

DISEÑO

1. Arquitectura del Sistema

La arquitectura de un sistema define cómo interactúan los distintos componentes del proyecto, organizando la estructura y las relaciones entre los módulos. En el caso de tu IA Baymax, podemos considerar una arquitectura por capas:

a) Capa de Presentación (Frontend)

Descripción: Es la interfaz de usuario, la parte visible donde los usuarios interactúan con Baymax, ya sea mediante una aplicación web o móvil y tiene las siguientes funcionalidades:

Recibe las consultas de los usuarios.

Pide información de usuario (siguiendo las normas de privacidad).

Valida la información para saber si es usuario afiliado o simplemente usuario.

Si es usuario afiliado despliega resultados médicos, citas, seguimiento psicológico e información pública.

Si es solo un usuario despliega resultados de información profesional sin compartir datos privados de alguna empresa

Provee un diseño amigable y sensible a las necesidades de usuarios, con accesibilidad

b) Capa de Lógica de Negocio (Backend)

Descripción: Procesa las solicitudes del frontend y las operaciones más complejas. Aquí es donde la IA de Baymax realiza consultas, genera respuestas y lleva el seguimiento de usuarios.

Funcionalidades:

Motor de IA que procesa la validación de usuario o usuario afiliado.

Motor de IA que procesa las consultas médicas y psicológicas.

Conexión y consulta a bases de datos (con permiso del usuario y según la ley).

Generación de recomendaciones informativas basadas en los síntomas y seguimiento.

c) Capa de Integración con Bases de Datos

Descripción: Esta capa gestiona la interacción con las bases de datos de las empresas de salud o bases de datos locales que Baymax puede consultar.

Funcionalidades:

Permite que Baymax acceda a la información médica de usuarios registrados en empresas de salud.

Garantiza el cumplimiento de normativas legales (como la ley de privacidad de datos).

d) Capa de Datos (Bases de Datos)

Descripción: Es donde se almacenan todos los datos que Baymax necesita consultar o gestionar.

Funcionalidades:

Base de datos interna de usuarios y consultas realizadas.

Base de datos externa (de empresas de salud) para usuarios afiliados.

2. Interfaces

Interfaz de usuario:

Se puede diseñar una interfaz simple con el clásico campo de texto donde el usuario se comunique y exprese cual sea su necesidad.

Debe ser responsiva y accesible para usuarios con distintas capacidades.

Integración con asistentes de voz podría mejorar la interacción.

Interfaz de consulta a bases de datos:

Una API segura que permita a Baymax conectarse a las bases de datos de las empresas de salud y siempre cifrando información de usuario.

Uso de estándares como OAuth para asegurar el acceso a la información sensible.

Interfaz de alimentación de datos (API externa):

API para la alimentación de información desde fuentes públicas o profesionales (internet, sistemas médicos) que Baymax pueda utilizar y la alimentación manual de la IA.

3. Modelo entidad-relación (E.R.)

El modelo entidad-relación de Baymax distingue entre usuarios comunes y usuarios afiliados a empresas de salud. Los usuarios comunes solo reciben información pública extraída de internet, mientras que los usuarios afiliados, tras una validación, pueden acceder a sus resultados médicos, citas y datos relacionados desde las bases de datos de empresas de salud. Baymax garantiza

el cumplimiento de normativas de privacidad, permitiendo acceso a la información sensible solo con el consentimiento del usuario. Las consultas, citas y resultados médicos se registran según el tipo de usuario y sus permisos.

4. Modelo direccionamiento

Entidad	Atributo	Descripción	Tipo de Dato
Usuario	ID	Identificador único del usuario	Integer
	Nombre	Nombre completo del usuario	String
	Email	Correo electrónico del usuario	String
	Es_Afiliado	Indica si el usuario está afiliado a una empresa de salud	Boolean
Usuario_Afiliado	Empresa_ID	Identificador de la empresa de salud a la que está afiliado	Integer (FK)
	Fecha_Afiliación	Fecha de afiliación del usuario	Date
	Permiso_Compartir_Información	Indica si el usuario permite compartir sus datos médicos	Boolean
Empresa_Salud	ID	Identificador único de la empresa	Integer
	Nombre	Nombre de la empresa de salud	String
	Base_Datos	URL o dirección de la base de datos de la empresa	String
	Normativas_Privacidad	Normativas de privacidad que rigen el uso de los datos	String
Consulta	ID	Identificador único de la consulta	Integer
	Usuario_ID	Identificador del usuario que realizó la consulta	Integer (FK)
	Sintomas	Detalles sobre los síntomas ingresados por el usuario	String

Entidad	Atributo	Descripción	Tipo de Dato
	Tipo_Consulta	Tipo de consulta: informativa o detallada	Enum (informativa, detallada)
	Respuesta	Respuesta generada por Baymax	String
Resultado_Médico	ID	Identificador del resultado médico	Integer
	Usuario_Afiliado_ID	Identificador del usuario afiliado al que pertenece el resultado	Integer (FK)
	Resultados	Detalles del resultado médico	String
	Fecha	Fecha del resultado médico	Date
Cita_Médica	ID	Identificador único de la cita médica	Integer
	Usuario_Afiliado_ID	Identificador del usuario afiliado que programó la cita	Integer (FK)
	Fecha_Cita	Fecha de la cita médica	Date
	Médico	Nombre del médico asignado a la cita	String
	Estado_Cita	Estado de la cita (confirmada, pendiente, cancelada)	Enum (confirmada, pendiente, cancelada)

5. Componentes del Sistema

- Lenguajes de Programación

Python: Ideal para el desarrollo de la lógica de IA. Es el lenguaje más utilizado en el campo de la inteligencia artificial y aprendizaje automático (machine learning) gracias a librerías como TensorFlow, Keras y Scikit-learn. Su integración con APIs y bases de datos es sencilla, lo que facilita la implementación de la lógica médica y psicológica de Baymax.

JavaScript (Node.js): Para el backend y la integración con bases de datos. Node.js permite manejar múltiples conexiones y solicitudes simultáneas de forma eficiente, siendo ideal para sistemas de consultas en tiempo real.

HTML, CSS, JavaScript (Frontend): Estas tecnologías son fundamentales para el desarrollo de la interfaz de usuario (web o móvil). HTML y CSS permiten el diseño de interfaces amigables, mientras que JavaScript (con frameworks como React o Vue.js) puede mejorar la interactividad.

Java: Si tu proyecto requiere más robustez y escalabilidad a nivel empresarial, Java es una opción sólida para el backend, además de ser compatible con aplicaciones críticas de salud y cumplir con estándares de seguridad.

- Sistema Gestor de Bases de Datos (SGBD)

PostgreSQL: Una base de datos relacional que ofrece robustez y seguridad, ideal para gestionar datos sensibles como información médica. Es compatible con operaciones complejas y permite el uso de JSON para almacenar datos no estructurados. Además, cuenta con mecanismos avanzados de control de acceso y auditoría, fundamentales para cumplir con normativas de privacidad.

- Herramientas de soporte

UML (Unified Modeling Language): Para la creación de diagramas que representen la arquitectura y los casos de uso de tu IA. UML es la herramienta estándar para modelar sistemas y te permitirá visualizar de manera clara los componentes de tu sistema (clases, relaciones, secuencias, etc.).

Microsoft Visio: Ideal para la elaboración de diagramas de flujo, arquitecturas de red, y seguridad. Te permite diseñar diagramas profesionales de forma rápida y con plantillas predefinidas. Es una herramienta muy útil para visualizar los procesos de tu sistema.

Lucidchart: Alternativa a Visio, con funcionalidad de colaboración en línea para crear diagramas UML, diagramas de flujo y modelos ER. Es intuitivo y facilita el trabajo en equipo.

TensorFlow/Keras: Para el desarrollo del motor de inteligencia artificial. TensorFlow es el framework más usado en IA y machine learning, mientras que Keras es su API más sencilla, ideal para prototipar y construir modelos avanzados.

Docker: Para garantizar que tu aplicación sea portátil y escalable. Docker permite encapsular todo el entorno del proyecto en contenedores, lo que

facilita el despliegue en diferentes servidores, asegurando que Baymax funcione de manera consistente.

Git/GitHub: Para control de versiones y colaboración en equipo. Git es esencial para el control de versiones del código fuente, y GitHub facilita la colaboración y el seguimiento del progreso del proyecto.

- Servicios en la Nube

AWS (Amazon Web Services) o Google Cloud: Ofrecen una variedad de servicios para proyectos basados en IA, desde almacenamiento de bases de datos hasta herramientas de machine learning y análisis de datos.

Además, cuentan con entornos seguros para cumplir con normativas de privacidad como HIPAA (para datos médicos).

- Seguridad y Cumplimiento de Normativas

OAuth 2.0: Protocolo estándar para autorización segura en aplicaciones web y móviles, asegurando que los datos médicos solo se compartan con el consentimiento del usuario.

HTTPS/SSL: Para asegurar que todas las comunicaciones entre el cliente y el servidor estén cifradas y sean seguras.

Normativas de privacidad: Asegúrate de que todas las soluciones de bases de datos y almacenamiento en la nube que utilices cumplan con normativas de privacidad de datos como GDPR (en Europa) o HIPAA (para EE. UU.).