



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

**TRABAJO FIN DE MÁSTER
APLICACIÓN DE LA INTELIGENCIA ARTIFICIAL PARA EL
USO DE BOTS EN EL PORTAL DEL EMPLEADO**

**D. PRESNO GARCÍA, Andrés
TUTOR: D. NUÑO HUERGO, Pelayo
CO-TUTOR: D. DE BENITO ARANGO, Rodolfo**

FECHA: Julio 2020

Quiero agradecer públicamente:

A Seresco por la oportunidad que me ha dado.

A mis compañeros de departamento y en especial a:

Antonio, Celia, Chus, Rodolfo, Francisco y Pelli.

A la E.P.I. y a todo el personal que conforma la escuela de Telecomunicaciones, porque debemos sentirnos orgullos de la alta calidad formativa que tenemos.

*Con mención especial para Pelayo Nuño,
por su compromiso con la formación universitaria y
por superar todas las expectativas que un alumno espera
de un docente.*

*A mi familia, pareja y compañeros,
por ser mi apoyo en este largo recorrido académico.*

Índice General

1. Memoria	15
1.1 Introducción.....	15
1.1.1 Enfoque desde una perspectiva empresarial.....	16
1.2 Objetivos y motivación.....	18
1.3 Demanda actual de los bots en RRHH	19
1.4 Fundamentos del procesamiento de lenguaje	20
1.4.1 Concepto: Intérprete de lenguaje natural (NLU).....	20
1.4.2 Concepto: Base de conocimiento	21
1.5 Estudio de las principales plataformas	22
1.5.1 Plataforma: Dialogflow	22
1.5.1.1 Consulta de información en Dialogflow.....	23
1.5.1.2 Solicitud en Dialogflow.....	26
1.5.1.3 Estadísticas y entrenamiento en Dialogflow	27
1.5.1.4 Eventos en Dialogflow	29
1.5.1.5 Multilenguaje en Dialogflow	30
1.5.1.6 API.ai de Google Cloud Platform (Dialogflow API).....	30
1.5.1.7 Integración en canales en Dialogflow	31
1.5.2 Plataforma: Azure Bot Service	32
1.5.2.1 LUIS (NLU)	33
1.5.2.2 QnA Maker (Base de Conocimiento).....	34
1.5.3 Plataforma: Amazon Lex.....	34
1.5.4 Plataforma: IBM Watson Assistant	35
1.5.5 Análisis de plataformas	36
1.5.6 Evaluación económica.....	39

1.5.6.1 Evaluación económica: Dialogflow	39
1.5.6.2 Evaluación económica: Azure	40
1.5.7 Elección de la tecnología.....	43
1.6 Arquitectura del chatbot	45
1.6.1 Despliegue del back-end en la red corporativa.....	45
1.6.2 Despliegue del back-end en el cloud de Azure.....	47
1.6.3 Análisis de arquitecturas.....	49
1.6.4 Análisis de servicios de interpretación de lenguaje.....	52
1.7 Flujo de conversación.....	55
1.7.1 Flujo de conversación: base de conocimiento	55
1.7.2 Flujo de conversación: procesos / procedimientos	56
1.7.3 Flujo de conversación: formularios	57
1.7.4 Plataforma de contenidos.....	58
1.8 Planificación del proyecto	59
1.8.1 Equipo de proyecto.....	59
1.8.2 Tareas del proyecto.....	60
1.9 Trabajo futuro y conclusiones	63
2. Presupuesto.....	65
2.1 Hardware: equipamiento y amortización.....	65
2.2 Software: equipamiento y amortización	66
2.3 Mano de obra de ingeniería	67
2.4 Presupuesto total.....	68
3. Documentos Técnicos.....	69
3.1 Requisitos	69
3.1.1 Requisitos funcionales.....	69

3.1.2 Requisitos no funcionales	78
3.2 Diseño de la herramienta	82
3.2.1 Diseño del back-end	82
3.2.1.1 Diseño y control del flujo de conversación	82
3.2.1.1.1 Diseño de los procesos y base de conocimiento.....	84
3.2.1.1.2 Diseño de formularios	87
3.2.2 Diseño del front-end	90
3.2.2.1 Comunicación servidor - cliente.....	91
3.2.3 Diseño de los agentes (API.ai Dialogflow)	91
3.2.4 Diseño de la base de datos (Oracle)	93
3.3 Descripción del sistema.....	95
3.3.1 Casos de uso	95
3.3.1.1 Casos de uso: subsistema de procesos y BC	96
3.3.1.2 Casos de uso: subsistema de formulario vacaciones	101
3.3.1.3 Casos de uso: subsistema de formulario permisos retribuidos.....	105
3.3.1.4 Casos de uso: subsistema de nómina.....	110
3.3.1.5 Casos de uso: subsistema de certificado retenciones.....	113
3.3.2 Diagramas de secuencia.....	116
3.3.2.1 Diagrama de secuencia: consulta funcionalidades	116
3.3.2.2 Diagrama de secuencia: formulario de permiso retribuido sin entidades...	117
3.3.2.3 Diagrama de secuencia: formulario de nómina con entidades	118
3.4 Descripción del back-end: Milena Server	119
3.4.1 Diagrama de estados de procesos	120
3.4.2 Diagrama de estados de los formularios.....	120
3.4.3 Clases, métodos y funciones del back-end	123

3.5 Pruebas del back-end	130
3.5.1 Verificación del back-end.....	130
3.5.1.1 Casos de prueba: subsistema de procesos y BC	131
3.5.1.2 Casos de prueba: subsistema de formulario vacaciones	136
3.5.1.2.1 Pruebas de vacaciones sin entidades en la expresión	136
3.5.1.2.2 Pruebas de vacaciones con entidades en la expresión	138
3.5.1.3 Casos de prueba: subsistema de formulario de permisos retribuidos	140
3.5.1.3.1 Pruebas de permisos retribuidos sin entidades en la expresión	140
3.5.1.3.2 Pruebas de permiso retribuido por matrimonio sin entidades en la expresión.....	142
3.5.1.3.3 Pruebas de compensación de horas sin entidades en la expresión	142
3.5.1.3.4 Pruebas de permisos con entidades en la expresión	144
3.5.1.4 Casos de prueba: subsistema de formulario de nómina.....	146
3.5.1.4.1 Pruebas de nómina sin entidades en la expresión.....	146
3.5.1.4.2 Pruebas de nómina con entidades en la expresión.....	146
3.5.1.5 Casos de prueba: subsistema de formulario de certificado de retenciones.	147
3.5.1.5.1 Pruebas de certificado sin entidades en la expresión.....	147
3.5.1.5.2 Pruebas de certificado con entidades en la expresión.....	147
3.5.1.6 Casos de prueba: cancelación de formularios.....	148
4. Manual de usuario	149
4.1 Manual de procesos y base de conocimiento.....	150
4.2 Manual de formularios	151
5. Manual de configuración	153
5.1 Google Cloud Platform: administración de agentes	153
5.2 Cliente ligero - cx_Oracle: cifrado de tráfico de red	155



Bibliografía..... 157

Índice de figuras

Figura 1 - Logo de MilenaBot	18
Figura 2 - Tareas del desarrollador en Dialogflow	22
Figura 3 – Flujo de conversación de consulta de información en Dialogflow	24
Figura 4 - Ejemplo de solicitud de información con el piloto de Dialogflow	25
Figura 5 - Respuestas de fallback en Dialogflow	25
Figura 6 - Consulta directa en Dialogflow	26
Figura 7 - Flujo de conversación de solicitud de información en Dialogflow	26
Figura 8 - Comportamiento piloto Dialogflow ante solicitud	27
Figura 9 - Estadísticas de comportamiento del usuario de Dialogflow	27
Figura 10 - Diagrama de flujo de Dialogflow	28
Figura 11 - Historial de Dialogflow	28
Figura 12 - Ejemplo de entrenamiento de Dialogflow	29
Figura 13 - Generación de evento mediante POST en Dialogflow	29
Figura 14 - Piloto Dialogflow interacción en inglés	30
Figura 15 - Primera llamada a la API de Dialogflow	31
Figura 16 - Configuración de un intent en LUIS	33
Figura 17 - Prueba de emulador con server con base de conocimiento	34
Figura 18 - Ejemplo de configuración de un agente de IBM Watson	35
Figura 19 - Estructura de la plataforma de Dialogflow e IBM Watson.	36
Figura 20- Estructura de la plataforma de Microsoft Bot Services.	36
Figura 21 - Desglose de costes de piloto mes de febrero	41
Figura 22 - Arquitectura local con servicios de IA de Azure	46
Figura 23 - Arquitectura local con servicios de IA de Google	46
Figura 24 - Arquitectura local con RASA	47
Figura 25 - Arquitectura Bot implementado en cloud Azure	48
Figura 26 - Evaluación de NLU	54
Figura 27 - Ejemplo de petición a base de conocimiento	55
Figura 28 - Ejemplo de proceso sobre explicación de nómina	56
Figura 29 - Ejemplo de proceso con elemento multimedia	57
Figura 30 - Ejemplo de formulario de solicitud vacacional	58

Figura 31 - Diagrama de Gantt.....	62
Figura 32 - Diseño del flujo de conversación de un proceso: paso 1º.....	85
Figura 33 - Diseño del flujo de conversación de un proceso: paso 2º.....	85
Figura 34 - Diagrama de flujo de ejemplo de un proceso.....	86
Figura 35 - Diseño de flujo tras contestación no-matched.....	86
Figura 36 - Diseño del flujo de conversación, control de permisos.....	87
Figura 37 - Diseño del flujo de conversación de un formulario: paso 1º.....	87
Figura 38 - Diseño del flujo de conversación de un formulario: paso 2º.....	88
Figura 39 - Diseño del flujo de conversación de un formulario: paso 3º.....	88
Figura 40 - Diagrama de flujo de ejemplo de un formulario.....	89
Figura 41 - Testeo con Bot Framework Emulator.....	90
Figura 42 - Diseño de flujo en fallo por conexión con Dialogflow.....	92
Figura 43 - Diagrama entidad - relación de la plataforma de contenidos.....	94
Figura 44 - Casos de uso del subsistema de procesos y base de conocimiento.....	96
Figura 45 - Casos de uso del subsistema de formulario vacaciones.....	101
Figura 46 - Casos de uso del formulario de permisos retribuidos.....	105
Figura 47 - Diagrama de casos de usos del subsistema de nómina.....	110
Figura 48 - Diagrama de casos de uso del certificado de retenciones.....	113
Figura 49 - Diagrama de secuencia: consulta funcionalidades.....	116
Figura 50 - Diagrama de secuencia de formulario de permiso retribuido.....	117
Figura 51 - Diagrama de secuencia de formulario de nómina.....	118
Figura 52 - Diagrama de clases de BotLogic.....	120
Figura 53 - Diagrama de estados del formulario de vacaciones.....	121
Figura 54 - Diagrama de estados del formulario de certificado.....	121
Figura 55 - Diagrama de estados del formulario de nómina.....	122
Figura 56 - Diagrama de estados del formulario de permisos retribuidos.....	122
Figura 57 - Estructura de las pruebas.....	130
Figura 58 - Manual de usuario: acceso a la interfaz.....	149
Figura 59 - Manual de usuario: primer ejemplo de proceso.....	150
Figura 60 - Manual de usuario: segundo ejemplo de proceso.....	151
Figura 61 - Manual de usuario: ejemplo de formulario.....	151

Figura 62 - Manual de usuario: cancelar formulario	152
Figura 63 - Manual de usuario: solicitud de información no disponible.....	152
Figura 64 - Primer paso de la configuración de los agentes	153
Figura 65 - Segundo paso de la configuración de los agentes.....	154
Figura 66 - Panel de control de Google Cloud Platform	154
Figura 67 - Configuración de parámetros de red de Oracle Net Manager	156

Índice de Tablas

Tabla 1 - Consulta sobre despliegue del bot.....	32
Tabla 2 - Comparación entre plataformas habilitadoras de bot.....	38
Tabla 3 - Límites y precios de Dialogflow	40
Tabla 4 - Precios de LUIS (Azure).....	40
Tabla 5 - Precios de QnA (Azure).....	41
Tabla 6 - Precio final del despliegue del cloud en Azure.....	42
Tabla 7 - Evaluación tecnología	44
Tabla 8- Análisis de arquitecturas	51
Tabla 9 - Resultados de análisis de LUIS y Dialogflow.....	53
Tabla 10 - Tareas del proyecto	61
Tabla 11 - Precio del equipamiento hardware	65
Tabla 12 - Amortización equipamiento hardware	66
Tabla 13 - Licencia de productos software.....	67
Tabla 14 - Mano de obra en ingeniería.....	67
Tabla 15 - Presupuesto total	68
Tabla 16 - Requisitos funcionales	77
Tabla 17 - Requisitos no funcionales	81
Tabla 18 - Caso de uso: Consulta de funcionalidades	97
Tabla 19 - Caso de uso: Consulta de información de la base de conocimiento	98
Tabla 20 - Caso de uso: Consulta de un proceso.....	99
Tabla 21 - Caso de uso: Consulta un paso exacto de un proceso	99
Tabla 22 - Caso de uso: Consulta de información no disponible	100
Tabla 23 - Caso de uso: Consulta de información sin permisos.....	100
Tabla 24 - Caso de uso: Conversación chatty.....	101
Tabla 25 - Caso de uso: Solicitud completa de vacaciones sin entidades en la expresión.	103
Tabla 26 - Caso de uso: Solicitud completa de vacaciones con entidades	104
Tabla 27 - Caso de uso: Cancelar solicitud de vacaciones.	105
Tabla 28 - Solicitud completa de permiso retribuido (con tipo de día laboral) sin entidades	107

Tabla 29 - Solicitud completa de permiso retribuido de compensación de horas sin entidades en la expresión.....	108
Tabla 30 - Solicitud completa de permiso con entidades en la expresión.....	109
Tabla 31 - Cancelar la solicitud de permiso retribuido	110
Tabla 32 - Solicitud completa de nómina sin entidades en la expresión.....	111
Tabla 33 - Solicitud completa de nómina con entidades en la expresión.....	112
Tabla 34 - Cancelar la solicitud de nómina	112
Tabla 35 - Solicitud completa de certificado de retenciones sin entidades.....	114
Tabla 36 - Solicitud completa de certificado de retenciones con entidades.....	115
Tabla 37 - Cancelar la solicitud de certificado de retenciones.....	115
Tabla 38 - Clases, métodos y funciones del back-end.....	129
Tabla 39 - Consulta de árbol de funcionalidades	131
Tabla 40 - Consulta de información en base de conocimiento.....	131
Tabla 41 - Consulta de un proceso	132
Tabla 42 - Consulta sobre un paso de un proceso sin acceso directo por expresión	132
Tabla 43 - Consulta sobre un paso de un proceso con acceso directo por expresión	133
Tabla 44 - Consulta de información no disponible	133
Tabla 45 - Consulta de información con cambio de disponibilidad durante la sesión	134
Tabla 46 - Consulta con fallo de conexión con la IA	134
Tabla 47 - Consulta de información sin permisos	135
Tabla 48 - Conversación chatty	135
Tabla 49 - Solicitud vacacional sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral	136
Tabla 50 - Solicitud sin entidades en la expresión con periodo superior al máximo y fecha de inicio y retorno en día laboral.....	137
Tabla 51 - Solicitud vacacional sin entidades en la expresión con fecha inicial no laboral	137
Tabla 52 - Solicitud vacacional sin entidades en la expresión con fecha de retorno no laboral	137
Tabla 53 - Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión.....	138

Tabla 54 - Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo superior al máximo en la expresión.....	138
Tabla 55 - Solicitud vacacional con fecha de inicio (laboral) y retorno (no laboral) con periodo inferior o igual al máximo en la expresión.....	139
Tabla 56 - Solicitud vacacional con fecha de inicio (no laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión.....	139
Tabla 57 - Solicitud de permisos sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral	140
Tabla 58 - Solicitud de permisos sin entidades en la expresión con periodo superior al máximo y fecha de inicio y retorno en día laboral	141
Tabla 59 - Solicitud de permisos sin entidades en la expresión con fecha inicial no laboral	141
Tabla 60 - Solicitud de permisos sin entidades en la expresión con fecha de retorno no laboral	141
Tabla 61 - Solicitud de permiso por matrimonio sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral	142
Tabla 62 - Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y periodo temporal inferior o igual a la jornada laboral.....	143
Tabla 63 - Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y periodo temporal superior a la jornada laboral.....	143
Tabla 64 - Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y fecha de retorno anterior a la fecha de inicio.....	143
Tabla 65 - Solicitud con tipo de permiso, fecha de inicio (laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión.....	144
Tabla 66 - Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo superior al máximo en la expresión.....	144
Tabla 67 - Solicitud con tipo de permiso con fecha de inicio (no laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión.....	145
Tabla 68 - Solicitud con tipo de permiso con fecha de inicio (laboral) y retorno (no laboral) con periodo inferior o igual al máximo en la expresión.....	145
Tabla 69 - Solicitud de nómina sin entidades en la expresión con mes y año validos.....	146

Tabla 70 - Solicitud de nómina con entidad de mes y año en la expresión del usuario	146
Tabla 71 - Solicitud de certificado sin entidad en la expresión con año válido	147
Tabla 72 - Solicitud de nómina con entidad de año en la expresión del usuario.....	147
Tabla 73 - Cancelar formulario	148
Tabla 74 - Comando instalación de cx_oracle.....	155
Tabla 75 - Parámetros de configuración del cliente, cx_oracle.....	156

Acrónimos

- NRH – Departamento de Nómina y Recursos Humanos
- RRHH – Recursos Humanos
- API – Application Programming Interface
- IA Services – Servicios de Inteligencia Artificial
- JSON – JavaScript Object Notation (Notación de objeto de JavaScript)
- NLU – Natural Language Understanding (Intérprete de lenguaje natural)
- BC – Base de conocimiento
- LUIS – Language Understanding Intelligent Service
- QnA Maker – Questions and Answer Maker
- GCP – Google Cloud Platform
- MD – Middleware
- CLI – Command-Line Interface
- BD – Base de Datos
- DBA – Database Administrator
- AIOHTTP – Asynchronous Hypertext Transfer Protocol Server

1. Memoria

En este documento se describe de forma detallada el proyecto e incluye todos los conceptos necesarios para su comprensión y las actividades realizadas para llevarlo a cabo.

1.1 Introducción

El proyecto se basa en el desarrollo de un bot conversacional o “*chatbot*”. La herramienta es un asistente virtual web que proporciona acceso a datos y servicios mediante una interfaz de conversación. El término inglés “*bot*” hace referencia a un sistema automatizado que imita el comportamiento humano y que, por lo general, realiza tareas repetitivas. Están implementados en muchos campos como el rastreo de información, análisis de perfiles en redes sociales o asistencia virtual.

Los chatbots priman la experiencia del usuario por encima de cualquier otro aspecto. Son herramienta con un grado de usabilidad muy alto. El principal objetivo es que no se diferencie entre un agente humano o virtual. Al igual que en la mayoría de asistentes, el usuario puede conversar mediante texto o voz y este, a su vez, le puede responder de la misma forma.

Las interacciones se pueden agrupar en tres diferentes tipos. En primer lugar, los formularios recopilan información mediante preguntas para ofrecer un servicio, por ejemplo, realizar una solicitud vacacional. Los procesos que están compuestos por sucesivas interfaces que guían al usuario durante una configuración o explicación. Normalmente, cada paso está formado por un texto descriptivo, una figura y varias opciones relacionadas con la interfaz actual. Por último, el asistente virtual tiene acceso a un repositorio de preguntas y respuestas sobre información corporativa. Esto se conoce como una base de conocimiento.

Sin embargo, las principales funcionalidades de los asistentes conversacionales apenas se diferencian de los otros tipos de software: lectura y escritura en bases de datos, uso de APIs para la comunicación con servicios externos y procesamiento de los datos del usuario. Lo que realmente define a un chatbot son los mecanismos de interpretación de lenguaje natural. Estos son servicios cognitivos que dada una expresión del usuario determinan cuál es su intención. Esta funcionalidad permite que el bot adquiera inteligencia y responda adecuadamente al mensaje.

El diseño es muy similar al de las aplicaciones web modernas. Contienen elementos multimedia que enriquecen el flujo de la conversación. Los dos más utilizados en este proyecto son los botones y las imágenes. No obstante, es posible diseñar tarjetas adaptativas donde se muestren vídeos, sonidos o incluso elementos de aplicaciones externas como calendarios o correos electrónicos. Como es lógico, los elementos varían considerablemente en función de su tipo y finalidad.

La arquitectura de la herramienta desarrollada en este trabajo está formada por tres capas:

- La capa del cliente o front-end es en la que se efectúa la comunicación entre el usuario y el sistema. Se encarga de procesar la expresión y dar formato a la respuesta del sistema.
- La capa middleware o back-end es donde reside la mayor parte de la lógica del bot. Se encarga de gestionar el flujo de conversación y establecer la conexión con los servicios de interpretación de lenguaje natural y la plataforma corporativa. También realiza un registro de la actividad del bot.
- Para la capa de capacidades cognitivas, se contrata el servicio externo Dialogflow. Las principales funcionalidades de este servicio son identificar las intenciones del usuario y entidades dentro de la expresión del usuario, extraer datos y estadísticas de las conversaciones.

1.1.1 Enfoque desde una perspectiva empresarial

Este trabajo fin de master describe el proyecto “*MilenaBot*” de Seresco S.A. Esta es una empresa de origen asturiano fundada en 1969. Su campo de negocio es el desarrollo de soluciones de software y la prestación de servicios de tipo TIC. Cuenta con sedes nacionales en Madrid, Barcelona, Oviedo y Vigo e internacionales: Lisboa, Quito, Ecuador y Costa Rica. Seresco es reconocida por su exitosa trayectoria en el desarrollo de proyectos en distintos campos. Los valores de la empresa destacan por ofrecer soluciones con alta calidad y eficiencia.

El servicio Nearshore desarrolla y mantiene aplicaciones a medida del usuario en los sectores de la industria y las administraciones públicas. Cuenta con equipos profesionales con alta cualificación en entornos J2EE y .NET. Actualmente, han desarrollado la aplicación

“Vayavaca” con el objetivo de digitalizar el mercado ganadero. Otro ejemplo de éxito es “Cultiva Decisiones”. Esta es una plataforma desarrollada para la toma de decisiones en el ámbito de la viticultura. Utiliza la inteligencia artificial para ofrecer agricultura de precisión con el fin de mejorar la calidad de la uva y el proceso de producción.

Sin embargo, donde destaca Seresco es en su larga trayectoria en el desarrollo de aplicaciones software para nómina y recursos humanos. Actualmente, el volumen de negocio de este departamento es muy alto llegando a liderar el mercado ibérico con 2,5 millones de nóminas procesadas al año. Todo ello es gracias a Milena una potente y flexible plataforma que es utilizada por más de 1.000 empresas y un total de 180.000 potenciales usuarios.

El proyecto “Milena Chatbot” o “MilenaBot” surge con el objetivo de cubrir las necesidades de soporte y asistencia de la plataforma. Sin embargo, Milena es una herramienta compleja. El chatbot pretende ayudar al personal de nueva incorporación en su uso, a la vez que sirve de guía al usuario en la plataforma.

Un segundo enfoque es ofrecer la capacidad de solicitar servicios de manera rápida y sencilla. P. ej. que el usuario acceda a pedir vacaciones expresándose textualmente o mediante voz: “Quiero pedir vacaciones del 5 de mayo al 8 de mayo”. Esto permite aumentar la usabilidad y rapidez de la plataforma. Debe predominar la sencillez y facilidad de acceso al servicio.

La tercera necesidad es solucionar las dudas que el usuario de una empresa tiene respecto a su convenio o relación laboral con la entidad. P. ej. explicarle al usuario su nómina o cómo debe proceder si ha enfermado. El objetivo es que el chatbot tenga la funcionalidad de cubrir preguntas sobre temas relacionados con nómina y recursos humanos.

Todos estos aspectos liberan al personal responsable de tareas repetitivas y por lo general, comunes a todos los empleados.

Seresco actualmente dispone de un blog público, masquenominas.com, que es la principal fuente de información en este ámbito. El departamento de Nómina y Recursos Humanos (NRH) es el principal encargado de gestionar la información del chatbot.

1.2 Objetivos y motivación

El objetivo es desarrollar un producto que contenga todas la claves que permiten que estas herramientas tengan éxito y sean tanto efectivas como eficientes.

1. El aspecto más importante es la usabilidad. Las guías de buenas prácticas establecen que una herramienta cumple satisfactoriamente si el usuario es capaz de utilizarla sin necesidad de una explicación previa.
2. El segundo objetivo es la accesibilidad. Desarrollar una herramienta que centralice la información respecto a temas corporativos y ofrezca un canal de distribución de acceso rápido.
3. El asistente virtual debe procesar la expresión del usuario tan rápido como sea posible. El tiempo de respuesta es un aspecto crítico.
4. El servicio de interpretación de lenguaje natural debe estar correctamente entrenado para que reduzca el número de intenciones mal interpretadas y ofrezca un servicio robusto.
5. El bot debe reducir el número de pasos que necesita para ofrecer un servicio mediante la identificación de parámetros dentro de la expresión del usuario.

Además, la herramienta tiene una motivación especial, debe cumplir con unos estándares de calidad debido a que este producto tiene un propósito comercial. En la figura 1, se muestra el icono y el logo de la herramienta. Por último, el lector debe conocer de antemano que este proyecto está formado por un equipo de trabajo que establece todos los requisitos y procedimientos para el correcto desarrollo de la herramienta.



Figura 1 - Logo de MilenaBot

1.3 Demanda actual de los bots en RRHH

En este apartado, se estudia cómo las empresas han automatizado los procesos de gestión de recursos humanos. El caso más significativo es el de la compañía CEPSA que ha mostrado públicamente su chatbot, “MAX BOT”. Resuelve preguntas relacionadas con licencias y permisos, seguros de salud, préstamos, teletrabajo, días de vacaciones. Su base de conocimiento desarrollada con la tecnología de inteligencia artificial IBM Watson cuenta con 5.000 preguntas distintas. El proyecto surgió a raíz del gran volumen de solicitudes recibidas por los empleados. La información de manera extendida se encuentra en el enlace [1].

Otra tendencia actual es automatizar el proceso de selección mediante una interacción directa con los candidatos y clasificarlos según sus respuestas. Arnie es un proyecto de la empresa W3 Digital Agency, que realiza preguntas contextuales basadas en los requisitos del puesto. Según un reciente artículo [2], mejoran el proceso de selección porque envían feedbacks inmediatos con información de seguimiento. En términos laborables, reducen en un 80% el tiempo de investigación y verificación de los datos, procedimiento también conocido como “*screening*”.

El estudio del enlace [3] de la compañía Sodexo realiza recomendaciones sobre cómo deben actuar los chatbots. Destaca la incorporación de guías de iniciación para el personal de nueva incorporación que sustituyan la lectura de documentos, los cursos de formación con sus preguntas evaluables y cuestionarios que permitan al empleado realizar sugerencias y opiniones sobre su puesto laboral. Su departamento de recursos humanos recomienda utilizar el canal como distribución de eventos y novedades personalizados relacionados con estos aspectos.

El estudio del enlace [4] sobre las tendencias de los chatbots para 2020 de la compañía Business Insider pronostican un incremento de un 80% en el ámbito laboral. En resumen, el 40% de las grandes empresas probablemente implementen un asistente conversacional para mejorar la productividad de los empleados. El objetivo principal es reducir las tareas de gestión y el coste asociado que suponen.

1.4 Fundamentos del procesamiento de lenguaje

La amplia mayoría de las plataformas que disponen de servicios de interpretación de lenguaje natural, utilizan una terminología similar. A continuación, todos los conceptos definidos son independientes del proveedor. Este apartado explica las características y diferencias de los dos principales servicios en el entorno de los chatbots: NLU (Natural Language Understanding) y Base de conocimiento.

En ambos casos, el elemento de nivel superior es el agente. Realiza la conversión de texto y audio a datos estructurados en formato JSON (JavaScript Object Notation) que son gestionados por el servicio de inteligencia artificial. El agente está formado por intents que es una clasificación de la expresión del usuario. En términos coloquiales, un intent determina cuál es el propósito del usuario.

1.4.1 Concepto: Intérprete de lenguaje natural (NLU)

NLU corresponde a las iniciales de Natural Language Understanding. Este servicio está formado principalmente por dos funcionalidades: la clasificación de la expresión y la identificación de entidades.

Cada intent está formado por múltiples frases que son utilizadas para el entrenamiento de la herramienta. Las frases son las expresiones probables que el usuario introduce para referirse a una acción. Por ejemplo, se necesita identificar expresiones con el objetivo de teletrabajar. El desarrollador debe entrenar la herramienta mediante frases como: *“Quiero teletrabajar”* y *“¿Cómo puedo teletrabajar?”*. El usuario llegado el momento, introduce *“Activar el teletrabajo”*. A pesar de que la herramienta no contuviera esta frase explícitamente, identifica adecuadamente dicho intent. Sin embargo, la identificación no se evalúa en términos absolutos, si no que cada expresión incluye un umbral o coeficiente de clasificación. Esto indica el grado de identificación, siendo el valor máximo la propia frase de entrenamiento. La terminología que se utiliza cuando supera o no el umbral es *“matched”* o *“no-matched”*

Los parámetros son los valores que se extraen de la expresión del usuario. Cada parámetro se clasifica por el tipo, también llamado entidad. Generalmente, todas las plataformas incorporan entidades definidas por defecto, por ejemplo: fecha y hora. Sin embargo, el desarrollador puede definir nuevos tipos. Al igual que en la definición de intents, necesita un

entrenamiento, pero en este caso mediante sinónimos. Situando esto en un caso particular, el desarrollador define una entidad llamada “*Departamentos*”, un parámetro: “*CYS*” y los sinónimos: “*Consultoría y Software*”, “*Consultoría*” y “*Software*”. De modo que si el usuario introduce:

“*Quiero consultar el organigrama de CYS*”

“*¿Quién es el responsable de Software?*”

Identificaría tanto “*CYS*” en el primer ejemplo como “*Software*” en el segundo, como entidades de “*Departamento*” del parámetros “*CYS*”.

A modo de resumen, los servicios NLU interpretan los objetivos del usuario (intenciones) y sintetizan la información de valor de las expresiones (entidades).

1.4.2 Concepto: Base de conocimiento

El concepto de base de conocimiento es similar al de algunos ámbitos de la informática. Está formado por preguntas y respuestas. El mecanismo de identificación de la pregunta es igual al de los NLU. A diferencia de este servicio, la respuesta no es un intent, es información que el propio usuario ya puede gestionar. Las bases de conocimiento están orientadas a almacenar preguntas frecuentes de manera que permitan guiar al usuario con preguntas relacionadas a su consulta. Para ello, introducen el concepto de contexto. Es similar al de una conversación natural entre personas, si una dice “*sabor vainilla*”, es necesario que especifique a qué se está refiriendo para comprender su expresión. La gestión de contextos permite desarrollar flujos de conversación en los que se minimiza la expresión que el usuario necesita introducir para acceder a información.

Algunos servicios como la API.ai de Dialogflow e IBM Watson ofrecen servicios híbridos en los que se integran la base de conocimiento y el NLU.

1.5 Estudio de las principales plataformas

Este apartado incluye un estudio de las actuales plataformas con sus respectivos servicios de API para la comprensión de lenguaje natural. Se analiza el proceso de desarrollo mediante la plataforma y sus funcionalidades disponibles con su respectivo coste asociado para evaluar cual es la solución óptima para el proyecto. En este apartado, no se evalúa el rendimiento de los servicios de interpretación de lenguaje NLU.

Las plataformas Dialogflow, IBM Watson Assistant y Amazon Lex ofrecen funcionalidades similares, por lo que se realiza una explicación más exhaustiva de esta primera tecnología con el fin de ofrecer una visión general de las capacidades de estas plataformas.

1.5.1 Plataforma: Dialogflow

Dialogflow es una plataforma desarrollada por Google Cloud con un único servicio de comprensión del lenguaje natural que permite el diseño de interfaces conversacionales de voz y texto y su integración en múltiples canales. La plataforma proporciona los mecanismos y procedimientos para desarrollar el producto en la nube, incluido la parte front-end y back-end. La figura 2 contiene un diagrama con todas las tareas del desarrollador que se muestran en los sucesivos apartados. Toda la documentación se encuentra en el enlace [5].

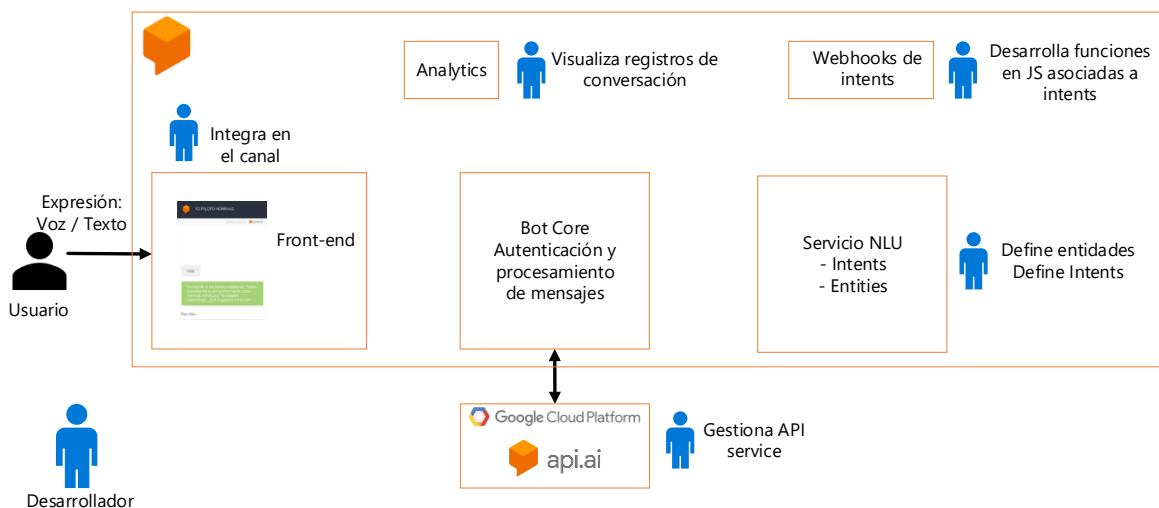


Figura 2 - Tareas del desarrollador en Dialogflow

1.5.1.1 Consulta de información en Dialogflow

En este apartado, se implementa un conjunto de preguntas y respuestas frecuentes. El objetivo es desarrollar una base de conocimiento. Para ello, se utiliza un ejemplo del citado blog sobre la gestión de personal y nóminas.

Para diseñar adecuadamente un flujo de conversación según la documentación del enlace [5], es necesario que el desarrollador se sitúe en la peor situación. Esto es que el usuario desconozca toda la información disponible. Es recomendable que en el mensaje de saludo inicial, el usuario sea informado sobre que está utilizando. Seguidamente establezca sus expectativas y a continuación, le sugiera información para que el usuario tome el control de la conversación. En este caso, este mensaje es respuesta de un saludo previo del usuario. Por lo general, Dialogflow es reactivo y responde a actividades del usuario. Para que el bot notifique una vez establecida la conversación, es necesario establecer un evento personalizado, aspecto que se comenta más adelante.

La figura 3 muestra el diseño de un flujo de conversación para que se comprenda el funcionamiento de la base de conocimiento en Dialogflow. Una vez que el usuario saluda, es guiado a que introduzca “*Novedades Legislativas*”. El bot responde indicándole información de tres años disponibles. Aquí, se incorpora el primer punto de decisión con el que se bifurca el flujo de conversación principal. Esto es gracias a la incorporación de un contexto de salida al intent “*I1-NovedadesLegislativas*” que a su vez es contexto de entrada para “*I2-NovedadesMarzo2019*”, “*I2-NovedadesEnero2019*” y “*I2-NovedadesEnero2020*”. Si el usuario, una vez iniciada la conversación introdujese directamente “*Enero 2020*”, Dialogflow emitiría el mensaje de error por defecto.

El concepto de contexto es similar al definido en el apartado 1.4.2. La gestión de contextos es distinta por parte de Dialogflow. Limita las posibles expresiones que puede introducir el usuario. De este modo, se reduce el número de posibles colisiones.

Partiendo de la primera elección “*Marzo 2019*”, el bot sugiere ampliar la información sobre los dos temas que cita la respuesta. No obstante, en esta bifurcación no se configura ningún contexto. De esta forma, se permite que el usuario pueda acceder en cualquier turno a la información sin obligarlo a recorrer este flujo de conversación.

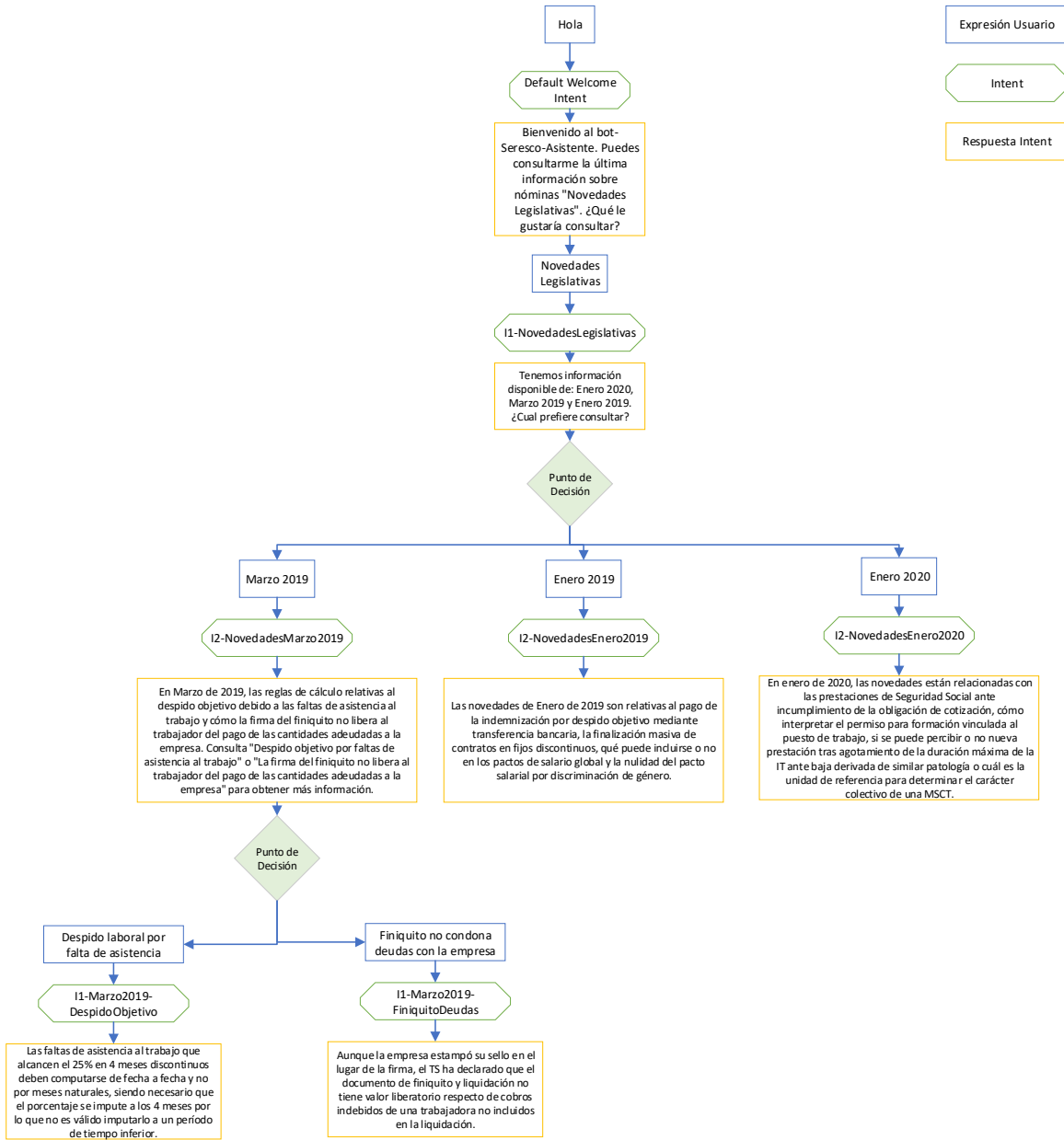


Figura 3 – Flujo de conversación de consulta de información en Dialogflow

La figura 4 muestra el comportamiento real del bot utilizando la interfaz web de la plataforma. Responde adecuadamente a las expresiones del usuario siguiendo el flujo de conversación citado anteriormente. En la última petición, se ha introducido únicamente “Despido objetivo”, el bot es capaz de asignar al intent “I1-Marzo2019-DespidoObjetivo”, sin embargo, en situaciones en las que estas palabras aparezcan reflejadas en otros intents, se podrían generar colisiones.



Figura 4 - Ejemplo de solicitud de información con el piloto de Dialogflow

La figura 5 contempla la posibilidad de cancelar la petición, permitiendo al usuario salir del flujo de conversación actual. Además, se ha configurado un intent “*fallback*” asociado al contexto específicamente, de manera que, si el usuario introduce una expresión por la que no hay asignación de intent, el bot emita un mensaje destacando que es lo que puede introducir. Esta funcionalidad evita lo que se conoce en el ámbito de chatbots como “*bot terco*”.



Figura 5 - Respuestas de fallback en Dialogflow

Es posible que el usuario ya conozca de antemano que quiere consultar, de modo que también se debe permitir acceder directamente a la información sin necesidad de recorrer la conversación antes mostrada. Esto se detalla en la figura 6.

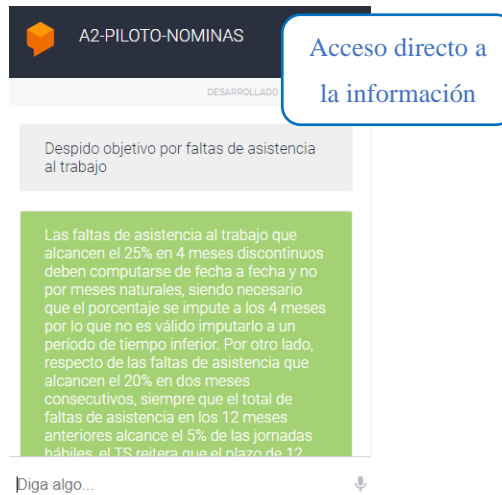


Figura 6 - Consulta directa en Dialogflow

1.5.1.2 Solicitud en Dialogflow

La figura 7 muestra el procedimiento de identificación de intent. En este caso, el objetivo es realizar una solicitud. En primer lugar, se ha definido la entidad, “*EIDepartamentos*”, que identifica el departamento dentro de la expresión del usuario. Por ejemplo, si introduce “*Software*”, automáticamente asocia a “*CYS*” que a su vez es un parámetro de entrada de la función de JavaScript. Estas funciones están restringidas a las APIs de Google en la capa gratuita.

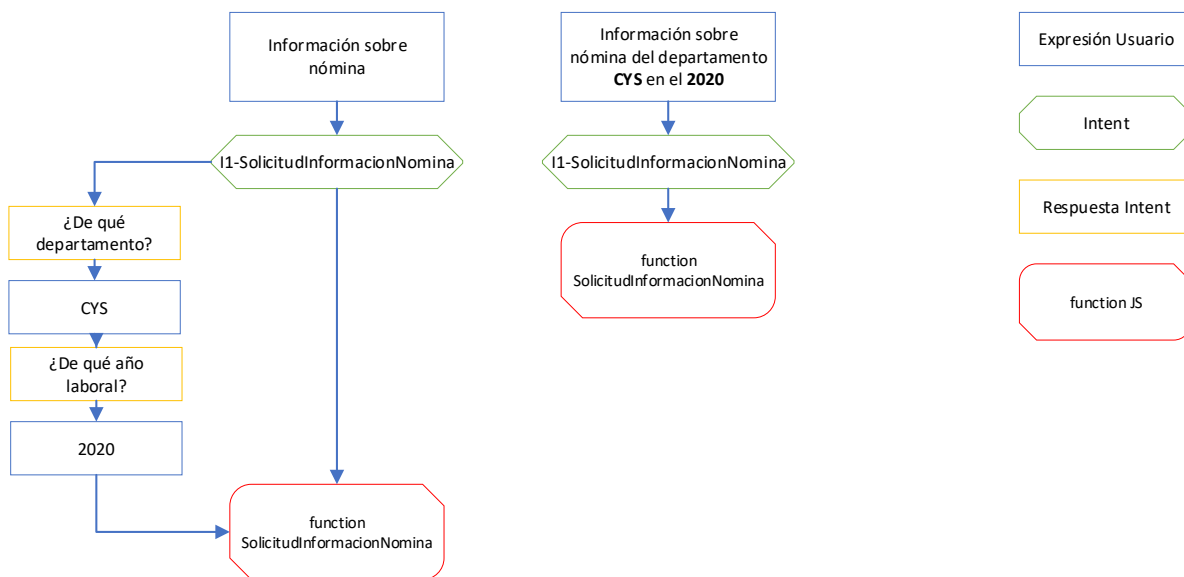


Figura 7 - Flujo de conversación de solicitud de información en Dialogflow

La figura 8 muestra la respuesta ante la solicitud incluyendo todos los parámetros necesarios en la expresión. Es posible que el usuario no inserte todos los parámetros necesarios en un único turno, de modo que se incluyen preguntas que se realizan automáticamente si no los detectan.



Figura 8 - Comportamiento piloto Dialogflow ante solicitud

1.5.1.3 Estadísticas y entrenamiento en Dialogflow

Dialogflow incluye estadísticas sobre el comportamiento del usuario. Como se muestra en la figura 9, la plataforma permite visualizar el número de sesiones con sus consultas por sesión y la tendencia de las expresiones de los usuarios.

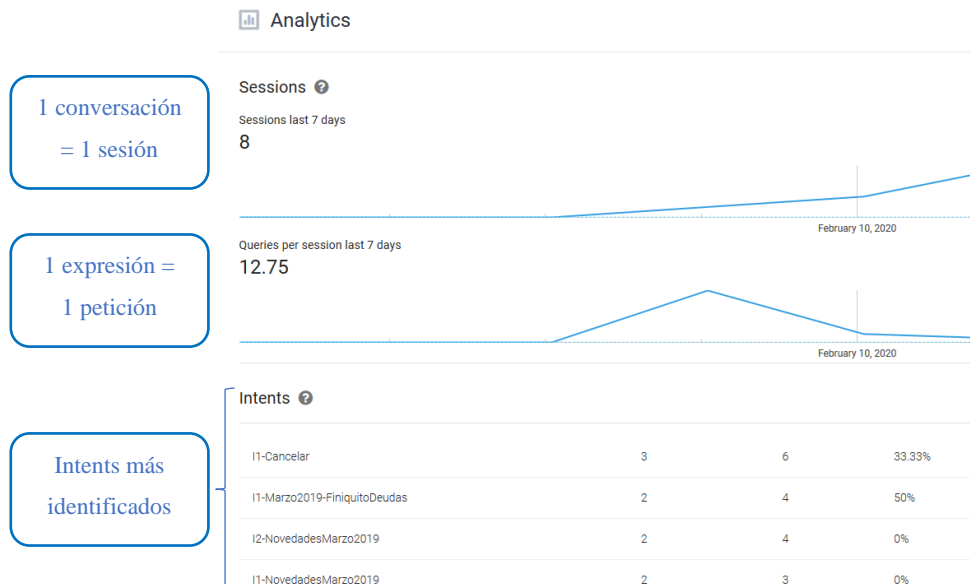


Figura 9 - Estadísticas de comportamiento del usuario de Dialogflow

También incluye el siguiente gráfico reflejado en la figura 10. Resume visualmente las rutas de conversación que han tomado los usuarios durante la interacción con el agente. Calcula el porcentaje de coincidencia con el intent, el número de solicitudes a cada uno y el porcentaje de salida. Esta herramienta permite estudiar la tendencia actual de las solicitudes.

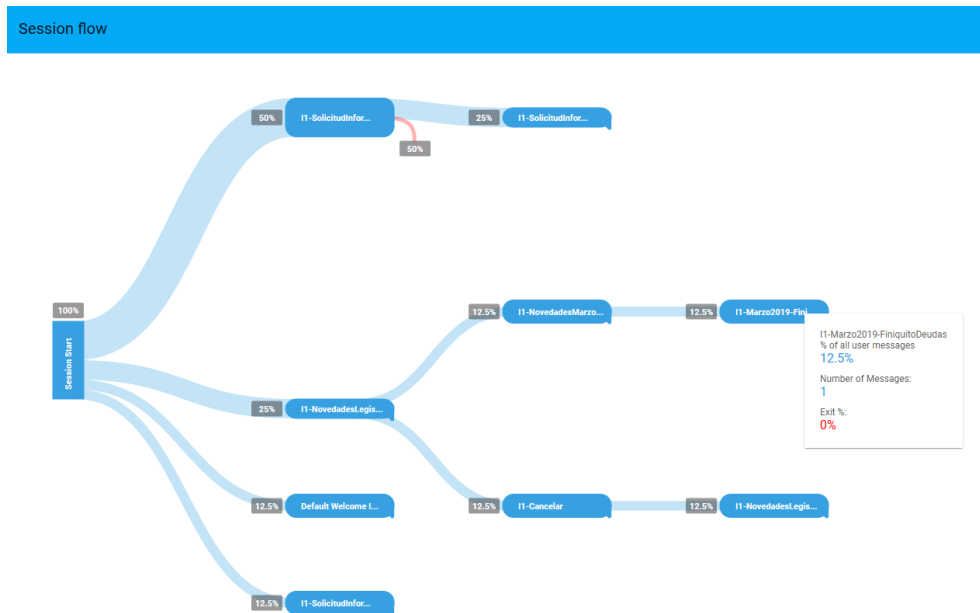


Figura 10 - Diagrama de flujo de Dialogflow

Sin embargo, esta herramienta no permite consultar con detalle qué expresiones del usuario han producido error. Para ello, es necesario acceder al historial de conversación, donde se describe la interacción entre el agente y el usuario junto a los errores producidos. Para comprobar esta funcionalidad, se introduce en el chat “Consulta sobre sueldo”, porque la palabra “sueldo” no está en ninguna de las frases de entrenamiento de “I1-SolicitudInformacionNomina”, de modo que se genera un error con su correspondiente respuesta por defecto. Esto se muestra en la figura 11.

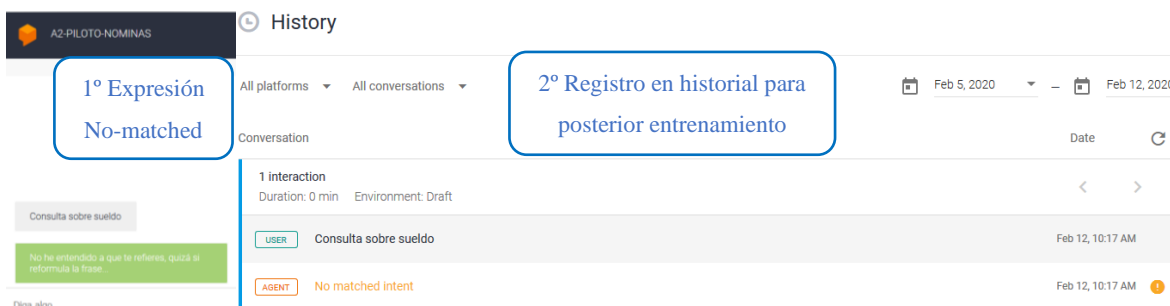


Figura 11 - Historial de Dialogflow

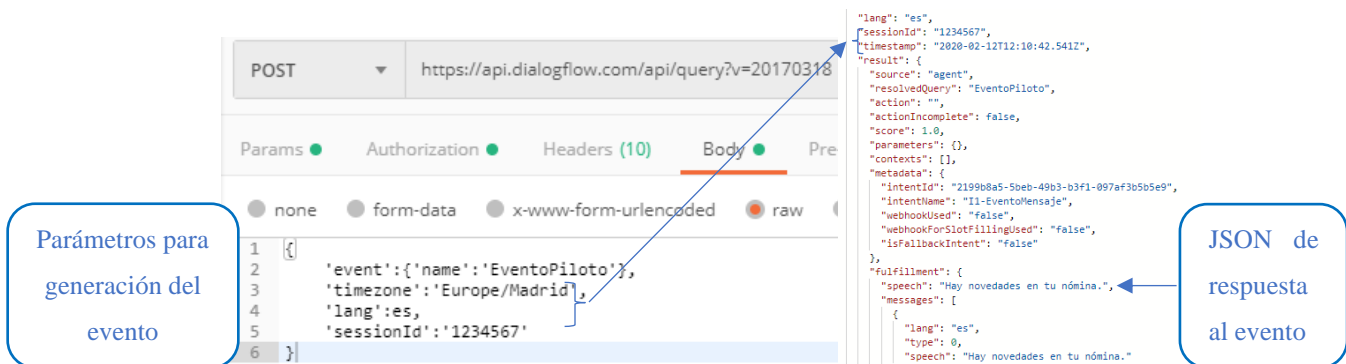
Para solucionar este error se debe entrenar al agente. Dialogflow permite el entrenamiento mediante registros internos y externos. En este caso, se utiliza el propio registro generado por la expresión no-matched de la figura 11. Una vez incorporado y entrenado, si se introduce nuevamente la expresión como en la figura 12, el bot responde adecuadamente. Este proceso es automático y tiene efecto directo sobre el chatbot en producción.



Figura 12 - Ejemplo de entrenamiento de Dialogflow

1.5.1.4 Eventos en Dialogflow

La siguiente funcionalidad de estudio son los eventos. Inicialmente, se genera un intent, “II-EventoMensaje”, con el evento “EventoPiloto” asociado. Responde con “Hay novedades en tu nómina.” como se muestra en la figura 13. Mediante la herramienta POSTMAN, se genera una petición POST con el token asociado al proyecto. En el caso de que se precise que el mensaje de bienvenida sea reactivo, es obligatorio que el front-end envíe un evento similar.



Parámetros para generación del evento

JSON de respuesta al evento

Figura 13 - Generación de evento mediante POST en Dialogflow

1.5.1.5 Multilinguaje en Dialogflow

Dialogflow permite configurar varios idiomas dentro de un mismo agente. Sin embargo, es obligatorio seleccionar uno como principal. La plataforma permite hasta 14 lenguajes que clasifica en idiomas raíz e idiomas de configuración regional, por ejemplo, español y español latino. Una vez añadido en la configuración, se definen nuevos intents con sus respectivas frases de entrenamiento y respuestas. Un ejemplo de adición de idioma se muestra en la figura 14.

Para identificar el lenguaje en las funciones del fulfillment se debe acceder a la propiedad `agent.locale`, que tiene asignado un código único.

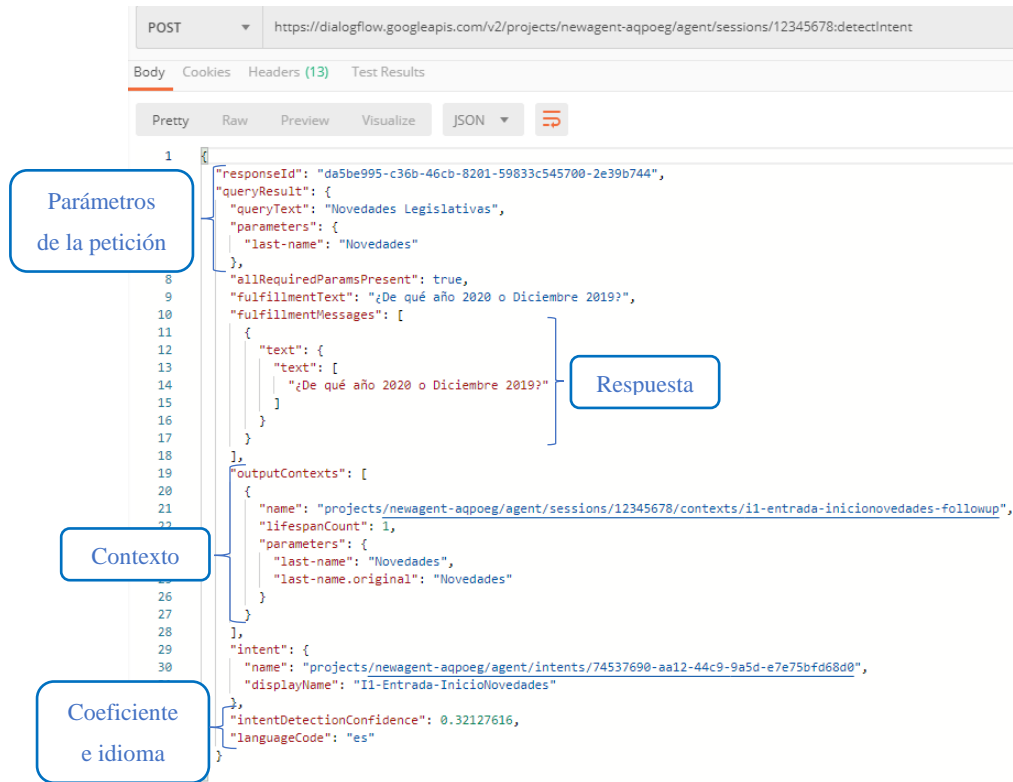


Figura 14 - Piloto Dialogflow interacción en inglés

1.5.1.6 API.ai de Google Cloud Platform (Dialogflow API)

La plataforma permite interactuar a través de una API configurada en la plataforma cloud de Google. Esto permite la posibilidad de utilizar Dialogflow como un intérprete de lenguaje natural de manera similar a LUIS de Microsoft. Por otro lado, Python dispone de la librería `Dialogflow_v2` para la creación de conexiones.

Para realizar peticiones a la API, se utiliza la aplicación POSTMAN. Las instrucciones para habilitar esta API se explican en el apartado 5.1. Este servicio utiliza el método de autenticación, OAuth 2.0. La figura 15 es un ejemplo de detección de intent, donde se señala información sobre la acción, los parámetros y la respuesta.



```

POST https://dialogflow.googleapis.com/v2/projects/newagent-aqpoeg/agent/sessions/12345678:detectIntent

Body
Pretty Raw Preview Visualize JSON
1
{
  "responseId": "da5be995-c36b-46cb-8201-59833c545700-2e39b744",
  "queryResult": {
    "queryText": "Novedades Legislativas",
    "parameters": {
      "last-name": "Novedades"
    }
  },
  "allRequiredParamsPresent": true,
  "fulfillmentText": "¿De qué año 2020 o Diciembre 2019?",
  "fulfillmentMessages": [
    {
      "text": {
        "text": [
          "¿De qué año 2020 o Diciembre 2019?"
        ]
      }
    }
  ],
  "outputContexts": [
    {
      "name": "projects/newagent-aqpoeg/agent/sessions/12345678/contexts/i1-entrada-inicionovedades-followup",
      "lifespanCount": 1,
      "parameters": {
        "last-name": "Novedades",
        "last-name.original": "Novedades"
      }
    }
  ],
  "intent": {
    "name": "projects/newagent-aqpoeg/agent/intents/74537690-aa12-44c9-9a5d-e7e75bfd68d0",
    "displayName": "I1-Entrada-InicioNovedades"
  },
  "intentDetectionConfidence": 0.32127616,
  "languageCode": "es"
}
  
```

Figura 15 - Primera llamada a la API de Dialogflow

1.5.1.7 Integración en canales en Dialogflow

El número de canales disponibles son 11: kik, Line, Skype, Cisco Spark, Telegram, Cisco Tropo, Twilio, Twitter, Viber y Web Chat. Se puede integrar en otros sistemas con un cliente de Dialogflow mediante la API.ai.

Por otra parte, la comunicación verbal con el usuario depende de la integración que se esté utilizando. Si se implementa en los canales disponibles, la propia plataforma realiza la conversión. Sin embargo, también es posible enviar la entrada de audio en una solicitud de intent mediante la API. En ese caso la aplicación tiene que enviar el audio codificado en Base64 en el campo content de la solicitud. Por otra parte, Google dispone de la API Cloud Text-to-Speech para generar respuestas de voz. Este servicio es independiente a Dialogflow. Las respuestas con elementos multimedia dependen del canal. Por ejemplo, en Telegram es posible integrar imágenes mientras que en un entorno web esto no es posible.

1.5.2 Plataforma: Azure Bot Service

Azure Bot Service es el servicio que dispone el cloud de Azure. Tiene un proceso de desarrollo totalmente distinto a las demás plataformas actualmente disponibles. Es la solución con mayor libertad de desarrollo, pero también con mayor complejidad. El enlace [6] contiene la documentación de esta tecnología.

Azure Bot Service dispone de una librería abierta, Bot Framework que consiste en un entorno de desarrollo completo que dispone de herramientas como un emulador de front-end y plantillas con funcionalidades específicas. Estas bibliotecas establecen el diseño de la arquitectura del núcleo, o back-end, del bot. Está disponible para los lenguajes C#, JavaScript, Typescript y Python.

A diferencia de Dialogflow, solo se puede desarrollar un asistente informativo. Si el desarrollador no desea más que un bot con una base de conocimiento asociada, puede desplegarlo sin necesidad de desarrollar software. Sin embargo, el proceso de desarrollo está detalladamente especificado y documentado por Microsoft en el enlace [6]. El primer paso es descargar la plantilla en el lenguaje preferido. Esta plantilla incluye un servidor de tipo asíncrono, aiohttp, que funciona como middleware de mensajería entre el front-end y los servicios de interpretación de lenguaje natural. Esto permite el desarrollo del chatbot en local y testarlo con el emulador. Bot Framework Emulator es una aplicación independiente para la depuración del bot. Permite visualizar los mensajes enviados y recibidos antes de ser procesados. Se puede ejecutar localmente o en remoto mediante una tunelización con ngrok. La explicación de cada una de las partes se encuentra en el apartado 3.2.2.

Una vez completada todas las fases de desarrollo, Azure recomienda la publicación en el cloud. Sin embargo, cabe la posibilidad de despliegue en servidores propios. En las preguntas frecuentes del enlace [6], se encuentra la información que se cita en la tabla 1.

¿Puedo hospedar el bot en mis propios servidores?

Sí. El bot puede hospedarse en cualquier lugar de Internet. En sus propios servidores, en Azure o en cualquier otro centro de datos. El único requisito es que el bot debe exponer un punto de conexión HTTPS accesible públicamente.

Tabla 1 - Consulta sobre despliegue del bot

Esta segunda opción obliga a disponer de una dirección pública si se desea conectar con los canales: Facebook, Slack, Microsoft Teams, Telegram, Kik, Skype, Messenger, mensajes de texto, Twilio, Skype, Cortana. Sin embargo, esta opción ofrece ciertas ventajas. Por ejemplo, la gestión de los datos y el almacenamiento de los registros de conversación.

Por otra parte, el cloud dispone de servicios que están relacionados con el asistente: Azure AD para la identificación y control de acceso al bot, Application Insights para la visualización de registros y estadísticas asociadas al usuario y Azure Blob para el almacenamiento.

Finalmente, a diferencia de las demás plataformas, el cloud dispone de los dos servicios de interpretación de lenguaje natural: LUIS (NLU) y QnA Maker (Base de conocimiento). Sin embargo, y dado el amplio abanico de posibilidades que ofrece Bot Framework, se pueden plantear soluciones en las que no se utilicen estos servicios cognitivos.

1.5.2.1 LUIS (NLU)

Language Understanding (LUIS) es un servicio API en la nube de interpretación de lenguaje natural (NLU) que aplica la inteligencia artificial para predecir el significado de la expresión y extraer parámetros. LUIS sirve para cualquier aplicación conversacional con un usuario, Azure no restringe esta herramienta solo a los asistentes virtuales.

La figura 16 muestra la configuración de un intent en LUIS. El procedimiento es similar al ya descrito en Dialogflow ya que el desarrollador debe incluir frases de entrenamiento y asociar los parámetros que desee a la entidad correspondiente. Además, permite realizar pruebas en la propia plataforma de LUIS. La documentación de este apartado se recoge en el enlace [7].

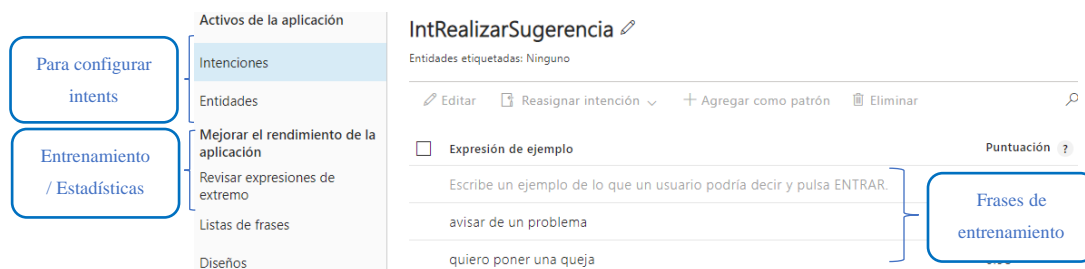


Figura 16 - Configuración de un intent en LUIS

1.5.2.2 QnA Maker (Base de Conocimiento)

QnA Maker (Questions and Answers Maker) es la base de conocimiento de Azure. Cada agente está compuesto por preguntas y respuestas. Incorpora contextos, pero a diferencia de Dialogflow, no restringe la expresión del usuario a las preguntas o expresiones asociadas a ese contexto. El objetivo es responder al usuario con preguntas relacionadas. Por ejemplo, como en la figura 17 se expresa “*Cambio smi*”, la respuesta incluye información como “¿Cuál es el incremento respecto al SMI de 2019?” e “*Reparto del SMI en función del número de pagas*”, para que sea mostrado al usuario por pantalla, pero no restringe la siguiente interacción a una de estas opciones. QnA también permite adjuntar preguntas mediante un documento Word. Toda la documentación se encuentra en el enlace [8].

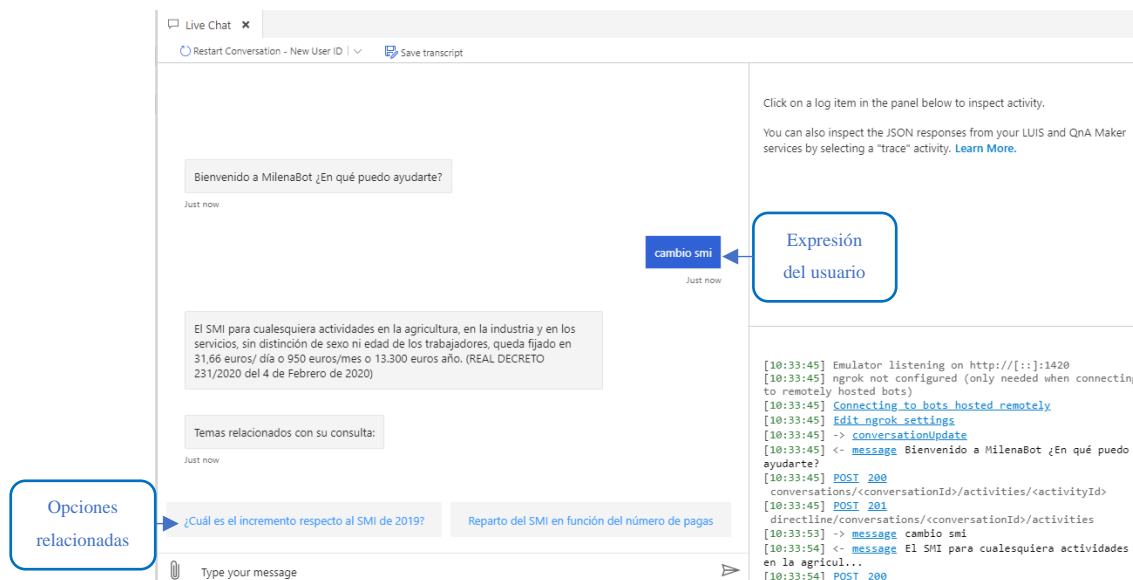


Figura 17 - Prueba de emulador con server con base de conocimiento

1.5.3 Plataforma: Amazon Lex

La plataforma Amazon Lex tiene una estructura de desarrollo similar a la Dialogflow e IBM Watson con integración directa de sus funciones, AWS Lambda. Permite generar rápidamente bots complejos desde altos niveles de abstracción integrables en los canales Facebook, Slack y Twilio SMS. Entre sus principales características, destaca el procesamiento de peticiones de voz que permite la integración en centros de llamadas.

El enlace [9] contiene toda la documentación de esta tecnología. Sin embargo, Amazon Lex presenta una desventaja ante sus competidores, ya que solo soporta inglés. Esto es un impedimento para el proyecto actual por lo que se descarta utilizar esta plataforma.

1.5.4 Plataforma: IBM Watson Assistant

La plataforma de IBM Watson Assistant presenta una estructura de desarrollo similar a Dialogflow. El servicio está disponible para 13 idiomas, entre ellos, el español. Cada nodo está formado por un intent. Los intents pueden ser por definidos, “*General_CursoPRL*” o por defecto, “*Bienvenido*”. Este es el único aspecto en el que se diferencia de Dialogflow. La figura 18 muestra un ejemplo de un flujo de conversación con el intent “*INFORMACION_CURSOS*” asociado al primer nodo.

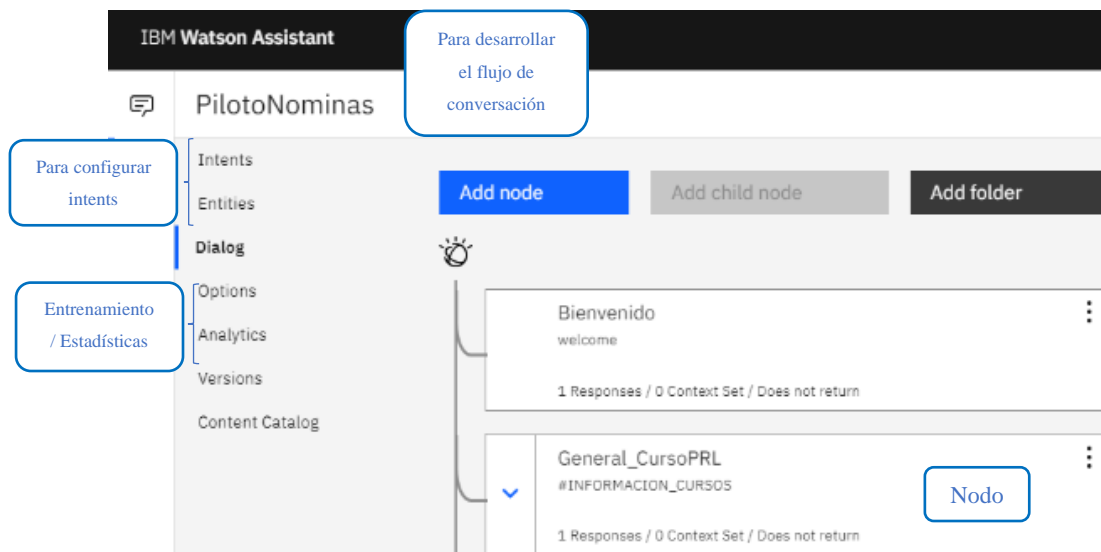


Figura 18 - Ejemplo de configuración de un agente de IBM Watson

En términos económicos, Watson Assistant ofrece tres capas disponibles. La opción Lite es la opción básica que tiene capacidades limitadas y restringe la integración en canales. El servicio es similar a Dialogflow pero no cuenta con una capa gratuita que no genere coste durante el desarrollo. El precio de los servicios de IA es un parámetro excluyente en la elección de la plataforma. Además, la empresa ni utiliza servicios del cloud de IBM, ni dispone de una API para la detección de intents.

1.5.5 Análisis de plataformas

Una vez analizadas las principales opciones, se debe comprender la diferencia en el paradigma de desarrollo de asistentes conversacionales. Como se muestra en la figura 19, Dialogflow e IBM Watson ofrecen la posibilidad de desarrollar la herramienta al completo mediante su PaaS.

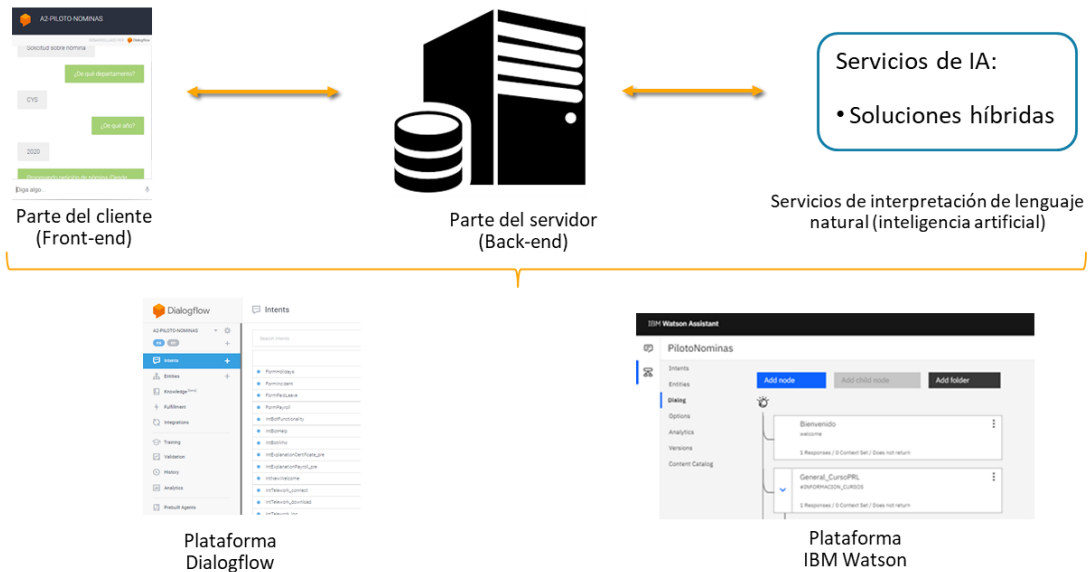


Figura 19 - Estructura de la plataforma de Dialogflow e IBM Watson.

El servicio de Microsoft es un cambio en el procedimiento de desarrollo. Cada capa es un servicio que se debe desplegar de manera independiente. La figura 20 muestra las tres principales capas de la arquitectura.

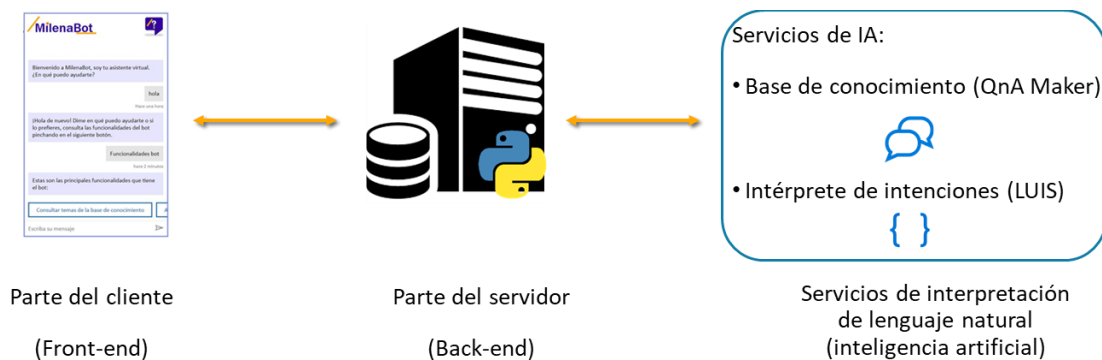


Figura 20- Estructura de la plataforma de Microsoft Bot Services.

A modo de resumen, la tabla 2 presenta una comparativa entre las características de las dos plataformas que mejor se adaptan a las necesidades del proyecto: Dialogflow y Azure Bot Services. Por parte de Azure Bot Services, se evalúa la solución con el despliegue completo en su cloud.

criterio	Dialogflow (Google)	Azure Bot Services (Microsoft)
Servicios cognitivos		
Nº servicios disponibles	Solo dispone de NLU, la base de conocimiento no está disponible en español, sin embargo, permite incorporar respuestas al intent. Se integran los dos servicios en uno.	Diferencia en dos servicios independientes: la base de conocimiento y el NLU. LUIS puede extraer entidades y reconocer expresiones del usuario mientras que QnA permite asociar preguntas-respuestas y contextos.
Entrenamiento	Solo se necesitan dos pasos, es posible utilizar tanto registros internos como externos. Además, permite variar la clasificación del intent.	Ambas plataformas permiten revisar expresiones sin matched para agregarlas al intent o pregunta correspondiente.
Contextos en Base de conocimiento	Limitan las siguientes expresiones del usuario.	Sirven como recomendación relacionada con la expresión del usuario.
Servidor		
Integración de servicios externos	La capa gratuita no permite conexiones desde el fulfillment a APIs externas, solo a las disponibles en Google Cloud Platform.	No hay restricciones en este ámbito.

Autenticación	Mediante mecanismos propios de la plataforma.	Si está desplegado en Azure, al igual que en Google, utiliza mecanismos propios.
Dificultad técnica	La configuración del agente no implica tener conocimiento técnicos excepto por el desarrollo de los webhooks.	El desarrollo de la lógica del bot requiere altos conocimientos técnicos. La configuración de la base de conocimiento está guiada, solo requiere conocer la configuración.
Estadísticas	La estadísticas solo están disponibles sobre los registros de las conversaciones. No permite la configuración por parte del desarrollador.	La visualización de los registros de las conversaciones es mediante una aplicación externa: Application Insight. Permite analizar todo tipo de datos asociados al bot, desde latencia a expresiones sin matched.
Parte del cliente		
Integración de elementos multimedia en la respuesta	No permite integrar botones. Las respuestas multimedia dependen del canal en el que el bot esté integrado. Las imágenes deben tener una URL pública.	Permite integrar botones, tarjetas enriquecidas con vídeo y audio, imágenes y vídeos almacenados. No requiere que tengan acceso público. Depende del desarrollo del núcleo del bot.
Canales disponibles	10 canales: kik, Line, Skype, Cisco Spark, Telegram, Cisco Tropa, Twilio, Twitter, Viber y Web Chat.	12 canales: Facebook, Slack, Microsoft Teams, Telegram, Kik, Skype, Messenger, mensajes de texto, Twilio, Skype, Cortana y Web chat.

Tabla 2 - Comparación entre plataformas habilitadoras de bot

1.5.6 Evaluación económica

En la evaluación económica, se analizan exclusivamente los servicios que se requieren para el proyecto. La pauta económica es el criterio más importante del proyecto.

1.5.6.1 Evaluación económica: Dialogflow

El desglose económico de cada capa de Dialogflow está en el enlace [11]. La solicitud se define como cualquier llamada al servicio, directamente mediante el uso de la API o indirectamente mediante la interacción con el asistente a través de un canal.

La capa Standard Edition proporciona la funcionalidades básicas. Como se puede comprobar en la tabla 3, el número de interacciones están limitadas por cuotas de uso. Además, el soporte solo está disponible por correo electrónico. Google recomienda esta capa para pequeñas y medianas empresas que desean iniciar proyectos o experimentar con Dialogflow.

A diferencia de la capa básica, la Enterprise Edition proporciona un mayor número de servicios como el análisis de sentimientos. La tarifa se ajusta al pago por uso y la asistencia es a través del cloud. Los tiempos de respuesta del soporte están acordados en un SLA público, la capa gratuita no dispone de ningún acuerdo de este tipo. Se recomienda para escenarios con grandes cambios en la demanda de los usuarios. Las condiciones de uso incluyen la aceptación del registro de datos de Google Cloud Platform. Toda la información está disponible en el enlace [12].

Servicio	Standard Edition	Enterprise Edition	
		Essentials	Plus
Texto	Gratis	0.002\$ / solicitud	0.004\$ / solicitud
Límite texto	180 solicitudes/min	600 solicitudes/min	
Detección de intents por voz (STT)	Gratis	0.0065 \$	0.0085 \$
		Cada 15 seg de audio	
Salida de audio (texto a voz, TTS).	Gratis	Voz estándar: 4\$ cada millón de caracteres	

Limites voz (igual para entrada y salida)	100 solicitudes/min 1,000 solicitudes/día 15,000 solicitudes/mes	300 solicitudes/min
--	--	---------------------

Tabla 3 - Límites y precios de Dialogflow

Dialogflow y los servicios Cognitivos de Azure presentan los mismos límites respecto a la entradas de audio, el máximo de cada solicitud son 60 segundos. Redondean el exceso como otra solicitud. Por ejemplo, si se realiza una petición de 15.14 segundos, se factura como una de 30 segundos.

1.5.6.2 Evaluación económica: Azure

La estimación del precio para los servicios de Azure es distinta. En primer lugar, es necesario conocer cuántos servicios de interpretación de lenguaje natural se van a contratar. Al igual que en Dialogflow, la capa gratuita no dispone de ningún acuerdo a nivel de servicio. Los precios mostrados en las siguientes tablas están disponibles en la calculadora de costes de Azure del enlace [13]. La tabla 4 contiene los de LUIS cuyo NLU dispone de un servicio de reconocimiento de voz pero no de “text-to-speech”.

Servicio	Gratis	Estándar (S1)
Texto	Gratis	€1,265 por 1000 solicitudes
Límite texto	5 solicitudes/segundo	50 solicitudes/segundo
Entrada de audio Detección de intents (reconocimiento de voz, voz a texto, STT).	No disponible	€4,639 por 1000 solicitudes
Limites voz	No disponible	50 solicitudes/segundo

Tabla 4 - Precios de LUIS (Azure)

QnA Maker dispone al igual que LUIS de dos versiones. Las solicitudes solo pueden ser de tipo texto según se muestra en la tabla 5.

Servicio	Gratis	Estándar (S1)
Texto	Gratis	8,433 € / mes

Límite texto	3 solicitudes/segundo 100 solicitudes/min 50.000 solicitudes/mes	100 solicitudes/segundo
Documentos administrados	3	Ilimitados

Tabla 5 - Precios de QnA (Azure)

El coste final no se reduce exclusivamente a los servicios de IA. Una vez que el desarrollador selecciona uno o ambos servicios, tiene que generar un grupo de recursos en el cloud con un coste asociado por instancia. Este proceso es obligatorio. Finalmente, si el desarrollador desea desplegar el bot en la nube, se añade también un coste por instancia del propio bot y por cada petición en un canal premium. Los canales básicos son suficientes para los actuales requisitos, por lo que se descarta de la estimación. Durante este análisis, se ha desplegado un piloto con ambos servicios en la nube. Los resultados de la figura 21 muestran que el precio final es de 51.28 €. Esto es un crédito gratuito de evaluación.

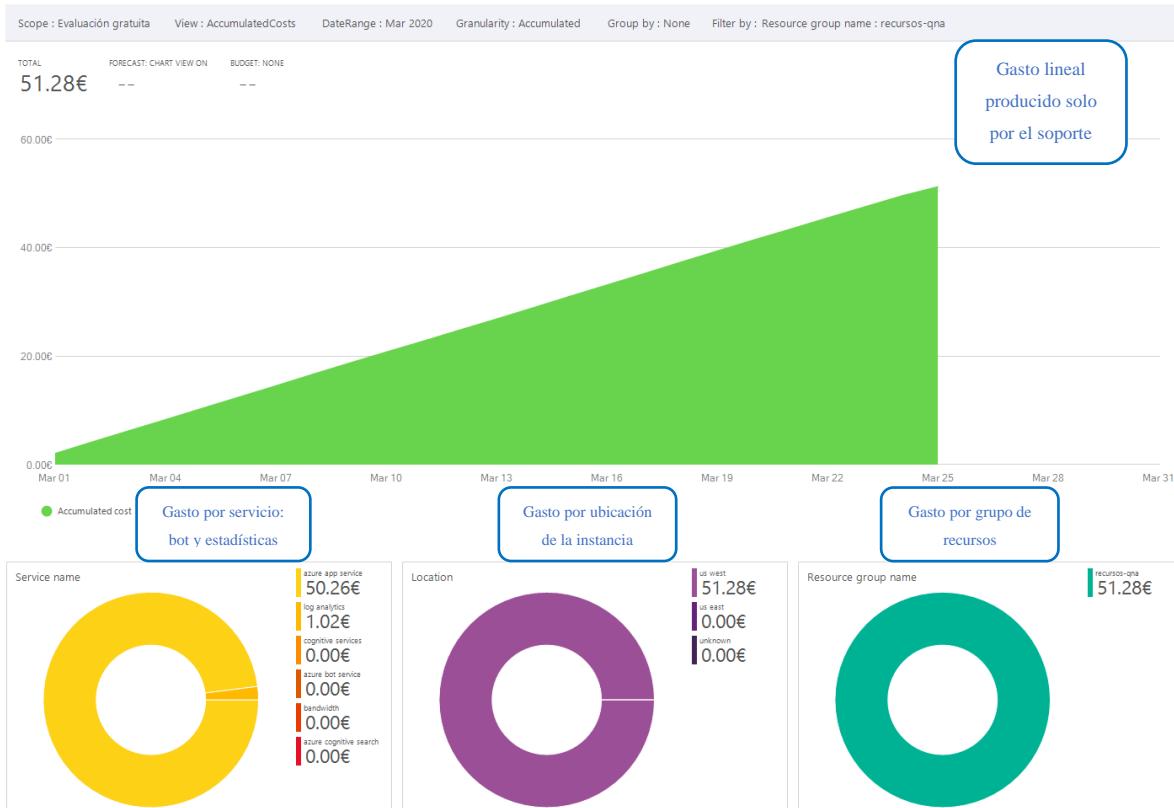


Figura 21 - Desglose de costes de piloto mes de febrero

Sin embargo, este análisis de costes es inexacto debido a que solo se ha generado tráfico de prueba. Para estimar de una forma más exacta, se ha utilizado la calculadora del enlace [13]. Debido a que en este momento se desconoce el número de personas que van a hacer uso del bot, se ha seleccionado la capa gratuita para la base de conocimiento y un servicio de S1 con 15.000 peticiones/mes para LUIS. Este es el límite máximo de la capa Standard de Dialogflow.

El siguiente paso es seleccionar el soporte, las opciones disponibles son: “Desarrollador” para periodos de prueba y evaluación, “Standard” para empresas con una dependencia crítica empresarial mínima de Azure y “Professional Direct” cuando el chatbot es crítico para el negocio. El precio final estimado se muestra en la tabla 6 junto con el soporte estándar.

Microsoft Azure Estimate					
Su presupuesto					
Service type	Custom name	Region	Description	SKU(s)	Estimated Cost
Azure Bot Service	Bot Service	West US	S1 nivel, 0 mensajes en canales Premium	AAA-92179	\$0.00
Cognitive Services	LUIS	West US	Language Understanding (LUIS) API: nivel Solicitudes de texto, 15.000 transacciones.	AAA-92213	\$22.50
Cognitive Services	QnA Maker	West US	Nivel Estándar; número ilimitado de documentos administrados.	{ "textAnalytics": "N7H-10244" }	\$10.00
Support			Standard level	Support	\$100.00
				Licensing Program	Enterprise Agreement (EA)
				Monthly Total	\$132.50
				Annual Total	\$1,590.00
Disclaimer					
All prices shown are in US Dollar (\$). This is a summary estimate, not a quote. For up to date pricing information please visit https://azure.microsoft.com/pricing/calculator/ This estimate was created at 3/30/2020 10:21:13 AM UTC.					

Tabla 6 - Precio final del despliegue del cloud en Azure

1.5.7 Elección de la tecnología

El objetivo de este apartado es justificar la elección de la tecnología sobre la que se desarrolla el chatbot. La tabla 7 incluye los criterios de decisión, si la plataforma tiene la funcionalidad descrita suma la puntuación correspondiente.

Criterio	Dialogflow		Azure Bot Service	
Ponderación muy alta: 5 puntos				
Idiomas disponibles en NLU y BC	Dispone de español, inglés y portugués.	5	Aunque permite integrar cualquier servicio cognitivo, LUIS y QnA están disponibles en español, inglés y portugués.	5
Integración de los servicios de Milena	No disponible en la capa gratuita. Desarrollo por webhooks limitado.	0	Libertad absoluta.	5
Integración de otros servicios de IA en el bot core	No es posible separar el núcleo del bot de los servicios de IA.	0	No hay acoplamiento entre el bot core y los servicios de IA.	5
Económico	Las capas de pago son más baratas debido al bajo coste por soporte.	5	Coste muy superior a Dialogflow.	0
Integración en interfaz web	Dispone de un Iframe pero no modificable.	0	Libertad de integración y diseño absoluta.	5
Ponderación alta: 3 puntos				
Despliegue en red local	La configuración es exclusiva en el nube.	0	Dispone de ambos enfoques.	3
Gestión de los datos	Todos los datos son gestionados por la plataforma y sometidos a las condiciones de uso.	0	Depende de la arquitectura y la IA utilizada.	3

Diseño de la capa cliente	No permite diseñar el webchat.	0	Libertad absoluta.	3
Ponderación media: 2 puntos				
Tiempo de desarrollo	Solo es necesario el desarrollo software de los webhooks.	2	La lógica del bot implica un desarrollo software.	0
Capacidad para delegar tarea de introducción de contenidos	Se necesita formación para el manejo de la plataforma.	0	Depende del IA contratado pero QnA Maker está preparado para ello.	2
Ponderación baja: 1 punto				
Extracción de datos de sesión para estadísticas	No es necesario contratar otro servicio. Estadísticas muy sencillas.	1	Es necesario contratar un servicio externo pero ofrece datos sobre todos los aspectos.	1
Tráfico	Solo una petición por expresión del usuario.	1	Depende del diseño y del número de servicios de IA contratados.	0
Seguridad	Ofrece mecanismos de cifrado (OAuth 2.0) e identificación del usuario.	1	Igual	1
Total	15		33	

Tabla 7 - Evaluación tecnología

La puntuación final es favorable a desarrollar la lógica del bot mediante el framework abierto de Bot Services. Esta solución permite la puesta en producción tanto en local como en el cloud. Además, esta elección no implica que la IA utilizada sea LUIS y/o QnA porque este enfoque permite integrar otros servicios mediante conexiones API REST.

1.6 Arquitectura del chatbot

La arquitectura está compuesta principalmente por la parte front-end, back-end y los servicios de interpretación de lenguaje natural. Para el desarrollo del back-end o middleware, Azure Bot Service dispone de cinco plantillas que contienen funcionalidades básicas y totalmente aptas para un despliegue en una red local. En la creación, se elige la plantilla y el lenguaje del SDK del bot. En este caso, se ha escogido la opción de funcionalidad básica y Python 3.7 respectivamente. La elección de este lenguaje se debe a un acuerdo con el equipo de proyecto.

1.6.1 Despliegue del back-end en la red corporativa

En este primer escenario de la figura 22, el usuario interactúa con el bot mediante voz o texto a través de la aplicación web. La aplicación autentica y autoriza al usuario a acceder a la lógica del bot. El middleware realiza llamadas a la API de QnA (Base de conocimiento) y LUIS (Natural Language Understanding) para obtener la respuesta sobre la pregunta del usuario y la intención de su expresión respectivamente. El desarrollador entrena con ejemplos y asocia expresiones nuevas y sin respuesta a intents ya definidos con el objetivo de aumentar la precisión de respuesta del bot.

Sin embargo, ambas plataformas no incluyen ninguna funcionalidad que permita ver los registros de las conversaciones. Para habilitarlo, es necesario que el propio desarrollador diseñe un panel mediante Kusto, que es un servicio de almacenamiento y análisis interactivos sobre macrodatos. Finalmente, el middleware se conecta con el microservicio para la gestión de funcionalidades de la plataforma Milena.

Este despliegue limita la integración al canal web chat. Para disponer de más canales como: Facebook, Slack, Microsoft Teams, Telegram, Kik, Skype, Messenger, mensajes de texto, Twilio, Skype y Cortana, es necesario realizar una configuración específica mediante el CLI (Command-Line Interface) de Azure. La información resumida sobre este proceso se encuentra en el enlace [12]. La gestión de mensajes en el middleware se realiza independientemente del canal en el que se encuentre publicado.

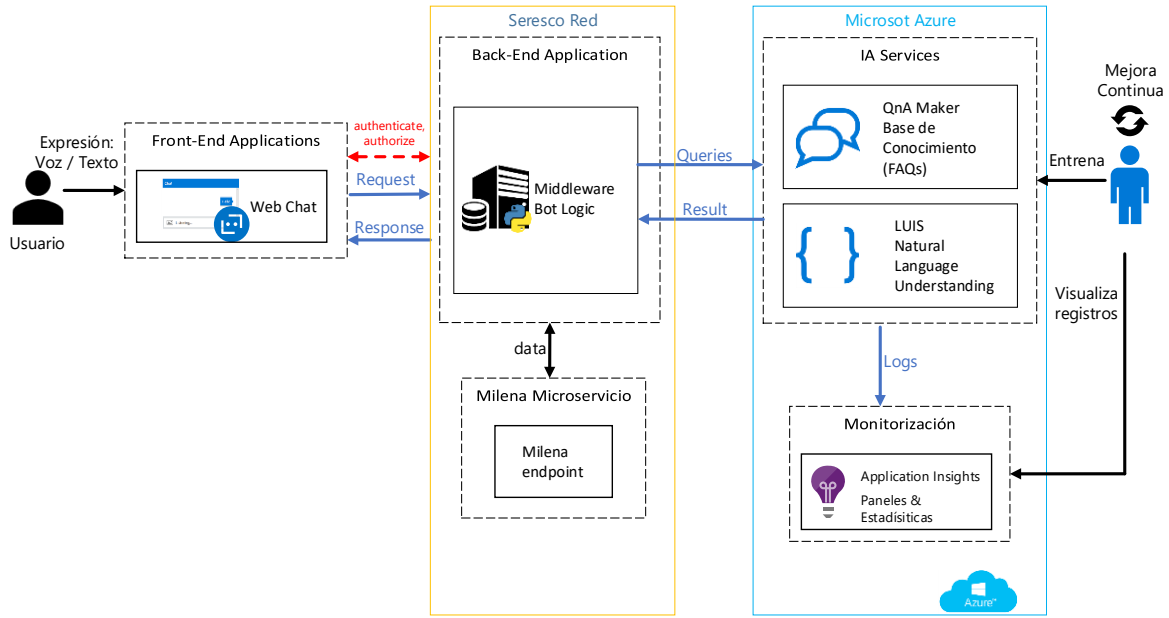


Figura 22 - Arquitectura local con servicios de IA de Azure

Se plantea como alternativa la arquitectura de la figura 18, donde se sustituyen los servicios de IA de Azure por la API.ai de Google. Este nuevo paradigma está motivado por el inferior coste que suponen los servicios de Dialogflow. Además, permite la visualización de los registros de las conversaciones y el entrenamiento en una única plataforma. El servicio es híbrido por lo que unifica la base de conocimiento y el NLU en una misma herramienta.

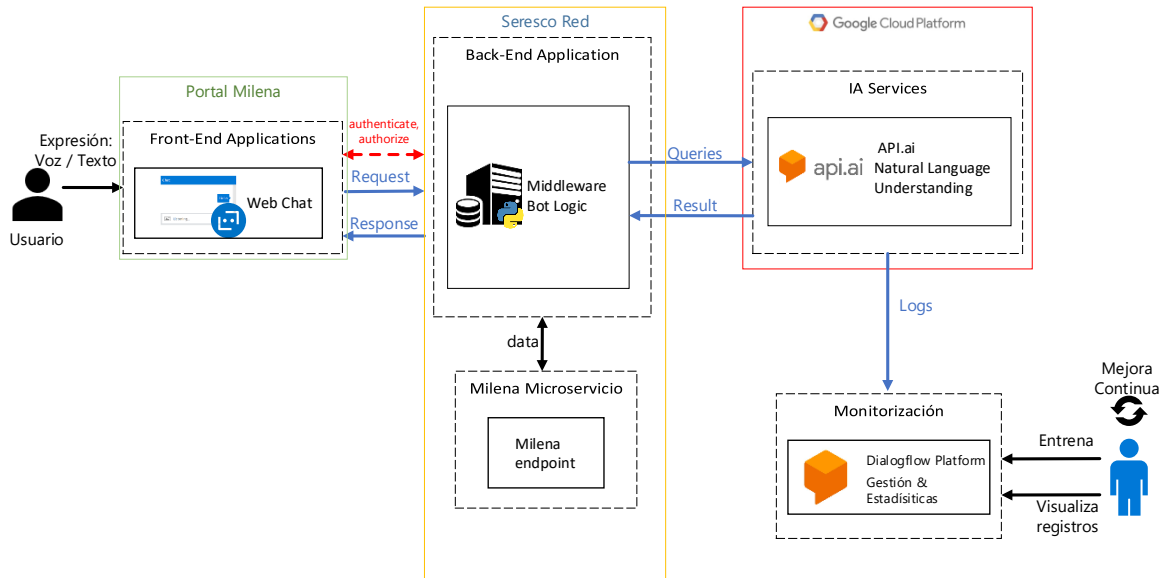


Figura 23 - Arquitectura local con servicios de IA de Google

El objetivo principal es desacoplar el middleware del servicio de IA. Sin embargo, existen soluciones que no dependen de servicios externos para la interpretación del lenguaje. Este es el caso de RASA, que es un proyecto de código abierto. Este NLU tiene un pequeño problema, necesita versiones específicas de Keras y Tersonflow, todavía no es posible ofrecer una solución estable en todas las versiones de Python. Estas dependencias han imposibilitado la integración de un agente RASA en el servidor. Toda la información sobre esta plataforma se encuentra en el enlace [15].

Además, es necesario que el desarrollador integre una aplicación para la monitorización de los registros de conversación. La figura 24 contiene el diseño de esta arquitectura.

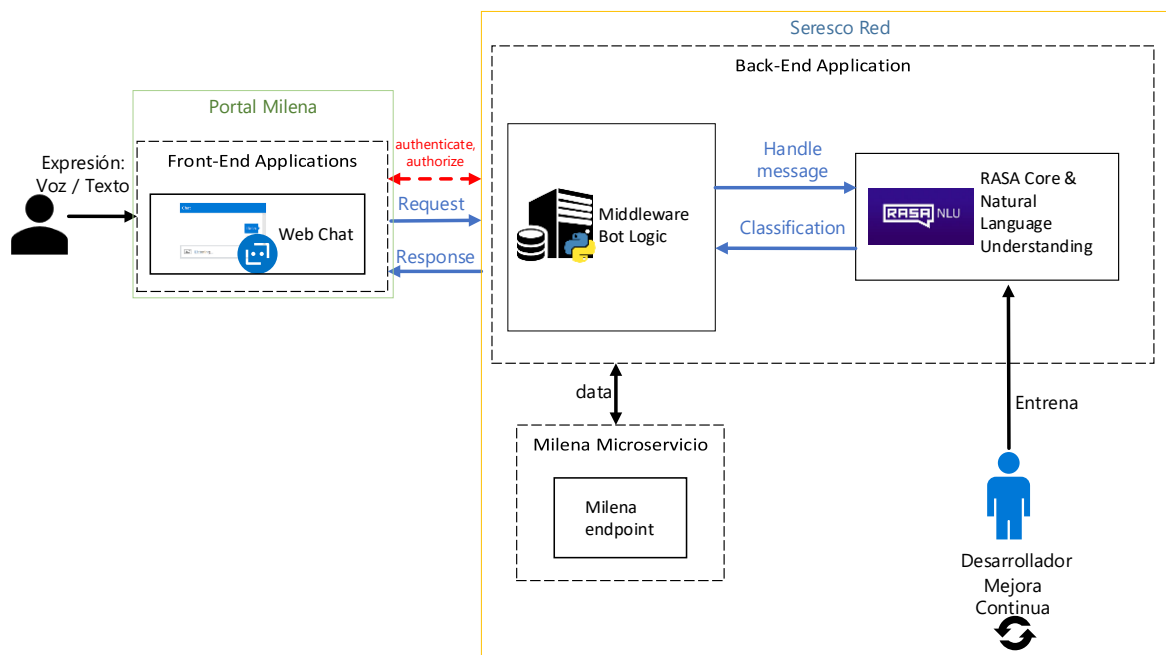


Figura 24 - Arquitectura local con RASA

1.6.2 Despliegue del back-end en el cloud de Azure

A pesar de que Azure Bot Service ofrezca plantillas para desarrollar el middleware en un servidor corporativo, la plataforma ofrece la posibilidad de implementarlo en la nube una vez que se ha desarrollado el prototipo en un entorno local. El enlace [16] detalla los pasos para desplegar el servicio mediante el CLI de Azure. De esta manera, Azure gestiona el acceso al bot y el enlace con los canales disponibles en la plataforma. Sin embargo, es incompatible como es lógico con bases de datos corporativas.

La arquitectura de la figura 25 es ideal en situaciones en las que no se disponga de servidor propio, esto es conocido comúnmente como enfoque “*serverless*”. Las ventajas principales son: la capacidad de escalado, la reducción de costes iniciales en hardware, la facilidad de mejora continua con el despliegue de nuevas versiones, la monitorización del middleware, el almacenamiento y la seguridad.

Esta arquitectura dispone de integración webchat mediante un iframe que es un componente que permite incrustar todo tipo de medios interactivos dentro de la interfaz web. No sería necesario, por tanto, desarrollar una dedicada a tal efecto. No obstante, Microsoft dispone de una aplicación gratuita que sirve de plantilla para el diseño de una interfaz propia. Azure cloud dispone de sus propios métodos de autenticación y autorización.

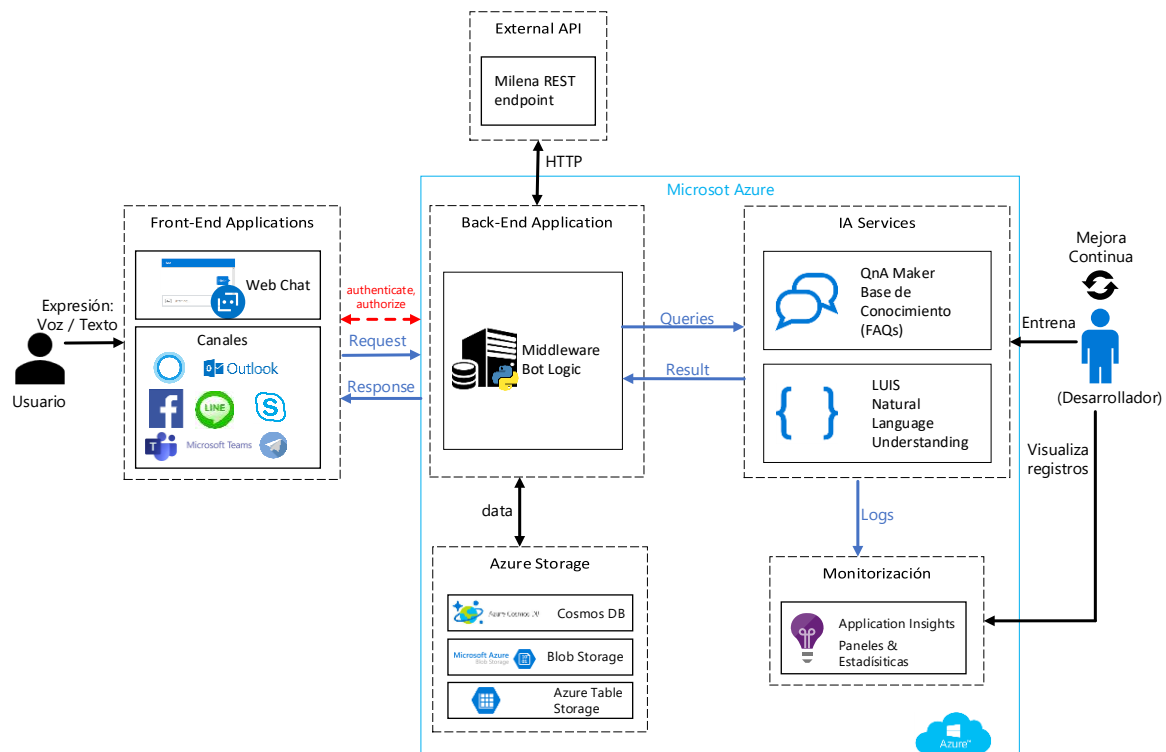


Figura 25 - Arquitectura Bot implementado en cloud Azure

1.6.3 Análisis de arquitecturas

La tabla 8 compara las tres principales arquitecturas: back-end desplegado en red corporativa con QnA y LUIS, esta misma arquitectura con la IA de Dialogflow y la implementación completa en el cloud de Azure. El proceso de análisis es igual que en la elección de tecnología. La solución que mejor se adapta a las necesidades vigentes es la segunda opción.

Criterio	MD + LUIS & QnA	MD + API.ai (Google)	Azure Bot Service
Ponderación alta: 5 puntos			
Gestión de los datos corporativos	Solo se enviarían algunas expresiones para reconocer la intención del usuario.	Igual que el primer caso.	Todos los datos se gestionan a través del cloud de Azure.
Nº servicios contratados en el cloud	El grupo de recursos en Azure debe contener: Cognitive Services, Application Insights, App Service, Servicio de búsqueda.	Solo el servicio de API.ai de Google Cloud Platform (Dialogflow).	Mismos servicios que en el primer caso, pero además incluyendo Bot Service.
Económico	Mínimo 100€ de mantenimiento más el pago por petición.	Solo pago por petición. La capa gratuita permite el desarrollo sin cobro.	Mínimo 100€ de mantenimiento más el pago por petición. Al desplegar el server en el cloud aumentaría el coste por petición.
Seguridad	Solo accesible mediante Milena.	Igual que en el caso 1.	La autenticación es obligatoria

	Se debe desplegar un mecanismo de autenticación integrado en los servicios de Milena. Comunicación con servicios cognitivos cifrada.		Se debe implementar OAuth 2.0 para el acceso a la API.ai.		mediante los mecanismos disponibles en la plataforma. Obligatorio sistema de tokens para la integración del cliente.	
Integración de servicios corporativos	El acceso a los servicios se encuentra en la misma red.	5	Igual que el primer caso.	5	Posible, pero es necesario establecer conexiones cifradas.	5
Ponderación media: 3 puntos						
Webchat	Disponible	3	Disponible	3	Disponible	3
Estadísticas, monitorizar y entrenamiento	Son dos aplicaciones totalmente independientes: Application Insights y los NLUs. Necesario implementar una monitorización en el middleware.	3	Dialogflow permite ambas funcionalidades en la misma plataforma. Las estadísticas solo incluyen datos sobre la interpretación del lenguaje. Obligatoria monitorización en el middleware.	3	Mismo servicio que el primer caso incluyendo la monitorización del middleware.	3

Ponderación baja: 1 punto						
Despliegue y mantenimiento	Mismo procedimiento que un servicio web.	1	Igual que el 1º.	1	Mediante el CLI de Azure, se despliega la versión deseada del bot.	1
Tráfico	Dependiendo del diseño de la aplicación. LUIS y QnA son servicios independientes, de modo que si se implementan ambos, se duplica el número de peticiones.	0	Al solo disponer de un único servicio que integra preguntas-respuestas y expresiones asociadas a procesos, el número de peticiones se reduce.	1	La aplicación web está conectada a la nube, cada expresión es una petición al servicio.	1
Total	22		33		18	

Tabla 8- Análisis de arquitecturas

1.6.4 Análisis de servicios de interpretación de lenguaje

Antes de proceder con un análisis de los servicios de IA, es importante explicar por qué se necesitan los servicios de inteligencia artificial. Es posible que llegados a este punto se pueda interpretar erróneamente que es posible prescindir de ellos. Para abordar esta cuestión, es necesario que el lector reflexione sobre una acción cotidiana como pedir ayuda. Seguramente, ha pensado en expresiones comunes como: “Necesito ayuda”, “Ayúdame” y “¿Puedes ayudarme?”, pero también es posible que en “SOS” o incluso con faltas ortográficas como: “Alludame”. Todas las combinaciones tienen la misma intención, pero expresadas de distinta forma.

Es por ello por lo que las interfaces conversacionales necesitan de un servicio de interpretación de lenguaje natural, para que al introducir cualquier expresión, esta sea procesada y respondida con lógica por la herramienta. Esta libertad implica cierta complejidad. Aunque el objetivo de este proyecto no es desarrollar servicios de interpretación de lenguaje natural, es importante entender cuál es el procedimiento para comprender su complejidad. A continuación, se resume la técnica de procesamiento de lenguaje natural, también conocido por sus siglas en inglés como NLP (Natural Language Processing). Está ampliamente desarrollado en el artículo [17].

1. El primer paso es la descomposición de las frases de entrenamiento en una lista secuencial de palabras. A este proceso se le conoce como "*tokenization*". Por ejemplo, “*Quiero jugar*” puede parecer sencillo para un castellano hablante, sin embargo se vuelve complejo en lenguajes en los que realicen contracciones como en el inglés. La descomposición de “*I’ll be there for you*” es [I, will, be, there, for, you].
2. El segundo paso es el “*stemming*”. Es el proceso por el que se obtiene la raíz de cada una de las palabras. En el caso particular de “*jugar*” es “*jug*”. Si en el primer paso el español tenía ventaja, ahora surge un problema con las palabras derivadas y conjugaciones.
3. A cada raíz extraída, se le conoce como “*feature*”. En este paso, se forma un vector en el que cada posición representa a una feature. Se compara el vector de la expresión del usuario con la frase de entrenamiento.

4. Llegados a este punto, se inicia el proceso matemático que respalda la inteligencia artificial. El perceptrón es una función donde cada término se multiplica por un valor llamado peso sináptico y al resultado se le suma un coeficiente llamado bias. Una vez calculado, se aplica una función de activación. Para calcular los pesos y bias se utiliza un optimizador que utiliza a su vez una función de coste con el que busca el mínimo de la función. Este proceso es sumamente complejo. En resumen, para cada intent hay asignado un perceptrón. Esta es la herramienta por la cual se estima la probabilidad de que la expresión del usuario vectorizada en “*features*” se corresponda con el intent.

No obstante, este proceso es totalmente ajeno al desarrollo del bot. Para el proyecto, los servicios de interpretación de lenguaje natural se comportan como una caja negra que es entrenada por frases, recibiendo como parámetro de entrada una expresión y devolviendo un intent con un coeficiente. Por tanto, la evaluación que se pretende realizar es mediante una simple prueba entre los servicios de Dialogflow y LUIS. Los resultados están en la tabla 9.

```
Frases_de_entrenamiento =[quiero realizar una sugerencia, hacer sugerencia, sugerir incidente, comentar problema, avisar de un problema, quiero hacer un aviso, quiero poner una queja]
```

```
texts_to_be_analyzed = ["sugerencia o queja", "solicitar sugerencia", "¿Cómo puedo realizar una sugerencia?", "comentar queja", "COMENTAR HINCIDENTE", "problema queja", "poner aviso", "acer sugerencia", "poner sugerencia", "sugerir un problema"]
```

Dialogflow:

Matched Promedio: 0.749753389093611 , Total: 9 / 10

NO-Matched Total: 1 / 10

Expresiones No-matched: ['poner sugerencia']

LUIS:

Matched Promedio: 0.95782570775 , Total: 8 / 10

NO-Matched Total: 2 / 10

Expresiones No-matched: ['poner sugerencia', 'sugerir un problema']

Tabla 9 - Resultados de análisis de LUIS y Dialogflow

Esta prueba es orientativa, no indica si un servicio es mejor que otro porque el número de intents y frases introducidos es muy pequeño. Aunque no es una muestra significativa, sí permite comprobar que Dialogflow consigue interpretar la expresión “sugerir un problema” a pesar de que la falta ortográfica se encuentre en la raíz léxica, “prov”. Si se omite esta expresión, ambas tecnologías detectan el mismo número de intents, pero LUIS presenta una mejora de dos décimas, un coeficiente de 0.957 por 0.765 de Dialogflow.

El estudio del enlace [17] es similar al realizado en este apartado, entre las pruebas principales destaca la evaluación de tres conjuntos de intents en igualdad de condiciones. El resultado es el mostrado en la figura 26.

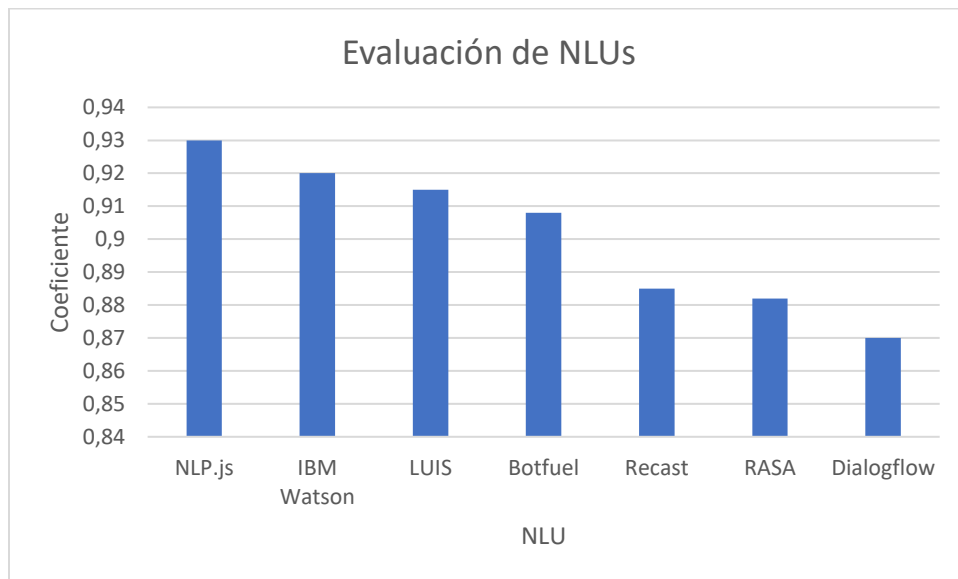


Figura 26 - Evaluación de NLU

En resumen, este análisis muestra que el comportamiento de LUIS es mejor que el de Dialogflow. Sin embargo, este no es un aspecto crítico para la elección de del servicio de interpretación de lenguaje natural.

1.7 Flujo de conversación

En este apartado se pretende ofrecer una visión general de las funcionalidades de la herramienta. El contenido del chatbot se puede clasificar en tres tipos: formularios, procesos y base de conocimiento. Los apartados siguientes muestran estas tres funcionalidades desde la perspectiva del usuario.

1.7.1 Flujo de conversación: base de conocimiento

La base de conocimiento está compuesta principalmente por información de Nómina y Recursos Humanos. La estructura de las preguntas de la base de conocimiento es simple. Generalmente, contiene una respuesta textual y temas relacionados con la consulta. Las buenas prácticas de Microsoft descritas en el enlace [6] recomiendan que el usuario reciba opciones similares. Este diseño deriva de escenarios ya probados en interfaces web, de modo que la probabilidad de que seleccione una opción sugerida es muy alta, pero no le obliga a escoger una. La figura 27 es un ejemplo de consulta sobre el salario mínimo interprofesional.

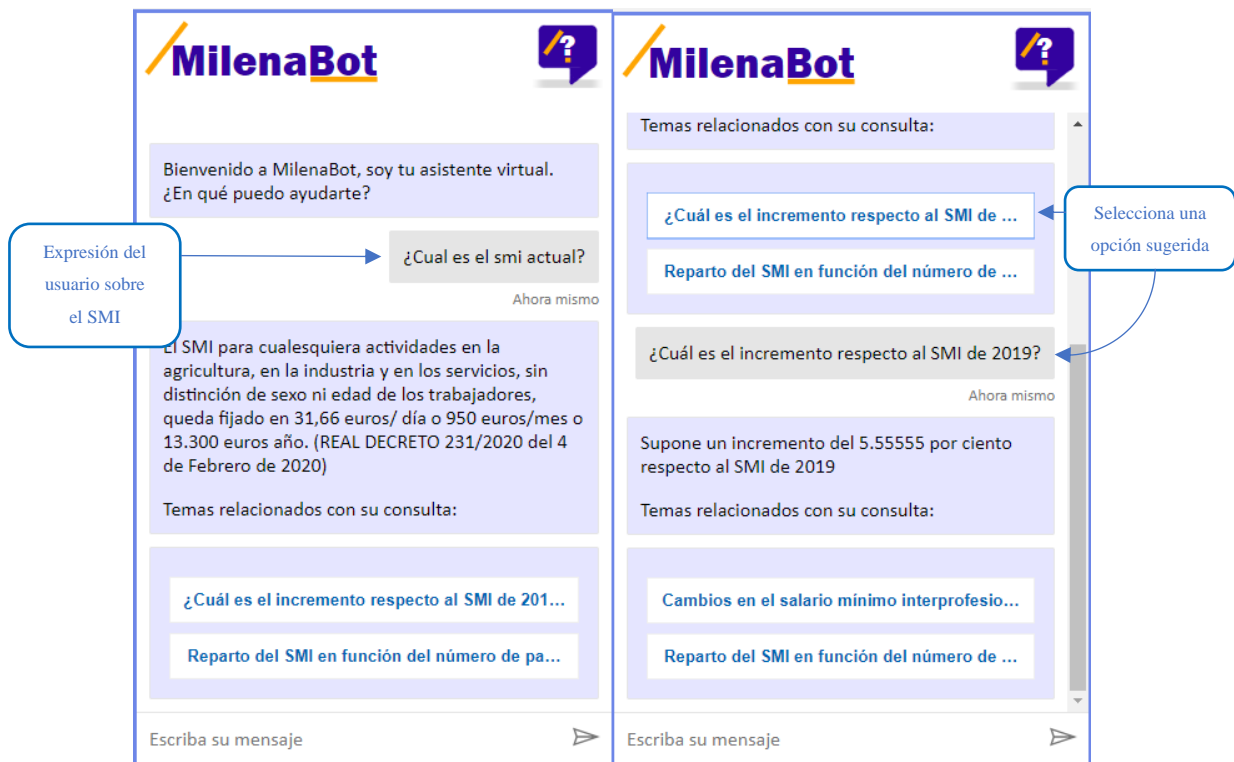


Figura 27 - Ejemplo de petición a base de conocimiento

1.7.2 Flujo de conversación: procesos / procedimientos

Para el usuario, los procesos y la base de conocimiento tienen una estructura similar. En ambos escenarios recibe información sobre la consulta. Sin embargo, los procesos o procedimientos tienen otro objetivo. Están orientados a guiar al usuario mediante sucesivas interfaces. El ejemplo de la figura 28, se muestra un recorrido por su nómina. El usuario parte de una primera introducción y avanza entre diferentes pasos mediante las opciones sugeridas. Es posible que no pueda acceder desde el estado inicial a un paso en concreto. Esto depende de cómo lo configure el personal de NRH, la principal diferencia es que los procedimientos tienen obligatoriamente definido un flujo de conversación mientras que en las preguntas de la base de conocimiento no es necesario.



Figura 28 - Ejemplo de proceso sobre explicación de nómina

Sin embargo, una de las ventajas de los asistentes conversacionales es que además de ofrecer esa libertad en la expresión del usuario, pueden incluir elementos multimedia similares los de una interfaz web convencional. La figura 29 muestra un ejemplo de configuración del

teletrabajo adjuntando imágenes en cada paso. Como se puede comprobar estos elementos favorecen la usabilidad de la herramienta.

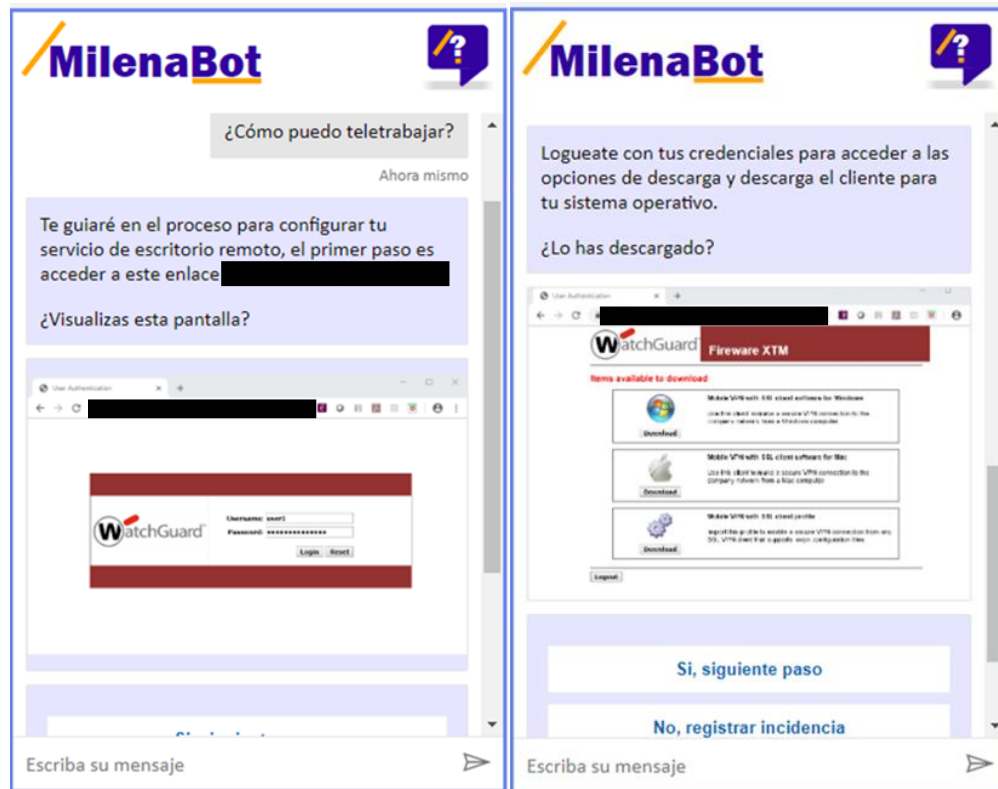


Figura 29 - Ejemplo de proceso con elemento multimedia

1.7.3 Flujo de conversación: formularios

Los formularios recopilan información mediante preguntas para ofrecer un servicio. El asistente dispone de identificación de entidades que permite extraer información de la expresión del usuario. La figura 30 muestra el ejemplo de solicitud de un periodo vacacional. A raíz de la expresión introducida, “*Solicito vacaciones del 8 de junio al 11 de junio*”, el bot identifica la fecha inicial y final y realiza las comprobaciones pertinentes. Si estas son validadas, el chatbot no pregunta sobre las fechas. De esta forma, se disminuye el número de pasos que se necesitan para gestionar la solicitud. Finalmente, todos los formularios incluyen un resumen previo a la confirmación con todos los parámetros introducidos.

Es importante destacar que los formularios no pretenden sustituir a la plataforma, sino ofrecer la posibilidad de aumentar la accesibilidad a los servicios ya existentes.

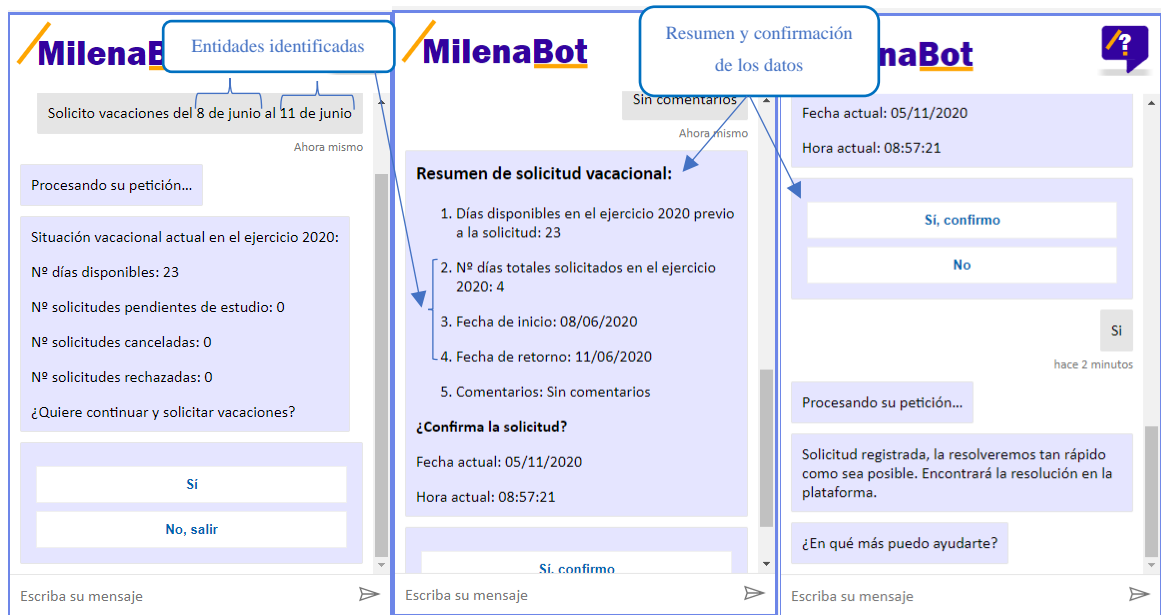


Figura 30 - Ejemplo de formulario de solicitud vacacional

1.7.4 Plataforma de contenidos

El objetivo de la plataforma de contenidos es evitar la dependencia de una tecnología y facilitar al personal de NRH la incorporación y mantenimiento de contenidos en MilenaBot. El lector puede interpretar por las explicaciones previas que la base de conocimiento de Microsoft (QnA Maker) cumple con los requisitos que se demandan de sencillez y formato de pregunta - respuesta. Sin embargo, si en un futuro se necesitase migrar los contenidos a otro sistemas, no sería posible. Esto que se conoce como *“matrimonio tecnológico”*, es un aspecto crucial a largo plazo. Además, al depender de dos servicios externos, el número de peticiones y coste se duplicaría.

La plataforma está formada por una interfaz web con 4 pestañas: creación del intent, visualización y búsqueda de los intents y expresiones no-matched y creación del flujo de conversación. Aunque el alcance de esta primera versión no incluya esta plataforma, cabe destacar la gran transcendencia que tiene en el proyecto. La información de los procesos y base de conocimiento proviene exclusivamente de la base de datos que es común a ambas herramientas.

1.8 Planificación del proyecto

En este apartado, se describe el equipo que conforma el proyecto “Milena Chatbot” y las tareas que ha desarrollado el alumno dentro del proyecto.

1.8.1 Equipo de proyecto

El proyecto es de gran envergadura por lo que el equipo no está únicamente formado por el alumno. Estos son los miembros que lo conforman y sus roles dentro del equipo:

- Antonio Manuel Campos López (Director del área de Tecnología e Innovación) es el responsable del proyecto. Ha establecido los requisitos y funcionalidades que debe tener la herramienta. Realiza la coordinación con el cliente, departamento de Nómina y Recursos Humanos.
- Celia García-Rovés Prieto (analista en Tecnología e Innovación) supervisa las tareas y asiste en los problemas técnicos.
- Jesús Rodríguez Pérez (analista en Tecnología e Innovación) mismo rol que Celia García-Rovés.
- Francisco Manuel Mendoza Soto (desarrollador en Tecnología e Innovación) es el encargado de todas las tareas asociadas al front-end (parte del cliente).
- Andrés Presno García (desarrollador en Tecnología e Innovación) está encargado del desarrollo del back-end (parte del servidor) y la gestión y administración de los servicios de IA.
- Rodolfo de Benito Arango (responsable del área de IA & Data Science Service) ofrece asistencia para temas relacionados con los servicios de IA contratados y los intérpretes de lenguaje natural.

La asignación de roles de los desarrolladores ha sido establecida por mutuo acuerdo entre los miembros del equipo. El alumno ha preferido desarrollar el back-end por tener conocimientos previos sobre el tema.

1.8.2 Tareas del proyecto

Las tareas que se describen a continuación son las realizadas exclusivamente por el alumno. Todas las descritas y mostradas en la tabla 10, son las que están registradas en la herramienta corporativa Redmine. Esta es una herramienta para la gestión de proyectos que permite a los miembros del equipo realizar el seguimiento y organización de las tareas. Todas las definidas a continuación pertenecen a la primera versión.

1. En el planteamiento del proyecto, se establece los objetivos del sistema acorde a los requisitos necesarios para alcanzarlos. Se determina qué ámbitos se abarcan y cómo desarrollarlos para que la herramienta sea viable.
2. La segunda parte es el estado del arte. Se analizan las plataformas habilitadoras de chatbot con sus respectivas funcionalidades.
3. La tercera tarea es el diseño de la arquitectura del bot. Se debe seleccionar y plantear la solución que mejor se adapta a las necesidades y requisitos vigentes.
4. La cuarta tarea es el diseño y desarrollo de la BD con el fin de almacenar la información de las preguntas y los datos corporativos.
5. El diseño de flujo de conversación tiene tres etapas: desarrollo de los procesos, base de conocimiento y formularios.
6. Una vez desarrollada la herramienta, se realizan varias tareas de integración de sistemas y pruebas con el objetivo de verificar el cumplimiento de todos los requisitos.
7. En la última parte, se redacta toda la documentación del proyecto.

EDT	Tarea	Fecha inicio estimada	Fecha finalización estimada (inclusive)	Horas totales planificadas
1	Planteamiento del proyecto	4 – Feb	4 – Feb	6 h
2	Estado del arte			
2.1	Análisis de la plataforma: Dialogflow	5 - Feb	7 - Feb	18 h

2.1	Análisis de la plataforma: Microsoft Bot Framework	10 - Feb	12 - Feb	18 h
2.1	Análisis de la plataforma: Amazon Lex	13 - Feb	13 - Feb	6 h
2.1	Análisis de la plataforma: IBM Watson	14 - Feb	14 - Feb	6 h
3	Diseño de la arquitectura			
3.1	Análisis de arquitecturas	17 - Feb	28 - Feb	60 h
4	Configuración de los agentes de Dialogflow	2 - Mar	6 - Mar	30 h
5	Base de datos			
5.1	Diseño de la base de datos con modelo relacional	9 - Mar	13 - Mar	30 h
5.2	Desarrollo BD (Oracle)	16 - Mar	20 - Mar	30 h
6	Diseño del flujo de conversación			
6.1	Desarrollo de base de conocimiento	23 - Mar	3 - Abr	60 h
6.2	Desarrollo de procesos y mensajes con elementos multimedia	6 - Abr	17 - Abr	48 h
6.3	Desarrollo de formularios	20 - Abr	15 - May	114 h
6.4	Multilenguaje	18 - May	20 - May	18 h
7	Integración de las capas cliente, servidor y servicios de IA	21 - May	29 - May	42 h
8	Planteamiento de plataforma de contenidos (Reunión con NRH)	1 - Jun	1 - Jun	6 h
9	Despliegue en server openSUSE	2 - Jun	5 - Jun	24 h
10	Pruebas de back-end	8 - Jun	26 - Jun	90 h
11	Monitorización y Logging	22 - Jun	26 - Jun	30 h
12	Documentación	29 - Jun	10 - Jul	60 h

Tabla 10 - Tareas del proyecto

El diagrama de Gantt con la tareas de la tabla 10, se muestra en la figura 31.

1.9 Trabajo futuro y conclusiones

Este apartado pretende responder cuál es la tendencia actual de los bots y hacia qué nuevas funcionalidades podría ser enfocado si bien actualmente, se ha cubierto todas las necesidades que por lo general, tienen este tipo de herramientas. El área de AI & Data Science Service ha propuesto el siguiente posible cambio de paradigma en la herramienta.

La información disponible es únicamente introducida por un grupo de usuarios y está relacionada con la herramienta. ¿Por qué no avanzar a una solución donde todos los usuarios puedan suministrar preguntas y responderlas? Esto generaría una base de conocimiento a todos los niveles. El personal podría acceder a consultar cualquier cuestión incluida las técnicas en cualquier momento. A través de esto, se generaría un mapa de conocimiento que permitiría detectar cómo se distribuye entre los distintos departamentos. Esto se conoce como “*skill mapping*”. Se debe empezar a pensar que el conocimiento no es propio si no un activo tangible y procesable dentro de la empresa. Un producto similar al planteado es Zapiens, para más información, consultar el enlace [19].

Actualmente, gran parte del personal de Seresco accede a Milena a través de su dispositivo móvil. Un campo de trabajo futuro podría ser disponer de esta herramienta en plataformas que permitan una accesibilidad total, principal garante de éxito.

Personalmente, cuando uno está en un proyecto de esta envergadura y adquiere cierta formación en la tecnología, lleva a preguntarse si realmente esto es un campo de negocio con una demanda propia y no producida por la dependencia de otras a las que da asistencia. Los datos del apartado 1.3 indican que los chatbots desplegados en plataformas son la primera opción y con una demanda creciente. De modo que se propone una reflexión, ¿Avanzan las corporaciones y entidades hacia servicios con interfaces conversacionales donde el usuario tenga total libertad para expresarse?

Actualmente, las grandes empresas tecnológicas tienen chatbots desplegados en sus dispositivos que facilitan el acceso a sus funcionalidades como es el caso de Siri o Cortana. Desde el punto de vista personal, el futuro enfoque está en conseguir que el usuario, que es a su vez trabajador, reduzca el tiempo que necesita para acceder a sus herramientas. El factor

catalizador es la capa de reconocimiento de voz a texto (text-to-speech) que quizás actualmente, no está lo suficientemente madura como para ofrecer una solución ágil y rápida.

2. Presupuesto

En este apartado se procede a realizar un presupuesto de la herramienta. Los precios incluidos en las siguientes tablas son estimaciones. El coste real está excluido por temas de confidencialidad.

2.1 Hardware: equipamiento y amortización

El puesto de trabajo del desarrollador está formado por un ordenador personal, dos pantallas y los periféricos: ratón y teclado. Este equipo es provisto por la empresa Seresco S.A. Los precios mostrados en la tabla 11 son el valor actual de mercado. El desarrollador no tiene acceso a estos datos dentro de la empresa.

- Las principales características del PC - ASUS F31CD-K-SP002T son: procesador (I3 I3-7100 3,9 GHZ), memoria RAM (4GB) y disco dura HDD (500GB).
- Los dos monitores son de tipo Asus VS197DE HD de 19 pulgadas.
- Los periféricos son el teclado (Logitech Wireless MK270) y el ratón inalámbrico (Logitech B100).

Concepto	Cantidad	Precio unitario (€)	Total (€)
PC Asus	1	368,7	368,70
Monitor Asus	2	65,47	130,94
Teclado Logitech	1	21,99	21,99
Ratón Logitech	1	4,99	4,99
Subtotal hardware			526,62

Tabla 11 - Precio del equipamiento hardware

Los precios de la tabla 11 son precios absolutos. No obstante, el equipo no ha sido comprado exclusivamente para el desarrollo del proyecto. Para el presupuesto final, se debe calcular su amortización fijando un horizonte de 2 años. Esto está especificado en la tabla 12.

Concepto	Cuota Adquisición (€)	Tiempo total (años)	Tiempo de uso (meses)	Amortización (€)
PC Asus	368,70	2	6	92,18
Monitor Asus	130,94	2	6	32,74
Teclado Logitech	21,99	2	6	5,50
Ratón Logitech	4,99	2	6	1,25
Subtotal amortizado de hardware				131,66

Tabla 12 - Amortización equipamiento hardware

2.2 Software: equipamiento y amortización

La tabla 13 recoge el coste de todas las licencias de los productos softwares utilizados y sus respectivos entornos de desarrollo. La empresa ha recomendado que se utilicen productos con licencia gratuita de modo que no se ha realizado la amortización.

- La licencia de PyCharm Community Edition (2019.3.3 x64) necesaria para el desarrollo del Back-end (servidor) en Python 3.7.
- El emulador para ejecutar las pruebas en el servidor es Bot Framework Emulator V4.
- La licencias de Dialogflow Standard Edition (Google) para el servicio de interpretación de lenguaje natural.
- La licencia seleccionada para la base de datos es Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 – 64 bit Production.
- El cliente para el desarrollo de la base de datos es Oracle SQL Developer 19.4.0.354.
- El entorno para el desarrollo de las pruebas y verificación del sistema es Selenium (3.141.59).

Concepto	Cantidad	Precio/Unidad (€)	Total (€)
PyCharm Community	1	0	0
Bot Framework Emulator V4	1	0	0
Dialogflow Standard Edition	1	0	0
Licencia Oracle Database 11g	1		
Oracle SQL Developer	1	0	0

Selenium	1	0	0
Subtotal hardware			0

Tabla 13 - Licencia de productos software

2.3 Mano de obra de ingeniería

La tabla de la mano de obra de ingeniería solo incluye las horas totales dedicadas al proyecto. Está dividido en tres partes: análisis y diseño, desarrollo y verificación. Por temas legales y contractuales con la empresa, no se especifica el precio real de cada hora de trabajo.

Concepto	Cantidad (horas)	Precio/Unidad (€)	Total (€)
Análisis	140	25	3.500
Diseño	120	25	3.000
Desarrollo	380	20	7.600
Implementación	30	20	600
Pruebas	50	20	1.000
Publicación	0	20	0
Preventa	0	20	0
Soporte	0	20	0
Calidad	0	20	0
Subtotal mano de obra	720		15.700

Tabla 14 - Mano de obra en ingeniería

2.4 Presupuesto total

El coste de la herramienta será el cómputo total de los costes de los tres apartados anteriores.

Concepto	Precio (€)
Equipamiento Hardware	131,66
Equipamiento Software	0,00
Mano de obra en ingeniería	15,700
Subtotal	15.831,66
Beneficio Industrial (6%)	949,90
Subtotal	16.781,56
IVA (21%)	3.524,13
Total	20.305,69

Tabla 15 - Presupuesto total

El presupuesto total asciende a la cuantía de VEINTE MIL TRESCIENTOS CINCO EUROS Y SESENTA Y NUEVA CÉNTIMOS#20.305,69#.

3. Documentos Técnicos

En este capítulo, se pretende dar una descripción de todos los documentos técnicos.

3.1 Requisitos

En este capítulo se describen los requisitos del sistema. Están divididos en dos categorías: funcionales, definen el comportamiento del sistema y los no funcionales, que indican cómo debe realizarse dicho sistema, imponiendo restricciones en su implementación. Los requisitos son de todo el proyecto incluidos los de la plataforma de contenidos y son establecidos por el equipo y el personal de NRH.

3.1.1 Requisitos funcionales

Los requisitos funcionales de la tabla 16 definen las operaciones de la herramienta. Cada uno de ellos está definido por:

- El identificador es un valor único del requisito. Su estructura es RFXX donde XX indica el número de secuencia del requisito.
- El nombre es un resumen del requisito.
- La descripción es una definición breve.
- Por último, la prioridad puede tener 3 niveles distintos: A (Alta) más prioritaria, M (Media) prioridad media y B (Baja). Esta prioridad influye en el orden de desarrollo.

ID	Nombre	Descripción	Prioridad
MilenaBot: requisitos generales			
RF01	Multilinguaje	El chatbot debe identificar el idioma del usuario y contestarle en el mismo idioma que la plataforma Milena.	A
RF02	Mensaje con texto	El usuario debe poder visualizar la respuesta del bot en texto.	A

RF03	Personalización mensaje de bienvenida con texto	MilenaBot deberá responder con un mensaje de bienvenida al usuario con el nombre del usuario identificado.	B
MilenaBot: autenticación			
RF04	Mecanismos de autenticación	El usuario debe ser identificado en el chatbot por su usuario de la plataforma Milena.	A
RF05	Control de permisos	La herramienta debe comprobar si el usuario tiene acceso a los recursos que está solicitando.	A
RF06	Mensaje de respuesta ante denegación de permisos	MilenaBot debe responder con un mensaje indicándole que no le ha entendido. No debe informar al usuario que no tiene permisos de acceso según el control de RF05.	A
MilenaBot: subsistema de procesos y base de conocimiento			
RF07	Mensaje con elemento multimedia	El proceso debe poder adjuntar un elemento multimedia a la respuesta.	A
RF08	Mensaje con opciones	El proceso debe poder sugerirle mediante botones al usuario interfaces relacionadas con la actual.	A
RF09	Acceso indistinto a recursos / pasos del proceso	El chatbot debe ofrecer la posibilidad de cambiar a otro proceso, o si el tipo de proceso lo permite, cambiar a un paso exacto.	A
RF10	Acceso a los contenidos de la BD	La herramienta accede a la base de datos para obtener los recursos de los requisitos RF02, RF08 Y RF09 y la ubicación del RF07.	A

RF11	Árbol de funcionalidades	El usuario debe poder acceder a un proceso en el que se le informe de las funcionalidades de MilenaBot.	M
RF12	Solicitud de ayuda	El usuario debe poder acceder a una explicación sobre el funcionamiento del chatbot.	M
RF13	Lenguaje “chatty”	MilenaBot deberá responder preguntas informales de forma similar a una conversación humana.	M
RF14	Consultar opinión usuarios	El bot debe disponer de alguna herramienta que permita conocer la opinión del usuario sobre el proceso al que ha accedido.	B
MilenaBot: subsistema de formularios			
RF15	Reducir preguntas	El bot debe optimizar el número de acciones / expresiones que debe introducir el usuario para responder durante un formulario mediante la identificación de entidades de los RF33, RF34, RF35 y RF36.	A
RF16	Cancelar formulario	La herramienta debe permitir al usuario cancelar el formulario en cualquier paso.	A
RF17	Resumen de solicitud	Todos los formularios deben resumir todos los parámetros introducidos por el usuario antes de procesar la solicitud.	A
RF18	Confirmar solicitud	Todos los formularios deben emitir un mensaje de confirmación antes de proceder con la solicitud	A

RF19	Modificar parámetros	Todos los formularios deben permitir que el usuario modifique cualquier parámetro antes de confirmar la solicitud.	A
RF20	Verificación de los parámetros modificados	La herramienta debe volver a verificar que los parámetros del RF19 cumplen con los respectivos requisitos RF23, RF26, RF28 y RF30.	A
MilenaBot: subsistema de formularios (Vacaciones)			
RF21	Solicitar vacaciones	El usuario debe poder solicitar vacaciones introduciendo fecha de inicio y fin y los comentarios de solicitud pertinentes.	A
RF22	Consultar días disponibles y estado solicitudes	El usuario debe recibir información sobre el estado de sus solicitudes vacacionales y del número de días disponibles.	A
RF23	Verificar vacaciones	La herramienta debe verificar los parámetros del RF21. Esto es que el periodo no exceda del número de vacaciones disponibles y las fechas de inicio y fin sean días laborables.	A
MilenaBot: subsistema de formularios (Permisos retribuidos)			
RF24	Solicitar permiso retribuido	El usuario debe poder solicitar los permisos retribuidos de la plataforma introduciendo las fechas y / u horas y los comentarios de solicitud pertinentes.	A
RF25	Informar sobre el N° de días u horas disponibles por permiso.	El usuario debe recibir información sobre el número de días u horas	A

		disponibles según el permiso seleccionado en RF24.	
RF26	Verificar permiso retribuido	La herramienta debe verificar los parámetros de RF24. Es decir que el periodo temporal no excede del número de vacaciones disponibles y que las fechas de inicio y fin sean días laborables.	A
MilenaBot: subsistema de formularios (Nómina)			
RF27	Solicitar nómina	El trabajador debe poder solicitar la nómina de cualquier mes y año.	A
RF28	Verificar nómina	MilenaBot debe comprobar que el mes y el año introducidos existan y que el trabajador tenga la nómina de esa fecha disponible en la plataforma Milena.	A
MilenaBot: subsistema de formularios (Certificado de retenciones)			
RF29	Solicitar certificado de retenciones	El trabajador debe poder solicitar el certificado de retenciones de cualquier año.	A
RF30	Verificar certificado de retenciones	MilenaBot debe comprobar que el mes y el año introducidos existan y que el trabajador tenga el certificado de retenciones de esa fecha disponible en la plataforma Milena.	A
MilenaBot: subsistema de formularios (Incidencia)			
RF31	Registrar incidencia	La herramienta debe permitir registrar una incidencia a través de su chatbot.	A
Servicios cognitivos (IA – Intérprete de lenguaje natural)			

RF32	Detección de intents	El servicio cognitivo seleccionado debe poder detectar la intención del usuario en función de un umbral.	A
RF33	Identificar entidad: día	El servicio cognitivo seleccionado debe detectar el día en la expresión del usuario.	A
RF34	Identificar entidad: mes	El servicio cognitivo seleccionado debe detectar el mes en la expresión del usuario.	A
RF35	Identificar entidad: año	El servicio cognitivo seleccionado debe detectar el año en la expresión del usuario.	A
RF36	Identificar entidad preconfigurada	El servicio cognitivo debe permitir la configuración para la detección de entidades específicas determinadas por el desarrollador.	A
MilenaBot: cliente			
RF37	Text-to-speech	El cliente debe tener algún mecanismo que al activarlo le permita convertir el texto a voz sintética.	B
RF38	Speech-to-text	El cliente debe tener algún mecanismo que al activarlo le permita convertir la voz del usuario en texto.	B
Plataforma de contenidos: interfaz web			
Plataforma de contenidos: interfaz web (creación de intent)			
RF39	Crear intent	El trabajador de NRH debe poder introducir el nombre del intent que considere.	A
RF40	Verificar intent	La herramienta deber comprobar que el intent de RF39 es único.	A

RF41	Seleccionar idioma del intent	El trabajador de NRH debe seleccionar el idioma del intent del RF39 que va a introducir.	A
RF42	Introducir preguntas / expresiones de entrenamiento	El usuario de NRH debe introducir mínimo 10 preguntas o frases de entrenamiento por intent.	A
RF43	Verificar pregunta / frase de entrenamiento es exclusiva	La herramienta debe verificar que la pregunta de entrenamiento es única para todos los intents.	A
RF44	Introducir respuesta textuales	El usuario de NRH debe añadir la respuesta textual al intent del RF39.	A
RF45	Dar formato a las respuestas	El usuario de NRH una vez introducida la respuesta del RF42, debe poder modificar el formato: establecer el tamaño de letra, cambiar el texto a negrita, hipervínculos y enumerar una lista.	A
RF46	Adjuntar elemento multimedia a la respuesta	La plataforma debe disponer de algún mecanismo que le permita adjuntar los elementos multimedia a la respuesta del RF42.	A
RF47	Verificar el formato del elemento multimedia	La herramienta debe comprobar si el formato del recurso es válido.	A
RF48	Asignar intent a un grupo	La plataforma debe permitir asignar al intent del RF39 a un grupo de recursos.	A
Plataforma de contenidos: interfaz web (Visualización de expresiones no-matched)			
RF49	Visualizar expresiones unmatched	El trabajador de NRH debe poder visualizar todas las expresiones que no han realizado matched en el servicio cognitivo.	A

RF50	Añadir expresión a un intent	El usuario debe poder añadir la expresión a un intent ya existente.	A
RF51	Descartar expresión	El usuario debe poder descartar la expresión.	A
Plataforma de contenidos: interfaz web (Visualización de intents)			
RF52	Visualizar intents	El usuario de NRH debe poder visualizar todos los intents con sus respectivas preguntas y respuestas asociadas, el nombre del usuario que lo ha creado o modificado y el grupo de recursos al que pertenece.	A
RF53	Filtrar por todos los parámetros	El usuario debe poder filtrar los intents por todos los parámetros que visualiza.	A
RF54	Modificar intent	El usuario de NRH debe poder modificar todos los parámetros del intent.	A
RF55	Eliminar intent	El usuario de NRH debe poder eliminar el intent.	A
Plataforma de contenidos: interfaz web (Flujo de conversación)			
RF56	Sugerir opciones en la respuesta	El usuario debe poder insertar opciones / botones relacionados con la información del intent actual para llevarlo a otro paso de un proceso o a un formulario.	A
Plataforma de contenidos: autenticación			
RF57	Crear usuario	El gestor debe poder dar de alta nuevos usuarios en la aplicación con sus respectivos permisos.	M

RF58	Definición de permisos de acceso personalizados.	La aplicación debe permitir otorgar distintos permisos de acceso para cada usuario/rol definido en el sistema.	M
RF59	Eliminar usuario	El gestor debe poder dar de baja usuarios en la plataforma.	M
RF60	Registrar responsable del intent	El gestor debe poder comprobar quién y cuándo ha registrado el intent / pregunta en la plataforma.	M
Monitorización y estadísticas			
RF61	Logs Server	MilenaBot debe disponer de un sistema de logs donde se registren todas las acciones de los usuarios y los problemas internos del servidor.	B
RF62	Registro de expresiones no-matched	El asistente, tras la no identificación de la expresión, debe registrarla en la BD junto al idioma.	A
RF63	Extracción de datos de sesión del usuario	El chatbot tiene que disponer de algún mecanismo que le permita extraer datos sobre la sesión del usuario y las valoraciones del RF14.	B
RF64	Visualización de datos y estadísticas	La plataforma debe disponer de alguna aplicación para la visualización de datos y estadísticas.	B
RF65	Mecanismo de monitorización	El servidor debe ser monitorizado constantemente y emitir eventos en caso de fallo.	B

Tabla 16 - Requisitos funcionales

3.1.2 Requisitos no funcionales

La tabla 17 muestra todos los requisitos no funcionales, se definen de igual forma que los funcionales salvo que el identificador está formado por RNFXX, donde XX es el número de secuencia del requisito.

ID	Nombre	Descripción	Prioridad
MilenaBot: requisitos generales			
RNF01	Acceso directo	El usuario tiene que poder interactuar con el bot una vez accedido a Milena independientemente del módulo en el que se encuentre.	A
RNF02	Sesión confidencial	Los usuarios no podrán acceder a la información introducida por otros usuarios.	A
RNF03	Peticiones anonimizadas	Las peticiones enviadas al servicio de interpretación de lenguaje natural deben ser anónimas.	A
RNF04	Control de los recursos de sesión	La herramienta debe tener un mecanismo para reservar recursos cuando el usuario se conecta al bot y destruirlos cuando se desconecta.	M
RNF05	Control de errores	El chatbot no debe enviar ningún mensaje al usuario advirtiéndole de que ha sucedido un error o excepción.	A
MilenaBot: subsistema de formularios			
RNF06	Datos solicitados a Milena	Los datos solicitados deben ser los mismos que se solicitan en Milena para cada servicio.	A

RNF07	Datos de formulario procesados servidor	Los datos recopilados durante un formulario deben ser exclusivamente procesados por el servidor. No deben ser enviados al servicio cognitivo.	A
MilenaBot: subsistema de procesos y base de conocimiento			
RNF08	Optimizar peticiones	El diseño debe optimizar el número de peticiones que se realizan a la API de interpretación de lenguaje natural.	A
MilenaBot: cliente			
RNF09	Canal de distribución	El canal de distribución debe ser exclusivamente web chat.	A
RNF10	Acceso simple	El usuario debe poder localizar con facilidad el bot dentro de la plataforma.	A
RNF11	Ventana desplegada	El chat del bot no puede superar el 25 % de la interfaz web de Milena.	A
RNF12	Colores corporativos	La interfaz debe seguir la guía de estilos corporativos.	A
RNF13	Identificación mensajes en chat	La interfaz debe permitir identificar los mensajes del usuario y del chatbot.	A
Servicios cognitivos			
RNF14	Entrenamiento	El servicio debe permitir el entrenamiento mediante registros internos (conversaciones del usuario) y externos (mediante la creación de intents)	A
RNF15	Nº de intents por agente	El número mínimo de intents por agente debe ser de 1.500.	A
RNF16	Capa básica sin coste	El servicio cognitivo debe disponer de una capa básica sin coste por petición.	A

RNF17	Límite de peticiones por segundo	El número mínimo de peticiones por segundo tiene que ser 5.	A
RNF18	Soporte	El servicio contratado debe ofrecer soporte al desarrollador en caso de que se produzca una incidencia.	A
RNF19	Disponibilidad	La disponibilidad del servicio contratado debe ser igual o superar el 99.9%	A
RNF20	Conexión cifrada serv. cognitivo – servidor	La conexión debe ser confidencial, íntegra con una disponible superior o igual al 99.9%.	A
Plataforma de contenidos: interfaz web			
RNF21	UI Web	El usuario de NRH debe introducir los contenidos a través de una interfaz web.	A
RNF22	Procedimiento guiado	El usuario de NRH debe ser guiado durante el procedimiento para introducir los contenidos.	M
Plataforma de contenidos: BD			
RNF23	Sistema gestor de base de datos.	La BD debe disponer de una herramienta para guardar, modificar y organizar la información	A
RNF24	Consistencia BD-Dialogflow	Las preguntas e intents de la BD deben coincidir.	A
RNF25	Mecanismo de comprobación de consistencia	El administrador de los servicios cognitivos debe disponer de un servicio para comprobar la consistencia siguiendo el RNF25.	A
RNF26	Restricciones entre tablas	El diseño debe evitar inconsistencias entre tablas.	A

RNF27	Acceso remoto	La base de datos debe tener algún servicio de acceso mediante API o un cliente ligero.	A
RNF28	Conexión cifrada BD – servidor	La conexión debe ser confidencial, íntegra y estar siempre disponible.	M
Disponibilidad			
RNF29	Disponibilidad mínima del 99.9%	La disponibilidad del chatbot debe ser igual o superar el 99.9%	A
RNF30	Nº de accesos simultáneos	El número de accesos simultáneos no puede ser superior al límite de peticiones por segundos establecidos por los servicios cognitivos.	A
Seguridad			
RNF31	Acceso restringido	El acceso a la plataforma Milena solo se puede realizar mediante la plataforma Milena.	A
Despliegue			
RNF32	Despliegue red local	La herramienta debe poder ser desplegada en la red corporativa.	A
MilenaBot: usabilidad			
RNF33	Tiempo de respuesta	El tiempo de respuesta no debe exceder los 2 segundos.	A
Reglamento General de Protección de Datos (RGPD)			
RNF34	RGPD	El asistente deberá cumplir el vigente reglamento sobre protección de datos	A
RNF35	Cumplimiento de la legalidad vigente	La herramienta debe cumplir con los requisitos legales establecidos por la empresa.	A

Tabla 17 - Requisitos no funcionales

3.2 Diseño de la herramienta

En este apartado se describe el diseño de las dos herramientas que conforman el sistema: el chatbot y la plataforma para contenidos.

3.2.1 Diseño del back-end

Desde el punto de vista de un diseño eficiente y práctico para el usuario, es recomendable que el flujo de conversación siga las recomendaciones que Microsoft sugiere en la documentación del enlace [20].

Para entender cómo proceder con el diseño, se puede tomar como punto de partida el diseño de una interfaz web. Por norma general, las aplicaciones se inician con una interfaz principal a la que los usuarios acceden y es esa la que proporciona información para acceder a otras con distintas funcionalidades. Esto es un aspecto que se debe implementar en los bots. En el paso inicial, se da la bienvenida al usuario, se establecen las expectativas y se deja que tome el control. Para que el usuario comprenda qué puede hacer la herramienta, es recomendable que se reserven intents que expliquen las funcionalidades. Los formularios de las aplicaciones web solicitan fragmentos de información. En un bot, se recopila la misma información pero con una conversación activa. El objetivo es conseguir un flujo de conversación natural con capacidad para controlar interrupciones y cambios de temas en la conversación correctamente.

La herramienta es eficaz si cumple los siguientes objetivos: solucionar fácilmente el problema planteado por el usuario con el número mínimo de pasos, ser más rápido que cualquier servicio alternativo, estar en plataformas con fácil accesibilidad y que el tiempo de respuesta sea adecuado.

3.2.1.1 Diseño y control del flujo de conversación

El objetivo de este apartado es demostrar que el diseño implementado mediante estados es la solución óptima al problema planteado. Antes de proceder, se debe explicar las diferencias de un proceso y un formulario desde el punto de vista del diseño. Cuando el usuario decide realizar una solicitud, por norma general, los servicios cognitivos interpretan que es un intent de ese tipo lo que desemboca en la identificación de las entidades. En función de eso, se

realizan las preguntas sobre los parámetros que no se han identificado. El número de interfaces de un formulario varía en función de la expresión del usuario. En cambio, los procesos no recogen información, simplemente guían a través de distintas interfaces.

El primer paso es enumerar los requisitos que afectan directamente al diseño. Estos son los referentes a los procesos:

1. El diseño debe optimizar el número de peticiones que se realizan a la API de interpretación de lenguaje natural (RNF08).
2. El chatbot debe ofrecer la posibilidad de cambiar a otro proceso, o si el tipo de proceso lo permite, cambiar a un paso exacto (RF09).
3. La herramienta accede a la base de datos para obtener los recursos (RF10).

Estos son los que afectan a los formularios:

1. Los datos recopilados durante un formulario deben ser exclusivamente procesados por el servidor. No deben ser enviados al servicio cognitivo (RNF07).
2. El bot debe optimizar el número de acciones / expresiones que debe introducir el usuario para responder durante un formulario mediante la identificación de entidades (RF15).
3. La herramienta debe permitir al usuario cancelar el formulario o antes de ser procesados (RF16).

El segundo paso es asociar a cada intent del NLU con un proceso o un formulario e integrarlo con un manejador que lo invoque. Actualmente, hay algunos proyectos de código abierto que ya han resuelto este problema. El enlace [21] de Code Project y el ejemplo de Microsoft del [22] mapean mediante un diccionario. Una vez que la herramienta reconoce el intent, se accede a un proceso o formulario específico desde donde se responde al usuario y se almacena el estado e información de la conversación. Generalmente, estos procesos tienen más de un paso, por lo que el usuario debe acceder en repetidas ocasiones a los recursos. Bot Framework resuelve este problema mediante su propia librería de diálogos y almacena en memoria el estado de la conversación. El enlace [23] asigna a cada paso del diálogo a una función asíncrona. Este enfoque de Microsoft está orientado a formularios donde los recursos estén integrados en el proyecto, no es el caso de MilenaBot.

3.2.1.1.1 Diseño de los procesos y base de conocimiento

Las dos principales diferencias que caracterizan al servidor es el acceso a la base de datos para enviar las respuestas y la optimización del número de peticiones que se realizan al servicio cognitivo. Además, los contenidos están relacionados entre sí, es decir el usuario de NRH puede incluir una opción que haga referencia a un formulario.

Para comprender esta idea, se presenta el siguiente ejemplo. El usuario solicita información para teletrabajar y el bot inicia un proceso, sin embargo, llegado a un paso no puede continuar. En ese momento, desea poner una incidencia pinchando en el botón sugerido. Esto es un proceso totalmente ajeno al teletrabajo. Una vez que es completado, el usuario quiere volver directamente al paso exacto donde estaba. Si se analiza desde el punto de vista del diseño, el usuario ha cambiado de proceso a formulario sin haber finalizado el actual y posteriormente, ha regresado al paso donde lo había dejado. Esto sería complejo de implementar si se utilizase la librería de diálogos de Microsoft.

El diseño planteado permite que si el usuario sigue las opciones sugeridas, solo se envíe la primera expresión inicial. Mediante el ejemplo de las figuras 32 y 33, se explica cómo este diseño cumple todos los requisitos. Inicialmente, el usuario se encuentra en el estado “*Inicial*”, desde aquí se realizan las peticiones a la NLU.

Cada respuesta del servicio cognitivo tiene asociado un intent con el que se accede a la base de datos. De modo que el mensaje que recibe el usuario no es “*IntTelework*”, sino un texto y una imagen seguido de 4 botones con las opciones: “*avanzar*”, “*retroceder*”, “*salir*” e “*informar del problema*”. El servicio cognitivo solo registra las frases de entrenamiento y la referencia al proceso o formulario correspondiente.

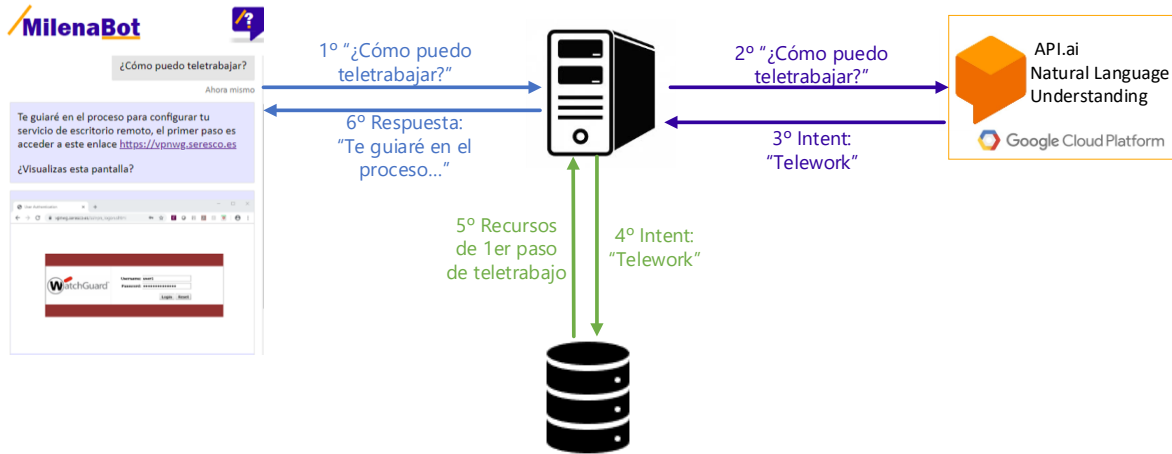


Figura 32 - Diseño del flujo de conversación de un proceso: paso 1º

El usuario puede responder simplemente pinchando en uno de los botones. Si optase por introducir una expresión textual, se enviaría al servicio cognitivo para su procesamiento. Con este planteamiento se reduce el número de peticiones realizadas.

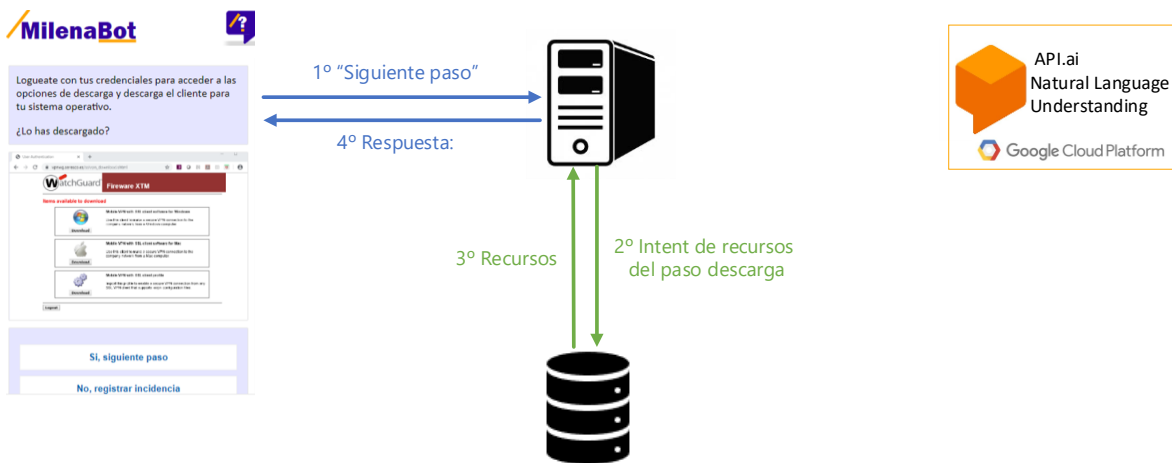


Figura 33 - Diseño del flujo de conversación de un proceso: paso 2º

Por último si el usuario deseara acceder a un paso específico puede expresar por ejemplo: "Quiero acceder al paso de descarga" y cambiaría directamente tras la interpretación del NLU. Esto permite cumplir con el RF09.

La figura 34 muestra el ejemplo desde el punto de vista de los estados. Todos los intents de procesos y base de conocimiento son procesados por la misma función asíncrona "Dialog_State".

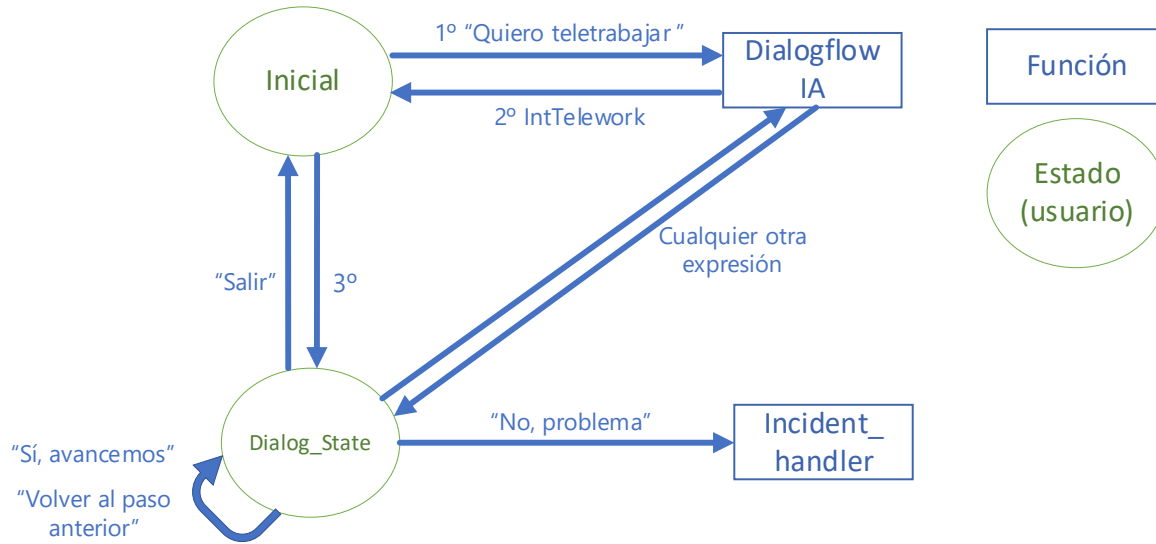


Figura 34 - Diagrama de flujo de ejemplo de un proceso

La figura 35 muestra qué sucede si la herramienta no tiene el contenido sobre lo que se refiere el usuario. En ese caso, se envía un mensaje informándole de que la aplicación no ha conseguido interpretar su expresión. Aunque este proceso parezca sencillo, implica también el establecimiento de la conexión con la base de datos. El objetivo es almacenar todas las expresiones del usuario con su respectivo lenguaje para que pueda ser visualizado en la plataforma de contenidos.

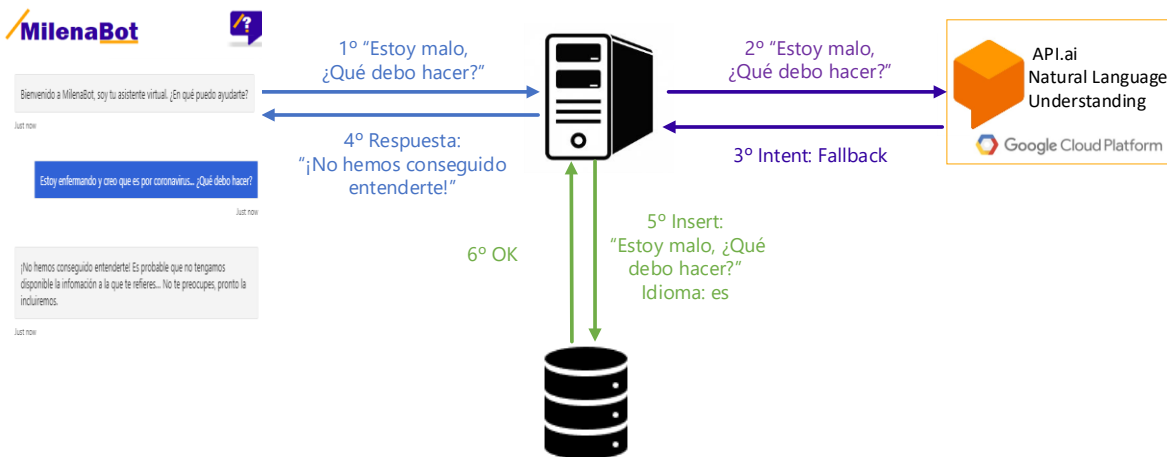


Figura 35 - Diseño de flujo tras contestación no-matched

Esto es similar a lo que sucede cuando no se tienen los permisos para acceder al contenido que se está solicitando, dependiendo del usuario se puede acceder a los distintos procesos y formularios. En todas las interacciones se realiza la verificación, y el usuario no recibe

información sobre este proceso. En el caso de que su acceso sea correcto, sucede lo descrito en las figuras 32 y 33. En caso de acceso incorrecto recibe el mismo mensaje que el de la figura 35. Esto es lo que sucede en la figura 36.

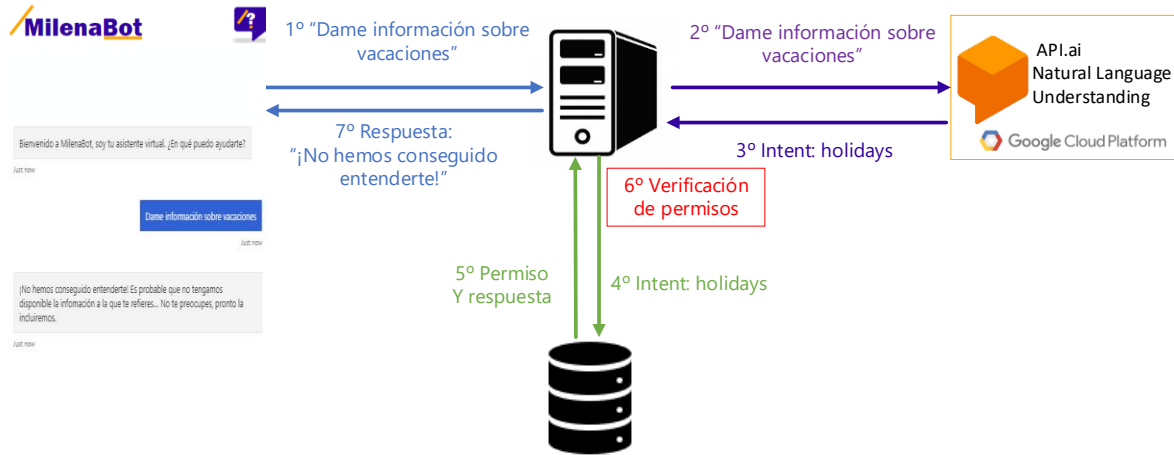


Figura 36 - Diseño del flujo de conversación, control de permisos

3.2.1.1.2 Diseño de formularios

Los formularios tienen la misma estructura que el proyecto de Code Project del enlace [21]. Cada uno tiene un intent y una serie de funciones asíncronas asociadas. A diferencia de los procesos, no se accede a la base de datos. La figura 37 muestra el primera paso del formulario vacaciones.

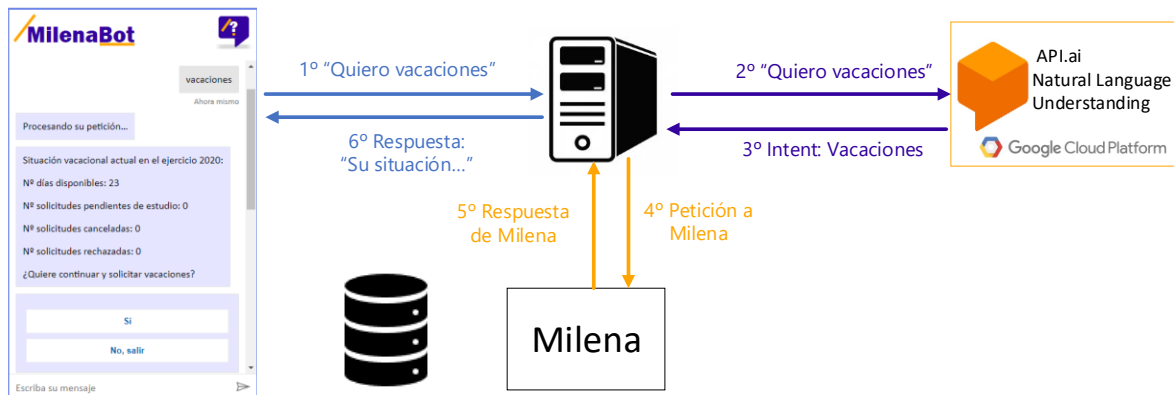


Figura 37 - Diseño del flujo de conversación de un formulario: paso 1º

En los siguientes pasos se realiza la validación de los datos introducidos y su almacenamiento en memoria en el propio servidor. De este modo, se cumple con los RF15 y RF16. Esto se muestra en la figura 38.

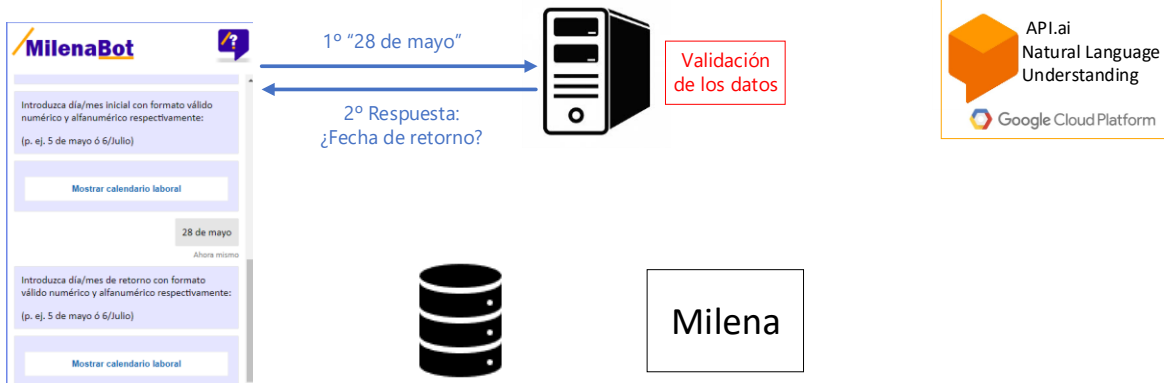


Figura 38 - Diseño del flujo de conversación de un formulario: paso 2º

El último paso (figura 39) es igual para todos los formularios. Contiene un resumen de todos los parámetros introducidos y obliga al usuario a confirmar si son estos los datos que desea enviar en su solicitud. Normalmente, es en este estado cuando se realiza la conexión con los servicios de Milena.

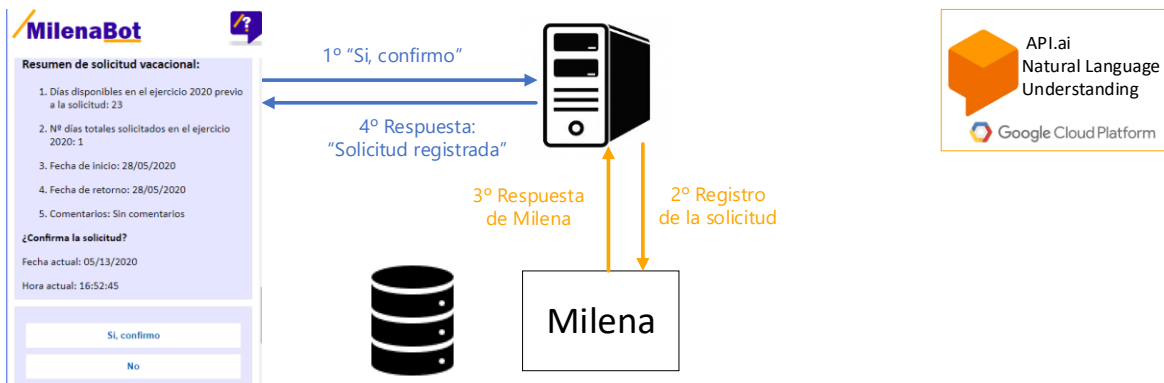


Figura 39 - Diseño del flujo de conversación de un formulario: paso 3º

La figura 40 muestra el ejemplo desde el punto de vista de los estados. Desde el estado inicial se accede a una función que comprueba las entidades identificadas por el servicio de interpretación de lenguaje natural, *“holidays_IAresponse”*. A continuación se accede al manejador, *“holidays_handler”*. En este paso, se selecciona a qué estados del formulario se debe acceder. Si la expresión del usuario incluye una fecha que cumpla con los requisitos, no se pregunta por ella. En los estados, solo se comprueba la expresión del usuario y se almacena en memoria o en el caso de ser el último, se procesa. Los mensajes se envían desde el manejador.

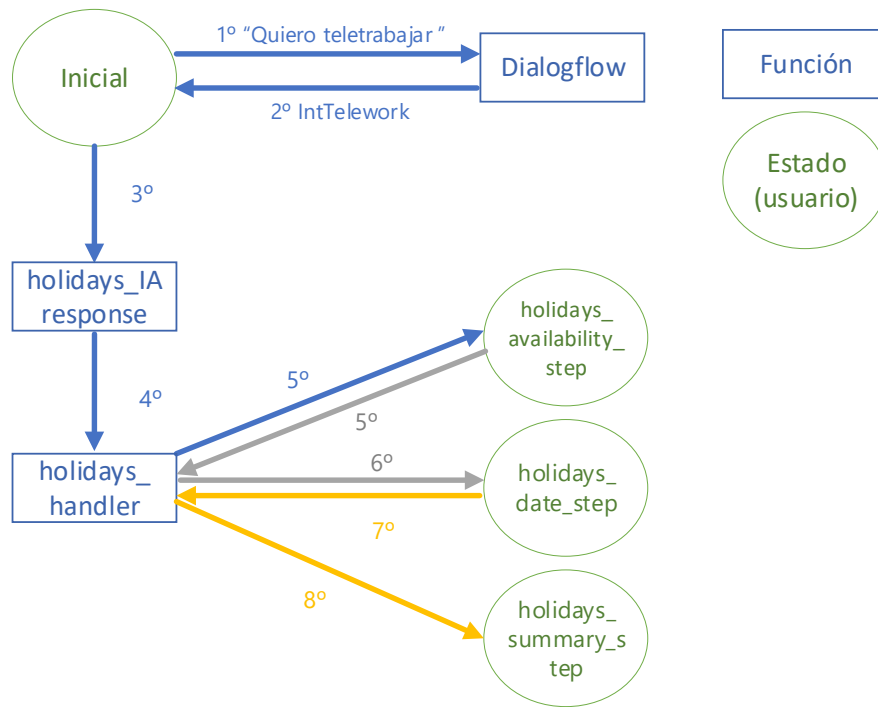


Figura 40 - Diagrama de flujo de ejemplo de un formulario

Por otra parte, es posible que el lector piense que la solución óptima es implementar un patrón estado. No hay ningún proyecto en el que se haya almacenado el estado mediante un descriptor de acceso en una clase concreta.

El almacenamiento actual de estados sigue la recomendaciones de buenas prácticas de Microsoft. Se recomienda el estado de conversación para realizar el seguimiento del contexto de la conversación. En cambio, el estado de usuario está orientado a realizar el seguimiento de la información recopilada al usuario, como por ejemplo, las respuestas del formulario de incidencia.

3.2.2 Diseño del front-end

La capa de cliente o front-end es la capa visible para los usuarios, además de ser la única con la que interactúan directamente. A su vez, esta se comunica exclusivamente con la capa servidor.

Durante la etapa de desarrollo, la capa de cliente puede ser emulada mediante Microsoft Bot Emulator [24]. Es una aplicación standalone que permite interactuar directamente con el servidor que utiliza Microsoft Bot Framework. Esta herramienta está orientada únicamente a probar el funcionamiento del bot y corregir posibles errores mediante la inspección de los mensajes en formato JSON. No está diseñada para ser utilizada en la etapa de producción. La figura 41 explica el testeo del back-end del apartado 3.5. También se puede conectar a bots desplegados en el cloud de Azure Bot Service.

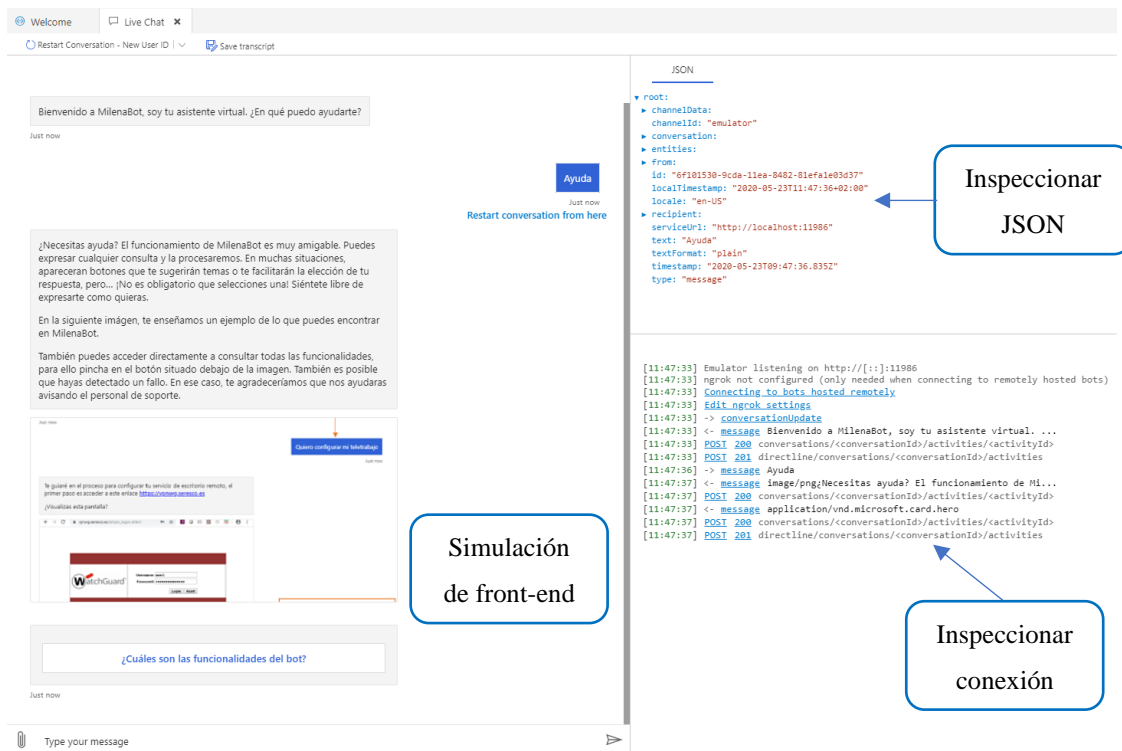


Figura 41 - Testeo con Bot Framework Emulator

Para la puesta en producción del sistema e integración con la plataforma web, Milena, se utiliza la tecnología Bot Framework Web Chat. El despliegue depende del servidor. A raíz de la elección de la arquitectura del apartado 1.6.3, se ha optado por la solución corporativa.

3.2.2.1 Comunicación servidor - cliente

La comunicación entre el cliente y el servidor no es trivial. Es diferente a un servicio API REST habitual. Para poder establecer dicha comunicación, es necesario seguir un protocolo establecido por Microsoft denominado Direct Line [26]. Cabe destacar que si se hubiese optado por el despliegue en el cloud, no sería necesario. Azure ofrece la integración completamente gratuita.

Este aspecto se soluciona mediante el paquete Offline-Directline del enlace [27]. La comunicación funciona de forma que se exponen un total de tres endpoints:

1. Directline, con el cual se comunica el cliente para enviar y recibir mensajes.
2. Conversation, con el cual se comunica el middleware para enviar mensajes.
3. Botstate, con el que se comunica el middleware para actualizar y leer el estado de la conversación.

De esta forma, se genera una estructura de tipo Proxy que sirve de intermediario el cliente y el bot, por lo que en ningún momento se realiza una comunicación directa entre ambos. Este protocolo es todo lo que se necesita para establecer la comunicación entre las dos capas.

3.2.3 Diseño de los agentes (API.ai Dialogflow)

Esta apartado pretende ofrecer una explicación sobre el diseño de los agentes para mejorar el rendimiento de las llamadas a la API. Se utiliza para la detección de intents en el servidor y para la administración de contenidos de los distintos agentes.

Siguiendo las recomendaciones de buenas prácticas de Google Cloud Platform del enlace [28], se reutiliza la misma instancia de cliente de sesión para todas las solicitudes de la aplicación. Eso se debe a que una vez instanciado, se realiza la autenticación, la autorización y la generación de un token de acceso. Este proceso puede producir una espera de unos pocos segundos. Para las llamadas posteriores, el cliente reutiliza el mismo token durante su período de validez que es de aproximadamente una hora. Cada vez que caduca, es el propio cliente de sesión el que actualiza el token automáticamente.

El cliente es robusto y soporta usuarios simultáneos. Sin embargo, hay algunos aspectos que deben ser gestionados por el desarrollador: los errores y el número de reintentos de conexión. Los errores se describen en la documentación de los errores de las API de Cloud. Respecto a los reintentos, si el segmento de red entre el servidor y GCP está congestionado y se supera el timeout de conexión. El chatbot redirige al usuario a una interfaz en el que le informa del problema y le permite reintentar la conexión. Este comportamiento es similar al de una interfaz web real. Esto se muestra en la figura 42.

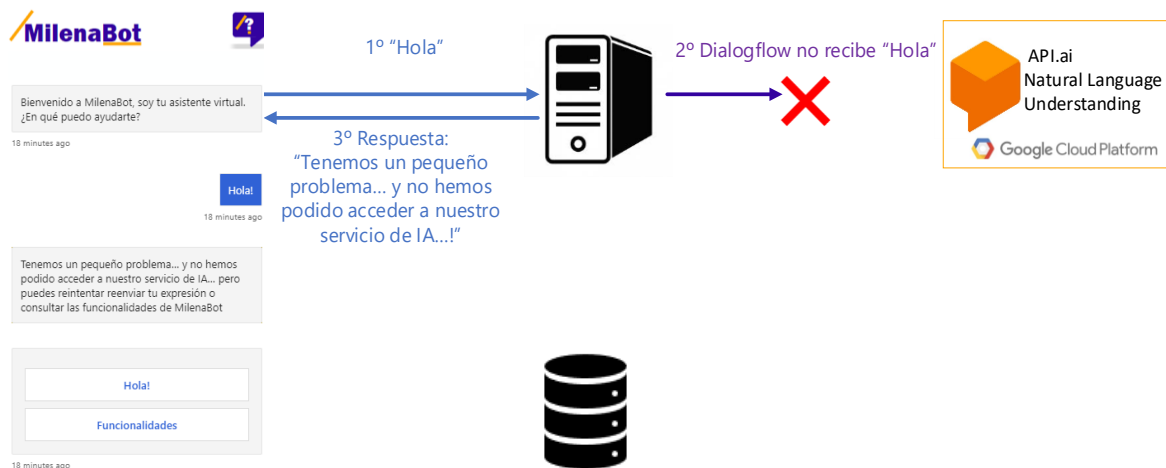


Figura 42 - Diseño de flujo en fallo por conexión con Dialogflow

Según el cloud de Google, la API no está diseñada para soportar picos de actividad abruptos de carga. El tráfico real generado por los usuarios no tiene este comportamiento. De este modo, la API no responde adecuadamente a abruptas demandas de carga. Para la realización de la prueba de carga, se debe aumentar de forma gradual la tasa de solicitudes hasta alcanzar la tasa deseada.

El tamaño del agente afecta al número de frases de entrenamiento que se necesitan para un correcto funcionamiento. Dialogflow recomienda al menos 10 frases por intent y este es el número mínimo que se solicita en la plataforma de contenidos. Se recomienda incluir variaciones de preguntas, verbos y sinónimos para cubrir el amplio abanico de solicitudes del usuario. También se ha habilitado la corrección ortográfica automática.

Las recomendaciones para la identificación de formularios y sus respectivas entidades no afectan directamente al diseño de la herramienta, pero sí que obliga a tenerlas en cuenta para la correcta detección del intent. La pregunta solo debe tener establecido la misma estructura

de entidad. Por ejemplo, para la pregunta “¿Cuál es mi nómina de junio del 2020?”, la entidad configurada como @month es “junio”. Para otro caso, “Nómina de julio del 18”, se debe establecer “julio”, pero no “de julio”. Además, como regla general, se debe tener tres veces más frases de entrenamiento que entidades.

El umbral actual está establecido en 0.3. Si es demasiado alto, solo las consultas que tengan coincidencias casi exactas con las frases de entrenamiento serán reconocidas como intent.

3.2.4 Diseño de la base de datos (Oracle)

En este apartado se describe el diseño de la base de datos. Es un elemento fundamental en el sistema porque en ella se almacenan todos los contenidos de procesos y base de conocimiento. La base de datos seleccionada es Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 – 64 bit Production. Respecto al DBA (Database Administrator), se utiliza el Oracle SQL Developer versión 19. Es una herramienta para guardar, modificar y organizar la información. Los miembros responsables del proyecto tomaron la elección de esta tecnología, se debe a que proporciona características más adaptadas al proyecto.

Además, Oracle utiliza el estándar de SQL. Es un lenguaje de consultas muy popular y de uso extendido. Gracias a que se trata de un lenguaje de alto nivel, simplifica la integración con la plataforma de contenidos y el servidor.

La base de datos desarrollada, “MilenaBot”, está basada en un modelo relacional. Su diseño se muestra en el siguiente diagrama entidad-relación de la figura 43. Es necesario para la herramienta que se reflejen las restricciones de los requisitos, de manera que nunca sea posible realizar acciones no válidas. Para ello, se utilizan reglas de modelo de datos y restricción. Por ejemplo, es imposible insertar en la tabla “RESOURCE_MESSAGE” si no existe el intent o el campo “MESSAGE” es nulo. Además, también delimita que se almacenen datos que no son coherentes, por ejemplo, registrar preguntas sin tener el idioma seleccionado.

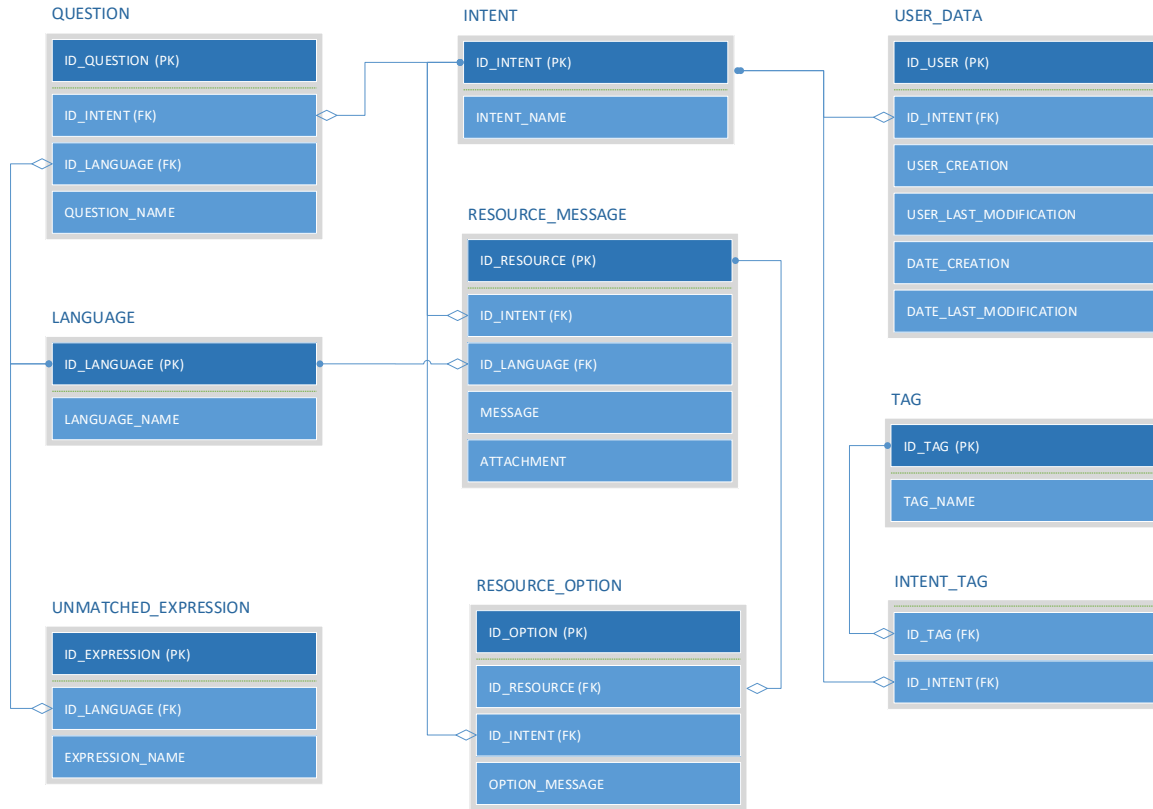


Figura 43 - Diagrama entidad - relación de la plataforma de contenidos

3.3 Descripción del sistema

Dentro de la descripción del sistema se pretende definir cada uno de los componentes que lo conforman, indicando todos sus aspectos.

3.3.1 Casos de uso

Un caso de uso es la definición de una funcionalidad del sistema que se basa en los requisitos establecidos en el punto 3.1. Solo están presentes aquellos con correspondencia con el RF01 al RF38. Cada uso está definido por los siguientes elementos:

- El Identificador que está formado por CU-XX siendo XX un número incremental.
- El nombre describe brevemente el caso de uso.
- El objetivo es la funcionalidad que se pretende cubrir.
- Las precondiciones son los requisitos previos.
- Las postcondiciones son efectos que genera en el sistema.
- El escenario define el procedimiento para alcanzar la funcionalidad del sistema.
- El actor es un elemento externo que interactúa con el sistema, no está solo delimitado a personas. En este caso existen cuatro tipos:
 - El usuario gestor / administrador es el actor que posee los privilegios de acceso tanto a los archivos de configuración del back-end como a la base de datos del sistema. Esto le permite realizar acciones de creación, modificación o eliminación de contenidos y formularios en la plataforma.
 - El usuario de NRH (Nómina y Recursos Humanos) es un usuario que tiene acceso a todas las funcionalidades de MilenaBot y de la plataforma de contenidos salvo las reservadas para el “Usuario gestor”.
 - El usuario estándar es en realidad un subtipo del actor “Usuario de recursos humanos” porque no tiene acceso a la plataforma de contenidos y dependiendo de los permisos de su usuario de Milena, tiene restringidos contenidos y formularios.

- La plataforma de contenidos provee a la herramienta de todas las respuestas de los procesos y base de conocimiento. Además, almacena las frases de entrenamiento de Dialogflow.

Para una mejor comprensión, se ha organizado en cinco subsistemas:

- Subsistema de procesos y base de conocimiento
- Subsistema de formulario vacaciones
- Subsistema de formulario permisos retribuidos
- Subsistema de formulario nómina
- Subsistema de formulario certificado de retenciones

3.3.1.1 Casos de uso: subsistema de procesos y BC

La figura 44 resume todos los casos de uso del subsistema de procesos y base de conocimiento descritas en las tablas 18 a 24.

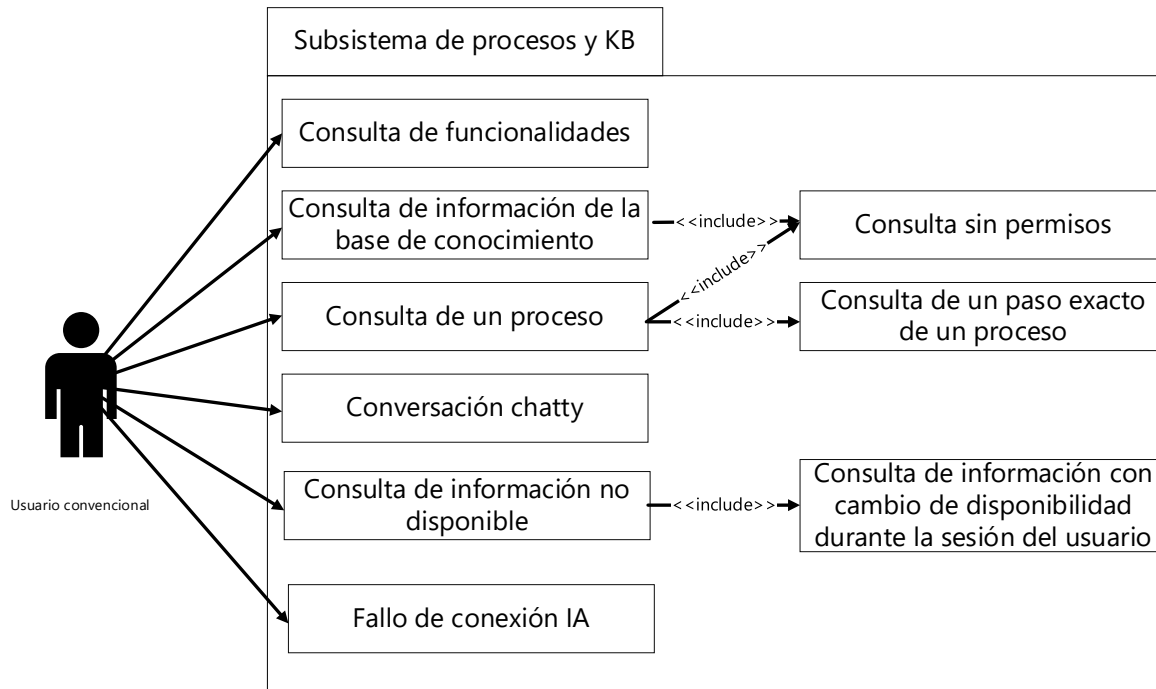


Figura 44 - Casos de uso del subsistema de procesos y base de conocimiento.

Nombre	Consulta de funcionalidades	Identificador	CU-01
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos ' <i>Generales</i> ' y desplegar la interfaz del asistente.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a las del intent de funcionalidades. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. El sistema accede a la base de datos y envía la respuesta formada por un mensaje, un elemento multimedia y las opciones. 5. Si el usuario selecciona uno de los botones, el asistente accede directamente a la base de datos y envía los respectivos recursos. 		
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF10, RF11 y RF32		

Tabla 18 - Caso de uso: Consulta de funcionalidades

Nombre	Consulta de información de la base de conocimiento	Identificador	CU-02
Actores	Usuario convencional		
Objetivo	Consultar información de la base de conocimiento		
Precondiciones	El usuario debe tener permisos de acceso a los recursos que está accediendo y desplegar la interfaz del asistente.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar o igual a la entrenada en un intent asociado a base de conocimiento. 		

	<ol style="list-style-type: none"> 3. El NLU identifica la expresión y la herramienta comprueba los permisos del recurso. 4. El sistema accede a la base de datos y envía la respuesta formada por un mensaje, un elemento multimedia y los temas relacionados con la consulta. 5. Si el usuario selecciona uno de los temas relacionados, el asistente accede directamente a la base de datos y envía la respuesta. En el caso de que introduzca cualquier otra expresión, esta será procesada de nuevo.
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF10 y RF32

Tabla 19 - Caso de uso: Consulta de información de la base de conocimiento

Nombre	Consulta de un proceso	Identificador	CU-03
Actores	Usuario convencional		
Objetivo	Consultar un proceso completo		
Precondiciones	El usuario debe tener permisos de acceso a los recursos que está accediendo y desplegar la interfaz del asistente.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar o igual a la entrenada en un intent. 3. El NLU identifica la expresión y la herramienta comprueba los permisos del recurso. 4. El sistema accede a la base de datos y envía la respuesta formada por un mensaje, un elemento multimedia y las opciones. 5. Si el usuario selecciona uno de los botones de una interfaz de proceso o base de conocimiento, el asistente accede directamente a la base de datos y envía los respectivos recursos. En el caso de que seleccione una opción con formulario, accede directamente al primer paso. 		

	6. En el caso de que introduzca cualquier otra expresión, esta será procesada de nuevo.
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF09, RF10 y RF32

Tabla 20 - Caso de uso: Consulta de un proceso

Nombre	Consulta de un paso exacto de un proceso	Identificador	CU-04
Actores	Usuario convencional, plataforma de contenidos		
Objetivo	Consultar un proceso completo		
Precondiciones	El usuario debe tener permisos de acceso a los recursos que está accediendo y que el usuario de NRH configure frases de entrenamiento para ese paso en concreto.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar o igual a la entrenada para un paso exacto de un proceso. 3. El NLU identifica la expresión y la herramienta comprueba los permisos del recurso. 4. El sistema accede a la base de datos y envía la respuesta formada por un mensaje, un elemento multimedia y las opciones. 5. Si el usuario selecciona uno de los botones de una interfaz de proceso o base de conocimiento, el asistente accede directamente a la base de datos y envía los respectivos recursos. En el caso de que seleccione una opción con formulario, accede directamente al primer paso. 6. En el caso de que introduzca cualquier otra expresión, esta será procesada de nuevo. 		
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF09, RF10 y RF32		

Tabla 21 - Caso de uso: Consulta un paso exacto de un proceso

Nombre	Consulta de información no disponible	Identificador	CU-05
Actores	Usuario convencional		
Objetivo	El servicio de IA no realice matched sobre la expresión.		
Precondiciones	El usuario debe tener permisos de acceso a los recursos que está accediendo, desplegar la interfaz del asistente y el NLU no esté entrenado con frases similares a la pregunta.		
Postcondiciones	La expresión del usuario es almacenada en la plataforma.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión no identificable. 3. El NLU no asigna el mensaje a un intent. 4. El sistema envía la respuesta por defecto de no identificación del mensaje. 5. La herramienta almacena esta expresión en la base de datos. 		
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF10 y RF62		

Tabla 22 - Caso de uso: Consulta de información no disponible

Nombre	Consulta de información sin permisos	Identificador	CU-06
Actores	Usuario convencional		
Objetivo	La herramienta deniega el acceso al recurso solicitado.		
Precondiciones	El usuario no debe tener permisos de acceso a los recursos que está accediendo.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión interpretable por el NLU. 3. El NLU identifica la expresión y la herramienta comprueba los permisos del recurso. 4. El sistema envía la respuesta por defecto de no identificación del mensaje. 		
Correspondencia	RF01, RF02, RF05, RF06, RF07, RF08, RF10 y RF32		

Tabla 23 - Caso de uso: Consulta de información sin permisos

Nombre	Conversación chatty	Identificador	CU-07
Actores	Usuario convencional		
Objetivo	La herramienta responda adecuadamente a mensajes de saludo, identificación y ayuda.		
Precondiciones	El usuario debe tener permisos de acceso 'Generales'.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión interpretable por el NLU. 3. El NLU identifica la expresión y la herramienta, tras comprobar los permisos del recursos, responde al usuario. 		
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF10, RF12, RF13 y RF32		

Tabla 24 - Caso de uso: Conversación chatty

3.3.1.2 Casos de uso: subsistema de formulario vacaciones

La figura 45 resume todos los casos de uso del subsistema de solicitud de vacaciones descritas en las tablas 25 a 27.

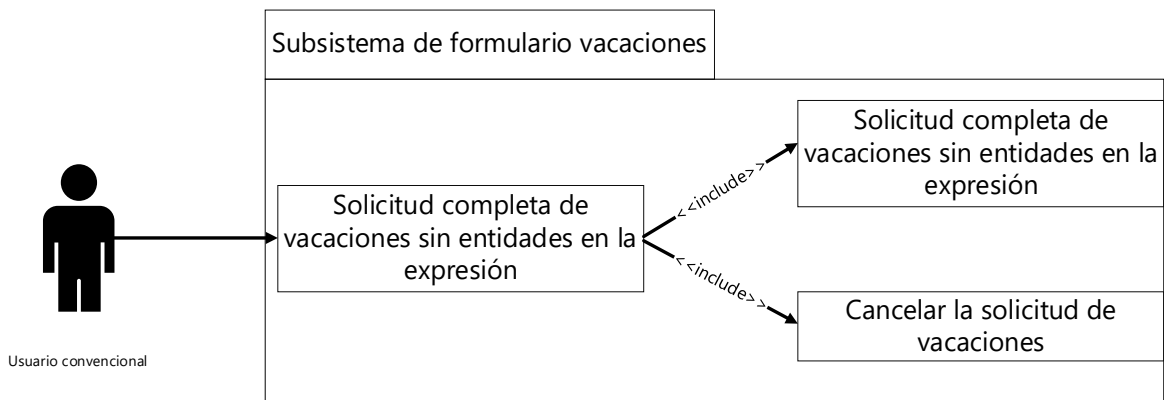


Figura 45 - Casos de uso del subsistema de formulario vacaciones

Nombre	Solicitud completa de vacaciones sin entidades en la expresión.	Identificador	CU-08
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de vacaciones. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. El sistema accede al servicio de Milena para obtener los días disponibles y el estado de las solicitudes y le envía la respuesta junto con una pregunta sobre si quiere continuar. 5. Si el usuario selecciona el botón de confirmación, se solicita la fecha de inicio. En caso contrario, retorna al estado inicial. 6. El usuario introduce la fecha inicial. 7. El sistema valida si la fecha no es igual o anterior a la actual, festivo o existe. En el caso de ser correcta solicita la fecha de retorno, sino la herramienta le envía un mensaje advirtiéndole del error y se la solicita de nuevo. 8. El usuario introduce la fecha de retorno. 9. El sistema valida si la fecha es posterior o igual a la inicial, festivo, existe y si el periodo de días laborables es inferior al enviado por Milena. En el caso de ser correcta solicita los comentarios, sino la herramienta le envía un mensaje advirtiéndole del error y se la solicita de nuevo. 10. El usuario introduce los comentarios. 11. El sistema almacena los comentarios y envía un resumen de los parámetros introducidos junto a una confirmación. 		

	<p>12. Si el usuario acepta, se procesa la solicitud, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones.</p> <p>13. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo que el periodo no exceda al disponible y sea posterior a la fecha actual.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF32, RF33 y RF34.

Tabla 25 - Caso de uso: Solicitud completa de vacaciones sin entidades en la expresión

Nombre	Solicitud completa de vacaciones con entidades en la expresión.	Identificador	CU-09
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de vacaciones con la fecha de inicio y/o de retorno. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. El sistema accede al servicio de Milena para obtener los días disponibles y el estado de las solicitudes y le envía la respuesta junto con una pregunta sobre si quiere continuar. En el caso de que el periodo de la expresión exceda del disponible o sea incorrecto, le envía un mensaje advirtiéndole de que se solicitarán las fechas de nuevo. 5. Si el usuario solo introdujo una fecha en la expresión, la herramienta la considera como fecha inicial y solicita la fecha de retorno. 		

	<ol style="list-style-type: none"> 6. El usuario, dependiendo de la expresión, no introduce ninguna fecha o solo la de retorno. Si se da el segundo caso, el sistema valida si la fecha es posterior o igual a la inicial, festivo, existe y si el periodo de días laborables es inferior al enviado por Milena. Si es correcta, solicita los comentarios, sino la herramienta le envía un mensaje advirtiéndole del error y se la solicita de nuevo. 7. El usuario introduce los comentarios. 8. El sistema almacena los comentarios y envía un resumen de los parámetros introducidos junto a una confirmación. 9. Si el usuario acepta, se procesa la solicitud, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones. 10. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo que el periodo no exceda al disponible y sea posterior a la fecha actual.
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF15, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF32, RF33, RF34 y RF35.

Tabla 26 - Caso de uso: Solicitud completa de vacaciones con entidades

Nombre	Cancelar la solicitud de vacaciones	Identificador	CU-10
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de vacaciones. 3. La herramienta comprueba los permisos de acceso a ese recurso. 		

	<p>4. El sistema accede al servicio de Milena para obtener los días disponibles y el estado de las solicitudes y le envía la respuesta junto con una pregunta sobre si quiere continuar.</p> <p>5. Si el usuario introduce en cualquiera de las interacciones una expresión similar a cancelar, retorna al estado inicial.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF32, RF33, RF34 y RF35.

Tabla 27 - Caso de uso: Cancelar solicitud de vacaciones.

3.3.1.3 Casos de uso: subsistema de formulario permisos retribuidos

La figura 46 resume todos los casos de uso del subsistema de solicitud de permisos retribuidos descritas en las tablas 28 a 31.

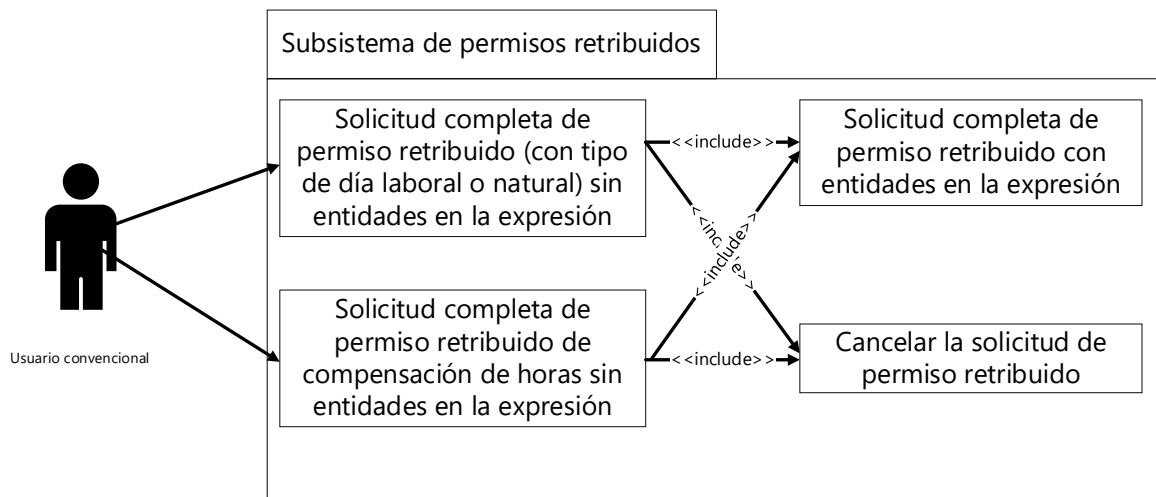


Figura 46 - Casos de uso del formulario de permisos retribuidos

Nombre	Solicitud completa de permiso retribuido (con tipo de día laboral o natural) sin entidades en la expresión.	Identificador	CU-11
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a permiso retribuido. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. El sistema presenta al usuario todos los permisos retribuidos disponibles. 5. El usuario selecciona el tipo de permiso a excepción de "Compensación de horas". 6. El sistema informa de los días laborables o naturales máximos disponibles y le consulta la fecha de inicio. 7. El usuario introduce la fecha de inicio. 8. El sistema valida si la fecha no es igual o anterior a la actual, festivo o no existe. En el caso de ser correcta solicita la fecha de retorno, sino la herramienta le envía un mensaje advirtiéndole del error y se la solicita de nuevo. Si el permiso es de un día, no se solicita fecha de retorno. 9. El usuario introduce la fecha de retorno. 10. El sistema valida si la fecha es posterior o igual a la inicial, festivo, existe y si el periodo de días laborables o naturales es inferior o igual al del servicio. En el caso de ser correcta solicita los comentarios, sino la herramienta le envía un mensaje advirtiéndole del error y se la solicita de nuevo. 11. El usuario introduce los comentarios. 		

	<p>12. El sistema almacena los comentarios y envía un resumen de los parámetros introducidos junto a una confirmación.</p> <p>13. Si el usuario acepta, se procesa la solicitud, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones.</p> <p>14. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo que el periodo no exceda al disponible y sea posterior a la fecha actual.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF17, RF18, RF19, RF20, RF24, RF25, RF26, RF32, RF33 y RF34.

Tabla 28 - Solicitud completa de permiso retribuido (con tipo de día laboral) sin entidades

Nombre	Solicitud completa de permiso retribuido de compensación de horas sin entidades en la expresión.	Identificador	CU-12
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a permiso retribuido. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. El sistema presenta al usuario todos los permisos retribuidos disponibles. 5. El usuario selecciona "Compensación de horas". 6. El sistema informa de que solo se dispone de un día laboral. 7. El usuario introduce la fecha. 8. El sistema valida si la fecha no es igual o anterior a la actual, festivo o no existe. En el caso de ser correcta se solicita la hora inicial. 		

	<p>9. El usuario introduce la hora inicial.</p> <p>10. El sistema valida la hora y solicita la hora de retorno.</p> <p>11. El usuario introduce la hora de retorno.</p> <p>12. La herramienta comprueba que el periodo no sobrepasa las 8 horas y 30 minutos y solicita los comentarios.</p> <p>13. El usuario introduce los comentarios.</p> <p>14. El sistema almacena los comentarios y envía un resumen de los parámetros introducidos junto a una confirmación.</p> <p>15. Si el usuario acepta, se procesa la solicitud, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones.</p> <p>16. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo que el periodo no exceda al disponible y sea posterior a la fecha actual.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF17, RF18, RF19, RF20, RF24, RF25, RF26, RF32, RF33 y RF34.

Tabla 29 - Solicitud completa de permiso retribuido de compensación de horas sin entidades en la expresión.

Nombre	Solicitud completa de permiso retribuido con entidades en la expresión.	Identificador	CU-13
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de permiso retribuido con el tipo y/o la fecha de inicio y de retorno. 3. La herramienta comprueba los permisos de acceso a ese recurso. 		

	<p>3.1 Si el usuario introduce el tipo de permiso, envía el número de días disponibles.</p> <p>3.2 Si el usuario introduce una sola fecha, el bot la valida y pregunta sobre la de retorno.</p> <p>3.3 Si introduce dos fechas, primero comprueba su validez. En el caso de que también introduzca el tipo de permiso 3.1, valida el periodo en caso contrario se realiza tras la elección del permiso con los botones.</p> <p>4. El usuario introduce los parámetros restantes en las sucesivas interacciones.</p> <p>5. El sistema almacena la información y envía un resumen de los parámetros introducidos junto a una confirmación.</p> <p>6. Si el usuario acepta, se procesa la solicitud, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones.</p> <p>7. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo que el periodo no exceda al disponible y sea posterior a la fecha actual.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF15, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF32, RF33, RF34, RF35 Y RF36.

Tabla 30 - Solicitud completa de permiso con entidades en la expresión.

Nombre	Cancelar la solicitud de permiso retribuido	Identificador	CU-14
Actores	Usuario convencional		
Objetivo	Consultar solicitud de permiso retribuido		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de permiso retribuido. 		

	<p>3. La herramienta comprueba los permisos de acceso a ese recurso.</p> <p>4. El sistema le informa de los días disponibles y le pregunta por la fecha actual.</p> <p>5. Si el usuario introduce en cualquiera de las interacciones una expresión similar a cancelar, retorna al estado inicial.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF32, RF33, RF34 y RF35.

Tabla 31 - Cancelar la solicitud de permiso retribuido

3.3.1.4 Casos de uso: subsistema de nómina

La figura 47 resume todos los casos de uso del subsistema de solicitud de nómina descritas en las tablas 32 a 34.

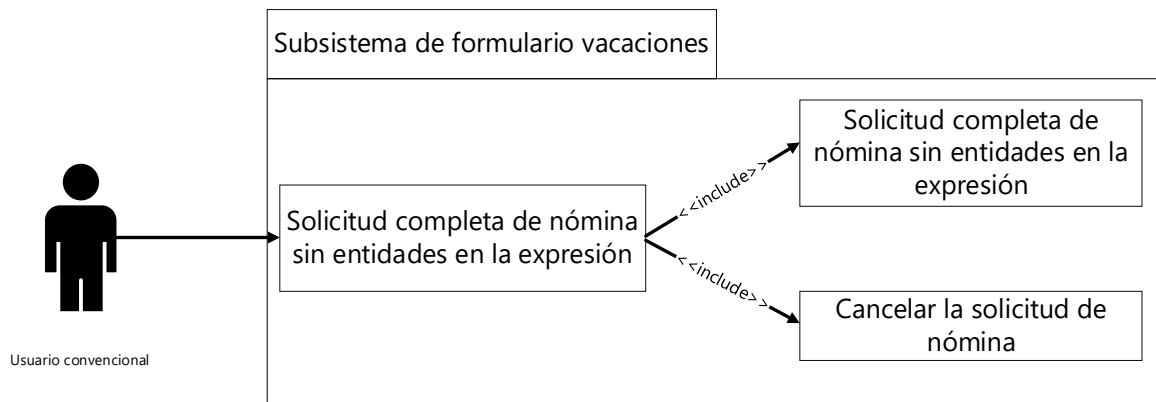


Figura 47 - Diagrama de casos de usos del subsistema de nómina

Nombre	Solicitud completa de nómina sin entidades en la expresión.	Identificador	CU-15
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Nómina'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida.		

	<ol style="list-style-type: none"> 2. El usuario introduce una expresión similar a la de solicitud de un intent de nómina. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. El sistema solicita el mes. 5. El usuario introduce el mes. 6. El bot comprueba la validez del mes. 7. El sistema solicita el año. 8. El usuario introduce el año. 9. El sistema comprueba que se dispone de la nómina solicitada y se genera un resumen. 10. Si el usuario acepta, se procesa la solicitud y se genera un enlace desde el que puede iniciar la descarga, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones. 11. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo el año y mes introducido.
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF17, RF18, RF19, RF20, RF27, RF28, RF32, RF33 y RF34.

Tabla 32 - Solicitud completa de nómina sin entidades en la expresión.

Nombre	Solicitud completa de nómina con entidades en la expresión.	Identificador	CU-16
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Nómina' .		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de nómina con entidades en la expresión. 		

	<p>3. La herramienta comprueba los permisos de acceso a ese recurso.</p> <p>1.1 Si la expresión contiene el mes, el chatbot no lo solicita.</p> <p>1.2 Si la expresión contiene el año, el chatbot no lo solicita.</p> <p>4. El sistema comprueba que se dispone de la nómina solicitada y se genera un resumen.</p> <p>5. Si el usuario acepta, se procesa la solicitud y se genera un enlace desde el que puede iniciar la descarga, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones.</p> <p>6. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo el año y mes introducido.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF17, RF18, RF19, RF20, RF27, RF28, RF32, RF33 y RF34.

Tabla 33 - Solicitud completa de nómina con entidades en la expresión.

Nombre	Cancelar la solicitud de nómina	Identificador	CU-17
Actores	Usuario convencional		
Objetivo	Consultar solicitud de permiso retribuido		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	-		
Escenario	<p>1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida.</p> <p>2. El usuario introduce una expresión similar a la de solicitud de un intent de nómina.</p> <p>3. La herramienta comprueba los permisos de acceso a ese recurso.</p> <p>4. Si el usuario introduce en cualquiera de las interacciones una expresión similar a cancelar, retorna al estado inicial.</p>		
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF32, RF33, RF34 y RF35.		

Tabla 34 - Cancelar la solicitud de nómina

3.3.1.5 Casos de uso: subsistema de certificado retenciones

La figura 48 resume todos los casos de uso del subsistema de solicitud de vacaciones descritas en las tablas 35 a 37.

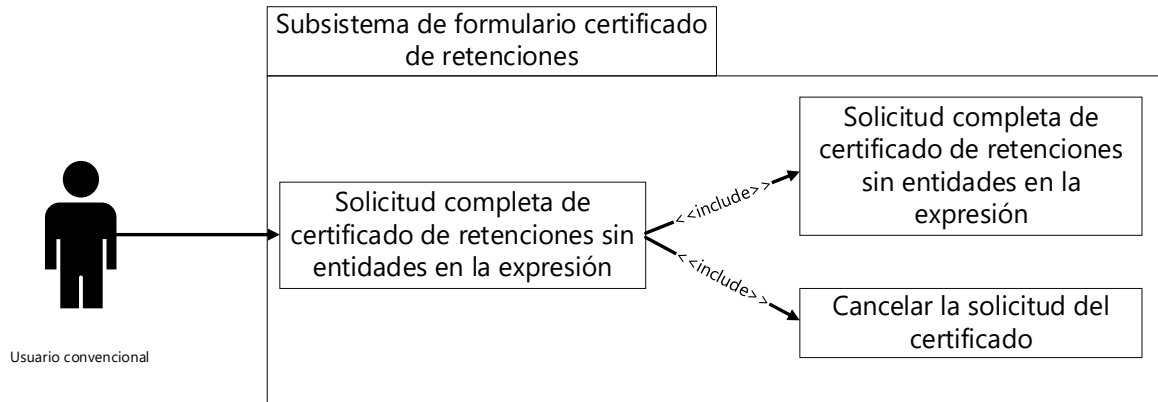


Figura 48 - Diagrama de casos de uso del certificado de retenciones

Nombre	Solicitud completa de certificado de retenciones sin entidades en la expresión.	Identificador	CU-18
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Nómina'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de certificado de retenciones. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. El sistema solicita el año. 5. El usuario introduce el año. 6. El sistema comprueba que se dispone del certificado solicitado y se genera un resumen. 		

	<p>7. Si el usuario acepta, se procesa la solicitud y se genera un enlace desde el que puede iniciar la descarga, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones.</p> <p>8. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo el año introducido.</p>
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF17, RF18, RF19, RF20, RF29, RF30, RF32, RF33 y RF34.

Tabla 35 - Solicitud completa de certificado de retenciones sin entidades.

Nombre	Solicitud completa de certificado de retenciones con entidades en la expresión.	Identificador	CU-19
Actores	Usuario convencional		
Objetivo	Consultar las funcionalidades de la herramienta		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Nómina'.		
Postcondiciones	La plataforma registra la solicitud.		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de certificado de retenciones con entidades en la expresión. 3. La herramienta comprueba los permisos de acceso a ese recurso. Si la expresión contiene el año, el chatbot no lo solicita. 2. El sistema comprueba que se dispone del certificado solicitado y se genera un resumen. 3. Si el usuario acepta, se procesa la solicitud y se genera un enlace desde el que puede iniciar la descarga, sino la herramienta le pregunta si desea modificar algún parámetro o salir del formulario mediante botones. 4. Si el usuario opta por modificar algún parámetro, la herramienta valida de nuevo el año introducido. 		

Correspondencia	RF01, RF02, RF05, RF07, RF08, RF17, RF18, RF19, RF20, RF29, RF30, RF32, RF33 y RF34.
-----------------	--

Tabla 36 - Solicitud completa de certificado de retenciones con entidades

Nombre	Cancelar la solicitud de certificado de retenciones.	Identificador	CU-20
Actores	Usuario convencional		
Objetivo	Consultar solicitud de permiso retribuido		
Precondiciones	El usuario debe tener permisos de acceso al módulo 'Vacaciones'.		
Postcondiciones	-		
Escenario	<ol style="list-style-type: none"> 1. El sistema identifica el idioma de la plataforma y envía un mensaje de bienvenida. 2. El usuario introduce una expresión similar a la de solicitud de un intent de certificado. 3. La herramienta comprueba los permisos de acceso a ese recurso. 4. Si el usuario introduce en cualquiera de las interacciones una expresión similar a cancelar, retorna al estado inicial. 		
Correspondencia	RF01, RF02, RF05, RF07, RF08, RF16, RF17, RF18, RF19, RF20, RF21, RF22, RF23, RF32, RF33, RF34 y RF35.		

Tabla 37 - Cancelar la solicitud de certificado de retenciones.

3.3.2 Diagramas de secuencia

Los diagramas de secuencia de los siguientes apartados son de los casos de uso más generales. No se realizan todos para no sobrecargar con información redundante.

3.3.2.1 Diagrama de secuencia: consulta funcionalidades

La figura 49 muestra el diagrama de secuencia para el caso de uso de consultar las funcionalidades del asistente. Este ejemplo sirve para todos los procesos y preguntas a la base de conocimiento. Se puede observar como el front-end interactúa con el back-end generando una actividad. Desde este evento, se accede a la función asociada al estado inicial, “*inital_state*”, que envía la expresión para que sea procesada por Dialogflow. Una vez que el servicio cognitivo determina el intent, se ejecuta “*set_process*”. Es en esta función desde la que se accede a la base de datos de Oracle para obtener los recursos y permisos. Finalmente, el front-end se encarga de mostrar una respuesta visible para el usuario.

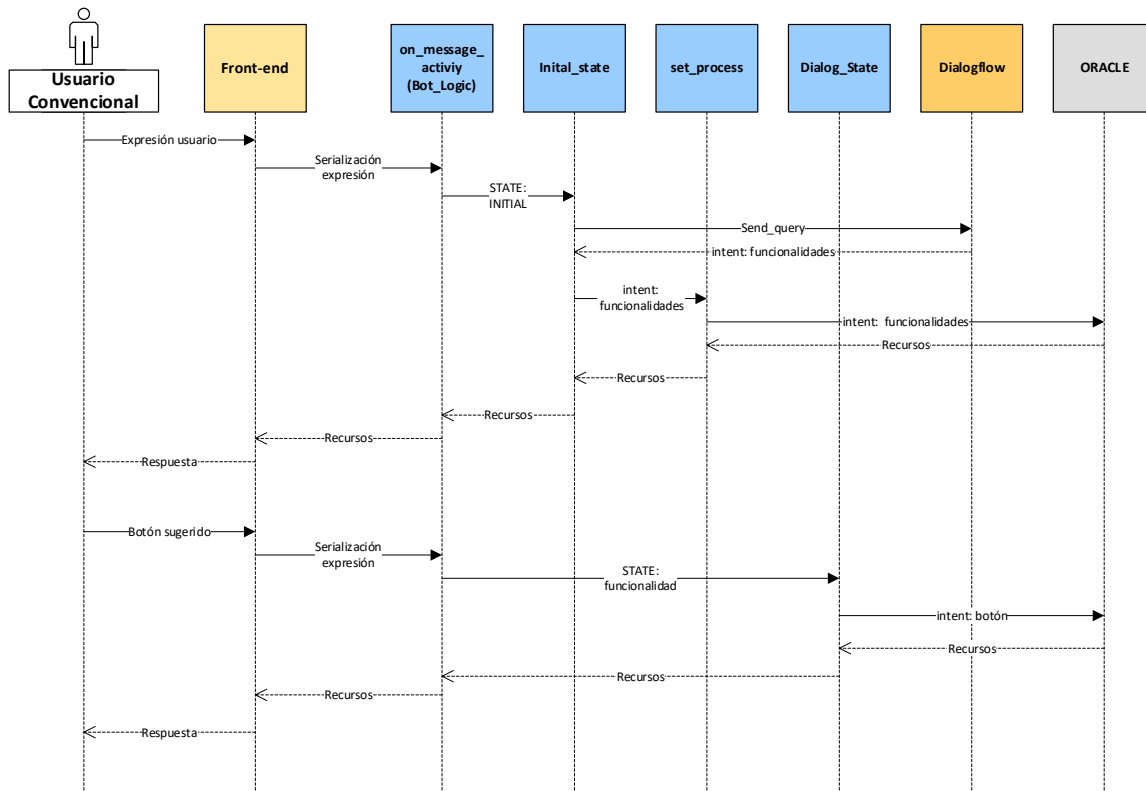


Figura 49 - Diagrama de secuencia: consulta funcionalidades

3.3.2.2 Diagrama de secuencia: formulario de permiso retribuido sin entidades

La figura 50 muestra el diagrama de secuencia para el caso de uso de solicitar un permiso retribuido sin entidades. Este ejemplo sirve para explicar la similar estructura que tienen estos formularios.

Desde el evento “*on_message_activity*”, se accede a la función asociada al estado inicial, “*intial_state*”, que envía la expresión para que sea procesada por Dialogflow. La IA determina que es un formulario sin entidades y se ejecuta “*set_form*”. Desde esta función, se accede al manejador que envía la pregunta y cambia al estado oportuno. Las zonas sombreadas indican bucles. El usuario solo puede avanzar si introduce el parámetro correctamente.

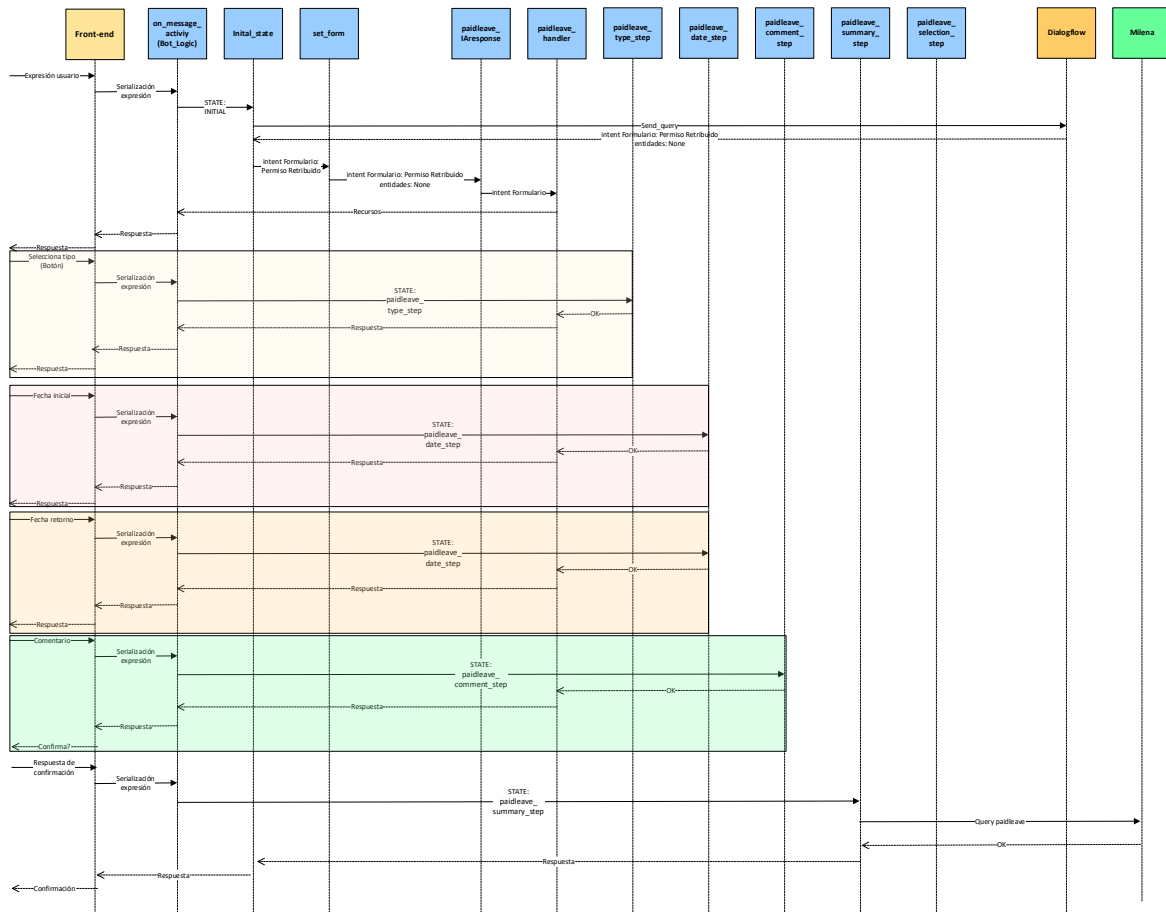


Figura 50 - Diagrama de secuencia de formulario de permiso retribuido

3.3.2.3 Diagrama de secuencia: formulario de nómina con entidades

La figura 51 muestra el diagrama de secuencia para el caso de uso de solicitar un permiso retribuido con entidades y modificación de los parámetros. Este ejemplo, al igual que el anterior, también sirve para detallar el similar comportamiento entre formularios.

Todos los mensajes son procesados por “*on_message_activity*” que accede a la función asociada al estado inicial. Esta primera parte es similar al 3.3.4.1. Sin embargo, la IA determina que es un formulario con entidades de modo que el manejador ya no se accede a todas las preguntas y estados. De hecho, al contener la expresión el año y mes, accede directamente al resumen de confirmación de solicitud.

Las zonas sombreadas indica un bucle, si el usuario al modificar un parámetro no introduce un dato correcto, el chatbot no le permite continuar.

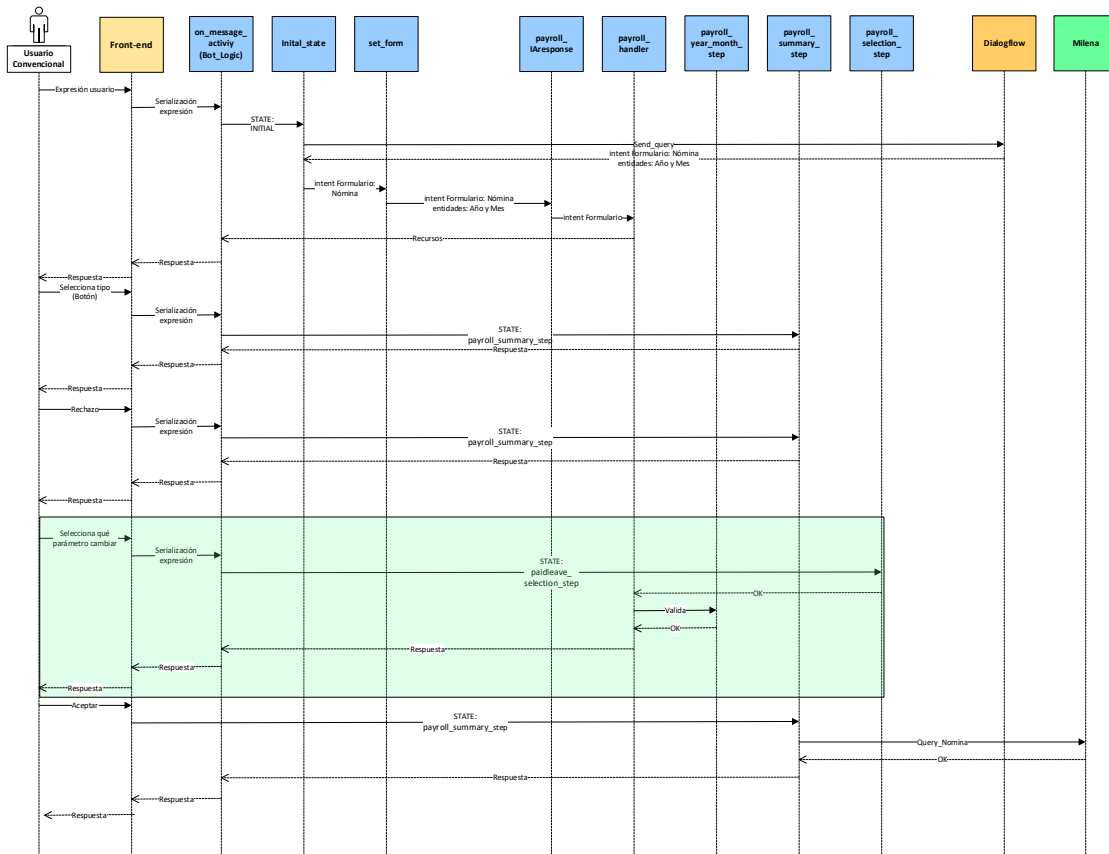


Figura 51 - Diagrama de secuencia de formulario de nómina

3.4 Descripción del back-end: Milena Server

El servidor es de tipo AIOHTTP (Asynchronous Hypertext Transfer Protocol Server) sobre el que Microsoft desarrolla una estructura para el procesamiento de mensajes, HTTP. Está basado en el módulo *asyncio* que proporciona herramientas para aplicaciones concurrentes con co-rutinas. *Asyncio* utiliza un enfoque de un solo hilo y proceso.

Sin embargo, el desarrollo tiene un mayor nivel de abstracción. En Bot Framework SDK, cada evento es una actividad del usuario o chatbot. Por ejemplo, al inicio de la conversación surgen dos actividades, ambas por actualización de la conversación de los dos agentes. Microsoft define el turno de conversación como el procesamiento de una actividad.

El *middleware* es similar a los back-ends de mensajería, incluye un conjunto lineal de componentes que se ejecutan secuencialmente. Los controladores definen la lógica del bot para las diferentes actividades que derivan de *ActivityHandler*. El controlador principal es “*on_turn*”, que funciona como manejador. Llama a dos controladores en función de la actividad:

- “*On_message_activity*” que se invoca una vez recibido un mensaje del usuario.
- “*On_event_activity*” es una función desde la que se definen los eventos.

La lógica principal está definida en `dialog_main/bot_logic.py`. *BotLogic* deriva de *ActivityHandler* que también deriva de la interfaz, *Bot*. En esta clase, se definen los controladores. Estos métodos se pueden sobrescribir aunque estén protegidos.

Debido a la particular estructura del back-end, los diagramas de estados de los siguientes apartados siguen el formato de la documentación de Microsoft del enlace [29]. No se considera una notación estandarizada. Las flechas indican que desde esa función o método se accede a otra.

3.4.1 Diagrama de estados de procesos

Al generarse la actividad de mensaje, se accede a una función dependiendo del estado del usuario. Si el usuario está en el inicial, se detecta el intent de la expresión mediante “*detect_intent*” de “*InitalDialog*” y a continuación, se establece el formulario o proceso / BC pertinente. Por último, si se encuentra en un proceso o pregunta de base de conocimiento, ejecuta “*process_dialog*” de “*ProcessDialogState*”. Todos los intents de este tipo son gestionados por esta función. Este diagrama se encuentra en la figura 52.

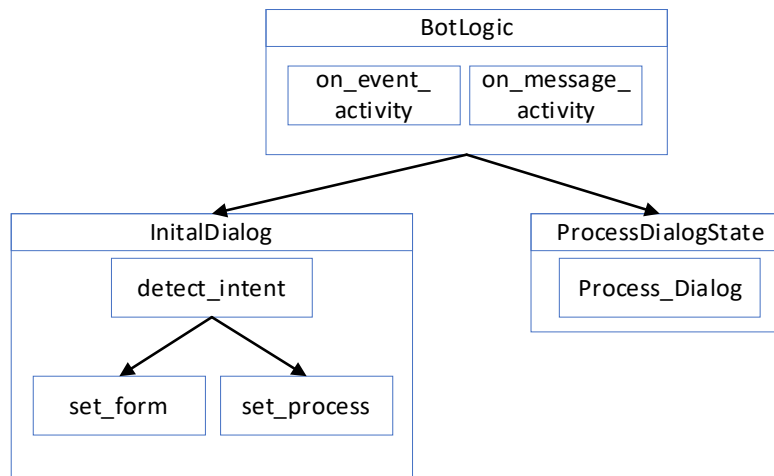


Figura 52 - Diagrama de clases de BotLogic

3.4.2 Diagrama de estados de los formularios

Desde “*set_form*”, se accede siempre a las funciones de tipo “*IAresponse*”. Son las encargadas de procesar las entidades del intent. Los manejadores, “*handler*”, envían los mensajes y establecen cada uno de los estados que conforman los formularios.

En el caso de que el usuario esté en cualquier de los estados de los formularios, la herramienta accede el diccionario de “*dict_mapping*” de “*StateFormHandler*” para ejecutar la función asociada. Todos los pasos que incluyen “*step*”, validan y almacenan la expresión del usuario.

Las figuras 53 a 56 contienen los diagramas de vacaciones, certificado de retenciones, nómina y permisos respectivamente.

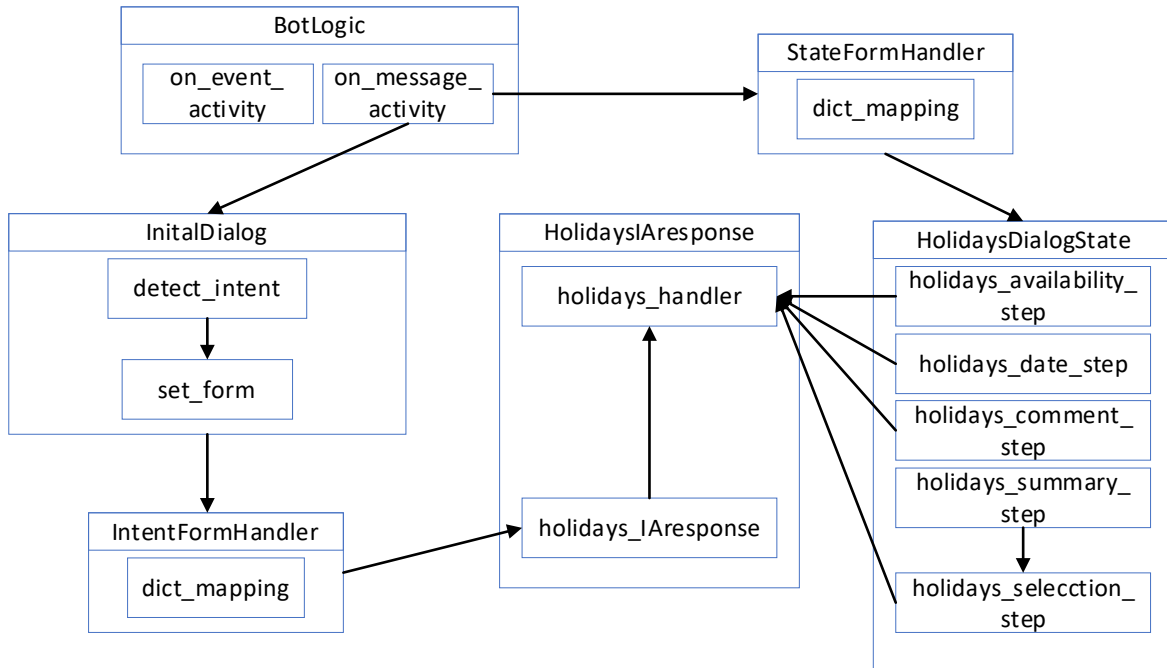


Figura 53 - Diagrama de estados del formulario de vacaciones

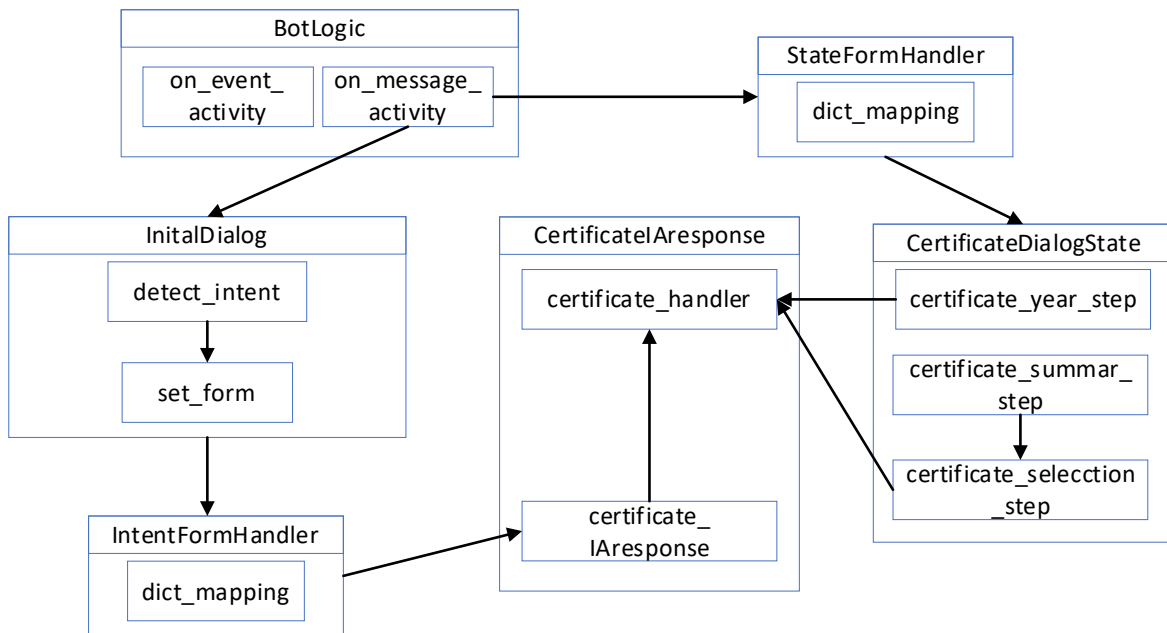


Figura 54 - Diagrama de estados del formulario de certificado

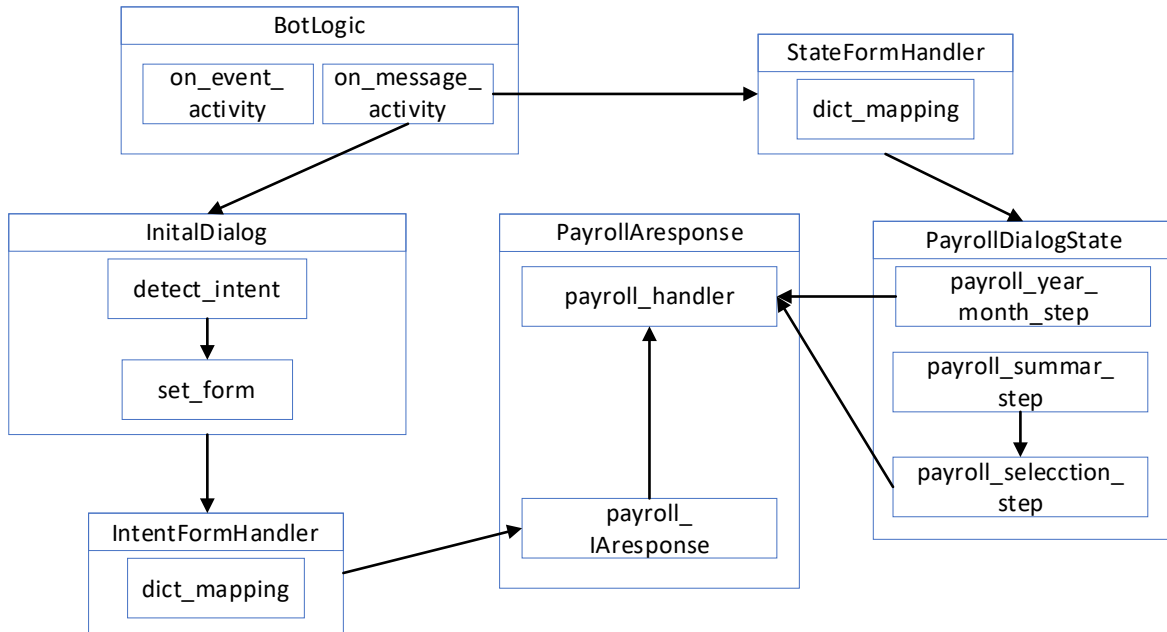


Figura 55 - Diagrama de estados del formulario de nómina

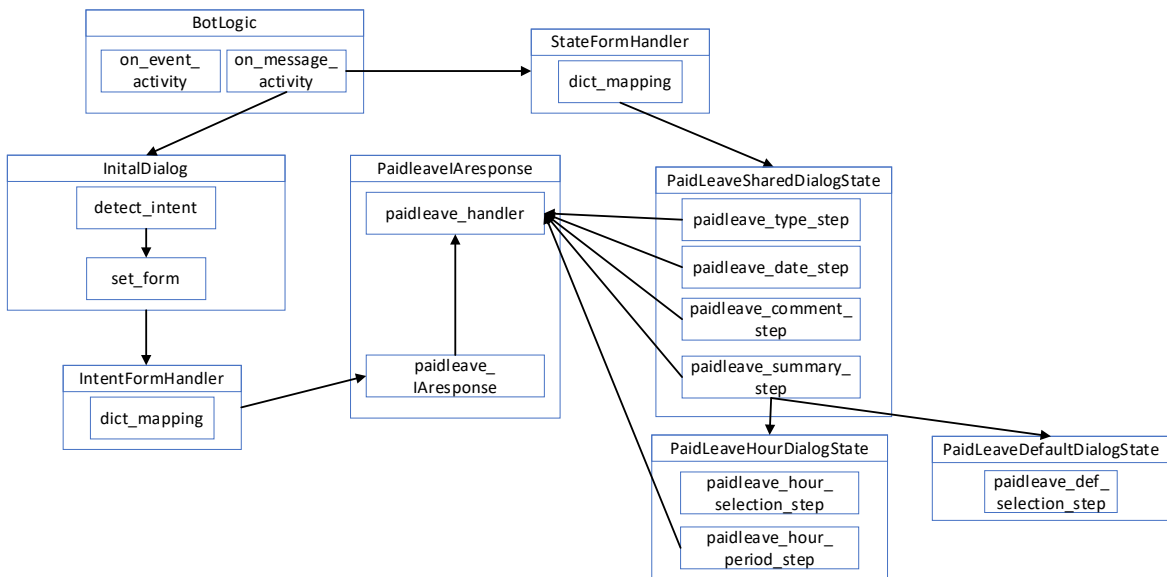


Figura 56 - Diagrama de estados del formulario de permisos retribuidos

3.4.3 Clases, métodos y funciones del back-end

La tabla 38 contiene todas las clases, métodos y funciones asíncronas que contiene actualmente el back-end.

dialog_main
class BotLogic(ActivityHandler)
<ul style="list-style-type: none"> - session_client : dialogflow_v2 - session_pool : cx_oracle - conversation_state - user_state
<ul style="list-style-type: none"> + __init__ (self, conversation_state, user_state, config) + on_turn (self, turn_context) + on_event_activity (self, turn_context) + on_message_activity (self, turn_context)
class InitialDialog
<ul style="list-style-type: none"> + detect_intent (turn_context, session_client, session_pool, conversation_data, user_profile) + check_user_permission (cursor, intent, user_profile) + set_process (turn_context, conversation_data, intent, cursor) + set_form (turn_context, session_client, conversation_data, user_profile, intent, entities) + send_query (turn_context, session_client, user_profile)
dialog_process
class ProcessDialogState
<ul style="list-style-type: none"> + process_dialog_state (turn_context, session_client, session_pool, conversation_data, user_profile)

class FailToConnectDialogState
+ process_error_connect (turn_context, conversation_data) + fail_select_option (turn_context, session_client, session_pool, conversation_data, user_profile)
dialog_form
class PayrollIAresponse
+ payroll_IAresponse (turn_context, session_client, conversation_data, user_profile, entities) + payroll_handler (turn_context, session_client, conversation_data, user_profile)
class PayrollDialogState
+ payroll_year_month_step (turn_context, session_client, conversation_data, user_profile) + payroll_summary_step (turn_context, session_client, conversation_data, user_profile) + payroll_selection_step (turn_context, session_client, conversation_data, user_profile)
class PaidLeaveIAresponse
+ paidleave_IAresponse (turn_context, session_client, conversation_data, user_profile, entities) + paidleave_handler (turn_context, session_client, conversation_data, user_profile) + paidleave_default_handler (turn_context, session_client, conversation_data, user_profile) + paidleave_hour_handler (turn_context, session_client, conversation_data, user_profile)

class PaidLeaveDefaultDialogState
+ paidleave_def_selection_step (turn_context, session_client, conversation_data, user_profile)
class PaidLeaveHourDialogState
+ paidleave_hour_period_step (turn_context, session_client, conversation_data, user_profile)
+ paidleave_hour_selection_step (turn_context, session_client, conversation_data, user_profile)
class PaidLeaveSharedDialogState
+ paidleave_type_step (turn_context, session_client, conversation_data, user_profile)
+ paidleave_date_step (turn_context, session_client, conversation_data, user_profile)
+ paidleave_comment_step (turn_context, session_client, conversation_data, user_profile)
+ paidleave_summary_step (turn_context, session_client, conversation_data, user_profile)
class CertificateIAresponse
+ certificate_IAresponse (turn_context, session_client, conversation_data, user_profile, entities)
+ certificate_handler (turn_context, session_client, conversation_data, user_profile)
class CertificateDialogState
+ certificate_year_step (turn_context, session_client, conversation_data, user_profile)
+ certificate_summary_step (turn_context, session_client, conversation_data, user_profile)

```
+ certificate_summary_step(turn_context, session_client, conversation_data,  
user_profile)  
+ certificate_selection_step(turn_context, session_client, conversation_data,  
user_profile)
```

```
class HolidaysIAresponse
```

```
+ holidays_IAresponse(turn_context, session_client, conversation_data,  
user_profile, entities)  
+ holidays_handler(turn_context, session_client, conversation_data,  
user_profile)
```

```
class HolidaysDialogState
```

```
+ holidays_availability_step (turn_context, session_client, conversation_data,  
user_profile)  
+ holidays_date_step (turn_context, session_client, conversation_data,  
user_profile)
```

```
class HolidaysDialogState
```

```
+ holidays_availability_step(turn_context, session_client, conversation_data,  
user_profile)  
+ holidays_date_step(turn_context, session_client, conversation_data,  
user_profile)  
+ holidays_comment_step(turn_context, session_client, conversation_data,  
user_profile)  
+ holidays_summary_step(turn_context, session_client, conversation_data,  
user_profile)  
+ holidays_selection_step(turn_context, session_client, conversation_data,  
user_profile)
```

class UserProfile

- name: str
- member_id: str
- language: str
- permission: str
- incident_description: str
- incident_priority: str
- holidays_start_date: str
- holidays_end_date: str
- holidays_comment: str
- holidays_num_requested: int
- holidays_num_available: int
- holidays_num_outstanding: int
- holidays_num_reject: int
- holidays_num_cancel: int
- paidleave_type: str
- paidleave_start_date: str
- paidleave_end_date: str
- paidleave_num_requested: int
- paidleave_start_hour: str
- paidleave_end_hour: str
- paidleave_hours_requested: str
- paidleave_comment: str
- payroll_year: str
- payroll_month: str
- certificate_year: str

class ConversationData
- state: str - state_option: str - evaluation_process: str - prompted_for_holidays: bool
handlers
class IntentFormHandler
- dict_mapping
class StateFormHandler
- dict_mapping
helpers
class HelperTimeFunctions
+ process_time(start_time, end_time) + check_time (time) + calculate_time_frame (start_time, end_time) + check_order_time(start_time, end_time) + change_time_format(raw_time)

class HelperDateFunctions
<ul style="list-style-type: none"> + process_entity_dates(raw_start_date, raw_end_date) + process_entity_period(start_date, end_date, type, num_available) + process_date(start_date, end_date, type, num_available) + process_input_date(raw_date) + change_format_date(raw_date) + check_date_in_calendar(raw_date) + check_holiday(date_valid) + check_order_date(start_date, end_date) + calculate_calendar_days(start_date, end_date) + calculate_working_days(start_date, end_date) + change_date_format(raw_date) + check_month(raw_month) + check_year(raw_year)
class HelperDialogFunctions
<ul style="list-style-type: none"> + attach_card(turn_context, introduction_message, image, response_list, conversation_data) + get_inline_attachment(image) + check_opt(turn_context, conversation_data) + check_break_form(turn_context, user_profile, conversation_data) + set_inital_state(turn_context, user_profile, conversation_data) + clean_form_values(user_profile, conversation_data)

Tabla 38 - Clases, métodos y funciones del back-end

3.5 Pruebas del back-end

En este apartado, las pruebas se centran exclusivamente en el back-end. La herramienta de “testing” utilizada es el emulador, su uso está explicado en el apartado de 3.2.2 Diseño del front-end. La estructura de las pruebas se detalla en la figura 57.

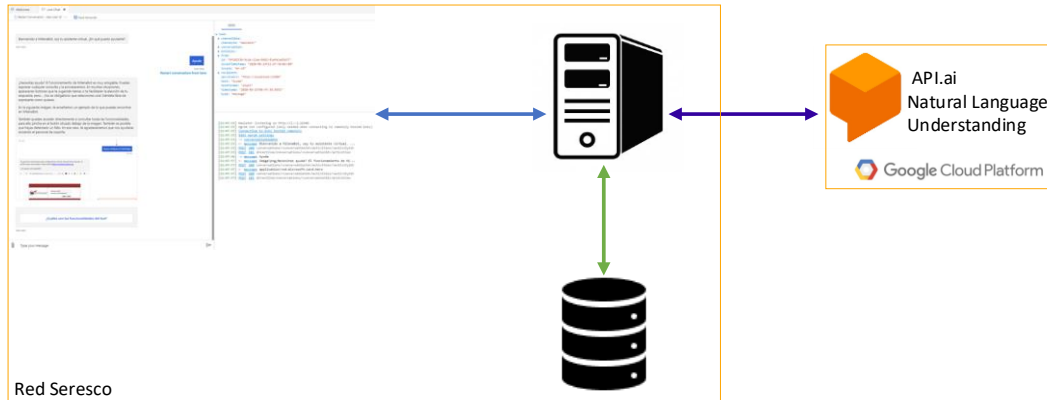


Figura 57 - Estructura de las pruebas

3.5.1 Verificación del back-end

El objetivo principal es verificar que el back-end desarrollado funciona correctamente en base a los requisitos establecidos. Cada prueba está definida por los siguientes elementos:

- El Identificador (ID) que puede ser de dos tipos.
 - PCNSI-VXX: Prueba de Caja Negra del chatbot. “V” significa que es un caso de una prueba en un uso correcto del sistema y “XX” es un número de secuencia incremental.
 - PCNSI-NXX: Prueba de Caja Negra del chatbot. “N” significa que es un caso de una prueba en un uso incorrecto del sistema y “XX” es un número de secuencia incremental.
- La prueba es el nombre la verificación que se va a realizar.
- La descripción es la definición detallada de la prueba.
- El resultado esperado es la descripción del resultado esperado.
- El resultado obtenido es la descripción del resultado final de la prueba.
- La validación indica de si el resultado de la prueba confirma el esperado.
- La correspondencia indica el caso de uso asociado a la prueba.

3.5.1.1 Casos de prueba: subsistema de procesos y BC

Los casos de prueba del subsistema de procesos y base de conocimiento comprenden de la tabla 39 a 48.

Prueba	Consulta del árbol de funcionalidades	Identificador	PCNSI-V01
Descripción	Esta prueba pretende verificar si la herramienta responde adecuadamente ante la consulta y el acceso al árbol de funcionalidades.		
Resultado esperado	El sistema responde con un mensaje y las tres opciones disponibles: “ <i>Base de conocimiento</i> ”, “ <i>Asistente Virtual</i> ” y “ <i>Acciones para el usuario</i> ”. Al seleccionar una, envía su definición.		
Resultado obtenido	Se obtiene la información sobre las funcionalidades de la herramienta.		
Validación	Correcta		
Correspondencia	CU-01		

Tabla 39 - Consulta de árbol de funcionalidades

Prueba	Consulta de información en base de conocimiento	Identificador	PCNSI-V02
Descripción	El objetivo es comprobar que el chatbot genere respuestas con elementos multimedia y botones sobre temas relacionados. Se realiza una pregunta sobre el salario mínimo interprofesional.		
Resultado esperado	El sistema responde con un mensaje textual sobre el estado actual del smi y dos botones con temas relacionados con la consulta.		
Resultado obtenido	El chatbot funciona adecuadamente como repositorio de preguntas.		
Validación	Correcta		
Correspondencia	CU-02		

Tabla 40 - Consulta de información en base de conocimiento

Prueba	Consulta de un proceso	Identificador	PCNSI-V03
Descripción	El objetivo es comprobar que el chatbot genere respuestas con elementos multimedia y botones que relacionen las interfaces entre sí. Se realiza una consulta sobre la configuración del teletrabajo.		
Resultado esperado	El sistema responde con interfaces manejadas mediante botones.		
Resultado obtenido	El chatbot funciona adecuadamente como repositorio de procesos.		
Validación	Correcta		
Correspondencia	CU-03		

Tabla 41 - Consulta de un proceso

Prueba	Consulta sobre un paso de un proceso sin acceso directo por expresión	Identificador	PCNSI-N01
Descripción	Esta prueba pretende verificar si la herramienta no permite acceder directamente a las definiciones de las distintas funcionalidades. Depende de la configuración realizada en la plataforma de contenidos.		
Resultado esperado	El sistema responde con el mensaje de no identificación por defecto: “¡No hemos conseguido entenderte! Es probable que no tengamos disponible la información a la que te refieres... No te preocupes, pronto la incluiremos.”.		
Resultado obtenido	No permite el acceso a ese recurso directamente y almacena la expresión.		
Validación	Correcta		
Correspondencia	CU-04		

Tabla 42 - Consulta sobre un paso de un proceso sin acceso directo por expresión

Prueba	Consulta sobre un paso de un proceso con acceso directo por expresión	Identificador	PCNSI-V04
Descripción	El objetivo es verificar si la herramienta permite acceder directamente a un paso de teletrabajo. Depende de la configuración realizada en la plataforma de contenidos.		
Resultado esperado	El sistema responde con el mensaje de ese paso de configuración, la imagen y las opciones.		
Resultado obtenido	Permite el acceso a ese recurso directamente.		
Validación	Correcta		
Correspondencia	CU-04		

Tabla 43 - Consulta sobre un paso de un proceso con acceso directo por expresión

Prueba	Consulta de información no disponible	Identificador	PCNSI-N02
Descripción	Se comprueba si la herramienta responde adecuadamente ante expresiones no registradas en la plataforma.		
Resultado esperado	El sistema responde con el mensaje de no identificación por defecto: “¡No hemos conseguido entenderte! Es probable que no tengamos disponible la información a la que te refieres... No te preocupes, pronto la incluiremos.” y almacena la expresión.		
Resultado obtenido	El sistema envía un mensaje por defecto y almacena la expresión.		
Validación	Correcta		
Correspondencia	-		

Tabla 44 - Consulta de información no disponible

Prueba	Consulta de información con cambio de disponibilidad durante la sesión	Identificador	PCNSI-V05
Descripción	Se comprueba que, durante la sesión del usuario, se realiza una actualización en los contenidos y en el agente de la IA, el chatbot responde adecuadamente.		
Resultado esperado	El bot inicialmente responde con el mensaje de no identificación por defecto: “ <i>¡No hemos conseguido entenderte! Es probable que no tengamos disponible la información a la que te refieres... No te preocupes, pronto la incluiremos.</i> ”. Tras la actualización de la plataforma de contenidos, MilenaBot responde adecuadamente.		
Resultado obtenido	La disponibilidad de los contenidos se actualiza durante la sesión del usuario.		
Validación	Correcta		
Correspondencia	CU-05		

Tabla 45 - Consulta de información con cambio de disponibilidad durante la sesión

Prueba	Consulta con fallo de conexión con la IA	Identificador	PCNSI-N03
Descripción	Esta prueba pretende verificar si la herramienta es capaz de responder ante fallos de conexión con el servicio cognitivo.		
Resultado esperado	El sistema responde con el mensaje de fallo: “ <i>Tenemos un pequeño problema... y no hemos podido acceder a nuestro servicio de IA... pero puedes reintentar reenviar tu expresión o consultar las funcionalidades de MilenaBot</i> ” junto a dos botones. El primero permite reenviar la expresión y el segundo acceder al árbol de funcionalidades.		
Resultado obtenido	El sistema ejecuta el proceso de fallo de conexión.		
Validación	Correcta		
Correspondencia	-		

Tabla 46 - Consulta con fallo de conexión con la IA

Prueba	Consulta de información sin permisos	Identificador	PCNSI-N04
Descripción	Esta prueba pretende verificar si la herramienta no permite acceder al usuario a recursos de los que no dispone de permisos. El usuario tiene acceso al módulo 'General' y se solicita información del de 'Vacaciones'.		
Resultado esperado	El sistema responde con el mensaje de no identificación por defecto: <i>"¡No hemos conseguido entenderte! Es probable que no tengamos disponible la información a la que te refieres... No te preocupes, pronto la incluiremos."</i>		
Resultado obtenido	No permite el acceso a ese recurso.		
Validación	Correcta		
Correspondencia	CU-06		

Tabla 47 - Consulta de información sin permisos

Prueba	Conversación chatty	Identificador	PCNSI-V06
Descripción	El objetivo es verificar si el bot ofrece respuestas adecuadas ante expresiones comunes del ser humano de saludo, pedir ayuda, identificación de MilenaBot y despedirse.		
Resultado esperado	El sistema responde adecuadamente ante estos cuatro tipos de mensajes.		
Resultado obtenido	MilenaBot responde según lo esperado en los cuatro casos.		
Validación	Correcta		
Correspondencia	CU-07		

Tabla 48 - Conversación chatty

3.5.1.2 Casos de prueba: subsistema de formulario vacaciones

Los casos de prueba del subsistema de formulario de vacaciones comprenden de la tabla 49 a 56.

3.5.1.2.1 Pruebas de vacaciones sin entidades en la expresión

Prueba	Solicitud vacacional sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral	Identificador	PCNSI-V07
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente las fechas introducidas en cada turno.		
Resultado esperado	El sistema responde aceptando ambos días y calculando el número de días hábiles solicitados.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-08		

Tabla 49 - Solicitud vacacional sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral

Prueba	Solicitud vacacional sin entidades en la expresión con periodo superior al máximo y fecha de inicio y retorno en día laboral	Identificador	PCNSI-N05
Descripción	El objetivo es comprobar que el sistema calcula adecuadamente el periodo y advierte de la superación de días disponibles.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: “El periodo introducido supera el límite disponible” y la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de retorno y la solicita de nuevo.		
Validación	Correcta		

Correspondencia	CU-10
-----------------	-------

Tabla 50 - Solicitud sin entidades en la expresión con periodo superior al máximo y fecha de inicio y retorno en día laboral

Prueba	Solicitud vacacional sin entidades en la expresión con fecha inicial no laboral	Identificador	PCNSI-N06
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de inicial: <i>“La fecha inicial es incorrecta o no es posterior a la actual”</i> y la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de inicial y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-08		

Tabla 51 - Solicitud vacacional sin entidades en la expresión con fecha inicial no laboral

Prueba	Solicitud vacacional sin entidades en la expresión con fecha de retorno no laboral	Identificador	PCNSI-N07
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: <i>“La fecha de retorno es incorrecta o no es posterior a la actual”</i> y la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de retorno y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-08		

Tabla 52 - Solicitud vacacional sin entidades en la expresión con fecha de retorno no laboral

3.5.1.2.2 Pruebas de vacaciones con entidades en la expresión

Prueba	Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión	Identificador	PCNSI-V08
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente las fechas introducidas y solo solicita los comentarios.		
Resultado esperado	El sistema no solicita las fechas de inicio – fin y calcula el número de días hábiles tras la primera confirmación.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-09		

Tabla 53 - Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión

Prueba	Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo superior al máximo en la expresión	Identificador	PCNSI-N08
Descripción	El objetivo es comprobar que el sistema calcula adecuadamente el periodo y advierte de la superación de días disponibles.		
Resultado esperado	El sistema responde tras la primera confirmación con “ <i>El periodo introducido no cumple con los requisitos. Te solicitaré de nuevo las fechas.</i> ”.		
Resultado obtenido	El sistema desestima ambas fechas y las solicita de nuevo		
Validación	Correcta		
Correspondencia	CU-09		

Tabla 54 - Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo superior al máximo en la expresión

Prueba	Solicitud vacacional con fecha de inicio (laboral) y retorno (no laboral) con periodo inferior o igual al máximo en la expresión	Identificador	PCNSI-N09
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: “ <i>La fecha de retorno es incorrecta o no es posterior a la actual</i> ” y tras la primera confirmación, la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de inicial y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-09		

Tabla 55 - Solicitud vacacional con fecha de inicio (laboral) y retorno (no laboral) con periodo inferior o igual al máximo en la expresión

Prueba	Solicitud vacacional con fecha de inicio (no laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión	Identificador	PCNSI-N10
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: “ <i>La fecha de inicio es incorrecta o no es posterior a la actual</i> ” y tras la primera confirmación, la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de retorno y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-09		

Tabla 56 - Solicitud vacacional con fecha de inicio (no laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión

3.5.1.3 Casos de prueba: subsistema de formulario de permisos retribuidos

Los casos de prueba del subsistema de permisos comprenden de la tabla 57 a 68.

3.5.1.3.1 Pruebas de permisos retribuidos sin entidades en la expresión

Prueba	Solicitud de permisos sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral	Identificador	PCNSI-V10
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente las fechas introducidas en cada turno.		
Resultado esperado	El sistema responde aceptando ambos días y calculando el número de días hábiles solicitados.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-11		

Tabla 57 - Solicitud de permisos sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral

Prueba	Solicitud de permisos sin entidades en la expresión con periodo superior al máximo y fecha de inicio y retorno en día laboral	Identificador	PCNSI-N11
Descripción	El objetivo es comprobar que el sistema calcula adecuadamente el periodo y advierte de la superación de días disponibles.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: <i>“El periodo introducido supera el límite disponible”</i> y la solicita de nuevo.		

Resultado obtenido	El sistema desestima la fecha de retorno y la solicita de nuevo.
Validación	Correcta
Correspondencia	CU-11

Tabla 58 - Solicitud de permisos sin entidades en la expresión con periodo superior al máximo y fecha de inicio y retorno en día laboral

Prueba	Solicitud de permisos sin entidades en la expresión con fecha inicial no laboral	Identificador	PCNSI-N12
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de inicial: “ <i>La fecha inicial es incorrecta o no es posterior a la actual</i> ” y la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de inicial y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-11		

Tabla 59 - Solicitud de permisos sin entidades en la expresión con fecha inicial no laboral

Prueba	Solicitud de permisos sin entidades en la expresión con fecha de retorno no laboral	Identificador	PCNSI-N13
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: “ <i>La fecha de retorno es incorrecta o no es posterior a la actual</i> ” y la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de retorno y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-11		

Tabla 60 - Solicitud de permisos sin entidades en la expresión con fecha de retorno no laboral

3.5.1.3.2 Pruebas de permiso retribuido por matrimonio sin entidades en la expresión

Prueba	Solicitud de permiso por matrimonio sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral	Identificador	PCNSI-V11
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente las fechas introducidas en cada turno.		
Resultado esperado	El sistema responde aceptando ambos días y calculando el número de días naturales solicitados.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-11		

Tabla 61 - Solicitud de permiso por matrimonio sin entidades en la expresión con periodo inferior al máximo y fecha de inicio y retorno en día laboral

3.5.1.3.3 Pruebas de compensación de horas sin entidades en la expresión

Prueba	Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y periodo temporal inferior o igual a la jornada laboral	Identificador	PCNSI-V12
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente los parámetros introducidos en cada turno.		
Resultado esperado	El sistema responde aceptando los parámetros y calculando el total de horas solicitadas.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		

Correspondencia	CU-12
-----------------	-------

Tabla 62 - Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y periodo temporal inferior o igual a la jornada laboral

Prueba	Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y periodo temporal superior a la jornada laboral	Identificador	PCNSI-N14
Descripción	El objetivo es comprobar que el sistema calcula adecuadamente el periodo y advierte de la superación de horas máximas.		
Resultado esperado	El sistema responde solicitando de nuevo la hora de retorno con el siguiente mensaje adjunto: <i>“El periodo es superior al periodo máximo laborable: 8 horas y 30 minutos”</i> .		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-12		

Tabla 63 - Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y periodo temporal superior a la jornada laboral

Prueba	Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y hora de retorno anterior a la hora de inicio.	Identificador	PCNSI-N15
Descripción	El objetivo es comprobar que el advierte de la inconsistencias de las horas.		
Resultado esperado	El sistema responde solicitando de nuevo las horas con el siguiente mensaje adjunto: <i>“La hora de retorno es anterior a la inicial, te solicitaré estos dos datos de nuevo”</i> .		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-12		

Tabla 64 - Solicitud de compensación de horas sin entidades en la expresión con fecha hábil y fecha de retorno anterior a la fecha de inicio

3.5.1.3.4 Pruebas de permisos con entidades en la expresión

Prueba	Solicitud con tipo de permiso, fecha de inicio (laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión	Identificador	PCNSI-V13
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente las fechas introducidas y solo solicita los comentarios.		
Resultado esperado	El sistema no solicita las fechas de inicio – fin y calcula el número de días hábiles tras la primera confirmación.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-13		

Tabla 65 - Solicitud con tipo de permiso, fecha de inicio (laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión

Prueba	Solicitud sin tipo de permiso y fechas en la expresión	Identificador	PCNSI-N16
Descripción	El objetivo es comprobar que el sistema al no tener un tipo de permiso retribuido establecido no almacena las fechas introducidas y advierte de que volverán a ser solicitadas.		
Resultado esperado	El sistema responde con: <i>“Has introducido el periodo sin haber escogido el tipo de permiso retribuido, se solicitará de nuevo.”</i>		
Resultado obtenido	El sistema desestima ambas fechas y las solicitará de nuevo		
Validación	Correcta		
Correspondencia	CU-13		

Tabla 66 - Solicitud vacacional con fecha de inicio (laboral) y retorno (laboral) con periodo superior al máximo en la expresión

Prueba	Solicitud con tipo de permiso con fecha de inicio (no laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión	Identificador	PCNSI-N17
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: <i>“La fecha inicial es incorrecta o no es posterior a la actual”</i> y tras la primera confirmación, la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de inicial y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-13		

Tabla 67 - Solicitud con tipo de permiso con fecha de inicio (no laboral) y retorno (laboral) con periodo inferior o igual al máximo en la expresión

Prueba	Solicitud con tipo de permiso con fecha de inicio (laboral) y retorno (no laboral) con periodo inferior o igual al máximo en la expresión	Identificador	PCNSI-N18
Descripción	El objetivo es comprobar que el sistema advierte de la solicitud en día no hábil.		
Resultado esperado	El sistema responde tras la introducción de la fecha de retorno: <i>“La fecha de retorno es incorrecta o no es posterior a la actual”</i> y tras la primera confirmación, la solicita de nuevo.		
Resultado obtenido	El sistema desestima la fecha de retorno y la solicita de nuevo.		
Validación	Correcta		
Correspondencia	CU-13		

Tabla 68 - Solicitud con tipo de permiso con fecha de inicio (laboral) y retorno (no laboral) con periodo inferior o igual al máximo en la expresión

3.5.1.4 Casos de prueba: subsistema de formulario de nómina

Los casos de prueba del subsistema de nómina s comprenden de la tabla 69 a 70.

3.5.1.4.1 Pruebas de nómina sin entidades en la expresión

Prueba	Solicitud de nómina sin entidades en la expresión con mes y año válido	Identificador	PCNSI-V14
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente el mes y año introducidos en cada turno de conversación.		
Resultado esperado	El sistema responde generando el enlace de descarga de la nómina del mes y año solicitado.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-15		

Tabla 69 - Solicitud de nómina sin entidades en la expresión con mes y año validos

3.5.1.4.2 Pruebas de nómina con entidades en la expresión

Prueba	Solicitud de nómina con entidad de mes y año en la expresión del usuario	Identificador	PCNSI-V15
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente el mes y año introducidos durante la expresión del usuario.		
Resultado esperado	El sistema responde generando el enlace de descarga de la nómina del mes y año solicitado.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-16		

Tabla 70 - Solicitud de nómina con entidad de mes y año en la expresión del usuario

3.5.1.5 Casos de prueba: subsistema de formulario de certificado de retenciones

Los casos de prueba del subsistema de certificado de retenciones comprenden de la tabla 71 a 72.

3.5.1.5.1 Pruebas de certificado sin entidades en la expresión

Prueba	Solicitud de certificado sin entidad en la expresión con año válido	Identificador	PCNSI-V16
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente el año introducido en cada turno de conversación.		
Resultado esperado	El sistema responde generando el enlace de descarga del certificado en el año solicitado.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-18		

Tabla 71 - Solicitud de certificado sin entidad en la expresión con año válido

3.5.1.5.2 Pruebas de certificado con entidades en la expresión

Prueba	Solicitud de nómina con entidad de año en la expresión del usuario	Identificador	PCNSI-V17
Descripción	El objetivo es comprobar que el sistema procesa adecuadamente el año introducido en el turno de conversación.		
Resultado esperado	El sistema responde generando el enlace de descarga del certificado en el año solicitado.		
Resultado obtenido	El sistema procesa adecuadamente la petición.		
Validación	Correcta		
Correspondencia	CU-19		

Tabla 72 - Solicitud de nómina con entidad de año en la expresión del usuario

3.5.1.6 Casos de prueba: cancelación de formularios

La tabla 73 recoge todas las pruebas de cancelaciones de los formularios: vacaciones, permisos retribuidos, nómina y certificado de retenciones.

Prueba	Cancelar formularios	Identificador	PCNSI-V18
Descripción	El objetivo es comprobar que es posible cancelar el formulario en cualquier turno.		
Resultado esperado	El sistema responde con: “¿En qué más puedo ayudarte?”		
Resultado obtenido	El sistema regresa al estado inicial según lo esperado.		
Validación	Correcta		
Correspondencia	CU-10, CU-14, CU-17, y CU-20		

Tabla 73 - Cancelar formulario

4. Manual de usuario

En este apartado se explica cómo el usuario debe utilizar el chatbot. Pese a que es una herramienta realmente intuitiva, se aconseja realizar una lectura detallada para conocer y comprender todas sus características.

Para acceder a la herramienta, el usuario debe desplegar la interfaz del chatbot dentro de la plataforma Milena según se muestra en la figura 58. Una vez realizada esta acción, el chatbot le envía un mensaje de bienvenida. A partir de ese preciso instante, el usuario tiene total libertad para expresarse como desee.

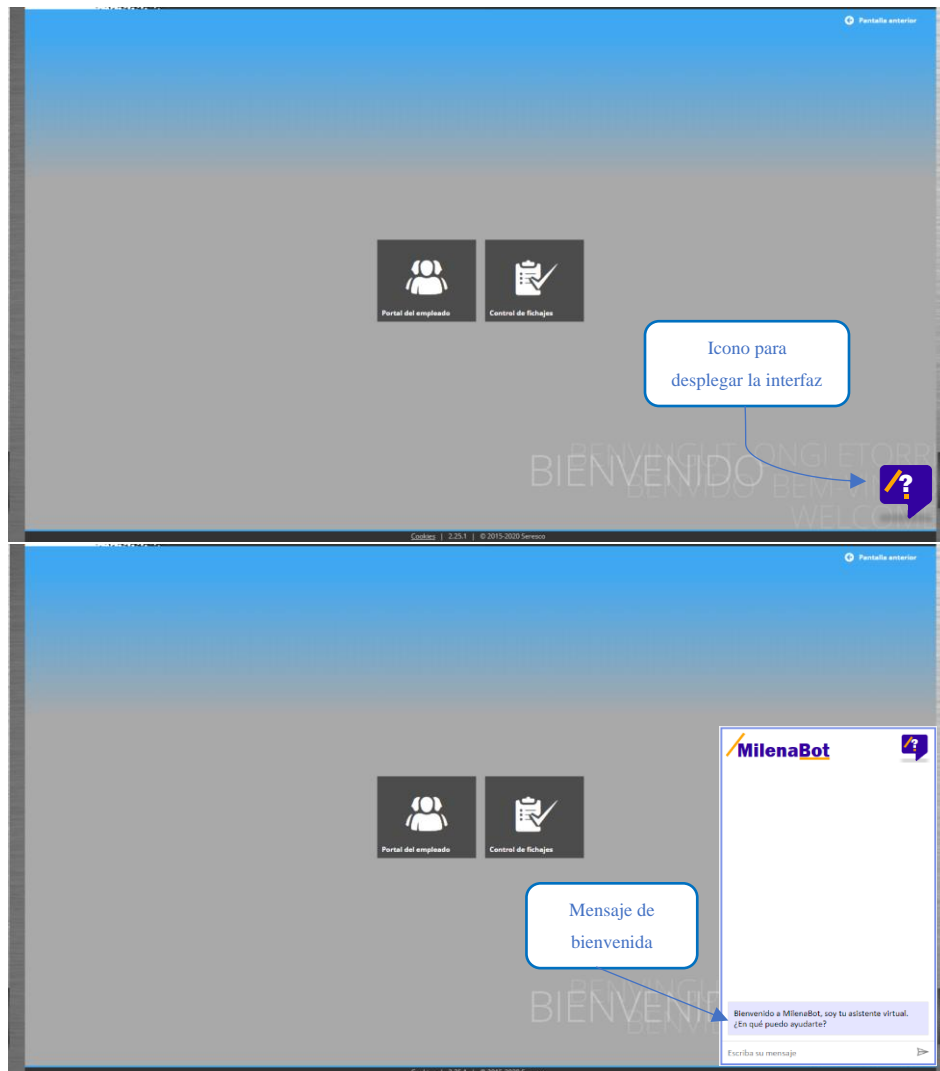


Figura 58 - Manual de usuario: acceso a la interfaz

4.1 Manual de procesos y base de conocimiento

Los mensajes del chatbot se identifican con el color azul mientras que los del usuario con un color grisáceo. Además, incluye la hora a la que se ha introducido la expresión. El usuario debe utilizar el espacio que se encuentra en la parte inferior del chat para redactar el mensaje. Si desea enviarlo, debe presionar la tecla “intro” o pinchar sobre el icono de flecha situado en la parte lateral derecha.

Los respuestas de MilenaBot pueden contener tres tipos distintos de elementos. Esto puede visualizarse en las figuras 59 y 60.

- Los mensajes textuales siempre están incluidos en la respuesta.
- Los elementos multimedia son principalmente imágenes. No todas las respuestas las incluyen. Puede ser descargadas por el usuario.
- Los botones de los procesos y base de conocimiento no obligan a que el usuario seleccione uno de ellos. Sirven para avanzar entre interfaces, acceder a otras preguntas relacionadas o formularios.

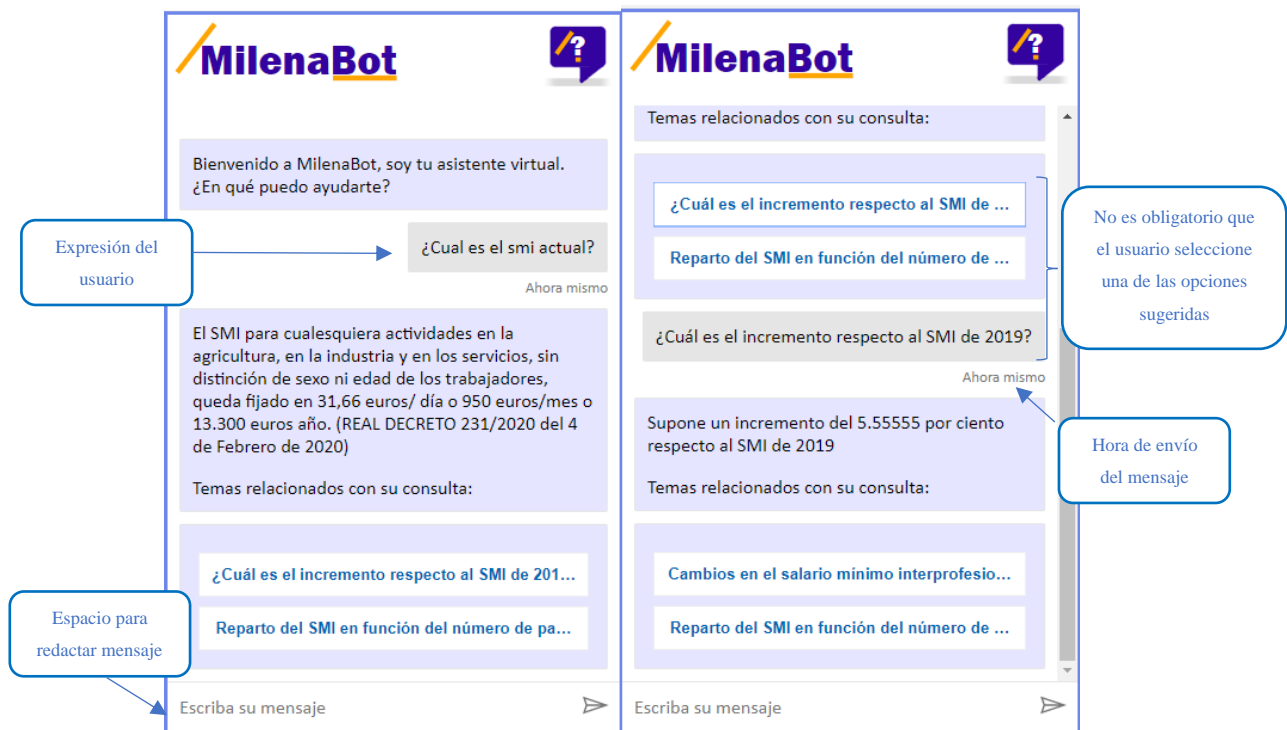


Figura 59 - Manual de usuario: primer ejemplo de proceso



Figura 60 - Manual de usuario: segundo ejemplo de proceso

4.2 Manual de formularios

MilenaBot identifica entidades en la expresión, que le permiten reducir el número de pasos del formulario. El usuario tiene total libertad para expresarse como desee. Si el parámetro introducido no cumple con la validación, vuelve a ser solicitado.

Los formularios incluyen un resumen previo a la confirmación con los parámetros introducidos. Desde esta interfaz, se puede acceder a su modificación. La figura 61 incluye un ejemplo para el formulario de solicitud de nómina.



Figura 61 - Manual de usuario: ejemplo de formulario

Si el usuario desea salir, puede optar por dos opciones: introducir “*Cancelar*” u optar por no confirmar la solicitud y a continuación, seleccionar el botón de “*Salir*”. Esto es lo que sucede en el ejemplo de la figura 62.



Figura 62 - Manual de usuario: cancelar formulario

Finalmente, es posible que el chatbot no disponga de la información que se está solicitando. De modo que MilenaBot, contesta con el siguiente mensaje de la figura 63.

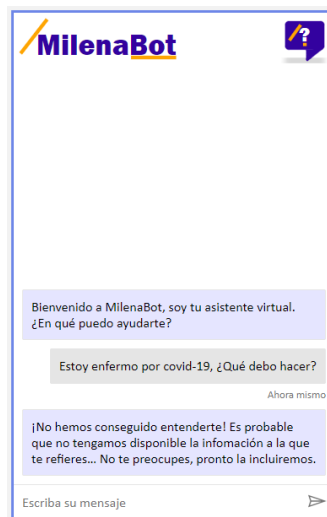


Figura 63 - Manual de usuario: solicitud de información no disponible.

5. Manual de configuración

El objetivo de este apartado es ofrecer una guía para el desarrollador para la administración, mantenimiento de la herramienta al completo.

5.1 Google Cloud Platform: administración de agentes

Este apartado describe el proceso para añadir un nuevo agente al proyecto de Dialogflow y la configuración de los mecanismos de seguridad pertinentes. El desarrollador debe acceder a la plataforma de Dialogflow del enlace [30]. La figura 64 muestra los pasos de configuración. En primer lugar, debe crear un nuevo agente, luego asignarle un nombre identificativo y el idioma principal.

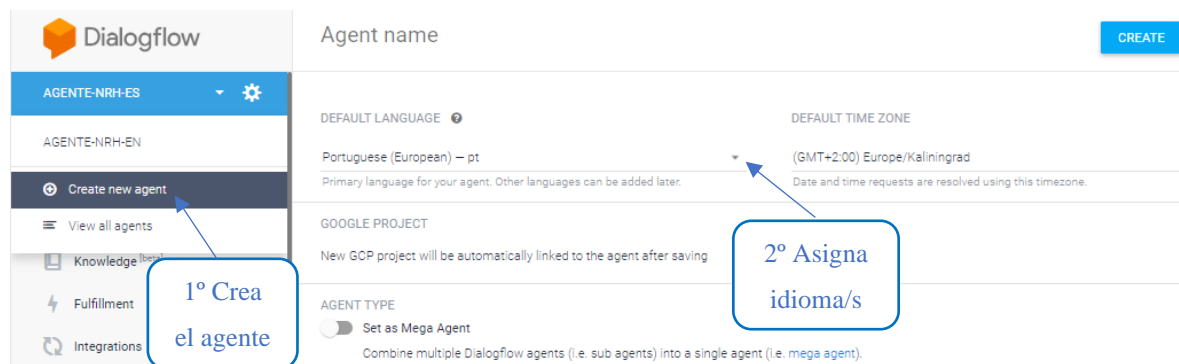


Figura 64 - Primer paso de la configuración de los agentes

Una vez creado el agente dentro del proyecto, se debe eliminar todos los intents por defecto menos el de “*Fallback*”. Toda la información se introduce mediante la plataforma de contenidos.

El siguiente paso es acceder a Google Cloud Platform del enlace [31]. En esta plataforma, también se genera un agente para configurar la API de acceso. Posteriormente, se debe seleccionar la opción de: “*Gestión de identidades y accesos*”. A continuación, se debe añadir el miembro que tiene una clave asociada. Esto es común para todos los agentes y permite establecer los mecanismos de autenticación para la creación, eliminación, modificación de los intents. Este procedimiento se muestra en la figura 65.

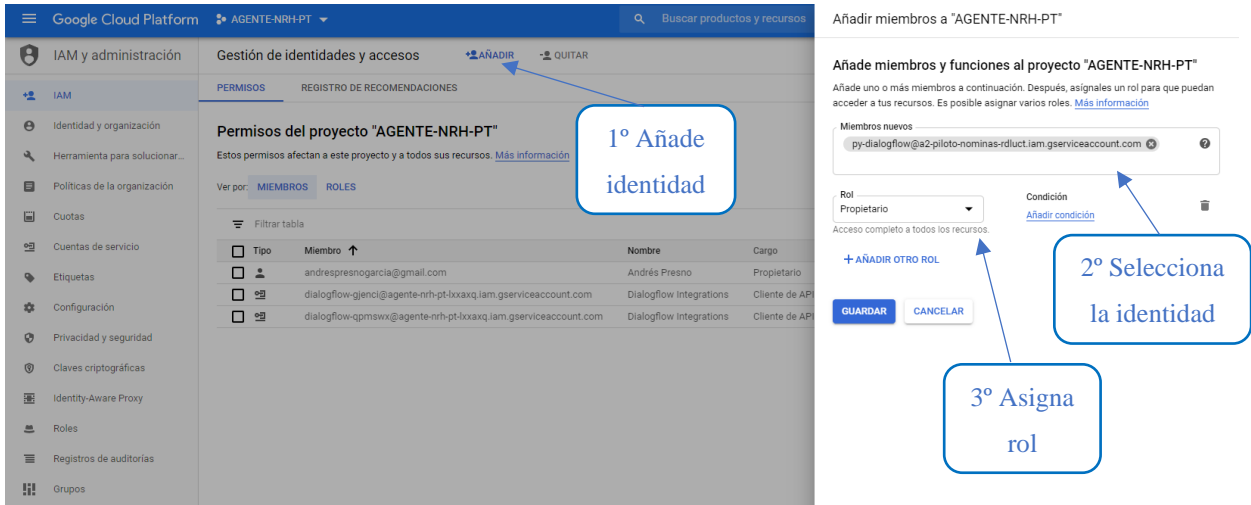


Figura 65 - Segundo paso de la configuración de los agentes

El último paso es añadir los parámetros de identificación al servidor. El cloud de Google genera automáticamente un ID. Este identificador, que es único es el parámetro que con el que se accede desde el back-end al agente. La figura 66 muestra un panel de control configurable con información general de la API asociada a este proyecto.

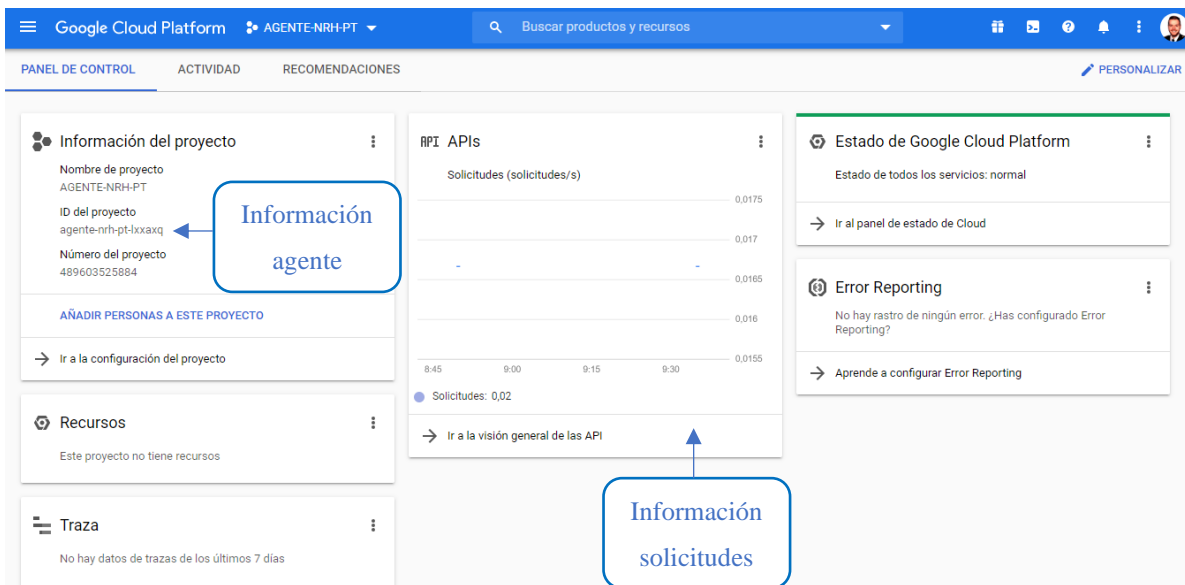


Figura 66 - Panel de control de Google Cloud Platform

5.2 Cliente ligero - cx_Oracle: cifrado de tráfico de red

Cx_Oracle es un módulo de Python que permite el acceso a la base de datos Oracle. Cumple con las especificaciones del estándar API 2.0 de bases de datos. Es válido para las versiones de Python 2.7 y de la 3.5 en adelante. El cliente 7.3 es interoperable con la actual versión de la base de datos según indica el enlace [32], de modo que puede conectarse a la Oracle Database 11.2. A nivel de red, utiliza el protocolo propietario, Transparent Network Substrate (TNS), que permite la conectividad punto a punto sobre TCP / IP.

El primer paso de la instalación es seguir las indicaciones de la tabla 74.

```
git clone https://github.com/oracle/python-cx_Oracle.git cx_Oracle
cd cx_Oracle
git submodule init
git submodule update
python setup.py install

mkdir -p /opt/oracle
cd /opt/oracle
unzip instantclient-basic-linux.x64-19.6.0.0.dbru.zip
sudo yum install libaio

sudo sh -c "echo /opt/oracle/instantclient_19_6 > /etc/ld.so.conf.d/oracle-
instantclient.conf"
sudo ldconfig
export LD_LIBRARY_PATH=/opt/oracle/instantclient_19_6:$LD_LIBRARY_PATH
```

Tabla 74 - Comando instalación de cx_oracle

Para conseguir confidencialidad e integridad de los datos transferidos, la opción más sencilla es la configuración mediante el propio mecanismo de Oracle. El cliente y el servidor (base de datos) realizan una negociación mediante el intercambio de claves Diffie-Hellman. El protocolo SSL es una opción si se dispone de infraestructura de clave pública (PKI), pero requiere un procedimiento más complejo.

Una vez realizada la instalación, el siguiente paso es acceder al archivo de configuración “*sqlnet.ora*” de la instancia de la base de datos y de cada una de las máquinas que alberguen el servidor y a su vez utilicen el cliente, cx_Oracle.

Para configurar los parámetros, se deben seguir las indicaciones de la guía oficial del enlace [33]. Oracle dispone para la configuración de una herramienta específica, Oracle Net Manager. Una vez lanzado, se debe acceder a las opciones de integridad y cifrado. A

continuación y como se muestra en la siguiente figura 67, se deben añadir los respectivos parámetros en el servidor y en el cliente. Es importante configurar el cifrado inicial.

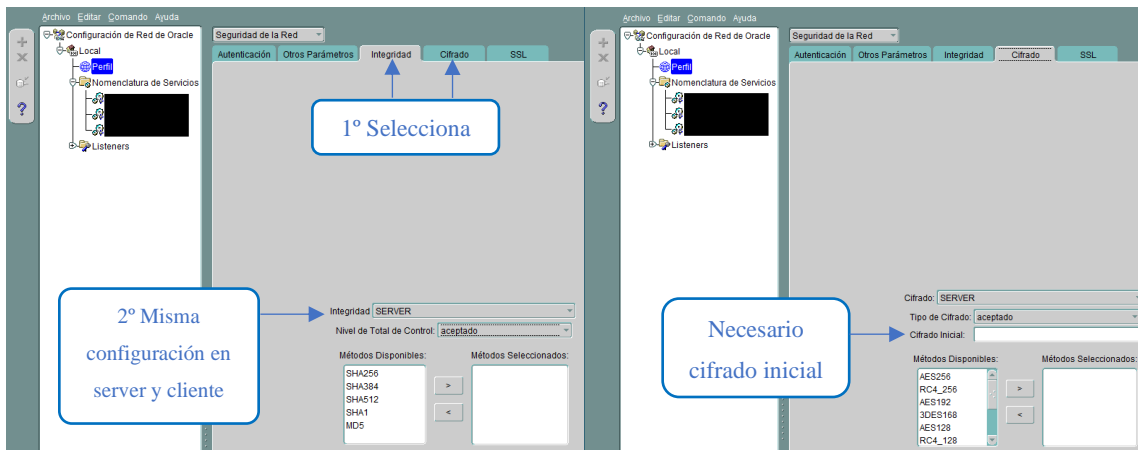


Figura 67 - Configuración de parámetros de red de Oracle Net Manager

El último paso es acceder al archivo de configuración del cliente y añadir o modificar los parámetros para que coincidan con los del servidor. La tabla 75 muestra un ejemplo con la función criptográfica SHA512 y el cifrado por clave simétrica, AES256.

```
SQLNET.CRYPTO_CHECKSUM_CLIENT = required
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT = (SHA512)
SQLNET.ENCRYPTION_CLIENT = required
SQLNET.ENCRYPTION_TYPES_CLIENT = (AES256)
```

Tabla 75 - Parámetros de configuración del cliente, cx_oracle

Bibliografía

- [1] *Información sobre el chatbot de RRHH de Cepsa.* (12 de Febrero de 2020). Obtenido de <https://www.cepsa.com/es/prensa/notas%E2%80%9393prensa/Cepsa-incorpora-la-inteligencia-artificial-a-la-gesti%C3%B3n-de-recursos-humanos-mediante-un-pionero-asistente-virtual>
- [2] *Información sobre bot de Arnie sobre reclutamiento de personal.* (12 de Febrero de 2020). Obtenido de <https://planetachatbot.com/inteligencia-artificial-y-recursos-humanos-recruiting-automation-99138f370e10>
- [3] *Estudio del impacto del chatbot en recursos humanos realizado por la compañía Sodexo.* (12 de Febrero de 2020). Obtenido de <https://www.sodexo.es/blog/chatbot-herramienta-recursos-humanos/>
- [4] *Tendencias y análisis globales sobre los chatbots para 2020.* (20 de Febrero de 2020). Obtenido de <https://observatorio-ia.com/tendencias-y-analisis-global-de-chatbots-para-2020>
- [5] *Documentación sobre Dialogflow.* (20 de Febrero de 2020). Obtenido de <https://cloud.google.com/dialogflow/docs>
- [6] *Documentación sobre Azure Bot Service.* (20 de Febrero de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/?view=azure-bot-service-4.0>
- [7] *Documentación sobre LUIS (NLU).* (20 de Febrero de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/?view=azure-bot-service-4.0>
- [8] *Documentación sobre QnA Maker (Base de conocimiento).* (20 de Febrero de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/?view=azure-bot-service-4.0>
- [9] *Documentación sobre Amazon Lex.* (20 de Febrero de 2020). Obtenido de <https://aws.amazon.com/es/lex/faqs/>

- [10] *Documentación sobre IBM Watson Assistant.* (20 de Febrero de 2020). Obtenido de <https://www.ibm.com/cloud/watson-assistant/docs-resources/>
- [11] *Precios de las capas disponibles en Dialogflow.* (20 de Febrero de 2020). Obtenido de <https://cloud.google.com/dialogflow/pricing>
- [12] *Condiciones de aceptación para el registro de datos de Dialogflow.* (20 de Febrero de 2020). Obtenido de <https://cloud.google.com/dialogflow/docs/data-logging-terms>
- [13] *Calculadora de precios de Azure.* (5 de Marzo de 2020). Obtenido de <https://azure.microsoft.com/es-es/pricing/calculator/>
- [14] *Información sobre el procedimiento de despliegue de Azure Bot Service.* (5 de Marzo de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-4.0>
- [15] *Documentación sobre RASA.* (5 de Marzo de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/bot-builder-deploy-az-cli?view=azure-bot-service-4.0&tabs=csharp>
- [16] *Procedimiento de implementación mediante CLI en Azure.* (5 de Marzo de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/bot-builder-deploy-az-cli?view=azure-bot-service-4.0&tabs=csharp>
- [17] *Procesamiento del lenguaje en aplicaciones de inteligencia artificial .* (14 de Marzo de 2020). Obtenido de <https://www.meneame.net/m/Art%C3%ADculos/como-entienden-bots-lenguaje>
- [18] Braun, D., Hernandez Mendez, A., & Matthes, F. (2017). Evaluating Natural Language Understanding Services. Proceedings of the SIGDIAL 2017 Conference, 174–185.
- [19] *Zapiens, herramienta de skill mapping.* (25 de Marzo de 2020). Obtenido de <https://zapiens.ai/>

- [20] *Principios de diseño de chatbots según Microsoft.* (14 de Marzo de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/bot-service-design-principles?view=azure-bot-service-4.0>
- [21] *Bot Framework utilizando la API de Dialogflow.* (25 de Marzo de 2020). Obtenido de <https://www.codeproject.com/Articles/1276434/Your-First-Chatbot-using-Microsoft-s-Bot-Framewo-2>
- [22] *Ejemplo de Microsoft sobre la incorporación de reconocimiento de lenguaje natural al bot.* (25 de Marzo de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/bot-builder-howto-v4-luis?view=azure-bot-service-4.0&tabs=python>
- [23] *Ejemplo de Microsoft sobre la Implementación de un flujo de conversación secuencial.* (25 de Marzo de 2020). Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/bot-builder-dialog-manage-conversation-flow?view=azure-bot-service-4.0&tabs=python>
- [24] *Bot Framework Emulator, aplicación diseñada para probar y debuggear bots que utilizan Bot Framework SDK.* (5 de Mayo de 2020). Obtenido de <https://github.com/microsoft/BotFramework-Emulator>
- [25] *Bot Framework Web Chat, tecnología que permite la construcción de clientes personalizables que se comunican con el middleware del sistema.* (5 de Mayo de 2020). Obtenido de <https://github.com/microsoft/BotFramework-WebChat>
- [26] *Protocolo Direct Line, protocolo establecido por Microsoft para permitir la comunicación entre un cliente chat y un bot desarrollado con Microsoft Bot Framework.* (5 de Mayo de 2020). Obtenido de <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-channel-directline?view=azure-bot-service-4.0>
- [27] *Offline-Directline, paquete no oficial para implementar el protocolo Direct Line en local.* (5 de Mayo de 2020). Obtenido de <https://www.npmjs.com/package/offline-directline>

- [28] *Buenas prácticas para el diseño de agentes en Dialogflow.* (5 de Mayo de 2020).
Obtenido de <https://cloud.google.com/dialogflow/docs/best-practices?hl=es#agent-design>
- [29] *Ejemplo de diagrama de clases de Microsoft Bot Framework.* (5 de Mayo de 2020).
Obtenido de <https://docs.microsoft.com/es-es/azure/bot-service/bot-builder-howto-v4-luis?view=azure-bot-service-4.0&tabs=python>
- [30] *Plataforma de Dialogflow para la configuración de los agentes.* (5 de Mayo de 2020).
Obtenido de <https://dialogflow.cloud.google.com/>
- [31] *Plataforma del cloud de Google para la configuración de las APIs de los agentes.* (5 de Mayo de 2020). Obtenido de <https://console.cloud.google.com/>
- [32] *Información sobre el cliente ligero cx_Oracle version 7.3.* (5 de Mayo de 2020).
Obtenido de https://oracle.github.io/python-cx_Oracle/
- [33] *Información sobre configuración mediante herramienta Oracle Net Manager.* (5 de Mayo de 2020). Obtenido de <https://docs.oracle.com/database/121/DBSEG/>