

GRUPO #10 CAMILO RODRIGUEZ TOMAS MARTINEZ DANIEL MILLAN NUESTRO CONJUNTO DE CONCEPTOS, PATRONES, ESTILOS, PRINCIPIOS, TECNOLOGÍAS Y FRAMEWORKS:

- ·microservicios
- ·front:svelt
- ·int:REST/XmI
- ·back:GO

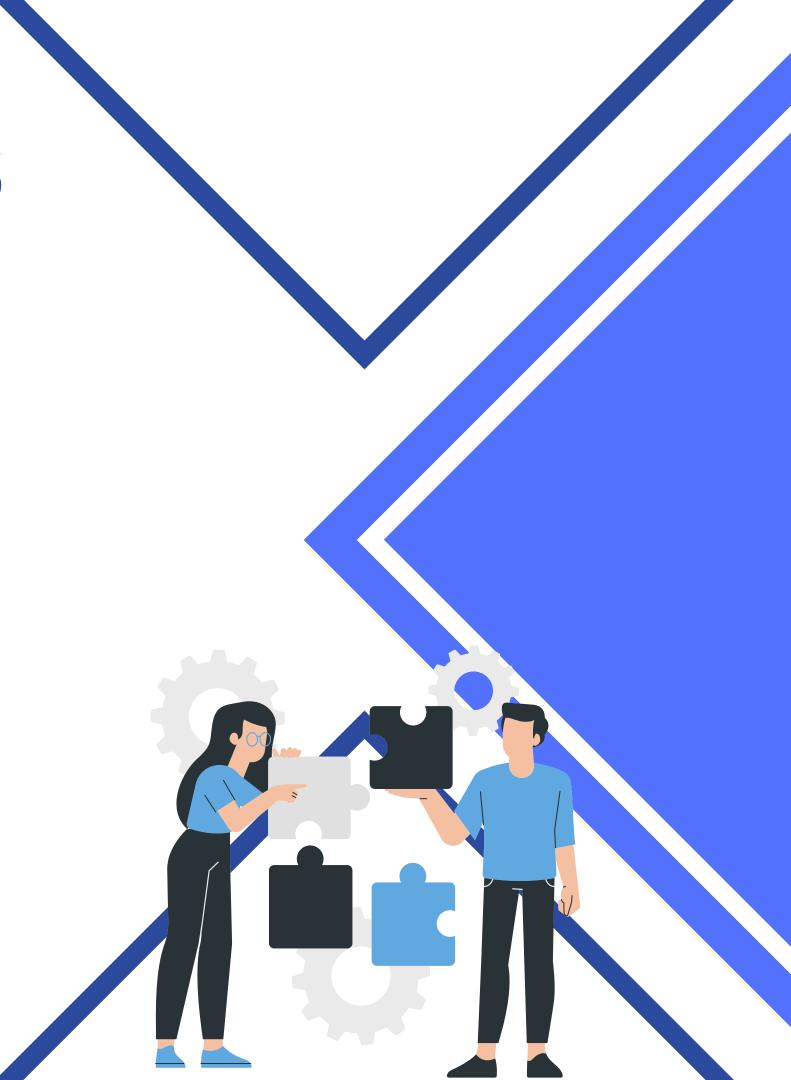
BD:mariaDB

DEFINICIÓN, HISTORIA Y EVOLUCIÓN

MICROSERVICIOS

LOS MICROSERVICIOS **SON UN PATRÓN ARQUITECTÓNICO** EN EL QUE UNA APLICACIÓN **SE DIVIDE EN COMPONENTES PEQUEÑOS E INDEPENDIENTES**, LLAMADOS MICROSERVICIOS. CADA MICROSERVICIO CUMPLE UNA FUNCIÓN ESPECÍFICA Y SE COMUNICA CON OTROS A TRAVÉS DE API. ALGUNAS CARACTERÍSTICAS CLAVE DE LOS MICROSERVICIOS SON:

- •INDEPENDENCIA: CADA MICROSERVICIO SE PUEDE DESARROLLAR, IMPLEMENTAR Y OPERAR DE FORMA INDEPENDIENTE SIN AFECTAR A OTROS SERVICIOS.
- •MANTENIBILIDAD Y PRUEBAS: FACILITAN LA EXPERIMENTACIÓN CON NUEVAS FUNCIONES Y PERMITEN REVERTIRLAS SI NO FUNCIONAN. TAMBIÉN SIMPLIFICAN EL AISLAMIENTO Y LA CORRECCIÓN DE ERRORES EN SERVICIOS INDIVIDUALES.
- • ORGANIZACIÓN EN TORNO A CAPACIDADES EMPRESARIALES: LOS SERVICIOS SE AGRUPAN SEGÚN FUNCIONALIDADES ESPECÍFICAS.
- ·INFRAESTRUCTURA AUTOMATIZADA: SE UTILIZAN PRÁCTICAS DE AUTOMATIZACIÓN DE INFRAESTRUCTURAS COMO CI/CD.



MICROSERVICIOS

HISTORIA Y EVOLUCIÓN:

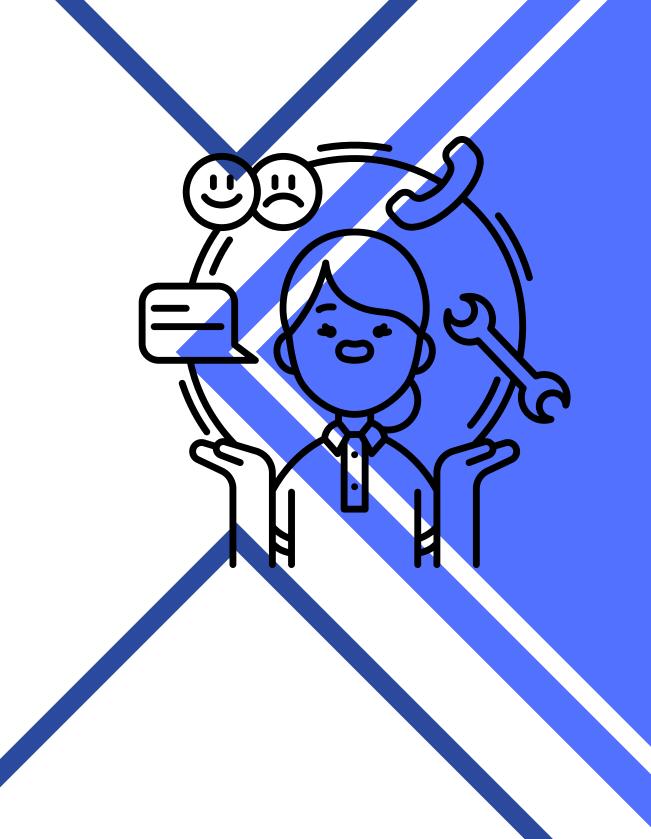
AUNQUE EL TÉRMINO "MICROSERVICIOS" SE POPULARIZÓ EN LA ÚLTIMA DÉCADA, LOS PRINCIPIOS SUBYACENTES HAN EXISTIDO DURANTE MUCHO TIEMPO.

EN LA DÉCADA DE 1970, SE PROMOVÍA LA MODULARIDAD Y LA SEPARACIÓN DE RESPONSABILIDADES EN EL DISEÑO DE SISTEMAS.

EN LA DÉCADA DE 2000, SURGIERON LAS ARQUITECTURAS ORIENTADAS A SERVICIOS (SOA), QUE COMPARTÍAN SIMILITUDES CON LOS MICROSERVICIOS.

SIN EMBARGO, SOA A MENUDO SE ASOCIABA CON SERVICIOS PESADOS Y ACOPLADOS.

LOS MICROSERVICIOS MODERNOS SE POPULARIZARON CON EL AUGE DE LA COMPUTACIÓN EN LA NUBE Y LA NECESIDAD DE ESCALABILIDAD Y MANTENIBILIDAD EN APLICACIONES WEB Y MÓVILES.



MICROSERVICES

BEFORE

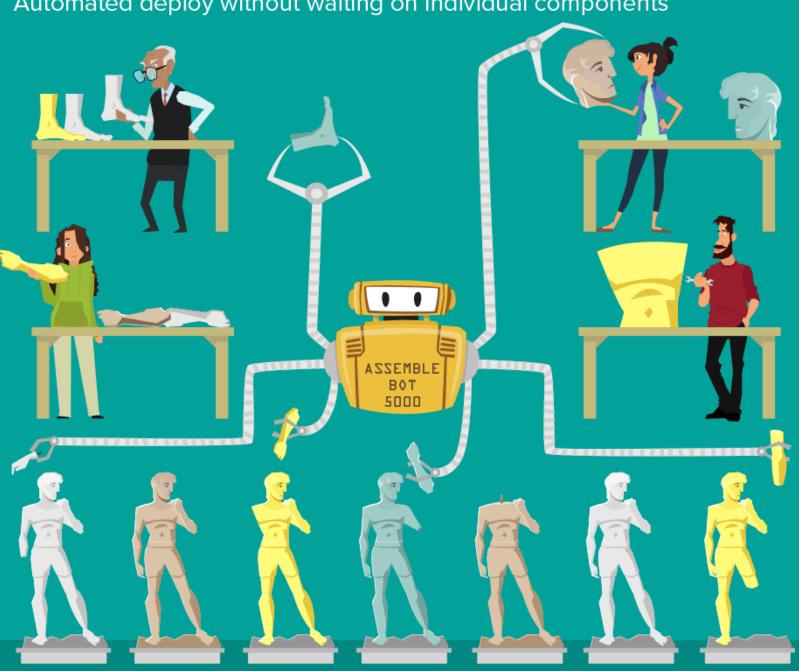
Tightly coupled components
Slow deploy cycles waiting on integrated tests and teams

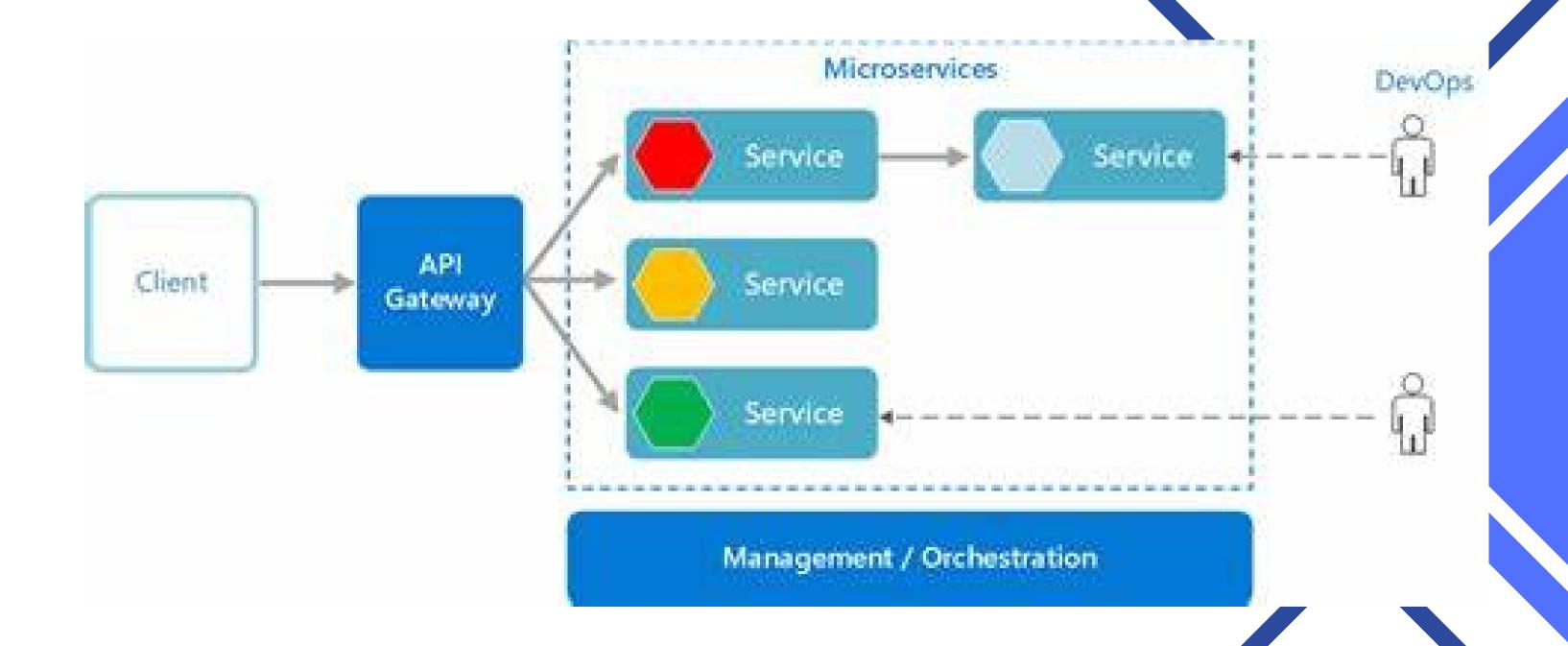


AFTER

Loosely coupled components

Automated deploy without waiting on individual components





SVELTE

SVELTE ES **UN FRAMEWORK DE DESARROLLO WEB** QUE COMPILA A **JAVASCRIPT** PURO EN TIEMPO DE COMPILACIÓN. PERMITE CREAR APLICACIONES WEB R**EACTIVAS Y EFICIENTES**.

HISTORIA Y EVOLUCIÓN:

SVELTE FUE CREADO POR RICH HARRIS Y SE LANZÓ EN 2016.
A DIFERENCIA DE OTROS FRAMEWORKS COMO REACT O VUE,
SVELTE NO SE EJECUTA EN TIEMPO DE EJECUCIÓN. EN SU LUGAR,
COMPILA EL CÓDIGO EN JAVASCRIPT OPTIMIZADO.
SU ENFOQUE EN LA EFICIENCIA Y LA SIMPLICIDAD LO HA
CONVERTIDO EN UNA OPCIÓN POPULAR PARA
DESARROLLADORES WEB.



SVELTE

Svelte Component File Structure

React	Vue	Svelte	
JavaScript	HTML	JavaScript	
HTML			
JavaScript	JavaScript	CSS	
HTML			
JavaScript	CSS	HTML	
.jsx	.vue	.svelte	







REST/XML

REST (REPRESENTATIONAL STATE TRANSFER) ES **UN ESTILO ARQUITECTÓNICO PARA SISTEMAS DISTRIBUIDOS** BASADO EN HTTP. SE CENTRA EN LA TRANSMISIÓN DE DATOS A TRAVÉS DE API Y EN REPRESENTACIONES DE RECURSOS.

XML (EXTENSIBLE MARKUP LANGUAGE): ES UN FORMATO DE MARCADO UTILIZADO PARA ESTRUCTURAR DATOS.

HISTORIA Y EVOLUCIÓN:

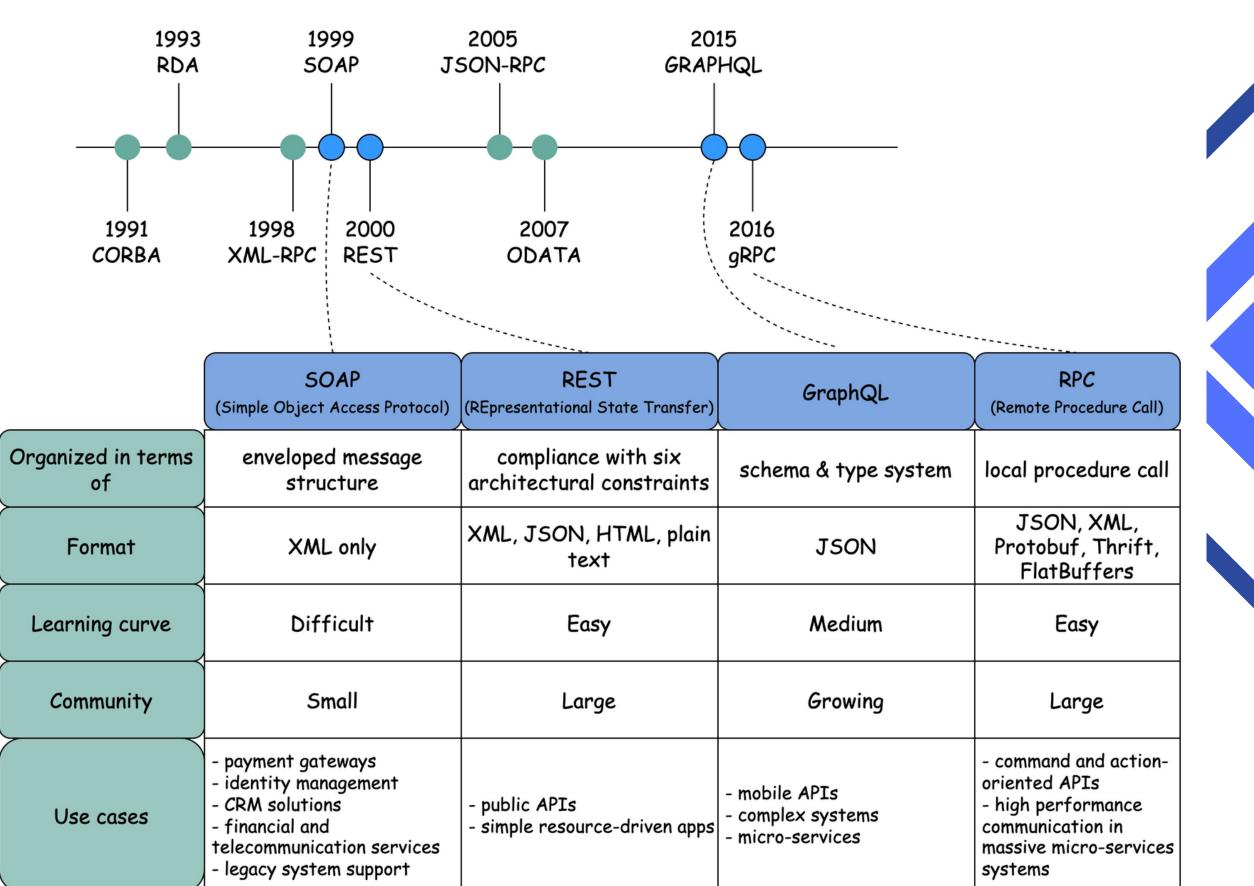
REST SE POPULARIZÓ EN LA DÉCADA DE 2000 COMO UNA ALTERNATIVA A LOS SERVICIOS WEB SOAP.

XML HA SIDO AMPLIAMENTE UTILIZADO PARA REPRESENTAR DATOS ESTRUCTURADOS EN APLICACIONES WEB Y SERVICIOS.



API Architectural Styles Comparison

Source: altexsoft



GO (GOLANG)

GO ES UN **LENGUAJE DE PROGRAMACIÓN** DE CÓDIGO ABIERTO CREADO POR GOOGLE. ES CONOCIDO POR SU RENDIMIENTO, EFICIENCIA Y FACILIDAD DE USO.

HISTORIA Y EVOLUCIÓN:

GO FUE ANUNCIADO EN 2009 Y SE LANZÓ OFICIALMENTE EN 2012.

DISEÑADO PARA SER SIMPLE, RÁPIDO Y SEGURO, GO SE HA CONVERTIDO EN UNA OPCIÓN POPULAR PARA EL DESARROLLO BACKEND Y LA CREACIÓN DE MICROSERVICIOS.





High

VS

Low

051		
May 27, 2009	INTRODUCED IN	Nov 10, 2009
Ryan Dahl	DEVELOPED BY	Robert Griesemer, Rob Pike, Ken Thompson
14.18.0	CURRENT VERSION	1.17.1
C, C++ and Javascript	PROGRAMMING LANGUAGES	Go, Assembly Language and C++
Windows, IBM, AIX, Linux, MacOS, SmartOS, etc.	SUPPORTING OS	Windows, DragonFly BSD, Linux, MacOS, Net BSD
Easy	LEARNING CURVE	Steep
Medium	PERFORMANCE	High
Medium	SCALABILITY	Easy
High	AVAILABILITY OF	Low

DEVELOPERS

Advantages of Using Go for Your Next Project



Simplified code

Powerful performance

Suitable for large projects

Designed for multicore processors

Internet friendly

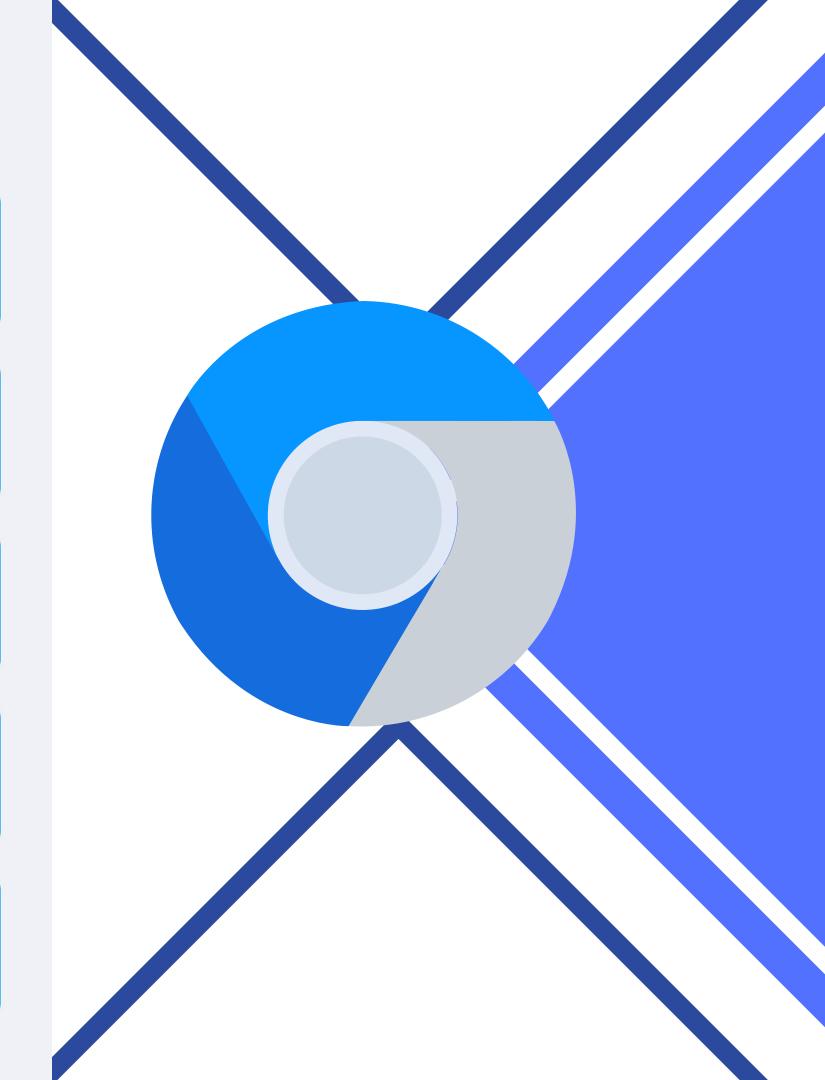
Fast garbage collection

Easy to learn

Easy maintenance

Open Source approach

Industry agnostic



MARIADB

MARIADB ES UNA **BASE DE DATOS RELACIONAL** DE CÓDIGO ABIERTO. ES UNA BIFURCACIÓN DE MYSQL Y OFRECE CARACTERÍSTICAS ADICIONALES Y MEJORAS.

HISTORIA Y EVOLUCIÓN:

MARIADB SE ORIGINÓ EN 2009 COMO UNA RESPUESTA A LAS PREOCUPACIONES SOBRE LA ADQUISICIÓN DE MYSQL POR PARTE DE ORACLE.

MARIADB MANTIENE LA COMPATIBILIDAD CON MYSQL Y HA SEGUIDO EVOLUCIONANDO CON NUEVAS CARACTERÍSTICAS Y MEJORAS.



What is MariaDB?

Open-source fork of MySQL RDBMS

Released in 2009

Very compatible with MySQL



2009



Supports same features that MySQL does, but with additional features as well

Some new features include: new storage engines, JSON API support & parallel data replication





WWW.TEMOK.COM

RELACIÓN ENTRE LOS COMPONENTES,

Microservicios y Svelte:

·La comunicación entre el frontend (Svelte)

y los microservicios se realiza a través de

API REST.

Microservicios y REST/XML:

·XML puede ser utilizado como formato de representación de datos en las respuestas de los microservicios.

- Microservicios y Go (Golang):
- ·Go proporciona bibliotecas estándar para crear servidores HTTP y gestionar rutas y controladores.
 - .Microservicios y MariaDB:
- ·Los microservicios se comunican con la base de datos a través de consultas SQL.

SITUACIONES Y PROBLEMAS

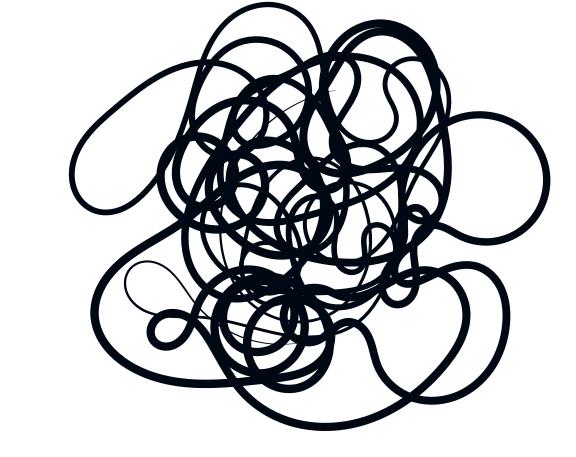
Escalabilidad y Mantenibilidad:

1.Microservicios:

- ·La gestión de múltiples microservicios requiere una planificación cuidadosa para evitar la complejidad.
- ·El acoplamiento entre microservicios debe manejarse con atención para evitar dependencias innecesarias.

2.Svelte:

- ·La escalabilidad de las interfaces de usuario creadas con Svelte debe considerar la optimización del rendimiento.
- ·El mantenimiento de componentes Svelte puede volverse complicado a medida que crece la aplicación. 3.Go (Golang):
- ·Aunque Go es eficiente en términos de recursos, se debe considerar la gestión de dependencias y la concurrencia.
- ·La concurrencia en Go puede ser desafiante, especialmente al escalar aplicaciones.





Integración de Sistemas Externos:

4.REST/XML:

- ·La integración de sistemas externos mediante REST/XML requiere una buena documentación y manejo de versiones.
- ·Cambios inesperados en las APIs externas pueden afectar la funcionalidad de los microservicios.

5. Microservicios:

- ·La dependencia de APIs externas puede afectar la disponibilidad y el rendimiento.
- ·Las pruebas exhaustivas de integración son esenciales para garantizar una comunicación fluida.

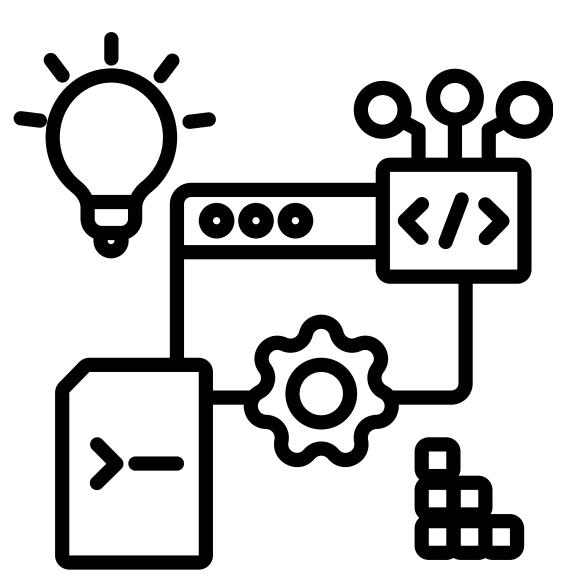
Almacenamiento de Datos:

6.MariaDB:

- ·Se debe optimizar adecuadamente la base de datos para evitar problemas de rendimiento.
- ·La falta de índices o consultas ineficientes puede afectar el rendimiento de los microservicios.

7. Microservicios:

·La sincronización de datos entre microservicios debe considerar transacciones atómicas y consistencia.



Ventajas / Desventajas

MICROSERVICIOS

Ventajas:

Acelera la escalabilidad. Mejora el aislamiento de los fallos. Costes más eficientes.

Desventajas:

Complejidad en la comunicación y gestión. Mayor consumo de memoria.

SVELTE

Ventajas:

Sencillo desde la perspectiva del programador. Compilación rápida de componentes.

Desventajas:

No tiene tantas características como otros frameworks. La interfaz de usuario no es tan intuitiva como otras bases de datos.

REST/XML

Ventajas:

Facilidad de uso. Flexibilidad.

Desventajas:

Puede ser difícil comprender y configurar para los principiantes.



Ventajas/ Desventajas

GO

Ventajas:

- Es un lenguaje concurrente que soporta canales de comunicación CSP.
- Cuenta con un recolector de basura que permite elevar al máximo la eficiencia y el rendimiento, y reducir al mínimo la latencia.

Desventajas:

No se encontraron desventajas específicas en las fuentes consultadas.

MARIADB

Ventajas:

MariaDB es más seguro que MySQL. Ofrece mejor rendimiento y escalabilidad. Al igual que MySQL, MariaDB es de software libre.

Desventajas:

MariaDB no tiene tantas características como MySQL. La interfaz de usuario no es tan intuitiva como otras bases de datos.



Ventajas / Desventajas

1







PROS.

CONS'F

- Extremely Fast
- **Easy to Learn Syntax**
- Statically Typed
- Garbage Collection
- Testing Support
- Interface Types

- Implicit Interfaces
- Poor Library Support
- No Generics
- Zero Community Support
- Internal Inconsistencies
- Limited Automation Scope





PRINCIPIOS SOLID

SRP

MCS

SVI

R/X

GO

MDB

CADA MICROSERVICIO DEBE TENER UNA ÚNICA RESPONSABILIDAD.

Cada componente en Svelte debe tener una única responsabilidad

Cada endpoint en una API Rest debe tener una única responsabilidad.

Cada paquete o función en Go debe tener una única responsabilidad.

Cada tabla en MariaDB debe tener una única responsabilidad.

OCP

MCS

SVT

R/X

GO

MDB

LOS MICROSERVICIOS DEBEN ESTAR ABIERTOS PARA LA EXTENSIÓN, PERO CERRADOS PARA LA MODIFICACIÓN.

Svelte permite extender componentes sin necesidad de modificarlos

Las APIs Rest deben permitir la extensión (agregar nuevos endpoints) sin necesidad de modificar los existentes.

Go permite extender el comportamiento de las estructuras a través de la implementación de interfaces, sin necesidad de modificar las estructuras.

MariaDB permite extender las tablas (agregar nuevas columnas) sin necesidad de modificar las existentes.

LSP

MCS

SVT

R/X

GO

MDB

EN EL CONTEXTO DE LOS MICROSERVICIOS, ESTE PRINCIPIO SE PUEDE INTERPRETAR COMO QUE CADA MICROSERVICIO DEBE SER REEMPLAZABLE POR OTRO SIN AFECTAR EL FUNCIONAMIENTO DEL SISTEMA

Los componentes en Svelte deben ser intercambiables sin afectar el funcionamiento de la aplicación.

Los endpoints en una API Rest deben ser intercambiables sin afectar el funcionamiento de la aplicación.

Las estructuras en Go que implementan una interfaz deben ser intercambiables sin afectar el funcionamiento del programa.

Las tablas en MariaDB deben ser intercambiables sin afectar el funcionamiento de la aplicación.

ISP

MCS

SVT

R/X

GO

MDB

LOS MICROSERVICIOS DEBEN PROPORCIONAR INTERFACES SIMPLES Y ESPECÍFICAS PARA LOS CLIENTES.

Svelte promueve la creación de componentes con interfaces simples y específicas.

Las APIs Rest deben proporcionar interfaces simples y específicas para los clientes.

Go promueve la creación de interfaces pequeñas y específicas.

MariaDB promueve la creación de tablas con interfaces simples y específicas para los clientes

DIP

MCS

SVT

R/X

GO

MDB

LOS MICROSERVICIOS DEBEN DEPENDER DE ABSTRACCIONES, NO DE IMPLEMENTACIONES CONCRETAS

En Svelte, los componentes deben depender de abstracciones (como props y eventos), no de implementaciones concretas

En una API Rest, los endpoints deben depender de abstracciones, no de implementaciones concretas.

En Go, las estructuras y funciones deben depender de interfaces, no de implementaciones concretas.

En MariaDB, las tablas deben depender de abstracciones, no de implementaciones concretas

RIMBERIO.CO



ATRIBUTOS DE CALIDAD

MICROSERVICIOS

EFICIENCIA: LOS MICROSERVICIOS FACILITAN EL MANTENIMIENTO, EVOLUCIÓN Y EL MEJORAMIENTO CONTINUO DE LOS ACTIVOS DIGITALES.

ESCALABILIDAD: DIVIDIR UNA APLICACIÓN EN MICROSERVICIOS PERMITE QUE CADA COMPONENTE SE PUEDA ESCALAR DE MANERA INDIVIDUAL.

COMPATIBILIDAD CON DIFERENTES LENGUAJES Y TECNOLOGÍAS.

SVELTE

RENDIMIENTO: SVELTE ES CAPAZ DE REALIZAR CONSULTAS MÁS COMPLEJAS EN COMPARACIÓN A OTROS FRAMEWORKS, GRACIAS A SU COMPILADOR QUE LE PERMITE ALMACENAR EN CACHÉ LOS DATOS.

FACILIDAD DE MODIFICACIÓN: SVELTE PERMITE EXTENDER COMPONENTES SIN NECESIDAD DE MODIFICARLOS.

PORTABILIDAD: SVELTE ES UN FRAMEWORK LIVIANO, LO QUE FACILITA SU IMPLEMENTACIÓN EN DIFERENTES PLATAFORMAS.



REST/XML

EFICIENCIA: LAS APIS REST SON EFICIENTES EN TÉRMINOS DE RENDIMIENTO Y USO DE RECURSOS.

INTEROPERABILIDAD: LAS APIS REST PUEDEN INTERACTUAR CON CUALQUIER SISTEMA QUE USE PROTOCOLOS HTTP.

MODIFICABILIDAD: LAS APIS REST PERMITEN LA EXTENSIÓN (AGREGAR NUEVOS ENDPOINTS) SIN NECESIDAD DE MODIFICAR LOS EXISTENTES.

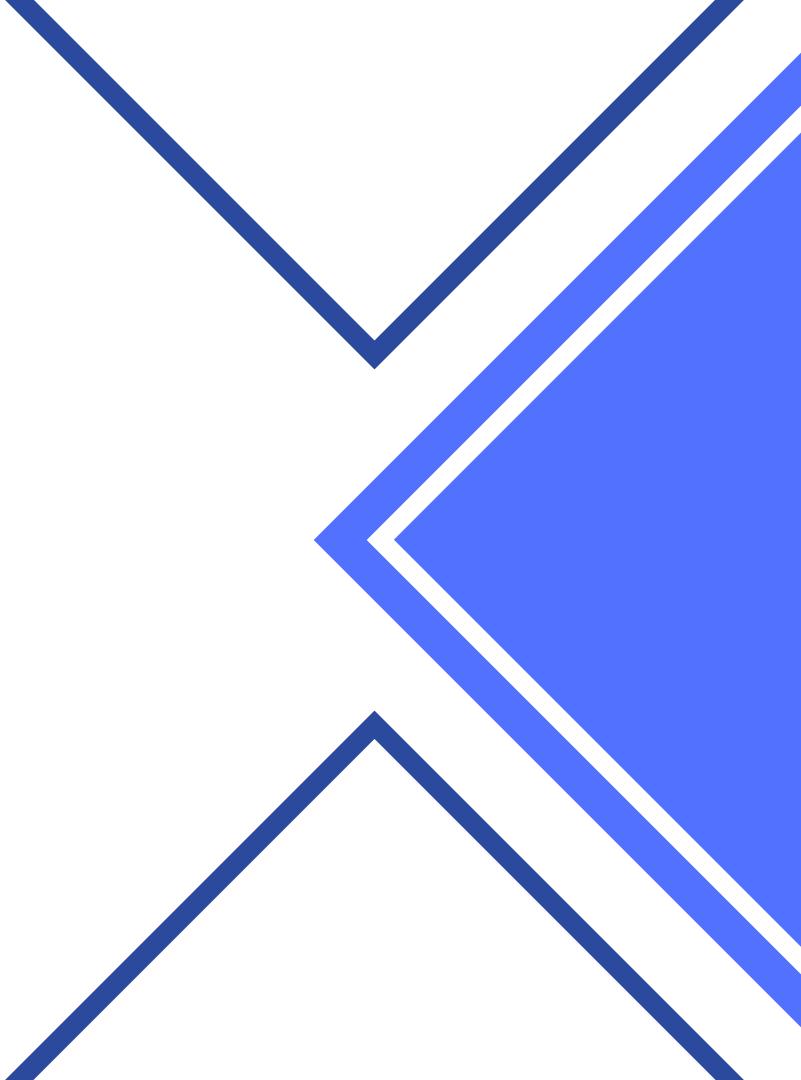


GO

EFICIENCIA: GO ES CONOCIDO POR SU EFICIENCIA EN TÉRMINOS DE RENDIMIENTO Y USO DE RECURSOS.

FACILIDAD DE MODIFICACIÓN: GO
PERMITE EXTENDER EL
COMPORTAMIENTO DE LAS ESTRUCTURAS
A TRAVÉS DE LA IMPLEMENTACIÓN DE
INTERFACES, SIN NECESIDAD DE
MODIFICAR LAS ESTRUCTURAS.

PORTABILIDAD: GO ES UN LENGUAJE COMPILADO QUE PUEDE EJECUTARSE EN DIFERENTES PLATAFORMAS.



MARIADB

EFICIENCIA: MARIADB ES CAPAZ DE REALIZAR CONSULTAS MÁS COMPLEJAS EN COMPARACIÓN A OTROS DBMS, GRACIAS A SU MOTOR DE ALMACENAMIENTO QUE LE PERMITE ALMACENAR EN CACHÉ LOS DATOS.

ESCALABILIDAD: MARIADB PERMITE QUE CADA COMPONENTE SE PUEDA ESCALAR DE MANERA INDIVIDUAL.

COMPATIBILIDAD CON DIFERENTES LENGUAJES Y TECNOLOGÍAS.



CASOS DE USO

Why Svelte Is the Next Big Thing

150%

growth in usage (from 8% to 20% in the last 2 years) awareness ratio (growth from 75% in the last 2 years)

90% satisfaction ratio (higher than React or Vue) position in the Stack Overfleranking of Most Loved Frameworks

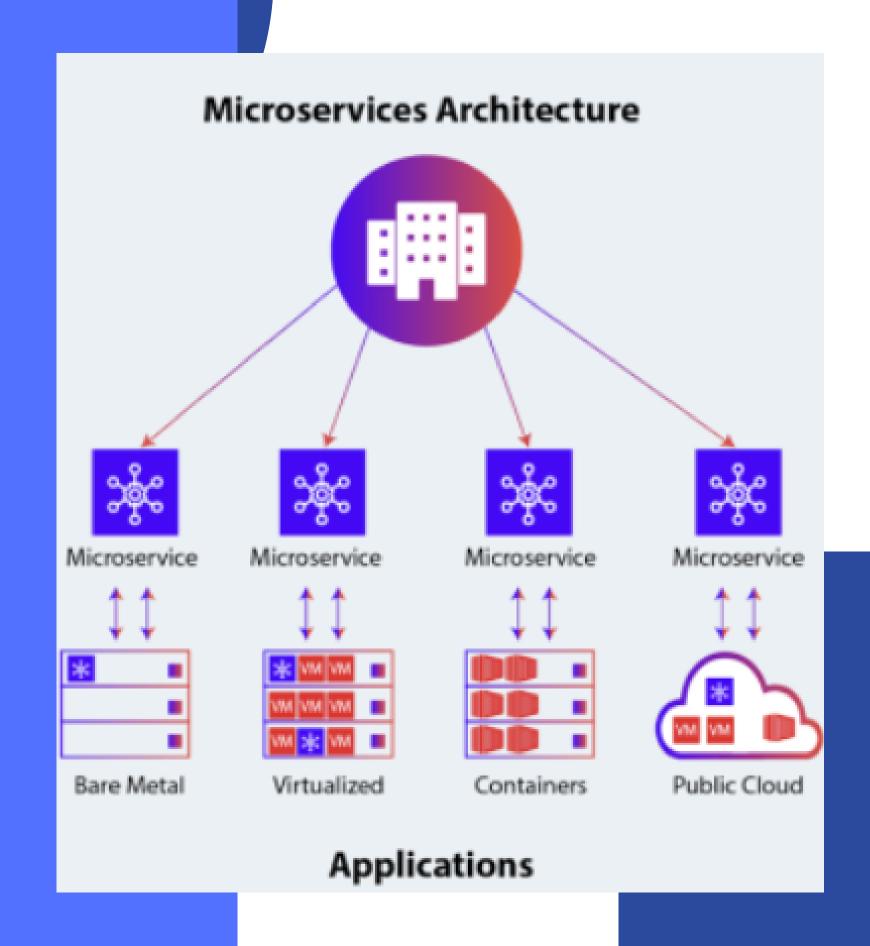


Sources: 2021 Developer Survey (Stack Overflow), The State of J

SVELTE

APLICACIONES DE UNA SOLA PÁGINA (SPA): SVELTE ES UNA GRAN OPCIÓN PARA LA CONSTRUCCIÓN DE APLICACIONES DE UNA SOLA PÁGINA QUE REQUIEREN ACTUALIZACIONES EN TIEMPO REAL E INTERFACES DE USUARIO DINÁMICAS.

APLICACIONES WEB PROGRESIVAS (PWA): SVELTE SE PUEDE UTILIZAR PARA CREAR APLICACIONES WEB PROGRESIVAS QUE PROPORCIONAN UNA EXPERIENCIA SIMILAR A UNA APLICACIÓN NATIVA EN LA WEB.



MICROSERVICIOS

NETFLIX: ESTA PLATAFORMA UTILIZA
MICROSERVICIOS PARA EL
FUNCIONAMIENTO DE SUS PRODUCTOS
(AWS).

SOFTWARE DE RECURSOS HUMANOS: SE PUEDE DESARROLLAR UN SOFTWARE DE RECURSOS HUMANOS UTILIZANDO UNA ARQUITECTURA DE MICROSERVICIOS.

GO

Apptunix

Top Golang Use Cases



GOOGLE: GO FUE CREADO EN GOOGLE EN 2007, Y DESDE ENTONCES, LOS EQUIPOS DE INGENIERÍA DE GOOGLE HAN ADOPTADO GO PARA CONSTRUIR PRODUCTOS Y SERVICIOS A GRAN ESCALA.

PAYPAL: PAYPAL UTILIZA GO PARA MODERNIZAR Y ESCALAR SUS SERVICIOS. AMERICAN EXPRESS: GO PROPORCIONA A AMERICAN EXPRESS LA VELOCIDAD Y ESCALABILIDAD QUE NECESITA PARA SUS REDES DE PAGOS Y RECOMPENSAS.

World's Most Popular Databases 1400 1200 1000 800 600 400 200

MARIADB

TRANSACCIONES: MARIADB ENTERPRISE SERVER ES LA BASE DE DATOS DE CÓDIGO ABIERTO LÍDER PARA EL PROCESAMIENTO DE TRANSACCIONES A CUALQUIER ESCALA. ANALÍTICA: MARIADB ENTERPRISE SERVER PUEDE SER DESPLEGADO COMO UN ALMACÉN DE DATOS O BASE DE DATOS ANALÍTICA UTILIZANDO ALMACENAMIENTO COLUMNAR Y PROCESAMIENTO MASIVAMENTE PARALELO (MPP) PARA REALIZAR CONSULTAS INTERACTIVAS Y AD HOC EN CIENTOS DE MILES DE MILLONES DE FILAS SIN CREAR ÍNDICES.

XML/REST

- USO DE LA WEB: LAS API REST SE UTILIZAN EN LA WEB PARA PERMITIR LA COMUNICACIÓN ENTRE DIFERENTES SERVICIOS WEB.
- XML EN SOAP: XML SE UTILIZA EN SOAP, UN PROTOCOLO DE INTERCAMBIO DE MENSAJES QUE PERMITE A LAS APLICACIONES COMUNICARSE A TRAVÉS DE UNA RED. EN UN MENSAJE SOAP, SE INCLUYE UN ELEMENTO <SOAP:BODY> QUE CONTIENE EL CONTENIDO DEL MENSAJE.