

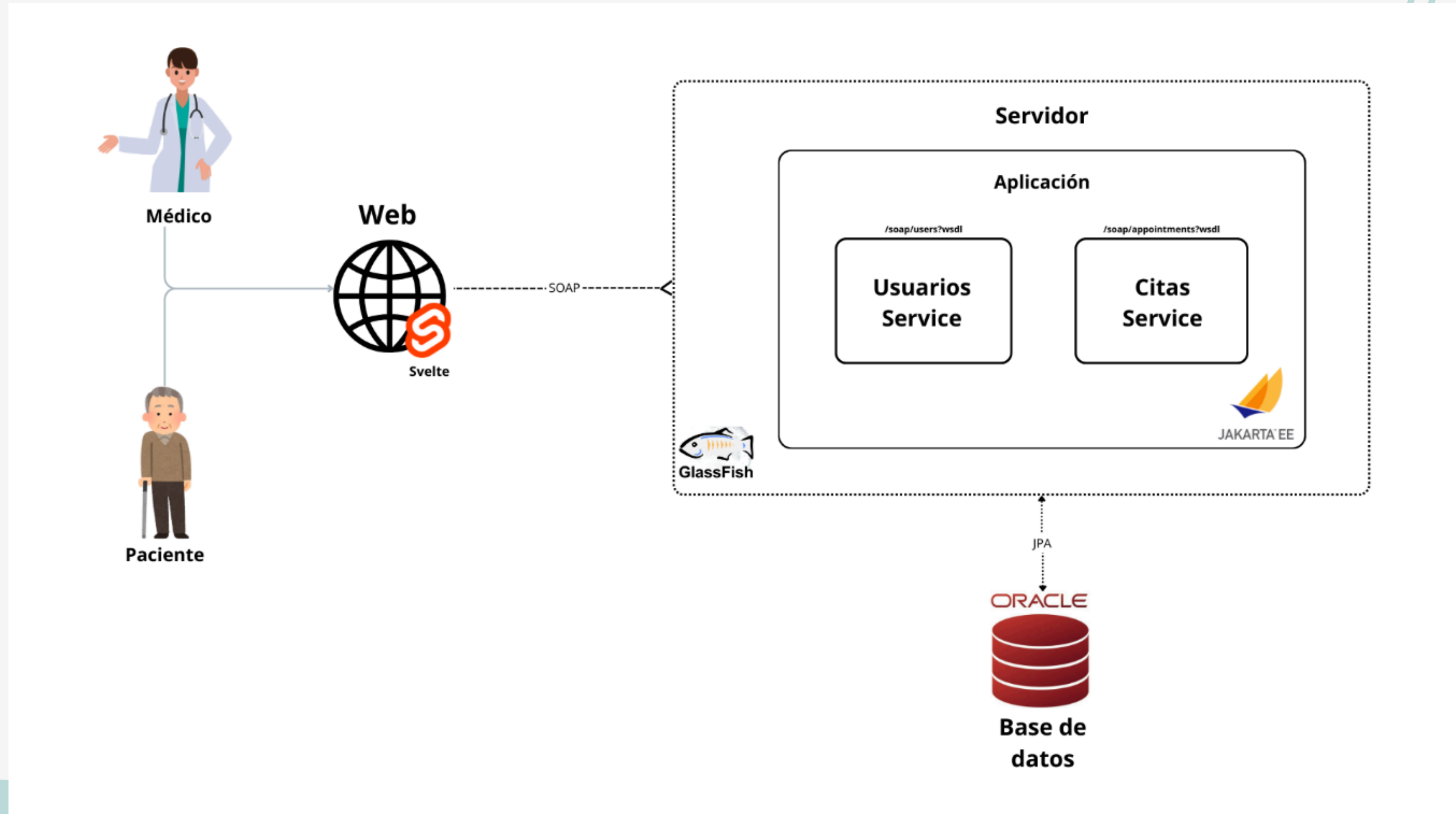


SISTEMA DE GESTIÓN DE CITAS MÉDICAS

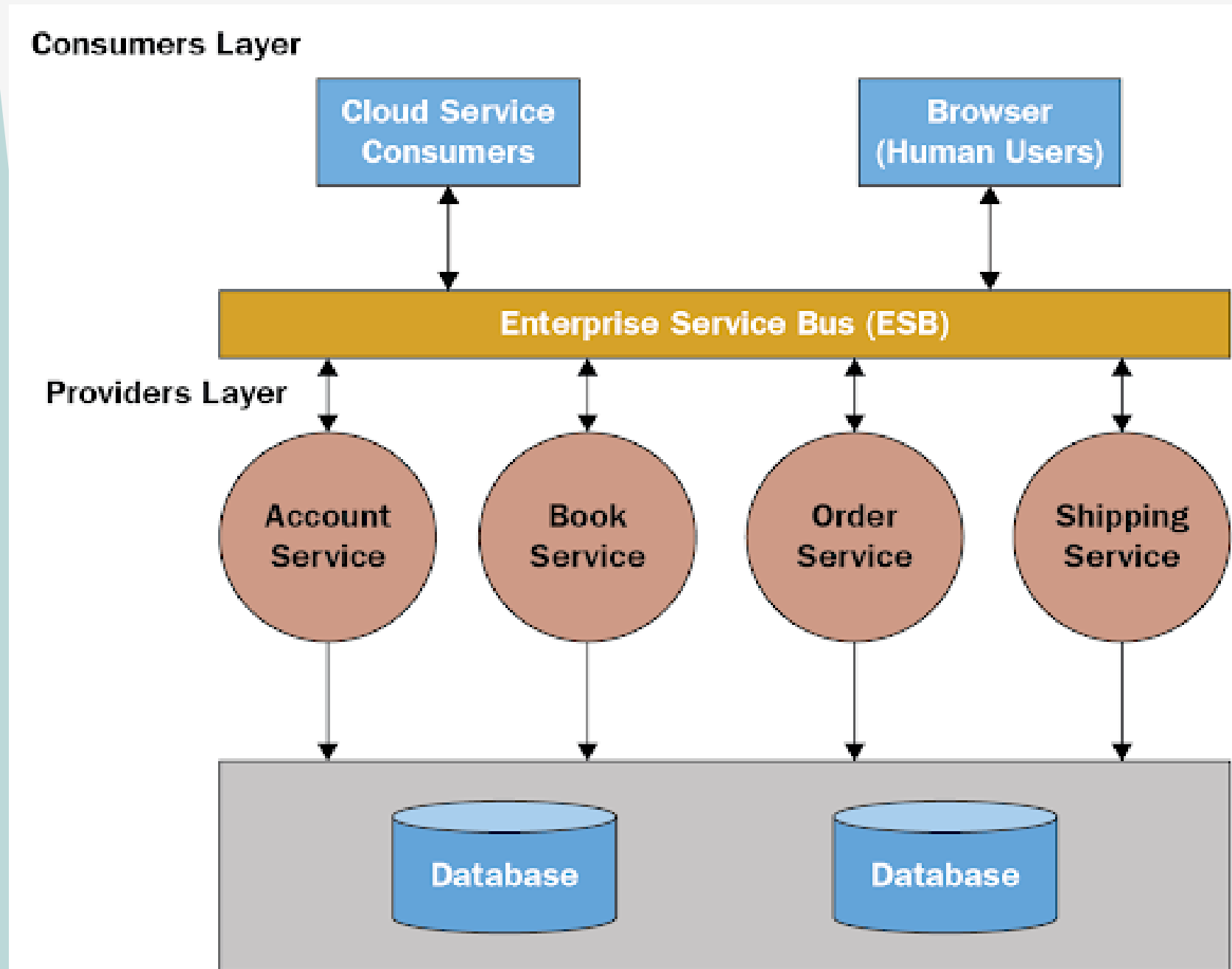


Camilo Mora
Santiago Guerrero
Santiago Barajas

Stack usado



Estilo SOA



¿Qué es?

Estilo arquitectónico que organiza un sistema en servicios independientes que se comunican entre sí usando protocolos estandarizados (como SOAP o WSDL). Cada servicio es autónomo, reutilizable e interoperable.

Historia y evolución

Surgió en los **2000s** como solución para integrar sistemas legados en grandes empresas. Fue impulsado por gigantes como IBM, Oracle y Microsoft. Aunque ahora compite con arquitecturas de microservicios, sigue siendo esencial en sectores donde se necesita robustez, trazabilidad y estándares estrictos.



Ventajas

- **Modularidad:** Servicios independientes que facilitan escalado.
- **Interoperabilidad:** Permite integrar tecnologías distintas.
- **Reusabilidad:** Servicios reutilizables en distintas plataformas.
- **Escalabilidad organizacional:** Ideal para equipos distribuidos.

Desventajas

- **Complejidad:** Mayor dificultad en diseño y despliegue inicial, especialmente para proyectos pequeños.
- **Overhead:** SOAP/XML pueden ser lentos comparados con REST.



¿Dónde usarlo?

- Sistemas empresariales legacy
- Orquestación compleja de servicios (ESB)
- Interoperabilidad entre plataformas heterogéneas
- Reutilización de servicios compartidos



¿Dónde se usa?



Svelte

¿Qué es?

Framework de desarrollo web que compila componentes a JavaScript puro en tiempo de compilación. No usa virtual DOM, lo que reduce el peso del bundle y mejora el rendimiento.



Historia y evolución

2016
Nace



Creado por Rich Harris.
Enfocado en eliminar el runtime en el navegador: los componentes se compilan a JS puro.

2019
Svelte 3



Cambio radical: nueva API reactiva.
Marca el inicio de su crecimiento real en la comunidad.

2021
SvelteKit



Framework oficial full-stack sobre Svelte.
Permite SSR, routing, prefetching, y más.

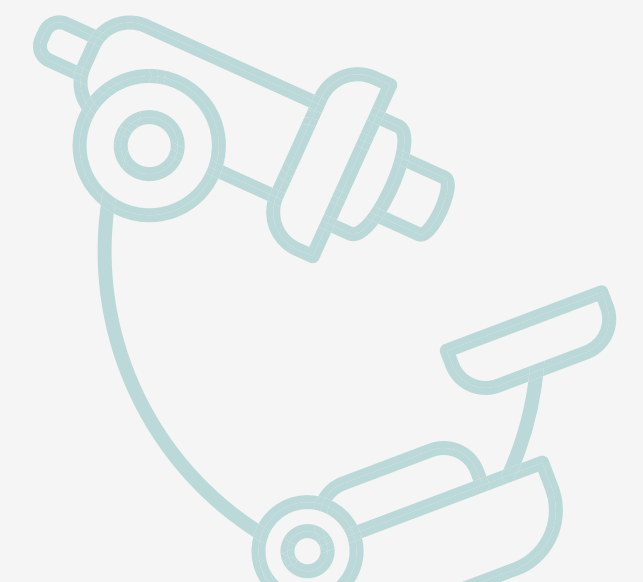


Ventajas

- **Rendimiento óptimo:** Archivos ligeros, carga más rápida.
- **Curva de aprendizaje baja:** HTML, CSS y JS puro.
- **Reactividad automática:** No requiere "hooks" o librerías externas.
- Ideal para zonas con baja conectividad.

Desventajas

- **Ecosistema pequeño:** Menos plugins que otros frameworks.
- **Sin soporte SOAP nativo:** Requiere librerías adicionales.
- Menor adopción.
- Curva de aprendizaje diferente.



¿Dónde usarlo?

- Interfaces web ligeras.
- Proyectos pequeños o medianos
- Dashboards administrativos.



¿Dónde se usa?



Jakarta EE

¿Qué es?

Conjunto de especificaciones Java para crear aplicaciones empresariales robustas. Proporciona APIs para persistencia, servicios web, inyección de dependencias y más.



Historia y evolución

1999
Java 2 Enterprise Edition

Sun Microsystems lanza J2EE como una plataforma para crear aplicaciones empresariales robustas en Java.

2006
Java EE

Oracle adquiere Sun y renombra la plataforma como Java EE.
Mejora el soporte para APIs modernas (JPA, JSF, CDI)

2017-2018
Jakarta EE

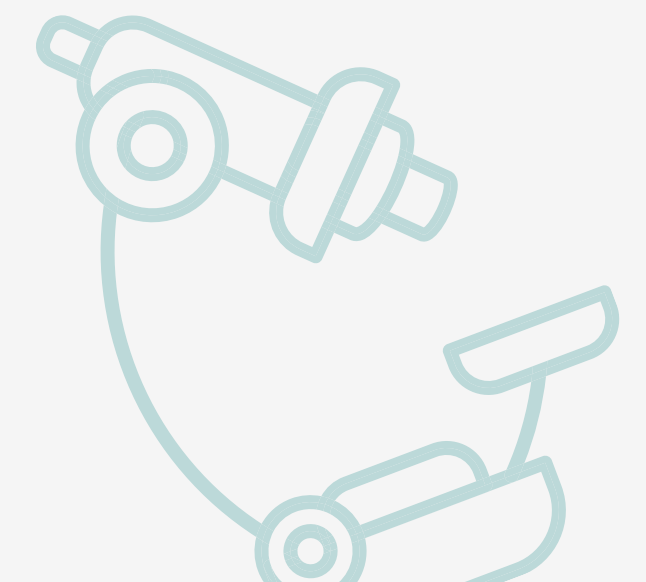
Oracle dona Java EE a la Eclipse Foundation. Cambia el nombre a Jakarta EE y se abre el desarrollo, se vuelve más comunitario y flexible.

Ventajas

- **Estandarización:** Basado en especificaciones maduras (JPA, CDI, JAX-WS).
- **Seguridad empresarial:** Control de acceso, encriptación, manejo de sesiones.
- **Escalabilidad:** Pensado para sistemas críticos.

Desventajas

- **Curva de aprendizaje:** Necesario conocer muchas especificaciones.
- Complejo para proyectos pequeños.
- Requiere servidor de aplicaciones como GlassFish.



¿Dónde usarlo?

- Aplicaciones empresariales de misión crítica.
- Sistemas transaccionales complejos.
- APIs RESTful y servicios backend estándar.



¿Dónde se usa?



GlassFish

¿Qué es?

Servidor open-source compatible con Jakarta EE. Permite desplegar aplicaciones empresariales, exponer servicios y administrar recursos como base de datos.



Historia y evolución

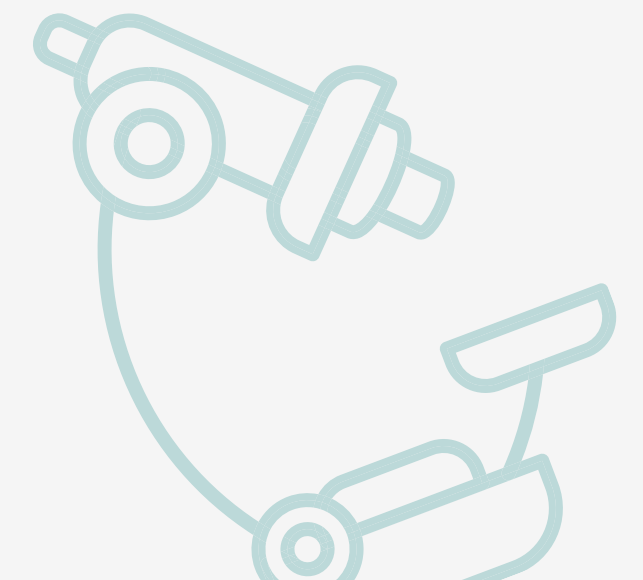


Ventajas

- Compatibilidad total con Jakarta EE.
- Consola web para despliegue rápido.
- Administración de conexiones con bases de datos.

Desventajas

- Menor rendimiento en producción.
- Actualizaciones lentas y comunidad reducida.

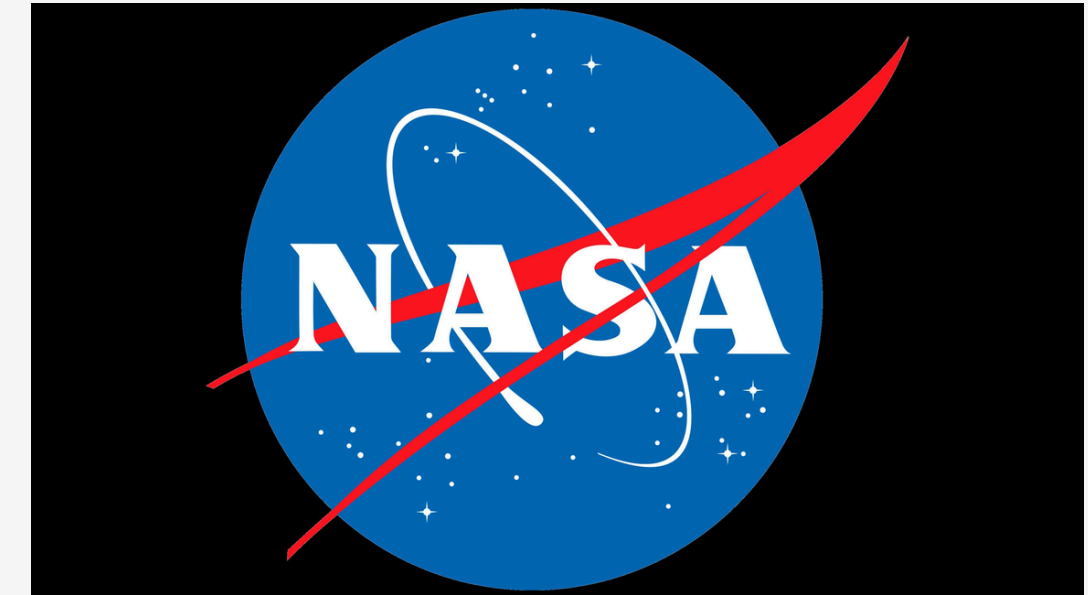


¿Dónde usarlo?

- Desarrollo y prueba de aplicaciones Jakarta EE.
- Sistemas administrativos y académicos.



¿Dónde se usa?



ORACLE

SOAP

¿Qué es?

Protocolo para enviar mensajes estructurados (en XML) entre aplicaciones. Utiliza contratos WSDL y garantiza seguridad, confiabilidad y compatibilidad.



Historia y evolución

Desarrollado en 1999 por Microsoft, IBM y otros. Dominó la industria durante años en sectores regulados. Hoy REST es más común, pero SOAP sigue en uso por su formalidad.

Ventajas

- **Contratos claros:** Basados en WSDL.
- **Seguridad:** Cifrado, firmas digitales, WS-Security.
- **Fiabilidad:** Transacciones distribuidas, entrega garantizada.

Desventajas

- Más pesado: Usa XML.
- Menos amigable para frontends modernos.



¿Dónde usarlo?

- Interoperabilidad con sistemas externos.
- Servicios con validación estricta.



¿Dónde se usa?



Amazon S3



Oracle DB

¿Qué es?

Gestor de bases de datos relacional empresarial. Soporta grandes volúmenes de datos, operaciones transaccionales, replicación y seguridad avanzada.



Historia y evolución

1979
Primer RDBMS comercial



Fundada como Software Development Laboratories (SDL) primer sistema de base de datos relacional comercial basado en



1990s-2000s
Consolidación



Oracle domina el mercado con versiones cada vez más robustas
Introduce características clave como replicación

2010s-presente
Oracle Cloud



Expansión al mundo cloud con Oracle Cloud Infrastructure (OCI)

Ventajas

- **Alta disponibilidad:** RAC, Data Guard.
- **Seguridad de nivel empresarial.**
- **Consultas optimizadas:** Vistas, particiones, índices.
- **Escalabilidad:** Vertical y horizontal.

Desventajas

- Licenciamiento costoso.
- Requiere infraestructura especializada.
- Demanda de recursos.



¿Dónde usarlo?

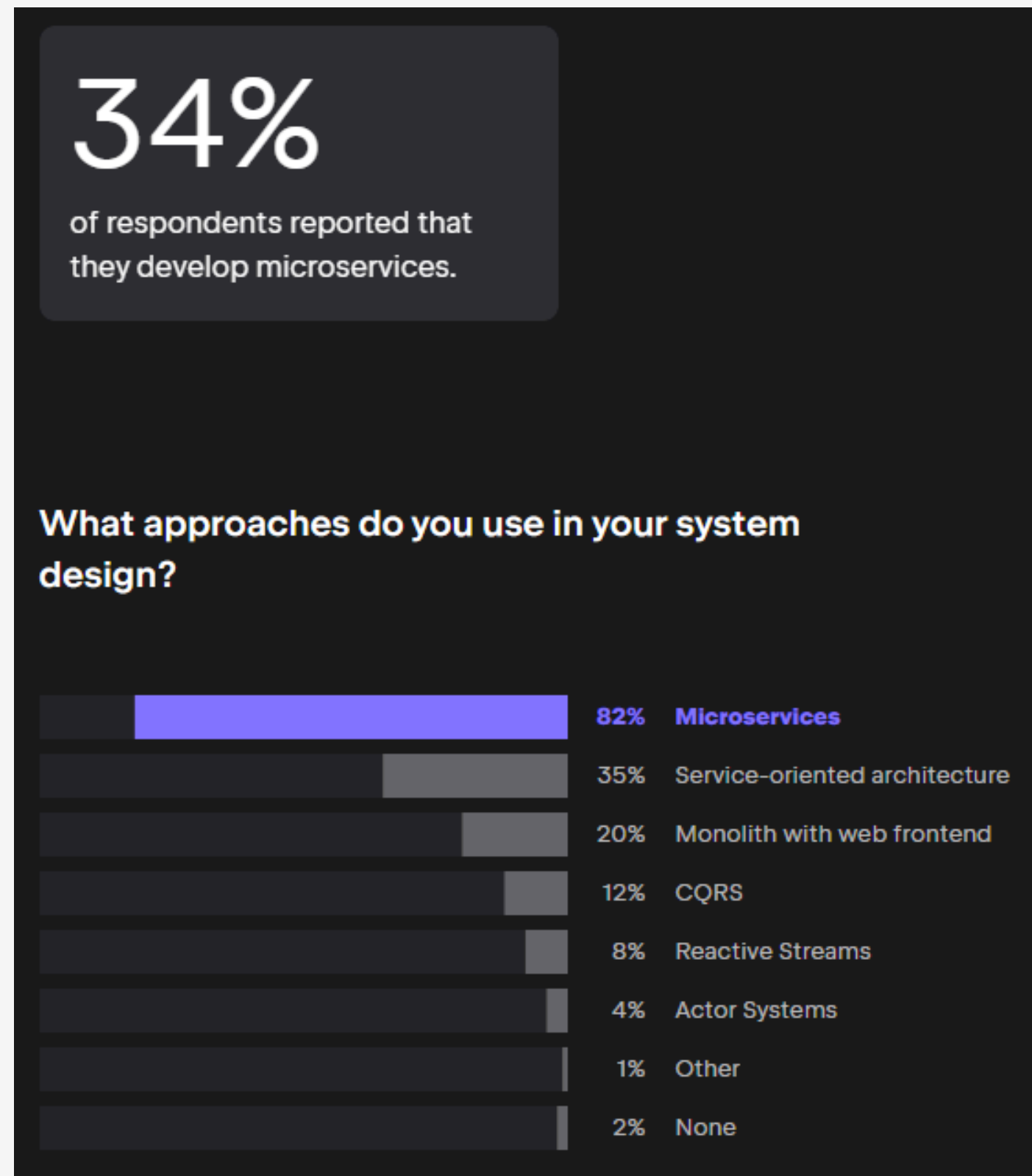
- Banca y servicios financieros.
- Sistemas de gestión empresarial (ERP, CRM).
- Gobiernos y servicios públicos.



¿Dónde se usa?



¿Qué tan común es el stack?

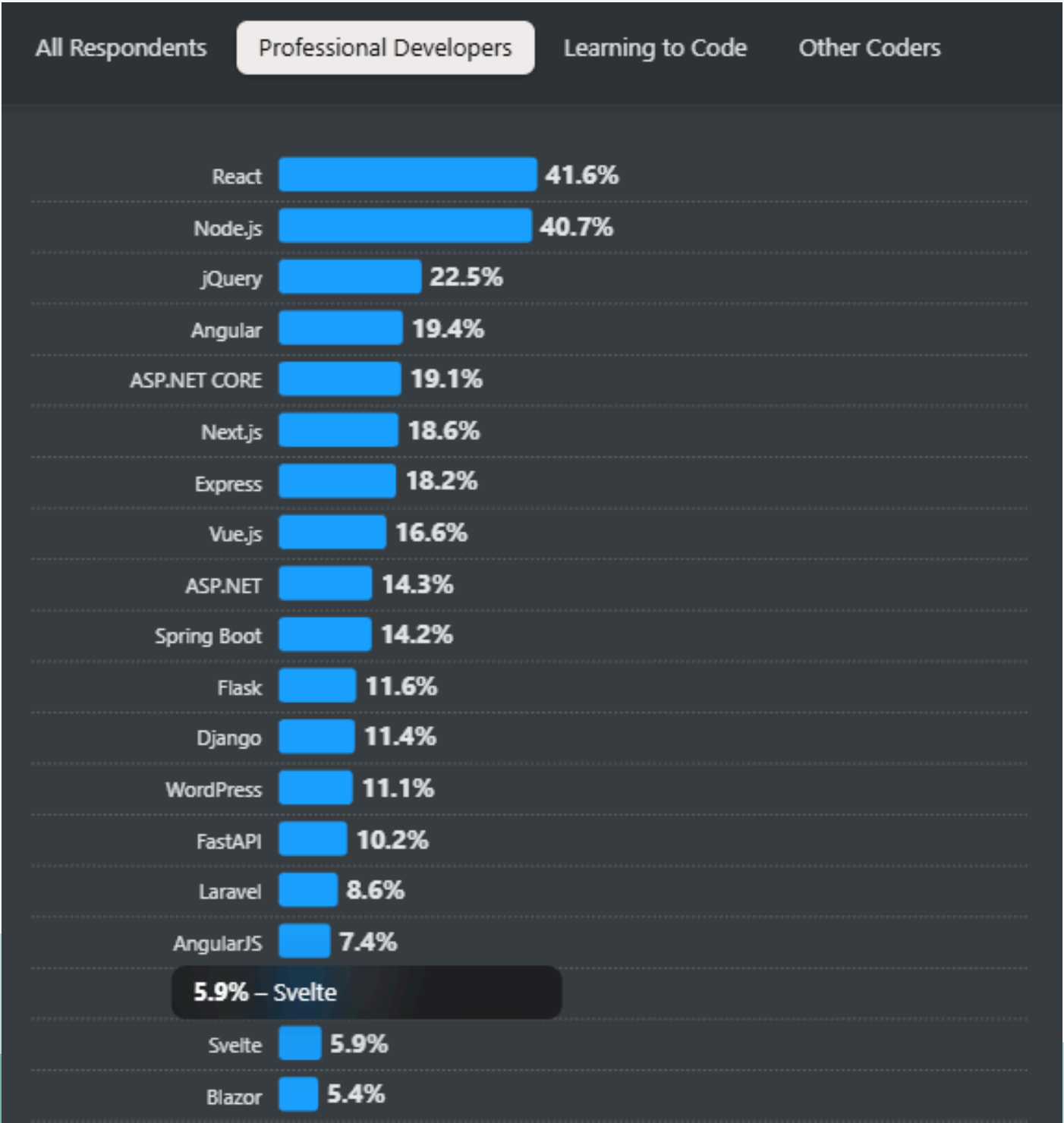


Estilo SOA

A pesar de dominar, especialmente entre 2005 y 2015, actualmente se ha visto desplazado por otras aproximaciones como microservicios

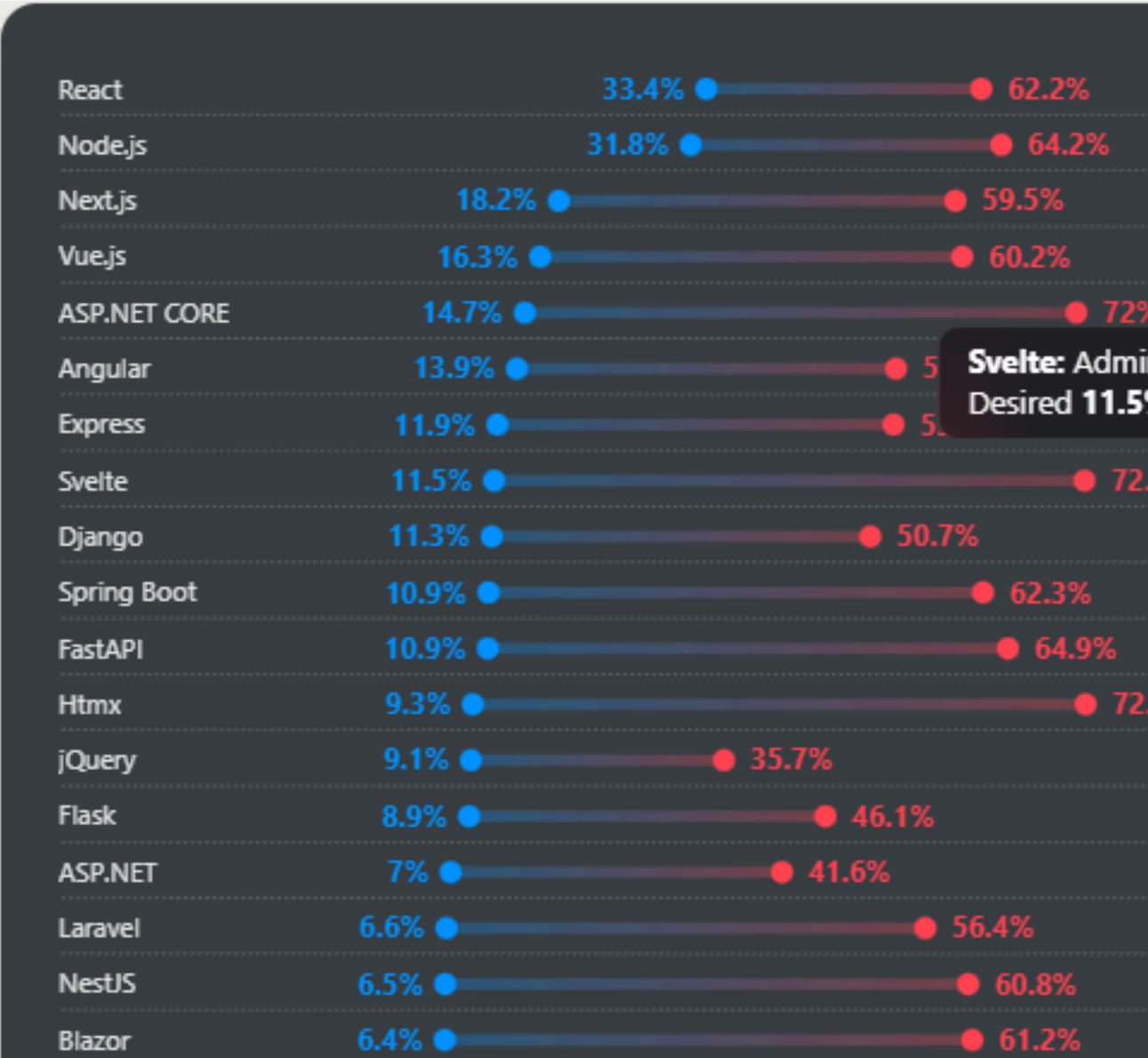
Svelte

Svelte ha ganado notable atención en la comunidad de desarrollo frontend. Aunque su adopción actual es moderada.



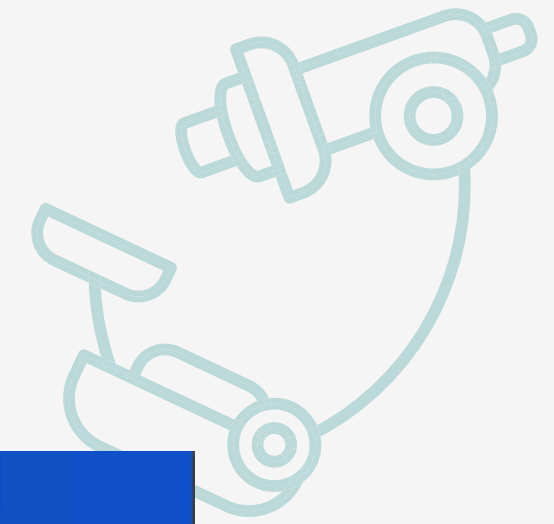
Web frameworks and technologies

73% of developers that used it want to keep working with Svelte. Fun fact: Our team at Stack Overflow used Svelte for the first time in building our 2024 Developer Survey results site. We could go on and on about Svelte, [listen to us do just that in a interview with one of our own.](#)



Jakarta EE

Continúa siendo una tecnología muy relevante, especialmente en contextos modernos de aplicaciones cloud-native.



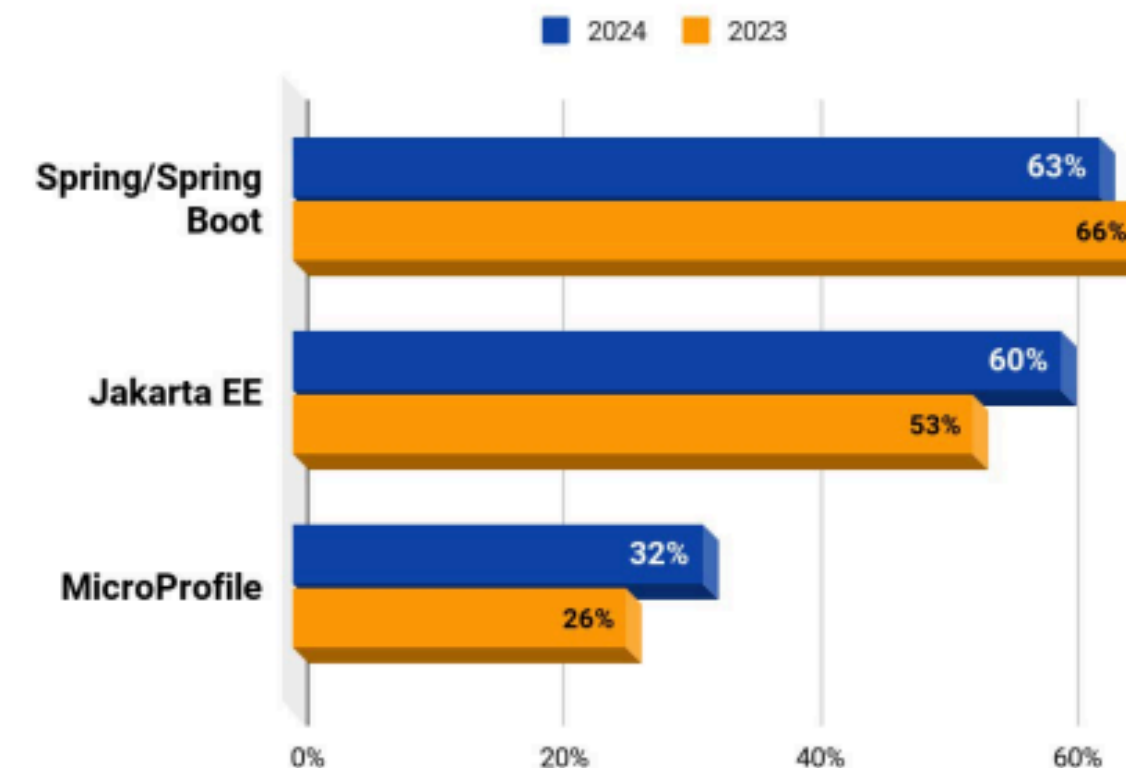
Key Takeaway

1

Top three Java frameworks for building cloud native applications:

- Spring/Spring Boot remains the top Java framework for building cloud native applications, with usage holding steady (66% in 2023 vs 63% in 2024).
- Jakarta EE has seen a notable rise in usage, growing from 53% in 2023 to 60% in 2024.
- MicroProfile has also experienced growth, with usage rising from 26% in 2023 to 32% in 2024.

These results reflect a growing interest and adoption of Jakarta EE and MicroProfile among developers, while Spring/Spring Boot maintains its strong position in the market.





Key Takeaway

5

▲

Top three architectural approaches for implementing Java systems in the cloud

46%

Hybrid

31%


Microservices

18%

Monolith

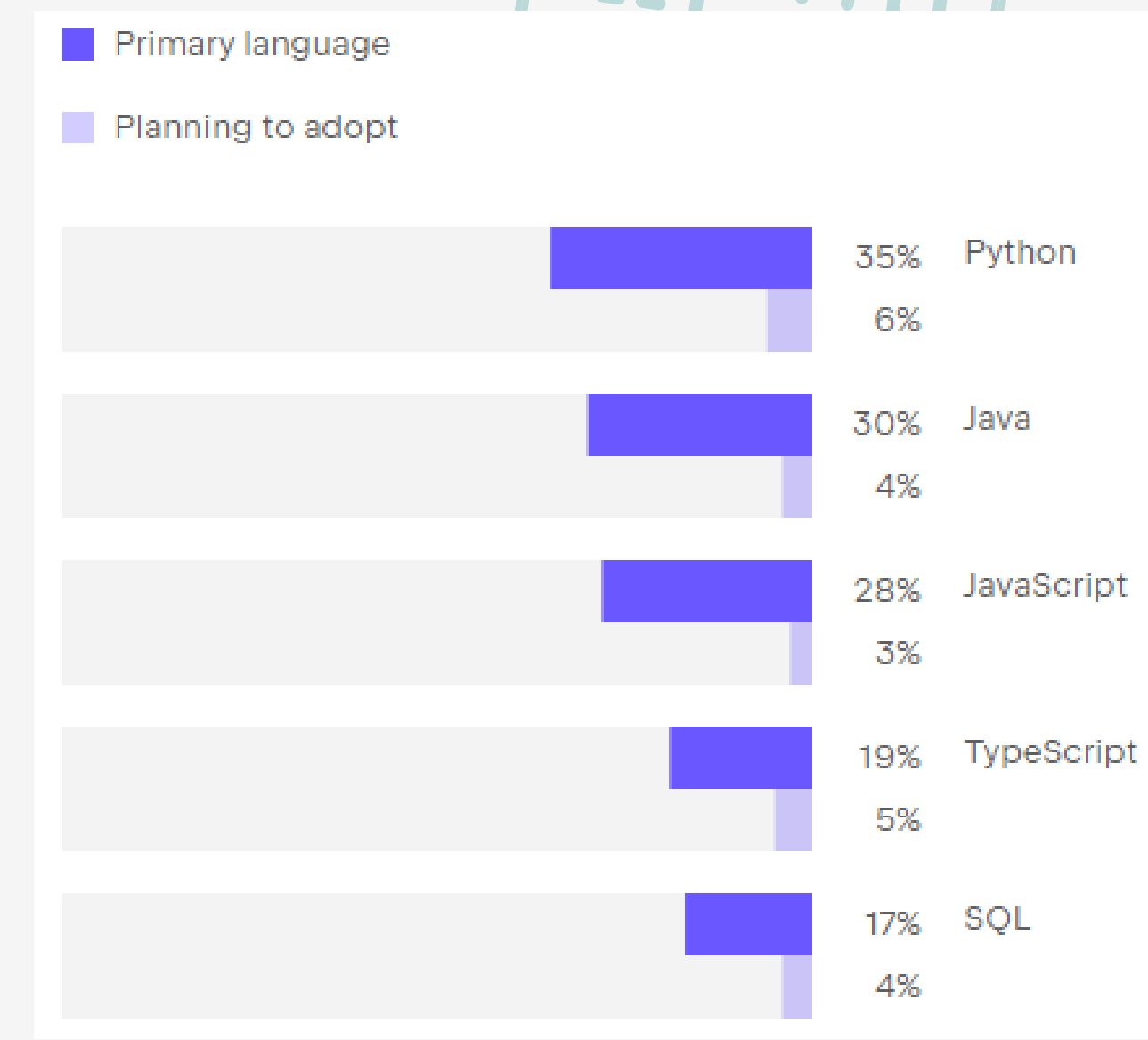
There has been a **notable increase in the adoption of a hybrid architectural approach** for implementing Java systems in the cloud, **rising from 41% in 2023 to 46% in 2024**. This suggests that more organisations are recognising the benefits of combining monolith and microservices, likely to leverage the strengths of both approaches.

Conversely, the usage of microservices architecture has decreased from 38% in 2023 to 31% in 2024. This decline may indicate that while microservices remain popular, some organisations are turning to hybrid solutions to more efficiently address complexity and integration challenges.

 JAKARTA EE

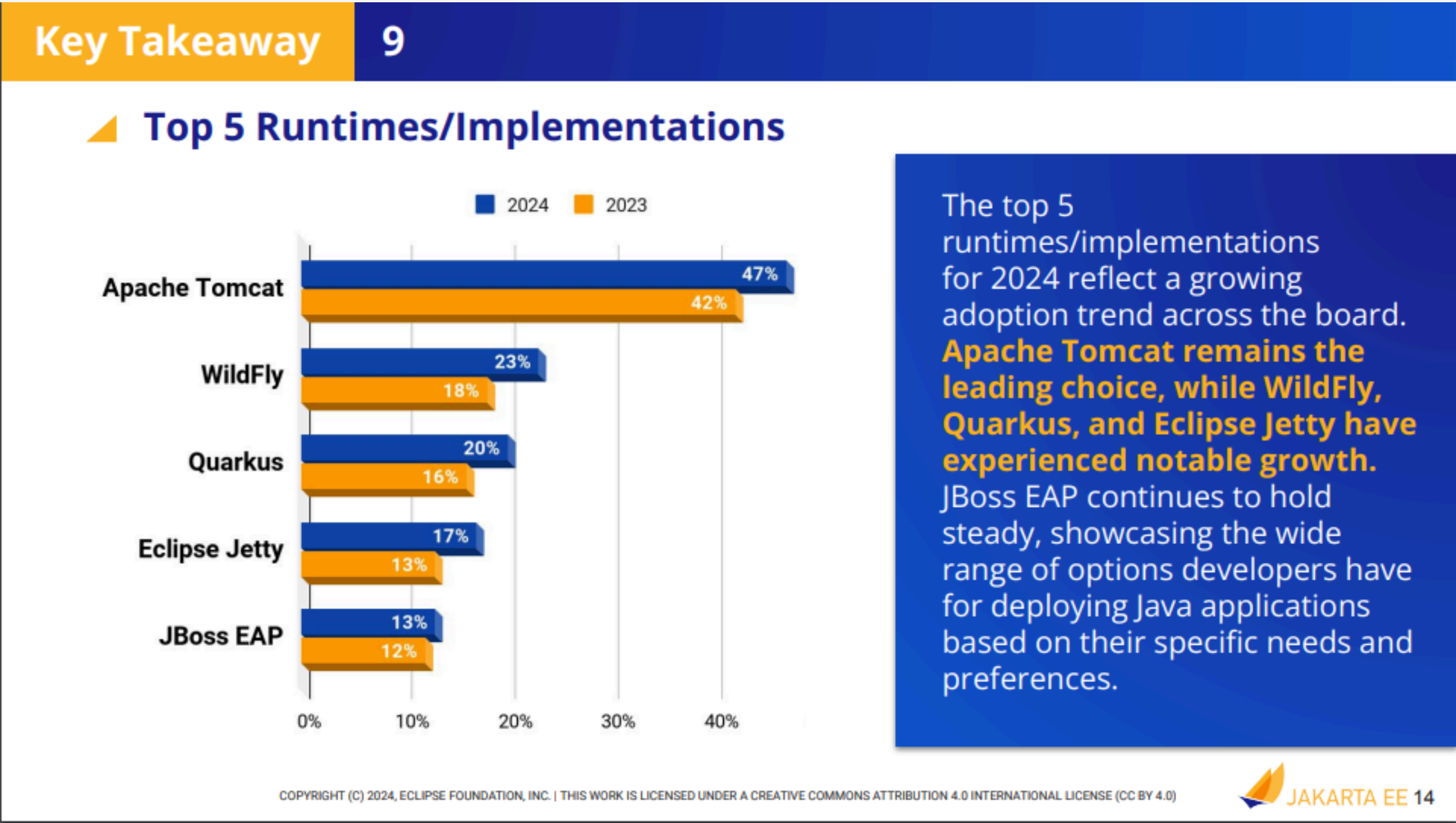
10

COPYRIGHT (C) 2024, ECLIPSE FOUNDATION, INC. | THIS WORK IS LICENSED UNDER A CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE (CC BY 4.0)



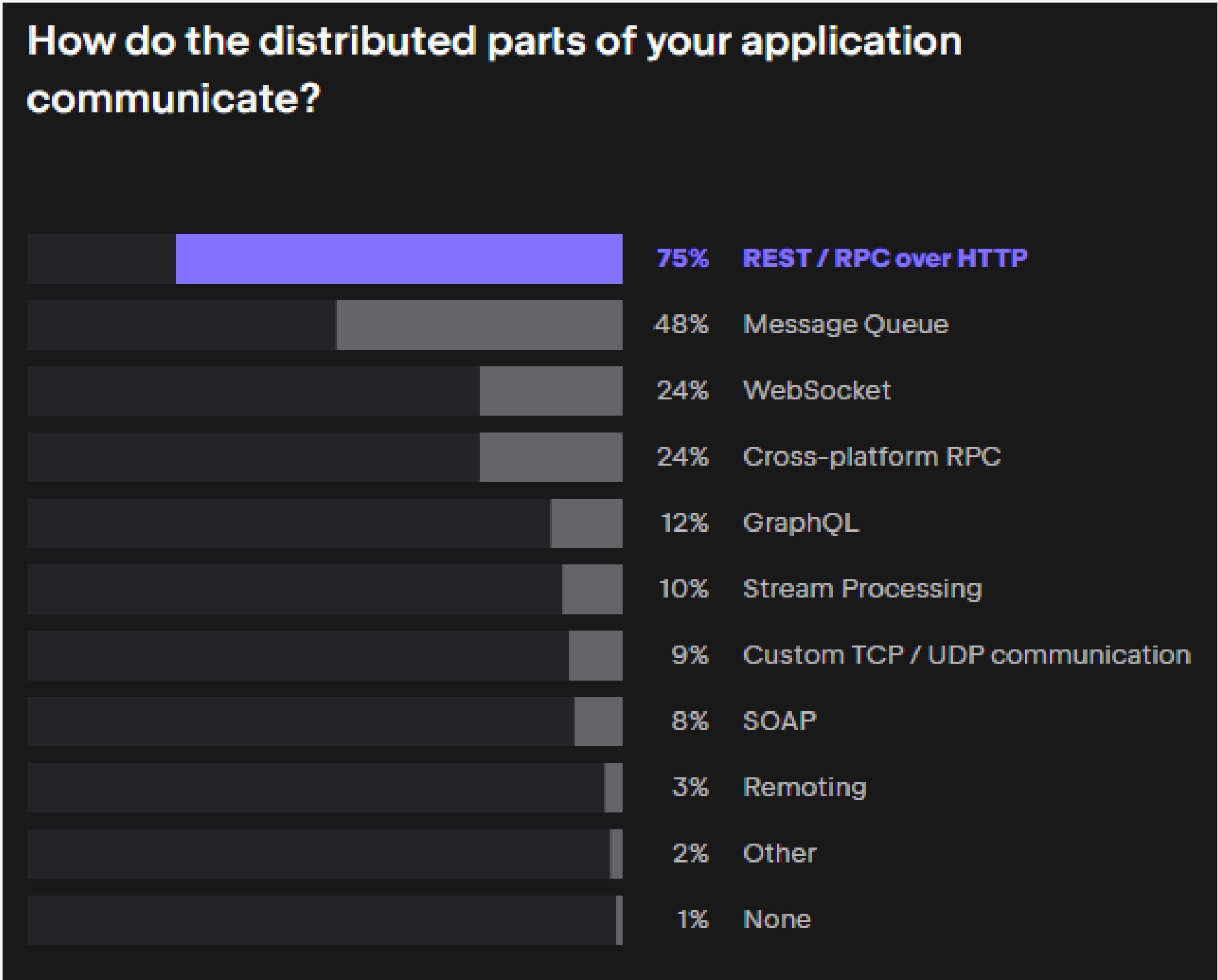
GlassFish

Hoy en día, su relevancia ha disminuido considerablemente y está siendo remplazado por otras tecnologías.



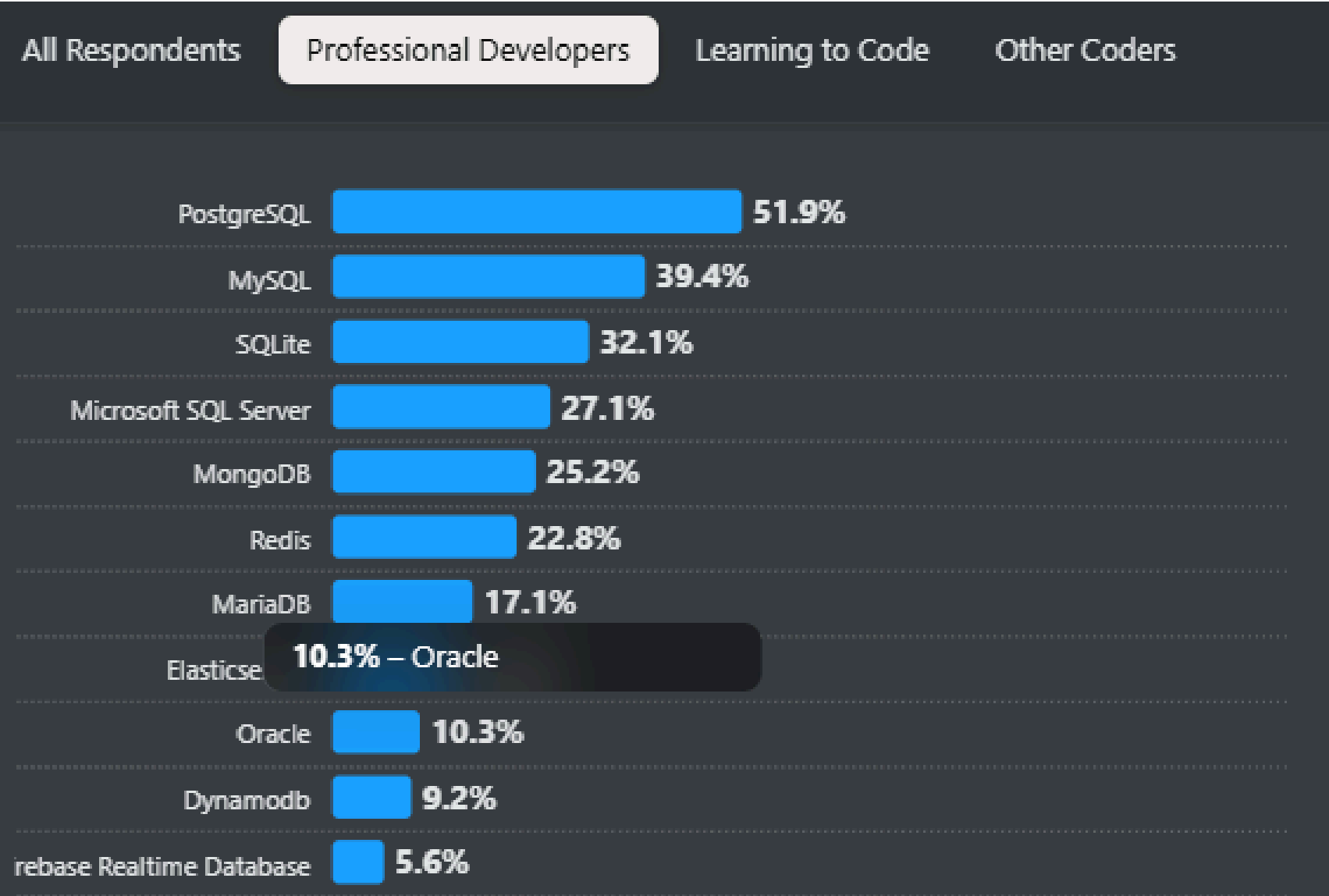
SOAP

Aunque fue el estándar dominante para servicios web empresariales durante muchos años, ha quedado relegado en la actualidad.



Oracle DB

Continúa siendo una opción válida en entornos empresariales que priorizan rendimiento, seguridad y disponibilidad. Sin embargo, su presencia en el ecosistema de desarrollo ha disminuido



Which databases have you used in the last 12 months, if any?

	2019	2020	2021	2022	2023	2024	
	60%	59%	61%	52%	51%	52%	MySQL
	32%	35%	36%	38%	38%	45%	PostgreSQL
	30%	32%	28%	27%	27%	30%	MongoDB
	29%	27%	29%	28%	25%	30%	SQLite
	27%	25%	29%	27%	26%	29%	Redis
	22%	20%	19%	18%	18%	20%	Microsoft SQL Server
	21%	19%	23%	18%	16%	16%	MariaDB
	—	—	—	—	—	13%	Elasticsearch
	16%	14%	13%	11%	13%	12%	Oracle Database
	—	—	—	—	—	10%	Amazon DynamoDB

Matriz de análisis de Principios SOLID vs Temas

Tecnología	SRP (S) – Responsabilidad única	OCP (O) – Abierto/cerrado	LSP (L) – Sustitución de Liskov	ISP (I) – Segregación de interfaces	DIP (D) – Inversión de dependencias
SOA	Cada servicio cubre una función clara e independiente.	Nuevas funcionalidades pueden añadirse como nuevos servicios.	Las interfaces expuestas pueden extenderse sin afectar clientes.	Servicios especializados con contratos (WSDL) bien definidos.	Implementaciones concretas ocultas detrás de contratos expuestos.
Svelte	Cada componente representa una unidad visual específica.	Reutilización de componentes sin modificar su núcleo.	Props y eventos se adaptan sin romper integraciones existentes.	Componentes manejan solo la UI necesaria, sin funciones extra.	Usa contextos y stores para desacoplar lógica de presentación.
Jakarta EE	Separación por capas: servicios, controladores, repositorios.	Soporta decoradores y extensiones sin modificar lógica base.	Interfaces y beans son sustituibles entre implementaciones.	Interfaces como JAX-RS o JPA siguen una responsabilidad clara.	CDI facilita la inversión de control entre capas.
GlassFish	Organiza módulos y servicios de forma separada.	Permite múltiples apps independientes en un mismo servidor.	Aplicaciones pueden ser sustituidas si respetan los contratos.	Cada módulo o EAR tiene sus propias funciones claramente aisladas.	Integración nativa con CDI para desacoplar capas.
SOAP	Cada operación de servicio tiene una finalidad específica.	Nuevas operaciones se agregan sin romper las existentes.	Contratos WSDL aseguran compatibilidad al evolucionar servicios.	Operaciones bien divididas según responsabilidades funcionales.	Clientes consumen servicios sin conocer su lógica interna.
Oracle DB	Tablas reflejan entidades concretas (clientes, citas, etc.).	Diseño permite escalar sin modificar el esquema base.	Vistas e índices reemplazables sin afectar el consumo.	Consultas especializadas por entidad, sin sobrecarga innecesaria.	Acceso abstracto a datos mediante JDBC/JPA, sin acoplar a SQL directo.

Matriz de análisis de Atributos de Calidad vs Temas

Tecnología	AF – Funcionalidad	ED – Rendimiento	CO – Compatibilidad	US – Usabilidad	FI – Fiabilidad	SE – Seguridad	MA – Mantenibilidad	PO – Portabilidad
SOA	Alta – Servicios bien definidos y enfocados.	Media – Puede haber sobrecarga por SOAP.	Alta – Diseñado para interoperar entre sistemas.	N/A – Es tecnología de backend.	Alta – Cada servicio es independiente y tolerante a fallos.	Alta – Compatible con WS-Security y estándares.	Alta – Arquitectura modular facilita ajustes.	Alta – Fácil de desplegar en múltiples entornos.
Svelte	Alta – Interfaces reactivas y enfocadas en funcionalidad.	Alta – Genera JS muy eficiente.	Media – Requiere adaptaciones para consumir SOAP u otros sistemas.	Alta – Experiencia de usuario moderna y fluida.	Alta – Los errores suelen estar acotados a componentes.	Media – No es enfoque principal, pero puede integrarse.	Alta – Componentes simples y fáciles de modificar.	Alta – Ideal para apps SPA multiplataforma.
Jakarta EE	Alta – Separación clara en capas y servicios.	Media – Algo más pesado que soluciones modernas.	Alta – Soporte robusto para APIs y protocolos.	N/A – Más enfocado en backend empresarial.	Alta – Fiabilidad respaldada por años de uso en producción.	Alta – Incluye JAAS, SSL y otras medidas.	Alta – Basado en estándares y prácticas sólidas.	Media – Portabilidad buena, pero requiere configuración.
GlassFish	Alta – Compatible con todo Jakarta EE.	Media-baja – No es ideal para entornos exigentes.	Alta – Compatible con múltiples estándares.	N/A	Media – Su estabilidad depende de una buena configuración.	Alta – Compatible con WS-Security y SSL.	Media – Menor comunidad y soporte que otros servidores.	Baja – Portabilidad limitada en comparación con alternativas más modernas.
SOAP	Alta – Contratos bien definidos aseguran precisión.	Media-baja – Verboso y menos eficiente que REST.	Alta – Amplio soporte en plataformas y lenguajes.	N/A	Alta – Transmisión estructurada confiable.	Alta – Soporte de seguridad robusto (WS-Security, XML Signature).	Alta – Servicios bien segmentados.	Alta – Puede integrarse fácilmente en distintos sistemas.
Oracle DB	Alta – Fuerte cumplimiento de reglas y ACID.	Alta – Excelente rendimiento en procesamiento de datos.	Media – Requiere drivers específicos según el entorno.	N/A	Alta – Resistencia probada en entornos críticos.	Alta – Gestión avanzada de roles, privilegios y auditoría.	Media – Mantenimiento técnico puede ser complejo.	Media – Migraciones pueden ser costosas o difíciles.

Matriz de análisis de Tácticas vs Temas

Tecnología	RA - Reducción del acoplamiento	SR - Separación de responsabilidades	MD - Monitoreo y diagnósticos	GE- Gestión de Errores	CA - Control de Acceso	BE - Balanceo de Carga y Escalabilidad	TF - Tolerancia a Fallos	IO - Interoperabilidad
SOA	Alta (servicios independientes)	Alta (servicios modulares por función)	Media (requiere herramientas externas)	Alta (gestión de errores por servicio)	Alta (seguridad a nivel de servicio)	Media (a través de orquestación)	Alta (si falla un servicio, no cae todo)	Alta
Svelte	Alta (componentes desacoplados)	Alta (UI por módulos)	Media (requiere integración con herramientas de log)	Alta (catch de errores UI)	Baja (seguridad delegada al backend)	Media (SPA optimiza carga)	Media	Media
Jakarta EE	Alta (inyección de dependencias CDI)	Alta (beans separados por capas)	Alta (APIs de logging y auditoría)	Alta (control por excepción)	Alta (JAAS, roles, filtros)	Alta (clusterización posible)	Alta (soporte a múltiples instancias)	Alta
GlassFish	Media (configuración modular)	Alta (soporta módulos WAR)	Media (consola de administración básica)	Media (requiere configuración)	Alta (configuración de seguridad)	Media-baja (no recomendado para producción masiva)	Media	Alta
SOAP	Alta (cada operación es independiente)	Alta (WSDL por funcionalidad)	Baja (complejo de monitorear)	Alta (manejo de errores estándar)	Alta (WS-Security)	Baja	Alta (respuesta predecible)	Alta
Oracle DB	Alta (modelo entidad-relación claro)	Alta (consultas por entidad)	Alta (auditorías, triggers)	Alta (integridad referencial, transacciones)	Alta (roles, privilegios)	Alta (RAC, particionamiento)	Alta (backup, recuperación)	Media (requiere drivers específicos)

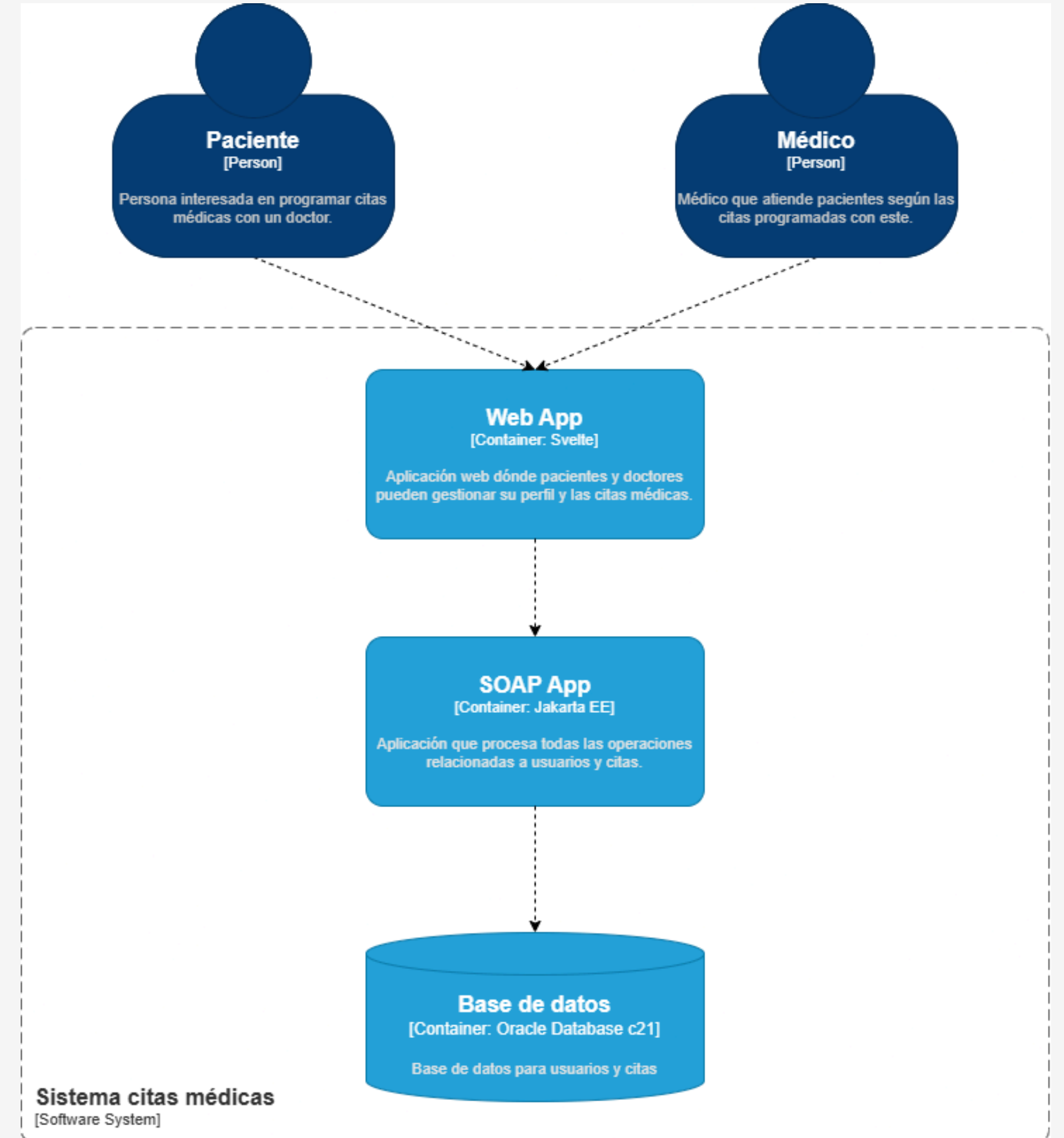
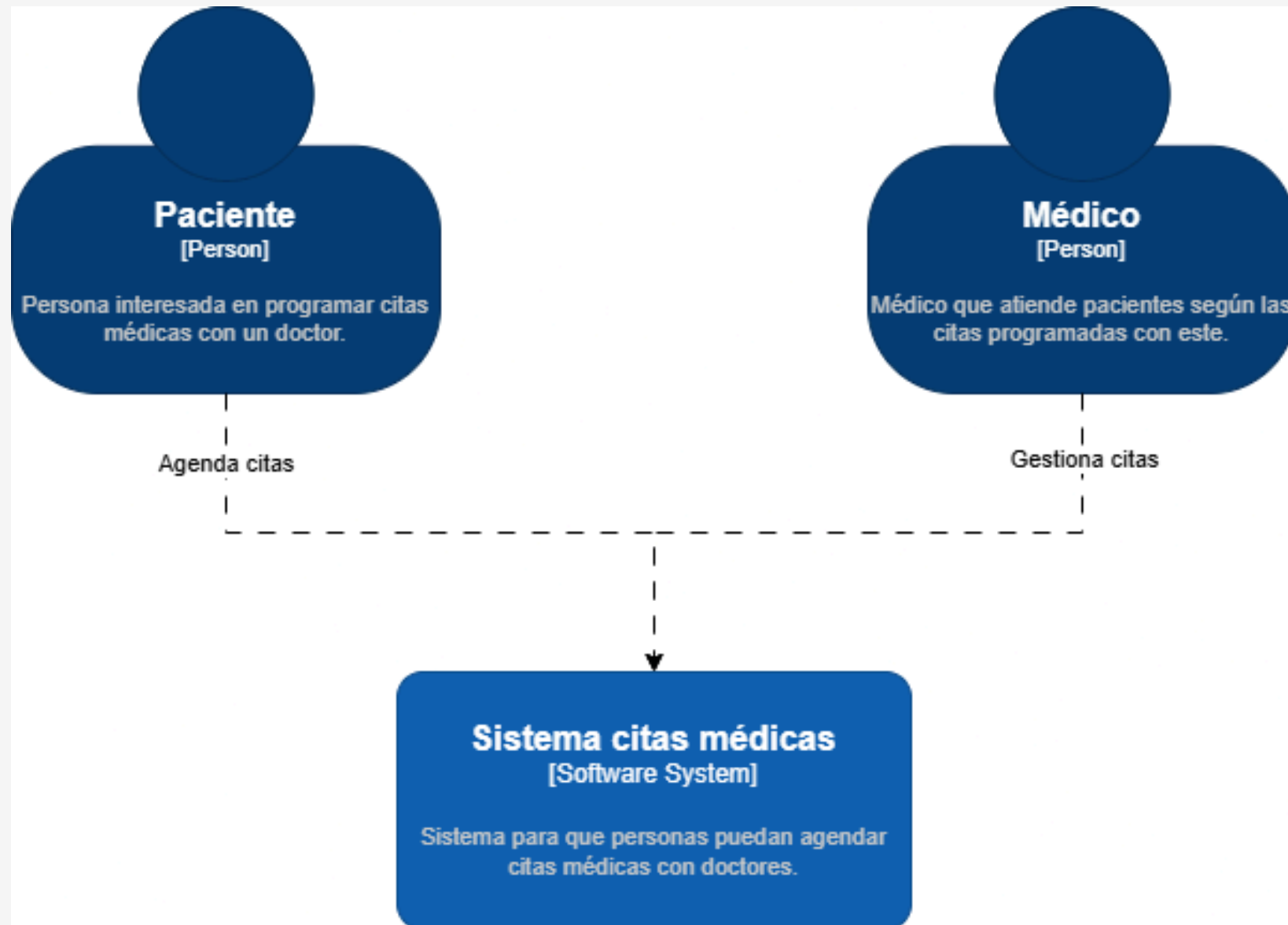
Matriz de análisis de Patrones vs Temas

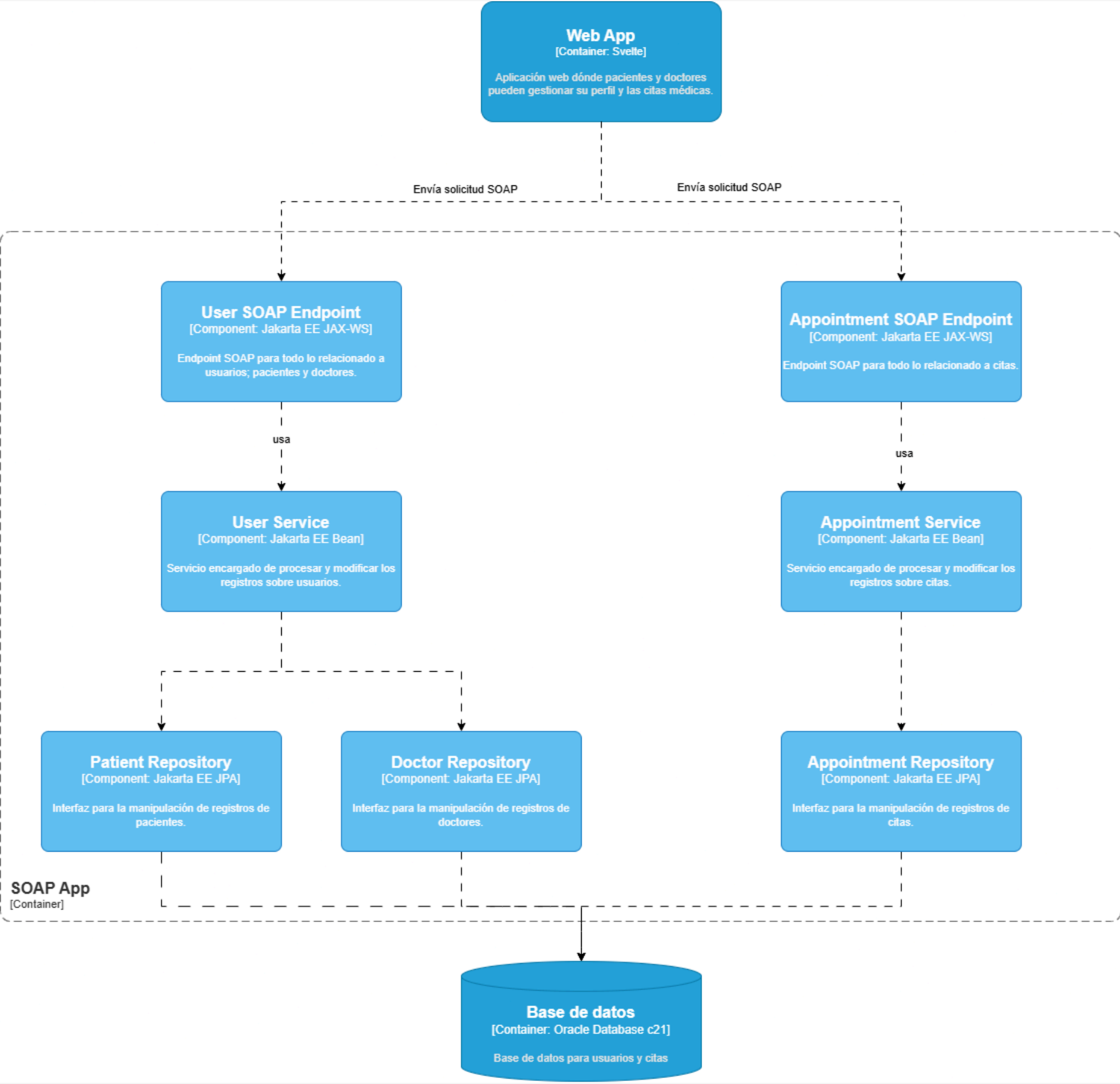
Tecnología	MVC (Modelo-Vista-Controlador)	DAO (Data Access Object)	DTO (Data Transfer Object)	Service Layer	Proxy o Gateway	Observer/Reactive UI	Facade	Orquestación de Servicios
SOA	N/A	N/A	Alta (envío de datos estructurados)	Alta (cada servicio actúa como capa intermedia)	Alta (a través de WSDL y bindings)	N/A	Alta (servicios simplifican procesos)	Alta (orquestación de múltiples servicios)
Svelte	Alta (estructura UI clara)	N/A	Media (usa objetos de datos desde el backend)	N/A	Media (puede implementar fetch wrappers)	Alta (reactividad, bindings)	N/A	N/A
Jakarta EE	Alta (Controladores, Servicios, DAOs)	Alta (repositorios JPA)	Alta (uso de DTOs entre capas)	Alta (Servicios entre endpoints y lógica)	Media (endpoints SOAP actúan como gateway)	N/A	Alta (Beans centralizan lógica)	Media (posible con EJB y Jakarta Batch)
GlassFish	Alta (facilita despliegue de MVC en Jakarta EE)	Alta	Alta	Alta	Media	N/A	Media	Media
SOAP	N/A	N/A	Alta (estructura XML definida)	Alta (cada operación expone una función)	Alta (definición formal de interfaces)	N/A	Alta (expone solo lo necesario al consumidor)	Alta
Oracle DB	N/A	Alta (acceso a datos con DAOs)	Media (requiere mapeo externo)	N/A	N/A	N/A	N/A	N/A

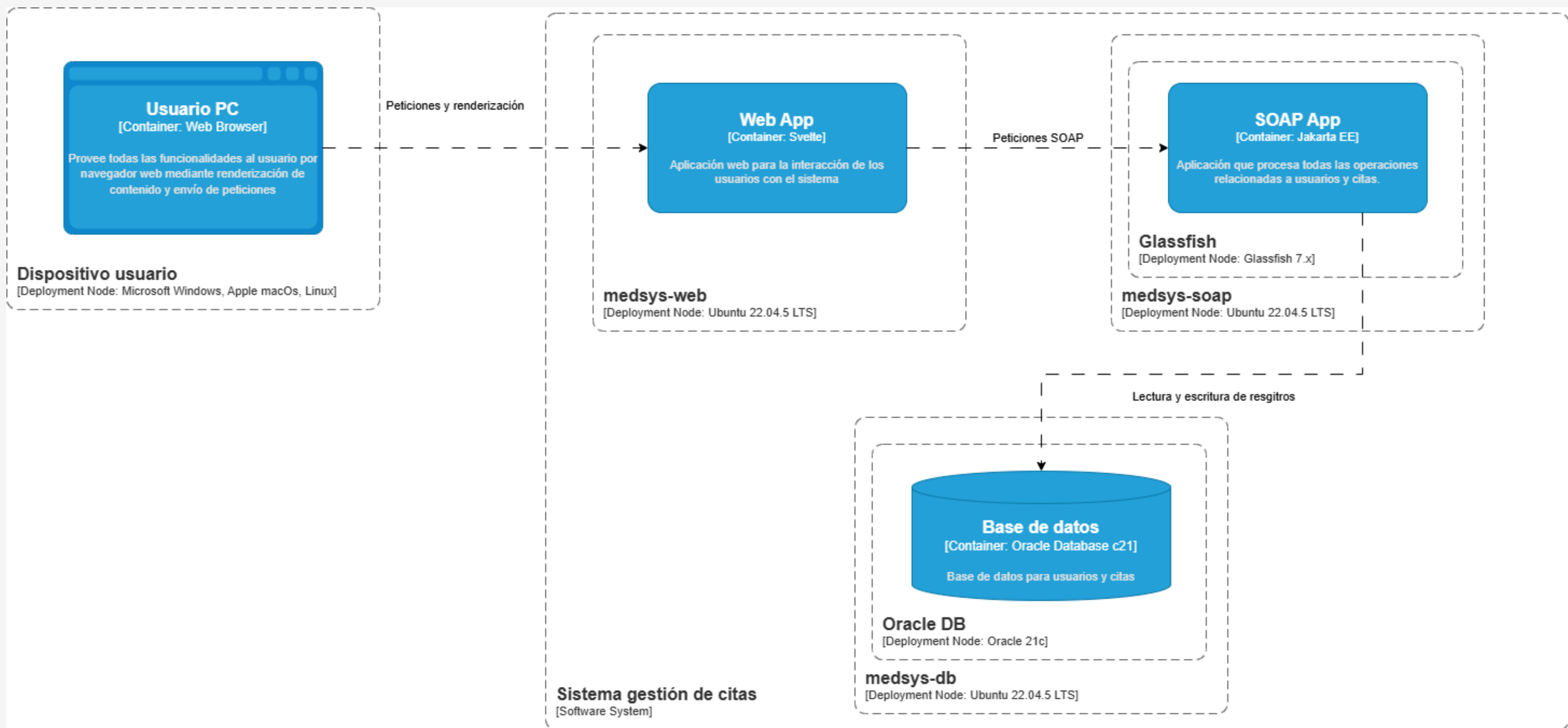
Matriz de análisis de Mercado Laboral vs Temas

Tecnología	Demanda laboral	Adopción en la industria	Madurez	Proyección de crecimiento	Comunidad / Soporte
SOA	Media (presente en grandes empresas, especialmente legacy)	Media-Alta (entornos bancarios, salud, gobierno)	Alta (consolidada)	Baja (reemplazada en muchos casos por microservicios)	Media (documentación formal, pero en declive activo)
Svelte	Media-Baja (nicho creciente, aún no tan demandado como React)	Baja-Media (usado en startups o proyectos personales)	Media (aún en expansión)	Alta (rápido crecimiento y satisfacción de desarrolladores)	Alta (comunidad muy activa y entusiasta)
Jakarta EE	Alta (muy usado en empresas grandes que usan Java)	Alta (corporativo, financiero, gubernamental)	Alta (evolución de Java EE)	Media (se mantiene vigente, pero no disruptiva)	Alta (respaldo de Eclipse Foundation)
GlassFish	Baja (no recomendado para producción moderna)	Baja (usado en educación y pruebas)	Media (estable pero superado por otros)	Baja (reemplazado por Payara, WildFly, etc.)	Media (existe documentación, pero comunidad limitada)
SOAP	Media (activo en industrias reguladas)	Alta (sector salud, banca, aseguradoras)	Alta (protocolo robusto y estandarizado)	Baja (en desuso frente a REST/GraphQL)	Alta (soporte continuo en plataformas empresariales)
Oracle DB	Alta (presente en grandes corporativos, gobiernos)	Alta (infraestructura crítica, ERP, CRM)	Alta (consolidada desde hace décadas)	Media (mantiene cuota pero compite con PostgreSQL)	Alta (documentación, soporte empresarial, certificaciones)

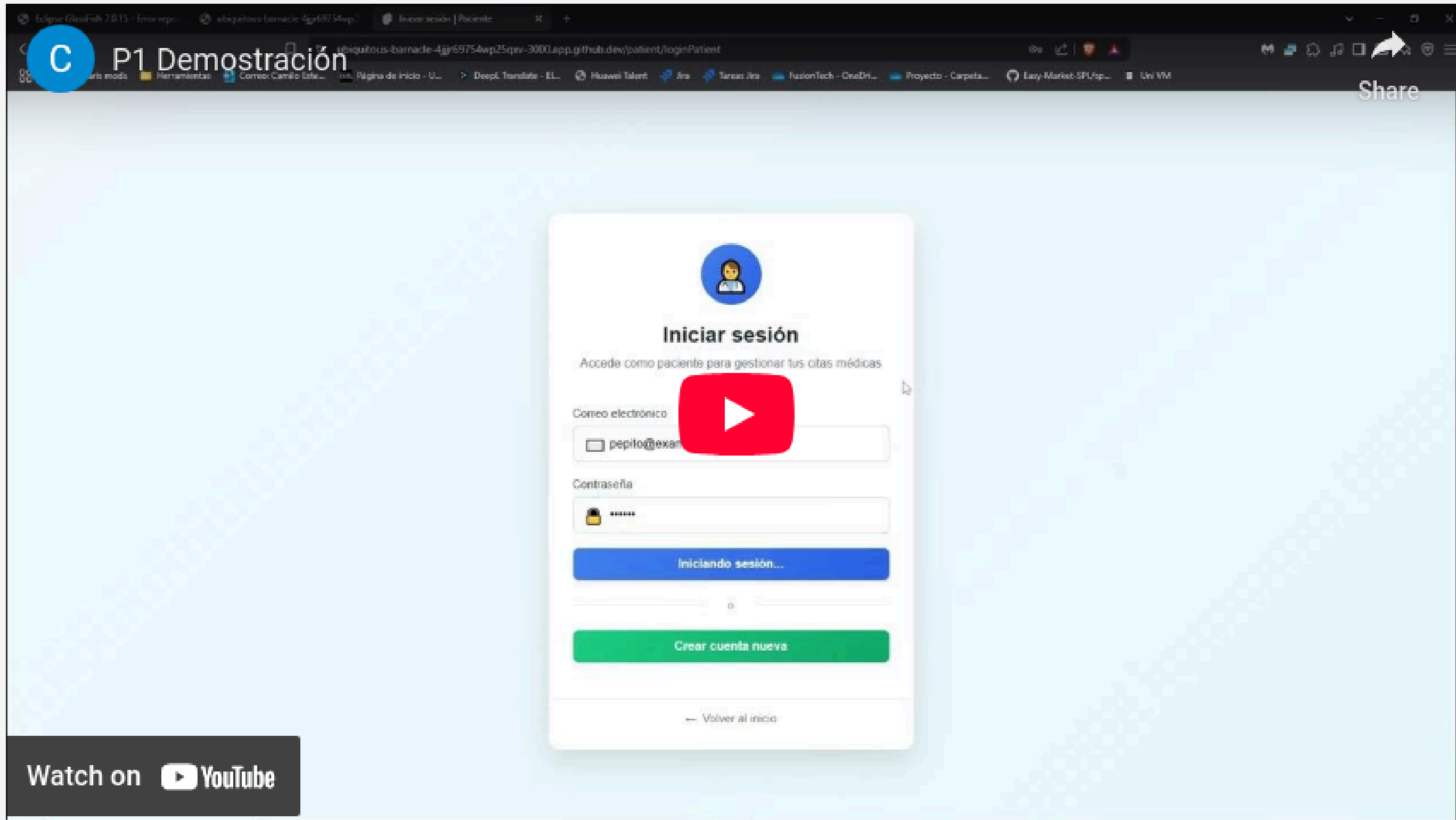
Diagramas







Demostración



GRACIAS

