que ser programado directamente en AIML aunque esto limita la potencia del chatbot por tener que tener en cuenta la gramática del lenguaje a utilizar.
Interfaces de programación disponibles
La interfaz de programación se basa en la línea de comandos y en la edición de los ficheros de categorías, pero además dispone de un editor de código online con algunas funcionalidades que facilita su programación.
Interfaces de comunicación disponibles
El bot puede desplegarse junto a un servidor web para con una API HTTP realizar preguntas al mismo. Está a disposición de los desarrolladores integraciones con Slack, Firebase y Heroku además de integraciones con diversas aplicaciones de mensajería como son Telegram, Twitter, WeChat y Line. Por último, posee SDKs de Java, Node.js y Python.
Tratamiento de datos aportados por los usuarios
No se realiza tratamiento de datos de los usuarios por ser de instalación en servidor propio y no enviar información a terceros.

Tabla 2.4: Tabla análisis de Pandorabots

Wit.ai (https://wit.ai/)
Presentación
Wit es una interfaz de lenguaje natural para aplicaciones capaces de convertir oraciones en datos estructurados. Dentro de las capacidades que esto permite está la creación de chatbots asociados a Facebook.
Disponibilidad
Está disponible de forma gratuita incluido su uso comercial, sin embargo, en los términos de uso y licencia, Facebook establece que es el propietario legal de la plataforma y hasta cierto punto de cualquier derivado, en concreto [96]: <i>“le otorgamos una licencia limitada, libre de regalías, no exclusiva, intransferible, no sub-licenciable y revocable para usar, acceder e integrar sus aplicaciones con la plataforma y el contenido de la plataforma”</i> sin embargo en las licencias por su uso que debemos conceder se menciona [97] : <i>“Usted nos otorga una licencia mundial no exclusiva, transferible, sub-licenciable, libre de regalías para alojar, usar, distribuir, modificar, ejecutar, copiar, almacenar, mostrar, traducir, analizar y crear trabajos derivados de su contenido, incluido cualquier derecho de propiedad intelectual contenido en el mismo, para cualquier propósito comercial en relación con la operación, prestación o mejora de los Servicios”</i> .
Definición del modelo del agente en el sistema
El modelo de agente en el sistema incluye la definición de propiamente un agente conversacional, bajo el que se pueden definir <i>intents</i> y <i>entities</i> . Además de los <i>intents</i> y <i>entities</i> el sistema permite definir lo que llama <i>utterances</i> que son frases de entrenamiento asociadas a un <i>intent</i> y a una serie de <i>entities</i> de ese <i>intent</i> que permiten mejorar el reconocimiento del <i>intent</i> y de las <i>entities</i> en frases nuevas por el sistema. Existen <i>intents</i> predefinidos en Wit.ai llamados <i>traits</i> que se corresponden con las intenciones más comunes en una frase como decir hola y adiós o contestar que no se ha entendido una frase. Asociadas a las <i>entities</i> se describen unos roles predefinidos por wit.ai que vienen pre-entrenados para reconocer el contenido más frecuente como valores de fecha y hora, localizaciones geográficas o direcciones, substrings dentro de la cadena de texto del mensaje, etc. aunque también es posible definir tus propios roles. Dentro de las <i>entites</i> es posible definir dos tipos y combinaciones de estos. Los tipos son: lo que llama <i>keywords</i> que son <i>entities</i> con valores predefinidos y que Wit.ai buscará que coincidan exactamente con los especificados o con uno de sus posibles sinónimos y las <i>free text</i> que son <i>entities</i> que no tienen valores predefinidos y que habrá que pre-entrenar antes de usar.
Procesamiento del lenguaje natural
El tipo de procesamiento que realiza Wit.ai del lenguaje natural se basa en técnicas de aprendizaje basadas en Machine Learning para determinar el <i>intent</i> asociado a la sentencia aportada por el usuario en base a las frases de entrenamiento recibido previamente para cada uno de los <i>intents</i> del agente conversacional. Una vez asocia la frase con un <i>intent</i> registrado, determina las posibles <i>entities</i> asociadas a la frase de entrada en base de nuevo a los valores de entrenamiento que ha recibido como posibles valores de las <i>entities</i> creadas para ese <i>intent</i> . La validación de nuevas sentencias asociadas a un <i>intent</i> debe realizarse manualmente en la fase de entrenamiento, seleccionando en la frase nueva el <i>intent</i> al que está asociado y las <i>entities</i> que contiene.
Idiomas en que está disponible
Está disponible para gran cantidad de idiomas, entre ellos el español y el inglés sin necesidad de instalar software por nuestra parte.
Interfaces de programación disponibles
La interfaz de configuración para el desarrollador del chatbot creado es una interfaz web.
Interfaces de comunicación disponibles
La comunicación con el chatbot se lleva a cabo con una API HTTP. El sistema dispone de SDKs para Python, Ruby, Go y Node.js. Es posible integrar un chatbot desarrollado con Wit.Ai alojándolo dentro de Facebook, WhatsApp y otras aplicaciones de mensajería.
Tratamiento de datos aportados por los usuarios
Implica el envío de datos personales a Facebook. Facebook como el procesador de los datos se compromete a seguir el GDPR.

Tabla 2.5: Tabla análisis de Wit.ai

IBM Watson Assistant (https://www.ibm.com/cloud/watson-assistant)
Presentación
IBM Watson Assistant es un servicio de IBM para la creación de chatbots desplegables sobre la nube de IBM. Permite a los desarrolladores de software de nivel empresarial incorporar un asistente virtual (VA) de inteligencia artificial (IA) en el software que están desarrollando y presentar el asistente como propio. Watson Assistant, que utiliza el aprendizaje automático (ML) de Watson AI y la comprensión del lenguaje natural (NLU), se comercializa para empresas que desean tener la opción de mantener como privados los datos que fluyen a través de su asistente virtual. A diferencia de otros proveedores que agregan la información que recopilan sus asistentes virtuales, IBM ofrece a los desarrolladores la opción de aislar la información que recopila su asistente en una nube privada para proteger los conocimientos adquiridos mediante la interacción del usuario.
Disponibilidad
Está disponible mediante licencias de uso. Existe un plan gratuito en el que se permite la conexión del asistente creado con hasta 1000 usuarios activos mensuales (MAU) únicos en conversación con el asistente, intercambiando hasta 10.000 mensajes al mes. Este paquete presenta limitaciones: Hasta 5 habilidades (diálogo, acción, búsqueda), 7 días de análisis de uso, 5 minutos de tiempo de espera de inactividad de sesión y el agente creado se suprime después de 30 días de inactividad (sin desarrollo). Las otras opciones no incluyen el alquiler o compra de una licencia sino el pago en función del uso que se le da al agente y las funcionalidades que se quiere incorporar al mismo. Además no permite el uso de las <i>Dialog Skills</i> .
Definición del modelo del agente en el sistema
La definición del agente conversacional se basa en la definición de las llamadas <i>skill</i> que componen el agente. Una <i>skill</i> o <i>conversational skill</i> se encarga de devolver respuestas creadas para responder preguntas comunes o realiza búsquedas en bases de datos externas para responder a la pregunta. Existen dos tipos de <i>skill</i> : <i>Conversational skills</i> que se dividen en dos subtipos: <i>Action Skills</i> donde la conversación se representa como un árbol de diálogo utilizando el editor de diálogo gráfico para crear una especie de script para determinar cuándo debe el agente interactuar con el usuario y <i>Dialog Skills</i> que contienen definiciones de <i>intents</i> , <i>entities</i> y <i>dialogs</i> a identificar en frases preentrenadas. Aparte de las <i>Conversational skills</i> existen las <i>Search skills</i> donde se aprovecha la información de las bases de conocimientos corporativos existentes u otras colecciones de contenido creadas por expertos en la materia para abordar consultas imprevistas o más matizadas de los clientes. Para una consulta de usuario determinada, se requiere el servicio IBM Watson Discovery para buscar una fuente de datos privada y devolver una respuesta.
Procesamiento del lenguaje natural
El procesamiento del lenguaje natural se realiza a través del Machine Learning de dos formas según se elija el tipo de skill de Action o Dialog. En las action es el sistema el responsable de identificar según los objetivos que tú has definido si la oración que el usuario introduce se corresponde con esos objetivos y se empieza a desencadenar la secuencia de acciones. Para el caso de dialog se tienen que definir los <i>intent</i> , <i>entities</i> y <i>dialogs</i> que regulen la conversación además de entrenarlos con frases de entrenamiento similares a las que puedan introducir los usuarios.
Idiomas en que está disponible
Está disponible en varios idiomas entre los que se incluye el español y el inglés.
Interfaces de programación disponibles
Interfaz web de desarrollo y configuración, el agente no se despliega fuera de la cloud de IBM.
Interfaces de comunicación disponibles
Para la comunicación con el agente está disponible una API HTTP para comunicación en base a peticiones GET/POST. También se dispone de integraciones Facebook, Slack, WhatsApp, etc.
Tratamiento de datos aportados por los usuarios
Los datos aportados por los usuarios en las conversaciones se alojan en la nube de IBM, existe un plan enterprise de pago que indica [98] que "Mantiene la privacidad de los datos de uso y entrenamiento en un entorno de arrendatario único aislado" pero el precio por ese plan excede los 5000 \$ así que no considero que sea adecuado su uso.

Tabla 2.6: Tabla análisis de IBM Watson Assistant

DialogFlow (https://cloud.google.com/dialogflow/docs/basics)
Presentación
DialogFlow es una plataforma de comprensión del lenguaje natural que se utiliza para diseñar e integrar una interfaz de usuario conversacional en aplicaciones móviles, aplicaciones web, dispositivos, bots, sistemas de respuesta de voz interactivos y usos relacionados. Pertenece a Google.
Disponibilidad
Está disponible mediante licencias de uso. Hay dos versiones disponibles DialogFlow ES y DialogFlow CX. DialogFlow ES provee de la funcionalidad básica, existen dos opciones de uso: la Trial Edition para hacer pruebas y aplicaciones no comerciales, siendo esta gratuita pero de uso limitado a, entre otras cosas, 2000 <i>intents</i> y no más de 50 MB totales de tamaño del agente. La otra opción de DialogFlow ES es la Essentials Edition cuyo precio es de 0.002 \$ por cada petición dirigida al chatbot. En el caso de CX 20 \$ por cada 100 sesiones de chat, entendiendo una sesión de chat como una conversación interrumpida por pausas menores a 30 minutos.
Definición del modelo del agente en el sistema
La definición del modelo de agente del sistema se compone de la creación de un agente conversacional al que se le deben definir una serie de <i>intents</i> , <i>entities</i> y contextos. El entrenamiento de <i>intents</i> , <i>entities</i> y contextos no está asociado a cada <i>intent</i> , <i>response</i> , <i>entity</i> o contexto sino al agente al que pertenecen, por tanto cada cambio en uno de estos elementos reentrena el agente con la nueva información. Los <i>intents</i> permiten definir dos tipos de respuestas: la primera es una respuesta prefijada en la configuración del <i>intent</i> para responderse siempre o al menos una de las variantes de esa respuesta. La segunda requiere el uso de un <i>webhook</i> que permite tanto activar contextos para llevar a cabo un diálogo con el usuario o bien peticiones HTTP de datos a servicios externos a DialogFlow.
Procesamiento del lenguaje natural
El procesamiento de lenguaje natural consiste en el entrenamiento de frases de ejemplo para los <i>intents</i> . Cuando un usuario final escribe o dice algo, Dialogflow compara la expresión con las frases de entrenamiento para cada intento para encontrar la mejor coincidencia. DialogFlow establece correspondencias mediante dos tipos de algoritmos: Coincidencia gramatical basada en reglas y ML Matching. La coincidencia gramatical se basa en el uso de coincidencias de patrones para encontrar frases similares a las utilizadas por el usuario, el ML Matching utiliza el entrenamiento supervisado de <i>intents</i> y <i>entities</i> con frases de ejemplo para que ante nuevas frases DialogFlow exprese un porcentaje de coincidencia para determinar las <i>entities</i> e <i>intents</i> posibles para esa frase.
Idiomas en que está disponible
Todas sus funcionalidades están disponible principalmente en inglés, pero también dispone de funcionalidades como conversaciones textuales (chats), TTS, STT, comunicación mediante llamadas y análisis de sentimientos en español.
Interfaces de programación disponibles
La interfaz de configuración del agente es una interfaz web.
Interfaces de comunicación disponibles
Dispone de integraciones con servicios de terceros como Slack, Messenger, Telegram, SmallTalk y Google Assistant. Puede dirigir peticiones de datos a servicios Rest web de terceros.
Tratamiento de datos aportados por los usuarios
Google toma posesión irrevocable de los datos utilizados para el entrenamiento del sistema sin embargo existe una opción para que no logee datos del sistema [99]: ” <i>El registro de interacción está habilitado de forma predeterminada para los agentes nuevos. Para habilitar o inhabilitar el registro de interacciones, sigue estos pasos: Ve a la consola de Dialogflow ES. Selecciona tu agente cerca de la parte superior del menú de la barra lateral izquierda. Haz clic en el botón de configuración ubicado junto al nombre del agente. Selecciona la pestaña General. Activa o desactiva la opción Registrar interacciones en Dialogflow (Log interactions to Dialogflow)</i> ” Esto lleva a pensar que es posible que se mantenga la privacidad de datos de los usuarios en el sistema. Para el resto de datos se sigue la Política de Privacidad de Google.

Tabla 2.7: Tabla análisis de DialogFlow

Azure Bot Services (https://azure.microsoft.com/es-es/services/bot-services/)
Presentación
Azure Bot Services es un producto propiedad de Microsoft en la forma de un servicio que permite al usuario implementar rápidamente con su servicio de Azure Bot Framework un servicio de chatbot que tomará las oraciones que se envían y las interpretará en términos de la intención que transmiten y las entidades clave que están presentes, tiene una interfaz web. que puede diseñar a medida un conjunto de intenciones y entidades que son relevantes para una aplicación y construcción de un sistema de comprensión del lenguaje.
Disponibilidad
Está disponible mediante licencias de uso. También es posible que en vez de alojar el servicio en la nube de Azure se haga localmente mediante la descarga de código del bot desarrollado. Dado que solo se aplican tarifas por lo que llama canales Premium que serían API REST propias de los desarrolladores y no un servicio como Slack o Facebook, se requeriría usar la versión Premium. Esta tiene una tarifa de 10.000 mensajes/mes gratuita a partir de la cual se cobrarían 0,422 Euros por 1000 mensajes.
Definición del modelo del agente en el sistema
Dentro de los servicios que ofrece el Azure Bot Service es LUIS el que aporta la parte de NLU. Dentro de LUIS, el agente se construye en base a apps que aglutinan la definición de <i>intents</i> , <i>entities</i> y patrones de expresiones regulares. La definición de intents se realiza indicando el objetivo del <i>intent</i> y una serie de frases de entrenamiento para ese <i>intent</i> . LUIS no recomienda exceder las treinta frases de entrenamiento ya que indica que esta fase del entrenamiento es mejor centrarse en el formato de la frase. En estas fases de entrenamiento también se identifican y marcan las <i>entities</i> asociadas al <i>intent</i> . Las <i>entities</i> de LUIS son de cinco tipos: Machine Learned que permiten extraer datos anidados y complejos aprendidos de ejemplos, listas que consisten en listas de definiciones y sus sinónimos comparando para encontrar coincidencia de texto exacta, pre-creadas para identificar un tipo de dato como una fecha o una url, y expresiones regulares.
Procesamiento del lenguaje natural
El procesamiento que se lleva a cabo en LUIS se realiza en base al entrenamiento de frases de prueba para asociarlas a <i>intents</i> y <i>entities</i> definidos previamente. Cómo se suministran las frases de prueba es lo que distingue a LUIS de otros agentes ya que LUIS permite no solo realizar pruebas individuales de una frase sino entrenar en base a un lote de frases si se aporta un archivo con estas. De cada entrenamiento LUIS devolverá un informe con los elementos identificados y la confianza en la identificación de estos.
Idiomas en que está disponible
Está disponible tanto en español como en inglés.
Interfaces de programación disponibles
La interfaz de configuración y programación consiste en el portal web de LUIS a partir del que se pueden realizar los entrenamientos y definición de agentes.
Interfaces de comunicación disponibles
Para interactuar con el chatbot están disponibles SDK de LUIS mediante C#, Python o JavaScript. También puede usar cURL para enviar solicitudes mediante la API REST
Tratamiento de datos aportados por los usuarios
Los datos de los usuarios son almacenados en Microsoft Azure como parte del modelo de LUIS salvo que se indique lo contrario. Microsoft especifica que [100] : “ <i>El contenido del cliente se almacena cifrado en el almacenamiento regional de Microsoft Azure e incluye:</i> • <i>El contenido de la cuenta de usuario recopilado en el registro.</i> • <i>Datos de entrenamiento necesarios para compilar los modelos.</i> • <i>Consultas de usuario registradas usadas por el aprendizaje activo para ayudar a mejorar el modelo Los usuarios pueden desactivar el registro de consultas mediante la anexión de <code>log=false</code> a la solicitud.</i> ” Además es posible mediante petición el borrado de datos de usuarios.

Tabla 2.8: Tabla análisis de Azure Bot Services

Amazon Lex (https://aws.amazon.com/es/lex/)
Presentación
Amazon Lex es un plataforma para la creación de agentes conversacionales a partir de interacciones basadas en voz y texto. Esto permite la creación de chatbots avanzados a través de las bibliotecas de <i>NL Processing</i> usadas para sistemas como Alexa. En la propia página de Amazon se menciona [101]: <i>“Amazon Lex proporciona la gran funcionalidad y flexibilidad de la comprensión del lenguaje natural (NLU) y el reconocimiento automático de voz (ASR), lo que permite crear experiencias de usuario sumamente atractivas con interacciones de conversación realistas y crear nuevas categorías de productos.”</i>
Disponibilidad
Está disponible mediante licencias de uso. Amazon explica en su página que tiene tarifas de uso por cada entrada del usuario al chatbot que se procesa como una llamada API independiente. El precio es de 0,004 \$ por cada solicitud de habla (voz) y 0,00075 \$ por solicitud de texto.
Definición del modelo del agente en el sistema
La definición del modelo del agente en Amazon Lex la componen el conjunto de <i>intents</i> , <i>slots</i> , <i>context</i> y <i>utterances</i> definidos para el bot así como el alias que referencia la versión e identidad del bot. Como en otras ocasiones, los <i>intent</i> definen el propósito de la interacción del usuario con el bot cuando produce entradas al mismo. Los <i>slot</i> actúan como las <i>entities</i> tradicionales además de que se pregunta automáticamente por ellas si al proveer la entrada el usuario, no se ha encontrado coincidencia para alguna de estas. Para hacer esto, por cada <i>slot</i> se ha de asociar una frase de pregunta que el bot se encargará de mostrar al usuario si no encuentra coincidencias en la frase de entrada para el <i>intent</i> . Por ejemplo, en el caso de un <i>intent</i> para solicitar un billete de tren puede haber <i>slots</i> para los lugares de origen y destino. Si el usuario no especifica en la frase inicial en la que solicita el billete alguno de estos datos, el bot se encargará de responder con las frases preparadas para preguntar por el dato que no ha encontrado en la frase aportada (origen y/o destino en este caso). Los <i>context</i> son la herramienta utilizada para llevar a cabo un diálogo efectivo con el usuario. Permiten una memoria dentro de la conversación para referirse a datos aportados previamente, permitiendo interacciones más complejas de esta forma. Por último, las <i>utterances</i> son frases de entrenamiento para los <i>intent</i> a partir de las que generalizar y asociar la intención del usuario con ese <i>intent</i> .
Procesamiento del lenguaje natural
De nuevo el procesamiento del lenguaje natural se realiza a partir de al generalización de frases de entrenamiento aportadas en los <i>intent</i> para permitir asociar una determinada entrada del usuario a un <i>intent</i> preentrenado. Sobre ese <i>intent</i> asociado, se buscarán coincidencias con los <i>slots</i> definidos en caso de encontrarlos. Si no se encuentran se preguntará al usuario por el <i>slot</i> específico no encontrado mediante una frase preparada para ese <i>slot</i> .
Idiomas en que está disponible
Está disponible tanto en español como en inglés.
Interfaces de programación disponibles
La interfaz de configuración y programación consiste en el portal web de Amazon Lex a partir del que se pueden realizar los entrenamientos y definición de bots.
Interfaces de comunicación disponibles
Para interactuar con el chatbot están disponibles SDK de Lex mediante C++, Go, Java, .NET, Node.js, PHP, Ruby, Python o JavaScript. También puede usar la API HTTP para enviar solicitudes GET y POST. También es posible utilizar integraciones con servicios de mensajería como Facebook Messenger, Slack y Twilio SMS.
Tratamiento de datos aportados por los usuarios
Amazon Lex está bajo el AWS shared responsibility model y del GDPR. Sin embargo en la sección de preguntas sobre privacidad se dice [102]: <i>“Amazon Lex puede almacenar y utilizar las entradas de voz y texto procesadas por el servicio únicamente para proporcionar y mantener el servicio, así como para mejorar y desarrollar la calidad de Amazon Lex y otras tecnologías de aprendizaje automático/inteligencia artificial de Amazon. El uso de su contenido es necesario para la mejora continua de la experiencia como cliente de Amazon Lex, incluido el desarrollo y el entrenamiento de tecnologías relacionadas. No utilizamos información de identificación personal que pudiera estar incluida en el contenido para venderle a usted o a sus usuarios finales productos o servicios, ni para campañas de marketing.”</i> También se menciona la posibilidad de eliminar el contenido almacenado por Amazon Lex[102] : <i>“Puede desactivar el uso de su contenido para mejorar y desarrollar el nivel de calidad de Amazon Lex y de otras tecnologías de aprendizaje automático e inteligencia artificial de Amazon. Para ello, utilice una política de no participación de AWS Organizations”</i> .

Tabla 2.9: Tabla análisis de Amazon Lex

2.5.3. Conclusión del análisis tecnológico

Del análisis presentado se puede concluir que solo las opciones que no se basan en el desarrollo de agentes cuyo funcionamiento depende de reglas son una opción viable para la construcción de un *bot* con interacción avanzada con el usuario. Esto descartó las opciones de Rasa, ChatScript, Chatterbot, PandoraBots y WIT. En una segunda inspección, prestando atención a los criterios de protección por GDPR, tarifas y facilidad de uso y programación, se descartó IBM Watson debido al elevado costo de mantener los datos aportados por los usuarios en su interacción con el *chatbot*. Amazon Lex fue descartado por no poder asegurar que los datos que manejase el *chatbot* en territorio europeo fuesen guardados en servidores en ese territorio y Azure Bot Service por la dificultad de programación y modificación del *chatbot* en comparación con las otras alternativas. Finalmente se seleccionó DialogFlow porque satisfacía los criterios exigidos de privacidad de datos, al mismo tiempo que mantenía un precio bajo en sus versiones comerciales y era familiar tanto a los tutores como al alumno implicados por lo que no se requería superar una primera curva de aprendizaje.

2.6. Metodologías de desarrollo de chatbots

En esta sección se presentan las metodologías de desarrollo de *chatbots* disponibles en la bibliografía. El enfoque mayoritario de estas metodologías ([34], [41], [50]) orientan la solución propuesta en torno a la fase de desarrollo del sistema que resuelva el problema. Solo una de las propuesta examinadas orienta la solución en cuanto al análisis y captura de la información para un posible tratamiento de los problemas conceptuales identificados, el Conversation Driven Development [51].

2.6.1. Metodologías orientadas hacia el desarrollo del sistema

Las metodologías orientadas hacia la solución del problema mediante un desarrollo tecnológico, identifican correctamente los problemas conceptuales anteriormente expuestos pero los abordan a través del desarrollo de una solución tecnológica. Este tipo de metodologías se categorizan en tres tipos principalmente: Las orientadas hacia un enfoque basado en reglas, las orientadas hacia un enfoque estadístico y las orientadas hacia un enfoque basado en técnicas de aprendizaje por inteligencia artificial. Las primeras [34] basan su funcionamiento en seguir reglas de diálogo rígidas de forma que se decide secuencialmente en función de condiciones lógicas como se estructura el diálogo. Las segundas [41], tienden a utilizar un modelo de máquina de estados en el que el diálogo es controlado en función de la probabilidad de transición a otro estado calculada sobre el contexto del diálogo ya existente y sobre el corpus de expresiones disponibles. Algunas de las técnicas usadas para resolver los problemas descritos con esta orientación, se basan en el uso de procesos de decisión de Markov entre distintos estados [49]. Las terceras [50] surgen como una evolución de las segundas, donde siendo el diálogo representado mediante una máquina de estados, las transiciones entre estados se controlan a través de distintas técnicas de clasificación sobre redes convolucionales y regresión sobre redes LSTM (Long Short Term Memory). Todas estas opciones, como se ha comentado, identifican correctamente los problemas, pero los tratan en una etapa de desarrollo. La propuesta que los trata en una etapa de análisis es el Conversation Driven Development.

2.6.2. Conversation Driven Development

El Conversation Driven Development (CDD), es una metodología planteada por Rasa para el desarrollo de *chatbots* que permitan llevar a cabo conversaciones más complejas en las que el usuario no tenga por qué saber como usar el sistema, solo concentrándose en su objetivo a conseguir mediante su interacción con él. La metodología del CDD se divide en cinco actividades a llevar a cabo iterativamente sobre un prototipo inicial del asistente que se desea desarrollar hasta su máxima funcionalidad. Cada una de las actividades se divide en subtarefas que hay que llevar a cabo de forma secuencial para poder obtener el resultado requerido.

Actividades del CDD

El listado de las actividades a desarrollar es el siguiente:

- Auto-evaluación
- Presentación del Prototipo
- Revisión de las conversaciones
- Correcciones y Mejoras
- Revisión de Métricas

Auto evaluación: La actividad de *Auto-evaluación* es la actividad inicial a desarrollar, consiste en identificar prácticas y hábitos de desarrollo dentro del equipo que pueden constituir oportunidades de mejora a la vez que se trabaja en ellas. La actividad de *Auto-evaluación* está a su vez formada por sub-tareas: La tarea de *Discusión de objetivos* consiste en llevar a cabo una discusión centrada en la especificación de los objetivos de mejora o construcción para el *chatbot*. En la tarea de *Auto-evaluación como equipo*, se puntúa una serie de tareas a realizar entre las que se incluyen las revisiones de conversaciones llevadas a cabo por usuarios, los datos de entrenamiento, valorando hasta que punto las conversaciones de los usuarios con el *chatbot* pueden ser clasificadas y utilizadas para ayudar en el proceso de aprendizaje del propio *chatbot*. Las pruebas con los usuarios, en donde se valora la implicación de los clientes en el desarrollo del *chatbot* a través de prototipos del mismo, permitiéndoles interactuar con el y recabar información de estas interacciones. El flujo de despliegue del sistema, donde se valora si se automatiza el despliegue del *chatbot* con técnicas de CI/CD (Continuous Integration / Continuous Delivery) según se hacen cambios en el mismo o si se evalúa manualmente el mismo. En la tarea sobre métricas, se valora cómo de avanzadas son las métricas utilizadas para la evaluación del éxito que tiene el *chatbot* en su interacción con los usuarios. En la tarea de *Compartir auto-evaluaciones* se comparte la puntuación evaluada por cada participante en el paso anterior para cada categoría y se discute las discrepancias en puntuaciones hasta que se llega a un consenso sobre la puntuación. Por último en la tarea de *Establecer pasos de mejora* se identifican las áreas donde el equipo debe mejorar su puntuación y se identifica una o dos ideas mediante un proceso de *Brainstorming* que se puedan poner en práctica en el corto plazo.

Presentación del Prototipo: La actividad de *Presentación del Prototipo* trata de incluir a los usuarios finales del producto en el proceso de desarrollo mediante el monitorizado de su interacción con prototipos del sistema a desarrollar para, por un lado, validar el prototipo creado directamente por parte del cliente (obteniendo el *feedback* requerido para hacer cambios si se necesitase), y por otro lado, para obtener frases de entrenamiento reales que los usuarios usarían en una interacción auténtica con el sistema. De esta última forma se pretende enriquecer el entrenamiento del *chatbot* con frases que realmente se usarán habitualmente. Al igual que las otras actividades, ésta se compone de una sucesión secuencial de tareas a desarrollar:

- Preparación para el test de usuarios. Consistente en preparar un listado de tareas que los usuarios que están probando el sistema deben llevar a cabo, mientras un miembro del equipo resuelve las dudas contextuales de las tareas para los usuarios, se pueden tomar notas de estas dudas para registrar qué puntos del contexto o de la interacción con el sistema no quedan claros.
- Explicación del proceso. Se da una explicación a los usuarios de en qué consiste la prueba a realizar así como en aportar explicaciones sobre el *background* del proyecto.
- Llevar a cabo el proceso de prueba. Consiste en solicitar a los usuarios que completen las tareas que han descrito y animarles a describir su proceso de pensamiento en voz alta, especialmente si encuentran algo negativo o inesperado.
- Entrevistar a los usuarios. Consiste en someter a los usuarios que han probado el sistema a la serie de preguntas preparadas previamente para recabar su opinión.

- Discutir y documentar los hallazgos. Discutir los comentarios importantes y poner en común las notas y el resumen de la sesión de prueba. Esto se realiza con el objetivo de obtener una lista de mejoras para el sistema.

Revisión de las conversaciones: En la actividad de *Revisión de las conversaciones* los datos de las conversaciones del asistente con los usuarios que lo han probado son revisados para ver exactamente cómo fue cada interacción de usuario. Se debe filtrar las conversaciones para ver interacciones en las que se produjeron acciones alternativas, la duración de la conversación, etc. También se debe etiquetar conversaciones para realizar un seguimiento de estas e incorporarlas como parte de los datos de entrenamiento para el próximo prototipo del sistema a construir. Al igual que las otras actividades, esta se compone de una sucesión secuencial de tareas a desarrollar:

- Filtrado de conversaciones. Esto permite descartar conversaciones en las que el usuario se desconecta de la conversación con el asistente antes de acabar así como descubrir fallos en la interacción con el agente.
- Clasificación de conversaciones. Dependiendo del motivo del filtrado aplicado pueden clasificarse conversaciones como fallos, resolución en el análisis de sentimientos, certidumbre de respuesta, u otros motivos.
- Búsqueda de conversaciones útiles. En función de la clasificación aplicada reunir un dataset de conversaciones que se puedan usar como motivo de mejora o para su uso en el entrenamiento del *chatbot*.
- Anotación de las conversaciones de los usuarios. Buscando en los mensajes erróneos es posible detectar el error del agente y anotarlo para un entrenamiento correctamente.
- Documentar cambios a realizar. Documentar los cambios o mejoras a realizar en el sistema.

Correcciones y Mejoras: En la actividad de *Correcciones y Mejoras* dependiendo en la naturaleza del problema o de las mejoras anotadas en la actividad previa se lleva a cabo las correcciones oportunas para obtener una versión mejorada del sistema para la próxima iteración. Dentro de esta actividad también se lleva a cabo las actividades relacionadas con el CI/CD. Esto incluye el hacer que cada vez que se modifique el agente se deban superar ciertos test conversacionales antes de que pueda pasar a producción. Esta actividad es dependiente de la tecnología de seguimiento que se utilice y por tanto su implementación concreta es variable en base a ello.

Revisión de Métricas: En la actividad de *Revisión de Métricas* se revisa el resultado de las métricas obtenidas como resultado de las evaluaciones directas e indirectas sobre las conversaciones de los usuarios que han probado la aplicación. La aplicación de esta actividad consiste en una variación del método de tarjetas de *Canvan* y comienza con una sesión de *brainstorming* en la que se escriben en estas las preguntas a plantear al agente. A continuación se posicionan las tarjetas con las preguntas en zonas de una pizarra que simbolizan la puntuación asignada a las tarjetas colocadas en esa zona. Posteriormente, mediante otro proceso de *brainstorming*, se encuentran respuestas a las preguntas plantadas y se deciden métricas directas e indirectas para evaluar la adecuación de las respuestas del *chatbot* a la pregunta en función de cada una de las respuestas dadas. Por último se documentan los hallazgos y se pone en común con el equipo de desarrollo lo averiguado.

Análisis del Conversation Driven Development

La Metodología del Conversation Driven Development se inspira fuertemente en técnicas de gestión *Agile* [103] para estructurarse. En concreto en la primera fase de la misma, *Auto-evaluación* se lleva a cabo un mapeado de lo que el equipo cree que se debe mejorar tanto como dinámicas de equipo de desarrollo como en cuanto a las características del *chatbot* y de los problemas detectados en fases previas del desarrollo del mismo. Esto comparte similitudes con el *Scrum Sprint Planning* [104] en cuanto a funcionalidad del *Product backlog* seleccionada para

llevar a cabo bajo el *Sprint* a comenzar. También hay similitudes en cuanto a la temática tratada en el *Daily Scrum* relativo a las dificultades que ha encontrado el equipo de desarrollo, salvo que en el caso del CDD, esta fase no se lleva a cabo todos los días como el *Daily Scrum* sino que se lleva a cabo únicamente una vez al principio de la iteración de desarrollo. En el caso del *Sprint Planning* se debe planificar cómo se va a llevar a cabo el proceso de desarrollo de las funcionalidades o cambios introducidos en el *sprint* así como de su dimensión temporal, sin embargo, en el CDD no se hace explícitamente hincapié en estos puntos dejándoselos al desarrollador. Otro punto a tener en cuenta de la primera fase del Conversation Driven Development es que hace un planteamiento del trabajo inicial desde el punto de vista del equipo de desarrollo pero no tanto desde el descubrimiento de las necesidades o intenciones del cliente a las que se pretenden dar solución con el sistema. De nuevo, en esta metodología al estar centrada en el punto de desarrollo basado en *Agile* se omite (erróneamente, pues *Agile* no implica olvidarse de esto) todo lo referente a las secciones de *Análisis y Diseño del Sistema* de los procesos de desarrollo software tradicionales.

En la segunda fase del Conversation Driven Development se introduce a los clientes mediante prototipos en el proceso de desarrollo del sistema con la recolección del resultado de su interacción con el *chatbot*, esto es hasta cierto punto similar a lo que propone Extreme Programming (XP) [105] en su BDD (Behaviour Driven Development) [106] en donde se extiende el TDD (Test Driven Development), junto con ideas del diseño guiado por el dominio y el análisis y diseño orientado a objetos para proveer al desarrollo de software y a los equipos de administración de herramientas compartidas y un proceso compartido de colaboración en el desarrollo de software. Esto se realiza en el CDD involucrando a los usuarios finales del sistema en las pruebas del prototipo construido para conseguir un *feedback* directo sobre el sistema desarrollado al mismo tiempo que permite a los mismos concretar la idea que tenían del sistema a través del prototipo probado permitiendo mejoras y alteraciones a la especificación inicial.

En la cuarta fase del Conversation Driven Development se introducen conceptos similares o iguales al Continuous Integration (CI) y Continuous Deployment (CD) que son clave (el CI por lo menos) en planteamientos organizativos como XP donde se prioriza la integración de código continuo en un repositorio común, además de ofrecer código disponible para producción y que haya pasado por el proceso de verificación de calidad, ejecutando una serie de tests para verificar esta de forma lo más automática que sea posible.

Como conclusión al análisis, la propuesta del CDD tiene algunos puntos fuertes y serias debilidades. Los puntos fuertes: se inspira fuertemente en técnicas de organización para el desarrollo de software basadas en *Agile*, pone el foco en la participación del cliente en la evaluación y corrección del sistema a través de sucesivos prototipos de forma que se acabe obteniendo un sistema lo más adecuado que sea posible de acuerdo a la especificación del cliente. Como puntos débiles y amenazando a su validez, está el hecho de que aunque el CDD Playbook es una propuesta de metodología para el tipo de problemas descritos, no pertenece esta a fuentes literarias de conocimiento reconocido (Artículos de Investigación Primaria, Secundaria, Especiales, literatura terciaria o gris), sino que pertenece a publicaciones en distintos blogs, incluido el de Rasa. Por esta razón no se puede afirmar que haya sido verificada (en la actualidad) como válida de forma pública por investigadores no pertenecientes a Rasa. Otro punto débil es que se justifica en su introducción mediante lo que llama *los 5 niveles de AI conversacional*. Éste es un concepto introducido por el propio personal de Rasa ([107], [108]) para clasificar la completitud que pueden tener los *chatbots* en su nivel de interacción con un usuario y poder justificar así la calidad de los *chatbots* ofrecidos como resultado de esta metodología. Otra debilidad es que, pese a decir en la introducción del CDD Playbook: “*CDD isn't proprietary, and it isn't tied to a particular framework. It's a set of activities and design principles that help conversational AI teams build AI assistants that really help users*”, en el mismo PlayBook se promociona el uso de su sistema de desarrollo de *chatbots* de pago, Rasa X, para la aplicación de la metodología. El último problema del CDD es que al centrarse en técnicas *Agile*, se centra bastante en el ambiente de trabajo de los desarrolladores del producto pero no tanto en cuanto a la participación del usuario en el mismo.

2.7. Conclusión del capítulo

En este capítulo se han revisado los principales retos tecnológicos y conceptuales, a los que se enfrenta un desarrollador de *chatbots*. Estos últimos han sido atendidos hasta el momento por metodologías que enfocan el

problema desde un punto de vista centrado en la fase de desarrollo mayoritariamente. La única alternativa que los enfoca desde un punto de vista centrado en la fase de análisis no resulta del todo adecuada. Es por ello necesario proponer una metodología que desde un punto de vista de análisis de la información necesaria para comprender el contexto de interacción con el usuario muestre su valor y pueda ser verificada por la comunidad informática.

Capítulo 3

Análisis de técnicas de elicitación de requisitos en chatbots

En este capítulo se realiza una exploración de las técnicas de elicitación de requisitos que más relevancia tienen a la hora de proponer soluciones a los problemas expuestos en el capítulo anterior. Las técnicas que se tendrán que seleccionar necesitarán aproximarse al problema de elicitación de requisitos desde un ángulo de análisis de información que preste atención al contexto de comunicación.

3.1. Clasificación de técnicas de elicitación de requisitos

Basándonos en el análisis realizado por Pacheco [71], es posible clasificar las distintas técnicas de elicitación de requisitos en sistemas de información disponibles en las siguientes categorías:

Técnicas tradicionales Aquellas que fueron utilizadas en las primeras etapas del desarrollo de la ingeniería de software. Se centran en la elicitación de datos genéricos para la identificación de las necesidades que exponen los clientes y las limitaciones del sistema a desarrollar. Dentro de estas técnicas están incluidas las entrevistas, las encuestas, el análisis de tareas y los cuestionarios.

Técnicas colaborativas Son técnicas que promueven el llegar a acuerdos con el cliente mientras se utilizan la dinámica del equipo de desarrollo. Estas técnicas son adecuadas si hay varios tipos de clientes participando en el proyecto. Además, las técnicas de esta categoría se utilizan para seleccionar y priorizar los requisitos dentro del proceso de elicitación y proporcionar una guía para descubrir conceptos básicos y mejorar el conocimiento del dominio de la aplicación. Dentro de esta categoría están incluidos los focus group, los talleres de elicitación de requisitos y las sesiones de brainstorming.

Técnicas de prototipado Son técnicas que utilizan prototipos (desde prototipado en papel a productos en beta) porque existe incertidumbre con respecto al funcionamiento del sistema de software en la vida real y se requiere la obtención de información detallada sobre el sistema a construir y promover la retroalimentación entre las partes interesadas.

Técnicas de modelado Estas técnicas proporcionan un modelo del tipo de información que se elicitará, y se utilizan para liderar el proceso de elicitación y obtener una mejor comprensión de las necesidades de los clientes, el contexto y el proyecto. Dentro de las técnicas agrupadas aquí, aparecen: escenarios, aproximaciones orientadas al objetivo, modelos de proceso de negocio y casos de uso.

Técnicas cognitivas Son técnicas orientadas a la adquisición de conocimiento en sistemas basados en información. Ofrecen técnicas de elicitación de requisitos destinadas a representar y estructurar el conocimiento de los clientes en base a la definición de un problema y la solución visionada por estos. Dentro de esta categoría aparecen técnicas como el uso de ontologías y el card sorting.

Técnicas contextuales Son técnicas creadas a partir de la combinación del prototipado y las entrevistas centradas en el entorno de trabajo del cliente en que se desplegará el sistema. Esto se realiza para recopilar datos sobre las partes interesadas, sus procesos, modelos y flujos de trabajo, el entorno u otro elemento relevante con su entorno de trabajo con el fin de obtener una detallada comprensión de los requisitos. Dentro de estas técnicas se incluyen aquellas de carácter etnográfico y etnometodológico.

Técnicas Agile Este tipo de técnicas se centran en la forma de trabajo del equipo de desarrollo, organizándolo para poder adaptarlo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. Dentro de las técnicas aquí consideradas se incluyen: el mind mapping, las user stories y el group storytelling.

3.2. Selección de técnicas

En base a la clasificación anterior se llevan a cabo las siguientes observaciones para configurar qué grupo de técnicas es conveniente que formen parte de la metodología:

Las **técnicas tradicionales**, aunque permiten prestar atención a las necesidades del cliente, lo hacen desde un punto de vista genérico. Los cuestionarios son rígidos en cuanto a formato y requieren de un cierto conocimiento y entendimiento del problema a resolver para poder prepararlos antes de enfrentarlos con los clientes. El análisis de tareas implicaría en estos sistemas el poder ver cómo el usuario interactúa con la institución al realizar las preguntas sobre los temas en los que se le presentan dudas, por lo que no es adecuado para este tipo de problemas. Por último las entrevistas presentan el mismo inconveniente que los cuestionarios. Requieren conocimiento previo sobre el dominio que se va a tratar, aunque no sean tan rígidas como las anteriores, además de no ser un proceso reproducible por ser un tipo de técnica variable en cuanto a estructura, dependiendo de las circunstancias a tratar bajo ellas en cada caso. Por ello, se decidió no utilizar las técnicas tradicionales como técnicas principales de la metodología. Sin embargo, se reconoce que estas técnicas permiten obtener información del cliente de forma eficiente, por lo que se propuso la búsqueda de una técnica que, estando basada en una entrevista tuviese un formato fijo reproducible en otras ocasiones ya que las entrevistas por su característica de ser específicas al problema a tratarse en cada momento no permitirían llevarse a cabo de forma reproducible (hasta cierto punto) como parte de una metodología. Por ello se ha seleccionado la técnica basada en *laddering*, que consiste en, a través de una serie de preguntas cortas y empezando en un nivel de abstracción bajo, obtener cada vez información de más nivel de abstracción permitiendo ordenarla de acuerdo a una cierta estructura o jerarquía, en concreto Zowghi menciona que se supone, al utilizar el *laddering*, que la información que se está elicitando se puede ordenar jerárquicamente [81]. El uso de entrevistas basadas en *laddering* viene además justificado porque, algunos autores [90] señalan que son efectivas para capturar requisitos de una forma similar a las entrevistas estructuradas [71].

Las **técnicas colaborativas**, aunque podrían ser adecuadas a la hora de abordar el problema desde el punto de vista de los clientes y teniendo en cuenta el contexto, agrupan técnicas que requieren sesiones o periodos de involucramiento con los clientes demasiado extensivos en cuanto a tiempo. Dado que en el caso específico de VirtualForest los clientes solo pueden dedicarnos periodos cortos de tiempo no es posible llevar a cabo este tipo de técnicas ni por separado ni como parte de una metodología.

Las **técnicas cognitivas** no se han seleccionado debido a que, para aplicarse, requieren el haber construido de antemano la información, o conjunto de información que caracteriza el sistema. Es en ese punto donde se aplican estas técnicas para conseguir estructurar la información en una representación comprensible para los involucrados en el proyecto. Dado que este no es el caso para el tipo de proyecto tratado, donde es frecuente que el propio cliente no sepa con seguridad qué información es necesaria para la construcción del *chatbot*, no es posible la aplicación de este tipo de técnicas.

Las **técnicas contextuales**, aunque podrían parecer idóneas para el problema que estamos tratando, de abordar un desarrollo de este tipo de sistemas desde un punto de vista contextual, centrado en la interacción y en el cliente, no son aplicables a este caso. Esto es debido a que ambas técnicas (análisis etnográfico y etnometodológico) conllevan trabajar con poblaciones de usuarios o fuentes de datos muy superiores a las disponibles en este caso.

Para poder llevarlas a cabo en este tipo de desarrollos se requeriría el estudio de información derivada de cientos de interacciones de usuarios reales con un sistema similar al proyectado. Cosa que no es posible en las condiciones actuales de realización de este trabajo.

Debido a las características de los problemas expuestos anteriormente nos concentramos en las técnicas basadas en modelado, en el prototipado y en *Agile*. Nos centramos en modelado para permitir dar solución (a la necesidad específica del proyecto VirtualForest) de claridad en cuanto a definición del problema pero no en cuanto a cómo solucionarlo. Estas técnicas nos permitirían el comprender mejor qué es lo que se espera del sistema y el contexto de uso del mismo. El prototipado, por su parte, permitiría mostrar a los clientes el sistema en desarrollo para acrecentar su confianza en el resultado a obtener con el uso del mismo. Por último, las técnicas agile podrían ayudar a delimitar el problema y tareas a resolver, considerando el tiempo que se dispone para interactuar con el cliente. Nos centraremos también en el uso de técnicas de laddering porque es razonable asumir que en un proceso de diálogo y negociación con un cliente se necesitarán técnicas similares a entrevistas. Dentro de los tipos de técnicas elegidos, vamos a centrarnos en la clasificación de Pacheco [71] para identificar las más apropiadas para nuestros objetivos.

3.2.1. Técnicas de modelado

En la categoría de modelado aparecen las técnicas de escenarios, aproximación basada en objetivos y modelos de proceso de negocio.

La técnica de *escenarios*, permite usar descripciones de situaciones y procesos en los que el usuario interactúa con el sistema con un propósito específico. El uso de escenarios implica afrontar un proceso iterativo de refinamiento de éstos, buscando todas las excepciones y casos alternativos en los que el sistema se encuentre debido a dicha interacción [75]. Esto es al mismo tiempo la mayor fortaleza, por su completitud, y la mayor debilidad, por la cantidad de tiempo necesario para llevarla a cabo, a la que se enfrenta esta técnica en su ejecución. Los escenarios se relacionan con los modelos mediante un proceso de abstracción y con los prototipos mediante un proceso de diseño. Los escenarios que realizan una representación del mundo real son abstraídos para formar modelos. Los modelos y especificaciones de requisitos se transforman en diseños y finalmente se implementan. Los escenarios permiten servir como inspiración a la hora de llevar a cabo el diseño de prototipos. Además los escenarios se pueden utilizar para razonar sobre el diseño y como scripts de prueba en métodos de evaluación.

La técnica de *Aproximación Basada en Objetivos* [77], consiste en, a partir de objetivos de alto nivel del sistema, descomponerlos en sub-objetivos de menor tamaño al mismo tiempo que se elabora una descripción más completa de estos sub-objetivos creados. Esta técnica permite representar más fielmente el objetivo del usuario en su interacción con el sistema a partir de una descripción más vaga de un objetivo general, hasta obtener sentencias más concretas y completas que describan uno o más objetivos a conseguir más específicamente. Sin embargo, hay que tener en cuenta que un defecto de esta técnica [77] es que los errores introducidos en las fases tempranas de la misma se reproducen y difuminan a medida que se descomponen los objetivos generales con errores en objetivos específicos. Además los frameworks disponibles para la aplicación de la técnica (KAOS, i*, etc.) requieren una inversión considerable de tiempo para poder aplicarlos.

La técnica de *Modelos de proceso de negocio* [80], consiste en realizar un diagrama que represente las actividades del negocio permitiendo el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow). Business Process Modeling (BPM) es la actividad de representar los procesos comerciales de una empresa, para que puedan ser analizados. Para garantizar la eficacia de BPM, es importante que se seleccione una notación BPM que sea completa y clara, es decir, capaz de expresar todos los conceptos relevantes del dominio en estudio. Esta técnica es útil para ayudar a las partes interesadas a analizar sus procesos y tareas siendo eficaz en mejorar la comunicación bidireccional entre el analista y los interesados y proporciona una base para una mayor comprensión, interpretación y validación de los requisitos obtenidos.

3.2.2. Técnicas de prototipado

La técnica de *Prototipado* [81] aunque costosa en cuanto a tiempo de llevar a cabo, permite obtener de primera mano de los interesados *feedback* relevante, además de ser especialmente útil en el caso de que los interesados no tengan del todo clara la finalidad que el sistema pretenda cubrir o cuando se disponen de una serie de requisitos de carácter general. La técnica referida en esta sección al prototipado es el prototipado iterativo. El prototipado iterativo [84] consiste en, a través de sucesivas iteraciones sobre un prototipo funcional, perfeccionar su funcionamiento hasta que en una iteración el cliente valide el alcance y funcionamiento del sistema dando como resultado un producto completo. Esta técnica es apropiada en la circunstancia de que el cliente no esté del todo seguro del funcionamiento final que quiere tenga el producto una vez sea lanzado a la fase de producción. Esta técnica le permite a la vez, el comprobar el estado del desarrollo y le da ideas que le permiten concretar qué es exactamente lo que necesita.

3.2.3. Técnicas Agile

La técnica de *Mind Mapping* [85] se ha definido como “*representaciones visuales, no lineales de ideas y sus relaciones*” [87]. Los mapas mentales comprenden una red de conceptos conectados y relacionados donde cualquier idea puede conectarse con cualquier otra. El objetivo de esta técnica es encontrar asociaciones creativas entre las ideas expuestas para lo que se requiere un pensamiento espontáneo e imaginativo al crear el mapa mental. De acuerdo con Pacheco, esta técnica es apropiada para trabajar siguiendo la forma en que la gente tiende a pensar, es decir, no linealmente y es eficaz en la organización y representación de información dentro de una jerarquía radial.

Las *User Stories* [81] son descripciones generales desde el punto de vista del usuario final de una funcionalidad que debe cubrir el sistema. El propósito de éstas, es explicar cómo proporciona valor una funcionalidad software al usuario o cliente final. Una historia de usuario describe la funcionalidad que será valiosa para un usuario o para el comprador de un sistema o software. De acuerdo con Mike Cohn [109], las *user stories* se componen de tres aspectos:

- Una descripción escrita de la historia utilizada para la planificación y como recordatorio.
- Conversaciones sobre la historia que sirven para desarrollar los detalles de la historia.
- Pruebas que transmiten y documentan detalles y que se pueden utilizar para determinar cuando una historia esta completa.

A estas tres partes se les suele llamar tarjeta, conversación y confirmación. Esta estructura está pensada porque si bien la tarjeta puede contener el texto de la historia, los detalles se resuelven en la Conversación y se verifican con la Confirmación. El enunciado de la *user story* suele ser: Como (Rol) Quiero (...) Para (...). Esta estructura permite determinar la función y la intención del usuario para proporcionarle la información correcta, que corresponde con el propósito definido en la story.

En la técnica del *Group Storytelling* [88] los participantes informan de sus actividades mediante relatos o narrativas colectivas. Las narrativas se consideran productos sociales dentro de contextos específicos, y un metodo interpretativo a través del cual las personas comunican conocimientos y definen su propia identidad. En esta técnica se llevan a cabo discusiones grupales en las que a través de la narración de historias individuales se construye una narración colectiva para unificar el grupo y unir los intereses de todas las partes. Esta técnica tiene como objetivo modelar el dominio desde diferentes perspectivas con el fin de desarrollar una descripción completa y coherente del sistema a desarrollar.

3.3. Selección de técnicas de análisis de requisitos para chatbots

En esta sección se va a hacer una selección de las técnicas más apropiadas para afrontar los problemas para el desarrollo de *chatbots* identificados en la sección 2.4, es decir, la variabilidad del lenguaje y el control del diálogo.

3.3.1. Técnicas a aplicar en la variabilidad del lenguaje

El problema de la variabilidad del lenguaje consiste en qué hacer con un sistema que acepta entradas equivalentes conceptualmente pero de distinta representación en el lenguaje. Dado que el problema a evitar es el no conseguir abarcar el contexto en que se comunica el usuario en una interacción determinada con el sistema, se propone el centrarse en técnicas que nos permitan abstraernos a lo que realmente desea el usuario o que nos permitan reducir la ambigüedad de los requisitos resultantes de las mismas. Debido a esto se seleccionaron las siguientes técnicas:

- User Stories
- Uso de entrevistas basadas en Laddering
- Uso de Prototipos

El uso de *user stories* viene justificado porque las *user stories* como herramienta son fácilmente adaptables a cambios en los requisitos derivados de las mismas ya que recogen el proceso de conversación que determina la propia *user story*. Las *user stories* permiten además adaptarse al desarrollo de *chatbots* ya que tanto las *user stories* como los *chatbots* necesitan prácticamente la misma información que se encuentra en los componentes de la story.

El uso de *entrevistas basadas en laddering* consiste en, a través de una serie de preguntas cortas y empezando en un nivel de abstracción bajo, obtener cada vez información de más nivel de abstracción encadenándola entre sí mediante una relación consecuencia-causa. Esto permite llevar a cabo un proceso de entrevista, pero ordenada de acuerdo a construcción de un modelo o jerarquía de información. Mediante esta técnica por tanto se puede llevar a cabo un proceso de entrevista reproducible en circunstancias distintas ya que se posee una estructura común a seguir entre ellas.

Por último el uso de *prototipos* permite, a través de las iteraciones, refinar el producto que se está elaborando al mismo tiempo que le permite al cliente verificar cómo se comportaría un sistema real ante la funcionalidad solicitada. Permitiría no solo recoger el feedback del cliente al ponerse a éste en contacto con el prototipo, concretando su idea sobre la funcionalidad requerida, sino también nuevas formas de expresión de acciones para el *chatbot* distintas de las elaboradas por el desarrollador y más cercanas al contexto real de uso del sistema.

3.3.2. Técnicas a aplicar en el control del diálogo

El problema, expuesto en la sección anterior, del control del diálogo consiste en cómo especificar el contexto y la estructura de la conversación que tendrá un *chatbot* con el usuario. Dentro de las técnicas examinadas en la sección anterior para poder tratar este problema las más relevantes para este problema son:

- Uso de escenarios.
- Uso de modelos de proceso de negocio.

El uso de escenarios viene justificado por permitir, sin restringirse a la notación de UML, estructurar un diálogo con el usuario a través de la aparición de condiciones en el flujo de la conversación. Permitiría mostrar también al cliente los caminos alternos al escenario de éxito planteado. El uso de escenarios permite analizar la interacción entre el software y el usuario, y reducir discrepancias y ambigüedades en el flujo de datos de la tarea mediante el estudio de los posibles canales de comunicación en la conversación.

El uso de modelos de proceso de negocio permite a través de los *flowcharts* estudiar visualmente como se estructura la conversación con el usuario cuando éste interacciona con el *chatbot* para cumplir un objetivo concreto. Permite esta técnica observar cómo fluye la información entre los distintos nodos de la conversación. Aunque esta técnica se basa en el uso de diagramas de flujo no utilizaremos directamente los diagramas de actividades de UML, ya que lo que queremos es modelar la conversación y con los diagramas de actividades nos veríamos forzados a limitarnos a las acciones que se cumplen en la conversación, pero no mostraríamos el intercambio de preguntas y respuestas con el *chatbot*.

3.4. Criterios de validación de las técnicas seleccionadas

En esta sección se explica el criterio de aplicación de cada una de las técnicas anteriormente seleccionadas. El objetivo es permitir verificar de una forma contrastable que se han aplicado las técnicas de la manera adecuada, validando el resultado obtenido por las técnicas como parte de la metodología. Para ello se siguen criterios conocidos para asegurar la calidad de la aplicación de las técnicas escogidas.

3.4.1. User stories

Para evaluar la aplicación de las user stories seguiremos el criterio INVEST de calidad tal y como hace Garm Lucassen en su estudio *The Use and Effectiveness of User Stories in Practice* [110]. En concreto el método de evaluación será para determinar si se está aplicando INVEST correctamente. El objetivo, como se demostró en el estudio de Garm Lucassen, es conseguir unos artefactos válidos y de calidad a través de este criterio. Por ello, cada *user story* obtenida en el proceso de selección de objetivos debe cumplir los siguientes criterios:

- Independencia (I): Las *user story* deben ser lo más independientes posible en el sentido de que deben poderse llevar a cabo en cualquier orden. Esto permitiría una priorización de unas *user story* sobre otras sin tener que tener en cuenta posibles relaciones entre ellas que desajusten esa priorización, obligando a reconsiderar esta debido a las dependencias.
- Negociables (N): La *user story* captura la esencia de lo que el cliente necesita. El resultado real debe ser el resultado de una negociación colaborativa entre el cliente y el desarrollador.
- Valor (V): La *user story* debe representar valor para el usuario de la *user story*. Debe estar escrita desde el punto de vista del valor que el usuario espera obtener en ella.
- Estimable (E): Una *user story* debe poder estimarse o dimensionarse para poder priorizarla adecuadamente.
- Pequeñas (S): Las *user story* no deben estar pensadas para representar grandes cargas de trabajo para el desarrollador. Deben representar pequeños conjuntos de tareas que provean una funcionalidad.
- Testeable (T): Cada historia debe tener algo que permita comprobar que se ha completado. Un criterio de aceptación demostrable.

Dado que las user stories se están aplicando para descubrir la funcionalidad que llevan detrás el cumplimiento de un objetivo del sistema, la granularidad de estas debe ser alta sin entrar tanto en detalles de implementación y centrándose en la forma o método de llevar a cabo las acciones del objetivo.

3.4.2. Escenarios

Para la aplicación de la técnica de recogida de información basada en escenarios nos vamos a inspirar en el meta-modelo de escenario propuesto por do Prado Leite [111]. El plantear un modelo de escenario tiene la finalidad de actuar como un modelo de sintaxis y pauta estructural con el objetivo de conseguir:

- Homogeneizar la estructura del conjunto de escenarios que se usarán como parte de la metodología.
- Plantear los aspectos que pueden cubrir los escenarios y hasta qué punto pueden ser completos modelando las metas de un diálogo entre el usuario y el *chatbot*.
- Obtener un método para llevar a cabo la verificación del escenario en función del correcto seguimiento o no de la estructura.

El meta-modelo de escenario propuesto es una estructura compuesta por las siguientes entidades: título, objetivo, contexto, recursos, actores, episodios y excepciones, y el atributo restricción.

- Título: El título descriptivo de la *user story* a la que está dando una explicación el escenario.

- **Objetivo:** La meta que el usuario persigue al interactuar con el *chatbot*.
- **Contexto:** Explicación de una restricción temporal o situacional que describe en qué circunstancias de interacción está el usuario iniciando la interacción con el *chatbot* bajo este escenario.
- **Recursos:** Recursos de información que el *chatbot* necesita disponer para que se pueda llevar a cabo el escenario.
- **Actor:** Bajo que rol está el usuario interactuando con el *chatbot*, qué motivación principal tiene ese rol asociado.
- **Episodios:** Conjunto de pasos que conducen a la consecución de la meta del escenario. Cada uno de ellos requiere una información de entrada perteneciente al contexto de la conversación y produce una información de salida necesaria a almacenar en el contexto de la conversación.
- **Excepciones:** Son comportamientos inesperados en la conversación que desvían al usuario de la meta que perseguía originalmente y abren nuevos diálogos ajenos a esta.

Al estructurar los escenarios de esta forma nos aseguramos los anteriores objetivos así como verificamos la posible conversación del usuario al perseguir la meta.

3.4.3. Uso de Prototipos

Como se ha explicado anteriormente se usará un modelo adaptado a nuestras necesidades a partir de lo expuesto en el modelo de Dante Carrizo [84]. En nuestro caso la sesión de validación del prototipo con los *stakeholders* se lleva a cabo en dos partes. La primera parte será una explicación por parte del desarrollador de la funcionalidad sobre la que se ha desarrollado el prototipo, esto permitirá a los *stakeholders* el familiarizarse con las capacidades del sistema en esta fase. La sesión estará siendo grabada, por lo que cualquier pregunta o sugerencia hecha por los *stakeholders* quedará registrada para su posterior análisis. Tras completar la explicación se pasará a la parte de interacción del sistema con los *stakeholders* en donde estos usarán el sistema, comentando en voz alta los fallos encontrados, diferencias de funcionamiento de lo proyectado originalmente o falta de completitud en lo proyectado. La aplicación del prototipado dentro de la metodología recibiría los artefactos resultantes (*user stories*, escenarios y diagramas de flujo conversacional) de la propia metodología aplicada antes de llegar al prototipado para con la información recogida hasta entonces poder elaborar el prototipo. El prototipo se validaría con los clientes en una sesión, que será grabada, de unos 30 minutos de duración con la estructura de dos partes descrita previamente.

3.4.4. Laddering

Para asegurar que, en caso de no coincidir la funcionalidad mostrada en el prototipo con lo que el cliente se esperaba, se obtiene una descripción del valor que realmente se perseguía, se aplica la técnica del laddering. Para la aplicación de la entrevista basada en laddering se usará una versión del Means Chain Theory (MCT) propuesto por Veludo de Oliviera [112]. Según la teoría de la MCT, existe una jerarquía de percepciones del consumidor y conocimiento del producto que va desde los atributos (A) hasta las consecuencias del consumo (C) y los valores personales (V), de la siguiente manera:

- **A:** Atributos o características del producto sobre el que se está investigando.
- **C:** Los atributos tienen consecuencias de consumo del producto para el usuario.
- **V:** Las consecuencias están conectadas a valores que se persiguen por el usuario.

Por cada área de producto se forma una cadena A-C-V que nos lleva desde los atributos expresados en lo que solicita el cliente a las consecuencias, a las razones reales de por qué pide eso el cliente. La técnica de *laddering* es lo que nos permite seguir esta cadena de razonamientos en una entrevista con el cliente. Para aplicarlo se seguirán los siguientes pasos:

1º : Buscar el por qué de las características de la funcionalidad que no encaja con lo solicitado. (Característica - Consecuencia).

2º : Partiendo de la consecuencia que da lugar a esas características preguntar sobre las causas que provocan esa consecuencia (Consecuencia - Causa).

3º : Partiendo de la causa se negocia por el problema que da origen a la causa explicada (Causa - Valor).

Un ejemplo de esto sería la siguiente situación: El cliente expresa que el precio de un máster ofertado en el sistema es demasiado preciso. Se le pregunta por qué es incorrecto que sea preciso el precio del máster. El cliente responde que esto puede llevar a engaño al usuario si el precio ha variado desde la última vez que se actualizó. Se le pregunta que a qué se debe que varíe el precio del máster. El cliente responde que no es la UVa la que fija el precio por crédito sino la JCyL y que por tanto varía cada año. Dado que el precio varía cada año se llega a un acuerdo de poner un precio aproximado y un enlace a la página de la UVa donde se muestre el precio por crédito en ese año, de esta forma no se cae en una imprecisión y se obtiene la funcionalidad requerida por el cliente. En esta secuencia se ha seguido la cadena Característica (precio demasiado preciso) - Consecuencia (engaño al usuario) - Causa (Precio variable por JCyL año a año) - Valor (Precio no calculado sino aproximado a partir de un valor guardado).

3.5. Conclusión de la selección de técnicas

En este capítulo se ha explicado la clasificación de las distintas técnicas para la elicitación de requisitos, se ha valorado como viable o no para la aplicación, sobre *chatbots* de interfaz textual, de cada una de ellas y se ha realizado la selección de las técnicas que formarán parte de la metodología propuesta. Se ha destacado la importancia en este caso de técnicas de modelado, basadas en agile y prototipado para el tratamiento de los problemas asociados al desarrollo del *chatbot*. De esta clasificación se han seleccionado las técnicas de *user stories*, escenarios, prototipos y laddering para incluir en la metodología. Por último se ha explicado los criterios de calidad y aplicación que guiarán su uso dentro de la propia metodología.

Capítulo 4

IMADTC: una Metodología Iterativa para el Análisis y Desarrollo de Chatbots Textuales

Una vez se han identificado los problemas a los que nos enfrentamos a la hora de analizar y diseñar chatbots, y las técnicas adecuadas a estos problemas, llega la hora de realizar la propuesta metodológica. El nombre de la propuesta desarrollada como resultado del trabajo realizado es IMADTC (*Metodología Iterativa para el Análisis y Desarrollo de Chatbots basados en Texto*). El capítulo presenta los objetivos a cubrir por la propuesta metodológica, la estructura de la propuesta y los métodos de validación de la misma.

4.1. Objetivos a cumplir con la propuesta

De acuerdo con los problemas detectados en la sección 2.4 sobre retos de desarrollo de *chatbots*, se presentan en esta sección los objetivos que la metodología debe cumplir si busca resolver o evitar estos problemas:

- Garantizar que el resultado de la metodología es un conjunto de artefactos que permiten llevar a cabo el análisis y diseño del sistema a desarrollar.
- Minimizar o hacer desaparecer los problemas de información identificados previamente mediante la aplicación de las técnicas escogidas e incorporadas en la metodología. Se pretende evaluar si la selección y combinación de técnicas escogidas es válido para minimizar o hacer frente a los problemas identificados.
- Mantener al cliente informado del estado de desarrollo del sistema al mismo tiempo que se obtiene *feedback* sobre el sistema. Para ello se pretende incorporar al cliente como pieza clave en el análisis y desarrollo del sistema.

4.2. Estructura de la Propuesta IMADTC

El diagrama genérico de la propuesta, cuyas fases se explicarán más en detalle seguidamente, se presenta en la figura 4.1. La secuencia de pasos es la siguiente:

- Contextualización del *chatbot* en su dominio. Una primera fase antes de empezar el proceso iterativo de contextualización del sistema.
- Ejecutar a continuación cíclicamente los siguientes pasos:
 - Preparación de objetivos

- Diseño de escenarios.
- Diseño del flujo conversacional
- Prototipo.
- Test de Usuario.
- Documentación.

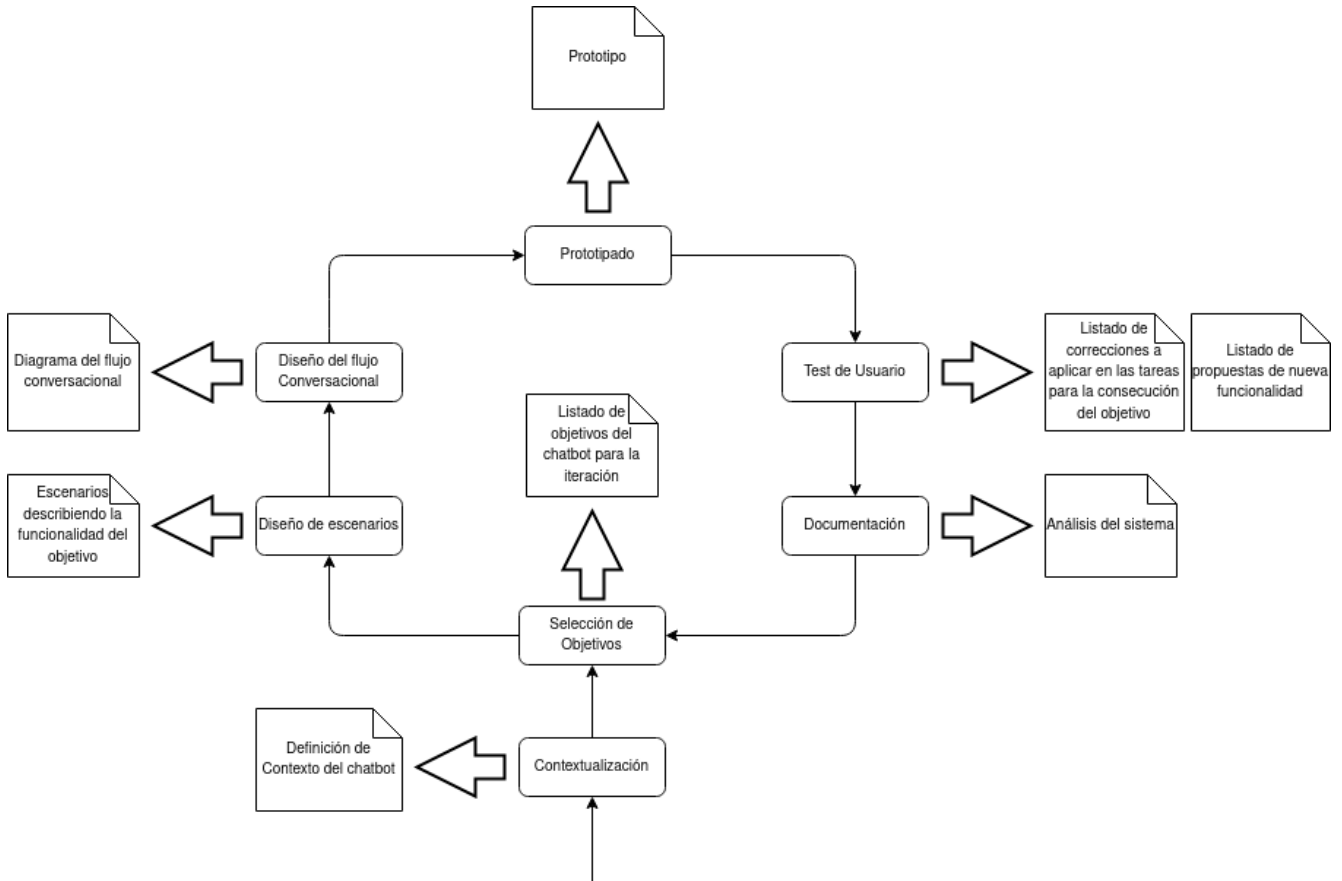


Figura 4.1: Esquema del flujo de fases a llevar a cabo en la metodología y los artefactos producidos en cada una

La metodología presentada es de carácter iterativo con un inicio secuencial para luego entrar en un bucle de iteraciones de continuo refinamiento sobre el producto a desarrollar. Dentro de este bucle se aplican las técnicas seleccionadas para el problema tratado, y explicadas en el capítulo 3.

Contextualización del *chatbot* en su dominio En esta primera fase, se lleva a cabo la especificación del entorno del *chatbot*, determinando el alcance del mismo y el rol que cumplirá en el entorno. El objetivo de esta fase es llevar a cabo una definición útil del entorno del *chatbot*, estableciendo los límites de este entorno así como los límites del rol que juega el *chatbot* en ese entorno. Para poder aplicar esta fase adecuadamente deberían estar presentes los principales actores relacionados con el proyecto de la parte de clientes y desarrolladores. Esta fase sería aconsejable realizarla antes, o al menos en el *kick off meeting*, que diese inicio al proyecto en que se enmarca el *chatbot* y que fuesen ellos los que lo completasen. En esta fase se aplican los principios referidos a contextualización de las guías de principios de diseño de *chatbots* de calidad que aporta Microsoft [113]. Para ello, se debe contestar a las preguntas de:

- ¿Para qué propósito está desarrollando el *chatbot* como sistema? Esta pregunta pretende hacer reflexionar a clientes y proveedores del sistema a desarrollar sobre la necesidad del desarrollo del sistema. Para ello,

se debe explicar de forma general cuál es el problema a resolver para el cliente final mediante el *chatbot* y cómo el *chatbot* puede ayudar al cliente final a resolver el problema explicado.

- ¿Bajo qué rol opera el *chatbot*? Esta pregunta fuerza a pensar el conjunto de información y funciones de las que el *chatbot* debe disponer. También dentro de esta pregunta se puede establecer el tono y la forma de interacción del *chatbot* con los usuarios finales.
- ¿Va a interaccionar el *chatbot* con otros sistemas software? La pregunta está pensada para evaluar desde un punto de vista tecnológico qué integraciones o interacciones con otros sistemas debe tener el *chatbot* en un futuro. Esta pregunta puede parecer prematura, pero debido a la variabilidad de las plataformas de desarrollo de *chatbots*, o en el caso de crear un sistema propio, permite no perder de vista el entorno tecnológico en el que el *chatbot* interactuaría.
- ¿Qué es lo que el *chatbot* debe no hacer explícitamente? Esto debe forzar a establecer hasta dónde llega el rol del *chatbot* en el sistema, a qué información no debe tener alcance el chatbot.

Como resultado de esta fase se obtendría como artefacto una declaración de no más de una cara en la que se contestaría a las cuatro preguntas anteriores. Esta declaración no debe modificarse una vez aceptada por todos los participantes en la sesión del *kick off meeting*. Esto se debe a que modificaciones en ésta redimensionarían el sistema a analizar y desarrollar, impidiendo una correcta organización en torno a su producción. Es por eso que la declaración se realiza en un tono general, sin entrar en detalles, pero permitiendo especificar desde un primer momento el alcance del proyecto. Se pasa ahora a la ejecución cíclica de las fases hasta obtener el visto bueno de los clientes acabando el proyecto.

Selección de Objetivos En la segunda fase, se lleva a cabo una declaración de objetivos del *chatbot* de forma general. Esta fase está pensada para definir los objetivos de alto nivel del proyecto y sus limitaciones, no tanto para obtener una especificación exhaustiva. Por ello para la definición de los objetivos se usarán técnicas *Agile* como las *user stories*. La definición de las *user stories* se hará de acuerdo a la plantilla tradicional de: “*Como (rol) quiero (hacer algo) para (finalidad)*”.

El número de *user stories* para cada objetivo debe estar limitado por el propio cliente, que será el responsable de decidir hasta qué punto se considera inicialmente que un conjunto de *user stories* definen y cubren correctamente el objetivo sobre el que se está trabajando. Las *user stories* deberán tener las notas asociadas que permitan recuperar el estado de la conversación con el cliente que llevaron a su definición, así como notas sobre problemas detectados en la propia definición de la *story*. De esta forma en el frontal de la tarjeta que contiene la *story* se incluirá su definición, las notas y las preguntas no resueltas sobre la misma y en la parte trasera se incluirá los test y pruebas para validar la *story*.

Diseño de escenarios En la tercera fase se usará la técnica de escenarios para llevar a cabo la especificación del objetivo cuya funcionalidad se debe llevar a cabo. Para la definición del escenario, el cliente ha debido proporcionar en la *user story* o *stories* a la que se refiere el escenario los elementos necesarios para poder definir el propio escenario de acuerdo al metamodelo explicado previamente.

Diseño del flujo conversacional En la cuarta fase, se lleva a cabo, con la información recabada hasta el momento, un esquema representativo de la conexión de las tareas encontradas como parte de los escenarios de la tercera fase. El objetivo de este esquema es servir de esqueleto para la lógica que deberá seguir el bot en un momento posterior de programación, al mismo tiempo que se tiene una representación visual del flujo de la conversación con el usuario en un diagrama de flujo. No se sigue ningún tipo de diagrama en concreto (como podrían ser los diagramas de actividad o los de modelo de proceso de negocio en UML), ya que el objetivo no es obtener un documento técnico sino una representación visual que sirva de ayuda para más adelante. Lo que sí se sigue, para mantener una notación coherente entre diagramas distintos, es la notación de la ISO 5807 para *Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*[114]. Esta ISO establece el significado de los símbolos más comunes en

diagramas de flujo, por lo que, pese a no seguir las reglas asociadas a un diagrama concreto, un desarrollador que conozca la notación de esta ISO podrá entender el significado tras esta.

Prototipado En la quinta fase, se llevaría a cabo la programación del prototipo y se llevarían a cabo pruebas de caja negra para verificar que lo programado concuerda con lo solicitado por el cliente.

Test de usuarios y documentación En la sexta fase, una vez el prototipo ha sido desarrollado y probado desde el punto de vista funcional, (pruebas de caja negra), se debe mostrar al cliente y permitir que el propio cliente interactúe con él. En estas pruebas se registran automáticamente las conversaciones además de llevar a cabo una pequeña entrevista no estructurada después de la interacción del cliente con el sistema para recoger sus opiniones sobre su interacción. El cliente en este punto validaría el desarrollo del prototipo y consecuentemente los requisitos que le dan cabida. En caso de validar el trabajo, se pasaría a la fase de documentación donde se formalizarían los requisitos y se haría el desarrollo de la funcionalidad prototipada, si quedasen objetivos sin tratar en el sistema se comenzaría una nueva iteración en la fase de preparación de objetivos. En caso de no validarla se pasaría a la entrevista no estructurada donde mediante preguntas sucesivamente más específicas (técnica de Laddering) se llegaría al punto de no concordancia entre el prototipo desarrollado y el objetivo que representa. Esto se documentaría para la próxima iteración y se comenzaría una nueva iteración.

4.3. Validación de la propuesta

La validación de la propuesta metodológica se realiza en dos partes. Una primera parte en la que se valora la aplicación de cada una de las técnicas que forman parte de la metodología y una segunda parte donde se valora la calidad de la información obtenida como resultado de la aplicación de la metodología.

4.3.1. Validación de las técnicas incorporadas en la metodología

En la primera parte se debe valorar la aplicación de las técnicas. Para ello se debe seguir los siguientes criterios de calidad. El objetivo es verificar de una forma contrastable que se han aplicado las técnicas de la manera adecuada validando el resultado obtenido por la metodología. En caso de no poder validar el resultado obtenido, se tendrá que acotar qué no es correcto desde el punto de vista de la técnica aplicada y se tendrá que corregir antes de la siguiente iteración.

User stories

Para evaluar la aplicación de las user stories seguiremos el criterio INVEST [110]. En concreto el método de evaluación será para determinar si se está aplicando INVEST correctamente. Por ello para cada user story obtenida en el proceso de selección de objetivos se debe aplicar los siguientes pasos:

- Evaluación de Independencia (I): determinar el número de dependencias con otras User Story ya elaboradas detectadas. El objetivo es conseguir que no haya ninguna dependencia con otras user stories.
- Evaluación de Negociables (N): determinar si hay al menos una nota referida al proceso de conversación con el cliente. *Acción de corrección:* exponer lo entendido de la user story al cliente para verificar que ambos tenemos el mismo concepto del objetivo que debe cumplir el chatbot.
- Evaluación de Valor (V): determinar si la user story está escrita desde el punto de vista del cliente o no. *Acción de corrección:* descartar la user story si no es posible reescribirla en términos del valor que aporta la interacción del chatbot al usuario. Si es posible reescribirla en estos términos, reescribirla y validar seguidamente frente al cliente que es el resultado esperado de ese objetivo.
- Evaluación de Estimable (E): determinar si la user story se puede concretar como trabajo realizable dentro de la duración fijada para la iteración. *Acción de corrección:* si no es el caso, determinar si es posible simplificarla, y de serlo, validar con el cliente que se sigue cumpliendo el objetivo final.

- Evaluación de Pequeñas (S): determinar si la user story tiene un tamaño que permita su trabajo durante toda la iteración o debe desagregarse en más. *Acción de corrección:* dividirla en varias user stories donde cada una de ellas pueda ser realizada dentro del periodo establecido para la iteración.
- Evaluación de Testeable (T): determinar si la user story posee test para verificar su completitud de acuerdo a lo establecido por el cliente. *Acción de corrección:* cuestionar al cliente sobre cómo se comprobaría que se ha cumplido el objetivo al que se refiere la story.

La user story estará bien especificada y será válida si podemos responder que las evaluaciones anteriores son positivas, o en el caso del número de dependencias entre user stories con un número igual a cero. En el caso de que no podamos contestar afirmativamente a cada una de las preguntas anteriores, dependiendo de a qué preguntas no se pueda contestar afirmativamente, se deberá llevar a cabo la acción de corrección correspondiente.

Escenarios

Para evaluar la aplicación de los escenarios tendremos que contestar a las siguientes preguntas sobre el modelo de escenario que hemos planteado. El objetivo es conseguir contestar para cada escenario afirmativamente a cada una de las preguntas planteadas:

- ¿El escenario está referido a un solo objetivo (user story)? *Acción de corrección:* Si está referido a más de una, dividir el escenario en el mismo número de user stories que están referidas. Si no lo está a ninguna, eliminarlo o bien crear una user story a la que de cabida.
- ¿Caracteriza el contexto la situación en la que se produce la story? *Acción de corrección:* Elaborar una de los siguientes al menos: Precondición para cumplir el escenario, Contexto Temporal de realización del escenario, descripción de la situación del actor al llevar a cabo el escenario.
- ¿Se ha identificado al menos un recurso de información para llevar a cabo el escenario? *Acción de corrección:* Determinar qué información se requiere que exista para poder llevar a cabo el escenario.
- ¿Se ha identificado al menos a un actor llevando a cabo las tareas del escenario? *Acción de corrección:* Identificar el rol sobre el que se ha especificado la user story a la que da cabida el escenario.
- ¿Cubren los episodios descritos el caso de éxito del escenario? *Acción de corrección:* Determinar cuál es el resultado que se considera un éxito y marca la completitud de la user story y traducirlo a episodios para el escenario.
- ¿Cubren los episodios descritos todos los escenarios alternativos que se pueden dar en la interacción con el usuario: cancelación, interrupción, abandono? *Acción de corrección:* Determinar entre qué episodios puede haber puntos en los que se den escenarios alternativos y elaborar los episodios de esos escenarios a partir de esos puntos.

Al igual que con las user stories, el escenario estará bien especificado y será válido si podemos responder afirmativamente a las anteriores cuestiones. En el caso de que no podamos contestar afirmativamente a cada una de las preguntas anteriores, dependiendo de a qué preguntas no se pueda contestar afirmativamente, se deberá llevar a cabo la acción de corrección correspondiente.

Prototipado

La técnica de prototipado se validará por nuestra parte mediante la comprobación de que se ha obtenido un prototipo referido a las stories de la iteración. La validación correrá más a cargo de los test de usuarios en los que el cliente validará si el prototipo se corresponde con la funcionalidad solicitada para llevar a cabo en el *chatbot*.

Laddering

La técnica del laddering se aplica durante el test de usuarios para examinar la causa principal de una incorrección en el prototipo que se ha llevado a cabo. Se debe validar que hemos aplicado la técnica correctamente si podemos contestar afirmativamente a las siguientes preguntas para cada uno de los problemas planteados por el usuario en la evaluación:

- ¿Se han determinado cuál son los atributos o características asociados al problema identificado?
- ¿Se ha determinado su asociación con las consecuencias que los provocan?
- ¿Se ha determinado la asociación de estas consecuencias con el valor central que no se ha cumplido?

Al contestar afirmativamente a estas preguntas y exponer el resultado al cliente para su confirmación se valida el resultado de la técnica.

4.3.2. Validación de la información obtenida por la aplicación de la metodología

Se indican aquí las preguntas objetivo a contestar como parte del proceso de test de usuarios durante la aplicación de la metodología para el desarrollo del sistema. Estas preguntas son las que validan la aplicación de la metodología como un todo y si realmente se está obteniendo información suficiente para aplicar el posterior análisis consiguiendo la elicitación de los requisitos del sistema.

- P1: ¿Qué número de participantes hemos podido involucrar?
- P2: ¿Qué características poseen estos participantes?
- P3: ¿Cuál es el tiempo que hemos necesitado de ellos?
- P4: ¿Se ha confirmado la mayoría de cosas enseñadas en el prototipo?
- P5: ¿Se han propuesto mejoras a lo mostrado en el prototipo?
- P5: ¿Se ha propuesto añadir mayor funcionalidad de lo mostrado en el prototipo?
- P6: ¿Podemos valorar positivamente el resultado obtenido en la prueba en términos de participación?
- P7: ¿Han entendido los participantes del test con usuarios lo que se requería de ellos o ha sido necesario aclararlo varias veces?
- P8: ¿Se requiere recoger más información de los usuarios tras haber realizado el test de usuario?

4.4. Conclusión de la metodología

Como conclusión a este capítulo hay que decir que se ha propuesto una metodología en torno a las técnicas anteriormente identificadas y esperando dar respuesta a los problemas asociados al desarrollo de un *chatbot* textual de preguntas y respuestas. Por último se ha propuesto los criterios de validación de la metodología, tanto técnica a técnica como de forma general para determinar si es viable.

Capítulo 5

Aplicación y Evaluación de la propuesta

En este capítulo se presenta el proceso de validación de la metodología propuesta, a través de la aplicación de la misma al caso de VirtualForest que fue descrito en el capítulo 1. En total se han identificado 14 user stories, escenarios y diagramas de flujo conversacional. Estos han sido expuestos en las secciones 5.3.2, 5.3.3 y 5.3.4. A continuación se presenta la organización del trabajo, el perfil de los participantes del sistema, los artefactos elaborados como parte de la metodología, el propio análisis del sistema en función de estos artefactos, el desarrollo y pruebas del sistema y la discusión de los resultados.

5.1. Organización del trabajo

En el transcurso de este TFM se han realizado 4 iteraciones. En esta sección se indica el trabajo llevado a cabo en cada iteración. Cada iteración ha comenzado al día siguiente del test de usuario con los clientes y ha tenido una duración de dos semanas.

- Primera iteración: Contextualización del *chatbot*. User Stories N^o 1 y 2 . Escenarios 1 y 2. Diagramas de flujo Conversacional 1 y 2. Prototipo 1. Test de usuario 1.
- Segunda iteración: User Stories N^o 3, 4, 5, 6 y 7 . Escenarios 3, 4, 5, 6 y 7. Diagramas de flujo Conversacional 3, 4, 5, 6 y 7. Prototipo 2. Test de usuario 2.
- Tercera iteración: User Stories N^o 8, 9, 10, 11 . Escenarios 8, 9, 10, 11. Diagramas de flujo Conversacional 8, 9, 10, 11. Prototipo 3. Test de usuario 3.
- Cuarta iteración: User Stories N^o 12, 13, 14 . Escenarios 12, 13, 14. Diagramas de flujo Conversacional 12, 13, 14. Prototipo 4. Test de usuario 4.

Los test de usuario han consistido en sesiones de 30 minutos donde en los primeros 5 minutos se ha explicado por parte del desarrollador la funcionalidad prototipada, en los siguientes 10 minutos se ha dejado a los usuarios interaccionar con el *chatbot* y en el resto del tiempo se ha recogido la opinión de los participantes sobre la funcionalidad inspeccionada.

5.2. Contextualización de los participantes en los test de usuario

Bajo el rol de participantes en la aplicación de la metodología y como clientes participaron tres personas pertenecientes a la ETSIIAA (Escuela Técnica Superior de Ingenierías Agrarias de Palencia). Dos de estas personas son profesores de la ETSIIAA y un alumno de la misma escuela que ha finalizado sus estudios en la misma y colabora como técnico en el IUFOR. Los profesores pertenecen también al IUFOR por lo que poseen los conocimientos suficientes tanto en materia Forestal como en entornos educativos para llevar a cabo un proceso de recogida de datos sobre ellos.

5.3. Artefactos elaborados por la aplicación de la propuesta

Se presenta en esta sección los artefactos resultantes de la aplicación de la metodología al proyecto VirtualForest. Su organización, estructura y formato se especifican de acuerdo a lo descrito tanto en los criterios de validación de las técnicas aplicadas, como en las fases descritas como parte de la metodología.

5.3.1. Contextualización del sistema

- ¿Para qué propósito está desarrollando el *chatbot* como sistema?
El *chatbot* a desarrollar pretende dar solución a un problema de gestión de respuestas a preguntas frecuentemente realizadas por estudiantes de grado a la hora de decidir si están interesados en cursar los estudios de máster disponibles en la ETSIIAA. Para ello se tendrán que contestar preguntas generales en torno a los másteres de Montes, Dataforest y el programa de Doble Máster en Montes y Dataforest. Además se contestarán preguntas sobre cuestiones relacionadas con el funcionamiento de los procesos administrativos de la universidad.
- ¿Bajo qué rol opera el *chatbot*?
El chatbot operará bajo un rol consultivo de tutor, respondiendo a las preguntas dirigidas por los estudiantes interesados en cursar un máster del ámbito forestal al *chatbot*. Las preguntas que le dirijan los estudiantes a este chatbot se centrarán en el contenido general, la organización y los trámites a realizar bajo los cursos que dure el máster en el que están interesados. Además el *chatbot* no debe utilizar un tono excesivamente coloquial pero debe utilizar un tono y una forma de expresarse cercana con el usuario. El nombre que se ha elegido para el *chatbot* será VirtualForestBot.
- ¿Va a interactuar el *chatbot* con otros sistemas software?
El *chatbot* tendrá que interactuar desde una interfaz web o desde un cliente de aplicación de mensajería instantánea como Telegram o WhatsApp con los usuarios.
- ¿Qué es lo que el *chatbot* debe no hacer explícitamente?
El chatbot no debe valorar casos demasiado particulares de un estudiante al ser un sistema destinado a un público general. Salvo en el caso concreto de que se requiera recabar información personal del usuario, el *chatbot* se limitará a dar explicaciones que no sean personalizadas para poder abarcar un público general.

5.3.2. Selección de objetivos a través de user stories

Se muestran a continuación una reproducción de las tarjetas utilizadas para las user story intentando mantener lo más fielmente posible el formato definido previamente:

User Story	Nº1
Enunciado:	<i>Como interesado en cursar un máster quiero conocer la oferta de másteres disponibles para informarme sobre uno afín a mis intereses.</i>
Notas:	Nota1: Para conocer la oferta basta con decir qué másteres hay disponibles. Nota2: Los másteres disponibles actualmente son Montes, Dataforest, el programa de Doble Máster en Montes y Dataforest, el programa de Doble Máster Internacional en AgroParisTech y Montes y el programa de Doble Máster Internacional en AgroParisTech y Dataforest. Nota3: X Nota4: X
Tests:	Test1: Se debe listar los másteres disponibles cuando se pregunte por la oferta de másteres. Test2: X Test3: X Test4: X

User Story	Nº2
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre requisitos de titulación para el acceso para saber si me puedo apuntar al máster de mi interés.</i>
Notas:	<p>Nota1: En el caso del máster de Montes, en el caso del Doble Máster en Montes y Dataforest y en el Doble Máster Internacional en AgroParisTech y Monte, para que un interesado se apunte debe tener un grado en Ingeniería Forestal validado por una universidad española. Si un interesado posee un graduado concedido por una entidad extranjera debe homologarlo a un graduado español en Ingeniería Forestal primero antes de acceder. Si un interesado no posee un graduado en Ingeniería Forestal no puede acceder.</p> <p>Nota2: En el caso de Dataforest y el Doble Máster Internacional en AgroParisTech y Dataforest para que un interesado acceda debe tener un grado. Si un interesado posee un graduado concedido por una entidad extranjera no tiene por qué homologarlo a su correspondiente graduado español. Si un interesado no posee un graduado no puede acceder.</p> <p>Nota3: En el caso de Montes o Dataforest los interesados de Francia o Vietnam pueden hacer uso de las colaboraciones con los centros de esos países.</p> <p>Nota4: En el caso de másteres con colaboraciones, se le debe indicar al interesado que pueda acceder la existencia de estas.</p>
Tests:	<p>Test1: Si el interesado tiene un graduado validado, homologado o que no requiere homologación, que le habilite acceso a un máster que lo requiere, se le debe proporcionar acceso a la preinscripción.</p> <p>Test2: Si el interesado tiene un graduado que le habilite para entrar a un máster que requiere graduado homologado pero expedido por una entidad extranjera y no homologado, se le debe indicar que se homologue.</p> <p>Test3: Si el interesado no posee un graduado que habilite para acceder al máster se le debe indicar que no puede acceder.</p> <p>Test4: Si el interesado ha preguntado por un programa de máster con colaboraciones y tiene un título de grado expedido por una entidad de ese país se le debe informar de esas colaboraciones.</p>

User Story	Nº3
Enunciado:	<i>Como interesado en cursar un máster quiero conocer el cronograma del programa en el que estoy interesado para saber cuanto tiempo necesitaría para completarlo.</i>
Notas:	<p>Nota1: El cronograma del programa se da en años académicos o en meses de inicio y fin del programa.</p> <p>Nota2: El número de cursos académicos de duración del máster de Montes es 1,5 , para el de Dataforest es 1,5 , para los másteres de AgroParisTech con Montes y con Dataforest son 2 y para el Doble máster en Montes y Dataforest es 2.</p> <p>Nota3: El primer curso del máster de Montes y de Dataforest empezaría en Septiembre para acabar en Junio del siguiente año. Y comenzaría el segundo curso en Septiembre para acabar en Febrero del siguiente.</p> <p>Nota4: El primer curso de los dobles máster empezaría en Septiembre para acabar en Junio del siguiente año. Y comenzaría el segundo curso en Septiembre para acabar en Junio del siguiente.</p>
Tests:	<p>Test1: Se debe dar el cronograma de un programa cuando se solicite tanto en años académicos como en meses de inicio y de fin.</p> <p>Test2: X</p> <p>Test3: X</p> <p>Test4: X</p>

User Story	Nº4
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre créditos ECTS para entender como se pondera cada asignatura dentro del programa de máster.</i>
Notas:	<p>Nota1: El crédito ECTS es el Sistema Europeo de Transferencia de Créditos académicos.</p> <p>Nota2: En nuestro caso un ECTS son 25 horas de trabajo del alumno (10 horas de clases + 15 horas de trabajo autónomo y estudio del alumno).</p> <p>Nota3: X</p> <p>Nota4: X</p>
Tests:	<p>Test1: Se debe dar una explicación breve de lo que es un crédito ECTS cuando se pregunte por ellos.</p> <p>Test2: X</p> <p>Test3: X</p> <p>Test4: X</p>

User Story	Nº5
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre los requisitos de idiomas para saber si puedo cursarlo adecuadamente.</i>
Notas:	<p>Nota1: Todos los másteres ofertados tienen un nivel mínimo de Inglés de B1. En el Dataforest y el Doble máster se recomienda el B2.</p> <p>Nota2: Un máster solo puede tener un único idioma distinto al español de impartición.</p> <p>Nota3: X</p> <p>Nota4: X</p>
Tests:	<p>Test1: Se debe dar una explicación breve de que se requiere mínimo un nivel B1 en Inglés.</p> <p>Test2: X</p> <p>Test3: X</p> <p>Test4: X</p>

User Story	Nº6
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre los requisitos de lenguajes de programación que se considera necesarios conocer para cursar el programa sin complicación.</i>
Notas:	Nota1: Es recomendable programación en R y Python aunque no hay requisitos de programación fijados. Nota2: No se exigirían, de exigirse, más que esos dos lenguajes de programación. Nota3: X Nota4: X
Tests:	Test1: Se debe informar que sería recomendable saber R o Python cuando se pregunte sobre lenguajes de programación pero que no es estrictamente necesario. Test2: X Test3: X Test4: X

User Story	Nº7
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre la modalidad de presencialidad necesaria en que se imparte el máster para saber si podría asistir a clase o no.</i>
Notas:	Nota1: Todos los másteres ofertados actualmente son presenciales. Nota2: Los niveles de presencialidad son: No presencial, Presencial y Semipresencial. Nota3: X Nota4: X
Tests:	Test1: Se debe informar de la necesidad de presencialidad 100% cuando se pregunte por la modalidad de presencialidad de un máster . Test2: X Test3: X Test4: X

User Story	Nº8
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre el precio de un programa de máster de mi elección para saber si puedo afrontar cursarlo o no.</i>
Notas:	Nota1: El precio de los programas de Máster suele variar de año en año pero está en torno a 2800 euros (31,14 Euros/crédito * 90 créditos) para Montes, 3600 euros (39,50 Euros/crédito * 90 créditos) para Dataforest, 4000 euros (31,14 * 129 créditos) para el Doble Máster de Montes y Dataforest, 3700 (31,14*120) para el de AgroParistech con Montes y 4700 (39,50*120) para el de AgroParisTech con Dataforest. Nota2: Los precios no son calculados, son orientativos y deben incluir una rebaja a aplicar este año, Montes (2800 euros), Dataforest (1940 euros), Doble máster (3000 euros), Montes + AgroParisTech (3700 euros), Dataforest + AgroParisTech (4700 euros). Nota3: X Nota4: X
Tests:	Test1: Se debe indicar un precio orientativo para el máster seleccionado por le interesado . Test2: X Test3: X Test4: X

User Story	Nº9
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre el contenido del programa de máster de mi elección para saber si me interesa cursarlo.</i>
Notas:	Nota1: La descripción del contenido del máster contiene información de en qué consiste el máster, si es un máster simple o doble y qué convenios con qué entidades posee. Nota2: Los másteres simples otorgan un único título mientras que los doble otorgan dos títulos cursando un único programa. Nota3: Los convenios con otras entidades son a nivel de máster y permiten llevar a cabo cursos del programa de máster en estas entidades. Nota4: X
Tests:	Test1: Se debe indicar una breve descripción del contenido del máster, si es un doble máster o no y de los convenios que posee . Test2: X Test3: X Test4: X

User Story	Nº10
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre las colaboraciones con entidades externas a la UVa para valorar la calidad del máster</i>
Notas:	Nota1: Los convenios de colaboración se caracterizan por el máster al que están referidos y por la entidad de acogida. Nota2: Los convenios deben explicarse mediante la identificación de la entidad y descripción de las actividades esperadas para su desarrollo. Nota3: X Nota4: X
Tests:	Test1: Se debe describir una colaboración indicando el nombre de la entidad de acogida, el máster bajo el que se produce, el país de la entidad de acogida y una descripción de la actividad que se realiza. Test2: X Test3: X Test4: X

User Story	Nº11
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre la posibilidad de convalidar asignaturas por experiencia previa para no tener que cursar asignaturas cuyo contenido ya conozco.</i>
Notas:	Nota1: Tanto por experiencia previa, por haber cursado asignaturas similares en otro master o por querer convalidar practicas en empresa se debe presentar antes de la matriculación, una solicitud en la sede electronica de la UVa. Nota2: X Nota3: X Nota4: X
Tests:	Test1: Se debe informar al usuario cuando pregunte sobre este tema que debe solicitarlo por la sede electrónica. Test2: X Test3: X Test4: X

User Story	Nº12
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre la asignatura de prácticas en empresa para valorar si me interesa cursar el máster con la oferta disponible.</i>
Notas:	Nota1: La información relevante de las prácticas en empresa es que son obligatorias y que es posible realizarlas con entidades internacionales. Nota2: X Nota3: X Nota4: X
Tests:	Test1: Se debe informar al usuario cuando pregunte sobre este tema de la obligatoriedad de las prácticas y de su posibilidad de cursarlas en el extranjero. Test2: Se debe proporcionar la URL de la ETSIA donde se explica la oferta disponible Test3: X Test4: X

User Story	Nº13
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre las ayudas económicas disponibles para valorar si puedo solicitar alguna para cursar el máster.</i>
Notas:	Nota1: La información relevante a las becas es muy variable de año en año, hay que consultarla y valorarla caso a caso. Nota2: Es mejor que se muestre el listado de becas de la UVa disponibles por la url correspondiente o que se contacte con el coordinador del master. Nota3: X Nota4: X
Tests:	Test1: Se debe proporcionar la URL de la UVa donde se muestra la oferta disponible. Test2: X Test3: X Test4: X

User Story	Nº14
Enunciado:	<i>Como interesado en cursar un máster quiero información sobre el correo de contacto de coordinación del máster para consultar información no presente en el chatbot.</i>
Notas:	Nota1: Sería necesario conocer el correo de coordinación de cada máster. No el personal del responsable actual. Nota2: X Nota3: X Nota4: X
Tests:	Test1: Se debe proporcionar el correo de coordinación general asociado a cada master. Test2: X Test3: X Test4: X

5.3.3. Diseño de escenarios

Escenario	Nº1
Objetivo:	User Story Nº1. Se requiere conocer los másteres de ciencias forestales disponibles:
Contexto:	Un interesado en cursar máster quiere conocer los másteres relacionados con el ámbito de la Ingeniería Forestal que ofrece la UVa en Palencia.
Recursos necesarios:	Listado de másteres disponibles en la UVa en Palencia para el ámbito de Ingeniería Forestal.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita la oferta de másteres disponibles. ▪ ES2: Se recupera el listado de másteres con docencia disponible para el siguiente curso. ▪ EC2.1: Si hay másteres disponibles para elaborar el listado se muestra el listado recuperado. ▪ EC2.2: Si no hay másteres disponibles para elaborar el listado se indica que no se dispone en ese momento de la información necesaria.

Escenario	Nº4
Objetivo:	User Story Nº 4 . Se requiere consultar información sobre créditos ECTS:
Contexto:	Un interesado en cursar máster quiere conocer qué es un crédito ECTS.
Recursos necesarios:	Descripción de un crédito ECTS.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre los créditos ECTS. ▪ ES2: Se aporta información sobre los créditos ECTS.

Escenario	Nº3
Objetivo:	User Story Nº 3 . Se requiere consultar el cronograma de un Máster:
Contexto:	Un interesado en cursar máster quiere conocer la temporización del máster en que está interesado antes de empezarlo.
Recursos necesarios:	Cronogramas para los másteres de Montes, el de Dataforest y el programa Dual.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita la temporización del máster. Restricción: Los másteres de los que puede solicitar los cronogramas son el de Montes, Dataforest o Dual. ▪ ES2: Se comprueba si el máster posee un cronograma. ▪ EC2.1: Si el máster posee un cronograma se informa al interesado con un mensaje indicando la temporización del máster en años académicos y en meses. ▪ EC2.2: Si el máster no posee un cronograma se informa al interesado con un mensaje indicando que en ese momento no se dispone de esa información.

Escenario	Nº2
Objetivo:	User Story Nº2 . Se requiere consultar las condiciones de acceso a un Máster:
Contexto:	Un interesado en cursar máster quiere conocer las condiciones de acceso de un máster del ámbito de la Ingeniería Forestal que ofrece la UVa en Palencia.
Recursos necesarios:	Listado de másteres disponibles en la UVa en Palencia para el ámbito de Ingeniería Forestal.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita las condiciones de acceso a un máster. Restricción: Los másteres de los que puede solicitar las condiciones de acceso son el de Montes, Dataforest, Doble máster en Montes y Dataforest y los programas de Doble máster de AgroParisTech. ▪ EC2: Se solicita al interesado el país de la institución que expidió el título de acceso al máster. ▪ ES3.1: Si el interesado cesa la comunicación la conversación se extingue. ▪ ES3.2: Si el país indicado por el interesado es España se consulta el título de grado específico para acceder. ▪ EC3.2.1: Se cuestiona al interesado si tiene el título específico de grado. ▪ ES3.2.1.1: Si el interesado cesa la comunicación la conversación se extingue. ▪ ES3.2.1.2: Si el interesado dispone del título entonces se recuperan las colaboraciones del programa y se informa al interesado de que cumple las condiciones de acceso. ▪ EC3.2.1.1: Si el título del interesado está expedido por una entidad del mismo país que una colaboración existente se le informa de la colaboración disponible. ▪ ES3.2.1.3: Si el interesado no dispone del título entonces se indica que no puede acceder a ese máster pero que puede acceder a un máster que no tenga un requerimiento de grado específico. ▪ ES3.3: Si el país indicado por el interesado no es España se consulta si el programa de máster requiere un título de grado homologado. ▪ EC3.3.1: Si se requiere un título de grado homologado se consulta al interesado sobre si posee el título. ▪ ES3.3.1.1: Si el interesado cesa la comunicación la conversación se extingue. ▪ ES3.3.1.2: Si el interesado no posee el título se le indica que debe homologarse para acceder a ese máster o que acceda a un programa que no requiera homologación. ▪ ES3.3.1.3: Si el interesado posee el título entonces se recuperan las colaboraciones del programa y se le informa de que cumple las condiciones de acceso. ▪ EC3.3.1.3.1: Si el título del interesado está expedido por una entidad del mismo país que una colaboración existente se le informa de la colaboración disponible. ▪ EC3.3.2: Si no se requiere un título de grado homologado se consulta al interesado sobre si posee un título de grado universitario. ▪ ES3.3.2.1: Si el interesado cesa la comunicación la conversación se extingue. ▪ ES3.3.2.2: Si el interesado no posee el título se le indica que no puede acceder al programa de máster. ▪ ES3.3.2.3: Si el interesado posee el título entonces se recuperan las colaboraciones del programa y se le informa de que cumple las condiciones de acceso. ▪ EC3.3.2.3.1: Si el título del interesado está expedido por una entidad del mismo país que una colaboración existente se le informa de la colaboración disponible.

Escenario	Nº5
Objetivo:	User Story Nº 5 . Se requiere consultar información sobre requisitos de idiomas de un máster:
Contexto:	Un interesado en cursar máster quiere conocer qué idiomas son necesarios para poder cursar un máster.
Recursos necesarios:	Nivel de idiomas para cada idioma en que se imparte el máster.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre los idiomas del máster. Restricción: Los másteres de los que puede solicitar el nivel de idiomas son el de Montes, Dataforest o Dual. ▪ ES2: Se recupera información sobre los idiomas en los que se imparte el máster. ▪ ES3: Se recupera el nivel para cada uno de los idiomas en que se imparte el máster. ▪ ES4: Se informa al interesado sobre el nivel de idiomas de cada uno de los idiomas del máster.

Escenario	Nº6
Objetivo:	User Story Nº 6 . Se requiere consultar información sobre requisitos de programación de un máster:
Contexto:	Un interesado en cursar máster quiere conocer si hay requisitos de conocimientos de programación para poder cursar un máster.
Recursos necesarios:	Lenguajes de programación necesarios para poder cursar el máster.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre los lenguajes de programación necesarios para cursar el máster. Restricción: Los másteres de los que puede solicitar los lenguajes de programación asociados son el de Montes, Dataforest o Dual. ▪ ES2: Se recupera información los lenguajes de programación necesarios para el máster. ▪ ES3: Se informa al interesado sobre los lenguajes de programación necesarios para el máster.

Escenario	Nº7
Objetivo:	User Story Nº 7 . Se requiere consultar información sobre requisitos de presencialidad de un máster:
Contexto:	Un interesado en cursar máster quiere conocer en qué modalidad de presencialidad se puede cursar un máster.
Recursos necesarios:	Modalidad de presencialidad de cada máster.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre la presencialidad para cursar el máster. Restricción: Los másteres de los que puede solicitar los requisitos de presencialidad son el de Montes, Dataforest o Dual. ▪ ES2: Se recupera información sobre la presencialidad en que se imparte el máster. ▪ ES3: Se elabora un resumen informando al interesado.

Escenario	Nº8
Objetivo:	User Story Nº 8 . Se requiere consultar información sobre precios de un máster:
Contexto:	Un interesado en cursar máster quiere conocer los precios orientativos de un máster.
Recursos necesarios:	Precios orientativos de los másteres.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre el precio de un máster. Restricción: Los másteres de los que puede solicitar el precio son el de Montes, Dataforest o Dual. ▪ ES2: Se recupera información sobre el precio de un máster. ▪ ES3: Se informa al interesado sobre el precio del máster.

Escenario	Nº9
Objetivo:	User Story Nº 9 . Se requiere consultar información del contenido de un máster:
Contexto:	Un interesado en cursar máster quiere conocer en qué consiste un máster específico.
Recursos necesarios:	Descripción de un máster y convenios de un máster.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre el contenido de un máster. Restricción: Los másteres de los que puede solicitar el contenido son el de Montes, Dataforest o el Dual. ▪ ES2: Se recupera información sobre el contenido de un máster. ▪ ES3: Se informa al interesado sobre el contenido del máster, indicando estructura del máster (simple o doble) y convenios con entidades.

Escenario	Nº10
Objetivo:	User Story Nº10
Contexto:	Un interesado en cursar máster quiere conocer las colaboraciones asociadas a un máster en específico.
Recursos necesarios:	Colaboraciones de entidades para masteres de la ETSIA de la UVa en Palencia.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita las colaboraciones de un máster de su elección. Restricción: Los másteres de los que puede solicitar el contenido son el de Montes, Dataforest o el Dual. ▪ ES2: Se recupera las colaboraciones del máster solicitado. ▪ EC2.1: Si hay colaboraciones se elabora un resumen exponiendo las características de cada una. ▪ EC2.2: Si no hay colaboraciones se informa al interesado de este hecho.

Escenario	Nº11
Objetivo:	User Story Nº11
Contexto:	Un interesado en cursar máster quiere conocer si puede convalidar alguna asignatura por su experiencia previa.
Recursos necesarios:	No se requiere ningún recurso.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre cómo convalidar asignaturas por experiencia previa. ▪ ES2: Se presenta un mensaje informando al usuario de que debe presentar una solicitud a través de la sede electrónica.

Escenario	Nº12
Objetivo:	User Story Nº12
Contexto:	Un interesado en cursar máster quiere información sobre prácticas en empresa.
Recursos necesarios:	URL de la página de la ETSIA con info de prácticas en empresa.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre prácticas en empresa. ▪ ES2: Se recupera la descripción sobre prácticas en empresa. ▪ ES3: Se recupera la url de la ETSIA de prácticas en empresa. ▪ ES4: Se forma un resumen con la información de prácticas en empresa y se presenta al usuario.

Escenario	Nº13
Objetivo:	User Story Nº13
Contexto:	Un interesado en cursar máster quiere información sobre ayudas económicas (becas) para cursar el master.
Recursos necesarios:	URL de la página de becas de la UVa.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita información sobre becas para cursar el máster. ▪ ES2: Se recupera la descripción sobre becas del máster. ▪ ES3: Se recupera la url de la UVa para cursar las becas. ▪ ES4: Se forma un resumen con la información de becas y se presenta al usuario.

Escenario	Nº14
Objetivo:	User Story Nº14
Contexto:	Un interesado en cursar máster quiere consultar información de contacto de la coordinación del master.
Recursos necesarios:	Correos de contacto de los coordinadores de másteres ofertados.
Actores:	Interesado: Persona con interés en cursar un programa de máster oficial en el próximo año.
Episodios (ES, EC, EO):	<ul style="list-style-type: none"> ▪ ES1: El interesado solicita el correo del coordinador de un máster. ▪ ES2: Se recupera el correo del coordinador del máster. ▪ ES3: Se presenta al usuario la dirección de correo.

5.3.4. Diseño del Flujo Conversacional

Se presentan a continuación los diagramas de flujo conversacional resultantes de los escenarios especificados.

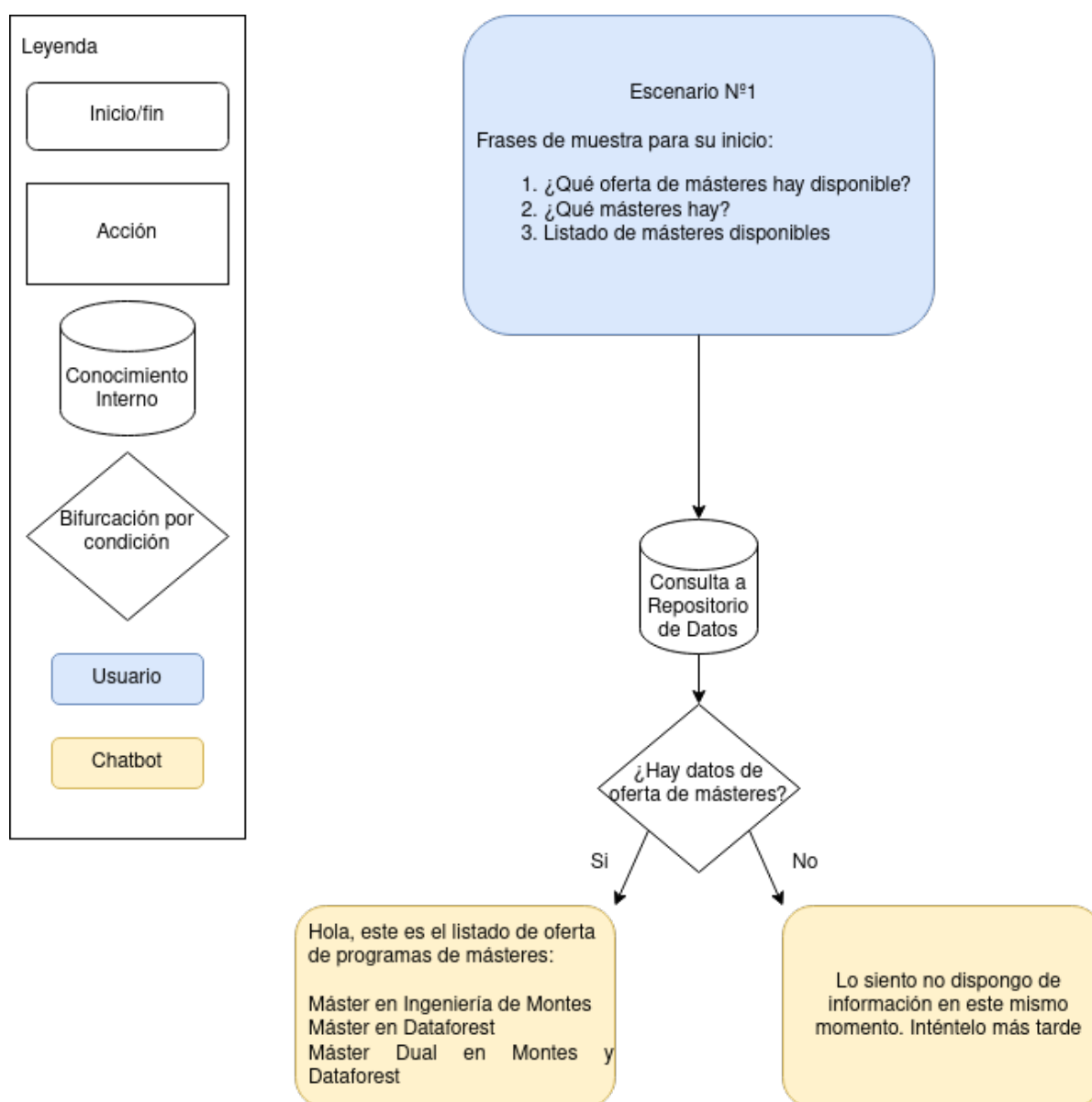


Figura 5.1: Diagrama de Flujo Conversacional 1: Listar la oferta de másteres

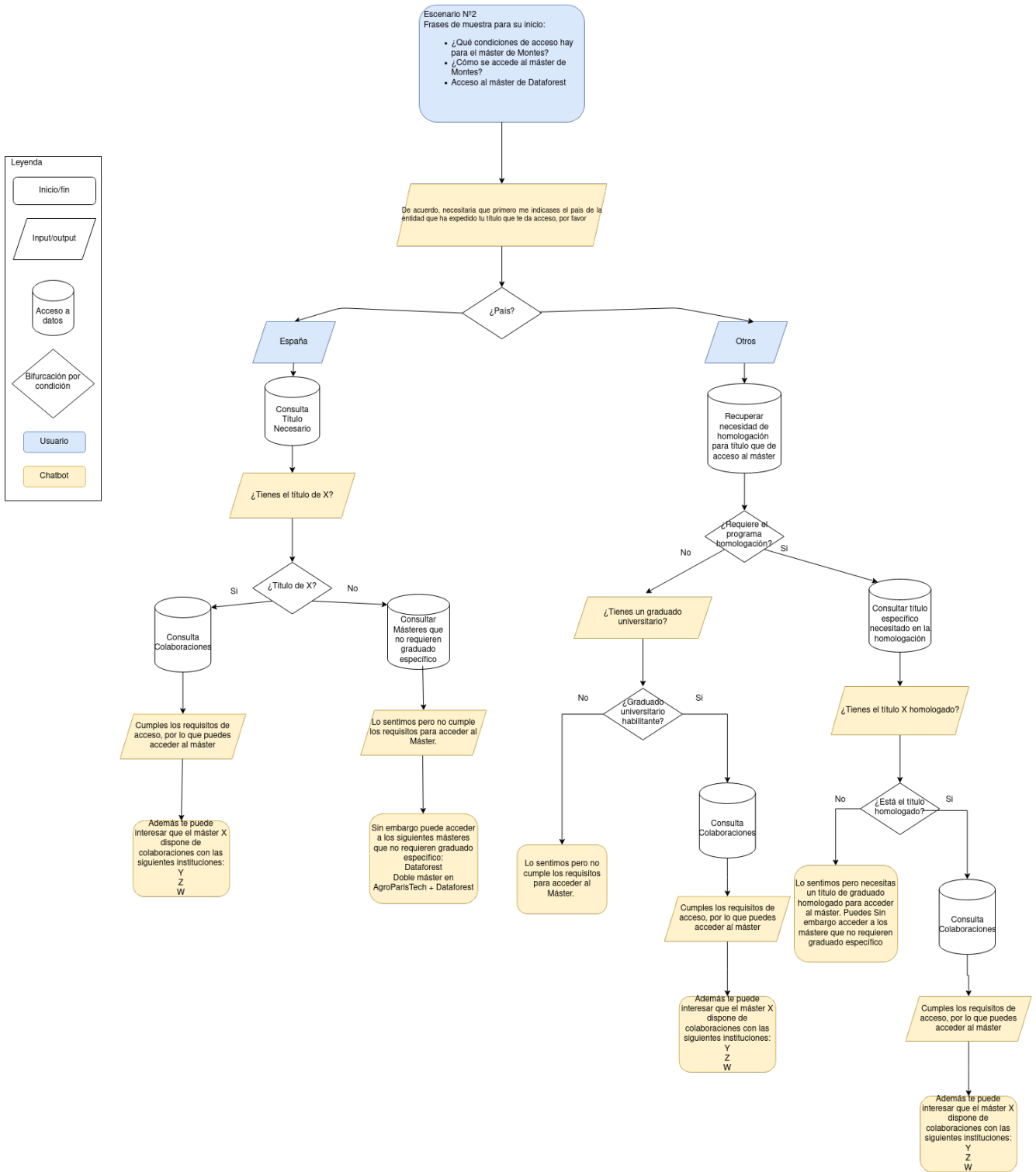


Figura 5.2: Diagrama de Flujo Conversacional 2: Consultar condiciones acceso a un máster

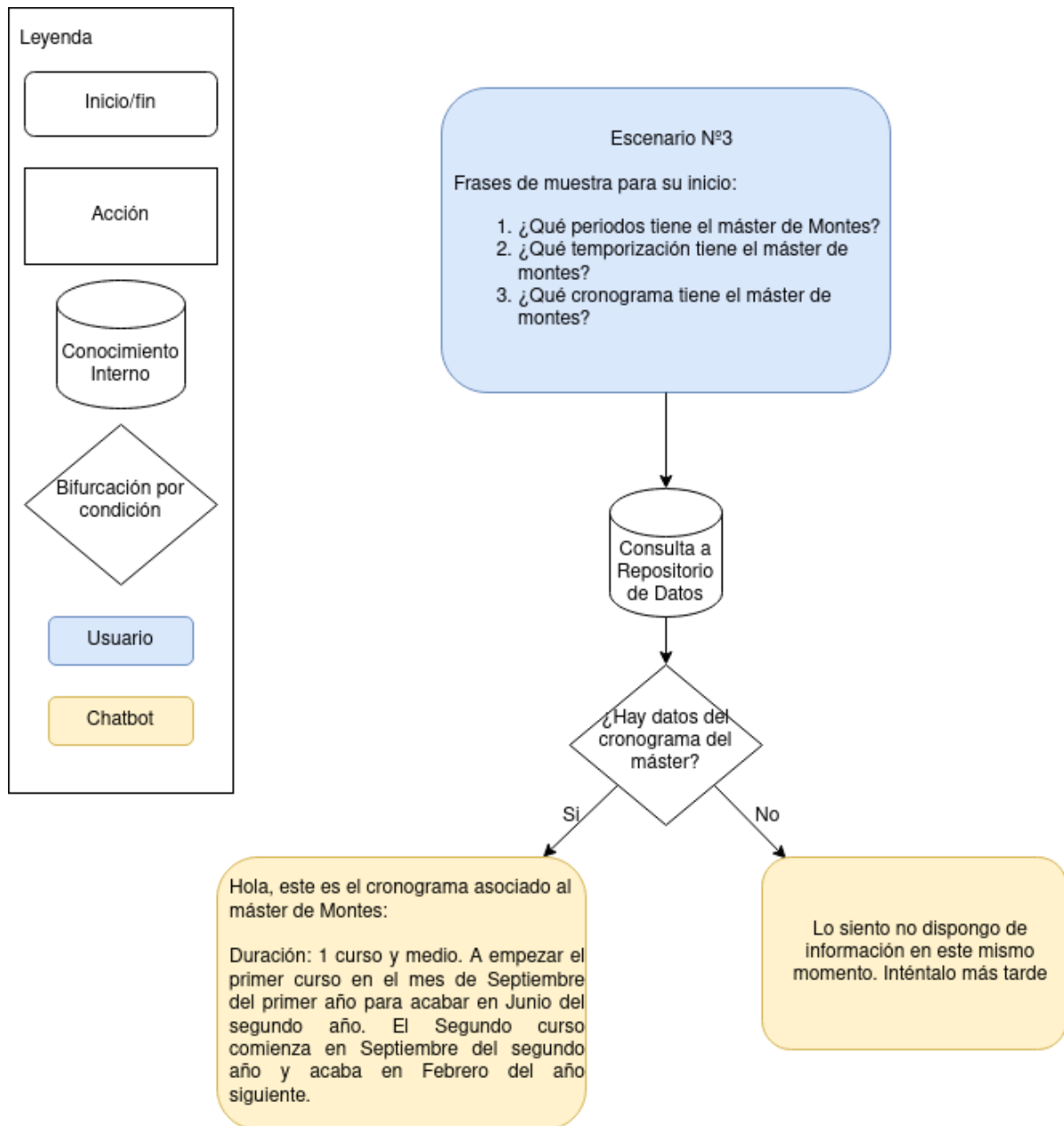


Figura 5.3: Diagrama de Flujo Conversacional 3: Consultar cronograma de un máster

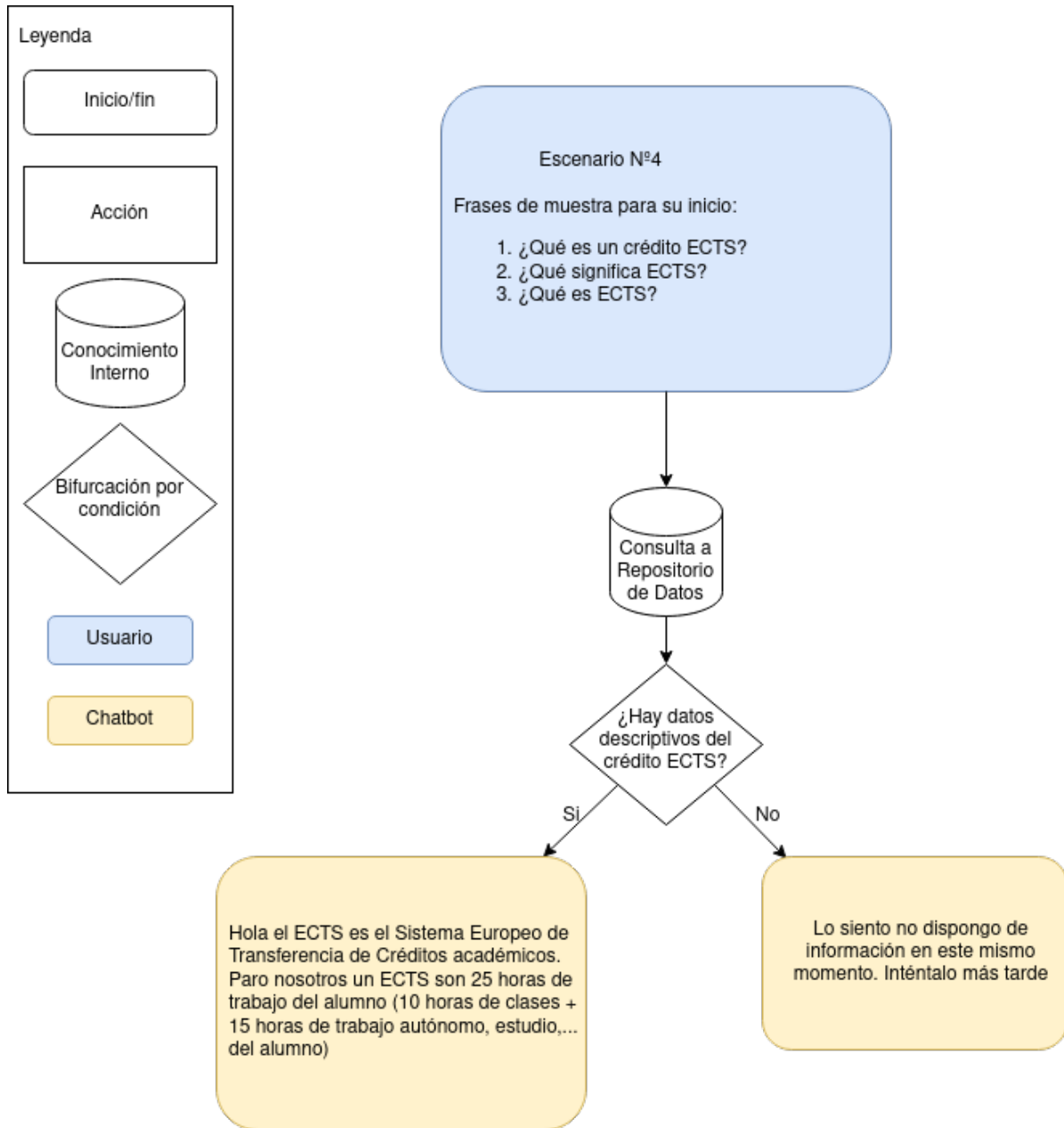


Figura 5.4: Diagrama de Flujo Conversacional 4: Consultar información sobre créditos ECTS

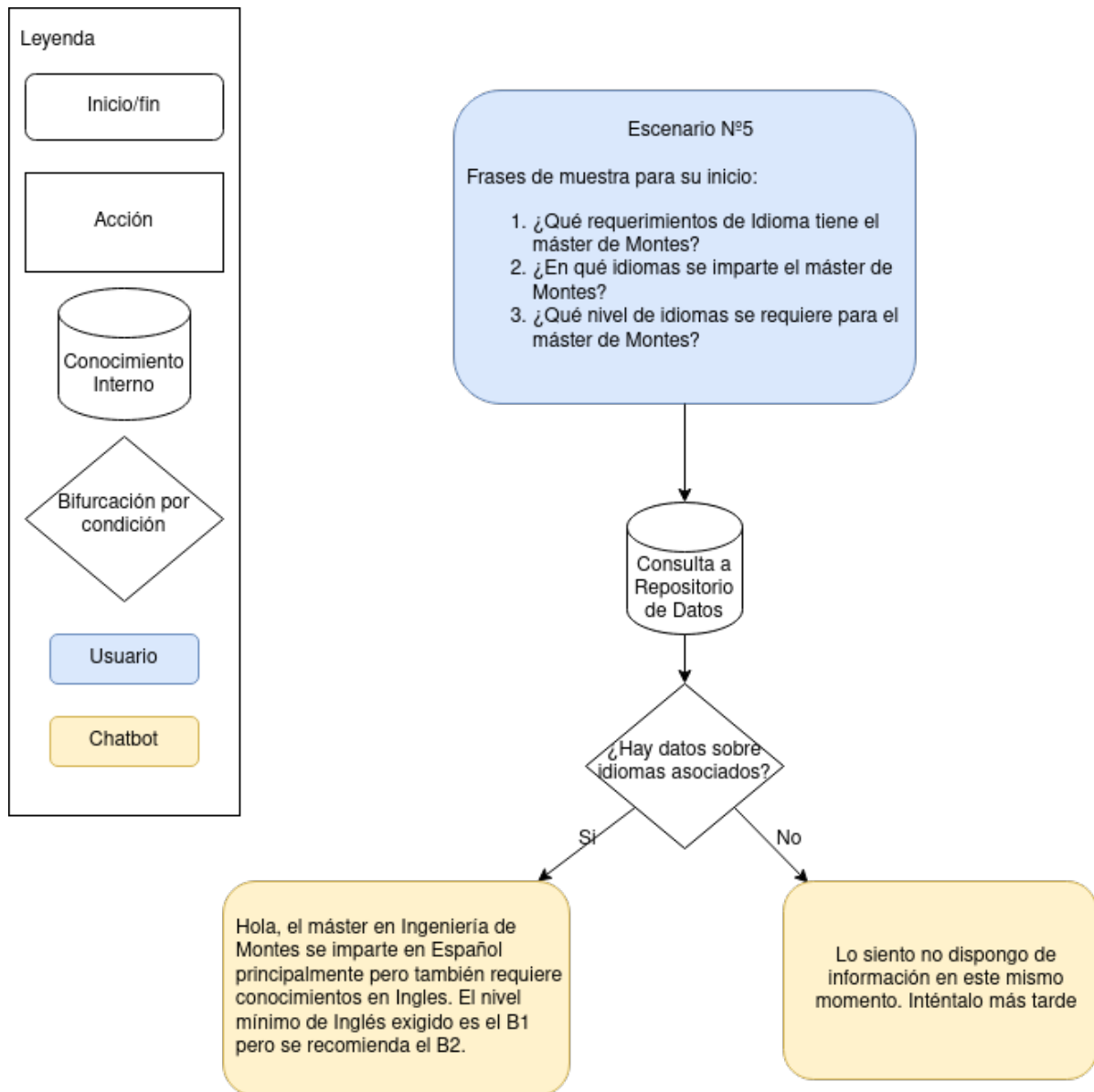


Figura 5.5: Diagrama de Flujo Conversacional 5: Consultar información sobre requisitos de idioma de un máster

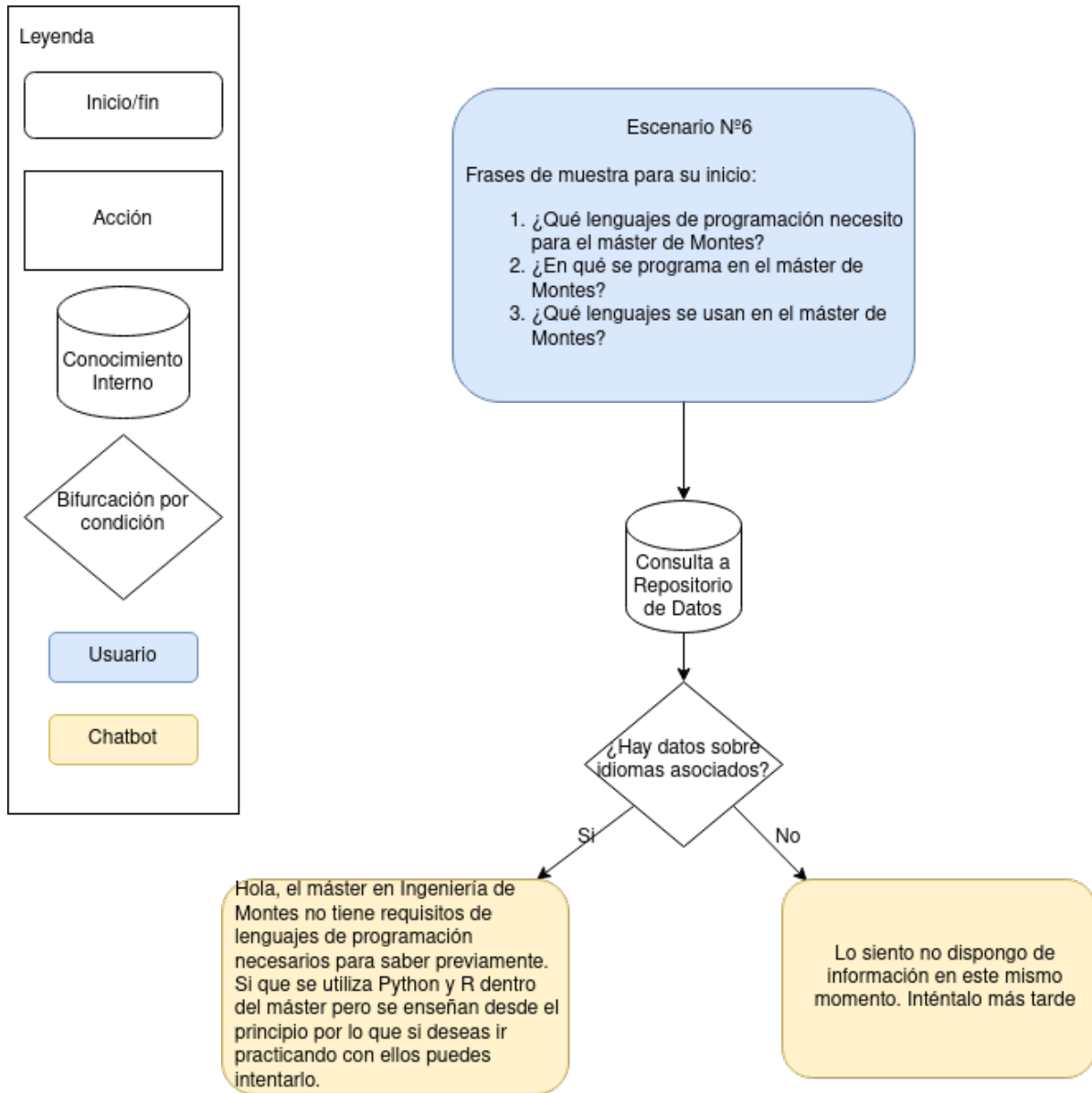


Figura 5.6: Diagrama de Flujo Conversacional 6: Consultar información sobre requisitos de lenguajes de programación de un máster

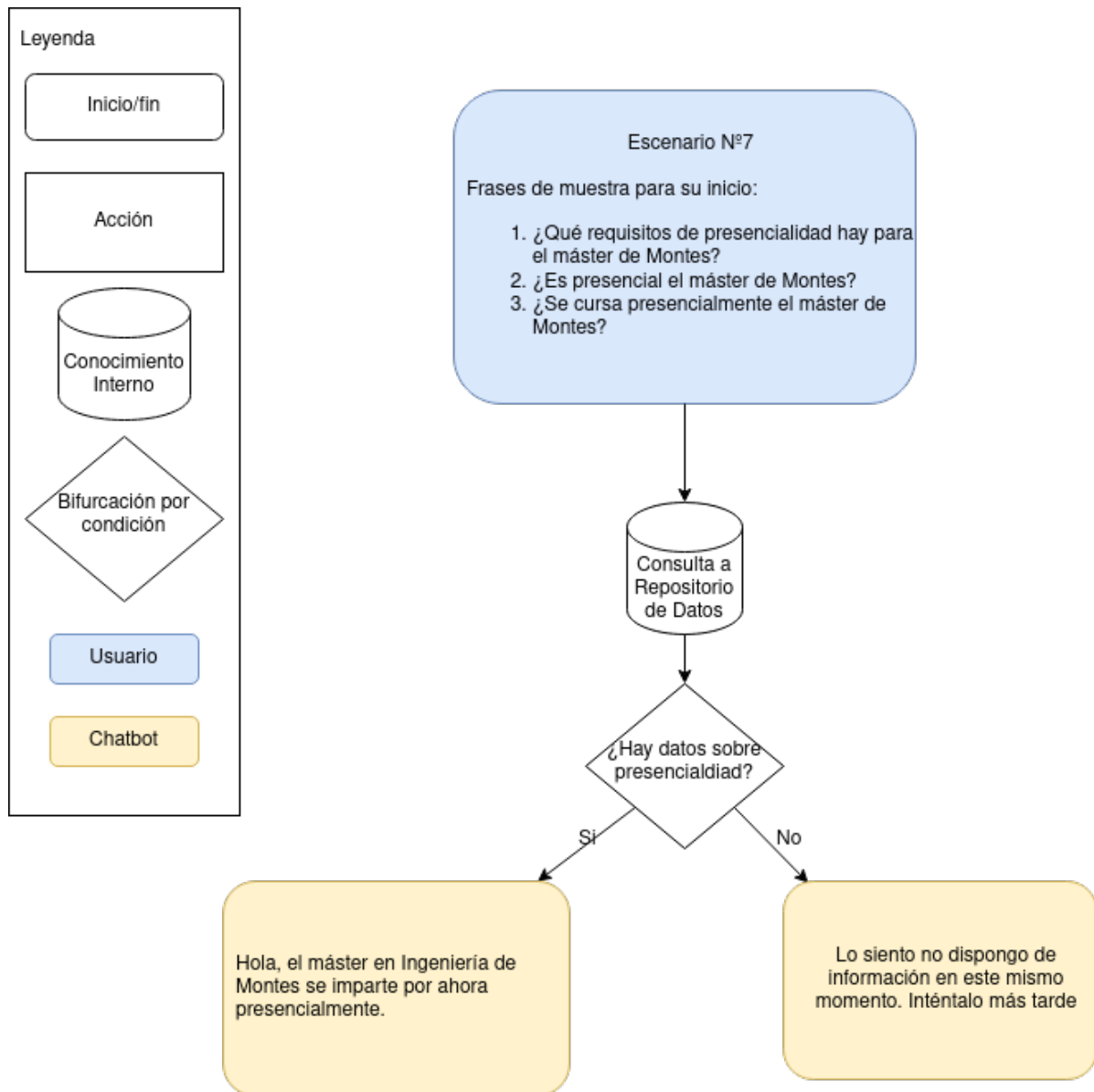


Figura 5.7: Diagrama de Flujo Conversacional 7: Consultar información sobre modalidad de presencialidad de un máster

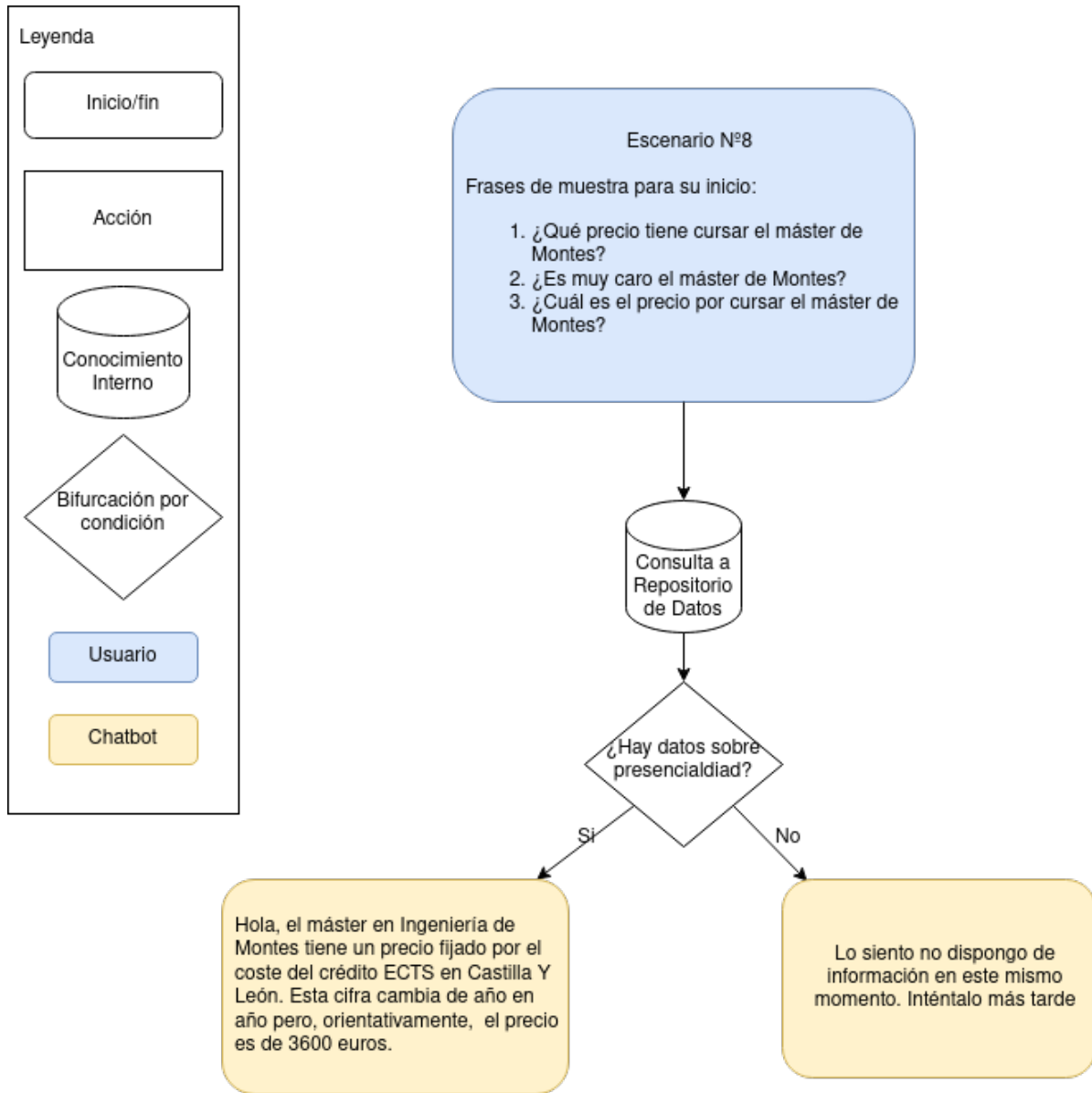


Figura 5.8: Diagrama de Flujo Conversacional 8: Consultar información sobre precios orientativos de un máster

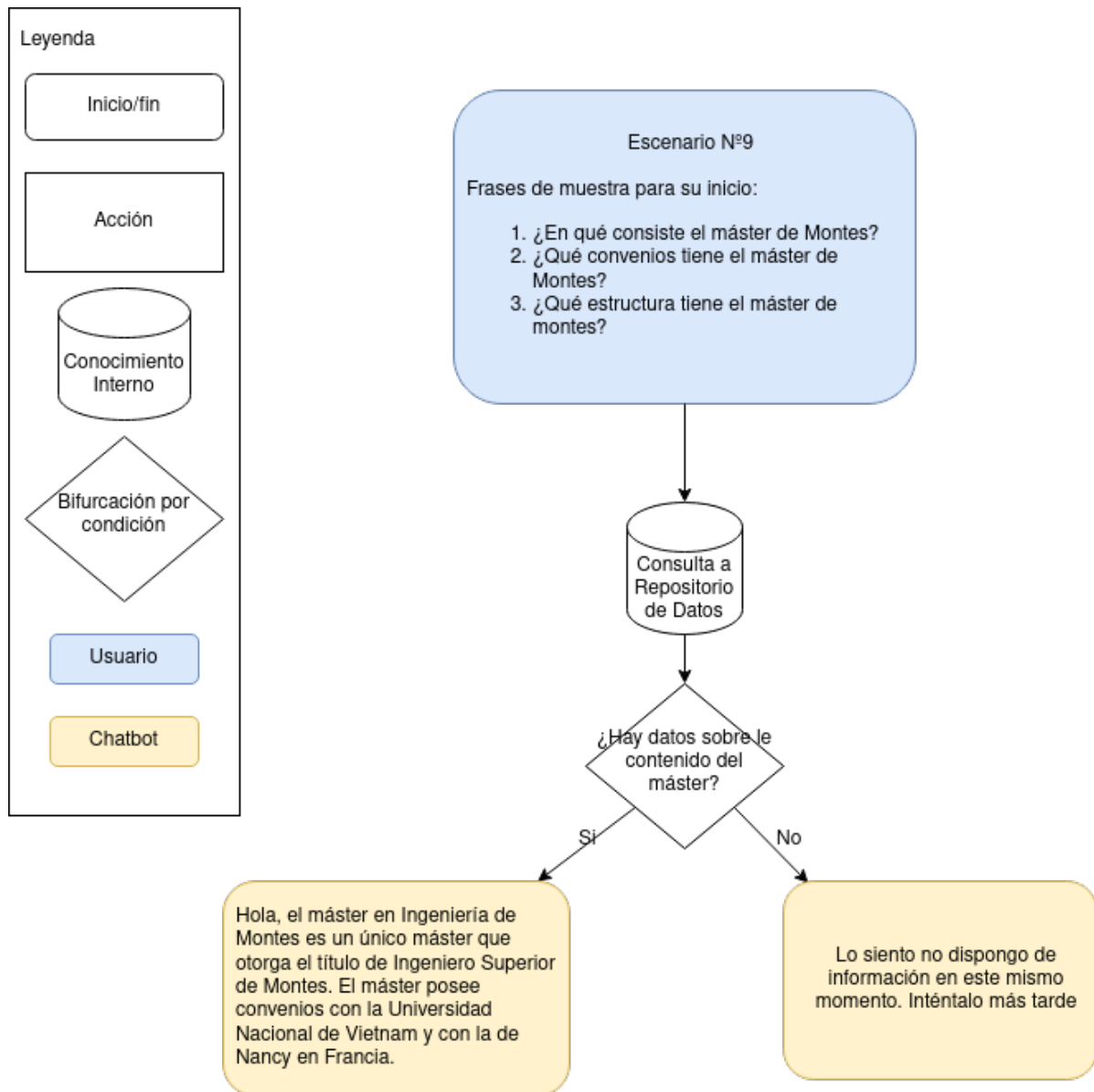


Figura 5.9: Diagrama de Flujo Conversacional 9: Consultar contenido de un máster

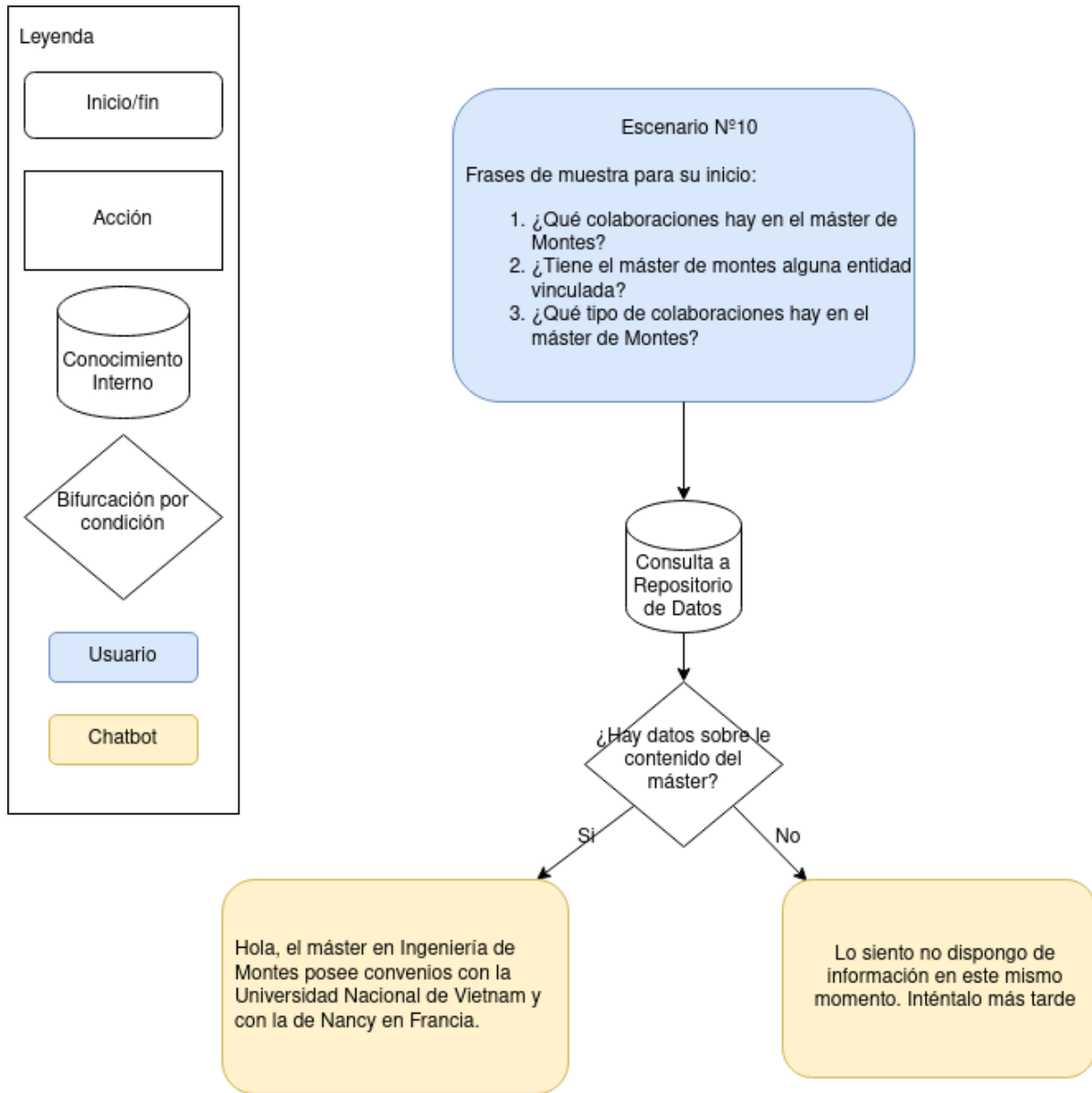


Figura 5.10: Diagrama de Flujo Conversacional 10: Consultar colaboraciones de un máster

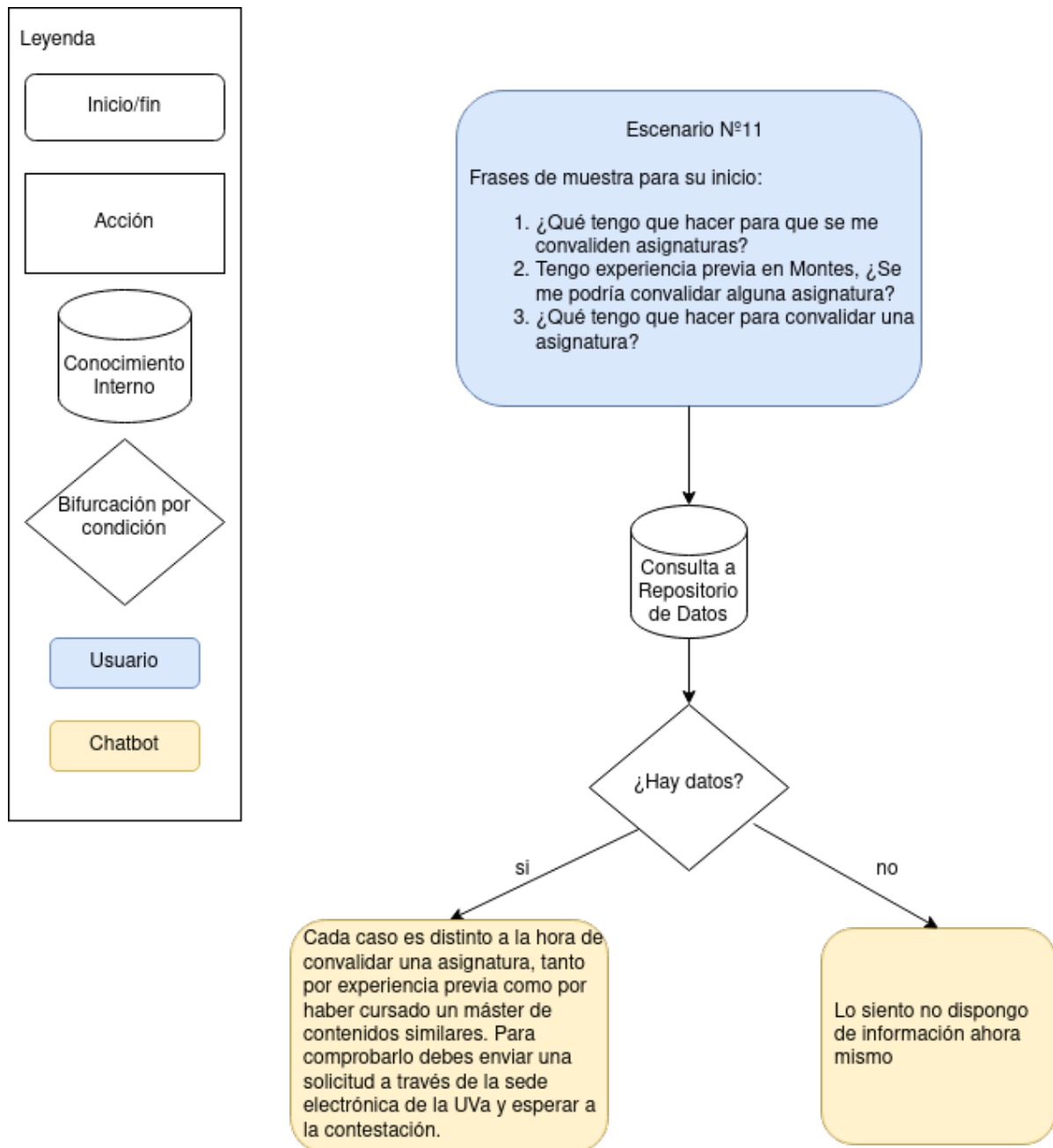


Figura 5.11: Diagrama de Flujo Conversacional 11: Solicitar reconocimiento de asignaturas

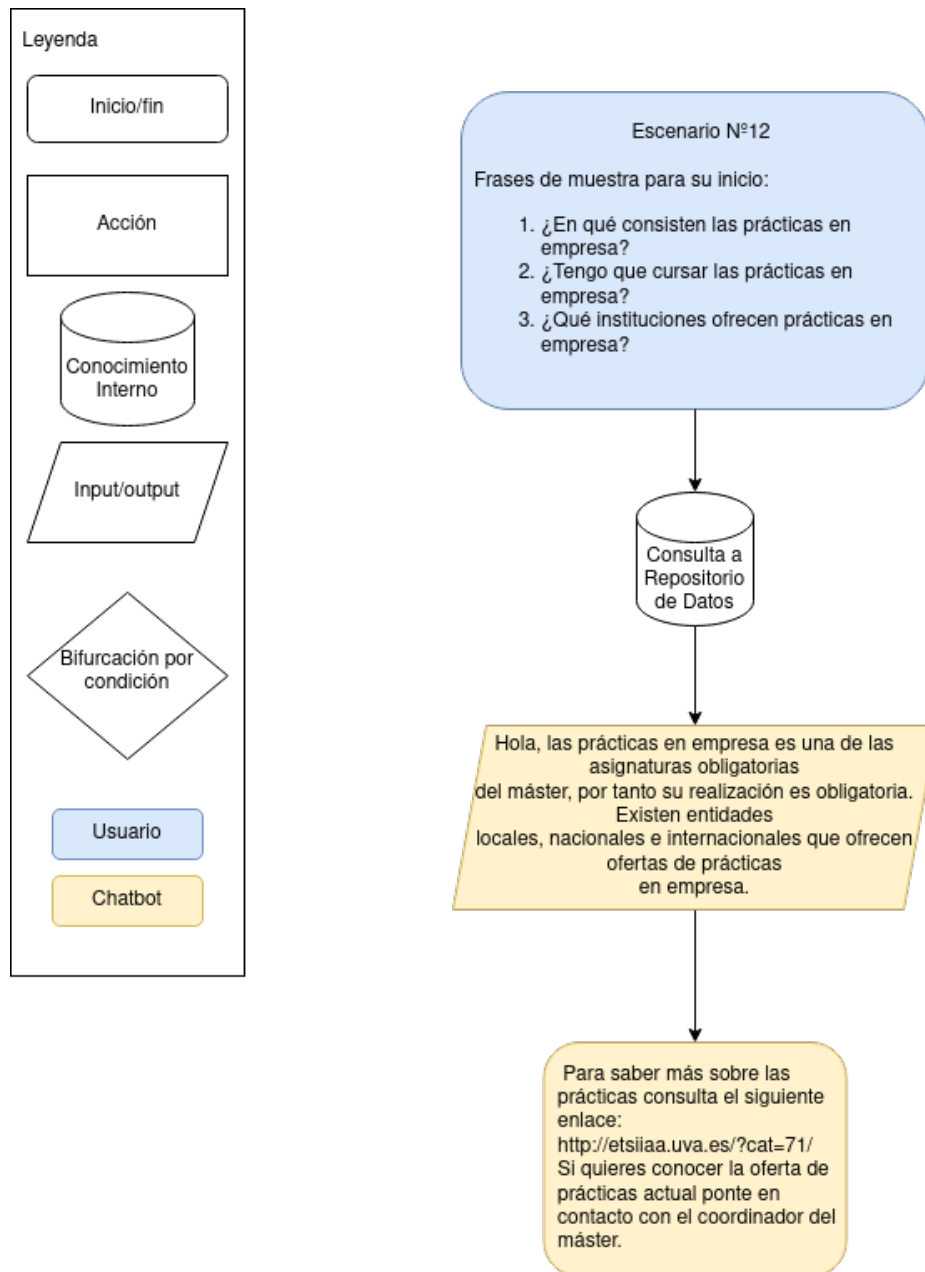


Figura 5.12: Diagrama de Flujo Conversacional 12: Solicitar información de prácticas en empresa

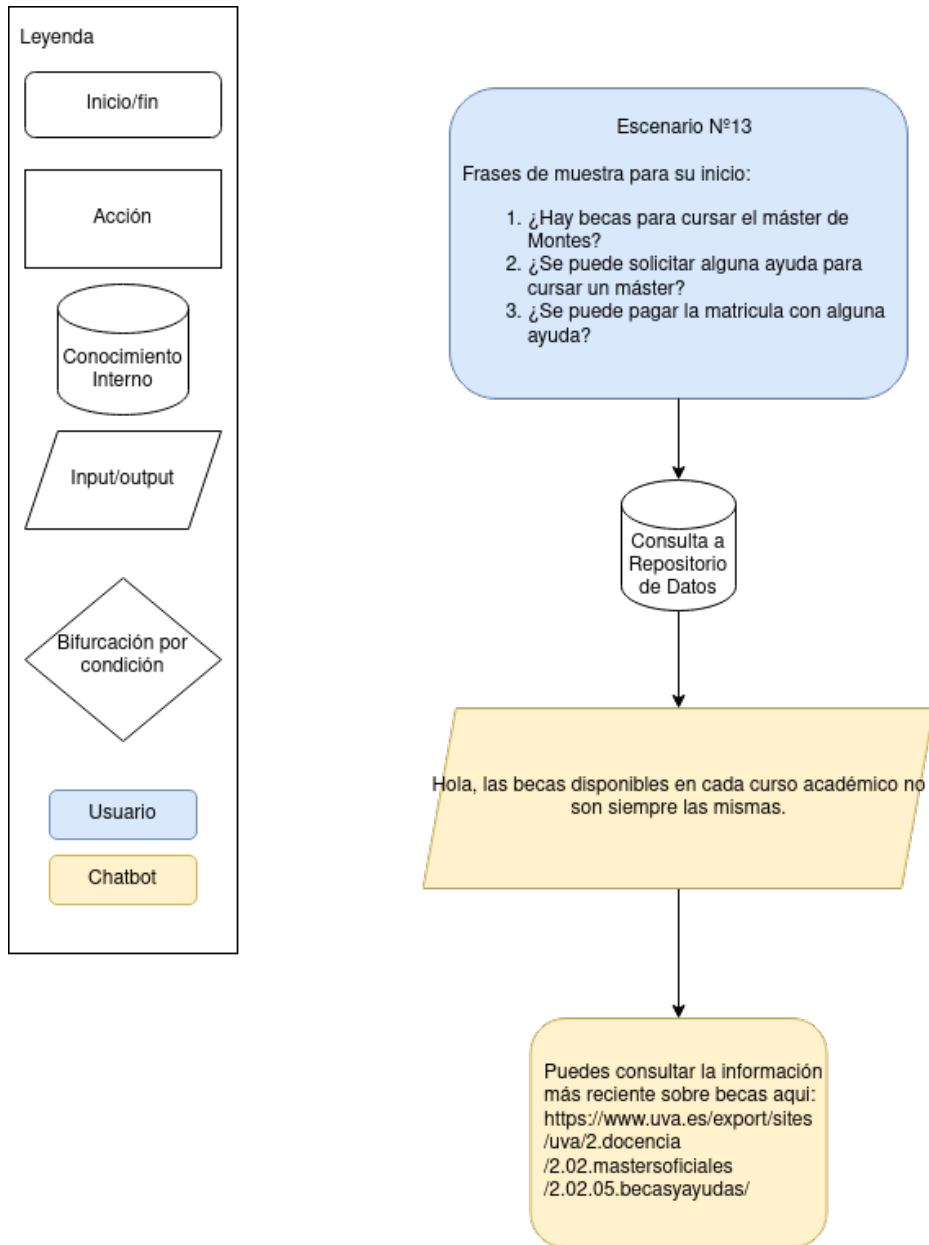


Figura 5.13: Diagrama de Flujo Conversacional 13: Solicitar información de becas para un máster

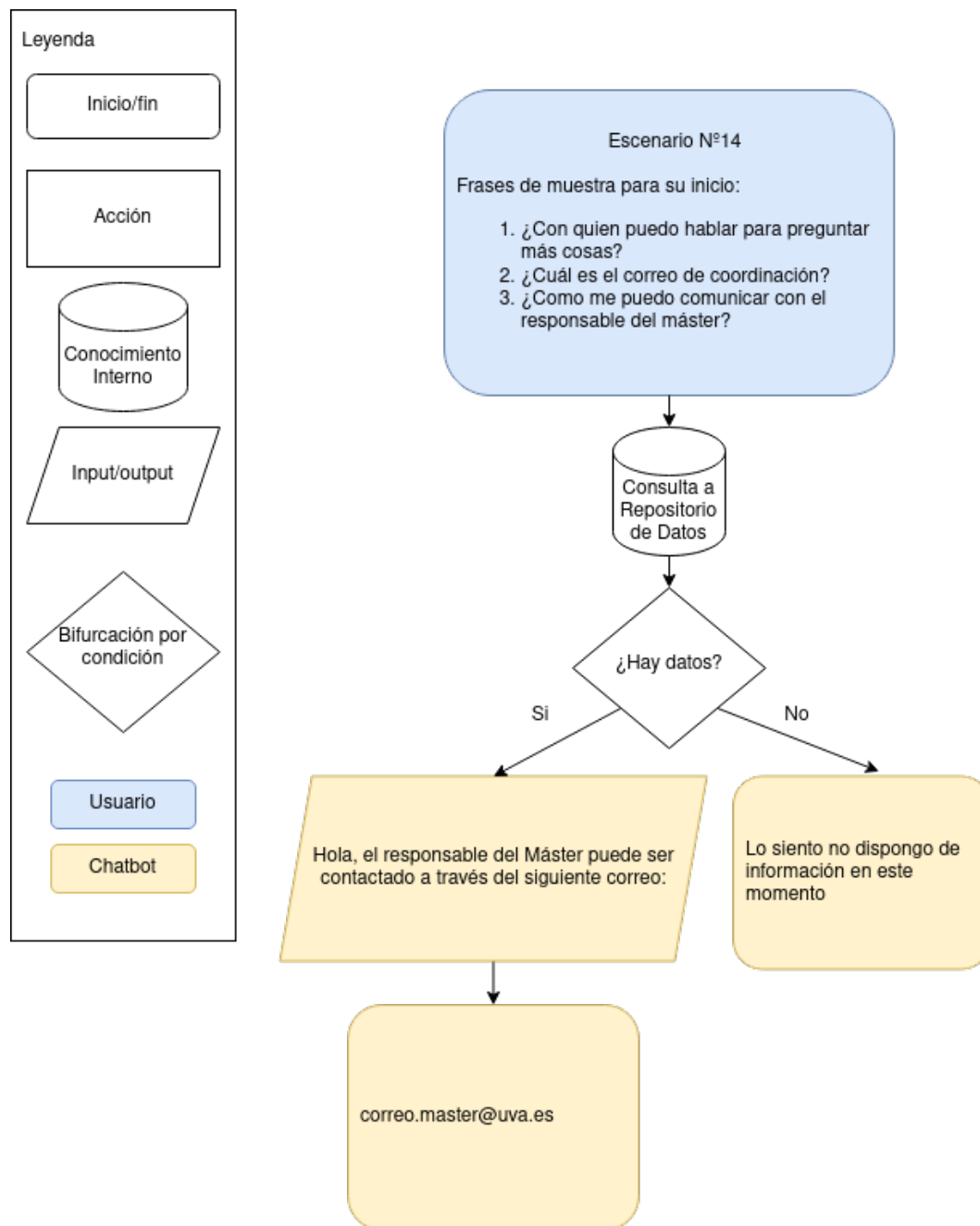


Figura 5.14: Diagrama de Flujo Conversacional 14: Solicitar información de contacto para un máster

5.4. Análisis del sistema

5.4.1. Requisitos funcionales del sistema

FRQ-001	Consultar oferta de másteres
Versión	1.0
Descripción	El sistema deberá permitir al usuario consultar la oferta de másteres disponibles.

FRQ-002	Consultar condiciones de acceso a máster
Versión	1.0
Descripción	El sistema deberá permitir al usuario verificar que cumple las condiciones de acceso para un máster de su elección entre los ofertados, informándole en caso de cumplirlas de las colaboraciones relevantes a ese máster.

FRQ-003	Solicitar información sobre créditos ECTS
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar una descripción sobre créditos ECTS.

FRQ-004	Solicitar resumen del máster
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar una descripción resumida del contenido del máster de su elección.

FRQ-005	Solicitar cronograma de máster
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre el cronograma del máster en que está interesado.

FRQ-006	Solicitar información nivel de idioma
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar los idiomas en que se cursa un máster de su elección incluyendo el nivel de idioma mínimo y recomendado a dominar.

FRQ-007	Solicitar información sobre lenguajes de programación de un máster
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre los lenguajes de programación recomendados para cursar un máster de su elección en el próximo año..

FRQ-008	Solicitar información sobre presencialidad de un máster
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre la modalidad de presencialidad para cursar un máster de su elección en el próximo año.

FRQ-009	Solicitar información sobre precio aproximado de un máster
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre el precio aproximado para cursar un máster de su elección en el próximo año.

FRQ-010	Solicitar información sobre colaboración
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre las colaboraciones de un máster de su elección de entre los ofertados.

FRQ-011	Solicitar información sobre convalidación de asignaturas
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre el procedimiento de convalidación de asignaturas de un máster de su elección.

FRQ-012	Solicitar información sobre prácticas en empresa
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre la obligatoriedad, internacionalización y oferta relacionada con prácticas en empresa.

FRQ-013	Solicitar información sobre becas
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información sobre becas disponibles para cursar un máster.

FRQ-014	Solicitar información sobre contacto con coordinación
Versión	1.0
Descripción	El sistema deberá permitir al usuario solicitar información de contacto con la coordinación de un máster.

FRQ-015	Número de lenguajes de programación de un máster
Versión	1.0
Descripción	El sistema deberá limitar el número de lenguajes de programación recomendados para cursar un máster a dos por máster.

FRQ-016	Precio de máster
Versión	1.0
Descripción	El sistema deberá mostrar el precio de un máster como una cantidad aproximada redondeada hacia abajo al número entero más próxima y múltiplo de 10.

FRQ-017	Numero idiomas impartición
Versión	1.0
Descripción	El sistema deberá limitar el número de idiomas de impartición de un máster a dos, uno principal en el que se impartirán la mayoría de asignaturas y otro secundario para algunas optativas.

5.4.2. Requisitos No Funcionales

NFR-001	Base de datos
Versión	1.0
Descripción	El sistema utilizará como proveedor de base de datos, uno compatible con el conjunto de bibliotecas de Apache Derby.

NFR-002	Servidor Web
Versión	1.0
Descripción	El sistema utilizará como servidor web, uno basado en Apache Tomcat.

NFR-003	Integración del servicio
Versión	1.0
Descripción	El sistema deberá permitir la integración con la plataforma de mensajería Telegram como interfaz de usuario textual.

NFR-004	Integración del servicio
Versión	1.0
Descripción	El sistema deberá permitir la integración con una página web como interfaz de usuario textual.

NFR-005	Número de transacciones
Versión	1.0
Descripción	El sistema deberá comunicar la información al usuario en el mínimo número de transacciones HTTP posible en cada caso.

NFR-006	Cronograma de máster
Versión	1.0
Descripción	El sistema deberá mostrar el cronograma de un máster como el número de cursos académicos y los meses de inicio y fin de cada curso .

NFR-007	Necesidad de homologación
Versión	1.0
Descripción	El sistema deberá informar a usuarios con graduados expedidos por una entidad universitaria no nacional de la necesidad de homologación del grado que de acceso al programa de máster de su elección previo a su inscripción.

NFR-008	Necesidad de graduado en el acceso a másteres
Versión	1.0
Descripción	El sistema deberá informar a usuarios nacionales de la necesidad de poseer un grado afín al máster de su elección previo a su inscripción.

5.4.3. Requisitos de Información

IRQ-001	Datos asociados a un máster
Versión	1.0
Descripción	La información mínima asociada a un Máster constará de los siguientes campos: Código del plan de máster, Nombre asociado al máster, Descripción del máster, cursos académicos que lo componen, meses de inicio y fin de los cursos que lo componen, precio aproximado, modalidad de presencialidad, idiomas, correo del coordinador, niveles de idiomas de impartición y lenguajes de programación asociados.

IRQ-002	Idiomas de impartición de un máster
Versión	1.0
Descripción	La información mínima referida a un idioma de impartición de un Máster constará de los siguientes campos: Idioma de impartición, nivel mínimo establecido, y nivel recomendado establecido.

IRQ-003	Modalidad de presencialidad de un máster
Versión	1.0
Descripción	Las modalidades de presencialidad para cursar un Máster pueden ser: Presencial, No Presencial, y Semipresencial.

IRQ-004	Colaboraciones de un máster
Versión	1.0
Descripción	La información mínima referida a las colaboraciones de un Máster constará de los siguientes campos: Entidad, Descripción de la colaboración y País de la Entidad.

5.4.4. Actores del sistema

Actor-001	Interesado
Versión	1.0
Descripción	El actor interesado es el actor que sin ser estudiante desea saber más sobre los programas de máster en Ingenierías Agrarias y Forestales de la UVa en Palencia.

5.4.5. Casos de Uso

CU-002	Consultar condiciones de acceso a un máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Verificar condiciones de acceso a un máster</i> , ocurre cuando el actor <i>Interesado</i> quiere verificar que posee los requisitos de acceso a un máster disponible en la ETS de Ingenierías Agrarias de Palencia.
Precondición	No se requiere.
Postcondición	Un mensaje informando de si posee o no las condiciones de acceso es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita consultar las condiciones de acceso a un máster. 2. El sistema solicita el país de la entidad de expedición del título que le da acceso al Interesado. 3. El actor interesado aporta el país. 4. El sistema solicita la confirmación de la posesión de un título válido que permita el acceso al máster. 5. El actor interesado confirma la posesión del título. 6. El sistema informa al actor interesado de que cumple los requisitos de acceso. 7. El sistema recupera las colaboraciones relacionadas con el programa de máster. 8. El sistema informa al actor Interesado de las colaboraciones.
Secuencias Alternativas	<ol style="list-style-type: none"> 4.1. Si la entidad expedidora del título que da acceso al interesado es de nacionalidad extranjera y el programa de máster requiere un título de grado específico, el sistema solicita la confirmación de que el título esté homologado. 4.2. Si la entidad expedidora del título es de nacionalidad española el sistema solicita al interesado la confirmación de que posee el título de graduado habilitante para el máster.
Secuencias de Excepción	3.1, 5.1 Si el interesado solicita cancelar la operación, esta queda sin efecto y el caso de uso termina.

CU-001	Consultar oferta de másteres
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Consultar oferta de másteres</i> , ocurre cuando el actor <i>Interesado</i> quiere consultar la oferta de másteres disponibles en la ETS de Ingenierías Agrarias de Palencia.
Precondición	No se requiere.
Postcondición	Un listado de nombres de programas de másteres es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita consultar la oferta de másteres. 2. El sistema recupera la información de cada máster registrado. 3. El sistema elabora un listado con una entrada para cada máster registrado. 4. El sistema muestra el listado elaborado al interesado.
Secuencias de Excepción	2.1. Si el interesado solicita cancelar la operación, esta queda sin efecto y el caso de uso termina.

CU-003	Solicitar cronograma de un máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar cronograma de un máster</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre la temporización de un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando del cronograma del máster es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre el cronograma de un máster. 2. El sistema recupera la información sobre la temporización del máster en cursos académicos y meses . 3. El sistema presenta el cronograma al actor Interesado.

CU-004	Solicitar información crédito ECTS
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Consultar información del crédito ECTS</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre qué es un crédito ECTS.
Precondición	No se requiere.
Postcondición	Un mensaje informando de en qué consiste un crédito ECTS es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre lo que es un crédito ECTS. 2. El sistema recupera la descripción sobre créditos ECTS. 3. El sistema muestra la descripción del crédito ECTS al actor Interesado.

CU-005	Solicitar idiomas del máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar idiomas del máster</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre el nivel de idiomas distintos al Español en que se imparte un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando del idioma y nivel de idioma del máster es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre el nivel de idioma de un máster. 2. El sistema recupera la información sobre los idiomas de impartición del máster y los niveles recomendados y requeridos . 3. El sistema presenta los idiomas de impartición del máster y por cada idioma sus niveles recomendados y requeridos.

CU-006	Solicitar información lenguajes de programación de un máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información lenguajes de programación de un máster</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre los lenguajes de programación requeridos sobre los que se imparte un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando de los lenguajes de programación del máster es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre los lenguajes de programación necesarios para un máster. 2. El sistema recupera la información sobre los lenguajes de programación asociados a un máster. 3. El sistema presenta al interesado los idiomas de impartición asociados a un máster.

CU-007	Solicitar información presencialidad del máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información presencialidad del máster</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre la presencialidad necesaria para cursar un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando de la presencialidad requerida en el máster es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre la presencialidad para cursar el máster en el próximo año. 2. El sistema recupera la información sobre la presencialidad asociada al máster. 3. El sistema presenta al interesado la presencialidad requerida para cursar el máster .

CU-008	Solicitar información precio del máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información precio del máster</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre el precio aproximado de un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando del precio aproximado del máster es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre el precio de cursar el máster en el próximo año. 2. El sistema recupera la información sobre el precio asociado al máster. 3. El sistema presenta al interesado el precio del máster, informándole de que es aproximado .

CU-009	Solicitar resumen de un máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar resumen de un máster</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre en qué consiste un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando del contenido del máster es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre el contenido de un máster. 2. El sistema recupera la información sobre la descripción del máster . 3. El sistema presenta el resumen elaborado al actor Interesado.

CU-010	Solicitar información colaboración de máster
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información colaboración de máster</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre las colaboraciones asociadas a un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando de las colaboraciones del máster es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre colaboraciones existentes en un máster. 2. El sistema recupera la información sobre colaboraciones de ese máster. 3. El sistema presenta al interesado las colaboraciones del máster .

CU-011	Solicitar información convalidación de asignaturas
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información convalidación de asignaturas</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre el proceso de convalidación de asignaturas asociadas a un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando del proceso de convalidación es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre convalidación de asignaturas de un máster. 2. El sistema recupera la información sobre el proceso de convalidación. 3. El sistema presenta al interesado un resumen del proceso de convalidación .

CU-012	Solicitar información prácticas en empresa
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información prácticas en empresa</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre las prácticas en empresa asociadas a un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando de prácticas en empresa es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre las prácticas en empresa de un máster. 2. El sistema recupera la información sobre prácticas. 3. El sistema presenta al interesado un resumen de las prácticas en empresa.

CU-013	Solicitar información becas
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información becas</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener información sobre las becas para cursar un máster de su elección.
Precondición	No se requiere.
Postcondición	Un mensaje informando de becas es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita información sobre las becas de un máster. 2. El sistema recupera la información sobre becas. 3. El sistema presenta al interesado un resumen de las becas disponibles.

CU-014	Solicitar información de contacto
Actor/Actores	Interesado (Actor-001)
Descripción	El caso de uso <i>Solicitar información de contacto</i> , ocurre cuando el actor <i>Interesado</i> quiere obtener la información de contacto del coordinador de un máster para un trato personal.
Precondición	No se requiere.
Postcondición	Un mensaje con la información de contacto es mostrado.
Secuencia Base	<ol style="list-style-type: none"> 1. El actor Interesado solicita el correo de contacto dle coordinador de un máster. 2. El sistema recupera el correo del coordinador. 3. El sistema presenta al interesado un resumen de información de contacto con el correo del coordinador.

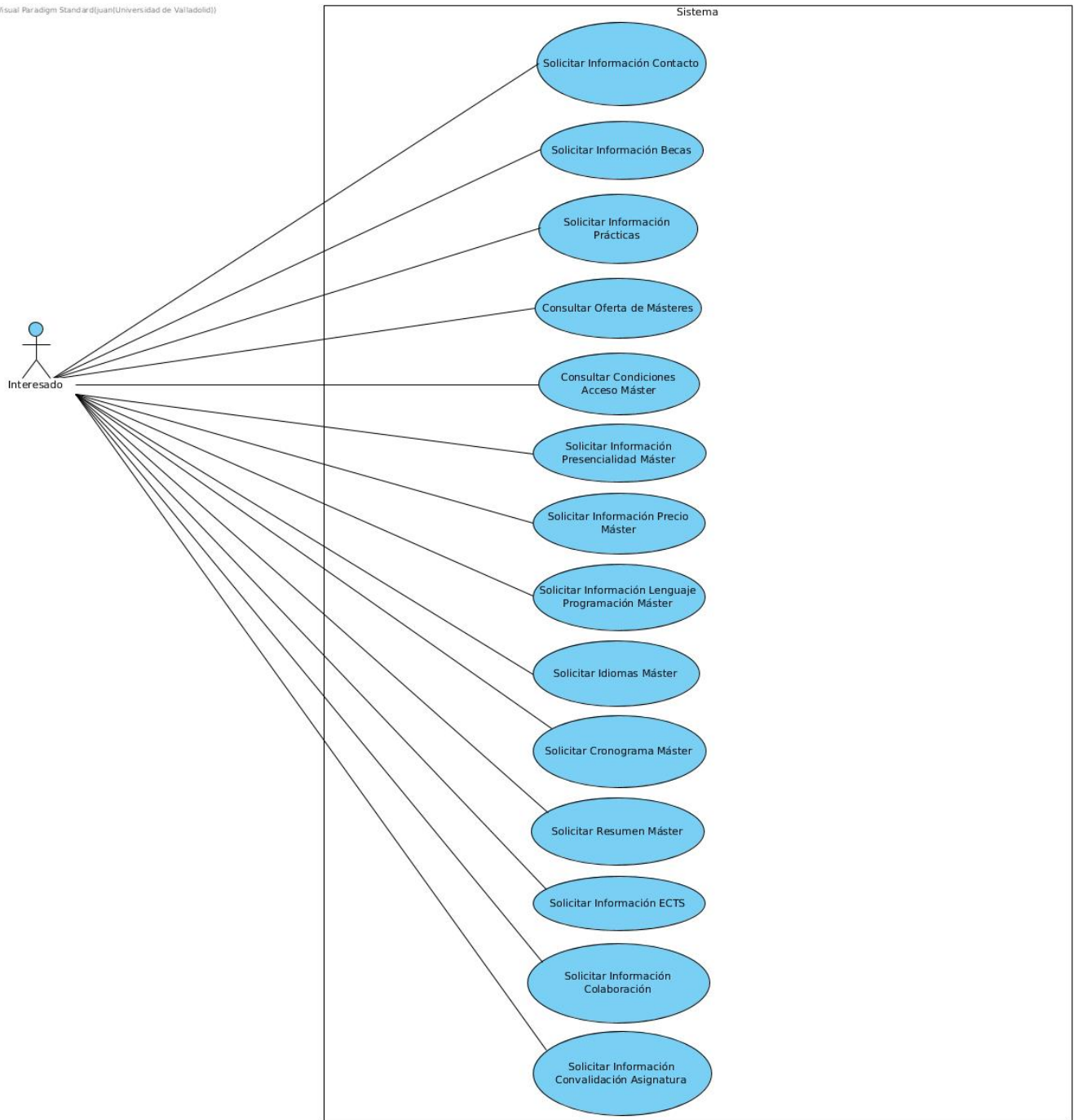


Figura 5.15: Diagrama de casos de uso

5.4.6. Diagrama de clases

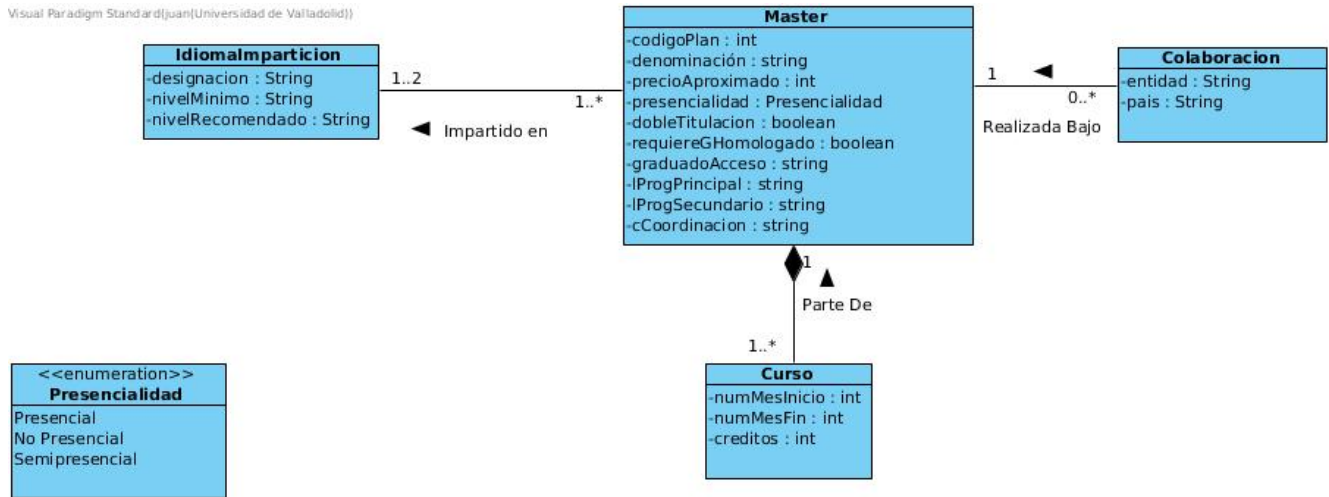


Figura 5.16: Diagrama de clases

5.5. Desarrollo del sistema

El sistema está estructurado, en primer lugar por la plataforma tecnológica escogida para el desarrollo del proyecto que es DialogFlow. En segundo lugar por el programa ngrok, instalado en una máquina virtual proporcionada por la Escuela Técnica Superior de Ingeniería Informática. Este programa se encarga de realizar túneles http para hacer que la dirección local especificada en su inicialización sea expuesta al exterior. En tercer lugar por un servidor web (Apache Tomcat) encargado de recibir y enviar peticiones POST con información en formato JSON a DialogFlow. En esta misma máquina hay un servidor Apache Derby actuando como Base de Datos para el proyecto. La forma actual de interactuar con DialogFlow consiste en utilizar una integración con el cliente de Telegram disponible desde el propio DialogFlow. El motivo de la elección de Telegram sobre otras alternativas (como directamente integrarlo con una plataforma web) se debe a que, debido a realizar las pruebas sobre la versión gratuita de DialogFlow, no es posible llevar a cabo pruebas en un entorno web que requiera el uso de un servidor web de terceros (es decir, en nuestro caso el servidor Tomcat instalado). Por este motivo se utiliza la integración de Telegram además de por ser una de las interfaces de interacción solicitadas en el proyecto.



Figura 5.17: Estructura del sistema

El flujo de información, en el funcionamiento normal del sistema, comienza en Telegram (o directamente en DialogFlow si se usa la interfaz propia de esa plataforma) al solicitarle el usuario información al *chatbot*. El *chatbot* transmite la petición del usuario a DialogFlow mediante el envío de una petición POST con el JSON conteniendo la información. En DialogFlow se evaluará el intent al que corresponde dicha petición de información. Si es un intent con webhook habilitado, el agente que representa al *chatbot* en DialogFlow crea un JSON conteniendo, entre otra información, el nombre asociado al intent con el que se ha identificado la petición del usuario y, como parámetros en el JSON, los valores de las entidades identificadas en la petición. Esta petición se envía mediante un método POST a la URL especificada en el apartado Fulfillment del propio agente creado. Esta URL es la proporcionada por ngrok al activarse en la máquina virtual de la Escuela, ngrok redirige el tráfico Https de esa dirección a la dirección localhost:8080 de la máquina virtual de la Escuela. Es en esta dirección donde está el servidor Web desplegado en Tomcat escuchando peticiones GET y POST. Al llegar el POST con la petición en formato JSON, se procesa y parsea realizándose peticiones relevantes a la BD de acuerdo con la lógica del caso de uso involucrado. Con la respuesta obtenida de la BD se introduce esta en un JSON de respuesta y se contesta a DialogFlow. Al recibir este el JSON con la estructura esperada, envía la respuesta (en este caso Telegram aunque como se ha explicado podría ser otra integración o el diálogo integrado directamente en DialogFlow) al cliente que realizó la pregunta inicial.

5.6. Pruebas del sistema

En esta sección se describen los métodos y pruebas utilizados para comprobar la corrección del sistema desarrollado.

5.6.1. Pruebas de Caja Blanca

Las pruebas de caja blanca son aquellas pruebas realizadas a la vista del código de la aplicación. Para estas pruebas se han utilizado las herramientas de testeo JUnit, JaCoCo y EasyMock. Estas pruebas se han ido realizando al final de cada iteración después de el desarrollo del prototipo en un sistema funcional, para comprobar no solo la funcionalidad de cada uno de los métodos de cada caso de uso sino la funcionalidad conjunta del sistema, tal y como se describe en la metodología explicada anteriormente. Dado que estas pruebas están relacionadas con el código, no se incluyen en esta memoria. Sin embargo, dado que se ha usado las herramientas SonarLint y SonarQube como medidor de la calidad del código desarrollado, se incluye a continuación el análisis realizado del código final desarrollado en el que se explica, entre otras cosas, la cobertura del código de lógica y modelo de la aplicación.

5.6.2. Pruebas de Caja Negra

Las pruebas de caja negra se hacen sin ser visible el código de la aplicación, únicamente teniendo en cuenta la especificación acordada de la aplicación. Por ello se prueban los casos extremos de acuerdo a la especificación de cada funcionalidad y se contrasta el funcionamiento obtenido con el esperado como parte del proceso de prueba. se muestran a continuación las condiciones de realización de cada prueba, los resultados esperados y obtenidos (salvo que la respuesta sea excesivamente larga, caso en el que se resume entre paréntesis) para esas condiciones.

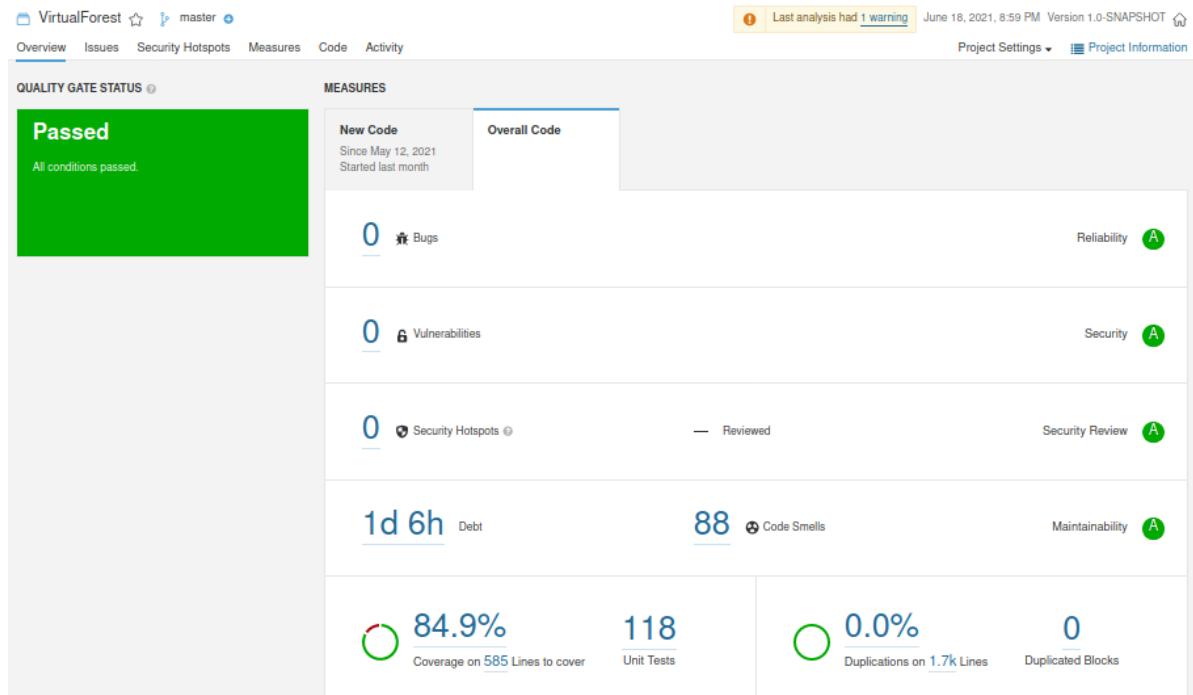


Figura 5.18: Informe Resumen de SonarQube

Prueba (P_ CU01_ 01)	
Descripción	Prueba para el CU 01 (Consultar oferta de másteres).
Entrada	¿Qué másteres hay disponibles?.
Salida Esperada	Un listado de másteres ofertados se muestra.
Salida Obtenida	Un listado de másteres ofertados se muestra (Resultado Correcto).

Prueba (P_ CU01_ 02)	
Descripción	Prueba para el CU 01 (Consultar oferta de másteres). Se prueba a interactuar con el sistema sin conexión a la BD
Entrada	¿Qué másteres hay disponibles?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento (Resultado Correcto).

Prueba (P_ CU02_ 01)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que requiere título de grado específico, nacionalidad española y posesión de título.
Entrada	¿Cómo puedo acceder al máster de Montes?, España, si.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/ .
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/ (Resultado Correcto).

Prueba (P_ CU02_ 02)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que requiere título de grado específico, nacionalidad española y no se posee título.
Entrada	¿Cómo puedo acceder al máster de Montes?, España, no.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso.
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso (Resultado Correcto).

Prueba (P_ CU02_ 03)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que requiere título de grado específico, nacionalidad extranjera y posesión de título homologado.
Entrada	¿Cómo puedo acceder al máster de Montes?, Vietnam, si.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/ Además, el programa posee las siguientes colaboraciones: Vietnam National University Of Science AgroParisTech.
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/ Además, el programa posee las siguientes colaboraciones: Vietnam National University Of Science AgroParisTech (Resultado Correcto).

Prueba (P_ CU02_ 04)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que requiere título de grado específico, nacionalidad extranjera y no se posee título homologado.
Entrada	¿Cómo puedo acceder al máster de Montes?, Vietnam, no.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso.
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Ingeniería de Montes. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso. (Resultado Correcto).

Prueba (P_ CU02_ 05)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que no requiere título de grado específico, nacionalidad española y posesión de título de grado.
Entrada	¿Cómo puedo acceder al máster de Dataforest?, España, si.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/ (Resultado Correcto).

Prueba (P_ CU02_ 06)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que no requiere título de grado específico, nacionalidad española y no se posee título de grado.
Entrada	¿Cómo puedo acceder al máster de Dataforest?, España, no.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso.
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso. (Resultado Correcto).

Prueba (P_ CU02_ 07)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que no requiere título de grado específico, nacionalidad extranjera y posesión de título de grado.
Entrada	¿Cómo puedo acceder al máster de Dataforest?, Canada, si.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/ Además, el programa posee las siguientes colaboraciones: Vietnam National University Of Science AgroParisTech
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que posees un título que permite el acceso al programa de máster es posible tu acceso. https://apps.stic.uva.es/preinsmaster/ Además, el programa posee las siguientes colaboraciones: Vietnam National University Of Science AgroParisTech (Resultado Correcto).

Prueba (P_ CU02_ 08)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso de máster que no requiere título de grado específico, nacionalidad extranjera y no se posee título de grado.
Entrada	¿Cómo puedo acceder al máster de Dataforest?, Canada, no.
Salida Esperada	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso.
Salida Obtenida	En primer lugar, gracias por tu interés en el Máster en Dataforest. Dado que no posees un título válido que permite el acceso al programa de máster no es posible tu acceso. (Resultado Correcto).

Prueba (P_ CU02_ 09)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema para el caso en que no se especifica un máster.
Entrada	¿Cómo puedo acceder al máster?.
Salida Esperada	Hola, ¿a qué máster querria acceder?
Salida Obtenida	Hola, ¿a qué máster querria acceder? (Resultado Correcto).

Prueba (P_ CU02_ 10)	
Descripción	Prueba para el CU 02 (Consultar condiciones de acceso a un máster). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Cómo puedo acceder al máster?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU03_ 01)	
Descripción	Prueba para el CU 03 (Solicitar Información Crédito ECTS).
Entrada	¿Qué es un crédito ECTS?.
Salida Esperada	(Explicación del crédito ECTS).
Salida Obtenida	(Explicación del crédito ECTS). (Resultado Correcto).

Prueba (P_ CU03_ 02)	
Descripción	Prueba para el CU 03 (Solicitar Información Crédito ECTS). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Qué es un crédito ECTS?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU04_ 01)	
Descripción	Prueba para el CU 04 (Solicitar Resumen de un Máster).
Entrada	Disponibilidad del máster de Montes.
Salida Esperada	(Resumen del master).
Salida Obtenida	(Resumen del master). (Resultado Correcto).

Prueba (P_ CU04_ 02)	
Descripción	Prueba para el CU 04 (Solicitar Resumen de un Máster). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	Disponibilidad del máster de Montes.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU04_ 03)	
Descripción	Prueba para el CU 04 (Solicitar Resumen de un Máster). Se prueba a interactuar con el sistema sin especificar el master.
Entrada	Disponibilidad del máster.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

Prueba (P_ CU05_ 01)	
Descripción	Prueba para el CU 05 (Solicitar Cronograma de un Máster).
Entrada	Duración del máster de montes.
Salida Esperada	Hola, el Máster en Ingeniería de Montes tiene 2 cursos académicos con la siguiente organización: El 1º curso tiene una duración de 9 meses, empezando en septiembre y acabando en junio. El contenido en créditos es de 60 El 2º curso tiene una duración de 5 meses, empezando en septiembre y acabando en febrero. El contenido en créditos es de 30
Salida Obtenida	Hola, el Máster en Ingeniería de Montes tiene 2 cursos académicos con la siguiente organización: El 1º curso tiene una duración de 9 meses, empezando en septiembre y acabando en junio. El contenido en créditos es de 60 El 2º curso tiene una duración de 5 meses, empezando en septiembre y acabando en febrero. El contenido en créditos es de 30 (Resultado Correcto).

Prueba (P_ CU05_ 02)	
Descripción	Prueba para el CU 05 (Solicitar Cronograma de un Máster). Se prueba a interactuar con el sistema sin especificar el master.
Entrada	Duración del máster.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

Prueba (P_ CU05_ 03)	
Descripción	Prueba para el CU 05 (Solicitar Cronograma de un Máster). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	Duración del máster.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU06_ 01)	
Descripción	Prueba para el CU 06 (Solicitar idiomas de un Máster).
Entrada	¿Qué nivel de idiomas es necesario para cursar el máster de Dataforest?..
Salida Esperada	Estos son los idiomas, aparte del español, en que se imparte el máster y sus niveles mínimos y recomendados: Idioma requerido Inglés (Alto), con nivel minimo B1 pero de nivel recomendado B2.
Salida Obtenida	Estos son los idiomas, aparte del español, en que se imparte el máster y sus niveles mínimos y recomendados: Idioma requerido Inglés (Alto), con nivel minimo B1 pero de nivel recomendado B2. (Resultado Correcto).

Prueba (P_ CU06_ 02)	
Descripción	Prueba para el CU 06 (Solicitar idiomas de un Máster). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Qué nivel de idiomas es necesario para cursar el máster de Dataforest?..
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU06_ 03)	
Descripción	Prueba para el CU 06 (Solicitar idiomas de un Máster). Se prueba a interactuar con el sistema sin especificar el master.
Entrada	¿Qué nivel de idiomas es necesario para cursar el máster?.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

Prueba (P_ CU07_ 01)	
Descripción	Prueba para el CU 07 (Solicitar información lenguaje de programación de un Máster). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Qué lenguajes de programación son necesarios para cursar el máster de Dataforest?.
Salida Esperada	Hola, el Máster en Dataforest no requiere conocer ningún lenguaje de programación previo, ya que en el se enseñan los siguientes lenguajes de programación: Python, R, MySQL, SAS.
Salida Obtenida	Hola, el Máster en Dataforest no requiere conocer ningún lenguaje de programación previo, ya que en el se enseñan los siguientes lenguajes de programación: Python, R, MySQL, SAS. (Resultado Correcto).

Prueba (P_ CU07_ 02)	
Descripción	Prueba para el CU 07 (Solicitar información lenguaje de programación de un Máster). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Qué lenguajes de programación son necesarios para cursar el máster de Dataforest?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU07_ 03)	
Descripción	Prueba para el CU 07 (Solicitar información lenguaje de programación de un Máster). Se prueba a interactuar con el sistema sin especificar el master.
Entrada	¿Qué lenguajes de programación son necesarios para cursar el máster?.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

Prueba (P_ CU08_ 01)	
Descripción	Prueba para el CU 08 (Solicitar información presencialidad de un Máster).
Entrada	¿Se puede cursar de forma no presencial el máster de Montes?.
Salida Esperada	Hola, el Máster en Ingeniería de Montes tiene una modalidad de impartición Presencial. Si en tu caso no pudieses seguir esta modalidad de impartición, tendrías que contactar con el coordinador del master para buscar una solución adecuada.
Salida Obtenida	Hola, el Máster en Ingeniería de Montes tiene una modalidad de impartición Presencial. Si en tu caso no pudieses seguir esta modalidad de impartición, tendrías que contactar con el coordinador del master para buscar una solución adecuada. (Resultado Correcto).

Prueba (P_ CU08_ 02)	
Descripción	Prueba para el CU 08 (Solicitar información presencialidad de un Máster). Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Qué lenguajes de programación son necesarios para cursar el máster de Dataforest?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU08_ 03)	
Descripción	Prueba para el CU 08 (Solicitar información presencialidad de un Máster). Se prueba a interactuar con el sistema sin especificar el master.
Entrada	¿Qué lenguajes de programación son necesarios para cursar el máster?.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

Prueba (P_ CU09_ 01)	
Descripción	Prueba para el CU 09 (Solicitar información precio de un Máster).
Entrada	¿Qué precio tiene cursar el máster de Montes?.
Salida Esperada	(Explicacion precio del máster de Montes).
Salida Obtenida	(Explicacion precio del máster de Montes). (Resultado Correcto).

Prueba (P_ CU09_ 02)	
Descripción	Prueba para el CU 09 (Solicitar información precio de un Máster).Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Qué precio tiene cursar el máster de Montes?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU09_ 03)	
Descripción	Prueba para el CU 09 (Solicitar información precio de un Máster). Se prueba a interactuar con el sistema sin especificar el master.
Entrada	¿Qué precio tiene cursar el máster.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

Prueba (P_ CU10_ 01)	
Descripción	Prueba para el CU 10 (Solicitar información colaboración de un Máster).
Entrada	¿Qué colaboraciones tiene el máster de Montes?.
Salida Esperada	(Explicacion colaboraciones del master de Montes).
Salida Obtenida	(Explicacion colaboraciones del master de Montes). (Resultado Correcto).

Prueba (P_ CU10_ 02)	
Descripción	Prueba para el CU 10 (Solicitar información colaboración de un Máster).Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Qué colaboraciones tiene el máster de Montes?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU10_ 03)	
Descripción	Prueba para el CU 10 (Solicitar información colaboración de un Máster). Se prueba a interactuar con el sistema sin especificar el master.
Entrada	¿Qué colaboraciones tiene el máster de Montes?.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

Prueba (P_ CU11_ 01)	
Descripción	Prueba para el CU 11 (Solicitar información convalidación asignatura).
Entrada	¿Cómo se puede convalidar una asignatura?.
Salida Esperada	(Explicacion proceso de convalidación).
Salida Obtenida	(Explicacion proceso de convalidación). (Resultado Correcto).

Prueba (P_ CU11_ 02)	
Descripción	Prueba para el CU 11 (Solicitar información convalidación asignatura).Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Cómo se puede convalidar una asignatura?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU12_ 01)	
Descripción	Prueba para el CU 12 (Solicitar información prácticas en empresa).
Entrada	¿En qué consisten las prácticas en empresa?.
Salida Esperada	(Explicacion de prácticas en empresa).
Salida Obtenida	(Explicacion de prácticas en empresa). (Resultado Correcto).

Prueba (P_ CU13_ 01)	
Descripción	Prueba para el CU 13 (Solicitar información becas).
Entrada	¿Hay becas disponibles para cursar los másteres?.
Salida Esperada	(Explicacion de becas).
Salida Obtenida	(Explicacion de becas). (Resultado Correcto).

Prueba (P_ CU14_ 01)	
Descripción	Prueba para el CU 14 (Solicitar información de contacto).
Entrada	¿Cómo puedo contactar con el coordinador del máster de Montes?.
Salida Esperada	(Explicacion de información de contacto).
Salida Obtenida	(Explicacion de información de contacto). (Resultado Correcto).

Prueba (P_ CU14_ 02)	
Descripción	Prueba para el CU 14 (Solicitar información de contacto).Se prueba a interactuar con el sistema sin conexión con el servidor de BD.
Entrada	¿Cómo puedo contactar con el coordinador del máster de Montes?.
Salida Esperada	El servicio no está disponible en este momento.
Salida Obtenida	El servicio no está disponible en este momento. (Resultado Correcto).

Prueba (P_ CU14_ 03)	
Descripción	Prueba para el CU 14 (Solicitar información de contacto).Se prueba a interactuar con el sistema sin especificar un máster.
Entrada	¿Cómo puedo contactar con el coordinador del máster de Montes?.
Salida Esperada	Perdona, ¿de qué máster estamos hablando?.
Salida Obtenida	Perdona, ¿de qué máster estamos hablando?. (Resultado Correcto).

5.7. Evaluación de la propuesta

Una vez realizada la aplicación de la propuesta para el desarrollo de un *chatbot*, llega la hora de llevar a cabo la evaluación de la misma con los criterios de validación definidos en la sección 4.3 de este documento. Para realizar la evaluación se explicará lo que ha funcionado correctamente y lo que no ha funcionado de acuerdo a lo planteado. A continuación se valora primero por técnicas y luego en general el resultado de aplicar la metodología.

5.7.1. Evaluación de las técnicas de elicitación de requisitos utilizadas

User stories En cuanto a *user stories*, dado que la forma de elaborarlas ha consistido en mantener una sola story por funcionalidad solicitada, y que las funcionalidades solicitadas no se solapaban entre ellas, se ha cumplido el criterio de independencia para las *user stories* en todo momento. Al no disponerse de mucho

tiempo de atención por parte de los clientes, en ciertos momentos se ha tenido que realizar suposiciones sobre la funcionalidad a incorporar en la siguiente iteración. Se ha incumplido por tanto en esas ocasiones el principio de negociación, teniendo que recurrirse a la confirmación en la reunión de test de usuarios de la siguiente iteración si la funcionalidad se correspondía con lo solicitado o no. En este último caso se ha tenido que reescribir la *user story* para incluir esta confirmación como proceso de negociación. Todas las *user stories* están escritas desde un punto de vista en el que se esté aportando valor al usuario, aunque también se han tenido que tomar en consideración reglas de negocio que no fuesen aconsejables ignorar. Un ejemplo de esto es la *user story* de consulta de precio de un máster, donde se ofrece una estimación aproximada del precio del máster en lugar del precio exacto que supondría cursarlo. Solo ha habido una *user story* que debido a su complejidad se ha considerado reducir para poder realizarla bajo una sola iteración, la *user story* número 2. Todas las *user stories* han podido priorizarse para incluirse, en función de su carga de trabajo, en una iteración con *user stories* iguales o más sencillas para poder realizarse en ese periodo de tiempo, cumpliendo así con el principio de (S) pequeñas. Se ha aplicado en la mayor parte de las veces y de forma preventiva la acción de corrección del principio de testeable de las *user stories*, ganando en cada iteración una forma de verificar la completitud de la story. Solo se ha necesitado modificar el test de completitud en cuatro ocasiones para reflejar realmente que la realización de la *user story* es acorde con lo especificado por el cliente.

Principio Incumplido	Users Stories con Incumplimiento
I (Independiente)	Ninguna.
N (Negociable)	3, 5, 6, 8, 11.
V (Valor)	Ninguna.
E (Estimable)	2.
S (Pequeña)	Ninguna.
T (Testeable)	1, 3, 5, 6, 9.

Tabla 5.1: Tabla de evaluación de user stories

Como se aprecia por la tabla la aplicación de las *user stories* tiene resultados positivos en términos de independencia, valor para el usuario y tamaño semejante, pero ha tenido problemas de negociabilidad, estimación y testing de completitud. Estos problemas vienen causados por poca comunicación con el cliente. Por tanto, para poder aplicar de forma eficiente las *user stories*, sería recomendable llevar a cabo una reunión inicial en cada iteración, diferente de la del test de usuarios, donde se especifiquen por parte del cliente las funcionalidades nuevas que requiera incluir en el prototipado de la iteración por empezar. En esta reunión no se tendría que prestar atención a modificaciones a realizar detectadas en el test de usuarios, si no a obtener especificaciones más concretas de la funcionalidad que debe cubrir la story .

Escenarios Dado que como fase previa al diseño de escenarios, se formaron las *user stories*, esto tiene como consecuencia que al no haber dependencias entre las *user stories*, tampoco hubiese escenarios que estuviesen referidos a múltiples objetivos, no necesitando dividirlos en varios escenarios de un solo objetivo. Por la misma razón que la característica anterior, al definirse la *user story* con la plantilla: *Como quiero... para....*, es posible determinar en cada escenario el contexto de realización, por parte del actor del mismo, de forma inmediata. Debido a que algunas *user stories* no se negociaron adecuadamente en su forma inicial con el cliente, los escenarios derivados de éstas no identificaban correctamente los recursos de información necesarios, teniendo que reescribirse en esas circunstancias. No ha sido necesario buscar identificaciones de actores adicionales ya que el *chatbot* está planteado para un solo tipo de actor, estudiantes de grado que quieren comenzar un máster del ámbito forestal en el próximo año. De nuevo, por la propia definición de la *user story* todos los escenarios cubren al menos el caso de éxito que representa el objetivo especificado como *user story*. Un criterio incumplido en la verificación de los escenarios es la completitud a la hora de describir todos los caminos alternativos. En una ocasión se ha tenido que modificar los episodios del escenario para

reflejar caminos alternativos no tenidos en cuenta inicialmente. Se ha tenido que replantear además en qué puntos es posible que el actor abandone la conversación con el *chatbot*.

Pregunta de verificación de escenario	Escenarios con Incumplimiento
P1 (Escenarios con múltiples objetivos)	Ninguna.
P2 (Contexto del escenario no definido)	Ninguna.
P3 (Recursos de información no válidos)	1, 3, 5, 6, 8.
P4 (Actores no identificados)	Ninguna.
P5 (Caso de éxito no cubierto)	Ninguna.
P6 (Escenarios alternativos cubiertos)	2.

Tabla 5.2: Tabla de evaluación de escenarios

Por los resultados de la tabla se puede ver que la aplicación de los escenarios ha sido mayoritariamente positiva. Solo se han localizado problemas en términos de recursos de información y de escenarios cubiertos. El problema de los recursos de información viene heredado de las *user stories*, al no tener éstas una especificación más completa que determinase de forma correcta la información necesaria para llevar a cabo el escenario. El problema de los escenarios alternativos cubiertos solo se ha dado en un solo escenario, en el que hubo un problema de comprensión de uno de los requisitos para llevar a cabo el escenario, dando como resultado que se tuviese que reorientar el conjunto de episodios a seguir para llegar a la meta. Esto provocó nuevos escenarios alternativos. Dado que el problema es de comprensión de la información, se considera que es derivado de las *user stories*. Convendría por tanto involucrar al cliente en este punto para, mostrándole una versión no técnica del escenario, confirmar que está de acuerdo con lo requerido.

Prototipado En la mayoría de funcionalidades prototipadas el cliente se ha mostrado satisfecho con la lógica y el funcionamiento de las mismas, indicando sobre todo correcciones relacionadas con la forma de expresar una información o con la información en sí misma y no con el funcionamiento del sistema prototipado. Se indica aquí por tanto las funcionalidades que no han coincidido con lo que el cliente tenía en mente al realizar la propuesta de funcionalidad. Estas funcionalidades se referencian sobre la numeración utilizada para los escenarios y las *user stories* hasta ahora por corresponderse con estas.

Funcionalidad	Explicación
Nº2 (Consultar condiciones de acceso a máster)	Se malinterpretó en la explicación de la funcionalidad por parte del cliente el concepto de nacionalidad, entendido como la del usuario. En realidad es la nacionalidad de la entidad expedidora del título que da acceso al máster.
Nº8 (Solicitar información del precio del máster)	El precio no debe ser calculado, debe ser recuperado de almacenamiento estando este precio aproximado al valor real del máster.
Nº9 (Solicitar resumen de un máster)	No se requiere tanto un resumen del contenido del máster como una indicación de que es posible cursarlo y un enlace a la página del máster.

Tabla 5.3: Tabla de evaluación de funcionalidades prototipadas

La realización de tests de usuario, tras el prototipado, fue un elemento clave para involucrar a los usuarios, que no respondían de otras maneras a nuestra llamada a colaborar. A raíz de los tests de usuario, se consiguió que proporcionaran datos y recursos importantes para continuar con el proceso además de los comentarios de correcciones y mejoras detectados. En esta etapa, los errores surgidos, más que de la propia etapa son, de nuevo, heredados de las etapas anteriores que aportan la información necesaria para la funcionalidad que

se requiere prototipar. La forma de evitar que los errores se propaguen hasta el prototipado sería realizar una sesión, compartida con el final de la fase anterior de diseño de escenarios, para evitar que estos errores de falta de comprensión se extiendan hasta el prototipo y se puedan corregir como muy tarde en la fase de escenarios.

Laddering Como se ha explicado, la técnica del laddering solo se ha aplicado en ocasiones muy concretas para determinar el valor que realmente perseguía el cliente con una funcionalidad que no ha sido adecuada. Estas ocasiones coinciden con las funcionalidades cuyo prototipo no se correspondía con lo exigido. Al igual que en la subsección anterior, estas aplicaciones del laddering se referencian sobre la numeración utilizada para los escenarios y las user stories hasta ahora.

Funcionalidad	Resultados de aplicación de laddering
Nº2 (Consultar condiciones de acceso a master)	Característica - Consecuencia: Nacionalidad asociada incorrectamente. Consecuencia - Causa: Necesidad de adquirir la nacionalidad de la entidad expedidora de grado para sabersi cumple con Bolonia. Causa -Valor: Necesidad de Homologación si no cumple con Bolonia.
Nº8 (Solicitar información del precio del máster)	Característica - Consecuencia: Cifra de precio calculado incorrecta. Consecuencia - Causa: Precio variable entre años pese a mantener número de créditos. Causa -Valor: Precio fijado por terceros, necesidad de indicar una cifra aproximada almacenada, no calculada.
Nº9 (Solicitar resumen de un máster)	Característica - Consecuencia: Resumen inadecuado para reflejar contenido del máster. Consecuencia - Causa: Contenido dependiente de cambio de planes. Causa -Valor: Mostrar contenido solo como referencia a la página de la UVa donde se cambiará.

Tabla 5.4: Tabla de evaluación de aplicación de laddering

A la vista de la tabla, se puede ver que en los casos que se ha aplicado el laddering se ha conseguido obtener el valor asociado a la funcionalidad que perseguía realmente el cliente para la interacción del usuario con el *chatbot*. Se ha podido identificar desde la conexión característica - consecuencia hasta la causa - valor, por lo que se considera satisfactoria la aplicación del laddering y los resultados obtenidos por esta técnica. De nuevo, sería recomendable no permitir que los problemas identificados se propaguen hasta la fase de test de usuario donde se aplica esta técnica para no tener que necesitar iteraciones a mayores donde aplicar las correcciones.

5.7.2. Evaluación general de la aplicación de la propuesta

Se contestan ahora a las preguntas de evaluación general de la metodología, alineadas con los objetivos de la propuesta, para poder valorar el alcance de esta.

- *P1: ¿Qué número de participantes hemos podido involucrar?*

El número de participantes ha sido en todas las iteraciones de tres personas. Se ha considerado incluir un número de participantes mayor, pero dadas las características del proyecto no es posible conseguir acceso a una población de tamaño más elevado. Se considera importante que se disponga de una población mayor para poder identificar y capturar las necesidades más relevantes de los usuarios y las formas más comunes de su interacción con el sistema.

- *P2: ¿Qué características poseen estos participantes?*

Los participantes consisten en dos profesores de la ETSIIAA perteneciente a la Universidad de Valladolid y un alumno de la misma escuela que ha finalizado sus estudios en la misma y colabora como técnico en el IUFOR. Los profesores pertenecen también al IUFOR por lo que poseen los conocimientos suficientes tanto en materia forestal como en entornos educativos para llevar a cabo un proceso de recogida de datos sobre ellos. Estos participantes reúnen las características necesarias de conocimiento del ámbito de ciencias forestales y relación con los programas de grado de este ámbito como para poder participar en el trabajo.

- *P3: ¿Cuál es el tiempo que hemos necesitado de ellos?*

Se han realizado sesiones de treinta minutos por cada iteración para mostrar el prototipo y obtener información tanto de correcciones como de nuevas funcionalidades, adicionalmente se ha necesitado contactar con los participantes mediante correo electrónico para realizar aclaraciones. Dadas las conclusiones obtenidas en la aplicación de las técnicas, comentadas en la sub sección anterior, sería recomendable incorporar una fase inicial de la iteración donde se especifiquen, con detalle, exclusivamente las funcionalidades necesarias a incluir en el prototipado de la iteración. La fase de test de usuario solo se ocuparía de correcciones a la funcionalidad que se ha mostrado en el prototipo y de obtener propuestas para nuevas funcionalidades. Estas propuestas se anotarían para su posterior especificación por parte del cliente en una reunión posterior, al comienzo de la siguiente iteración. Esto otorgaría responsabilidad sobre el desarrollo del prototipo al propio cliente, forzándole a pensar en las funcionalidades a prototipar, ya que las que no recibiesen información suficiente en la reunión de la fase inicial no se podrían llevar a cabo en la iteración a comenzar.

- *P4: ¿Se ha confirmado la mayoría de cosas enseñadas en el prototipo?*

Si, se han realizado aclaraciones e indicado correcciones, pero la funcionalidad se ha confirmado como mayormente correcta. Como se ha indicado en la evaluación de las técnicas, la causa mayoritaria de no confirmar una funcionalidad es no disponer de descripciones suficientes por parte del cliente de funcionalidades proyectadas para la iteración. Esto se debe a que el propio cliente no está del todo seguro sobre qué es lo que quiere para que le resuelva el problema que tiene. Es más fácil, desde su punto de vista, el plantear correcciones o modificaciones sobre un resultado “palpable” en forma de prototipo, que sobre un concepto.

- *P5: ¿Se han propuesto mejoras a lo mostrado en el prototipo?*

Si, se han propuesto mejoras tanto en contenido conversacional del *chatbot* como en cuanto a correcciones funcionales. En las iteraciones iniciales no se dio tanta importancia por parte del cliente al contenido conversacional pero a medida que se ha familiarizado con el sistema, a través de los test de usuario, a partir de la tercera iteración se ha empezado a pedir mejoras conversacionales que modulen el diálogo del *chatbot* de una forma más natural. Las correcciones funcionales se han realizado, como se proyectaba, durante el test de usuario al involucrarse ellos mismos en el desarrollo del sistema.

- *P6: ¿Se ha propuesto añadir mayor funcionalidad de lo mostrado en el prototipo?*

Si, en casi todas las iteraciones (salvo en la última) se ha propuesto nueva funcionalidad para la siguiente iteración que amplía las capacidades del *chatbot* respecto a lo propuesto en el prototipo de la iteración. Aquí es donde han surgido algunos de los problemas de información detectados. Al comprobar mediante el test de usuarios los clientes el alcance del prototipo, han podido entender qué problemas no estaba cubriendo el *chatbot* en cada iteración. En base a los problemas que no se estaba cubriendo con el *chatbot* han podido plantear nuevas funcionalidades que diesen respuesta. Han conseguido declarar el problema que no se cubría, pero no han aportado la información que se necesitaba para cubrirlo, o han quedado en aportar información pero finalmente no lo han realizado. Esto se debe a que no han dedicado siempre tiempo a pensar en el sistema fuera de las reuniones. Dado que la interacción con ellos es puramente reactiva, es decir, reaccionando en función de incorrecciones que hayan detectado en algo que les hemos enseñado, no hay respuesta proactiva por su parte. Esto puede deberse a que no perciben que tengan una responsabilidad asignada para cumplir antes del próximo test de usuarios. Para solventar esto se propone, como se ha explicado ya, indicar que solo se pueden implementar las propuestas de funcionalidades de las que se tiene información. Se necesita por tanto la reunión de la fase inicial expuesta para que un representante del cliente exponga la información y se decida en función de esta qué es lo que va a prototiparse en la iteración.

- *P7: ¿Podemos valorar positivamente el resultado obtenido en la prueba en términos de participación?*

Si, los participantes han colaborado en el test de usuario y han respondido a las preguntas formuladas posteriormente. Solo ha habido problemas en torno a la asistencia en un par de ocasiones donde gente que ha llegado tarde a la reunión ha tenido que recibir un resumen de lo hablado hasta el momento.

- *P8: ¿Han entendido los participantes del test con usuarios lo que se requería de ellos o ha sido necesario aclararlo varias veces?*

Se ha requerido aclarar la necesidad de concretar a mayores el dominio sobre el que opera el *chatbot*, pero en cuanto a la operativa y la participación en el test de usuarios no se ha requerido repetir las instrucciones a seguir. En una ocasión al conocer los usuarios las funcionalidades del prototipo de la iteración anterior se han intentado poner a probar estas funcionalidades por su cuenta, teniendo que redirigirles a las funcionalidades prototipadas a probar en la iteración en la que se estaba. Esto muestra la importancia de seguir un guión conocido por todas las partes, para que antes del inicio de la reunión de test de usuarios tengan la oportunidad de consultar qué es lo que se va a tratar en la reunión.

- *P9: ¿Se ha requerido recoger más información de los usuarios tras haber realizado los test de usuario?*

Se ha requerido realizar aclaraciones sobre la funcionalidad en ocasiones concretas y siempre en la siguiente reunión de test de usuarios. En ocasiones se ha necesitado solicitar por correo electrónico información para aclarar funcionalidades específicas. Sin embargo, no se han realizado reuniones a mayores para recabar más información de la funcionalidad a prototipar en la siguiente iteración.

5.8. Discusión

El resultado presentado mediante la aplicación de la propuesta de metodología al caso VirtualForest muestra que, aunque se cumplen los objetivos de la metodología, se está sufriendo de falta de información en el prototipado de ciertas funcionalidades porque el cliente no la está aportando en las fases iniciales de cada iteración.

Los objetivos que se han podido cumplir y que sí han funcionado como parte de la metodología han sido, en primer lugar, el de garantizar que los artefactos obtenidos permiten el análisis y diseño del sistema y en segundo lugar, el de mantener informado al cliente obteniendo feedback sobre el sistema. Solo se ha podido cumplir parcialmente el objetivo de la minimización de los problemas de información ya que se ha minimizado el riesgo de no obtener la información pero aún se puede mejorar más la metodología para obtener toda la información necesaria para el prototipado de una funcionalidad sin necesidad de arrastrar el problema de falta de información entre fases.

El primer objetivo cumplido ha permitido obtener, como resultado de la aplicación de la metodología, artefactos de información necesarios para comprender el contexto comunicativo y de interacción con un usuario a la hora de realizar la elicitación de requisitos en el análisis del sistema y posteriormente en la documentación de casos de uso y diagrama de clases elaborados a partir de estos. Los artefactos obtenidos permiten mostrar el diseño y lógica del sistema sobre el contexto de la comunicación con un usuario. A través de estos artefactos se obtiene la noción de cómo se comunica el sistema y qué lógica se sigue para cumplir los objetivos del diálogo.

El segundo objetivo cumplido ha sido el involucrar al cliente en el desarrollo del sistema para permitir obtener información tanto de nuevas funcionalidades, como para permitir realizar correcciones a las ya prototipadas. Como prueba de que se haya cumplido es que, en todas las iteraciones, se ha obtenido información sobre correcciones o ampliaciones de la funcionalidad para continuar el desarrollo. Por lo tanto el involucrar al cliente en el proceso de aplicación de la metodología ha tenido un resultado satisfactorio.

El objetivo cumplido parcialmente de falta de información ha provocado problemas derivados de esta falta de información, en una fase inicial de la metodología, que se transforman en problemas de incorrección de funcionalidades al extenderse por las distintas fases hasta que se resuelven en la aplicación de la técnica del *laddering*. Para poder cumplir el objetivo totalmente sería necesario un proceso de confirmación y aporte de información por parte del cliente en las fases tempranas de la metodología. Este impediría que los problemas se propaguen más allá de las fase inicial, mejorando la adquisición de información. Aunque se ha estructurado la sesión de test de usuario mediante planes de reunión, para dejar tiempo suficiente a plantear funcionalidades para la siguiente iteración, sería más conveniente que los clientes plantearan las funcionalidades en esa sesión pero que concretasen y aportasen información en una sesión posterior a celebrarse en un momento previo al comienzo de la fase de selección de objetivos para permitir que los clientes tomen un papel proactivo en al metodología, piensen de forma adecuada y

recopilen la información necesaria para las funcionalidades proyectadas si se han de prototipar en la iteración por comenzar.

La propuesta para mejora de la metodología incluiría una fase inicial con el cliente donde se especificase la información de las propuestas de funcionalidad nueva, realizadas como parte de la fase de test de usuarios. En base a la información recopilada se decidiría en la fase de selección de objetivos si, a la vista de esta, es viable llevar a cabo el prototipado de la funcionalidad o existen demasiadas dudas sobre el posible funcionamiento del prototipo correspondiente como para llevarlo a cabo en esta iteración. La propuesta metodológica con esta modificación se muestra en la figura 5.19.

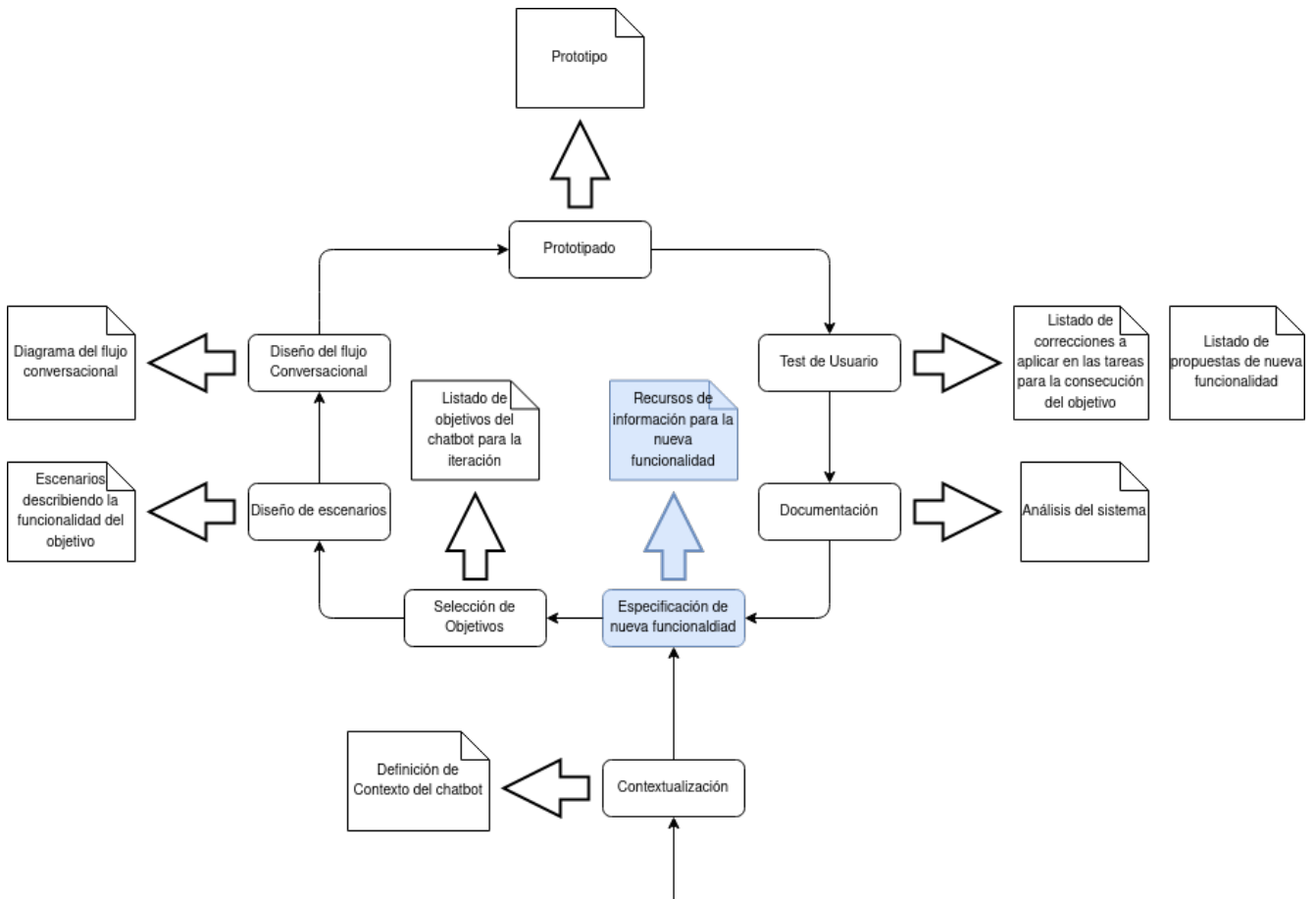


Figura 5.19: Esquema del flujo de fases a llevar a cabo en la propuesta de mejora de metodología final. En azul se muestra la ampliación propuesta.

Como se puede observar en la propuesta de mejora, se ha incluido una fase previa a la selección de objetivos que indica la especificación por parte del cliente de los recursos de información necesarios para los objetivos que éste quiere que se aborden en el prototipo de la iteración por empezar. La contextualización inicial ya no daría paso a la fase de selección de objetivos, sino que daría paso a esta fase donde se solicitaría al cliente los recursos de información iniciales para la primera iteración.

Dada la poca capacidad de tiempo disponible del cliente para la atención a este proyecto, la realización de esta fase se tendría que realizar probablemente a través de medios de comunicación asíncronos tal como comunicación a través de correos electrónicos. Sin embargo, esto tiene el peligro de no ser eficaz, ya que implicaría que el cliente proporcionase todos los recursos de información necesarios para poder llevar a cabo todas las funcionalidades propuestas y que no quedase ninguna duda en torno a esta información. Por lo tanto, se recomienda una reunión síncrona con el cliente en la que, a partir de un plan de reunión de no más de 10 minutos de duración, se proporcionase al desarrollador un diálogo de prueba que mostrase la funcionalidad a reproducir con el *chatbot* en su

interacción con el usuario y los recursos de información necesarios para poder construir ese diálogo.

5.9. Conclusión del capítulo

Como conclusión a este capítulo hay que decir que se ha presentado la organización del trabajo de aplicación de la metodología en cuatro iteraciones separadas, se ha caracterizado a los participantes del proceso de evaluación de prototipos por el test de usuario, se ha mostrado y explicado los artefactos resultantes de la aplicación de la metodología ,el análisis de la información que se ha realizado y las pruebas de validación del trabajo realizadas. Por último se ha discutido la evaluación del resultado favorable, obtenido por la aplicación de la metodología.

Capítulo 6

Conclusiones y trabajo futuro

En este trabajo se ha explicado que los *chatbots* son agentes conversacionales que a través de distintas técnicas, entre ellas el NLU, intentan simular el flujo no estructurado de la conversación humana. Se ha visto que los dominios de aplicación más populares de estos sistemas son la atención al cliente, la salud y bienestar físico y la educación. Dentro de la educación se ha demostrado los roles que puede tomar el *chatbot* en la comunicación con los alumnos y que el rol más popular es el de preguntas y respuestas. Por la investigación realizada, se ha determinado que los principales problemas que afectan a este tipo de sistema son de naturaleza tecnológica (necesitándose un análisis de alternativas de implementación para poderlos evitar) y conceptuales, relacionados con el contexto de comunicación (necesitándose un análisis de técnicas de elicitación de requisitos para escoger técnicas que ayuden a abordarlos).

Ante la situación del proyecto VirtualForest, donde se solicitó un *chatbot* para responder a problemas administrativos del alumnado de programas de másteres de la ETSIIAA, en el que los clientes no tenían claro el problema a cubrir, el cómo cubrirlo, ni la necesidad de definirlo que se les transmitió, se hizo patente la necesidad de un proceso pensado y refinado, una metodología, cuya aplicación resultase en un *chatbot* que diese respuesta a las necesidades propuestas por los clientes. Por esta razón y ante la falta de alternativas que abordasen el problema desde un punto de vista de análisis y no de desarrollo, en el trabajo presentado se ha realizado la propuesta de IMADTC y los criterios de validación de la misma tanto por la aplicación de las técnicas que la componen como por los resultados generales obtenidos en su aplicación.

La propuesta de metodología IMADTC, se basa en la realización iterativa de una serie de tareas para obtener y refinar información, hasta realizar un prototipo aprobado por el cliente, que de esta forma define el análisis de necesidades del cliente. Como método de validación de la metodología se ha aplicado ésta al caso del proyecto VirtualForest obteniéndose el desarrollo de este sistema como una contribución secundaria del trabajo y que se detalla en el cap. 5. La evaluación de la metodología muestra que la selección de técnicas de elicitación de requisitos es adecuada, pero que algunas debilidades ocasionales a la hora de recoger recursos de información, provocan a su vez imprecisiones en la funcionalidad prototipada. La forma propuesta de resolverlo sería la aplicación de una breve entrevista con el cliente en las fases iniciales de la metodología para aportar estos recursos de información e impedir que los fallos iniciales se extiendan por fases posteriores de la metodología.

Como trabajo futuro, quedaría seguir aplicando la propuesta a nuevos casos para por un lado, comprobar la completitud de la propuesta respecto a otros *chatbots* de características similares pero no iguales y por otro lado para seguir refinándola, aplicando modificaciones y correcciones a la misma y para poder determinar hasta qué punto es una propuesta metodológica válida y eficaz para los objetivos propuestos.

Bibliografía

- [1] “Virtual forest, virtualization of forest studies, proyecto financiado por la comisión europea dentro de las acciones ka2 - cooperation for innovation and the exchange of good practices, ka226 - paternships for digital education readiness. duración: 01/03/2021 al 28/02/2023.”
- [2] R. Adrion, “Research methodology in software engineering,” *Summary of the Dagstuhl Workshop on Future Directions in Software Engineering, volumen 18*, pp. 36–37, 1993.
- [3] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, “A New Chatbot for Customer Service on Social Media,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, (New York, NY, USA), pp. 3506–3510, Association for Computing Machinery, May 2017.
- [4] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, “Chatbot for university related FAQs,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1525–1530, Sept. 2017.
- [5] F. Colace, M. D. Santo, M. Lombardi, F. Pascale, A. Pietrosanto, and S. Lemma, “Chatbot for e-learning: A case of study,” *IJMERR issue 7 n^o5 pp. 528-533*, 2018.
- [6] N. Albayrak, A. Ozdemir, and E. Zeydan, “An overview of artificial intelligence based chatbots and an example chatbot application,” in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, (Izmir), pp. 1–4, IEEE, May 2018.
- [7] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, “SuperAgent: A Customer Service Chatbot for E-commerce Websites,” in *Proceedings of ACL 2017, System Demonstrations*, (Vancouver, Canada), pp. 97–102, Association for Computational Linguistics, 2017.
- [8] B. AbuShawar and E. Atwell, “ALICE Chatbot: Trials and Outputs,” in *scielomx*, vol. 19, pp. 625–632, 12 2015.
- [9] Y. Zhang, T. Xu, and Y. Dai, “Research on Chatbots for Open Domain: Using BiLSTM and Sequence to Sequence,” in *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*, AICS 2019, (New York, NY, USA), pp. 145–149, Association for Computing Machinery, July 2019.
- [10] D. Rooein, “Data-Driven Edu Chatbots,” in *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, (New York, NY, USA), pp. 46–49, Association for Computing Machinery, May 2019.
- [11] A. Følstad and M. Skjuve, “Chatbots for customer service: user experience and motivation,” in *Proceedings of the 1st International Conference on Conversational User Interfaces*, CUI '19, (New York, NY, USA), pp. 1–9, Association for Computing Machinery, Aug. 2019.
- [12] P. Kucherbaev, A. Psyllidis, and A. Bozzon, “Chatbots as conversational recommender systems in urban contexts,” in *Proceedings of the International Workshop on Recommender Systems for Citizens*, CitRec '17, (New York, NY, USA), Association for Computing Machinery, 2017.

- [13] M. Dahiya, “A Tool of Conversation: Chatbot,” *International Journal of Computer Sciences and Engineering*, p. 5, 2017.
- [14] S. Bell, C. Wood, and A. Sarkar, “Perceptions of chatbots in therapy,” in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, (New York, NY, USA), p. 1–6, Association for Computing Machinery, 2019.
- [15] A. Abdellatif, K. Badran, and E. Shihab, “MSRBot: Using bots to answer questions from software repositories,” *Empirical Software Engineering*, vol. 25, pp. 1834–1863, May 2020.
- [16] E. Amer, A. Hazem, O. Farouk, A. Louca, Y. Mohamed, and M. Ashraf, “A Proposed Chatbot Framework for COVID-19,” *IEEE*, pp. 263–268, May 2021.
- [17] S. Hobert, “Say hello to ‘coding tutor’! design and evaluation of a chatbot-based learning system supporting students to learn to program,” in *Proceedings of the 40th International Conference on Information Systems, ICIS 2019, Munich, Germany, December 15-18, 2019* (H. Krcmar, J. Fedorowicz, W. F. Boh, J. M. Leimeister, and S. Wattal, eds.), Association for Information Systems, 2019.
- [18] J. Jia, “The study of the application of a web-based chatbot system on the teaching of foreign languages,” in *Proceedings of Society for Information Technology Teacher Education International Conference 2004* (R. Ferdig, C. Crawford, R. Carlsen, N. Davis, J. Price, R. Weber, and D. A. Willis, eds.), (Atlanta, GA, USA), pp. 1201–1207, Association for the Advancement of Computing in Education (AACE), 2004.
- [19] D. Griol, J. M. Molina, Z. Callejas, and R. López-Cózar, “La plataforma educagent: Agentes conversacionales inteligentes y entornos virtuales aplicados a la docencia,” *Revista Iberoamericana de Informática Educativa issue 13 pp. 15-31*, 2011.
- [20] Q. Liu, J. Huang, L. Wu, K. Zhu, and S. Ba, “CBET: design and evaluation of a domain-specific chatbot for mobile learning,” *Universal Access in the Information Society*, vol. 19, pp. 655–673, Aug. 2020.
- [21] L. Ni, C. Lu, N. Liu, and J. Liu, “MANDY: Towards a Smart Primary Care Chatbot Application,” in *Knowledge and Systems Sciences* (J. Chen, T. Theeramunkong, T. Supnithi, and X. Tang, eds.), vol. 780, pp. 38–52, Singapore: Springer Singapore, 2017. Series Title: Communications in Computer and Information Science.
- [22] A. A. Qaffas, “Improvement of chatbots semantics using wit.ai and word sequence kernel: Education chatbot as a case study,” *International Journal of Modern Education and Computer Science*, vol. 11, pp. 16–22, 2019.
- [23] P. A. Tamayo, A. Herrero, J. Martin, C. Navarro, and J. M. Tranchez, “Design of a chatbot as a distance learning assistant,” *Open Praxis*, vol. 12, no. 1, p. 145–153, 2020.
- [24] V. Deshmukh and S. J. Nirmala, “Open domain conversational chatbot,” in *Information, Communication and Computing Technology* (A. B. Gani, P. K. Das, L. Kharb, and D. Chahal, eds.), (Singapore), pp. 266–278, Springer Singapore, 2019.
- [25] M. B. Ibáñez, J. J. García, S. Galán, D. Maroto, D. Morillo, and C. D. Kloos, “Design and implementation of a 3d multi-user virtual world for language learning,” *Journal of Educational Technology Society*, vol. 14, no. 4, pp. 2–10, 2011.
- [26] S. Roos, “Chatbots in education, a passing trend or a valuable pedagogical tool?,” *Uppsala University*, pp. 6–49, 6 2018.
- [27] K. Chung and R. C. Park, “Chatbot-based healthcare service with a knowledge base for cloud computing,” *Cluster Computing*, vol. 22, pp. 1925–1937, Jan 2019.

- [28] S. Laumer, C. Maier, and F. T. Gubler, “Chatbot acceptance in healthcare: Explaining user adoption of conversational agents for disease diagnosis,” in *27th European Conference on Information Systems - Information Systems for a Sharing Society, ECIS 2019, Stockholm and Uppsala, Sweden, June 8-14, 2019* (J. vom Brocke, S. Gregor, and O. Müller, eds.), 2019.
- [29] J. Santamarta Martín, “Desarrollo de un sistema de interacción basado en google home y dialog flow,” *Universidad de Valladolid, Escuela Ingenierías Industriales*, p. 106, 2021.
- [30] A. Følstad, C. B. Nordheim, and C. A. Bjørkli, “What makes users trust a chatbot for customer service? an exploratory interview study,” in *Internet Science* (S. S. Bodrunova, ed.), (Cham), pp. 194–208, Springer International Publishing, 2018.
- [31] T. Nguyen, “Potential effects of chatbot technology on customer support: A case study,” master’s thesis, Aalto University. School of Science, 2019.
- [32] F. Clarizia, F. Colace, M. Lombardi, F. Pascale, and D. Santaniello, “Chatbot: An education support system for student,” in *Cyberspace Safety and Security* (A. Castiglione, F. Pop, M. Ficco, and F. Palmieri, eds.), (Cham), pp. 291–302, Springer International Publishing, 2018.
- [33] J. Bang, H. Noh, Y. Kim, and G. G. Lee, “Example-based chat-oriented dialogue system with personalized long-term memory,” in *2015 International Conference on Big Data and Smart Computing (BIGCOMP)*, pp. 238–243, Feb. 2015.
- [34] H. Aust, M. Oerder, F. Seide, and V. Steinbiss, “The philips automatic train timetable information system,” *Speech Commun.*, vol. 17, p. 249–262, Nov. 1995.
- [35] M.-C. Hsieh, W.-S. Hung, S.-S. Shin, S.-W. Lin, and T.-H. Huang, “Spoken Dialogue Agent Interface Requirements Modeling Based on PASSI Methodology,” in *2008 Eighth International Conference on Intelligent Systems Design and Applications*, vol. 1, pp. 339–342, Nov. 2008. ISSN: 2164-7151.
- [36] G. Chung, “Developing a flexible spoken dialog system using simulation,” in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL ’04*, (Barcelona, Spain), pp. 63–es, Association for Computational Linguistics, 2004.
- [37] S. Lutfi, F. Fernández-Martínez, J. Lorenzo-Trueba, R. Barra-Chicote, and J. Montero, “I Feel You: The Design and Evaluation of a Domotic Affect-Sensitive Spoken Conversational Agent,” *Sensors*, vol. 13, pp. 10519–10538, Aug. 2013.
- [38] D. Griol, Z. Callejas, and R. López-Cózar, “Statistical dialog management methodologies for real applications,” in *Proceedings of the SIGDIAL 2010 Conference*, (Tokyo, Japan), pp. 269–272, Association for Computational Linguistics, Sept. 2010.
- [39] V. Hung, M. Elvir, A. Gonzalez, and R. DeMara, “Towards a method for evaluating naturalness in conversational dialog systems,” in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1236–1241, Oct. 2009.
- [40] D. Griol, Z. Callejas, R. López-Cózar, and G. Riccardi, “A domain-independent statistical methodology for dialog management in spoken dialog systems,” *Computer Speech & Language*, vol. 28, pp. 743–768, May 2014.
- [41] D. Griol, L. Hurtado Oliver, E. Segarra, and E. Sanchis, “A Statistical Approach to Spoken Dialog Systems Design and Evaluation,” *Speech Communication*, vol. 50, pp. 666–682, Aug. 2008.
- [42] A. Loisel, N. Chaignaud, and J.-P. Kotowicz, “Designing a human-computer dialog system for medical information search,” in *2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, pp. 350–353, Dec. 2007.

- [43] T. Bickmore and T. Giorgino, “Methodological Review: Health Dialog Systems for Patients and Consumers,” *Journal of Biomedical Informatics - JBI*, June 2021.
- [44] J. Nievergelt, C. Muller, and H. Sugaya, *Dialog Design: Principles and Experiments*, pp. 171–193. Boston, MA: Springer US, 1986.
- [45] G. M. Linders and M. M. Louwerse, “Zipf’s Law in Human-Machine Dialog,” in *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents, IVA ’20*, (New York, NY, USA), pp. 1–8, Association for Computing Machinery, Oct. 2020.
- [46] W. R. Kearns, N. Kaura, M. Divina, C. Vo, D. Si, T. Ward, and W. Yuwen, “A wizard-of-oz interface and persona-based methodology for collecting health counseling dialog,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI EA ’20*, (New York, NY, USA), p. 1–9, Association for Computing Machinery, 2020.
- [47] F. Enembreck and J.-P. Barthès, “Architecture of a dialog system with an assistant agent,” in *Proceedings of the 15th Conference on l’Interaction Homme-Machine, IHM ’03*, (New York, NY, USA), p. 96–103, Association for Computing Machinery, 2003.
- [48] H. Ryu, S. Kim, D. Kim, S. Han, K. Lee, and Y. Kang, “Simple and steady interactions win the healthy mentality: Designing a chatbot service for the elderly,” *Proc. ACM Hum.-Comput. Interact.*, vol. 4, Oct. 2020.
- [49] L. Zhou, J. Gao, D. Li, and H. Shum, “The design and implementation of xiaoice, an empathetic social chatbot,” *CoRR*, vol. abs/1812.08989 p. 35, 2018.
- [50] A. Celikyilmaz, L. Deng, and D. Hakkani-Tür, “Deep Learning in Spoken and Text-Based Dialog Systems,” in *Deep Learning in Natural Language Processing* (L. Deng and Y. Liu, eds.), pp. 49–78, Singapore: Springer, 2018.
- [51] A. Nichol, “CDD Playbook,” *Rasa Technologies Incorporated*, 2020.
- [52] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, “Challenges in Chatbot Development: A Study of Stack Overflow Posts,” in *Proceedings of the 17th International Conference on Mining Software Repositories, MSR ’20*, (New York, NY, USA), pp. 174–185, Association for Computing Machinery, June 2020.
- [53] T. P. Nagarhalli, V. Vaze, and N. K. Rana, “A Review of Current Trends in the Development of Chatbot Systems,” in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 706–710, Mar. 2020. ISSN: 2575-7288.
- [54] N. C. Bradley, T. Fritz, and R. Holmes, “Context-aware conversational developer assistants,” in *Proceedings of the 40th International Conference on Software Engineering, ICSE ’18*, (New York, NY, USA), pp. 993–1003, Association for Computing Machinery, May 2018.
- [55] G. Cameron, D. W. Cameron, G. Megaw, R. R. Bond, M. Mulvenna, S. B. O’Neill, C. Armour, and M. McTear, “Best Practices for Designing Chatbots in Mental Healthcare – A Case Study on iHelp,” July 2018. Publisher: BCS Learning & Development.
- [56] M. Nuruzzaman and O. K. Hussain, “A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks,” in *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, pp. 54–61, Oct. 2018.
- [57] S. Westberg, “Applying a chatbot for assistance in the onboarding process : A process of requirements elicitation and prototype creation,” Master’s thesis, Linköping University, Human-Centered systems, 2019.

- [58] H. Jung, J. Lee, and C. Park, “Deriving Design Principles for Educational Chatbots from Empirical Studies on Human–Chatbot Interaction,” *Journal of Digital Contents Society*, vol. 21, pp. 487–493, Mar. 2020.
- [59] E. Hoff, “How social contexts support and shape language development,” *Developmental Review*, vol. 26, pp. 55–88, Mar. 2006.
- [60] P. Knoeferle, “Predicting (variability of) context effects in language comprehension,” *Journal of Cultural Cognitive Science*, vol. 3, pp. 141–158, Nov. 2019.
- [61] A. P. Chaves, E. Doerry, J. Egbert, and M. Gerosa, “It’s How You Say It: Identifying Appropriate Register for Chatbot Language Design,” in *Proceedings of the 7th International Conference on Human-Agent Interaction*, (Kyoto Japan), pp. 102–109, ACM, Sept. 2019.
- [62] A. P. Chaves, J. Egbert, T. Hocking, E. Doerry, and M. A. Gerosa, “Chatbots language design: the influence of language variation on user experience,” *CoRR p. 34*, Jan. 2021.
- [63] T. Rietz and A. Maedche, “LadderBot: A Requirements Self-Elicitation System,” in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pp. 357–362, Sept. 2019.
- [64] J. H. Hayes and J. Offutt, “Input validation analysis and testing,” *Empirical Software Engineering*, vol. 11, pp. 493–522, Dec. 2006.
- [65] H. Bunt, “Context and dialogue control,” *THINK Quarterly*, vol. 3, 1994.
- [66] P. Fodor, “Dialog Management for Decision Processes,” Master’s thesis, Stony Brook University, 8 2008.
- [67] T. Paek and D. M. Chickering, “The Markov assumption in spoken dialogue management,” in *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, (Lisbon, Portugal), pp. 35–44, Special Interest Group on Discourse and Dialogue (SIGdial), Sept. 2005.
- [68] C. Liebrecht, L. Sander, and C. van Hooijdonk, “Too informal? how a chatbot’s communication style affects brand attitude and quality of interaction,” 2020. Conversations 2020 : 4th international workshop on chatbot research ; 23-11-2020 Through 24-11-2020.
- [69] H. Cuayahuitl, S. Renals, O. Lemon, and H. Shimodaira, “Human-computer dialogue simulation using hidden Markov models,” in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, (San Juan, Puerto Rico), pp. 290–295, IEEE, 2005.
- [70] E. Ribeiro, R. Ribeiro, and D. M. de Matos, “The Influence of Context on Dialogue Act Recognition,” *arXiv:1506.00839 [cs]*, Jan. 2017. arXiv: 1506.00839.
- [71] C. Pacheco, I. García, and M. Reyes, “Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques,” *IET Software*, vol. 12, no. 4, pp. 365–378, 2018.
- [72] A. Sajid, A. Nayyar, and A. Mohsin, “Modern trends towards requirement elicitation,” in *Proceedings of the 2010 National Software Engineering Conference, NSEC ’10*, (New York, NY, USA), Association for Computing Machinery, 2010.
- [73] L. Jiang, A. Eberlein, B. H. Far, and M. Mousavi, “A methodology for the selection of requirements engineering techniques,” *Software & Systems Modeling*, vol. 7, pp. 303–328, Jul 2008.
- [74] L. R. Wong, D. S. Mauricio, and G. D. Rodriguez, “A systematic literature review about software requirements elicitation,” *journal of engineering science and technology*, vol. 12, pp. 296–317, 2017.
- [75] A. Sutcliffe, “Scenario-based requirements engineering,” in *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.*, pp. 320–329, Sept. 2003. ISSN: 1090-705X.

- [76] D. Carrizo, O. Dieste, and N. Juristo, “Systematizing requirements elicitation technique selection,” *Information and Software Technology*, vol. 56, pp. 644–669, June 2014.
- [77] S. Parveen and R. Kumar, “Software requirement elicitation using goal oriented requirement engineering: A literature review,” *International Journal of Advances in Electronics and Computer Science*, volume 3 Issue 8, vol. 3, p. 5, 8 2016.
- [78] M. Yousuf and M. M. Asger, “Comparison of Various Requirements Elicitation Techniques,” *International Journal of Computer Applications*, vol. 116, pp. 8–15, Apr. 2015.
- [79] T.-Y. Lim, F.-F. Chua, and B. B. Tajuddin, “Elicitation Techniques for Internet of Things Applications Requirements: A Systematic Review,” in *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, ICNCC 2018, (New York, NY, USA), pp. 182–188, Association for Computing Machinery, Dec. 2018.
- [80] J. L. de la Vara and J. Sánchez, “BPMN-Based Specification of Task Descriptions: Approach and Lessons Learnt,” in *Requirements Engineering: Foundation for Software Quality* (M. Glinz and P. Heymans, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 124–138, Springer, 2009.
- [81] D. Zowghi and C. Coulin, *Requirements Elicitation: A Survey of Techniques, Approaches, and Tools*, pp. 19–46. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
- [82] S. Khan, A. B. Dulloo, and M. Verma, “Systematic Review of Requirement Elicitation Techniques,” *International Journal of Information and Computation Technology*, Volume 4, Number 2, pp. 133–138, 2014.
- [83] “Characterizing Software Requirements Elicitation Processes: A Systematic Literature Review | Proceedings of the XII Brazilian Symposium on Information Systems on Brazilian Symposium on Information Systems: Information Systems in the Cloud Computing Era - Volume 1,” 2016.
- [84] D. Carrizo and I. Quintanilla, “Prototyping Use as a Software Requirements Elicitation Technique: A Case Study,” in *Trends and Advances in Information Systems and Technologies* (I. Rocha, H. Adeli, L. P. Reis, and S. Costanzo, eds.), Advances in Intelligent Systems and Computing, (Cham), pp. 341–350, Springer International Publishing, 2018.
- [85] M. Davies, “Concept mapping, mind mapping and argument mapping: what are the differences and do they matter?,” *Higher Education*, vol. 62, pp. 279–301, Sept. 2011.
- [86] D. C. Moreno, *Selecting Software Requirements Elicitation Techniques*. LAP LAMBERT Academic Publishing, 2011.
- [87] E. N. Biktimirov and L. B. Nilson, “Show Them the Money: Using Mind Mapping in the Introductory Finance Course,” *Journal of Financial Education*, vol. 32, pp. 72–86, 2006. Publisher: Financial Education Association.
- [88] R. Escalfoni, V. Braganholo, and M. R. Borges, “A method for capturing innovation features using group storytelling,” *Expert Systems with Applications*, vol. 38, pp. 1148–1159, Feb. 2011.
- [89] D. Horváth, “From user feedback to requirements using chatbots,” master thesis, Utrecht University, 2019.
- [90] O. D. Tubio, M. López, and F. Ramos, “Updating a systematic review about selection of software requirements elicitation techniques,” *11th Workshop em Engenharia de Requisitos, WER*, p.8 2009.
- [91] S. Jha, “Chatbot application Life cycle.” <https://medium.datadriveninvestor.com/chatbot-application-life-cycle-8b2d083650a8>, Jan. 2019. Web, accedido el 12-04-2021.

- [92] S. Sharma and S. K. Pandey, “Article: Revisiting requirements elicitation techniques,” *International Journal of Computer Applications*, vol. 75, pp. 35–39, August 2013.
- [93] R. Nuseibeh, “What is a chatbot?,” *Chatbot magazine*, 2021. [Web; accedido el 12-04-2021].
- [94] L. Vishnoi, “Conversational agent: A more assertive form of chatbots,” *Towards Data Science*, Jun 30, 2020. [Web; accedido el 12-04-2021].
- [95] S. Demetriadis, S. Tegos, G. Psathas, T. Tsiatsos, A. Weinberger, S. Caballé, Y. Dimitriadis, E. G. Sánchez, P. M. Papadopoulos, and A. Karakostas, “Conversational agents as group-teacher interaction mediators in moocs,” in *2018 Learning With MOOCS (LWMOOCS)*, pp. 43–46, 2018.
- [96] Wit.ai, “Frequently asked questions of wit.ai.” <https://wit.ai/faq>, 2020. [Web; accedido el 19-04-2021].
- [97] Wit.ai, “Terms of use of wit.ai.” <https://wit.ai/terms>, 2020. [Web; accedido el 19-04-2021].
- [98] IBM, “Ibm watson plan information.” <https://cloud.ibm.com/docs/assistant?topic=assistant-services-information>, 2020. [Web; accedido el 20-04-2021].
- [99] Google, “Registro de interacción de dialogflow.” <https://cloud.google.com/dialogflow/es/docs/interaction-logging>, 2020. [Web; accedido el 22-04-2021].
- [100] Microsoft, “Exportación y eliminación de datos del cliente en language understanding (luis) en cognitive services.” <https://docs.microsoft.com/es-es/azure/cognitive-services/luis/luis-user-privacy>, 2020. [Web; accedido el 22-04-2021].
- [101] Amazon, “¿qué es amazon lex?.” https://docs.aws.amazon.com/es_es/lex/latest/dg/what-is.html, 2020. [Web; accedido el 22-04-2021].
- [102] Amazon, “Preguntas frecuentes de amazon lex.” <https://aws.amazon.com/es/lex/faqs/>, 2020. [Web; accedido el 22-04-2021].
- [103] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, “Manifiesto for agile software development,” 2001.
- [104] Atlassian, “Scrum: qué es, cómo funciona y por qué es excelente.” <https://www.atlassian.com/es/agile/scrum>. Web, accedido el 03-05-2021.
- [105] “What is Extreme Programming (XP)?.” <https://www.agilealliance.org/glossary/xp/>, June 2017. Web, accedido el 03-05-2021.
- [106] “Behaviour Driven Development: Embrace Uncertainty | Agile Alliance.” <https://www.agilealliance.org/resources/sessions/behaviour-driven-development-embrace-uncertainty/>, May 2015. Web, accedido el 03-05-2021.
- [107] A. Nichol, “5 Levels of Conversational AI.” <https://blog.rasa.com/5-levels-of-conversational-ai-2020-update/>, 2020. Web, Accedido el 24/05/2021.
- [108] A. Nichol, “The next generation of AI assistants in enterprise Chatbots are just the first step in the journey to achieve true AI assistants and autonomous organizations.” <https://www.oreilly.com/radar/the-next-generation-of-ai-assistants-in-enterprise/>, 8 2018. Web, Accedido el 24/05/2021.
- [109] M. Cohn, *User Stories Applied For Agile Software Development*. Addison-Wesley Professional, 2004.
- [110] G. Lucassen, F. Dalpiaz, J. M. E. M. v. d. Werf, and S. Brinkkemper, “The Use and Effectiveness of User Stories in Practice,” in *Requirements Engineering: Foundation for Software Quality* (M. Daneva and O. Pastor, eds.), Lecture Notes in Computer Science, (Cham), pp. 205–222, Springer International Publishing, 2016.

- [111] J. C. S. do Prado Leite, G. D. S. Hadad, J. H. Doorn, and G. N. Kaplan, “A Scenario Construction Process,” *Requirements Engineering*, vol. 5, pp. 38–61, July 2000.
- [112] T. Veludo-de Oliveira, A. Ikeda, and M. Campomar, “Discussing Laddering Application by the Means-End Chain Theory,” *The Qualitative Report*, Jan. 2015.
- [113] “Responsible bots: 10 guidelines for developers of conversational AI,” Nov. 2018.
- [114] “Information processing — documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts,” standard, International Organization for Standardization, Feb. 1985.

Apéndice A: Acrónimos

- AIML (Artificial Intelligence Mark-up Language)
- API (Application Programming Interface)
- ASR (Automatic Speech Recognition)
- CDD (Conversation Driven Development)
- CD (Continuous Delivery)
- CI (Continuous Integration)
- ES (Essential Edition)
- GDPR (General Data Protection Regulation)
- HTTP (Hypertext Transfer Protocol)
- IA (Inteligencia Artificial)
- IMADTC (Iterative Methodology for the Analysis and Development of Text based Chatbots)
- LSTM (Long Short Term Memory)
- MAU (Monthly Active Users)
- MIT (Massachusetts Institute of Technology)
- ML (Machine Learning)
- NLU (Natural Language Understanding)
- REST (Representational State Transfer)
- SDK (Software Development Kit)
- STT (Speech To Text)
- TTS (Text to Speech)
- VA (Virtual Assistant)
- XP (Extreme Programming)
- YAML (Acrónimo recursivo: YAML(Yet Another Markup Language) Ain't Markup Language)

Apéndice B: Código del backend del prototipo de VirtualForest

El código se encuentra en formato descargable en <https://github.com/juaalon/CodigoPrototipoEntregaTFM>.
git.

A este código se aplican las mismas licencias que al resto del trabajo.