Formato para evaluación de la arquitectura de software

1. Datos generales

Campo	Descripción
Nombre del proyecto:	Solidarity: Plataforma solidaria
Integrantes:	Carlos Daniel Sánchez Palomino Dylan De Vega Daniel Puerta Yesid Soto Cristian Mendoza
Fecha:	18/11/2025
Entorno de prueba:	Local

2. Atributos de calidad evaluados

Seleccionamos 6 **atributos clave** (los más visibles en un prototipo):

1. **Rendimiento (Performance)**
2. **Mantenibilidad (Modularity / Code Organization)**
3. **Escalabilidad básica (Potential to Grow)**
4. **Seguridad básica (Safe Access & Data Handling)**
5. **Usabilidad (User Interaction)**
6. **Independencia tecnológica**

---

3. Escala de valoración (1 a 5)

Nivel	Descripción general
1 - Deficiente	El atributo no está considerado o falla constantemente.
2 - Bajo	Se perciben intentos, pero hay fallas frecuentes o mala implementación.
3 - Aceptable	Cumple de forma básica los criterios. Puede mejorar.
4 - Bueno	Cumple bien el atributo con pocas observaciones.
5 - Excelente	Supera las expectativas; el diseño refleja madurez técnica.

4. Tabla de evaluación simplificada

Atributo	Indicador observable o prueba sencilla	Valor (1-5)	Comentarios / Evidencia
Rendimiento	El sistema responde en menos de 3 segundos en operaciones comunes (login, guardar, listar).	5	El sistema presenta tiempos de respuesta óptimos en sus funcionalidades básicas. Las consultas a la base de datos y el procesamiento de lógica de negocio están optimizados, garantizando fluidez incluso con cargas moderadas de usuarios.
Mantenibilidad	El código está organizado por capas o módulos (MVC, servicios, repositorios).	5	El sistema sigue una arquitectura modular (MVC y capas de servicios/repositorios), lo que facilita la lectura, comprensión y mantenimiento del código. Se aplican principios SOLID y se implementan excepciones personalizadas para un manejo de errores claro y consistente.
Escalabilidad básica	La arquitectura permitiría agregar más funcionalidades o usuarios sin re-diseñar todo.	5	La arquitectura modular y orientada a servicios permite la incorporación de nuevas funcionalidades o la integración con otros sistemas sin reescribir el núcleo. Además, se han considerado patrones de diseño que soportan crecimiento y cambios futuros.
Seguridad básica	Hay control de acceso (login), validación de datos y no exposición de información sensible.	5	El sistema implementa control de acceso con login seguro, validación exhaustiva de datos de entrada y buenas prácticas para proteger información sensible. Se han implementado capas de seguridad tanto en la lógica de negocio como en el acceso a la base de datos.
Usabilidad	La interfaz es comprensible	5	La interfaz es intuitiva y consistente, guiando al usuario a través de los flujos de trabajo de

Atributo	Indicador observable o prueba sencilla	Valor (1-5)	Comentarios / Evidencia
	y permite completar tareas sin errores confusos.		manera clara. La interacción evita errores comunes y proporciona retroalimentación útil ante acciones incorrectas o excepciones del sistema.
Independencia Tecnológica	Capaz de modificar Lenguaje , BD y servidores	5	La arquitectura desacoplada permite reemplazar o actualizar componentes tecnológicos (bases de datos) sin afectar la lógica principal del sistema. El diseño orientado a interfaces y patrones de abstracción permite adaptabilidad a nuevas tecnologías.

5. Cálculo del puntaje final

$$\text{Puntaje total} = \frac{5+5+5+5+5+5}{6} = 5$$

Resultado global	Interpretación
1.0 – 2.0	Nivel bajo – La arquitectura requiere replanteamiento.
2.1 – 3.5	Nivel medio – Cumple lo básico, puede mejorar.
3.6 – 4.5	Nivel bueno – Arquitectura bien estructurada.
4.6 – 5.0	Nivel sobresaliente – Arquitectura sólida y bien fundamentada.

6. Retroalimentación global

Fortalezas:

- Arquitectura modular del sistema, que permite fácil mantenimiento y extensión de funcionalidades.
- Rápido tiempo de respuesta en operaciones críticas.
- Alta cohesión y bajo acoplamiento, facilitando la comprensión y modificación del código.

- Respeto de los principios SOLID, asegurando un diseño robusto y flexible.
- Manejo de excepciones personalizado y consistente, que mejora la estabilidad y la experiencia del usuario.

Oportunidades de mejora:

- Implementación de características faltantes del sistema.
- Escalamiento para agregar nuevas funcionalidades, como la gestión de domicilios y la apertura de un programa específico para domiciliarios, lo que permitirá ampliar la cobertura del sistema sin re-diseñar su arquitectura central.