



UNIVERSIDAD DE VALLADOLID



Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad de Valladolid

TRABAJO FIN DE GRADO

**GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN**

RECONOCIMIENTO DE VOZ EN REUNIONES ÁGILES

AUTOR: MARIO GONZALEZ GOMEZ
TUTOR: ISABEL DE LA TORRE DÍEZ

19 de abril de 2021

TÍTULO: RECONOCIMIENTO DE VOZ EN REUNIONES
ÁGILES
AUTOR: MARIO GONZALEZ GOMEZ
TUTOR: ISABEL DE LA TORRE
DEPARTAMENTO: TSCIT

Miembros del Tribunal

PRESIDENTE: Miguel López-Coronado
SECRETARIO: Isabel de la Torre Díez
VOCAL: Beatriz Sainz de Abajo
CALIFICACIÓN:

Resumen del proyecto

Las metodologías ágiles son usadas a diario por gran cantidad de empresas con el objetivo de desarrollar sus proyectos de una manera más flexible y eficiente. La gestión de los equipos de trabajo consiste fundamentalmente en la gestión de las reuniones diarias para evaluar el estado del proyecto. El uso de reconocimiento de voz en estas reuniones hace que sean más dinámicas y eficientes, logrando que mediante comandos de voz se realicen las acciones propias de una metodología de desarrollo de software ágil.

Palabras clave

Reconocimiento de voz, Metodologías ágiles, Google Assistant, Dialogflow, ServiceNow.

Keywords

Voice Recognition, Agile Methodology, Google Assistant, Dialogflow, ServiceNow.

Abstract

Agile methodologies are used daily by many companies with the aim of developing their projects in a more flexible and efficient way. The management of the work teams consists mainly of the management of daily meetings to assess the status of the project. Using speech recognition in these meetings, makes them more dynamic and efficient, achieving the actions of an agile software development methodology through voice commands.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecerle a todo mi equipo de SilverStorm, ya que me han ayudado durante todos estos meses, enseñándome y formándome en numerosas herramientas y tecnologías que seguro me serán de utilidad a lo largo de mi carrera profesional.

Me gustaría agradecerle a mi familia todo el apoyo que me han dado durante estos años de carrera, en especial a mi padre Alfonso, mi madre Estrella y mi hermano Samuel por todo lo que han hecho por mí porque mis logros son suyos. Por último, gracias a mis compañeros y amigos Alejandro y David, ya que las siglas AFA significarán siempre una forma de vida.

ÍNDICE GENERAL

INTRODUCCIÓN.....	8
1.1 LAS METODOLOGÍAS ÁGILES	8
1.2 BENEFICIOS DE TRABAJAR CON METODOLOGÍAS ÁGILES	9
1.3 METODOLOGÍAS MAS UTILIZADAS	10
1.4 VOZ EN REUNIONES AGILE.....	11
1.5 MEDIOS NECESARIOS EMPLEADOS PARA EL DESARROLLO.....	12
1.6 ESTRUCTURA DEL DOCUMENTO.....	13
DESARROLLO DEL PROCESO.....	14
2.2 OBJETIVOS DEL PROCESO	15
2.3 DESCRIPCIÓN GENERAL DE LA SOLUCIÓN	17
2.3.1 RECONOCIMIENTO DE VOZ E INTERPRETACIÓN DEL LENGUAJE	17
2.3.2 INTERACCIÓN A TRAVÉS DE LA INTERFAZ	18
2.3.3 ACTIONS LISTENER	19
2.4 ARQUITECTURA GLOBAL	20
CAPACIDADES DEL ASISTENTE(S) DE VOZ.....	22
3.1 CORTANA	23
3.2 GOOGLE ASSISTANT	23
3.3 SIRI	23
3.4 ALEXA.....	24
3.5 BIXBY.....	24
DISEÑO TÉCNICO	27
4.1 DIALOGFLOW ESSENTIALS	28
4.1.1 INTENTS.....	29
4.1.2 ENTITIES.....	30
4.1.3 CONTEXTS.....	30
4.1.4FULFILLMENT	30
4.2 SERVICENOW BACKEND	31
4.2.1 REMOTE END POINT -REST API.....	31
4.2.2 CORE UTILS.....	32
4.2.2.1 ACTIONS RESOLVER	33
4.2.2.2 VOICE RESPONER.....	34
4.2.2.3 ACTION EXECUTOR.....	35
4.2.3 SERVICENOW FRONTEND.....	35
4.2.3.1 USER INTERFACE.....	35
4.2.3.2 ACTIONS LISTENER	39
4.3 DATA MODEL	40
4.4 SEGURIDAD Y AUTENTICACIÓN	41
MANUAL DE USUARIO.....	43
5.1 INICIAR EL ASISTENTE DE VOZ Y LA INTERFAZ DE SERVICENOW	43
5.2 INVOCACIÓN DE LOS INTENTS	45
CONCLUSIONES Y LÍNEAS FUTURAS.....	48
6.1 CONCLUSIONES	49
6.2 LIMITACIONES Y LÍNEAS FUTURAS	50
BIBLIOGRAFÍA.....	51

ÍNDICE DE ILUSTRACIONES

3.1. Ilustración 1 - Ciclo de vida en SCRUM.....	15
3.2. Ilustración 2 - Interpretación del lenguaje natural.....	17
3.3. Ilustración 3 - Uso de la interfaz de usuario.....	18
3.4. Ilustración 4 - Action Resolver	20
3.5. Ilustración 5 - Arquitectura global del sistema.....	21
3.6. Ilustración 6 - Arquitectura global final.....	27
3.7. Ilustración 7 - Arquitectura típica bot.....	28
3.8. Ilustración 8 - Configuración del Fulfillment	31
3.9. Ilustración 9_ Tabla sys_ui_message.....	35
3.10. Ilustración 10 - Pestaña de Project Backlog.....	37
3.11. Ilustración 11 - Pestaña de Sprint	38
3.12. Ilustración 12 - Widgets de información incrustados.....	39
3.13. Ilustración 13 - Configuración de la autenticación	42
3.14. Ilustración 14 - Configuración de la entidad OAuth.....	42
3.15. Ilustración 15 - Selección del portal	43
3.16. Ilustración 16 - Invocación del asistente	44
3.17. Ilustración 17 - Autenticación en ServiceNow.....	44
3.18. Ilustración 18 - Creación de un sprint.....	45
3.19. Ilustración 19 - Pantalla del nuevo sprint creado.....	46
3.20. Ilustración 20 - Obtención del Project backlog	46
3.21. Ilustración 21 - Interfaz del Project backlog.....	47
3.22. Ilustración 22 - Creación de una story.....	47
3.23. Ilustración 23 - Nueva story creada.....	48
3.24. Ilustración 24 - Pestaña de Logs	48

ÍNDICE DE TABLAS

3.25. Tabla 1 - Fases del proceso.....	18
3.26. Tabla 2 - Fases del uso de la interfaz de usuario.....	19
3.27. Tabla 3 - Porcentaje de acierto de los asistentes.....	25
3.28. Tabla 4 - Comparativa DialogFlow Y Alexa Developers.....	26
3.29. Tabla 5 - Intents.....	29
3.30. Tabla 6 - Entities.....	30
3.31. Tabla 7 - Contexts.....	30
3.32. Tabla 8 - Funciones del Action resolver.....	34
3.33. Tabla 9 - Funciones del Voice Responser.....	35
3.34. Tabla 10 - Funciones del Action Executor.....	36
3.35. Tabla 11 - Tablas de la aplicación.....	40
3.36. Tabla 12 - Campos de la tabla Actions History.....	41
3.37. Tabla 13 - Campos de la tabla GraphQL Actions.....	41

Capítulo 1

INTRODUCCIÓN

El propósito principal de este capítulo es introducir el concepto de metodología ágil dentro del ámbito de los equipos de trabajo encargados del desarrollo de software. Se explicarán cómo funcionan dichas metodologías, así como los beneficios de su uso sobre el rendimiento y productividad de una empresa. También se explicarán los principios de distintas metodologías para finalmente elegir una de las más utilizadas en el mundo laboral.

1.1 LAS METODOLOGÍAS ÁGILES

En el proceso de transformación digital, las empresas buscan la implementación de metodologías ágiles en sus distintos equipos de trabajo para lograr entregar sus proyectos y servicios en unos tiempos más reducidos y con una mayor calidad.

En la actualidad, las empresas se enfrentan a una revolución digital que provocan cambios en su forma de trabajar, y algunas no logran adaptarse a esta nueva era digital, quedando a la cola en innovación y productividad. Para ello surgen las metodologías ágiles o metodologías *Agile* que pretenden que la interacción del cliente y del equipo de desarrollo sea lo más provechosa posible, haciendo que los cambios requeridos por el cliente tengan una respuesta más rápida y eficaz. Así, se consiguen unos equipos de desarrollo capaces de adaptarse a las condiciones y cambios del proyecto de una manera sencilla y simple [1].

Un artículo de Forbes titulado “Por qué Agile se está comiendo el mundo” describe como Marc Andreessen -cofundador de *Netscape Communications Corporation* advirtió en 2011 de que las empresas tecnológicas e innovadoras dominarían el mercado, algo que queda tangente hoy en día cuando empresas como Amazon, Apple, Google o Microsoft son las más grandes compañías, algunas de las cuales utilizan metodologías ágiles en sus desarrollos [2].

La metodología Agile es por tanto una fórmula para el desarrollo de proyectos software que necesitan flexibilidad y capacidad de adaptación a las necesidades del cliente cuyo objetivo principal es la mejora de resultados. Una de las bases de estas metodologías es la

retroalimentación, trabajando en periodos de tiempo conocidos como *sprints* que pueden tener una duración variable como una o dos semanas. Al final de dicho sprint, se entregan los desarrollos y avances y se comienza de nuevo el proceso, así el cliente es consciente de los avances y novedades, y en caso de que este requiera nuevas funcionalidades o cambios, el equipo podrá responder de manera rápida y eficaz.

Así, la “agilidad” permite que el cliente establezca prioridades y rectifique según se van consiguiendo avances en el desarrollo del proyecto, mejorando la experiencia del cliente, que no tendrá que esperar al desarrollo completo del proyecto para poder revisar su funcionamiento y exigir cambios. Según un estudio de *Pulse of the Profession*, en 2017, ya el 79% de las organizaciones en todo el mundo utilizaban una metodología agile [3].

Siguiendo con la encuesta *Pulse of the Profession* de 2021, un 53% de las organizaciones encuestadas consideran de alta prioridad la creación de una cultura receptiva a los cambios, siendo una de sus áreas de mayor importancia la agilidad organizacional, punto en el que el uso de una metodología Agile puede ser clave para mejorar los resultados [4].

1.2 BENEFICIOS DE TRABAJAR CON METODOLOGÍAS ÁGILES

El uso de metodologías Agile presenta una serie de ventajas frente al desarrollo de proyectos usando formas tradicionales. A continuación, se enumeran algunas de estas ventajas:

- **Versatilidad.** La gestión de proyectos no es algo preestablecido que responda a un patrón, sino que se orienta hacia la consecución de objetivos reales desde el inicio del proyecto. El uso de Agile permite que el proyecto se ajuste a las necesidades cambiantes del cliente.
- **Mejora de la calidad del proyecto.** El uso de Agile fomenta la búsqueda de la excelencia del producto por parte de los equipos de trabajo. Los desarrollos son comprobados y mejorados continuamente, incrementando así la calidad del resultado final.
- **Mayor satisfacción del cliente.** La mejora de calidad del proyecto conlleva inherentemente una mayor satisfacción del cliente, que además es participe de todo el proceso de desarrollo, pudiendo comprobar cada poco tiempo el estado de su proyecto.
- **Mayor motivación de los trabajadores.** El hecho de que los equipos se reúnan cada poco tiempo para poner en común sus avances hacen que la ayuda en el equipo de desarrollo sea mayor y, por tanto, los equipos sean más innovadores y creativos.
- **Uso de métricas reales.** El desarrollo de proyectos conlleva el uso de métricas como tiempo coste. Las estimaciones de estos valores suelen ser más realistas en proyectos que utilizan metodologías Agile.
- **Reducción de costes.** La gestión Agile de los proyectos supone que el seguimiento continuo de los posibles errores o fallos, por lo que es muy poco probable que un proyecto fracase.

1.3 METODOLOGÍAS MAS UTILIZADAS

Existen muchas metodologías Agile, por eso se van a estudiar algunas de las más utilizadas por las empresas actualmente [1].

- **Extreme Programming XP.** Es una metodología muy usada por startups y empresas en proceso de consolidación, ya que su principal objetivo es el de ayudar en las relaciones entre empleados y clientes, potenciando las relaciones personales a través del trabajo en equipo. Sus principales fases son:
 - Planificación con el cliente.
 - Diseño del proyecto.
 - Codificación, donde se desarrolla en equipo para maximizar el rendimiento.
 - Testing periódico para comprobar que los desarrollos funcionan correctamente.
- **Kanban.** Es una herramienta que resulta de especial utilidad para los responsables de proyectos ya que consiste en la elaboración de un cuadro con tres columnas en las que se reflejan los tres estados en los que puede encontrarse una tarea: pendiente, en proceso y terminada. El cuadro debe ser visible por parte de todos los miembros del equipo para que todas sean conscientes del estado de cada tarea. Sus ventajas principales son:
 - Métricas visuales e intuitivas.
 - Mejora del rendimiento del equipo.
- **Agile Inception.** Se basa en el concepto de “*elevator pitch*” que consiste en tener una breve reunión entre el equipo de trabajo y los socios o jefes, y que normalmente duran menos de cinco minutos. Estas reuniones tienen como objetivo establecer los objetivos generales de la empresa.
- **Design Sprint.** Es la metodología Agile de Google, orientada a la promoción de startups tecnológicas. Se basa en un proceso que suele durar 5 días en el que la empresa o negocio tiene que definir las cuestiones de diseño, prototipado y testeo de clientes. Las etapas de trabajo se dividen en periodos de tiempo denominados *sprint* de manera que se busca acortar el tiempo de trabajo necesario para desarrollar un producto ya que con la fase de prototipado se evalúa la calidad de la idea de la *startup*.
- **SCRUM.** Se basa en una estructura de desarrollo incremental en el que el proyecto se divide en pequeñas partes llamadas “*story*” que siguen un proceso dividido en tres etapas: análisis, desarrollo y testing. Cada *story* sigue una serie de iteraciones conocidas como Sprint. Estos sprint tienen una duración variable que suele ir desde una semana a dos semanas. En cada iteración en la fase de desarrollo se comprueba el estado de las stories del proyecto y se fijan cuáles son los pasos a seguir en el siguiente sprint para completar

la story y así pase a la fase de testeo. Esta metodología permite abordar proyectos complejos y con desarrollos de larga duración, ya que el utilizar stories que se revisan cada poco tiempo permite seguir el estado del proyecto con facilidad e introducir nuevos cambios si el cliente así lo desea.

El proceso que sigue una story es el siguiente:

- Se parte de la lista de objetivos priorizada que se utiliza como plan del proyecto. En la lista, se balancean y priorizan las tareas en función de los beneficios que aportan para el cliente y el coste de estos.
- El equipo establece las tareas de la iteración para desarrollar los requisitos seleccionados y se estima (en horas) el coste o esfuerzo que suponen dichas tareas. Se asignan las tareas pudiendo una tarea ser asignada a más de un miembro del equipo para así compartir conocimientos.
- Cada día se celebra una reunión conocida como “daily” en el que el equipo repasa el trabajo que se está realizando, como está avanzando la story y en caso de existir, compartir los problemas o limitaciones que están impidiendo avanzar en la resolución de las tareas.
- Al final de cada iteración (sprint) se revisa el estado del proyecto, las stories completadas y testeadas y se analiza cómo ha sido la manera de trabajar del equipo, como se podría mejorar la productividad y se fijan las siguientes stories a completar.

De aquí en adelante, se usará Scrum como la metodología Agile de referencia para este proyecto, utilizando reconocimiento de voz para gestionar una reunión típica de esta metodología [5].

1.4 VOZ EN REUNIONES AGILE

En las anteriores secciones se ha introducido el concepto de “metodologías Agile” y se ha visto como la gran mayoría de estas se basan en las reuniones entre equipo y cliente, así como reuniones del propio equipo de trabajo.

Estas reuniones pueden tener una duración variable y requieren de una persona del equipo, que se encarga de crear, actualizar o eliminar registros propios de una reunión como las ya mencionadas stories o sprints. El uso de reconocimiento de voz permite agilizar las acciones y reducir el tiempo de reunión garantizando la misma calidad.

1.5 MEDIOS NECESARIOS EMPLEADOS PARA EL DESARROLLO

Para la correcta realización del proyecto planteado, será necesaria la utilización de diferentes herramientas de *hardware* y *software*:

Hardware

- PC portátil con distribución Windows y las siguientes características:
 - Procesador Intel ® Core ™ i7-10510U CPU@ 1.80 GHz
 - 16 GB de memoria RAM.
 - Disco duro HDD de 500 GB de capacidad.

Software

- Dialogflow: Herramienta de creación de chatbots capaz de entender el lenguaje natural y que provee infraestructura para recrear conversaciones y construir diálogos
- Google Assistant: Asistente en disponible en móviles, tables y dispositivos inteligentes que se encarga de la gestión de las conversaciones.
- Instancia de ServiceNow: Plataforma como servicio (PaaS) basada en *cloud* que provee la infraestructura necesaria para el desarrollo, ejecución y administración de aplicaciones.

1.6 ESTRUCTURA DEL DOCUMENTO

El presente documento consta de seis capítulos, incluyendo el capítulo de introducción. El contenido de los posteriores capítulos es el que sigue:

Capítulo 2- Desarrollo del proyecto. Se presenta como se va a implementar la solución del uso de un asistente de voz para la gestión de reuniones Agile. Se explicarán las distintas partes de la arquitectura, así como una introducción a su funcionamiento.

Capítulo 3- Capacidades del asistente(s) de voz. Se realiza un estudio exhaustivo de los distintos asistentes de voz disponibles en el mercado, comparando su usabilidad y sus características para escoger uno que se utilizará en el desarrollo de este producto. Además, se encuadra el mercado de los asistentes de voz en la situación económica actual.

Capítulo 4- Diseño técnico. En este capítulo se detalla cual es la arquitectura global final y se detalla cómo funciona y se ha configurado cada parte del diagrama global de la aplicación.

Capítulo 5- Manual de usuario. Se realiza un manual de cómo debe ser inicializada la aplicación por un usuario. Se explica cómo debe realizarse la autenticación, así como algunos casos de uso para observar el funcionamiento de la aplicación.

Capítulo 6- Conclusiones y líneas futuras. Por último, se extraen las ventajas del uso de un asistente de voz en la gestión de reuniones Agile, así como un estudio de los posibles futuros desarrollos que aumentarían el valor de esta aplicación.

Capítulo 2

DESARROLLO DEL PROCESO

Las metodologías ágiles se fundamentan en una serie de principios que anteponen:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

Scrum es un modelo de desarrollo ágil caracterizado por:

- Adoptar una estrategia de desarrollo incremental e iterativa, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.

2.1 DEFINICIÓN DEL PROCESO

SCRUM modela el ciclo de vida del proyecto en base a incrementos e iteraciones del ciclo de desarrollo, que en Scrum se conocen como Sprints. El proceso parte de una lista (Product o Project Backlog) de funcionalidades (Stories). En base a las priorizaciones se realiza una planificación inicial estableciendo qué incluir en cada iteración (Sprint) generando un Sprint Backlog.

Antes de cada Sprint se realiza la planificación más detallada (Sprint Plan) refinando la información de las Stories, definiendo el tamaño de las Stories mediante puntos, identificando las tareas que deben realizarse en cada Story y asignándolas a miembros del equipo. Esta planificación de las tareas a realizar en la iteración se divide en dos partes:

1. Qué: Primera parte de la reunión

En esta parte, el cliente presenta al equipo la lista de requisitos priorizada del producto o

proyecto, pone nombre a la meta de la iteración y propone los requisitos más prioritarios a desarrollar en ella. Después, el equipo examina la lista, pregunta al cliente las dudas que le surgen, se añaden más detalles y se seleccionan los objetivos/requisitos más prioritarios que el equipo prevé completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

2. Cómo: Segunda parte de la reunión.

El equipo planifica la iteración y define las tareas necesarias para poder completar cada objetivo/requisito, creando la lista de tareas de la iteración (Sprint Backlog) basándose en la definición de hecho (DoD).

Los miembros del equipo realizan una estimación conjunta del esfuerzo necesario para realizar cada tarea y se autoasignan las tareas que pueden realizar.

Al final del Sprint el equipo realiza una Demo en la Sprint Review que debería ofrecer un producto entregable. Fruto del Sprint Review se añaden o eliminan Stories y se prepara el siguiente Sprint en la reunión de planificación de Sprint (Sprint Planning). La Ilustración 1 muestra un ejemplo de ciclo de vida en Scrum (imagen cedida por SilverStorm).

El proceso concreto que se aborda en este proyecto es el de una reunión de planificación de Sprint (Sprint Planning).

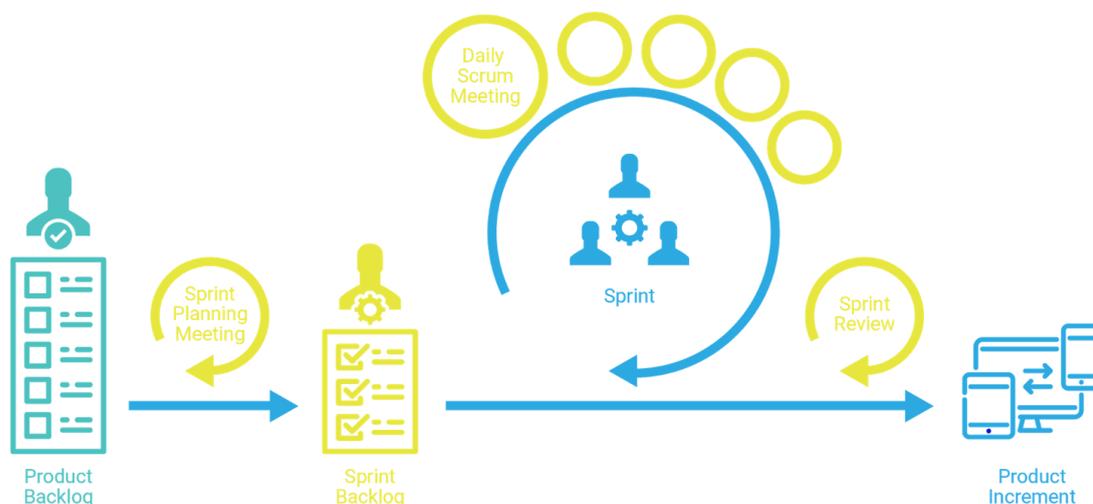


Ilustración 1 - Ciclo de vida en SCRUM

2.2 OBJETIVOS DEL PROCESO

El objetivo principal del proyecto es el de desarrollar, probar y validar una solución de *cloud computing* para la gestión a través de comandos de reconocimiento de voz naturales de los equipos de trabajo que utilizan metodología de desarrollo de software ágiles. La gestión de

los equipos de trabajo consiste fundamentalmente en la planificación de las reuniones diarias para evaluar el estado del proyecto, de las reuniones periódicas para la planificación de sprints o iteraciones y de las reuniones retrospectivas para mejorar la forma de trabajar.

Para poder alcanzar el objetivo principal del proyecto es necesario conseguir los siguientes objetivos específicos:

OBJETIVOS TÉCNICOS:

- Identificar y analizar funcionalmente las acciones más habituales llevadas a cabo por equipos ágiles durante sus reuniones para su estandarización y modelización.
- Desarrollar un habilitador digital para la gestión completa de las acciones de los equipos de trabajo que sigan metodologías ágiles sobre la plataforma ServiceNow. de servicios Saas
- Desarrollar una interfaz API para ejecutar en el sistema estas acciones de forma entendible y accesible.
- Configurar y desarrollar un sistema intérprete de voz que reconozca patrones relacionados con reuniones ágiles y los traduzca a comandos básicos que puedan ser enviados al sistema gestor ServiceNow.
- Desarrollar la integración en el sistema intérprete que permita consumir el API publicado por el sistema gestor para enviarle los comandos adecuadamente. Además, en base a la respuesta recibida, para que eventualmente interactúe con el usuario solicitando información necesaria para completar la transacción si es que se requiere.
- Validar técnicamente en los propios equipos de SilverStorm el correcto funcionamiento del habilitador.
- Garantizar la fiabilidad del nuevo producto. Como en todos los productos, ha de garantizarse una alta calidad y fiabilidad del producto.

OBJETIVOS ESTRATÉGICOS PARA LA EMPRESA:

- Mejorar y ampliar la línea de negocio para mantener el estatus adquirido en este tipo de soluciones resulta clave la modernización y actualización constante, que se conseguirá con la realización de este proyecto.
- La incorporación de la tecnología del reconocimiento de voz a los desarrollos realizados por SilverStorm sobre la plataforma ServiceNow. de servicios Saas.

2.3 DESCRIPCIÓN GENERAL DE LA SOLUCIÓN

El objetivo de esta sección es la explicación del proceso que sigue una orden desde que es pronunciada por el usuario en lenguaje natural, hasta que es recibida por la API y provoca una acción. De la misma manera, estos cambios quedarán registrados una interfaz de usuario a la que el usuario podrá acceder para utilizar las acciones propias del asistente de voz de forma intuitiva. En esta sección se definirán los distintos componentes y las interacciones entre cada uno de ellos.

Se pretende proporcionar una vista global de todo el sistema, donde se muestre la función que cada componente desempeña dentro de él, así como la forma en que cada uno se comunica con el resto de los componentes de su entorno.

2.3.1 RECONOCIMIENTO DE VOZ E INTERPRETACIÓN DE LENGUAJE

La primera parte de la solución consistirá en el desarrollo de un asistente de voz. El usuario emitirá una orden relacionada con una acción propia de una reunión Agile y dicha orden deberá ser escuchada, traducida y finalmente ejecutará una acción. Estas acciones se definirán más adelante.

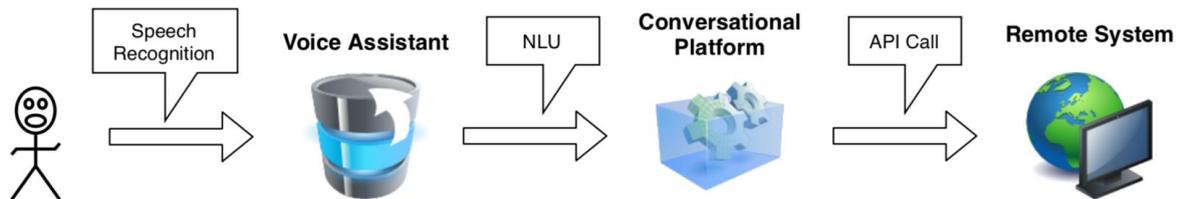


Ilustración 2 - Interpretación del lenguaje natural

Como puede observarse en la Ilustración 2, desde que el usuario pronuncia una frase hasta que la API del sistema remoto emite una respuesta, la información pasa por tres estados. Estos estados pueden observarse en la Tabla 1.

Fase	Descripción
<i>Speech Recognition</i>	Proceso por el cual se transcribe en formato texto la frase literal expresada por el usuario a través de la voz.
NLU (<i>Natural Language Understanding</i>)	A partir del texto escrito en la fase anterior se extrae la petición real del usuario. Se identifica el “ <i>Intent</i> ” de la frase a partir de una serie de frases de entrenamiento previamente configuradas.

<i>API Call</i>	<p>A partir del “<i>Intent</i>” identificado en la fase anterior se realiza una llamada al API del sistema remoto, el cual será el responsable de devolver la información solicitada.</p> <p>La plataforma conversacional es la encargada de definir el formato y la estructura de la llamada al API remoto, tanto la llamada como la respuesta.</p>
-----------------	--

Tabla 1 - Fases del proceso

De la misma manera, como puede verse en la Ilustración 2, en el proceso intervienen los siguientes actores:

- **Usuario:** Se refiere a la persona que va a interactuar a través de la voz con el asistente virtual.
- **Voice Assistant:** Asistente virtual que transcribirá la petición del usuario y enviará el texto transcrito para ser procesado. No se implementará ningún tipo de funcionalidad adicional, sino que simplemente se utilizará uno de los asistentes de voz disponibles en el mercado.
- **Conversational platform:** Es el sistema encargado de entender y extraer la petición real del usuario del texto expresado en lenguaje natural, utilizando técnicas de NLU. Además, realizará el enlace con el sistema remoto encargado de ejecutar dicha petición.
- **Remote System:** Sistema externo que recibirá la petición y realizará la acción adecuada para atender la petición del usuario.

2.3.2 INTERACCIÓN A TRAVÉS DE LA INTERFAZ WEB

La segunda parte de la solución consistirá en una interfaz de usuario donde el usuario podrá comprobar información relativa a las acciones que va realizando, así como mostrar diferentes listas y tablas de registros.



Ilustración 3 – Uso de la interfaz de usuario

Como puede observarse en la Ilustración 3, en el proceso del uso de la interfaz por parte del usuario intervienen dos fases que son descritas en la Tabla 2.

Fase	Descripción
<i>Open URL</i>	Apertura por parte del usuario del <i>Front-End</i> (URL) diseñado sobre plataforma Web.
API Call	Integración a través del API REST publicado en el sistema remoto. Desde el <i>Front-End</i> se realizan llamadas sucesivas al API que refrescarán el estado de la pantalla en tiempo real. El formato y estructura de la llamada al API vendrán determinados por el sistema remoto.

Tabla 2 - Fases del uso de la interfaz de usuario

En el proceso intervienen los siguientes actores:

- **Usuario:** Se refiere a la persona que va a acceder a la página web donde se publica la información.
- **Web Interface:** Representa el *Front-End* publicado sobre plataforma web y accesible desde cualquier localización y dispositivo (Portátil, PC, Smartphone, Tablet). Será el encargado de representar gráficamente el resultado de las peticiones realizadas por el usuario en tiempo real.
- **Remote System:** Representa el *Back-End* que contiene la lógica necesaria para ejecutar finalmente la petición realizada por el usuario y devolver el resultado al *Front-End* (*Web Interface*).

2.3.3 ACTIONS RESOLVER

Ambos elementos no funcionan de forma aislada, sino que existe un punto de unión entre ellos, el cual se encarga de sincronizar las peticiones que el usuario solicita a través de la voz con la ejecución de estas y su presentación a través de una interfaz web.

Este elemento es el **Action Resolver** cuya función es precisamente, sincronizar ambos componentes, es decir, sincronizar la petición del usuario con su representación gráfica en pantalla, como puede observarse en la Ilustración 4.

El Action Resolver actúa como un intérprete de las acciones a llevar a cabo, de forma que la solicitud de un usuario se pueda traducir a un lenguaje que la interfaz web sea capaz de entender y, por tanto, la lógica del *front-end* pueda presentarlo en pantalla.

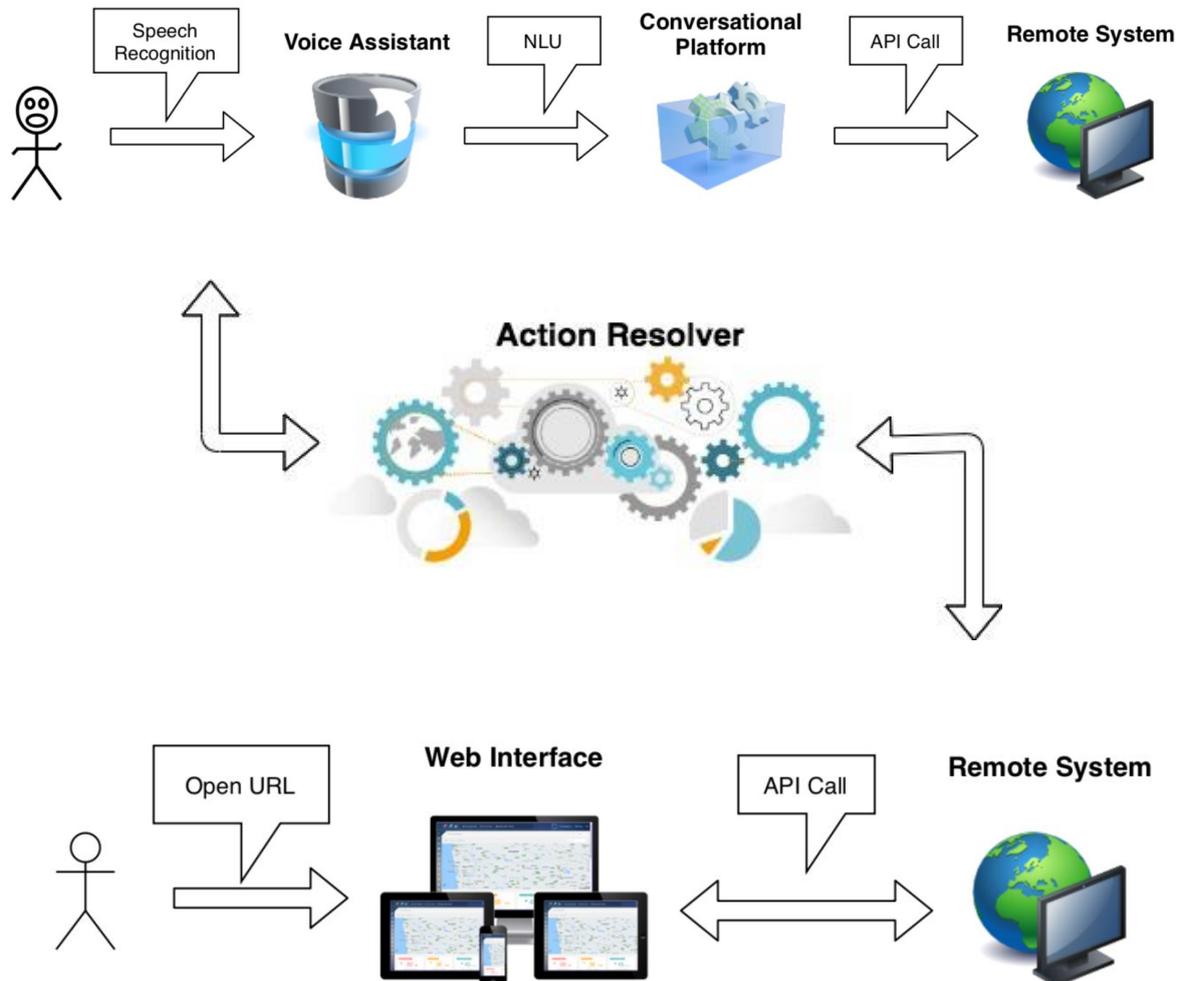


Ilustración 4 - Action Resolver

Una vez se defina la arquitectura global, se explicarán en detalle los diferentes componentes del sistema, así como su funcionamiento.

2.4 ARQUITECTURA GLOBAL

En la sección anterior, se ha explicado cuales son los componentes del sistema, indicando sus funciones principales y cómo funciona la comunicación entre componentes. A nivel global, la arquitectura del sistema puede verse en la Ilustración 5.

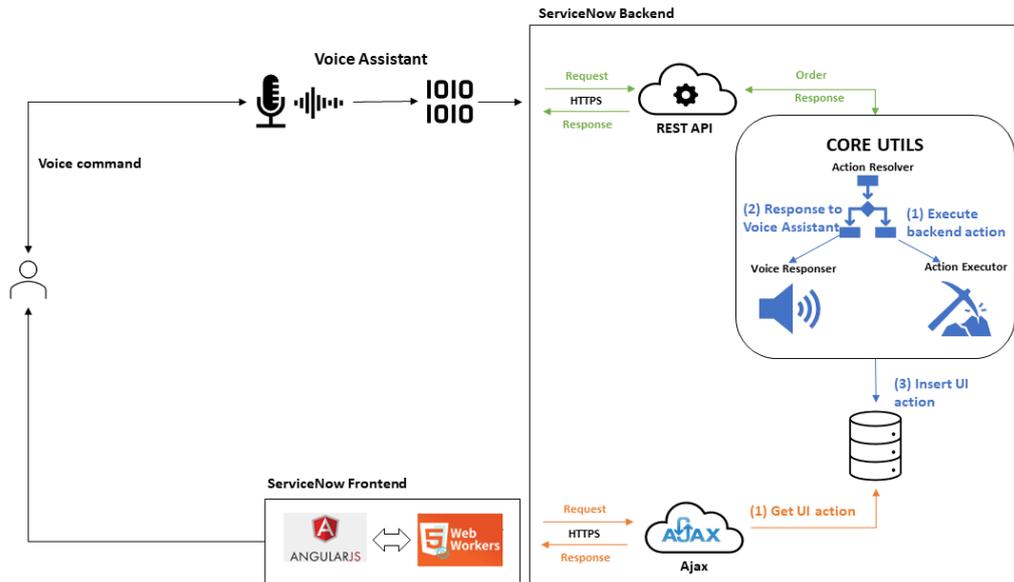


Ilustración 5 - Arquitectura global del sistema

Aunque en este capítulo se han introducido los componentes que componen el proceso, que pueden observarse en la Ilustración 5, más adelante se definirá cada uno en detalle, no sin antes realizar un estudio, en el capítulo tres, de los asistentes de voz disponibles en el mercado para finalmente escoger el que mejor se adapte a los requerimientos del proyecto

Capítulo 3

CAPACIDADES DEL ASISTENTE(S) DE VOZ

Un asistente de voz es un software que permite a un usuario interactuar con él a través de comandos de voz para solicitar información o llevar a cabo acciones. Cada día estos asistentes de voz son usados por multitud de personas a través de sus móviles, dispositivos inteligentes o computadoras, y el número de personas que hacen uso de asistentes de voz no para de crecer, al igual que la calidad y el número de funciones que estos cubren.

Según algunas consultoras especializadas en tecnología como Best o Strategy Analytics en el año 2020 estas fueron algunas de las estadísticas más importantes acerca de los asistentes de voz [6]:

- El **70%** de la población utiliza un asistente de voz en su vida diaria.
- El **50 %** de las búsquedas totales en Internet son búsquedas por voz.
- Según Ovum, a lo largo de 2021 habrá casi tantos asistentes activados por voz en el planeta como personas.

Para poner en contexto estos datos, los teléfonos móviles tardaron casi 30 años en superar al número de humanos en el planeta, mientras que los dispositivos asistentes a través de voz pueden llegar en menos de la mitad de ese tiempo. Además, se predice que la tasa de adopción de altavoces inteligentes será mucho más rápida que la de Internet, televisión, teléfonos inteligentes e incluso radio [7].

Según el informe “Global Speech and Voice Recognition Market-Industry Trends and Forecast 2027” de Data Bridge Market Research, se espera que el mercado mundial de la voz, conversación y la tecnología del reconocimiento de voz alcance los 10.731 millones de dólares para 2027, con un crecimiento anual 25.9% en el periodo de pronóstico del año 2020 al 2027. Los principales factores que impulsan este importante desarrollo son la integración cada vez más efectiva de estos dispositivos en nuestras vidas debido a la mayor demanda en el uso de sistemas biométricos [8].

Es por ello, que el hecho de elegir un asistente para este proyecto se convierte en un punto de importancia, estudiando cuales son los principales asistentes del mercado y cuáles son sus puntos fuertes, para finalmente acabar eligiendo uno de ellos [9].

Para ello se analizarán los siguientes asistentes:

1. Cortana
2. Google Assistant
3. Siri
4. Alexa
5. Bixby

3.1 CORTANA

Cortana es el asistente de voz de Microsoft que se incluye en Windows 10 para PC y en las versiones de Windows Phone 8.1 para móviles, aunque también está disponible para el altavoz inteligente de Microsoft. Como asistente de voz, puede realizar búsquedas, ayudar con tareas cotidianas o el tiempo, recordatorios, etc. Es un asistente que en comparación con Google Assistant o Alexa tiene mucho que mejorar, ya que muchas veces no reconoce correctamente las frases que emite el usuario.

Dado que es un asistente propio de Windows, su uso se limita a estos dispositivos por lo que por estas limitaciones se descarta su uso.

3.2 GOOGLE ASSISTANT

El asistente de Google apareció por primera vez en el mercado en 2016 de la mano de Google Allo, algo más tarde que Alexa, sin embargo, con dos años menos de vida, su evolución ha sido tan grande que es un competidor de Alexa por el primer puesto entre los asistentes digitales.

El asistente de Google es capaz de acceder a cualquiera de las funcionalidades de Google como noticias, tiempo, tráfico, música, alarmas, etc. Además, a través de la aplicación de Google Home es posible controlar bombillas, altavoces o termostatos.

Una de las ventajas encontradas es la posibilidad de integrar el asistente de Google con Dialogflow, una plataforma de desarrollo de asistentes de voz en la que es posible crear agentes que, a través de comandos de voz, realicen acciones programables a través de llamadas a un End Point. Dado que en Dialogflow es posible definir respuestas y llamadas al End Point a través del uso de *webhooks*, Google Assistant se postula como un candidato óptimo para el desarrollo de este proyecto, ya que es posible utilizar el asistente de Google desde cualquier tipo de dispositivo, ya sea un dispositivo móvil, computadora o altavoz inteligente.

3.3 SIRI

Siri es el asistente digital de Apple, además de ser el pionero de los asistentes que aquí se describen, puesto que su fecha de lanzamiento se remonta al 2011. Siri permite controlar

diversas funcionalidades del móvil además de los dispositivos conectados compatibles con Apple Homekit.

Esto supone un problema porque mientras Alexa o Google Assistant pueden estar disponibles en distintos dispositivos de diferentes marcas, Siri solo está disponible en dispositivos Apple, es decir para iOS, macOS, tvOS y watchOS. La cantidad de dispositivos compatibles con Siri son mucho menores que los disponibles para Alexa o Google Assistant por lo que se descarta su uso para este proyecto.

3.4 ALEXA

Alexa es el asistente digital de Amazon, fue lanzado en 2014 en Estados Unidos a través de los altavoces inteligentes Amazon Echo. Este asistente cubre gran número de funcionalidades, su tiempo de respuesta es muy rápido y su uso no está restringido a una marca de dispositivos como ocurre con Cortana o Siri.

Una de sus ventajas es que al igual que con Google Assistant, es posible desarrollar aplicaciones para este asistente digital, disponible para altavoces inteligentes Amazon Echo, en otros altavoces inteligentes y en la aplicación móvil. Mediante el uso de su consola de desarrollo, Amazon permite crear aplicaciones conocidas como “Skills” que son funcionalidades que se pueden añadir al asistente de voz. En cuanto a la forma de crear el proyecto es similar a la utilizada por Google Assistant, utilizando frases de entrenamiento que, al coincidir con lo dicho por el usuario, lanzan intents que realizan llamadas a un End Point.

Al igual que Google Assistant, se trata de un asistente óptimo para el desarrollo de este proyecto, dado que está disponible para diferentes marcas y que posee un entorno de desarrollo de aplicaciones con una API muy desarrollada.

3.5 BIXBY

Es el asistente digital de Samsung, lanzado en 2017 y solo está disponible para smartphones y tablets de Samsung. Llegó a anunciarse para un altavoz inteligente llamado Galaxy Home que se anunció en 2018 pero que aún no se ha lanzado al mercado.

Sus funcionalidades son más limitadas que la del resto de asistentes, y el hecho de que esté solo disponible para dispositivos Samsung reduce mucho su uso, por lo que se descarta para el desarrollo de este proyecto.

La decisión de elegir un asistente se resume por tanto en las diferencias entre Google Assistant y Alexa, ya que estos asistentes están disponibles para cualquier dispositivo, independientemente de su marca, mientras que el resto son específicos de una marca. El entorno de desarrollo de ambos es muy similar, basándose en la definición de “intents” que se lanzan cuando la frase emitida por el usuario coincide con una de las que los intents tienen definidas como frases de entrenamiento de entrada.

En cuanto a usabilidad, Alexa tiene un mayor número de dispositivos compatibles a su disposición y permite descargar aplicaciones desde un catálogo de Skills. Por otra parte, Alexa mejora progresivamente sus respuestas a medida que el usuario utiliza dicho asistente. Por su parte, Google Assistant ofrece respuestas más precisas ya que utiliza machine learning para generar más frases de entrenamiento que puedan coincidir con las peticiones del usuario, aumentando su eficacia [10].

En el siguiente estudio que puede verse en la Tabla 3 realizado por la web Loup Ventures que compara el porcentaje de peticiones respondidas correctamente y las peticiones entendidas, de un total de 800 peticiones, de los diferentes asistentes de voz, es Google Assistant el asistente que mejores resultados demuestra [11].

Asistente	Peticiones respondidas correctamente	Peticiones entendidas correctamente
Google Assistant	87,9%	100%
Siri	74,6%	99,6%
Alexa	72,5%	99,0%
Cortana	63,4%	99,4%

Tabla 3 - Porcentaje de acierto de los asistentes

Otro punto importante a la hora de elegir uno de estos dos asistentes es el entorno de desarrollo y los costes que pueda tener el uso de este asistente. Todos estos puntos se pueden observar en la Tabla 4, que trata la comparativa de algunos de los puntos más importantes de ambos entornos de desarrollo [12], [13].

	Dialogflow	Alexa Developers
Canales	Texto y voz	Texto y voz
Facilidad de uso	La interfaz web es muy intuitiva y sencilla para la creación de bots.	La interfaz web es algo más compleja que la provista por Dialogflow.
Integraciones	Google Assistant, Slack, Viber, Facebook Messenger, Twitter, Twilio...	SMS, Slack, Facebook Messenger, Kik, Twilio
Idiomas	Más de 20 idiomas incluyendo el español, inglés, portugués, etc.	Solo disponible en inglés.
Detección de Intents	Muy fuerte	Débil como puede verse en la Tabla 1
Web Chatbot	Si-	No

	Proporciona integración lista para usar con un chatbot incorporado	
Costes	<ul style="list-style-type: none"> - El plan estándar es gratuito (óptimo para pequeños proyectos) - Versión para empresas: \$0,002/petición de texto y 0,0065/petición de voz cada 15 segundos de audio 	<ul style="list-style-type: none"> - El primer año, 10k peticiones de texto y 5k peticiones de voz al mes gratuitas - Después \$0,004/petición de voz y \$0,000075/petición de texto

Tabla 4 - Comparativa DialogFlow Y Alexa Developers

Teniendo todos estos puntos en cuenta y en base a que nuestro asistente de voz debe ser robusto y que nuestras peticiones se entiendan lo mejor posible, dado que podría provocar cambios no deseados en la instancia de ServiceNow y que puede ser necesario que esté disponible en diversos idiomas se ha decidido escoger Google Assistant como el asistente utilizado para este proyecto.

DISEÑO TÉCNICO

El objetivo de este apartado es detallar el diseño técnico de los componentes definidos en la arquitectura de la solución.

Se pretende mostrar el diseño de los distintos componentes, sus integraciones, el modelo de datos y la interfaz de usuario. La arquitectura global final se muestra en la Ilustración 6 tras haber decidido que el asistente que se va a utilizar es Google Assistant junto con su plataforma de desarrollo Dialogflow, se ha sustituido esa parte en el diagrama. Se irán explicando una por una las partes de la arquitectura detallando su implementación y funcionalidades.

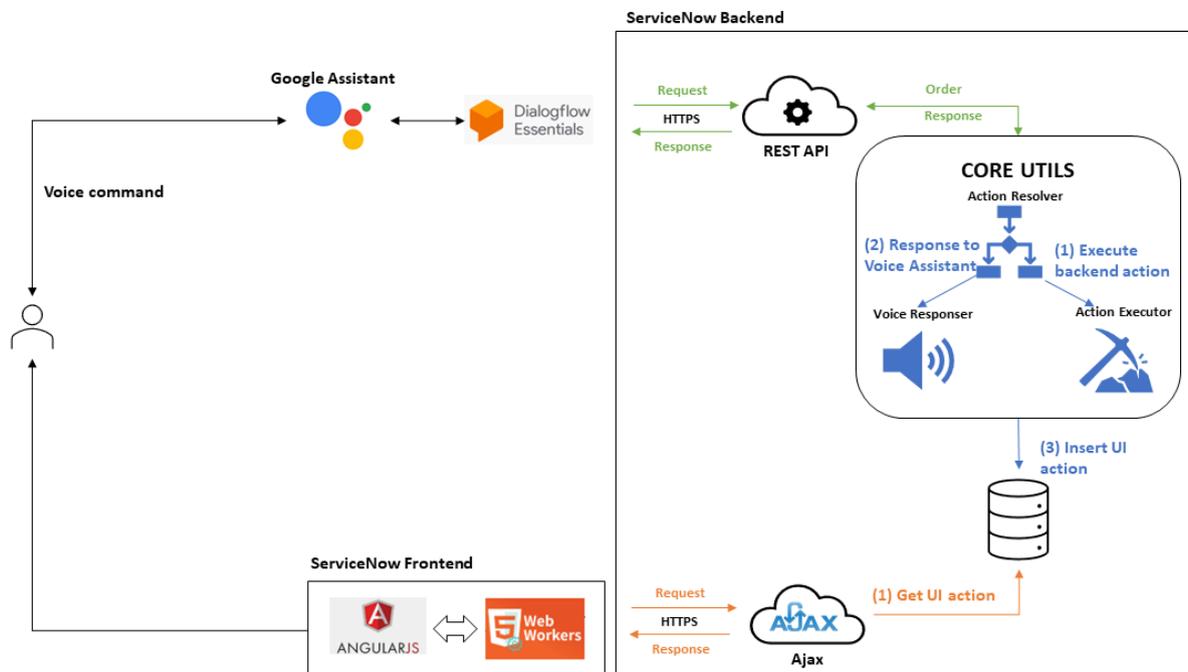


Ilustración 6 - Arquitectura global final

4.1 DIALOGFLOW ESSENTIALS

Dialogflow es una herramienta de creación de chatbots capaz de entender el lenguaje natural que Google pone a disposición para trabajar con tecnologías conversacionales[14].

En un bot se tiene típicamente una arquitectura de este tipo[15]:



Ilustración 7 - Arquitectura típica bot

La Ilustración 7 es una descripción a nivel global, las partes de las que se compone un “agente” en Dialogflow son las siguientes:

1. **Intents:** Acciones que corresponden a la lógica de negocio y que va a realizar nuestro agente.
2. **Entities:** Corresponden a las entidades de primer nivel que queremos manejar en nuestro agente. Parámetros asociados a la petición del usuario. Como, por ejemplo, el numero de una historia de usuario.
3. **Context:** Contexto asociado a la petición del usuario. El contexto puede modificar la forma en que se interpreta una petición. Por ejemplo: Si el usuario se encuentra creando una historia de usuario, una petición del tipo “Añadir comentario” tendrá que procesarse de forma que se añada el comentario a la historia en curso. Sin embargo, si esa misma frase se pronuncia cuando se está creando un “Sprint”, el comentario se añadirá al sprint en curso. Será el contexto el que determine donde se añade el comentario a partir de una misma frase.
4. **Fullfilment:** Es el apartado donde se configura el uso de webhooks y cuál es el End Point con el que Dialogflow debe comunicarse.

4.1.1 INTENTS

Los intents categorizan la intención de un usuario en un turno de la conversación. Se han definido los intents que pueden observarse en la Tabla 5, que se encargan de reconocer todas las peticiones de los usuarios relacionadas con la gestión de metodologías Agile.

Nombre del Intent	Descripción
activateSprint	Es el encargado de cambiar el estado de un sprint a Current en caso de que no haya otro en ese estado para el mismo equipo
createSprint	Es el encargado de crear un Sprint y de solicitar todos los campos obligatorios
createStory	Es el encargado de crear una Story y de solicitar todos los campos obligatorios
createTask	Es el encargado de crear una Task y de solicitar todos los campos obligatorios
Default Fallback Intent	Es el encargado de las respuestas cuando las peticiones del usuario no son entendidas o no coinciden con ningún intent
Default Welcome Intent	Es el encargado de reconocer la invocación del usuario y de iniciar el asistente de voz
deleteTask	Es el encargado de eliminar una Task. En lugar de eliminarla se cambia el estado a CANCELLED y se marca como "Inactiva".
getInfoStory	Es el encargado de mostrar el resumen de una Story (Name, Short Description and Points)
getProjectBacklog	Es el encargado de mostrar el backlog de un Project
getSprintBacklog	Es el encargado de mostrar el backlog de un Sprint
getTasks	Es el encargado de mostrar un resumen de las Tasks de una Story
includeStory	Es el encargado de incluir una Story en un Sprint
removeStory	Es el encargado de excluir una Story de un Sprint
setSprintCompleted	Es el encargado de marcar un sprint como completo, es decir, cambiar su estado a Complete
sprintCapacity	Es el encargado de editar la Capacity de un equipo para un Sprint
sprintObjectives	Es el encargado de editar los objetivos de un equipo para un Sprint
TaskDuration	Es el encargado de editar la duración de una Task
updatePoints	Es el encargado de editar los puntos de una Story
updateStory	Es el encargado de modificar los campos más relevantes de una Story
updateTask	Es el encargado de modificar los campos más relevantes de una Task

Tabla 5 - Intents

4.1.2 ENTITIES

Cada parámetro de los intents tiene un tipo, denominado entity que determina de forma exacta como se extraen los datos de una expresión de usuario final. Dialogflow proporciona entities predefinidas que pueden coincidir con muchos tipos de datos comunes. En este proyecto ha sido necesario definir los tipos de intents personalizados que pueden observarse en la Tabla 6.

Nombre de la Entity	Descripción
Number_of_hours	Estructura para referirse a una cantidad de horas
storyFields	Define los campos de una Story que es posible editar.
Task_type	Define los tipos de Task que el usuario puede introducir.
taskFields	Define los campos de una Task que es posible editar.

Tabla 6 - Entities

4.1.3 CONTEXTS

Los contextos de Dialogflow funcionan de la misma manera que el contexto en el lenguaje natural. Mediante contexts se puede controlar el flujo de una conversación haciendo que unos intents se lancen después de otros, dotando a la conversación de fluidez y naturalidad. En este proyecto ha sido necesario definir los contexts que pueden observarse en la Tabla 7.

Nombre del Context	Descripción
Default Welcome Intent - fallback	Encargado de la conversación posterior al account linking
deleteTask - followup	Encargado de la confirmación por parte del usuario antes de eliminar una Task
RemoveStory - followup	Encargado de la confirmación por parte del usuario antes de eliminar una Story de un Sprint

Tabla 7 - Contexts

4.1.4 FULFILLMENT

Usando las opciones de integración es posible emitir respuestas dinámicas y complejas habilitando el uso de webhooks. Cuando un intent con webhook activado tiene una coincidencia, envía una solicitud al end point con información sobre el intent coincidente. Es entonces cuando el sistema remoto o End Point (en este caso la URL de la instancia de ServiceNow) realiza la acción requerida y responde a DialogFlow. Esta configuración puede observarse en la Ilustración 8.

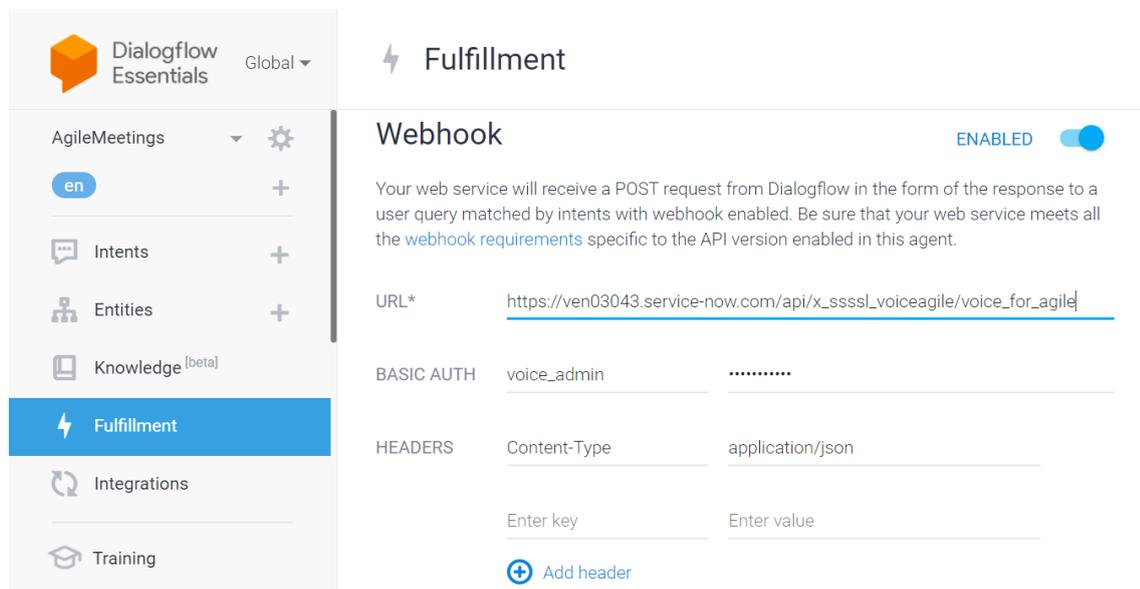


Ilustración 8 - Configuración del Fulfillment

4.2 SERVICENOW BACKEND

Las acciones que solicita el usuario desde DialogFlow provocan la creación, eliminación o actualización de registros en la instancia de ServiceNow, así como la lectura de algunos de ellos. Esto se gestiona mediante un End Point y una serie de scripts que se detallan en esta sección.

4.2.1 REMOTE END POINT – REST API

ServiceNow recibe desde Dialogflow una solicitud POST cuando una petición del usuario coincide con alguno de los intents, y es en el End Point de ServiceNow donde se recogen y analizan estas solicitudes POST.

El principal propósito del End Point es el de definir la comunicación entre Dialogflow y ServiceNow. Cuando hay una coincidencia con un intent configurado para la entrega, Dialogflow envía una solicitud POST HTTPS del webhook al servicio de webhook. El cuerpo de esta solicitud es un objeto JSON con información sobre el intent que coincide.

Este es un ejemplo de solicitud que llegaría a ServiceNow:

```
{
  "responseId": "response-id",
  "session": "projects/project-id/agent/sessions/session-id",
  "queryResult": {
    "queryText": "End-user expression",
    "parameters": {
      "param-name": "param-value"
    }
  }
}
```

```

},
"allRequiredParamsPresent": true,
"fulfillmentText": "Response configured for matched intent",
"fulfillmentMessages": [
  {
    "text": {
      "text": [
        "Response configured for matched intent"
      ]
    }
  }
],
"outputContexts": [
  {
    "name": "projects/project-id/agent/sessions/session-id/contexts/context-name",
    "lifespanCount": 5,
    "parameters": {
      "param-name": "param-value"
    }
  }
],
"intent": {
  "name": "projects/project-id/agent/intents/intent-id",
  "displayName": "matched-intent-name"
},
"intentDetectionConfidence": 1,
"diagnosticInfo": {},
"languageCode": "en"
},
"originalDetectIntentRequest": {}
}

```

Una vez ServiceNow recibe esta estructura JSON, debe decidir qué hacer y para ello en el End Point se consulta la información del JSON. En primer lugar, en base al campo **intent.displayName** se decide recibe cual ha sido el intent que ha coincidido con la petición del usuario en Dialogflow, y en baso a ello, se realiza la acción mediante llamada al CORE UTILS.

4.2.2 CORE UTILS

Está conformado por tres partes, que se encargan de gestionar las llamadas que se hacen desde la REST API. Son las encargadas de comprobar que cambios deben hacerse en la base de datos en base al JSON recibido, construir las respuestas de éxito o error para el usuario y de ejecutar los cambios en la base de datos.

4.2.2.1 ACTIONS RESOLVER

El Actions Resolver se encarga de interpretar el JSON que recibe de Dialogflow, además de la comprobación de autenticación del usuario.

La parte de la autenticación se realiza mediante el uso de un token único para cada vez que el usuario realiza el account linking y que tiene un periodo de tiempo de validez. En caso de que el usuario tenga un token válido, podrá realizar cualquiera de las acciones propias de una reunión Agile.

Una vez se recibe el JSON y se comprueba que el usuario disponga de un token válido, a través del campo intent.displayName del JSON, se decide cuál de los intents ha invocado el usuario y se prepara la llamada al VOICE RESPONSER Y AL ACTIONS EXECUTOR.

Principalmente el Script encargado de esto es VoiceAgileUtils. En la Tabla 8 pueden observarse todas las funciones del Actions Resolver.

Función	Descripcion
findIntent	Busca dentro del mensaje de Dialogflow el Intent pasado por parámetro
findParameters	Busca la variable dentro del mensaje de Dialogflow que contiene las variables del Intent
extractParameter	Extrae un campo concreto de los parámetros de un Intent
findGroupID	Busca el ID de un grupo a partir del nombre proporcionado
findUserID	Busca el ID de un usuario a partir del nombre proporcionado
findProject	Busca el ID de un proyecto a partir del nombre proporcionado
findRelease	Busca el ID de un Release a partir del nombre proporcionado
findProduct	Busca el ID de un Product a partir del nombre proporcionado
findTheme	Busca el ID de un Theme a partir del nombre proporcionado
findEpic	Busca el ID de un Epic a partir del nombre proporcionado
findStory	Busca el ID de una Story a partir del nombre proporcionado
findSprint	Devuelve los Sprint encontrados a partir del número proporcionado
findTask	Devuelve los Task encontrados a partir del número proporcionado
checkSprintDates	Comprueba que un sprint no se solape en el tiempo con ningún otro sprint
parseDate	Cambio de formatos de fecha con respecto al formato que devuelve el asistente
findDates	Recibe el Context del mensaje traído desde Dialogflow y extrae la fecha de inicio y fin de sprint

findGroupSprint	Comprueba si el grupo de un Sprint se encuentra entre una lista dada de grupos
findUserGroups	Devuelve los grupos a los que pertenece un usuario
checkCurrentSprint	Recibe el ID de un Sprint y comprueba que para el grupo al que pertenece el sprint no haya Sprints en estado "current"

Tabla 8 - Funciones del Action resolver

4.2.2.2 VOICE RESPONSER

Se encarga de confeccionar la respuesta del asistente, que debe ser enviada a Dialogflow en una estructura del tipo:

```
responseBody = {
  "payload": {
    "google": {
      "expectUserResponse": true,
      "richResponse": {
        "items": [{
          "simpleResponse": {
            "textToSpeech": message
          }
        }]
      }
    }
  }
};
```

El campo message se construye en función de si la acción se ha realizado correctamente, o si se trata de un error y por tanto es diferente en cada comunicación entre ServiceNow y Dialogflow. Se construye mediante una combinación entre el CORE UTILS y la funcionalidad de ServiceNow getMessage(). Mediante la definición de mensajes en la tabla sys_ui_message que puede observarse en la Ilustración 9, y su llamada desde el VOICE RESPONSER utilizando la funcionalidad getMessage() se construye el campo introducido en el JSON de respuesta hacia Dialogflow. En la Tabla 9 pueden observarse todas las funciones del Voice Responser.

Key key	Language language	Message message	Updated sys_updated_on	Application sys_scope
vr.create_story_answer_error_epic_multiple	English	More than one epic matches with your des...	2021-03-01 07:10:33	VoiceForAgile
vr.create_story_answer_error_multiple_epics	English	More than one epic matches with your epl...	2021-02-25 03:44:25	VoiceForAgile
vr.update_story_answer_error_epic_multiple	English	More than one epic matches with your epl...	2021-03-03 02:04:26	VoiceForAgile
vr.update_story_answer_error_assignment_...	English	More than one group matches with your gr...	2021-03-03 02:16:09	VoiceForAgile
vr.update_task_answer_error_assignment_g...	English	More than one group matches with your gr...	2021-03-03 03:07:22	VoiceForAgile
vr.create_sprint_answer_error_multiple_g...	English	More than one group matches with your gr...	2021-02-24 03:34:28	VoiceForAgile
vr.create_story_answer_error_multiple_gr...	English	More than one group matches with your gr...	2021-02-25 03:46:56	VoiceForAgile
vr.create_story_answer_error_multiple_pr...	English	More than one product matches with your ...	2021-02-25 03:35:54	VoiceForAgile

Ilustración 9_ Tabla sys_ui_message

Función	Descripción
findFieldFromArrays	Transforma un array de ID's a un array compuesto por uno de los campos del registro que identifica el ID
findNameOfStory	Encuentra el nombre de la Story que le llega como parámetro
findDescOfStory	Encuentra el nombre de la Story que le llega como parámetro
findStoryPoints	Devuelve los puntos de una Story
findTasksOfStory	Realiza el conteo de Tasks asignadas a una Story
findTotalHours	Realiza el conteo de las horas planeadas de las Tasks asignadas a una Story
findSprintStartDate	Busca la fecha de inicio de un Sprint a partir del ID del sprint
findStoriesNum	Cuenta el número de Stories asignadas a un sprint a partir del ID del Sprint
findSprintEndDate	Devuelve la fecha de fin de un Sprint a partir del ID del Sprint
findTotalPoints	Devuelve el número total de puntos de un Sprint
findTotalPercent	Devuelve el porcentaje de Stories completadas
findCompleteStories	Cuenta las Stories completas de un Sprint
findIncompleteStories	Cuenta las Stories incompletas de un Sprint
calculateAggregate	Calcula la operación indicada sobre un campo indicado de la tabla pasada por argumento

Tabla 9 - Funciones del Voice Responser

4.2.2.3 ACTION EXECUTOR

Es el encargado de realizar las acciones de borrado, creación o actualizado propias de cada intent. En base a la decisión que se toma en el ACTIONS RESOLVER haciendo uso del nombre del intent, se realizan las llamadas a los métodos necesarios que principalmente se encuentran en el script VoiceAgileUtils.

Las funciones del Action Executor pueden observarse en la Tabla 10.

Función	Descripción
createEntry	Creación de registro en una tabla
updateEntry	Actualización de registro en una tabla
checkCompleteSprint	Comprueba que el Sprint esté completo, pasándolo a ese estado en caso contrario
checkCurrentSprint	Comprueba que el grupo tenga un Sprint marcado como "Current" y, de no tenerlo, marca el enviado como actual
deleteTask	Cambia el estado de la Task indicada a "Canceled" y lo marca como inactivo
includeStory	Inserta una Story en un Sprint
createEntryInGraphQLActions	Crea una entrada en la tabla de GraphQL Actions
insertLog	Crea una entrada en el Log de la aplicación

Tabla 10 - Funciones del Action Executor

4.2.3 SERVICENOW FRONTEND

4.2.3.1 USER INTERFACE

La potencia de la solución radica en el aplicar el reconocimiento e interpretación del lenguaje natural para realizar acciones en el sistema, pero se ha desarrollado una interfaz de usuario con 2 principales objetivos:

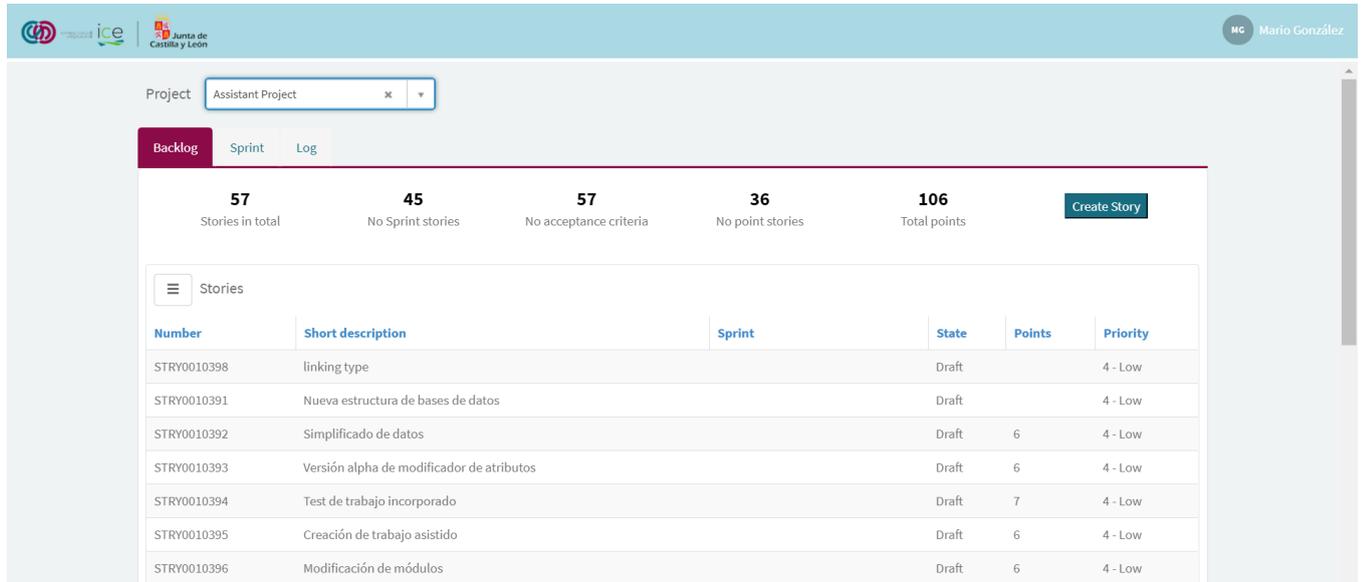
- Facilitar el seguimiento de las reuniones.
- Proporcionar una herramienta visual para poder realizar demostraciones del funcionamiento de la solución.

Se creará una única pantalla con 3 pestañas para visualizar información del Backlog y del Sprint en base a un filtro con el listado de proyectos. Las diferentes pestañas contendrán la siguiente información:

- La primera mostrará información sobre el backlog del proyecto seleccionado (Stories en total, Stories sin sprint, Stories sin Acceptance Criteria, Stories sin puntos asignados y el total de puntos de todo el Backlog). A su vez, mostrará el listado de Stories contenidas en dicho proyecto. También contará con el botón "Create Story" que nos permitirá crear una Story nueva.
- La segunda contendrá dos selectores que permitirán seleccionar el grupo y el Sprint de dicho grupo y que actuarán a su vez como filtros del listado de Stories que estén asignadas a Sprints. Desde esta misma vista, se podrá también ver información sobre el Sprint seleccionado (fecha de inicio, fecha de fin, capacidad de grupo y puntos), así como crear un Sprint desde esta misma vista mediante el botón "Create Sprint".
- Por último, la última pestaña nos mostrará un log de los últimos comandos utilizados por el usuario y de las correspondientes respuestas de voz del sistema.

Para esto, se pretende utilizar un widget que permita mostrar la información descrita (en widgets incrustados para mantener una estructura sencilla y legible) a la vez que actualizar sus datos con los nuevos cambios realizados por los usuarios mediante comandos de voz. Paralelamente, se creará otro widget que actuará como "vigilante", observando las acciones realizadas por el usuario para informar mediante la interfaz gráfica de los mismos. Sus acciones se resumirán en:

- Mensajes de éxito en caso de que la solicitud del usuario se complete correctamente.
- Mensaje de error en caso de que la solicitud del usuario no se pueda completar.
- Redirección en caso de que el usuario haya pedido la creación de un elemento y esta se haya realizado correctamente.



The screenshot shows a web application interface for a project backlog. At the top, there are logos for 'ice' and 'Junta de Castilla y León', and a user profile for 'Mario González'. Below the header, a 'Project' dropdown menu is set to 'Assistant Project'. There are three tabs: 'Backlog' (active), 'Sprint', and 'Log'. A summary bar displays the following statistics: 57 Stories in total, 45 No Sprint stories, 57 No acceptance criteria, 36 No point stories, and 106 Total points. A 'Create Story' button is visible on the right. Below the summary, a table lists stories with columns for Number, Short description, Sprint, State, Points, and Priority.

Number	Short description	Sprint	State	Points	Priority
STRY0010398	linking type		Draft		4 - Low
STRY0010391	Nueva estructura de bases de datos		Draft		4 - Low
STRY0010392	Simplificado de datos		Draft	6	4 - Low
STRY0010393	Versión alpha de modificador de atributos		Draft	6	4 - Low
STRY0010394	Test de trabajo incorporado		Draft	7	4 - Low
STRY0010395	Creación de trabajo asistido		Draft	6	4 - Low
STRY0010396	Modificación de módulos		Draft	6	4 - Low

Ilustración 10 - Pestaña de Project Backlog

La pestaña del Backlog, que puede verse en la Ilustración 10, contendrá 3 widgets incrustados en el widget de información:

- El selector de Backlog, que contendrá el título del propio selector y permitirá filtrar entre aquellos proyectos que se encuentren activos. Actuará, además, como un filtro para la información mostrada en el siguiente punto y para la lista de Stories mostrada en el cuerpo de la página. Este recibirá como argumentos el título, la tabla de ServiceNow donde buscará y el filtro que utilizará para decidir qué entradas mostrará.
- La información del proyecto, que recibirá como argumento el identificador único del propio proyecto, y cuyos datos serán calculados en el momento de ser mostrados.
- La lista de Stories del proyecto, filtrada por el proyecto del selector anterior, que mostrará información como los puntos, el grupo al que estarán asignados y el usuario que trabajará con dicha Story. Esta recibirá como argumentos la tabla y columnas a mostrar (en este caso, "rm_story"), el filtro a utilizar y el campo que utilizarán para ordenarse, así como la dirección de dicho orden.

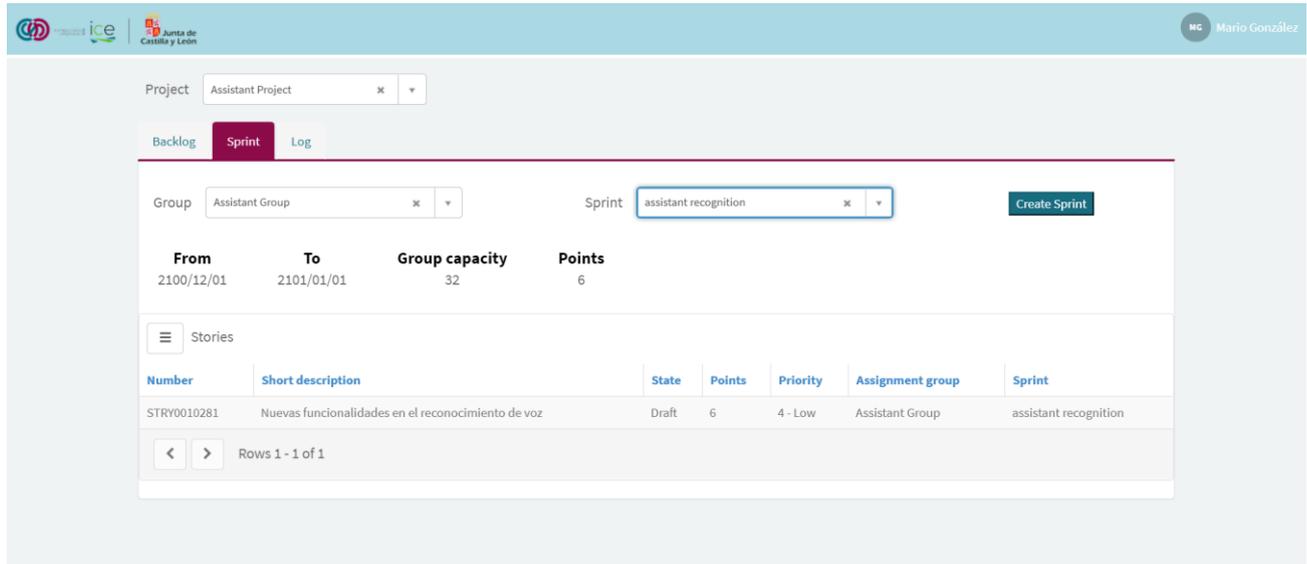


Ilustración 11 - Pestaña de Sprint

La pestaña del Sprint, que puede observarse en la Ilustración 11, contendrá otros 4 widgets incrustados en el widget de información:

- Los selectores de Group y Sprint, que serán el mismo widget que el selector de Backlog de la pestaña anterior para reutilizar código, y que permitirán filtrar la lista de Stories por dichos datos.
- La información del Sprint, que recibirá como argumento el identificador único del Sprint seleccionado, y cuyos datos serán calculados en el momento de ser mostrados.
- La lista de Stories del proyecto, que serán también el mismo widget que en el caso de la lista de Stories de la pestaña anterior (utilizando un filtro distinto) y, si hay elementos en los selectores, filtrando las Stories con estos Groups y Sprints.

En la tercera pestaña, que puede observarse en la Ilustración 12, la pestaña “Log”, habrá únicamente una tabla con las respuestas ofrecidas por el sistema de voz al usuario en el día, de manera que se pueda seguir una traza de los comandos y las acciones realizadas.

La lista total de widgets de información utilizados en la aplicación puede verse en la ilustración 12 y es:

- Widget general de pestañas: `vfa_tabbed_list`
- Widget de información de Backlog: `vfa_backlog_information`
- Widget de información de Sprint: `vfa_sprint_information`
- Widget de selector: `vfa_selector`
- Widget de tablas: `vfa_data_table_filter`



Ilustración 12 - Widgets de información incrustados

4.2.3.2 ACTIONS LISTENER

Dentro de la misma pantalla donde está contenida la información anterior se encontrará un widget (**vfa_voice_updates**) que anteriormente hemos descrito como un “vigilante” que observará las acciones realizadas por los usuarios mediante comandos de voz. Para observar dichas acciones, se utilizará la función integrada en el objeto `spUtil` de los portales de ServiceNow, que ejecuta la función especificada cuando detecta una actualización en la tabla que se le especifique.

Para que este widget pueda tener una referencia en la que basarse para observar cambios, crearemos una tabla (**x_sssl_voiceagile_graphql_actions**) en la que se insertarán todas las

acciones que deberá realizar este widget en el instante en que un usuario realice un comando. Así, cuando el usuario ordene un cambio, la orden se traducirá como una entrada en esta tabla y el widget observará este cambio, leyendo dicha entrada y mostrando por pantalla un mensaje o una ventana concreta.

Para leer las entradas de la tabla, el widget utiliza la técnica AJAX para comunicarse con el servidor de manera asíncrona. Este responderá con las entradas no procesadas pertenecientes únicamente al usuario, de modo que el cliente reduce el número de operaciones y de información a procesar.

Al recibir el script de cliente la información, realizará uno de los dos tipos de acciones:

- Mensaje de información/error: los datos traídos desde servidor contendrán el mensaje que debe mostrarse en el portal, así como si es un error o es información para el usuario.
- Redirección: los datos en este caso contendrán la información suficiente como para montar la URL al objeto que se pretende mostrar (tabla e identificador único).

4.3 DATA MODEL

El listado completo de las tablas se muestra en la Tabla 11.

Tabla	Name	Principal contenido
Action History	x_ssssl_voiceagile_action_history	Log de acciones solicitadas por los usuarios junto con la acción resuelta
GraphQL Actions	x_ssssl_voiceagile_graphql_actions	Tabla de eventos, tanto a realizar como realizados, para la interfaz gráfica

Tabla 11 - Tablas de la aplicación

Estas tablas no tienen relación entre sí y su utilidad es exclusivamente para dar apoyo al proceso, siendo la primera (Action History) para mostrar al usuario un historial de sus comandos lanzados y la segunda (GraphQL Actions) para indicar al sistema los mensajes o redirecciones que debe realizar. A continuación, se muestran los campos contenidos en las dos diferentes tablas. En la Tabla 12, los campos de la tabla Action History, y en la Tabla 13 los campos de la tabla GraphQL Actions.

Etiqueta del campo	Nombre del campo	Tipo	Descripción
Action	action	Choice	Define el tipo de acción realizada por el usuario (created, read, updated, deleted)
Answer	answer	String	Muestra la respuesta de voz lanzada por el sistema al usuario
Interpreted command	command	String	Muestra el comando ordenado por el usuario
Object	object	String	Contiene la tabla sobre la que recae la acción
Object field	object_field	String	Contiene el campo sobre el que recae la acción (en caso de recaer sobre un campo)
Object ID	object_id	String	Contiene el identificador único del elemento sobre el que recae la acción

Tabla 12 - Campos de la tabla Actions History

Etiqueta del campo	Nombre del campo	Tipo	Descripción
Action type	action_type	Choice	Define las acciones que puede realizar el Action Listener del Portal
Context	context	String	Contiene el contexto de la conversación Sistema-Usuario
Message	message	String	Contiene, de ser necesario, el mensaje a mostrar por el sistema en el Portal
Object	object	String	Contiene, de ser necesario, el objeto a mostrar en el portal
Object ID	object_id	String	Contiene, de ser necesario, el elemento a mostrar en el portal
Token	token	String	Contiene el token utilizado por la comunicación entre ServiceNow y Dialogflow

Tabla 13 - Campos de la tabla GraphQL Actions

4.4 SEGURIDAD Y AUTENTICACIÓN

El asistente necesita de un sistema de autenticación ya que es necesario disponer de una cuenta en la instancia de ServiceNow para poder realizar modificaciones. Por ello, se ha utilizado un sistema de autenticación basado en OAuth que exige que el usuario introduzca sus credenciales en la instancia de ServiceNow antes de poder realizar modificaciones en esta [16].

En primer lugar, se ha realizado la configuración en el apartado Account Linking del Actions Console que puede verse en la Ilustración 13.

Actions Console Overview **Develop** Test Deploy Analytics agilemeetings-nhse

Account linking

Linking type

OAuth

Implicit

OAuth Client Information

Client ID issued by your Actions to Google ⓘ

46f893c7d2ee6450629b9ab4df3bb989

Authorization URL ⓘ

https://ven03043.service-now.com/oauth_auth.do

Ilustración 13 - Configuración de la autenticación

Por otra parte, el hecho de autenticarse provoca la generación de un “token” único para cada sesión y que tiene una validez temporal, configurable en la tabla `oauth_entity`, que puede observarse en la Ilustración 14.

* Name | name VoiceRecognition

* Client ID | client_id 46f893c7d2ee6450629b9ab4df3bb989

Client Secret | client_secret *****

Redirect URL | redirect_url https://oauth-redirect.googleusercontent.com/r/agilemeetin-gs-nhse

Logo URL | logo_url

Comments | comments OAuth for Voice Recognition Project

Application | sys_scope VoiceForAgile ⓘ

Accessible from | access All application scopes | public

Active | active

* Refresh Token Lifespan | refresh_token_lifespan 8,640,000

* Access Token Lifespan | access_token_lifespan 5,400

Ilustración 14 - Configuración de la entidad OAuth

Este token queda registrado en la tabla `oauth_credential` de ServiceNow. En caso de que un usuario inicie sesión y su token no haya expirado podrá acceder al asistente sin necesidad de introducir sus credenciales de nuevo, quedando todas las acciones de este identificadas por dicho token.

Dado que a la hora de usar el asistente se emiten mensajes de confirmación/error en la interfaz de usuario, es necesario que cada usuario pueda ver solo las notificaciones asociadas a sus acciones, y esto se consigue mediante el uso del token. Antes de mostrar una notificación se comprueba que el token de usuario asociado a la acción sea el mismo que el del usuario, y si estos coinciden, se muestra la notificación.

Capítulo 5

MANUAL DE USUARIO

Una vez está desarrollado el agente, es necesario lanzar una versión Alpha, dirigida a usuarios de testeo, que comprueban que el asistente funciona correctamente y que no incumple las normas de desarrollo de Google. Una vez lanzado y configurados los usuarios de testing, los usuarios cuyas cuentas de Google aparezcan en la sección de usuarios de testeo podrán utilizar la aplicación de la forma que se describe en esta sección [17].

5.1 INICIAR EL ASISTENTE DE VOZ Y LA INTERFAZ DE SERVICENOW

Para iniciar la iniciar de ServiceNow es necesario disponer de una cuenta en la instancia. Una vez el usuario haya introducido sus credenciales, para iniciar el portal basta con introducir en el Navigation Filter “Voice for Agile” y aparecerá el Voice for Agile Portal, como puede verse en la Ilustración 15.

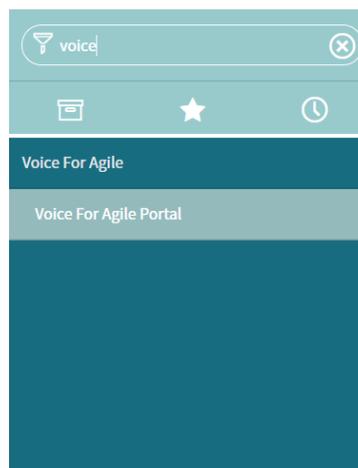


Ilustración 15 - Selección del portal

Haciendo click sobre la opción “Voice for Agile Portal” se abrirá automáticamente la interfaz de usuario en ServiceNow.

En cuanto al asistente, basta con iniciar el asistente de voz de Google (teniendo el inglés como el idioma predeterminado del asistente) y ejecutar la invocación mediante la frase “Talk to Stormy Assistant”. Una vez invocado el asistente se abre en pantalla completa y solicita al usuario que realice la autenticación en la instancia de ServiceNow, como puede observarse en la Ilustración 16. Al pulsar la opción “Yes” el usuario será redirigido a la página de autenticación.

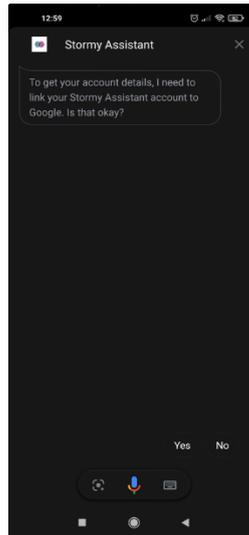


Ilustración 16 - Invocación del asistente

Primero el usuario deberá introducir sus credenciales en la instancia de ServiceNow, como se observa en la Ilustración 17, y tras ello, el sistema solicitará permiso para permitir la conexión entre el asistente de voz y la instancia de ServiceNow. Al aceptarlo, el usuario podrá comenzar a utilizar el asistente.

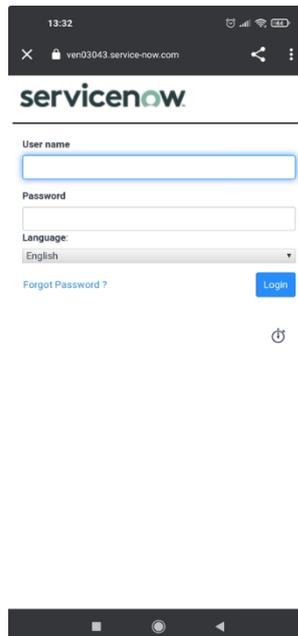


Ilustración 17 - Autenticación en ServiceNow

5.2 INVOCACIÓN DE LOS INTENTS

A partir de este punto, el usuario puede comenzar a utilizar el asistente para gestionar una reunión Agile. Por ejemplo, una acción típica de una reunión Agile podría ser la creación de un sprint. Para ello el usuario debería solicitar al asistente la creación de un sprint con la frase “*create a Sprint*” o similares, pudiendo incluir en la petición parámetros obligatorios para un sprint. Si el usuario proporciona estos datos y son correctos, el sprint se creará. Si por el contrario el usuario limita su petición a solicitar la creación de un sprint, el asistente ira pidiendo al usuario todos los campos obligatorios como puede observarse en la Ilustración 18.

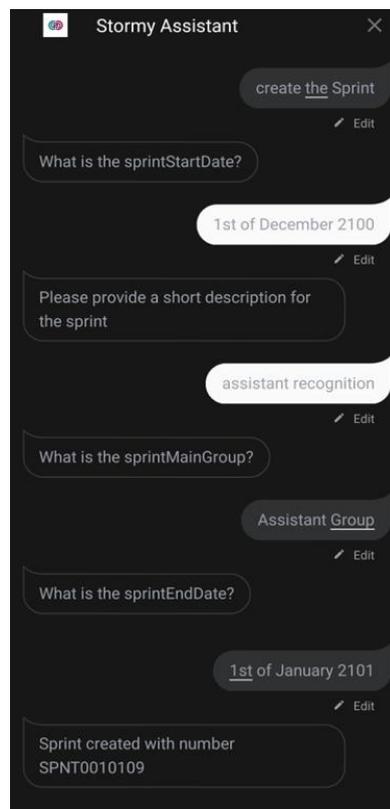


Ilustración 18 - Creación de un sprint

Al crearse correctamente, el asistente emite un mensaje de éxito y el registro del nuevo sprint se abre automáticamente en la interfaz de usuario, pudiendo modificarse o rellenar cualquier campo en caso de ser necesario. Además, el nuevo sprint aparecerá en el elegible de la pestaña Sprint de la interfaz de usuario, para listar las stories asignadas a dicho sprint, como se muestra en la Ilustración 19. También existe la posibilidad de crear un nuevo Sprint desde el botón “Create Sprint”.

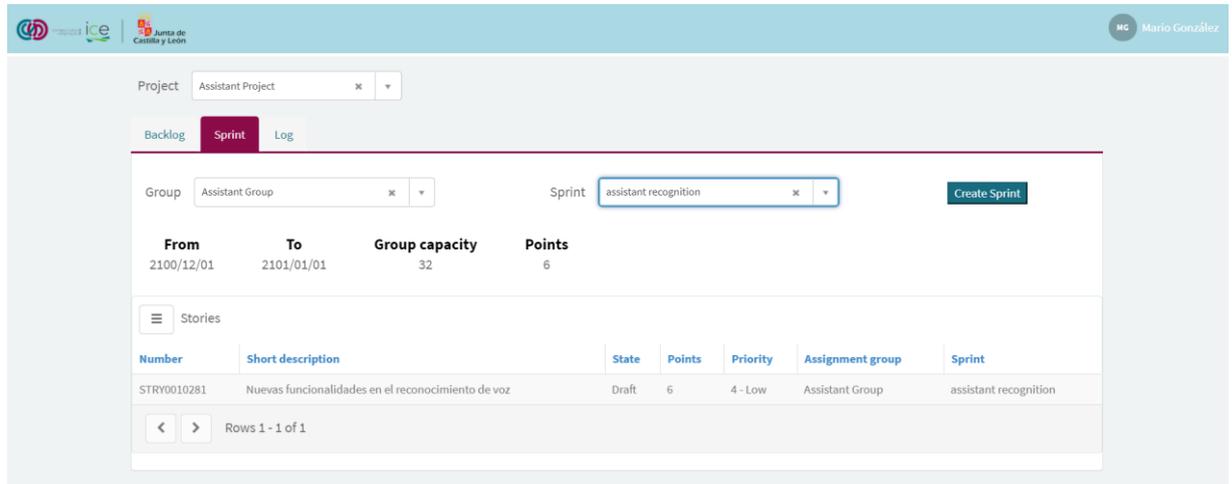


Ilustración 19 - Pantalla del nuevo sprint creado

De igual manera se procedería para cualquiera de las funcionalidades. Por ejemplo, para obtener el backlog de un project, sería necesario ordenarle al asistente “*get información about Project XYZ*”. En el caso del ejemplo de la Ilustración 20, puede observarse el backlog del proyecto “*assistant project*”.

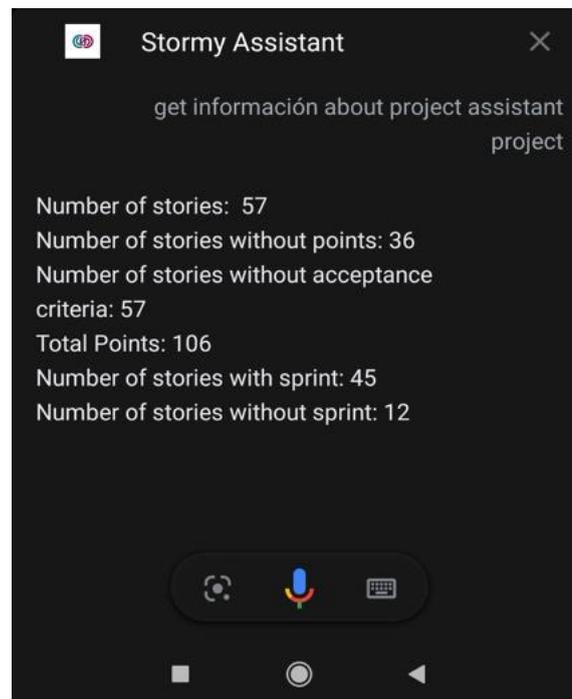
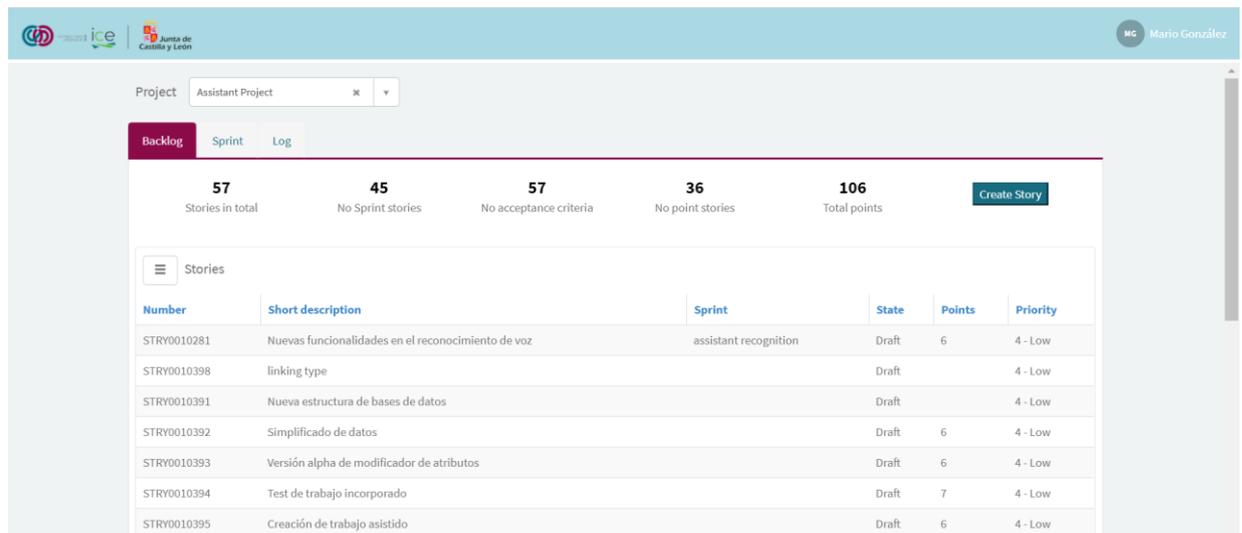


Ilustración 20 - Obtención del Project backlog

De igual manera, en la interfaz de usuario puede obtenerse el backlog del proyecto en la pestaña “Backlog” y filtrando por el Project deseado en el filtro habilitado para ello. Como se observa en la Ilustración 21, la información ofrecida por el asistente y por la interfaz coincide.



The screenshot shows the 'Project Backlog' interface for the 'Assistant Project'. At the top, there are navigation tabs for 'Backlog', 'Sprint', and 'Log'. Below these, summary statistics are displayed: 57 Stories in total, 45 No Sprint stories, 57 No acceptance criteria, 36 No point stories, and 106 Total points. A 'Create Story' button is visible on the right. The main area contains a table of stories with the following data:

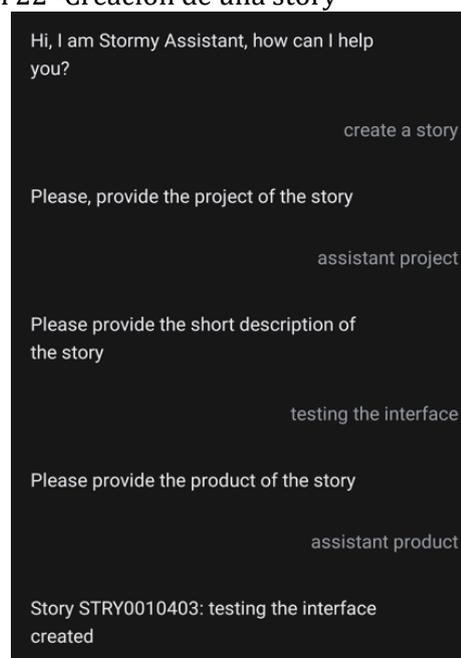
Number	Short description	Sprint	State	Points	Priority
STRY0010281	Nuevas funcionalidades en el reconocimiento de voz	assistant recognition	Draft	6	4 - Low
STRY0010398	linking type		Draft		4 - Low
STRY0010391	Nueva estructura de bases de datos		Draft		4 - Low
STRY0010392	Simplificado de datos		Draft	6	4 - Low
STRY0010393	Versión alpha de modificador de atributos		Draft	6	4 - Low
STRY0010394	Test de trabajo incorporado		Draft	7	4 - Low
STRY0010395	Creación de trabajo asistido		Draft	6	4 - Low

Ilustración 21 - Interfaz del Project backlog

Siguiendo con el curso de una reunión, el siguiente paso podría ser crear una story. Para ello es necesario ordenar al asistente que realice dicha acción con una frase del estilo “*create a story*”. En caso de proporcionar algún campo, el asistente lo identificará y si no se proporciona algún campo obligatorio, el asistente lo solicitará.

Si alguno de los campos introducidos no tiene un campo válido, tanto el asistente como la interfaz lo notificarán.

Una vez se hayan introducido todos los datos, la story se crea y el asistente lo notifica como puede verse en la Ilustración 22. Ilustración 22 - Creación de una story



The screenshot shows a chat interface with a dark background. The conversation is as follows:

- Assistant: Hi, I am Stormy Assistant, how can I help you?
- User: create a story
- Assistant: Please, provide the project of the story
- User: assistant project
- Assistant: Please provide the short description of the story
- User: testing the interface
- Assistant: Please provide the product of the story
- User: assistant product
- Assistant: Story STRY0010403: testing the interface created

Ilustración 22 - Creación de una story

Además, en la interfaz se abrirá el registro de la nueva story y aparecerá la nueva story en el listado del backlog del proyecto al que ha sido asignada, en este caso, el *assistant Project*, como puede verse en la Ilustración 23. Además, la acción quedará registrada en la tabla de Logs de la interfaz, siendo visible solo para el propio usuario debido al uso de token explicado en el apartado SEGURIDAD Y AUTENTICACIÓN. Estos logs pueden verse en la Ilustración 24 donde se ve reflejada la creación de la story.

Number	Short description	Sprint	State	Points	Priority
STRY0010403	testing the interface		Draft		4 - Low
STRY0010402	Creación de una story	assistant recognition	Draft	8	4 - Low
STRY0010281	Nuevas funcionalidades en el reconocimiento de voz	assistant recognition	Draft	6	4 - Low
STRY0010398	linking type		Draft		4 - Low
STRY0010391	Nueva estructura de bases de datos		Draft		4 - Low

Ilustración 23 - Nueva story creada

Interpreted command	Answer
assistant product	Story STRY0010403: testing the interface created
8	Story points for Story STRY0010402 updated to 8

Ilustración 24 - Pestaña de Logs

El resto de las funcionalidades se invocan de la misma manera, sin tener una expresión obligatoria para que se lance la acción, si no que a través del *machine learning* de Dialogflow, gracias a las frases de entrenamiento introducidas, se generan variantes de estas para que expresiones similares se identifiquen como peticiones.

CONCLUSIONES Y LÍNEAS FUTURAS

6.1 CONCLUSIONES

En este documento se ha explicado el desarrollo de un asistente de voz capaz de gestionar las reuniones Agile propias de un equipo de trabajo, agilizando estas reuniones y consiguiendo que los equipos puedan trabajar más eficientemente.

Tras estudiar una arquitectura viable y los posibles asistentes del mercado, se eligió Google Assistant como asistente para este proyecto dada su versatilidad, el amplio número de usuarios de Google y su fiabilidad en cuanto a reconocimiento de peticiones. Al tratarse de la gestión a través del uso de la voz de una base de datos alojada en una instancia de ServiceNow cuya integridad es de importancia para la empresa, el hecho de que este asistente tenga un alto porcentaje de acierto en las peticiones es clave, ya que, si el usuario realiza una petición y esta es interpretada incorrectamente, puede producir cambios en la base de datos no deseados.

Una vez que se entrenó al asistente de voz introduciendo frases de entrenamiento que el usuario pudiera utilizar para realizar sus peticiones y se programó el End Point en ServiceNow para gestionar esas peticiones, el usuario pudo ejecutar el asistente realizando todas las peticiones que se han detallado en las secciones anteriores.

A través del uso de este asistente se ha conseguido gestionar reuniones Agile reales de un grupo de trabajo y se han observado los siguientes beneficios.

- Reducción del tiempo de la reunión debido a la no necesidad de acceder a cada registro propio de la reunión. Ahora las ordenes por voz se ejecutan directamente sin hacer uso de los registros de ServiceNow.
- Reducción del contacto con los equipos personales debido a que la tarea de liderar la reunión realizada antes mediante el uso de una computadora por uno de los miembros del equipo es ahora realizada por voz. Esto puede ayudar a evitar contagios por contacto dada la actual situación sanitaria.
- El hecho de que ya no sea necesario que una persona lidere la reunión y vaya actualizando, borrando, o creando registros, hace que esa persona pueda centrar su atención en la reunión, así como realizar otras tareas en paralelo.
- El hecho de que no sea necesario utilizar una computadora para gestionar la reunión y que estas acciones puedan ser realizadas por órdenes de voz, facilita la inclusión en el equipo de personas con algún tipo de discapacidad que pueda impedir el uso de teclados.

6.2 LIMITACIONES Y LÍNEAS FUTURAS

Una de las partes más importantes a la hora de desarrollar un asistente de voz , en este caso para la reunión de reuniones Agile, es la elección del propio asistente. El hecho de haber utilizado Google Assistant ha sido de gran ayuda dado su entorno de desarrollo, Dialogflow. Sin embargo, esta funcionalidad podría extenderse a multitud de asistentes diferentes para que así pudiera ser utilizada desde distintos dispositivos.

Sería de especial interés hacer que el código que se encarga de los cambios en ServiceNow fuera capaz de identificar que asistente de voz se está utilizando y fuese válido para cualquiera de ellos. Otra posible línea futura sería la de expandir el asistente a otros idiomas, como el francés o el español para que diversas empresas de habla no inglesa pudieran utilizarlo de manera más intuitiva en su lengua natal.

Actualmente son veinte las funcionalidades que el asistente cumple, aunque existen algunas funcionalidades adicionales que se podrían cubrir dentro de la gestión de reuniones ágiles, como la posibilidad de crear reuniones de seguimiento y añadir participantes, aunque esto supondría la integración con herramientas como Teams o Skype.

En definitiva, dado el gran porcentaje de aciertos del asistente y de las ventajas que su uso ofrece para el equipo de desarrollo, sería de interés extender sus funcionalidades y los idiomas en los que está disponible para así conseguir llegar a un número de usuarios mayor

BIBLIOGRAFÍA

- [1] «¿Qué es la metodología Agile y qué beneficios tiene para tu empresa?» <https://www.wearemarketing.com/es/blog/que-es-la-metodologia-agile-y-que-beneficios-tiene-para-tu-empresa.html> (accedido feb. 24, 2021).
- [2] S. Denning, «Why Agile Is Eating The World», Forbes. <https://www.forbes.com/sites/stevedenning/2018/01/02/why-agile-is-eating-the-world/> (accedido feb. 24, 2021).
- [3] «pulse-of-the-profession-2017.pdf». Accedido: feb. 24, 2021. [En línea]. Disponible en: <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf>.
- [4] «Nuevos datos reafirman el rol de los directores de proyectos en futuro laboral», feb. 11, 2020. <https://www.businesswire.com/news/home/20200211006086/es/> (accedido feb. 24, 2021).
- [5] «Qué es SCRUM», Proyectos Ágiles, ago. 04, 2008. <https://proyectosagiles.org/que-es-scrum/> (accedido mar. 15, 2021).
- [6] «2021 BÚSQUEDA POR VOZ EN GOOGLE Y SU IMPORTANCIA EN 2020». <https://www.rankingonline.es/busqueda-por-voz-y-su-importancia-en-2020/> (accedido abr. 14, 2021).
- [7] «Datos y estadísticas sobre el crecimiento del mercado de asistentes de voz», Roberto Carreras, ene. 31, 2019. <https://robertocarreras.es/datos-y-estadisticas-sobre-el-crecimiento-del-mercado-de-asistentes-de-voz/> (accedido abr. 14, 2021).
- [8] «Speech and Voice Recognition Market – Global Industry Trends and Forecast to 2027 | Data Bridge Market Research». <https://www.databridgemarketresearch.com/reports/global-speech-voice-recognition-market#> (accedido abr. 14, 2021).
- [9] T. Cuenca, «Alexa, Google Assistant y Siri. Los mejores asistentes de voz hoy en día», Domoticada, may 15, 2020. <https://www.domoticada.com/alex-google-assistant-siri-asistentes-de-voz/> (accedido abr. 14, 2021).
- [10] J. Pollard, «Google Assistant vs Alexa Development», Medium, may 30, 2018. <https://medium.com/@yeltzland/google-assistant-vs-alex-development-94340a3505> (accedido abr. 15, 2021).
- [11] «Annual Smart Speaker IQ Test», Loup Ventures, dic. 20, 2018. <https://loupventures.com/annual-smart-speaker-iq-test/> (accedido abr. 15, 2021).
- [12] «AWS Lex vs Google DialogFlow», Technosoft Solutions, jul. 03, 2019. <https://technosoft.com/aws-lex-vs-google-dialogflow.html> (accedido abr. 15, 2021).
- [13] «Dialogflow vs Lex vs Watson vs Wit vs Azure Bot | What to Choose?», Kommunicate Blog, may 10, 2019. <https://www.kommunicate.io/blog/dialogflow-vs-lex-vs-watson-vs-wit-vs-azure-bot/> (accedido abr. 15, 2021).
- [14] P. Durán, «Qué es DialogFlow y cuáles son sus nuevas actualizaciones en beta», Medium, feb.

14, 2020. <https://planetachatbot.com/que-es-dialogflow-y-cuales-son-actualizaciones-en-beta-37f7ac827aec> (accedido abr. 15, 2021).

[15] «Despliega tu chatbot en una hora con Dialogflow». <https://www.paradigmadigital.com/dev/despliega-tu-chatbot-en-una-hora-con-dialogflow/> (accedido abr. 15, 2021).

[16] «Account linking with OAuth | Actions on Google account linking». <https://developers.google.com/assistant/identity/oauth2> (accedido abr. 19, 2021).

[17] «Policies for Actions on Google | Actions console», Google Developers. <https://developers.google.com/assistant/console/policies/general-policies> (accedido abr. 19, 2021).