



Explorando la evolución del lenguaje de programación Dart

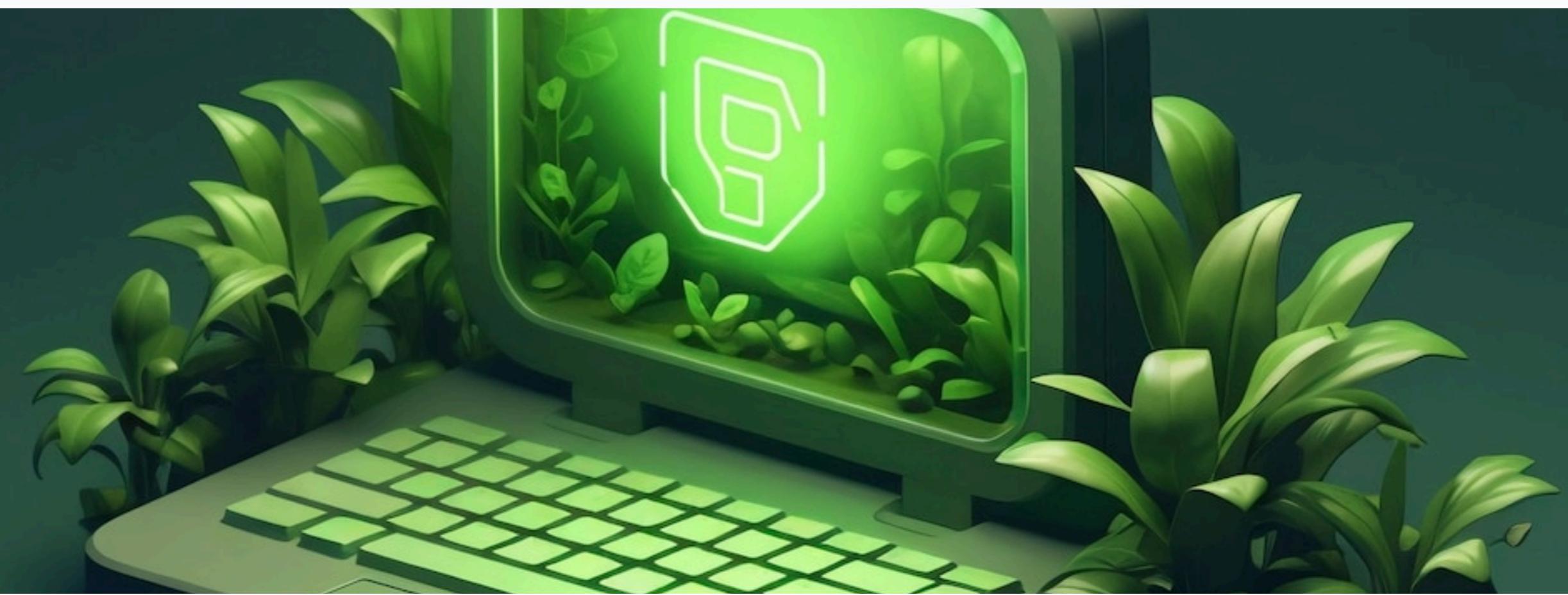
INTRODUCCIÓN

Bienvenidos a la exploración de la **evolución del lenguaje de programación Dart**. En esta presentación, analizaremos el impacto de Dart en el mundo de la programación y su desarrollo a lo largo del tiempo.



ORÍGENES DE DART

Los **orígenes de Dart** se remontan a 2011, cuando fue presentado por Google como un lenguaje de programación de código abierto. Su objetivo era ofrecer una alternativa eficiente para el desarrollo web y de aplicaciones.

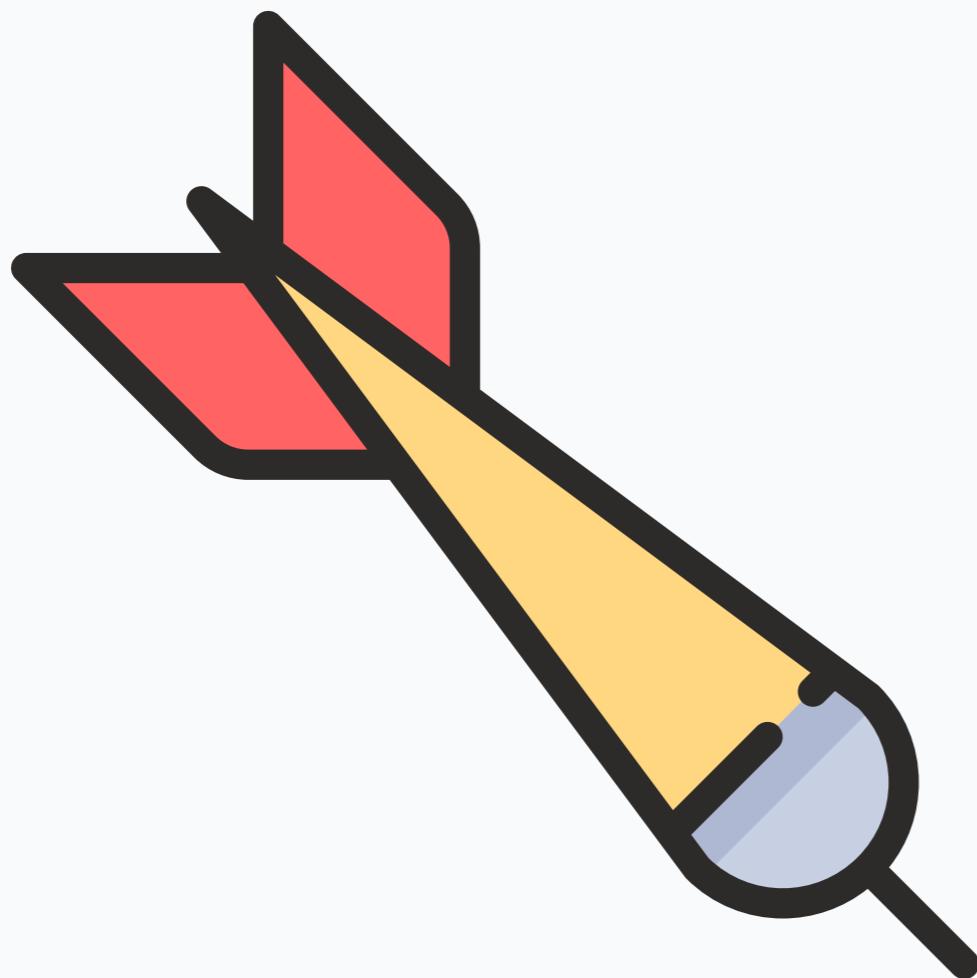


CARACTERÍSTICAS PRINCIPALES

Dart se destaca por su **sintaxis clara** y su capacidad para compilar a código nativo. Además, su soporte para programación orientada a objetos y su sistema de tipado estático lo hacen atractivo para desarrolladores.



CARACTERÍSTICAS PRINCIPALES



Tipado fuerte y seguro: Dart es un lenguaje de programación de tipado fuerte, lo que significa que cada variable tiene un tipo específico asociado y no se puede cambiar dinámicamente.

Soporte para programación orientada a objetos y funcional: Dart combina lo mejor de ambos mundos.

Hot Reload: Esta característica permite la visualización instantánea de cambios en el código sin reiniciar toda la aplicación.

CARACTERÍSTICAS PRINCIPALES

Flexibilidad: Dart fue diseñado para ser familiar y cómodo para los programadores, independientemente de su experiencia previa con otros lenguajes¹. Además, Dart puede ejecutarse en múltiples plataformas, desde aplicaciones móviles hasta aplicaciones de escritorio¹.

Máquina virtual (VM): Un rasgo distintivo de Dart es su máquina virtual (VM), que permite una ejecución rápida y un desarrollo más fluido.

Herramientas integradas: Dart incluye múltiples componentes, como son las librerías y su propio editor, así como una máquina virtual que tiene la capacidad de llevar a cabo la ejecución del código Dart.



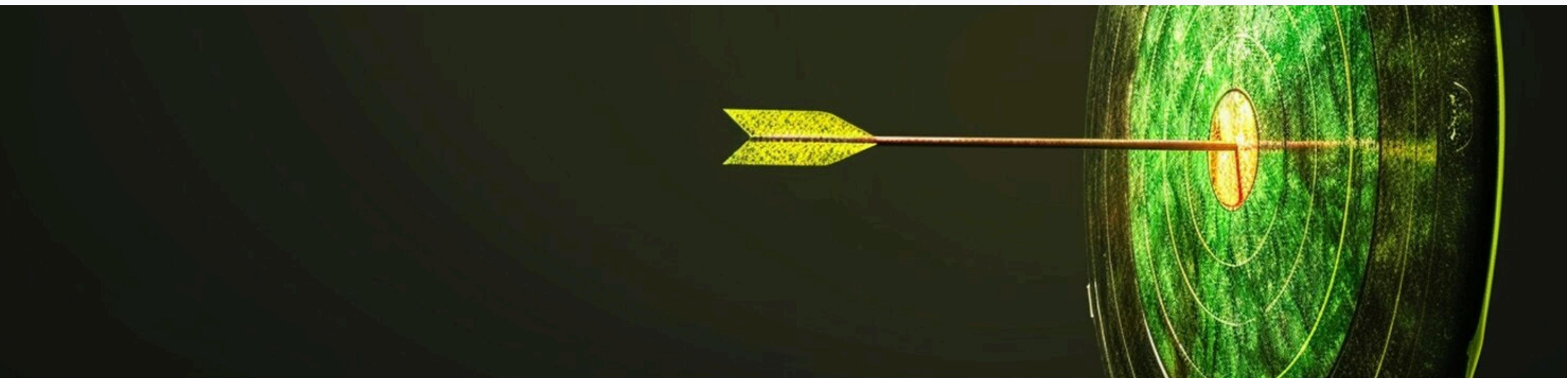
ADOPCIÓN EN LA INDUSTRIA

La **adopción de Dart** ha ido en aumento, especialmente en el desarrollo de aplicaciones móviles y web. Grandes empresas como Google, Alibaba y others están utilizando Dart para sus proyectos.



EVOLUCIÓN DEL ECOSISTEMA

El **ecosistema de Dart** ha evolucionado con el tiempo, con la introducción de herramientas como Flutter para desarrollo de interfaces de usuario. Esta evolución ha ampliado su alcance y utilidad.



DART 2 Y MÁS ALLÁ

Con el lanzamiento de **Dart 2**, se introdujeron mejoras significativas en rendimiento y usabilidad. Además, se ha mantenido un enfoque en la innovación continua para futuras versiones de Dart.



DART 3

Aquí te presento algunas de las mejoras más destacadas:

100% Verificación de nulos (100% Sound null safety): A partir de Dart 2.12, se introdujo una nueva característica llamada verificación de nulos (null safety). Esta característica tiene como objetivo principal mejorar la seguridad y la estabilidad del código al proporcionar verificaciones de nulos más sólidas. Si intentas asignar un valor nulo a una variable no nula, el compilador generará un error en tiempo de compilación. Esto ayuda a reducir la posibilidad de errores relacionados con nulos y a mejorar la robustez del código. Ahora en Dart 3 el lenguaje ahora cuenta con un 100% de verificación de nulos.



DART 3

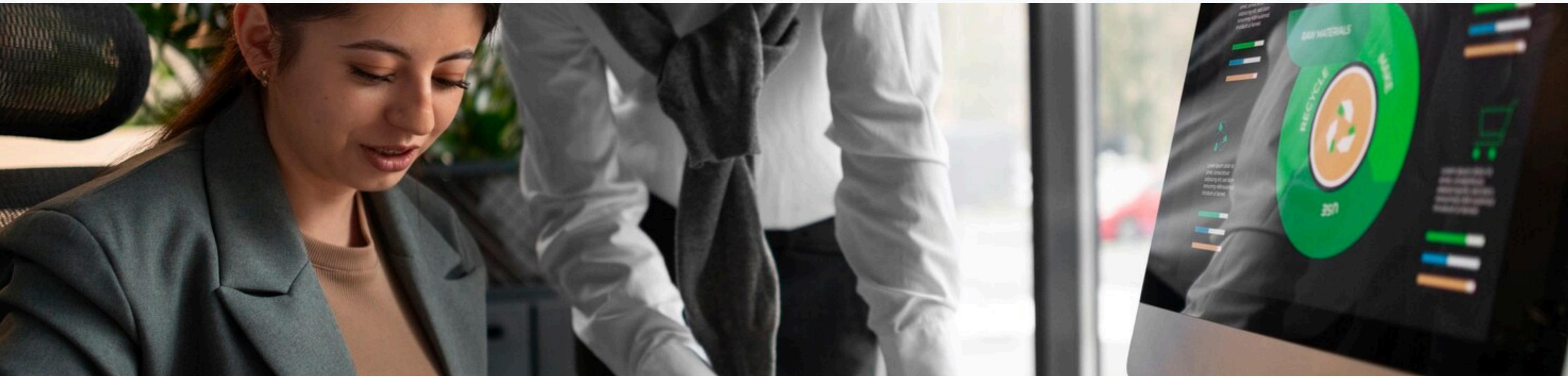
Records: Una nueva característica de Dart 3 son los Records que permiten que un solo objeto pueda contener varios objetos. Un caso de uso es cuando queremos regresar dos o más valores de una función.

Patterns: Retomando el ejemplo anterior, un problema al usar records es que al no crear la clase Usuario no vamos a escribir usuario.nombre para acceder al nombre, sino que vamos a escribir usuario.\$1, y esto hace el código más difícil de comprender. Pero utilizando patterns podemos volver a escribir un código fácil de comprender.



COMUNIDAD Y COLABORACIÓN

La **comunidad de Dart** ha crecido y se ha vuelto más activa, con contribuciones significativas de desarrolladores de todo el mundo. La colaboración abierta ha enriquecido el ecosistema de Dart.



DESAFÍOS Y OPORTUNIDADES

A pesar de su crecimiento, Dart enfrenta **desafíos en su adopción** a gran escala. Sin embargo, las oportunidades en el desarrollo web y de aplicaciones siguen siendo prometedoras para Dart.



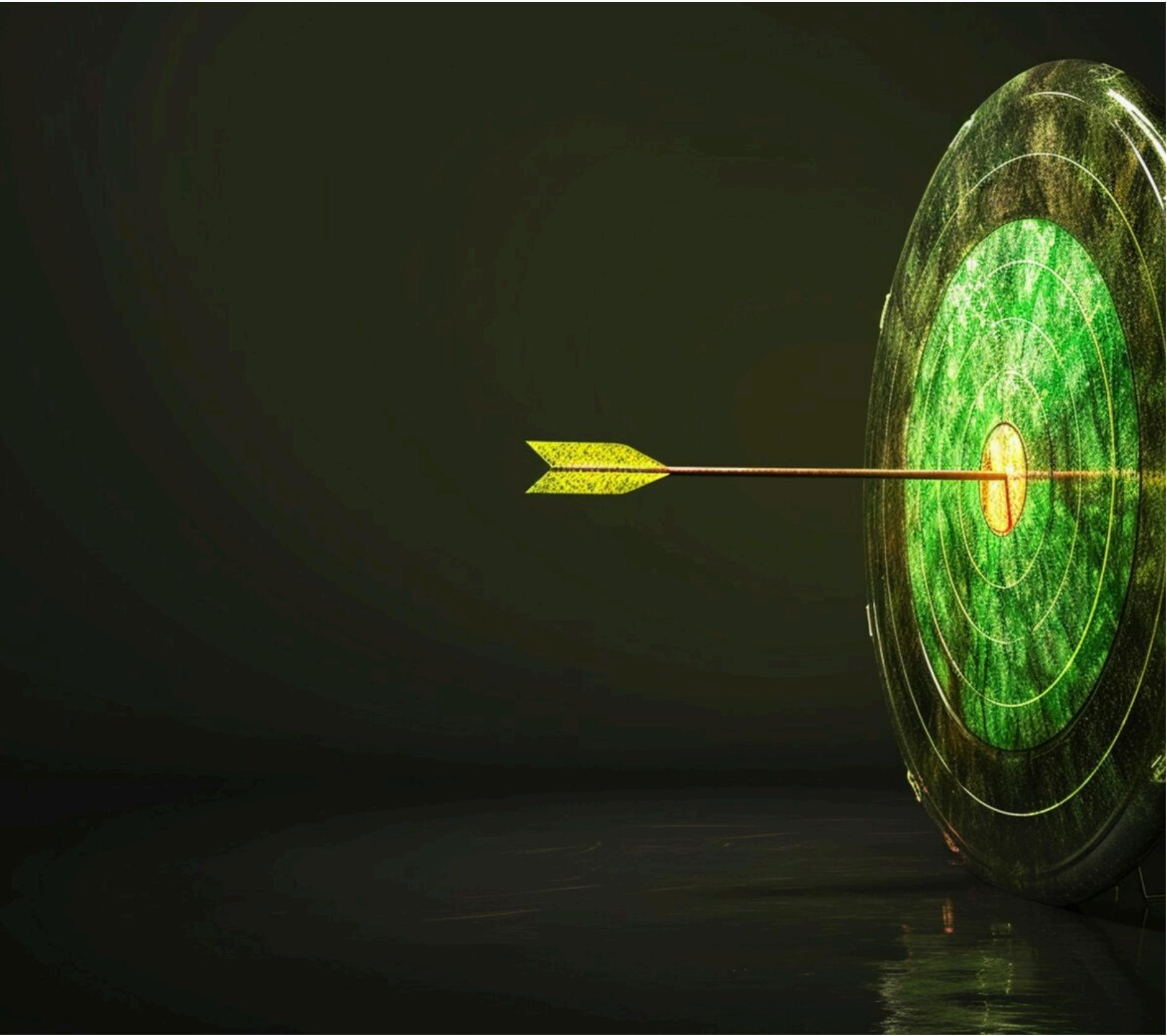
OPTIMIZACIÓN DE RENDIMIENTO

La **optimización de rendimiento** es un área clave de desarrollo para Dart, con un enfoque en la mejora de la velocidad de ejecución y la eficiencia del código generado.



FUTURO DE DART

El **futuro de Dart** se vislumbra prometedor, con un enfoque en la expansión de su uso en el desarrollo de aplicaciones web y móviles. La continua innovación y colaboración impulsarán su evolución.



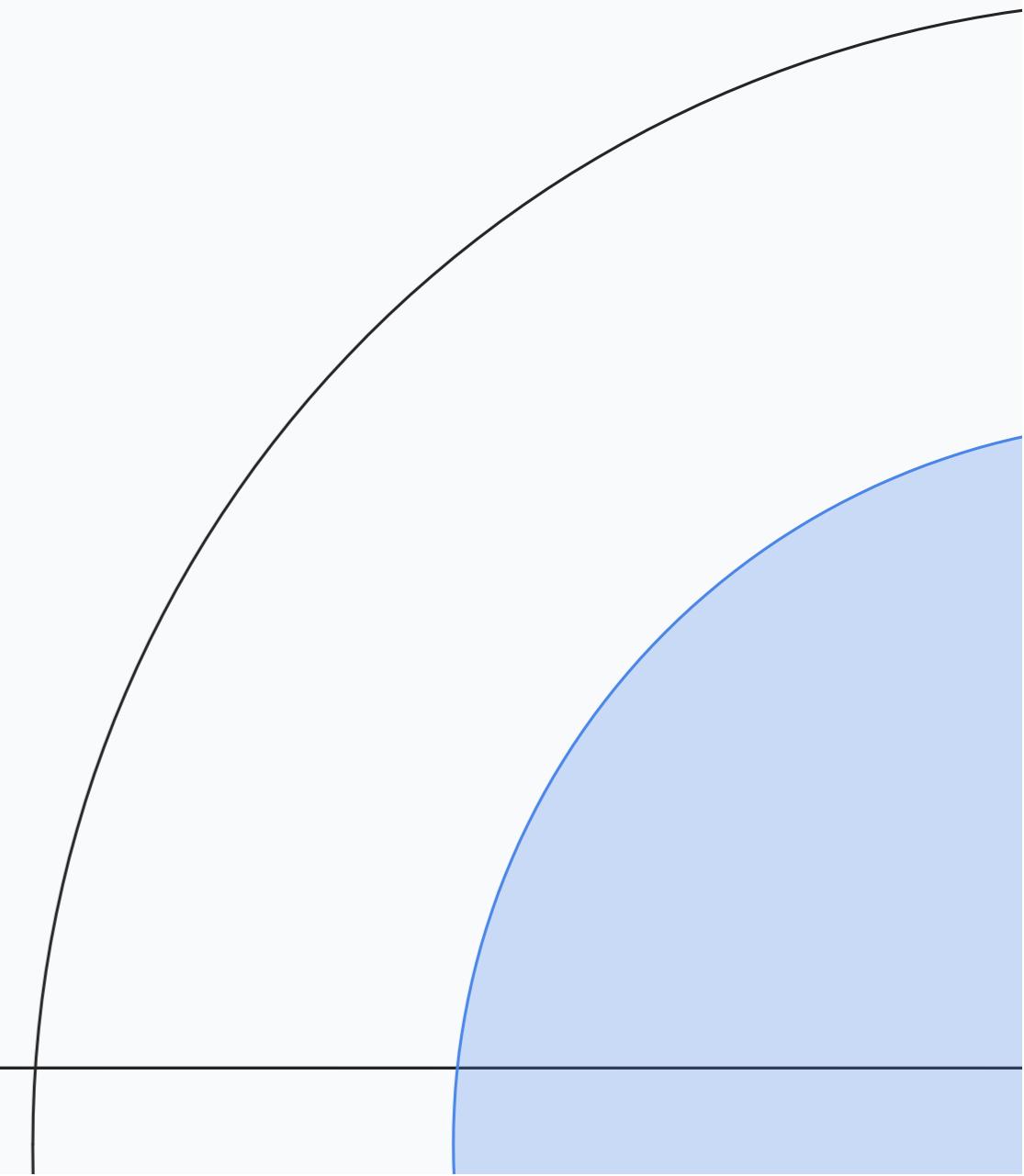
CONCLUSIONES

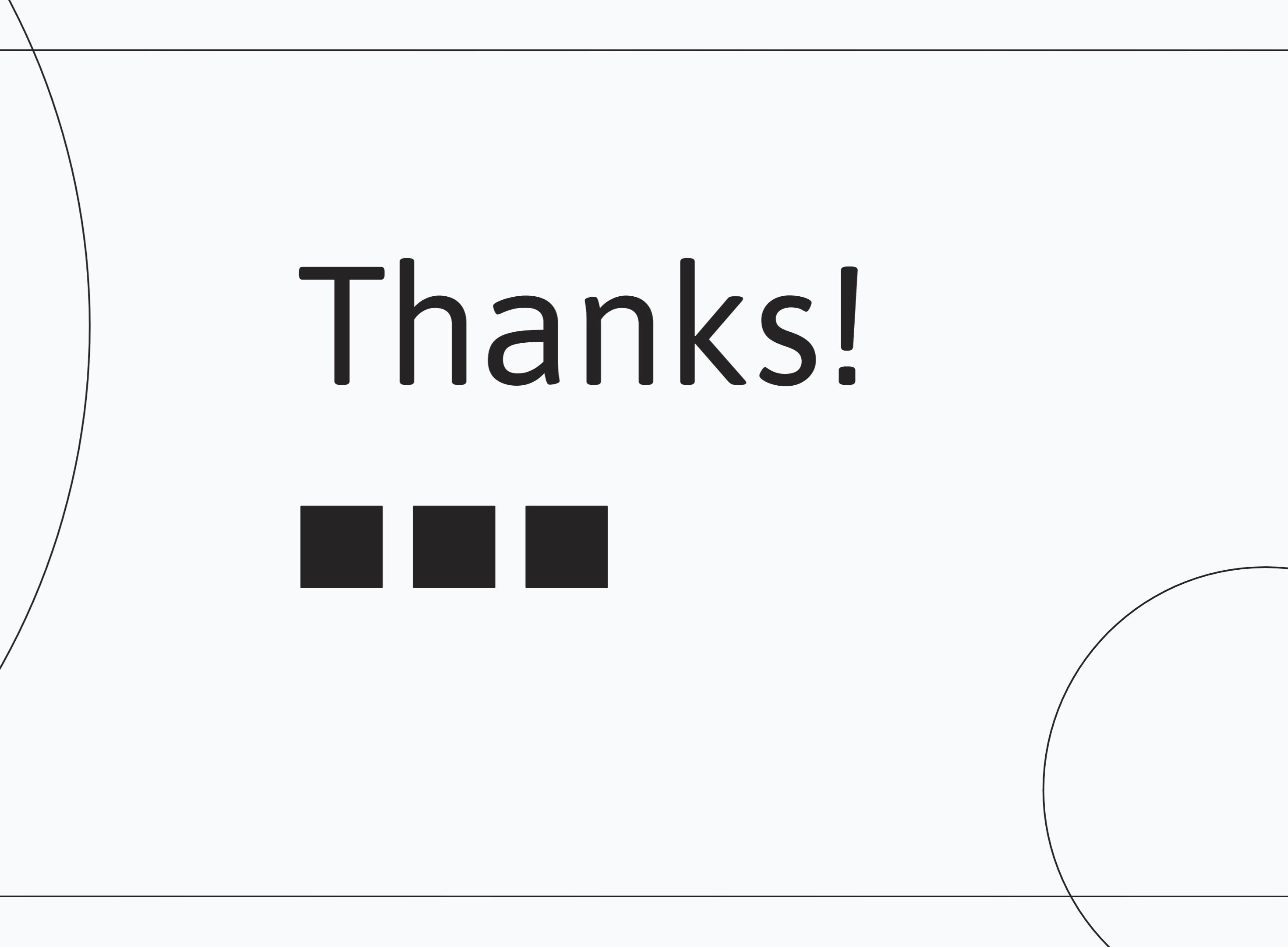
En resumen, la **evolución del lenguaje de programación Dart** ha sido notable, con avances significativos en su rendimiento, usabilidad y adopción. Su futuro promete seguir impactando el mundo del desarrollo de software.



PREGUNTAS

¿Tienen alguna pregunta sobre la evolución del lenguaje de programación Dart? ¡Estoy aquí para responderlas!





Thanks!

