DERECHOS DIGITALES
América Latina

NO ES MAGIA

Interfaces y protocolos para las videollamadas

Martu Isla
cacu
la_jes
Juliana Guerra

Texto por Martu Isla, Juliana Guerra, la_jes (Sursiendo), cacu (Tierra Común Cooperativa).
Portada y diagramación: Constanza Figueroa.
Edición: Vladimir Garay.

Octubre de 2020

Esta publicación fue posible gracias al apoyo de Article 19

i. En aislamiento, ¿podemos mantenernos cerca?

A raíz de la pandemia por Covid-19, durante los últimos meses nos hemos enfrentado a la promesa sobre la que se construyó internet. De un momento a otro, buena parte del mundo se volcó a los entornos digitales. En muchas ciudades del mundo solo las actividades "esenciales" de cuidado, aseo y vigilancia se mantuvieron en marcha, mientras gran parte de la población se vio obligada al confinamiento.

Además de la inminente crisis generada por la suspensión de actividades económicas, en regiones como América Latina y el Caribe se evidenció también la persistente brecha de acceso a internet, en sus múltiples dimensiones. Las cifras son muy desiguales entre países, pero también a nivel interno, entre los contextos urbanos y rurales. ¿Y qué significa acceder a internet? Quizás tener acceso: a un dispositivo, aunque no sea lo mismo una computadora, una tableta o un celular; aunque no sea lo mismo conectarse a banda ancha fija o móvil. Porque el costo tampoco es el mismo.

El costo depende de la infraestructura. En los países más pobres es más costoso conectarse y las conexiones son más lentas. Depende de cuánta infraestructura hay disponible y qué tan robusta es, lo que a su vez depende de los materiales que se utilicen para establecer la conexión y las tecnologías que ejecuten los dispositivos que la median. Pero esta es solo una dimensión del acceso: una vez que podemos conectarnos, nuestras habilidades técnicas, lingüísticas y culturales determinan también nuestra capacidad para "navegar".

Pero por ahora, volvamos sobre la promesa de internet.

En 1989, en el Centro Europeo de Investigaciones Nucleares (CERN) se propuso una herramienta para la colaboración y el intercambio de información que se convertiría en la World Wide Web,[1] materializándose en el protocolo *HTTP* (Hypertext Transfer Protocol o protocolo de transferencia de hipertexto) que utilizamos hoy para navegar. La propuesta consistía en un sistema distribuido de hipertextos, legible para las personas y donde la información estuviera conectada de forma ilimitada, no a partir de un sistema jerárquico fijo.

Inicialmente diseñada para organizar y agilizar el trabajo de una comunidad científica específica, desde el comienzo la Web se planteó como un sistema universal de información interconectada, que soportara distintas plataformas y que fuera extensible a nuevos formatos de datos. Con ese propósito, entre 1993 y 1994 empezó a ser un producto atractivo para el mercado y, poco a poco, fue entrando en oficinas gubernamentales, empresas y hogares, en un proceso expansivo que continúa hasta hoy.

Casi diez años después, la transmisión de video por internet aumentó enormemente el tráfico de información digital, mientras que el desarrollo de la tecnología 3G en telefonía móvil, que

1 Tim Berners-Lee. CERN. 1989-1990. Information Management: A Proposal. https://www.w3.org/History/1989/proposal.html

permitía la conexión a algunos servicios de internet, disparó el nivel de conexiones. Hoy no solo nos conectamos para trabajar, aprender o hacer trámites administrativos, nuestras emociones y sentimientos también están conectados. En pandemia, nuestros círculos de afecto, confianza y organización política están necesariamente mediados por las tecnologías de internet.

La edición colaborativa, el intercambio de archivos y la transmisión de audio y video en tiempo real son quizás las herramientas digitales más útiles en tiempos de confinamiento, pero ¿por qué las videollamadas y videoconferencias se hicieron tan populares en los últimos meses? A pesar de consumir muchos recursos y que muchas veces la comunicación no es fluida ni comprensible, optamos por vernos: en las clases; en las reuniones con pocas o muchas personas; en las presentaciones y talleres; en las fiestas; en el sexo.

Más allá de los motivos que nos lleven a preferir herramientas de audio y video en tiempo real, o de las alternativas comerciales y sus características en términos de calidad, seguridad o privacidad, en este documento queremos entender cómo es técnicamente posible esta comunicación, y nos preguntamos si el acceso a videollamadas y videoconferencias es universal. O, dicho de otra forma, de qué depende el poder utilizar estos servicios de manera óptima.

ii. Las limitaciones del contacto

Cuando comenzaron las medidas de distanciamiento físico para frenar la curva de contagio por Covid-19, desde muchos lugares reclamamos fortalecer el encuentro social, y ahí estaba internet para satisfacernos. De acuerdo con la Cepal,[2] durante el primer semestre de 2020 aumentó vertiginosamente el consumo de servicios de comunicación de banda ancha en América Latina y el Caribe: el uso de soluciones de teletrabajo aumentó 324%, la educación en línea 62% y el comercio electrónico 157%.

Esto significó un aumento del tráfico y mayor exigencia en capacidad y resiliencia para las redes que operan en la región, aunque el potencial de conectividad siga siendo bastante limitado. Dice la Cepal que para garantizar en la región una participación efectiva en los entornos digitales, incluyendo acceso a salud, educación y trabajo, así como a compras, servicios de banca y entretenimiento, se requiere primero ampliar la cobertura de banda ancha fija y mejorar la velocidad de conexión de banda ancha móvil. Además, para la prestación de servicios de salud en línea, llama la atención sobre la necesidad de garantizar acceso a información médica digitalizada e interoperabilidad de los servicios, así como privacidad y seguridad de los datos.

Pero más allá del acceso a bienes y servicios digitales, o de las cifras sobre la calidad de conexión a banda ancha fija o móvil, en confinamiento se ha hecho evidente cómo en los entornos digita-

2 Comisión Económica para América Latina y el Caribe, Cepal. 2020. Universalizar el acceso a las tecnologías digitales para enfrentar los efectos del COVID-19. https://repositorio.cepal.org/bitstream/handle/11362/45938/4/S2000550_es.pdf

les tenemos experiencias y entablamos relaciones en las que necesariamente estamos encarnando múltiples identidades y realidades. Así lo reconocen los Principios Feministas para Internet,[3] que desde hace varios años reclaman acceso "universal, aceptable, asequible, incondicional, abierto, significativo e igualitario", especialmente para las mujeres y disidencias sexogenéricas.

La posibilidad de acceder a internet está atravesada por múltiples dimensiones y, mientras la industria busca soluciones rentables para conectar a la otra mitad de la población mundial, avanza rápidamente hacia tecnologías de punta, cada vez más complejas y que requieren de mejores infraestructuras para funcionar de manera óptima. Justamente por eso, hoy es urgente trabajar para que la ampliación de la cobertura se haga con criterios de calidad y dignidad para las personas usuarias, pues se trata de conectar a comunidades tradicionalmente marginadas y sometidas a distintos tipos de violencia.

Durante los primeros meses de 2020, distintas organizaciones publicaron guías para orientar el buen uso de plataformas y aplicaciones de videollamadas, algunas dirigidas a audiencias amplias,[4] otras a grupos críticos como periodistas,[5] maestras de escuela[6] o activistas.[7] Analistas de seguridad y privacidad voltearon su mirada sobre las plataformas más populares y muchas de estas tuvieron que actualizar sus políticas, diseños y configuraciones, para responder a las necesidades del momento.

El caso de Zoom es paradigmático. Esta empresa con sede en Silicon Valley, desde 2013 estaba tratando de posicionarse como competencia frente a Google, Apple o Microsoft, ofreciendo una interfaz sencilla y amigable, al mismo tiempo que garantizaba una transmisión estable de audio y video. Conforme las medidas tempranas de confinamiento comenzaban a entrar en vigencia, Zoom se volvió la opción de videollamadas más popular en empresas, entidades estatales y centros educativos. Así, pasó de 10 millones de participantes por día en diciembre de 2019, a 300 millones en abril de 2020.[8]

---

3 Principios Feministas de Internet. Declaraciones que ofrecen una perspectiva de género y derechos sexuales sobre derechos críticos relacionados con Internet. 2014-2015. https://feministinternet.org/en

4 En inglés https://foundation.mozilla.org/en/privacynotincluded/categories/video-call-apps/ y https://videoconferencing.guide/, entre otros recursos. En Español, y para América Latina https://www.derechosdigitales.org/wp-content/uploads/pub_videollamadas.pdf

5 Choosing the right video conferencing tool for the job. https://freedom.press/training/blog/videoconferencing-tools/

6 Protecting Students in Virtual Classrooms: Considerations for Educators. https://cdt.org/insights/protecting-students-in-virtual-classrooms-considerations-for-educators/

7 Guía sobre herramientas seguras para conferencias y chats grupales. https://www.frontlinedefenders.org/es/resource-publication/guide-secure-group-chat-and-conferencing-tools

8 Datos publicados en su blog. Disponible en https://blog.zoom.us/a-message-to-our-users/ y https://blog.zoom.us/90-day-security-plan-progress-report-april-22/

A partir de marzo se empezaron a publicar una serie de críticas respecto a las vulnerabilidades en la plataforma y el discurso engañoso con que se publicitaba. Ya en 2019 se había denunciado su capacidad de "eludir la configuración de seguridad del navegador y habilitar remotamente la cámara web de una usuaria sin su conocimiento o consentimiento",[9] a lo que se sumaron críticas por la función de *seguimiento de atención*, que permite a la anfitriona ver si alguna asistente no tiene el cliente de escritorio o la aplicación móvil en foco durante más de 30 segundos.[10]

También se levantaron alertas por el llamado *Zoom Bombing* ("bombardeos en Zoom"),[11] por los datos que la plataforma enviaba a Facebook para notificar cuando alguien abría la aplicación,[12] y por el filtrado de datos de quienes se suscribían con cuentas de correo electrónico en servidores diferentes a los más populares, como Gmail, Hotmail o Yahoo.[13] Luego vinieron análisis sobre los mecanismos de preinstalación ejecutados en macOS,[14] la implementación de "lo que la empresa llama cifrado punto a punto",[15] sus alternativas de ruteo utilizando servidores en China desde que comenzó la pandemia,[16] y una vulnerabilidad en la sala de espera de una videollamada.[17]

Según explicó The Intercept a fines de marzo,[18] hasta ese momento en Zoom solo se cifraba la conexión entre el cliente y la plataforma, del mismo modo como se cifra la navegación en un sitio web que tiene *https*. La comunicación no se cifraba de extremo a extremo (e2e) sino

---

9 EPIC Files Complaint with FTC about Zoom
https://epic.org/2019/07/epic-files-complaint-with-ftc-.html

10 Working From Home? Zoom Tells Your Boss If You're Not Paying Attention https://www.vice.com/en_us/article/qjdnmm/working-from-home-zoom-tells-your-boss-if-youre-not-paying-attention

11 Beware of 'ZoomBombing': screensharing filth to video calls
https://techcrunch.com/2020/03/17/zoombombing/

12 Zoom iOS App Sends Data to Facebook Even if You Don't Have a Facebook Account
https://www.vice.com/en_us/article/k7e599/zoom-ios-app-sends-data-to-facebook-even-if-you-dont-have-a-facebook-account. Esta característica fue rápidamente removida, de acuerdo con la misma fuente https://www.vice.com/en_us/article/z3b745/zoom-removes-code-that-sends-data-to-facebook

13 Zoom is Leaking Peoples' Email Addresses and Photos to Strangers
https://www.vice.com/en_us/article/k7e95m/zoom-leaking-email-addresses-photos

14 https://twitter.com/c1truz_/status/1244737675191619584

15 Move Fast and Roll Your Own Crypto. A Quick Look at the Confidentiality of Zoom Meetings
https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/

16 Zoom admits some calls were routed through China by mistake
https://techcrunch.com/2020/04/03/zoom-calls-routed-china/

17 Zoom's Waiting Room Vulnerability
https://citizenlab.ca/2020/04/zooms-waiting-room-vulnerability/

18 Zoom Meetings Aren't End-to-End Encrypted, Despite Misleading Marketing
https://theintercept.com/2020/03/31/zoom-meeting-encryption/

solo el chat, es decir los mensajes de texto. De acuerdo con el reporte realizado por Citizen-Lab,[19] Zoom implementaba su propio protocolo de transporte, con algunas modificaciones sobre el estándar RTP (Real-Time Transport Protocol o Protocolo de transporte en tiempo real) existente, y se cifraba y descifraba todo el tráfico de medios con una única clave AES-128 (Advanced Encryption Standard o Estándar avanzado de cifrado, de 128bits), generada y distribuida por el servidor de la plataforma a las participantes, en modo ECB (Electronic Codebook o Libro de códigos electrónicos), considerado como muy débil dentro de los estándares existentes.

Para no extendernos mucho más en Zoom, vale decir que la empresa asumió un compromiso con la privacidad de sus usuarias y en abril puso en marcha un plan de 90 días para reparar errores y vulnerabilidades.[20] Sin embargo, los servicios de pago de esta y otras plataformas como Meet (de Google), Teams (de Microsoft) y Webex (de Cisco) continúan ofreciendo un mejor servicio en términos de calidad, estabilidad y privacidad.[21] Quizás por eso, con el avance de la pandemia, fueron estas las empresas que mejor respondieron a la demanda institucional de los servicios de videollamadas y videoconferencias, y son quienes hoy dominan el mercado. ¿Y qué pasa con las organizaciones, movimientos, grupos y personas que no pueden costear el acceso a los servicios ofrecidos por los grandes de internet?

Frente a las dificultades y riesgos asociados al creciente uso de plataformas digitales gratuitas, algunas organizaciones compartieron recomendaciones para el trabajo remoto[22] basadas en herramientas libres y respetuosas de la privacidad, donde Jitsi Meet aparecía como una de las mejores opciones para videollamadas.[23] Jitsi es un proyecto de código abierto que en 2003 empezó a desarrollar una aplicación de escritorio para transmisión de voz y mensajes de texto por internet. Con los años ha venido implementando diferentes tecnologías para

19 Move Fast and Roll Your Own Crypto... https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/

20 CEO Report: 90 Days Done, What's Next for Zoom https://blog.zoom.us/ceo-report-90-days-done-whats-next-for-zoom/

21 Zoom is Making Privacy and Security a Luxury https://foundation.mozilla.org/en/blog/zoom-making-privacy-and-security-luxury/ y Google Meet acabará con el 'gratis total' de los últimos meses, ¿cuándo https://cincodias.elpais.com/cincodias/2020/09/29/lifestyle/1601370323_374814.html

22 Conectadas y seguras en tiempos de cuarentena https://blog.torproject.org/Conectadas-seguras-tiempos-cuarentena, Recomendaciones de software libre para usar en contexto de distanciamiento físico (pero no social) https://www.vialibre.org.ar/2020/05/04/recomendaciones-de-software-libre-para-usar-en-contexto-de-distanciamiento-fisico-pero-no-social/, Recomendaciones para una mejor experiencia en línea https://ranchoelectronico.org/recomendaciones-cuarentena/, entre otras.

23 Vídeollamadas con Jitsi: la alternativa a las plataformas comerciales https://labekka.red/novedades/2020/04/21/jitsi.html, Alternativas a las reuniones en vivo https://mayfirst.coop/es/post/2020/node-167915/, ¿Qué está pasando con Zoom? https://sursiendo.org/blog/2020/05/que-esta-pasando-con-zoom/

integrar video y garantizar una comunicación fluida, que no requiera tantos recursos en los clientes finales.[24]

Como su código está abierto, es posible instalar instancias propias y muchas organizaciones lo hicieron durante la pandemia.[25] En Argentina se desarrolló Jitsimeter,[26] un comparativo sobre la calidad de las instancias y las condiciones de privacidad en que operan, a partir del uso de servidores intermedios, propiedad de grandes empresas del mercado de datos como Amazon, Google o Microsoft. Y es que la infraestructura detrás de una videollamada es mucho más compleja que levantar una instancia.

iii. Los engranajes de la infraestructura

La posibilidad de comunicarnos con audio y video en tiempo real es un proyecto que comenzó a finales de la década de 1980, cuando internet era una herramienta para conectar computadoras que pudieran intercambiar información digital entre ellas, con un uso principalmente militar y académico. Pero la lógica de internet es cambiante y no fue solo la Web la que permitió convertirse en una herramienta de comunicación a nivel mundial. El despliegue comercial de fibra óptica fue, tal vez, el factor más importante en el crecimiento exponencial de internet, ya que permitió transportar volúmenes de tráfico cada vez más grandes, a costos cada vez más bajos en comparación con los cables de cobre.

La fibra óptica permitió no solo el transporte de datos, sino también la transmisión de audio y, años después, video de alta calidad.[27] Si en sus inicios internet permitió el intercambio de correos electrónicos, desde 2010 la mayor parte del tráfico en internet corresponde a la transmisión de video y audio.

Aunque la base de usuarias también ha crecido exponencialmente, el mercado de internet está siendo monopolizado por cada vez menos empresas, que no solo desarrollan herramientas con las que interactuamos a diario (motores de búsqueda, plataformas de redes sociales o de trabajo colaborativo) sino que recolectan, alojan y capturan nuestros datos, al mismo tiempo que desarrollan y estandarizan las reglas con las cuales opera la infraestructura, para garantizar que toda esa información permanezca disponible en internet, pues nos hemos

24 Jitsi User FAQ https://jitsi.org/user-faq/

25 Maadix es un proveedor de infraestructura que ofrece herramientas de trabajo en línea al tempo que garantiza la autonomía, seguridad y privacidad de sus usuarias. A principios de abril dispusieron el servicio de instalar instancias propias de Meet Jitsi https://maadix.net/es/instala-jitsi-meet-con-un-clic y publicaron una serie de recomendaciones para optimizar su rendimiento https://maadix.net/es/optimizar-rendimiento-jitsi

26 Jitsimeter ¿Qué instancia de Jitsi me conviene usar? https://ladatano.partidopirata.com.ar/jitsimeter/

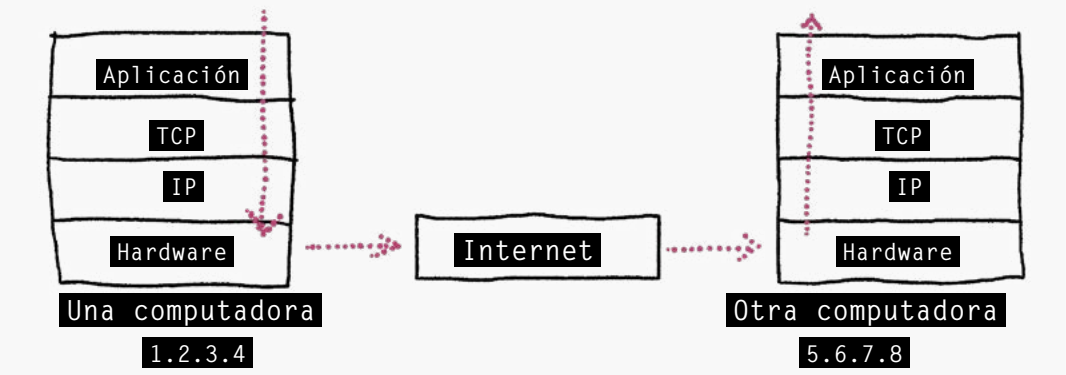
27 Clark, D. 2018. *Designing an Internet*. Massachusetts Institute of Technology.

acostumbrado a que prácticamente todo esté alojado en "la nube".

Pero internet no es una nube. No es etérea, es material y sólida. Aunque los dispositivos con que nos conectamos sean cada vez más pequeños y la información viaje a altísimas velocidades, se trata de un enorme complejo técnico y comercial. Y en tiempos de pandemia, cuando una parte importante de nuestras vidas transcurre en distintas pantallas, y "conectar a la otra mitad" es una prioridad para empresas y gobiernos, las preguntas por la soberanía sobre nuestra información —y por nuestra propia autonomía cuando interactuamos en línea— se vuelven urgentes.

¿Qué tanto sabemos de la información que sobre nosotras se captura e intercambia cada vez que hacemos una videollamada? ¿De quién son las redes por donde se transmite? ¿Quién instala, mantiene y accede a esas redes? Estas preguntas pueden sobrepasar nuestros intereses y capacidades si solo queremos sostener una reunión que no puede hacerse presencialmente; pero, precisamente por la necesidad en que nos pone este contexto, consideramos relevante mirar más allá de las opciones de software libre o infraestructuras alternativas, y entender mejor los estándares y protocolos que rigen el funcionamiento de internet.

Imagen 1 : Comunicación entre máquinas



Basado en Shuler, R. 2018. How Does the Internet Work? Standford University.

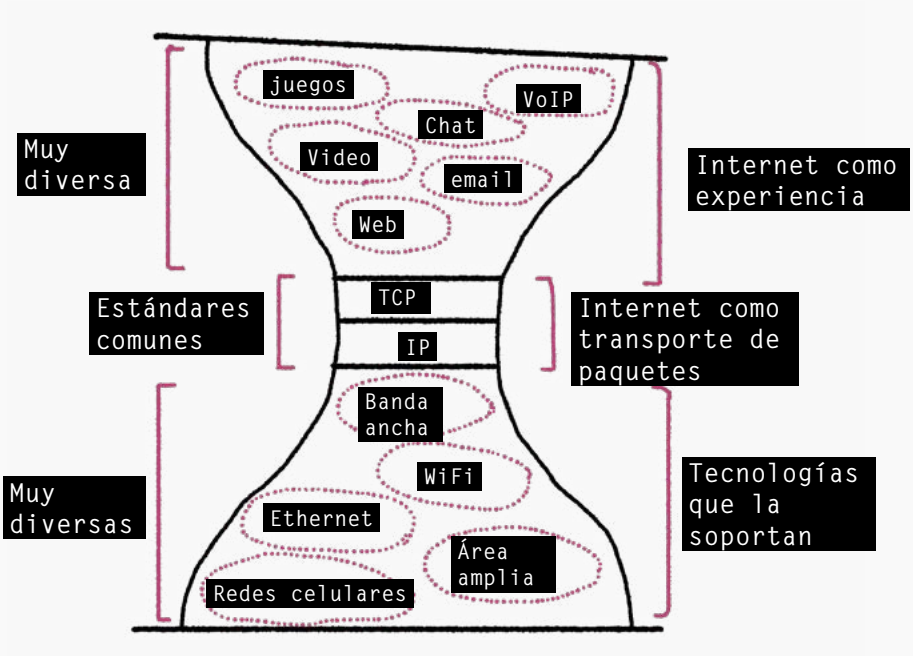
Para acceder a una plataforma web es necesario que muchas capas se entiendan. En la capa de aplicación corre el protocolo *HTTP* o *HTTPS,*[28] en la de transporte corre TCP (Transport Control Protocol o Protocolo de control del transporte) que se encarga de dirigir la información utilizando diferentes puertos (por ejemplo, el puerto 80 para *HTTP* y el 443 para *HTTPS*). En la capa de red cada dispositivo conectado obtiene una dirección IP (Internet Protocol o Protocolo de internet) que le identifica. Finalmente, el hardware convierte toda la

28 HTTPS agrega una capa de cifrado al protocolo de transferencia de hipertextos HTTP. A través de la generación de un certificado SSL (Secure Sockets Layer o Capa de conexión segura), se garantiza la integridad de la información compartida con un sitio web específico, así como la identidad del sitio y privacidad en la información. Mejor explicado (en inglés) en este comic https://howhttps.works

información de conexión en código binario.[29]

Podríamos decir que la red (IP) es hasta el día de hoy la base de cómo funciona internet. Junto con TCP, se han encargado de garantizar que muy distintos tipos de aplicaciones se comuniquen entre ellas, pero funcionen en conjunto utilizando diferentes tecnologías de comunicación, más o menos así:

Imagen 1 :Modelo de comunicación a través de internet



Basado en Clark, D. 2018. Designing an Internet. Massachusetts Institute of Technology.

Se supone que IP está en todo lo que pasa en internet, porque todo corre sobre IP. Pero como veremos más adelante, TCP no es la única forma de transporte que se puede emplear. TCP funciona en los nodos finales: es una información que llevan los paquetes, pero que no debería ser revisada por los *routers* intermedios, que solo deberían mirar la información IP. La idea del trabajo por capas es que la mayoría del trabajo ocurra en los puntos finales y no en la red.[30]

En la práctica, mientras se van desarrollando funcionalidades de internet cada vez más complejas, como la transmisión de audio y video en tiempo real, para garantizar que la mayoría del trabajo se desarrolle en los puntos finales parece necesario que quien diseña una aplicación no necesite conocer los detalles de cada tecnología que va a utilizar, sino simplemente las especificaciones necesarias para su trabajo. Para eso sirven los protocolos y estándares.

29 Shuler, R. 2018. How Does the Internet Work? Standford University. https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm

30 Aunque este principio no se cumple actualmente, porque la red está llena de intermediarios que analizan la información para reenviarla.

WebRTC es el estándar para las comunicaciones de audio, video y datos en tiempo real sobre la web, y es quizás una de las muestras más claras de cómo algo muy complejo en su lógica interna, resulta sencillo de implementar, tanto para quienes desarrollan aplicaciones de videollamadas (por ejemplo, Jitsi), para quienes desarrollan y mantienen navegadores (como Firefox o Chrome) y para las personas que utilizamos esas aplicaciones. A partir de ahora vamos a recorrer en profundidad el proceso de una videollamada y los protocolos involucrados.

Si bien los protocolos técnicos suelen referirse a "los clientes" como aquellos con el poder de comunicarse, haremos un esfuerzo grande por diferenciarlos de las usuarias, quienes interactuamos con esos clientes (sea una aplicación de videollamadas o de navegación) y que, finalmente, somos las interesadas en entablar una comunicación.

Audio, video y datos en tiempo real

WebRTC es un proyecto de código abierto impulsado inicialmente por Google en 2011. Su objetivo principal es permitir la transmisión en tiempo real de audio, video y datos genéricos entre navegadores, garantizando calidad y privacidad en las comunicaciones. El beneficio que ofrece WebRTC a una usuaria final es que puede establecer una comunicación desde su navegador sin necesidad de crear un perfil, instalar una aplicación ni descargar complementos o *plug-ins*.

Para lograr interoperabilidad entre diferentes navegadores (que, aunque sea de empresas distintas, puedan comunicarse entre ellos) el proyecto está basado en estándares abiertos, desarrollados en el World Wide Web Consortium[31] al nivel de API (Application Programming Interface o Interfaz de programación), y en la Internet Engineering Task Force[32] al nivel de protocolos.

Actualmente, los navegadores más utilizados soportan WebRTC, es decir, cuentan con la API necesaria para que se establezca una comunicación entre pares. Esto no quiere decir que los navegadores tienen esta capacidad en sí mismos, pues para garantizar una comunicación en tiempo real de buena calidad se requiere contar con altas velocidades de procesamiento de información, entre otros recursos con los que normalmente no cuenta un dispositivo casero como las computadoras, tabletas o celulares.

Por otra parte, si bien su objetivo principal es establecer una comunicación P2P entre dos o más navegadores, WebRTC también se puede implementar en una aplicación independiente que además puede integrarse con otros sistemas de comunicación existentes, tales como VoIP (voz sobre IP), clientes SIP (Session Iniciation Protocol o Protocolo de Inicio de Sesión) o PSTN (Public Switched Telephone Network o Red telefónica pública conmutada), tradicionalmente utilizados en el servicio de telefonía digital. Así, WebRTC no solo se trata de llevar la comunicación en tiempo real al navegador, sino también de llevar las capacidades de la web al mundo de las telecomunicaciones.

En el modelo WebRTC se espera que el navegador tenga la capacidad de trabajar en conjunto con servidores de respaldo que sí cuenten con recursos suficientes para implementar las funciones requeridas. Por eso, antes de comenzar cualquier transmisión, debe hacerse una señalización entre dispositivos, esto es, identificarse como los puntos finales que establecerán una comunicación entre pares (P2P), utilizando internet.

Una vez identificados los dispositivos, se abre una sesión WebRTC entre pares, que no utiliza TCP para el transporte, sino UDP (User Datagram Protocol o Protocolo de datagramas de usuaria), ya que, al tratarse de medios en tiempo real, es más importante que la información

31 WebRTC 1.0: Real-Time Communication Between Browsers https://www.w3.org/TR/webrtc/

32 Internet Engineering Task Force https://ietf.org/

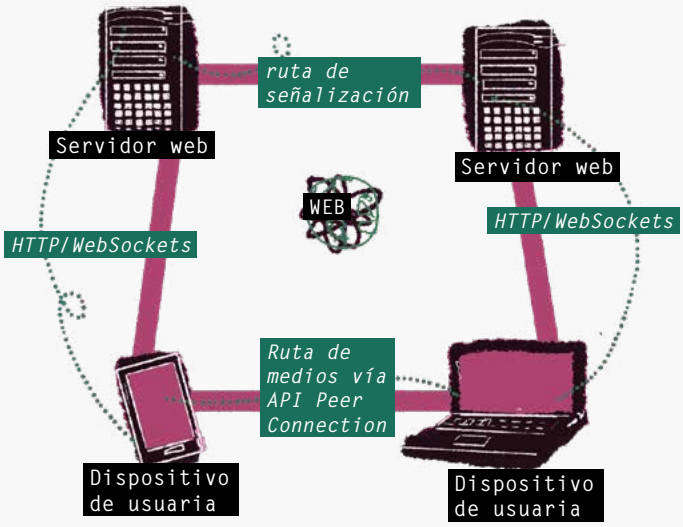
se transmita inmediatamente, y no que cada paquete sea fiable. A diferencia de TCP, UDP no ofrece ninguna promesa sobre la fiabilidad o el orden de los datos.

1. Señalización

Supongamos que un grupo de personas se dispone a comenzar una reunión. Al conectarse a la hora acordada, sus dispositivos enviarán una señal para identificarse entre ellos a través de la URL. Este proceso de señalización consiste en la búsqueda de un servidor intermedio que permita establecer un canal directo de transmisión entre los navegadores, que comenzarán luego un flujo de comunicación P2P.

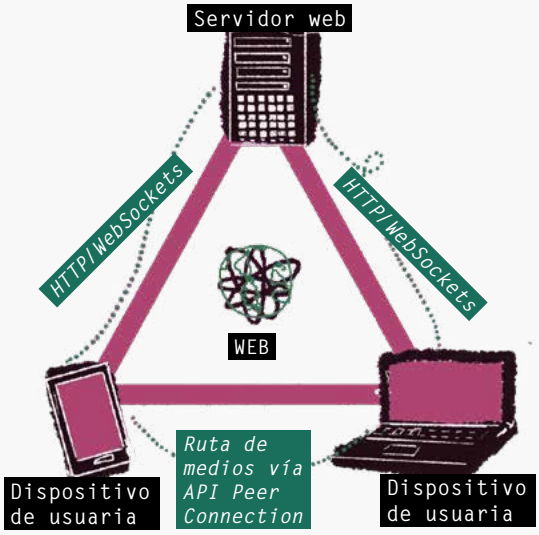
Este proceso no es parte de los estándares WebRTC, pero es necesario como paso previo para el establecimiento de una conexión P2P. Hasta este momento no se ha definido un mecanismo para el transporte de la información entre los navegadores conectados, pues el servidor intermediario no tiene capacidad de interpretar el contenido de los datos.[33] El intercambio de información ocurrirá a través de *RTCPeerConnection*, una vez creado el perfil de la sesión con el protocolo SDP. Mientras tanto, para la señalización se pueden utilizar distintos mecanismos como SIP sobre WebSockets, XMPP, MQTT o soluciones propietarias.[34] Este proceso puede hacerse utilizando uno o dos servidores intermedios, pero el modelo más común es el triangular.

Imagen: Modelo trapezoidal SIP



Ambos dispositivos ejecutan una aplicación web desde servidores diferentes. Basado en https://www.tutorialspoint.com/webrtc/index.htm

Imagen: Modelo triangular SIP



Ambos dispositivos ejecutan una sola aplicación web desde el mismo servidor. Basado en https://www.tutorialspoint.com/webrtc/index.htm

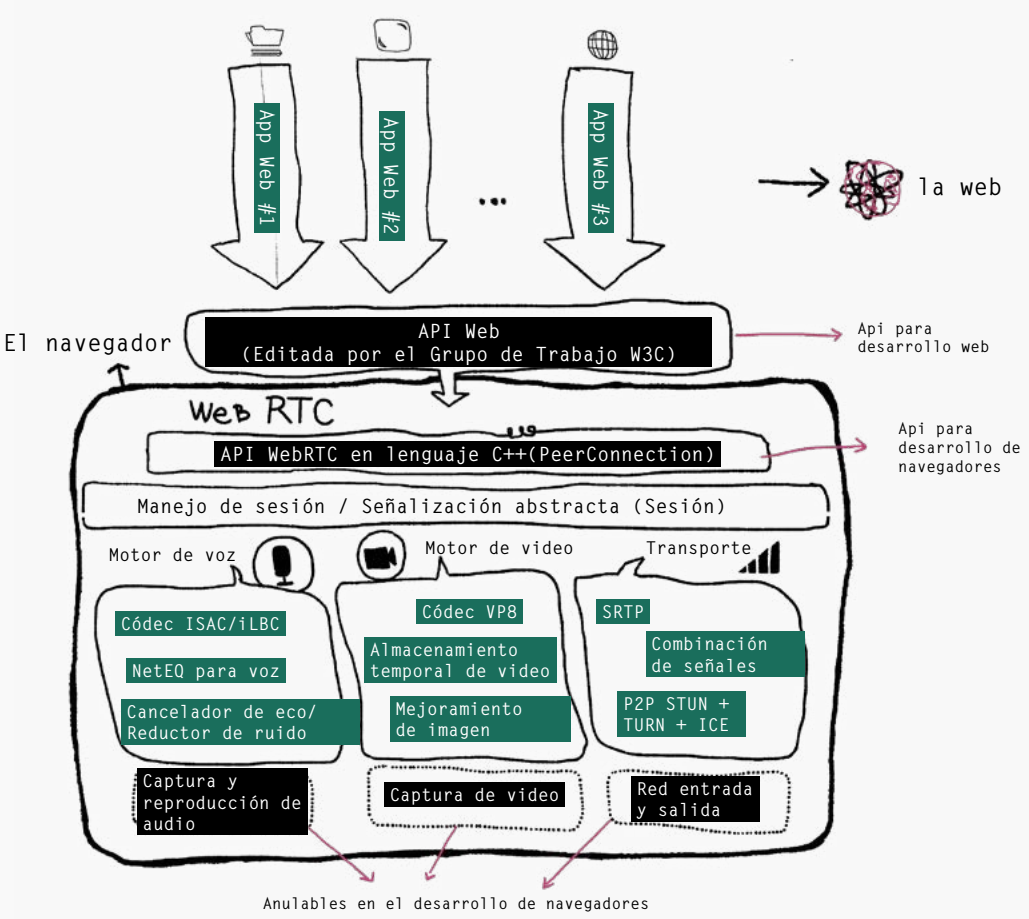
33 Signaling and video calling https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Signaling_and_video_calling

34 Sobre los servidores que intervienen en una sesión WebRTC https://bloggeek.me/webrtc-server/

2. Arquitectura de WebRTC

Una vez los dispositivos finales se han identificado entre ellos, se ejecuta la API,[35] cuyas piezas cumplen diferentes tareas para establecer los flujos de información y medios en tiempo real, directamente entre navegadores. La calidad de la transmisión de la que pueden gozar las personas usuarias depende del modo en que esté implementada la API, tanto en el navegador (por ejemplo, Firefox o Chrome) como en las aplicaciones (por ejemplo, Jitsi o Zoom), específicamente por la configuración de los códecs o formatos en que se hará la transferencia de audio y video.

Imagen 2: Arquitectura de la API WebRTC



Basado en https://webrtc.github.io/webrtc-org/architecture/

Además del establecimiento, gestión y mantenimiento de la sesión o canal de comunicación P2P (*RTCPeerConnection*) entre navegadores, en la API WebRTC se cumplen otras dos tareas principales: la captura, desde el navegador, de las pistas de audio y video que serán transmitidas (*MediaStream*), y la transmisión de datos diferentes a los de audio y video (*RTCDataChannel*). Además, se establecen los parámetros para el transporte de datos en tiempo real.

35 WebRTC - Architecture https://www.tutorialspoint.com/webrtc/webrtc_architecture.htm

2.1. RTCPeerConnection

Mucho de lo que se considera WebRTC está en el establecimiento de la P2P: el procesamiento de los protocolos SDP y ICE, que describiremos en el apartado siguiente; la gestión de una conexión UDP con otra usuaria; la posibilidad de comunicarse con una sesión de WebRTC a través de una llamada telefónica; la apertura de un canal de datos; la verificación de identidad de los pares conectados; el mantenimiento y monitoreo de la conexión, así como el cierre de la conexión una vez que no se necesite más; y el reporte de estadísticas.[36]

2.2. MediaStream

Permite capturar, desde el navegador local, tanto la cámara o como el micrófono del dispositivo, preguntándole antes a la usuaria si permite acceder a estos y, en caso de que haya más de una cámara o micrófono, escoger a cuál permite o no acceder. La interfaz *MediaStream* representa un flujo de medios, que puede consistir en varias pistas de video o audio, si es una sesión de varias participantes. El flujo se abre con la descripción de la sesión utilizando un servidor intermedio, pero una vez abierto, los medios se comparten a través de RTCPeerConnection, sin pasar por el servidor.

2.3. RTCDataChannel

Además del envío de medios P2P, en WebRTC también se pueden enviar bidireccionalmente datos que no requieren la negociación de códecs ni la sincronización de flujos. La tarea principal de *RTCDataChannel* es crear un canal que provenga de un objeto *RTCPeerConnection* existente. Utiliza la misma API que los *WebSockets* y tiene una latencia muy baja.

2.4. Códecs

En el contexto de WebRTC, un códec es una pieza de software cuya función es comprimir y descomprimir un flujo digital de medios (audio y video), desechando toda la información que no sea perceptible para el ojo o el oído de una persona, con el fin de que el proceso de codificación y decodificación ocurra en el menor tiempo posible (esto es, con baja latencia), pero cuidando que la transmisión sea clara para las participantes.

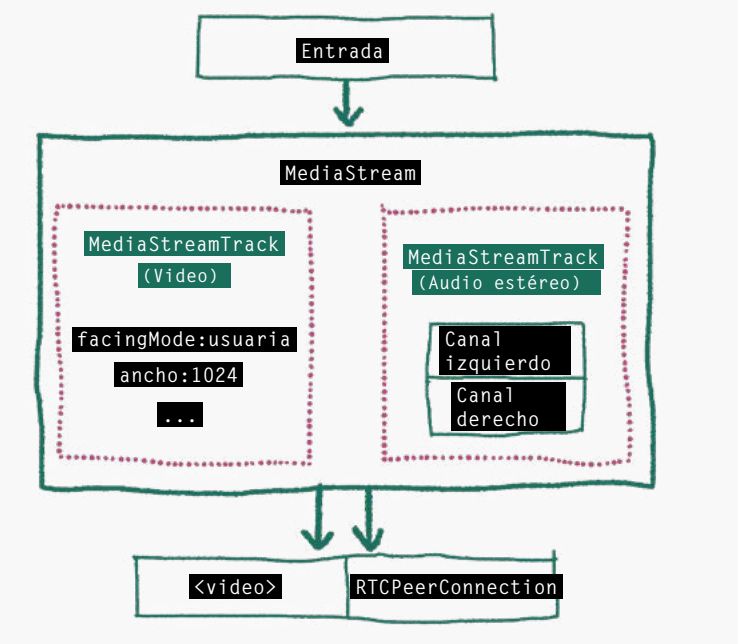
En general, para que los flujos de audio y video puedan ser almacenados o transmitidos, es necesario encapsularlos juntos en contenedores para que las personas usuarias se presentan como formatos o extensiones de archivo, por ejemplo, .mpg, .avi, .mov, .mp4, .rm, .ogg, .mkv. Para la transmisión en tiempo real, el objetivo es que las distintas pistas puedan ser sincronizadas, pues una misma usuaria podría querer compartir su cámara y su pantalla al mismo tiempo, o varias usuarias podrían estar compartiendo su cámara mientras se escuchan entre ellas.

Si bien el éxito de una transmisión WebRTC depende en gran medida de la calidad de conexión

36 RTCPeerConnection https://developer.mozilla.org/es/docs/Web/API/RTCPeerConnection

con que cuenten las participantes,[37] la configuración de códecs en los navegadores y en las aplicaciones, así como la negociación de códecs que se hace a través del protocolo SDP, permiten contar con una mejor calidad de transmisión utilizando la menor cantidad de recursos.

Imagen 3: Codificación de flujos en WebRTC



MediaStream sincroniza varias pistas de audio y video (MediaStreamTrack). Basado en https://hpbn.co/webrtc/

Aunque los códecs han ido evolucionando, el estándar de audio que más se utiliza hoy en WebRTC es Opus, de acuerdo con el RFC 7874,[38] aunque se contempla el uso de códecs adicionales para tener una mayor interoperabilidad, de acuerdo con el RFC7875.[39] Opus está diseñado para soportar aplicaciones de audio interactivas como VoIP, videoconferencia y chat de voz en juegos, entre otras. Este, como los demás códecs utilizados en WebRTC, se caracterizan por tener pérdidas, es decir, que no conservan toda la información original.

El estándar de video que más se utiliza hoy en WebRTC, VP8 + H264, fue desarrollado por Google y es de fuente abierta, por lo que se ha adoptado en distintas aplicaciones, no solo las desarrolladas con el estándar WebRTC. Actualmente el navegador Chrome, propiedad de Google, tiene implementado el códec VP9, el mismo utilizado en Youtube, también propiedad de Google.

37 Y, por ejemplo en muchos lugares la infraestructura disponible simplemente no es posible compartir cámara y pantalla al mismo

38 WebRTC Audio Codec and Processing Requirements https://tools.ietf.org/html/rfc7874

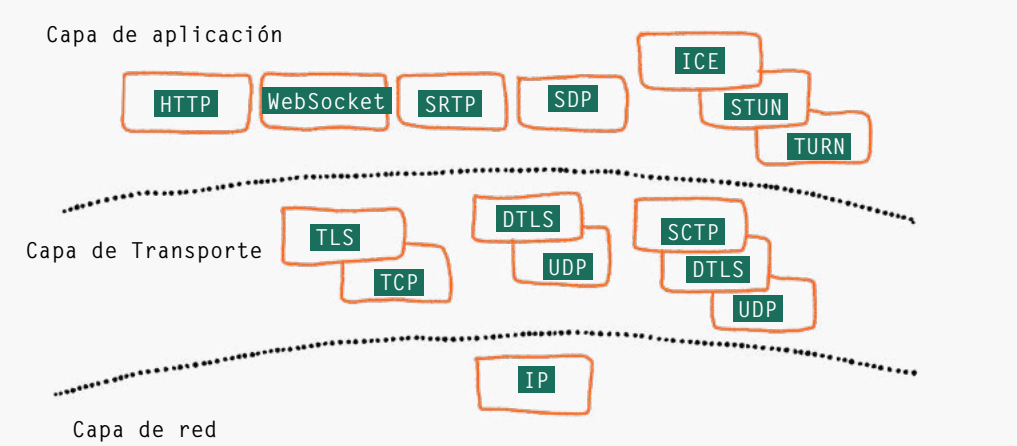
39 Additional WebRTC Audio Codecs for Interoperability https://tools.ietf.org/html/rfc7875

3. Protocolos para WebRTC

Hasta aquí hemos revisado la API WebRTC, que permite capturar audio y video en cada uno de los navegadores que estarán dentro de la sesión, con autorización de sus respectivas usuarias; comenzar un flujo de audio y video, es decir, reconocer y sincronizar las pistas (pueden ser dos de video y cuatro de audio, por ejemplo) con las mejores capacidades que tenga cada navegador; y adjuntar las pistas de medios que vayan a ser transmitidas. Además, una vez establecida la conexión, permite intercambiar y acordar detalles de comunicación entre navegadores (códecs, información sobre el ancho de banda y direcciones IP). Y —en paralelo a la transmisión de medios— permite también abrir un canal de datos entre los navegadores.

Para que este proceso sea posible, se utilizan distintos protocolos, tanto en la capa de aplicación como en la de transporte, que no han sido desarrollados específicamente para WebRTC, pues antes de su invención existían infraestructuras para la telefonía IP y la transmisión unidireccional de audio y video, entre otras funcionalidades. Por ello se han desarrollado las extensiones necesarias para soportar audio y video en tiempo real, en las condiciones definidas por el estándar WebRTC.

Imagen 4: Protocolos involucrados en WebRTC



Basado en WebRTC Tutorial, IETF 100. 2017. https://youtu.be/viZC1G4tmVM

Si bien WebRTC corre sobre el protocolo de transporte UDP, la señalización previa se hace sobre TCP. Con los protocolos NAT, STUN y TURN se establece y mantiene una conexión P2P sobre UDP. Pero, como veremos en seguida, ICE es el proceso mediante el cual esa interacción entre navegadores es posible, pues es el que procesa las solicitudes de establecimiento de conexión que registra cada navegador con el objeto *RTCPeerConnection*. Una vez que ese proceso se completa, se genera la oferta SDP y se utiliza el canal de señalización para alcanzar a los pares.

Paralelo a la transmisión de medios se abrirá un canal de datos entre los navegadores. Ese canal utiliza el protocolo de transporte SCTP para hacer control de flujo y congestión, y medir la calidad del servicio, teniendo en cuenta que el transporte sobre UDP no es fiable (a

diferencia de TCP), sino que se basa en el principio del "mejor esfuerzo". Adicionalmente, en todo el proceso están presentes los protocolos TLS, DTLS y SRTP, para garantizar la seguridad y privacidad de la información transmitida.

3.1. Capa de aplicación

Uno de los beneficios más grandes que ofrece WebRTC es que toda la gestión de una transmisión en tiempo real se hace desde el navegador web. Para las usuarias, esto facilita la posibilidad de acceder, aprender y utilizar este tipo de sistemas, y quizás por eso es que la comunidad *gamer* es la que más ha contribuido a su desarrollo e implementación. Desde el punto de vista técnico, esto implica que se deben desarrollar capacidades en el navegador que permitan abrir y mantener flujos estables de comunicación a través de la infraestructura de internet. Ese es el trabajo de los protocolos de aplicación.

3.1.1. NAT - Network Address Translation

RFC 2663 https://tools.ietf.org/html/rfc2663

NAT (Traducción de direcciones de red) existe para gestionar la limitada cantidad de direcciones IP que hay en la versión 4 del protocolo IP (IPv4). Cuando nos conectamos, nuestro dispositivo hace una solicitud a la empresa que nos presta el servicio de internet; es a través del *router* administrado por esa empresa que podemos acceder a una dirección IP pública y navegar. Las solicitudes se traducirán de la IP privada a la pública utilizando un puerto único.

Por seguridad, hoy muchos *router* domésticos sirven a la vez como cortafuegos y dispositivos NAT.[40] Además, existen distintos tipos de configuración de NAT, dependiendo de las restricciones para comunicar los dispositivos de la red privada local con los dispositivos externos.

Para el establecimiento de la interacción a través de ICE, es necesario enviar y recibir paquetes entre los dispositivos internos y externos, y esto se hace a través de STUN, pero la configuración de NAT simétrico no soporta ese protocolo, pues la traducción de la dirección IP privada a una pública está condicionada por la dirección IP de destino a la que se quiere enviar el tráfico. Para eso se utiliza TURN.

3.1.1. STUN - Session Traversal Utilities for NAT

RFC 5389 https://tools.ietf.org/html/rfc5389

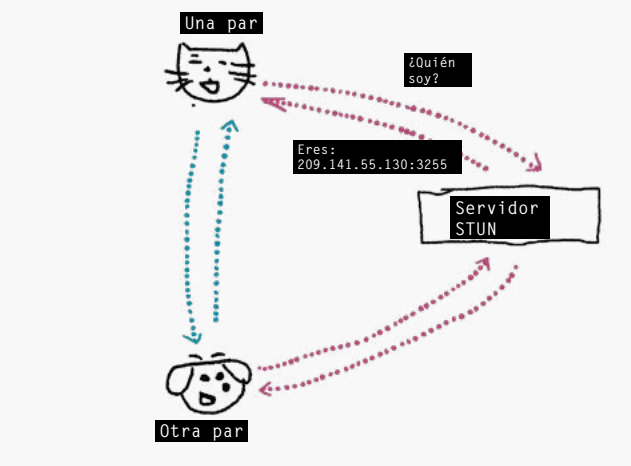
Las utilidades de sesión transversal para NAT es el protocolo que permite a una usuaria conocer la dirección IP pública con que navega en internet. Funciona bajo el modelo cliente/servidor, ya que permite a clientes NAT (como un navegador) encontrar su dirección IP pública, el tipo de NAT en que se encuentra y el puerto de internet asociado con el puerto

40 Clark, D. 2018. ibid, p. 25.

local a través de NAT.

En el contexto de WebRTC, esta información es usada para configurar una comunicación UDP entre dos dispositivos que se encuentren detrás de *routers* NAT. El software debe incorporar un cliente STUN que envía peticiones a un servidor STUN, el cual informa al cliente su IP pública y qué puerto ha sido abierto por NAT para permitir el tráfico entrante a la red del cliente. Los distintos tipos de NAT manejan los paquetes UDP entrantes de manera diferente, aunque normalmente se hace a través del puerto 3478 sobre UDP.

Imagen 5: Un servidor STUN descubre la IP pública del cliente



Basado en https://developer.mozilla.org/es/docs/Web/API/WebRTC_API

3.1.3. TURN - Traversal Using Relays around NAT

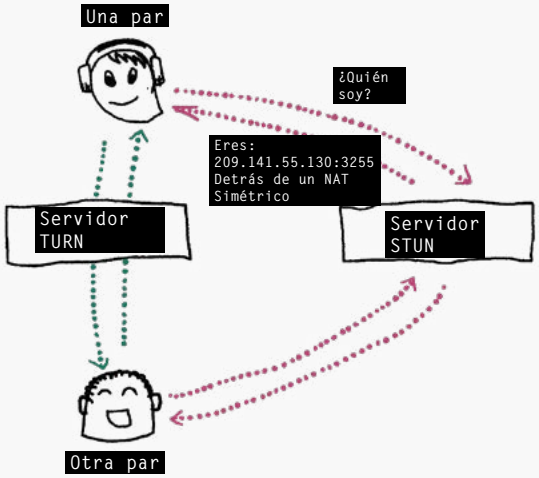
RFC 5766 https://tools.ietf.org/html/rfc5766

Cuando un *router* utiliza NAT simétrica o un sistema cortafuegos, el protocolo de desplazamiento con Relay NAT (mejor conocido como TURN) permite eludir estas restricciones y utiliza un tercer servidor para retransmitir todos los mensajes entre dos clientes. Para eso, el cliente debe conectarse al servidor TURN y es ese servidor el que se conecta al destino en su nombre, retransmitiendo los paquetes.

Si bien este proceso conlleva un mayor gasto de recursos y por lo tanto se usa solo como última alternativa, hoy la mayoría de las conexiones tienen protecciones de seguridad, así que prácticamente cualquier servicio WebRTC debe soportar el uso de TURN. Para optimizar recursos, también se han desarrollado servidores COTURN[41] que hacen de TURN y STUN a la vez.

41 Free open source implementation of TURN and STUN Server https://github.com/coturn/coturn

Imagen 6: Un servidor TURN soluciona la restricción NAT simétrico



Basado en https://developer.mozilla.org/es/docs/Web/API/WebRTC_API

3.1.4. ICE - Interactive Connectivity Establishment

RFC 5245 https://tools.ietf.org/html/rfc5245

El protocolo para el establecimiento de conectividad interactiva es un proceso mediante el cual un navegador web se conecta con otros, identificando una ruta fiable para hacerlo. La oferta de conectividad que se establece en el objeto *RTCPeerConnection* contiene una lista de IP candidatas, así como títulos de puertos de los que dispone una entidad remota. Con esto, el agente de ICE puede verificar las condiciones de conectividad para ver si puede alcanzar a la otra entidad.

El proceso es más o menos así: el agente de ICE envía una solicitud de unión STUN que la otra entidad debe reconocer con una respuesta exitosa de STUN. Si esto se completa, se abre una vía para la conexión P2P. Si los candidatos fallan, puede pasar que la *RTCPeerConnection* se marque como fallida o que la conexión recaiga en un servidor de retransmisión TURN para establecer la conexión.

El agente ICE clasifica y prioriza automáticamente el orden en que se realizan las comprobaciones de la conexión de las entidades candidatas: primero se comprueban las direcciones IP locales, luego las IP públicas a través de STUN, y como último recurso se utiliza TURN. Una vez establecida la conexión, el agente de ICE continúa emitiendo solicitudes periódicas de STUN al otro par. Esto sirve para mantener viva la conexión y para ver si se puede ofrecer un mejor rendimiento a través de una ruta alternativa.

3.1.5.SDP - Session Description Protocol

RFC 4566 https://tools.ietf.org/html/rfc4566

Una sesión se describe con una serie de atributos. Cada atributo en una línea, por ejemplo, así:

v= (Versión del protocolo)
o= (Origen e identificador de sesión)
s= (Nombre de sesión)
i=* (Información de la sesión)
u=* (URI de descripción)
e=* (Correo electrónico)
p=* (Número telefónico)
c=* (Información de conexión)
b=* (Cero o más líneas con información de ancho de banda)
Una o más líneas de descripción de tiempo (Ver abajo "t=" y "r=")
z=* (Ajustes de zona horaria)
k=* (Clave de cifrado)
a=* (Cero o más líneas de atributos de sesión)
Cero o más descripciones de medios

SDP se encarga de describir el perfil de una sesión. Este protocolo es ampliamente utilizado para la transmisión en tiempo real, y en el marco de WebRTC se utiliza junto con SIP, principalmente para definir cómo codificar los medios (audio y video) que luego serán transmitidos utilizando SRTP.

Este protocolo permite hacer una negociación bajo el modelo de oferta/respuesta, reconociendo las capacidades de cada una de las entidades que participará para establecer los parámetros con los cuales se abrirá una sesión. Su función no es entregar contenidos, sino entablar una negociación para definir qué códecs utilizar, con qué ancho de banda se puede hacer la conexión y cuáles son las IP candidatas para conectarse.

Una vez que se ha creado el *RTCPeerConnection,* es necesario crear la cadena de textos de oferta/respuesta SDP, tanto para la entidad que llama como para la que recibe. Cuando ya se reconocen ambas entidades, el servidor que hizo posible dicha conexión pierde el control sobre la sesión y se inicia la conexión directa entre pares que correrá sobre el protocolo de transporte UDP. De esto se encarga ICE. La comunicación durará mientras haya flujo de datos.

3.2. Capa de transporte

En el contexto de WebRTC, en la capa de transporte se cumplen funciones de señalización, control de congestión y gestión de la cola en el tráfico, con el fin de garantizar una buena calidad en el servicio. Si bien en esta capa se ha desarrollado toda un área de trabajo en el transporte en medios en tiempo real (que normalmente van encapsulados en UDP), los protocolos que intervienen en WebRTC deben dar soporte a un servicio de comunicaciones en

tiempo real que corre encima de la web, que a su vez corre sobre TCP.

3.2.1. TCP – Transport Control Protocol

RFC 793 https://tools.ietf.org/html/rfc793

El Protocolo de control de transmisión es tan antiguo como internet. Fue diseñado para satisfacer necesidades concretas de los sistemas de comunicación en el campo militar, es decir, en un entorno susceptible de ser atacado. Por eso TCP está orientado a la conexión: para ejecutarse requiere la sincronización previa de las partes que se van a comunicar. Además, está diseñado para que la información sea transmitida de manera confiable de extremo a extremo, y para eso emplea un sistema de verificación cada vez que se envía y recibe un paquete. Aplicaciones como la web, el correo electrónico, FTP (para compartir archivos) o SSH (para conectarse remotamente a un servidor) corren sobre TCP.

Para el transporte, TCP organiza y envía individualmente cada byte, garantizando que lleguen a su destino todos, en orden y sin errores. Además, con el sistema de puertos, permite transportar simultáneamente datos de distintas aplicaciones. Inicialmente TCP e IP eran un mismo protocolo base de internet. Pero, dada su complejidad, TCP muchas veces presentaba un retraso en el envío de paquetes, lo cual no resultaba útil, por ejemplo, para la transmisión de audio. Es por eso que a finales de la década de 1970 se separó la capa de red (IP) de la de transporte (TCP) y se comenzó a desarrollar el protocolo UDP.[42]

3.2.2. UDP – User Datagram Protocol

RFC 768 https://tools.ietf.org/html/rfc768

Publicado inicialmente en 1980, UDP (protocolo de datagramas de usuario) está orientado a las transacciones, es decir que para ejecutarse no requiere el establecimiento previo de una conexión. Protocolos de aplicación como DNS (para la resolución de nombres de dominio), DHCP (para la asignación de direcciones IP privadas en redes locales) o RIP (con información para el enrutamiento de paquetes) trabajan sobre UDP, pues requiere un mínimo de recursos de red para ejecutarse.

Como su nombre lo indica, UDP trabaja con mensajes, es decir, con datagramas o paquetes de bytes, no con bytes individuales. Esto le permite ser más ágil, porque no garantiza el orden en que llegan los paquetes a su destino, ni tampoco que lleguen.

Los paquetes RTP (Real Time Transport Protocol o Protocolo de transporte en tiempo real) viajan encapsulados en UDP. Publicado inicialmente en 1996, RTP ha servido al desarrollo de sistemas de comunicación y transmisión de medios, incluyendo telefonía sobre IP y sistemas de televisión, entre otros sistemas. En el estándar WebRTC se utiliza el protocolo SRTP

42 Clark, D. 2018. ibid.

(Secure Real-time Transport Protocol o Protocolo seguro de transporte en tiempo real) ya que el cifrado es obligatorio.

3.2.3. SCTP - Stream Control Transmission Protocol

RFC 4960 https://tools.ietf.org/html/rfc4960

Desarrollado para la señalización del transporte en redes de telefonía pública conmutada (PSTN) y publicado inicialmente en 2000, SCTP funciona directamente sobre el protocolo IP. En la capa de transporte es una alternativa a TCP y UDP, pues está orientado a la conexión, provee confiabilidad, control de flujo y secuenciación de paquetes, como TCP. Pero, de manera similar a UDP, utiliza delimitadores de mensajes, no de bytes, para garantizar la llegada de toda la información, permitiendo su envío en desorden, lo que hace el transporte más eficiente.

Se trata de un protocolo mucho menos complejo que TCP y que, sin embargo, permite controlar la congestión y mejorar la tolerancia de fallos durante el envío de paquetes, ofreciendo soporte *multihoming* (conexión simultánea a varias redes) y *multistreaming* (varios flujos de datos por el mismo puerto, para que no se bloquee la comunicación si hay un fallo), garantizando mayor seguridad en la comunicación con un *handshake* de cuatro vías que incluye una *cookie* de autenticación y una etiqueta de verificación obligatoria en la cabecera de cada paquete enviado.

Paralelo a la transmisión P2P de medios, En WebRTC se abre un canal de datos entre los navegadores. Ese canal utiliza SCTP para hacer control de flujo y congestión, pero aquí SCTP se conecta a través de un túnel DTLS, para garantizar la confidencialidad de la información. A su vez, DTLS se ejecuta sobre UDP, que provee el transporte a través de NAT una vez se ha abierto un canal mediante ICE.

4. Seguridad y privacidad en WebRTC

WebRTC se ha desarrollado pensando en la facilidad de acceso a transmisión de audio y video en tiempo real. Por eso el estándar propone que corra sobre la web y no requiera la instalación de *plug-ins* o aplicaciones específicas. Teniendo en cuenta los riesgos asociados a esta facilidad de acceso para las usuarias, en WebRTC el cifrado es una característica obligatoria y por ello la seguridad se basa principalmente en los protocolos DTLS y SRTP, y exige que los navegadores implementen la gestión de autorización de acceso a la cámara y el micrófono.

Si bien WebRTC es cuidadoso en la configuración de seguridad y privacidad para la transmisión de medios, el proceso previo de señalización quedó fuera del estándar. Sin embargo, se basa en la exposición de capacidades y flujos locales por parte de clientes finales, tanto de navegadores como de dispositivos. Esto supone un ejercicio de confianza tanto de quienes desarrollan aplicaciones como de quienes las utilizamos, pues en el establecimiento de la

conexión necesariamente intervienen terceros (en este caso servidores) con acceso a la información de las participantes.

Este ha sido un problema permanente en el desarrollo de extensiones y protocolos para la implementación de WebRTC. En IETF se conformó un grupo de trabajo para el mejoramiento de la privacidad en conferencias basadas en el protocolo RTP,[43] que trabaja específicamente sobre el protocolo SRTP, pero también sobre SIP (protocolo ampliamente utilizado en la señalización de aplicaciones WebRTC). Las consideraciones de seguridad relacionadas con el conjunto de API y protocolos utilizados por WebRTC se describen un *Internet-draft* próximo a ser publicado como RFC.[44]

4.1. DTLS - Datagram Transport Layer Security

RFC 6347 https://tools.ietf.org/html/rfc6347

Este protocolo proporciona privacidad en las comunicaciones y previene su interceptación y manipulación. Está basado en el protocolo TLS (Transport Layer Security o Seguridad en la capa de transporte), que es un extendido protocolo de seguridad para comunicaciones. La principal diferencia entre estos dos protocolos es que TLS corre sobre TCP y DTLS sobre UDP. Actualmente DTLS se encuentra en la versión 1.2, publicada en 2012.

Durante el proceso ICE, los datos se cifran utilizando DTLS, que debe estar integrado en todos los navegadores que soportan WebRTC. Este se utiliza para asegurar todas las transferencias de datos entre pares.

4.2. SRTP -Secure Real-Time Transport Protocol

RFC 3711 https://tools.ietf.org/html/rfc3711

Aparte de DTLS, WebRTC también cifra los datos de video y audio a través de SRTP, para garantizar que terceros sin autorización puedan escuchar o ver las transmisiones, y minimizar los riesgos de ataques como denegación de servicio. Publicado en 2004, SRTP establece un sistema de cifrado y autenticación del tráfico en los protocolos RTP y RTCP.

RTP fue publicado inicialmente en 1996 y actualizado en 2003. Es uno de los fundamentos técnicos de la VoIP, así que está implementado en muchos otros sistemas de comunicación además de WebRTC. RTP corre sobre UDP y se utiliza en conjunto con RTCP (Real-Time Control Protocol o Protocolo para el control de tiempo real), que permite hacer seguimiento y monitoreo al envío de paquetes. Mientras que RTP transmite contenidos, RTCP captura estadísticas de transmisión y calidad del servicio, al tiempo que ayuda a sincronizar múltiples transmisiones.

43 Privacy Enhanced RTP Conferencing (perc) https://datatracker.ietf.org/wg/perc/about/

44 WebRTC Security Architecture. https://tools.ietf.org/html/draft-ietf-rtcweb-security-arch-20

4.3. Permisos en el navegador o aplicación web

De acuerdo con el RFC7478 sobre casos de uso y requerimientos en comunicaciones en tiempo real sobre web,[45] el navegador que establece una comunicación por WebRTC debería proveer varios mecanismos que garanticen el consentimiento de acceso a la cámara, al micrófono y a la pantalla, por parte de las usuarias. Normalmente esto se implementa a través de un mensaje donde se puede aceptar o denegar el acceso. Además, los navegadores deberían implementar algún mecanismo para informar cuando la cámara y el micrófono estén siendo usados. Usualmente esto se hace a través de un ícono. Además, las usuarias deberían poder revisar y revocar este permiso en cualquier momento y, para eso, las aplicaciones que implementen WebRTC deberían asegurarse de que sus usuarias consienten el establecimiento de una comunicación entre ellas, tanto para recibir como para enviar cualquier flujo de datos.

4.4. Sobre el cifrado de extremo a extremo

WebRTC puede ser usado para comunicaciones entre dos personas o entre grupos más grandes y la implementación de protocolos de seguridad DTLS y SRTP será diferente en cada caso. En las comunicaciones entre dos personas (P2P), la comunicación se cifra de extremo a extremo (e2e) usando los protocolos DTLS y SRTP, como se detalla en el RFC5763,[46] incluso si el envío de paquetes pasa por servidores intermedios, por ejemplo, servidores TURN.

En cambio, cuando las comunicaciones se establecen entre más de dos personas (sesiones *multiparty*), la capa de cifrado que proporciona DTLS y SRTP se elimina cuando los paquetes atraviesan los servidores intermedios. Algunos servicios de videollamadas, como Jitsi, están probando una implementación de cifrado punto a punto en estas sesiones grupales,[47] a partir de la API WebRTC Insertable Streams (API de flujos insertables de WebRTC).[48]

45 Web Real-Time Communication Use Cases and Requirements. https://tools.ietf.org/html/rfc7478

46 Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS) https://tools.ietf.org/html/rfc5763

47 This is what end-to-end encryption should look like! https://jitsi.org/blog/e2ee/

48 WebRTC Insertable Streams https://www.chromestatus.com/feature/6321945865879552 Avances para cifrado punto a punto en https://webrtchacks.com/true-end-to-end-encryption-with-webrtc-insertable-streams/ y https://webrtchacks.com/you-dont-have-end-to-end-encryption-e2ee/

5. WebRTC en los Navegadores

Para la implementación de aplicaciones que utilicen protocolos de tiempo real,[49] siguiendo la estructura de capas, los navegadores deberían contar con estas funcionalidades:

Apoyo al sistema local	No necesitan ser especificadas de manera uniforme pues cada participante puede elegir cómo hacerlo, sin afectar la transmisión. Por ejemplo: cancelación de eco, mecanismos locales de autenticación y autorización, acceso al sistema operativo, capacidad de grabar localmente.
Presentación y control	Para asegurar que las interacciones no se comportan de manera sorpresiva, para lo cual se requiere cooperación de las participantes. Muchas aplicaciones han sido construidas sin interfaces estandarizadas para estas funciones. Por ejemplo: control del suelo, disposición de la pantalla, activación por voz de la conmutación de imágenes, entre otras.
Gestión de la conexión	Establecimiento de conexiones, acuerdo sobre formatos de datos, cambios en los formatos de datos durante una llamada. Protocolos para la señalización como SDP, SIP, y Jingle/XMPP pertenecen a esta categoría.
Formatos de datos	Especificaciones de códecs para audio y video, así como de formato y funcionalidad para los datos que pasan entre los sistemas.
Encuadre de datos	Protocolos como RTP, SRTP, DTLS y otros. Sirven como contenedores y garantizan la confidencialidad e integridad de los datos.
Transporte de datos	Protocolos como TCP, UDP, SCTP y los medios para establecer conexiones de forma segura entre las participantes, así como funciones para decidir cuándo enviar datos: gestión de la congestión, ancho de banda y estimación.

WebRTC promete interoperabilidad y facilidad para establecer conexiones de video y audio entre navegadores, pero no encontramos una documentación clara y actualizada sobre qué tan bien nos podemos conectar desde determinados sistemas operativos y navegadores web. Según Wikipedia —y en este punto es importante decir que la información técnica sobre comunicaciones en tiempo real es muy clara y completa, sobre todo la que está en inglés—[50] WebRTC es compatible con los siguientes navegadores:

En computadoras	En dispositivos Android	En dispositivos con iOS
Microsoft Edge 12+	Google Chrome 28+	MobileSafari/WebKit (iOS 11+)
Google Chrome 28+	Mozilla Firefox 24+	
Mozilla Firefox 22+	Opera Mobile 12+	
Safari 11+		
Opera 18+		
Vivaldi 1.9+		

49 Overview: Real Time Protocols for Browser-based Applications https://datatracker.ietf.org/doc/draft-ietf-rtcweb-overview/

50 WebRTC https://en.wikipedia.org/wiki/WebRTC#support

Sin embargo, parece una información desactualizada y con falta de referencias, que contradice los datos de otros proyectos como caniuse.com[51] y las pruebas realizadas por nosotras mismas. caniuse.com ofrece una comparativa más detallada y reporta específicamente que WebRTC no está soportado por los navegadores Internet Explorer, UC Browser y Opera Mini. Otras referencias en internet[52] especifican que WebRTC es compatible con Chrome, Mozilla Firefox, Safari, Opera y otros navegadores basados en Chrome, sin dar muchos más detalles.

Para aportar más datos actualizados sobre qué navegadores soportan WebRTC, elaboramos una tabla.

Tabla 1. Soporte WebRTC en navegadores[53]

Navegador *(porcentaje de uso*)*	Chrome *(63,97 %)*		Safari *(16,96%)*		Firefox *(4,44%)*		Samsung Internet *(3,39%)*		UC Browser *(2,69 %)*		Opera *(2,2%)*		Edge *(2,11%)*		IE *(1,79%)*	
Sistema operativo	Pruebas	Versión	P	V	P	V	P	V	P	V	P	V	P	V	P	V
Windows 10	❶❷❸	83	⊗	⊗	❶❷❸	78	⊗	⊗	❶❷❸	13	❶❷❸	69	❶❷❸	83	❶❷❸	11
Debian 10	❶❷❸	83	⊗	⊗	❶❷❸	68	⊗	⊗	⊗	⊗	❶❷❸	69	⊗	⊗	⊗	⊗
MacOS	❶❷	83	❶❷	13	❶❷		⊗	⊗	⊗	⊗	❶❷	69	❶❷	83	⊗	⊗
Android	❶❷	83	⊗	⊗	❶❷	68	❶❷	12	❶❷	13	❶❷	Touch 2	❶❷	45	⊗	⊗
iOS	❶❷	83	❶❷	13	❶❷	28	⊗	⊗	❶❷	13	❶❷	Touch 2	❶❷	45	⊗	⊗
Basado en Chromium	*Sí*		*No*		*No*		*Sí*		*No*		*Sí*		*Sí*		*No*	

Leyenda:

- ❶ prueba de llamada con jitsi
- ❷ prueba de llamada con bbb
- ❸ test wpt
- ⊗ No existe este navegador para este sistema operativo

Código de colores:

Verde	Funciona
Amarillo	Funciona con errores
Rojo	No funciona

Web-platform-tests ofrece un desarrollo público de pruebas para los estándares web[54] que pueden ser ejecutados en el navegador que elija. Allí hay una serie de tests para WebRTC,[55]

51 WebRTC Peer-to-peer connections https://caniuse.com/#feat=rtcpeerconnection

52 Who Supports WebRTC? https://www.3cx.com/webrtc/which-browsers-support-webrtc/; Which web browsers are currently supporting WebRTC? https://support.pexip.com/hc/en-us/articles/216077528-Which-web-browsers-are-currently-supporting-WebRTC

53 Porcentaje de uso según: https://gs.statcounter.com/browser-market-share#monthly-201906-202006-bar

54 The Web platform: Browser technologies https://platform.html5.org/

55 Directory listing for /webrtc/ https://wpt.live/webrtc/

que realizamos siguiendo las indicaciones[56] y recogemos en las tablas 2 y 3.

Tabla 2. Tests de web-platform-tests en Windows 10

Navegador	Versión	Test superados	Test fallados
Firefox	78	1058	706
Chrome	83	1288	304
Edge	83	1270	314
Opera	69	1244	313
UC Browser	6	87	17

Tabla 3. Tests de web-platform-tests en Debian 10

Navegador	Versión	Test superados	Test fallados
Firefox	68	Error al ejecutar	Error al ejecutar
Chrome	83	1334	332
Chromium	83	1221	310
Opera	69	1264	307

A partir de los datos obtenidos en nuestras pruebas, podemos concluir que WebRTC no está plenamente soportado por los navegadores más utilizados a nivel mundial, mientras que Chrome y los navegadores basados en su código se destacan por su compatibilidad.

Frente a estos resultados nos parece importante saber más acerca de qué dificultades están encontrando las desarrolladoras de navegadores para hacerlos compatibles con WebRTC, pero también por qué los navegadores basados en Chrome soportan mejor WebRTC y cómo se traduce, en términos de consumo, la compatibilidad que ofrece Chrome y otros navegadores basados en su código.

Teniendo en cuenta que Google fue la marca que impulsó WebRTC como estándar abierto, que hoy sigue siendo uno de los principales promotores del proyecto y que sus principales servicios de videollamadas (Google Hangouts, Google Meets y Google Duo) estén basados en WebRTC, nos preguntamos: ¿Cómo están influyendo las grades corporaciones en el desarrollo de un estándar como WebRTC? ¿Cómo afectar esto en nuestra libertad y autonomía como usuarias, a la hora de elegir qué software utilizar para hacer videollamadas?

56 Running Tests from the Local System. https://web-platform-tests.org/running-tests/from-local-system.html

6. Y todo esto, ¿para qué?

En mayo de 2020, IETF retomó un trabajo comenzado en 2017 para cifrar e2e los flujos de audio y video cuando necesariamente debe haber un servidor intermedio, como ocurre con las videollamadas grupales. Hasta ahora, algunas soluciones de cifrado se habían implementado en aplicaciones específicas y algunos navegadores le han dado soporte, pero todavía no hay un estándar abierto al respecto.[57] La conversación se puede seguir en una lista abierta de correo,[58] a la que le vendría bien mayor diversidad en la participación.

Sumarse a una conversación técnica es difícil, más cuando hay desacuerdos y divergencias. Sin embargo, para transformar algo es necesario entender cómo funciona, o al menos plantearnos la pregunta. Este documento, y el ejercicio previo de indagación y comprensión por parte de quienes lo trabajamos, tiene ese fin.

La web está llena de información sobre cómo funciona WebRTC. Casi toda en inglés y dirigida a desarrolladores interesados en implementar o adaptar el estándar a sus necesidades. Este documento, con sus fallas y aciertos, es el resultado de un ejercicio que buscaba reunir toda esa información y explicarla de acuerdo con un orden coherente para nosotras, usuarias con muy distintas capacidades técnicas, esperando que pueda ser de interés y utilidad para nuestras compañeras.

Porque si durante la pandemia logramos continuar con nuestros procesos de organización gracias al uso de herramientas digitales, especialmente de las videollamadas, la exposición de nuestras voces y cuerpos en las pantallas también nos hizo objeto de ataques.[59] Este, por supuesto, no es un escenario nuevo. La pandemia solo hizo más evidentes las violencias a las que mujeres y diversidades sexogenéricas estamos expuestas. Violencias que, por supuesto, se intensifican en diversas condiciones de raza, clase, capacidades, edad y ubicación geográfica.[60]

Y si las herramientas que utilizamos para trabajar, organizarnos y difundir ideas son las mismas con las que sostenemos relaciones afectivas de distintos tipos en la distancia, reclamar privacidad implica, necesariamente, reclamar el control sobre nuestra información. Si lo personal es político, ¿también debería ser público? Si hacemos una videollamada segura, ¿qué tan seguras estamos de que nuestra información está protegida? ¿Protegida de quién o de qué? ¿En manos de quién estamos delegando esa protección?

---

57 Secure Frames (SFrames): end-to-end media encryption with #webrtc now in chrome. https://webrtcbydralex.com/index.php/2020/03/30/secure-frames-sframes-end-to-end-media-encryption-with-webrtc-now-in-chrome/

58 Frame-based end-to-end encryption of real-time media https://www.ietf.org/mailman/listinfo/sframe

59 Trolls pandémicos https://www.pikaramagazine.com/2020/05/trolls-pandemicos/

60 La otra pandemia: internet y violencia de género en América Latina https://www.derechosdigitales.org/14716/la-otra-pandemia-internet-y-violencia-de-genero-en-america-latina/

Con o sin pandemia, hay muchas estrategias sobre las que podemos trabajar[61] para la eliminación de violencias basadas en sistemas tradicionales de opresión. Entender cómo funcionan nos permite imaginar sistemas otros, donde el sometimiento no sea la regla, ni en su uso ni en su desarrollo.

El diseño de complejísimos sistemas de comunicación nos pone cada vez más lejos de esa posibilidad. ¿Es posible comunicarnos con sistemas menos complejos? ¿Es posible hacer más visible y legible su complejidad?

Dejamos las preguntas abiertas.

61 Emergencia.Acoso.Online. Materiales disponibles para saber qué hacer ante un caso de difusión de imágenes íntimas sin consentimiento u otro tipo de violencia de género en línea. https://acoso.online/cl/emergencia/

@ | DERECHOSDIGITALES
Derechos Humanos y Tecnología en América Latina