**Programación Web 3**

**UNLaM - Tecnicatura en Desarrollo Web**

**Trabajo Práctico de Investigación**

**SignalR y la Comunicación en Tiempo Real para Aplicaciones Web Colaborativas Modernas**



**Índice**

[Integrantes 2](#_heading=h.1fob9te)

[Objetivo 2](#_heading=h.3znysh7)

[Situación Actual 2](#_heading=h.2et92p0)

[¿Qué es SignalR? 3](#_heading=h.r5yeykdekhui)

[Flujo de comunicación cliente - servidor 6](#_heading=h.ljwv5iscxwep)

[Integración con ASP.NET y otros marcos de trabajo 7](#_heading=h.b45oqcjgj2y9)

[Características clave de SignalR 7](#_heading=h.er7q36bu824i)

[SignalR - Casos de uso 9](#_heading=h.embcpr4n811e)

[Aplicaciones de chat: Mensajería y colaboración en tiempo real 9](#_heading=h.bq3gom9f0yl)

[Actualizaciones de datos en directo: Cotizaciones del mercado de valores, resultados deportivos, apuestas en línea 10](#_heading=h.yeaf4crl6c48)

[Juegos en línea: Interacciones y sincronización multijugador 10](#_heading=h.bu3tbfnvs20y)

[IoT: Monitoreo y control remoto de sensores 11](#_heading=h.18j6e18yy538)

[Consideraciones de implementación 12](#_heading=h.t4o5ufeapi0m)

[Elección del mecanismo de transporte apropiado 12](#_heading=h.swc5jwq24kzw)

[Manejo del ciclo de vida de la conexión: Gestión de conexiones y estrategias de reconexión 12](#_heading=h.iu0xjpiaw26c)

[Consideraciones de seguridad: Autenticación, Autorización y Cifrado de datos 14](#_heading=h.27og9o1jqnak)

[Técnicas de optimización de rendimiento: Caché, Comprensión y Minimización de latencia 14](#_heading=h.itv6xetqg939)

[Tendencias Emergentes en la Comunicación Web en Tiempo Real 15](#_heading=h.t2lsqhrp2twk)

[Potencializando la Tecnología SignalR 15](#_heading=h.bldrzn8yh2zx)

[Desafíos de Escalabilidad en Implementaciones a Gran Escala 15](#_heading=h.8299bryk3d6a)

[Conclusiones 16](#_heading=h.7lbzu6rb4exy)

[Bibliografía 18](#_heading=h.c3i08rggxvtk)

# 

# **Integrantes**

* Del Pino Iván
* Mongelos Joaquín
* Paz Alejandro Daniel
* Pereyra Camila Anabella
* Torres Fabio Ezequiel

# **Objetivo**

Este trabajo de investigación se centrará en la exploración exhaustiva de SignalR, una biblioteca avanzada para la implementación de comunicación en tiempo real dentro del entorno de ASP.NET. Analizaremos detalladamente la arquitectura de la biblioteca, identificando sus componentes principales y las funciones que desempeñan, así como el flujo de trabajo que facilita. Además, se presentarán ejemplos prácticos de la aplicación de esta tecnología en diversos contextos de desarrollo, destacando cómo SignalR resuelve desafíos relacionados con la seguridad, la escalabilidad y la optimización de recursos para asegurar un rendimiento eficiente en proyectos de diversa magnitud.

También se discutirán las proyecciones futuras de SignalR en términos de su integración con tecnologías emergentes y cómo estas pueden influir en su evolución. Finalmente, se identificarán los principales desafíos que la actualización continua de las tecnologías de mercado presenta para SignalR.

Este estudio pretende proporcionar una comprensión profunda de SignalR, ofreciendo una perspectiva clara de su relevancia y aplicabilidad en el desarrollo de soluciones web modernas.

# **Situación Actual**

En un principio, con el nacimiento de la web moderna, los usuarios enviaban una petición al servidor y se esperaba de vuelta una respuesta para que se pueda actualizar la página y poder visualizar la respuesta a nuestra solicitud. Durante muchos años la comunicación en la web se efectuó de este modo y a partir de las demandas de los usuarios esto comenzó a cambiar. Es aquí donde surge la web moderna en la que los usuarios pueden mantener una interacción fluida entre sí generando feedback comunicacional a través de una aplicación web de modo dinámico, obteniendo en tiempo real una respuesta sincrónica a su solicitud enviada.

SignalR surge frente a la necesidad de comunicarse en tiempo real a través de la web, para mejorar la experiencia del usuario, dado que tecnologías desarrolladas anteriormente, como por ejemplo Comet, permitían la transmisión de datos del servidor al cliente pero de manera asíncrona, brindando actualización del contenido sin correspondencia temporal con la petición del usuario. Además, estas tecnologías son limitadas en cuánto a su implementación, dada su complejidad, su escalabilidad y compatibilidad con diversos navegadores.

Esta tecnología marcó un antes y un después en avances tecnológicos en el área del desarrollo de la comunicación web, ya que introdujo un modelo de programación en eventos que brinda a los desarrolladores la oportunidad de escalar y crear sus aplicaciones web y que estas puedan ser distribuídas, con carga de trabajo variable, interactivas, colaborativas, inmersivas y receptivas permitiendo la implementación en tiempo real de la comunicación cliente - servidor.

A diagram of a network

Description automatically generated

*"Figura 1: Integración de SignalR con .NET Core."*

# **¿Qué es SignalR?**

SignalR es una biblioteca tecnológica de software libre y de código abierto para Microsoft ASP.NET, de fácil utilización, que simplifica la incorporación de funciones web en tiempo real a las aplicaciones. Esta herramienta es especialmente útil para desarrollos que requieren una comunicación rápida y bidireccional. SignalR gestiona automáticamente las conexiones, incluyendo la apertura, el cierre y la reconexión, en caso de fallos, facilitando a los desarrolladores centrarse en la lógica de la aplicación en lugar de en la infraestructura subyacente.

Adicionalmente, esta herramienta facilita la comunicación masiva al permitir el envío de mensajes a todos los clientes conectados de manera simultánea, lo cual es conveniente para apps destinadas a la difusión. Asimismo, ofrece la capacidad de dirigir mensajes a clientes específicos o a grupos seleccionados, lo que resulta ideal para el envío de notificaciones personalizadas.

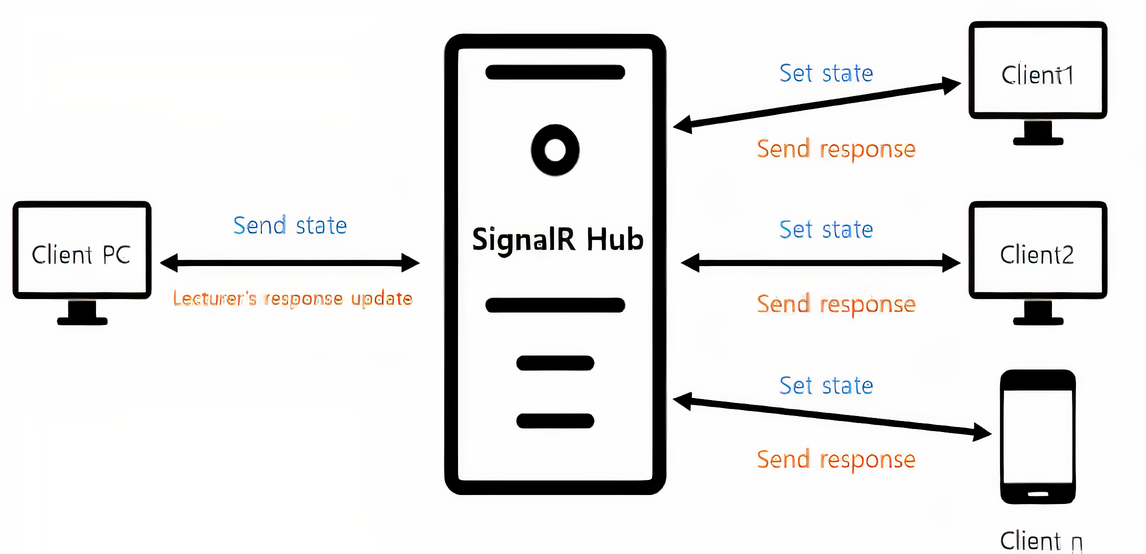
La capacidad de escalar eficientemente para manejar aumentos de tráfico en aplicaciones distribuidas, utilizando técnicas como múltiples servidores y servicios de mensajería escalables, asegura un rendimiento óptimo incluso con un gran número de usuarios. SignalR proporciona una API para llamadas a procedimientos remotos (RPC) de servidor a cliente, facilitando una comunicación inmediata mediante la estructura del Hub que actúa como intermediario. Compatible con varias plataformas, SignalR permite que las aplicaciones en tiempo real funcionen de manera consistente en diferentes entornos y dispositivos, utilizando lenguajes de programación apropiados para cada plataforma, como JavaScript para aplicaciones web y Swift o Kotlin para aplicaciones móviles. Ejemplos de su uso incluyen plataformas online como Trello y Microsoft Teams, monitoreo y control de dispositivos, y actualización de estados de videojuegos en línea.

Es pertinente destacar que el *Remote Procedure Call* (RPC) es un protocolo que permite a una computadora ejecutar código en otra máquina remota, administrando la comunicación de manera que el código parece ejecutarse localmente. Este enfoque representa un avance significativo sobre los antiguos sockets de Internet, ya que encapsula los detalles de la comunicación, liberando al desarrollador de la necesidad de manejarlos directamente.

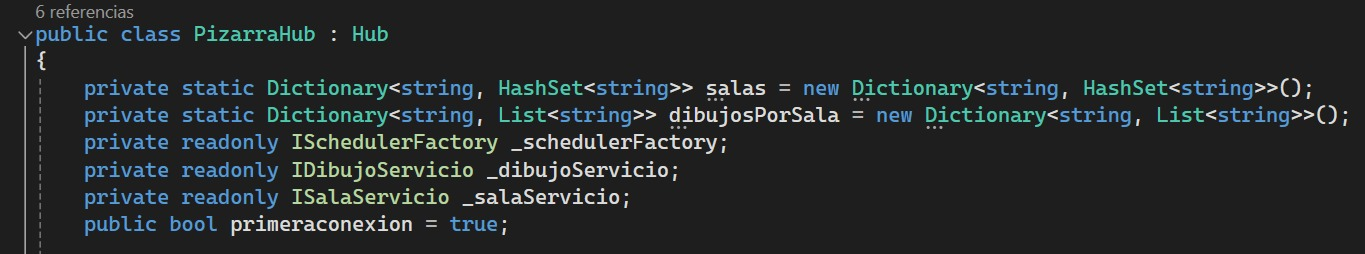
**Arquitectura de SignalR**

SignalR posee una arquitectura flexible que consta de componentes clave creados para simplificar y favorecer la comunicación sincrónica entre cliente y servidor. Los principales componentes de esta arquitectura son:

* **Hubs:** Son esenciales para la comunicación en tiempo real, funcionan como centros que agrupan diversos métodos. Los hubs actúan como el principal punto de entrada para las interacciones entre cliente y servidor, permitiendo la invocación de métodos desde ambas direcciones. Ofrecen una capa de abstracción que simplifica la complejidad inherente a estas interacciones, facilitando así la implementación de lógica de aplicación dinámica y en tiempo real. Esta estructura permite a los desarrolladores concentrarse más en la funcionalidad de la aplicación, mejorando la eficiencia y la claridad en el desarrollo de soluciones interactivas.

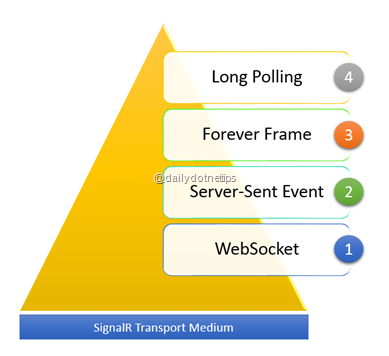


*"Figura 2: Diagrama de flujo de comunicación en SignalR: Interacciones entre el hub y clientes múltiples."*



*"Figura 3: Ejemplo de utilización de Hub en el Proyecto Pizarra SignalR."*

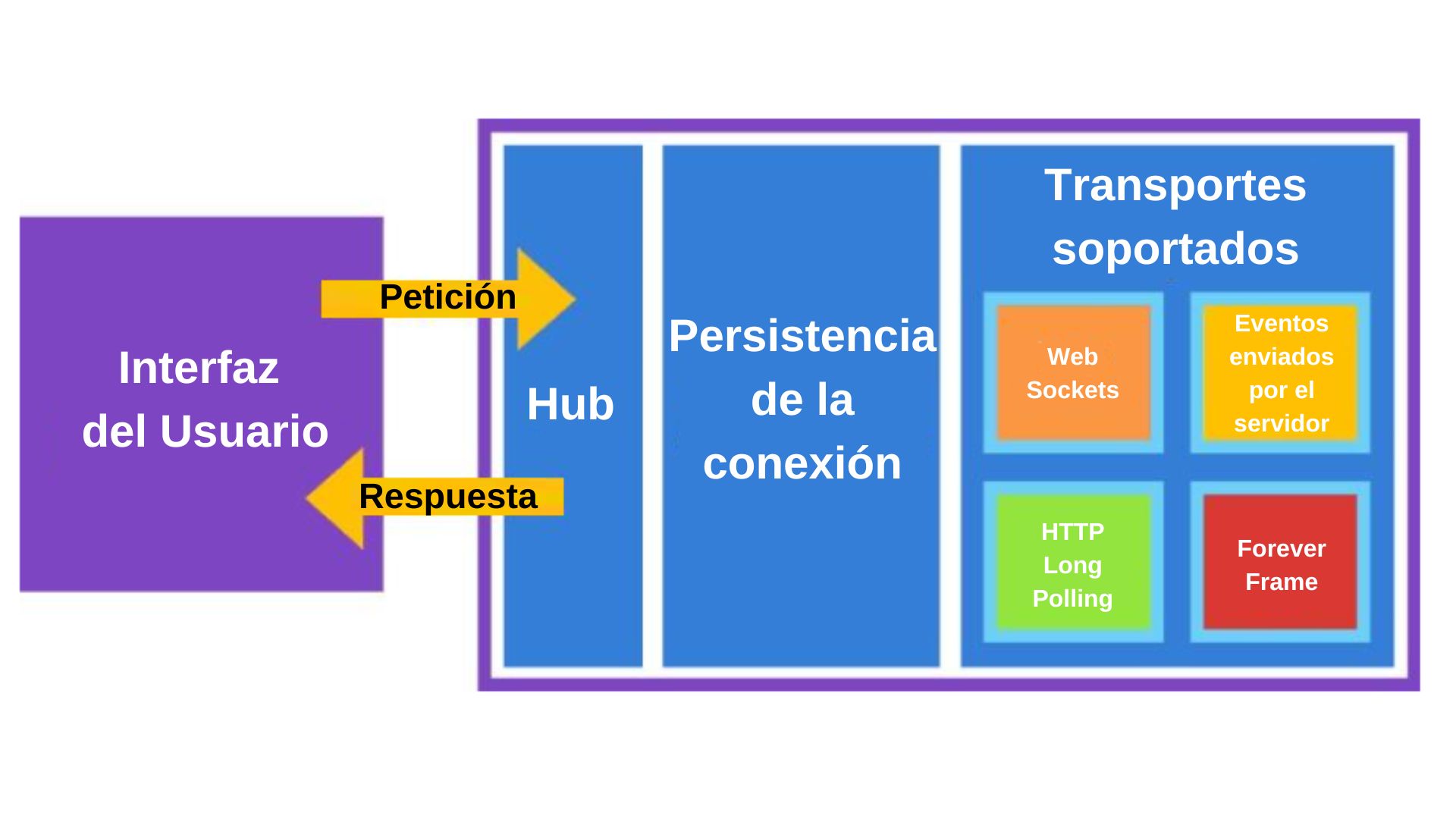
* **Conexiones:** Las conexiones en SignalR representan el vínculo entre cliente y servidor, y son establecidas cuando el cliente se conecta al servidor mediante una aplicación SignalR que crea una conexión única. Estas conexiones pueden mantenerse activas por períodos prolongados de tiempo, facilitando una comunicación sincrónica y bidireccional en tiempo real, donde ambas partes intercambian información de manera continua. SignalR automatiza la gestión de estas conexiones, manejando aspectos como la detección y resolución de desconexiones para asegurar una interacción fluida y confiable, mejorando así la experiencia del usuario.
* **Transportes:** SignalR utiliza diversos métodos de transporte para facilitar la comunicación entre cliente y servidor, incluyendo Server-Sent Events (SSE), Long Polling y Forever Frame. Sin embargo, WebSocket es preferido por su capacidad de ofrecer una conexión de baja latencia, lo que resulta en una comunicación más rápida y eficiente. Además, SignalR está equipado con funcionalidades de reconexión automática para mantener la estabilidad de la conexión ante interrupciones de red o cambios en el estado de la conexión.

****

*"Figura 4: Jerarquía de Transportes en SignalR: Comparativa de Eficiencia.”*

# **Flujo de comunicación cliente - servidor**

El flujo comunicacional en SignalR inicia cuando un cliente se conecta y establece una conexión con el servidor, conectándose a uno o más hubs para recibir actualizaciones en tiempo real. Tanto el cliente como el servidor pueden enviar y recibir mensajes, lo que permite una interacción continua y en tiempo real a través de los hubs.



*"Figura 5: Estructura de Comunicación en SignalR: Interfaz de Usuario, Hub y Transportes Soportados."*

# **Integración con ASP.NET y otros marcos de trabajo**

SignalR, integrado con el marco de trabajo web ASP.NET de Microsoft, proporciona una solución eficaz para el desarrollo de aplicaciones web dinámicas. Es especialmente útil para apps que necesitan actualizaciones frecuentes del servidor, tales como juegos en línea, redes sociales, y aplicaciones de votación o subastas. Además, es ideal para sistemas de navegación y monitoreo en tiempo real, así como para desarrollar softwares que requieran notificaciones, incluyendo alertas de viaje. SignalR también facilita la creación de plataformas de comunicación como chats y herramientas colaborativas, incluyendo pizarras virtuales y software de reuniones.

La incorporación de SignalR en proyectos web existentes que emplean ASP.NET se realiza sin contratiempos, permitiendo a los desarrolladores integrar funcionalidades, manteniendo las capacidades ya existentes de ASP.NET. Además de su compatibilidad con ASP.NET, SignalR también se extiende a otros marcos y plataformas como .NET Core, Xamarin y JavaScript, ofreciendo una amplia gama de posibilidades y permitiendo su implementación en diversos entornos de desarrollo. Esta versatilidad convierte a SignalR en una herramienta preferida por desarrolladores que buscan mejorar la interactividad y la capacidad de respuesta de sus soluciones de software.

# **Características clave de SignalR**

* **Comunicación Bidireccional en tiempo real:**SignalR proporciona capacidades avanzadas para establecer una comunicación bidireccional en tiempo real entre el cliente y el servidor. Esta funcionalidad permite a ambos actores iniciar interacciones y transmitir mensajes de manera sincronizada y fluida, lo que es esencial para aplicaciones que demandan una respuesta inmediata y constante intercambio de información.
* **Escalabilidad y balanceo de cargas:** SignalR está estratégicamente diseñado para facilitar la escalabilidad horizontal, permitiendo manejar un alto volumen de conexiones simultáneas sin degradar el rendimiento. Utiliza un enfoque de "escala hacia afuera", es decir, distribuye la carga de trabajo de manera eficiente entre varias instancias del servidor, lo que permite añadir más servidores según sea necesario en lugar de sobrecargar un solo servidor, esto posibilita el manejo de un alto volumen de conexiones simultáneas sin sobrecargar ningún servidor individual.

Adicionalmente, esta herramienta incluye configuraciones avanzadas para el balanceo de carga, asegurando una distribución equitativa de las solicitudes entre todos los servidores disponibles, lo cual optimiza la capacidad de respuesta y la estabilidad del sistema.

* **Compatibilidad Multiplataforma:** SignalR destaca por su compatibilidad multiplataforma, facilitando la comunicación en tiempo real entre dispositivos y plataformas variadas, desde aplicaciones web y móviles hasta soluciones de escritorio. Esta característica es esencial para el desarrollo de aplicaciones cohesivas y conectadas que pueden interactuar con una audiencia diversa, sin que se vea afectada la calidad de la experiencia del usuario. SignalR proporciona bibliotecas cliente específicas para cada plataforma, permitiendo a los desarrolladores integrar funcionalidades de comunicación en tiempo real de manera sencilla. Estas bibliotecas están disponibles no solo para entornos web como JavaScript y TypeScript, sino también para plataformas móviles como Android e iOS, y plataformas de escritorio como .NET y Java. Esta versatilidad asegura una interoperabilidad extensa y facilita el desarrollo en diferentes entornos de aplicación, potenciando así la universalidad y accesibilidad de las aplicaciones desarrolladas con SignalR.

A diagram of different applications

Description automatically generated with medium confidence

*"Figura 6: Arquitectura de Aplicaciones Multiplataforma con .NET 6."*

# **SignalR - Casos de uso**

# **Aplicaciones de chat: Mensajería y colaboración en tiempo real**

SignalR es fundamental en el desarrollo de aplicaciones de chat que requieren interacción instantánea y colaboración en tiempo real entre múltiples usuarios. Esta biblioteca asegura una conexión continua y eficiente entre el servidor y los clientes, facilitando la transmisión inmediata de mensajes y actualizaciones. Ejemplos destacados incluyen aplicaciones de mensajería instantánea como WhatsApp y Telegram, que dependen de la rapidez y fiabilidad de SignalR para proporcionar una experiencia de usuario óptima. Además, se emplea en entornos empresariales en plataformas como Slack y Microsoft Teams, donde permite a equipos compartir recursos y comunicarse eficazmente. La capacidad de SignalR para manejar notificaciones instantáneas y sincronizar datos en múltiples dispositivos la convierte en una solución versátil para plataformas que buscan interacciones fluidas y dinámicas en comunidades, foros y eventos de streaming en vivo.

A screenshot of a chat room

Description automatically generated

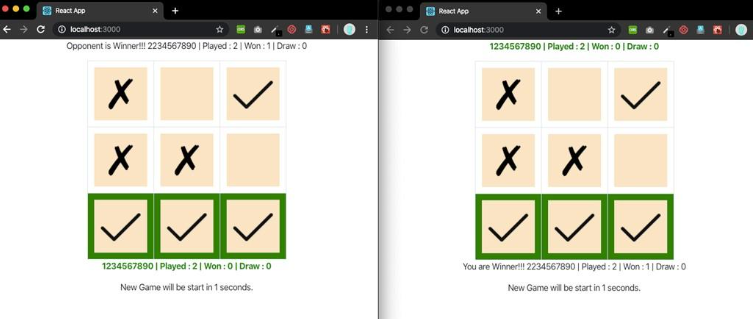
*"Figura 7: Mensajería de chat en tiempo real utilizando SignalR."*

# **Actualizaciones de datos en directo: Cotizaciones del mercado de valores, resultados deportivos, apuestas en línea**

SignalR es esencial en aplicaciones que requieren actualizaciones de datos en tiempo real, como las cotizaciones del mercado de valores, resultados deportivos y apuestas en línea. Esta tecnología permite la transmisión continua de información crucial, ayudando a los usuarios a mantenerse al día con eventos significativos y facilitando decisiones rápidas e informadas. En el sector financiero, SignalR es clave para actualizar instantáneamente los precios de las acciones, lo que es vital para los traders que responden a los cambios del mercado. En el ámbito deportivo, facilita la entrega de resultados en tiempo real, estadísticas y comentarios detallados, enriqueciendo la experiencia de los aficionados. Además, en las apuestas en línea, asegura que las cuotas se mantengan actualizadas, permitiendo a los apostadores realizar apuestas basadas en la información más reciente.

# **Juegos en línea: Interacciones y sincronización multijugador**

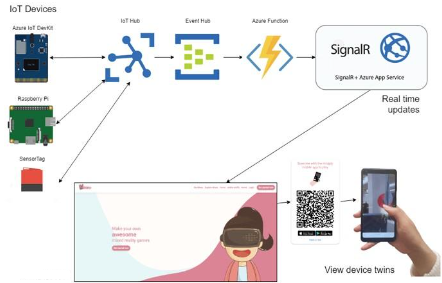
SignalR es fundamental en el desarrollo de juegos en línea, desde juegos de mesa tradicionales hasta juegos de acción en tiempo real, facilitando la sincronización del estado del juego y la comunicación entre múltiples jugadores. En juegos de mesa, SignalR asegura que movimientos como el desplazamiento de fichas o cartas se actualicen instantáneamente en todas las pantallas, manteniendo el flujo del juego fluido y sin interrupciones. En juegos de acción, como shooters multijugador o juegos de rol, gestiona la transmisión de datos vitales —posiciones de personajes, ataques y actualizaciones del entorno— asegurando una experiencia de juego coherente y sincronizada para todos los jugadores. Además, la capacidad de SignalR para facilitar la comunicación en tiempo real potencia la interacción social dentro del juego, permitiendo a los jugadores chatear, colaborar y coordinar estrategias efectivamente, enriqueciendo la experiencia de juego y fortaleciendo la comunidad de jugadores.



*"Figura 8: Interfaz de Usuario de un Juego de Tic-Tac-Toe desarrollado con SignalR, mostrando resultados en tiempo real para dos jugadores."*

# **IoT: Monitoreo y control remoto de sensores**

SignalR se emplea eficazmente en aplicaciones de IoT para el monitoreo y control remoto de sensores y dispositivos. Esta tecnología facilita la transmisión en tiempo real de datos desde sensores a aplicaciones, permitiendo el seguimiento continuo de variables como temperatura, humedad y presión. Los datos se transmiten instantáneamente a paneles de control o apps móviles, permitiendo a los usuarios analizar la información sin demoras. SignalR también habilita el control remoto de dispositivos IoT, permitiendo a los usuarios ajustar sistemas como la iluminación o la calefacción de hogares y supervisar maquinaria en entornos industriales desde cualquier lugar. Su capacidad para sostener comunicaciones bidireccionales en tiempo real lo convierte en una herramienta clave para desarrollar soluciones IoT avanzadas y mejorar la eficiencia operativa.

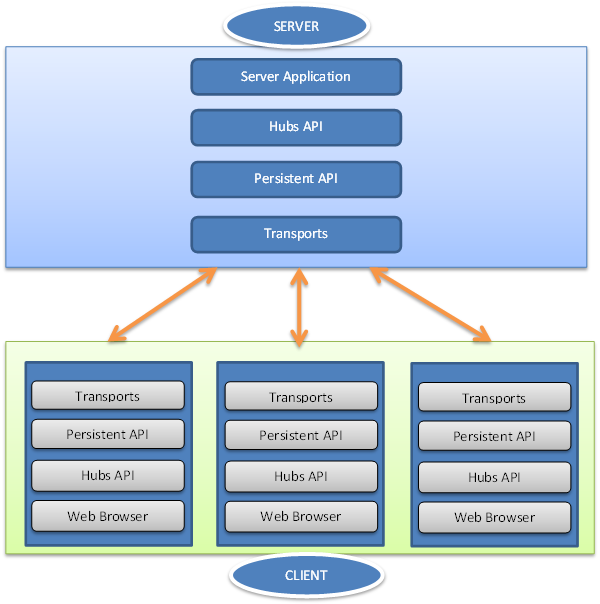


*"Figura 9: Diagrama de la arquitectura de una solución IoT integrada con SignalR para actualizaciones en tiempo real."*

# **Consideraciones de implementación**

# **Elección del mecanismo de transporte apropiado**

La selección del mecanismo de transporte es crítica para el rendimiento de aplicaciones que utilizan SignalR. WebSocket es generalmente preferido por su baja latencia y eficiencia, manteniendo conexiones bidireccionales persistentes. No obstante, en entornos donde WebSocket no es compatible, SignalR ofrece alternativas como Server-Sent Events (SSE) para comunicación unidireccional eficiente, y Long Polling, que garantiza compatibilidad universal a costa de menor eficiencia. La elección del transporte debe basarse en las necesidades específicas de la aplicación y las características del entorno de despliegue.

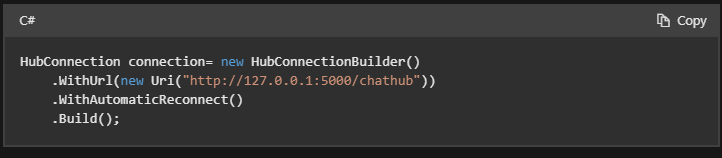


*"Figura 10: Arquitectura de Comunicación en SignalR.*

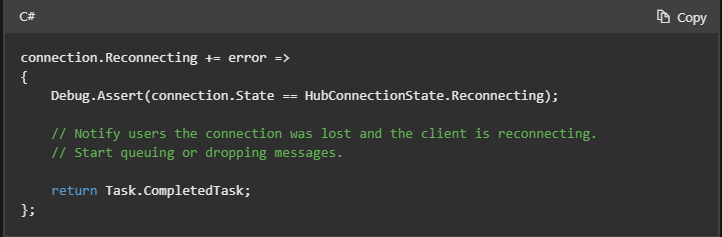
*Integración de los Diferentes Mecanismos de Transporte entre el Servidor y los Clientes."*

# **Manejo del ciclo de vida de la conexión: Gestión de conexiones y estrategias de reconexión**

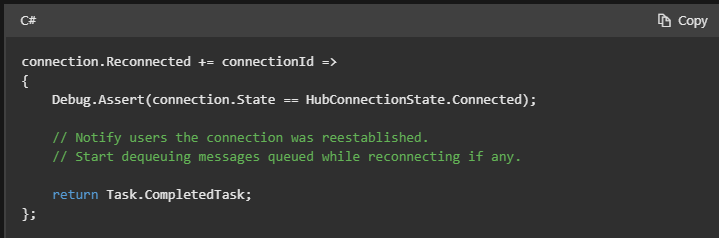
SignalR mejora la robustez y resiliencia de las aplicaciones mediante una gestión eficiente del ciclo de vida de la conexión, que incluye la apertura, cierre y monitoreo de desconexiones. Es crucial implementar estrategias de reconexión para asegurar una experiencia de usuario continua, particularmente en entornos con conectividad fluctuante. SignalR proporciona mecanismos automáticos de reconexión que pueden ser configurados por los desarrolladores para definir la cantidad de intentos y el intervalo entre estos. Adicionalmente, permite la integración de lógica personalizada para manejar la reconexión, facilitando acciones como notificaciones al usuario o procedimientos específicos al restablecer la conexión.

**

*“Figura 11: Ejemplo de manejo automático de reconexión.”*

**

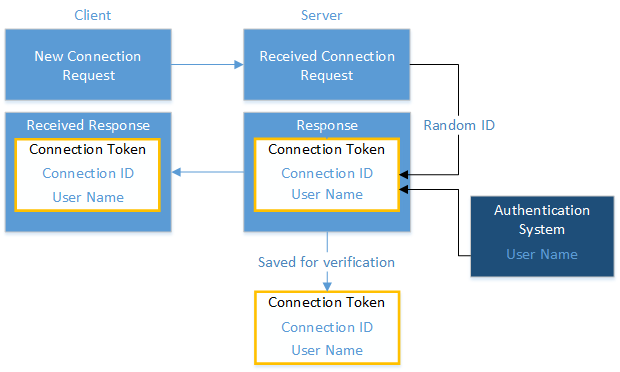
*“Figura 12: Estado del cliente luego de sufrir problemas con la conexión y activación del evento Reconnecting.”*

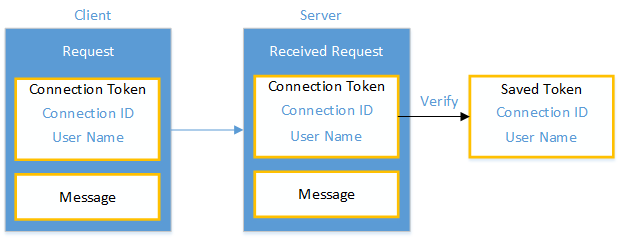
**

*“Figura 13: Reconexión exitosa del cliente.”*

# **Consideraciones de seguridad: Autenticación, Autorización y Cifrado de datos**

La seguridad es esencial en las aplicaciones que emplean SignalR, necesitando la implementación de mecanismos robustos de autenticación y autorización para garantizar el acceso exclusivo a usuarios autorizados. SignalR facilita su integración con los sistemas de autenticación de ASP.NET Core, permitiendo el uso de cookies, tokens JWT (JSON Web Tokens) y otros estándares de autenticación. Además, se pueden establecer políticas de autorización para controlar el acceso a los hubs y sus métodos específicos. Es crucial también cifrar los datos transmitidos entre cliente y servidor para proteger la información sensible. Utilizando HTTPS, SignalR asegura una comunicación segura, mitigando potenciales riesgos de interceptaciones o ataques como CSRF.





*“Figura 14: Uso y validación de tokens de conexión en el marco de SignalR Security.”*

# **Técnicas de optimización de rendimiento: Caché, Comprensión y Minimización de latencia**

Para asegurar un funcionamiento eficiente de una aplicación SignalR bajo condiciones de carga elevada, es vital adoptar estrategias avanzadas de optimización de rendimiento. El uso de técnicas de caché minimiza la carga en el servidor y acelera los tiempos de respuesta al almacenar datos de acceso frecuente en memoria. Además, la compresión de datos mediante tecnologías como Gzip reduce el tamaño de los mensajes intercambiados, disminuyendo el consumo de ancho de banda y mejorando la velocidad de transmisión.

Es crucial minimizar la latencia, especialmente en aplicaciones en tiempo real; esto se puede lograr optimizando la infraestructura de red, por ejemplo, ubicando servidores cerca de los usuarios finales a través de redes de entrega de contenido (CDN), gestionando eficientemente el tráfico de mensajes y afinando el código del servidor para respuestas rápidas. Implementar estas técnicas asegura que la aplicación permanezca responsiva y eficiente, incluso bajo altas demandas.

**Futuras Direcciones y Desafíos de SignalR**

# **Tendencias Emergentes en la Comunicación Web en Tiempo Real**

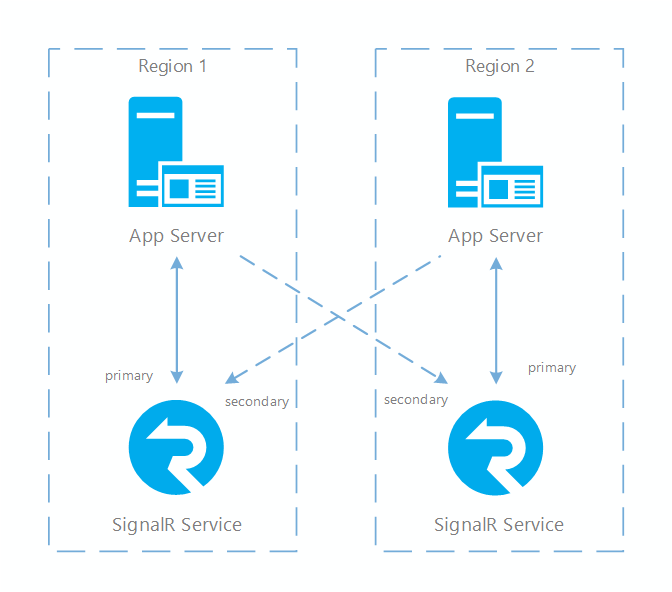
SignalR está bien posicionado para capitalizar las tendencias emergentes en comunicación web en tiempo real. Con la creciente demanda de aplicaciones más rápidas y responsivas, tecnologías como WebSocket y Server-Sent Events se destacan por su baja latencia y conexiones persistentes. La adopción de estándares como HTTP/2 y WebRTC podría potenciar la eficiencia y seguridad de SignalR, abriendo oportunidades en IoT y plataformas de streaming en tiempo real.

# **Potencializando la Tecnología SignalR**

La evolución tecnológica señala áreas clave para la mejora de SignalR, incluyendo la optimización del rendimiento y la gestión eficiente de recursos, especialmente en entornos de gran escala. Fortalecer la gestión de conexiones y las estrategias de reconexión aumentaría su robustez y fiabilidad. La implementación de funcionalidades avanzadas de seguridad, como autenticación multifactor y cifrado end-to-end, consolidaría a SignalR como una solución segura para aplicaciones críticas.

# **Desafíos de Escalabilidad en Implementaciones a Gran Escala**

La escalabilidad sigue siendo un desafío para SignalR en contextos de alto volumen. Es esencial desarrollar arquitecturas que soporten una gran cantidad de usuarios y conexiones, utilizando técnicas como la separación de capas, balanceadores de carga y almacenamiento en memoria distribuida. La colaboración con servicios gestionados en la nube, como Azure SignalR Service, ofrece soluciones escalables que manejan la demanda variable y garantizan la alta disponibilidad.

****

*“Figura 15: Distribución de la carga y redundancia del servicio SignalR entre múltiples regiones.”*

# **Conclusiones**

SignalR se consolida como una herramienta imprescindible en el ámbito del desarrollo web moderno, proporcionando una solución eficaz para la implementación de comunicaciones en tiempo real entre servidores y clientes. Esta biblioteca se distingue por su habilidad para manejar conexiones automáticas, transmitir mensajes de manera eficiente a múltiples clientes y escalar adecuadamente para soportar cargas fluctuantes, lo que la convierte en una opción robusta para proyectos de cualquier magnitud.

Es crucial seleccionar con precisión los mecanismos de transporte adecuados para cada aplicación, tales como WebSocket, Server-Sent Events y Long Polling, adaptándolos a los requerimientos específicos del proyecto. Además, es recomendable una gestión proactiva del ciclo de vida de las conexiones, implementar medidas de seguridad sólidas y aplicar técnicas de optimización de rendimiento para asegurar la estabilidad, la protección de los datos y una experiencia de usuario superior.

Mirando hacia el futuro, SignalR muestra un potencial significativo para integrarse con tecnologías emergentes como HTTP2 y WebRTC, y anticipa mejoras continuas en seguridad y escalabilidad. Estas evoluciones prometen fortalecer aún más su posición como una solución vital para el desarrollo de aplicaciones web innovadoras y la implementación de comunicación en tiempo real avanzada.

En resumen, ASP.NET Core SignalR representa una herramienta poderosa y versátil que facilita la creación de experiencias de usuario dinámicas y responsivas. Su capacidad para gestionar eficazmente la comunicación en tiempo real y la escalabilidad de aplicaciones la establece como una opción destacada para una amplia variedad de escenarios de desarrollo, consolidando su relevancia en la industria del desarrollo web.

# 

# **Bibliografía**

1. Microsoft. (s.f.). *Diferencias de versión en SignalR*. Recuperado de<https://learn.microsoft.com/es-es/aspnet/core/signalr/version-differences?view=aspnetcore-8.0&source=recommendations>
2. Wikipedia. (s.f.). *Llamada a procedimiento remoto*. Recuperado de la página de Wikipedia en [https://es.wikipedia.org/wiki/Llamada\_a\_procedimiento\_remoto].
3. c-sharpcorner.com. (s.f.). *Hub Architecture*. [Imagen]. Recuperado de [c-sharpcorner.com].
4. dailydotnettips.com. (s.f.). *Identifying the SignalR Transport Medium*. Recuperado de [dailydotnettips.com].
5. c-sharpcorner.com. (s.f.). *What, Why and How About SignalR*. Recuperado de [c-sharpcorner.com].