

Pub - Sub



RabbitMQ

The RabbitMQ logo, featuring a white rabbit head icon on an orange square background, followed by the text 'RabbitMQ' in white.

.NET

SQL server

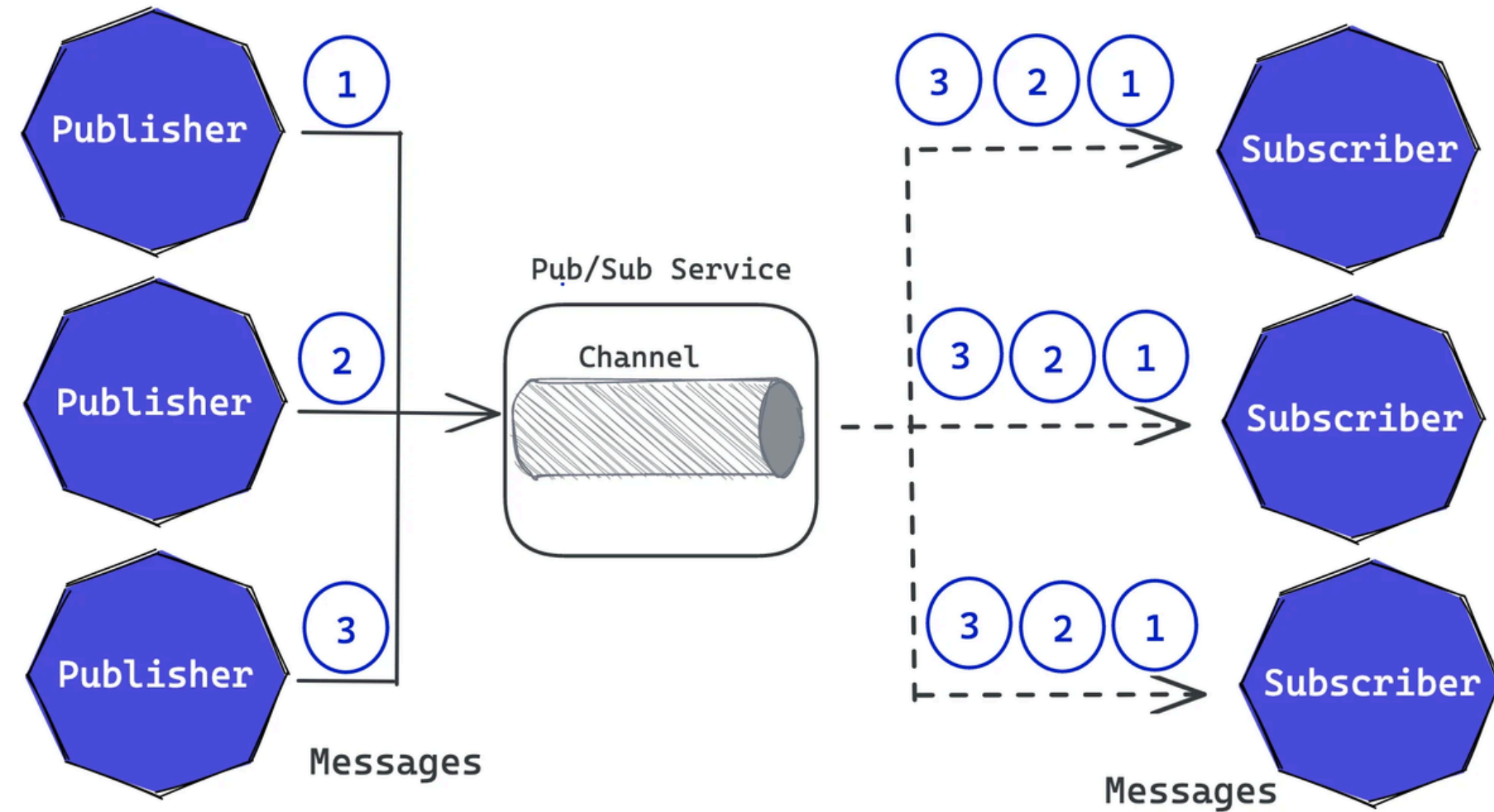
GRUPO 7
NICOLAS Q, NICOLAS R,
SEBASTIAN I

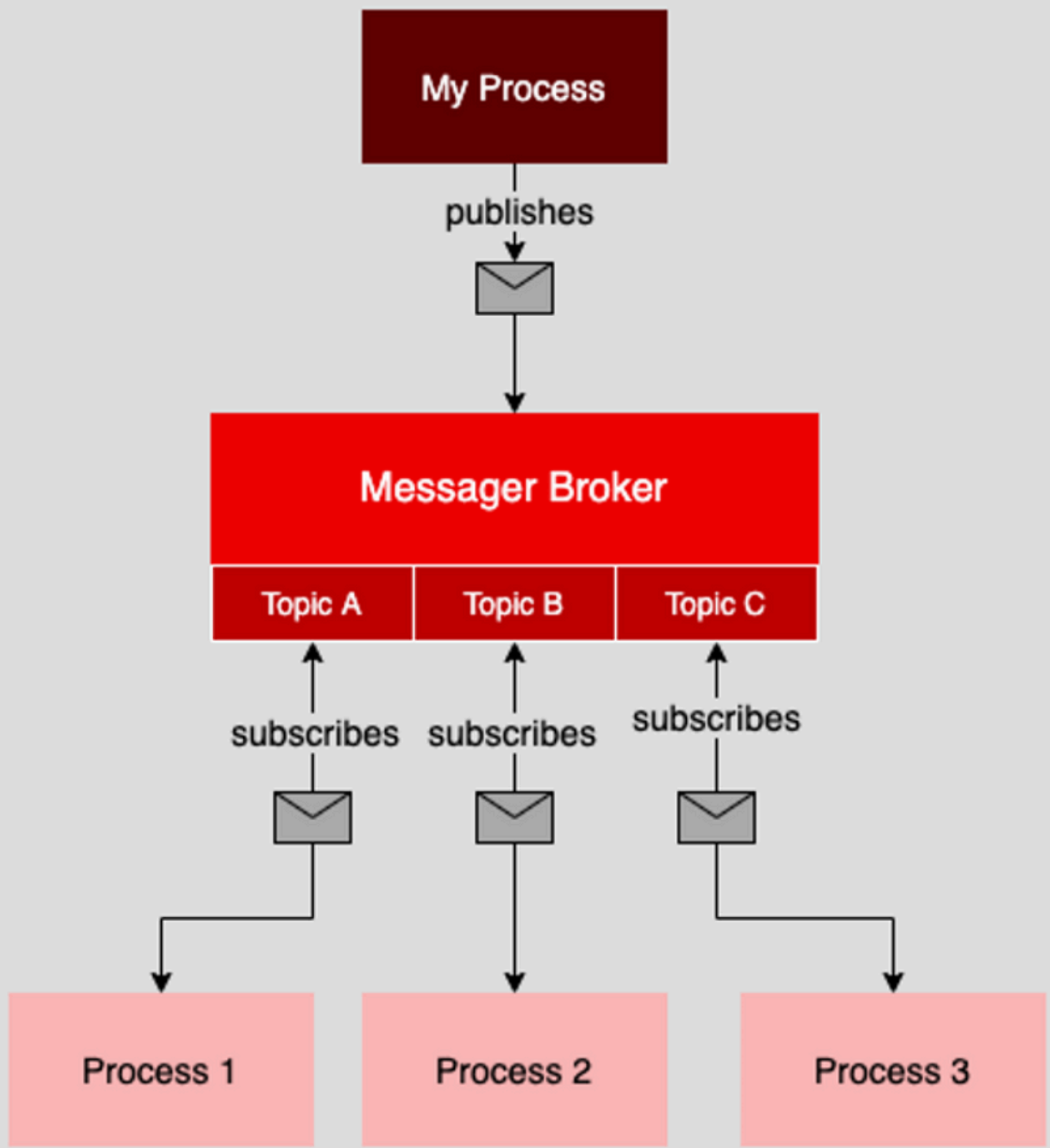
CONTENIDO

- | | | | |
|----|---|----|----------------------------|
| 01 | DEFINICION | 06 | MATRIZ DE PRINCIPIOS SOLID |
| 02 | CARACTERÍSTICAS | 07 | ATRIBUTOS DE CALIDAD |
| 03 | HISTORIA Y EVOLUCION | 08 | ANALISIS DE TACTICAS |
| 04 | VENTAJAS Y DESVENTAJAS | 09 | MERCADO LABORAL |
| 05 | SITUACIONES DONDE SE
PUEDE USAR Y CASOS DE
APLICACION | 10 | DIAGRAMAS |
| | | 11 | CONCLUSIÓN |

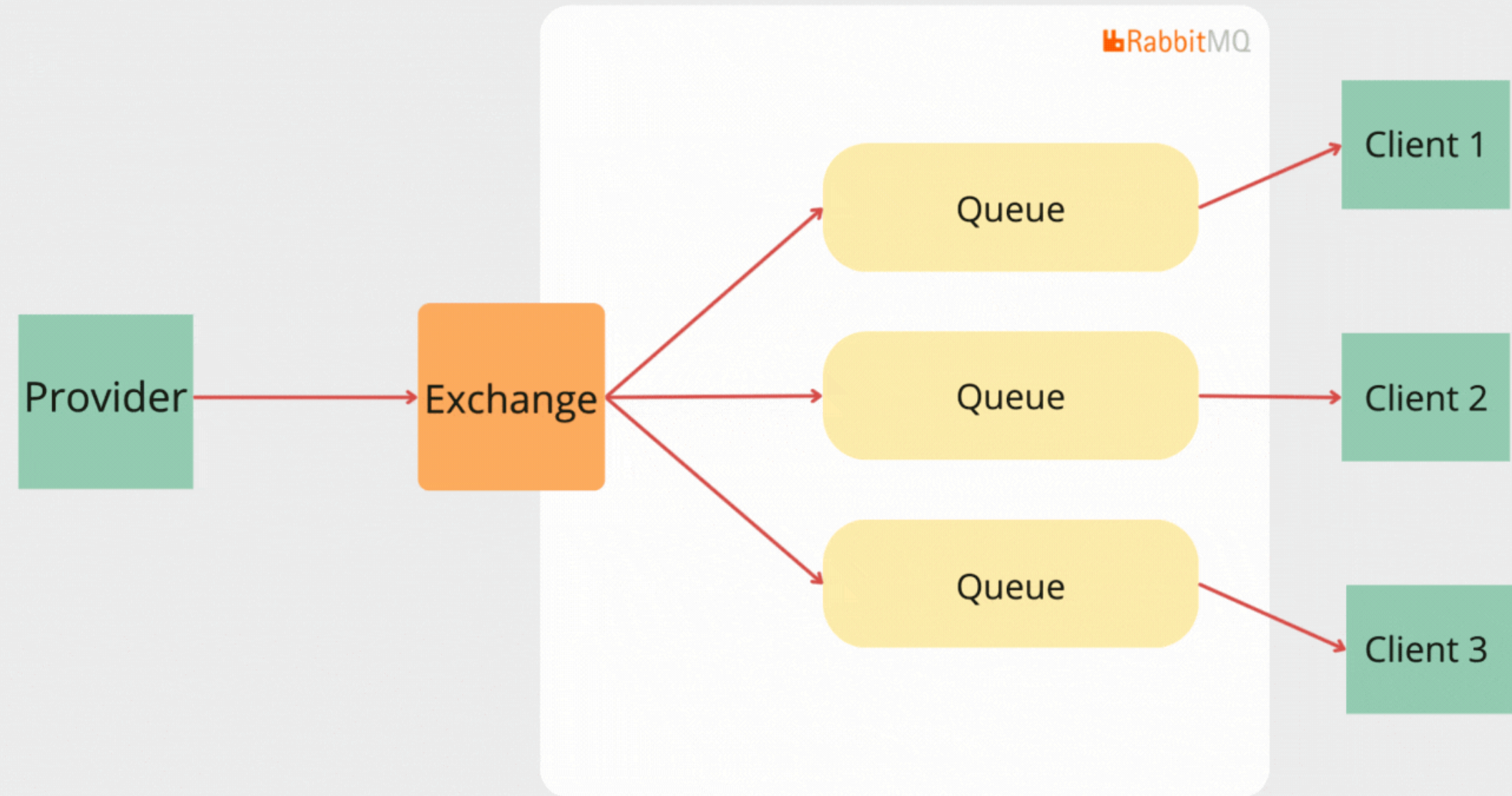
PUB-SUB

Es un patrón de diseño de mensajería que permite que diferentes componentes de un sistema se comuniquen entre sí de manera asíncrona y desacoplada. Este patrón se basa en la idea de que un publicador emite mensajes o eventos y uno o más suscriptores reciben esos mensajes si están interesados en ellos.

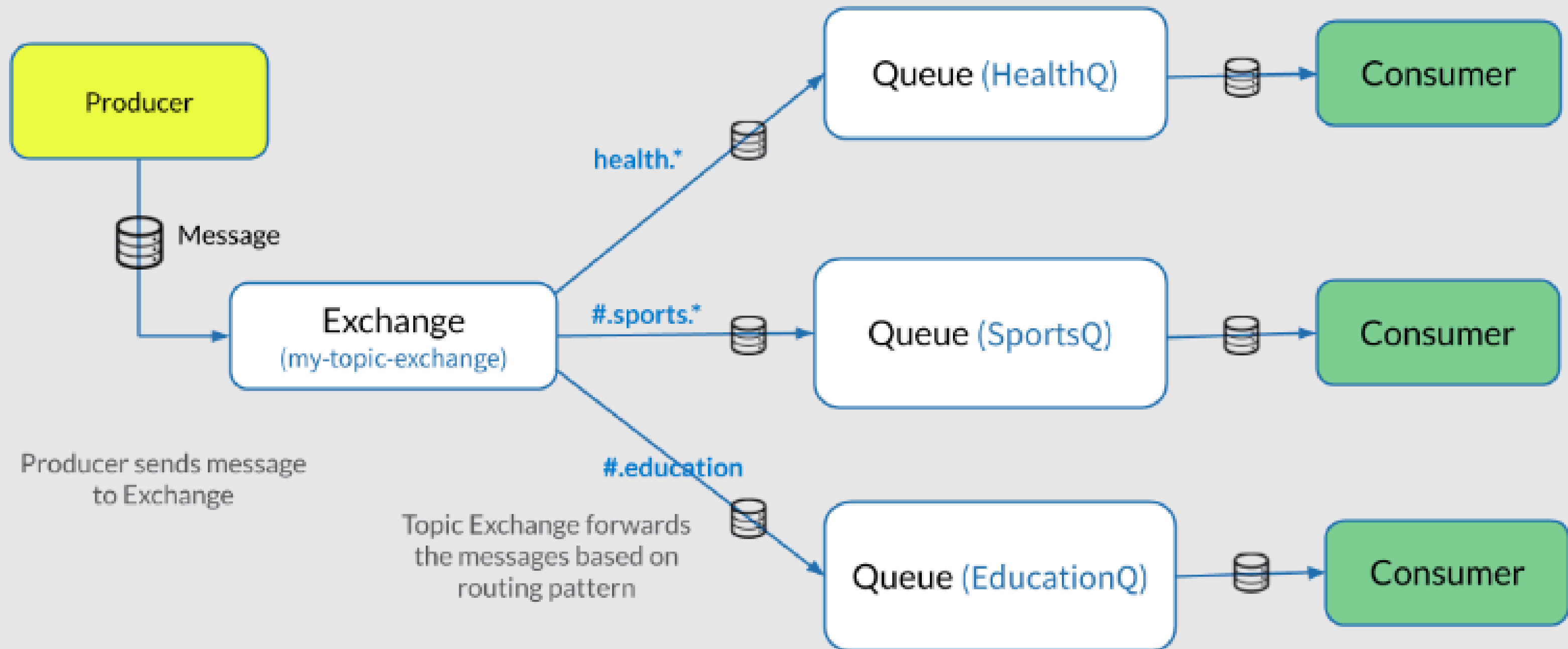




Este "interes" es mediante topicos.



Topic Exchange in RabbitMQ



DEFINICIÓN



Una pieza para utilizar dentro del marco de desarrollo de aplicaciones, que puede funcionar de manera autónoma y comunicarse con diversos sistemas: **Broker de mensajería**



Un marco de JavaScript para crear interfaces de usuario. Se basa en HTML, CSS y JavaScript estándar y proporciona un modelo de programación declarativo basado en componentes. SPA (Single Page Application)

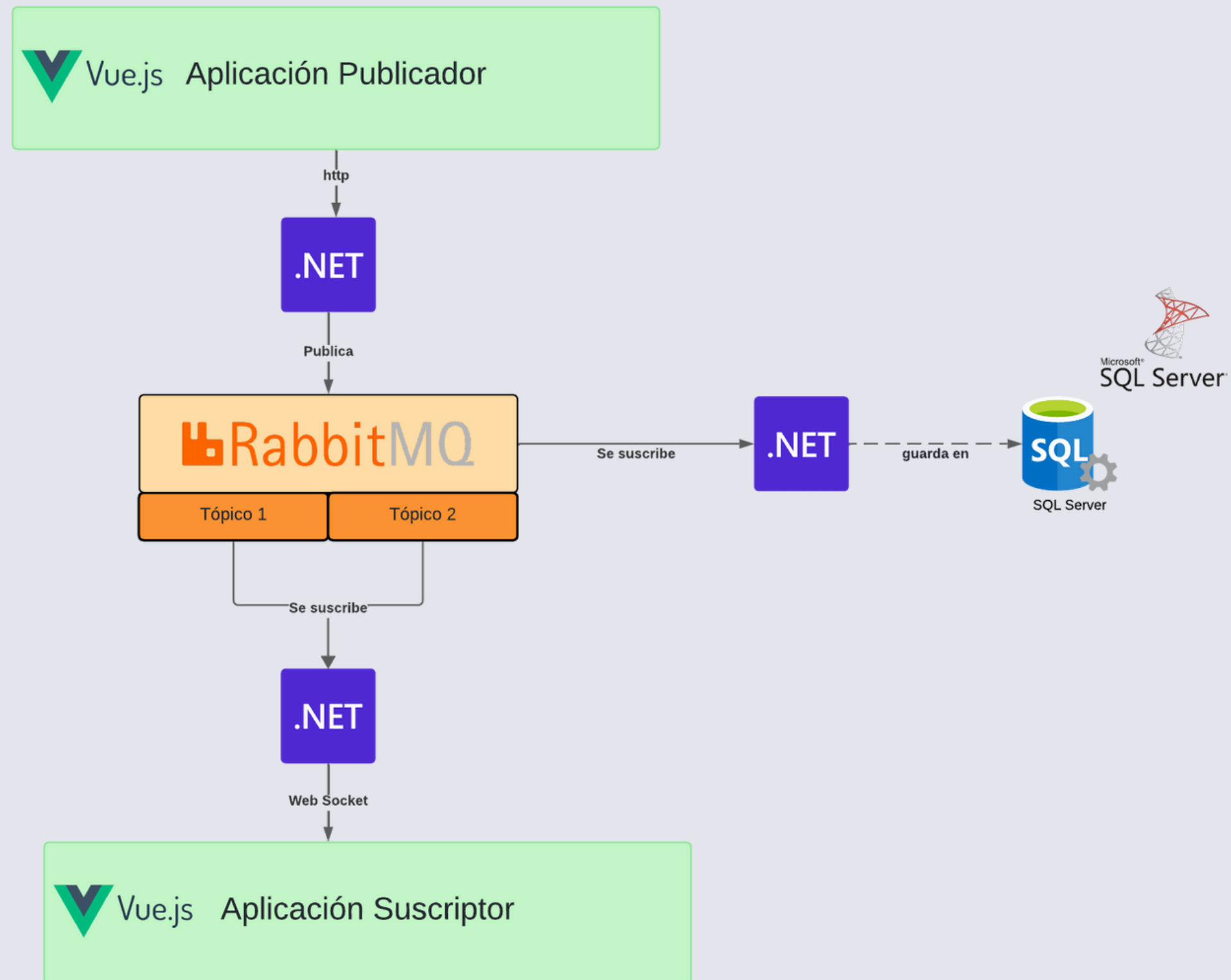


Plataforma de código abierto para crear aplicaciones de escritorio, web y móviles que se pueden ejecutar de forma nativa en cualquier sistema operativo. El sistema .NET incluye herramientas, bibliotecas y lenguajes que admiten el desarrollo de software moderno, escalable y de alto rendimiento.



Es una plataforma para generar soluciones de integración de datos de alto rendimiento, lo que incluye paquetes que proporcionan procesamiento de extracción, transformación y carga para almacenamiento de datos

CARACTERÍSTICAS



HISTORIA EVOLUCIÓN

PUBLICADOR SUBSCRIPTOR

Tiene sus raíces en los sistemas de comunicación de mensajes en los años 60 y 70, principalmente con los trabajos en sistemas distribuidos y redes de computadoras

En los sistemas de comunicaciones tempranas, como los basados en ARPANET (predecesora de Internet), el modelo principal de comunicación era punto a punto

El modelo de publicador-suscriptor surgió como una respuesta a esta necesidad, permitiendo que los emisores (publicadores) no tuvieran que conocer a los receptores (suscriptores) de los mensajes.

Event-Driven Architecture: Microservicios

IOT : Donde millones de dispositivos distribuidos publican datos a plataformas que los procesan y los envían a los suscriptores interesados.

HISTORIA EVOLUCIÓN



RABBIT-MQ

Creado en 2007 por la empresa **LShift**, liderada por Alexis Richardson y Matthias Radestock, como un **mensaje broker** que implementara el protocolo **AMQP** (Advanced Message Queuing Protocol), diseñado para la interoperabilidad entre diferentes sistemas.

En 2010, RabbitMQ fue adquirido por **VMware** como parte de su estrategia de expansión en la infraestructura de nube y software. En 2012, VMware lanzó Pivotal Software, bajo la cual RabbitMQ se continuó desarrollando.

Medida que las arquitecturas distribuidas y los microservicios se volvieron más populares, RabbitMQ evolucionó para soportar múltiples protocolos de mensajería como STOMP (Streaming Text Oriented Messaging Protocol) y MQTT (Message Queuing Telemetry Transport). Esta flexibilidad permitió que RabbitMQ se adaptara a una amplia gama de aplicaciones.



SQL-SERVER

Fue lanzado en 1989, fruto de una colaboración entre Microsoft, Sybase, y Ashton-Tate. Inicialmente, SQL Server estaba basado en la tecnología de Sybase, diseñado para correr en sistemas operativos OS/2. Sin embargo, en 1994, Microsoft se separó de Sybase y continuó desarrollando SQL Server de manera independiente.

A principios de los 2000, con SQL Server 2000 y posteriormente SQL Server 2005, Microsoft comenzó a incluir herramientas para Business Intelligence (BI), como SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS), y SQL Server Integration Services (SSIS), posicionando a SQL Server como una solución integral para bases de datos y análisis de datos empresariales.

En años recientes, SQL Server ha adoptado características modernas como Big Data Clusters, integraciones con Azure para la nube, y mejoras en el manejo de grandes volúmenes de datos con tecnologías como PolyBase.

HISTORIA EVOLUCIÓN

.NET



.NET

Microsoft buscaba una manera de competir con Java y otros entornos de desarrollo multiplataforma. Esto llevó al desarrollo de .NET Framework, anunciado en 2000 por Bill Gates como una plataforma integral para construir aplicaciones web, de escritorio y móviles en entornos Windows.

En 2002, se lanzó la primera versión de .NET Framework 1.0, que incluía el Common Language Runtime (CLR), un entorno de ejecución que soportaba múltiples lenguajes de programación (C#, VB.NET, etc.). Esta versión permitió la creación de aplicaciones robustas sobre el sistema operativo Windows.

Microsoft lanzó .NET Core, una versión de .NET diseñada para ser multiplataforma (Windows, Linux, y macOS) y de código abierto, lo que marcó un cambio significativo en la estrategia de Microsoft para hacer .NET más accesible y adaptable a las nuevas arquitecturas en la nube y microservicios.

VUE.JS



Fue creado en 2014 por **Evan You**, un exdesarrollador de Google. Frustrado por la complejidad de otros frameworks de JavaScript como Angular, decidió crear una herramienta más accesible y fácil de aprender. Vue fue diseñado para ser **progresivo**, lo que significa que los desarrolladores podían adoptarlo en proyectos existentes de manera incremental.

En 2016, Vue 2 fue lanzado, consolidando a Vue como una de las opciones más populares en el desarrollo frontend. Esta versión trajo mejoras en rendimiento, una mejor gestión de los componentes, y una sintaxis más refinada para trabajar con datos reactivos.

En 2020, se lanzó Vue 3, que introdujo la **Composition API**, permitiendo una mayor flexibilidad en la creación y organización de componentes. Vue 3 también se centró en mejorar el rendimiento y la escalabilidad, haciéndolo ideal para aplicaciones más grandes y complejas. Además, Vue.js continúa siendo una de las opciones favoritas para el desarrollo de interfaces de usuario debido a su simplicidad y rendimiento.

VENTAJAS

PUB/SUB

Desacoplamiento: Los publicadores **no necesitan saber quiénes son los suscriptores** ni cuántos son. Esto facilita la evolución independiente de los componentes, **permitiendo añadir nuevos suscriptores sin modificar a los publicadores.**

Escalabilidad: ya que **múltiples suscriptores pueden recibir mensajes simultáneamente sin sobrecargar al publicador.** Esto es ideal para sistemas distribuidos y de gran escala.

Flexibilidad: Soporta una arquitectura orientada a eventos y permite que los mensajes se publiquen cuando ocurren eventos relevantes, desencadenando acciones en los suscriptores. Esto es útil en sistemas donde los eventos ocurren de manera impredecible.

Asincronía : Los publicadores y suscriptores funcionan de manera **asíncrona**. lo que mejora el rendimiento al evitar bloqueos y tiempos de espera innecesarios

.

VENTAJAS



Facilita la **comunicación entre servicios sin que dependan unos de otros**, lo que permite flexibilidad y escalabilidad en sistemas distribuidos.

Puede **manejar grandes volúmenes de mensajes**, siendo una solución robusta para aplicaciones que requieren procesamiento en paralelo.

Ofrece **confirmación de entrega de mensajes y permite reintentar en caso de fallos**.

Compatible con **varios patrones** de mensajería (cola de mensajes, pub/sub, routing, etc.), lo que lo hace versátil.



Puede **integrarse gradualmente en proyectos existentes** o ser usado para aplicaciones grandes gracias a su enfoque modular.

La actualización automática del **DOM (Document Object Model)** cuando cambian los datos mejora la experiencia del usuario con aplicaciones más rápidas y fluidas.



Desarrollar aplicaciones en **Windows, Linux y macOS**, lo que brinda más opciones y flexibilidad

Ofrece un **rendimiento sólido**, siendo adecuado para aplicaciones que manejan grandes cargas de trabajo,

Cuenta con un **amplio ecosistema de herramientas, bibliotecas y soporte** tanto de la comunidad como de Microsoft.

Incluye características **avanzadas de seguridad**, como **autenticación y manejo de identidades**.



Reconocido por su capacidad de **manejar grandes volúmenes de datos y ofrecer un rendimiento consistente y fiable**

Cuenta con el **respaldo de Microsoft**, lo que garantiza actualizaciones regulares y soporte.

Incluye herramientas como **SSRS (Reporting Services) y SSIS (Integration Services)**, que facilitan la integración y análisis de datos.

Ofrece **sólidas características de seguridad**, como cifrado y control de acceso basado en roles

DESVENTAJAS

PUB/SUB

Garantía de entrega: puede ser difícil garantizar que todos los suscriptores reciban todos los mensajes. Algunos sistemas ofrecen solo entrega "al mejor esfuerzo", lo que significa que podrían perderse mensajes en situaciones de alta carga o fallos.

Latencia: La latencia puede aumentar debido a la naturaleza asíncrona de la comunicación y el uso de intermediarios (brokers).

Orden de los mensajes: Garantizar el orden de los mensajes en todos los suscriptores puede ser un desafío, especialmente si los mensajes se distribuyen en un sistema distribuido o con múltiples brokers. Esto puede complicar el manejo de datos secuenciales o dependientes.

Sobrecarga en el broker : Si el broker tiene que manejar grandes volúmenes de mensajes o gestionar muchas conexiones, puede convertirse en un punto único de fallo o en un cuello de botella para el rendimiento global.

DESVENTAJAS



La implementación de RabbitMQ **puede requerir configuración avanzada y manejo de errores**, lo que introduce complejidad en la arquitectura.

Si no se configura adecuadamente, **puede haber problemas de acumulación de mensajes en colas, afectando el rendimiento**.

La administración y monitoreo **continuo de las colas de mensajes es esencial** para asegurar un rendimiento óptimo.



La comunidad es más **pequeña** que la de React o Angular, lo que puede hacer que algunos paquetes y bibliotecas estén menos desarrollados.

no cuenta con el **respaldo de grandes corporaciones** tecnológicas como Angular (Google) o React (Facebook)

Las aplicaciones basadas en Vue.js, como las SPAs, pueden tener dificultades **con el SEO si no se implementan soluciones como SSR (Server-Side Rendering)**.



Aunque es poderoso, **puede ser más complejo para desarrolladores sin experiencia previa en .NET** o en el ecosistema de Microsoft.

Aunque .NET Core es open-source, si se implementa en entornos **Windows, puede haber costos asociados a licencias de Windows Server y SQL Server**.

Las aplicaciones desarrolladas en .NET pueden ser **más pesadas en términos de consumo de recursos**



SQL Server no es una base de datos gratuita (aunque tiene una versión Express limitada)

A diferencia de las bases de datos open-source como MySQL o PostgreSQL, SQL Server **tiene una comunidad más cerrada** y depende en gran medida del soporte de Microsoft.

Algunas versiones de **SQL Server están optimizadas** para ejecutarse en entornos Windows, lo que puede limitar su adopción en entornos basados en Linux.

CASOS DE USO

CASOS DE APLIACIÓN

Rabbit MQ

MICRO-SERVICIOS

NOTIFICACIONES EN
TIEMPO REAL

NETFLIX: COLAS DE TRABAJOS
ASINCRONOS Y NOTIFICACIONES

UBER: PARA EVENTOS Y MANEJO
DE TRANSACCIONES DISTRIBUIDAS
EN TIEMPO REAL,

.NET

APIS RESTFUL

APLICACIONES
EMPRESARIALES A
GRAN ESCALA: .NET

GODADDY: LA
INFRAESTRUCTURA DE SU
SERVICIO DE DOMINIOS.

STACK OVERFLOW: EL
BACKEND

Vue.js

SPAS (SINGLE PAGE
APPLICATIONS)

APLICACIONES WEB
INTERACTIVAS

ALIBABA

BEHANCE

SQL Server

SISTEMAS OLTP
(PROCESAMIENTO DE
TRANSACCIONES EN LÍNEA)

APLICACIONES DE
ANÁLISIS DE DATOS

DOMINO'S PIZZA: MANEJAR SU
SISTEMA DE PEDIDOS EN LÍNEA,
PROCESANDO TRANSACCIONES EN
TIEMPO REAL

STACK OVERFLOW: PARA
ALMACENAR MILLONES DE
REGISTROS DE PREGUNTAS

POPULARIDAD STACK

La combinacion es algo
popular en articulos y GitHub



<https://6sense.com/tech/queueing-messaging-and-background-processing/rabbitmq-market-share>
<https://www.wearedevelopers.com/magazine/most-popular-backend-frameworks>
<https://db-engines.com/en/ranking>
<https://inveritasoft.com/article-10-reasons-why-vue-js-is-so-popular>
https://db-engines.com/en/ranking_trend/system/Microsoft+SQL+Server
<https://medium.com/@kova98/pub-sub-in-net-with-masstransit-44cb27024497>
<https://alikhzlda.medium.com/a-simple-pub-sub-scenario-with-masstransit-6-2-rabbitmq-net-core-3-1-elasticsearch-mssql-5a65c993b2fd>

01

.Net, C#, Altisima popularidad y demanda

02

Vue.js, JavaScript, creciente popularidad y demanda

03

Colas, RabbitMQ, #2 en popularidad, altamente demandado

04

SQL Server EE, perdido popularidad al igual que demanda, sigue siendo de los más usados

MATRIZ DE ANÁLISIS

PRINCIPIOS SOLID VS TEMAS

	SQL Server EE	Vue.js	Rabbit MQ	.NET	Publicador/ Subcriptor
Single Responsibility	Medio	Bajo	Alto	Alto	Alto
Open/Closed	Bajo	Medio	Medio	Alto	Medio
Liskov Substitution	Bajo	Medio	Bajo	Alto	Medio
Interface Segregation	Bajo	Alto	Alto	Alto	Alto
Dependency Inversion	Medio	Bajo	Alto	Alto	Alto

MATRIZ DE ANÁLISIS

ATRIBUTOS DE CALIDAD VS TEMAS

	SQL Server EE	Vue.js	Rabbit MQ	.NET	Publicador/s ubscriptor
Funcionalidad	Alta	Media	Alta	Alta	Alta
Fiabilidad	Alta	Media	Alta	Alta	Alta
Usabilidad	Media	Alta	Media	Alta	Media
Eficiencia	Alta	Alta	Alta	Alta	Alta
Mantenibilidad	Media	Alta	Media	Alta	Alta
Portabilidad	Media	Alta	Alta	Media	Alta

MATRIZ DE ANÁLISIS

TÁCTICAS VS TEMAS

	SQL Server EE	Vue.js	RabbitMQ	.NET	PublicadorSuscriptor
Caché	Medio	Bajo	Medio	Alto	Medio
Replicación	Alto	Bajo	Alto	Medio	Alto
Particionamiento	Alto	N/A	N/A	Medio	Alto
Balanceo de carga	Bajo	Medio	Alto	Alto	Alto
Compensación	Medio	Bajo	Alto	Alto	Alto
Monitoreo	Alto	Medio	Alto	Alto	Alto
Fallback	Medio	Bajo	Alto	Alto	Alto

MATRIZ DE ANÁLISIS

MERCADO LABORAL VS TEMAS

	SQL Server EE	Vue.js	Rabbit MQ	.NET	Publicador/ Subscriber
Demanda	Alto	Alto	Media	Alto	Medio
Salario promedio	Alto	Media	Alto	Alto	Alto
Nuevas ofertas	Alto	Alto	Media	Alto	Medio
Competencia	Media	Alto	Media	Media	Medio

MATRIZ DE ANÁLISIS

PATRONES LABORAL VS TEMAS

	SQL Server EE	Vue.js	Rabbit MQ	.NET	Publicador /Subscriber
Trabajo remoto	Medio	Alto	Medio	Alto	Medio
Metodologías ágiles	Alto	Alto	Alto	Alto	Alto
Contratos freelance	Medio	Alto	Medio	Alto	Medio
Roles de DevOps	Bajo	Medio	Alto	Medio	Alto

DIAGRAMA DYNAMIC C4

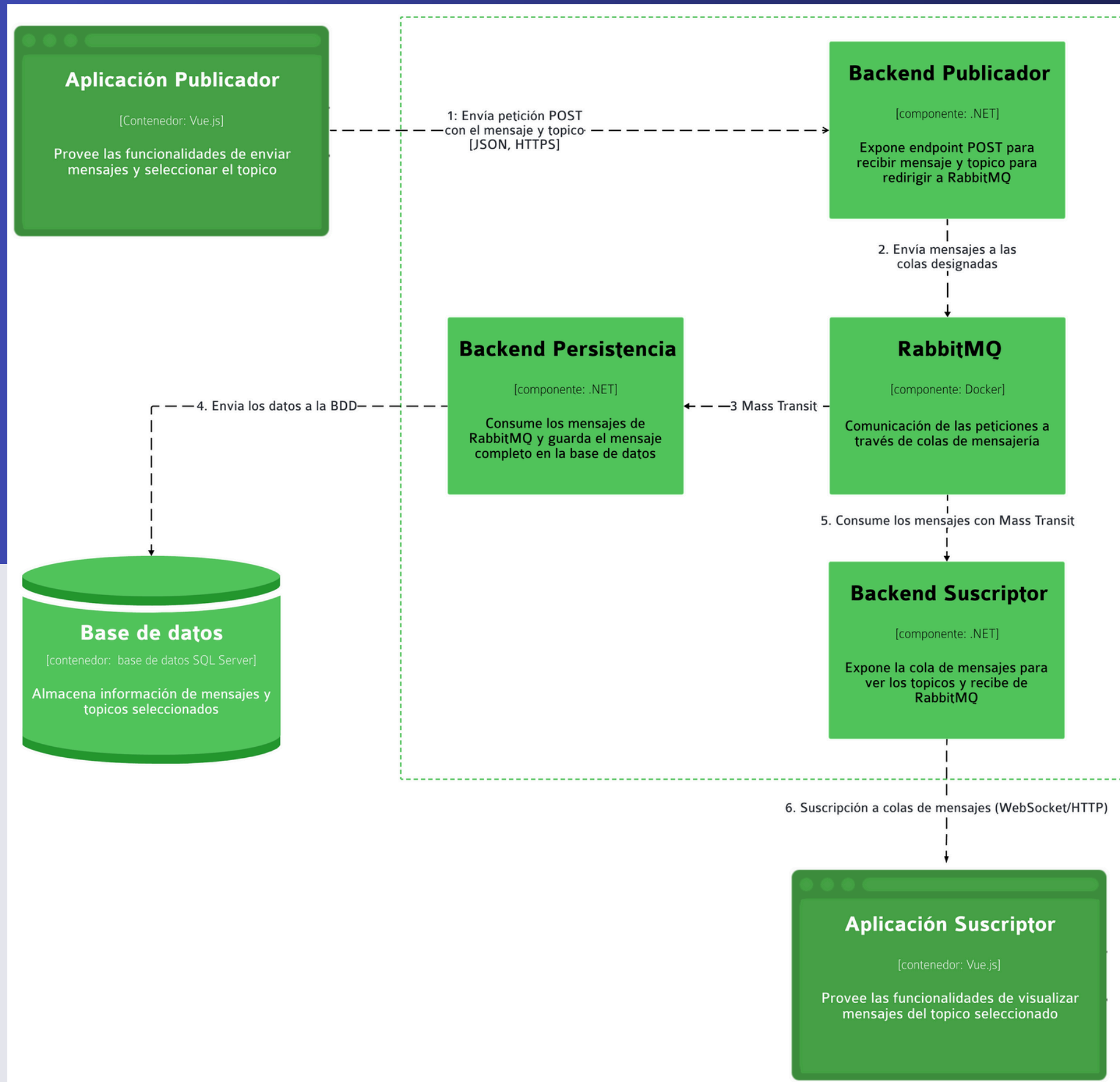


DIAGRAMA DE CONTEXTO



DIAGRAMA DESPLIEGUE C4

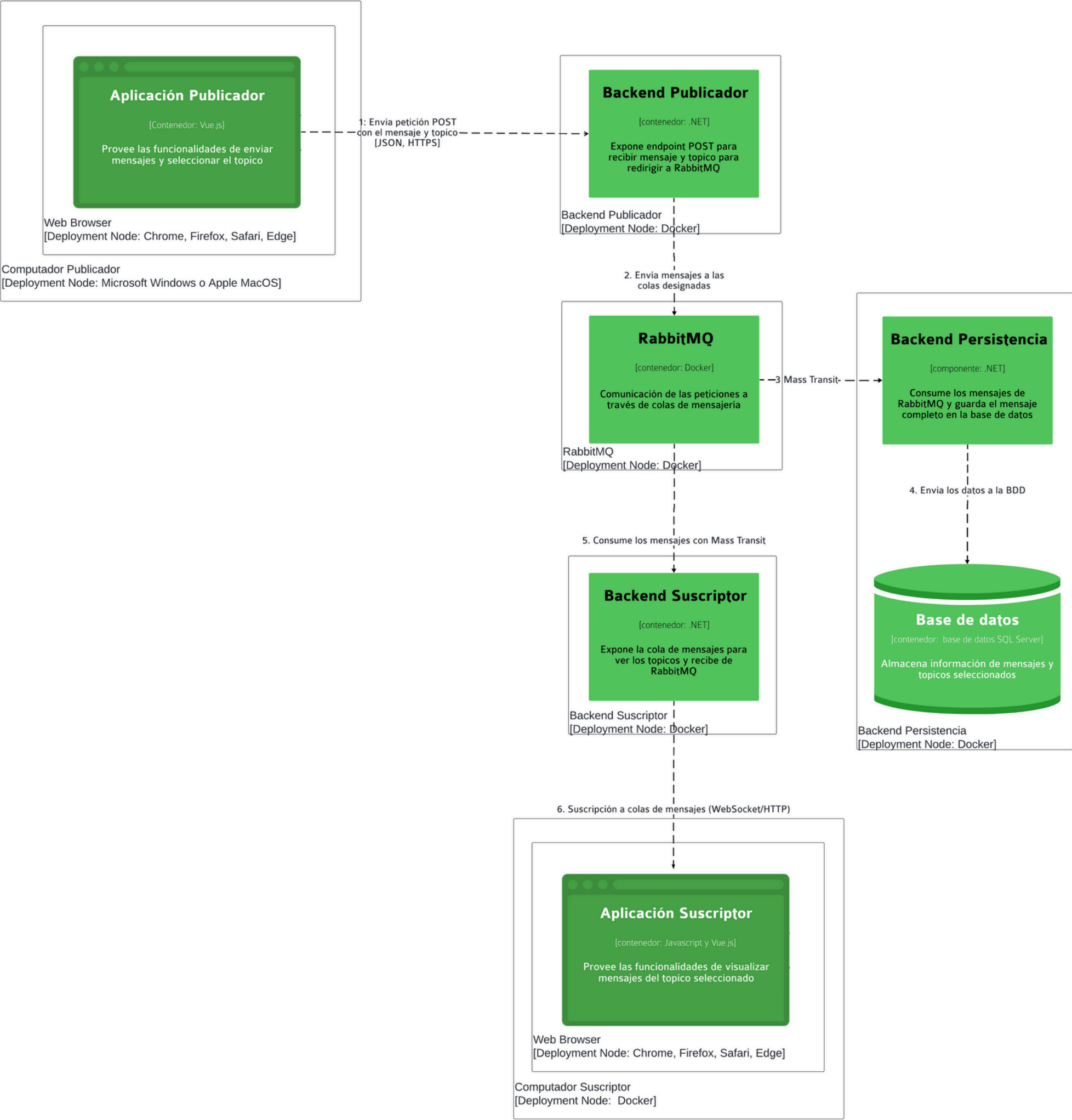


DIAGRAMA COMPONENTE C4

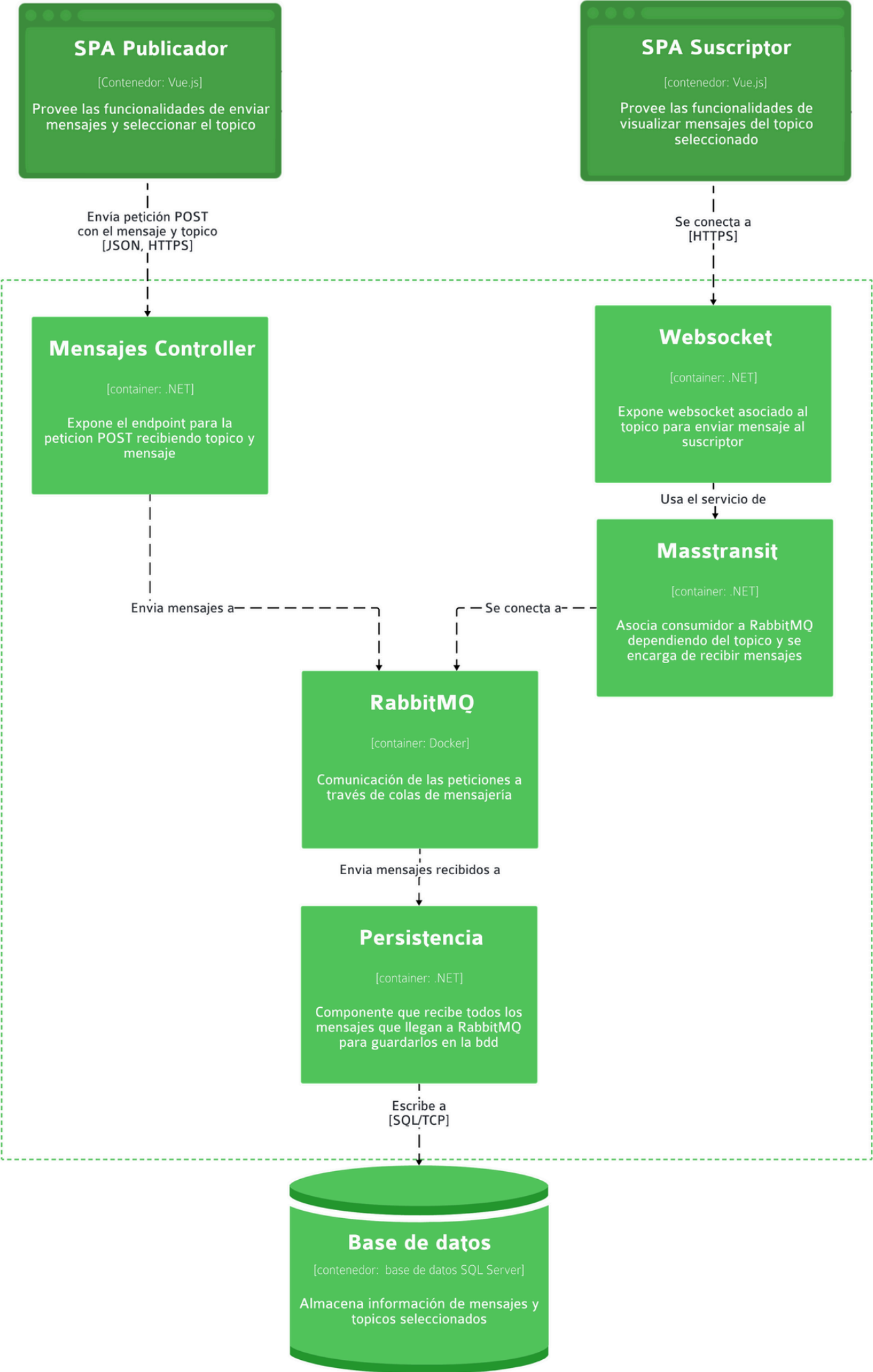


DIAGRAMA CONTENEDORES

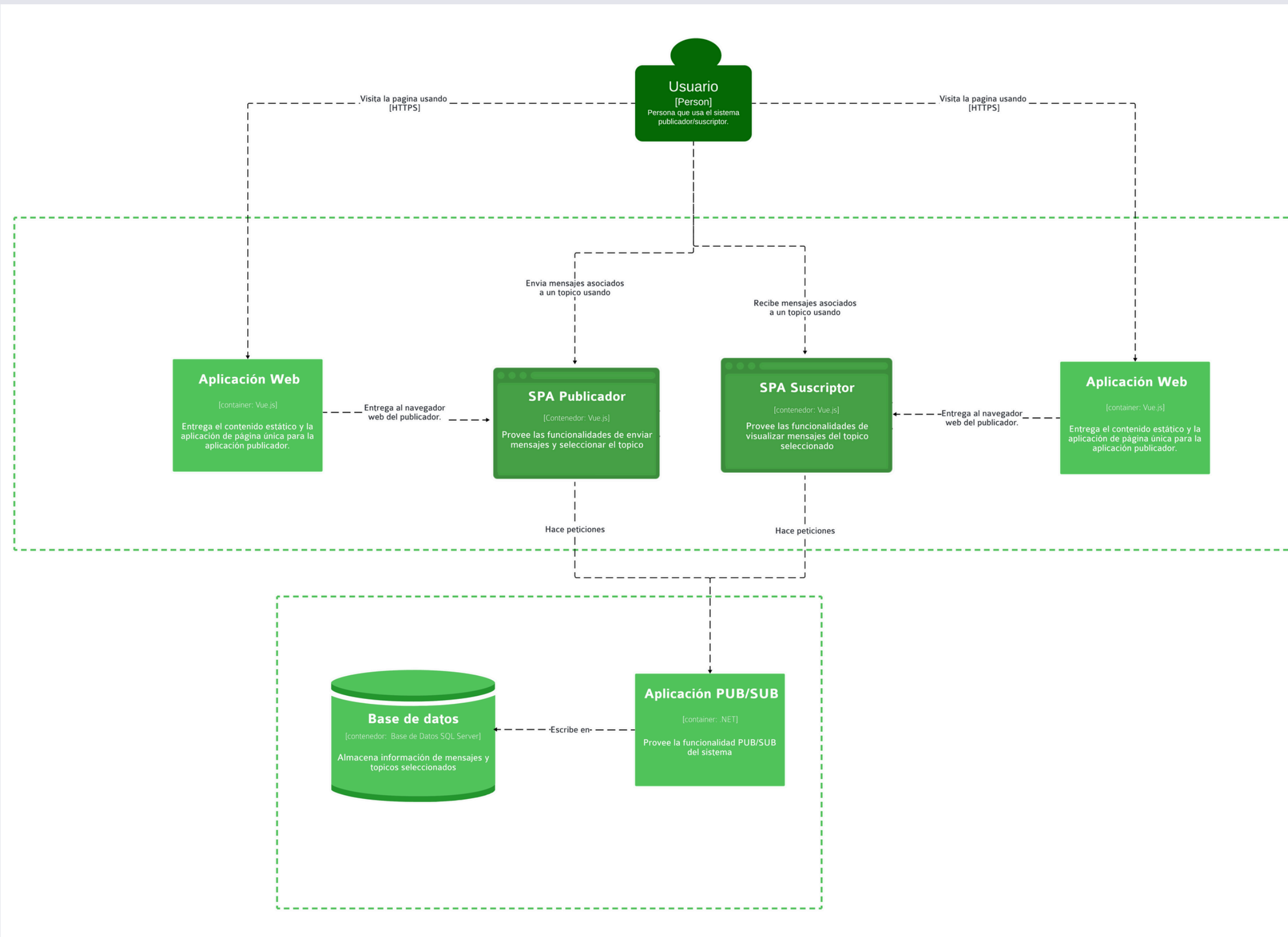
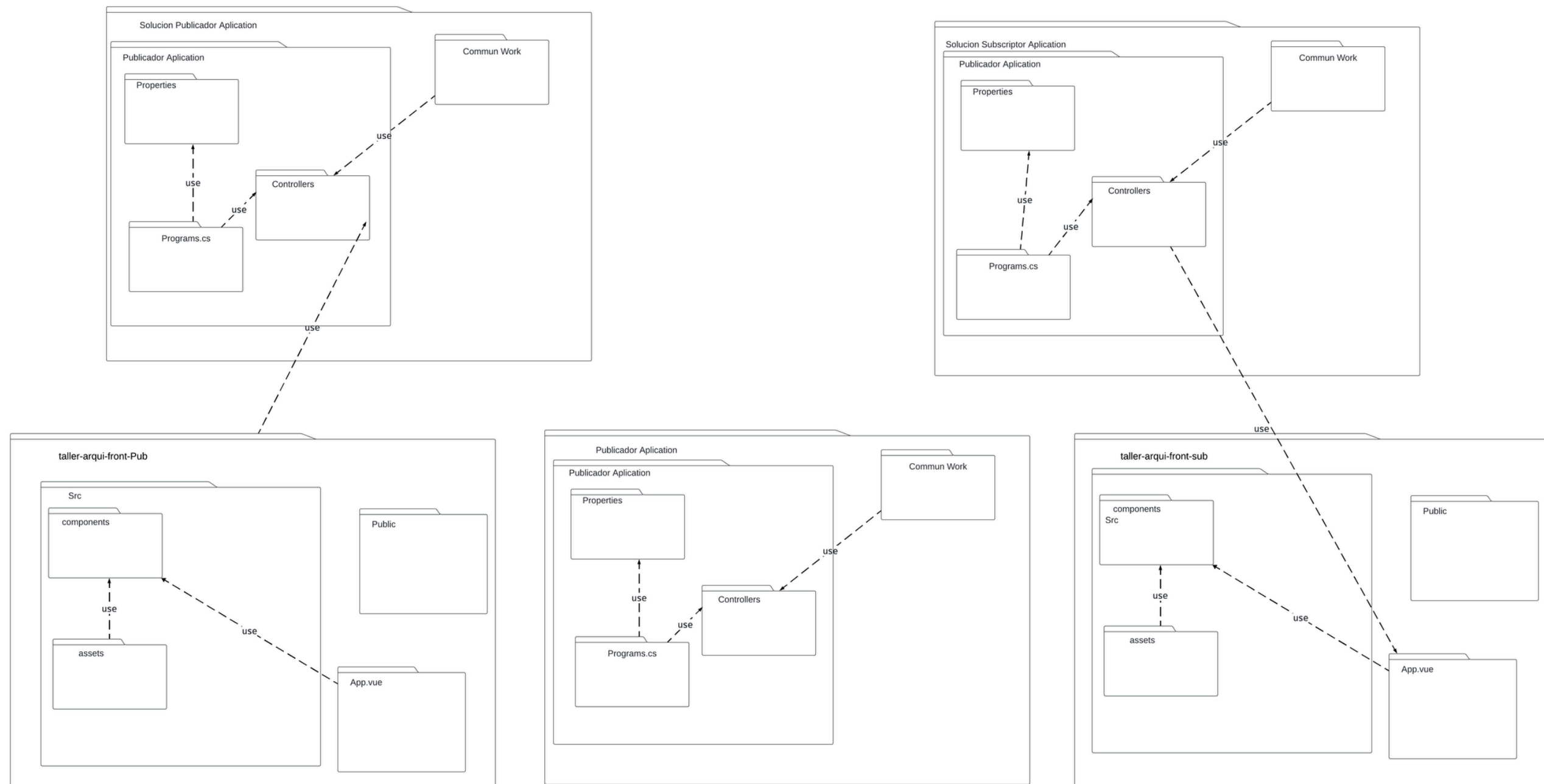


DIAGRAMA UML



VIDEO

