



*myMind: Plataforma móvil para la detección y seguimiento de emociones*

**Documento SAD**

**Enero 2025**

**Versión 3**

**Natalia Mejía Estrada**

**Luis Fernando Lee Rodríguez**

**Juan David Castillo Laverde**

**Juan José Gómez Arenas**

**Planeación de Proyecto de Grado**

**PONTIFICIA UNIVERSIDAD JAVERIANA**

**BOGOTÁ D.C**

Sección	Fecha	Sección del documento modificada	Descripción del Cambio	Responsables
1.	16/02/2025	Objetivo Atributos de calidad	Documento inicial	Juan José Gómez Arenas
2.	18/02/2025	Atributos de calidad Arquitectura	Definición arquitectura y atributos de calidad	Juan José Gómez Arenas
3.	22/02/2025	Arquitectura	Corrección arquitectura basada en retroalimentación	Juan José Gómez Arenas

## Contenido

<b>Objetivo</b> .....	4
<b>Atributos de Calidad</b> .....	4
Descripción de atributos de calidad .....	4
Atributos de calidad en myMind .....	5
Funcionalidad .....	5
Desempeño .....	5
Usabilidad .....	5
Fiabilidad .....	5
Seguridad .....	6
Mantenibilidad .....	6
Escalabilidad .....	6
Disponibilidad .....	6
Priorización de atributos de calidad .....	6
Prioridad Alta .....	6
Prioridad Media .....	7
Prioridad Baja .....	7
<b>Arquitectura</b> .....	11
Descripción plataforma .....	11
Decisiones arquitectónicas .....	11
Arquitectura basada en servicios .....	11
Consideraciones de diseño y mitigación de limitaciones .....	12
Diagrama de arquitectura de alto nivel .....	13
<b>Riesgos</b> .....	13
<b>Restricciones</b> .....	15
<b>Referencias</b> .....	16

# Objetivo

El presente documento tiene como propósito ofrecer una visión detallada de la arquitectura del sistema de myMind, abordando aspectos clave como los atributos de calidad, la arquitectura de alto nivel y los factores de riesgo y restricciones asociados. Se establecerá una estructura clara del sistema, alineada con sus objetivos y requisitos arquitectónicos, tanto funcionales como no funcionales. A lo largo del documento, se analizarán las decisiones arquitectónicas tomadas, justificando su elección y evaluando su impacto en el desarrollo del proyecto. Además, se incluirán representaciones gráficas para facilitar la comprensión de la estructura del sistema, y se definirán los pasos a seguir para asegurar que la arquitectura se mantenga alineada con los objetivos estratégicos de myMind.

## Atributos de Calidad

Los atributos de calidad son características esenciales de un sistema de software que determinan su comportamiento y desempeño más allá de sus funcionalidades principales. Estos atributos permiten evaluar el sistema en términos de factores como eficiencia, seguridad, mantenibilidad y escalabilidad, asegurando que la aplicación cumpla con los requerimientos tanto funcionales como no funcionales.

En arquitectura de software, cada decisión conlleva “trade-offs”, lo que implica que mejorar un atributo de calidad puede afectar negativamente a otro. Por ejemplo, aumentar la seguridad del sistema mediante mecanismos de cifrado puede impactar el rendimiento al requerir mayor capacidad de procesamiento. De esta manera, el diseño arquitectónico debe encontrar un balance adecuado entre estos atributos, alineándose con los objetivos del sistema y las necesidades del usuario.

## Descripción de atributos de calidad

Para estructurar la evaluación de los atributos de calidad, se utilizará el marco de referencia de la norma ISO 25010, que define diferentes categorías de atributos de calidad, cada una con subcaracterísticas específicas. En el contexto de myMind, se han identificado los siguientes atributos como los más relevantes para la arquitectura del sistema:

- *Funcionalidad*: Evalúa si el sistema proporciona las funciones necesarias para cumplir con los objetivos del usuario de manera precisa y completa.
- *Desempeño*: Analiza el uso eficiente de los recursos y el tiempo de respuesta del sistema al ejecutar sus funciones.
- *Usabilidad*: Determina la facilidad con la que los usuarios pueden interactuar con el sistema y aprender a utilizar sus funcionalidades.
- *Fiabilidad*: Examina la estabilidad del sistema y su capacidad para operar sin fallos o interrupciones prolongadas.
- *Seguridad*: Garantiza la protección de los datos del usuario, la prevención de accesos no autorizados y la resistencia a ataques maliciosos.

- *Mantenibilidad*: Evalúa la facilidad con la que el sistema puede ser actualizado, corregido y adaptado a nuevas necesidades sin comprometer su estabilidad.
- *Escalabilidad*: Determina la capacidad del sistema para manejar un crecimiento en la cantidad de usuarios o volumen de datos sin degradar su rendimiento.
- *Disponibilidad*: Mide la capacidad del sistema para estar operativo y accesible en todo momento, minimizando los tiempos de inactividad.

## Atributos de calidad en myMind

A continuación, se describe cómo cada uno de estos atributos de calidad se aplican específicamente en el contexto de myMind:

### Funcionalidad

myMind debe garantizar que todas las funciones necesarias para el análisis de emociones sean implementadas de manera completa y precisa. Esto incluye la grabación de notas de voz, la transcripción precisa del contenido, el análisis de emociones mediante NLP y la generación de reportes comprensibles. La funcionalidad del sistema debe estar alineada con las necesidades del usuario, asegurando que los resultados obtenidos sean relevantes y útiles para el seguimiento emocional.

### Desempeño

El desempeño es crítico para proporcionar una experiencia fluida a los usuarios. El sistema debe ser capaz de procesar la transcripción de voz en tiempo real o en el menor tiempo posible, así como realizar análisis de NLP sin generar demoras significativas. Además, debe gestionar los recursos computacionales de manera eficiente, evitando el uso excesivo de memoria o procesamiento que pueda afectar la escalabilidad y la estabilidad del sistema.

### Usabilidad

Dado que la aplicación está dirigida a usuarios que pueden no tener conocimientos técnicos, la usabilidad es un factor clave. La interfaz debe ser clara y fácil de navegar, permitiendo que los usuarios accedan rápidamente a sus grabaciones y reportes. Además, las representaciones visuales de los análisis emocionales deben ser comprensibles y permitir que el usuario identifique patrones sin necesidad de un entrenamiento previo.

### Fiabilidad

La fiabilidad implica que el sistema debe ser capaz de operar de manera continua y sin errores críticos. La pérdida de datos debe minimizarse a través de mecanismos de respaldo y recuperación, asegurando que los registros de los usuarios sean almacenados de manera segura y accesible en todo momento. Además, el sistema debe ser resistente a fallos inesperados y capaz de mantener su operatividad bajo condiciones normales de uso.

## Seguridad

La aplicación debe garantizar la protección de los datos, asegurando que la información almacenada no pueda ser accedida sin autorización. Además, debe implementar mecanismos de control de acceso y autenticación segura para restringir el acceso a los datos, de manera que los usuarios únicamente tengan acceso a su propia información y no a la de algún otro usuario.

## Mantenibilidad

El sistema debe estar diseñado de manera modular para facilitar su mantenimiento y evolución. La implementación de nuevas funcionalidades o la corrección de errores debe realizarse sin afectar la estabilidad del sistema. Además, la documentación del código y la arquitectura deben mantenerse actualizadas para permitir una rápida comprensión por parte de los desarrolladores.

## Escalabilidad

El sistema debe ser capaz de manejar un crecimiento progresivo en la cantidad de usuarios sin afectar su desempeño. La arquitectura del sistema debe permitir la distribución eficiente de la carga de trabajo, utilizando tecnologías que soporten la escalabilidad horizontal y vertical.

## Disponibilidad

La disponibilidad es un atributo fundamental, ya que los usuarios deben poder acceder a la aplicación en cualquier momento. Para ello, el sistema debe contar con mecanismos de redundancia y recuperación ante fallos, minimizando los tiempos de inactividad. Además, la infraestructura debe ser capaz de manejar picos de tráfico sin degradar la experiencia del usuario.

## Priorización de atributos de calidad

Para garantizar que myMind cumpla con sus objetivos y ofrezca una experiencia estable y confiable, es esencial priorizar los atributos de calidad en función de su impacto en el sistema. La siguiente clasificación justifica la relevancia de cada atributo dentro del contexto de la aplicación.

### Prioridad Alta

La funcionalidad es un aspecto crítico, ya que myMind debe garantizar que todas las funciones esenciales para el análisis emocional sean implementadas de manera completa y precisa. Esto incluye la grabación de notas de voz, la transcripción del contenido y el análisis de emociones mediante NLP. Dado que el propósito de la aplicación es permitir que los estudiantes universitarios registren y den seguimiento a sus emociones, cualquier falla en estos procesos afectaría directamente la utilidad del sistema. Además, la funcionalidad debe alinearse con las necesidades de los usuarios, asegurando que los resultados obtenidos sean relevantes y aplicables para el seguimiento emocional.

Con relación a la fiabilidad, este es el uno de los atributos más críticos en el sistema, ya que los registros emocionales de los usuarios representan información valiosa para su bienestar y corresponde al eje central de la aplicación. El sistema debe ser capaz de operar de manera continua y sin errores que puedan comprometer la integridad de los datos almacenados. La pérdida de registros o fallos en la recuperación de información pueden afectar negativamente la experiencia del usuario. Además, los mecanismos de respaldo y recuperación ante fallos son fundamentales para garantizar la persistencia de los datos a lo largo del tiempo.

Por otra parte, la disponibilidad también es clave, dado que los usuarios pueden necesitar acceder a sus registros emocionales en cualquier momento. Para ello, el sistema debe minimizar los tiempos de inactividad y contar con una infraestructura capaz de manejar solicitudes de acceso sin interrupciones. La redundancia en los servicios y estrategias de recuperación ante fallos son necesarias para garantizar que los usuarios siempre puedan interactuar con la aplicación cuando lo requieran.

La seguridad en este contexto es esencial debido a la naturaleza sensible de los datos que gestiona la aplicación. La información relacionada con el estado emocional de los usuarios debe estar protegida, así como a través de controles de acceso estrictos para evitar accesos no autorizados. Adicionalmente, es crucial implementar mecanismos de autenticación robustos y monitoreo de actividad para detectar y mitigar posibles amenazas de seguridad.

## Prioridad Media

El desempeño del sistema es clave para garantizar una experiencia de usuario fluida. La transcripción de voz y el análisis de NLP deben realizarse en tiempos razonables, evitando retrasos que puedan afectar la percepción de eficiencia del sistema.

Por otra parte, la usabilidad es fundamental para garantizar que los estudiantes universitarios puedan interactuar con la aplicación sin dificultades. Dado que este grupo suele estar familiarizado con la tecnología, pero enfrenta altos niveles de estrés y carga académica, la interfaz debe ser eficiente y directa, evitando elementos que dificulten la navegación. Las funciones principales, como la grabación de notas de voz y la visualización de reportes emocionales, deben ser accesibles con la menor cantidad de pasos posible. Además, los reportes emocionales deben presentarse en un formato visual claro y de rápida interpretación, permitiendo que los estudiantes identifiquen patrones emocionales sin necesidad de dedicar demasiado tiempo a su análisis.

## Prioridad Baja

La escalabilidad es importante en el largo plazo para garantizar que se pueda soportar un crecimiento en la cantidad de usuarios sin degradar su rendimiento. No obstante, en el contexto de este proyecto, el enfoque principal debe estar en la estabilidad y confiabilidad del sistema antes de optimizarlo para manejar grandes volúmenes de datos. La arquitectura debe estar diseñada de manera que permita la expansión futura sin requerir cambios disruptivos.

Finalmente, la mantenibilidad es relevante para la evolución de la aplicación, ya que un diseño modular facilita la implementación de mejoras y la corrección de errores sin afectar la estabilidad del sistema. Sin embargo, en la fase inicial, se puede gestionar progresivamente siempre que se sigan buenas prácticas de desarrollo y documentación.

## Escenarios de calidad

Escenario de calidad N-1	
<i>Funcionalidad</i>	
El sistema debe permitir a los usuarios subir notas de voz, transcribirlas con precisión y generar análisis emocionales relevantes.	
<b>Fuente del estímulo</b>	Un usuario usa la funcionalidad de grabación y análisis emocional.
<b>Estímulo</b>	Subida de un archivo de voz para su procesamiento.
<b>Artefacto</b>	Módulo de transcripción y análisis.
<b>Ambiente</b>	Operación normal del sistema
<b>Respuesta</b>	El sistema transcribe el contenido, identifica emociones y guarda el resultado en la base de datos.
<b>Medida de Respuesta</b>	El análisis debe completarse exitosamente en al menos el 95% de los casos.

Escenario de calidad N-2	
<i>Desempeño</i>	
El sistema debe procesar transcripciones y análisis emocionales sin generar demoras perceptibles.	
<b>Fuente del estímulo</b>	Un usuario solicita el análisis emocional de una nota de voz.
<b>Estímulo</b>	Petición de transcripción de audio.
<b>Artefacto</b>	Modelo de análisis NLP.
<b>Ambiente</b>	Condiciones normales de carga ( $\leq 50$ usuarios concurrentes).
<b>Respuesta</b>	El sistema entrega el análisis emocional completo.
<b>Medida de Respuesta</b>	Respuesta en $\leq 10$ segundos

Escenario de calidad N-3	
<i>Fiabilidad</i>	
El sistema debe continuar operando de forma estable ante entradas de baja calidad sin generar errores críticos.	
<b>Fuente del estímulo</b>	Un usuario envía una nota de voz con bajo volumen y ruido ambiental.
<b>Estímulo</b>	Petición de transcripción de una nota con calidad de audio deficiente.
<b>Artefacto</b>	Módulo de transcripción y análisis.
<b>Ambiente</b>	Condiciones normales de carga ( $\leq 50$ usuarios concurrentes).



<b>Respuesta</b>	El sistema procesa la transcripción sin fallos y retorna un análisis, aunque impreciso, sin interrupciones ni pérdida de servicio.
<b>Medida de Respuesta</b>	Sin errores críticos ni caídas del servicio; la operación completa exitosamente en $\leq 15$ segundos.

<b>Escenario de calidad N-4</b>	
<i>Desempeño</i>	
El sistema debe procesar la transcripción y análisis en un tiempo aceptable.	
<b>Fuente del estímulo</b>	Un usuario graba una nota de voz estándar ( $\leq 30$ segundos).
<b>Estímulo</b>	Petición de análisis emocional a partir del audio.
<b>Artefacto</b>	Módulo de transcripción y análisis
<b>Ambiente</b>	Condiciones normales de carga ( $\leq 50$ usuarios concurrentes).
<b>Respuesta</b>	El sistema completa la transcripción y devuelve el análisis emocional asociado sin demoras significativas.
<b>Medida de Respuesta</b>	Transcripción y análisis entregados en $\leq 10$ segundos.

<b>Escenario de calidad N-5</b>	
<i>Seguridad</i>	
El sistema debe garantizar que los usuarios no puedan acceder a los datos de otros.	
<b>Fuente del estímulo</b>	Un usuario no-autenticado intenta acceder a una transcripción que no le pertenece
<b>Estímulo</b>	Petición de transcripciones con un token inválido.
<b>Artefacto</b>	Módulo de usuarios
<b>Ambiente</b>	Condiciones normales de uso con múltiples usuarios autenticados.
<b>Respuesta</b>	El sistema bloquea el acceso, retorna un error 401 (Unauthorized).
<b>Medida de Respuesta</b>	Acceso denegado en 100% de los casos no autorizados; sin exposición de datos ajenos.

<b>Escenario de calidad N-6</b>	
<i>Usabilidad</i>	
El usuario debe poder visualizar los análisis emocionales consolidados desde la aplicación móvil	
<b>Fuente del estímulo</b>	Un usuario autenticado solicita ver el análisis emocional de la última semana.
<b>Estímulo</b>	Petición de registro de última semana

<b>Artefacto</b>	Módulo de visualizaciones
<b>Ambiente</b>	Condiciones normales, peticiones autenticadas y backend activo.
<b>Respuesta</b>	El sistema muestra de forma clara y estructurada las emociones predominantes por día, usando visualizaciones.
<b>Medida de Respuesta</b>	Visualizaciones generadas en el frontend en $\geq$ 95% de los casos

Escenario de calidad N-7	
<i>Usabilidad</i>	
El sistema debe mantenerse disponible para los usuarios incluso bajo condiciones de carga.	
<b>Fuente del estímulo</b>	Un conjunto de usuarios accede simultáneamente a funcionalidades de transcripción y visualización.
<b>Estímulo</b>	50 peticiones concurrentes
<b>Artefacto</b>	Infraestructura completa del sistema: APISIX + Backend + MongoDB + Celery.
<b>Ambiente</b>	Condiciones normales, peticiones autenticadas y backend activo.
<b>Respuesta</b>	El sistema responde correctamente a las peticiones sin fallos ni interrupciones del servicio. (HTTP 200)
<b>Medida de Respuesta</b>	$\geq$ 95% de las peticiones completadas exitosamente bajo carga media, sin caídas de servicio.

Escenario de calidad N-8	
<i>Seguridad</i>	
El sistema debe aceptar peticiones únicamente del Api Gateway	
<b>Fuente del estímulo</b>	Un usuario intenta acceder directamente a los servicios sin haber sido autenticado.
<b>Estímulo</b>	Petición para obtener datos demográficos de usuario
<b>Artefacto</b>	Módulo de Usuarios
<b>Ambiente</b>	Condiciones normales
<b>Respuesta</b>	El sistema responde a las peticiones con un error de origen invalido. (HTTP 403)
<b>Medida de Respuesta</b>	100% de las peticiones de origen invalido son bloqueadas.

# Arquitectura

## Descripción plataforma

La plataforma está diseñada como una aplicación móvil para brindar a los estudiantes universitarios una herramienta accesible y flexible que les permita registrar y analizar sus emociones en cualquier momento. A través de una interfaz intuitiva, los usuarios podrán acceder a diversas funcionalidades, incluyendo la grabación de notas de voz, la visualización de reportes emocionales y el seguimiento de tendencias en su estado emocional.

La decisión de desarrollar myMind como una aplicación móvil responde a la necesidad de accesibilidad, flexibilidad y facilidad de uso, asegurando que los estudiantes puedan interactuar con la plataforma cuando lo consideren necesario, sin restricciones de ubicación o tiempo. La interfaz estará diseñada para minimizar la fricción en la navegación, permitiendo que los usuarios accedan rápidamente a sus funcionalidades sin procesos complejos. Esta aproximación permite que la aplicación se adapte a la rutina universitaria, en la que los momentos de reflexión pueden surgir en cualquier lugar, ya sea en la biblioteca, en casa o en un descanso entre clases, garantizando una experiencia intuitiva y fluida.

En términos de experiencia de usuario, la interfaz de la aplicación estará diseñada para facilitar el acceso inmediato a sus funciones principales, reduciendo la fricción en la interacción con la plataforma. Los usuarios podrán grabar sus notas de voz con pocos pasos, revisar sus reportes emocionales de manera visual y comprender la evolución de su estado emocional sin requerir conocimientos técnicos.

Esta estructura busca maximizar la disponibilidad y usabilidad de la aplicación, asegurando que los usuarios puedan acceder a sus registros sin interrupciones y sin necesidad de configuraciones complejas. Al priorizar la accesibilidad móvil, la plataforma se convierte en una herramienta útil y práctica para los estudiantes, alineándose con su estilo de vida dinámico y sus necesidades emocionales.

## Decisiones arquitectónicas

La arquitectura de myMind adopta un enfoque basado en microservicios, combinando servicios desacoplados que optimizan la escalabilidad y el mantenimiento del sistema. Se ha diseñado una arquitectura distribuida pragmática, evitando la sobrecarga operativa de un sistema completamente basado en servicios, pero manteniendo la modularidad y separación de responsabilidades.

Se priorizó la independencia de cada componente, asegurando la escalabilidad y la fiabilidad del sistema a largo plazo.

## Arquitectura basada en servicios

La aplicación sigue los principios de la arquitectura basada en microservicios, lo que permite dividir la funcionalidad en servicios independientes alineados con dominios específicos del negocio. Este enfoque garantiza una mayor modularidad y facilita la evolución del sistema sin afectar otros componentes. Entre sus principales ventajas se encuentran:

- Flexibilidad y mantenibilidad: Cada microservicio puede actualizarse o reemplazarse sin afectar el funcionamiento global del sistema.
- Escalabilidad eficiente: Los servicios pueden escalarse de manera independiente en función de la demanda, optimizando el uso de recursos.
- Desacoplamiento y resiliencia: La independencia de cada microservicio reduce el impacto de fallos en el sistema, mejorando su disponibilidad y confiabilidad.

Este enfoque sigue las recomendaciones establecidas por Richards & Ford (2023), quienes enfatizan que la elección de una arquitectura de servicios debe considerar la relación entre escalabilidad y complejidad operativa. Según estos autores, una segmentación excesiva puede generar sobrecarga en la comunicación entre servicios y aumentar la dificultad en la gestión. En este sentido, la arquitectura de myMind busca un equilibrio entre modularidad y simplicidad operativa, garantizando independencia entre servicios sin introducir una fragmentación innecesaria que afecte el desempeño y la mantenibilidad.

## Consideraciones de diseño y mitigación de limitaciones

Para abordar las limitaciones inherentes a esta arquitectura y mejorar tanto el rendimiento como la experiencia del usuario, se implementaron varias estrategias que optimizan la disponibilidad, eficiencia y respuesta del sistema, minimizando latencias y asegurando una interacción fluida con la aplicación.

1. *API Gateway con Balanceo de Carga Interno:* Se utiliza Apache APISIX como API Gateway, el cual gestiona las solicitudes de los usuarios y distribuye el tráfico de manera eficiente entre los microservicios. Gracias a sus capacidades de balanceo de carga interno, se evita la necesidad de un Load Balancer externo, reduciendo la complejidad y los puntos de fallo.
2. *Replicación y Alta Disponibilidad en Bases de Datos:* Se emplea un sistema híbrido de almacenamiento con NoSQL para gestionar los datos de emociones y un Data Warehouse en MySQL optimizado para consultas analíticas y utilizado una réplica ante posibles fallos. Ambas bases de datos cuentan con replicación y failover automático, lo que garantiza resiliencia y evita la pérdida de datos en caso de fallos.
3. *Procesamiento de Datos con ETL:* Para optimizar la integración entre los datos almacenados en NoSQL y el Data Warehouse, se implementa un proceso ETL basado en Apache Airflow, el cual permite transformar y migrar datos de manera estructurada, asegurando la integridad y disponibilidad de la información en los reportes.
4. *Contenedorización con Docker:* Todos los servicios están dockerizados, lo que facilita despliegues consistentes y escalables. Esto permite la ejecución de múltiples instancias de los microservicios en diferentes entornos sin conflictos de configuración.
5. *Monitoreo y Observabilidad en Tiempo Real:* Se integró Grafana y Prometheus para supervisar el estado y rendimiento de los servicios en producción. Además, se recopilan métricas clave como latencia de respuesta, uso de CPU/memoria y disponibilidad de los microservicios, permitiendo detectar problemas antes de que afecten la experiencia del usuario.
6. *Gestión de Autenticación Externa:* En lugar de manejar credenciales dentro del sistema, se implementa Auth0 con OAuth 2.0, delegando la autenticación a un proveedor externo.

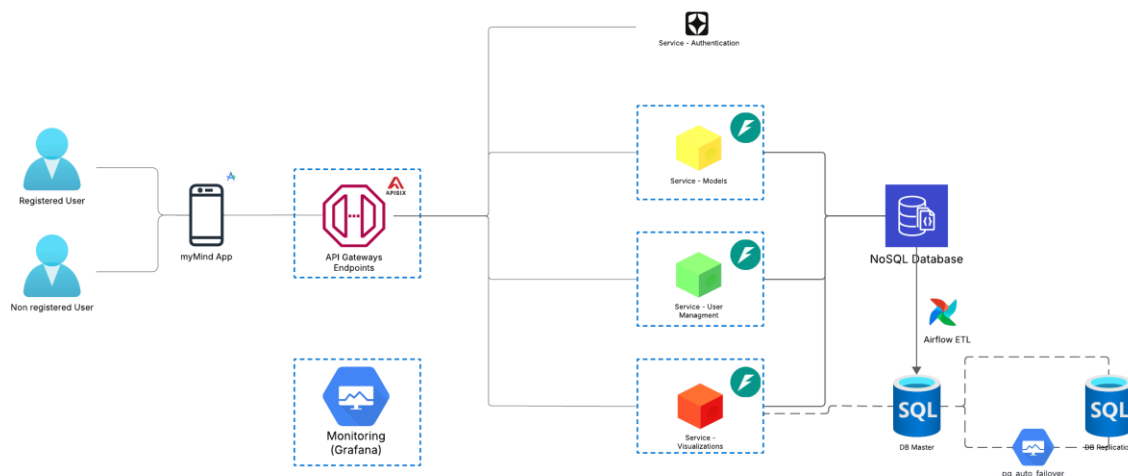
especializado. Esto no solo mejora la seguridad, sino que reduce la carga operativa en los microservicios.

Con estas estrategias, la arquitectura de myMind logra mitigar riesgos relacionados con disponibilidad, escalabilidad y seguridad, asegurando un funcionamiento óptimo y confiable del sistema.

## Diagrama de arquitectura de alto nivel

El sistema se estructura en una serie de servicios interconectados que optimizan el procesamiento y análisis de la información. La aplicación móvil actúa como punto de entrada, permitiendo a los usuarios grabar notas de voz que son enviadas a la infraestructura backend. Un API Gateway gestiona el tráfico y la autenticación, asegurando una distribución eficiente de las solicitudes. La gestión de usuarios se maneja a través de un servicio externo.

El procesamiento de las notas de voz se realiza en un servicio especializado para el manejo de modelos de IA. La información procesada se almacena en un Data Warehouse, permitiendo la generación de reportes y visualización de tendencias emocionales.



Esta arquitectura garantiza un sistema escalable, fiable y seguro, alineado con los atributos de calidad prioritizados. La replicación de bases de datos y los mecanismos de failover refuerzan la fiabilidad, reduciendo el riesgo de pérdida de información. La alta disponibilidad se asegura mediante balanceadores de carga y múltiples instancias de servicios clave, evitando puntos únicos de fallo y optimizando el rendimiento del sistema.

## Riesgos

Los riesgos identificados se agrupan en tres categorías principales: riesgos de producto, riesgos de proceso y riesgos de proyecto. Cada uno de ellos puede afectar de forma directa o indirecta la calidad, estabilidad y éxito del sistema de análisis emocional que se está desarrollando.

## *Riesgos de Producto*

Estos riesgos se relacionan con la calidad, funcionalidad, seguridad y rendimiento del sistema final.

### Riesgos de Seguridad:

- Posibles vulnerabilidades en las rutas expuestas a través de APISIX, como inyecciones en los parámetros, ataques de tipo XSS o uso inadecuado del plugin openid-connect.
- Errores en la validación de tokens de acceso, lo que podría permitir accesos no autorizados a transcripciones u otra información sensible.
- Gestión deficiente del almacenamiento de datos personales, como registros de voz o resultados emocionales, sin un adecuado cifrado o control de acceso.

### Riesgos de Rendimiento:

- Latencia en el procesamiento de transcripciones debido a la arquitectura distribuida (FastAPI + Celery + MongoDB).
- Sobrecarga del sistema durante picos de peticiones simultáneas, especialmente en el análisis emocional, que puede ser computacionalmente costoso.
- Cuellos de botella en la comunicación entre servicios (por ejemplo, entre APISIX y los contenedores del backend).

### Riesgos de Disponibilidad:

- Interrupciones del servicio debido a errores en la configuración de infraestructura, contenedores o la red compartida entre servicios.
- Tiempo de inactividad por reinicios no planificados o mantenimiento de servicios fundamentales como MongoDB o APISIX.
- Dependencia de servicios externos (por ejemplo, Auth0) cuya caída puede afectar la autenticación y, por ende, el acceso al sistema.

## *Riesgos de Proceso*

Estos riesgos están asociados al desarrollo, integración y mantenimiento del sistema.

### Riesgos de Mantenibilidad:

- Aumento en la complejidad del código base, especialmente por la integración de múltiples tecnologías (FastAPI, Celery, MongoDB, Prometheus, etc.), lo que puede dificultar la localización y corrección de errores.
- Cambios o actualizaciones en las librerías utilizadas (por ejemplo, modelos de NLP o bibliotecas de métricas) que generen incompatibilidades o introduzcan nuevos errores.

- Falta de documentación técnica clara sobre la arquitectura y la configuración, lo que puede dificultar futuras modificaciones o el traspaso del proyecto a nuevos desarrolladores.

### Riesgos relacionados con la implementación:

- Posible falta de experiencia del equipo con herramientas como Celery, Prometheus o configuración avanzada de APISIX, lo que puede ocasionar errores de diseño o implementación.
- Disponibilidad limitada de tiempo por parte de los desarrolladores, lo que puede afectar la cobertura de pruebas, la implementación de casos borde o la entrega oportuna de funcionalidades clave.

## *Riesgos de Proyecto*

Estos riesgos corresponden a factores externos o limitaciones generales que pueden afectar el cumplimiento de los objetivos del proyecto.

### Recursos Financieros Limitados:

- Restricciones presupuestarias que impidan acceder a servicios de infraestructura más robustos, suscripciones avanzadas de APIs externas (como Auth0), o mayor capacidad de cómputo en la nube.

### Disponibilidad de Recursos:

- Limitaciones en el uso de herramientas clave (como licencias de dashboards de visualización o monitoreo), que podrían restringir la capacidad de observabilidad del sistema o la generación de reportes gráficos.
- Falta de acceso a entornos de prueba replicables (por ejemplo, staging o integración continua) para garantizar la calidad del software antes del despliegue.

### Limitaciones de Hardware:

- Las máquinas virtuales disponibles para pruebas o despliegue podrían no contar con suficiente capacidad de procesamiento, memoria o almacenamiento, afectando directamente el rendimiento del sistema, especialmente durante la transcripción de audio o análisis en lote.

## **Restricciones**

Estas son limitaciones específicas que afectan la capacidad del sistema para cumplir con ciertos requisitos o estándares, y deben ser consideradas durante el desarrollo.

### **1. Cumplimiento Normativo:**

- a. El sistema debe cumplir con regulaciones legales y normativas vigentes, como protección de datos y privacidad, lo que puede limitar el diseño y la implementación de ciertas funcionalidades o el manejo de información. Como es el caso de la Ley 1581 de 2012 referente al tratamiento de datos personales de los usuarios.

## **Referencias**

- Richards, M. and Ford, N. (2023) *Fundamentals of Software Architecture: An Engineering Approach*. Sebastopol, CA: O'Reilly Media, Inc.