



# Unidad 4 - Procesamiento de video en tiempo real

**UNR - TUIA - Procesamiento de Imágenes y Visión por Computadora**

Docente teoría: Juan Pablo Manson

Docentes práctica: Lucas Brugé, Constantino Ferrucci

Cuaderno de práctica:

Google Colab

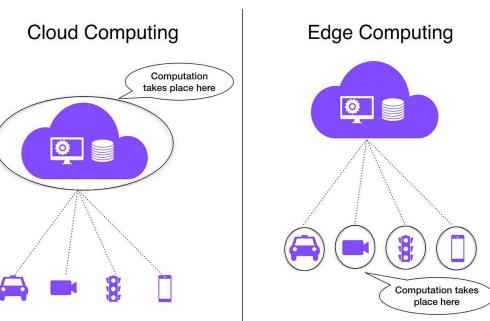
🔗 [https://colab.research.google.com/drive/1SfWMRTDIEs35RvoemUfGk1ke9IN\\_zOwC?usp=sharing](https://colab.research.google.com/drive/1SfWMRTDIEs35RvoemUfGk1ke9IN_zOwC?usp=sharing)



## 1. Edge computing

¿Qué es el Edge Computing?

El edge computing (computación en el borde) es el concepto de capturar y procesar datos tan cerca de su fuente o usuario final como sea posible. Típicamente, la fuente de los datos es un dispositivo del internet de las cosas (IoT). El procesamiento se realiza localmente colocando poder computacional (hardware) cerca de la ubicación física de las fuentes de datos para procesar los datos.

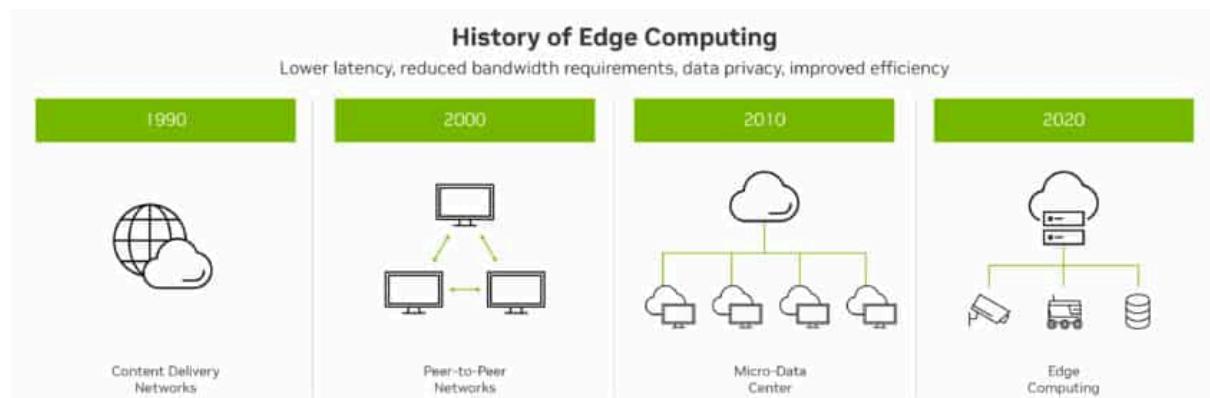


Dado que la computación en el borde procesa los datos localmente, en el borde de la red, en lugar de la nube o un centro de cómputos, se minimiza la latencia y los costos de tránsito de los datos, permitiendo una retroalimentación y toma de decisiones en tiempo real.

La retroalimentación instantánea y siempre activa que ofrece la computación en el borde es especialmente crítica para aplicaciones donde la seguridad humana es un factor. Por ejemplo, es crucial para los coches autónomos, donde ahorrar incluso milisegundos en el procesamiento de datos y los tiempos de respuesta puede ser clave para evitar accidentes. También es fundamental en hospitales, donde los médicos dependen de datos precisos en tiempo real para tratar a los pacientes.

Aunque la computación en el borde es particularmente importante para aplicaciones modernas como la ciencia de datos y el aprendizaje automático, también conocido como IA en el borde, no es un concepto nuevo.

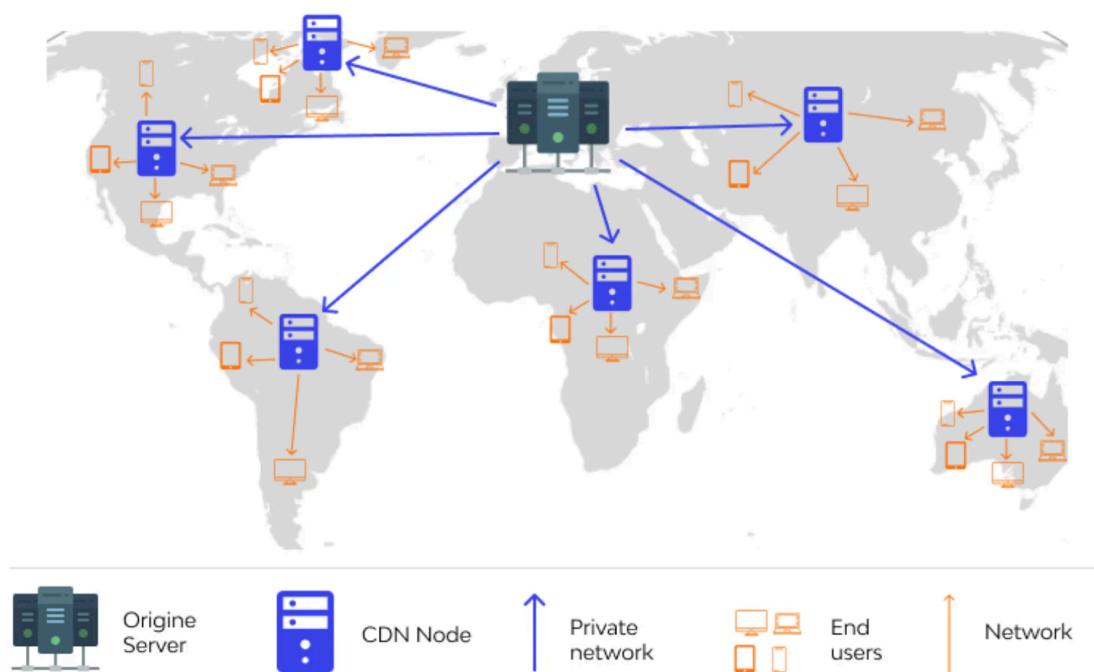
## La Historia de la Computación en el Borde



La computación en el borde se remonta a la década de 1990, cuando las redes de entrega de contenido (CDNs) actuaban como centros de datos distribuidos. En ese momento, las CDNs se limitaban a almacenar en caché imágenes y videos, no grandes cargas de trabajo de datos.

Para la década de 2000, la explosión de dispositivos inteligentes tensionó la infraestructura de TI existente. Sin embargo, inventos como las redes peer-to-peer (P2P), donde las computadoras están conectadas y comparten recursos sin pasar por un servidor centralizado, alivieron la tensión.

A mediados de la década de 2000, grandes empresas comenzaron a alquilar recursos de computación y almacenamiento de datos a los usuarios finales a través de nubes públicas. A medida que las aplicaciones basadas en la nube y las empresas que trabajan desde muchas ubicaciones crecían en popularidad, procesar datos de la manera más eficiente posible se volvió cada vez más importante.



Todas estas tecnologías han llevado a nuestra forma actual de computación en el borde, en la que los nodos en el borde tienen la capacidad de ofrecer acceso de baja latencia a recursos e información intensivos en datos. Estas capacidades se construyeron sobre principios de las habilidades de baja latencia de las CDNs, la plataforma descentralizada de las redes P2P y la

escalabilidad y resiliencia de la nube. Juntas, estas tecnologías han creado un marco de computación más eficiente, resiliente y confiable.

### **¿Cómo funciona la Computación en el Borde?**

La computación en el borde funciona procesando datos tan cerca de su fuente o del dispositivo del usuario final como sea posible. Mantiene los datos, las aplicaciones y la potencia de computación alejados de una red o centro de datos centralizados.

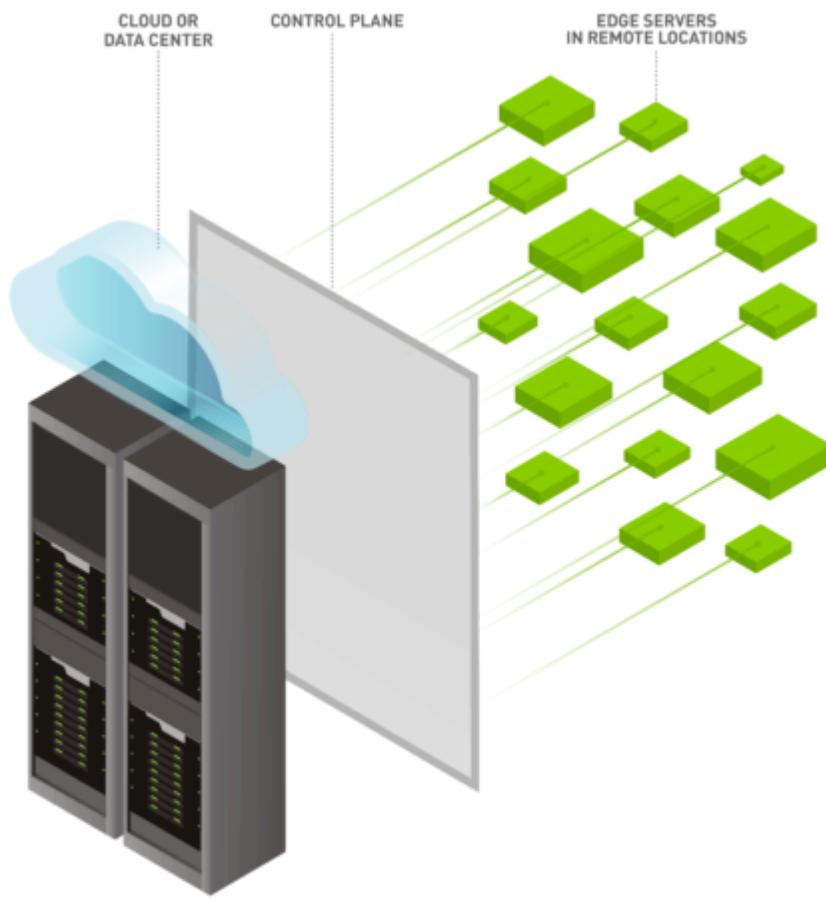
Tradicionalmente, los datos producidos por los sensores a menudo son revisados manualmente por humanos, dejados sin procesar o enviados a la nube o a un centro de datos para su procesamiento, y luego devueltos al dispositivo. Confiar únicamente en revisiones manuales resulta en procesos lentos e inefficientes. Y aunque la computación en la nube proporciona recursos de computación, la transmisión y el procesamiento de datos pone una gran tensión en el ancho de banda y la latencia.

El ancho de banda es la tasa a la que se transfieren datos a través de internet. Cuando los datos se envían a la nube, viajan a través de una red de área amplia, lo cual puede ser costoso debido a su cobertura global y altas necesidades de ancho de banda. Al procesar datos en el borde, se pueden utilizar redes de área local, resultando en un mayor ancho de banda a menores costos.

La latencia es el retraso en el envío de información de un punto a otro; afecta los tiempos de respuesta. Se reduce cuando se procesa en el borde porque los datos producidos por sensores y dispositivos IoT ya no necesitan ser enviados a una nube centralizada para ser procesados.

Al llevar la computación al borde, o más cerca de la fuente de datos, se reduce la latencia y se incrementa el ancho de banda, resultando en información y acciones más rápidas.

La computación en el borde puede ejecutarse en uno o varios servidores para cerrar la distancia entre donde se recopilan y procesan los datos, reduciendo cuellos de botella y acelerando las aplicaciones. Una infraestructura de borde ideal también implica una plataforma de software centralizada que puede gestionar remotamente todos los sistemas de borde en una sola interfaz.

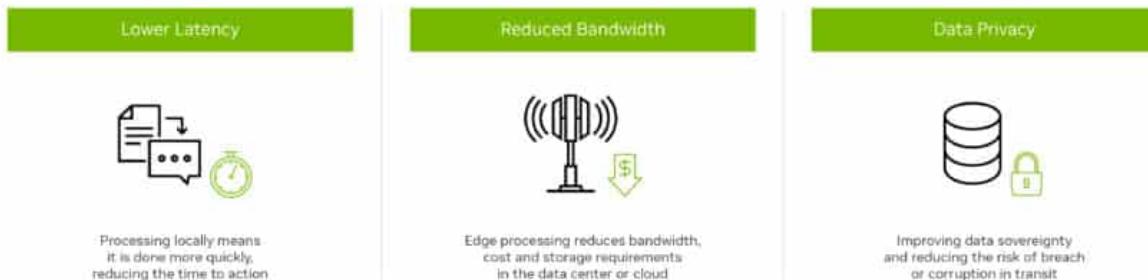


## **¿Por qué la Computación en el Borde? ¿Cuáles son los Beneficios de la Computación en el Borde?**

El cambio hacia la computación en el borde ofrece a las empresas nuevas oportunidades para obtener conocimientos a partir de sus grandes conjuntos de datos. Los principales beneficios de la computación en el borde son:

- **Menor latencia:** La latencia se reduce cuando se procesa en el borde porque los datos producidos por sensores y dispositivos IoT ya no necesitan ser enviados a una nube centralizada para ser procesados.
- **Reducción del ancho de banda:** Cuando los datos se envían a la nube, viajan a través de una red de área amplia, lo cual puede ser costoso debido a su cobertura global y altas necesidades de ancho de banda. Al procesar datos en el borde, se pueden utilizar redes de área local, resultando en un mayor ancho de banda a menores costos.
- **Soberanía de los datos:** Cuando los datos se procesan en el lugar donde se recopilan, la computación en el borde permite a las organizaciones

mantener todos sus datos y procesamiento en una ubicación adecuada. Esto resulta en una menor exposición a ataques cibernéticos y en el cumplimiento de estrictas y cambiantes leyes de ubicación de datos.



Más allá de la computación en el borde, muchas compañías buscan beneficiarse de la inteligencia artificial en el borde (edge AI), que es la fusión de la computación en el borde y la IA.

### **¿Por qué se Necesita la Computación en el Borde?**

En los últimos años, la computación en el borde se ha vuelto cada vez más importante debido a la convergencia del IoT y el 5G. Estas tecnologías están creando casos de uso que requieren que las organizaciones consideren la computación en el borde.

Con la proliferación de dispositivos IoT llegó la generación de grandes volúmenes de datos. Las organizaciones que de repente comenzaron a recopilar datos de todos los aspectos de sus negocios se dieron cuenta de que sus aplicaciones no estaban diseñadas para manejar tales volúmenes de datos.

Además, se dieron cuenta de que la infraestructura para transferir, almacenar y procesar grandes volúmenes de datos puede ser extremadamente costosa y difícil de gestionar. Es por eso que solo una fracción de los datos recopilados de los dispositivos IoT se procesa alguna vez. En algunas situaciones, es tan bajo como el 25%.

Y el problema está aumentando. A medida que el número de dispositivos conectados crece y aumenta la cantidad de datos que necesitan ser transferidos, almacenados y procesados, las organizaciones están cambiando a la computación en el borde para aliviar los costos necesarios para usar los mismos datos en modelos de computación en la nube.

Las redes 5G, que pueden ser 10 veces más rápidas que las 4G, están diseñadas para permitir que cada nodo sirva a cientos de dispositivos en el

borde, aumentando las posibilidades de servicios habilitados por IA en ubicaciones en el borde.

Con la poderosa, rápida y confiable capacidad de procesamiento de la computación en el borde, las empresas tienen el potencial de explorar nuevas oportunidades comerciales, obtener información en tiempo real, aumentar la eficiencia operativa y mejorar sus experiencias de usuario.

### ¿Cuáles son los Tipos de Computación en el Borde?

La definición de computación en el borde es amplia. A menudo, se refiere a cualquier computación fuera de una nube o un centro de datos tradicional.



Aunque existen diferentes tipos de computación en el borde, tres categorías principales incluyen:

- **Provider Edge (Borde del proveedor):** El borde del proveedor es una red de recursos informáticos a la que se accede a través de internet. Se utiliza principalmente para ofrecer servicios de compañías de telecomunicaciones, proveedores de servicios, empresas de medios y otros operadores de redes de entrega de contenido (CDN).
- **Enterprise Edge (Borde empresarial):** El borde empresarial es una extensión del centro de datos empresarial, que consiste en centros de datos en sitios de oficinas remotas, micro centros de datos o incluso racks de servidores ubicados en un armario de computación en una fábrica. Al igual que con un centro de datos centralizado tradicional, este entorno generalmente es propiedad y está operado por TI. Sin embargo, puede haber limitaciones de espacio o energía en el borde empresarial que cambien el diseño de estos entornos.

- **Industrial Edge (Borde industrial):** El borde industrial también se conoce como el borde lejano. Generalmente abarca instancias de computación más pequeñas, como uno o dos pequeños servidores robustos en el borde, o incluso un sistema embebido desplegado fuera de un entorno de centro de datos. Debido a que funcionan fuera de un centro de datos normal, hay una serie de desafíos únicos en cuanto a espacio, refrigeración, seguridad y gestión.



Dispositivo industrial para vision 3D, con una computadora NVIDIA Jetson y una cámara industrial incorporada.

## ¿Cuáles son las Diferencias y Similitudes entre la Computación en el Borde y la Computación en la Nube?

La principal diferencia entre la computación en la nube y la computación en el borde es dónde se realiza el procesamiento. Para la computación en el borde, el procesamiento ocurre en el borde de una red, más cerca de la fuente de datos, mientras que para la computación en la nube, el procesamiento ocurre en el centro de datos.

El siguiente cuadro detalla las diferencias entre las dos tecnologías:

Computación en la Nube	Computación en el Borde
Procesamiento de datos no sensibles a la latencia	Procesamiento de datos en tiempo real
Conexión a internet confiable	Ubicaciones remotas con conectividad limitada o sin conectividad a internet
Cargas de trabajo dinámicas	Grandes conjuntos de datos que son demasiado costosos de enviar a la nube
Datos en almacenamiento en la nube	Datos altamente sensibles y leyes estrictas de datos

La computación en el borde y en la nube tienen características distintas, y la mayoría de las organizaciones se benefician del uso de ambas. Una

arquitectura híbrida en la nube permite a las empresas aprovechar la seguridad y la capacidad de gestión de los sistemas locales, mientras también utilizan recursos de nube pública de un proveedor de servicios.

Las tecnologías nativas de la nube, como la “dockerización”, pueden ayudar a gestionar soluciones de computación en el borde. IDC predice que, para 2024, el 75% de las nuevas aplicaciones operativas desplegadas en el borde aprovecharán la dockerización para permitir una arquitectura más abierta y componible, necesaria para operaciones resilientes.

### | Fuente:

#### What Is Edge Computing?

Edge computing is the concept of capturing and processing data as close to its source or end user as possible.

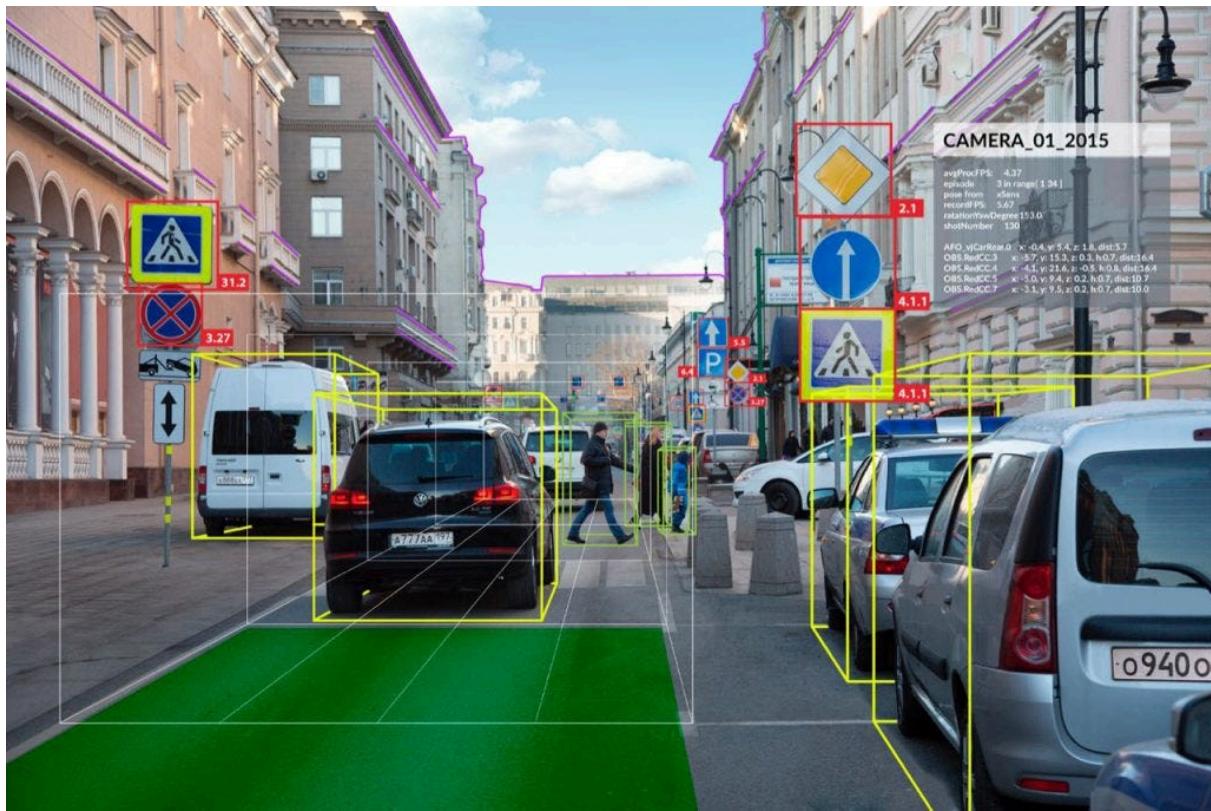
 <https://blogs.nvidia.com/blog/what-is-edge-computing/>



## Aplicaciones de Edge Computing basadas en Computer Vision

A continuación se enumeran algunos ejemplos de aplicación del Edge Computing:

### **Coches Autónomos**

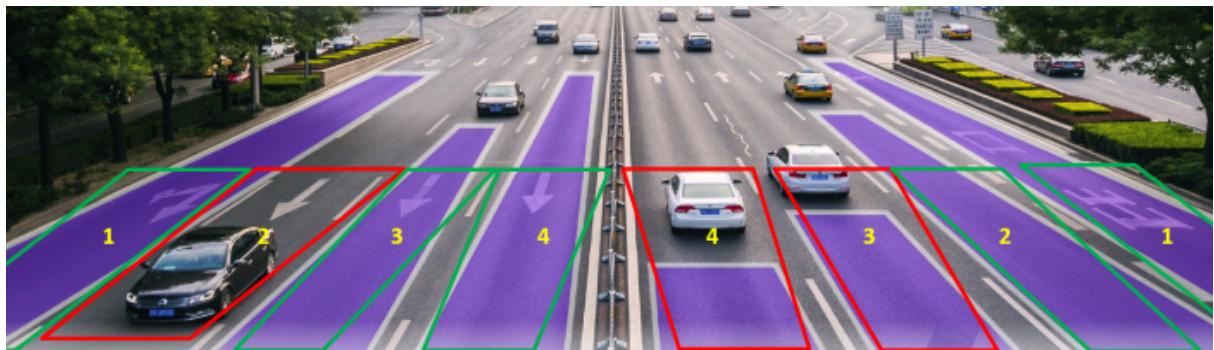


Los coches autónomos dependen en gran medida de la visión por computadora para tomar decisiones en tiempo real sobre la conducción. Estos vehículos recopilan y procesan datos de múltiples sensores y cámaras montados en el coche para:

- Detectar y reconocer objetos como otros vehículos, peatones, señales de tráfico y obstáculos en la carretera.
- Evaluar las condiciones de la carretera y el tráfico para ajustar la velocidad y la dirección.
- Realizar maniobras de seguridad instantáneamente para evitar accidentes.

El procesamiento de estos datos debe ser extremadamente rápido, con latencias mínimas, lo cual no es viable si los datos se envían a servidores remotos. Por ello, se utiliza la computación en el borde para procesar la información localmente en el vehículo.

### Monitoreo de Tráfico Vehicular



El monitoreo de tráfico vehicular a través de cámaras de vigilancia y sensores en las carreteras ayuda a:

- Controlar el flujo de tráfico y detectar congestiones.
- Identificar infracciones de tránsito y situaciones de emergencia.
- Gestionar la sincronización de semáforos en función del flujo de tráfico en tiempo real.

Enviar flujos de video continuos a servidores remotos para su procesamiento resultaría en una alta latencia y uso excesivo de ancho de banda. En cambio, la computación en el borde permite procesar los datos localmente, proporcionando información y alertas en tiempo real.

## Automatización Industrial



En la automatización industrial, la visión por computadora se utiliza para:

- Inspección de calidad de productos en líneas de producción.

- Identificación de defectos y control de procesos en tiempo real.
- Guiado de robots industriales y sistemas de ensamblaje automatizado.

El procesamiento local de imágenes y datos es esencial para mantener la eficiencia y precisión de los procesos industriales. La computación en el borde reduce la latencia y permite la toma de decisiones instantánea, fundamental en ambientes de manufactura rápida.

## Agricultura de Precisión



En la agricultura de precisión, la visión por computadora se utiliza para:

- Detectar y clasificar malezas en campos de cultivo.
- Aplicar herbicidas de manera precisa solo en áreas afectadas, reduciendo el uso de químicos.
- Monitorear la salud de las plantas y la calidad del suelo.

El procesamiento en el borde permite a los equipos agrícolas analizar imágenes y datos en el campo sin necesidad de conexión a internet, lo que es vital en áreas rurales donde la conectividad puede ser limitada.

## Exploración espacial



Curiosity Sol 1060 Aug 2015

Credit: NASA/JPL/MSSS/Ken Kremer/Marco Di Lorenzo

El Mars Rover utiliza la visión por computadora para:

- Navegar el terreno, evitando obstáculos y seleccionando rutas seguras.
- Analizar el entorno y realizar investigaciones científicas, como la identificación de rocas y minerales.
- Tomar decisiones autónomas en tiempo real, ya que la comunicación con la Tierra tiene una latencia significativa.

El procesamiento de datos en el borde es esencial debido a la imposibilidad de enviar continuamente datos a la Tierra para su análisis. El Rover debe ser capaz de procesar y actuar sobre los datos localmente para asegurar una exploración efectiva y segura.

## Dispositivos para Edge Computing

Los dispositivos de edge computing ofrecen una solución eficiente en términos de ancho de banda. En lugar de saturar las redes con grandes volúmenes de datos que deben ser procesados en la nube, estos dispositivos manejan los datos localmente. Esto no solo reduce los costos asociados con el uso del ancho de banda, sino que también permite un procesamiento continuo incluso en entornos con conectividad limitada, como zonas rurales o áreas remotas.

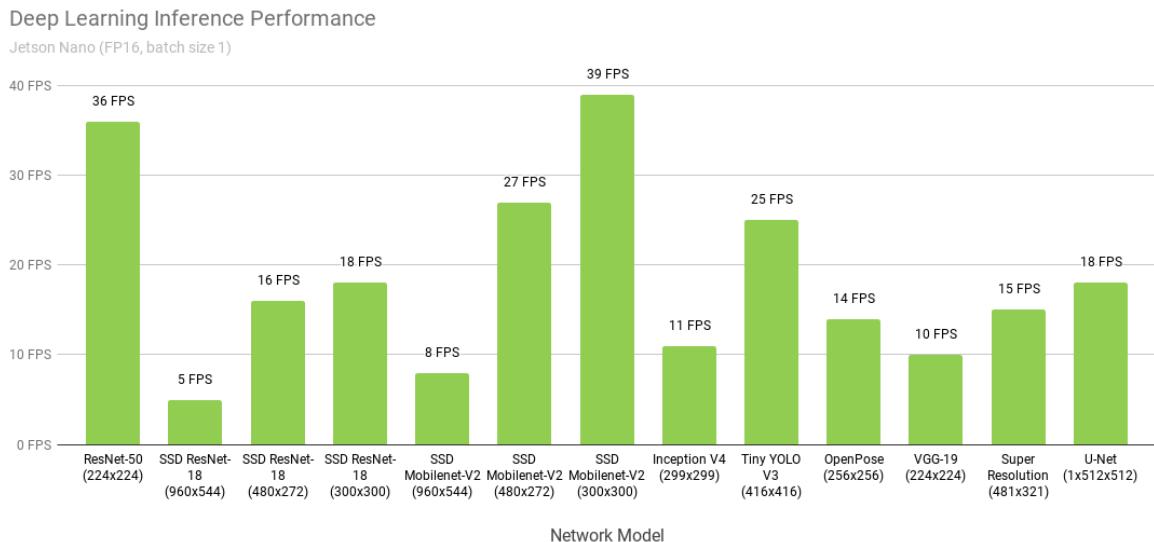
La capacidad de ejecutar modelos de deep learning directamente en el borde es fundamental. Los modelos de visión por computadora, que suelen ser complejos y exigentes en términos de recursos, pueden beneficiarse enormemente de la potencia de cómputo local proporcionada por dispositivos equipados con GPUs o TPUs. Esto permite realizar inferencias rápidas y precisas en tiempo real.



NVIDIA Jetson con una cámara industrial

### Características principales

- **Capacidad de Procesamiento:** La capacidad de procesamiento es fundamental para los dispositivos de edge computing destinados a la visión por computadora. Contar con GPUs potentes o múltiples núcleos de CPU permite ejecutar modelos de deep learning a altas tasas de fotogramas por segundo (FPS). Esto es fundamental para el procesamiento de video en tiempo real, necesario en aplicaciones como coches autónomos, donde las decisiones deben tomarse instantáneamente, o en monitoreo de tráfico, donde la detección y respuesta rápida a eventos es vital. Una alta capacidad de procesamiento garantiza que los modelos puedan operar con la precisión y velocidad requeridas sin retrasos.



Performance de diferentes modelos optimizados con TensorRT, corriendo en un dispositivo Jetson Nano

- Coneectividad con Cámaras:** La conectividad con diversas configuraciones de cámaras, como sistemas estéreo o cámaras de 360 grados, es esencial para una visión por computadora efectiva. Los dispositivos de edge computing deben ser capaces de integrar y sincronizar múltiples fuentes de video para proporcionar una visión completa y precisa del entorno. Esto es especialmente importante en aplicaciones como la automatización industrial, donde se puede necesitar una vista detallada y multifacética de un proceso, o en agricultura de precisión, donde la combinación de diferentes perspectivas puede mejorar la detección de malezas y la salud del cultivo.



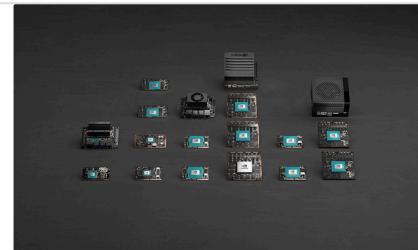
- **Soporte de frameworks de aceleración:** La compatibilidad con frameworks de aceleración como TensorRT y OpenVINO permite optimizar y acelerar la ejecución de modelos de deep learning. Estos frameworks están diseñados para aprovechar al máximo el hardware disponible, reduciendo los tiempos de inferencia y mejorando la eficiencia del procesamiento. Esta capacidad es crucial para aplicaciones que requieren procesamiento en tiempo real y alta eficiencia, asegurando que los modelos puedan operar de manera óptima en entornos con recursos limitados.
- **Comunicación:** Las capacidades de comunicación son vitales para la integración y operabilidad de los dispositivos de edge computing. La disponibilidad de múltiples opciones de comunicación, como Ethernet, WiFi o CAN-Bus, asegura que los dispositivos puedan conectarse y comunicarse eficazmente con otros sistemas y dispositivos en una red. En aplicaciones industriales, una conexión fiable y rápida es esencial para la transferencia de datos en tiempo real y la coordinación de diferentes sistemas. En vehículos autónomos, la comunicación rápida y segura entre sensores y sistemas de control es fundamental para una operación segura y eficiente.
- **Sistema Operativo y contenedores:** Un sistema operativo robusto y la compatibilidad con tecnologías de contenedorización como Docker o Kubernetes permiten una gestión y despliegue más eficiente de aplicaciones y servicios. Esto facilita la actualización y mantenimiento de los modelos de deep learning, así como la escalabilidad y flexibilidad de las soluciones implementadas. En entornos industriales o de investigación, donde las necesidades pueden cambiar rápidamente, la capacidad de adaptar y gestionar el software de manera eficiente es crucial para mantener la operatividad y la eficacia.
- **Energía y disipación:** La eficiencia energética y una gestión eficaz de la disipación de calor son críticas para la operatividad continua de los dispositivos de edge computing. Los dispositivos deben ser capaces de operar con un consumo de energía mínimo y gestionar el calor generado para evitar sobrecalentamientos y fallos. Esto es particularmente importante en aplicaciones móviles, como vehículos autónomos, donde la disponibilidad de energía puede ser limitada, y en entornos industriales, donde la fiabilidad y la duración operativa son esenciales.
- **Soporte del Fabricante y Actualizaciones:** El soporte continuo del fabricante y las actualizaciones regulares del software son esenciales para

mantener la seguridad y funcionalidad de los dispositivos de edge computing. Un buen soporte asegura que cualquier problema técnico se resuelva rápidamente y que los dispositivos se mantengan al día con las últimas mejoras y parches de seguridad. Esto es fundamental en entornos donde la seguridad y la operatividad constante son críticas, como en la salud, donde los datos y sistemas deben estar protegidos y operativos en todo momento, o en la industria, donde las interrupciones pueden resultar costosas.

#### NVIDIA Jetson Modules, Support, Ecosystem, and Lineup

Jetson modules help you build and manage AI on the edge, roll out new features, and deploy innovative products across industries.

 <https://developer.nvidia.com/embedded/jetson-modules>



#### Developer Kits for IoT

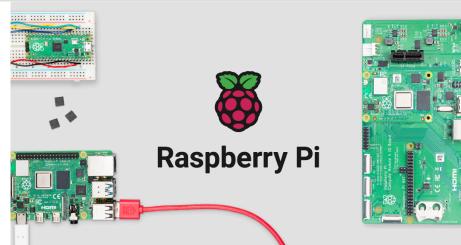
Evaluate the best IoT developer kit for your needs. Learn how to get started with your project.

 <https://www.intel.com/content/www/us/en/developer/tool-pc-technology/edge-5g/hardware/overview.html>

#### Raspberry Pi

From industries large and small, to the kitchen table tinkerer, to the classroom coder, we make computing accessible and affordable for everybody.

 <https://www.raspberrypi.com/>



#### Coral

Build intelligent ideas with our platform for local AI

 <https://coral.ai/>



## **2. Streaming, Video Encoding/Decoding.**

### **Formatos de video.**

#### **Codificación y Decodificación de imágenes**

Después de capturar el video con una cámara — ya sea un sistema de TV en vivo, una cámara IP o desde dispositivos móviles —, los datos de video deben digitalizarse para su transporte eficiente a través de internet. La codificación de video es esencial para la transmisión en vivo, ayudando a asegurar una entrega y reproducción rápidas.

##### **¿Qué es la Codificación (Encoding)?**

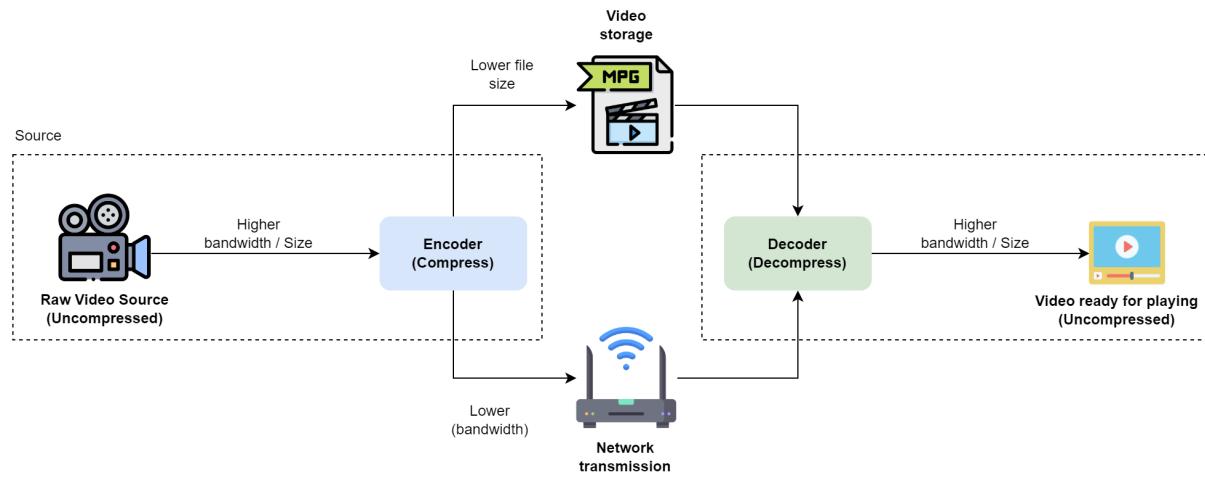
La codificación de video se refiere al proceso de convertir video crudo en un formato digital que sea compatible con muchos dispositivos. Los videos a menudo se reducen de gigabytes de datos a megabytes de datos. Este proceso involucra una herramienta de compresión de dos partes llamada codec.

##### **¿Qué es un Codec?**

Generalmente al referirse a codec, nos referimos a "codificador-decodificador" o "compresor-descompresor". Los codecs aplican algoritmos para comprimir de manera ajustada un video voluminoso para su entrega. El video se reduce para el almacenamiento y la transmisión, y más tarde se descomprime para su visualización.

Cuando se trata de transmisión, los codecs emplean compresión con pérdida descartando datos innecesarios para crear un archivo más pequeño. Se llevan a cabo dos procesos de compresión separados: video y audio. Los codecs de video actúan sobre los datos visuales, mientras que los codecs de audio actúan sobre el sonido grabado.

H.264, también conocido como AVC (Codificación de Video Avanzada), es el codec de video más común. AAC (Codificación de Audio Avanzada) es el codec de audio más común.



## Estándares de codificación y decodificación

A continuación, se describen algunos de los sistemas de codificación de video más utilizados:

- **Motion JPEG (MJPEG)**

**Descripción:** MJPEG es una técnica de compresión de video donde cada fotograma se comprime como una imagen JPEG independiente. No utiliza compresión inter-frame, lo que significa que no hay compresión temporal entre los fotogramas.

**Ventajas:**

- Alta calidad de imagen por fotograma.
- Baja latencia, adecuado para aplicaciones en tiempo real como videoconferencias y cámaras de seguridad.

**Desventajas:**

- Mayor tamaño de archivo comparado con otros códecs que utilizan compresión inter-frame.
- Menos eficiente en términos de compresión.

**Aplicaciones:** Videovigilancia, grabación de video en cámaras digitales, transmisión de video en tiempo real.

- **H.264 (AVC - Advanced Video Coding)**

**Descripción:** H.264 es uno de los códecs de video más utilizados, conocido por su alta eficiencia de compresión y buena calidad de imagen. Utiliza compresión intra-frame e inter-frame para reducir significativamente el tamaño de los archivos de video.

### **Ventajas:**

- Excelente relación de compresión/calidad.
- Amplia compatibilidad con una variedad de dispositivos y plataformas.
- Soporte para resoluciones de alta definición (HD) y ultra alta definición (UHD).

### **Desventajas:**

- Mayor complejidad de codificación y decodificación en comparación con MJPEG.

**Aplicaciones:** Streaming de video en línea (YouTube, Netflix), videoconferencias, grabación y distribución de video en HD.

- **H.265 (HEVC - High Efficiency Video Coding)**

**Descripción:** H.265 es el sucesor de H.264 y ofrece una eficiencia de compresión aún mayor, aproximadamente el doble que H.264, manteniendo una calidad de video similar.

### **Ventajas:**

- Reducción significativa del tamaño del archivo con la misma calidad de video.
- Mejor rendimiento en resoluciones 4K y superiores.
- Soporte para contenido HDR (High Dynamic Range).

### **Desventajas:**

- Aún mayor complejidad de codificación y decodificación.
- No tan ampliamente soportado en dispositivos más antiguos.

**Aplicaciones:** Streaming de video 4K y 8K, almacenamiento de video de alta calidad, transmisión en línea.

- **VP8**

**Descripción:** VP8 es un códec de video desarrollado por On2 Technologies y ahora mantenido por Google como parte del proyecto WebM. Es una alternativa gratuita y de código abierto a H.264.

### **Ventajas:**

- Sin costos de licencia, lo que lo hace atractivo para desarrolladores y plataformas abiertas.

- Buen equilibrio entre calidad y compresión.

**Desventajas:**

- No tan eficiente como H.265 en términos de compresión.
- Menos soporte en hardware comparado con H.264 y H.265.

**Aplicaciones:** Transmisión de video en web, aplicaciones de videoconferencia como Google Hangouts.

• **VP9**

**Descripción:** VP9 es el sucesor de VP8, también desarrollado por Google. Ofrece una eficiencia de compresión comparable a H.265, siendo también de código abierto y libre de regalías.

**Ventajas:**

- Alta eficiencia de compresión, similar a H.265.
- Libre de costos de licencia.
- Soporte mejorado para resoluciones 4K.

**Desventajas:**

- Soporte de hardware limitado en comparación con H.264, aunque está creciendo.
- Codificación y decodificación más compleja que VP8.

**Aplicaciones:** YouTube (transmisión de video en 4K), aplicaciones web y móviles, streaming en línea.

• **AV1**

**Descripción:** AV1 es un códec de video de nueva generación desarrollado por la Alliance for Open Media (AOMedia). Es un códec gratuito y de código abierto diseñado para reemplazar VP9 y competir con H.265.

**Ventajas:**

- Mejor eficiencia de compresión que H.265 y VP9.
- Libre de regalías.
- Optimizado para streaming de video en alta calidad.

**Desventajas:**

- Complejidad de codificación y decodificación.

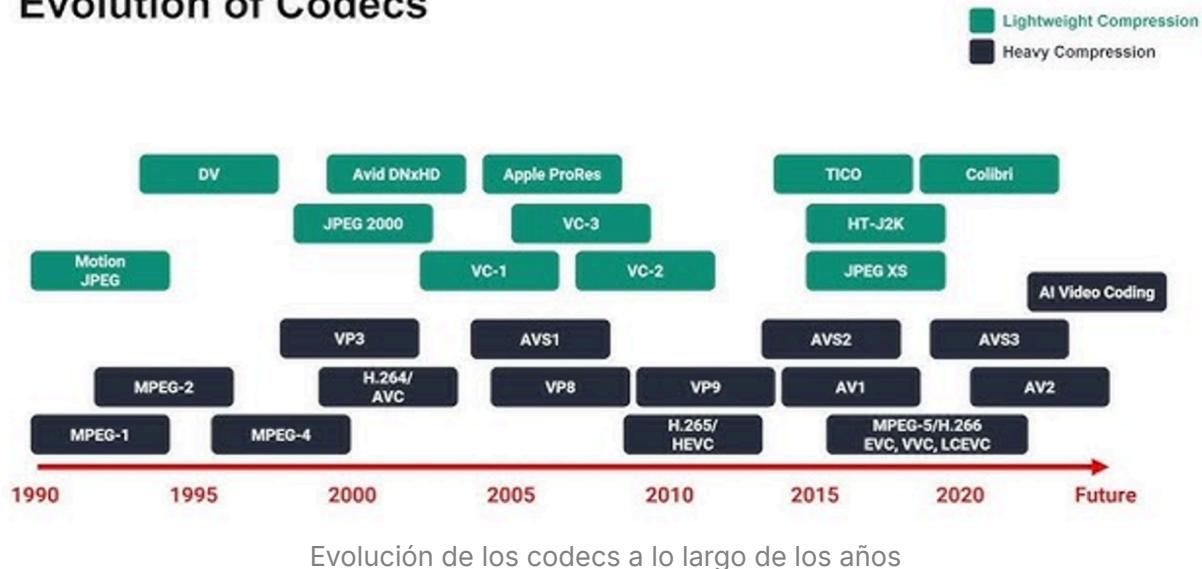
- Implementación de hardware aún en desarrollo.

**Aplicaciones:** Streaming de video en línea, plataformas de video como YouTube y Netflix están comenzando a adoptar AV1.

## Factores importantes en la codificación

Al seleccionar y configurar encoders de video, varios factores son cruciales para determinar la calidad final del video, la eficiencia de la transmisión y la compatibilidad con diferentes dispositivos y plataformas.

### Evolution of Codecs



Evolución de los codecs a lo largo de los años

A continuación se describen algunos de los factores más importantes a considerar:

- **Bitrate (Tasa de Bits)**

**Descripción:** El bitrate es la cantidad de datos procesados por segundo en un flujo de video, generalmente medido en kilobits por segundo (kbps) o megabits por segundo (Mbps).

**Importancia:**

- **Calidad del Video:** Un mayor bitrate generalmente se traduce en una mejor calidad de video, ya que hay más datos disponibles para representar la imagen.
- **Tamaño del Archivo:** Un mayor bitrate también aumenta el tamaño del archivo, lo que puede ser una consideración importante para el almacenamiento y la transmisión.

- **Ancho de Banda:** Un bitrate alto requiere más ancho de banda para la transmisión, lo que puede no ser viable en redes con limitaciones de velocidad.

- **Calidad**

**Descripción:** La calidad del video se refiere a la fidelidad visual del video codificado en comparación con el original.

**Importancia:**

- **Algoritmos de Compresión:** Diferentes encoders utilizan distintos algoritmos que afectan la calidad del video. Por ejemplo, H.265 ofrece una mejor calidad a menores bitrates en comparación con H.264.
- **Configuraciones de Encoder:** Ajustes como el modo de compresión (CBR, VBR), el nivel de compresión y los filtros de procesamiento también afectan la calidad final del video.

- **Resolución**

**Descripción:** La resolución es la cantidad de píxeles en cada dimensión que el video contiene, comúnmente expresada como ancho × alto (por ejemplo, 1920×1080).

**Importancia:**

- **Detalles Visuales:** Una mayor resolución proporciona más detalles y claridad en la imagen, esencial para videos en alta definición (HD), 4K y superiores.
- **Recursos de Procesamiento:** Codificar y decodificar video en alta resolución requiere más recursos de procesamiento y memoria.

- **Keyframes (Fotogramas Clave)**

**Descripción:** Los keyframes son fotogramas completos en el video que se almacenan periódicamente para permitir la reconstrucción de otros fotogramas intermedios basados en cambios (frames diferenciales).

**Importancia:**

- **Búsqueda y Edición:** Los keyframes permiten una búsqueda y edición más eficientes del video.
- **Calidad de Transición:** Un mayor número de keyframes puede mejorar la calidad de las transiciones y reducir artefactos, pero también puede aumentar el tamaño del archivo.

- **Latencia**

**Descripción:** La latencia es el retraso temporal entre la captura del video y su visualización después de ser codificado y transmitido.

**Importancia:**

- **Aplicaciones en Tiempo Real:** En aplicaciones como videoconferencias, transmisión en vivo y videojuegos, una baja latencia es crítica para una experiencia fluida.
- **Buffering:** La latencia también afecta cómo se maneja el buffering y la estabilidad de la transmisión en condiciones de red fluctuantes.

- **Procesamiento**

**Descripción:** El procesamiento se refiere a los recursos computacionales necesarios para codificar y decodificar el video.

**Importancia:**

- **Eficiencia del Encoder:** Encoders más eficientes pueden producir video de alta calidad con menos recursos de procesamiento.
- **Costos de Hardware:** La cantidad de procesamiento requerido puede influir en los costos de hardware y la infraestructura necesaria para manejar la codificación y decodificación del video.

- **Formato de Compresión**

**Descripción:** El formato de compresión (por ejemplo, H.264, H.265, VP9, AV1) determina cómo se comprimen los datos del video.

**Importancia:**

- **Compatibilidad:** Diferentes plataformas y dispositivos soportan diferentes formatos de compresión.
- **Eficiencia de Compresión:** Formatos más avanzados como H.265 y AV1 ofrecen mejores tasas de compresión a la misma calidad de video comparados con H.264.

- **Modo de Compresión**

**Descripción:** Los modos de compresión incluyen CBR (Constant Bit Rate) y VBR (Variable Bit Rate).

**Importancia:**

- **CBR:** Proporciona un bitrate constante, útil para transmisión en tiempo real donde la estabilidad es crucial.
  - **VBR:** Ajusta el bitrate según la complejidad de la escena, ofreciendo mejor calidad en escenas complejas y eficiencia en escenas simples.
- **Redundancia y Robustez**

**Descripción:** La capacidad de un encoder/decoder para manejar errores y pérdida de datos.

**Importancia:**

- **Fiabilidad de la Transmisión:** Encoders con técnicas de corrección de errores y robustez pueden mantener la calidad del video incluso en condiciones de red imperfectas.

- **Compatibilidad y Soporte de Aplicaciones**

**Descripción:** La capacidad del encoder para ser utilizado en diferentes aplicaciones, dispositivos y plataformas.

**Importancia:**

- **Compatibilidad de Hardware y Software:** Algunos encoders son ampliamente soportados en una variedad de dispositivos y sistemas operativos, lo que facilita su integración en diferentes entornos.
- **Soporte de Plataforma:** Encoders como H.264 y H.265 son soportados por la mayoría de las plataformas de streaming y dispositivos de reproducción, mientras que formatos más nuevos como AV1 están ganando soporte.
- **Interoperabilidad:** La capacidad de trabajar bien con otros componentes del sistema, como reproductores de video, servidores de streaming y editores de video.

## Protocolos de transmisión

Un protocolo es un conjunto de reglas que gobiernan cómo viajan los datos de un dispositivo a otro. Por ejemplo, el Protocolo de Transferencia de Hipertexto (HTTP) se ocupa de documentos de hipertexto y páginas web. La entrega de video en línea utiliza tanto protocolos de transmisión como protocolos basados en HTTP. Protocolos de transmisión como el Protocolo de Mensajería en Tiempo Real (RTMP) ofrecen una entrega de video rápida, mientras que los

protocolos basados en HTTP como HLS pueden ayudar a optimizar la experiencia de visualización.

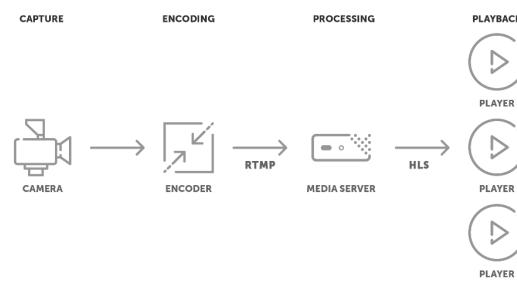


El protocolo utilizado puede aumentar la latencia de transmisión hasta en 45 segundos.

### Protocolos de Transmisión Tradicionales con Estado

En los primeros días, los protocolos tradicionales como el Protocolo de Transmisión en Tiempo Real (RTSP) y el Protocolo de Mensajería en Tiempo Real (RTMP) eran los métodos preferidos para transmitir video por internet y reproducirlo en dispositivos domésticos. Estos protocolos son con estado, lo que significa que requieren un servidor de transmisión dedicado.

Mientras que RTSP y RTMP admiten una entrega de video extremadamente rápida, no están optimizados para grandes experiencias de visualización a gran escala. Además, cada vez menos reproductores admiten estos protocolos. Muchos emisores eligen transportar transmisiones en vivo a su servidor de medios usando un protocolo con estado como RTMP y luego transcodificarlo para la entrega en múltiples dispositivos.

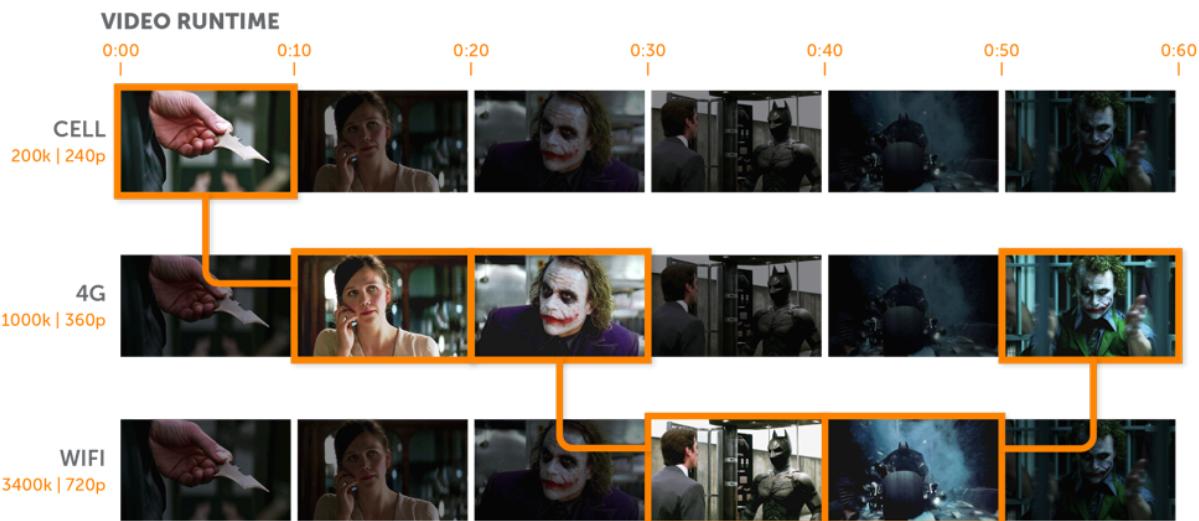


RTMP y RTSP mantienen la latencia en aproximadamente 5 segundos o menos.

### Protocolos de Transmisión Adaptativos Basados en HTTP

Eventualmente, la industria se inclinó a favor de las tecnologías basadas en HTTP. Más bien, son descargas progresivas enviadas a través de servidores web regulares.

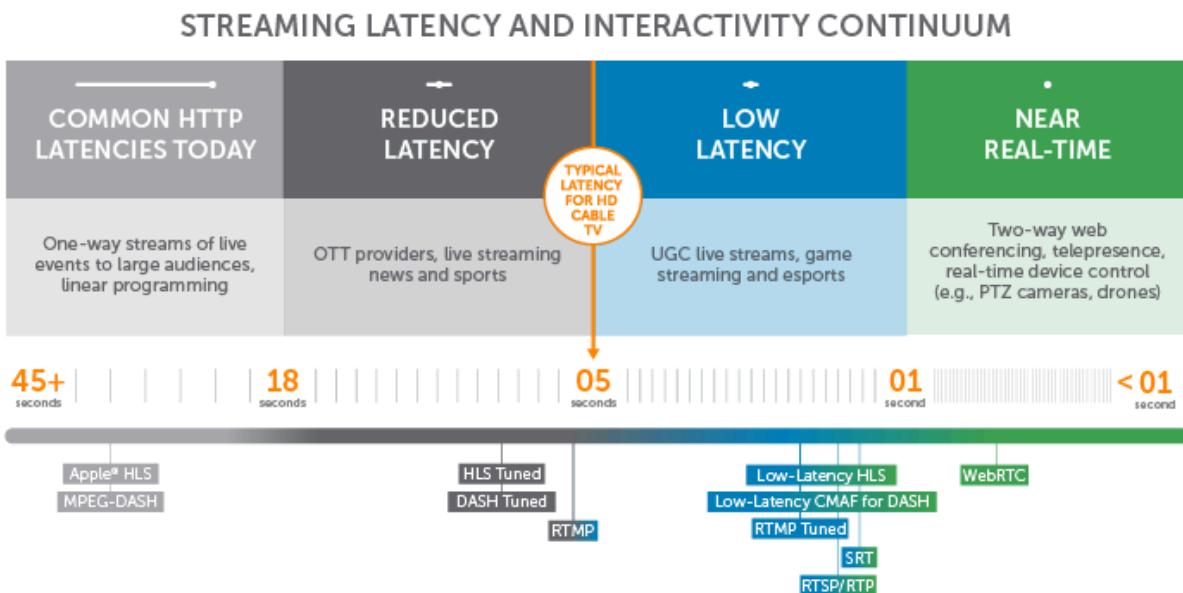
Utilizando la transmisión adaptativa de bitrate, los protocolos basados en HTTP entregan la mejor calidad de video y experiencia de visualización posible, sin importar la conexión, el software o el dispositivo.



Según el ancho de banda disponible, el protocolo se encarga de seleccionar el stream más apropiado.

Algunos de los protocolos basados en HTTP más comunes incluyen MPEG-DASH y HLS de Apple. Los protocolos basados en HTTP son sin estado, lo que significa que pueden ser entregados usando un servidor web común y corriente. Dicho esto, se encuentran en el extremo superior del espectro de latencia.

Los protocolos basados en HTTP pueden causar de 10 a 45 segundos de latencia.

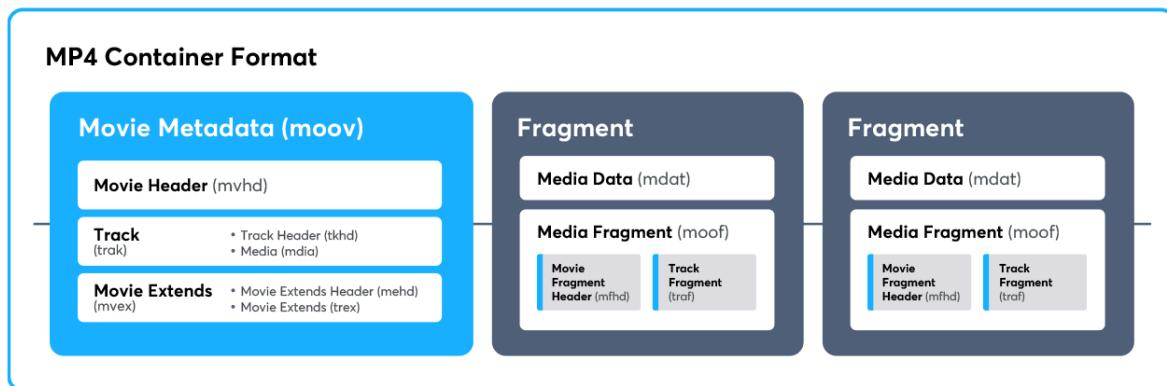


## Protocolos emergentes para entrega casi en tiempo real

Con un número creciente de videos entregados en vivo, los líderes de la industria continúan mejorando la tecnología de transmisión. Estándares emergentes como WebRTC, SRT, HLS de baja latencia y CMAF de baja latencia para DASH (que es un formato más que un protocolo) admiten la entrega casi en tiempo real, incluso sobre conexiones deficientes. Estos nuevos conjuntos de tecnologías prometen reducir la latencia a 3 segundos o menos.

## Formato contenedor de video

Los formatos de contenedor de video, también llamados wrappers (envoltorios), contienen todos los componentes de una transmisión comprimida. Esto podría incluir el codec de audio, el codec de video, subtulado oculto y cualquier metadato asociado, como subtítulos o imágenes de vista previa.



Formato MP4 (<https://bitmovin.com/container-formats-fun-1>)

Los contenedores más comunes son:

- **.mp4** : (MPEG-4 Parte 14) es uno de los formatos de contenedor más comunes y ampliamente compatible. Utilizado para almacenar video y audio, pero también puede contener subtítulos y otros datos. MP4 es un formato estándar para la distribución de contenido multimedia en internet y es compatible con casi todos los dispositivos y reproductores.
- **.avi** : (Audio Video Interleave) fue desarrollado por Microsoft a principios de los años 90. Es un formato de archivo de video muy antiguo y popular que permite la reproducción sincrónica de audio y video. AVI puede contener

tanto datos de audio como de video en un contenedor que permite la reproducción simultánea.

- **.mkv** : (Matroska Video) es un formato de contenedor moderno que puede contener un número ilimitado de video, audio, imágenes o pistas de subtítulos en un solo archivo. Es conocido por su capacidad para contener una amplia gama de tipos de datos y por ser un formato abierto, no patentado.
- **.mov** (QuickTime File Format) Desarrollado por Apple y originalmente destinado para el reproductor QuickTime. MOV es utilizado comúnmente por editores de video debido a su capacidad de almacenar datos complejos en diferentes pistas multimedia.
- **.ts** : (Transport Stream) Utilizado para transmitir video y audio en aplicaciones de transmisión de medios, especialmente para la transmisión de televisión digital. Es capaz de multiplexar digitalmente flujos de audio, video y otros datos.
- **.wmv** : (Windows Media Video) fue desarrollado por Microsoft, este formato es parte del framework de Windows Media. WMV está diseñado principalmente para aplicaciones de streaming en el entorno de Windows.
- **.webm** : Es un formato de contenedor de medios abierto y libre desarrollado específicamente para ser utilizado en la web. WebM es compatible con video que se comprime con el códec VP8 o VP9 y audio comprimido con Vorbis o Opus.
- **.ogg** : Aunque más conocido por contener datos de audio (especialmente con el códec Vorbis), Ogg también puede encapsular video (usualmente con Theora). Es un formato de contenedor libre y abierto, desarrollado por la fundación Xiph.Org.
- **.3gp** : Diseñado para uso en teléfonos móviles 3G, este formato se utiliza para simplificar el almacenamiento y la transferencia de archivos de video y audio entre dispositivos móviles.
- **.mpg** : es otro contenedor de archivos multimedia muy conocido, asociado principalmente con los estándares MPEG (Moving Picture Experts Group). En general, el término **.mpg** se refiere a archivos de video codificados bajo los estándares MPEG-1 o MPEG-2.

## Práctica

Google Colab

🔗 [https://colab.research.google.com/drive/1SfWMRTDIEs35RvoemUfGk1ke9IN\\_zOwC?usp=sharing](https://colab.research.google.com/drive/1SfWMRTDIEs35RvoemUfGk1ke9IN_zOwC?usp=sharing)



## 3. Despliegue y optimización de inferencia

### Introducción

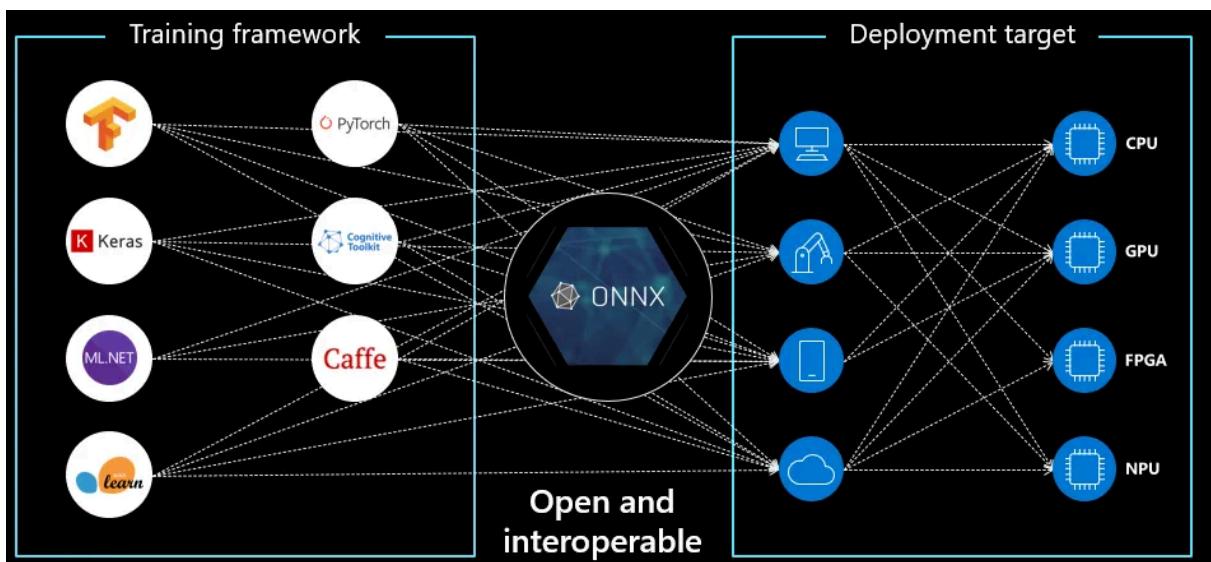
El tiempo de inferencia para un modelo de aprendizaje automático depende de muchos factores que comienzan desde capacidades a nivel de sistema como la GPU y sus núcleos CUDA, hasta el marco del modelo, la precisión de los pesos del modelo, el uso de diferentes técnicas de agrupación de entradas y los métodos utilizados para la inferencia. Mientras que las capacidades a nivel de sistema para la aceleración de hardware están limitadas en la mayoría de los proyectos, todos los demás factores pueden optimizarse mediante la aceleración de software y la aceleración algorítmica o de red para aumentar el rendimiento.

### Frameworks de aprendizaje profundo

Tensorflow y PyTorch son dos de los marcos más estándares en el campo de la Inteligencia Artificial y el Aprendizaje Automático, particularmente para la comunidad de aprendizaje profundo. Ambos han recibido una aceptación generalizada para casos de uso de despliegue y producción, además de análisis y desarrollo. Sin embargo, debido a que Python y estos marcos están estrechamente vinculados, existe una falta de interoperabilidad y una restricción en su uso y flexibilidad en otros ecosistemas y entornos. Además, otros marcos son más adecuados para etapas particulares del proceso de desarrollo, como el entrenamiento rápido, la flexibilidad de la arquitectura de la red, como el ecosistema C++ (software nativo de Linux), o la inferencia en dispositivos móviles como aplicaciones Android e iOS y frameworks de frontend como JavaScript.

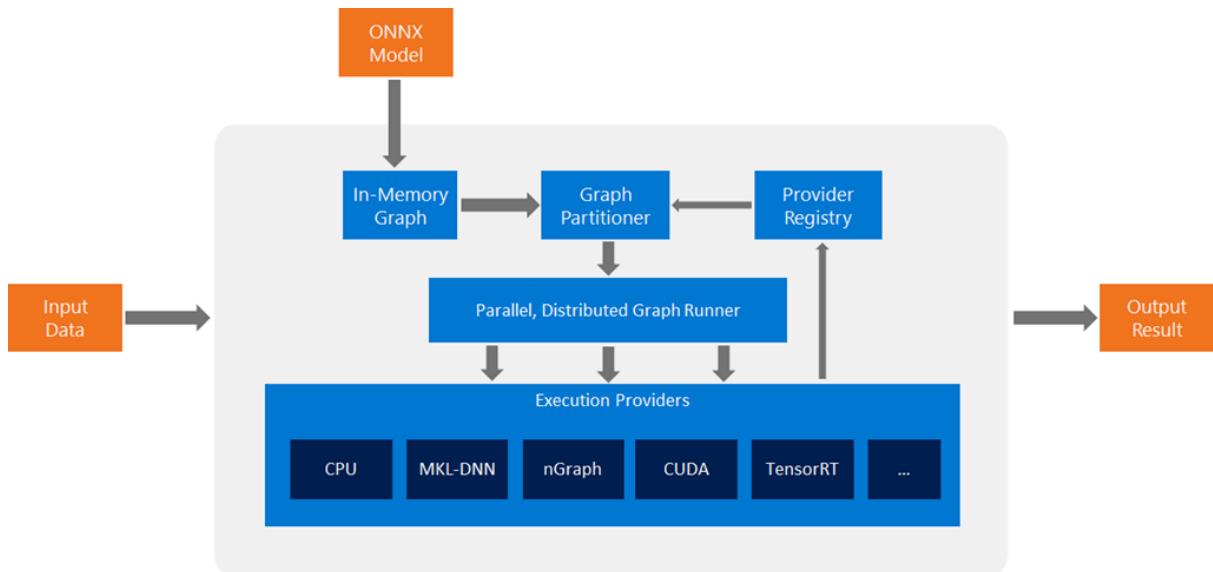
### ONNX

ONNX actuó como este formato intermedio utilizado para convertir los pesos de modelos de aprendizaje automático entrenados de un formato a otro e introdujo la optimización compartida, ya que permitió a los proveedores de hardware y otros mejorar el rendimiento de las redes neuronales profundas de múltiples marcos de trabajo al mismo tiempo, enfocándose en las representaciones de ONNX.



Con el objetivo de lograr la interoperabilidad, Toffee fue desarrollado por el equipo de PyTorch en Facebook. En septiembre de 2017, este proyecto interno fue renombrado a Open neural network exchange (ONNX) y anunciado por Facebook y Microsoft a la comunidad de código abierto. Más tarde, otros gigantes tecnológicos como IBM, Huawei, Intel, AMD, Arm y Qualcomm proclamaron su apoyo a esta iniciativa. — [Wikipedia](#)

ONNX abstrae las similitudes entre frameworks para crear una definición estándar de modelo de un modelo de grafo de cómputo extensible, operadores incorporados y tipos de datos estándar, centrados en la inferencia. La plataforma ofrece soporte para varios frameworks de inteligencia artificial populares, para crear, desplegar, optimizar y visualizar un modelo.



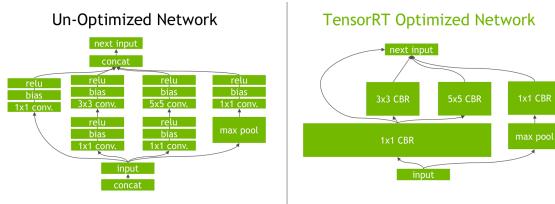
La estructura de ejecución de Open Neural Network Exchange (ONNX) consta de un conjunto de API, bibliotecas y herramientas. Estas permiten una ejecución eficiente y multiplataforma de modelos representados en el formato. Fuente: <https://onnxruntime.ai/docs/reference/high-level-design.html>

ONNX no es solo un formato de modelo conveniente, también podemos utilizar nuestros modelos en este formato con el framework de aceleración de aprendizaje automático [ONNX Runtime](#) de Microsoft, para procesos de reentrenamiento e inferencia más rápidos. PyTorch tiene soporte nativo de funciones para exportar modelos a ONNX.

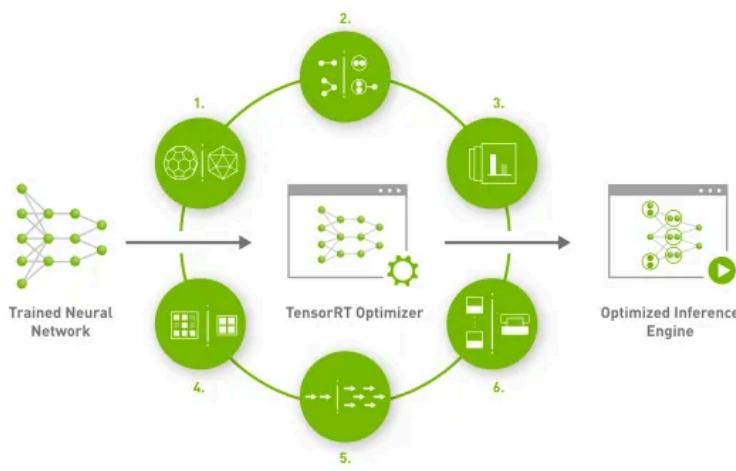
## TensorRT

El mundo está experimentando una creciente necesidad de aceleración, impulsada principalmente por preocupaciones empresariales como la reducción de costos o la mejora de la experiencia del usuario final al reducir la latencia y consideraciones tácticas como el despliegue de modelos en dispositivos periféricos con menos recursos computacionales.

TensorRT, desarrollado por NVIDIA, surgió de esta necesidad de velocidad como un SDK para inferencia de aprendizaje profundo de alto rendimiento. Incluye un optimizador de inferencia de aprendizaje profundo y un entorno de ejecución que proporciona baja latencia y alto rendimiento para estas aplicaciones de inferencia. Emplea métodos como la calibración de precisión de pesos y activaciones para lograr la cuantificación INT8 con una huella de modelo mucho menor, fusión de capas y tensores en conjunto con la autoajuste de núcleos, maximizando aún más la utilización de la GPU.



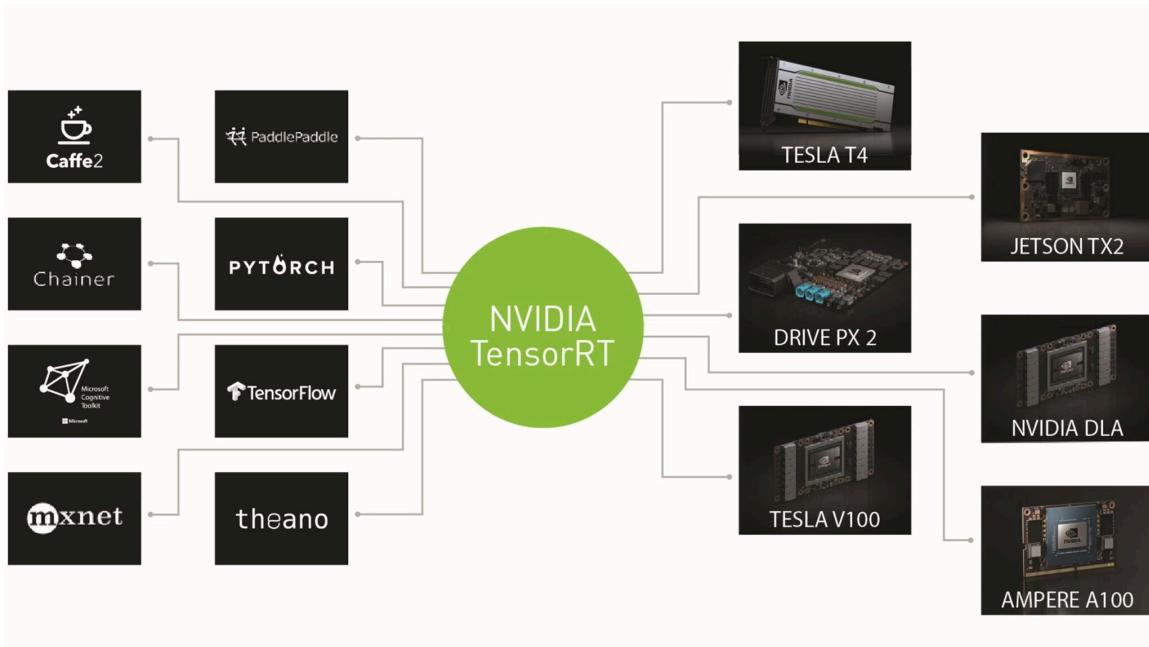
**Fusión de Capas:** El proceso de optimización de TensorRT incluye la fusión de capas, donde múltiples capas de una red neuronal se combinan en una sola operación. Esto reduce la sobrecarga computacional y mejora la velocidad de inferencia al minimizar el acceso a la memoria y la computación.



1. **Weight & Activation Precision Calibration**  
Maximizes throughput by quantizing models to INT8 while preserving accuracy
2. **Layer & Tensor Fusion**  
Optimizes use of GPU memory and bandwidth by fusing nodes in a kernel
3. **Kernel Auto-Tuning**  
Selects best data layers and algorithms based on target GPU platform
4. **Dynamic Tensor Memory**  
Minimizes memory footprint and re-uses memory for tensors efficiently
5. **Multi-Stream Execution**  
Scalable design to process multiple input streams in parallel
6. **Time Fusion**  
Optimizes recurrent neural networks over time steps with dynamically generated kernels

Diferentes herramientas para la optimización. Imagen de <https://developer.nvidia.com/tensorrt>

Con soporte para cada framework, TensorRT ayuda a procesar grandes cantidades de datos con baja latencia a través de optimizaciones poderosas, uso de precisión reducida y uso eficiente de la memoria. ONNX facilita una conversión más fluida desde todos los formatos de modelo posibles.

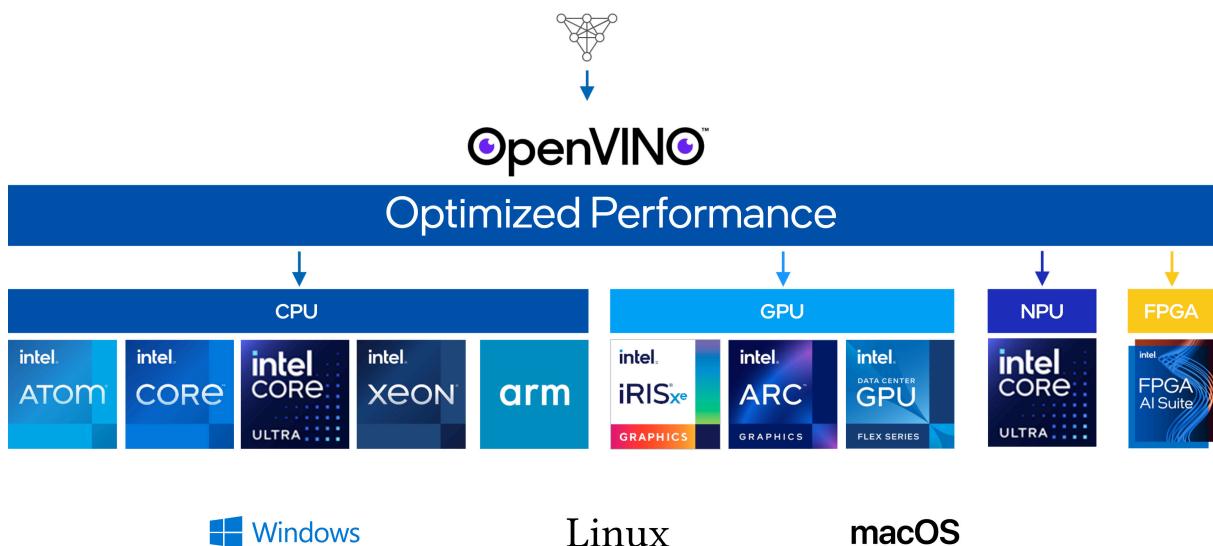


Las aplicaciones del mundo real tienen demandas de computación mucho mayores, necesidades de procesamiento adicionales y límites de latencia más ajustados con modelos de aprendizaje profundo más grandes. TensorRT es una herramienta muy potente para la inferencia.

## OpenVINO

OpenVINO, que significa "Open Visual Inference and Neural Network Optimization", es un kit de herramientas de Intel diseñado para facilitar el desarrollo y la implementación de aplicaciones de inteligencia artificial (IA) y visión por computadora. Este marco está optimizado especialmente para productos de hardware de Intel, incluyendo CPUs, GPUs integradas, Intel Movidius Neural Compute Stick (VPU), y FPGAs de Intel. La finalidad de OpenVINO es mejorar la eficiencia de las aplicaciones de IA en términos de velocidad y consumo de energía, permitiendo que los dispositivos en el borde (edge devices) realicen tareas de procesamiento de visión computarizada y decisiones de IA de manera más rápida y eficiente. Características Principales de OpenVINO:

- 1. Soporte de Hardware Diversificado:** Aunque está optimizado para hardware de Intel, OpenVINO soporta una amplia gama de dispositivos, lo que permite una mayor flexibilidad en el despliegue de aplicaciones de IA.
- 2. Optimización de Modelos de Redes Neuronales:** OpenVINO transforma los modelos de redes neuronales en una Representación Intermedia (IR). Esta representación está optimizada para ejecutarse de manera eficiente en dispositivos de destino.
- 3. Aceleración de la Inferencia:** Utiliza optimizaciones avanzadas para mejorar el rendimiento de la inferencia de modelos de aprendizaje profundo, reduciendo la latencia y aumentando el rendimiento en comparación con las ejecuciones en dispositivos no optimizados.
- 4. Conversión de Modelos:** OpenVINO incluye herramientas que convierten modelos de varios marcos de trabajo de aprendizaje profundo, como TensorFlow, PyTorch, y otros, al formato IR. Esto facilita la integración y el despliegue en una variedad de plataformas.
- 5. Backends de Aceleración:** Soporta backends como MKL-DNN para optimizar las operaciones en CPUs y GPUs de Intel, aprovechando las características de hardware específicas para maximizar el rendimiento.
- 6. Facilidad de Uso:** Proporciona scripts y soporte para facilitar la instalación y gestión de dependencias, permitiendo que los desarrolladores se enfoquen más en el desarrollo de aplicaciones en lugar de en la configuración del entorno.

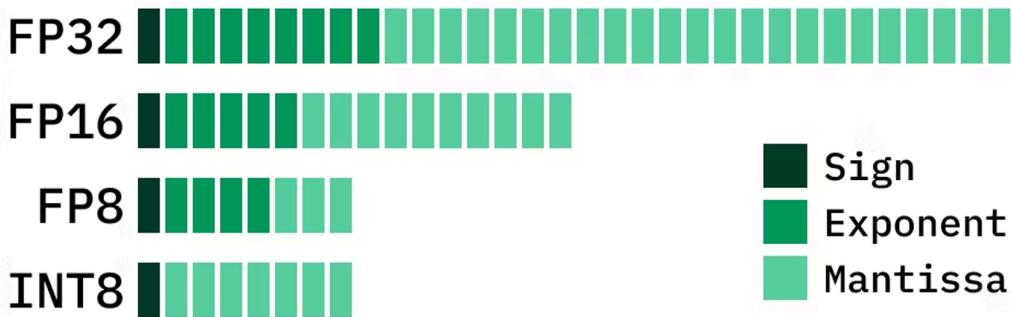


OpenVINO es utilizado ampliamente en aplicaciones industriales, de vigilancia, de salud, y de retail, donde la capacidad de procesar rápidamente grandes volúmenes de datos visuales y tomar decisiones inteligentes en el dispositivo es crucial. Su capacidad para funcionar en dispositivos de cómputo en el borde (Edge Computing) también lo hace ideal para escenarios donde la latencia o la privacidad de los datos es una preocupación significativa.

## Precisión de Punto Flotante en el Aprendizaje Profundo

La precisión de punto flotante de un modelo, no suele ser el factor clave al trabajar con modelos de aprendizaje profundo. Si nuestro modelo no está funcionando bien, entonces el formato de punto flotante ciertamente no nos va a salvar.

Pero dado el aumento en complejidad, tamaño y tiempo de entrenamiento para los modelos de aprendizaje profundo, la precisión de punto flotante puede tener un impacto significativo en el tiempo de entrenamiento/inferencia e incluso en el rendimiento del modelo. Algunas precisiones comúnmente utilizadas son FP32, FP16, BFP16, INT8, etc.



### Más información:

<https://moocaholic.medium.com/fp64-fp32-fp16-bfloat16-tf32-and-other-members-of-the-zoo-a1ca7897d407>

Comparación de Precisión entre Formatos Numéricos | Claude

Try out Artifacts created by Claude users

 <https://claude.ai/public/artifacts/98243a20-ce12-4cf0-98f3-9e30c8f2992c>

## Procesamiento de video en tiempo real

Cuando procesamos un flujo (stream) de video es de vital importancia de optimizar los tiempos de procesamiento, para evitar la pérdida de cuadros. En este contexto, esto significa que ingresan más cuadros de los que nuestro sistema es capaz de procesar. Por cada cuadro, nuestro sistema de vision realiza normalmente estas tareas:

1. **Preprocesamiento:** Esta etapa prepara los cuadros de video para la inferencia. Puede incluir tareas como ajuste de tamaño, normalización de intensidad de píxeles, y quizás la conversión de color. La optimización en esta etapa es fundamental para asegurar que los datos estén listos y sean adecuados para el modelo de inferencia sin retrasos.
2. **Inferencia:** Aquí es donde el modelo de aprendizaje automático ya entrenado, procesa el cuadro para obtener resultados, como la detección de objetos, reconocimiento facial, o cualquier otra tarea específica. El tiempo de inferencia depende de la complejidad del modelo y del hardware utilizado (CPU, GPU, etc.).

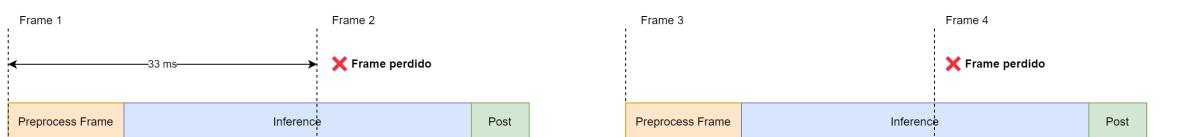
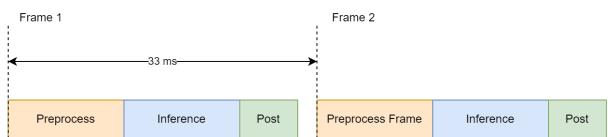
### 3. Post-procesamiento:

Esta fase maneja los resultados de la inferencia.

Puede incluir la interpretación de los datos de salida del modelo, como filtrar o agregar información adicional, y preparar los datos para su visualización o transmisión.

## Importancia de Optimizar los Tiempos

En el diagrama, cada bloque (preproceso, inferencia y post-proceso) debe completarse idealmente en menos de 33 milisegundos para alcanzar un flujo de procesamiento en tiempo real a 30 cuadros por segundo (FPS). Los 33 ms surgen de realizar 1/30 (1 segundo dividido 30 cuadros).



Para procesar un video a 30 FPS, se requiere de tiempos mínimos de pre-procesamiento, inferencia y post-procesamiento, para no perder cuadros de imagen. En el ejemplo de abajo, perdemos una de cada dos imágenes, lo que produce un procesamiento efectivo de 15 FPS.

La optimización es fundamental, por lo siguiente:

- **Evitar la Pérdida de Cuadros:** Si cualquier etapa toma más tiempo de lo permitido, los cuadros subsiguientes tendrán que ser descartados o se acumularán, causando retrasos y pérdida de información en tiempo real. En caso de acumulación, necesitaríamos grandes cantidades de memoria para almacenar los cuadros en un buffer hasta que sean procesados, y además perderíamos el comportamiento de tiempo real.
- **Calidad y Fluidez de la Transmisión:** Para aplicaciones como la videovigilancia en tiempo real, la telemedicina, o las transmisiones en vivo, mantener una alta calidad y fluidez sin interrupciones es esencial para la funcionalidad y la experiencia del usuario.
- **Uso Eficiente del Hardware:** La optimización del tiempo de procesamiento permite un uso más eficiente del hardware, reduciendo la necesidad de recursos computacionales excesivamente poderosos y costosos.

## Estrategias de Optimización

Para lograr un procesamiento eficiente que se ajuste a los límites de tiempo requeridos, podemos considerar:

- **Optimización del Modelo de Aprendizaje Automático:** Utilizar modelos más ligeros o aplicar técnicas de reducción de precisión (como cuantificación) para acelerar la inferencia sin una gran pérdida de precisión.
- **Mejorar el Hardware:** Usar GPUs o hardware especializado como TPUs que pueden procesar operaciones de inferencia mucho más rápido que las CPUs estándar.
- **Paralelización:** Procesar múltiples cuadros en paralelo, si el hardware y software lo permiten, puede ayudar a manejar mejor los tiempos de procesamiento, especialmente en sistemas con múltiples cámaras.
- **Ajustes en el Pre-procesamiento y Post-procesamiento:** Simplificar estas etapas tanto como sea posible, o realizarlas también en hardware especializado.

Optimizar cada etapa del procesamiento de video es vital para garantizar que el sistema sea capaz de operar en tiempo real sin perder cuadros, manteniendo la integridad y la eficacia de cualquier aplicación crítica que dependa de la fluidez y la actualidad del video procesado.