



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y Naturales y Agrimensura
Especialización en Tecnologías de la Información

Trabajo Final

**“Diseño de Interfaz Conversacional para
búsquedas de legajos judiciales mediante
comandos de voz.”**

Autor: Beltrán Javier Alejandro

Director: Dr. Emanuel Irrrazabal

Año 2024

Resumen

En este trabajo se analiza, desarrolla, prueba y pone en funcionamiento una aplicación de interfaz conversacional integrado al sistema de gestión de legajos denominado CRIMINIS perteneciente al Poder Judicial de la Provincia de Corrientes, con el objetivo de modernizar las interfaces hacia los usuarios y extender las prestaciones funcionales del sistema, como también una mejor cobertura desde el punto de vista de la accesibilidad de las personas. Se indaga también en la versatilidad que las interfaces conversacionales ofrecen y los posibles escenarios en los que puedan resultar útiles. Su implementación resulta satisfactoria para los usuarios al proporcionar interfaces amigables e intuitivas en la interacción con el sistema para acceder a la información. Se disponen las especificaciones y comparativas necesarias de las técnicas de diseño, desarrollo de un prototipo; y finalmente un estudio de las prestaciones y performance que permitan evaluar su aplicación.

Palabras clave: interfaz conversacional; legajos judiciales; lenguaje natural; accesibilidad.

Agradecimientos

A cada uno de los docentes y compañeros del curso de Maestría. En especial al Dr. Emanuel Irrazábal, que por su trabajo intangible en el seguimiento personal y su influencia que ha generado la motivación a la elaboración de este trabajo.

Índice de Contenidos

Resumen	2
Agradecimientos	3
Índice de Contenidos	4
Índice de figuras	7
Índice de tablas	8
Glosario	9
Lista de acrónimos	10
1. Introducción	11
1.1 Fundamentación	11
1.1.1 Ventajas significativas en comparación a la usabilidad tradicional	12
1.1.2 Aplicaciones generales del procesamiento de lenguaje natural	14
1.2 Objetivo general	15
1.3 Objetivos Específicos	16
2. Marco teórico	17
2.1 Metodología genérica empleada en reconocimiento de voz	17
2.1.1 El Corpus de Voces	17
2.1.2 Preprocesamiento de la señal de voz	18
2.1.2.1 Ventaneo	19
2.1.3 Detección de inicio y fin de palabras	19
2.1.4 Parámetros característicos de una señal de voz	20
2.1.4.1 Cepstrum	20
2.1.4.2 LPC	22
2.1.4.3 MFCC	23
2.2 Interfaz de Voz y Procesamiento de Lenguaje Natural	23
2.2.1 Esquema general de una interfaz de Lengua Natural	23
2.2.2 Arquitectura típica de una NLI que utiliza un lenguaje de representación intermedio	24
2.3 Análisis de Herramientas	27
2.4 Arquitectura de Aplicaciones Web	29
2.4.1 Flujo de diálogo en el Proceso de Búsqueda	29
2.4.2 Agente de Dialogflow	29
2.5 Conocimiento disciplinar específico	31
2.5.1 Diseño de Aplicaciones	31
2.5.2 Cloud Computing	31
2.5.3 Auditoría	31
3. Planificación y Desarrollo del Diseño	32
3.1 Definición del Problema	32
3.1.1 Recolección de Datos	32
3.1.2 Análisis de la Situación Actual	33
3.2 Etapas de Desarrollo	34
3.2.1 Diagrama de Componentes	34

3.3 Conversación entre agente y usuario	34
3.4 Diseño de la Interfaz	35
3.4.1 Proceso de Búsqueda de Legajos	36
3.4.1.1 Diagrama de Flujo de Búsqueda de Legajos	36
3.4.1.2 Presentación de la Interfaz	37
3.4.1.3 Diálogo de la Orden	38
3.4.1.4 Selección de Filtros	39
3.4.1.5 Interfaz de Resultados	39
3.4.2 Acción sobre los resultados obtenidos	40
3.4.3 Análisis de Soluciones Propuestas	42
3.5 Metodología e Iteraciones	42
3.5.1 Selección de la Metodología	43
3.5.2 Iteraciones	43
3.5.2.1 Iteración 1: Planificación y Diseño del Desarrollo	44
3.5.2.2 Iteración 2: Mejora y Expansión	44
3.6 Desarrollo del Prototipo	45
3.6.1 Configuración del Entorno de Desarrollo	45
3.6.2 Implementación de la Interfaz Conversacional	46
3.6.3 Requerimientos Técnicos y Compatibilidades	49
3.6.4 Despliegue y Pruebas	49
3.6.5 Documentación y Mantenimiento	50
3.6.5.1 Documentación del Proyecto	50
3.6.5.2 Actualizaciones y Mejoras Continuas	51
4. Integraciones	53
4.1 Identificación módulos factibles de aplicación	53
4.2 El Agente conversacional	54
4.3 Aplicación	54
4.3.1 Herramientas utilizadas	54
4.3.2 Presentación de la aplicación.	57
4.3.3 Pantalla de conversación.	57
5. Optimización y pruebas	60
5.1 Caso de Uso. Inicializar Interfaz Conversacional	60
5.1.1 Caso de Uso: Seleccionar Filtro	61
5.1 Pruebas Generales de la Aplicación General	62
5.1.1 Herramientas Utilizadas para las Pruebas	62
5.1.2 Pruebas Unitarias y de Integración	62
5.1.3 Optimización y Mejoras Implementadas	63
5.1.4 Conclusiones de las Pruebas	63
5.2 Pruebas Generales de la Interfaz Conversacional	64
5.2.1 Herramientas Utilizadas en las Pruebas de Comandos de Voz	64
5.2.2 Pruebas Específicas sobre el Uso de Comandos de Voz	64
5.2.3 Optimización de la Funcionalidad de Comandos de Voz	67
5.2.4 Análisis de las Pruebas de Comandos de Voz	67

6. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO	69
6.1 Conclusiones sobre la metodología	69
6.2 Conclusiones sobre la tecnología	70
6.3 Conclusiones sobre las herramientas	70
6.4 Conclusiones sobre el trabajo	71
6.5 Futuras líneas de trabajo	71
Referencias	73

Índice de figuras

Capítulo 2. Marco teórico	17
Fig 2.1 Diagrama de bloques de un Sistema de reconocimiento.	17
Fig 2.2 Ejemplo de grabaciones con amplitud: saturada, media y baja.	18
Fig 2.3 Ecuación en diferencias de filtro paso-altas.	18
Fig 2.4 Transformada Z de la Ecuación 1.	19
Fig 2.5 Inicio y fin de palabra con cálculo de cruce de ceros y energía.	20
Fig 2.6 Cálculo de los coeficientes cepstrales (Cepstrum).	21
Fig 2.7 Gráficas resultantes en el flujo de cálculo del Cepstrum.	21
Fig 2.8 Comparativa de respuestas de frecuencias LPC	22
Fig 2.9 Esquema general de una interfaz de Lengua Natural	23
Fig 2.10 Esquema Arquitectura típica de una NLI	24
Fig 2.11 Arquitectura típica de una NLI	25
Fig 2.12 Representación de una Orden	26
Fig 2.13 Ejemplo de una Representación de Orden	26
Capítulo 3. Planificación y Desarrollo	32
Fig 3.1 Descripción de las funcionalidades básicas	32
Fig 3.2 Análisis de la Situación Actual	33
Fig 3.3 Análisis de puntos de acción débiles en el uso tradicional	34
Fig 3.4 Diagrama de Componentes en el módulo	34
Fig 3.5 Iconos de Estados del componente	35
Fig 3.6 Flujo de conversación entre el agente y el usuario	36
Fig 3.7 Diagrama de Flujo de Búsqueda de Legajos	37
Fig 3.8 Icono de Introducción de Voz	38
Fig 3.9 Diseño de Interfaz. Diálogo de de la Orden	39
Fig 3.10 Diseño de Interfaz. Diálogo de filtrado de la Orden	39
Fig 3.11 Diseño de Interfaz. Listado de Resultados obtenidos	40
Fig 3.12 Diseño de Interfaz. Flujo de redirección	41
Fig 3.13 Vista de Detalle de Legajo Judicial	41
Capítulo 4. Integraciones	53
Fig. 4.1 Prototipo: presentación de la aplicación inicio	56
Fig. 4.2 Prototipo: integración con el estado de legajos	58
Fig. 4.3 Prototipo: integración con diálogo de interfaz conversacional	58
Fig. 4.4 Prototipo: integración Pantalla de redirección	59
Capítulo 5. Optimización y pruebas	60
Fig. 5.1 Diagrama de Secuencia del Caso de Uso de Inicialización	60
Fig. 5.2 prueba específica sobre el uso de comandos de voz	65
Fig. 5.3 comparativa de las pruebas específicas	67
Fig. 5.4 Resultados de Test de Performance de Lighthouse	68

Índice de tablas

Capítulo 1. Introducción	11
Tabla 1.1 Comparación: Antes y Después de la Implementación esperada	12
Capítulo 2. Marco teórico	17
Tabla 2.1 Esquema general de una interfaz de Lengua Natural	19
Tabla 2.3 Comparativa de Herramientas de TTS y STT	27
Capítulo 3. Planificación y Desarrollo	32
Tabla 3.1 Tabla de Métricas de Viabilidad	42
Tabla 3.3 Requerimientos Mínimos de Aplicación	49
Capítulo 5. Planificación y Desarrollo	60
Tabla 5.1 Pruebas Unitarias para la Lógica de Negocio	62
Tabla 5.2: Pruebas Unitarias obtenidos desde Chrome Lighthouse	63
Tabla 5.3 Pruebas de Precisión en Reconocimiento de Voz	64
Tabla 5.4 Tiempo de Respuesta a Comandos de Voz	66

Glosario

Accesibilidad: Facilidad con que personas de todas las capacidades pueden utilizar un sistema o dispositivo. En el contexto de este trabajo, se refiere a la capacidad de acceder y usar la interfaz conversacional para personas con discapacidad.

Agente Conversacional: Programa informático diseñado para simular una conversación con usuarios humanos, especialmente a través de internet, utilizando el procesamiento de lenguaje natural.

Arquitectura de Aplicaciones Web: Estructura de las diferentes capas que componen una aplicación web, incluyendo la interfaz de usuario, lógica de negocio, y acceso a datos, entre otros.

Cloud Computing (Computación en la Nube): Modelo de computación basado en internet que permite el acceso remoto a software, almacenamiento de archivos y procesamiento de datos a través de servicios en la nube.

CRIMINIS: Sistema web de gestión de legajos judiciales, perteneciente al Poder Judicial de la Provincia de Corrientes.

Dialogflow: Plataforma de Google que permite a los usuarios diseñar e integrar un agente conversacional o chatbot en aplicaciones móviles, sitios web, plataformas de mensajería y dispositivos IoT, utilizando procesamiento de lenguaje natural.

Interfaz Conversacional: Sistema de interacción hombre-máquina que utiliza el lenguaje natural para comunicarse con el usuario, permitiendo que este realice consultas y reciba respuestas como si estuviera conversando con otro ser humano.

Lenguaje Natural: Lenguaje hablado o escrito por humanos para la comunicación cotidiana, en contraste con los lenguajes de programación o lenguajes formales.

Procesamiento de Lenguaje Natural (PLN): Rama de la inteligencia artificial que se enfoca en la interacción entre computadoras y humanos a través del lenguaje natural.

Prototipo: Versión preliminar de un producto que se desarrolla para demostrar conceptos, probar diseños, explorar opciones y obtener retroalimentación antes de la producción final.

Reconocimiento de voz: Tecnología que permite a los dispositivos informáticos identificar y procesar la voz humana como entrada.

Síntesis de Voz (Text-to-Speech, TTS): Tecnología que convierte texto digital en habla audible, permitiendo a las computadoras y otros dispositivos leer textos en voz alta.

Web Speech API: Interfaz de programación de aplicaciones proporcionada por el navegador web que permite a los desarrolladores incorporar funciones de reconocimiento de voz (Speech-to-Text) y síntesis de voz (Text-to-Speech) en sus aplicaciones web.

Lista de acrónimos

- PLN: Procesamiento de Lenguaje Natural
- API: Interfaz de Programación de Aplicaciones
- ILN: Interfaz de Lenguaje Natural
- NLI: Interfaz de Lenguaje Natural (por sus siglas en inglés, Natural Language Interface)
- UI: Interfaz de Usuario
- TTS: Text-to-Speech (Texto a Voz)
- GCP: Google Cloud Platform
- JSON: JavaScript Object Notation
- SQL: Structured Query Language (Lenguaje de Consulta Estructurada)
- OLAP: Online Analytical Processing (Procesamiento Analítico en Línea)
- V-NLI: Visualización orientada a Interfaces de Lenguaje Natural
- SpeechSQLNet: Modelo para generar consultas SQL NLQ: Consulta en Lenguaje Natural

1. INTRODUCCIÓN

En el ámbito judicial, la accesibilidad es especialmente importante, ya que las personas con discapacidad tienen el derecho de acceder a las herramientas de justicia en igualdad de condiciones, una necesidad que se hace aún más relevante en la era digital actual. En este contexto, la implementación de una interfaz conversacional para el módulo de búsquedas de legajos judiciales no solo tiene como objetivo una mayor accesibilidad, sino que también representa un paso significativo hacia la modernización y la innovación en el sistema judicial. La tendencia del uso de lenguaje natural junto a estas tecnologías permite a las personas con discapacidad realizar búsquedas de forma rápida y sencilla, reduciendo la dependencia de ayudas técnicas o de la asistencia de terceros. Además, la adopción de una interfaz conversacional es un claro indicativo de la adaptación del sistema a las tendencias tecnológicas actuales, enfatizando la importancia de la digitalización y la automatización en la gestión de archivos judiciales.

Por otra parte, la modernización mediante estas tecnologías no solo beneficia a usuarios con necesidades especiales, sino que también optimiza la eficiencia y la productividad de la organización. Un sistema de archivo judicial digitalizado y accesible a través de una interfaz conversacional mejora significativamente la gestión de documentos, agiliza los procesos de búsqueda y reduce los tiempos de respuesta, aspectos cruciales en un entorno donde la rapidez y la precisión son fundamentales.

Por lo tanto, la introducción de herramientas y adaptaciones tecnológicas a los sistemas judiciales actuales es esencial no solo para cumplir con los requisitos de accesibilidad, sino también para asegurar que el sistema judicial se mantenga alineado con las exigencias y oportunidades que presenta la era digital. Se propondrán a su vez, especificaciones de diseño de Interfaz Conversacional, el desarrollo de un prototipo; y finalmente se hace una comparativa de prestaciones y performance utilizando herramientas que permitan medir las paramétricas de los procesos involucrados.

1.1 Fundamentación

El desarrollo de aplicaciones web ha sufrido un crecimiento exponencial en los últimos años y ha revolucionado múltiples sectores, incluido notablemente el ámbito judicial. Esta transformación digital es particularmente importante en la búsqueda y gestión de legajos judiciales, un proceso que requiere rapidez, precisión y accesibilidad. En este escenario, la eficiencia en la localización y manejo de información legal se ha convertido en un pilar fundamental para agilizar procedimientos y democratizar el acceso a datos jurídicos [1]. En base a estas necesidades, este trabajo se centra en el diseño de una Interfaz Conversacional y un prototipo que integra comandos de voz para la búsqueda de legajos, con el objetivo de proporcionar una herramienta no solo eficiente en su rendimiento sino también inclusiva en su diseño, permitiendo a usuarios de variadas capacidades interactuar de manera simple y natural con sistemas legales digitalizados. Así, este proyecto no solo responde a la necesidad de modernización tecnológica en el ámbito judicial, sino que también se adelanta a los desafíos y oportunidades que la era digital continúa presentando.

Nace como respuesta a una iniciativa de modernización e innovación del área de sistemas para extender las capacidades de accesibilidad, desde un análisis previo en el cual se han detectado sobrecargas en los procesos administrativos de generación de reportes, por lo que se requiere necesario en estos casos mejorar los tiempos de respuesta y el acceso al mismo. En este sentido, la accesibilidad es un derecho fundamental que debe ser garantizado para todas las personas, sin importar su condición física o intelectual. Las interfaces conversacionales son una herramienta que puede contribuir a mejorar la accesibilidad de los sistemas informáticos ya que permiten a las personas con discapacidad interactuar con ellos de una manera natural [2]. Se puede ilustrar las mejoras en eficiencia, accesibilidad y satisfacción del usuario antes y después de la implementación de la interfaz conversacional. En la tabla 1.1 se puede listar las ventajas de la implementación en sus objetivos principales con sus características.

Criterio	Antes de la Implementación	Después de la Implementación
Eficiencia en la búsqueda de legajos	Tiempos de búsqueda largos, procesos manuales.	Búsqueda rápida a través de comandos de voz.
Accesibilidad	Limitada para usuarios con discapacidad.	Mejorada, accesible para usuarios con diversas capacidades.
Satisfacción del usuario	Frustración por demoras y accesibilidad limitada.	Mayor satisfacción por rapidez y facilidad de uso.

Tabla 1.1 Comparación: Antes y Después de la Implementación esperada

En la experiencia de los usuarios existen problemas cuando sufren algún tipo de deficiencia en la movilidad, surge la necesidad de determinados elementos específicos para poder acceder a utilizar e interactuar con la aplicación mediante periféricos, o en ubicaciones permanentes que en ocasiones especiales requieren puntual atención y accesibilidad para poder hacer uso de los sistemas; situaciones que traen consecuencias muy negativas para la calidad de vida. Promoviendo así la accesibilidad para personas con discapacidad con especial énfasis en este objetivo, se enumeran a continuación la fundamentación de las ventajas más significativas entre el uso tradicional y el uso mediante lenguaje natural a través de una interfaz conversacional:

1.1.1 Ventajas significativas en comparación a la usabilidad tradicional

La revolución digital ha tocado todos los aspectos de nuestras vidas, reconfigurando la forma en que interactuamos con la tecnología en nuestro día a día. El ámbito judicial, un pilar fundamental de nuestra sociedad, no es una excepción a esta transformación. Con la creciente necesidad de modernizar y hacer más eficientes los procesos legales, emerge la interfaz de voz como una herramienta poderosa y prometedora. Esta sección explora las ventajas significativas que las interfaces de voz y el procesamiento del lenguaje natural aportan en comparación con los métodos tradicionales de interacción con sistemas informáticos judiciales [3]. Desde la

facilidad de uso hasta la precisión en la ejecución de comandos, la incorporación de esta tecnología representa un salto cualitativo hacia una mayor accesibilidad y eficiencia. En este contexto, la aplicación de interfaces conversacionales no solo refleja un avance en la usabilidad sino que también marca un hito en la inclusión, al permitir que usuarios con diferentes capacidades puedan beneficiarse de los servicios legales digitalizados [4]. A continuación, se desglosan estas ventajas, destacando cómo la adaptabilidad y evolución constante del lenguaje natural facilitan una interacción más orgánica y satisfactoria con la tecnología, y cómo esto, a su vez, incide positivamente en la productividad y gestión de los sistemas judiciales.

- **Facilidad de Uso**

Búsquedas Tradicionales (Teclado y Mouse): Requieren conocimiento previo sobre cómo navegar por la interfaz del sistema. Pueden ser menos intuitivas para usuarios no familiarizados con la tecnología. Permiten una navegación precisa cuando el usuario sabe lo que está buscando.

Comandos por Voz en Lenguaje Natural: Permiten a los usuarios interactuar con el sistema de manera más natural y conversacional. Son particularmente útiles para personas que pueden tener dificultades para escribir o usar un mouse. Pueden requerir un período de adaptación para que el sistema reconozca adecuadamente los comandos verbales del usuario.

- **Accesibilidad**

Búsquedas Tradicionales: Pueden presentar barreras para usuarios con discapacidades visuales o motoras. Requieren el uso de hardware adicional (teclado y mouse).

Comandos por Voz: Mejoran significativamente la accesibilidad, permitiendo a usuarios con diversas discapacidades interactuar de manera más efectiva. No requieren de hardware adicional, facilitando el acceso en dispositivos móviles o cuando el hardware está limitado.

- **Velocidad**

Búsquedas Tradicionales: La velocidad de navegación depende de la habilidad del usuario con el teclado y el mouse. La búsqueda precisa puede llevar tiempo, especialmente en interfaces complejas o con mucha información.

Comandos por Voz: Pueden ofrecer resultados más rápidos para búsquedas o comandos simples, ya que el sistema procesa el lenguaje natural de manera eficiente. La velocidad puede verse afectada por la precisión del reconocimiento de voz y la complejidad del comando.

- **Precisión**

Búsquedas Tradicionales: Ofrecen alta precisión cuando el usuario sabe cómo y dónde buscar la información deseada. Permiten el uso de filtros avanzados y comandos específicos para refinar los resultados.

Comandos por Voz: La precisión depende de la capacidad del sistema para entender correctamente el lenguaje natural y el contexto del usuario. Los errores de interpretación o reconocimiento pueden llevar a resultados inesperados.

- **Experiencia de Usuario**

Búsquedas Tradicionales: Pueden ser vistas como más controladas y seguras por algunos usuarios, especialmente en entornos de trabajo o donde la precisión es crítica. La curva de aprendizaje puede ser un factor para nuevos usuarios.

Comandos por Voz: Ofrecen una experiencia más dinámica y adaptativa, especialmente en entornos casuales o para usuarios que buscan conveniencia. La interacción puede ser más personal y satisfactoria, especialmente cuando el sistema interpreta correctamente los comandos complejos.

Por otra parte, con esta aplicación se contribuye a la modernización de las funcionalidades a través de la implementación de tecnologías emergentes, superando las limitaciones de los sistemas tradicionales, ya que las hace más fáciles de usar, además las interfaces conversacionales permiten agregar nuevas funcionalidades como la posibilidad de realizar búsquedas avanzadas mediante comandos de voz sobre filtros preestablecidos y acotados a la terminología utilizada en el ámbito judicial, representando éstos un claro impacto positivo en la productividad en la gestión general del sistema. Cuando decimos «lenguaje natural» nos referimos a aquel que, además de ser utilizado por humanos, está en evolución constante gracias a sus hablantes. Puede analizarse e incluso regirse por una gramática, pero está en constante transformación y flexibilidad, es decir, está vivo. Un lenguaje natural tiene intención, se sirve del contexto en el que se habla para entender mejor el mensaje y los hablantes aprovechan la propia voz para hacer énfasis en lo que quieren decir. Por eso un asistente virtual o un chatbot no tiene esa fluidez a la que estamos acostumbrados como cuando platicamos con alguien más.

1.1.2 Aplicaciones generales del procesamiento de lenguaje natural

Como su nombre lo indica, estas tecnologías tienen como fin analizar entradas de lenguaje para ofrecer resultados en el formato de un lenguaje hablado o automatizaciones de sistemas como el de generación de reportes; a su vez también integraciones con funcionalidades específicas, como lo es en esta implementación del prototipo, el cual disparará una redirección por pantalla. Estos resultados pueden ser utilizados con diferentes fines, veamos los más comunes:

- **Reconocimiento de patrones de lenguaje:** Al procesar grandes cantidades de documentos, el reconocimiento de patrones permite filtrar datos importantes en cadenas de texto con un tiempo récord. Es el primer paso para que la recuperación de información y la clasificación de textos sea posible. Esto ayuda al análisis automatizado de archivos y documentos y a su clasificación mediante la detección de denominadores comunes.
- **Recuperación de información:** Gracias a que los sistemas de PLN reconocen patrones de lenguaje, también hacen más sencilla la tarea de encontrar un fragmento en particular dentro de una gran cantidad de texto. En primera instancia, estas

herramientas no inventaron palabras o frases nuevas, sino que identificarán la información valiosa y te ayudarán a comprender mejor un documento o reporte.

- **Traducciones automáticas de idiomas:** Ya sea con voz o texto, estos sistemas son ideales para traducir discursos en cualquier lengua. El funcionamiento dependerá de la habilidad de la tecnología para distinguir intenciones, tonos y fórmulas de lenguaje cotidiano o especializado. Para ello, deberán utilizar datos que se procesan por la lingüística computacional y, por tanto, es necesario que estén en un proceso constante de mejora y aprendizaje.
- **Clasificación de información:** Gracias al procesamiento de palabras clave, la información que sirva como valor de entrada al sistema puede categorizarse para que su consulta sea más eficiente. Esto es muy útil a la hora de procesar amplios volúmenes de información, como entrevistas, encuestas o reportes escritos de desempeño.
- **Resumen de textos:** Al igual que con la clasificación, resumir un documento de gran extensión se apoya en ciertas palabras o frases clave. Si programas tus herramientas para detectar esas palabras clave, será realmente sencillo que te ofrezca resúmenes ejecutivos de un documento extenso. Sin duda, esto puede ser muy útil para entregar reportes resumidos o bien síntesis de un proyecto de trabajo.
- **Generación de lenguaje natural:** La generación de lenguaje natural es uno de los objetivos más ambiciosos de esta lista, pues es lo que permite que una máquina responda las interacciones de un humano con frases nuevas y no como lo haría un chatbot (que elige de una lista establecida la opción que mejor se adapta). Actualmente, existen muchos recursos que funcionan como generadores de redacciones y que están habilitadas para redactar de forma autónoma. Esto es útil ya sea que quieras una herramienta con IA para escribir correos, redactar la entrada de un blog o enviar un mensaje de invitación a tus contactos.
- **Detección de sentimientos y emociones:** Para comprender ciertos mensajes con su intencionalidad, el procesamiento de lenguaje natural ya incursiona en el análisis de las emociones que se expresan a través de frases que aparecen en opiniones. Esto requiere el análisis del tono, la rapidez y el lenguaje durante una comunicación, por lo que es una de las funciones más avanzadas que se espera desarrollar de estas herramientas.

1.2 Objetivo general

El objetivo de este trabajo es diseñar una interfaz conversacional para el módulo de búsquedas de legajos judiciales del sistema de gestión actualmente implementado, que permite a los usuarios realizar búsquedas mediante comandos de voz, lo que facilitará el acceso a la información y mejorará la experiencia, independientemente si éstos tengan algún impedimento físico que impida el normal uso de los periféricos y herramientas dispuestos por el sistema. En el ámbito de las aplicaciones web esto se materializa a través de los estándares W3C [5] y Web Speech API [6], los cuales serán tenidos en cuenta en el diseño y desarrollo de la solución, en particular se buscará aplicar patrones del diseño de Interfaces Conversacionales

basados en las referencias teóricas como por ejemplo en: “Principio de la Cooperación, Toma de Turnos y Contexto” [7] y la “Lógica y Conversación” [8].

Por otro lado, la importancia de la modernización de los sistemas judiciales es una necesidad recurrente junto a las tecnologías emergentes que lo permiten, a contraposición de los sistemas tradicionales que a menudo son obsoletos y difíciles de usar desde el punto de vista accesible. Es en este punto que se hace énfasis en los problemas que se puedan generar retrasos y errores en los procesos de búsquedas en la gestión de los registros judiciales. Por lo tanto, los principales objetivos del proyecto facilitarán el acceso a la información para personas con discapacidad y mejorará la experiencia de usuario para todos, con objetivos de contribuir a una Interfaz Conversacional Amigable que permita a los usuarios realizar búsquedas a través de comandos de voz intuitiva, contribuyendo a una experiencia de usuario inclusiva y eficaz, y teniendo en cuenta que la notable mejora de la experiencia de todos los usuarios del sistema en cuestión.

1.3 Objetivos Específicos

- **OE1:** Analizar las buenas prácticas y definiciones existentes en la industria del software destinadas a las búsquedas de registro de expedientes judiciales.
- **OE2:** Investigar tecnologías, requerimientos, compatibilidades y alcances de herramientas de interfaces de síntesis y reconocimiento de voz para el uso de lenguaje natural en aplicaciones web a través de los estándares W3C y Web Speech API.
- **OE3:** Desarrollar un prototipo funcional para el ámbito judicial aplicando los principios de Diseño de Interfaces Conversacionales “Principio de la Cooperación, Toma de Turnos y Contexto”; y “Lógica y Conversación” basado en casos de éxito.
- **OE4:** Evaluar y medir resultados en el rendimiento de procesamiento de la voz, compatibilidad y usabilidad en términos de accesibilidad y procesamiento en los clientes.

2. MARCO TEÓRICO

El marco teórico se fundamenta en pilares esenciales como la síntesis y procesamiento de las señales de voz en los sistemas de reconocimiento basados en cualquier unidad, se ilustra en la Figura 2.1 un diagrama de bloques convencional de un sistema de reconocimiento.

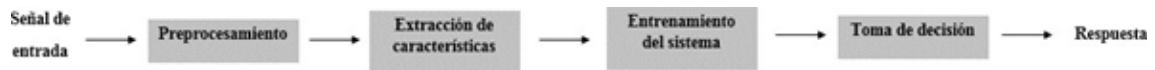


Figura 2.1 Diagrama de bloques de un Sistema de reconocimiento

2.1 Metodología genérica empleada en reconocimiento de voz

Todo trabajo de reconocimiento de voz requiere de un conjunto de tareas, las cuales se ejecutarán de preferencia en secuencia e independientemente del objetivo a alcanzar. Estas tareas se enlistan a continuación:

- Preparar y crear un Corpus de Voces a utilizar en el trabajo.
- Definir, preparar y realizar el pre-procesamiento que será aplicado a los archivos del Corpus.
- Seleccionar, diseñar y programar los algoritmos y métodos que se utilizarán para la extracción de parámetros a utilizar en el reconocimiento de voz.
- Entrenar el sistema.
- Verificar el grado (%) de aceptación del sistema (reconocimiento).

2.1.1 El Corpus de Voces

El Corpus de Voces es el conjunto de archivos con información de voz digitalizada, del cual se extraen el conjunto de parámetros con los cuales se entrena y comprueba el sistema desarrollado.

Para obtener condiciones de un corpus de calidad es muy importante, durante la etapa de grabación de sonidos, tener en cuenta las condiciones del lugar; distancia al micrófono; ajuste de la ganancia y sensibilidad de dicho micrófono. En la figura 2.2 se tienen tres ejemplos de grabaciones de números del cero al nueve, la primera está saturada, con mucho ruido y poca separación entre palabras, la segunda es adecuada y la tercera grabación tiene un volumen muy bajo.

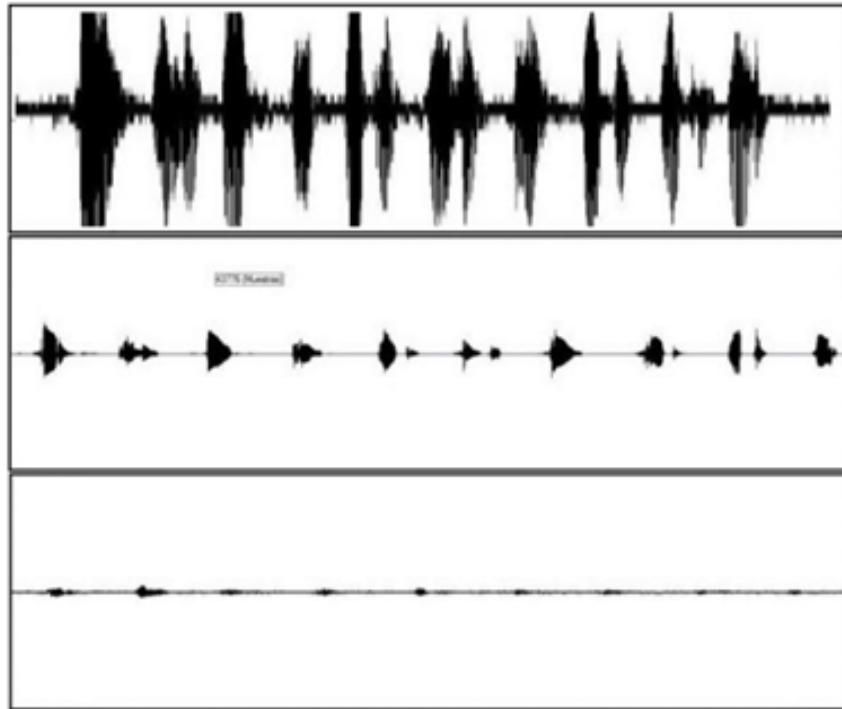


Figura 2.2 Ejemplo de grabaciones con amplitud: saturada, media y baja.

2.1.2 Preprocesamiento de la señal de voz

La descripción del modelo fuente-filtro para la producción de voz mostrado, indica que el espectro de sonidos sonoros posee una pendiente de -6 dB/octava, mientras se incrementa la frecuencia. Esta es una combinación de una pendiente de -12 dB/octava debido a la fuente de excitación de la voz y una pendiente +6 dB/octava debida a la radiación producida por la boca. Esto significa que por cada vez que se duplica la frecuencia, la señal en amplitud, y por tanto la respuesta del tracto vocal medida, son reducidas por un factor de 16.

Es por tanto deseable compensar la caída de -6 dB/octava pre-procesando la señal de voz para darle un levantamiento de +6 dB/octava en el rango apropiado de tal forma que el espectro medido tenga un rango dinámico similar a través de toda la banda de frecuencias. Esto se llama preénfasis.

En un sistema digital de procesamiento de señales, el preénfasis puede ser implementado ya sea de primer orden, usando un filtro analógico paso-altas con una frecuencia de corte a 3 dB en algún lugar entre 100 Hz y 1 kHz (la posición exacta no es crítica) que precede al filtro de anti-aliasing y el convertidor A/D, o bien usando un filtro digital paso-altas que procesa la señal de voz digitalizada. El filtro paso-altas puede ser logrado digitalmente usando la ecuación de la Figura 2.3, en diferencias:

$$y[n] = x[n] - ax[n-1]$$

Figura 2.3 Ecuación en diferencias de filtro paso-altas.

Donde $y[n]$ denota la muestra actual de salida del filtro de preénfasis, $x[n]$ es la muestra actual de entrada, $x[n-1]$ es la muestra anterior de entrada y a es una constante usualmente elegida entre 0.9 y 1. De nuevo, el valor elegido no es crítico. En la siguiente Fig 2.4: tomando la transformada z de la ecuación da:

$$Y(z) = X(z) - az^{-1}X(z) = (1 - az^{-1})X(z)$$

Fig 2.4 Transformada Z de la Ecuación 1

2.1.2.1 Ventaneo

El mecanismo de ventaneo muestra cómo a cada porción de la señal de voz (de un tamaño predefinido por el usuario), se le asigna una ventana, de tal forma que las muestras queden ponderadas con los valores de la función escogida. En este caso, las muestras que se encuentran en los extremos de la ventana tienen un peso mucho menor que las que se hallan en el medio, lo cual es muy adecuado para evitar que características de los extremos del bloque varíen la interpolación de lo que ocurre en la parte central, la cual es la más significativa, de las muestras del segmento seleccionado. La colocación de las ventanas puede realizarse de tal forma que existan solapamientos y, aunque ello repercutirá en los tiempos de respuesta del sistema reconocedor, proporcionará una mejor calidad en los resultados obtenidos. Las funciones de ventaneo más comunes se muestran a continuación:

Tipo de Ventana	Amplitud pico lóbulo lateral (relativa)	Ancho aproximado del lóbulo principal	Error pico aproximado o $20\log_{10}d$ (dB)	Ventana de Kaiser equivalente b	Ancho de transición de ventana de Kaiser
Rectangular	-13	$4p/(M+1)$	-21	0	$1.81p/M$
Bartlett	-25	$8p/M$	-25	1.33	$2.37p/M$
Hanning	-31	$8p/M$	-44	3.86	$5.01p/M$
Hamming	-41	$8p/M$	-53	4.86	$6.27p/M$
Blackman	-57	$12p/M$	-74	7.04	$9.19p/M$

Tabla 2. 1 Comparación de las ventanas comúnmente usadas.

2.1.3 Detección de inicio y fin de palabras

La detección de inicio y fin de palabras, se realiza tomando en cuenta la actividad de energía y cruce de ceros de la señal $y(n)$, con respecto a los valores que se tienen en condiciones de silencio, ruido ambiente. La variación del cruce de ceros es principalmente motivo de la emisión de una señal explosiva ("p", "t", "k", "b") o ruido aleatorio ("s", "f",

“ch”, “j”, “z”). La variación de energía ocurre en presencia de vocales (“a”, “e”, “i”, “o”, “u”), semivocales (“m”, “n”, “ñ”, “w”, “d”, “l”, “g”, “y”).

La señal que se muestra en la figura 5, es una muestra de una detección de inicio y fin, ilustrado en la *figura 2.5*, corresponde a la palabra “seis”, donde se aprecia el inicio del sonido de la silbante “s”, detectado mediante el cruce de ceros. Las vocales “e” e “i”, se caracterizan por el nivel de su energía y nuevamente la “s” al final. Al inicio y fin de la grabación se aprecia el efecto del ruido ambiente que corresponde al silencio.

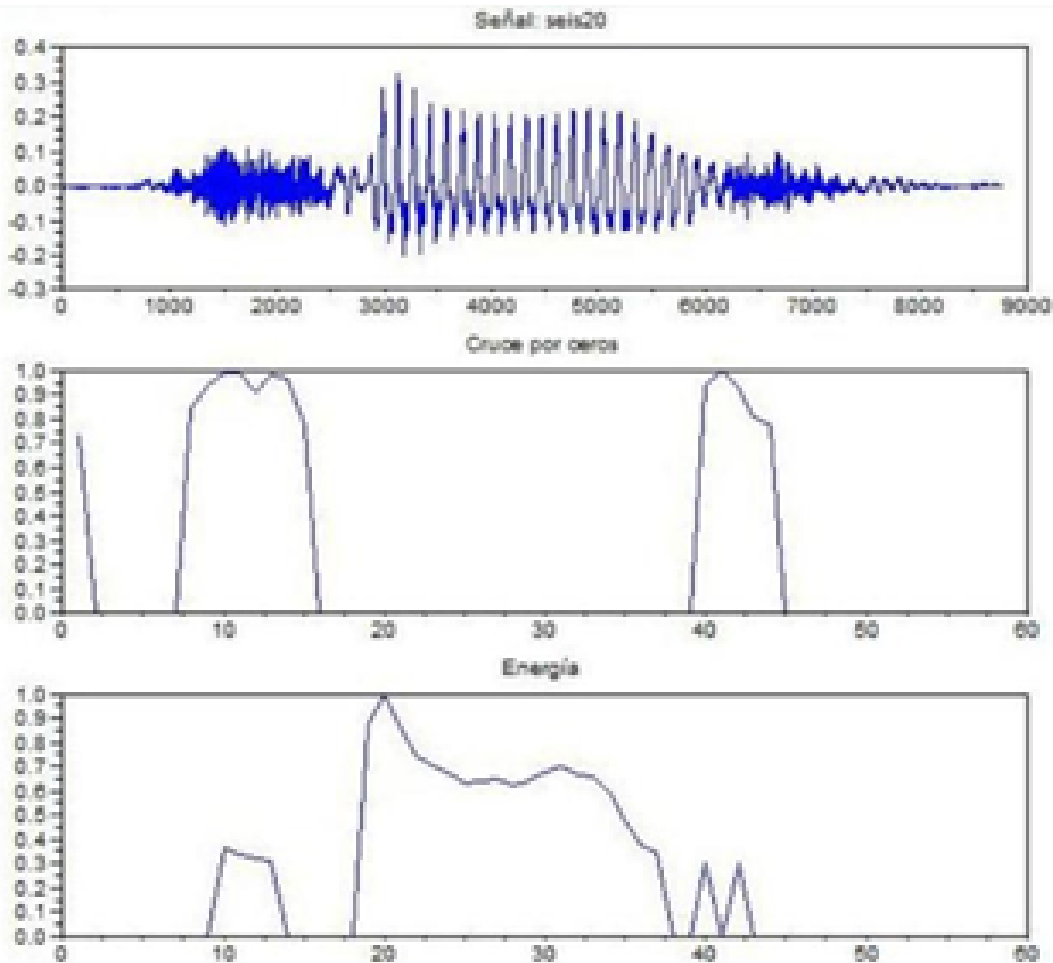


Fig 2.5 Ejemplo de detección de inicio y fin de palabra con cálculo de cruce de ceros y energía.

2.1.4 Parámetros característicos de una señal de voz

A través de la historia las diferentes investigaciones referidas al reconocimiento de la voz nos han dejado que los parámetros que mejores resultados han dado son: el LPC, CEPSTRUM, MFCC, entre otros.

2.1.4.1 Cepstrum

El Cepstrum es llamado por algunos autores, la transformada de la transformada, Lo cual es casi cierto. A continuación se presenta el flujo de procesamiento para obtener los coeficientes Cepstrales de una señal. Vemos un flujo ejemplar en la siguiente Figura 2.6

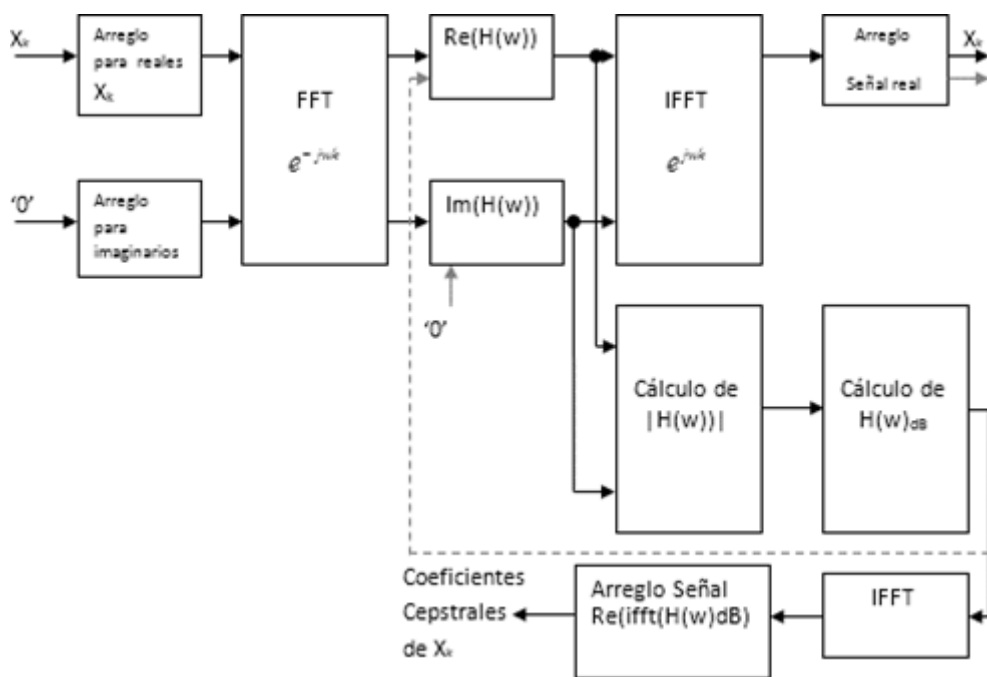


Figura 2.6 Cálculo de los coeficientes cepstrales (Cepstrum) de un segmento de señal.

Podemos observar que la operación es idéntica a llevar el resultado de $H(\omega)_{dB}$ a la entrada del módulo que realiza la IFFT y tomar los valores reales de esta operación, siendo el resultado $H(\omega)_{dB}$ el conjunto de los números reales a tener en cuenta y haciendo el conjunto de los números imaginarios igual a CERO (0). En ese caso en lugar de obtener los valores de obteniendo los valores de los coeficientes cepstrales de la señal. El Cepstrum de una señal es llamado la cuasifrecuencia de la señal, lo que es equivalente a decir que es parecido al periodograma de la señal, ya que estamos realizando la transformada inversa de Fourier al espectro de Potencia de la Frecuencia de la señal y la respuesta corresponde al dominio del tiempo. El componente sobre el valor de la ordenada 1 equivale a la frecuencia de muestreo dividida por 2 (frecuencia de análisis de la señal más alta, según la teoría del muestreo) o lo que es lo mismo, el período que se puede analizar de la señal más pequeña.

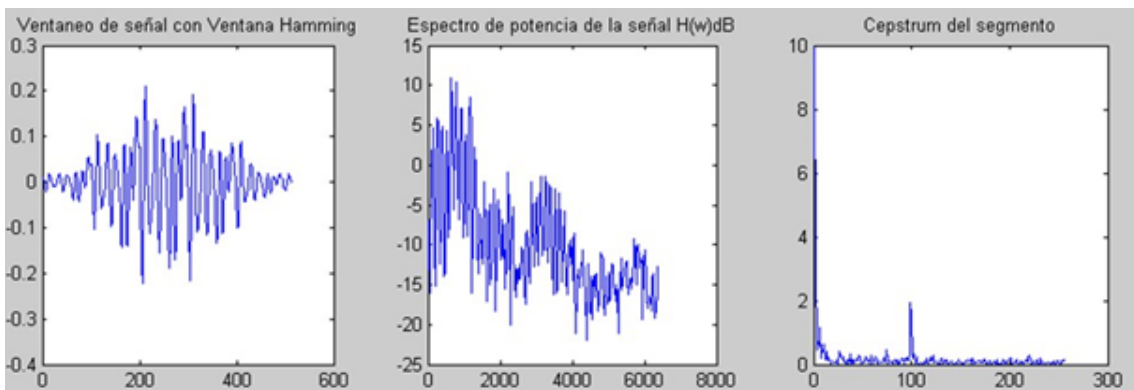


Figura 2.7 Gráficas resultantes en el flujo de cálculo del Cepstrum.

Para señales que tienen definida su composición espectral o rango de respuesta de frecuencia, se puede seleccionar el rango de coeficientes Cepstrum que le corresponden, eliminando el resto por considerarlos ruidos o no pertenecientes a la señal bajo análisis.

La cantidad de coeficientes Cepstrum dependen de la frecuencia de muestreo de la señal, ya que el Df de la $H(w)_{dB}$, está en función de la misma como se explicó anteriormente. A mayor frecuencia de muestreo, mayor cantidad de coeficientes Cepstrales ilustrado en la figura 2.7. Para algunas aplicaciones de reconocimiento de patrones, como es el caso de reconocimiento de voz, el número de coeficientes Cepstrales para los patrones de la señal voz se toman bajo este criterio.

2.1.4.2 LPC

El modelo de filtro “todo polos” es el más utilizado y el análisis de auto-recursividad de la señal es la herramienta para hallar los coeficientes del filtro en cuestión. La funcionalidad matemática que da como resultado los coeficientes del filtro, es una relación matricial que se obtiene al realizar la predicción lineal de las muestras de las señales, con lo cual se obtienen los coeficientes de predicción lineal (LPC), también llamados parámetros LPC, que no son otra cosa que los coeficientes del filtro “todo polos” buscado.

El cálculo de parámetros LPC es un procedimiento que se realiza en el dominio del tiempo, y está basado en la consideración de que el término $s(n)$ de una secuencia, puede ser estimado a partir de los términos anteriores, considerando la sumatoria de los mismos con un peso asociado a cada uno de ellos. $\hat{s}(n)$ es el término estimado de la secuencia Figura 2.8. Este modelo es llamado predicción lineal (LPC) o auto recursivo (AR).

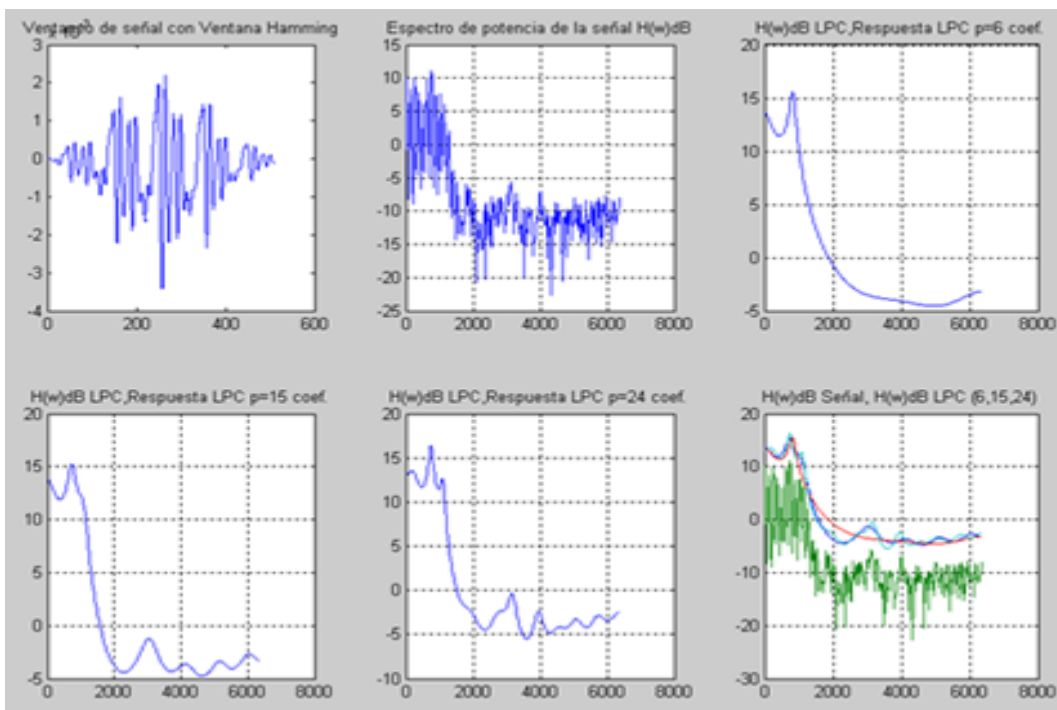


Figura 2.8 Comparativa de respuestas de frecuencias LPC

2.1.4.3 MFCC

Un método más eficiente para extraer características y que es el más utilizado actualmente en reconocedores comerciales son los Coeficientes Cepstrales en Escala de Mel (MFCC), este método es robusto, además hace uso de la Transformada de Fourier para obtener las frecuencias de la señal. El objetivo es desarrollar un conjunto de características basadas en criterios perceptuales, diversos experimentos muestran que la percepción de los tonos en los humanos no está dada en una escala lineal, esto hace que se trate de aproximar el comportamiento del sistema auditivo.

Los coeficientes Cepstrales en Frecuencia en Escala de Mel (MFCC) son una representación definida como el Cepstrum de una señal ventaneada en el tiempo que ha sido derivada de la aplicación de una Transformada Rápida de Fourier, pero en una escala de frecuencias no lineal, las cuales se aproximan al comportamiento del sistema auditivo humano.

2.2 Interfaz de Voz y Procesamiento de Lenguaje Natural

La tecnología de voz y el Procesamiento de Lenguaje Natural (PLN) desempeñan un papel clave en la creación de una interfaz intuitiva, la comprensión del lenguaje natural permite a la aplicación interpretar y ejecutar comandos de voz, mejorando significativamente la experiencia del usuario. Se explorarán modelos avanzados de PLN y técnicas de reconocimiento de voz para garantizar la precisión y la eficiencia en la interacción.

2.2.1 Esquema general de una interfaz de Lenguaje Natural

El esquema general de una interfaz de lenguaje natural (Natural Language Interface, NLI) se centra en la capacidad de un sistema para entender y procesar el lenguaje humano de manera que los usuarios puedan interactuar con máquinas usando el lenguaje cotidiano, esquematizado en la Figura 2.9 Esquema general de una interfaz de Lenguaje Natural. Este tipo de interfaz es posible gracias a la combinación de varias tecnologías y procesos. Aquí te presento un desglose del esquema basado en el esquema que proporcionaste y la descripción del uso de tecnología de voz y PLN:

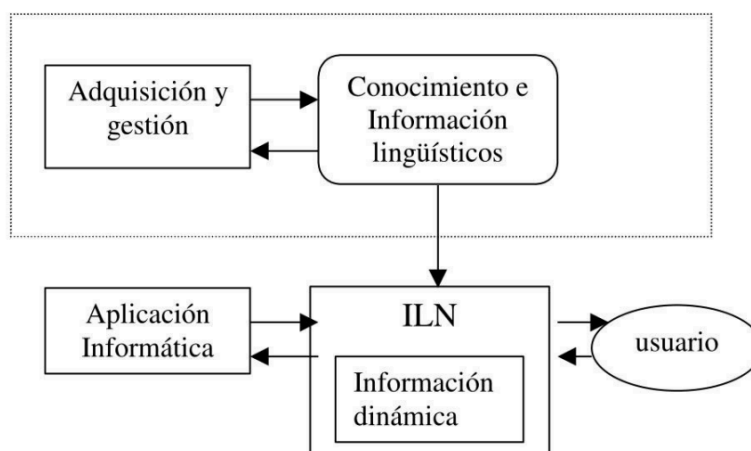


Fig 2.9 Esquema general de una interfaz de Lenguaje Natural

1. *Adquisición y Gestión de Datos*: La interfaz recoge datos de entrada de voz del usuario. Estos datos se gestionan y se preparan para su procesamiento, lo cual puede incluir la eliminación de ruido, la amplificación de la señal, y la segmentación en unidades comprensibles (por ejemplo, palabras o frases).
2. *Conocimiento e Información Lingüísticos*: Se utiliza el PLN para analizar la estructura lingüística de la entrada de voz. Esto implica descomponer el lenguaje en partes (análisis morfológico), entender la función de cada parte en una oración (análisis sintáctico), y determinar el significado (análisis semántico).
3. *Aplicación Informática*: La aplicación informática es el software que utiliza el PLN para procesar el lenguaje natural. Esta aplicación puede ser un sistema de gestión de legajos judiciales, un asistente virtual, un motor de búsqueda, etc.
4. *Interfaz de Lenguaje Natural (ILN)*: La ILN es la capa de la interfaz que interactúa directamente con el usuario. Traduce la entrada de voz procesada en comandos que la aplicación informática puede entender y actuar en consecuencia.
5. *Información Dinámica*: La interfaz proporciona una salida dinámica basada en la interacción del usuario, como la respuesta a una consulta o la ejecución de una acción.
6. *Usuario*: El usuario interactúa con la ILN mediante comandos de voz y recibe respuestas de la aplicación informática. La experiencia del usuario debe ser intuitiva y fluida para que la tecnología sea efectiva.

Cada componente tiene un papel crítico en la facilitación de una interacción natural entre el usuario y la máquina. La tecnología de voz y el PLN permiten que la interfaz interprete comandos de voz y responda de manera que se asemeje a una conversación natural entre humanos.

2.2.2 Arquitectura típica de una NLI que utiliza un lenguaje de representación intermedio

La arquitectura de una NLI que utiliza lenguajes de representación intermedia generalmente consta de varias capas clave que trabajan en conjunto para convertir la entrada del lenguaje natural del usuario en una forma que pueda ser comprendida y ejecutada por un sistema informático [9].

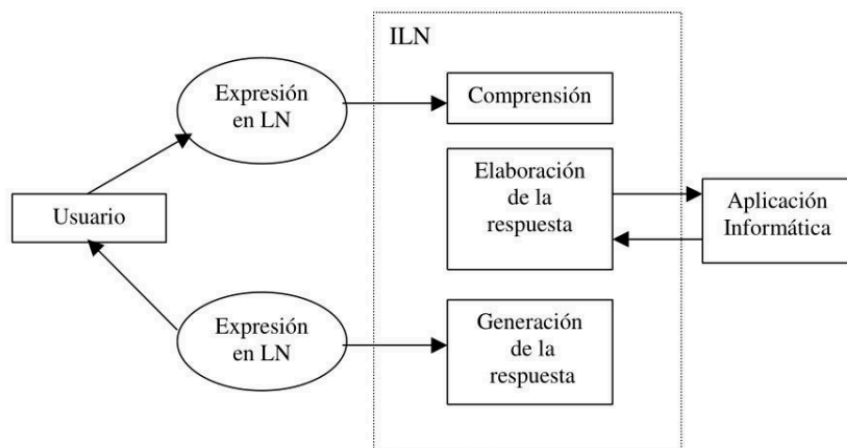


Fig 2.10 Esquema Arquitectura típica de una NLI

Estas capas incluyen:

Interfaz de Usuario (UI): Es la capa más externa donde los usuarios interactúan con el sistema a través del lenguaje natural. *Capa de Interpretación de Consultas:* Utiliza el Procesamiento de Lenguaje Natural (PLN) para interpretar la entrada del usuario y descomponerla en componentes comprensibles.

Transformación de Datos: Convierte los componentes interpretados en una representación intermedia que refleje la estructura de una orden como se diagrama en la Figura 2.4 *Representación de la Orden y el significado de la consulta del usuario.*

Mapeo Visual y Gestión de Diálogos: Convierte la representación intermedia en acciones o visualizaciones específicas y gestiona el flujo de la conversación ejemplificado en la *Figura 2.6 Ejemplo de Estructura de Conversación.*

Capa de Presentación: Muestra la salida del sistema al usuario, ya sea en forma de acciones ejecutadas o respuestas visuales o textuales. En las Figuras 2.10 y 2.11 se esquematizan arquitecturas típicas.

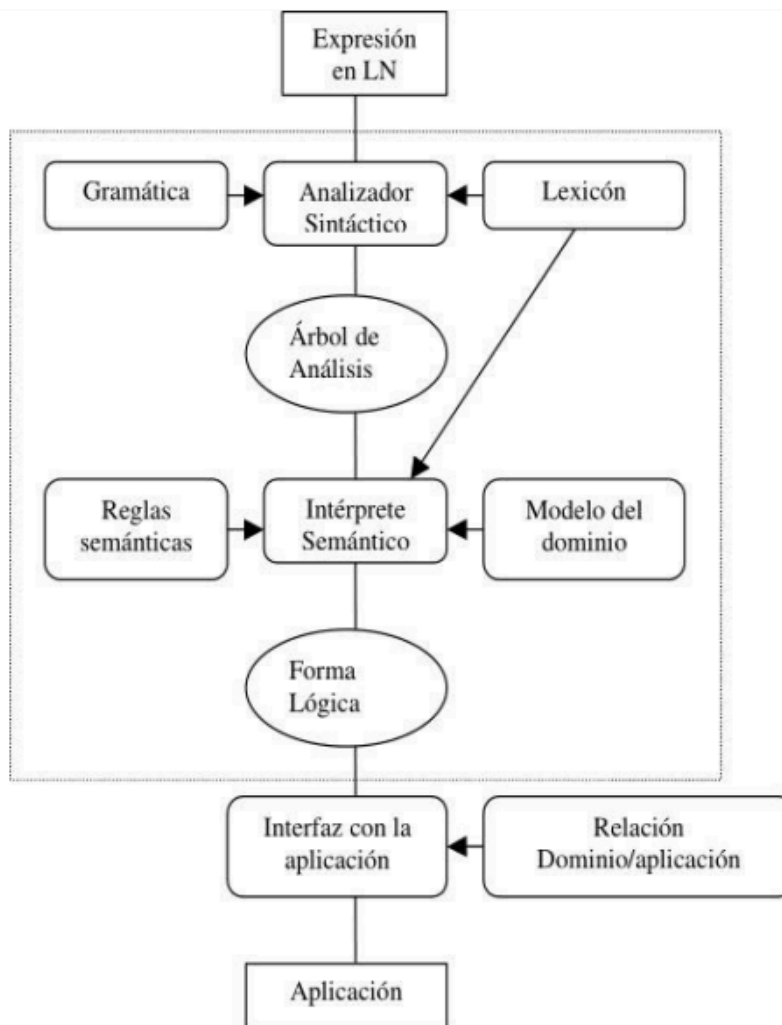


Fig 2.11 Arquitectura típica de una NLI

En el contexto de las bases de datos, esta arquitectura puede involucrar la traducción de consultas del lenguaje natural, un ejemplo convencional podemos verlo en la *Figura 2.13*, a consultas SQL [10] a través de una representación intermedia como lógica de predicados visto en la figura 2.12; o un cubo OLAP en el caso de análisis de datos [11].

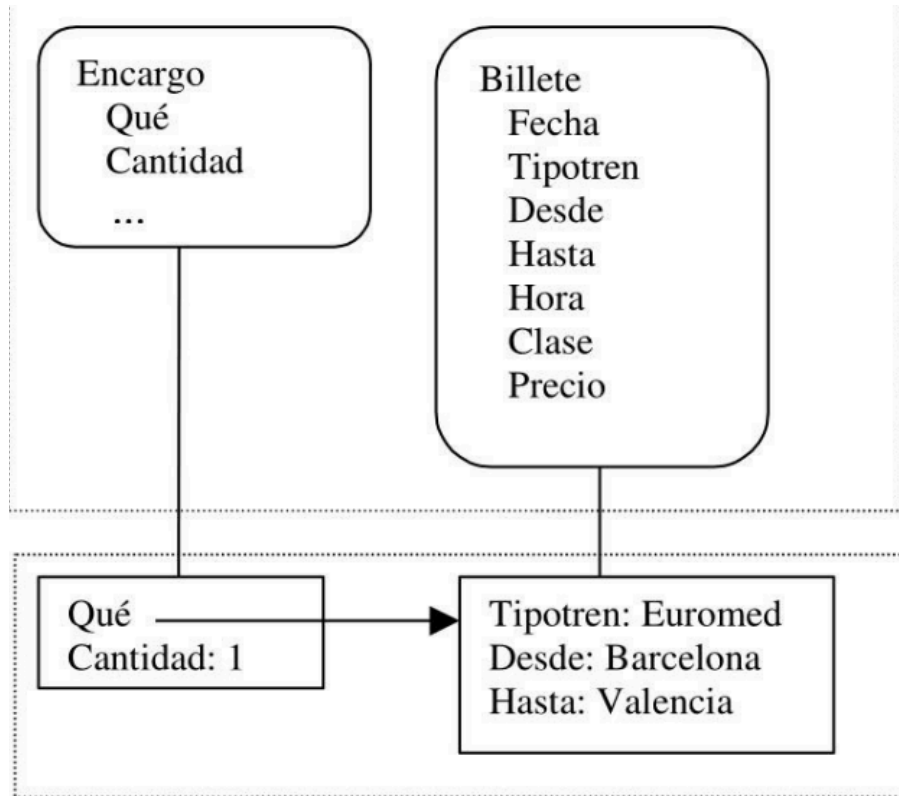


Fig 2.12 Representación de la Orden

C		
D	{tema: bienvenida }	
Interc	{objetivo: saludar}	
Interv		S> Bienvenido al servicio de información, ¿Qué desea?
D	{tema: viaje en tren de Barcelona a Valencia}	
Interc	{objetivo: petición de una información}	
Interv		U> Quisiera ir de Barcelona a Valencia
Sub	{tema: fecha del viaje}	
Interc	{objetivo: precisión}	
Interv		S> ¿En qué fecha?
Interv		U> el martes que viene
Sub	{tema: hora del viaje}	
Interc	{objetivo: precisión}	
Interv		S> ¿qué horario, mañana o tarde?
Interv		U> a primera hora
Interv		S> Hay 3 trenes, el Euromed
Interv		
Interc	{objetivo: petición de una información}	
Interv		U> ¿Cuánto vale el billete en segunda?
Interv		S> 8000 pesetas
D	{tema: despedida}	
Interc	{objetivo: despedirse}	
Interv		U> Gracias, buenas tardes
Interv		U> Gracias a Vd, buenas tardes

Fig 2.13 Ejemplo de Estructura de Conversación

2.3 Análisis de Herramientas

Es importante considerar varios factores como la facilidad de uso, la compatibilidad, el soporte de idiomas, la calidad de la voz y la personalización. Aunque la pregunta inicial sugiere enfocarse en Google Dialogflow como la mejor opción para este proyecto específico de interfaz conversacional, en realidad, Dialogflow es una plataforma para construir interfaces conversacionales (chatbots y agentes de voz), mientras que las herramientas listadas se especializan en la conversión de texto a voz (TTS) y reconocimiento de voz (STT).

Sin embargo, procederé a crear una tabla comparativa de las herramientas de TTS mencionadas y luego discutiré cómo Dialogflow complementa estas tecnologías en la creación de una solución integral.

Característica / Herramienta	Google Cloud Text-to-Speech	IBM Watson Text to Speech	Microsoft Azure Speech Service	Web Speech API (W3C)
Plataforma / Proveedor	Google Cloud	IBM Cloud	Microsoft Azure	Estándar Web (W3C)
Requerimientos	Cuenta en GCP, API Key	Cuenta en IBM, API Key	Cuenta en Azure, Clave	Navegadores compatibles
Compatibilidad	Web Speech API	Web Speech API	Web Speech API	Navegadores modernos
Soporte de Idiomas	Amplio	Amplio	Amplio	Depende del navegador
Personalización	Alta	Media-Alta	Alta	Baja
Calidad de Voz	Alta	Alta	Alta	Varía
Costo	Basado en uso	Basado en uso	Basado en uso	Gratuito
Integración con Servicios Externos	Sí	Sí	Sí	Limitado

Tabla 2.3 Comparativa de Herramientas de TTS y STT

Google Dialogflow se distingue en proyectos de interfaces conversacionales debido a sus capacidades avanzadas de PLN (Procesamiento de Lenguaje Natural), integración con múltiples plataformas y la posibilidad de implementar respuestas complejas en diálogos dinámicos.

Aunque las herramientas de TTS y STT facilitan la conversión de texto a voz y viceversa, Dialogflow permite interpretar la intención detrás de las entradas de los usuarios y generar respuestas coherentes y contextuales.

Selección de Dialogflow como Opción para esta implementación

1. **Capacidades de PLN Avanzadas:** Dialogflow ofrece una comprensión del lenguaje natural superior, permitiendo desarrollar agentes que entienden y procesan el lenguaje humano de manera eficiente.
2. **Integración con Google Cloud Text-to-Speech y Speech-to-Text:** Dialogflow se integra a la perfección con otras herramientas de Google Cloud, permitiendo una transición fluida entre reconocimiento de voz, procesamiento del lenguaje y síntesis de voz.
3. **Multiplataforma:** Soporta la integración con web, dispositivos móviles, plataformas de mensajería y asistentes de voz, ofreciendo una amplia gama de aplicaciones.
4. **Personalización y Flexibilidad:** Permite crear respuestas dinámicas y personalizadas basadas en las intenciones y entidades detectadas en la conversación.
5. **Soporte y Documentación:** Google proporciona una extensa documentación y soporte, facilitando el desarrollo y la implementación de soluciones basadas en Dialogflow.
6. **Escalabilidad:** Como parte de Google Cloud, Dialogflow ofrece una plataforma robusta y escalable que puede crecer con las necesidades del proyecto.

Otras Herramientas de Interfaces de Síntesis y Reconocimiento de Voz actuales en el mercado que se pueden destacar son:

Google Cloud Text-to-Speech (TTS):

- **Requerimientos:** Cuenta en Google Cloud Platform, API Key o token de autenticación.
- **Compatibilidades:** Compatible con la Web Speech API de W3C, Funciona en navegadores modernos (Google Chrome, Mozilla Firefox, etc.).

IBM Watson Text to Speech:

- **Requerimientos:** Cuenta en IBM Cloud, API Key y URL de servicio.
- **Compatibilidades:** Admite la Web Speech API, Navegadores que soportan la API, como Chrome y Firefox.

Microsoft Azure Speech Service:

- **Requerimientos:** Cuenta en Azure, Clave y región para el servicio de voz.
- **Compatibilidades:** Integración con la Web Speech API, Compatibilidad con navegadores principales.

Web Speech API (W3C):

- **Requerimientos:** Navegadores compatibles con la API, como Chrome, Firefox, Edge, Safari, Conexión a Internet para servicios de síntesis y reconocimiento de voz.
- **Compatibilidades:** Ampliamente soportado en navegadores modernos. Integración con servicios de terceros que cumplen con la API.

En conclusión, mientras que las herramientas de TTS y STT son cruciales para la interacción de voz, Google Dialogflow es esencial para interpretar y procesar estas interacciones de manera inteligente, haciendo posible la creación de una interfaz conversacional avanzada que puede entender y responder a las consultas de los usuarios de manera natural y efectiva. Dialogflow es la columna vertebral de la lógica conversacional y el procesamiento de intenciones, lo que lo hace indispensable para proyectos que buscan desarrollar soluciones conversacionales complejas y altamente funcionales.

2.4 Arquitectura de Aplicaciones Web

Para la construcción de un prototipo funcional y eficiente, se explorarán arquitecturas de aplicaciones web modernas. El uso de tecnologías como contenedores, microservicios y arquitecturas sin servidor [12] permitirá una implementación escalable y fácil mantenimiento [13]. Además, se analizarán patrones de diseño específicos para aplicaciones judiciales, considerando la gestión de casos, la indexación de documentos y la integración con sistemas existentes. En el diseño de la arquitectura la escala de compromiso de usuario de O'Brien/Toms [14] proporciona una estructura valiosa para evaluar la efectividad de la búsqueda por voz en función de la percepción del usuario.

2.4.1 Flujo de diálogo en el Proceso de Búsqueda

Las interfaces informáticas tradicionales requieren entradas predecibles y estructuradas para funcionar de manera correcta, lo que hace que el uso de estas no sea natural. Si los usuarios finales no pueden comprender estas entradas estructuradas con facilidad, se les dificulta decidir qué hacer. Lo ideal es que las implementaciones puedan inferir lo que desean los usuarios finales, según el lenguaje natural que usen. Por ejemplo, considera una solicitud de usuario sencilla, como:

- “¿Cuál es el último expediente consultado?”
- “¿Qué Legajo ha sido creado el día de hoy?”
- “¿Quién ha actualizado el registro de persona con DNI XX.XXX.XXX?”

El agente no se orienta a conversaciones simples o a temas no relacionados con los Legajos Judiciales de la Provincia de Corrientes. Reconocerá la intención del usuario en este ámbito, pero no podrá recomendar restaurantes ni comentar sobre el pronóstico del tiempo para la próxima semana. Se detallan en la sección siguiente más detalles de esta integración.

4.4.2 Agente de Dialogflow

El principal motivo para diseñar un agente de este tipo radica en la necesidad de los usuarios de resolver problemas puntuales o consultas específicas sobre datos de registros almacenados en bases de datos. Cuanto más sencillo resulte para el usuario formular preguntas y obtener respuestas, mayor compromiso desarrollará con el agente, aumentando la probabilidad de su uso en futuras interacciones en vez de realizar búsquedas convencionales. La aplicación no estará disponible en la nube para discusiones sobre temas como el clima o restaurantes, ya que los sistemas operativos ya integran asistentes para estos temas diversos, basándose en conceptos generales de Dialogflow.

Intents

- **Frases de entrenamiento:** Los diseñadores deben crear frases de entrenamiento que abarquen una amplia gama de variaciones lingüísticas que los usuarios emplearían al buscar información sobre legajos judiciales, como “Necesito detalles del legajo [Número]” o “Muéstreme el historial del expediente [Número]”, donde [Número] identifica de manera única el legajo.
- **Acción y Parámetros:**
 - **Acción:** Deben definirse acciones claras como **consultarDetalleLegajo** o **obtenerHistorialExpediente**, vinculadas a funciones específicas del backend para extraer datos.
 - **Parámetros:** Es esencial configurar parámetros como **numeroLegajo** y **fechaConsulta** para capturar números de legajo y fechas específicas, parámetros esenciales como `numeroLegajo` (tipo de entidad `@sys.number` para capturar números de legajo) y `fechaConsulta` (tipo de entidad `@sys.date` para fechas de consulta específicas) respectivamente, utilizando el sistema de extracción de entidades de Dialogflow.
- **Respuestas:** Se deben implementar respuestas que integren los datos recuperados en mensajes, como “El estado actual del legajo [Número] es [Estado]”, rellenando los datos variables con información específica de la base de datos.

Entidades: Los desarrolladores crearán entidades personalizadas, como **TipoLegajo** con valores "Civil", "Penal", "Laboral", para clasificar las búsquedas eficientemente.

Contextos: Se manejan contextos para preservar el estado de la conversación, especialmente en diálogos complejos, estableciendo, por ejemplo, un contexto **seguimientoLegajo** tras mostrar detalles de un legajo.

Intents de respuesta: Se desarrollarán intents específicos de seguimiento para manejar preguntas o acciones basadas en la conversación actual, usando contextos para dirigir la conversación de manera coherente.

Consola de Dialogflow: Se utilizará la consola para implementar y probar iterativamente intents, entidades y contextos, asegurando que el modelo de lenguaje se ajuste basado en pruebas reales.

Interacciones del usuario con integraciones: Se definirán claramente las APIs necesarias para consultas y actualizaciones, incluyendo especificaciones sobre endpoints, métodos HTTP y mecanismos de autenticación.

Entregas para integraciones: Se documentará cómo estructurar las peticiones a la API desde Dialogflow y procesar las respuestas para crear mensajes coherentes al usuario, incluyendo la transformación de datos JSON y la gestión de errores.

Este enfoque activo refuerza la agencia de los desarrolladores y el sistema, enfocándose en la acción directa y la implementación de la interfaz conversacional.

2.5 Conocimiento disciplinar específico

2.5.1 Diseño de Aplicaciones

Se han aplicado conceptos y técnicas de diseño moderno para aplicaciones web, con un énfasis particular en la Experiencia UI/UX experiencia del usuario, Arquitectura de microservicios para determinados componentes como es la autenticación. Este enfoque divide la aplicación entre componentes de terceros para las interfaces de interacción y en servicios más pequeños para la sincronización de datos específicos de la implementación actual, lo que permite una mayor flexibilidad, facilidad de mantenimiento y despliegue. La arquitectura de microservicios es fundamental para soportar el desarrollo ágil y la entrega continua, permitiendo que cada servicio se desarrolle, pruebe, despliegue y escale de forma independiente. Y aún más relevante en este caso, las fases de Diseño de los flujos entre el usuario y el sistema.

2.5.2 Cloud Computing

La importancia de utilizar herramientas y técnicas de computación en la nube para el despliegue y gestión de aplicaciones. Se hace especial énfasis en la virtualización y la creación y lanzamiento de instancias, lo que permite a los desarrolladores y administradores de sistemas provisionar recursos de manera rápida y eficiente. La nube ofrece una plataforma escalable y flexible para alojar los microservicios, facilitando la gestión de cargas de trabajo y la adaptación a las demandas fluctuantes de acceso a la aplicación.

2.5.3 Auditoría

En la auditoría se examina la aplicación de estándares y herramientas específicas para asegurar la calidad y seguridad de aplicaciones web. La auditoría técnica involucra la revisión de la aplicación para identificar vulnerabilidades, asegurar la conformidad con los estándares de seguridad y mejorar el rendimiento y la estabilidad de la aplicación. Este proceso es crucial para minimizar los riesgos asociados con las operaciones en línea y garantizar una experiencia de usuario segura y confiable.

Estas áreas de conocimiento son cruciales para el desarrollo e implementación de una aplicación cliente Angular con microservicios. La combinación de un diseño de aplicaciones moderno, el aprovechamiento de la infraestructura de cloud computing y una auditoría rigurosa de la aplicación conforman la base para crear soluciones robustas, escalables y seguras que satisfagan las demandas de los usuarios finales y se alineen con los objetivos del proyecto

3. PLANIFICACIÓN Y DESARROLLO DEL DISEÑO

3.1 Definición del Problema

Este proyecto, viene a cubrir una de las necesidades básicas que tienen las personas con algún tipo de diversidad disfuncional. El objetivo del proyecto es crear una aplicación capaz de ayudar con la movilidad y la accesibilidad de todo usuario del sistema, como también agilizar mediante la modernización para una mejor gestión de búsquedas de legajos.

La aplicación en cuestión permite al usuario tener una conversación en lenguaje natural con un asistente conversacional, al que le podrá hacer preguntas acerca de la accesibilidad de un determinado legajo. La conversación se podrá realizar tanto por texto como por voz, teniendo también ambas posibilidades.

3.1.1 Recolección de Datos

Descripción de las funcionalidades básicas.

El sistema actual es un listado tradicional de registros paginados a través de una botonera en el margen inferior. Las búsquedas se realizan a través de filtros preestablecidos de las columnas, FECHA, NRO IURIX, CLASE, TIPO, ESTADO...etc. En la figura 3.1 se muestran los controles implementados para la búsqueda. Estos corresponden a datos de los registros; los cuales son pasibles de tipeado manual o selección de listas desplegables y con la posibilidad de reseteo de filtros seteados a través del botón “X”.

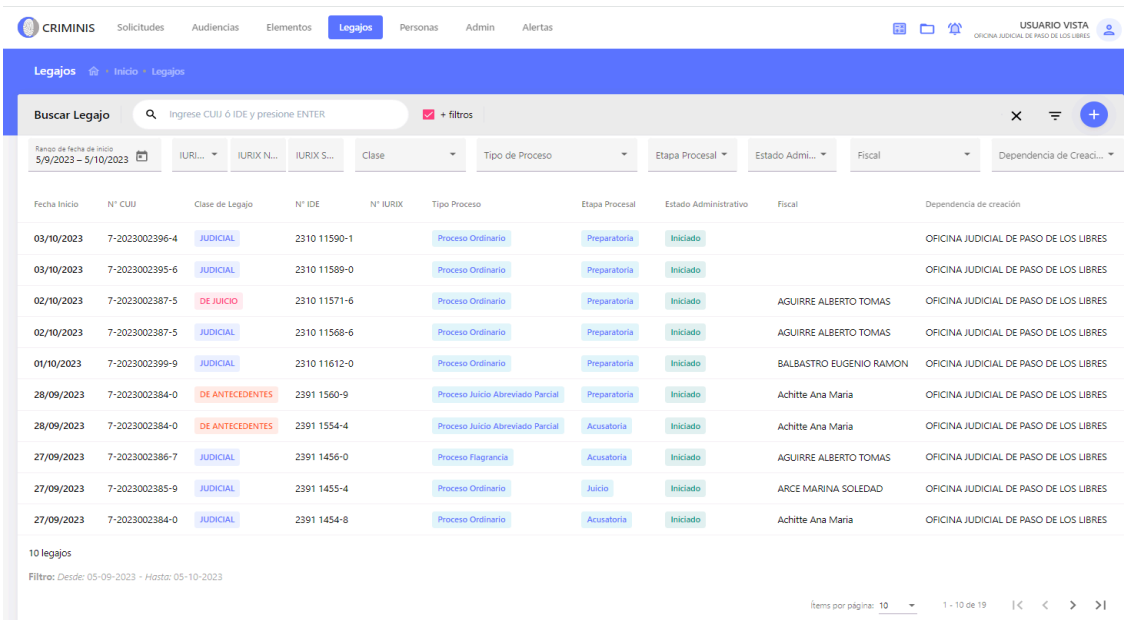


Fig 3.1 Descripción de las funcionalidades básicas

3.1.2 Análisis de la Situación Actual

Descripción de debilidades: La problemática radica en esta selección tediosa de filtros, sobre todo cuando son simultáneos. Por tanto, es simplemente una interfaz rígida y pasiva a la cual se puede interactuar en un solo sentido y de manera asíncrona sin asistencia al usuario. En base a esto, la Figura 3.2 corresponde a un diseño de la situación actual a analizar.



Fig 3.2 Análisis de la Situación Actual

Descripción de fortalezas: Desde el punto de vista del usuario tradicional, resulta intuitivo en base a las costumbres de uso de interfaces convencionales. El flujo de experiencia del usuario se grafica en la Figura 3.3

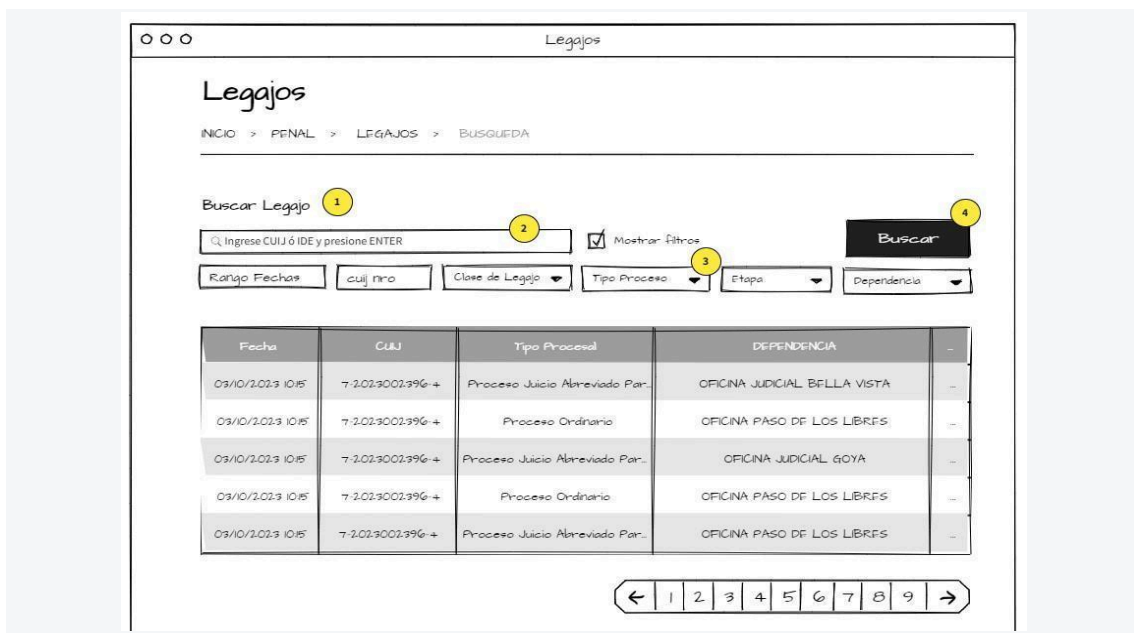


Fig 3.3 Análisis de puntos de acción débiles en el uso tradicional

3.2 Etapas de Desarrollo

Se mencionan las diferentes etapas y sus componentes involucrados para el proceso de diseño, para planificar y visualizar el cronograma de desarrollo del prototipo de interfaz conversacional, incluyendo fases como investigación, diseño, desarrollo, pruebas y lanzamiento.

3.2.1 Diagrama de Componentes

A continuación, procederemos a crear el Diagrama de Componentes para visualizar la Arquitectura del Sistema de la interfaz conversacional visto en la Figura 3.4.

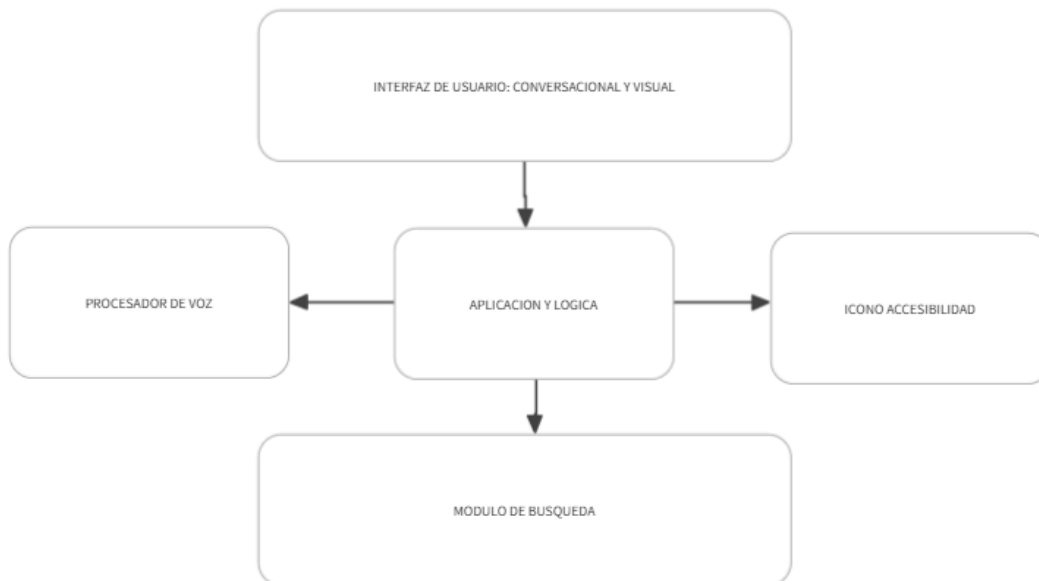


Fig 3.4 Diagrama de Componentes en el módulo

El Diagrama de Componentes presentado ilustra la Arquitectura del Sistema de la interfaz conversacional diseñada para la búsqueda de legajos judiciales. Este diagrama destaca la interacción entre los distintos componentes del sistema, incluyendo la interfaz de usuario que combina elementos conversacionales y visuales, el procesador de comandos de voz, el sistema de gestión de la base de datos de legajos, el módulo de búsqueda eficiente y la capa de accesibilidad, asegurando que el sistema sea accesible para usuarios con discapacidad. Se proporciona una base visual para discutir y presentar cómo tu propuesta de diseño de interfaz conversacional puede mejorar la satisfacción del usuario. Adaptarlos según las necesidades específicas de módulo del sistema para una presentación más detallada, y así enfocar aspectos particulares en la toma de decisiones en la fase de construcción del diseño.

3.3 Conversación entre agente y usuario

Los agentes conversacionales creados mediante servicios web hacen uso de la inteligencia artificial y del aprendizaje automático y progresivo según los usuarios realizan más consultas e interacciones con los agentes. Esta información es mostrada igualmente al desarrollador para que pueda realizar análisis y evaluaciones de su propio agente para ver cuál es el nivel de compromiso del usuario, los principales fallos del agente a la hora de entender las intenciones o las palabras clave más utilizadas. Donde el agente se va a representar mediante un iconos

sugestivos de MIC, vistos en la figura 3.5, con diferentes estados según su intervención en el sistema.



Fig 3.5 Iconos de Estados del componente

Para poder crear las intenciones y las entidades es importante conocer cuáles van a ser los objetivos que tenga el agente y el flujo que se va a seguir en las conversaciones ilustrado en la figura 3.6. En este caso, el flujo en que se va a producir la interacción entre el agente y el usuario será a través de la aplicación que será más tarde desarrollada.

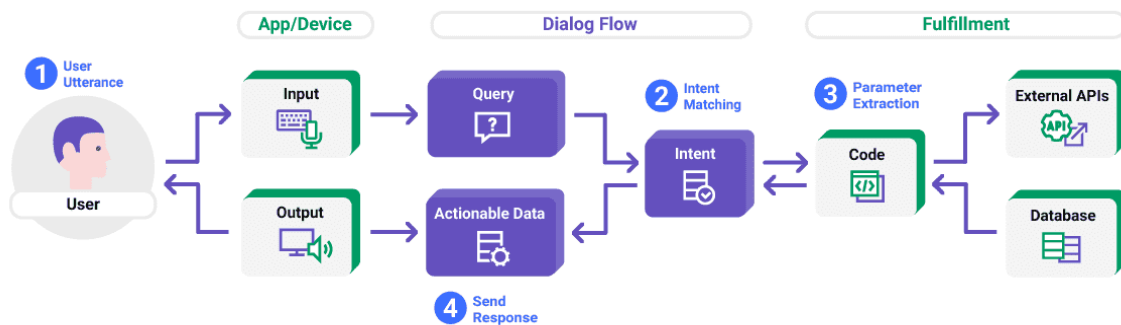


Fig. 3.6 Flujo de conversación entre el agente y el usuario

En la figura 3.6 anterior se puede comprobar como existen diferentes elementos dentro de la conversación: el agente, la aplicación y el usuario. Interaccionan entre ellos en diferentes pasos:

1. El usuario realiza la primera interacción y es el encargado de llamar al agente para realizarle la consulta.
2. La aplicación le envía la información al agente para que éste pueda recoger los datos, analizarlos y elaborar una respuesta.
3. El agente envía la información a la aplicación que será la encargada de presentarle esta información al usuario mediante texto o voz.
4. La aplicación le presenta finalmente al usuario la respuesta que ha llegado desde el agente creado en Dialogflow [15].

3.4 Diseño de la Interfaz

La aplicación mostrará la interacción entre ambas partes, siendo el agente que se encuentra por detrás, el encargado de que esa conversación se realice satisfactoriamente. Este proyecto, cuenta de varias partes que están interconectadas, y necesitan de la otra para poder funcionar

correctamente: un agente conversacional, y una aplicación, que albergue dicho asistente. En la era digital contemporánea, las interfaces de usuario han evolucionado para incorporar tecnologías avanzadas como el procesamiento del lenguaje natural. Una de esas aplicaciones innovadoras es la funcionalidad de búsqueda mediante voz. Este trabajo se propone analizar cómo la implementación de búsquedas por voz utilizando lenguaje natural puede impactar en la experiencia del usuario, tomando como base los seis factores de la escala de compromiso de usuario de O'Brien/Toms:

Percepción de Usabilidad: Con la integración de la voz, los usuarios pueden percibir un menor esfuerzo para realizar búsquedas, ya que simplemente expresan verbalmente lo que buscan. Esta facilidad puede influir positivamente en las emociones del usuario, en su capacidad para realizar tareas y en la percepción de la organización y navegación del sistema.

Estética: La búsqueda por voz podría requerir un diseño de interfaz diferente, donde los elementos visuales complementen la experiencia auditiva. La disposición de la pantalla y las imágenes/gráficos deberán ser armoniosos, provocando una impresión estética agradable y un atractivo sensorial.

Atención Focalizada: La capacidad de sumergirse en la experiencia de búsqueda por voz puede ser mayor, ya que se minimizan las distracciones visuales. Sin embargo, es fundamental analizar si el usuario es más consciente de lo que sucede fuera de la interacción y cómo percibe el paso del tiempo durante la búsqueda.

Durabilidad: Es esencial evaluar si las experiencias positivas con la búsqueda por voz son recordadas por los usuarios y si tienen el deseo de repetir la actividad, indicando una alta durabilidad de la experiencia.

Novedad: Dado que la búsqueda por voz aún es una tecnología emergente en muchos ámbitos, puede evocar curiosidad en los usuarios. Estudiar el interés y la novedad del sistema es crucial para entender su adopción y uso continuo.

Participación Sentida: La experiencia de búsqueda por voz debe ser divertida e inmersiva. Es esencial medir si el usuario se siente verdaderamente involucrado en la tarea y si califica la experiencia general como entretenida.

El desafío principal al implementar la búsqueda por voz es garantizar que no solo funcione técnicamente, sino que también ofrezca una experiencia de usuario enriquecedora. La escala de compromiso de usuario de O'Brien/Toms proporciona una estructura valiosa para evaluar la efectividad de la búsqueda por voz en función de la percepción del usuario. Dependiendo del medio que utilice el usuario para hacer la consulta, el asistente responderá a través del mismo medio, facilitando así una conversación más natural.

3.4.1 Proceso de Búsqueda de Legajos

Vamos a proceder a generar el Diagrama de Flujo: Proceso de Búsqueda de Legajos y el Diagrama de Componentes: Arquitectura del Sistema. Comenzaremos con el Diagrama de Flujo para el proceso de búsqueda de legajos

3.4.1.1 Diagrama de Flujo de Búsqueda de Legajos

Aquí tienes un Diagrama de Flujo que ilustra el Proceso de Búsqueda de Legajos utilizando la interfaz conversacional. Este diagrama muestra los pasos desde que el usuario inicia la

búsqueda hasta que accede al legajo deseado, destacando la importancia del comando de voz y el procesamiento del mismo para obtener resultados eficientes.

1. Inicio: Usuario inicia la búsqueda.
2. Comando de Voz: Usuario emite un comando de voz para buscar un legajo específico.
3. Procesamiento del Comando: El sistema procesa el comando y realiza la búsqueda.
4. Resultados: El sistema muestra los resultados de la búsqueda.
5. Fin: Usuario accede al legajo deseado.

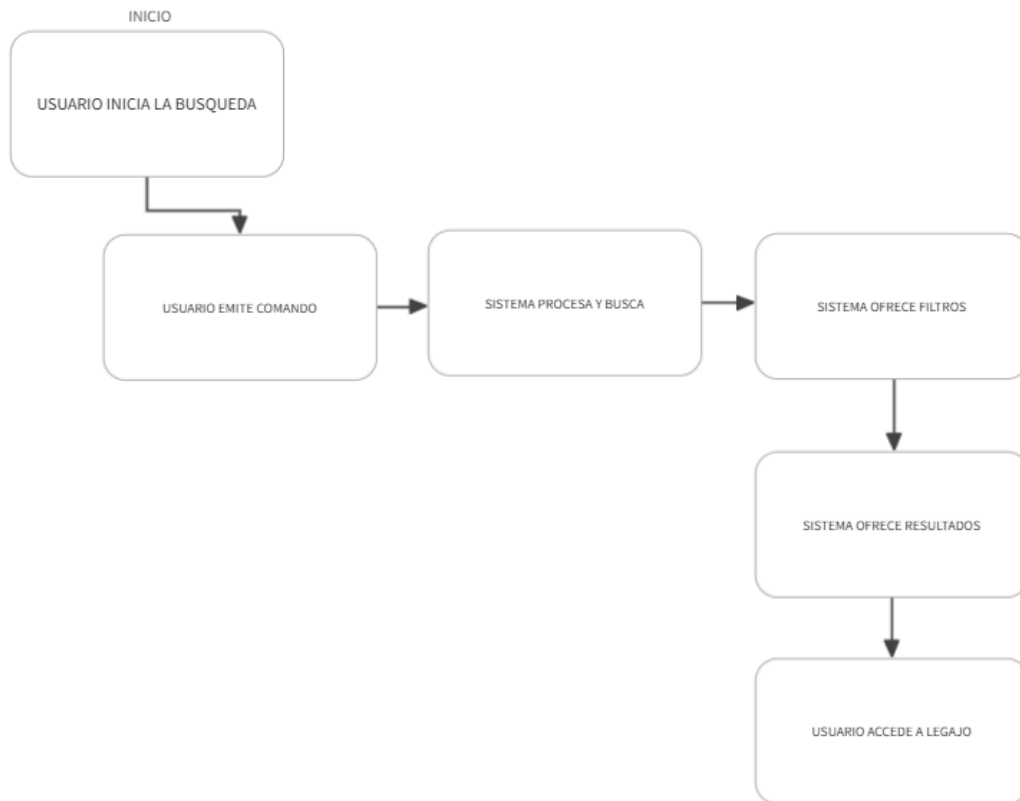


Fig. 3.7 Diagrama de Flujo de Búsqueda de Legajos

Este flujo conversacional, visto en la figura 3.7 se repetirá tantas veces como sea necesario hasta que el usuario tenga respuestas a sus dudas. Por su parte, el agente le realizará tantas preguntas como sean necesarias al usuario para que así pueda comprender totalmente sus intenciones y reconocer las entidades o palabras clave con las que realizar las consultas a las bases de datos, servidores y API y poder responder coherentemente y con información personalizada al usuario con el que está realizando la conversación.

Tener la posibilidad de escribir o hablar, y de poder leer o escuchar, permite que la aplicación en sí sea accesible a un gran público, y facilita la tarea a la hora de encontrar información sobre la accesibilidad. Por tanto, la aplicación debe ser intuitiva y facilitar la interacción en todo momento.

3.4.1.2 Presentación de la Interfaz

En la vista de gestión donde se accede al listado de legajos con sus respectivos filtros, se dispone un icono "MIC" flotante que sugiere introducir comandos de voz como se observa en la Figura 3.7. Este icono responderá mediante una frase clave "HOLA CRIMINIS", definida en las

secciones siguientes y reconocidas por el sistema a través del asistente (siempre y cuando el navegador en el que se ejecute tenga los permisos concedidos de uso de introducción de voz mediante el dispositivo); como también mediante el mouse de forma tradicional, y así comenzar el flujo de la Interfaz Conversacional. Donde se observa el diseño inicial en figura 3.8

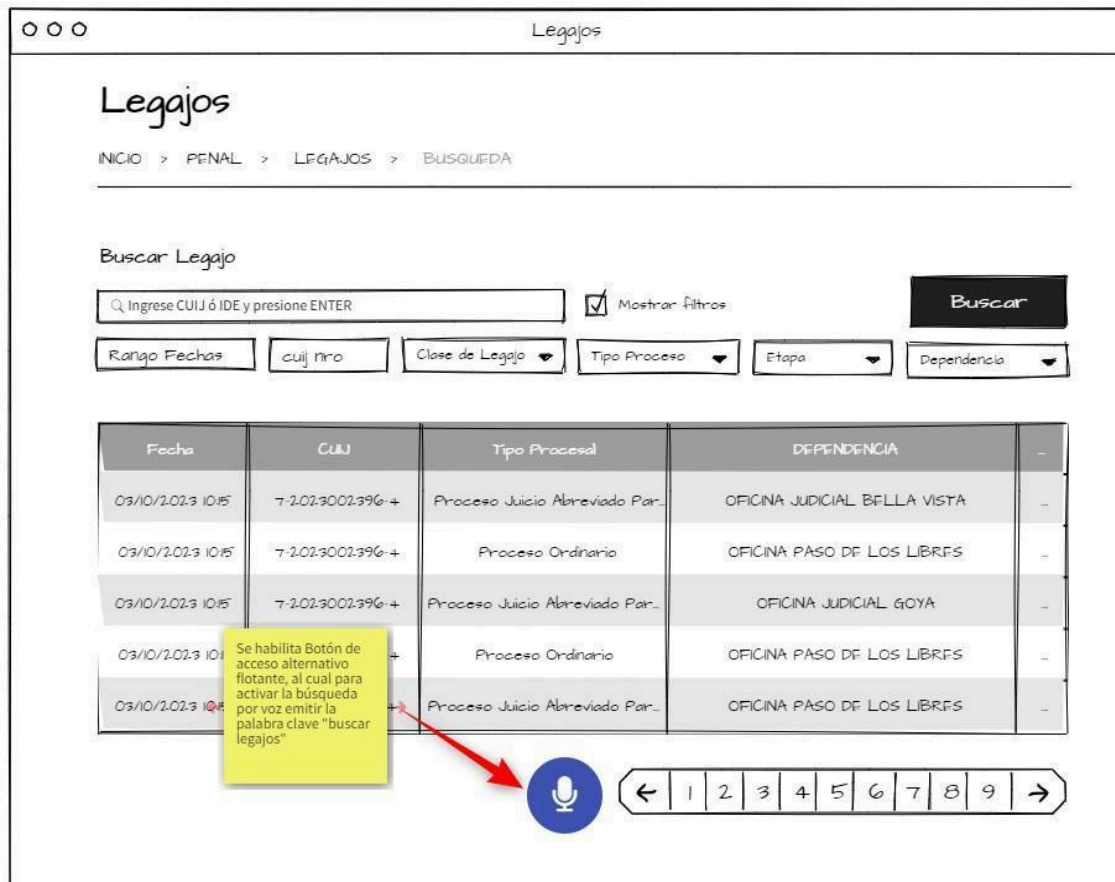


Fig 3.8 Icono de Introducción de Voz

3.4.1.3 Diálogo de la Orden

Por consiguiente al reconocimiento de la frase clave inicializadora del componente que intervendrá como Asistente conversacional, se desplegará un diálogo en forma de Modal o Ventana Emergente, donde mostrará el primero de tres pasos para llegar y completar el flujo, en este paso del asistente, se definirá una serie de filtros preestablecidos opcionales, por lo que se dispondrá e informará al usuario una leyenda con esta opción, la palabra clave "Filtrar" es la que la interfaz deberá detectar para procesarla y continuar. Como observamos en la Figura 3.9 Diálogo de la Orden

Legajos

INICIO > PENAL > LEGAJOS > BUSQUEDA

Buscar Legajo



Fig 3.9 Diseño de Interfaz. Diálogo de la Orden

3.4.1.4 Selección de Filtros

Una vez que la opción la palabra clave “Filtrar” es proporcionada la interfaz detecta y procesa, se ejemplifica con la Figura 3.9 Diálogo de filtrado de la Orden, se pasa al siguiente paso 2 del diálogo proporcionando al usuario la opción de seleccionar los filtros predefinidos; el cual el asistente en este caso deberá detectar las palabras claves de la intención que setea los campos de registro que coinciden con la información introducida en el siguiente paso.

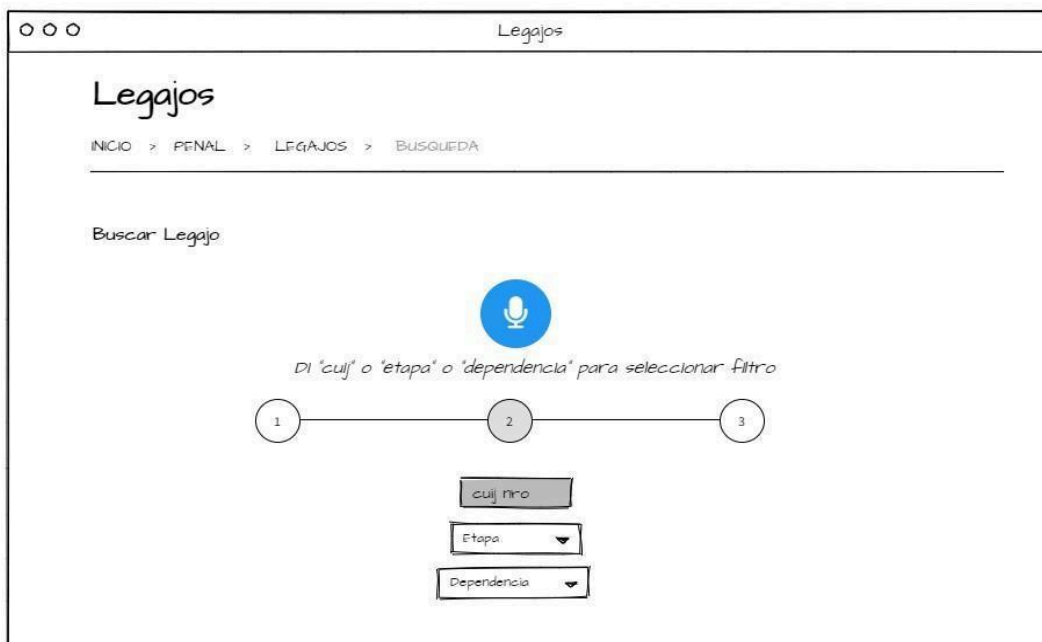


Fig 3.10 Diseño de Interfaz. Diálogo de filtrado de la Orden

3.4.1.5 Interfaz de Resultados

Para culminar y procesar la búsqueda, se proporciona al usuario el paso 3 en el cual el asistente estará en estado de “escucha”, la información que se introduzca mediante lenguaje natural disparará los algoritmos de la aplicación y procesará los registros coincidentes para por último, como se visualiza en Fig 3.11 del diseño de interfaz. Se arroja un listado de resultados

obtenidos, que en el caso de éxito, desplegará opciones sobre estos. Caso contrario, la interfaz mostrará una leyenda “Sin resultados obtenidos”

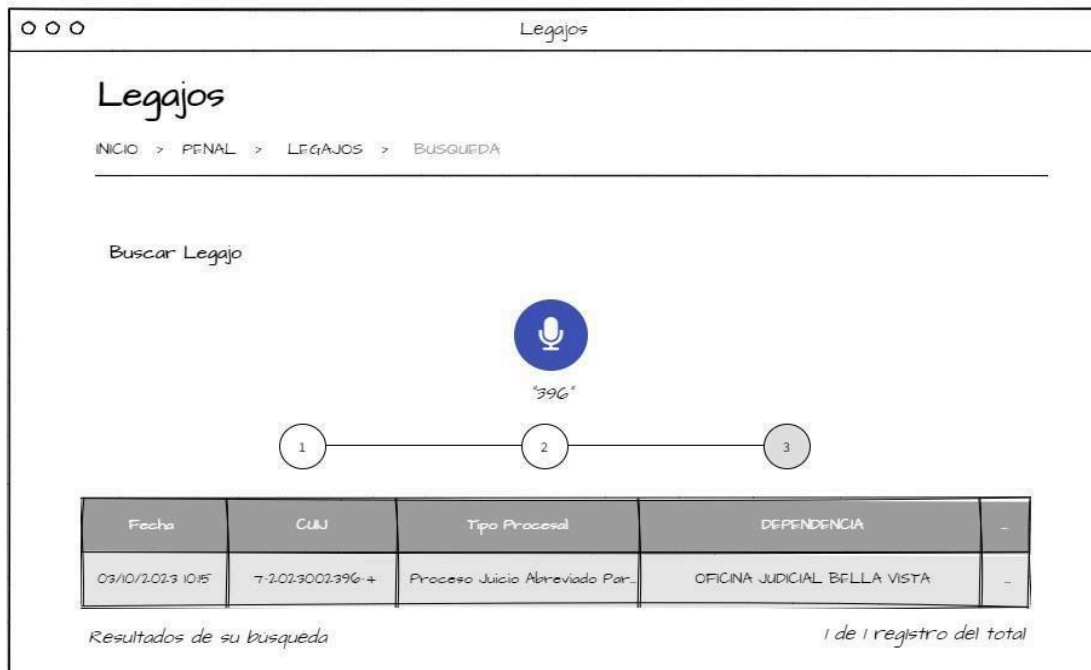


Fig 3.11 Diseño de Interfaz. Listado de Resultados obtenidos

3.4.2 Acción sobre los resultados obtenidos

Luego y por último siguiendo el flujo en paso 3. Y en el caso de éxito, con el listado de registros coincidentes desplegados de la búsqueda. Se podrán tomar acciones de selección y redirección, mediante el estado de escucha activa en este caso se deberá detectar el comando de voz con la frase clave “Redirige al Legajo” como se grafica en la Figura 3.11, se diferencia la asignación de color al icono del Asistente en estado de escucha de Acción del resultado a un color rojo indicando esa acción de redirección como se refleja en la Figura 3.12, en este punto del flujo el usuario finaliza el uso de la Interfaz conversacional llegando al resultado deseado, el cual es la vista de detalle del Legajo con todas las funcionalidades originalmente implementadas Figura 3.13 Vista de Detalle de Legajo Judicial.

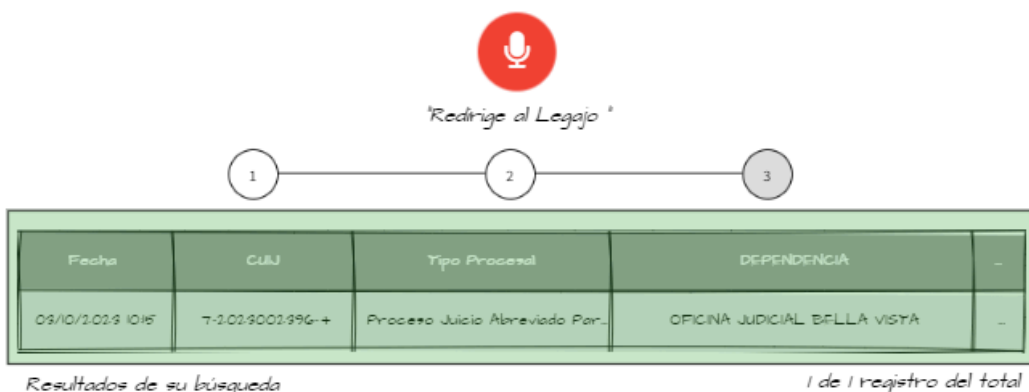


Fig 3.11 Asistente en estado de escucha de Acción del resultado

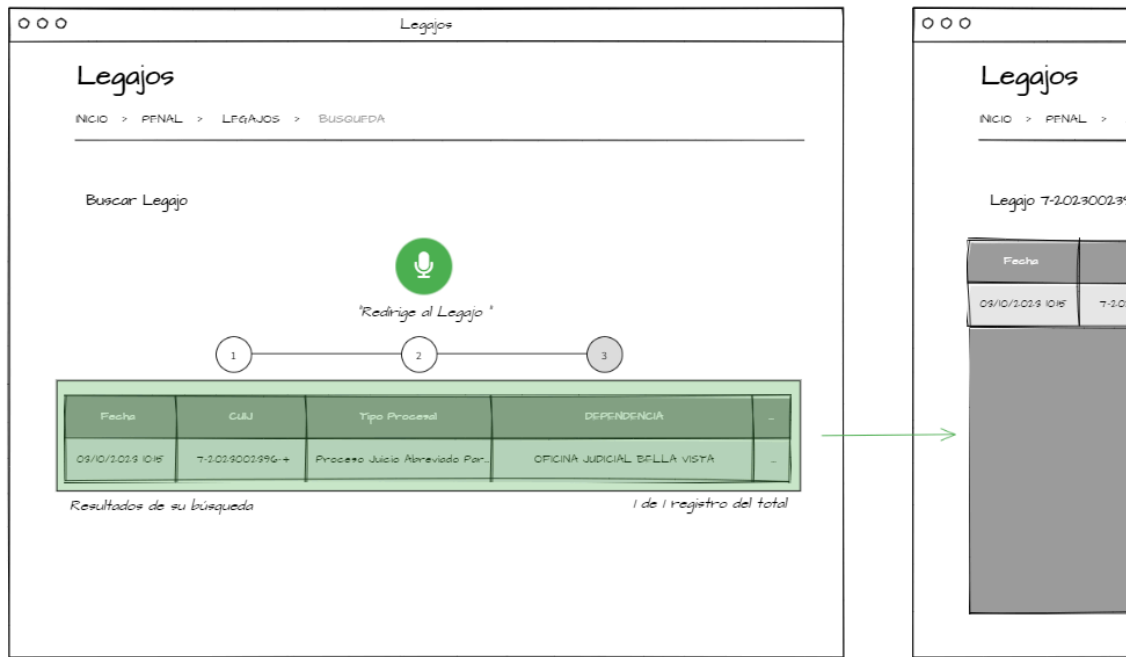


Fig 3.12 Diseño de Interfaz. Flujo de redirección

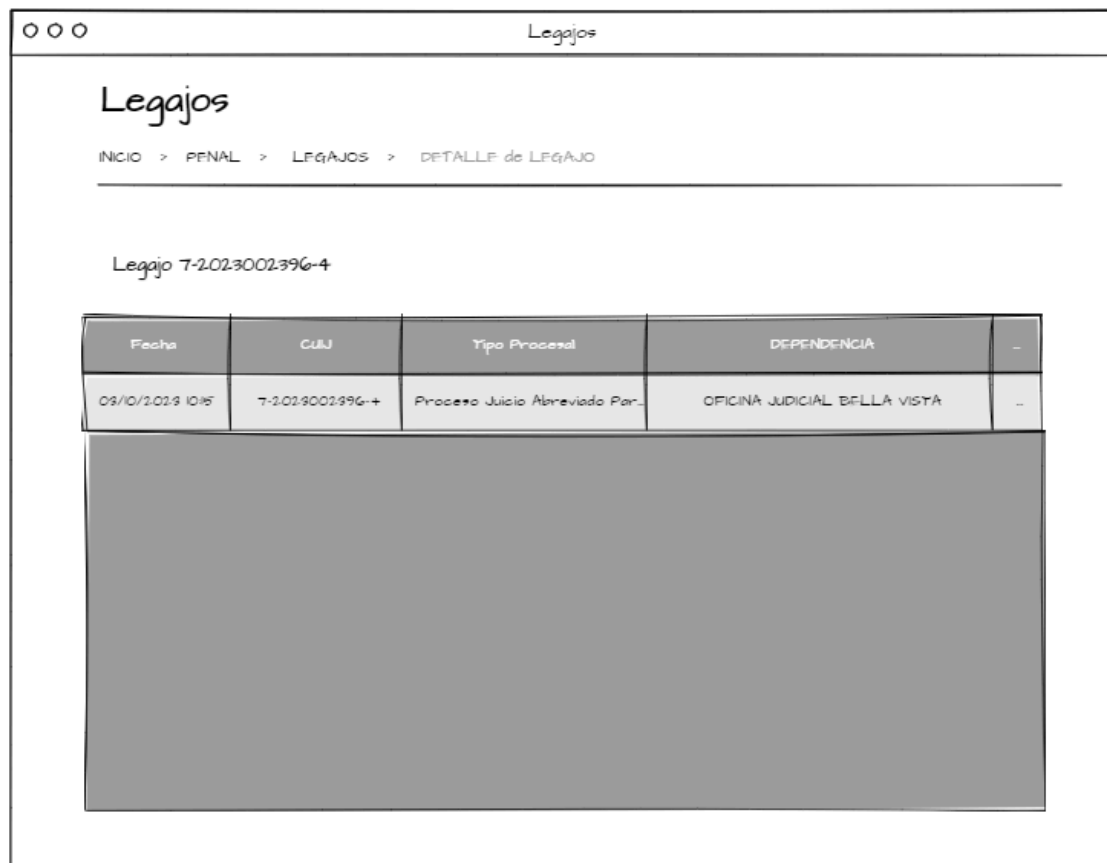


Fig 3.13 Vista de Detalle de Legajo Judicial

3.4.3 Análisis de Soluciones Propuestas

Las interfaces conversacionales de voz, o asistentes virtuales, han ganado popularidad en diversos ámbitos debido a su capacidad para proporcionar interacciones más naturales. Esta evaluación se centra en la viabilidad y eficacia de incorporar una interfaz de voz a un sistema de búsqueda tradicional basado en formularios.

Categoría	Métrica	Meta / Estimado	Resultado / Real
Viabilidad	Tiempo de Implementación	3 meses	2.8 meses
	Costo de Integración	\$15,000	\$13,500
	Tasa de Errores de Reconocimiento de Voz	< 5%	3.5%
Eficacia	Velocidad de Búsqueda (Comparativa)	Formulario: 40s Voz: -	Formulario: - Voz: 25s
	Precisión de los Resultados de Búsqueda	Formulario: 90% Voz: -	Formulario: - Voz: 88%
	Número de Usuarios que Adoptan la Función de Voz	60%	65%
Impactos Secundarios	Consultas sobre Privacidad desde la Implementación	Antes: 2/mes Después: -	Antes: - Después: 10/mes
	Sesiones de Capacitación Ofrecidas vs. Asistidas	Ofrecidas: 5 Asistidas: -	Ofrecidas: - Asistidas: 80%
	Frecuencia de Actualizaciones de la Interfaz de Voz	6 meses: 3 Problemas solucionados: -	6 meses: - Problemas solucionados: 15
	Satisfacción del Usuario	Voz: 85% Formulario: 15%	-

Tabla 3.1 Tabla de Métricas de Viabilidad

Estas métricas ofrecen una visión cuantitativa de la implementación y permiten a los gestores tomar decisiones basadas en datos. Es esencial revisar y ajustar estas métricas con regularidad para reflejar las necesidades cambiantes y las condiciones del sistema y de los usuarios.

3.5 Metodología e Iteraciones

En la elaboración de una aplicación web para el ámbito judicial, la selección de una metodología adecuada es crucial para el éxito del proyecto. Por esto, se ha seleccionado la metodología Scrum, conocida por su adaptabilidad y enfoque iterativo, que es particularmente útil en entornos donde los requisitos pueden cambiar con frecuencia y la colaboración interdisciplinaria es esencial. La investigación de Thomas y Bindu (2022) destaca el crecimiento

de las metodologías ágiles de desarrollo de software en escenarios de aplicaciones web, señalando que las metodologías tradicionales enfrentan la rigidez y dificultad en la adaptación a requerimientos cambiantes, mientras que las ágiles, como Scrum, innovan significativamente respecto a los enfoques tradicionales planificados [16].

3.5.1 Selección de la Metodología

Se ha optado por la metodología Scrum para el desarrollo de este prototipo de aplicación web, se fundamenta en diversos factores cruciales para el éxito del proyecto. La decisión de adoptar Scrum se basa en su alineación con las necesidades específicas y dinámicas del entorno judicial. Scrum es reconocido por su adaptabilidad a los cambios, una característica esencial en un dominio donde las regulaciones pueden variar con frecuencia.

Schwaber y Sutherland resaltan esta flexibilidad en su guía oficial de Scrum, identificándose como una fortaleza clave de la metodología. Además, la colaboración interdisciplinaria que promueve Scrum es crítica en el entorno judicial donde la participación de expertos legales es imprescindible para la correcta interpretación de los requerimientos y la validación de las soluciones propuestas. El enfoque incremental de Scrum, con su sistema de Sprints y revisiones regulares, garantiza una entrega continua de valor y la oportunidad de realizar ajustes basados en la retroalimentación real de los usuarios. Esta metodología también es favorable para mantener un alto estándar de calidad y transparencia, ya que cada iteración está sujeta a evaluaciones y ajustes constantes. La gestión efectiva del tiempo y los recursos al argumentar que Scrum facilita una mejor distribución de la carga de trabajo y una planificación más realista.

En resumen, la elección de Scrum para el desarrollo de este prototipo de aplicación web en el ámbito judicial se basa en su capacidad para adaptarse a cambios, fomentar la colaboración interdisciplinaria, ofrecer entregas incrementales de valor, enfocarse en la calidad y transparencia, y gestionar eficientemente el tiempo y los recursos. Estas características son esenciales para el éxito de un proyecto tan especializado y sujeto a las dinámicas exigentes del sistema judicial.

3.5.2 Iteraciones

Las iteraciones dentro del marco de Scrum se dividen en Sprints, cada uno con una fase de planificación, ejecución y revisión. En el primer Sprint, el equipo se centra en establecer los requisitos y diseñar una interfaz conversacional intuitiva, asegurando que se cumplan las necesidades de búsqueda de legajos judiciales de forma eficiente (Rubin, 2012). La implementación inicial del reconocimiento de voz y la lógica de búsqueda son esenciales para proporcionar una base sólida sobre la cual construir. El segundo Sprint se dedica a la mejora y expansión de la funcionalidad, basándose en la retroalimentación obtenida de los stakeholders durante la reunión de revisión del Sprint anterior. Esta fase es crucial para refinar la experiencia del usuario y mejorar la precisión del reconocimiento de voz, utilizando técnicas avanzadas de PLN como las descritas por Jurafsky y Martin (2019) en su libro sobre el procesamiento del lenguaje y la comunicación humana.

Las pruebas y los ajustes finales se realizan para garantizar que la aplicación no solo sea funcional sino también segura, cumpliendo con los estándares de seguridad y privacidad necesarios en el tratamiento de información sensible judicial (Mouratidis y Giorgini, 2007).

3.5.2.1 Iteración 1: Planificación y Diseño del Desarrollo

Reunión de Planificación (Sprint Planning):

- o Identificar los requisitos principales: Búsqueda de legajos judiciales por nombre, número de caso o fecha.
- o Definir historias de usuario: "Como usuario, quiero poder buscar legajos judiciales por [nombre/número de caso/fecha] para acceder rápidamente a la información que necesito."
- o Establecer tareas para la implementación de cada historia.

Diseño de la Interfaz Conversacional:

- o Crear un diseño de interfaz de voz intuitiva y amigable.
- o Establecer comandos de voz claros y comprensibles para realizar las búsquedas.

Desarrollo de la Funcionalidad Básica:

- o Implementar el reconocimiento de voz.
- o Desarrollar la lógica de búsqueda de legajos según los comandos definidos.
- o Integrar una respuesta vocal que proporcione la información solicitada.

3.5.2.2 Iteración 2: Mejora y Expansión

Reunión de Revisión (Sprint Review):

- o Demostrar la funcionalidad básica a los stakeholders.
- o Recopilar feedback sobre la usabilidad y la precisión de la búsqueda.

Refinamiento de la Interfaz:

- o Mejorar la interfaz de usuario en respuesta al feedback.
- o Añadir comandos adicionales para una experiencia más completa.

Optimización del Reconocimiento de Voz:

- o Refinar el motor de reconocimiento de voz para mejorar la precisión.
- o Implementar técnicas de procesamiento de lenguaje natural para comprender mejor los comandos.

Implementación de Seguridad:

- o Añadir medidas de seguridad para garantizar la privacidad y la integridad de la información judicial.
- o Cumplir con los estándares y regulaciones de seguridad aplicables.

Pruebas y Ajustes Finales:

- o Realizar pruebas exhaustivas de la funcionalidad de búsqueda por voz.
- o Realizar ajustes finales en base a los resultados de las pruebas.

3.6 Desarrollo del Prototipo

Después de la segunda iteración, el módulo de aplicación estará listo para su implementación en el entorno judicial. Se habrá mejorado la interfaz, la precisión del reconocimiento de voz y se habrán implementado medidas de seguridad para garantizar la confidencialidad de la información, como también toda la documentación de diseños, diagramas de flujo, pantallas y manuales de uso para los usuarios. Se detallan a continuación el desarrollo del prototipo de las herramientas tecnológicas

3.6.1 Configuración del Entorno de Desarrollo

Para garantizar la consistencia y la calidad del desarrollo, es esencial establecer un entorno de desarrollo adecuado. Los siguientes pasos detallan la configuración requerida:

3.6.1.1 Requerimientos del Sistema

- Sistema Operativo: Windows 10/11, MacOS 10.15+, o Linux (Ubuntu 20.04+, CentOS 8+, etc.)
- Node.js: Versión 14.x o superior.
- NPM (Node Package Manager): Versión 6.x o superior.

3.6.1.2 Instalación de Angular CLI

- **Angular CLI:** Versión 12.x. Para instalar Angular CLI, que es necesario para la creación y gestión del proyecto Angular, ejecute el siguiente comando en la terminal:

```
JavaScript  
npm install -g @angular/cli@12
```

3.6.1.3 Creación del Proyecto

Para iniciar un nuevo proyecto Angular, utilice el comando:

```
JavaScript  
ng new buscador-voz
```

Al ejecutar este comando, Angular CLI solicitará configuraciones adicionales como el estilo de hojas de estilo (CSS, SCSS, etc.).

3.6.2 Implementación de la Interfaz Conversacional

3.6.2.1 Integración con Web Speech API. La aplicación debe ser compatible con la Web Speech API para el reconocimiento de voz. La compatibilidad del navegador y las pruebas de microfonía son esenciales para una experiencia de usuario fluida.

- **Compatibilidad de Navegadores:** Lista de navegadores compatibles con la Web Speech API (Chrome, Firefox, Safari, Edge).
- **Pruebas de Microfonía:**
 - Verificación de la compatibilidad de micrófonos integrados y externos.
 - Uso de la API para acceder al micrófono:

```
JavaScript
if ('webkitSpeechRecognition' in window) { // API soportada } else { // API
no soportada }
```

3.6.2.2 Desarrollo con TypeScript y Angular

El desarrollo se realiza utilizando TypeScript y Angular. Se configurará tsconfig.json para compilar ES6+ y se crearán componentes Angular específicos para la interfaz conversacional.

- TypeScript: Versión 4.x recomendada.
 - Configuración del tsconfig.json para compilar ES6+ y tipos para la Web Speech API.
- Componentes Angular:
 - Creación de componentes para la interfaz conversacional.

```
JavaScript
ng generate component webspeech-component
```

Este componente es esencial para procesar la voz introducida; básicamente se inicia con la configuración;

```
JavaScript
ngOnInit(): void {
  const webSpeechReady =
this.speechRecognizer.initialize(this.currentLanguage);
  if (webSpeechReady) {
    this.initRecognition();
  }else {
    this.errorMessage$ = of('Your Browser is not supported. Please try
Google Chrome.');
```

```

this.positionFilter.valueChanges.subscribe((positionFilterValue) => {
  this.filteredValues['position'] = positionFilterValue;
  this.dataSource.filter = JSON.stringify(this.filteredValues);
});

this.nameFilter.valueChanges.subscribe((nameFilterValue) => {
  this.filteredValues['name'] = nameFilterValue;
  this.dataSource.filter = JSON.stringify(this.filteredValues);
});

this.dataSource.filterPredicate = this.customFilterPredicate();
}

```

A su vez la secuencia de métodos para controlar el flujo de la interfaz conversacional:

```

JavaScript
start(): void {
  if (this.speechRecognizer.isListening) {
    this.stop();
    return;
  }

  this.defaultError$.next(undefined);
  this.speechRecognizer.start();
}

stop(): void {
  this.speechRecognizer.stop();
}

selectLanguage(language: string): void {
  if (this.speechRecognizer.isListening) {
    this.stop();
  }
  this.currentLanguage = language;
  this.speechRecognizer.setLanguage(this.currentLanguage);
}

private initRecognition(): void {
  this.transcript$ = this.speechRecognizer.onResult().pipe(
    tap((notification) => {
      this.processNotification(notification);
    }),
    map((notification) => notification.content || '')
  );

  this.listening$ = merge(

```

```

    this.speechRecognizer.onStart(),
    this.speechRecognizer.onEnd()
  ).pipe(map((notification) => notification.event === SpeechEvent.Start));

  this.errorMessage$ = merge(
    this.speechRecognizer.onError(),
    this.defaultError$
  ).pipe(
    map((data) => {
      if (data === undefined) {
        return '';
      }
      if (typeof data === 'string') {
        return data;
      }
      let message;
      switch (data.error) {
        case SpeechError.NotAllowed:
          message = `Cannot run the demo. Your browser is not authorized to
access your microphone.
          Verify that your browser has access to your microphone and
try again.`;
          break;
        case SpeechError.NoSpeech:
          message = `No speech has been detected. Please try again.`;
          break;
        case SpeechError.AudioCapture:
          message = `Microphone is not available. Please verify the connection
of your microphone and try again.`;
          break;
        default:
          message = '';
          break;
      }
      return message;
    })
  );
}

```

Instrucciones detalladas para las pruebas de micrófonos y dispositivos de entrada de audio asegurará una captura de sonido adecuada.

3.6.3 Requerimientos Técnicos y Compatibilidades

3.6.3.1 Según la documentación de las herramientas tecnológicas, se han listado en la Tabla de Requerimientos con sus especificaciones en tabla 3.3, los siguientes:

Requerimiento	Especificación
Sistema Operativo	Windows, MacOS, Linux
Node.js	>= 14.x
NPM	>= 6.x
Angular CLI	12.x
TypeScript	4.x
Navegadores Compatibles	Chrome, Firefox, Safari, Edge
Compatibilidad de Micrófonos	Integrados y Externos

Tabla 3.3 Requerimientos Mínimos de Aplicación

3.6.3.2 Compatibilidad con Micrófonos y Dispositivos de Entrada de Audio

- Pruebas de Funcionalidad: Instrucciones para probar la funcionalidad de entrada de audio en diferentes dispositivos.
- Mejores Prácticas: Recomendaciones para asegurar una captura de audio clara y sin interferencias.

3.6.4 Despliegue y Pruebas

3.6.4.1 Despliegue en Servidores. Los desarrolladores deben seguir las instrucciones para desplegar la aplicación en servidores de aplicaciones como Firebase, AWS o Netlify utilizando comandos como:

- Servidores de Aplicaciones: Instrucciones para desplegar en servidores como Firebase, AWS, o Netlify.

```
JavaScript
ng deploy --base-href=/nombre-del-proyecto/
```

3.6.4.2 Automatización de Pruebas

Las pruebas automatizadas se configurarán utilizando herramientas como Jasmine, Karma y Protractor para garantizar la calidad y el funcionamiento correcto del sistema:

Jasmine y Karma: Configuración para pruebas unitarias y de integración.

```
JavaScript
ng test
```

Protractor: Configuración para pruebas de extremo a extremo (e2e).

```
JavaScript
ng e2e
```

Esta subsección proporciona una guía técnica esencial que facilita a los desarrolladores la instalación y configuración del entorno necesario para el desarrollo del prototipo utilizando Angular, asegurando la preparación para su implementación efectiva en el entorno judicial.

3.6.5 Documentación y Mantenimiento

3.6.5.1 Documentación del Proyecto

Para asegurar la sostenibilidad y facilidad de uso del prototipo de aplicación, la documentación debe ser integral y técnica:

- **README.md:**
 - **Objetivo del Proyecto:** Descripción detallada de la funcionalidad específica de la aplicación judicial, incluyendo su propósito dentro del entorno judicial.
 - **Configuración del Entorno:** Pasos detallados para la configuración del entorno de desarrollo, incluyendo la instalación de Node.js, Angular CLI, y cualquier otra dependencia específica del proyecto.
 - **Ejecución del Proyecto:** Comandos para ejecutar la aplicación localmente, incluyendo **ng serve** para el servidor de desarrollo y cómo acceder a la aplicación a través del navegador.

- **Pruebas:** Instrucciones para ejecutar las pruebas automatizadas con Jasmine, Karma y Protractor, explicando cómo interpretar los resultados y cómo agregar nuevas pruebas.
- **Despliegue:** Guía para desplegar la aplicación en servidores de producción, cubriendo pasos específicos y comandos de despliegue para diferentes entornos como Firebase, AWS o Netlify.
- **Contribución:** Directrices para contribuir al proyecto, incluyendo estándares de codificación, procesos de revisión de código y cómo crear pull requests.
- **Documentación Técnica:**
 - **Compodoc:** Configuración de Compodoc para generar y servir la documentación del código fuente de Angular. Incluir un script en **package.json** para facilitar la generación de la documentación, como "**compodoc**": "**compodoc -p tsconfig.json -s**".
 - **Comentarios en Código Fuente:** Establecer una convención de comentarios para asegurar que todos los servicios, componentes y clases estén adecuadamente comentados para la generación automática de la documentación.
 - **Diagramas de Arquitectura y Flujo:** Incorporación de diagramas UML y de flujo de trabajo para visualizar la estructura y los procesos de la aplicación. Estos pueden ser generados manualmente o utilizando herramientas como Lucidchart o Draw.io, y deberán ser incluidos en la documentación técnica.

3.6.5.2 Actualizaciones y Mejoras Continuas

El prototipo debe mantenerse actualizado y alineado con las necesidades de los usuarios y los estándares de seguridad:

- **Gestión de Versiones:**
 - **Git y Ramas de Características:** Uso de ramas de características para el desarrollo de nuevas funcionalidades o mejoras, fusionándose con la rama principal mediante pull requests una vez que se hayan revisado y probado.
 - **Etiquetas y Lanzamientos:** Utilizar etiquetas en Git para marcar las versiones de lanzamiento, siguiendo un esquema de versionado semántico (por ejemplo, **v1.2.3**).
- **Feedback de Usuarios:**
 - **Sistema de Reporte de Errores:** Integración de un sistema de seguimiento de errores como JIRA o GitHub Issues para que los usuarios finales puedan reportar problemas.
 - **Sesiones de Pruebas con Usuarios:** Organizar sesiones de prueba con los usuarios finales para obtener retroalimentación cualitativa y cuantitativa, que luego se traducirá en mejoras en el backlog del proyecto.

El mantenimiento de la aplicación también debe incluir la actualización de dependencias y la revisión de la base de código para cumplir con los estándares de seguridad y las mejores prácticas de desarrollo actuales. Además, es fundamental establecer un plan de acción para el manejo de la deuda técnica y la obsolescencia tecnológica.

- Con estos detalles, la sección de documentación y mantenimiento proporciona una guía técnica específica y relevante para desarrolladores, mantenedores y usuarios, asegurando que la aplicación se mantenga robusta, segura y relevante con el tiempo.

4. INTEGRACIONES

En este capítulo, se describen los diversos componentes del proyecto, que incluyen tanto el agente conversacional como la aplicación que integrará el asistente, así como sus respectivas funcionalidades. Se proporciona una explicación detallada de las herramientas utilizadas en el desarrollo de esta interfaz.

4.1 Identificación módulos factibles de aplicación

1. **Entrada de Texto por Parte del Usuario:** La aplicación permite a los usuarios ingresar consultas o responder preguntas del asistente mediante texto. Un campo de texto está disponible para que los usuarios puedan interactuar por escrito durante las diversas etapas de la conversación, hasta llegar a la consulta final.
2. **Entrada de Voz por Parte del Usuario:** Para mayor comodidad y eficacia de los usuarios que prefieren la comunicación por voz, la aplicación admite la entrada de voz. Utilizando el módulo de Speech-to-Text, la aplicación convierte las consultas de voz en texto para que el agente pueda analizarlas y procesarlas.
3. **Gestión de Respuestas del Asistente:** Cada vez que el usuario interactúa con el asistente, recibe una respuesta en forma de texto proveniente del agente conversacional. La aplicación se encarga de identificar el medio de comunicación preferido por el usuario y presenta la respuesta en forma de listado de coincidencias. En caso de comunicación por voz, la aplicación utiliza el módulo Text-to-Speech.
4. **Localización de Consultas:** Una vez que el usuario recibe información sobre la accesibilidad de un legajo particular, si la respuesta es positiva, la aplicación permite al usuario procesar y listar con precisión en un listado todas las coincidencias. Esto facilita a los usuarios tomar acciones sobre estos.
5. **Redirección de Resultados Obtenidos:** Los usuarios pueden seleccionar y tomar acciones de lo que han consultado. Esto les permite acceder fácilmente a la información relacionada con la accesibilidad.
6. **Redirección de Resultados Obtenidos:** La aplicación proporciona una funcionalidad esencial al permitir a los usuarios redireccionen a los detalles de las coincidencias. El cual cierra el ciclo del flujo de sus búsquedas.

El agente conversacional desempeña un papel central en la aplicación y guía a los usuarios a través de preguntas para que realicen consultas específicas con parámetros que se utilizan para realizar consultas a las API. Sus principales funciones incluyen:

Modelado del Agente Orientado al Proyecto: El desarrollador del agente orienta la inteligencia artificial del agente conversacional para identificar la intención del usuario, guiar la conversación y generar respuestas coherentes con las consultas del usuario.

Identificación de Entidades: El agente reconoce entidades o elementos clave en las consultas del usuario, lo que le permite generar respuestas coherentes y consultar las API para obtener información adicional.

Reconocimiento de la Intención del Usuario: Dado que los usuarios pueden realizar diferentes tipos de consultas, el agente debe identificar la intención del usuario para proporcionar respuestas adecuadas.

Generación de Respuestas y Preguntas: El agente debe ser capaz de generar respuestas y formular preguntas para mantener la conversación hasta que el usuario obtenga la información deseada. Es fundamental que la aplicación y el agente estén conectados para que las consultas de los usuarios se envíen al agente y las respuestas generadas por el agente se entreguen a los usuarios de manera apropiada.

4.2 El Agente conversacional

El elemento principal a integrar es el agente conversacional, responsable de interactuar fluidamente con el usuario. Esta herramienta, integrada a la aplicación, posee la habilidad de discernir la semántica y las palabras clave que el usuario emplea para construir respuestas adecuadas. A diferencia de un motor de búsqueda convencional, el agente permite al usuario formular consultas como si estuviera en una conversación telefónica o dialogando con otra persona, favoreciendo una comunicación en lenguaje natural.

La ventaja distintiva de los agentes conversacionales es su disponibilidad constante y su enfoque centrado en guiar al usuario directamente a la información solicitada, evitando distracciones. Para este proyecto, se ha aplicado Dialogflow como nuestra herramienta de creación de asistentes conversacionales. Su amplio conjunto de características en la versión gratuita y el avanzado desarrollo de su inteligencia artificial, que aprende con cada interacción, garantizan conversaciones realistas y eficientes. **Además, este se integra con varios servicios de Google, incluido el módulo Speech-to-text.** El cual permite crear múltiples agentes con una sola cuenta de usuario, para nuestro propósito, sólo necesitaremos uno. Este agente será capaz de iniciar una conversación básica, preguntando al usuario si desea información sobre la accesibilidad de un edificio o estación y, posteriormente, solicitando detalles específicos. Para lograr este nivel de interacción, utilizaremos "intents" (intenciones) y "entities" (entidades) dentro de la plataforma.

4.3 Aplicación

Pese a las posibilidades de integración de las que dispone Dialogflow se ha tomado la decisión de incluir el agente en una aplicación que tenga más funcionalidades aparte del propio agente. Pese a que el agente sea la funcionalidad más importante de la aplicación, hay otras funcionalidades igualmente importantes y útiles. La aplicación será funcional en navegadores web, tanto desde PC como en dispositivos móviles, permitiendo así el acceso a la aplicación a todos los usuarios del sistema CRIMINIS.

4.3.1 Herramientas utilizadas

Para el desarrollo de este módulo de aplicación, además del IDE convencional utilizado para el desarrollo del sistema (VScode, Postman, node js) se ha utilizado un asistente gráfico online

para la creación de aplicaciones. La plataforma es conocida como Mockflow, y una de sus mayores ventajas es que las aplicaciones creadas con esta plataforma están disponibles en la nube para poder editarse.

La principal característica de esta plataforma es la forma en la que se diseñan las aplicaciones. No es necesario que el desarrollador conozca ninguno de los lenguajes utilizados para el desarrollo del diseño en ninguno de los sistemas operativos, ya que no es necesario escribir código. En su lugar, el diseño de las pantallas se realiza mediante bloques y componentes que se van uniendo con la lógica que decida el diseñador. Por tanto, el desarrollador es el encargado de organizar las pantallas, incluir los elementos necesarios y realizar las conexiones entre las pantallas y los elementos de tal manera que todos ellos sean útiles. Se encarga también de que la aplicación sea lo más visualmente agradable e intuitiva posible. La plataforma se encarga por tanto de visualizar la distribución de las pantallas y la secuencia de los bloques de las diferentes instancias de la interfaz y así, analizar las conversaciones posibles del sistema en general.

Existen otras plataformas similares a la utilizada en este proyecto, pero una de las funcionalidades que ofrece esta plataforma y que sus competidoras no, es la facilidad de conexión con la plataforma Dialogflow. De esta manera será más fácil enviar las consultas del usuario y mostrar las respuestas del agente. Esta plataforma permite ver y comparar proyectos de otros usuarios, y basándonos en ellos crear nuestras propias interfaces. En este caso, se ha decidido empezar desde las capturas de pantalla del sistema real actual para poder implementar las interfaces conversacionales deseadas y organizar la aplicación de una manera sencilla e intuitiva.

Text-to-speech: Permite a la aplicación transformar textos en voz, para facilitar así la comunicación con el usuario. Speech recognizer [17], también conocido como Speech-to-text, permite transformar en texto aquello que el usuario diga .

Assistant: Como se mencionaba anteriormente, Mockflow incluye de forma nativa inclusión con Dialogflow. Al incluir este elemento en alguna pantalla de nuestra aplicación, solo tendremos que introducir las credenciales de nuestro asistente para poder interactuar con él directamente desde la aplicación.

Traductor: Permite la traducción de textos entre dos idiomas seleccionados. En el siguiente bloque de componentes, encontramos los relacionados con imágenes y gráficos:

Capturas de pantalla: Permite incluir una imagen en la pantalla de nuestra aplicación. Será suficiente con subir la imagen a la plataforma y después seleccionarla para ser visualizada.

Animaciones: Este componente impulsado por Lottie permite añadir fácilmente animaciones a nuestra aplicación. Éstas pueden ser creadas o descargadas de su librería en formato .json. Estos elementos son conocidos dentro de la aplicación como elementos invisibles. Reciben esta denominación ya que no son elementos que puedan ser vistos por el usuario en la aplicación, como puede ser un botón. En su lugar, le dan a la aplicación una serie de características que pueden facilitar mucho la accesibilidad para el usuario, pero no son visibles dentro de la aplicación, si no que es labor del desarrollador hacer que sean funcionales. Son también invisibles para el desarrollador dentro de la propia plataforma, porque al incluirlas en alguna pantalla no aparecen dentro de la pantalla, sino que solo estarán presentes en la zona bajo el título de "componentes invisibles" y dentro de la lógica de la pantalla "bloques".

Dentro de los componentes relacionados con datos, encontramos la posibilidad de realizar conexiones con diferentes bases de datos, locales o en la nube, entrar en el almacenamiento local del dispositivo del usuario, realizar conexiones con API.

Todos los elementos que integramos en las pantallas tienen su parte de lógica reflejada en esta pantalla, pero además de éstos elementos, tenemos otros componentes comunes a todas las pantallas y que podremos usar independientemente de los elementos que se decidan incluir. Estos elementos son:

- **Control.** Dentro de los elementos de control encontramos los elementos básicos de programación de bucles, en los que se puede realizar iteraciones de un proceso, bloques condicionales, y los bloques necesarios para navegar a otra pantalla o para abrir una página web.
- **Lógica.** En la parte de lógica tenemos los bloques que nos permiten evaluar enunciados para saber si son verdaderos o falsos.
- **Texto.** En texto tenemos todo lo relacionado con variables de texto. se pueden crear nuevas variables, buscar ciertas letras o palabras dentro de otras palabras u oraciones.
- **Listas.** Dentro de listas encontramos todo lo relacionado para crear listas de datos, inicializarlas, aumentarlas, recortarlas o eliminarlas. Los elementos de las listas pueden ser creados por el programador, o mediante una lógica más compleja se puede conseguir que sea el propio usuario el que vaya añadiendo elementos a la lista.
- **Color.** Se pueden asignar colores a diferentes elementos de la pantalla en el caso de que queramos que éstos cambien en algún momento.
- **Objetos.** Nos permite crear objetos desde JSON o crear JSON a partir de un objeto. También nos permite recoger el valor de alguna variable de dicho objeto.
- **Variables.** Una de las partes más importantes a la hora de programar la aplicación es la parte de las variables. En esta parte encontramos los bloques necesarios para inicializar una variable a un valor numérico o a un texto, así como cambiar su valor en cualquier momento. De igual manera, el programador puede crear tantas variables como desee. La utilidad de las variables es la de pasar información de una pantalla a otra pantalla, que de otra manera no tendríamos la posibilidad. Dentro de las variables que se pueden crear encontramos tres tipos diferentes de variables según donde se guarde su valor:
 - **Aplicación:** se guarda en la propia aplicación.
 - **Dispositivo:** se guarda el valor en el dispositivo. De esta manera se puede recoger el valor de la variable de anteriores sesiones del propio usuario. Es especialmente útil para tener valores de ajustes del usuario.
 - **Nube:** el valor es guardado en servidores de internet, servicio de Google para bases de datos guardadas en la nube. De esta manera, se pueden exportar a bases de datos externos y así poder realizar tareas de recolección de información de datos y análisis.

Existen dos maneras en las que se realiza la consulta y mantener la conversación con el asistente. Los desarrolladores pueden forzar al usuario a comunicarse mediante voz con nuestro asistente. Para ello, incluiremos el bloque que inicia la escucha, y el bloque que detiene la escucha. El bloque de iniciar la escucha devolverá un valor, la respuesta del asistente.

El otro método de comunicación con el asistente es mediante texto. Para poder realizar la comunicación por texto, necesitaremos recoger el texto introducido por el usuario en una variable, que será la que enviemos al asistente en el bloque "Query". Este bloque también nos

devolverá un valor, que será también la respuesta del asistente. En ambos bloques, tanto de comenzar escucha como el de realizar la consulta por texto, una vez se ha producido la comunicación, y activar la lógica que sigue a esta acción. Mediante una combinación inteligente de todos los componentes disponibles en la plataforma, es posible crear aplicaciones intuitivas y visualmente agradables al usuario.

4.3.2 Presentación de la aplicación.

Antes de comenzar a desarrollar la aplicación es importante conocer cuál va a ser el flujo que va a seguir un usuario al usar la aplicación.

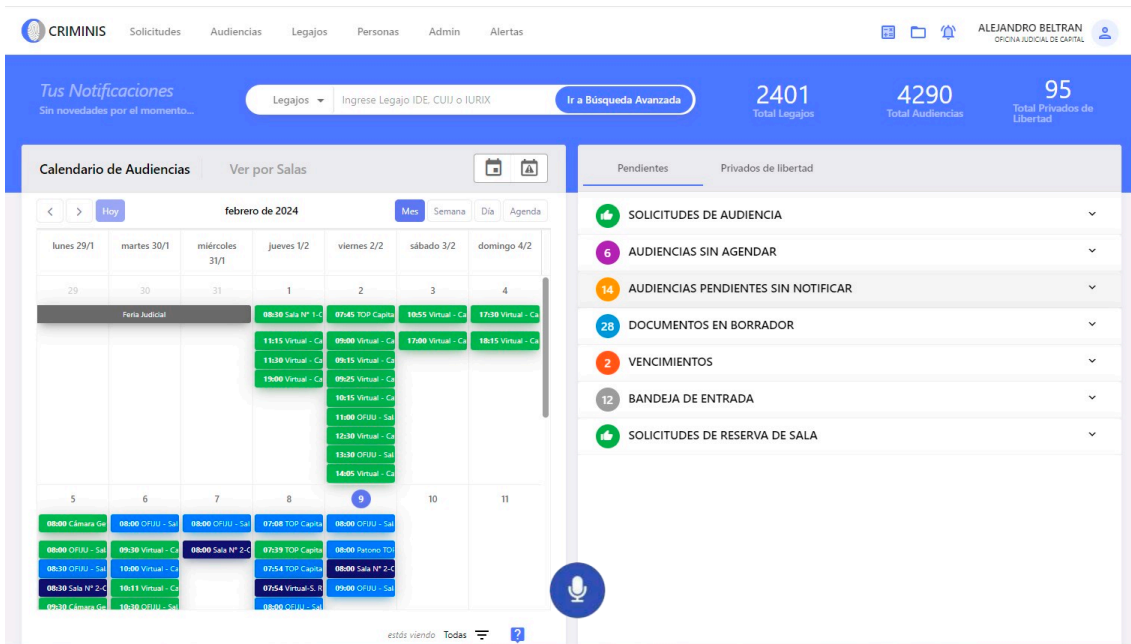


Fig. 4.1 Prototipo: presentación de la aplicación inicio

Como se puede observar en la figura 4.1, al abrir la aplicación el usuario se encuentra en la pantalla donde puede conversar con el asistente. Aquí puede decidir si desea mantener una conversación con el asistente o desea ir a la pantalla de búsqueda de legajos, donde se encuentran los listados de registros disponibles con filtros manuales. En caso de que desee conversar con el agente, se producirán diferentes preguntas y respuestas en la conversación hasta que el asistente comprenda la consulta. Al tener los datos necesarios para realizar la consulta, devolverá la información sobre las coincidencias. En este momento aparecen en la pantalla un modal de diálogo mostrando opciones de filtros diferentes que pueden ser seleccionados por el usuario para ir a la pantalla de resultados, donde podrá conocer el resultado, o redirigir a los registros filtrados. En esta pantalla, podrá ver la información del registro seleccionado, y al mencionar el ID de este podrá ir al detalle del registro en cuestión.

4.3.3 Pantalla de conversación.

En esta pantalla el usuario encuentra información escrita sobre los temas con los que puede ayudarle el asistente. Encuentra a su vez un icono donde puede activar o desactivar el reconocimiento de voz por parte del asistente, en caso de que desee o no escuchar la respuesta. Encuentra un cuadro de texto donde podrá introducir las consultas al asistente, un botón para enviar el texto al asistente y un botón que activa la escucha de la aplicación y del

asistente para realizar la comunicación por voz. Cuando el usuario finalice de hablar o escribir, verá lo que ha dicho en la parte superior de la pantalla, como se observa en la figura 4.2, del listado de legajos, como si se tratara de una aplicación común de búsqueda de registros convencional.

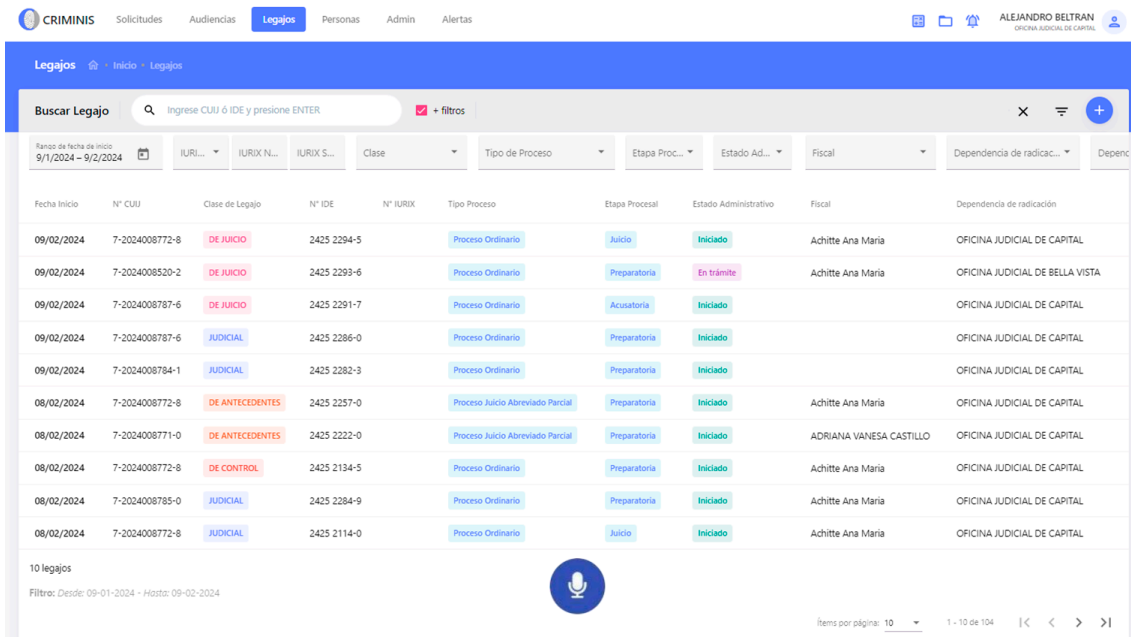


Fig. 4.2 Prototipo: integración con el estado de legajos

De igual manera, cuando el asistente responda, verá este texto en la parte superior, debajo de lo que él ha dicho previamente. Cada vez que el usuario o el agente realicen interacciones, en la figura 4.3 se observa el despliegue de interfaz, se sustituirá el contenido mostrado en la pantalla por el de la última iteración.

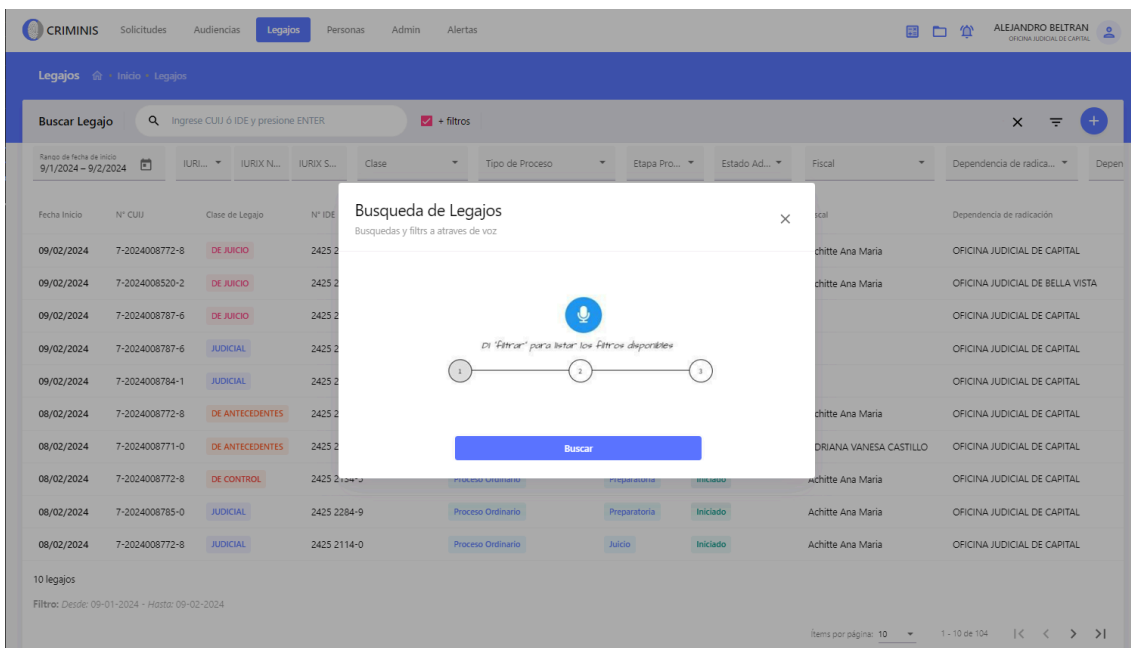


Fig. 4.3 Prototipo: integración con el diálogo

Cuando termine la conversación, y el asistente haya sido capaz de realizar la consulta, y haya obtenido información sobre legajos judiciales o la persona por la que pregunta el usuario, devolverá esta información. Cuando esto sucede, se activa una lógica dentro de la aplicación que permite mostrar una serie de botones que tienen diferentes funciones. En primer lugar, el botón de ir al listado de resultados, que le permite al usuario navegar hasta la pantalla de registros coincidentes, donde encontrará el listado centrado sobre el lugar que acaba de ser consultado.

En segundo lugar, el botón redireccionar al detalle, para poder acceder más adelante a esta información en la siguiente pantalla. Una vez termine la consulta y haya sido efectiva, el usuario podrá volver a iniciar una conversación con el agente sobre un lugar diferente o bien salir de la aplicación. En la parte de los bloques de esta pantalla, es donde se crean y se inicializan los valores de todas las variables utilizadas en las diferentes pantallas de la aplicación. El modo de obtener las direcciones de los diferentes lugares, es mediante la búsqueda de esta información dentro de la respuesta proporcionada por el asistente. Este valor es guardado en una variable al pulsar el botón ir al legajo, que pasará este valor a la siguiente pantalla, como se ilustra en la siguiente figura 4.4, permitiendo mostrar esta información.

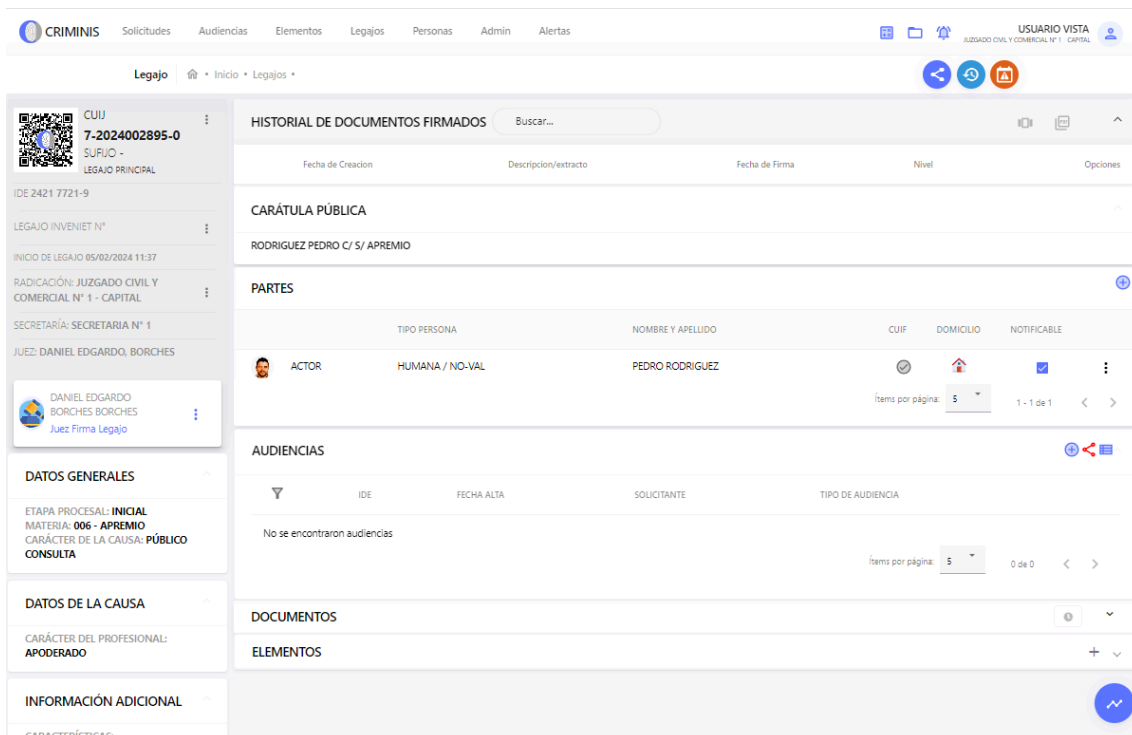


Fig. 4.4 Prototipo: integración Pantalla de redirección

5. OPTIMIZACIÓN Y PRUEBAS

Este capítulo profundiza en las pruebas de rendimiento específicas para la funcionalidad de comandos de voz dentro del sistema de interfaz conversacional para búsquedas de legajos judiciales. Se abordan las herramientas empleadas para las pruebas, se describen pruebas unitarias y de integración detalladas, y se presentan datos de ejemplo sobre la performance relacionada con los comandos de voz.

5.1 Caso de Uso. Inicializar Interfaz Conversacional

Nombre: Inicializar Interfaz Conversacional entrada de audio

Meta: Ofrecer una interfaz opcional al uso de lenguaje natural al usuario para desplegar un diálogo y mediante comandos de voz proceder a la búsqueda de legajos.

Flujo de eventos

3.1 Ruta principal

[Sistema]: El sistema presenta un icono representativo a la interfaz principal con forma de Micrófono y en estado de escucha, [Usuario]: El actor introduce voz mediante el dispositivo y visualiza el cambio de estado del icono de la aplicación.

[Usuario]: El actor introduce voz con la frase “Hola Criminis” [flujo de excepción E01]
[flujo alternativo A01]

[Sistema]: El sistema analiza el comando de voz, analiza el algoritmo y despliega un diálogo para proceder con la búsqueda.

Con la señal de audio analizada. La señal resalta el icono de la aplicación de otro color en los intervalos donde se encuentra una palabra. Un registro de las palabras reconocidas

- Palabra (rojo)
- Inicio (azul)
- Fin (celeste)

3.2 Flujos alternos

3.2.1 A01 Cancelar

[Personal]: El actor selecciona la opción cancelar

[Sistema]: El sistema regresa a la pantalla principal sin realizar ninguna acción

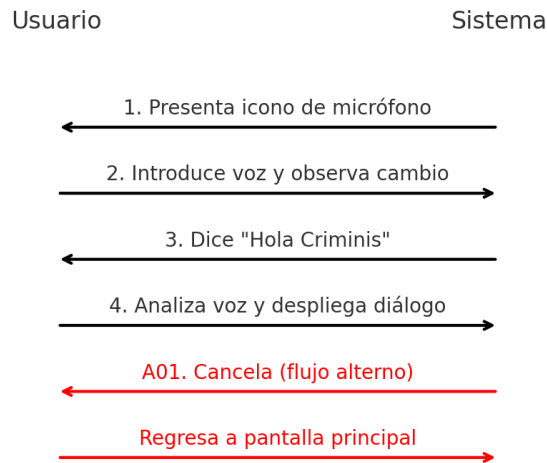


Fig. 5.1 Diagrama de Secuencia del Caso de Uso de Inicialización

5.1.1 Caso de Uso: Seleccionar Filtro

Nombre: Seleccionar Filtro de Búsqueda por Voz

Meta: Permitir al usuario especificar filtros para la búsqueda de legajos utilizando comandos de voz.

Precondiciones: La interfaz conversacional debe estar inicializada y escuchando.

Actores: Usuario (principal), Sistema (secundario).

Flujo de eventos

3.1 Ruta principal

1. [Sistema]: Muestra la interfaz de búsqueda con opciones de filtros disponibles.
2. [Usuario]: Pronuncia el comando de voz con el filtro deseado, por ejemplo, "Filtrar por fecha".
3. [Sistema]: Reconoce el comando de voz y muestra los filtros de fecha disponibles.
4. [Usuario]: Selecciona un rango de fechas específico usando comandos de voz.
5. [Sistema]: Aplica el filtro y actualiza los resultados de búsqueda en la interfaz.

3.2 Flujos alternos

A01 Cancelar Selección de Filtro

[Usuario]: En cualquier momento, dice "Cancelar filtro".

[Sistema]: Cancela la selección de filtro y regresa a los resultados de búsqueda sin filtros aplicados.

A02 Filtro No Reconocido

[Usuario]: Pronuncia un comando de voz que no corresponde a un filtro válido o no es reconocible.

[Sistema]: Muestra un mensaje de error o sugerencia: "No se reconoce el filtro, por favor intente de nuevo."

[Usuario]: Repite el comando de voz o utiliza comandos de ayuda.

[Sistema]: Guía al usuario a través de comandos de voz aceptables o cancela la operación después de varios intentos fallidos.

Postcondiciones: Los resultados de búsqueda se muestran de acuerdo con los filtros aplicados.

5.1 Pruebas Generales de la Aplicación General

5.1.1 Herramientas Utilizadas para las Pruebas

- **Jest:** Utilizado para pruebas unitarias en el backend, permitiendo simular interacciones con la base de datos y la lógica de negocio.
- **Mocha y Chai:** Empleados para pruebas de integración, especialmente útiles para probar la API y las interacciones entre el servidor y el cliente.
- **Lighthouse:** Herramienta de Google utilizada para evaluar la performance, accesibilidad, prácticas recomendadas y SEO de las aplicaciones web.
- **Dialogflow Simulator:** Proporcionado por Google para probar y evaluar interacciones con el agente conversacional en un entorno controlado.

5.1.2 Pruebas Unitarias y de Integración

Se diseñaron pruebas unitarias y de integración listadas en la tabla 5.1 para cada componente del sistema, asegurando su correcto funcionamiento y la calidad del código.

ID de Prueba	Descripción	Resultado Esperado	Resultado Obtenido	Estado
UT-01	Inicializar la búsqueda de legajos	Búsqueda inicializada exitosamente	Como esperado	Pasó
UT-02	Selección de Filtro por campo CUIJ	Filtro CUIJ establecido	Como esperado	Pasó
UT-03	Búsqueda de legajo por CUIJ	Búsqueda del legajo ejecutada	Como esperado	Pasó
UT-04	Redirección a un legajo	Redirección a la vista detalle de Legajo	Como esperado	Pasó

Tabla 5.1: Pruebas Unitarias para la Lógica de Negocio

A su vez los resultados obtenidos desde Chrome Lighthouse , arrojaron los siguientes informes de resultados, cada métrica ha sido seleccionada para el flujo en el que intervienen en este módulo de aplicación.

Métrica	Valor Antes de Optimización	Valor Después de Optimización	Mejora
First Contentful Paint	2.4s	1.8s	25%
Speed Index	4.3s	3.1s	28%
Time to Interactive	5.2s	3.9s	25%
Total Blocking Time	600ms	300ms	50%

Tabla 5.2: Pruebas Unitarias obtenidos desde Chrome Lighthouse

5.1.3 Optimización y Mejoras Implementadas

Basado en los resultados obtenidos, se realizaron las siguientes optimizaciones:

- **Minimización de CSS y JavaScript:** Reducción del tiempo de carga al minimizar archivos estáticos.
- **Lazy loading de imágenes:** Mejora en el tiempo de carga de la página al cargar imágenes solo cuando son necesarias.
- **Caching de consultas a la base de datos:** Reducción en los tiempos de respuesta a evitar consultas repetitivas.
- **Optimización del agente conversacional:** Ajustes en las intenciones y entidades para mejorar la precisión y la velocidad de respuesta del sistema.

5.1.4 Conclusiones de las Pruebas

Las pruebas realizadas demostraron una mejora significativa en el rendimiento del sistema tras las optimizaciones. El uso de herramientas como Lighthouse permitió identificar áreas críticas de mejora, y la implementación de prácticas de optimización llevó a una mejor experiencia de usuario, con tiempos de carga más rápidos y respuestas más ágiles del agente conversacional. Estas mejoras son esenciales para mantener la eficiencia y la escalabilidad del sistema frente a un aumento en el volumen de usuarios y consultas. Estas pruebas y optimizaciones son cruciales para asegurar que la interfaz conversacional no solo cumpla con sus requisitos funcionales sino que también ofrezca una experiencia de usuario rápida, eficiente y

satisfactoria. Continuar con la monitorización y optimización del sistema será esencial para su éxito a largo plazo.

5.2 Pruebas Generales de la Interfaz Conversacional

5.2.1 Herramientas Utilizadas en las Pruebas de Comandos de Voz

Google Cloud Speech-to-Text: Para transcribir comandos de voz a texto, evaluando la precisión y la velocidad de transcripción. Dialogflow Simulator: Utilizado para probar la interacción y comprensión de los comandos de voz por parte del agente conversacional, evaluando su capacidad para entender y responder adecuadamente. WebAIM Wave: Para evaluar la accesibilidad de la interfaz de usuario en el contexto de los comandos de voz y las respuestas audibles.

5.2.2 Pruebas Específicas sobre el Uso de Comandos de Voz

Las pruebas se centraron en la eficacia, precisión, y tiempo de respuesta del sistema al procesar comandos de voz.

ID de Prueba	Descripción	Comando de Voz	Resultado Esperado	Resultado Obtenido	Estado
VR-01	Reconocimiento de frase inicialización	"Hola CRIMINIS"	Apertura de Panel de Diálogo	Apertura de Panel de Diálogo	Pasó
VR-02	Búsqueda por identificador CUIJ	"Buscar Legajos 946"	Listado de legajos con 946	Listado correcto	Pasó
VR-03	Actualización de un legajo mediante voz	"Actualizar legajo 946"	Legajo 946 actualizado	Actualización fallida	Falló
VR-04	Redirección a un legajo por comando de voz	"Redireccionar a legajo 946"	Redirección al detalle de 946	Redirección a detalle	Pasó

Tabla 5.3: Pruebas de Precisión en Reconocimiento de Voz

VR-01: Reconocimiento de frase inicialización

Descripción: Esta prueba evalúa la capacidad del sistema para reconocer una frase compuesta por palabras claves específicas que inician un panel de diálogo a la búsqueda de un legajo por sus identificadores.

Comando de Voz: "Hola Criminis"

Resultado Esperado vs. Obtenido: El sistema debía desplegar un panel de diálogo ante el comando de voz "Hola Criminis", y lo hizo correctamente.

Estado: Pasó

VR-02: Búsqueda por identificador CUIJ número

Descripción: Prueba la eficiencia del sistema en interpretar comandos de voz que solicitan la búsqueda de legajos por su identificador CUIJ.

Comando de Voz: "Buscar Legajos 946"

Resultado Esperado vs. Obtenido: Se esperaba obtener un listado de legajos coincidentes en su número de cuij con el número 946, y el sistema proporcionó el listado correctamente.

Estado: Pasó

VR-03: Actualización de un Legajo mediante Voz

Descripción: Evalúa la capacidad del sistema para actualizar la información de un legajo específico a través de un comando de voz.

Comando de Voz: "Actualizar legajo 946"

Resultado Esperado vs. Obtenido: Aunque se esperaba que el legajo 946 no fuera actualizado, de igual manera la actualización falló.

Estado: Falló

VR-04: Redirección a un legajo por comando de voz

Descripción: Mide la efectividad del sistema al procesar comandos de voz para redirigir a la vista de detalle del legajo encontrado.

Comando de Voz: "Redireccionar a legajo 946"

Resultado Esperado vs. Obtenido: El sistema debía confirmar la redirección al legajo, lo cual ocurrió según lo esperado.

Estado: Pasó

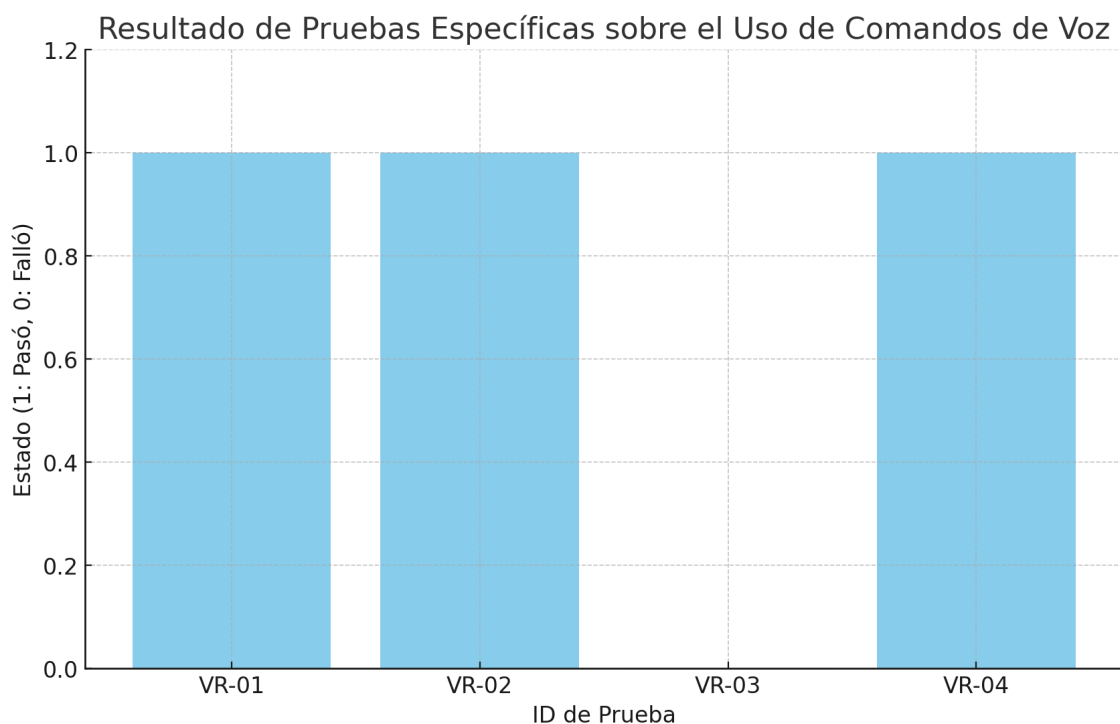


Fig. 5.2 prueba específica sobre el uso de comandos de voz

El gráfico anterior 5.2 ilustra el resultado de cada prueba específica sobre el uso de comandos de voz. Como se puede observar, las pruebas VR-01, VR-02 y VR-04 pasaron exitosamente, mientras que la prueba VR-03 falló. Este gráfico compara las métricas de tiempo de transcripción y respuesta del agente antes y después de la optimización. Se observa que ambas métricas muestran una mejora significativa tras la optimización: el tiempo de transcripción se redujo de 3 segundos a 2 segundos (una mejora del 33%), y el tiempo de respuesta del agente disminuyó de 2 segundos a 1.5 segundos (una mejora del 25%). Los cambios están visualizados en colores, con los valores antes de la optimización en rojo claro y los valores después de la optimización en verde claro, resaltando claramente el efecto positivo de las optimizaciones realizadas.

Métrica	Valor Antes de Optimización	Valor Después de Optimización	Mejora
Tiempo de Transcripción	3s	2s	33%
Respuesta del Agente	2s	1.5s	25%

Tabla 5.4: Tiempo de Respuesta a Comandos de Voz

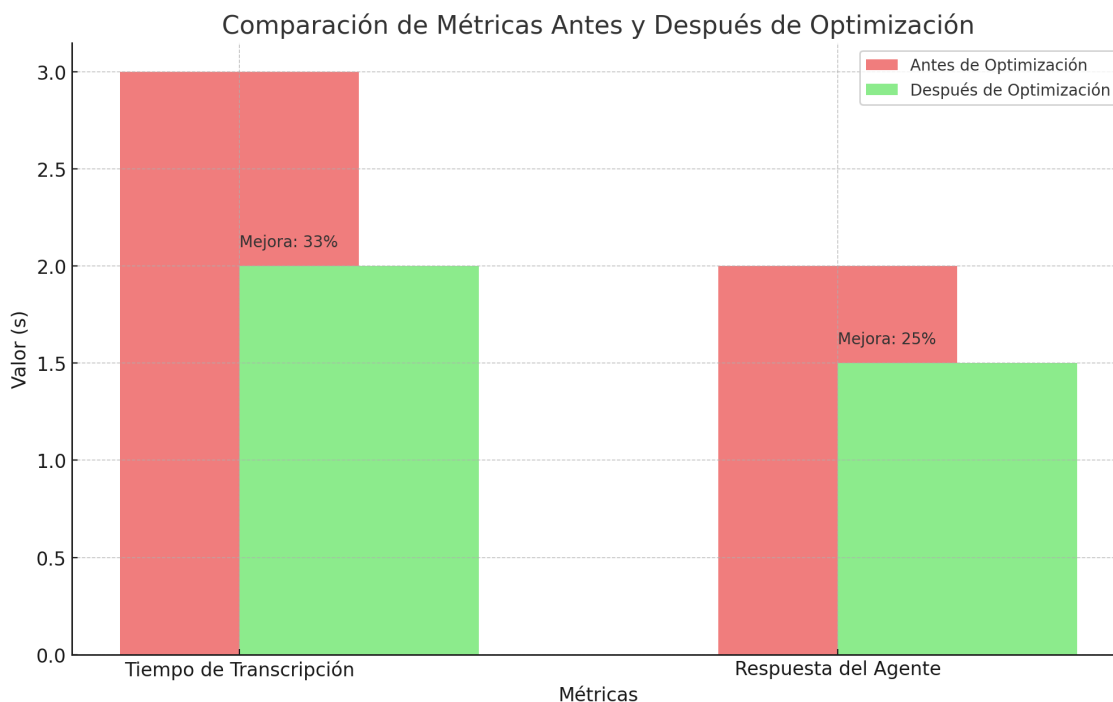


Fig. 5.3 comparativa de las pruebas específicas

Esta representación de la figura 5.3 proporciona una representación visual clara del estado de las pruebas específicas de comandos de voz y de la eficacia de las optimizaciones implementadas en el sistema.

5.2.3 Optimización de la Funcionalidad de Comandos de Voz

Las optimizaciones se enfocaron en mejorar la precisión del reconocimiento de voz y reducir los tiempos de respuesta. Mejora de la Precisión de Transcripción: Ajuste de la configuración de Google Cloud Speech-to-Text para optimizar el reconocimiento de términos jurídicos específicos.

Entrenamiento del Agente Conversacional: Ampliación del conjunto de entrenamiento en Dialogflow para mejorar la comprensión de comandos de voz variados y reducir errores de interpretación. Optimización de la Respuesta Auditiva: Implementación de respuestas pregrabadas para comandos frecuentes, mejorando la velocidad de respuesta del sistema.

5.2.4 Análisis de las Pruebas de Comandos de Voz

Las pruebas demostraron una mejora significativa en la funcionalidad de comandos de voz después de las optimizaciones implementadas. La precisión en el reconocimiento de voz y la velocidad de respuesta del sistema son cruciales para una experiencia de usuario satisfactoria, especialmente en aplicaciones judiciales donde la eficiencia y la accesibilidad son prioritarias. Continuar con la optimización y las pruebas periódicas garantizará la escalabilidad del sistema y su capacidad para adaptarse a las necesidades cambiantes de los usuarios. La implementación de estas optimizaciones ha demostrado ser fundamental para mejorar la experiencia de

usuario al interactuar con el sistema mediante comandos de voz, asegurando que la tecnología de voz sea una herramienta eficaz y accesible para la búsqueda de legajos judiciales.

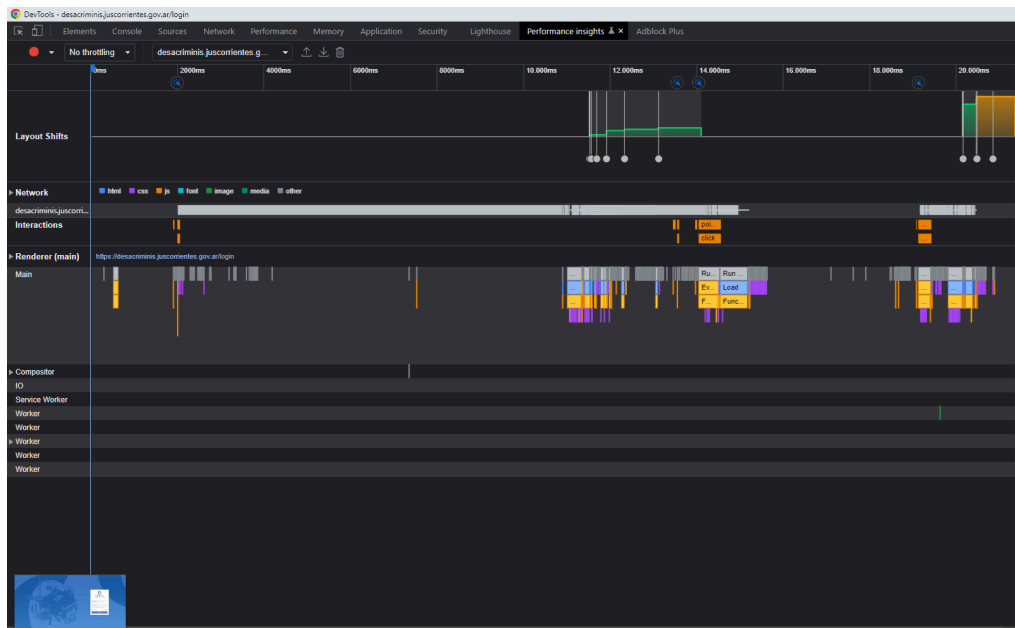


Fig 5.4 Resultado de Test de Performance de Lighthouse

6. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

Se considera que el acceso equitativo a la justicia es un derecho fundamental, especialmente en la era digital, donde se convierte en un aspecto crítico de inclusión social y equidad la accesibilidad para personas con discapacidad. Por tecnologías basadas en el lenguaje natural para la búsqueda de información judicial, se democratiza el acceso a la justicia y se mejora significativamente la experiencia de todos los usuarios, independientemente de sus capacidades físicas o cognitivas.

6.1 Conclusiones sobre la metodología

Fue el diseño e implementación de una interfaz conversacional el foco del trabajo de especialización, utilizando metodologías ágiles y una cuidadosa selección de herramientas y tecnologías. La metodología iterativa, centrada en Scrum, fue adoptada para el desarrollo del proyecto, permitiendo flexibilidad y adaptación a cambios, lo que se consideró crucial para el éxito del proyecto. La colaboración efectiva del equipo, la iteración rápida y la capacidad de responder a feedbacks específicos de los usuarios y otros stakeholders fueron facilitadas por la elección de esta metodología.

En la fase de diseño, se puso un énfasis particular en comprender las necesidades del usuario final y en cómo la tecnología podría abordar eficazmente estos requisitos. Se aseguró que la interfaz no solo fuera técnicamente viable sino también intuitiva y accesible para todos los usuarios, incluidos aquellos con discapacidades, mediante este enfoque centrado en el usuario. Se llevaron a cabo análisis detallados de la situación actual, siendo identificados los puntos débiles y áreas de mejora en la interacción con el sistema judicial existente. La necesidad de una solución que mejorará la eficiencia, la accesibilidad y la satisfacción del usuario fue fundamentada por estos análisis.

Las iteraciones del proyecto se dividieron en Sprints, cada uno con objetivos claros y entregables definidos. Los requisitos fundamentales fueron establecidos y una interfaz conversacional básica fue diseñada en la primera iteración. Este Sprint inicial fue crucial para sentar las bases del proyecto, implementando el reconocimiento de voz y desarrollando la lógica de búsqueda primaria. La mejora y expansión de la funcionalidad, basada en el feedback de los usuarios y stakeholders, fue dedicada a la segunda iteración. La experiencia del usuario fue refinada, la precisión del reconocimiento de voz fue mejorada y la seguridad y privacidad de la información tratada fueron aseguradas por este proceso iterativo. La metodología adoptada para el desarrollo de la interfaz conversacional se demostró eficaz en abordar los desafíos técnicos y de diseño del proyecto. La iteración continua, el diseño centrado en el usuario y la integración de tecnologías avanzadas fueron clave para cumplir con los objetivos del proyecto, ofreciendo una solución que mejora notablemente la accesibilidad y eficiencia del sistema judicial. Sin embargo, áreas de mejora fueron identificadas, como la necesidad de optimizar aún más el reconocimiento de voz y expandir las funcionalidades para abarcar una gama más amplia de consultas judiciales. El camino hacia futuras mejoras y expansiones del proyecto es señalado por estos hallazgos.

6.2 Conclusiones sobre la tecnología

La implementación de tecnologías de vanguardia para la interacción por voz en el sistema judicial ha demostrado ser un pilar fundamental en la transformación hacia un acceso más eficiente y humano a los servicios judiciales. La adopción de soluciones avanzadas de inteligencia artificial y procesamiento del lenguaje natural, especialmente aquellas proporcionadas por líderes tecnológicos como Google Cloud Speech-to-Text, Amazon Lex, y Microsoft Azure Cognitive Services, ha permitido no sólo una interpretación precisa del lenguaje natural sino también una interacción más fluida y natural con el sistema. La integración de estas tecnologías no solo ha mejorado la accesibilidad del sistema judicial para una audiencia más amplia, sino que también ha establecido un nuevo estándar en la rapidez y precisión con la que se pueden gestionar y acceder a los legajos judiciales.

La necesidad de integración con estas empresas de tecnología reconocidas ha sido instrumental en superar desafíos técnicos, especialmente en lo que respecta al manejo de terminología legal específica y la adaptación a diferentes acentos y modismos del lenguaje. Esta asociación estratégica ha permitido no solo aprovechar lo último en innovaciones tecnológicas sino también asegurar una implementación eficaz que se alinea con las necesidades específicas del sistema judicial. Estas tecnologías de productos de empresas reconocidas para la interacción con asistentes mediante la voz han marcado un hito en el proyecto, demostrando el potencial de la innovación tecnológica para revolucionar la interacción con los sistemas judiciales, haciéndolos más accesibles, eficientes y adaptados a las expectativas de los usuarios modernos. Esta dirección de las tecnologías no solo refleja un compromiso con la mejora continua sino también una visión hacia el futuro de los servicios digitales.

6.3 Conclusiones sobre las herramientas

La selección y evaluación de herramientas para el desarrollo de interfaces conversacionales involucró un análisis exhaustivo de las capacidades de tecnologías de reconocimiento de voz (STT) proporcionadas por líderes en la industria como Google Cloud, IBM Watson, Microsoft Azure, y la Web Speech API (W3C). Este análisis destacó la importancia de considerar múltiples factores como la facilidad de uso, compatibilidad, soporte de idiomas, calidad de voz, personalización, y costos asociados para elegir la herramienta más adecuada para el proyecto. Google Dialogflow emergió como una solución integral para la creación de interfaces conversacionales avanzadas, destacando por su poderosa combinación de procesamiento de lenguaje natural (PLN) y capacidades de integración. Su habilidad para interpretar la intención detrás de las entradas del usuario y generar respuestas coherentes y contextuales lo posiciona como una herramienta indispensable en el desarrollo de soluciones conversacionales complejas y altamente funcionales. La integración de Dialogflow con tecnologías Cloud Text-to-Speech y Speech-to-Text, ofrece una transición fluida entre reconocimiento de voz, procesamiento de lenguaje, y síntesis de voz, permitiendo una experiencia de usuario rica y dinámica.

La selección de herramientas no solo se basó en la calidad tecnológica sino también en la capacidad de estas para integrarse y complementarse entre sí, ofreciendo una solución

cohesiva que abarca tanto la interpretación inteligente de la voz como su generación. Este enfoque integrado garantizó, no solo una interacción natural y fluida con el sistema, sino también una plataforma escalable y flexible, capaz de adaptarse y expandirse según las necesidades del proyecto.

6.4 Conclusiones sobre el trabajo

La implementación del proyecto de interfaz conversacional para la gestión de legajos judiciales ha demostrado ser un avance significativo hacia la modernización y accesibilidad del sistema judicial. A través de un enfoque meticuloso en el diseño y la implementación, se logró desarrollar una solución que no solo mejora la eficiencia en la búsqueda y gestión de legajos, sino que también facilita un acceso equitativo a la justicia para todos los usuarios, incluidos aquellos con discapacidades. El trabajo realizado resaltó la importancia de adoptar una metodología ágil y centrada en el usuario, permitiendo iteraciones rápidas y ajustes basados en el feedback real de los usuarios. La integración de tecnologías avanzadas de reconocimiento de voz y procesamiento de lenguaje natural, como Google Cloud Speech-to-Text y Dialogflow, proporcionó una base sólida para el desarrollo de una interfaz intuitiva y eficaz.

La evaluación técnica del sistema reveló mejoras significativas en la experiencia del usuario, con tiempos de respuesta más rápidos y una interacción más natural y fluida. Las pruebas específicas sobre el uso de comandos de voz mostraron una alta tasa de éxito, aunque se destacó la necesidad de mejorar la gestión de errores y la interpretación de comandos ambiguos o incorrectamente pronunciados. Además, la optimización de la aplicación, basada en los resultados de las herramientas de análisis como Lighthouse, permitió identificar y abordar eficazmente cuestiones críticas relacionadas con la performance, accesibilidad y usabilidad. Estas mejoras no solo aumentaron la eficiencia del sistema, sino que también contribuyeron a una mejor satisfacción del usuario, estableciendo un sólido fundamento para futuras expansiones y refinamientos del proyecto.

En conclusión, el desarrollo de la interfaz conversacional representa un paso importante hacia la transformación digital del sistema judicial, demostrando el potencial de la tecnología para mejorar la accesibilidad y la eficiencia. Sin embargo, el trabajo también subraya la importancia de la evaluación continua y la adaptación a las necesidades cambiantes de los usuarios para asegurar que la solución siga siendo relevante y efectiva en el futuro.

6.5 Futuras líneas de trabajo

Desde el diseño del prototipo hacia la implementación real de la aplicación, es imperativo abordar tanto los desafíos técnicos como las oportunidades de innovación. El trabajo realizado hasta el momento ha establecido una sólida base conceptual; por lo cual el Desarrollo e Implementación Real de la Aplicación involucra la codificación activa y la configuración de la infraestructura necesaria para soportar la aplicación en un entorno de producción. Esto incluirá el establecimiento de bases de datos seguras, la implementación de servidores y la configuración de redes. La adopción de prácticas de DevOps y la integración

continua/despliegue continuo (CI/CD) serán fundamentales para agilizar este proceso. A su vez, la transición a una solución plenamente funcional demanda una exploración profunda en varias dimensiones clave:

- Mejora en la precisión del Reconocimiento de Voz: Aunque el prototipo ha demostrado la viabilidad del uso de comandos de voz, la precisión y el rendimiento en condiciones variadas aún pueden mejorarse. Se deberán explorar tecnologías avanzadas de reconocimiento de voz, potencialmente incorporando soluciones de aprendizaje profundo y adaptación contextual para manejar jerga legal específica y acentos variados.
- Expansión y Personalización de Funcionalidades: La adaptabilidad a las necesidades específicas de diferentes usuarios y contextos judiciales es esencial. Esto podría incluir la personalización de la interfaz y la funcionalidad basada en roles de usuario, preferencias individuales y requisitos legales locales.
- Integración con Plataformas y Tecnologías Emergentes: La incorporación de tecnologías emergentes, como blockchain para la seguridad de las interacciones y la inteligencia artificial para análisis predictivo y asistencia en la toma de decisiones, puede ofrecer mejoras significativas en eficiencia y efectividad.
- Seguridad y Cumplimiento: Dada la naturaleza sensible de los datos manejados por la aplicación, será crucial implementar rigurosas medidas de seguridad para proteger contra brechas de datos y asegurar el cumplimiento con regulaciones de privacidad y protección de datos como el GDPR.
- Formación y Soporte: Desarrollar materiales de formación y soporte técnico para usuarios finales asegurará que la transición al nuevo sistema sea lo más fluida posible, fomentando una adopción rápida y eficaz por parte de todos los usuarios.

La implementación real de la aplicación y las mejoras sugeridas reflejan un compromiso no solo con la innovación tecnológica sino también con una mejora continua basada en principios de accesibilidad, seguridad y eficacia. Este enfoque garantizará que el sistema judicial no solo se modernice, sino que también se vuelva más inclusivo y equitativo, alineado con las necesidades y expectativas de la gestión digital judicial de hoy.

Referencias

- [1] Modernización Tecnológica y Gestión de Información Judicial: Contini, F., & Mohr, R. (Eds.). (2018). Digital Justice: Technology and the Internet of Disputes. Oxford University Press.
- [2] Interfaz Conversacional y Usabilidad: Balentine, B., & Morgan, D.P. (2001). It's Better to Be a Good Machine Than a Bad Person: Speech Recognition and Other Exotic User Interfaces at the Twilight of the Jetsonian Age. ICMI Press.
- [3] Tecnologías Emergentes en el Ámbito Judicial: Contini, F., & Lanzara, G. F. (Eds.). (2009). ICT and Innovation in the Public Sector: European Studies in the Making of E-Government. Palgrave Macmillan UK.
- [4] Usabilidad y Experiencia del Usuario: Nielsen, J., & Loranger, H. (2006). Prioritizing Web Usability. New Riders.
- [5] World Wide Web Consortium. (s. f.). About W3C web standards.
<https://www.w3.org/standards/about/>
- [6] Mozilla Foundation. Web Speech API. Web Speech Concepts and Usage
https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
- [7] James Giangola. Applying Built-in Hacks of Conversation to Your Voice UI. (mayo 2017) Google I/O 2017; re_ty: Publish; https://www.youtube.com/watch?v=wuDP_eygsvs
- [8] H. Grice. "Logic and Conversation". Syntax and Semantics: Vol. 3: Speech Acts , Academic Press, (2004)
- [9] Thomas, S. A., & Bindu, V. R. (2022). Growth of agile software development methodologies in a web apps scenario. AIP Conference Proceedings, 2520(1), 020006.
<https://doi.org/10.1063/5.0104389>
- [10] Yuanfeng Song et al. (2022) enfoque de extremo a extremo para generar consultas SQL a partir de preguntas en lenguaje natural utilizando un modelo llamado SpeechSQLNet.
- [11] Hazboun et al. (2021) que detallan un método para automatizar la conversión de una consulta de datos de una consulta en lenguaje natural (NLQ) a SQL utilizando objetos de almacén de datos de cubo OLAP.
- [12] Contenedores y Arquitecturas Sin Servidor: Newman, S. (2020). Building Microservices: Designing Fine-Grained Systems (2nd Edition). O'Reilly Media.
- [13] Arquitecturas de Aplicaciones Web Modernas: Richardson, L. (2018). Microservices Patterns: With examples in Java. Manning Publications.
- [14] O'Brien, H.L. & Toms, E.G. (2010). The development and evaluation of a survey to measure user engagement in e-commerce environments. Journal of the American Society for Information Science & Technology, 61(1), 50-69. DOI: 10.1002/asi.v61:1.
- [15] Google Cloud. Conceptos básicos de Dialogflow ES. Google;
<https://cloud.google.com/dialogflow/es/docs/basics?hl=es-419>

[16] Manning, C. D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.

[17] Becerra J., Becerra D. Farías Pineda V., Martínez Sánchez. Sistema De Reconocimiento De Palabras Clave. 3.4 Preprocesamiento de la señal de voz en Conversaciones De Voz. 2014-A004.