

Planeación de sistemas de software

Gpo 101

Evidencia Sprint 4

Alejandra Coeto Sánchez

Diego de Jesús Esparza Ruiz

Jesús Adrián López Gaona

Luis Gerardo Juárez García

Mónica Soberón Zubía

Santiago De la Riva Juárez

Monterrey, NL.

31 de mayo, 2025

Índice

Resumen.....	3
Introducción.....	4
Contexto General.....	5
Delimitación del Objeto de Estudio.....	6
Planteamiento del Problema.....	7
Justificación.....	8
Marco Teórico.....	9
Procesamiento de Lenguaje Natural y Análisis Conversacional.....	9
Transcripción Automática de Audio.....	9
Análisis de Sentimientos y Detección Emocional.....	9
Generación Automática de Reportes Inteligentes.....	10
Inteligencia Artificial Conversacional y RAG.....	10
Retrieval-Augmented Generation (RAG).....	10
Chat Inteligente con Memoria Histórica.....	10
Sistema de Recomendaciones Inteligentes.....	11
Integración con APIs de Inteligencia Artificial.....	11
Visualización de Datos y Dashboards Interactivos.....	11
Business Intelligence y Analítica.....	11
Procesamiento de Formularios y Feedback.....	12
Seguridad y Control de Acceso.....	12
Role-Based Access Control (RBAC).....	12
Objetivo.....	12
Generales del proyecto:.....	12
Específicos del sprint:.....	13
Infraestructura.....	13
Arquitectura de Software y Tecnologías de Implementación.....	13
Arquitectura Modular y Escalable.....	13
Base de Datos Relacional Optimizada.....	13
Metodologías de Desarrollo y Calidad.....	15
Metodologías Ágiles y Scrum.....	15
Integración Continua y Control de Versiones.....	16
Sprint Backlog.....	16
Daily Sessions.....	18
Sprint Retrospective.....	20
Review Session.....	21
Estrategia Sprint 5.....	22
Hipótesis y Resultados.....	22
Conclusiones.....	22

Resumen

Este documento presenta el desarrollo de Howl, una plataforma web que emplea inteligencia artificial para el análisis automatizado de llamadas de servicio al cliente. A partir de las transcripciones de estas interacciones, el sistema aplica técnicas de procesamiento de lenguaje natural para generar resúmenes automáticos, detección de emociones, alertas de riesgo y recomendaciones personalizadas. La solución incluye el diseño e implementación de una base de datos relacional optimizada, que modela de manera eficiente entidades como usuarios, empresas, clientes, llamadas y resultados analíticos.

Howl está construido sobre una arquitectura modular que facilita su mantenimiento, escalabilidad e integración con otras plataformas. El desarrollo se llevó a cabo mediante metodologías ágiles, principalmente Scrum, con ciclos iterativos de planificación, implementación, pruebas y validación. Se incorporaron prácticas de integración continua, control de versiones colaborativo y pruebas automatizadas para asegurar la calidad del software.

El sistema cuenta con un panel administrativo para la gestión de roles y entidades, así como una interfaz conversacional inteligente que permite explorar los resultados del análisis de manera intuitiva. Además, ayuda a los consultores a prepararse mejor para su siguiente llamada al brindarles una visión clara del historial, necesidades y tono emocional del cliente.

Finalmente, Howl integra una sección de analítica con dashboards interactivos que muestran el rendimiento de los consultores y métricas clave de las llamadas. También procesa automáticamente formularios de evaluación enviados a los clientes, cuyos resultados se utilizan para retroalimentación y toma de decisiones. En conjunto, Howl convierte datos conversacionales en conocimiento estructurado y útil para las organizaciones.

Introducción

Hoy en día, muchas empresas tienen llamadas constantes con clientes, pero no aprovechan lo que se dice en ellas. Aunque en esas conversaciones hay datos valiosos sobre cómo se siente el cliente, lo que necesita o lo que salió mal, normalmente solo se almacenan como texto sin procesar. Esto significa que se pierden oportunidades para mejorar la atención, prevenir problemas o entender mejor a cada usuario.

Howl busca resolver ese problema. Es una plataforma web que analiza las transcripciones de llamadas con inteligencia artificial. Usa modelos de lenguaje para generar resúmenes, detectar emociones, identificar palabras de riesgo y dar recomendaciones. También ayuda a los consultores a prepararse para su siguiente llamada, dándoles contexto sobre el historial y estado emocional del cliente.

Además, Howl permite a las empresas ver todo lo que está pasando con sus llamadas. Tiene dashboards que muestran cómo van los consultores, qué tan satisfechos están los clientes y qué temas aparecen más. El sistema se desarrolló con metodología ágil (Scrum), integrando una base de datos relacional diseñada específicamente para el proyecto, una arquitectura modular y buenas prácticas como control de versiones, pruebas automatizadas y CI/CD. El objetivo es convertir datos conversacionales en algo que realmente se pueda usar para tomar decisiones.

Contexto General

Howl es un proyecto desarrollado como parte de un curso universitario de planeación de sistemas de software. El enfoque fue construir una solución funcional y escalable que resolviera un problema real usando tecnologías actuales. Desde el inicio, se planteó como una plataforma web con backend, base de datos y modelos de inteligencia artificial integrados.

El proyecto parte de la necesidad que tienen muchas empresas de entender mejor lo que ocurre en sus llamadas de atención al cliente. Aunque estas llamadas se graban o transcriben, en la mayoría de los casos no se hace nada con esa información. Esto representa una pérdida de datos valiosos que podrían usarse para mejorar el servicio, tomar decisiones o capacitar al personal.

Howl fue pensado como una herramienta que pueda adaptarse a distintos tipos de organizaciones, con un sistema de roles, múltiples empresas, clientes y consultores. Se desarrolló en equipo utilizando Scrum, organizando las tareas en sprints, y usando herramientas como GitHub Projects, CI/CD, y pruebas automatizadas para mantener la calidad. También se buscó desde el inicio que el sistema fuera modular y que el backend pudiera integrarse fácilmente con otras fuentes de datos o sistemas externos si fuera necesario.

Delimitación del Objeto de Estudio

Howl se enfoca en el análisis automatizado de llamadas de servicio al cliente usando inteligencia artificial. A diferencia de sistemas que dependen de integraciones específicas, Howl puede trabajar con cualquier plataforma de videollamada como Zoom, Microsoft Teams o Google Meet. El sistema puede recibir archivos de audio directamente o grabar desde la computadora del consultor mientras se realiza la llamada, lo que lo hace flexible y adaptable a distintos entornos.

El proceso parte del audio, que luego es transcrito y procesado por modelos de lenguaje para extraer información relevante. Esto incluye resúmenes automáticos, detección de emociones, alertas por palabras de riesgo y recomendaciones personalizadas. La herramienta no está limitada a un tipo de empresa, cliente o flujo; funciona con cualquier organización que tenga interacciones verbales con sus usuarios.

El sistema incluye también paneles para administración de usuarios, historial de llamadas, dashboards con métricas clave y un sistema de formularios automáticos para evaluar a los consultores. No se contempla por ahora el análisis de texto en tiempo real durante la llamada, ni procesamiento de otros canales como correo o chat. El enfoque principal es capturar lo que ocurre en la llamada, procesarlo y convertirlo en información útil para mejorar el servicio.

Planteamiento del Problema

La mayoría de las empresas que ofrecen atención al cliente por llamada no aprovechan lo que se dice en esas conversaciones. Aunque las llamadas pueden ser grabadas o transcritas, rara vez se analizan de forma estructurada. Esto significa que se pierde información valiosa sobre lo que quieren los clientes, cómo se sienten o qué problemas se repiten.

Además, muchas veces los consultores no tienen acceso a un historial claro de interacciones previas antes de hablar con un cliente, lo que dificulta dar una atención personalizada y rápida. Tampoco existen herramientas que ayuden a los administradores a entender con datos qué tan bien están funcionando sus equipos o qué temas están generando más insatisfacción.

El problema no es solo de almacenamiento, sino de interpretación. Se necesita una herramienta que no solo guarde las llamadas, sino que las analice automáticamente y presente la información más relevante de forma clara. También se requiere que funcione con cualquier plataforma de videollamada, que sea fácil de integrar al flujo de trabajo actual y que proporcione valor tanto al consultor como al administrador.

Howl nace como respuesta a este problema, convirtiendo conversaciones en información útil para mejorar el servicio y tomar mejores decisiones.

Justificación

Howl es una herramienta que ayuda a las empresas a aprovechar un recurso que ya tienen pero no están usando bien: sus llamadas con clientes. Estas conversaciones contienen información valiosa sobre dudas, emociones, quejas y oportunidades, pero normalmente solo se almacenan sin análisis. Howl permite transformar esas llamadas en datos útiles sin cambiar el flujo de trabajo actual.

La plataforma genera automáticamente resúmenes, identifica emociones y posibles riesgos, y muestra esta información de forma visual para que los administradores puedan tomar decisiones basadas en hechos. Además, ayuda directamente a los consultores al mostrarles el historial del cliente y recomendaciones antes de cada llamada, lo que mejora la atención y reduce tiempos de respuesta.

También automatiza la recolección de feedback por parte de los clientes a través de formularios y lo integra todo en dashboards con métricas claras. Esto permite evaluar el desempeño y detectar áreas de mejora sin necesidad de procesos manuales.

Howl no depende de una plataforma específica. Funciona con Zoom, Meet, Teams o cualquier otro servicio de videollamada, lo que lo hace flexible. Su diseño modular, su base de datos optimizada y su uso de IA lo convierten en una solución escalable, útil y lista para aplicarse en escenarios reales.

Marco Teórico

El desarrollo de HowlX se fundamenta en la convergencia de múltiples disciplinas tecnológicas y marcos conceptuales que permiten el procesamiento inteligente de conversaciones empresariales. Este marco teórico aborda los fundamentos tecnológicos, metodológicos y arquitectónicos que sustentan la plataforma.

Procesamiento de Lenguaje Natural y Análisis Conversacional

Transcripción Automática de Audio

La transcripción automática constituye el primer eslabón en la cadena de procesamiento de HowlX. El sistema se basa en modelos de reconocimiento automático de voz (ASR, por sus siglas en inglés) como OpenAI Whisper, que utiliza arquitecturas transformer para convertir señales de audio en texto estructurado. Estos modelos han demostrado alta precisión en múltiples idiomas y contextos conversacionales, especialmente en entornos empresariales donde la calidad del audio puede variar significativamente.

La transcripción no se limita a la conversión básica de audio a texto, sino que incluye la preservación de metadatos temporales, identificación de hablantes y mantenimiento del contexto conversacional. Esta información adicional resulta fundamental para el análisis posterior y la generación de insights contextuales.

Análisis de Sentimientos y Detección Emocional

El análisis de sentimientos en HowlX se implementa mediante modelos de procesamiento de lenguaje natural que clasifican las emociones expresadas durante las interacciones. Esta funcionalidad utiliza técnicas de aprendizaje profundo para identificar patrones lingüísticos, tonos emocionales y indicadores de satisfacción o insatisfacción del cliente.

El sistema emplea algoritmos de clasificación multiclase que pueden detectar emociones específicas como frustración, satisfacción, confusión o urgencia, proporcionando a los consultores y administradores una comprensión más profunda del estado emocional del cliente durante la interacción.

Generación Automática de Reportes Inteligentes

La generación de reportes se basa en técnicas de extracción de información y resumen automático. El sistema analiza las transcripciones para identificar elementos clave como:

- **Temas principales:** Clasificación automática de los asuntos discutidos
- **Decisiones tomadas:** Identificación de acuerdos y resoluciones
- **Tareas pendientes:** Extracción de compromisos y seguimientos requeridos

- **Puntos de riesgo:** Detección de palabras clave o frases que indican posibles problemas

Esta funcionalidad utiliza modelos de lenguaje grandes (LLMs) que han sido entrenados para comprender contextos empresariales y extraer información relevante de manera estructurada.

Inteligencia Artificial Conversacional y RAG

Retrieval-Augmented Generation (RAG)

HowlX implementa RAG como mecanismo central para proporcionar respuestas contextuales basadas en el historial de interacciones. Esta tecnología combina la capacidad de recuperación de información con la generación de lenguaje natural, permitiendo que el sistema:

1. **Indexe el contenido conversacional** mediante técnicas de embedding vectorial
2. **Recupere información relevante** basada en consultas específicas del usuario
3. **Genere respuestas contextuales** que incluyen citas exactas de las transcripciones originales

El componente de recuperación utiliza modelos de embedding que convierten el texto en representaciones vectoriales densas, permitiendo búsquedas semánticas eficientes. El componente de generación emplea modelos como GPT para producir respuestas coherentes y contextualmente apropiadas.

Chat Inteligente con Memoria Histórica

El sistema de chat inteligente mantiene contexto a través de múltiples sesiones, permitiendo conversaciones continuadas sobre el historial de llamadas. Esta funcionalidad se implementa mediante:

- **Gestión de estado conversacional:** Mantenimiento del contexto durante sesiones extendidas
- **Integración multi-llamada:** Capacidad de analizar información de múltiples interacciones simultáneamente
- **Respuestas con referencias:** Provisión de citas exactas y referencias temporales a las fuentes originales

Sistema de Recomendaciones Inteligentes

Las recomendaciones se generan mediante algoritmos de aprendizaje automático que analizan patrones en el historial de interacciones. El sistema considera factores como:

- Historial específico del cliente
- Patrones de comportamiento identificados

- Resultados de interacciones similares
- Contexto temporal y empresarial

Integración con APIs de Inteligencia Artificial

HowlX se diseñó con flexibilidad para integrar múltiples proveedores de servicios de IA:

APIs de Transcripción: Integración con servicios como OpenAI WhisperX, Gemini y otros proveedores de ASR que permiten a las organizaciones seleccionar la solución más adecuada para sus necesidades específicas.

Modelos de Lenguaje: Conectividad con diversos LLMs incluyendo GPT, Claude y otros modelos especializados, proporcionando opciones para optimizar costos y rendimiento.

Servicios Especializados: Capacidad de integrar APIs específicas para análisis de sentimientos, detección de idiomas y otras funcionalidades especializadas.

Visualización de Datos y Dashboards Interactivos

Business Intelligence y Analítica

Los dashboards de HowlX implementan principios de business intelligence para transformar datos conversacionales en insights accionables:

- **Métricas de Rendimiento:** Visualización de KPIs de consultores y equipos
- **Análisis Temporal:** Identificación de tendencias y patrones a lo largo del tiempo
- **Segmentación de Clientes:** Análisis comparativo por diferentes dimensiones empresariales
- **Alertas Proactivas:** Identificación automática de situaciones que requieren atención

Procesamiento de Formularios y Feedback

El sistema automatiza la recolección y procesamiento de feedback mediante:

- **Generación automática de formularios** de evaluación post-llamada
- **Procesamiento inteligente de respuestas** utilizando NLP para análisis de texto libre
- **Integración de resultados** en el sistema de métricas y dashboards
- **Correlación de feedback** con métricas de desempeño y análisis conversacional

Seguridad y Control de Acceso

Role-Based Access Control (RBAC)

HowlX implementa un sistema robusto de control de acceso que permite:

- **Definición granular de roles** adaptados a estructuras organizacionales específicas
- **Permisos contextuales** que varían según la relación entre usuarios, empresas y clientes
- **Auditoría de accesos** para mantener trazabilidad de operaciones sensibles
- **Escalabilidad de permisos** que se adapta al crecimiento organizacional

Esta implementación asegura que los datos conversacionales sensibles se mantengan protegidos mientras se facilita el acceso apropiado a la información relevante para cada rol organizacional.

Objetivo

Generales del proyecto:

El objetivo de HowlX es desarrollar una plataforma potenciada por inteligencia artificial que optimice la comunicación empresarial, específicamente en el contexto de llamadas con clientes.

Nuestra visión va más allá de facilitar las llamadas: queremos que HowlX sea una herramienta versátil que permita recolectar, analizar y aprovechar datos provenientes de conversaciones previas. Para lograrlo, planeamos integrar funcionalidades que permitan grabar o subir llamadas, las cuales serán analizadas automáticamente mediante IA para generar reportes, minutas, transcripciones y smart insights relevantes.

Además, la plataforma contará con un chat inteligente impulsado por tecnología RAG (Retrieval-Augmented Generation), el cual permitirá a los usuarios hacer preguntas específicas sobre llamadas pasadas y recibir respuestas acompañadas de citas exactas del transcript donde se encontró la información.

Por último, HowlX busca ser una herramienta estratégica de preparación: queremos facilitar un proceso ágil y efectivo para que los usuarios puedan prepararse para futuras conversaciones con clientes, proporcionando contexto sobre interacciones anteriores y recomendaciones personalizadas sobre cómo abordar cada llamada.

Específicos del sprint:

El objetivo de este cuarto sprint consistió en finalizar el flujo completo de la aplicación, lo cual implicaba terminar o avanzar considerablemente en los siguientes puntos:

- Flujo administrativo: que los administradores pudieran definir los roles de los usuarios y pudieran crear nuevas empresas y clientes.
- Página AI: Recomendaciones inteligentes. chat general, y Q&A con RAG
- Modularización de IA donde el cliente puede seleccionar el proveedor de el IA del sistema

Infraestructura

Arquitectura de Software y Tecnologías de Implementación

Arquitectura Modular y Escalable

HowlX se construye sobre una arquitectura modular que facilita el mantenimiento, la escalabilidad y la integración con sistemas externos. Esta arquitectura se compone de:

Capa de Presentación: Interfaz web responsiva que proporciona acceso a todas las funcionalidades del sistema mediante dashboards interactivos y interfaces conversacionales.

Capa de Lógica de Negocio: Implementa los procesos core del sistema, incluyendo el procesamiento de transcripciones, análisis de IA y generación de recomendaciones.

Capa de Datos: Gestiona el almacenamiento y recuperación de información mediante una base de datos relacional optimizada para consultas complejas y análisis temporal.

Capa de Integración: Facilita la conectividad con APIs externas de IA y servicios de terceros, permitiendo flexibilidad en la selección de proveedores de servicios.

Base de Datos Relacional Optimizada

El diseño de la base de datos modela eficientemente las relaciones complejas entre entidades empresariales:

- **Usuarios y Roles:** Implementación de RBAC (Role-Based Access Control) para gestión granular de permisos
- **Empresas y Clientes:** Soporte multi-tenant que permite a múltiples organizaciones utilizar la plataforma
- **Llamadas y Transcripciones:** Almacenamiento optimizado de contenido conversacional con indexación para búsquedas rápidas
- **Análisis y Métricas:** Estructuras de datos diseñadas para agregaciones eficientes y generación de reportes

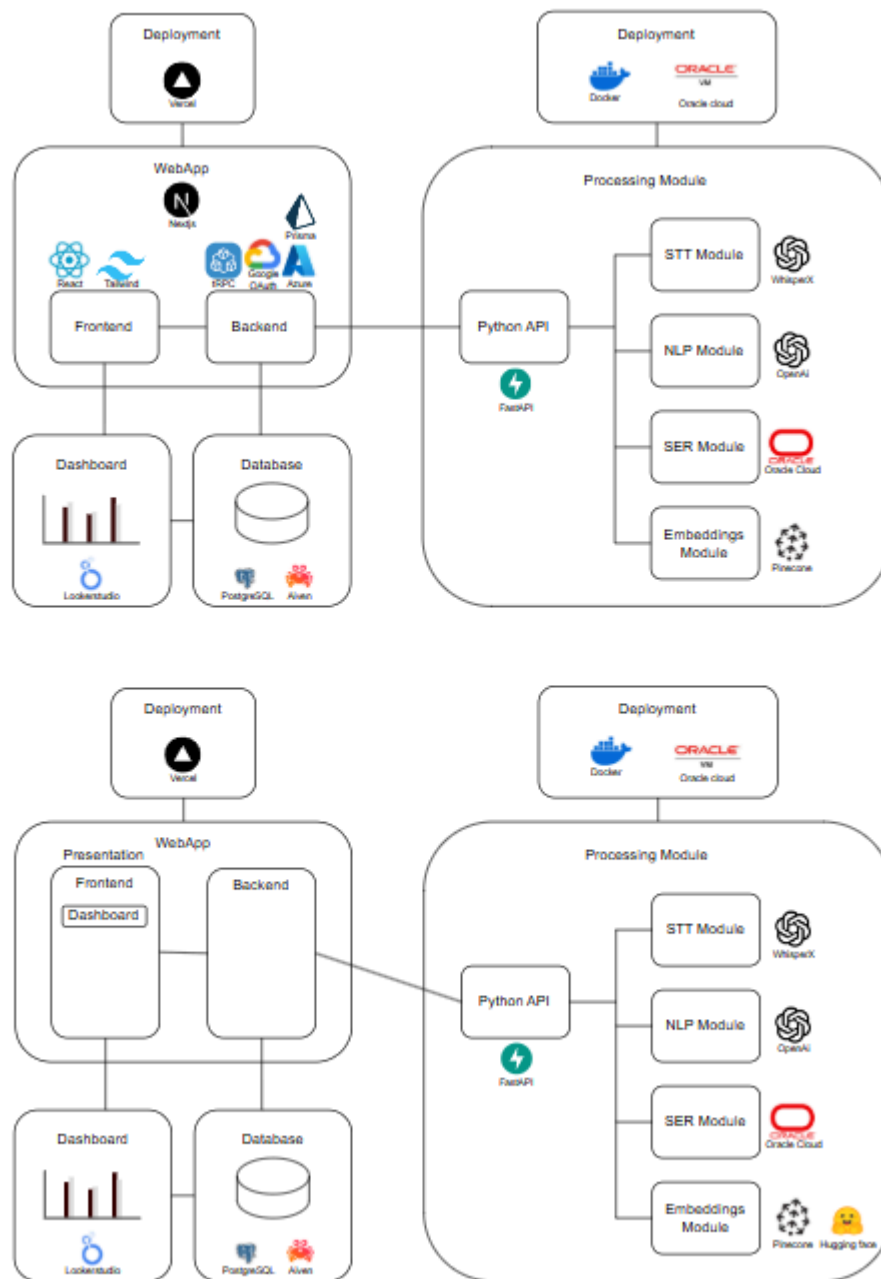


Imagen: Arquitectura del software.

Metodologías de Desarrollo y Calidad

Metodologías Ágiles y Scrum

El desarrollo de HowlX se llevó a cabo utilizando metodologías ágiles, específicamente Scrum, que permitió iteraciones rápidas y adaptación continua a los requerimientos cambiantes. Esta aproximación facilitó:

- **Sprints iterativos** con entregas incrementales de funcionalidad
- **Colaboración continua** entre miembros del equipo y stakeholders
- **Adaptabilidad** a cambios en requerimientos y prioridades
- **Mejora continua** mediante retrospectivas y refinamiento de procesos

Integración Continua y Control de Versiones

El proyecto implementa prácticas de DevOps modernas:

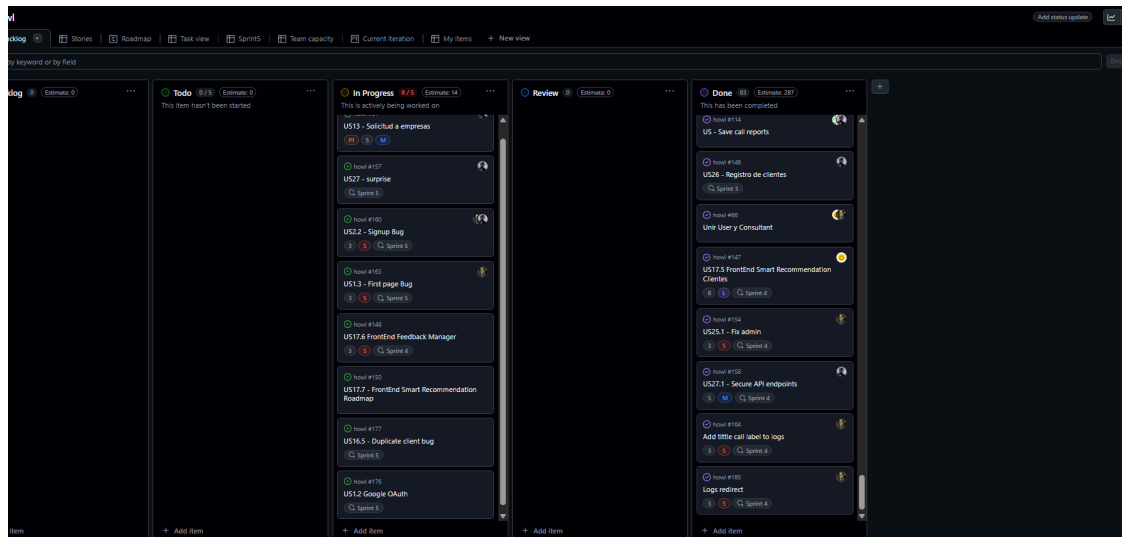
Control de Versiones Colaborativo: Utilización de Git y GitHub para gestión de código fuente, con workflows que facilitan la colaboración y revisión de código.

Integración Continua (CI/CD): Pipelines automatizados que ejecutan pruebas, validaciones de calidad y despliegues, asegurando la estabilidad del sistema en cada cambio en gi.

Pruebas Automatizadas: Implementación de suites de pruebas unitarias, de integración y de aceptación que garantizan la calidad del software y detectan regresiones tempranamente.

Sprint Backlog

Tomando en cuenta el objetivo, se seleccionaron las siguientes tasks para el backlog del sprint 4, asignando responsables, la prioridad y estimación.



Iteration: Sprint 4											
Title	Assignee	Type	Status	Milestone	P. S.T.2	Estimate	Size	Labels	Reas.	S.T.1	
1 US17 - Smart recommendations #78	monicasoberon	Spec	Done	Development	(90)		11	New Feature		Sprint 4	
2 US24 - Conexión AI con BD #132	adrian16 an...		Done	Development	(90)		8	New Feature		Sprint 4	
3 US17.4 - Fetch Call Data #149	Ale-Corbo	Issue	Done	Development	(90)		3	Enhancement		Sprint 4	
4 US26.1 - Update call report: add client field #131	Ale-Corbo	Issue	Done	Development	(90)		2	Bug		Sprint 4	
5 US17.41 Database Functions para recoleccion de datos de llamada... #169	monicasoberon		Done		(90)		1	Enhancement		Sprint 4	
6 US26 Sandbox AI Setup #173	adrian16		Done		(90)		8	Enhancement		Sprint 4	
7 US29 - Enhance AI #113	adrian16		Done	Development	(91)		8	Enhancement		Sprint 4	
8 US21 - AI modular input #117	adrian16		Done	Development	(92)		8	Enhancement		Sprint 4	
9 US19.3 - Frontend Forms #162	monicasoberon		Done				5	Enhancement		Sprint 4	
10 US17.8 Refactor Smart Insight backend for frontend #132	SantiagoDin		Done				3	Enhancement		Sprint 4	
11 US17.9 Feedback manager Backend #133	SantiagoDin		Done				5	Enhancement		Sprint 4	
12 US17.5 Frontend Smart Recommendation Clientes #147	monicasoberon		Done				8	Enhancement		Sprint 4	
13 US25.1 - Fix admin #154	LUISG57		Done				3	Enhancement		Sprint 4	
14 US17.6 Frontend Feedback Manager #148		In Prog...	Development				3	New Feature		Sprint 4	
15 US27.1 - Secure API endpoints #158	Ale-Corbo	Issue	Done				5	New Feature		Sprint 4	
16 Add title call label to logs #164	LUISG57		Done				3	Bug		Sprint 4	
17 Logs redirect #165	LUISG57		Done				2	Bug		Sprint 4	

Imagen 1 y 2: Sprint backlog en Github Projects.

De igual manera, se asignaron los criterios de aceptación en las tasks de historias de usuario:

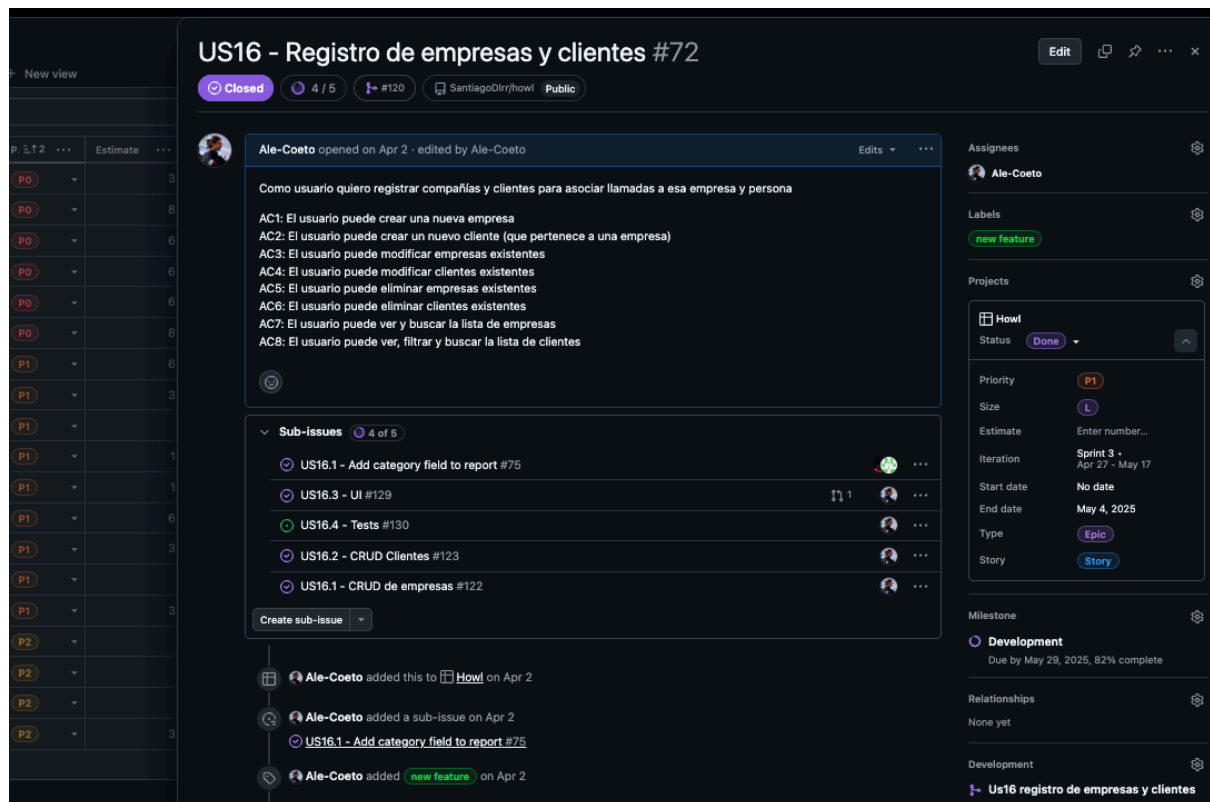


Imagen 3: Ejemplo de una historia de usuario.

[Github Project](#)

Daily Sessions

Para dar seguimiento del proyecto, se tuvieron daily sessions, dónde cada integrante comentaba su estatus, lo que había trabajado, lo que trabajará hoy y si tuvo algún impedimento.

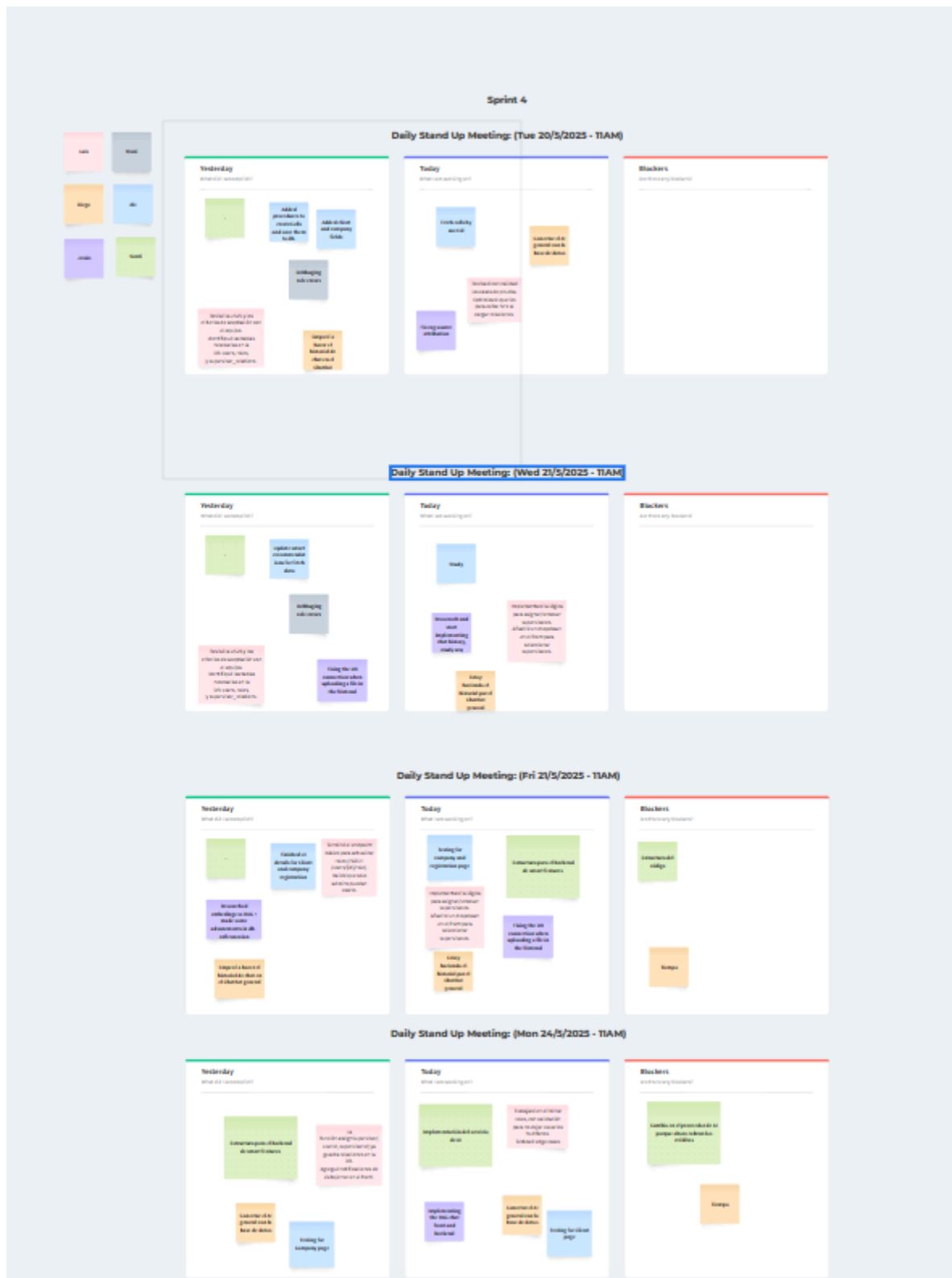


Imagen 4: Daily sessions del sprint 4.

2. No saltarnos dailys

Responsable: todos los integrantes. Cada quién debe comprometerse a asistir a las dailys.

3. Asegurar que CI/CD esté actualizado y las pruebas sean correctas

Responsables: Ale y Jesús. Dar seguimiento al CI/CD para verificar que valide correctamente los cambios.

Review Session

En este caso, no se identificaron riesgos que pudieran afectar la entrega del producto. Esto se debe al avance que se ha realizado y a que se ha identificado un ritmo constante de trabajo. Esto se puede ver reflejado en la gráfica de velocidad:

Gráfica de velocidad

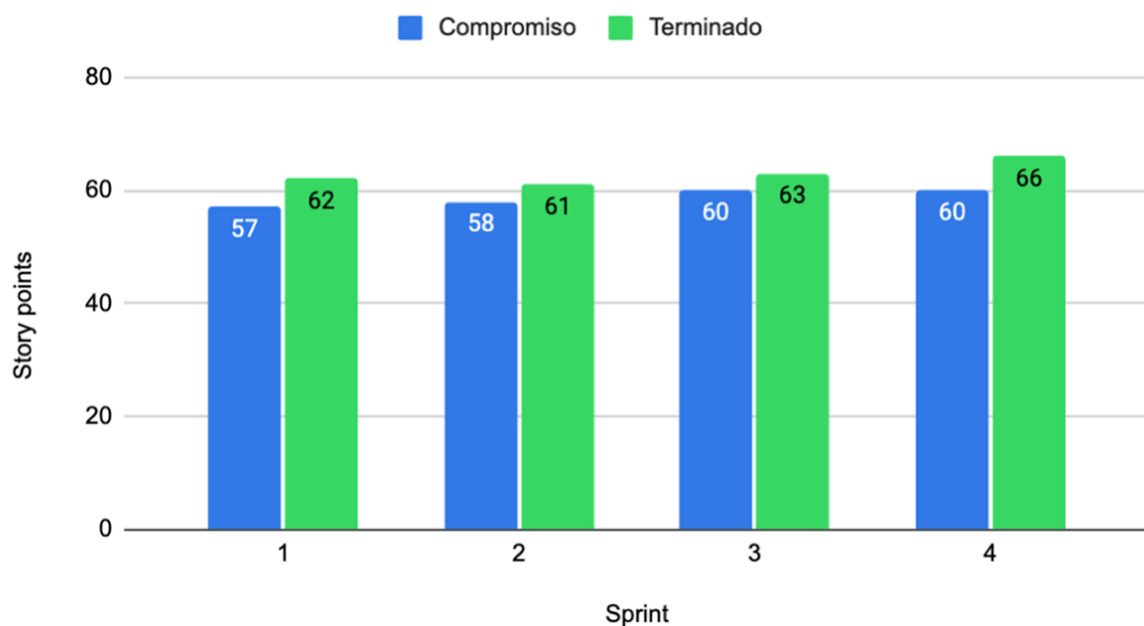


Imagen 7: Gráfica de velocidad.

Retro Neoris

Al finalizar la presentación, la retroalimentación fue bastante positiva y se nos recomendó seguir con el mismo ritmo para lograr finalizar el producto con buena calidad. De igual manera nos recomendaron presentar algunas métricas que nos llevaron a tomar algunas decisiones por ejemplo sobre los modelos de AI a utilizar, lo cual consideraremos para la siguiente presentación.

Estrategia Sprint 5

El enfoque principal será finalizar las mejoras a nuestra interfaz gráfica para el cliente, corregir pequeños errores presentes en el sistema y, lo más importante, realizar pruebas de aceptación, integración y seguridad para asegurarnos de que el producto entregado sea de alta calidad y utilidad para el cliente.

Hipótesis y Resultados

Planteamos la hipótesis de que el uso de la herramienta HowlX mejoraría el trato al cliente por parte de los empleados, especialmente en aspectos como la memoria de interacciones pasadas, el manejo emocional y la reducción de malentendidos relacionados con conversaciones previas.

Aunque no fue posible implementar la herramienta en un entorno laboral real para validar completamente esta hipótesis, realizamos una prueba en una junta de negocios simulada utilizando HowlX. En este contexto, observamos una mejora notable en la retención de detalles de las conversaciones y una disminución en los malentendidos sobre lo discutido anteriormente, lo cual respalda parcialmente nuestra hipótesis.

Conclusiones

El proyecto logró cumplir su objetivo principal: desarrollar una plataforma capaz de transformar conversaciones en conocimiento accionable. A través del uso de inteligencia artificial, se logró extraer información valiosa de llamadas con clientes, optimizando los procesos de análisis y toma de decisiones.

El impacto fue notable en la mejora de la atención al cliente, gracias al acceso a recomendaciones personalizadas, alertas tempranas sobre posibles problemas y un entendimiento más profundo del estado emocional del usuario. Estas funcionalidades fortalecen la relación con los clientes y permiten una atención más empática y efectiva.

La arquitectura del sistema demostró ser altamente flexible, permitiendo una integración ágil con plataformas de videollamada, servicios de IA y flujos empresariales existentes. Esta adaptabilidad amplía el rango de aplicación de la plataforma sin requerir grandes modificaciones en los entornos actuales.

Próximos pasos

Para mejorar el proyecto, una posible línea de evolución sería implementar procesamiento en tiempo real, lo que permitiría generar alertas y análisis durante el transcurso de una llamada. Asimismo, se podría ampliar la cobertura del sistema hacia otros canales de comunicación, como chat y correo electrónico, enriqueciendo así el análisis multicanal.

Otra oportunidad de mejora sería desarrollar modelos de inteligencia artificial especializados por industria, lo que permitiría adaptar los análisis a contextos específicos y mejorar la precisión de las recomendaciones. También podría incorporarse análisis predictivo, con el fin de anticipar comportamientos o resultados a partir de patrones históricos.

Finalmente, el proyecto podría escalar su impacto mediante la integración con sistemas CRM existentes, la creación de APIs públicas que faciliten su adopción por terceros, y el fortalecimiento de sus capacidades de analítica avanzada para apoyar decisiones estratégicas en entornos empresariales complejos.

Lecciones aprendidas por el equipo

Ale Coeto: Este fue el cuarto sprint y sin duda uno de los más productivos hasta ahora. Logramos cerrar muchas funcionalidades clave, lo que dejó la aplicación prácticamente terminada. Creo que gran parte de este avance se debe a una mejor planificación desde el inicio, con objetivos claros y bien definidos. Además, ver tantos progresos reflejados en el proyecto fue una gran fuente de motivación para todos, ya que impulsa el compromiso individual y colectivo por entregar un producto sólido. También considero que el hecho de estar aplicando buenas prácticas ha facilitado bastante el trabajo, especialmente en aspectos como la integración del código, que ahora resulta mucho más fluida.

Luis Juarez: En este sprint trabajé más en algunos cambios del front end y arreglando bugs refinando un poco lo que llegue a hacer durante sprints pasados, entre algunas cosas que hice fue hacer que la pestaña de clientes y role management fuera solo visible para usuario con rol de administrador y pase estos 2 permisos de la navbar al menú del perfil, a su vez implemente testing para role management en cypress y refactorize código que ya tenía en role management, también implemente una función de un botón que te permite ver un reporte pasado que se encuentre en logs reutilizando código que ya habían implementado mis compañeros, esta última parte me gustó bastante debido a que se hizo todo con buena planeación el código demostró ser escalable y fácil de implementar.

Jesus Lopez: Durante este sprint tuve la oportunidad de profundizar en los riesgos asociados al desarrollo de interfaces que integran inteligencia artificial, particularmente en lo relacionado con ataques de prompt injection. Realicé una investigación sobre diversos casos en los que estas vulnerabilidades han sido explotadas, lo cual me permitió adquirir un entendimiento más sólido de los desafíos de seguridad en este ámbito. Esta experiencia representó un aprendizaje significativo que amplió considerablemente mis conocimientos, por lo cual me siento muy agradecido. Adicionalmente, participé en la implementación de varios cambios que requirieron trabajo tanto en la base de datos como en el frontend y backend. Esta integración de tareas multidisciplinarias resultó especialmente enriquecedora y estimulante desde el punto de vista del desarrollo profesional.

Diego Esparza: En este sprint trabajé en finalizar ya todo el front del Chatbot general y arreglé los bugs de sign out y estoy trabajando en otros bugs como el del chat de la ai en

llamadas, también ya acabé lo que es toda la página de perfil con todo funcional, en este sprint sigo aprendiendo mucho sobre back end que es en lo que aun flaqueo mucho pero me está sirviendo mucho el terminar todo lo que es el chatbot general.

Mónica Soberón: En este sprint trabajé principalmente en documentación y frontend. Aun que estas dos partes no son mis fuertes, ni las que más disfruto, considere muy importante trabajar en ellas para tener conocimiento sobre todas las partes del proyecto.

Santiago De La Riva: Aprendí mucho durante este sprint, fue uno de los más productivos donde pude terminar de desarrollar el módulo que tenía que desarrollar. Disfruté de diseñar e implementar toda la funcionalidad, a pesar de que representó un desafío muy complejo, quedé satisfecho con el producto final, y gracias a todas las técnicas y metodologías que hemos visto en clase, la comunicación, desarrollo y planeación de todo el proyecto fluye de forma natural . Me entusiasma presentar mis avances durante la presentación final.