

Asignatura: Módulos Técnicos

Actividad: Proyecto de Módulo 2.2 (Caso de Estudio)

Grado: II Año “B”

Especialidad: Desarrollo de Software (DSW)

Docente: Ing. Carlos Amilcar Chávez Iraheta

Grupo: #4	
N°	Apellidos//nombres
28	López Rodríguez Josué Iván
29	Lopez Vasquez Wilson Daniel
20	Grimaldi Hernández Anderson Isaac

Soyapango, 12 de junio de 2025

Índice

1. Introducción.....	3
1.1 Antecedentes.....	3
1.2 Fase del problema.....	4
1.3 Objetivos de negocio.....	5
1.4 VISIÓN DE LA SOLUCIÓN.....	6
1.4.1 Fase de visión.....	6
1.4.2. Características del sistema.....	6-7
1.4.3 Alcance.....	8
1.5. CONTEXTO DEL SISTEMA.....	9
1.5.1. Interesados.....	9
1.5.2. Diagrama de contexto.....	10
1.5.3. Entorno de operación.....	10-11
1.6. INFORMACIÓN ADICIONAL.....	12
2. Requerimientos de la arquitectura.....	13
2.1. Drivers funcionales.....	13-14
2.1.2. Elección de casos de uso primarios.....	15
2.2. Drivers de atributos de calidad.....	16-17
2.3. Drivers de restricciones.....	18
3. Diseño de la arquitectura.....	19
3.1. Primera iteración: estructuración general del sistema.....	19
3.3. Tercera iteración: desempeño en capa de datos.....	29-30
4. Documentación.....	31
4.1 Generar vistas candidatas.....	32-33
4.2 Combinar vistas.....	34-35
4.3 Priorizar las vistas.....	36
4.4 Ejemplo de vistas.....	37
4.4.3 Catálogo de elementos.....	37-42
5. Evaluación.....	44
5.1 Escenarios básicos o secundarios.....	51
5.2. Resultados de la evaluación.....	52

1. Introducción

El desarrollo del sistema Braincare responde a la necesidad de crear una plataforma digital accesible y confiable que contribuya a la salud mental de la población. Este proyecto busca ofrecer una solución tecnológica centrada en la educación, orientación y detección temprana de posibles trastornos mentales, promoviendo así la concientización y desestigmatización de la salud mental en la sociedad actual.

La plataforma se diseñará como un sitio web interactivo que pondrá a disposición de los usuarios contenido educativo basado en evidencia científica, pruebas de autoevaluación orientativas y un chatbot capaz de brindar asistencia básica. El sistema está enfocado en ofrecer información clara y recursos útiles para personas que desean conocer más sobre su estado emocional y tomar decisiones informadas sobre su salud mental.

En esta etapa, se desarrollará un prototipo funcional que permita evaluar la usabilidad, estructura y principales características del sistema, con miras a su posterior implementación completa. Este prototipo incluirá las funciones principales como el acceso a contenido informativo, herramientas de autoevaluación y el chatbot integrado, permitiendo así validar su utilidad en contextos reales.

1.1 Antecedentes

La salud mental se ha convertido en un desafío a nivel mundial, especialmente luego de la pandemia del COVID-19, que provocó un aumento del 30% en los casos de depresión y ansiedad en América Latina según (CEPAL, 2021). Este incremento ha dejado en evidencia las limitaciones que poseen los sistemas de salud tradicionales, que en muchos casos no han

podido escalar su capacidad de atención, dejando a millones de personas sin acceso a diagnóstico, orientación o tratamiento psicológico.

Aunque actualmente existen plataformas internacionales que abordan la salud mental, como Woebot o MindDoc, estas soluciones no están adaptadas cultural ni lingüísticamente al contexto salvadoreño. Además, requieren cierto nivel de alfabetización digital.

1.2 Fase del problema

En El Salvador, se estima que alrededor del 80% de las personas con trastornos mentales no reciben atención adecuada (MINSAL). Esta problemática se origina en una combinación de factores como el estigma social, la falta de recursos económicos, la escasa cobertura de servicios psicológicos y la ausencia de soluciones digitales accesibles y culturalmente pertinentes.

La falta de diagnóstico y atención oportuna genera consecuencias en cadena: disminución de la productividad laboral y académica, deterioro de las relaciones interpersonales y afectación significativa de la calidad de vida. Esto perpetúa un círculo vicioso donde los trastornos psicológicos se agravan y continúan invisibilizados.

Por lo tanto, se vuelve urgente desarrollar alternativas tecnológicas que permitan prevenir, identificar y orientar sobre estas condiciones desde un enfoque accesible, educativo y empático, como lo propone Braincare.

1.3 Objetivos de negocio

ID	Descripción del objetivo de negocio
ON-1	Incrementar en un 40% el acceso a información educativa sobre salud mental entre jóvenes salvadoreños, mediante la creación de una plataforma web en los próximos 6 meses.
ON-2	Desarrollar e integrar un chatbot basado en inteligencia artificial para brindar orientación básica sobre salud mental, con al menos un 80% de tasa de respuesta funcional desde su fase inicial.
ON-3	Obtener datos anónimos de uso y comportamiento de los usuarios para mejorar continuamente los contenidos y funcionalidades del sistema a partir del tercer mes de operación.
ON-4	Aumentar la interacción con el chatbot en un 15% mensual durante los primeros 6 meses posteriores a su implementación.
ON-5	Desplegar una versión estable y funcional del sistema antes del 30 de agosto del presente año.
ON-6	Reducir la brecha de acceso a información y orientación psicológica básica para sectores de la población salvadoreña que actualmente no cuentan con recursos adecuados.
ON-7	Registro y consulta Fomentar el seguimiento individual del bienestar emocional mediante el uso de una bitácora personal diaria, promoviendo la autorreflexión y detección temprana.bitácora personal diaria.
ON-8	Fortalecer la comunicación entre usuarios a través de foros categorizados que permitan compartir experiencias, opiniones y apoyo mutuo.
ON-9	Mejorar la experiencia del usuario mediante funcionalidades de personalización como favoritos, descargas y edición de perfil.
ON-10	Garantizar la seguridad y calidad del entorno digital mediante herramientas administrativas para la moderación de contenido y gestión de usuarios.
ON-11	Difundir el sistema como una solución factible a los usuarios interesados.

1.4 VISIÓN DE LA SOLUCIÓN

1.4.1 Fase de visión

Braincare surge como una propuesta digital local, diseñada específicamente para la población salvadoreña. Combina contenido educativo, herramientas de autoevaluación y asistencia automatizada a través de un chatbot basado en inteligencia artificial. La plataforma se apoyara en información confiable y en la asesoría de un usuario experto, priorizando un enfoque centrado en el usuario y en la accesibilidad.

1.4.2. Características del sistema

ID	Descripción	Prioridad	Objetivos de negocios asociados
CAR-01	La plataforma permitirá a los usuarios registrar y guardar su estado emocional diariamente, generando una bitácora personal que podrá ser consultada en cualquier momento como parte del seguimiento de su bienestar.	Alta	ON-7
CAR-02	La plataforma permitirá a los usuarios leer artículos educativos relacionados con la salud mental, así como marcarlos con "me gusta", guardarlos para futuras consultas y eliminarlos de su lista de guardados si así lo desean.	Alta	ON-9
CAR-03	La plataforma ofrecerá guías temáticas que los usuarios podrán leer, guardar en su perfil, añadir a favoritos o eliminar de sus listas de guardado o favoritos según su preferencia.	Alta	ON-6 ON-1
CAR-04	Cada artículo y guía contará con una sección de comentarios donde los usuarios podrán dejar su opinión.	Media	ON-9
CAR-05	El sistema permitirá a los usuarios descargar artículos y guías en formato digital para su consulta offline.	Media	ON-9

CAR-06	La plataforma contará con una biblioteca digital que reunirá una amplia variedad de recursos educativos, incluyendo artículos y guías, accesibles en cualquier momento por los usuarios.	Alta	ON-6 ON-1
CAR-07	El sistema proporcionará la opción de crear un foro dentro de la plataforma, con el objetivo de fomentar la interacción entre los usuarios en torno a temas relacionados con la salud mental.	Alta	ON-8
CAR-08	Los usuarios podrán escribir y publicar mensajes en el foro, responder a otros participantes, así como expresar su apoyo a las publicaciones mediante la opción de "me gusta".	Media	ON-8
CAR-09	Cada usuario tendrá la posibilidad de eliminar sus propios mensajes del foro cuando así lo desee	Media	ON-8
CAR-10	Los usuarios podrán reportar publicaciones de otros miembros del foro si consideran que el contenido es inapropiado o viola las normas de la comunidad.	Media	ON-8
CAR-11	La plataforma permitirá a los usuarios modificar datos de su perfil como: nombre, nombre de usuario, género, correo electrónico, dirección, contraseña.	Alta	ON-9
CAR-12	El sistema permitirá al administrador gestionar a los usuarios registrados, incluyendo la visualización de perfiles, edición de información, suspensión o eliminación de cuentas.	Alta	ON-10
CAR-13	El sistema permitirá al administrador moderar el foro mediante herramientas para eliminar publicaciones inapropiadas, revisar reportes de usuarios y aplicar sanciones cuando sea necesario.	Alta	ON-10
CAR-14	El sistema permitirá al administrador crear, editar o eliminar artículos y guías dentro de la biblioteca de recursos.	Alta	ON-10
CAR-15	El sistema permitirá al administrador revisar y gestionar los reportes realizados por los usuarios sobre publicaciones o comportamientos inadecuados.	Alta	ON-10
CAR-16	El sistema contará con redes sociales para su promoción al público.	Baja	ON-11
CAR-17	El sistema permitirá al usuario poder recibir orientación y análisis básico al igual que poder mantener	Alta	ON-6

	conversaciones con un asistente virtual de manera coherente en un 98%.		
CAR-18	El sistema permitirá al administrador poder alimentar el asistente virtual mediante un usuario experto	Media	ON-10
CAR-19	El sistema permitirá la compra de paquetes premium, los cuales otorgarán acceso a ventajas adicionales dentro de la plataforma, como recursos exclusivos, beneficios relacionados con el uso del foro o el chatbot.	Alta	

1.4.3 Alcance

Número de entrega	Tema principal	ID de características a incluir
1.0	Funcionalidad básica	CAR-01, CAR-02, CAR-03, CAR-06, CAR-07, CAR-11, CAR-12, CAR-13, CAR-14 y CAR-15
2.0	Funciones secundarias del sistema	CAR-04, CAR-05, CAR-08, CAR-09 y CAR-10
3.0	Visibilidad o funciones de última importancia	CAR-16

1.4.4 Formato Diagrama de Gantt (Cronograma de actividades)

		Feb		Marzo					Abril				Mayo					Junio					Julio				Agosto					Sep		
ACTIVIDADES	Propietario	S3	S4	S1	S2	S3	S4	S5	S1	S2	S3	S4	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5	S1	S2	S3	S4	S1	S2	S3	S4	S5	S1	S2	S3
Preparacion de proyecto																																		
Conformación de equipos y presentación de propuestas.	Todos																																	
Reunión de equipo para presentación de propuestas.	Todos																																	
Entrega de perfil de proyecto	Todos																																	
Respuesta de preaprobación del perfil del proyecto y/o sugerencias de mejora.	Todos																																	
Entregar final de perfil de proyecto	Todos																																	
Avances del proyecto																																		
Seguimiento de avance de proyectos																																		
Redacción del anteproyecto	Ivan, Anderson, Wilson																																	
Creación de los organizadores graficos (Ishikawa, Gantt, RACI)	Ivan y Anderson																																	
Investigación de temas de salud mental	Ivan, Anderson, Wilson																																	
Redacción del documento de requerimientos	Ivan, Anderson, Wilson																																	
Maquetación de diagramas UML	Anderson y Ivan																																	
Ideas para el diseño de la página	Ivan, Anderson, Wilson																																	
Diseño de Mockups	Anderson y Wilson																																	
Reunion para maquetación y diseñado de la página	Ivan, Anderson, Wilson																																	
Creación de guías de y afiches informativos	Ivan, Anderson, Wilson																																	
Entrega de Anteproyecto (al docente para retroalimentación)	Ivan, Anderson, Wilson																																	
Trabajar Front end y Backend de la página	Ivan, Anderson, Wilson																																	
Entrega de Anteproyecto revisado para mejoras	Ivan, Anderson, Wilson																																	
Redacción del manual de usuario	Ivan y Anderson																																	
Final del proyecto																																		
Entrega final de Anteproyecto CREA J 2025	Ivan																																	
Supervisión del proyecto constructivo	Ivan, Anderson, Wilson																																	
Evaluación final del proyecto	Ivan, Anderson, Wilson																																	
Proyecto finalizado	Ivan, Anderson, Wilson																																	
Crea-J																																		

Para una mejor visualización del diagrama de adjunta el enlace directamente al documento:

https://docs.google.com/spreadsheets/d/1rp4m5MRmtHN_XVeJ17gVeWlcrTcZ2VFkBL2KX8M9dhc/edit?usp=sharing

1.4.5 Formato de Matriz R.A.C.I (Diagrama de responsabilidades)

ACTIVIDADES	Propietario	Anderson Grimaldi	Iván López	Wilson López
Preparación de proyecto				
Conformación de equipos y presentación de propuestas.	Todos	RACI	RACI	RACI
Reunión de equipo para presentación de propuestas.	Todos	RACI	RACI	RACI
Entrega de perfil de proyecto	Todos	RACI	RACI	RACI
Respuesta de preaprobación del perfil del proyecto y/o sugerencias de mejora.	Todos	RACI	RACI	RACI
Entregar final de perfil de proyecto	Todos	RACI	RACI	RACI
Avances del proyecto				
Seguimiento de avance de proyectos	Docente encargado	Docente encargado	Docente encargado	Docente encargado
Redacción del anteproyecto	Todos	RACI	RACI	RACI
Creación de los organizadores gráficos (Ishikawa, Gantt, RACI)	Ivan y Anderson	R, A	R,A	C, I
Investigación de temas de salud mental	Todos	RACI	RACI	RACI

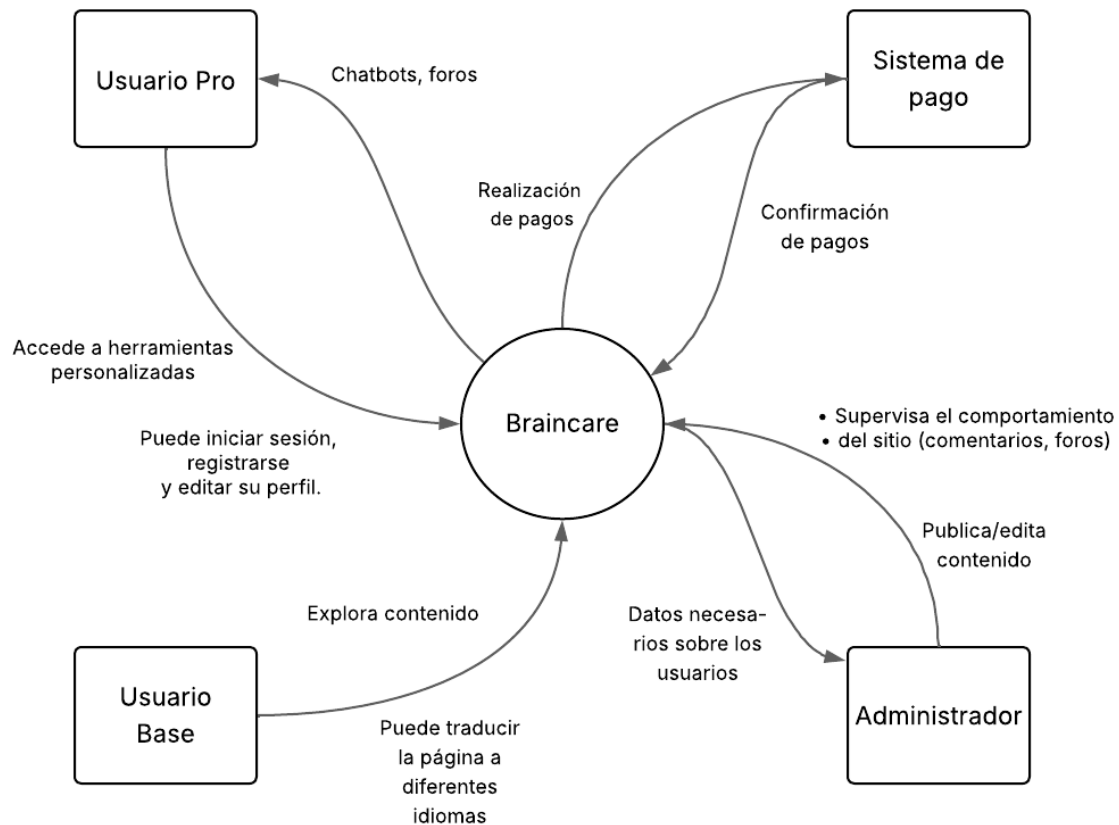
Redacción del documento de requerimientos	Todos	RACI	RACI	RACI
Maquetación de diagramas UML	Anderson y Ivan	R, A	R, A	C, I
Ideas para el diseño de la página	Todos	RACI	RACI	RACI
Diseño de Mockups	Anderson y Wilson	R, A	C, I	R, A
Reunión para maquetación y diseño de la página	Todos	RACI	RACI	RACI
Creación de guías de y afiches informativos	Todos	RACI	RACI	RACI
Entrega de Anteproyecto (al docente para retroalimentación)	Todos	RACI	RACI	RACI
Trabajar Frontend y Backend de la página	Todos	RACI	RACI	RACI
Entrega de Anteproyecto revisado para mejoras	Todos	RACI	RACI	RACI
Redacción del manual de usuario	Ivan y Anderson	R, A	R, A	C, I
Final del proyecto				
Entrega final de Anteproyecto	Ivan y Anderson	R, A	R, A	C, I
Supervisión del proyecto constructivo	Todos	RACI	RACI	RACI
Evaluación final del proyecto	Todos	RACI	RACI	RACI
Proyecto finalizado	Todos	RACI	RACI	RACI
Crea-J				

1.5. CONTEXTO DEL SISTEMA

1.5.1. Interesados

Nombre	Descripción	Responsabilidades
Anderson Grimaldi	Director y administrador del sistema	Aprueba decisiones importantes del sistema, encargado del diseño virtual del sistema, representante público del sistema, está presente en todas las fases del desarrollo del sistema
Iván López	Asesor de la documentación, traductor	Da seguimiento a la parte escrita del sistema, documenta los procesos y diagramas lógicos del sistema, encargado de la verificación del uso de normas APA dentro del documento del software, director general del sistema en el segundo idioma (así como su documentación)
Wilson López	Supervisor técnico	Encargado mayormente de la codificación y funciones esenciales del sistema, controla las diferentes bases de datos, supervisa el correcto flujo de la información dentro del sistema

1.5.2. Diagrama de contexto



1.5.3. Entorno de operación

El sistema será usado desde navegadores web y deberá soportar como mínimo los siguientes:

- **Google Chrome 17+**
- **Mozilla Firefox 10+**
- **Microsoft Edge** (considerando la de Bing y su reemplazo en entornos modernos)
- **Brave**
- **Opera GX**

Los dispositivos móviles siguientes deberán ser soportados:

- **iPhone/iPod Touch, iOS 6+**
- **Android 4+**

Se deberá descartar el uso de flash y applets. El sistema se ejecutará en un servidor que será adquirido para albergar el sistema, preferiblemente uno que permita la escalabilidad y seguridad necesarias para manejar datos sensibles de salud. A su vez, se debe usar el servidor de base de datos legado con Oracle v11.2, asegurando las configuraciones de seguridad adecuadas para la protección de datos en reposo y en tránsito.

La integración con sistemas externos, incluyendo el sistema de autorización de pagos y cualquier otro servicio relacionado con la salud, se llevará a cabo mediante **servicios web (APIs)** con estricto apego a protocolos de seguridad y privacidad como para la autenticación y autorización.

Consideraciones Adicionales Clave para Salud Mental:

- **Seguridad de Datos:** Implementación de encriptación de extremo a extremo (End-to-End Encryption) para todas las comunicaciones y encriptación de datos en reposo en la base de datos Oracle.
- **Cumplimiento Normativo:** Adherencia estricta a regulaciones de protección de datos de salud y normativas locales pertinentes en El Salvador.
- **Alta Disponibilidad y Recuperación ante Desastres:** Implementación de soluciones de redundancia y copias de seguridad robustas para asegurar la continuidad del servicio y la integridad de los datos.
- **Monitoreo y Auditoría:** Herramientas de monitoreo de rendimiento de la aplicación y la infraestructura, así como sistemas de auditoría para rastrear accesos y modificaciones de datos.

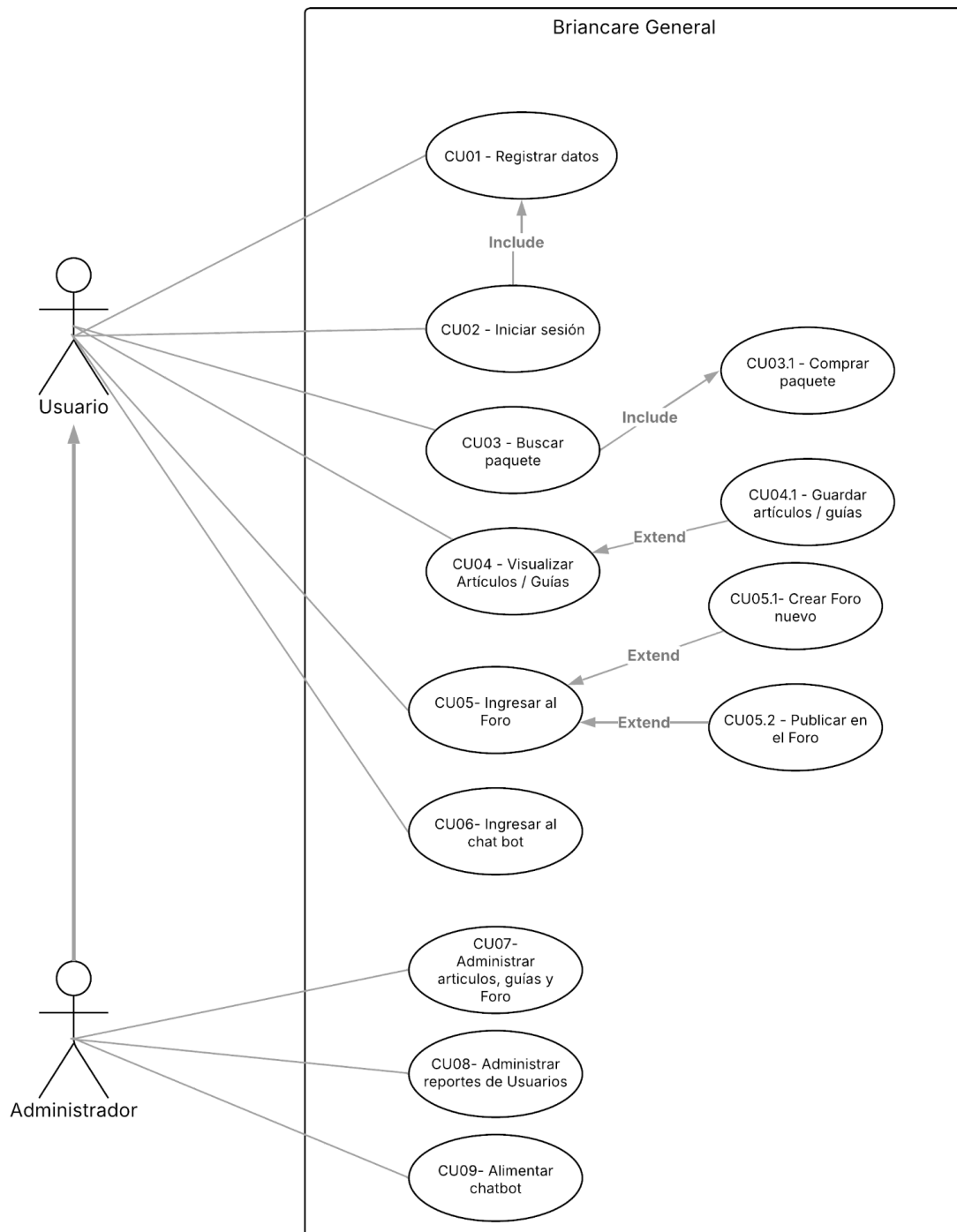
1.6. INFORMACIÓN ADICIONAL

Los criterios de aceptación del sistema “Braincare” incluyen:

- Plataforma web funcional y accesible desde navegadores modernos, desplegada en servidores en un plazo no mayor a 5 meses.
- Módulo de registro, inicio de sesión, selección de idioma y compra de paquetes completamente operativo antes del mes 7.
- Disponibilidad del sistema de pagos integrado con pasarelas seguras y operativas en un plazo máximo de 7 meses.
- Sistema traducido al menos en dos idiomas (español e inglés) y con posibilidad de ampliación.
- Pruebas de aceptación del sistema superadas en un 100%, incluyendo pruebas funcionales, de usabilidad y de seguridad.
- Manual de usuario, manual de administración y documentación técnica entregados al 100% junto con el código fuente del sistema.
- Interfaz accesible y adaptada a usuarios desde los 12 años, con contenido validado por especialistas en bienestar emocional.
- Entorno de pruebas funcional disponible para validación por parte del equipo de desarrollo y cliente antes del despliegue final.

2. Requerimientos de la arquitectura

2.1. Drivers funcionales



ID	Descripción	Característica asociada
CU-1	Permite al usuario registrar su información personal en la plataforma para crear una cuenta. Involucra datos como nombre, correo, contraseña, género, entre otros.	CAR-01
CU-2	Permite al usuario autenticarse en la plataforma ingresando su correo y contraseña. Es un paso previo obligatorio para acceder a funciones personalizadas.	CAR-01
CU-3	Permite al usuario explorar paquetes que le ofrecen beneficios y servicios más personalizados.	CAR-19
CU-3.1	Permite al usuario adquirir paquetes de contenido o funcionalidades específicas, si existiera una versión premium o de pago dentro del sistema.	CAR-19
CU-04	Permite al usuario acceder al contenido educativo como artículos o guías sobre salud mental, pudiendo leerlos desde la plataforma.	CAR-02 CAR-06 CAR-04
CU-4.1	Permite al usuario guardar artículos o guías en su perfil para consulta posterior. Puede incluir funcionalidades como marcar como favorito, eliminar de guardados o descargar el contenido.	CAR-03 CAR-05 CAR-06
CU-05	Permite al usuario acceder a la sección del foro para explorar temas publicados por otros usuarios o participar en discusiones.	CAR-07
CU-5.1	Permite al usuario generar un nuevo hilo de discusión en el foro, agregando un título, descripción y categoría.	CAR-07
CU-5.2	Permite al usuario escribir y publicar mensajes dentro de un hilo del foro, responder a otros mensajes, interactuar mediante "me gusta" o reportar en caso que el mensaje sea inapropiado.	CAR-08 CAR-09 CAR-10
CU-6	Permite al usuario interactuar con un asistente virtual basado en inteligencia artificial para resolver dudas y recibir orientación básica.	CAR-17
CU-7	Permite al administrador crear, editar o eliminar artículos, guías y publicaciones del foro para mantener la calidad y relevancia del contenido.	CAR-14
CU-8	Permite al administrador revisar y gestionar los reportes generados por los usuarios sobre mensajes ofensivos o inapropiados en el foro, aplicando sanciones si es necesario.	CAR-13 CAR-15 CAR-12
CU-9	Permite al administrador agregar o actualizar la base de conocimiento del chatbot para mejorar sus respuestas y adaptarlo a nuevas consultas o temas emergentes.	CAR-18

2.1.2. Elección de casos de uso primarios

ID	Criterios de elección (justificación)
CU-01	Es el primer momento entre el usuario y la plataforma. Sin esta acción no es posible personalizar la experiencia ni hacer seguimiento del uso del sistema.
CU-02	Es esencial para proteger la información personal y garantizar el acceso seguro a funcionalidades avanzadas.
CU-04	Corresponde a una de las funcionalidades principales del sistema, alineada con el objetivo de educar al usuario sobre salud mental.
CU-4.1	Mejora la experiencia del usuario permitiendo la consulta posterior. Aporta valor en términos de personalización y retención.
CU-05	Promueve la participación y apoyo entre personas, alineado con el objetivo de crear una comunidad y combatir el estigma de la salud mental. ON-8
CU-06	Es una de las razones casi primarias del sistema siendo un asistente especializado en dar tratamiento y compañía básica.

2.2. Drivers de atributos de calidad

ID	Categoría	Fuente	Estímulo	Ambiente	Respuesta del sistema	Medida de respuesta	Prioridad
EAC-01	Usabilidad	Usuario	El usuario accede por primera vez a la plataforma.	Normal	El sistema muestra una interfaz clara, con navegación intuitiva y accesible.	El usuario puede acceder a las funciones clave sin ayuda externa.	(A,A)
EAC-02	Seguridad	Usuario malintencionado	Intenta acceder sin autorización a información personal o privada.	Cualquier momento	El sistema bloquea el acceso no autorizado y protege los datos.	Cero filtraciones, intentos registrados y notificados.	(A,A)
EAC-03	Disponibilidad	Usuario	Intenta ingresar al sitio web o al asistente virtual en un horario nocturno.	Posible baja carga del sistema	El sistema responde sin interrupciones.	Tiempo de actividad del sistema superior al 99%.	(M, A)
EAC-04	Mantenibilidad	Administrador / Usuario experto	Solicita actualizar contenido o extender funcionalidades.	Durante operación del sistema	El sistema permite modificar artículos, alimentar chatbot.	Actualizaciones sin afectar el servicio, en menos de 1 día.	(M, M)
EAC-05	Portabilidad	Usuario	Accede al sistema desde un dispositivo móvil con conexión	Red móvil, navegador móvil	El sistema adapta correctamente la interfaz y conserva todas las funciones	Funcionalidad completa en versión móvil, sin errores de visualización o uso.	(A,A)

			limitada.		clave.		
EAC-06	Rendimiento	Usuario	Accede al chatbot o abre artículos simultáneamente con otros usuarios.	Carga media del sistema	El sistema responde sin demoras perceptibles en cada solicitud.	Tiempo de respuesta menor a 3 segundos en un 95% de los casos.	(M,M)

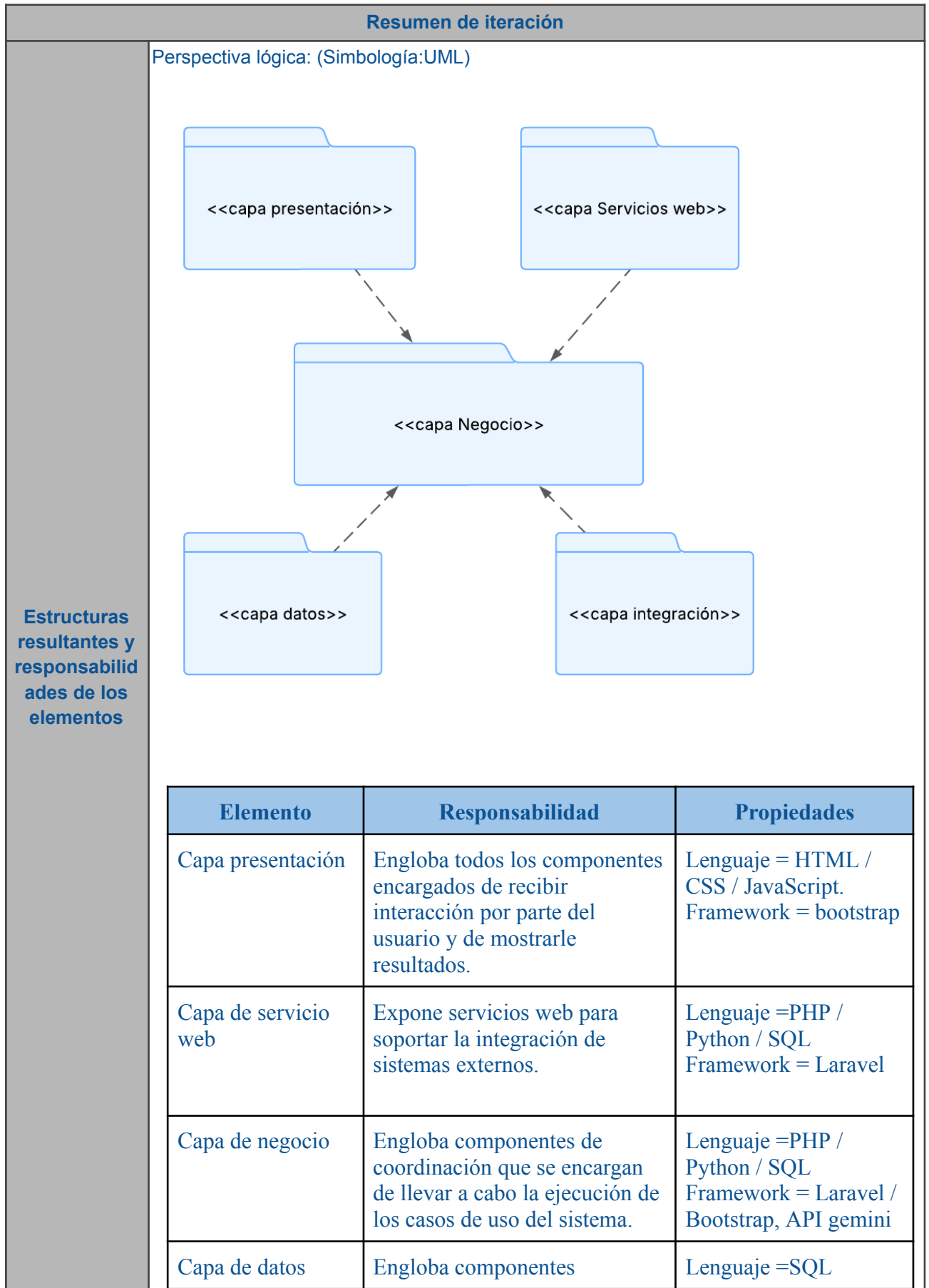
2.3. Drivers de restricciones

ID	Tipo de restricción	Descripción
R-01	Técnica	<ul style="list-style-type: none">- El sistema debe desarrollarse como aplicación web y ser accesible desde dispositivos móviles.- El sistema debe utilizar tecnologías de código abierto y gratuitas por presupuesto, evitando utilizar soluciones comerciales como Microsoft Azure, licencias de chatbot privado.
R-02	Económica	<ul style="list-style-type: none">- El presupuesto que se tiene para hosting, almacenamiento y manejo del chatbot es limitado.- Optimizar recursos y priorizar funcionalidades esenciales en versiones iniciales.
R-03	Organizacional	<ul style="list-style-type: none">- El equipo de desarrollo es pequeño y tiene experiencia en desarrollo web con tecnologías como JavaScript, PHP, etc.
R-05	De tiempo	<ul style="list-style-type: none">- El prototipo funcional debe estar operativo antes del 10 de septiembre del presente año.
R-06	Legal	<ul style="list-style-type: none">- El sistema debe garantizar la privacidad de los datos personales conforme a normativas locales como Ley de Protección de Datos de El Salvador.

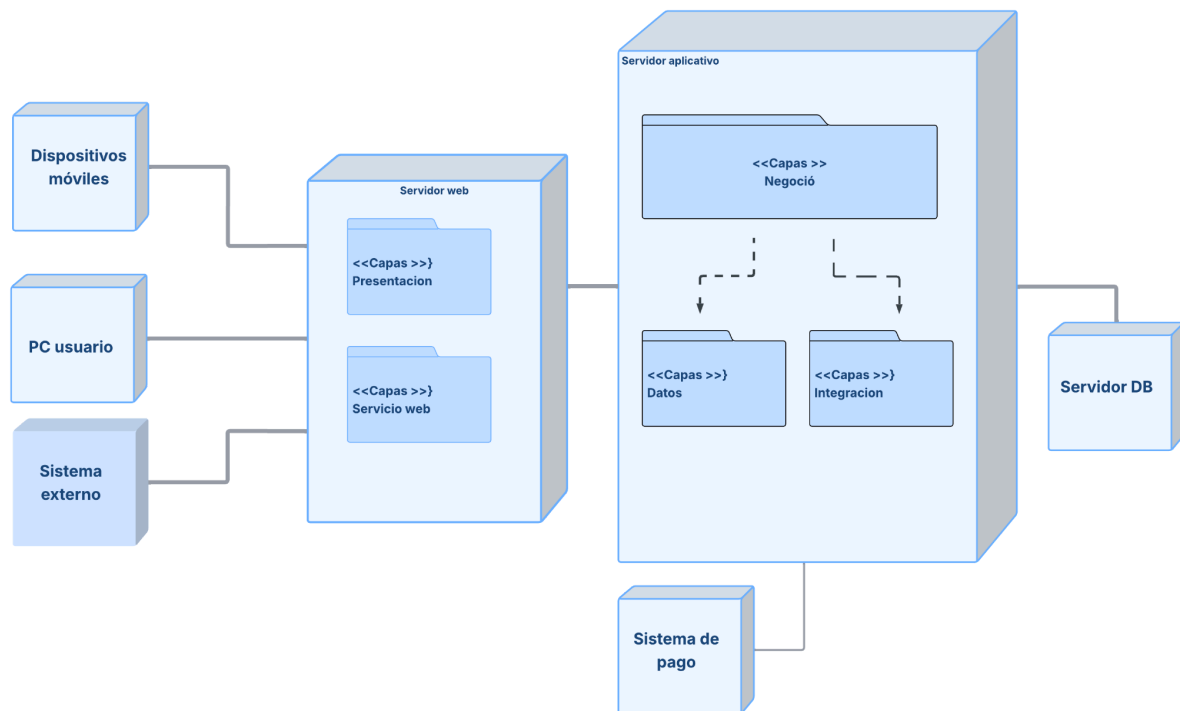
3. Diseño de la arquitectura

3.1. Primera iteración: estructuración general del sistema

Primera iteración – Estructuración general del sistema		
Elementos a descomponer	Sistema completo (BRAIN CARE)	
Drivers elegidos para la iteración	Funcionalidad, usabilidad, seguridad, disponibilidad, portabilidad, mantenibilidad, restricciones económicas y tecnológicas.	
Conceptos de diseño	Concepto	Justificación
	Patrón de arquitectura en capas (3 capas) y enfoque cliente-servidor web con modularidad por componentes.	Las capas permiten una clara separación de responsabilidades entre la interfaz de usuario, la lógica de negocio y la gestión de datos. Esta organización facilita la mantenibilidad, escalabilidad y reutilización del sistema a futuro, además de alinearse con las restricciones técnicas del proyecto y la experiencia del equipo de desarrollo. Asimismo, se adopta un enfoque cliente-servidor web para garantizar accesibilidad desde múltiples dispositivos, especialmente considerando la importancia de la portabilidad y usabilidad en el contexto de Braincare.
	Elementos instanciados	Capa de presentación, lógica de negocio, persistencia de datos; módulos de foro, contenido, chatbot y administrador.
	Interfaces definidas	Interfaces entre: presentación ↔ lógica de negocio; lógica ↔ persistencia. Se definen de forma abstracta aún sin detalle técnico.
	Resultado de la iteración	Modelo lógico inicial del sistema estructurado en capas + propuesta de despliegue físico inicial con tres nodos: cliente web, servidor aplicativo y servidor de base de datos.



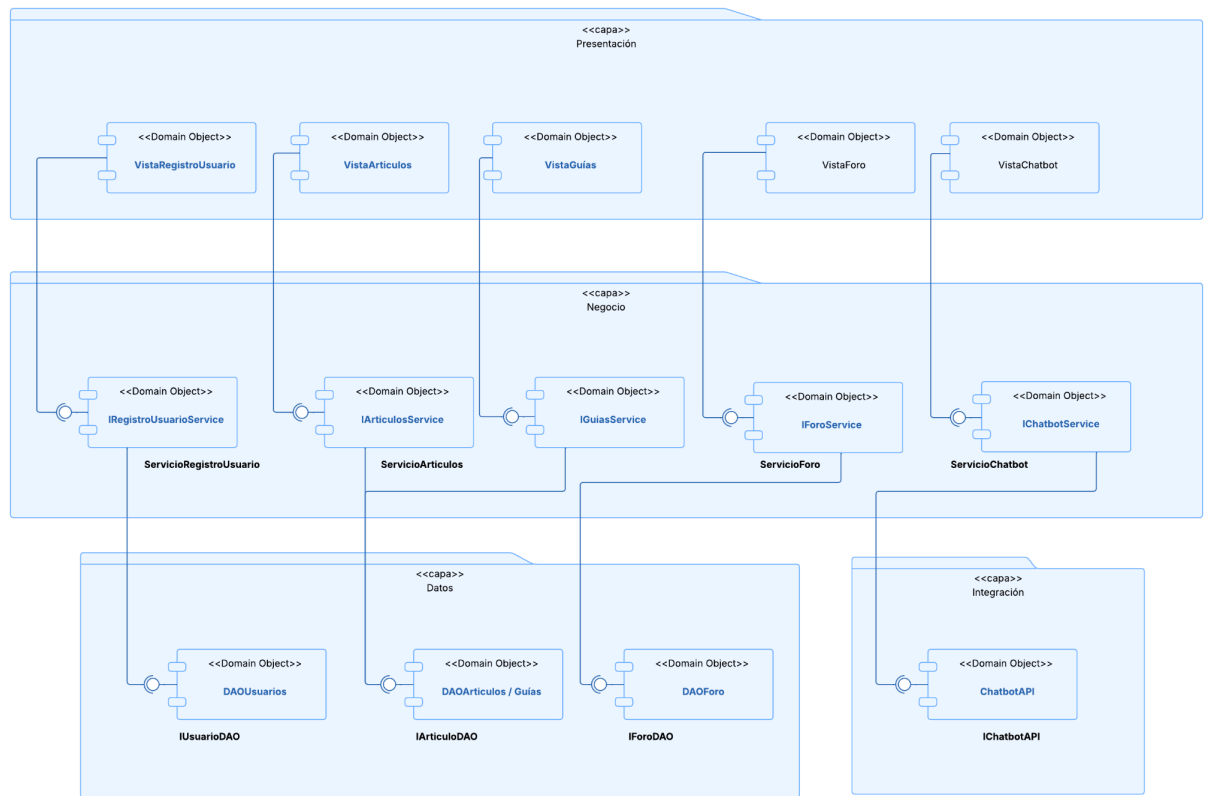
		encargados de almacenar objetos en una base de datos, ocultando del resto de la aplicación el hecho de que se trata de una base de datos relacional.	Framework = Laravel
	Capa de integración	Permite la integración con sistemas externos (sistema de pagos).	Lenguaje = PHP /Python / TypeScript Framework =Node.js / Laravel / API Paypal



Elemento	Responsabilidad	Propiedades
PC del usuario	Computadora personal mediante la cual acceden los usuarios y administradores del sistema.	Navegador = Internet Explorer 10+, Firefox 10+, Google Chrome 17+
Dispositivo móvil	Dispositivos por medio de los cuales se accederá en el futuro a la aplicación.	OS = iPhone / , Android 4+
Servidor web	El servidor web alberga las capas de presentación y servicios <i>web</i> .	Tipo = Firebase
Servidor Aplicativo	El servidor aplicativo alberga las capas de negocio, datos e integración de la aplicación.	Memoria = Por definir Procesador = Por definir
Servidor BD	Este servidor alberga la base de datos (del otro servidor legado).	SQL
Sistema externo	Sistema externo que interactúa con el sistema que se desarrolla mediante servicios <i>web</i> .	Lenguaje = JavaScript, PHP

Resumen iteración		
Elemento a descomponer	Las distintas capas del sistema definidas en la primera iteración	
Drivers elegidos para la iteración	Casos de uso primarios:	
	CU-01: Registrar usuario	
	CU-04: Visualizar artículos / guías	
	CU-05.2: Publicar en el foro	
	CU-06: Usar el chatbot	
Resumen iteración		
Conceptos de diseño	Concepto	Justificación
	Patrón Objeto de dominio (Domain Object	Permite encapsular funcionalidades clave (registro, publicación, consulta) en objetos autocontenidos, facilitando la reutilización y claridad.
	Patrón Interfaz explícita + implementación encapsulada	Facilita pruebas y separación de responsabilidades, permitiendo que la lógica esté desacoplada de la interfaz.
	Patrón Mapeador de datos (Data Mapper)	Permite desacoplar la lógica de acceso a la base de datos. Cada módulo de acceso (registro, artículos, foro) se mapea a través de DAOs independientes.

Perspectiva lógica: (simbología: UML)



**Estructura
resultantes y
responsabilida
des de los
elementos**

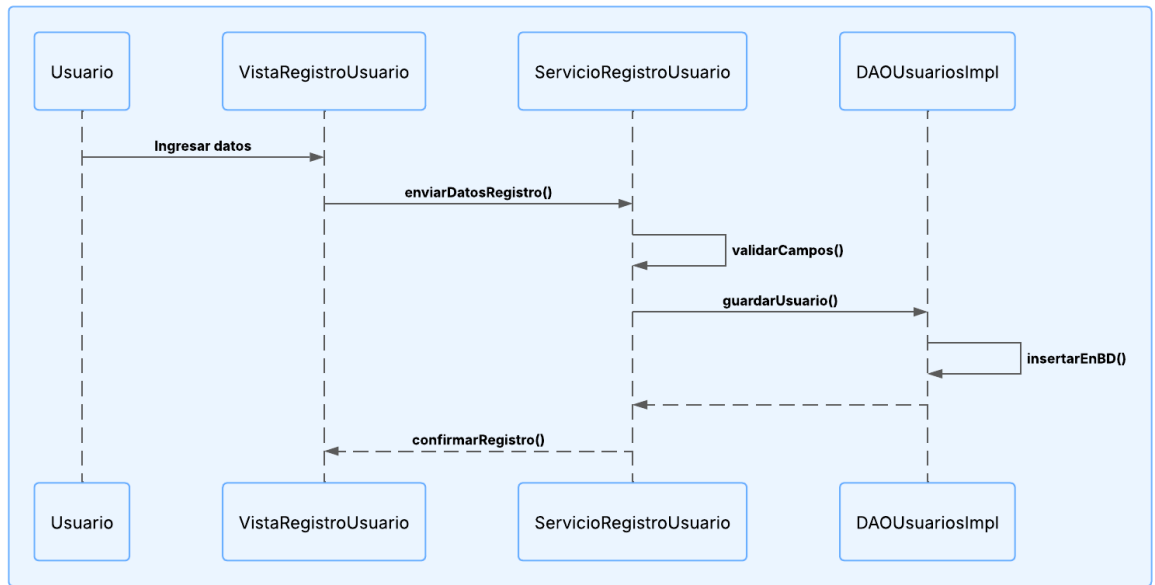
Para visualizar de mejor manera el diagrama que explica la perspectiva lógica se adjunta el link correspondiente:

https://lucid.app/lucidchart/7744f4eb-cc09-4123-b9a2-6e9d52bc4628/edit?viewport_loc=-956%2C-206%2C3807%2C1699%2CQban4hz2kE~6&invitationId=inv_2a125d23-ffed-4a86-a10b-601e12e3c589

Estructuras resultantes y responsabilidades de los elementos	Resumen de la iteración		
	Elemento	Responsabilidad	Propiedades
	VistaRegistroUsuario	Desplegar el formulario de registro y recibir datos del usuario	Lenguaje = HTML, JS / Framework = Laravel, vue.
	VistaArticulos	Mostrar guías/artículos al usuario y permitir guardarlos o descargarlos	Lenguaje = HTML, JS / Framework = Vue
	VistaForo	Mostrar hilos del foro y permitir publicación/respuesta	Lenguaje = HTML, JS / Framework = Laravel, vue.
	VistaChatbot	Proveer una interfaz conversacional de texto	Lenguaje = HTML, JS / Framework = Laravel
	ServicioRegistroUsuario	Validar datos, cifrar contraseña y crear usuario	Lenguaje = JavaScript / PHP / Python
	ServicioArticulos	Consultar y entregar guías/artículos disponibles	Lenguaje = JavaScript / Php /Python
	ServicioForo	Controlar la publicación y validación de mensajes en el foro	Lenguaje = JavaScript / PHP / Python
	ServicioChatbot	Procesar preguntas del usuario y solicitar respuesta al servicio externo	Lenguaje = JavaScript / Node.js – Stateless
	DAOUsuariosImpl	Insertar y consultar datos del usuario en la BD	Lenguaje = SQL / Framework = Laravel
	DAOArticulosImpl	Obtener artículos o guías desde la base de datos	Lenguaje = SQL / Framework = Laravel
	DAOForoImpl	Guardar y recuperar publicaciones del foro	Lenguaje = SQL / Framework = Laravel
	ChatbotAPIImpl	Comunicarse con un servicio externo (OpenAI / IA) para obtener respuestas del bot	Lenguaje = JS / Framework = REST Client / API Gemini

Interfaces de los elementos

Diagrama de Secuencia UML: Registrar Usuario en Sistema Web con Arquitectura en Capas



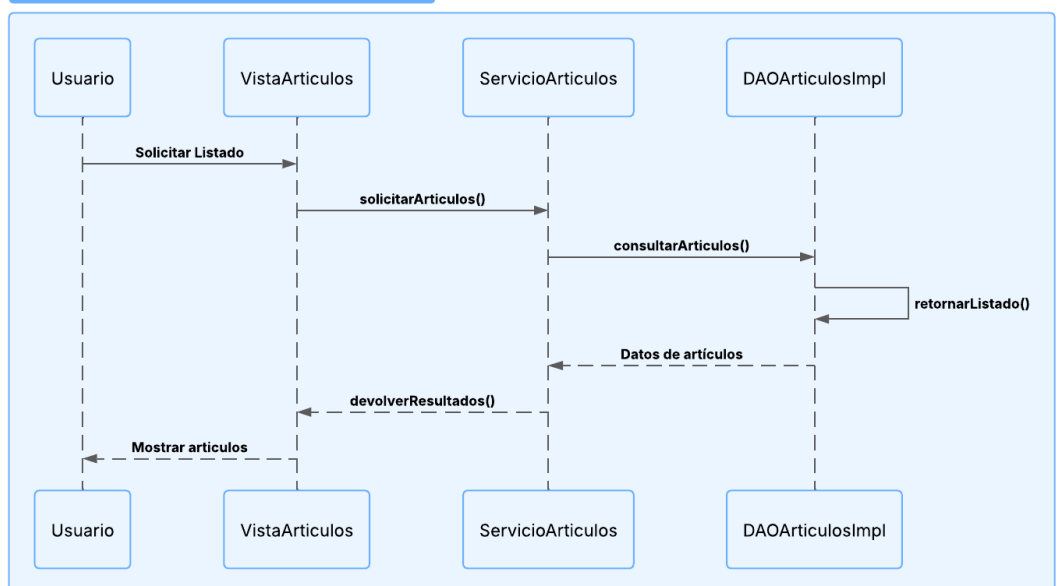
Para visualizar de mejor manera el diagrama que explica la perspectiva de CU-01: Registrar usuario se adjunta el link correspondiente:

https://lucid.app/lucidchart/7744f4eb-cc09-4123-b9a2-6e9d52bc4628/edit?viewport_loc=285%2C-1068%2C2137%2C1092%2C6Lbno4R.WLz9&invitationId=inv_2a125d23-ffed-4a86-a10b-601e12e3c589

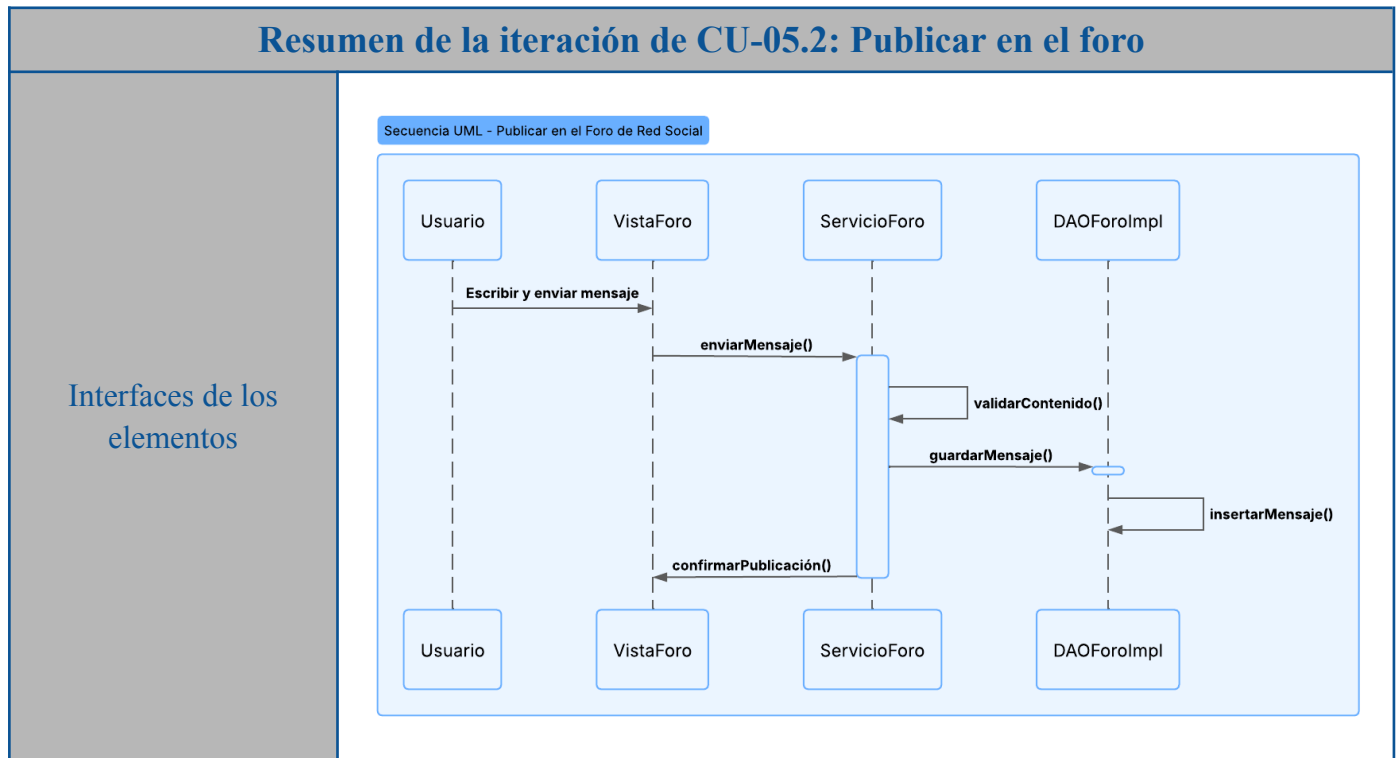
Resumen de la iteración de CU-04: Visualizar artículos / guías

Interfaces de los elementos

UML Sequence Diagram for 'Visualize Articles or Guides' Use Case



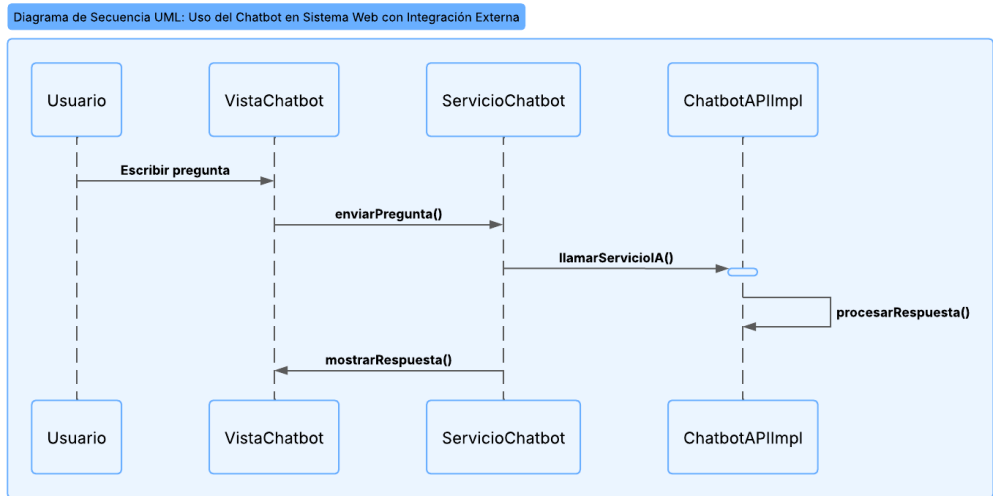
Para visualizar de mejor manera el diagrama que explica la perspectiva de CU-04: Visualizar artículos / guías se adjunta el link correspondiente:
https://lucid.app/lucidchart/7744f4eb-cc09-4123-b9a2-6e9d52bc4628/edit?viewport_loc=347%2C-1090%2C2192%2C1214%2CfSbnHMMicMMK&invitationId=inv_2a125d23-ffed-4a86-a10b-601e12e3c589



Para visualizar de mejor manera el diagrama que explica la perspectiva de CU-05.2: Publicar en el foro se adjunta el link correspondiente:
https://lucid.app/lucidchart/7744f4eb-cc09-4123-b9a2-6e9d52bc4628/edit?viewport_loc=177%2C-1096%2C1972%2C1092%2CsVbnM9B.nf2V&invitationId=inv_2a125d23-ffed-4a86-a10b-601e12e3c589

Resumen de la iteración de CU-06: Usar el chatbot

Interfaces de los elementos



Para visualizar de mejor manera el diagrama que explica la perspectiva de CU-06: Usar el chatbot se adjunta el link correspondiente:

https://lucid.app/lucidchart/7744f4eb-cc09-4123-b9a2-6e9d52bc4628/edit?viewport_loc=236%2C-1095%2C1972%2C1092%2CZWbnf1EokA~G&invitationId=inv_2a125d23-ffed-4a86-a10b-601e12e3c589

3.3. Tercera iteración: desempeño en capa de datos

Resumen de la Iteración		
Elementos a descomponer	Capas	
Drivers elegidos para la iteración	EAC-01 (Usabilidad) EAC-03 (Disponibilidad) EAC-06 (Rendimiento) CU-04 (Visualizar artículos/guías) CU-06 (Usar el chatbot	
Conceptos de diseño	Concepto	Justificación
	Patrones de Adquisición (Early Acquisition y Lazy Acquisition)	Optimiza el rendimiento cargando sólo datos necesarios inicialmente, crucial para objetos dependientes y datos que podrían no usarse inmediatamente.
	Táctica: Mantener copias múltiples	Cache a nivel de procesos para contenido estático como artículos educativos, mejorando tiempos de respuesta.
	Cachés	Query cache para almacenar resultados de consultas frecuentes (ej. listado de artículos populares), reduciendo carga en BD.
	Táctica: Manejo de recursos - Paginación	Consultas por subconjuntos de datos, esencial para el rendimiento en dispositivos móviles con conexión limitada.
	Replicación de datos	Garantiza alta disponibilidad del servicio, especialmente para el chatbot en horarios pico.

	Particionamiento horizontal	Distribuye tablas grandes (bitácoras emocionales) para optimizar tiempos de consulta.
Estructuras resultantes y responsabilidades de los elementos	Perspectiva lógica (UML)	

Estructuras resultantes y responsabilidades de los elementos	Resumen de la iteración		
	Elemento	Responsabilidad	Propiedades
	UsuarioForos	Sección en la cual los usuarios podrán compartir sus experiencias ya sea anónimo o con sus respectivo name users.	Lenguaje = HTML,CSS,BOOTSTRAP, PHP, PYTHON, JS / Framework = Laravel
	VistaArticulos	Mostrar guías/artículos al usuario y permitir guardarlos o descargarlos	Lenguaje = PHP, PYTHON, JS / Framework = Laravel

4. Documentación

En el proyecto, el arquitecto de software debe aplicar el método de “**vistas y más allá**” para estructurar la arquitectura del sistema desde múltiples perspectivas, con el fin de abordar las diversas preocupaciones de los distintos interesados. Este enfoque le permitirá definir y documentar las vistas necesarias, como la vista lógica (organización del software en módulos y componentes), la vista de desarrollo (estructura del código fuente y asignación de responsabilidades), la vista de procesos (comportamiento en tiempo de ejecución y concurrencia) y la vista física (despliegue del sistema en la infraestructura tecnológica). Además, debe asegurar que todas estas vistas están alineadas con los requerimientos funcionales y no funcionales del sistema.

4.1 Generar vistas candidatas

	Vistas de módulos					Vistas de Componentes y Conectores		Vistas de Localización		Otra documentación				
	Descomposición	Uso	Generalización	Capas	Modelo de datos	Cliente-servidor	Comunicación entre procesos	Puesta en marcha	Documentación de interfaces	Diagrama de contexto	Mapeo entre vistas	Variabilidad	Análisis	Decisiones arquitectónicas
Líder del proyecto (Anderson Grimaldi)	m	m	m	m	m	p	p	m	m	d	m	m	d	d
Equipo de desarrollo (Equipo en general)	d	d	d	d	d	d	d	d	d	m	d	m	d	d
Equipo de pruebas (Iván López)	d	d	p	d	d	m	m	m	d	m	d	m	m	m
Arquitecto de software (Equipo en general)	d	d	d	d	d	d	d	d	d	d	d	d	d	d
Representante de usuarios/tester funcional (Iván López)	p	m	m	p	p	p	p	p	m	d	p	m		p
Encargado de operaciones/soporte (Wilson López)	p	m	m	m	p				m	d	p	m		p
D = muy detallada (indispensable para ese rol)														
M = considerable detalle (útil pero no central)														
P = poco detalle (solo como referencia)														

Tabla de vistas arquitectónicas - Para una mejor visualización se adjunta el enlace a la tabla en excel:

<https://docs.google.com/spreadsheets/d/1jmieo5kCvHmG8iQnD8kPAsYaFNCnaruHYDPtgrghMk0/edit?usp=sharing>

Explicación de la tabla de vistas arquitectónicas

La tabla presenta el nivel de detalle requerido para cada tipo de vista arquitectónica según el rol desempeñado por los miembros del equipo en el proyecto “**Braincare**”. Esta clasificación se basa en el método de vistas utilizado en arquitectura de software, que permite representar el sistema desde distintas perspectivas para facilitar la comprensión, diseño, implementación y mantenimiento del sistema.

Cada vista (**de módulos, componentes y conectores, localización y otra documentación**) ha sido evaluada con tres niveles de necesidad de detalle:

- **D (Muy detallada): Información esencial para el desempeño del rol.**
- **M (Considerable detalle): Información relevante, pero no central.**
- **P (Poco detalle): Información referencial o poco utilizada por el rol.**

Vistas de módulos: Incluyen aspectos como la descomposición, uso, generalización, capas y modelo de datos. Estas vistas son fundamentales para el equipo de desarrollo y el arquitecto, ya que permiten visualizar la estructura lógica del sistema, sus relaciones internas y su organización en capas o módulos funcionales.

Vistas de componentes y conectores: Representan la estructura dinámica del sistema, es decir, cómo interactúan los módulos a través de conectores. Son clave para el equipo de pruebas y de infraestructura, ya que implican aspectos de despliegue, comunicación entre procesos y arquitectura cliente-servidor.

Vistas de localización: Especifican la distribución física del software en el hardware, considerando la puesta en marcha y asignación de procesos a nodos. Son especialmente relevantes para roles como el arquitecto, infraestructura y operaciones/sophorte.

Otra documentación: Incluye diagramas de contexto, mapeo entre vistas, análisis de variabilidad, decisiones arquitectónicas y documentación de interfaces. Esta información proporciona una visión integral del sistema y es útil para todos los actores involucrados, especialmente en actividades de coordinación, mantenimiento, pruebas y evolución del sistema.

Esta tabla permite establecer claramente las necesidades de información de cada rol y facilita la planificación del proceso de documentación. Al distribuir el detalle de las vistas de forma estratégica, se optimiza el trabajo en equipo y se asegura que cada miembro cuente con los recursos necesarios para el cumplimiento efectivo de sus funciones dentro del proyecto

Braincare.

4.2 Combinar vistas

Una vez finalizada la tabla de interesados y vistas, se llevó a cabo un análisis detallado con el objetivo de **reducir el número de vistas a documentar sin comprometer la calidad ni la utilidad de la información arquitectónica**. Este análisis fue realizado por el equipo de arquitectura del proyecto *Braincare*, considerando principalmente la utilidad individual de cada vista y el impacto de su combinación en términos de **complejidad de elaboración, mantenimiento e interpretación**.

A partir de este análisis, se ha decidido documentar las siguientes vistas combinadas:

- **Vista 1: Capas, Descomposición y Uso** (*muy detallada*)

Esta vista es esencial para comprender la estructura lógica del sistema y su

organización funcional. La capa de presentación (interfaz del usuario), la lógica de negocio, los servicios y la capa de datos se encuentran claramente delimitadas en *Braincare*, lo que permite una combinación eficiente de estas vistas sin introducir complejidad adicional. Esta vista será clave para el equipo de desarrollo, pruebas y mantenimiento.

- **Vista 2: Modelo de datos** (*muy detallada*)

Dado que el sistema *Braincare* incluye funcionalidades como registro de usuarios, gestión de suscripciones, pagos, traducción de idiomas y almacenamiento de historial de uso, el **modelo de datos** requiere una representación independiente. Esta vista contiene información de naturaleza diferente a las anteriores y es crucial para el diseño de la base de datos, por lo cual **se documentará por separado**.

- **Vista 3: Cliente-servidor y Comunicación entre procesos** (*muy detallada*)

El sistema **Braincare** tiene una arquitectura basada en el modelo cliente-servidor, donde los visitantes y usuarios interactúan desde sus navegadores con los servicios desplegados en el servidor web. Esta vista permitirá ilustrar adecuadamente los principales flujos de información, incluyendo la interacción con el sistema de pagos y la carga dinámica de contenidos. La vista de comunicación entre procesos se ha combinado con esta sin generar complejidad adicional, ya que ambas comparten el mismo enfoque estructural.

- **Vista 4: Puesta en marcha e Instalación** (*muy detallada*)

Estas vistas han sido consolidadas en una sola ya que están directamente relacionadas con la configuración del entorno, la asignación de procesos a dispositivos físicos y el

despliegue del sistema en producción. En **Braincare**, el sistema será implementado de manera web, por lo que el entorno de instalación y su documentación puede integrarse sin dificultad en una vista unificada.

Otras vistas complementarias: Aunque no se documentan como vistas principales, se incorporarán referencias al **diagrama de contexto**, la **variabilidad** y el **mapeo entre vistas** dentro de la documentación general. Asimismo, se mantendrá una sección de **decisiones arquitectónicas** para justificar los enfoques tomados durante el diseño.

4.3 Priorizar las vistas

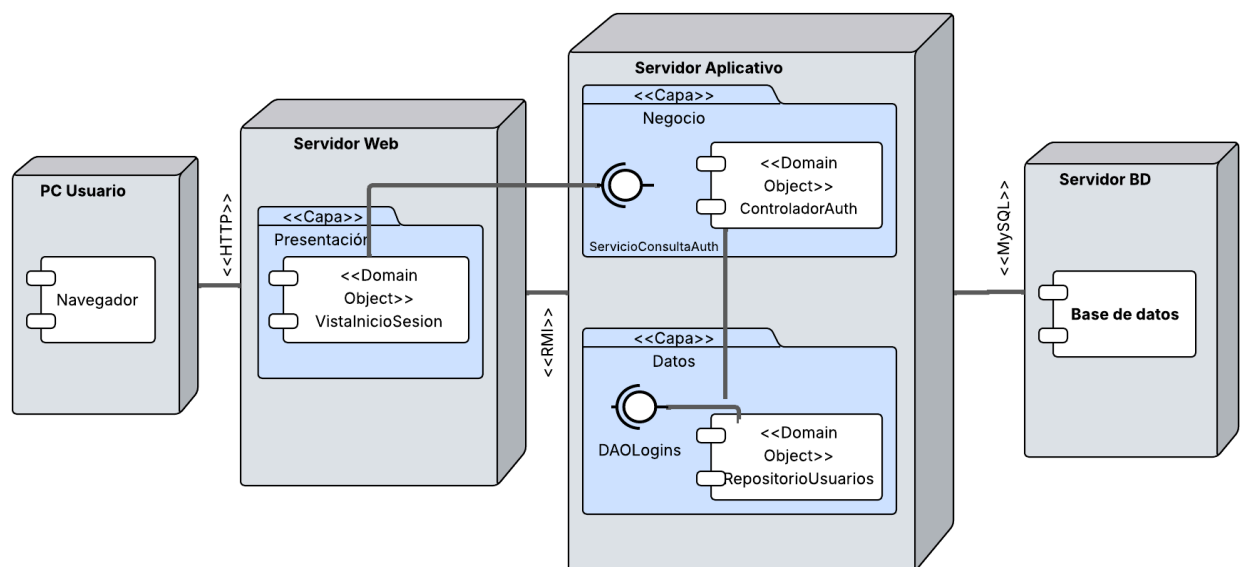
Categoría	Vista combinada o individual	Prioridad
Vistas de módulos	Vista 1: Capas, Descomposición y Uso	1
	Vista 2: Modelo de datos	1
Vistas de comportamiento	Vista 3: Cliente-servidor y Comunicación entre procesos	1
Vistas de localización	Vista 4: Puesta en marcha e Instalación	1

Esta priorización está basada en el hecho de que las vistas de módulos y comportamiento representan la estructura lógica y operativa del sistema, fundamentales para su construcción inicial. Las vistas complementarias se desarrollarán posteriormente o conforme a la necesidad del equipo técnico.

4.4 Ejemplo de vistas

El siguiente ejemplo representa la vista del inicio de sesión:

https://lucid.app/lucidchart/22a50d96-3e37-4444-ac40-40a8f94789f4/edit?viewport_loc=41%2C-74%2C1995%2C891%2CsXonDeGEO1a5&invitationId=inv_0c18541a-558b-4e37-ba55-debc0a351237



4.4.3 Catálogo de elementos

Elemento	Responsabilidad Principal	Propiedades
Navegador	Interfaz entre el usuario y la aplicación web. Permite enviar solicitudes y mostrar respuestas.	- Ejecuta HTML, CSS y JS- Usa el protocolo HTTP/HTTPS- Compatible con dispositivos múltiples
VistaInicioSesion	Presentar el formulario de inicio de sesión y capturar credenciales del usuario.	- Componente frontend (HTML/CSS/JS)- Validación básica

ControladorAuth	Procesar el inicio de sesión, autenticación y redirección según el rol del usuario.	- Backend (Laravel, Python y PHP)- Lógica de negocio de autenticación- Interactúa con repositorioUsuarios
RepositorioUsuarios	Acceder a los datos de los usuarios en la base de datos (login, registro, recuperación, etc.).	- Encapsula consultas SQL
Base de datos	Almacenar persistentemente los datos de los usuarios, contraseñas (hash), roles, etc.	- MySQL- Tablas: usuarios, roles, etc.- Acceso mediante SQL

Elemento	Responsabilidad Principal	Propiedades
Request/Response (L)	Comunicación entre el navegador y el servidor web/backend.	Protocolo HTTP/HTTPS Formato: JSON, HTML
Request/Response (R)	Comunicación entre el backend (controlador) y la base de datos o repositorio.	Consultas SQL Formato: datos estructurados

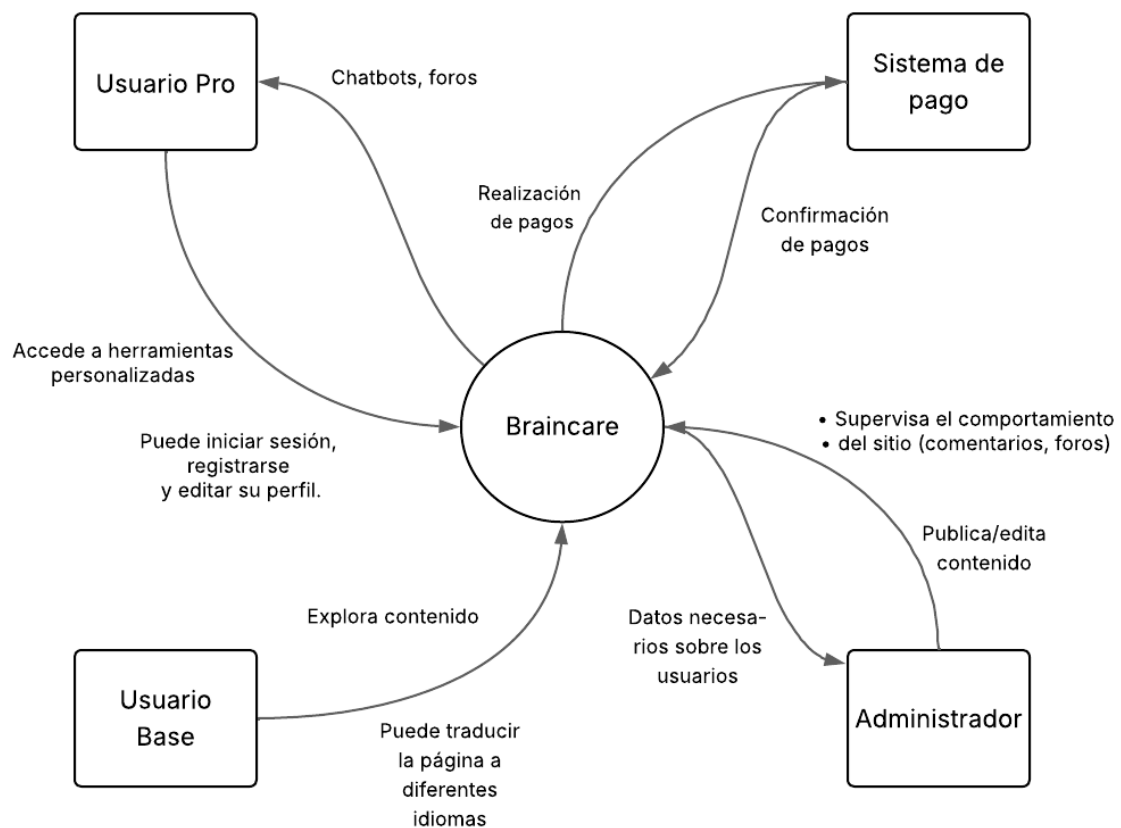
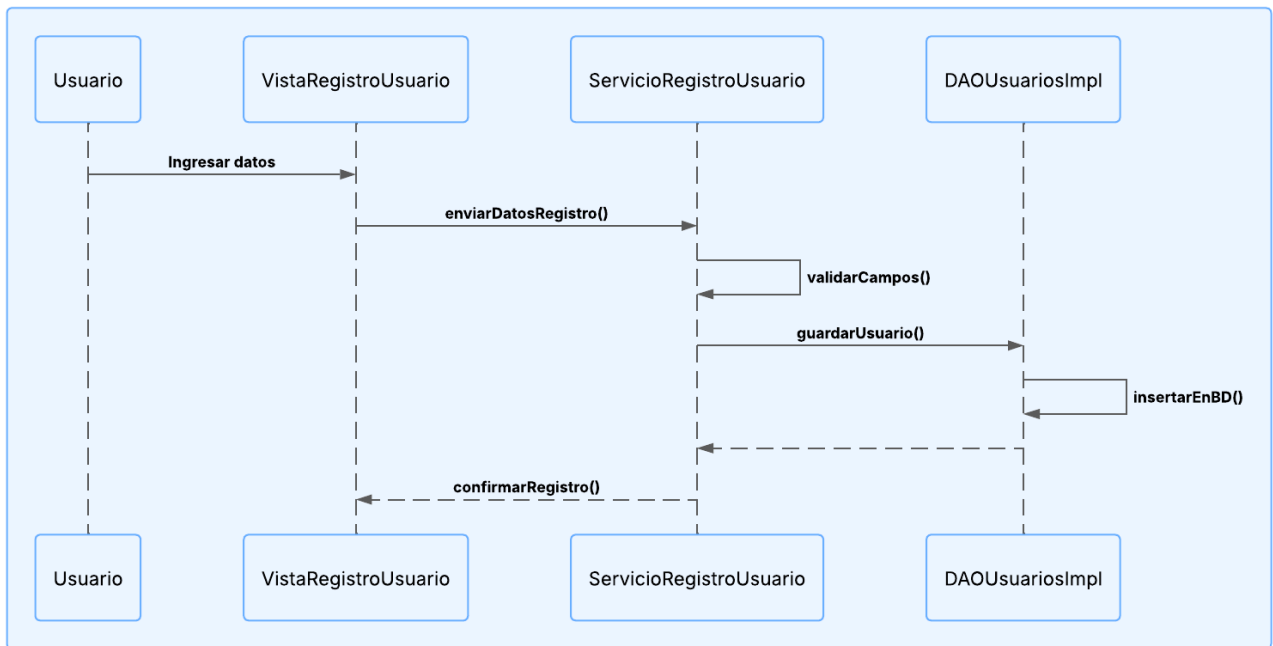
Nombre de método:	iniciarSesionUsuario	
Precondiciones:	El usuario debe haber ingresado correctamente su correo electrónico y contraseña en el formulario de inicio de sesión.	
Poscondiciones:	Si las credenciales son válidas, el sistema inicia una sesión para el usuario y lo redirige a la página principal de la plataforma. Si no lo son, se muestra un mensaje de error.	
Parámetros y retorno		
Nombre	Tipo	Descripción
correo	String	Correo electrónico registrado por el usuario.

contraseña	String	Contraseña correspondiente al correo ingresado.
Valor de retorno:	boolean – Retorna “true” si el inicio de sesión fue exitoso, false si las credenciales son inválidas.	
Excepciones		
Nombre	Descripción	
IllegalArgumentEx ception	Indica que uno o ambos campos fueron enviados vacíos o con formato incorrecto.	
AuthenticationExce ption	Se lanza cuando las credenciales no coinciden con ningún usuario registrado.	
BusinessComponen tException	Señala un error interno en la lógica de autenticación o comunicación con la base de datos.	

Nombre de método:	iniciarSesionUsuario	
Precondiciones:	El usuario debe haber ingresado un correo electrónico válido. Aparte, la contraseña no debe estar vacía y debe cumplir con el formato requerido.	
Poscondiciones:	Si las credenciales son válidas, se crea una sesión activa para el usuario.	
	El usuario es redirigido a la interfaz principal de la plataforma.	
	Si las credenciales son incorrectas, se muestra un mensaje de error.	
Parámetros y retorno		
Nombre	Tipo	Descripción
correo	String	Correo electrónico del usuario que intenta iniciar sesión.
contraseña	String	Contraseña ingresada para validar el acceso del usuario.

ServAutenticacionUsuario					
Valor de retorno:	boolean	Indica si el inicio de sesión fue exitoso (true) o fallido (false).			
Excepciones					
Nombre	Descripción				
IllegalArgumentException	Indica que el correo o la contraseña tienen un formato inválido o están vacíos.				
AuthException	Señala que el usuario no fue encontrado o que la autenticación falló.				
DAOUsuario					
Nombre de método:	verificarCredenciales				
Precondiciones:	El correo electrónico y la contraseña han sido provistos y tienen un formato válido.				
Poscondiciones:	True (si las credenciales coinciden; de lo contrario, se lanza una excepción.)				
Parámetros y retorno					
Nombre	Tipo	Descripción			
correo	String	Correo electrónico con el que el usuario intenta iniciar sesión.			
contraseña	String	Contraseña proporcionada para verificar la identidad del usuario.			
Valor de retorno:	boolean	(Retorna true si las credenciales son válidas y se autentica correctamente.)			
Excepciones					
Nombre	Descripción				
IllegalArgumentException	Indica que algún parámetro ingresado no es válido (por ejemplo, campos vacíos).				
SQLException	Señala que hubo un error al acceder a la base de datos durante la verificación.				

Diagrama de Secuencia UML: Registrar Usuario en Sistema Web con Arquitectura en Capas

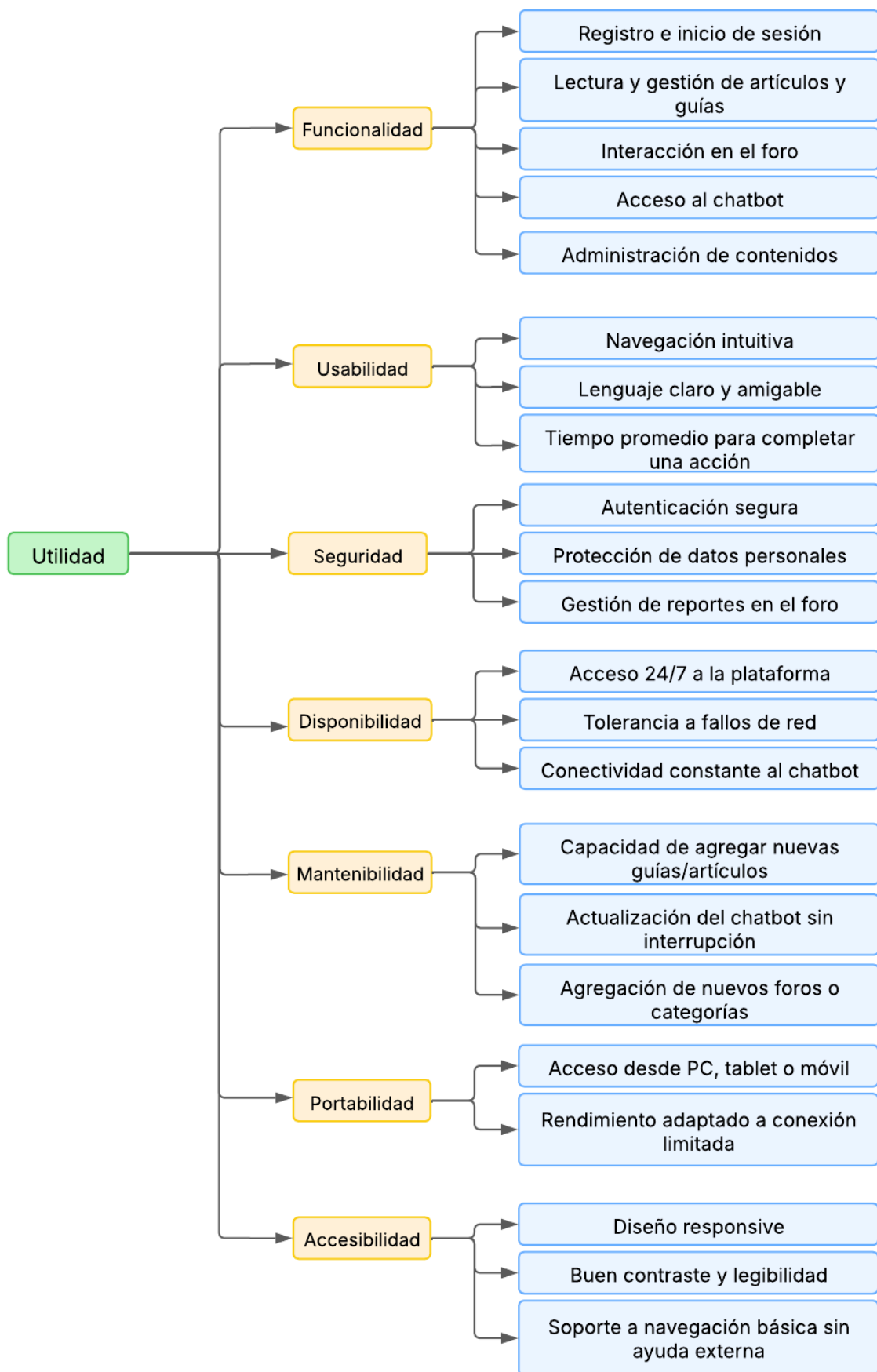


	Táctica	Justificación	Comentarios
EAC-01, EAC-02, EAC-05	Middleware de Adaptación	Esta táctica mejora la usabilidad al garantizar una experiencia consistente en diferentes plataformas (web, móvil, APIs) mediante una capa de abstracción que unifica las interacciones. En seguridad, centraliza mecanismos de autenticación y autorización (como OAuth2 o JWT), evitando redundancias y reduciendo puntos vulnerables. Para la portabilidad, desacopla la lógica de negocio de dependencias externas (bases de datos, servicios cloud), facilitando migraciones entre entornos sin modificar el código base.	Se implementa con API Gateways (Kong, Apigee) o brokers de mensajería (RabbitMQ). Requiere estandarizar protocolos (REST/GraphQL) y definir contratos claros entre componentes.
EAC-01, EAC-02, EAC-05	Configuración Externa	Favorece la usabilidad al permitir personalizar interfaces (idiomas, temas) mediante ajustes en tiempo real, sin recompilaciones. En seguridad, habilita la rotación segura de credenciales y certificados mediante herramientas como HashiCorp Vault, minimizando exposiciones. Para la portabilidad, separa configuraciones específicas por entorno (dev, producción, cloud), asegurando compatibilidad con infraestructuras diversas mediante archivos externos (.env, YAML) o servicios gestionados (Spring Cloud Config).	Requiere integración con orquestadores (Kubernetes) y validación de permisos para evitar accesos no autorizados a configuraciones sensibles.

	Decisión	Justificación	Comentarios
EAC-01, EAC-02, EAC-05	Táctica: Asegurar un inicio de sesión seguro	Garantizar un inicio de sesión seguro es esencial para proteger la confidencialidad e integridad de las credenciales del usuario. Esto se logra mediante el uso de protocolos de comunicación cifrados (como HTTPS), almacenamiento seguro de contraseñas (usando hash y salt), y la implementación de mecanismos contra ataques comunes como fuerza bruta, inyección de SQL o robo de sesiones.	Esta táctica se implementa en la capa de autenticación y está estrechamente relacionada con políticas de seguridad de la aplicación. Requiere la integración de librerías o servicios confiables de gestión de identidad y autenticación. También puede implicar el uso de autenticación multifactor (MFA) para mejorar la seguridad del sistema.
EAC-01, EAC-02, EAC-05	Táctica: Verificar que el inicio de sesión se complete con éxito	Verificar el éxito del inicio de sesión garantiza que los usuarios accedan al sistema solo si sus credenciales han sido validadas correctamente. Esta verificación permite establecer una sesión autenticada o token de acceso y evita accesos indebidos. Además, proporciona retroalimentación adecuada al usuario ante errores de autenticación.	Esta táctica se implementa en la lógica de negocio de autenticación y en el manejo de sesiones. Implica gestionar correctamente los estados de sesión o tokens (por ejemplo, JWT), y registrar eventos de inicio de sesión exitoso o fallido. Su correcta implementación es crucial para mantener un control de acceso confiable en la aplicación.

5. Evaluación

ID	Decisión arquitectónica
DA-01	Adoptar una arquitectura en capas (presentación, lógica de negocio, datos) para organizar el sistema.
DA-02	Implementar el sistema como una aplicación web responsive, accesible desde navegadores en PC y dispositivos móviles
DA-03	Utilizar tecnologías web de código abierto como, HTML, CSS, JavaScript, PHP, MySQL.
DA-04	Incorporar un chatbot basado en IA como módulo externo integrado a través de API de Gemini y seteado para ser adiestrado en conversaciones psicológicas.
DA-05	Usar base de datos relacional para almacenar usuarios, bitácoras, contenido (artículos/guías), comentarios y reportes.
DA-06	Aplicar sistema de roles para diferenciar funcionalidades entre usuarios normales y administradores.
DA-07	Incorporar medidas básicas de seguridad, como autenticación cifrada (hash de contraseñas) y validación de sesiones.
DA-08	Permitir la descarga de contenido educativo formato PDF bajo control de permisos del usuario.
DA-09	Diseñar el sistema para ser modular y extensible, permitiendo agregar nuevos paquetes, artículos, guías o foros.



Análisis de enfoque arquitectónicos	
Escenario núm 1.	Registro y navegación desde un dispositivo móvil con red limitada
Atributo(s)	Usabilidad, disponibilidad, portabilidad
Entorno	Un nuevo usuario accede desde un teléfono móvil con conexión débil
Estímulo	El usuario intenta registrarse, navegar por artículos y usar el chatbot
Respuesta	El sistema debe cargar correctamente, mostrar contenido accesible, responder a interacciones clave sin fallos y permitir registro fluido.

Decisión arquitectónica	Punto de sensibilidad	Equilibrio	Riesgo	No-riesgo
Arquitectura en capas	Cómo se comunican las capas en condiciones de red inestable	Permite separación lógica que facilita gestión de errores	R1: lentitud en la capa de presentación afecta la percepción del sistema	NR1: Portabilidad gracias a separación de capas
Diseño responsive / mobile-first	Adaptabilidad a pantallas pequeñas y rendimiento en dispositivos móviles	Mejora la usabilidad en móviles	R2: textos o botones pueden quedar inaccesibles	NR2: Navegación fluida en pantallas móviles pequeñas
Infraestructura ligera (hosting básico)	Carga bajo red lenta	Ahorra recursos	R3: tiempo de carga del chatbot o de artículos	NR3: Optimización de carga por modularidad

Tabla que explica el enfoque de arquitectura en el escenario de: “Registro y navegación desde un dispositivo móvil con red limitada”

Análisis de enfoque arquitectónicos	
Escenario núm 2.	Publicación y respuesta en el foro
Atributo(s)	Funcionalidad, seguridad, usabilidad
Entorno	Un usuario inicia sesión, entra al foro, publica una pregunta y responde a otro usuario
Estímulo	El usuario interactúa con otros mediante publicaciones
Respuesta	El sistema debe permitir crear, editar o eliminar publicaciones propias, mientras protege contra spam y publicaciones ofensivas.

Decisión arquitectónica	Punto de sensibilidad	Equilibrio	Riesgo	No-riesgo
Control de acceso por roles	Foro accesible solo tras autenticación	Previene interacciones anónimas no deseadas	R1: acceso no autorizado por mala validación de sesión	NR1: Moderación basada en autenticación
Moderación y reportes por admin	Posibilidad de revisar publicaciones	Permite control comunitario	R2: saturación de reportes	NR2: Filtrado y eliminación de contenido ofensivo
Separación de lógica y presentación	Foro como módulo separado	Aísla errores	R3: foros desacoplados no sincronizan con perfiles	NR3: Foro estable y reutilizable en otras secciones

Tabla que explica el enfoque de arquitectura en el escenario de: “Publicación y respuesta en el foro”

Análisis de enfoque arquitectónicos	
Escenario núm 3.	Interacción con el chatbot
Atributo(s)	Usabilidad, rendimiento, mantenibilidad
Entorno	Un usuario hace varias preguntas en poco tiempo al chatbot
Estímulo	Solicita orientación sobre síntomas o contenido disponible
Respuesta	El chatbot debe responder rápido y sin errores, y poder actualizar sus respuestas fácilmente

Decisión arquitectónica	Punto de sensibilidad	Equilibrio	Riesgo	No-riesgo
Chatbot como módulo independiente	Requiere comunicación con base de datos y otros servicios como gemini	Escalabilidad, se puede actualizar aparte	R1: latencia alta si API no responde bien	NR1: Fácil actualización sin afectar el sistema
Base de conocimientos actualizable	Permite modificar respuestas del bot en tiempo real	Evita reinicio del sistema	R2: errores humanos al alimentar el bot	NR2: Admin puede mejorar respuestas progresivamente
Comunicación asincrónica	El usuario espera respuesta sin bloqueos	Mejora experiencia del usuario	R3: respuesta fuera de contexto si se pierde orden	NR3: Conversación fluida en la mayoría de escenarios

Tabla que explica el enfoque de arquitectura en el escenario de: “Interacción con el chatbot ”

Análisis de enfoque arquitectónicos	
Escenario núm 4.	Administración de contenido por parte del administrador
Atributo(s)	Mantenibilidad, seguridad, funcionalidad
Entorno	Un administrador desea actualizar una guía y eliminar una publicación del foro
Estímulo	Ingresa al panel administrativo y realiza cambios
Respuesta	El sistema debe permitir cambios inmediatos y mantener trazabilidad segura

Decisión arquitectónica	Punto de sensibilidad	Equilibrio	Riesgo	No-riesgo
Módulo exclusivo para administrador	Gestión de contenido aislada del usuario	Mejora seguridad y claridad de permisos	R1: error en permisos permite cambios indebidos	NR1: Separación clara evita interferencias
Panel de control web	Facilidad para editar desde navegador	Aumenta velocidad de administración	R2: no carga correctamente en dispositivos móviles	NR2: Admin puede operar desde cualquier ubicación
Uso de base de datos relaciona	Artículos y foros bien organizados	Asegura integridad de los datos	R3: eliminación accidental de recursos compartidos	NR3: Datos normalizados y seguros en cada módulo

Tabla que explica el enfoque de arquitectura en el escenario de: “ Administración de contenido por parte del administrado ”

Análisis de enfoque arquitectónicos	
Escenario núm 5.	Guardado y descarga de artículos o guías
Atributo(s)	Funcionalidad, usabilidad, disponibilidad
Entorno	Un usuario navega en la biblioteca, guarda una guía y la descarga para verla offline
Estímulo	El usuario marca como favorito y descarga el contenido
Respuesta	El sistema debe permitir estas acciones sin errores y guardar las preferencias del usuario

Decisión arquitectónica	Punto de sensibilidad	Equilibrio	Riesgo	No-riesgo
Gestión de contenido descargable	Formato y tamaño de los archivos	Archivos livianos significa mejor experiencia	R1: archivos pesados no se descargan bien	NR1: Descarga en PDF
Perfil de usuario con favoritos	Relación entre usuario y contenido personalizado	Mejora retención de usuario	R2: mal funcionamiento del guardado	NR2: Guardado inmediato y persistente
Base de datos organizada por recursos	Identificación única de cada artículo o guía	Facilita búsquedas y gestión	R3: múltiples versiones de la misma guía	NR3: Sistema actualizado sin duplicidad

Tabla que explica el enfoque de arquitectura en el escenario de: “ Guardado y descarga de artículos o guías”

5.1 Escenarios básicos o secundarios

La tabla contiene posibles escenarios básicos y considerados secundarios por el hecho de que un usuario no los tome en cuenta o no le interesen, describe de manera breve los atributos involucrados y el nivel de riesgo, si en un futuro se requiere abordar cada uno de los escenarios planteados en esta tabla el arquitecto tendrá que solicitarlo.

Escenario	Atributos involucrados	Nivel de riesgo
El usuario edita su perfil de cuenta	Usabilidad, funcionalidad	Bajo
El administrador revisa reportes de usuarios en el foro	Seguridad, mantenibilidad	Medio
El sistema responde ante error de conexión del chatbot	Disponibilidad, rendimiento	Medio
El usuario intenta acceder a la plataforma sin conexión a Internet	Disponibilidad	Bajo
El administrador alimenta el chatbot con nuevas respuestas	Mantenibilidad	Bajo
El usuario elimina una guía de favoritos	Funcionalidad	Bajo
El sistema no permite guardar artículos sin iniciar sesión	Seguridad, funcionalidad	Bajo
Un usuario reporta contenido ofensivo	Seguridad, funcionalidad, usabilidad	Medio

5.2. Resultados de la evaluación

Al analizar los escenarios propuestos para la evaluación, se tienen identificados los siguientes resultados planteados como riesgos y no-riesgos:

Riesgos:

R1: No se ha especificado el protocolo de comunicación con el servicio externo del chatbot, lo cual genera incertidumbre respecto al manejo de errores, latencia o caídas del servicio externo.

R2: No se evidencia cómo se manejarán transacciones concurrentes, como múltiples usuarios guardando bitácoras o escribiendo en el foro al mismo tiempo.

R3: El sistema expone funcionalidades (como guardar favoritos o eliminar publicaciones) que modifican estado, pero no se ha definido explícitamente el manejo de sesiones, lo cual puede afectar la consistencia si se usa un modelo stateless.

R4: No hay pruebas ni detalles específicos sobre el rendimiento del chatbot bajo carga o su comportamiento en conexiones lentas, lo cual representa un riesgo de desempeño.

R5: Aunque se menciona la modularidad del sistema, no se han modelado explícitamente mecanismos de autenticación y autorización, lo que representa un riesgo en seguridad.

R6: No se ha establecido cómo se manejarán los errores entre capas, ni se han propuesto mecanismos de logging o trazabilidad de fallos, lo cual podría dificultar el mantenimiento.

No riesgos:

- **NR1:** El uso de una arquitectura en capas y componentes bien definidos **garantiza mantenibilidad y claridad estructural**, permitiendo escalabilidad y fácil actualización.

- **NR2:** La interacción entre presentación, lógica de negocio, datos e integración está bien delimitada, lo que **favorece la modularidad** y permite mantener o sustituir componentes sin afectar el sistema completo.
- **NR3:** El diseño responsivo y centrado en el usuario, así como el enfoque en accesibilidad móvil, **favorecen la usabilidad**, especialmente en un contexto como el salvadoreño, con alta demanda de portabilidad.
- **NR4:** El chatbot y el sistema de artículos están desacoplados y definidos como módulos independientes, lo que permite **facilidad de prueba, evolución y despliegue por separado**.