



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



FACULTAD DE  
INGENIERÍA

# Generación de un renderizador acústico

Informe de Proyecto de Grado presentado por

Agustín Bologna, Agustín dos Santos, Stefano Graziani

en cumplimiento parcial de los requerimientos para la graduación de la carrera de  
Ingeniería en Computación de Facultad de Ingeniería de la Universidad de la  
República

Supervisor

Eduardo Fernández

Montevideo, 14 de octubre de 2024



Generación de un renderizador acústico por Agustín Bologna,  
Agustín dos Santos, Stefano Graziani tiene licencia [CC Atribución 4.0](#).

# Agradecimientos

En primer lugar, queremos extender nuestros más sinceros agradecimientos a nuestra familia y amigos por su apoyo incondicional, paciencia y comprensión no solo a lo largo del desarrollo de este trabajo, sino durante toda nuestra carrera académica.

Desde el primer momento hasta la culminación del proyecto, el apoyo y la orientación de nuestro tutor Eduardo Fernández fueron fundamentales para el desarrollo exitoso de este trabajo. Le agradecemos profundamente por su guía, dedicación y experiencia brindada a lo largo de este proceso.

También queremos expresar nuestro agradecimiento a Camilo Satut por su receptividad y disposición para responder nuestras dudas y brindar apoyo cuando se le solicitó.

Gracias a todos.



# Resumen

El presente proyecto se centra en el diseño e implementación de un renderizador acústico con capacidad de auralización en tiempo real en escenas virtuales tridimensionales. La auralización se define como el proceso de hacer audible el campo sonoro generado por una fuente en un entorno determinado, mediante la aplicación de modelos físicos o matemáticos. Para lograr este objetivo, se utilizan técnicas avanzadas de computación gráfica y procesamiento de señales, que permiten modelar con precisión la interacción del sonido en entornos virtuales. Particularmente, se emplea el trazado de rayos, una técnica utilizada tradicionalmente para modelar la propagación de la luz y su interacción con los objetos.

A lo largo de este trabajo, se justifica esta decisión describiendo los fundamentos físicos del sonido y presentando un modelo simplificado de su propagación, sobre el que se basa para la construcción de este motor acústico. Este enfoque permite adaptar los principios del trazado de rayos al ámbito del sonido, proporcionando una solución eficiente y precisa para la simulación acústica en tiempo real.

Las GPUs (Unidades de Procesamiento Gráfico) se han mostrado particularmente adecuadas para el trazado de rayos debido a su capacidad para manejar cálculos paralelos de alta complejidad de manera eficiente. Este paralelismo es crucial tanto para el trazado de rayos, ya que cada rayo se puede procesar de forma independiente, como para el procesamiento de la señal acústica. De este modo, el uso adecuado de esta tecnología permite acelerar significativamente el proceso de auralización, logrando que se realice en tiempo real.

El resultado del proyecto es un programa que permite realizar la auralización de una entrada de audio determinada, a partir de una escena virtual tridimensional existente dentro de la que se posicionan objetos virtuales para representar el emisor del sonido y el receptor. Este motor acústico se limita a simular la reflexión y absorción del sonido, y no contempla otros fenómenos acústicos como la difracción o la transmisión. A pesar de esta limitación, el sistema proporciona una base robusta y eficiente para la auralización en tiempo real, facilitando futuras expansiones que incluyan estos fenómenos adicionales, logrando así simulaciones más completas.

Los resultados de este proyecto de grado están disponibles en el repositorio [AudioRenderingV2](#). En este repositorio, se presentan ejemplos de sonidos auralizados en diversos escenarios, demostrando los resultados del trabajo realizado.

**Palabras clave:** Simulación Sonora, Trazado de Rayos, Unidades de Procesamiento Gráfico, Auralización

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Trabajos previos	2
1.2. Estructura del documento	2
<b>2. Revisión de antecedentes</b>	<b>5</b>
2.1. Fundamentos del sonido	5
2.1.1. La onda de sonido	5
2.1.2. Nivel de sonido	7
2.1.3. Interacción del sonido con el ambiente	7
2.1.4. Tiempo de reverberación	11
2.1.5. HRTF	11
2.1.6. Representación digital del sonido	12
2.1.7. Respuesta al impulso	15
2.1.8. Modelado del sonido como rayo	17
2.2. Computación gráfica en el problema abordado	19
2.2.1. Ray Tracing	19
2.2.2. Path Tracing	20
2.2.3. Aceleración por hardware	22
2.2.4. CUDA	23
2.2.5. OptiX	24
2.3. Simulación sonora	25
2.3.1. Transformada de Fourier	25
2.3.2. Convolución de funciones	27
2.3.3. Auralización	27
2.4. Soluciones existentes para la simulación	28
2.5. Tecnologías complementarias	30
<b>3. Diseño e Implementación</b>	<b>33</b>
3.1. Objetivos	33
3.2. Requerimientos	34
3.3. Diseño	34
3.3.1. Entradas	35
3.3.2. Módulo geometría	35

3.3.3.	Módulo rasterizador . . . . .	37
3.3.4.	Módulo trazador de caminos . . . . .	38
3.3.5.	Módulo auralizador . . . . .	38
3.3.6.	Módulo audio . . . . .	39
3.4.	Implementación . . . . .	39
3.4.1.	Concurrencia . . . . .	39
3.4.2.	Contexto . . . . .	40
3.4.3.	Trazado de rayos . . . . .	40
3.4.4.	Construcción del IR . . . . .	45
3.4.5.	Simulación de HRTF . . . . .	46
3.4.6.	Convolución . . . . .	47
3.4.7.	Auralización en tiempo real . . . . .	48
<b>4.</b>	<b>Experimentación</b>	<b>53</b>
4.1.	Metodología Experimental . . . . .	53
4.1.1.	Pruebas Realizadas . . . . .	54
4.1.2.	Escenas Utilizadas . . . . .	57
4.1.3.	Hardware Utilizado . . . . .	57
4.2.	Resultados obtenidos . . . . .	59
4.2.1.	Pruebas de precisión . . . . .	59
4.2.2.	Pruebas de exactitud . . . . .	61
4.2.3.	Pruebas de rendimiento . . . . .	76
4.3.	Simulaciones en otros escenarios . . . . .	77
4.3.1.	Conference Room . . . . .	78
4.3.2.	Caja . . . . .	79
4.4.	Análisis de la auralización . . . . .	81
4.4.1.	Convolución de audio pregrabado . . . . .	81
4.4.2.	Convolución estéreo de audio pregrabado . . . . .	82
4.4.3.	Entrada por micrófono . . . . .	87
4.4.4.	Auralización en tiempo real . . . . .	87
<b>5.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>89</b>
5.1.	Conclusiones . . . . .	89
5.2.	Trabajo Futuro . . . . .	90
5.2.1.	Mejorar modelado de la propagación . . . . .	90
5.2.2.	Múltiples emisores . . . . .	91
5.2.3.	Canales de salida parametrizables . . . . .	91
<b>A.</b>	<b>Archivo de configuración</b>	<b>99</b>
<b>B.</b>	<b>Experimentos extra</b>	<b>101</b>
B.1.	Experimentos con limite de rebotes . . . . .	101

# Capítulo 1

## Introducción

La capacidad de simular y predecir comportamientos es una cualidad altamente valorada en el campo de la ingeniería. Comprender un concepto a tal nivel que se puedan realizar simulaciones desbloquea un gran potencial tanto para fines analíticos y didácticos como para la reducción de costos operativos y de desarrollo.

En el ámbito audiovisual, la motivación para realizar simulaciones sonoras es diversa. En la industria del cine y los videojuegos, por ejemplo, estas simulaciones permiten crear ambientes sonoros más envolventes y realistas, enriqueciendo significativamente la experiencia del usuario. En el campo de la arquitectura, facilitan a los diseñadores el entendimiento de cómo se propaga el sonido en espacios cerrados, aspecto crucial para el diseño de teatros, salas de conciertos y otros espacios acústicamente sensibles. Además, en la planificación urbana, esta tecnología podría predecir y mitigar la contaminación sonora en las ciudades, mejorando así la calidad de vida de los residentes.

Para simular este fenómeno, se puede recurrir a una analogía con la luz, otro fenómeno físico común cuyas técnicas de simulación están bien establecidas y popularizadas. Una de estas técnicas es el trazado de rayos, comúnmente aplicada en computación gráfica para simular la interacción de los rayos de luz con un entorno y así generar imágenes realistas. Esta técnica ha experimentado avances significativos en las últimas décadas, gracias al progreso en computación que caracteriza a la era actual. Por un lado, la luz y el sonido comparten algunas características comunes, como emanar de una fuente, propagarse en forma de ondas e interactuar con objetos, lo que indica suficientes paralelismos para poder asumir que el trazado de rayos puede ser aplicado para realizar simulaciones acústicas. Sin embargo, presentan diferencias extremadamente significativas en varios aspectos, como la longitud de onda y la velocidad, entre otros. Estas diferencias implican que la aplicación del trazado de rayos en sonido no es directamente trasladable desde su uso en la luz, sino que debe ser ajustado correspondientemente.

En el ámbito del trazado de rayos, la industria del entretenimiento, especialmente los sectores de videojuegos y efectos visuales, ha sido un motor fundamental en el avance de las técnicas de trazado de rayos. Además, han estimulado el desarrollo de herramientas tecnológicas avanzadas, como las tarjetas gráficas, también conocidas como unidades de procesamiento gráfico (GPU, por sus siglas en inglés). Estas GPUs no solo

destacan por su gran potencia, sino que también son suficientemente accesibles para un amplio espectro de usuarios. La accesibilidad de estas tecnologías ha incentivado a los competidores en el mercado a promover su uso en la programación abierta y habilitar a sus usuarios poder utilizar su poder computacional para un fin deseado, por ejemplo, la elaboración de un renderizador acústico utilizando trazado de rayos.

## 1.1. Trabajos previos

En el campo de la acústica, investigaciones previas sobre la amplificación y propagación del sonido en diversos entornos constituye una base fundamental para el presente trabajo. En particular, el estudio titulado “*Impact of multiple reflections on urban acoustics*” por (Bort y Beckers, 2021) se enfoca en comprender la influencia de la morfología urbana en la generación de ruido, identificando la multiplicidad de reflexiones del sonido entre estructuras como un factor crucial. En el marco del capítulo 4, se emplearán los resultados empíricos proporcionados por este estudio como referencia, contrastándolos con los hallazgos obtenidos mediante la implementación realizada en este proyecto.

Similarmente, el trabajo “*Simulación del sonido con trazado de rayos*” por (Satut, 2022) constituye un punto de partida esencial para el desarrollo de este proyecto. En dicho estudio, se establecen objetivos claros que orientan a la implementación de la simulación del sonido en entornos virtuales. De forma similar, sus resultados obtenidos serán referenciados más adelante en el documento.

El análisis exhaustivo de los resultados obtenidos en estos trabajos constituyó un pilar fundamental que impulsó significativamente los resultados obtenidos en este proyecto.

## 1.2. Estructura del documento

El contenido restante de este informe se dedica a proporcionar una visión detallada de todos los aspectos abordados a lo largo del proyecto. A continuación se describe la estructura y el contenido de cada capítulo:

Capítulo 2: Revisión de antecedentes. En este capítulo se aborda, entre otras cosas, el estado del arte en el campo de la acústica, la computación gráfica y la simulación sonora. Se detallan tanto los fundamentos físicos como los conocimientos matemáticos y las técnicas computacionales relevantes para el desarrollo del presente proyecto.

Capítulo 3: Diseño e Implementación. En el Capítulo 3 se plantean los desafíos a abordar, centrándose principalmente en el diseño e implementación de la solución propuesta. En cuanto al diseño, se conceptualiza la aplicación y se detallan las decisiones tomadas, así como los distintos módulos de la aplicación y su interacción. La sección de implementación abarca cómo se resuelven los problemas específicos encontrados durante el desarrollo del proyecto, junto con otros aspectos técnicos relevantes.

Capítulo 4: Experimentación. Este capítulo se dedica a presentar y analizar los resultados obtenidos a través de la simulación. Se llevan a cabo comparaciones con mediciones reales y resultados de otras simulaciones previas para evaluar el desempeño del sistema desarrollado.

Capítulo 5. Conclusiones y trabajo futuro. El último capítulo del documento se presentan las conclusiones derivadas del estudio, así como las posibles áreas de mejora del sistema o investigaciones relevantes a realizar.



# Capítulo 2

## Revisión de antecedentes

En este capítulo se examinan los fundamentos teóricos y el estado del arte relacionado con el proyecto. Se exploran los conceptos esenciales del sonido, las técnicas de computación gráfica relevantes para el proyecto, así como tecnologías complementarias y las soluciones existentes.

### 2.1. Fundamentos del sonido

En el marco del proyecto, la comprensión detallada de los fundamentos del sonido adquiere una importancia crítica, sirviendo como guía para el desarrollo de este proyecto de grado.

#### 2.1.1. La onda de sonido

El sonido se define como la percepción de vibraciones mecánicas que se propagan a través de un medio elástico, como el aire, el agua o un sólido, hasta llegar a un receptor, como un humano o un instrumento medidor (Beranek, 1954) (Möser y Barros, 2009).

Estas vibraciones se propagan en gases, líquidos y sólidos, que son medios elásticos. Cuando una partícula se desplaza de su posición original, las fuerzas elásticas tienden a restaurarla, pero la inercia de la partícula hace que sobrepase su posición de reposo, activando nuevamente las fuerzas elásticas en la dirección opuesta (Figura 2.1). Por esta razón, la elasticidad y la inercia son dos características que todos los medios deben poseer para ser capaces de conducir el sonido.

El movimiento previamente descrito en las partículas de un medio constituye la onda sonora. Dicha onda genera la compresión y rarefacción (o escasez) de dichas partículas, (Figura 2.2)

La intensidad acústica de un sonido, o volumen, es proporcional al cuadrado de la presión sonora eficaz, que es medida en Pascales (Pa). Viéndolo desde una perspectiva matemática, esta presión se representa con la amplitud de la onda, es decir, la diferencia del valor de la presión entre los “valles” y las “crestas”.

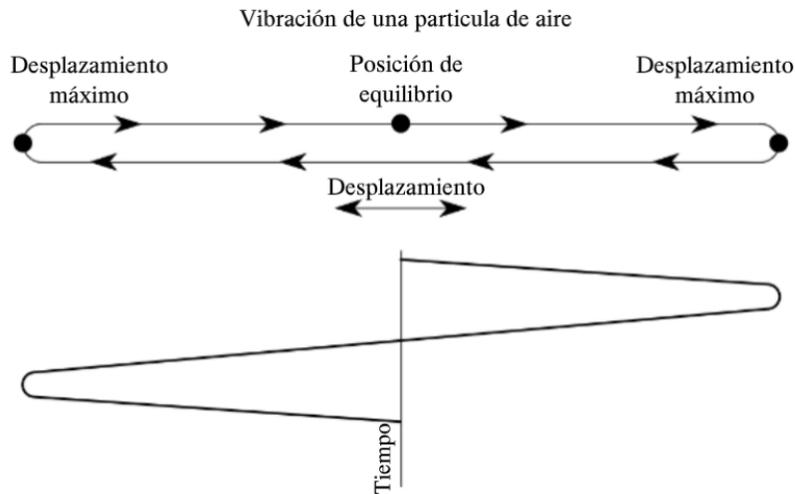


Figura 2.1: Posición en el tiempo de una partícula de aire bajo los efectos de fuerzas elásticas. Extraído de (Everest, 2001).

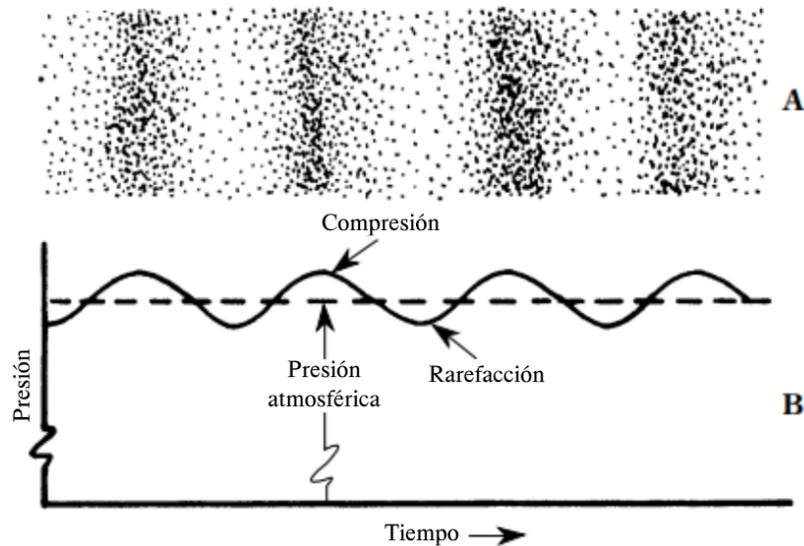


Figura 2.2: (A) Ilustración de un grupo de partículas comprimidas y enrarecidas debido a la transmisión del sonido. (B) Variaciones en la presión causadas por una onda de sonido con respecto a la presión atmosférica. Extraído de (Everest, 2001).

Además, la onda sonora exhibe otra propiedad fundamental: su frecuencia, definida como el número de ciclos completos por unidad de tiempo, que determina el tono del sonido y es medida en Hertz (Hz).

Finalmente, la longitud de onda representa la distancia en el espacio que se desplaza una onda en el tiempo que esta demora en realizar un ciclo.

Dados estos dos valores, es posible calcular la velocidad del sonido<sup>1</sup>.

### 2.1.2. Nivel de sonido

La existencia del sonido depende fuertemente de la capacidad del receptor de percibirlo. Nuestra percepción del sonido no se relaciona linealmente con los atributos físicos del sonido, más precisamente, el oído humano percibe cambios en el nivel sonoro de forma logarítmica.

El decibel (dB) es una unidad logarítmica usada para medir el nivel de presión sonora. Cuantifica el nivel al comparar la presión de dicho sonido con la presión de un sonido referencia (denominado  $P_0$ ) con una presión de 20  $\mu\text{Pa}$  (micropascales) en el aire, valor que representa la diferencia de presiones entre el momento de compresión y rarefacción de las partículas (es decir, el valle y la cresta de la onda). Este valor aproxima a la mínima presión perceptible por un humano con excelentes habilidades auditivas.

La fórmula para el nivel de presión sonora de un sonido con una determinada presión  $P_1$  es definida de la siguiente forma

$$L_p = 20 * \log_{10}(P_1/P_0) \quad (2.1)$$

Además de esto, el decibel también se puede usar para expresar el nivel de potencia sonora que en lugar de representar la variación de la presión representa la cantidad de energía sonora emitida por la fuente por unidad de tiempo. También se calcula a partir de un valor referencia  $W_0$  ( $10^{-12}$  W) mediante la siguiente fórmula

$$L_w = 10 * \log_{10}(W_1/W_0) \quad (2.2)$$

### 2.1.3. Interacción del sonido con el ambiente

Mientras las ondas de sonido se propagan, estas interactúan con el ambiente de formas complejas y variadas, dando lugar a distintos fenómenos como la reflexión, absorción, transmisión, refracción y difracción de la onda. En esta sección se procederá a explicar cada fenómeno mencionado.

Cuando una onda de sonido encuentra una superficie u objeto que se consideran “grandes” en comparación a su longitud de onda, la onda de sonido sufre una **reflexión** de forma similar a la que lo haría un rayo de luz. Un libro sería un buen reflector para un sonido de 10 kHz (donde la longitud de onda es aproximadamente 3,5 cm), sin embargo, un sonido de 20 Hz (donde la longitud de onda es aproximadamente 17 metros) atravesaría tanto el libro como a la persona que lo está sosteniendo sin ser reflejado.

Las frecuencias audibles superiores a 300–400 Hz son consideradas frecuencias especulares, (Everest, 2001) dado que sonidos en este rango actúan como rayos de luz en

---

<sup>1</sup>En la atmósfera terrestre la velocidad del sonido es de 343,2 m/s a 20 °C de temperatura

un espejo, reflejándose especularmente con un ángulo de incidencia igual al ángulo de reflexión.

Sin embargo, en la realidad otros factores pueden afectar cómo el sonido se refleja, dado que ninguna superficie es realmente plana, sino que poseen diversas irregularidades, estas tienen un gran efecto sobre la forma en la que el sonido se refleja (ver Figura 2.3). Nuevamente, la clave está en la longitud de onda del sonido, si esta es suficientemente mayor a las dimensiones de las irregularidades entonces la superficie puede ser tratada como “lisa” y se refleja acordemente; sin embargo, si esta es suficientemente menor a las dimensiones de las irregularidades, entonces se refleja según la dirección de la superficie de la irregularidad contra la que impacta. En el caso intermedio, se producirá un reflejo especular junto con un reflejo difuso, donde una gran porción de la energía original de la onda se refleja en todas las direcciones.

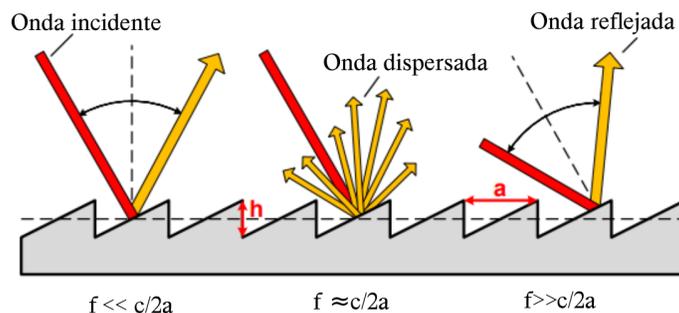


Figura 2.3: Reflexiones en una superficie con irregularidades de altura  $h$  y largo  $a$ .  $f$  denota la frecuencia de la onda incidente y  $c$  su velocidad.

La **absorción** del sonido refiere al proceso en el que una estructura u objeto es impactada por una onda de sonido determinada y, además de reflejar la onda, la absorbe y transforma parte de su energía. Típicamente, la energía absorbida es transformada en calor. Es un concepto crucial para controlar y manipular el sonido de forma predecible en determinados ambientes.

La efectividad de un material para reflejar un sonido está dada por su coeficiente de reflexión  $R$  observado, inversamente, la efectividad en la absorción sobre el sonido se representa con su coeficiente de absorción  $\alpha$ , que varía de 0 (reflexión de onda absoluta) a 1 (absorción absoluta). Este coeficiente depende de varios factores del material, como su grosor, densidad, y la estructura de su superficie. Por ejemplo, la espuma plástica con la que se elaboran los paneles acústicos que se utilizan para evitar la propagación del sonido es un material altamente poroso que permite que la onda de sonido traspase su superficie, para luego disipar la energía de la misma. Además, no solo depende del material, sino que depende de la frecuencia del sonido y el ángulo con el que la onda de sonido impacta con el material.

$R$  y  $\alpha$  se calculan de la siguiente forma (Schröder, 2011):

$$\alpha = 1 - \frac{I_r}{I_i}, \quad R^2 = 1 - \alpha \quad (2.3)$$

Donde  $I_r$  es la intensidad de la onda reflejada, e  $I_i$  es la intensidad de la onda incidente o emitida.

Además de la reflexión y la absorción, cuando el sonido impacta con un obstáculo hay un tercer fenómeno incidente, la **transmisión** del sonido a través del obstáculo (Figura 2.4). Se trata de un concepto bastante natural, se puede observar fácilmente al escuchar el sonido de otra habitación propagarse a través de la pared que la separa. Con seguridad, el sonido transmitido tiene características más silenciosas y apagadas, especialmente en comparación con el sonido original, dado que las frecuencias más altas se aíslan más efectivamente que las más bajas (Schröder, 2011).

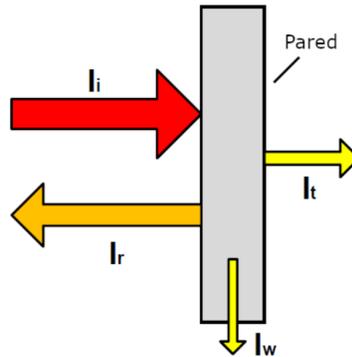


Figura 2.4:  $I_i$  es la intensidad incidente del sonido,  $I_r$  es la reflejada,  $I_w$  es la absorbida y  $I_t$  es la transmitida. Extraído de (Schröder, 2011).

Además de los mencionados hasta ahora, la **refracción** del sonido es otro fenómeno identificable en la propagación, este se relaciona con cómo los cambios en el medio de propagación modifican su trayectoria. Esto es debido a que causa que distintas partes de la onda se propaguen a diferentes velocidades. Un ejemplo de esto se puede apreciar en la Figura 2.5, donde la velocidad de un sonido transmitido sobre un medio denso es mayor que la velocidad de un medio menos denso. Esto causa que los frentes de onda marcados por los segmentos A-B y C-D ya no sean paralelos.

Nuestra atmósfera no representa un medio uniforme para la propagación del sonido, es un sistema intensamente dinámico que desafía a los meteorólogos constantemente. Debido a esto, el comportamiento del sonido en la atmósfera se ve afectado por factores como la velocidad del aire, su temperatura, su composición, su densidad, entre otros. Sin embargo, en el contexto del proyecto el principal escenario de estudio son los espacios cerrados, donde sí se puede considerar el aire como un medio uniforme, ya que gran parte de estos factores se minimizan. El principal factor a considerar son las diferencias en la temperatura del aire causadas por artefactos como el aire acondicionado, ya que la velocidad del sonido será afectada y variante a lo largo del trayecto. De todas formas, no se consideró un factor relevante sobre la propagación del sonido en este proyecto.

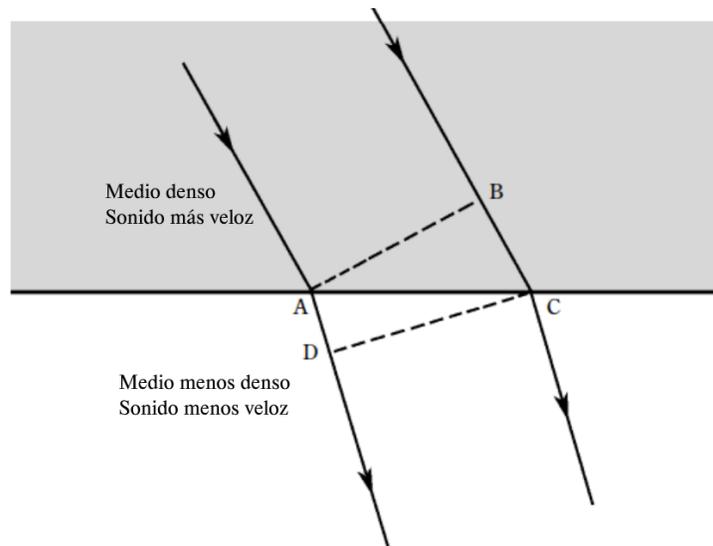


Figura 2.5: Ejemplo de la trayectoria de una onda que atraviesa medios con distintas densidades. Extraído de (Everest, 2001).

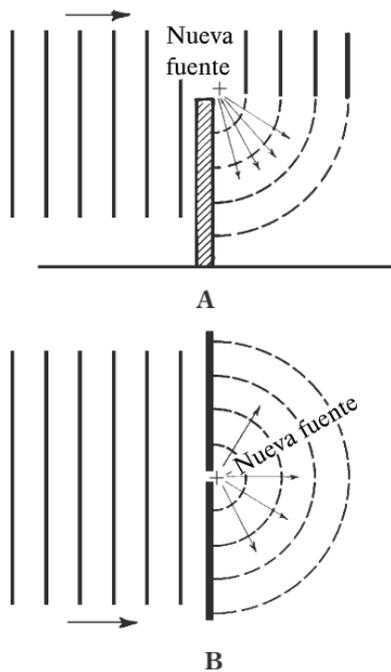


Figura 2.6: Ejemplo de la difracción de ondas sonoras. Extraído de (Everest, 2001).

Finalmente, el concepto de **difracción** describe el fenómeno donde la dirección del sonido es alterada no por cambios en el medio, sino por obstáculos presentados en su trayectoria. Un ejemplo básico de este fenómeno se puede observar fácilmente, reproduciendo música en una habitación y notando que esta es audible a lo largo de un pasillo o en otras habitaciones, “doblando” en cada esquina. En la Figura 2.6 muestra ejemplos de esto. Al impactar el sonido con una pared en el ejemplo A, se genera una nueva fuente virtual de sonido. El ejemplo B es similar, donde el sonido que atraviesa un agujero en la pared se convierte en una nueva fuente virtual, radiando el sonido omnidireccionalmente.

Existen más ejemplos y cada caso depende tanto del obstáculo como de la longitud de onda del sonido. Es importante destacar que la influencia de la difracción en espacios cerrados suele ser de un orden menor que la de las reflexiones (Tsingos et al., 2002).

### 2.1.4. Tiempo de reverberación

El tiempo de reverberación se define como la duración necesaria para que la densidad de energía acústica promediada en un recinto disminuya en 60 dB después de que la emisión de la fuente haya cesado (International Organization for Standardization, 2008).

Existen diversas razones para medir este tiempo. En primer lugar, el nivel de presión sonora proveniente de fuentes de ruido, la inteligibilidad del habla y la percepción de privacidad en una habitación dependen en gran medida de esta medida. En segundo lugar, se mide el tiempo de reverberación para determinar la absorción en la habitación inherente a muchas mediciones acústicas. En algunos países, los códigos de construcción especifican los tiempos de reverberación requeridos en aulas y otras categorías de habitaciones.

Sin embargo, medir el tiempo de reverberación puede ser una tarea difícil. En muchas habitaciones, el número de personas presentes puede influirlo fuertemente, por lo que las mediciones deben realizarse en una habitación sin personas. Algunos otros requisitos que dificultan el procedimiento incluyen el equipamiento necesario para producir y capturar sonido, y que el espacio no posea ruido de fondo, dado que este puede tapar la señal que se desea estudiar o simplemente la señal puede ser muy tenue como para que el micrófono capte esta disminución (International Organization for Standardization, 2008) (Everest, 2001).

Por este motivo, el tiempo de reverberación suele estimarse a partir de otras medidas (Claus, George y Jens, 2013). La curva de decaimiento o *decay curve* mide cómo disminuye el nivel sonoro en función del tiempo. La curva establece en su primera sección una recta que puede ser estimada mediante regresión lineal y usada para obtener estimaciones (Figura 2.7). La pendiente de esta recta se denomina *decay rate* ( $d$ ) y se expresa en  $dB/s$ , su relación con el tiempo de reverberación es lineal, de forma que un tiempo de reverberación de 0,3 segundos es equivalente a  $60\text{ dB}/0,3\text{ sec} = 200\text{ dB/s}$ .

Es entonces posible estimar el tiempo de reverberación mediante la siguiente ecuación:

$$d * t_{60} = -60 \rightarrow t_{60} = \frac{-60}{d} \quad (2.4)$$

Otra aproximación utilizada para este tiempo es el tiempo de descomposición temprana (EDT por sus siglas en inglés) (Jordan, 1970), este mide el decay rate sobre los primeros 10 dB de la caída y de esta forma contempla solo a las reflexiones tempranas. Se reconoce que el EDT está altamente relacionado con la reverberación percibida (Barron, 1995) por lo que resulta una medida adecuada para realizar la aproximación.

### 2.1.5. HRTF

La *Head Related Transfer Function* (Función de Transferencia Relacionada con la Cabeza) o HRTF por sus siglas en inglés, es una función característica del receptor del sonido. Esta establece cómo cierto sonido es percibido por el oído del usuario dado los

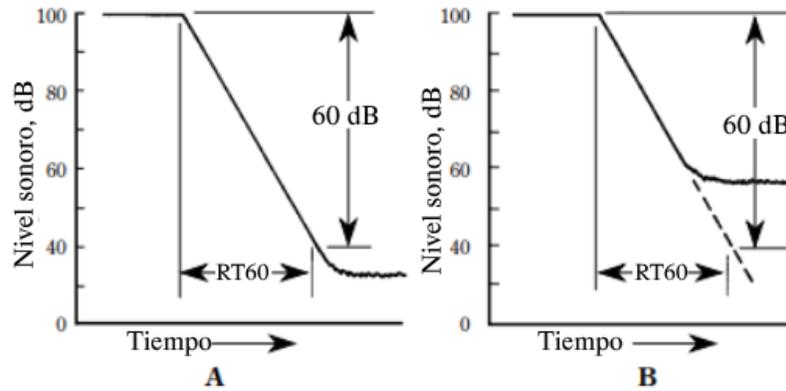


Figura 2.7: Ejemplos de curvas de decaimiento. (A) Se puede registrar una disminución de 60 dB en el nivel sonoro, el tiempo de reverberación se puede medir directamente. (B) El nivel de la señal cae por debajo del ruido de fondo antes de registrarse la disminución de 60 dB y por consiguiente el tiempo de reverberación se estima a partir de la pendiente de la curva. Extraído de (Everest, 2001).

elementos del cuerpo que lo rodean (cabeza, oreja, hombros, torso, etc), su tiempo de propagación, intensidad, frecuencia y ángulo con el que llega al receptor.

Algunos de los factores que determinan esta función son la **diferencia de tiempo interaural**, que ayuda a determinar la posición horizontal del emisor dependiendo de la diferencia de tiempo entre que un sonido llega a un oído y el otro. De forma similar, la **diferencia de niveles interaural** ayuda a determinar la ubicación del emisor dependiendo de las diferencias en la intensidad y distribución de frecuencias entre el oído derecho e izquierdo, causadas por el efecto de la cabeza sobre la señal (Google, s.f.).

Se ha demostrado que estos parámetros tienen una influencia drástica en la localización del sonido y en su evaluación subjetiva general (L. y J, 1989) por lo que su simulación para la auralización es un gran aporte. Sin embargo, aunque se puede estimar la HRTF humana para su utilización, la función es única para cada persona dada su biología, existe una necesidad importante en la personalización de la función según el oyente para realizar una simulación realista (Yang y Choudhury, 2021).

### 2.1.6. Representación digital del sonido

En el mundo moderno, los sonidos digitales son omnipresentes, ya sea en la música que escuchamos, en los efectos de sonido de las películas que vemos o en las llamadas que realizamos a través de nuestros dispositivos móviles. Estos sonidos, a menudo generados por computadoras, se basan en la representación digital de las ondas sonoras. Para comprender este proceso en detalle, en esta sección se describirá cómo se captura un sonido natural para luego convertirlo en formato digital.

## Muestreo

El primer paso en el almacenamiento digital del sonido es capturar la onda sonora. Esto se realiza típicamente utilizando un micrófono. Estos dispositivos poseen un “diafragma” o superficie móvil similar que es excitada por las variaciones en la presión generadas por la onda sonora y convierten la energía acústica en energía eléctrica, que se utiliza para construir una señal analógica continua (Ballou, 2008).

El proceso de muestreo se realiza posteriormente utilizando un conversor analógico-digital, donde la señal analógica continua previamente capturada por el micrófono, se convierte en una señal digital discreta. El conversor toma instantáneas regulares o muestras de la amplitud de la señal analógica a intervalos fijos, lo que resulta en una serie de valores que aproximan la onda a lo largo de un tiempo dado

La frecuencia con la que se toman estas muestras se conoce como tasa de muestreo (o *sample rate*), medida en hercios (Hz). Un *sample rate* común para audio de calidad CD estándar es de 44,1 kHz, lo que significa que el audio se muestrea 44.100 veces por segundo. En la actualidad también es frecuente utilizar *sample rates* de 88,2 kHz o 96 kHz para grabaciones de alta fidelidad. Es relevante notar que tasas de muestreo aún más altas pueden efectivamente reducir la precisión en la conversión de audio a una señal digital al introducir energía ultrasónica no deseada y distorsiones de intermodulación (Lavry Engineering Inc., 2012), además de requerir más tiempo de procesamiento y un mayor espacio de almacenamiento. Es por esta razón también que muchas veces se decide trabajar con *sample rates* bajos, como las llamadas telefónicas que utilizan un *sample rate* de 8 kHz, cantidad suficiente para lograr la inteligibilidad de la voz de un ser humano.

Es importante destacar el concepto presentado por el teorema de Nyquist-Shannon. Este teorema es impuesto sobre el muestreo ya que establece que una señal continua puede ser muestreada adecuadamente solo si no contiene componentes de frecuencia por encima de la mitad de la frecuencia de muestreo. Por ejemplo, una frecuencia de muestreo de 44,1 kHz *samples* por segundo requiere que la señal analógica esté compuesta por frecuencias por debajo de 22,05 kHz ciclos por segundo (Smith, 1997).

## Canales de audio

Los distintos canales de audio, en su esencia, representan diferentes vías de transmisión de información sonora. Se utilizan con el propósito de crear una experiencia auditiva más inmersiva o realista para el oyente. En un contexto técnico, los canales de audio se refieren a las diferentes pistas o componentes independientes que componen una grabación o una transmisión de sonido. Estos canales pueden corresponder a instrumentos musicales, voces, efectos de sonido o cualquier otro elemento sonoro que se desee separar o distinguir.

La cantidad de canales de un micrófono varía según el tipo y el diseño en cuestión. Algunos micrófonos son monofónicos, lo que significa que solo capturan una señal de audio. Otros son estereofónicos o incluso multicanal, lo que les permite capturar múltiples señales de audio simultáneamente. Esta capacidad de captura multidireccional es

especialmente útil en entornos como estudios de grabación, conciertos en vivo o grabaciones de campo donde se desea una reproducción más precisa y detallada del sonido ambiente.

Por otro lado, los reproductores de sonido, ya sean dispositivos físicos o software, también pueden tener varios canales de salida. Estos canales determinan la forma en que se distribuye el sonido a través de altavoces o auriculares. Una señal mono se reproduce a través de un único canal de audio, significando que se envía el mismo buffer de sonido a todas las salidas. Sin embargo, el estándar para la salida de audio es una señal estéreo que utiliza dos canales de audio (izquierda y derecha), lo que permite reproducir experiencias auditivas más realistas. Además del mono y el estéreo, existen formatos de audio de mayor complejidad, como el sonido envolvente, que utiliza múltiples canales para simular una experiencia sonora tridimensional. Sin embargo, el uso de señales mono sigue siendo común en situaciones donde el espacio de almacenamiento o el ancho de banda son limitados.

## Cuantización

Tras el proceso de muestreo, se inicia el proceso de cuantización, en el que cada valor muestreado de la señal de audio se debe representar mediante un valor numérico en el dominio digital.

Este proceso conlleva inherentemente un grado de aproximación, dado que el espectro continuo de la señal analógica necesita ser representado por un conjunto finito de valores digitales. La exactitud de esta representación depende de la profundidad de bits (*bit depth*). Esta profundidad determina la cantidad de bits asignados a cada valor muestreado. Por ejemplo, una profundidad de 16 bits, estándar en el audio de CD, permite 65,536 valores diferentes para cada muestra.

Un mayor bit depth ofrece dos ventajas significativas. En primer lugar, posibilita una mayor cantidad de valores digitales posibles para cada muestra, lo que se traduce en una precisión incrementada. Además, estos valores no solo abarcan una precisión más elevada, sino también un rango dinámico más amplio, permitiendo representar digitalmente sonidos tanto sutiles como intensos.

Un mayor bit depth también implica ciertas desventajas. Por un lado, incrementa el requerimiento de espacio de almacenamiento; por ejemplo, un sonido almacenado con una profundidad de 32 bits ocupará el doble de espacio que el mismo sonido cuantizado con una profundidad de 16 bits. Además, al permitir la digitalización de sonidos con una mayor precisión, se pueden introducir variaciones mínimas en la señal, que podrían ser captadas del ambiente, generando así ruido, distorsión, y un aumento en la complejidad de procesamiento y manejo del archivo de audio.

## Codificación y compresión

Las muestras cuantificadas finalmente se codifican en un formato digital, lo que resulta en una representación binaria del sonido. Estos datos binarios pueden comprimi-

mirse aún más para reducir el tamaño del archivo, utilizando métodos de compresión con pérdida o sin pérdida.

La capacidad de comprimir estos datos de audio, sin importar el método utilizado, brinda flexibilidad en el almacenamiento y transmisión de archivos de audio, permitiendo una reproducción fiel del original, y facilitando su manipulación, distribución y consumo en diversos contextos.

### 2.1.7. Respuesta al impulso

En el contexto de la acústica se denomina impulso a una única ráfaga corta o serie de ráfagas cortas de presión sonora eficaz (International Organization for Standardization, 1997). Se trata de un sonido caracterizado por un pico de presión (o picos) repentinos y de corta duración.

Cuando una fuente de sonido emite sonido en una habitación cerrada, las ondas sonoras que se propagan interactúan en los elementos constructivos de la habitación hasta que finalmente llegan al receptor. Por lo tanto, un evento auditivo percibido por un oyente no solo consiste en la onda inicial generada por la fuente, sino también en la respuesta de la habitación, que comprende reflexiones retrasadas y atenuadas.

Si se emite una señal única e instantánea dentro de una escena (como por ejemplo un aplauso) y se utiliza un micrófono como receptor, se puede registrar la señal generada junto con la interacción de la señal con la escena misma. Se define a esta medición como la respuesta al impulso de un sistema (o IR por sus siglas en inglés). Esta medición puede verse como la “huella acústica” de la configuración usada para registrar el IR, dicha configuración contempla las posiciones del emisor y receptor, y la naturaleza física de la escena, como su geometría y materiales.

Objetivamente, el IR refleja la energía que llega al receptor en cada instante de tiempo, esta energía puede expresarse en intensidad o con el cuadrado de la presión sonora eficaz.

Se pueden identificar tres secciones de un IR causadas por distintas fuentes (ver Figura 2.8). Las secciones difieren en sus propiedades y también en la percepción y procesamiento por el oído humano (Schröder, 2011)

1. Sonido Directo - Este primer impulso que llega, el más fuerte, es evaluado por el oído humano para localizar la fuente de sonido. El sonido directo se retrasa por la distancia entre la fuente y el receptor y solo se atenúa por el aire.
2. Reflexiones Tempranas - Las primeras reflexiones de los elementos constructivos circundantes u otros obstáculos en la habitación llegan al oyente. Estas reflexiones (de bajo orden) se agregan al sonido directo inicial por el oído humano, lo que significa que no se pueden percibir por separado. La información sobre la fuente de sonido, como la posición, distancia, anchura de la fuente y volumen, están relacionadas principalmente con estas primeras reflexiones.

3. Reverberación Tardía - Típicamente con un retraso de 50-80 ms al sonido directo, el número de reflexiones aumenta gradualmente y el oído humano ya no puede percibir las como eventos individuales. Este campo de sonido difuso forma una reverberación tardía que es casi independiente de la posición del oyente, ya que el oído humano comienza a realizar una integración energética bastante aproximada durante un cierto intervalo de tiempo y campo angular. La reverberación es un atributo acústico muy importante y probablemente el más notable de una habitación, ya que ciertas características, como el volumen y la forma de la habitación, están directamente asociadas con ella, lo que le da a la habitación su sonido muy individual.

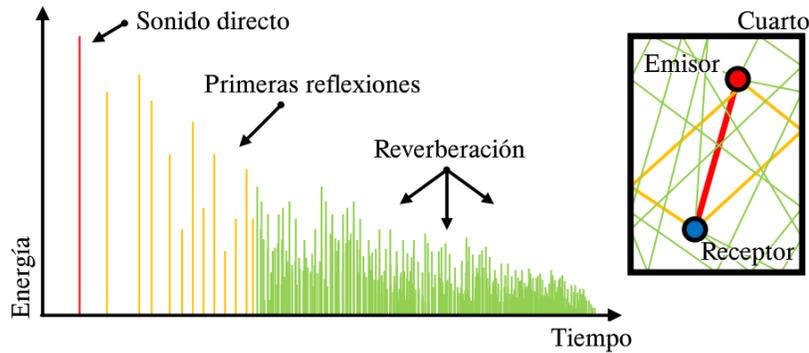


Figura 2.8: Ejemplo de un IR y sus secciones. Extraído de (Ballou, 2008).

El IR es de interés para este trabajo, ya que dado el mismo, es posible utilizarlo para simular cómo un sonido se escucharía si este fuera emitido por un emisor y recibido por un receptor posicionados dentro de la misma escena en la que se midió el IR, en sus posiciones originales. Este proceso se simula a través de una operación llamada **convolución** y se describe en mayor detalle posteriormente (ver Sección 2.3.2).

## Métodos de medición del IR

Para grabar una respuesta al impulso fiel de un espacio, la fuente de sonido utilizada debe excitar el ambiente por igual en todas las frecuencias y en todas las direcciones.

Las primeras grabaciones de la respuesta al impulso se realizaban utilizando una fuente cuasi-impulsiva, como el estallido de un globo (Abel et al., 2010), un instrumento de percusión como un clapper (ver Figura 2.9) o un disparo (Stevens y Murphy, 2014).

El uso de estas fuentes puede ser indeseable, ya que un disparo de un arma es inapropiado o prohibido en algunos contextos, y un instrumento de palmada es difícil de activar de forma remota y sin involucrar objetos reflectantes.

Una onda sinusoidal “barrida” (de la traducción al inglés “sine sweep”) es una función sinusoidal que gradualmente cambia de frecuencia en el tiempo. Puede ser utilizada como una alternativa a los métodos anteriores para emular la respuesta al

impulso en todos los valores de frecuencia, proporcionando un barrido de amplitud constante a través de un rango de frecuencia adecuado que cubre típicamente el rango del oído humano (aproximadamente 20 Hz - 20 kHz). A diferencia de los métodos anteriores, el resultante es un audio extenso que se deconvoluciona para obtener la respuesta al impulso y “remover” la onda sinusoidal de la grabación (Stevens y Murphy, 2014).

La onda sinusoidal barrida es un método altamente popular dada la exactitud que se obtiene. Sin embargo, requiere un conjunto de equipos de grabación y reproducción, incluyendo un altavoz potente capaz de emitir omnidireccionalmente en un amplio ancho de banda, lo que resulta su principal desventaja en comparación a otros métodos, ya que los requisitos materiales y logísticos de esta técnica son significativos.

Dadas muchas de las alternativas presentadas para capturar el IR, resulta claro que ninguna brinda una ventaja absoluta sobre las demás. Sin embargo, la simulación del IR sí presenta ventajas (como se presentó en el Capítulo 1) principalmente, que la escena a estudiar no tiene por qué existir en la realidad. Por lo tanto, este trabajo opta por simular el IR en lugar de capturarlo.



Figura 2.9: Un *clapper*, instrumento de percusión que consiste de dos tablas de maderas que se golpean para producir sonido.

### 2.1.8. Modelado del sonido como rayo

Para realizar una simulación del IR es importante primero definir el método por el que se simula la fuente de sonido que lo caracteriza. Existen dos enfoques principales en el modelado acústico digital. El más preciso se basa en resolver numéricamente la ecuación de onda de sonido. Las técnicas que utilizan este enfoque se denominan *basadas en ondas*. Estos típicamente mantienen un conjunto constante de demandas computacionales para toda etapa de la simulación, ya que normalmente utilizan un número constante de elementos. Desafortunadamente, los costos computacionales son altos, ya que en muchos casos se debe resolver un sistema lineal de ecuaciones de gran magnitud. El tamaño del sistema depende de la frecuencia, ya que debe haber un cierto número de elementos por longitud de onda, lo que implica que estos métodos son más

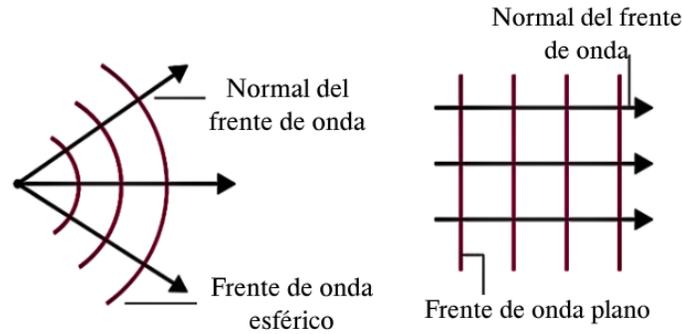


Figura 2.10: Dos tipos de frentes de onda y sus normales.

adecuados para bajas frecuencias (Siltanen, Lokki y Savioja, 2012).

El otro enfoque se basa en la acústica geométrica, donde se supone que el sonido actúa como rayos, dado que la dirección de la energía transmitida sigue la normal del frente de onda (ver Figura 2.10).

Sin embargo, utilizar esta simplificación significa una reducción en la precisión del modelado, ya que varios de los fenómenos basados en ondas, como la difracción y la refracción, están ausentes en estos métodos (en contraste a los métodos basados en ondas, ya que estos modelan los fenómenos de manera inherente). Es relevante mencionar que la difracción y la refracción se ven más presentes a bajas frecuencias, por lo que el modelado de la onda de sonido como rayos resulta adecuado para simular ondas de altas frecuencias (Siltanen, Lokki y Savioja, 2012).

Otro factor que afecta la precisión de la simulación en la acústica geométrica es su capacidad limitada para encontrar todas las trayectorias de reflexión especular. Los métodos basados en ondas garantizan encontrar todas ellas, mientras que el trazado de rayos solo muestrea el espacio de trayectorias y modela cierto subconjunto de todos los posibles caminos del sonido. Sin embargo, con una cantidad infinita de rayos de sonido, es posible generar resultados similares entre los métodos, pero cuanto menos rayos se usen, es más probable que se pierdan trayectorias de reflexión.

Claramente existen ventajas y desventajas inherentes a cada enfoque, y cabe destacar que a lo largo del tiempo han surgido métodos híbridos que buscan combinar los aspectos más beneficiosos de ambos. Sin embargo, para los propósitos de este trabajo, se ha decidido optar por representar el sonido mediante el enfoque de trazado de rayos. Esta decisión se fundamenta principalmente en el conocimiento de los estudiantes y del tutor en el trazado de rayos y la potencialidad para modelar exitosamente un enorme conjunto de situaciones.

Siguiendo este modelo, se puede considerar un impulso de sonido como la emisión omnidireccional de infinitos rayos, cada uno representando una ínfima cantidad de la presión de la onda de sonido original. En el contexto de la simulación, esta cantidad de

rayos es finita, y el impacto de uno de estos rayos con un hipotético receptor tridimensional representa la percepción del sonido. Para poder calcular la cantidad de energía acústica percibida se puede seguir la siguiente ecuación (Yang y Shield, 2000):

$$I_{rayo} = \frac{sE_0e^{-hd} \prod_i(1 - \alpha_i)}{V_{receptor}} \quad (2.5)$$

- $s$  es la distancia que el rayo recorre dentro del receptor expresada en metros.
- $E_0$  es la energía acústica inicial del rayo expresada en watts.
- $d$  es la distancia recorrida hasta el receptor en metros.
- $h$  el coeficiente de absorción del aire.
- $i$  es la cantidad de reflexiones del rayo hasta llegar al receptor.
- $\alpha_i$  es el coeficiente de absorción del objeto que produce la  $i$ -ésima reflexión. La fórmula  $(1 - \alpha_i)$  representa el factor reflejado y se le llama coeficiente de reflexión.
- $V_{receptor}$  es el volumen del receptor, dado que, por ejemplo, el receptor puede ser modelado como una esfera.

La intensidad dada por esta ecuación tiene como unidad  $\frac{W}{m^2}$ .

## 2.2. Computación gráfica en el problema abordado

Dado que se ha optado por modelar el sonido mediante el enfoque de trazado de rayos, resulta altamente conveniente aprovechar las tecnologías existentes de computación gráfica que simulan la propagación de la luz en forma de rayos. En esta sección, exploraremos las herramientas y tecnologías de la computación gráfica que serán relevantes para el modelado acústico realizado en este trabajo.

### 2.2.1. Ray Tracing

El trazado de rayos, o *ray tracing* como es conocido en inglés, es una técnica inspirada en el comportamiento físico de la luz, utilizada principalmente en la computación gráfica para la generación de imágenes digitales de una escena tridimensional. En el contexto de un *ray tracer* primitivo, este método simula la luz como una serie de rayos originarios desde la posición de un observador, que son emitidos en una dirección “a través” de un plano de imagen. El plano representa la pantalla, formada por una matriz de píxeles donde a cada píxel le corresponde usualmente un rayo emitido. Estos rayos se propagan hasta detectar una colisión con un objeto de la escena, y una vez que esto sucede, se realizan cálculos para determinar el color del píxel en función de la posición de las fuentes de iluminación, texturas y materiales de los objetos involucrados (Arvo

et al., 2019). Después de calcular el color del píxel, el rayo puede rebotar, dependiendo de las propiedades de los materiales (por ejemplo, si son reflectantes o refractivos), y se repite el proceso para el nuevo rayo generado. Este proceso de emisión de rayos continúa hasta que se haya calculado el color de cada píxel en la imagen final.

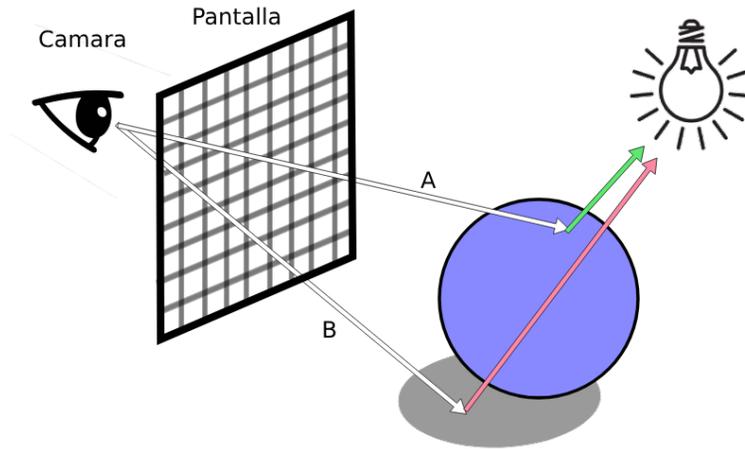


Figura 2.11: Elementos básicos de un *ray tracer*: un punto de emisión (ojo del observador), un plano que representa la imagen y direcciona los rayos a través de sus “píxeles”. Para el rayo A no se detectan colisiones hacia la luz, para el rayo B si, mediante esta lógica, se pueden detectar sombras.

Es una técnica que surge como una evolución de métodos previos, como la rasterización (Caulfield, 2018), adquiriendo una mayor complejidad y peor rendimiento para convertirse en un método más versátil capaz de lograr resultados más realistas (Chang, Chen y Chuang, 2015). Es destacable que cualquier fenómeno físico de ondas o partículas con movimiento aproximadamente lineal, como por ejemplo la propagación del sonido puede ser simulada con el trazado de rayos (Wu y Zheng, 2014).

### 2.2.2. Path Tracing

Una limitación de los *ray tracers* a nuestra aplicación es que el algoritmo no es capaz de brindar información sobre el camino del rayo hasta el receptor, en este caso la cámara. Esto es necesario dado que en el contexto de la propagación del sonido, la dimensión del tiempo es sumamente importante, así como también la existencia de los posibles caminos que el sonido puede recorrer, ya que estos causan que el receptor experimente diferentes fenómenos auditivos, como el eco o la reverberación.

En este sentido, los *path tracers*, presentan una opción viable para la simulación. El *path tracing* es una técnica moderna derivada del *ray tracing* para crear imágenes de

escenas tridimensionales que se fundamenta en los métodos de Monte Carlo (Kajiya, 1986).

El mecanismo básico de *path tracing* implica seguir la trayectoria de los rayos emitidos desde la cámara hasta las fuentes de luz en la escena. Para lograr imágenes más realistas que mejor simulen diversos efectos de iluminación, cada rayo interactúa con los objetos en su camino, reflejándose, refractándose o siendo absorbido. Cuando un rayo llega a una fuente de luz, se considera como una “trayectoria completa” y se calcula la contribución de esa trayectoria a la imagen final. Pueden generarse más rayos con el fin de encontrar más caminos que acumulen contribuciones al color del píxel final. Aunque el algoritmo de *path tracing* puede ser relativamente ineficiente para la generación de imágenes, dado que requiere un gran número de rayos para obtener imágenes de alta calidad sin artefactos de ruido (Figura 2.12), esta limitación no es tan relevante en el contexto de la simulación acústica, donde el enfoque está en los impactos sonoros en sí mismos y no en generar suficientes impactos para aproximar estadísticamente la imagen completa y en buena calidad.

Considerando la adecuación de los *path tracers* en este contexto, se puede construir una variante de un *path tracer* para la propagación del sonido, trazando caminos entre un emisor y el micrófono, construyendo un IR con cada camino incidente en el micrófono receptor. Esto se explora más detalladamente en el Capítulo 3.

---

**Algoritmo 1** Algoritmo básico de *path tracing*. (Pharr, Jakob y Humphreys, 2023)

---

```
1: for píxel do
2:   Definir  $N \geq 1$  muestras en ese píxel.
3:   for muestra do
4:     Generar un rayo (GenRay, Algoritmo 3) y emitirlo.
5:     while no se ha alcanzado una superficie emisora
6:       and no se alcanzo el número máximo de rebotes para el camino
7:       and no se alcanzo la máxima longitud permitida para el camino do
8:         if interseca con la escena then
9:           Calcular la contribución de luz directa e indirecta del camino en el
10:          punto de intersección
11:          Combinar las contribuciones de luz con texturas y efectos adicionales
12:          Calcular una nueva dirección para el rayo
13:          Almacenar los resultados de los cálculos para la siguiente iteración
14:         end if
15:       end while
16:     end for
17:   Promediar los resultados de las N muestras para producir el color final del píxel.
18: end for
```

---



(a)



(b)

Figura 2.12: (a) Imagen de una escena generada por un *path tracer* con 1024 rayos por píxel. (b) Imagen de una escena generada por un *path tracer* con 8 rayos por píxel.

Extraídas de (Pharr, Jakob y Humphreys, 2023).

### 2.2.3. Aceleración por hardware

Las unidades de procesamiento gráfico (conocidas por sus siglas en inglés *GPU*) han impactado en la computación científica como una tecnología innovadora que ha permitido grandes mejoras en rendimiento y eficiencia energética para numerosas aplicaciones (Farber, 2011). Estas herramientas están dotadas de una poderosa capacidad computacional, modelos de programación accesibles y arquitecturas diseñadas específicamente para la computación matemática paralela, ya que el procesamiento de gráficos presenta ese requerimiento. En comparación, las CPUs tradicionales se centran en la ejecución secuencial de instrucciones de mayor complejidad, por lo que no resultan tan adecuadas para la computación gráfica.

Su utilización para algoritmos de *ray tracing* es reciente. En la década del 2000 se empezaron a ver trabajos que exitosamente pudieron mapear algoritmos de trazado de rayos al hardware gráfico (Purcell et al., 2005) (Christen, 2005). Aunque la idea de utilizar las GPUs con este fin era previa, desarrollos en la arquitectura de las mismas y en sus técnicas de programación dieron lugar a tales avances. En la actualidad el trazado de rayos en tiempo real sigue sin ser una realidad, pero se busca aproximarla o emularla. De parte de NVIDIA, desde 2018 (Altavilla, 2018) con la arquitectura Turing y la inclusión de núcleos dedicados para *ray tracing* denominados *RT cores*, la plataforma RTX combina tecnología de *ray tracing* con redes neuronales (NVIDIA, s.f.[e]) para el renderizado de imágenes en tiempo real. Por otra parte, AMD lanzó su arquitectura en 2020 con soporte dedicado para *ray tracing* con núcleos similares denominados *ray accelerators*. Su inclusión mejoró la eficiencia para el trazado de rayos, pero no es capaz de realizarlo en tiempo real.

## 2.2.4. CUDA

Los primeros avances en las aplicaciones de GPU fueron liderados por expertos que invirtieron un esfuerzo considerable en aprovechar el potencial de las GPUs sin demasiadas herramientas auxiliares. Sin embargo, para extender la capacidad de estos dispositivos a una base de usuarios más amplia surgió la necesidad crítica de reducir la barrera de entrada para el desarrollo de software con utilización de GPU.

La respuesta de NVIDIA a esta demanda es CUDA (NVIDIA, s.f.[a]), una plataforma de computación paralela y una interfaz de programación que fue lanzada por primera vez en 2007. Una de las funcionalidades principales de CUDA es su capacidad para aprovechar el paralelismo masivo inherente en la arquitectura de las GPUs para ejecutar miles de hilos de forma concurrente. Además, la plataforma proporciona un conjunto de herramientas y bibliotecas optimizadas para todo tipo de funciones matemáticas complejas. Cabe destacar que el ecosistema CUDA abarca varios lenguajes de programación, pero el más adecuado y el que brinda una selección más diversa de librerías es C++. También ofrece soporte para la interoperabilidad entre la CPU y la GPU, lo que permite la transferencia eficiente de datos entre los dos dispositivos y la ejecución combinada de código en ambos. En este trabajo se aprovechan todas estas ventajas.

Dado que muchas aplicaciones paralelas involucran datos organizados de forma multidimensional, la plataforma organiza su jerarquía de hilos paralelos de forma tridimensional. En la práctica, esto implica que los identificadores de los hilos son vectores de tres componentes, lo que permite identificar los hilos utilizando un índice unidimensional, bidimensional o tridimensional. A su vez, estos hilos forman un bloque de hilos unidimensional, bidimensional o tridimensional, y estos a su vez se organizan en una grilla de bloques unidimensional, bidimensional o tridimensional. Esta jerarquía proporciona una manera natural de invocar una función que opere a través de los elementos en una estructura de memoria de una, dos o tres dimensiones.

En el contexto de CUDA, la memoria se categoriza en dos tipos, memoria *host* y memoria *device*, o de dispositivo. La memoria del dispositivo es la memoria ubicada en la GPU, siendo accesible solo por este hardware, ofrece una alta tasa de transferencia y es especialmente adecuada para operaciones de cálculo paralelo. Es aquí donde se almacenan los datos que serán procesados por los programas que ejecuten en GPU. Por otro lado, la memoria del *host* es la memoria del sistema central de la computadora, accesible por la CPU y caracterizada por ser más lenta que la memoria del dispositivo debido a la naturaleza de acceso a través del bus de memoria. En el modelo de programación CUDA, la comunicación entre la memoria del *host* y *device* es una consideración importante, ya que es necesaria y también costosa en términos de rendimiento, por lo que es crucial minimizar las transferencias para obtener un mejor rendimiento.

## 2.2.5. OptiX

OptiX es un motor de trazado de rayos acelerado por hardware desarrollado por NVIDIA bajo el ecosistema de CUDA, diseñado específicamente para facilitar la creación de aplicaciones de renderizado de imágenes (NVIDIA, s.f.[d]). Una de las funcionalidades destacadas de esta librería es su gran eficiencia, ofreciendo desde primeras versiones la capacidad de realizar *ray tracing* en tiempo real (Ludvigsen y Elster, 2010). Otra de sus grandes características es que ofrece un entorno de programación flexible que permite implementar algoritmos de renderizado personalizados que por ejemplo, no generan resultados visuales, sino que utilizan la información del trazado de rayos de alguna otra forma.

El proceso de trazado de rayos proporcionado por OptiX se implementa mediante diferentes tipos de programas, cada uno con propósitos distintos (NVIDIA, s.f.[c]).

- **Generación de rayos:** Es el punto de entrada al proceso de trazado de rayos, invocado por el sistema para cada asignación de trabajo definida por el usuario.
- **Intersección:** Implementa una prueba de intersección primitiva, invocada durante el trayecto del rayo.
- ***Closest-hit:*** Se activa cuando un rayo encuentra el punto de intersección más cercano. Se activa solo una vez por cada lanzamiento de rayo.
- ***Any-hit:*** Se activa cuando un rayo encuentra un nuevo punto de intersección, no necesariamente el más cercano. Por lo tanto, puede ser activado múltiples veces durante un solo lanzamiento de rayo.
- ***Miss:*** Se activa cuando un rayo no alcanza ninguna geometría de escena.
- ***Exception:*** Invocado para condiciones de error.
- ***Callable* directo:** Una llamada de función CUDA regular, las llamadas directas se ejecutan inmediatamente.
- ***Callable* continuo:** Llamadas ejecutadas por el planificador dados eventos particulares.

El flujo de trazado de rayos se basa en la estructura de llamadas interconectadas de los ocho programas y su relación con la búsqueda a través de los datos geométricos en la escena. La Figura 2.13 ilustra su relación.

Es importante resaltar que la implementación de todos los programas no es necesaria y varía según el caso de uso del usuario, se explora esta ventaja en más detalle en la Sección 3.4.3.

Otro factor de utilidad de la librería Optix es la capacidad de asignar una carga útil (*payload*) a cada rayo. Este *payload* es único para cada rayo y es un recurso versátil utilizado para pasar datos entre los distintos tipos de programas durante la ejecución.

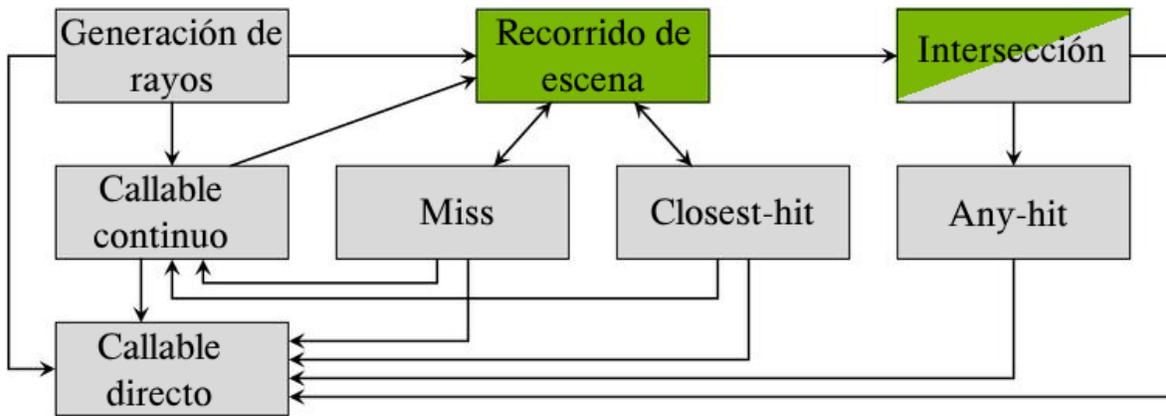


Figura 2.13: Relaciones entre los tipos de programa de OptiX. Las funciones en verde representan funciones fijadas por la librería, las grises son implementadas por el usuario.

Los *payloads* están limitados en tamaño y pueden ser definidos en su totalidad por el usuario.

Otra estructura notable en el contexto de la librería es el *shader binding table* o SBT. El mismo almacena las direcciones de memoria de los posibles *shaders* para la renderización y otros datos relacionados, necesarios u opcionales, para procesar intersecciones de rayos durante la ejecución. Es mediante esta tabla que se puede asociar a los *meshes* de la geometría propiedades personalizadas, como por ejemplo la absorción acústica del material.

## 2.3. Simulación sonora

Hasta ahora, no se ha explorado cómo la respuesta al impulso de un entorno se utiliza para simular la percepción auditiva dentro del mismo. En la siguiente sección se explorará en detalle el proceso y sus elementos.

### 2.3.1. Transformada de Fourier

La transformada de Fourier es una transformada integral que convierte una función continua en una representación que describe las frecuencias presentes en la función original. Es una herramienta matemática y física para analizar señales y sistemas (Gomes y Velho, 1999).

La transformada descompone una función compleja  $f(x)$  en sus frecuencias constituyentes y sus amplitudes. Se define de la siguiente manera:

$$\mathcal{F}\{f(x)\} = F(k) = \int_{-\infty}^{\infty} f(x) e^{-ikx} dx \quad (2.6)$$

Donde:

- $F(k)$  representa la transformada de Fourier de  $f(x)$
- $k$  es la variable de frecuencia
- $i$  es la unidad compleja

La función exponencial compleja  $e^{-ikx}$  representa la oscilación a la frecuencia  $k$  y es esencial para capturar el contenido de frecuencia de la función (Heckbert, 1998).

La inversa de la transformada de Fourier nos permite reconstruir la función original  $f(x)$  a partir de su representación de frecuencia  $F(k)$ . Se define como:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(k) e^{ikx} dk \quad (2.7)$$

### Transformada Discreta de Fourier

Como se explicó en la Sección 2.1.6, el sonido digital se representa mediante *samples*, por lo que es una señal discreta. Cuando una señal es discreta y periódica, no se necesita de la transformada de Fourier para funciones continuas. En su lugar se utiliza la transformada discreta de Fourier, o DFT por sus siglas en inglés (Weaver, 1989).

La DFT toma una secuencia de  $N$  puntos de entrada y produce una secuencia de  $N$  puntos de salida, cada uno representando una componente de frecuencia de la señal original (Heckbert, 1998). Se define matemáticamente como:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{(-i2\pi kn)/N} \quad (2.8)$$

Donde:

- $x(n)$  es la secuencia de entrada de  $N$  puntos.
- $X(k)$  es la secuencia de salida de  $N$  puntos que representa las amplitudes y fases de las componentes de frecuencia.

Esta transformada también posee su versión inversa.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{(i2\pi kn)/N} \quad (2.9)$$

### Transformada Rápida de Fourier

Una Transformada Rápida de Fourier (FFT, por sus siglas en inglés) es un algoritmo que calcula la DFT de una secuencia de datos. Es un algoritmo deseable dado que calcular la DFT de una secuencia de  $n$  puntos mediante la definición directa de la ecuación 2.8 tiene una complejidad de  $O(n^2)$ . El algoritmo FFT reduce significativamente la complejidad computacional a  $O(n \log n)$ . Esto es posible dado que FFT factoriza

la matriz del algoritmo DFT como el producto de matrices dispersas (principalmente ceros) (Heckbert, 1998).

Además de la FFT, también existe la Transformada Rápida de Fourier Inversa (IFFT), que calcula la inversa de la DFT. Al igual que la FFT, la IFFT es computacionalmente eficiente, con una complejidad de  $O(n \log n)$ .

### 2.3.2. Convolución de funciones

La convolución es una operación matemática que combina dos funciones para producir una tercera función que describe cómo una función modifica a la otra a medida que se superponen y se desplazan a lo largo del eje independiente.

La definición formal de la convolución de dos funciones  $f(t)$  y  $g(t)$  es:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) \cdot g(t - \tau) d\tau$$

Donde  $*$  denota la operación de convolución y  $(f * g)(t)$  representa la función resultante de la convolución de  $f(t)$  y  $g(t)$  en el punto  $t$ .

#### Teorema de la convolución 1 (Heckbert, 1998)

*La transformada de Fourier de una convolución de dos señales es el producto de sus transformadas:  $F(f * g) \leftrightarrow F \cdot G$ .*

*Además, la transformada de Fourier del producto de dos señales es la convolución de sus transformadas:  $F(f \cdot g) \leftrightarrow F * G/2\pi$*

### 2.3.3. Auralización

La auralización es un proceso que busca representar cómo se percibe la acústica de un espacio tridimensional que pueda o no existir. Implica hacer audible, a través de modelos físicos o matemáticos, el campo sonoro generado por una fuente dentro de un entorno dado (Everest, 2001).

Para llevar a cabo el proceso de auralización, se necesitan las características reverberantes del espacio en forma de IRs. Estas señales se convolucionan con una señal de audio para infundir este audio con la “huella acústica” del entorno (ver Sección 2.1.7). Para realizar la convolución se utiliza el Teorema 1, realizar la inversa de la transformada de Fourier de la convolución retornará el resultado deseado (ver Sección 3.4.6). Este proceso permite “escuchar” la habitación desde varios puntos de vista y evaluar sus características sonoras.

Los beneficios de la auralización se extienden más allá del diseño arquitectónico; encuentra aplicaciones en campos diversos, como la acústica de salas de conciertos, entornos de realidad virtual y estudios de producción de audio. Al aprovechar técnicas computacionales avanzadas, la auralización capacita a arquitectos, ingenieros y artistas

para crear experiencias auditivas realistas y optimizar el rendimiento acústico de los entornos construidos.

Dado este procedimiento, es posible construir un sistema de auralización en tiempo real que permita al usuario moverse dentro de la habitación y escuchar los cambios resultantes en el sonido mientras realmente ocurren. Dado que un IR está determinado por una escena y las posiciones del emisor y receptor dentro de esta, el desplazamiento de alguno de estos elementos implica un recálculo de un nuevo IR para realizar una auralización actualizada. Este enfoque requiere la computación casi instantánea de la respuesta al impulso para que todos los parámetros relacionados con el sonido directo y las reflexiones puedan ser calculados, desafío que puede ser resuelto con la utilización de técnicas de *ray tracing* eficientes (Savioja, 2000).

## 2.4. Soluciones existentes para la simulación

A continuación se presentan algunas soluciones para la simulación del sonido espacial. Algunas de estas comparten el mismo enfoque en *ray tracing* que este trabajo.

### NVIDIA VRWorks Audio

NVIDIA VRWorks Audio (NVIDIA, s.f.[f]) es un kit de desarrollo de software (SDK por sus siglas en inglés) para el desarrollo de experiencias de audio inmersivas en entornos de realidad virtual de la plataforma NVIDIA. Este SDK proporciona funcionalidades avanzadas para la simulación y auralización del sonido, permitiendo a los desarrolladores crear ambientes sonoros envolventes y realistas para aplicaciones de realidad virtual.

La librería está diseñada para aprovechar al máximo la aceleración por hardware disponible por las GPUs de NVIDIA, ofreciendo operaciones optimizadas para las arquitecturas de estas. Utiliza un motor de *ray tracing* llamado NVAR (NVIDIA Acoustic Raytracer) desarrollado sobre CUDA y OptiX.

Las características principales del NVIDIA VRWorks Audio incluyen la modelización de la reflexión difusa y especular, así como la transmisión y una aproximación de la difracción para el sonido directo. Los desarrolladores pueden construir escenas sonoras detalladas con diferentes coeficientes de absorción y transmisión, así como múltiples fuentes de sonido. Estas escenas proveen la base para el llamado “*game loop*” de la librería, donde para cada par de emisor y receptor, la biblioteca construye un par de filtros de alta resolución (de hasta 48000 Hz) y largos (de hasta 2 segundos). Los dos filtros modelan cómo el sonido del emisor particular será escuchado por los oídos izquierdo y derecho del receptor en esa posición y orientación.

### Wayverb

Wayverb (Thomas, 2016) es una aplicación gratuita y de código abierto diseñada para simular la respuesta al impulso de una escena a partir de un modelo CAD. Desarrollada en C++ y con aceleración por GPU, Wayverb opera exclusivamente en la

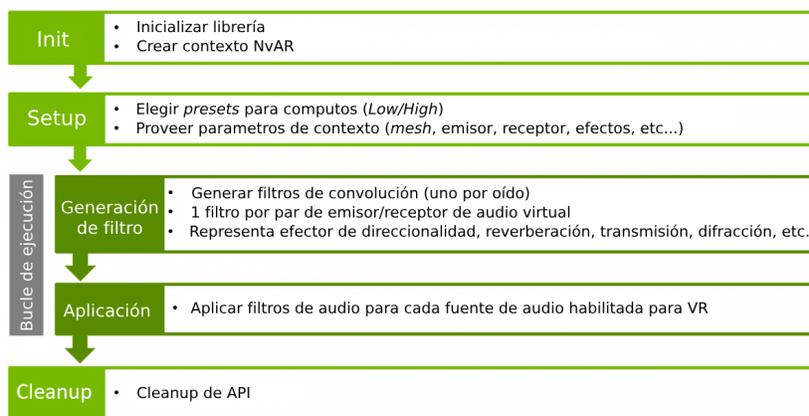


Figura 2.14: Flujo de aplicacion típico de VRWorks Audio.

plataforma MacOS. Los cálculos de la simulación no se producen en tiempo real, y la aplicación carece de la capacidad de aplicar la IR obtenida a una señal de audio para realizar la auralización. Wayverb ofrece una interfaz gráfica que permite la visualización de la escena junto con diversas configuraciones.

Aunque las características específicas de esta aplicación no están descritas explícitamente, su código fuente está disponible y el repositorio del proyecto incluye una explicación detallada de su implementación.

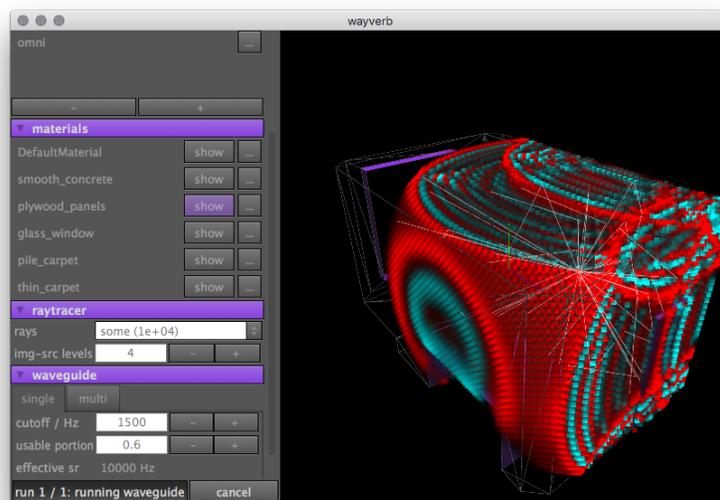


Figura 2.15: Ejemplo de la interfaz de usuario de Wayverb.

## Resonance Audio

Resonance Audio es una SDK de audio espacial multi-plataforma que provee herramientas para simular de manera realista la propagación del sonido en juegos y aplicaciones de realidad virtual/aumentada brindado por Google (Mauskopf, s.f.). Se destaca por su simpleza desde la perspectiva de desarrollo y su eficiencia de ejecución, prometiendo fidelidad en dispositivos con pocos recursos para la computación del audio, como los celulares.

No utiliza *ray tracing*, sino que se basa en utilizar HRTFs detalladas para poder simular las interacciones reales de las ondas con nuestros oídos, creando una detallada ilusión y permitiendo al usuario identificar el origen preciso del emisor.

Posee algunas funcionalidades únicas, como por ejemplo permitir personalizar la dirección de propagación del sonido y la utilización del estándar Ambisonics (Gerzon. M and Ambisonics at Oxford, s.f.) como base del sistema de sonido envolvente. Se desconoce detalles de su funcionamiento interno y las técnicas que utiliza para determinar como la geometría del espacio afecta al sonido.

## Steam Audio SDK

Steam Audio SDK (Valve Software, s.f.) es una SDK multiplataforma para integrar audio espacial que ofrece una amplia gama de funcionalidades para juegos y aplicaciones de realidad virtual/aumentada. Permite renderizar audio posicional en 3D y modelar la oclusión, reflexiones y efectos ambientales, adaptar la acústica en tiempo real a la geometría en movimiento, utilizar aceleración de GPU.

A diferencia de otras herramientas, esta SDK no se encarga de calcular la geometría del espacio para efectuar la renderización del audio espacial; en cambio, opera mediante la utilización de parámetros suministrados por el usuario. Estos parámetros incluyen entre otros: la dirección de la fuente sonora en relación con el oyente, el método de interpolación para estimar la HRTF (ver Sección 2.1.5), así como el grado de mezcla espacial deseado. Al otorgar al usuario el control sobre estos parámetros, el SDK ofrece una flexibilidad significativa para adaptarse a una diversidad de contextos de audio espacial, prescindiendo de la necesidad de procedimientos complejos de modelado geométrico del entorno.

## 2.5. Tecnologías complementarias

En esta sección se presentan algunas tecnologías que acuden a ciertas necesidades de la solución, relacionados a la rasterización, el manejo de señales de sonido y la convolución de señales. Estas son utilizadas en la solución propuesta en el siguiente capítulo

## OpenGL

OpenGL (The Khronos Group Inc, [s.f.](#)) es una biblioteca gráfica desarrollada por la organización The Khronos Group. Su propósito principal es proporcionar una API para la rasterización de imágenes en una variedad de plataformas.

Aunque no es indispensable para la simulación o auralización del sonido en una escena, la visualización de la escena es fundamental en determinados casos de uso, para permitir la exploración detallada de la escena y simplificar el proceso de pruebas y evaluación dentro del entorno. Se destaca que es posible ejecutar la rasterización utilizando exclusivamente con recursos de la CPU, lo que resulta una ventaja si se quiere dedicar los recursos de la GPU a otro fin.

## RtAudio

RtAudio es una biblioteca de C++ que proporciona una API para la entrada y salida de audio en diversas plataformas (Scavone, [s.f.](#)). El propósito fundamental de esta biblioteca de audio es simplificar el acceso al hardware de audio de una computadora, permitiendo que los programas desarrollados puedan grabar y reproducir sonido de manera efectiva.

Una de las características principales de RtAudio es su capacidad para acceder a través de su interfaz a muestras de audio provenientes de un micrófono en tiempo real. Esta funcionalidad simplifica significativamente el procesamiento del audio en vivo para su posterior auralización. Además, ofrece un amplio control sobre parámetros de audio y realiza conversiones internas automáticas para garantizar la interoperabilidad entre diferentes formatos de audio y configuraciones de dispositivos.

## cuFFT

La biblioteca cuFFT (NVIDIA, [s.f.\[b\]](#)) es una herramienta diseñada por NVIDIA para ejecutar el algoritmo FFT en el entorno CUDA (ver Sección [2.2.4](#)) con una eficacia computacional garantizada.

Atributos destacados de cuFFT incluyen su versatilidad para adaptarse a múltiples tamaños de datos de entrada, una complejidad computacional de  $O(n * \log n)$  para todas las dimensiones de entrada, capacidad para procesar tanto entradas como salidas complejas o reales con igual destreza, y además, la biblioteca cuenta con la capacidad para realizar cálculos de transformaciones simultáneas.

cuFFT abarca todas las funcionalidades necesarias para ejecutar convoluciones entre señales de audio e IRs, realizando así eficientemente la auralización deseada.



# Capítulo 3

## Diseño e Implementación

En este capítulo se describe el enfoque adoptado para la elaboración del renderizador acústico, comenzando con la definición de los objetivos y requerimientos del proyecto. A continuación, se presentan los diferentes módulos que componen el sistema desde una perspectiva de diseño. Desde el punto de vista de la implementación, se detalla el desarrollo de componentes clave, abordando aspectos esenciales como la concurrencia, el trazado de rayos y la auralización en tiempo real.

### 3.1. Objetivos

Mediante la comprensión de los conceptos teóricos presentados anteriormente en la Sección 2, principalmente:

1. La propagación del sonido, tanto en la realidad como su simulación en un entorno virtual.
2. La obtención de una respuesta al impulso a partir de técnicas de computación gráfica de trazado de rayos.
3. La convolución de señales para realizar una auralización estéreo en tiempo real.

El presente trabajo tiene como principal objetivo el diseño e implementación de una aplicación destinada a la auralización en tiempo real del sonido en una escena virtual, ya sea anteriormente grabado o generado en tiempo real. La finalidad de la solución es proporcionar una salida estéreo que permita al usuario experimentar una simulación auditiva de alta fidelidad.

Es importante destacar que la solución propuesta se restringe a escenarios donde la propagación del sonido tiene lugar dentro de una atmósfera de aire uniforme. Además, si bien se abordan los efectos esenciales de la propagación del sonido, como la reflexión y absorción, se omite considerar el resto de los efectos mencionados en la Sección 2.1.3, centrándose únicamente en aquellos más relevantes en la mayoría de las situaciones.

Cabe señalar, por último, que no se realizarán simulaciones complejas de HRTFs, que por ejemplo incorporen la geometría corporal o la transmisión del sonido a través del mismo.

## 3.2. Requerimientos

A continuación se detallan los requerimientos de la solución:

1. Carga de escenas tridimensionales virtuales, incluyendo geometría y materiales con coeficientes de absorción.
2. Desarrollo de una interfaz que permita explorar la escena virtual desde la perspectiva del oyente, así como posicionar una fuente de sonido en ella.
3. Implementación de la interfaz utilizando OpenGL sin utilizar recursos de la GPU.
4. Capacidad para especificar si la entrada de audio es de un archivo existente o en tiempo real (a través de un micrófono).
5. Medición de un IR para cada canal de salida (dos en total, para auralización estéreo).
6. Capacidad para configurar diversos parámetros para la medición del IR, con el propósito de facilitar la experimentación
7. Considerar solo reflexiones especulares para la propagación del sonido.
8. El coeficiente de absorción de un material es constante para todas las frecuencias del sonido. El aire no tiene coeficiente de absorción.
9. Aplicación de convolución de las señales utilizando FFT en tiempo real. Cada convolución de cada canal de salida se realiza con su respectivo IR.
10. Medición de los IR periódica mientras el usuario cambia de posición, lo que implica un ajuste correspondiente en la convolución.

## 3.3. Diseño

A continuación se presentan la división en módulos diseñada para la solución. Los módulos son conjuntos de clases y/o partes de clases organizados para minimizar el acoplamiento y reducir la redundancia. También se detallarán las entradas o inputs necesarios por parte del usuario para configurar la aplicación previa a su ejecución, así como las decisiones de diseño que se tomaron para realizar la implementación de la solución.

### 3.3.1. Entradas

Dada la flexibilidad requerida de la solución, se decidió que el usuario deba configurar el comportamiento de la aplicación previo a su ejecución. Se optó por este enfoque dado que en algunos casos, la configuración durante la ejecución podría agregar complejidad al código y dificultar su mantenimiento.

Se cuenta con una gran cantidad de parámetros de entrada, estos se organizan en un solo archivo de configuración en un formato elegido JSON (JSON, [s.f.](#)). Este archivo se divide en tres secciones principales: *renderer\_parameters*, *scene\_parameters*, y *pathtracer\_parameters*.

La sección *scene\_parameters* se enfoca en los detalles de la escena virtual y los elementos de audio. Aquí se especifica la ubicación de los archivos de audio y modelos 3D. Además, se define la posición inicial del receptor (oyente) y del emisor (fuente de sonido) en la escena.

La sección *pathtracer\_parameters* contiene los parámetros específicos del algoritmo de trazado de rayos utilizado para calcular la respuesta al impulso. Esto incluye el número de rayos indexados en cada dimensión (ver Sección 2.2.4), y las características de cada uno: su intensidad inicial (ver Sección 3.3.4), umbrales para la distancia y la intensidad, así como el número máximo de rebotes permitidos. También se pueden especificar materiales personalizados para los objetos en la escena con sus respectivos coeficientes de absorción.

Por último, en la sección *renderer\_parameters*, se especifican los parámetros relacionados con la interfaz de usuario utilizada, así como aquellos parámetros que no encajen en las categorías anteriores. Esto incluye entre otros: el volumen inicial, el umbral de distancia para volver a ejecutar el trazado de rayos (ver Sección 3.4.7), la duración deseada de la respuesta al impulso (IR), la resolución de la pantalla para la visualización. Además, se incluyen opciones para la experimentación, como la capacidad de escribir la respuesta al impulso en un archivo, la distancia y el ángulo que el usuario debe de desplazarse dentro de la escena para volver a realizar la simulación (más sobre esto en la Sección 3.3.5).

### 3.3.2. Módulo geometría

El módulo de geometría es el responsable de cargar la geometría de la escena y el receptor para la utilización del rasterizador y del *path tracer*. Dado que por requerimientos, el rasterizador ejecuta sin utilizar recursos de la GPU y que el *path tracer* ejecuta únicamente a base de recursos de la GPU, la geometría de la escena debe de ser cargada dos veces, una en memoria *host* y otra en memoria de *device*.

La lectura de la geometría incluye la organización de *meshes* poligonales que construyen las figuras, a estas se les asignan un material para identificarlos tanto para la rasterización, como la identificación del coeficiente de absorción posteriormente en el momento de la simulación del IR.

Para cumplir con los requerimientos pautados, durante la implementación de este

módulo se tomaron las siguientes decisiones de diseño:

- La geometría de la escena se carga a partir de un archivo OBJ (Congress, s.f).
- El receptor se modela como dos medias esferas para representar una salida estéreo, cada media esfera registra impactos independientemente. Ver Figura 3.1.
- Con el fin de aumentar la probabilidad del impacto con el receptor, se utilizará un “micrófono” de gran tamaño, la esfera tiene 1 metro de radio.

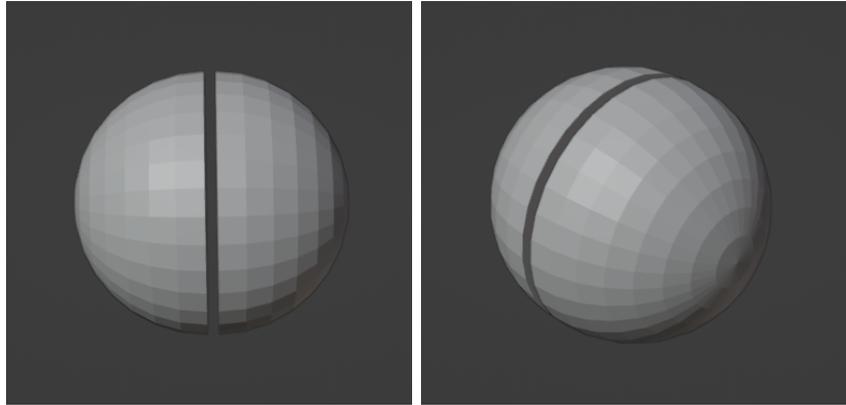


Figura 3.1: Modelo 3D del receptor utilizado.

El cargado de la escena se hace con la ayuda de la librería *tinyobjloader*<sup>1</sup>

Se puede apreciar un diagrama con el flujo de ejecución de las responsabilidades del módulo en la Figura 3.2.

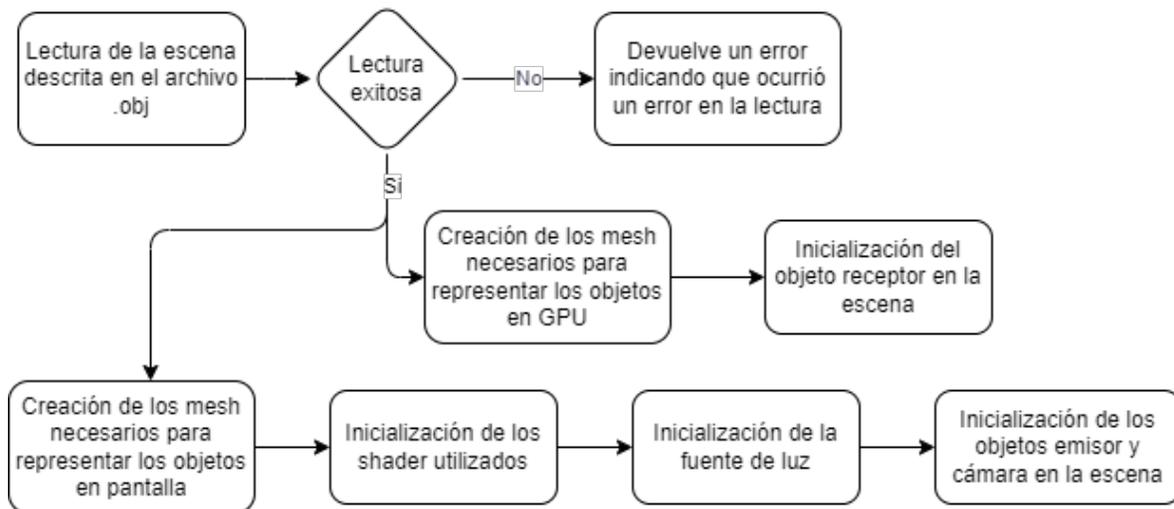


Figura 3.2: Flujo de ejecución típico para el módulo geometría.

<sup>1</sup><https://github.com/tinyobjloader/tinyobjloader>

### 3.3.3. Módulo rasterizador

Este módulo está totalmente ligado a la librería OpenGL. Este módulo recibe como entrada la información del módulo de geometría para compilar *shaders* para manipular la iluminación y finalmente construir la representación tridimensional adecuada para la interfaz del usuario.

El bucle principal de la rasterización controla la actualización continua de la escena, incluida la interacción con la cámara, el dibujo de los objetos en la ventana y las entradas disponibles durante la ejecución del programa (ver Tabla 3.1).

Entrada	Acción
Movimiento del <i>mouse</i>	Rotación de la cámara
W,A,S,D	Movimiento de la cámara
R	Realizar la auralización desde la posición actual del emisor y receptor
E	Colocar el emisor en la posición actual de la cámara
V	Silenciar salida de audio
P	Imprimir próximo IR simulado

Tabla 3.1: Entradas durante la ejecución y sus respectivas acciones en el programa.

Se puede apreciar un diagrama con el flujo de ejecución de las responsabilidades del módulo en la Figura 3.3.

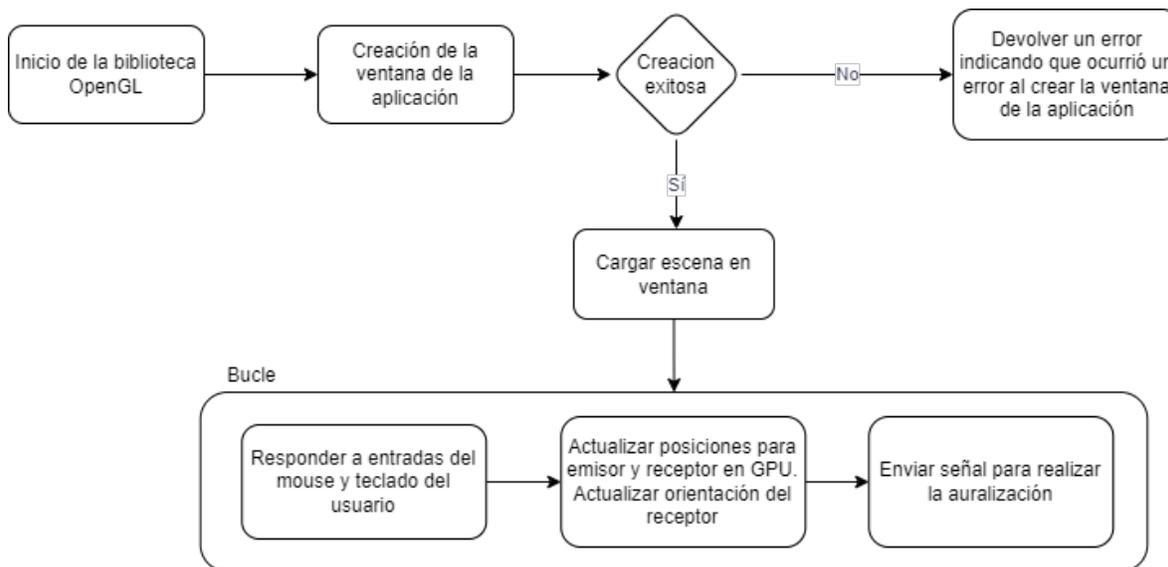


Figura 3.3: Flujo de ejecución típico para el módulo rasterizador.

### 3.3.4. Módulo trazador de caminos

Este módulo es responsable de la configuración y la ejecución del algoritmo de trazado de caminos utilizando el entorno Optix (ver Sección 2.2.5). Es inicializada a partir de varios de los parámetros encontrados en el archivo de configuración así como la geometría cargada por el correspondiente módulo. A partir de estos parámetros el *path tracer* ejecutará el algoritmo a través de la GPU y copiará el resultado en memoria *host* para luego ser accedido por el resto de la aplicación.

Se puede apreciar un diagrama con el flujo de ejecución de las responsabilidades del módulo en la Figura 3.4.

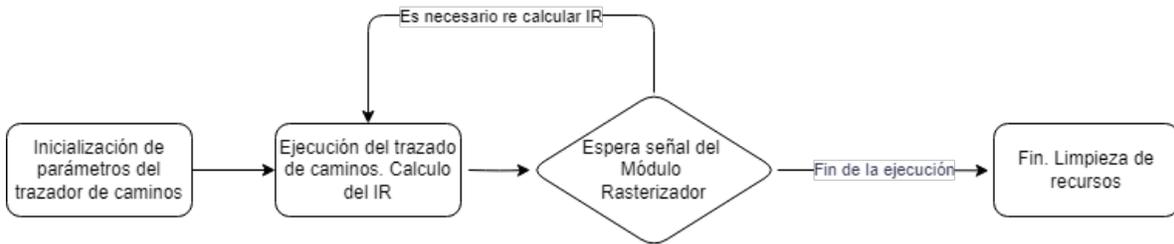


Figura 3.4: Flujo de ejecución típico para el módulo trazador de caminos.

### 3.3.5. Módulo auralizador

Este módulo proporciona las operaciones a realizar para el proceso de auralización. Este parte de las salidas del módulo anterior (IR) para realizar su convolución con el buffer de audio actual, que es representado como un arreglo. En caso de ser un audio pregrabado el que se decide auralizar, el buffer contiene el audio que va a ser emitido en su totalidad. En caso de ser una ejecución con entrada en vivo por micrófono, se trata con un buffer circular que será actualizado con nuevos valores constantemente.

Se denomina buffer circular a una estructura de datos que consiste de un arreglo de tamaño fijo  $N$  y dos punteros o índices que denotan el elemento inicial y final de dicho arreglo. Si se agrega un elemento, el puntero de la cola se incrementa en 1. Si se borra el primer elemento, el puntero de la cabeza se incrementa en 1. Si alguno de los punteros apunta a la posición  $N - 1$  y se quiere incrementar dicho puntero a la siguiente posición, el puntero se mueve a la posición 0. De ahí el nombre buffer circular.

El procedimiento de convolución se resume en la ejecución de la transformada para cada entrada y la realización de la convolución mediante el método *overlap-add* (Smith, 1997). Este método permite la convolución de una señal con un filtro, donde el filtro en este caso es el IR y se explorará en más detalle en la Sección 3.4.6.

Se puede apreciar un diagrama con el flujo de ejecución de las responsabilidades del módulo en la Figura 3.5.

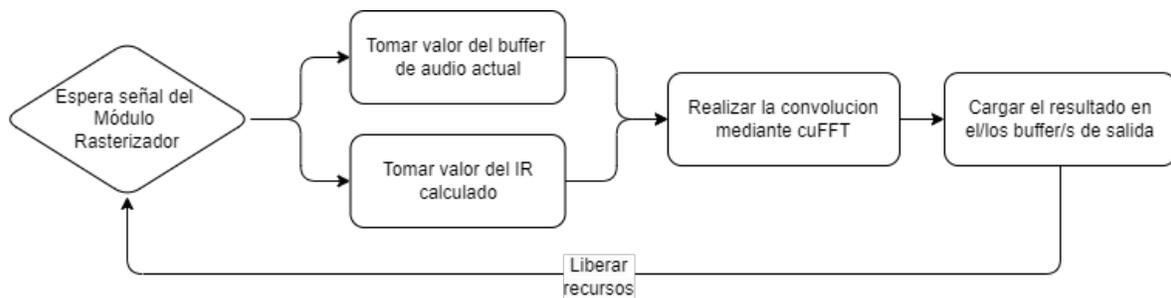


Figura 3.5: Flujo de ejecución típico para el módulo auralizador.

### 3.3.6. Módulo audio

El módulo de audio es el encargado de interactuar con las interfaces del sistema para gestionar la entrada y salida de audio. Para su implementación, se ha empleado la biblioteca RtAudio, como se detalla en la Sección 2.5.

Mediante esta biblioteca, se crean *streams* de audio. Estos *streams* se encargan de capturar los datos de entrada, procesarlos o no mediante una función de *callback*, y dirigirlos hacia un dispositivo de salida correspondiente. Además de esta función principal, RtAudio proporciona herramientas para la manipulación de archivos de audio y el control de los canales de audio disponibles.

El uso de la función de *callback* depende de la entrada elegida para la ejecución. En el caso de que se esté auralizando un archivo de audio, la función de *callback* se limita a tomar la salida del módulo auralizador y modificar el volumen según lo desee el usuario y asignar la señal a la salida de audio. Sin embargo, si la entrada es mediante un micrófono, la recepción del audio de entrada se realiza dentro de esta función de *callback*. Esta debe de ser redirigida hacia el módulo auralizador, para luego proceder normalmente, modificando el volumen y emitiendo la señal por el dispositivo de salida correspondiente.

En el caso de optar por utilizar archivos de audio como fuente de entrada, estos deben estar en formato WAV (Corporation y Corporation, 1991). Para cargar y manipular estos archivos, se hace uso de la biblioteca AudioFile (Stark, s.f.).

## 3.4. Implementación

La siguiente sección desarrolla la elaboración del diseño planteado, así como los fundamentos para la implementación de ciertos puntos clave de la aplicación, como algoritmos o la concurrencia de los procesos.

### 3.4.1. Concurrencia

La decisión de implementar mecanismos de concurrencia en la solución surge de la necesidad de trabajar con flujos de audio, video y sincronización de la información

entre GPU y CPU utilizando recursos compartidos. Estos recursos atraviesan llamadas de lectura y escritura frecuentemente durante la ejecución.

Para abordar esta situación, se opta por separar los módulos en hilos de ejecución (*threads*) distintos. Por un lado, el módulo de geometría se ejecuta secuencialmente antes que el módulo rasterizador, además no interactúan con los recursos compartidos de audio, por ende estos módulos pueden ejecutar en un mismo hilo. Luego, el módulo rasterizador controla el módulo de trazado de caminos de cierta forma, ya que este realiza el llamado para ejecutar el trazado de camino y determina alguno de sus parámetros. Por los mismos argumentos, estos módulos pueden ejecutar en un mismo hilo. El trazador de caminos está ligado al auralizador, que sí interactúa con el buffer de entrada para lectura y con el de salida para escritura. Por otro lado, el módulo de audio es responsable de leer el buffer de salida y escribir en el buffer de entrada.

Dada esta interacción, se decidió separar en un hilo denominado hilo “*screen*” un primer grupo de módulos (geometría, rasterizador, trazador de caminos, auralizador), y en un segundo hilo denominado hilo “*audio*” a todo lo referente al audio (módulo audio y funciones auxiliares).

Para manejar la concurrencia, se utiliza la funcionalidad *mutex*, proporcionada por la biblioteca estándar `<mutex>`. Este mecanismo permite controlar el acceso a los buffers de audio compartidos, asegurando que solo un hilo pueda acceder a ellos en un momento dado y evitando posibles condiciones de carrera.

### 3.4.2. Contexto

Dada la complejidad de los módulos y sus interdependencias, se hace necesaria una arquitectura que facilite la comunicación sin generar un acoplamiento excesivo. Además, dado que ciertos módulos trabajan de forma independiente en threads separados, la coordinación entre ellos es fundamental. Para abordar estas necesidades, hemos definido una clase mediadora llamada Contexto.

El Contexto se implementa como un *Singleton* (Gamma et al., 1994), encargado de mantener referencias importantes, como los buffers de entrada y salida de audio, la geometría tanto de OpenGL como la de Optix, así como otros parámetros compartidos que resulten útiles para su acceso.

En la figura 3.6, se puede observar de manera general su interacción con cada módulo.

### 3.4.3. Trazado de rayos

Se incorporó a la solución el algoritmo base de trazado de rayos (ver Algoritmo 1) realizando los siguientes ajustes. El Algoritmo 2 refleja estas modificaciones.

1. Inversamente al algoritmo base, la emisión se realiza desde la fuente de luz, que en este contexto se trata del emisor de sonido, los rayos terminan su recorrido al colisionar con el receptor, que cumple el rol de “cámara”. Aunque los rayos

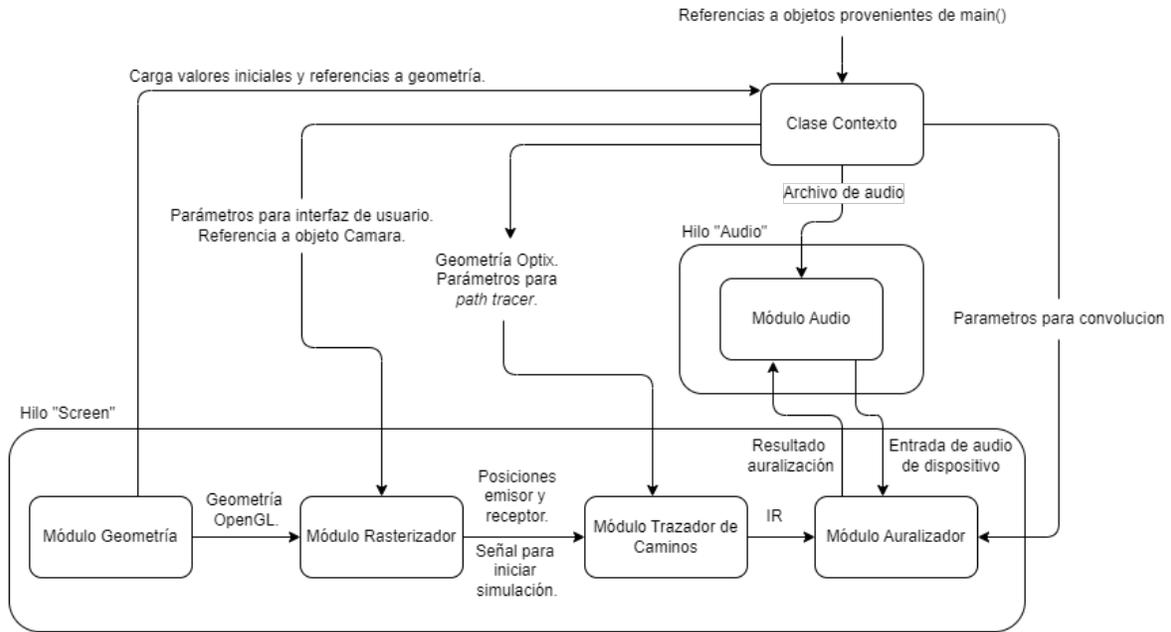


Figura 3.6: Interacción entre los módulos y sus respectivos hilos.

podrían ser emitidos desde la posición del micrófono o receptor, se eligió este ajuste a modo de alinearse conceptualmente con la realidad.

2. El emisor se representa como un punto en el espacio que emite  $N$  rayos o muestras en todas las direcciones, con distribución aleatoria.
3. En este contexto la cámara corresponde al objeto receptor y este claramente no posee píxeles, por lo que se descarta ese factor del algoritmo.
4. El receptor registrará el impacto de los rayos, trazando el camino recorrido para determinar la intensidad que aporta y el momento en el que lo hace.

La implementación de este algoritmo de *path tracing* se apoya fuertemente en la flexibilidad de Optix y la capacidad que la librería posee para modularizar el comportamiento de cada parte del trazado de rayos. El programa correspondiente a la implementación del Algoritmo 2 se aprecia en el Algoritmo 3, representado por una única función llamada *GenRay*. Este se ejecuta individualmente para cada rayo y corresponde a la sección contenida en el bucle *for*.

Se puede apreciar en esta función cómo la distribución de los rayos es pseudoaleatoria. Las secuencias pseudoaleatorias son construidas por un algoritmo a partir de una semilla. Son secuencias de números que no presentan ningún patrón o regularidad aparente desde un punto de vista estadístico.

Cada rayo tiene tres condiciones de parada, cada una correspondiente a límites establecidos por la entrada del usuario (ver Sección 3.3.1).

---

**Algoritmo 2** Algoritmo modificado de *path tracing* para simular propagación del sonido.

---

```
1: Sobre la posición del emisor, distribuir  $N \gg 1$  direcciones aleatorias y uniformes.
2: for dirección do
3:   Generar un rayo (GenRay, Algoritmo 3) y trazar un camino en la escena.
4:   while no se ha alcanzado una superficie receptora
5:     and no se cumple una condición de parada para el camino do
6:       if interseca con la escena then
7:         Calcular la absorción sonora que aporta esta intersección
8:         Calcular una nueva dirección para el rayo
9:         Almacenar los resultados de los cálculos para la siguiente iteración
10:      end if
11:    end while
12: end for
12: Registrar los resultados de las N muestras para producir el IR.
```

---

1. *maxDistanceValue* establece la máxima distancia que puede recorrer un rayo antes de terminar su propagación, es determinado según el parámetro que indica la longitud deseada para el IR en segundos (ver Anexo A) multiplicado por la velocidad del sonido utilizada. De esta forma se evita que los rayos se propaguen innecesariamente a través de la escena dado que si un impacto se produce en un tiempo posterior a la longitud del IR, el mismo no se puede contabilizar.
2. *maxBounces* establece la máxima cantidad de rebotes que puede tener un rayo (que es indicada por su parámetro *recursionDepth*), con este parámetro se apunta a descartar rayos que dada una gran cantidad de rebotes, perdieron una cantidad considerable de su intensidad original, por lo que su aporte se minimiza.
3. *minIntensityValue* está muy relacionado a este último argumento, donde si la *intensity* de un rayo se considera descartable (por ejemplo, luego de impactar contra una superficie de gran absorción), se mejora el rendimiento general del programa al descartarlo y liberar sus recursos.

Se agrega una condición extra, con el propósito de brindar un mecanismo para finalizar la propagación de un rayo. Esto es deseable por ejemplo, cuando se impacta al receptor.

4. *recursionDepth* debe ser mayor a cero, si un programa desea finalizar el bucle de trazado puede lograrlo al establecer esta variable en un valor negativo.

Si estas condiciones lo permiten, el rayo es emitido desde el origen y con la dirección que se almacena en su *payload*, usando la función *optixTraceRay* provista por la librería. De lo contrario, el programa finaliza. Nótese además que la energía acústica inicial de

cada rayo (correspondiente con  $E_0$  en la Ecuación 2.5) es igual a la intensidad total del sonido repartida en iguales partes para cada rayo.

---

**Algoritmo 3** Programa para generación de un rayo.

---

```
1: function GENRAY
2:   rayIndex = optixGetRayIndex()
3:   totalRays = optixGetLaunchDimensions()
4:   // Valores iniciales
5:   payload.intensity =  $\frac{\text{totalIntensity}}{\text{totalRays}}$ 
6:   payload.originPosition = emitter.position
7:   payload.traveledDistance = 0
8:   payload.recursionDepth = 0
9:   // Determinar dirección aleatoria, con distribución uniforme
10:  random.initSeed(rayIndex)
11:   $\theta = 2\pi \times \text{random}()$ 
12:   $\phi = \text{acos}(2 \times \text{random}() - 1)$ 
13:  rayDir.x =  $\sin(\phi) \times \cos(\theta)$ 
14:  rayDir.y =  $\sin(\phi) \times \sin(\theta)$ 
15:  rayDir.z =  $\cos(\phi)$ 
16:  rayDir.normalize()
17:  payload.direction = rayDir
18:  while payload.traveledDistance < maxDistanceValue
19:    and payload.intensity > minIntensityValue
20:    and payload.recursionDepth  $\geq 0$ 
21:    and payload.recursionDepth < maxBounces do
22:    optixTraceRay(payload.originPosition, payload.direction)
23:  end while
24: end function
```

---

La función *optixTraceRay* se encarga de trazar el rayo y llamar al programa de OptiX adecuado según se produzca un impacto o no. Una vez ejecutado este programa, se retorna al bucle de emisión de rayos, donde este puede volver a ser emitido o no dependiendo de sus nuevos valores de *payload*.

Para continuar profundizando en la implementación, a continuación se describen los dos programas posibles a ser invocados por *optixTraceRay*.

El programa de *closest-hit* es el designado a ejecutar en la primera intersección entre un rayo y la geometría de la escena. Es la función responsable de determinar el comportamiento a seguir del rayo, dependiendo de las circunstancias de la intersección.

El primer factor a determinar es la naturaleza del cuerpo con el que el rayo impacta. Para identificar el cuerpo se decidió utilizar el valor del coeficiente de absorción sonora asociado al objeto, que se almacena en la SBT (ver Sección 2.2.5). En el mundo real, los coeficientes de absorción siempre están entre 0 y 1. Sin embargo, en la implementación del programa se decidió asociar a los materiales de los hemisferios del objeto receptor

valores imposibles para sus coeficientes de absorción, en este caso valores negativos. Estos valores negativos de absorción se utilizan exclusivamente como una forma eficiente de identificar al receptor y no tienen ningún otro propósito.

En caso de intersectar al receptor se procede a registrar la intensidad del rayo incidente en la posición correspondiente de cada IR (izquierdo y derecho). En caso de intersectar otro objeto, se determinan las propiedades necesarias para reflejar el rayo, continuando con el trazado de un posible camino. Estos valores incluyen un nuevo origen, una nueva dirección en la que el rayo debe de lanzarse y otros valores de su *payload*. Luego que se establecen estos valores, el programa termina para retornar al programa *GenRay* del algoritmo 3, donde este determina si con estos nuevos parámetros de *payload* se debe de lanzar el rayo o no.

El programa completo se puede apreciar en el pseudo-código del algoritmo 4.

---

**Algoritmo 4** Programa a ejecutar dada la intersección de un rayo con la escena.

---

```

1: function CLOSESTHIT
2:   SBTData = optixGetSBTData()
3:   if SBTData.materialAbsorption = -1,0 then
4:     // Receptor, canal derecho
5:     Ejecutar Algoritmo 6 para el hemisferio derecho
6:   else if SBTData.materialAbsorption = -2,0 then
7:     // Receptor, canal izquierdo
8:     Ejecutar Algoritmo 6 para el hemisferio izquierdo
9:   else
10:    // Objeto de la escena, actualizar parámetros del payload
11:    payload.direction = calculateReflection(payload.direction, SBTData.surfaceNormal)
12:    payload.traveledDistance = payload.traveledDistance + getTraveledDistance()
13:    payload.intensity = payload.intensity * (1 - SBTData.materialAbsorption)
14:    payload.recursionDepth++
15:   end if
16: end function

```

---

Alternativamente, si el rayo no causa ninguna intersección, el programa de *miss* se encarga de terminar la ejecución del rayo, el mecanismo se puede ver el Algoritmo 5.

---

**Algoritmo 5** Programa a ejecutar si el rayo no interseca ningún objeto.

---

```

1: function MISS
2:   payload.recursionDepth = -1
3: end function

```

---

En la Figura 3.7 se aprecia de una forma simplificada el flujo de los programas detallados en esta sección.

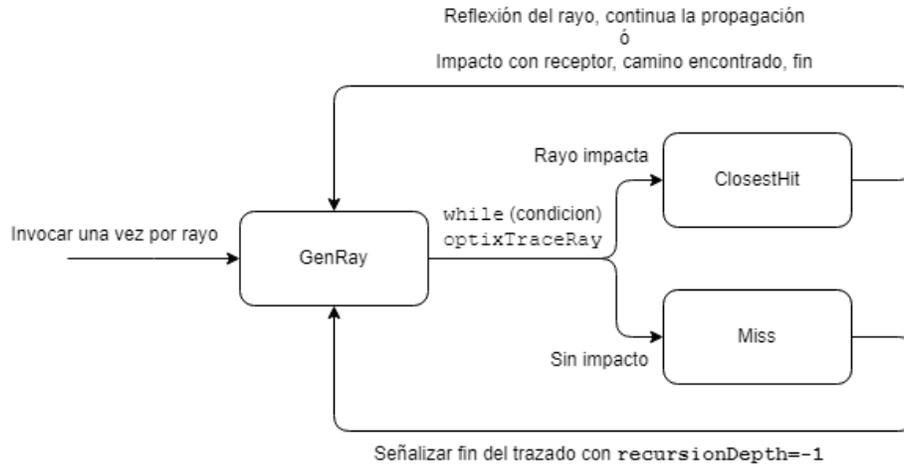


Figura 3.7: Interacción de los programas de trazado de rayos.

### 3.4.4. Construcción del IR

El Algoritmo 6 describe la construcción de los IRs necesarios para la auralización estéreo mediante la interacción de los rayos con el objeto receptor, el resultado es un arreglo de intensidades expresadas en  $W/m^2$  acorde a la ecuación 2.5. Como se muestra en el Algoritmo 4, la construcción se realiza a lo largo de la ejecución del trazado de caminos, con cada impacto con el receptor. El cálculo del aporte de cada rayo al IR del hemisferio donde incide se realiza a partir de la ecuación mencionada y de la carga útil de cada rayo.

- $s$  se calcula dada la dirección actual del rayo y el centro de la esfera obteniendo los puntos de intersección para luego calcular la distancia del recorrido en metros.
- $E_0$  es determinado en el momento de lanzar el rayo.
- Dado los requerimientos, se considera que el aire no tiene coeficiente de absorción ( $h = 0$ )
- Por el Algoritmo 4, dado un impacto con un objeto distinto del receptor con coeficiente de absorción  $\alpha$ , inmediatamente se realiza la multiplicación de la intensidad remanente del rayo por  $(1 - \alpha)$ , por lo que el cálculo de la productoria  $\prod_i (1 - \alpha_i)$  ya fue realizado.
- $V_{receptor}$  es una constante calculada como el valor de una esfera de 1 metro de radio, dividido dos (dado que se está calculando el impacto del rayo con un hemisferio del receptor).

Entonces la Ecuación 2.5 se simplifica a:

$$I_{rayo} = \frac{s * ray\_payload.remaining\_intensity}{V_{hemisferio\ receptor}} \quad (3.1)$$

Una vez calculado el valor de la intensidad del impacto, se identifica la distancia recorrida desde la emisión original ( $d$ ) gracias al valor que los rayos contienen en su carga útil la distancia total recorrida, usando el valor constante de la velocidad del sonido, se puede determinar en que instante de tiempo se realizó el impacto. Con este parámetro temporal, conociendo el *sample rate* del IR, se puede registrar el impacto en la posición deseada.

Para realizar la suma de  $I_{rayo}$  en la posición de tiempo indicada se utiliza la función *atomicAdd* de CUDA. Esta función permite sumar un valor determinado en una posición de memoria sin tener que tratar con una condición de carrera. Una gran ventaja de la utilización de esta función es que, aunque el IR se trate de una estructura de memoria constantemente solicitada y accedida por posiblemente una gran cantidad de hilos de ejecución, no hay que implementar ni utilizar mecanismos para la mutua exclusión.

---

**Algoritmo 6** Construcción del IR dado un impacto de un rayo.

---

```

1: function ADDHITTOIR(HEMISFERIO H)
2:   elapsedTime = payload.traveledDistance / SPEED_OF_SOUND
3:   IRPosition = elapsedTime * sampleRate
4:   Calcular intensityOfRay según la Ecuación 3.1
5:   atomicAdd(IRHemisferio[IRPosition], intensityOfRay)
6:   Llamar a HRTF auxiliar para sumar intensityOfRay al IR del hemisferio opuesto
7: end function

```

---

### 3.4.5. Simulación de HRTF

Trabajos de (Yang y Choudhury, 2021) y (Gardner y Martin, 1994) permiten estimar la HRTF (ver Sección 2.1.5) de un oyente mediante análisis experimentales, donde se han colocado micrófonos en los oídos de personas o maniqués, para grabar cómo se percibe el sonido según la ubicación el emisor. El resultado de dichos experimentos es la representación de una HRTF como un par de IRs (para cada canal de salida) los que pueden ser convolucionados con un audio existente para posicionar virtualmente la fuente de sonido desde el lugar que se generó el IR.

Sin embargo, implementar esta solución para la auralización estéreo resulta problemática. Primeramente, si se asume que se realiza una convolución del sonido con el IR generado a partir de la escena, se necesitaría realizar una segunda convolución con las HRTFs correspondientes a los canales de salida. Además de esto, se necesitaría un repositorio extenso de HRTFs para cada configuración posible del emisor (teniendo en cuenta su distancia, elevación, y ángulo con respecto a la orientación del receptor).

La implementación detallada en la sección anterior (3.4.4) soluciona ambos problemas, ya que cada IR para cada canal tiene información conjunta sobre la escena, la

posición del emisor y sobre como los rayos impactan individualmente a cada hemisferio. De esta forma, solo es necesaria una convolución y no se necesita tener almacenadas HRTFs previas. La desventaja que presenta es que no es una HRTF exacta, ya que no fue generada considerando la anatomía humana ni su efecto en la transmisión del sonido.

Para incluir un factor que simule la interacción del sonido con la anatomía humana se implementa una HRTF simplificada basada en la diferencia de tiempo interaural (ITD), siendo este uno de sus factores más significativos. Determinar el ITD exacto es una tarea compleja, considerando que la velocidad del sonido varía considerablemente en diferentes tejidos biológicos (y también varía según la frecuencia de la onda), además de la complejidad de la anatomía involucrada. También se debe tener en cuenta que depende del punto de impacto y los ángulos horizontales y verticales de incidencia.

Dado un impacto de un rayo en un hemisferio del receptor, se procede a invocar la HRTF simplificada para resolver la intensidad a agregar al IR del hemisferio opuesto, y en qué instante de tiempo hacerlo. Estos dos parámetros se resuelven de la siguiente manera:

- El factor sumado es igual a la intensidad del rayo incidente, pero atenuada por un coeficiente destinado a simular la absorción del sonido por parte de la cabeza humana y su anatomía, el valor del mismo es configurable. Denominamos a este coeficiente  $A$ .
- La posición del IR donde realizar la suma es determinada por el tiempo que llevaría al rayo incidente impactar con el hemisferio opuesto si continúa con su trayectoria actual. Esta distancia está acotada por el radio del micrófono, que en este caso es de 1 m. Por ejemplo, en el caso de que se esté modelando una cabeza adulta, la medida a usar corresponde con la distancia promedio para el ancho de la cabeza de un adulto, que es de 14,5 cm (Lee, Shin e Istook, 2006). Denominamos a este coeficiente  $IDT'$ .

Entonces siendo  $I$  una intensidad y  $T$  un tiempo, la HRTF utilizada es la siguiente

$$HRTF : (I, T) \rightarrow (I, T), \quad HRTF(i, t) = (i * (1 - A), t + IDT') \quad (3.2)$$

### 3.4.6. Convolución

Para realizar el procedimiento, se tomó como base el algoritmo *overlap-add* (Smith, 1997), que se fundamenta en el Teorema 1.

1. Se divide la señal de entrada en segmentos de una duración determinada.
2. Posteriormente, estos segmentos son expandidos agregando suficientes ceros al final del segmento para permitir la expansión del audio durante la convolución. Un ejemplo de la expansión del audio sería al agregar eco.

3. El IR también se expande de la misma forma hasta tener exactamente el mismo largo que el segmento expandido.
4. Una vez preparados los segmentos y el IR, estos se transforman ejecutando la FFT (utilizando la librería `cuFFT`, Sección 2.5) para obtener una señal compleja correspondiente a cada uno.
5. Luego, se realiza la multiplicación de ambas transformadas, estas son representadas como vectores de números complejos de igual largo.
6. Posterior a la multiplicación, se toma el resultado y se le aplica la transformada inversa, que resulta en el segmento original convolucionado, la Figura 3.8 ilustra este procedimiento.

El término *overlap-add* del método empleado se debe a que los segmentos resultantes de la convolución son más extensos que los segmentos originales (debido a la expansión mencionada anteriormente), lo que implica que se superponen en el tiempo y deben de sumarse para reconstruir el audio completo, la Figura 3.9 ilustra el procedimiento.

En caso de estar realizando la convolución de un audio pre-grabado, se puede procesar el audio entero en una sola operación de convolución. Sin embargo, si se está trabajando con audio en vivo esto no es posible, y la operación debe de realizarse periódicamente.

Otra particularidad de trabajar con audio en vivo es que dado un segmento de entrada de tamaño  $n$  *samples*, al realizar su convolución se obtiene una salida de  $n + d$  *samples* (como fue explicado anteriormente debido a la expansión del audio). Estos  $d$  *samples* restantes se superponen con los  $n$  *samples* del siguiente segmento. Si se estuviera trabajando con un audio existente, ya se dispone del siguiente segmento por lo que se suma directamente.

Sin embargo, al trabajar con audio en vivo, el siguiente segmento no se ha generado, además, en la función de *callback*, los *buffers* de entrada y salida tienen el mismo tamaño, por lo que se debe retornar un segmento a la salida de audio de exactamente  $n$  *samples*. Entonces los siguientes  $d$  *samples* deben de ser almacenados para su uso posterior en la siguiente iteración de la función de *callback*. El manejo de este comportamiento se ilustra en la Figura 3.10.

### 3.4.7. Auralización en tiempo real

La auralización en tiempo real en el programa se lleva a cabo mediante un proceso que busca garantizar la continuidad del sonido en función del movimiento del oyente en el entorno virtual. Se basa en el siguiente mecanismo:

1. Tras el trazado de los rayos, se almacena la posición del oyente utilizada, el ángulo del micrófono respecto al eje  $z$  (uno de los ejes horizontales) y el instante de tiempo en el que se realizó.

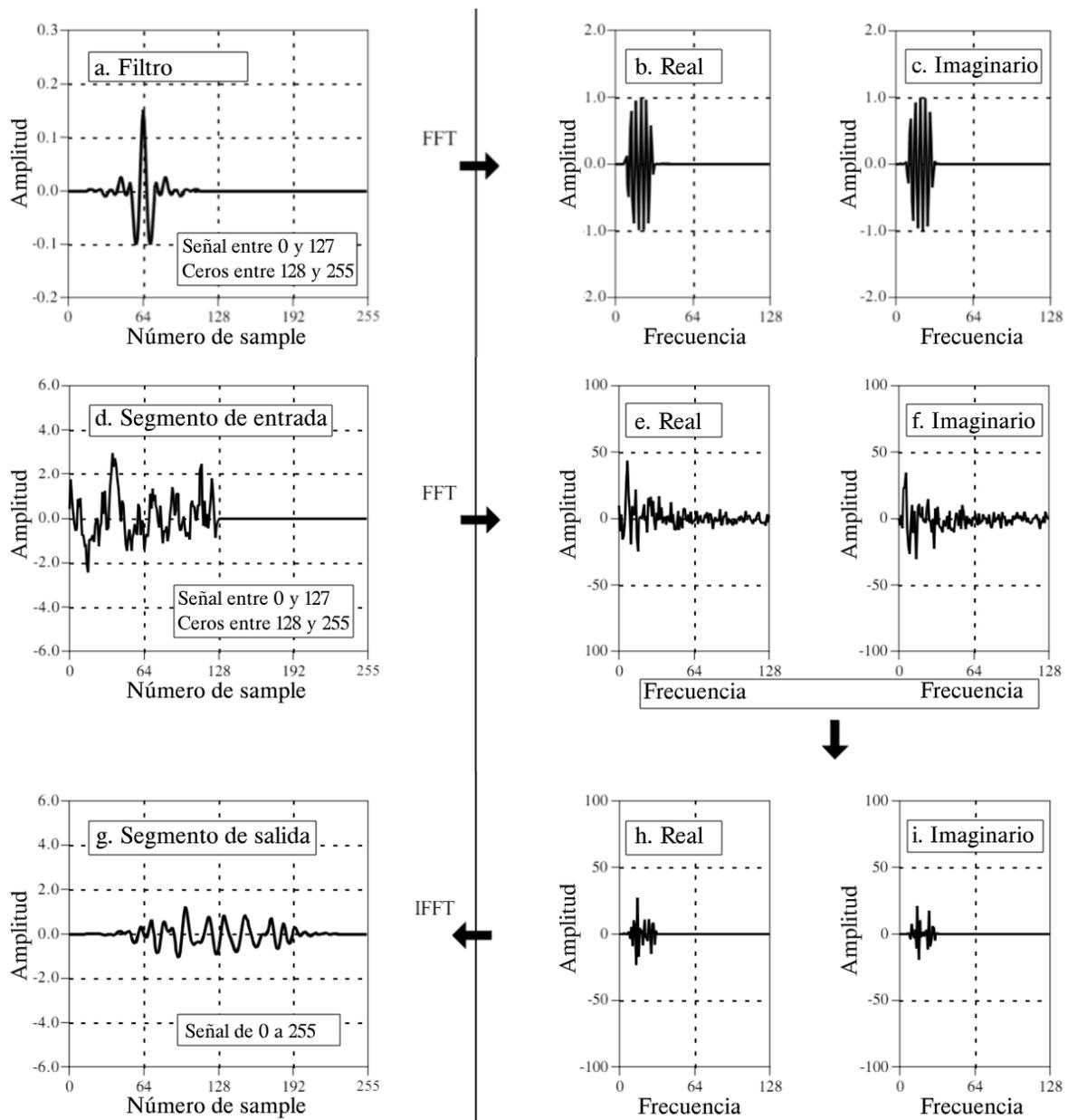


Figura 3.8: El input (d) y el correspondiente filtro (a) expandidos con ceros se transforman mediante FFT en sus componentes complejas (b), (c), (e) y (f). Estos se multiplican para obtener los componentes de la señal de salida (g) y (h), y construyen la salida mediante la inversa de la transformada. Extraído de (Smith, 1997).

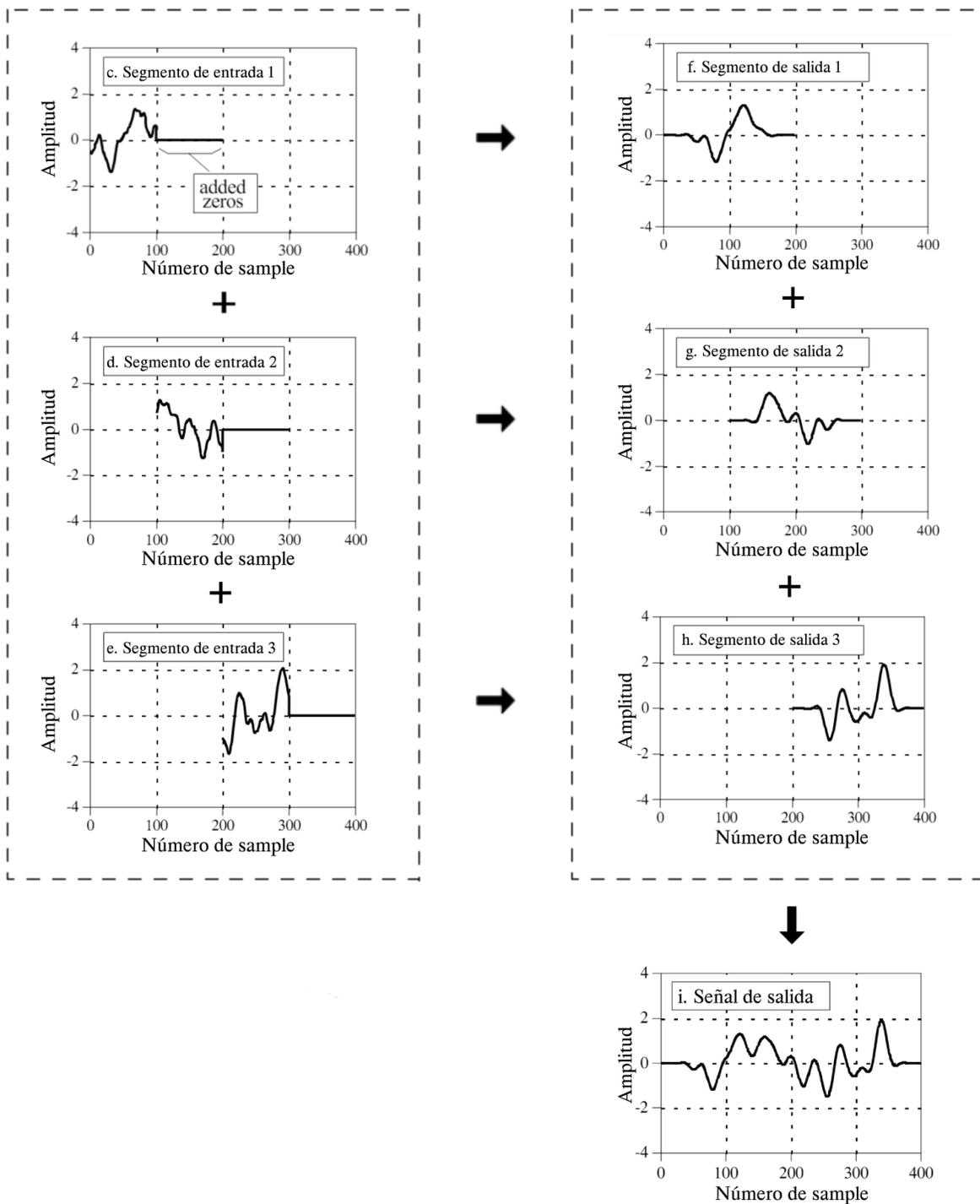


Figura 3.9: Segmentos de la señal expandidos con ceros (c), (d) y (e), con sus segmentos resultantes de la convolución (f), (g), (h). Los segmentos resultantes se superponen en el tiempo. Para construir el audio final es necesario sumarlos. Extraído de (Smith, 1997).

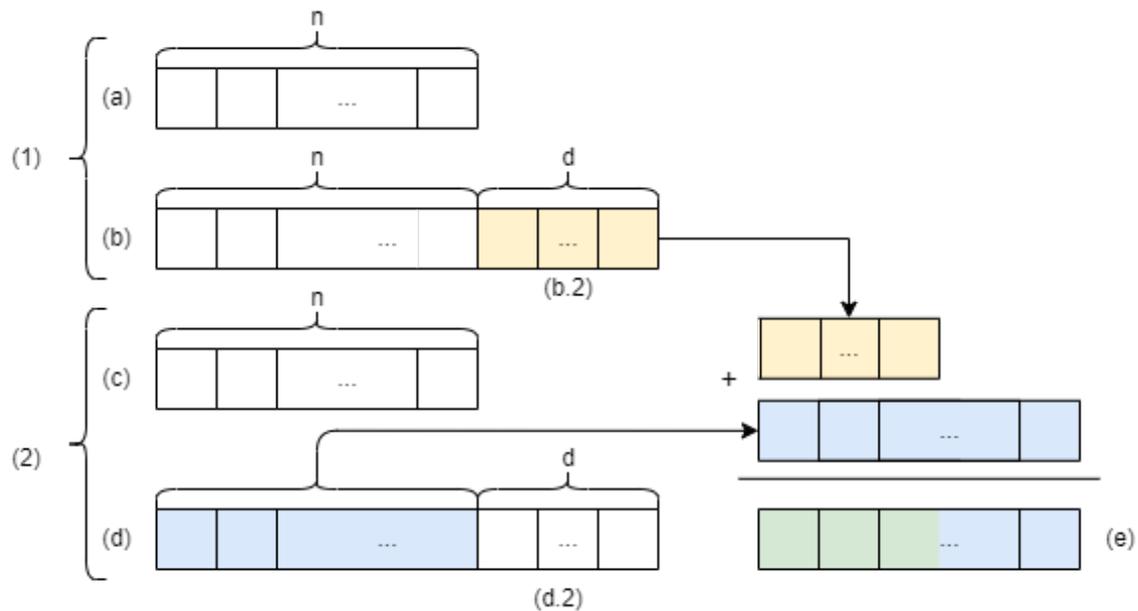


Figura 3.10: Algoritmo para convolución en vivo.

Iteración 1: La entrada (a) se convoluciona para generar el vector (b), los primeros  $n$  *samples* se emiten como salida, y se almacenan los  $d$  restantes (b.2).

Iteración 2: La entrada (c) se convoluciona para generar el vector (d), se toman los primeros  $n$  *samples* de este y se le suma (b.2). El resultado (e) es emitido como salida. El resto de esta iteración (d.2) se almacena para la posterior iteración.

2. Durante el desplazamiento del micrófono en la escena, se calcula:

- La posición actual del micrófono y su distancia con respecto al punto de referencia tomado en el paso anterior.
- El ángulo actual del micrófono y su distancia en grados con respecto al ángulo anterior.
- El tiempo actual

3. Cuando se cumpla alguna de las siguientes condiciones se inicia una nueva ejecución del trazado de rayos:

- La distancia con el punto de referencia anterior supera un umbral determinado por el parámetro *re\_render\_distance\_threshold* (configurable por los parámetros de entrada).
- La rotación del micrófono con respecto al ángulo de referencia anterior supera un umbral determinado por el parámetro *re\_render\_angle\_threshold* (configurable por los parámetros de entrada).
- Si ha pasado más de un segundo desde la última ejecución.

Este proceso implica la modificación de la geometría en la escena de Optix, colocando el receptor en la nueva posición y eliminando el receptor anterior. Acto seguido, se procede a generar los nuevos IRs.

Para finalizar este flujo, en caso de que la fuente de sonido provenga de un archivo pregrabado, la obtención de este nuevo par de IRs implica realizar nuevamente la convolución del audio. Una vez realizado esto, la reproducción del audio continúa desde el último *sample* emitido, asegurando así una reproducción continua. En caso de que la entrada de audio sea en tiempo real, no es necesario realizar pasos adicionales, ya que los nuevos IRs reemplazan automáticamente a los anteriores, y el próximo segmento recibido del micrófono se auralizará con los nuevos parámetros.

Cabe destacar que el movimiento libre del usuario en la escena representa un desafío en términos de sincronización, ya que si se define un valor suficientemente pequeño para *re\_render\_distance\_threshold* este flujo descrito se realizará muy frecuentemente. Esto puede causar un problema debido a que la generación de nuevos IRs y su convolución con el audio deben realizarse antes de que el segmento de audio actual finalice su reproducción. De lo contrario, el siguiente segmento a emitir podría aún no estar convolucionado con la nueva posición, o podría demorar cierto tiempo en salir, ya que se está escribiendo el resultado de la nueva convolución y el *buffer* no está disponible para la lectura, causando así una interrupción en la salida de audio. Se explorará más sobre esta condición en la Sección 4.

# Capítulo 4

## Experimentación

En esta sección se describe el enfoque experimental adoptado para evaluar la implementación de la solución. Se detalla la metodología experimental, que incluye los aspectos que se decidieron evaluar y los sistemas bajo los que se llevaron a cabo las pruebas. También se detalla la metodología de evaluación, especificando los criterios y métricas utilizados para analizar el rendimiento, exactitud y la precisión del sistema.

Posteriormente, se presentan y analizan los resultados obtenidos. Además, se presentan experimentos adicionales que resultan de interés, contribuyendo a una comprensión más amplia del desempeño de la aplicación.

### 4.1. Metodología Experimental

Los experimentos a realizar tienen como objetivo evaluar la solución elaborada en diversas dimensiones: la precisión que presentan los resultados en aspectos de consistencia y reproducibilidad, la exactitud de los mismos en cuanto al resultado esperado, y finalmente el rendimiento de los componentes claves de la implementación

La precisión es esencial para un sistema que realiza simulaciones ya que garantiza la consistencia y confiabilidad de los resultados. Un sistema preciso produce resultados repetibles y uniformes cuando se ejecuta bajo las mismas condiciones, y esto es crucial para validar la consistencia del modelo simulado. Además, dado que este sistema es altamente configurable por diversos parámetros, la precisión es fundamental para comparar los resultados de diferentes escenarios o configuraciones, permitiendo atribuir las diferencias observadas principalmente a los cambios en las variables, y no a la variabilidad del sistema mismo.

Se evalúa la exactitud debido a que el objetivo central del proyecto depende de que los resultados de la solución implementada representen efectivamente la realidad modelada. Para esto es de interés hallar los valores de los parámetros a usar para que las diversas configuraciones brinden una exactitud razonable. Los resultados deseados están dados por las mediciones de respuestas al impulso realizadas por (Bort y Beckers, 2021) sobre escenas reales. Las mismas se presentaran en la siguiente sección. Al modelar

las escenas reales y realizar experimentos sobre estas es posible analizar la correctitud de los resultados de las simulaciones. De forma similar, las simulaciones realizadas por (Satut, 2022) presentan otro punto de comparación que cumple la misma finalidad.

Además, es de interés evaluar el rendimiento de los componentes. Esta dimensión principalmente es motivada por el requerimiento de que la solución sea capaz de realizar la auralización en tiempo real. Para esto es necesario tomar métricas sobre el rendimiento de los procesos más significativos del sistema, como por ejemplo el trazado de caminos y la convolución de su resultado con una entrada de audio. Analizar estos resultados además brindará perspectiva sobre los límites de la aplicación y su dependencia al hardware donde se ejecuta.

### 4.1.1. Pruebas Realizadas

#### Precisión

El principal factor que determina la precisión está dado por la distribución de los rayos a trazar, que es pseudoaleatoria con distribución uniforme en todas las direcciones. Esto significa que dadas dos simulaciones consecutivas utilizando la misma configuración, los resultados producidos no serán iguales. La forma de identificar qué tan distintos son dos IRs se basa en la siguiente simplificación: en lugar de analizar la totalidad de los *samples* de un IR, se toma un *sample* representativo de cada IR. Se elige el que tiene la máxima intensidad (correspondiente al sonido directo que llega al receptor), y a partir de estos máximos se definirá el criterio para evaluar si son suficientemente similares o no.

Se denomina *Just Noticeable Difference* (diferencia de intensidad apenas perceptible) o JND a la medida que corresponde con la diferencia de intensidad más pequeña que es perceptible por el oído (Berg, 2024). El valor de esta diferencia es de aproximadamente un dB según las curvas de Fletcher-Munson (también conocidas como curvas isofónicas) (ver Figura 4.1), en otras palabras, para que no se perciba diferencia entre la intensidad de dos sonidos tiene que haber una diferencia menor a un dB.

Si se tienen dos intensidades diferentes  $I_1$  e  $I_2$ , con  $I_1 > I_2$ , pero que se perciben como iguales, entonces estas deben de cumplir

$$10 * \log_{10} \frac{I_1}{I_2} \leq 1$$

$$\frac{I_1}{I_2} \leq 10^{0,1} = 1,2589$$

Esto significa que dado el cociente entre dos intensidades, la variación debe ser menor a 25,89 % para que el cambio sea imperceptible.

Entonces para evaluar la precisión brindada por una cantidad de rayos determinada se toman las escenas trabajadas y se procede a realizar la simulación del IR un total de 100 veces. Dadas estas simulaciones, como se mencionó anteriormente, se toman los valores máximos y se evaluará que estos no varíen más del 25,89 % con una probabilidad determinada, se eligió para esta prueba un 95 % de probabilidad.

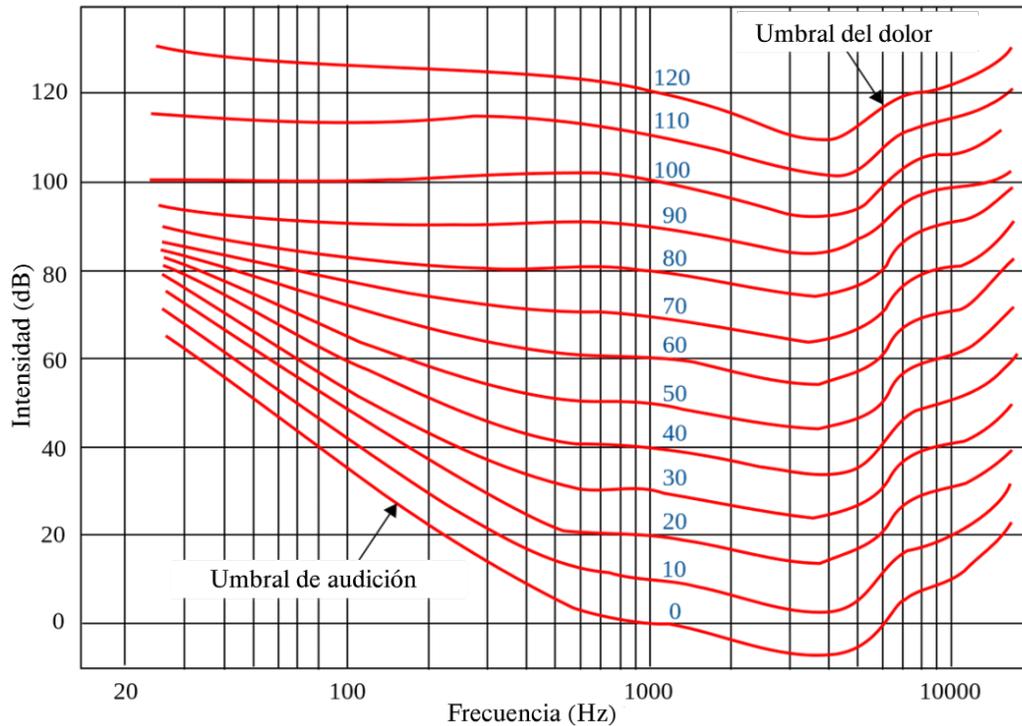


Figura 4.1: Curvas de Fletcher-Munson. Expresan la relación existente entre la frecuencia y la intensidad, permitiendo así evaluar como dos sonidos con distinta frecuencia e intensidad son percibidos como igual de fuertes por el oído. Todos los puntos sobre una misma curva isofónica tienen la misma sonoridad (medida en fonios). Extraído de (Fletcher y Munson, 1933).

Si la distribución de estos valores máximos a utilizar fuese normal, la precisión se puede verificar con el siguiente cálculo:

$$\frac{3,92*\sigma}{\mu} \leq 0,2589$$

Donde

- 3,92 proviene de considerar un intervalo de confianza del 95 %.
- $\mu$  es la media de los máximos de los IRs.
- $\sigma$  es la varianza de los máximos de los IRs.

Si la distribución de estos valores no es normal, se puede estimar el intervalo al centrarlo en la mediana y tomando el 95 % de los datos usando los percentiles 2,5 % y 97,5 %. Luego se verifica la precisión con:

$$\frac{\text{percentile}(97,5) - \text{percentile}(2,5)}{\mu} \leq 0,2589$$

Para determinar cuál de los dos enfoques utilizar, se realiza un test de normalidad de Kolmogorov-Smirnov. Se espera que al aumentar la cantidad de rayos, la varianza baje y por ende la precisión aumente.

### **Exactitud**

Para evaluar la exactitud, se identificaron tres variables que la afectan:

1. **Cantidad de rayos:** Según el trabajo de (Satut, 2022), un mayor número de rayos proporciona un IR simulado con mayor detalle y, por ende, una mayor semejanza con la realidad.
2. **Absorción:** La absorción afecta los rebotes tempranos y la reverberación, ya que la energía que los rayos poseen después de reflejarse en la escena y propagarse influye en el resultado.
3. **Cantidad de rebotes permitidos:** Si no se permiten rebotes, la simulación no presentará reflexiones y a medida que se permiten más rebotes, es más probable que un rayo impacte al receptor. Sin embargo, dado que el propósito del parámetro es optimizar el rendimiento de la aplicación, resulta de interés utilizar esta cantidad lo más baja posible, tratando de mantener la exactitud de los resultados.

Se realizaron diversos experimentos modificando estas variables para cada escena, buscando determinar la configuración de valores que brinda la mayor semejanza posible con la realidad.

A diferencia de la precisión, la exactitud buscada no se define por un criterio cuantitativo, sino que se estima mediante el análisis cualitativo del IR y sus características con respecto a la medida de la realidad obtenida gracias a los experimentos realizados en (Bort y Beckers, 2021).

### **Rendimiento**

Para evaluar el rendimiento, es de principal interés evaluar los tiempos necesarios para la ejecución del flujo de convolución. De esta forma se puede estimar si para un sistema determinado la auralización en tiempo real será posible o no. Se realizaron mediciones de tiempos sobre los siguientes aspectos clave:

1. **Tiempos de trazado de rayos:** El tiempo de trazado de rayos se verá afectado por parámetros como la cantidad de rayos y la complejidad de la escena. Se asume que a más rayos y mayor complejidad, mayor será el tiempo destinado al trazado de rayos.
2. **Tiempos de función de convolución:** Se midió cuánto tiempo toma la ejecución de la convolución de un audio con un IR. Este aspecto se ve afectado principalmente los tamaños de las señales a convolucionar.

3. **Tiempos del proceso completo de convolución:** Dado que la operación de convolución se ejecuta en GPU, establecer los recursos necesarios para la ejecución (como por ejemplo, la transferencia de la información) implica un costo de tiempo. Esta métrica está destinada a poder identificar el peso de dichas operaciones. Se define el tiempo del proceso completo de convolución como la diferencia de tiempo entre el instante previo a la inicialización del proceso y el instante de tiempo posterior a la liberación de los recursos utilizados y la obtención del resultado en memoria *host* (Ver Sección 2.2.4). Este tiempo contiene al tiempo correspondiente al punto anterior.

Para cada experimento se realizaron 5 mediciones y se promediaron los resultados. Se considera que la aplicación es eficiente si los tiempos de ejecución son suficientemente bajos para permitir la auralización en tiempo real, característica que se verificó empíricamente.

#### 4.1.2. Escenas Utilizadas

Para realizar el análisis se estudian tres escenas reales (configuraciones urbanas) trabajadas por (Bort y Beckers, 2021). La primera escena, denominada “1D”, corresponde a una fachada con vista abierta. La segunda, “2D”, representa un cañón urbano. La tercera, “3D”, se refiere a una pequeña plaza o patio interno. Las mismas se pueden ver representadas en la Figura 4.2.

Cada escena fue simplificada y modelada geoméricamente por (Satut, 2022) como se puede apreciar en la Figura 4.3. Aunque se pueden apreciar diferencias en las medidas y distancias con respecto a las originales, se deciden utilizar estas escenas para poder referenciar y comparar los resultados con los del trabajo realizado por (Satut, 2022).

#### 4.1.3. Hardware Utilizado

Se utilizan tres ambientes de prueba correspondientes a los ambientes de trabajo de los estudiantes. Cada sistema mantiene el mismo versionado en cuanto al software del entorno de desarrollo pero difiere en cuanto a hardware, como se puede apreciar en la Tabla 4.1. A modo de comparación, también se incluye en la tabla el hardware utilizado en (Satut, 2022).

En el proyecto se utilizan tres dependencias de software principales. Para la programación de actividades en GPU se utilizó CUDA en su versión 12.1.105 y OptiX en su versión 7.7. Asimismo, se empleó RtAudio en su versión 6.0.1, este último responsable de la entrada y salida de audio. Las diferencias en hardware son de gran utilidad al comparar los resultados de las pruebas de rendimiento sobre cada sistema (ver Sección 4.2.3). Es importante destacar que los resultados de las simulaciones realizadas son independientes del sistema sobre el que se ejecuten. Esto garantiza que los resultados obtenidos son comparables entre los diferentes sistemas.

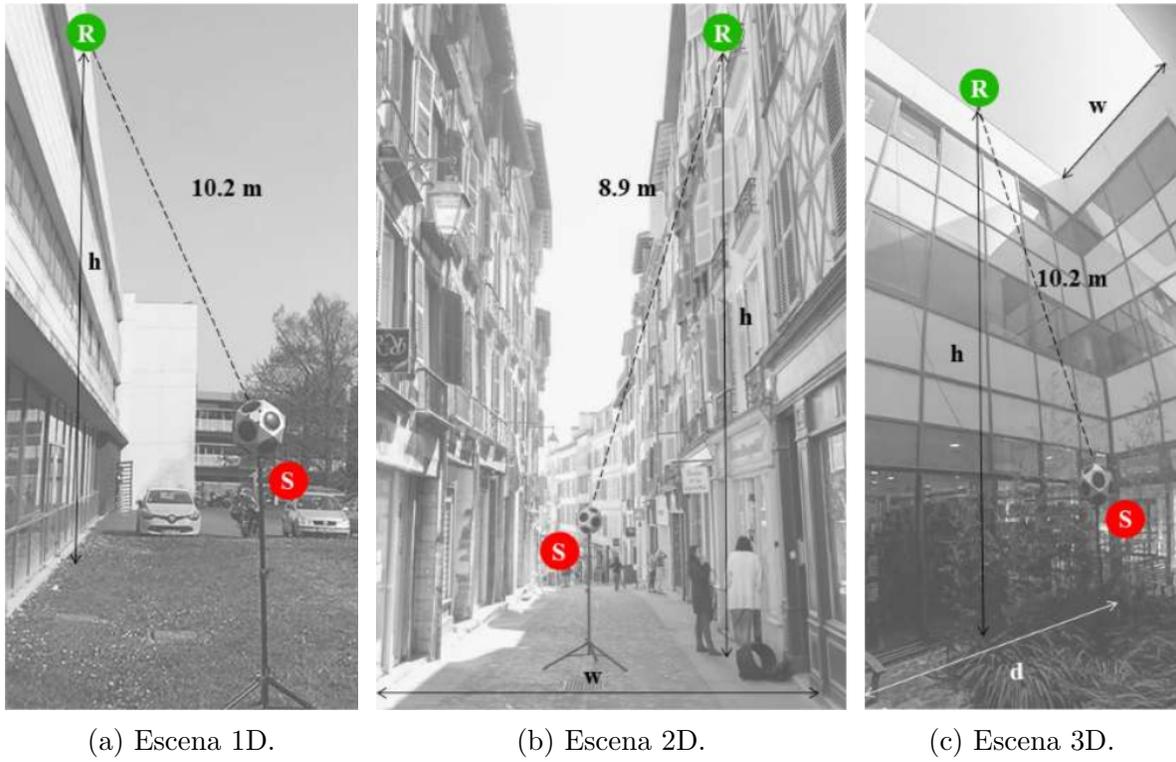


Figura 4.2: Escenas reales estudiadas. S marca la posición de la fuente (*source*) y R marca la posición del receptor (*receiver*). Extraído de (Bort y Beckers, 2021).

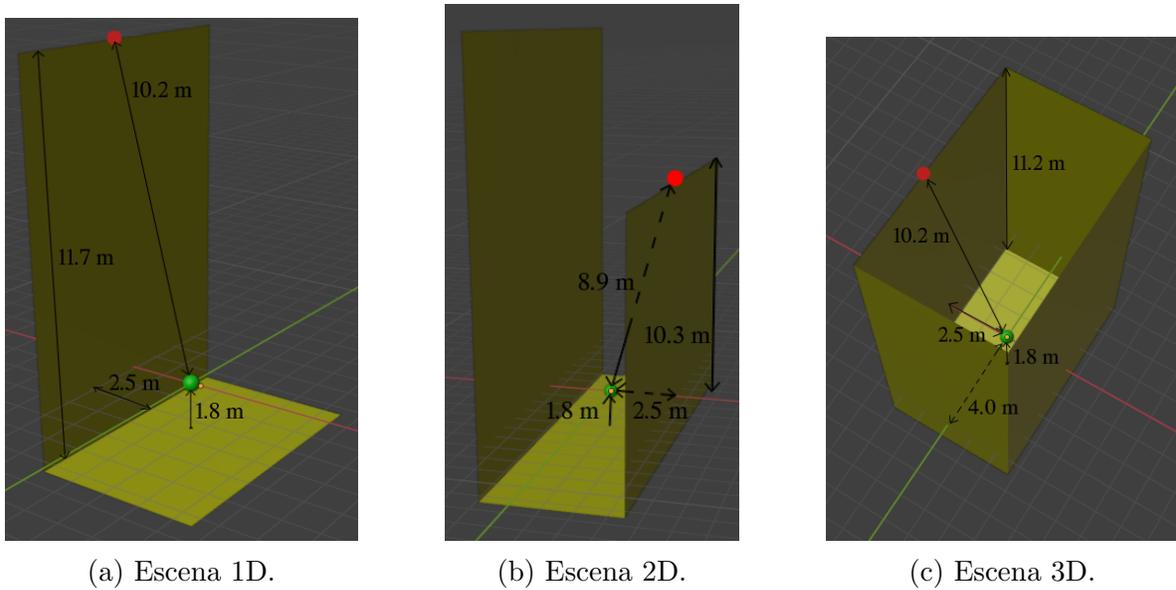


Figura 4.3: Capturas de los modelos geométricos asociados a las escenas reales de la Figura 4.2 con sus medidas, el receptor es equidistante a paredes opuestas. Escenas generadas por (Satut, 2022).

Ambiente	1	2	3	(Satut, 2022)
SO	Windows 11	Windows 11	Windows 11	Windows 10
GPU	RTX 3070 - 8GB	GTX 1050 Ti - 4GB	RTX 2080S - 8GB	-
CPU	Intel i7-12700k	AMD Ryzen 3 3200G	Intel i5-9600k	Intel i7-6500U
RAM	64 GB - 3200 MHz	16 GB - 2400 MHz	32 GB - 3200 MHz	8 GB - 2133 MHz

Tabla 4.1

## 4.2. Resultados obtenidos

En esta sección se brindan los resultados de los experimentos detallados en la Sección 4.1.1.

### 4.2.1. Pruebas de precisión

En las siguientes tablas se pueden observar los resultados de las pruebas de precisión para la escena 1D (Tabla 4.2), 2D (Tabla 4.4) y 3D (Tabla 4.4).

Recordando que el criterio de evaluación de la precisión a aplicar depende de si la distribución de los valores se considera normal, en primer lugar se realizó un test de normalidad sobre los resultados. Los únicos resultados que no cumplen con la normalidad son los valores generados con  $10^3$  rayos, por lo que su variación fue calculada con la formula  $(\text{percentile}(97,5) - \text{percentile}(2,5))/\mu$ . Por otro lado se determina que para los resultados de  $10^5$  rayos en adelante la distribución es normal, por lo que se evalúa calculando  $3,92 * \sigma/\mu$

Dado un resultado para la variación, se considera que los resultados son suficientemente precisos si son menores o iguales a 0,2589.

Contemplando los resultados de la varianza, en cada una de las tablas se puede observar cómo esta disminuye mientras aumenta la cantidad de rayos, que afirma las hipótesis sobre la relación entre esta variable y una mejora de precisión.

Teniendo esto en consideración, los únicos resultados que podemos concluir como suficientemente precisos son aquellos donde la cantidad de rayos fue por lo menos  $10^7$  para todos los casos estudiados. Se deben realizar más análisis con otros escenarios para indagar si la precisión persiste o si la cantidad de rayos debe de aumentar.

<b>Cantidad de rayos</b>	Test de normalidad	$\mu$	$\sigma \times 10^3$	Variación	$\leq 0,2589$
$10^3$	Negativo	0,01148	N/A	1,846	No
$10^5$	Positivo	0,00194	0,329	0,558	No
$10^6$	Positivo	0,00172	0,139	0,319	No
$10^7$	Positivo	0,00167	0,051	0,120	Si
$10^8$	Positivo	0,00166	0,015	0,035	Si
$10^9$	Positivo	0,00166	0,004	0,009	Si

Tabla 4.2: Resultados de precisión escena 1D. Los valores de  $\mu$  se expresan en  $W/m^2$ , y los valores para  $\sigma$  en  $W/m^2$ .

<b>Cantidad de rayos</b>	Test de normalidad	$\mu$	$\sigma \times 10^3$	Variación	$\leq 0,2589$
$10^3$	Negativo	0,01216	N/A	1,707	No
$10^5$	Positivo	0,00251	0,383	0,598	No
$10^6$	Positivo	0,00169	0,133	0,310	No
$10^7$	Positivo	0,00215	0,053	0,097	Si
$10^8$	Positivo	0,00216	0,019	0,034	Si
$10^9$	Positivo	0,00216	0,005	0,009	Si

Tabla 4.3: Resultados de precisión escena 2D. Los valores de  $\mu$  se expresan en  $W/m^2$ , y los valores para  $\sigma$  en  $W/m^2$ .

Cantidad de rayos	Test de normalidad	$\mu$	$\sigma \times 10^3$	Variación	$\leq 0,2589$
$10^3$	Negativo	0,01094	N/A	1,238	No
$10^5$	Positivo	0,00197	0,362	0,720	No
$10^6$	Positivo	0,00169	0,158	0,368	No
$10^7$	Positivo	0,00167	0,048	0,113	Si
$10^8$	Positivo	0,00166	0,016	0,038	Si
$10^9$	Positivo	0,00166	0,005	0,012	Si

Tabla 4.4: Resultados de precisión escena 3D. Los valores de  $\mu$  se expresan en  $W/m^2$ , y los valores para  $\sigma$  en  $W/m^2$ .

#### 4.2.2. Pruebas de exactitud

Para estas pruebas se procedió a variar los parámetros para el trazado de caminos para buscar determinar la mínima configuración que brinda la mayor semejanza posible con la realidad. Los parámetros bajo experimentación son las absorciones de los materiales en la escena y la cantidad de rayos emitidos. Dentro de una misma escena se mantienen constantes para todos los experimentos: las posiciones de receptor y emisor, el nivel sonoro ( $L_w = 100$  dB, mismo valor que el usado en (Satut, 2022) para las simulaciones y en (Bort y Beckers, 2021) para las mediciones), la longitud del IR (2 segundos, a un *sample rate* de 16 KHz) y el tamaño del objeto receptor. Además, no se limitaran la cantidad de rebotes.

Los resultados presentados son IRs generados en modo *mono*, es decir que no se distinguen impactos entre hemisferios del receptor, y la orientación del receptor con respecto al emisor no es una variable que afecte los resultados.

#### Escena 1D

Como fue mencionado en 2.1.1 la medición empírica del IR posee valores positivos y negativos ya que se tratan con valores de la presión efectiva. Por lo tanto, dado que la intensidad es directamente proporcional al cuadrado de la presión efectiva, los valores de la medición son elevados al cuadrado para poder compararlos con los resultados de la simulación. El resultado de la medición incluyendo esta modificación es visible en la Figura 4.4.

La escena 1D es el escenario más sencillo de analizar. Al ser una fachada abierta no existen muchos caminos por el que un rayo puede llegar hasta el receptor. Como se ve en la medición de la Figura 4.4, esta escena presenta 2 picos bien distinguidos, que evidencian la poca cantidad de caminos mencionada. El primer pico es a los 29,69 ms mientras que el segundo a los 37,45 ms. El pico más prominente, el primero, representa

los impactos directos mientras que el siguiente es el resultante de los caminos que llegan al receptor luego de algún rebote, con el suelo por ejemplo. La presencia de los rebotes se ve ilustrada en la Figura 4.5.

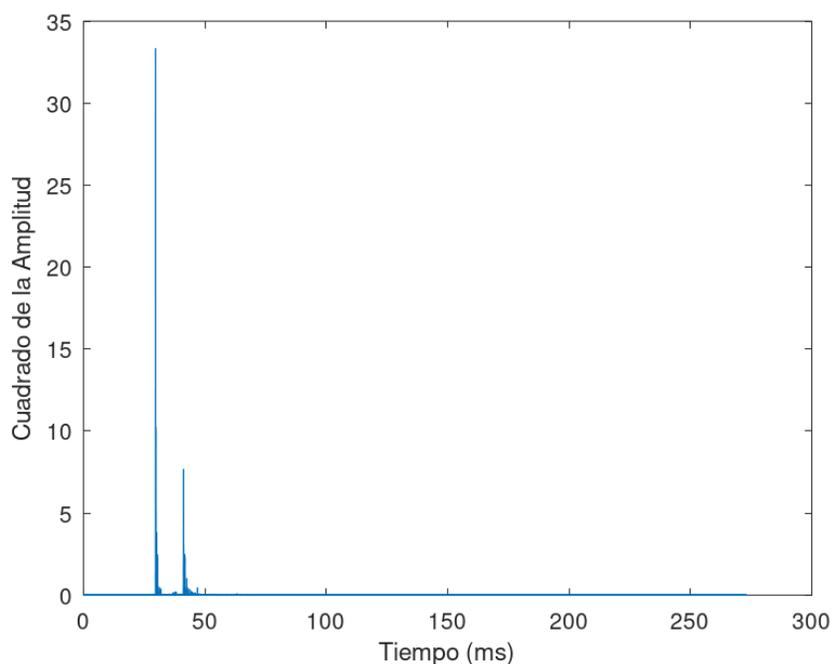


Figura 4.4: Respuesta al impulso medida en la escena 1D. Generado a partir de los resultados de (Bort y Beckers, 2021). Se presentan únicamente los primeros *samples* de este IR para apreciar el detalle.

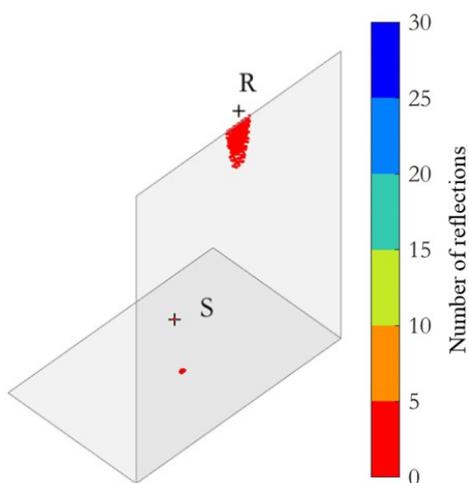


Figura 4.5: Rebotes en la escena 1D, se colorea el último rebote en la propagación de un rayo antes que este alcance al receptor. Extraído de (Bort y Beckers, 2021).

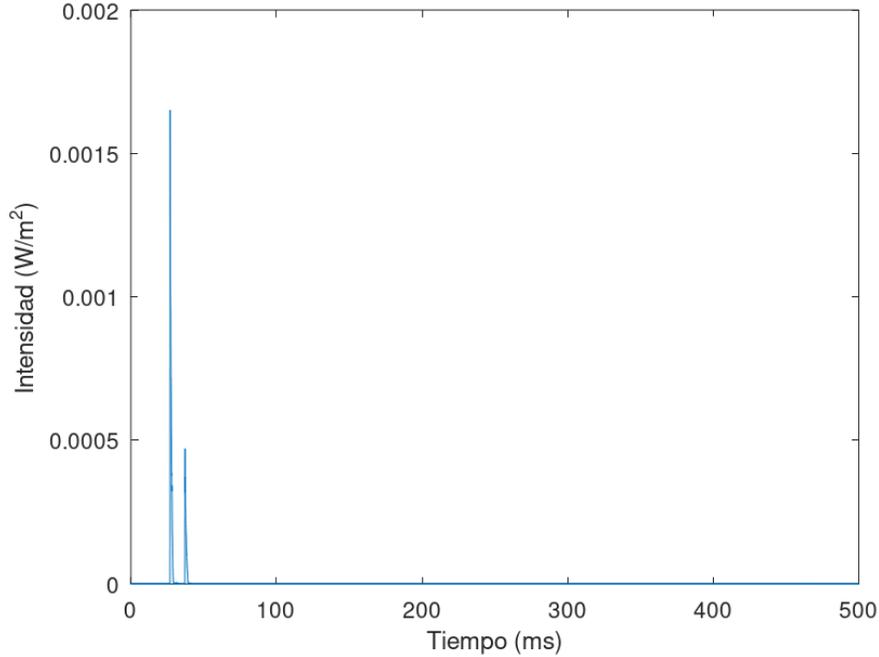


Figura 4.6: IR simulado obtenido con  $10^7$  rayos.

Por el análisis presentado y la Figura 4.5 se puede ver que no existen caminos con más de 2 rebotes. Se selecciono la **cantidad de rayos emitidos** como la primera variable a investigar, para este fin se mantuvo constante el coeficiente de absorción de los materiales en 0,5.

Se obtuvo el IR de la Figura 4.6 al emitir  $10^7$  rayos dentro de la escena. Se puede observar que también posee dos picos, donde el primero ocurre en el milisegundo 27,11, y el segundo en 37,05. En la escena virtual el centro del receptor se encuentra a 10,3 m del punto de emisión, teniendo en cuenta que se representa como una esfera de radio 1 m, su superficie está a 9,3 m. del punto de emisión. Entonces, el sonido que se propaga a través de un camino directo entre el emisor y receptor lo hace en  $9,3/343 = 27,11$  ms, lo que explica el tiempo para el primer pico y la sutil diferencia con respecto al tiempo del primer pico en la medición. El segundo pico corresponde con rayos que deben de rebotar contra el piso, recorriendo aproximadamente un total de 12,7 m desde la emisión, distancia que es recorrida en  $12,7/343 = 37,02$  ms.

Similarmente, todos los experimentos realizados en esta escena poseen dos picos en momentos similares de tiempo, mostrando consistencia en cuanto a los caminos encontrados por cada ejecución independiente del trazado de rayos.

La Figura 4.7, tomada de (Satut, 2022) muestra una simulación del IR de la escena 1D, que en dicho trabajo se describió como “la que generó resultados más similares a la medición”. Se pueden encontrar variadas similitudes al contrastar el resultado con la Figura 4.7, que resulta esperable dado que se utilizaron parámetros idénticos.

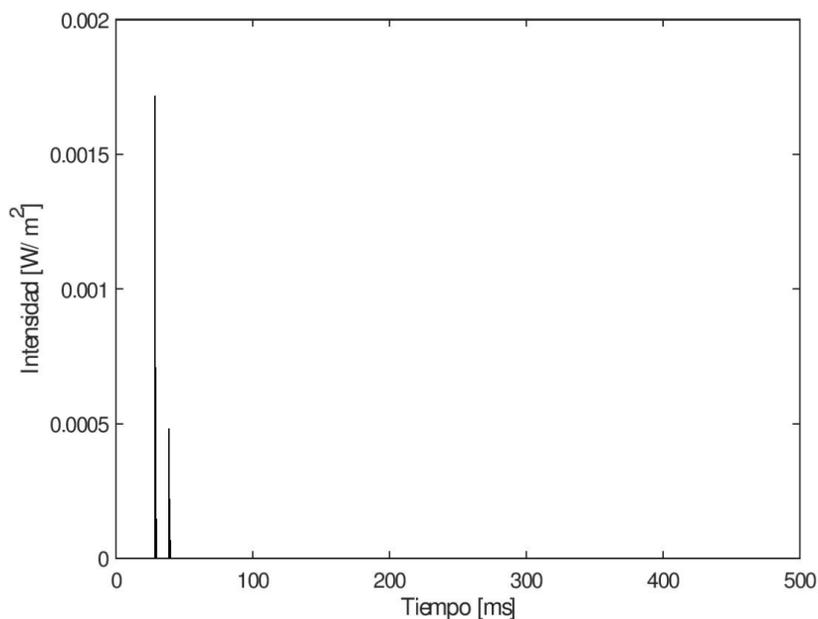


Figura 4.7: IR generado con  $10^7$  rayos, coeficiente de absorción de 0,5, máxima cantidad de rebotes 30. Simulación extraída de (Satut, 2022)

Dado que resultados con cantidades mayores a  $10^7$  no fueron presentadas en (Satut, 2022) se procede a experimentar al aumentar la cantidad de rayos emitidos a  $10^8$ . La Figura 4.8 muestra el resultado de la simulación.

Por lo visto en los resultados de la Sección 4.2.1 para la escena 1D,  $10^7$  es suficiente para obtener resultados precisos, y como es esperable no se obtuvieron cambios perceptibles en los resultados. Se puede afirmar que aunque  $10^8$  produce resultados menos variables, para la escena 1D,  $10^7$  rayos emitidos es cantidad suficiente para obtener resultados exactos y comparables con las mediciones reales.

Procediendo con el análisis de las variables dentro de la escena, se eligió investigar la exactitud del **coeficiente de absorción** utilizado.

Por el análisis previo realizado se afirma que el segundo pico presentado en el IR de la escena corresponde con los caminos que recorren una distancia mayor a la de los impactos directos, por lo que deben de ser aquellos que rebotan contra el suelo. Claramente, estos caminos presentan por lo menos un rebote y su aporte al IR se ve afectado por el coeficiente de absorción. Además, se afirma que el valor máximo del segundo pico corresponde con el primer valor del pico, es decir, el que llegó más rápido hasta el receptor y tuvo un recorrido menor.

Por la Figura 4.5 se puede ver que además de rebotar contra el suelo, los caminos pueden rebotar contra el área en la pared debajo del receptor. Se pueden despreciar aquellos que también rebotan contra la pared debido a que sus trayectos son más largos que los que solo rebotan contra el piso, por lo que su aporte es posterior y no corresponde

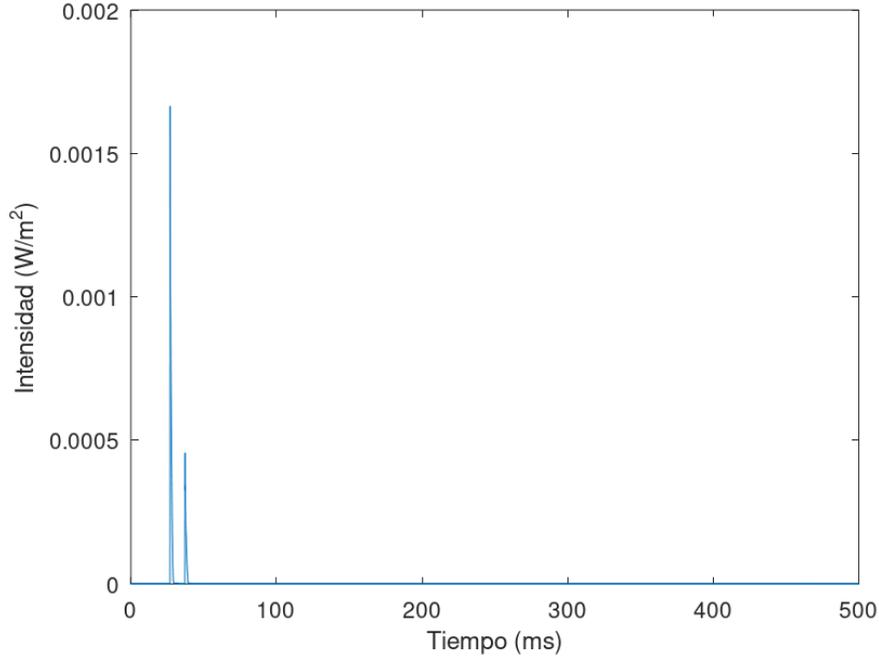


Figura 4.8: IR simulado obtenido con  $10^8$  rayos.

con el máximo del segundo pico. Finalmente, por este razonamiento se puede afirmar que el valor máximo del segundo pico corresponde con la intensidad aportada por ciertos caminos que rebotan una única vez.

Usando la medición de la Figura 4.4 se identifica que la relación entre los valores de los picos de la siguiente forma:

$$\frac{\text{Valor máximo del primer pico}}{\text{Valor máximo del segundo pico}} = \frac{33,33}{7,68} \approx 4,34$$

Dados los resultados de los experimentos de precisión (ver Sección 4.2.1) se puede ver que usando  $10^7$  rayos, la media para el máximo del primer pico corresponde a 0,00167. Usando una metodología similar, se procedió a realizar 100 simulaciones de  $10^7$  rayos con un coeficiente de absorción igual a 0 (es decir, no absorbe). Dados estos resultados se procedió a analizar los IRs y obtener la media para el segundo pico, equivalente a aproximadamente  $0,00095 W/m^2$ .

Sabiendo que al segundo pico se le aplica la absorción una única vez, y el resultado debe de ser 4,34 veces menor a la intensidad máxima del primer pico, se puede calcular el coeficiente de absorción a utilizar (representado por  $x$ ) en esta escena por la siguiente ecuación, donde  $\mu_1$  es la media para el primer pico, y  $\mu_2$  es la media para el segundo si no se le aplica una absorción:

$$\frac{\mu_1}{(1-x)*\mu_2} = \frac{0,00167}{((1-x)*0,000955)} = 4,34 \rightarrow x \approx 0,598$$

La Figura 4.9 muestra el resultado obtenido con este coeficiente de absorción.

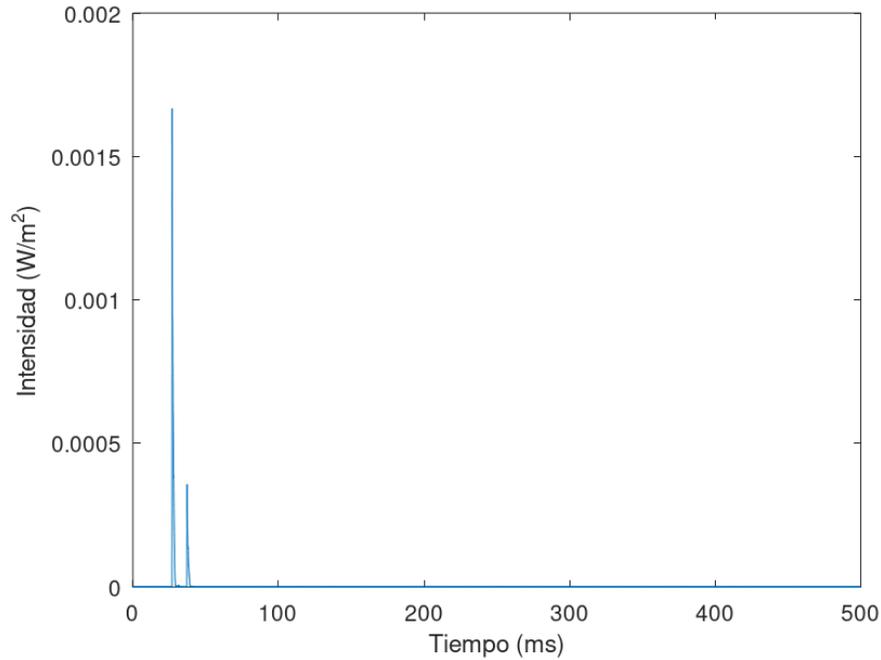


Figura 4.9: IR simulado obtenido con  $10^7$  rayos y un coeficiente de absorción de 0,6.

## Escena 2D

La escena 2D introduce una segunda pared que brinda una nueva superficie para que los rayos puedan reflejarse, aumentando la cantidad de caminos con el receptor encontrados. Como se ve en la Figura 4.11 solo aquellos rayos con direcciones semi-alineadas con el plano perpendicular al suelo que atraviesa el origen de emisión y el centro del receptor lograrán impactar.

En la medición de la Figura 4.10 se puede apreciar como la cantidad de picos ha aumentado de gran forma, causando que el IR sea más extenso en el tiempo. Se destacan unos 3 picos al principio de la gráfica, alrededor de los 25,95 ms, 33,41 ms y 37,16 ms, seguidos de una cola de decaimiento exponencial. Es importante recordar que la ubicación del receptor en esta escena cambia con respecto a la anterior, por lo que los tiempos de los picos claramente también lo harán.

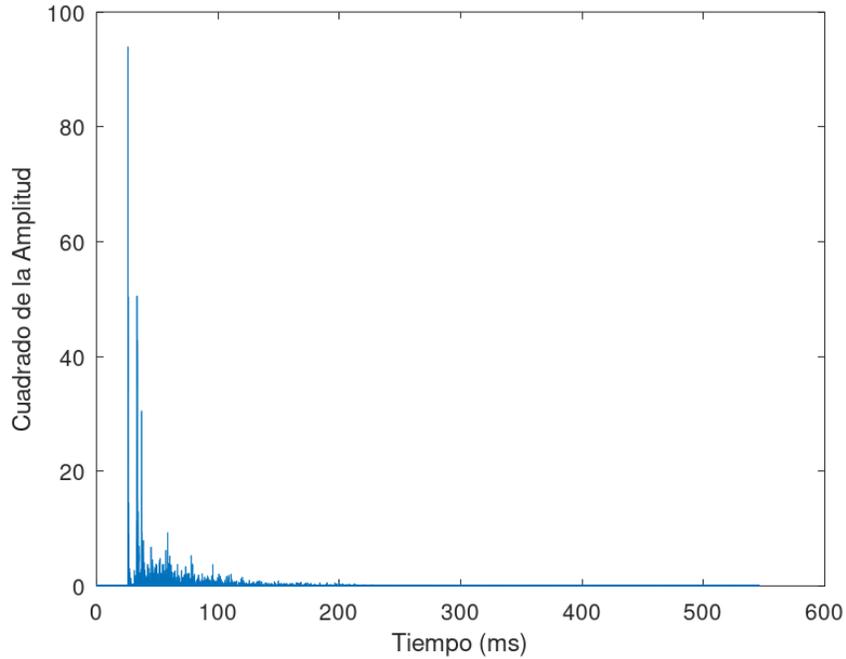


Figura 4.10: Respuesta al impulso medida en la escena 2D. Generado a partir de los resultados de (Bort y Beckers, 2021). Se presentan únicamente los primeros *samples* de este IR para apreciar el detalle.

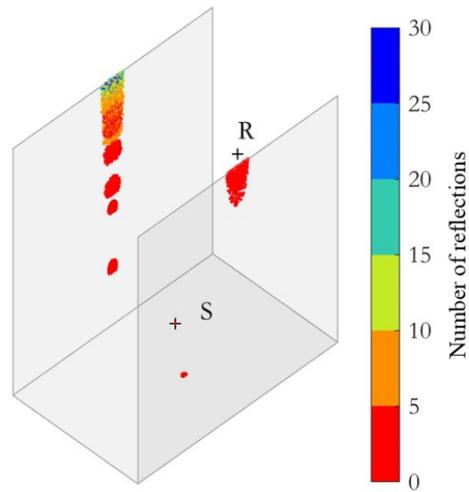


Figura 4.11: Rebotes en la escena 2D, se colorea el último rebote en la propagación de un rayo antes que este alcance al receptor. Extraído de (Bort y Beckers, 2021).

De forma similar a la escena anterior se comienza analizando el impacto de la cantidad de rayos mientras que el coeficiente de absorción se mantiene constante en 0,5.

El resultado del primer experimento se aprecia en la Figura 4.12. Este presenta una forma similar con respecto a la de la medición. Los tiempos similares pero no idénticos para sus tres primeros picos: 23,25 ms, 30,68 ms y 33,37 ms. Nuevamente, con las medidas presentadas en el modelo geométrico 2D (Figura 4.2b), la distancia recorrida por un impacto directo debe de ser ajustada teniendo en cuenta el radio de 1 m de la esfera. Considerando entonces que un impacto directo recorre un mínimo de 7,9 m, podemos realizar un cálculo estimado del tiempo en el que se dará el primer impacto, esto es  $7,9/343 = 23,03$  ms, que es coherente con el resultado mostrado en la gráfica.

El segundo y tercer pico corresponden con caminos de aproximadamente 10,52 y 11,45 m. Mediante otros experimentos realizados, que se exponen en el Anexo B, se verifica que estos picos se conforman en gran parte por caminos de menos de un rebotes. Además, se afirma que estos dos picos se conforman únicamente por caminos de un rebote debido a que resulta imposible encontrar un camino de dos rebotes que tenga un recorrido de 11,45 m o menos.

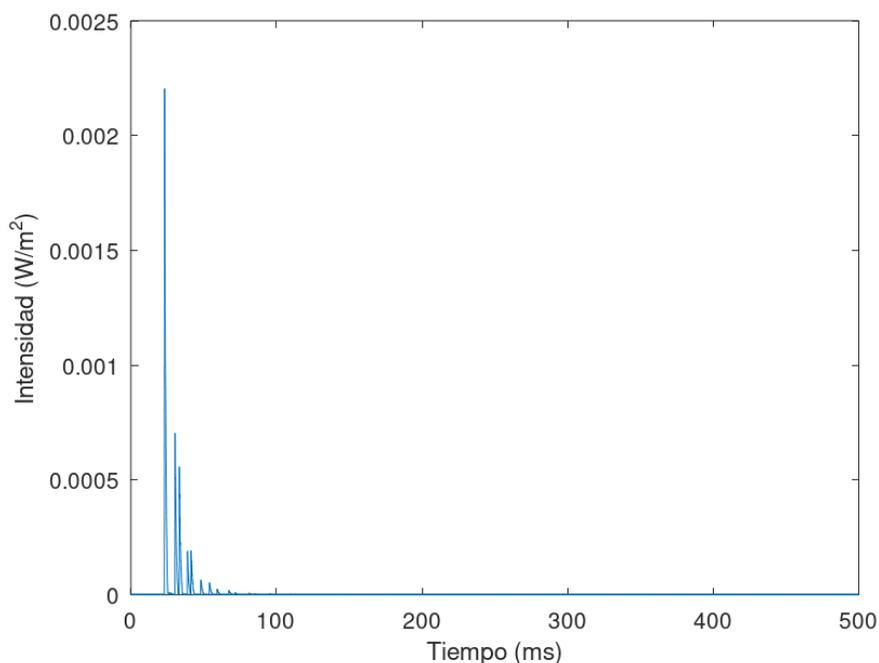


Figura 4.12: IR simulado obtenido con  $10^7$  rayos.

Es evidente que los resultados no son exactos con respecto a la medición realizada, esto se puede explicar por una variedad de razones. Por ejemplo, los resultados pueden ser exactos para el modelo geométrico, pero como fue mencionado en la Sección 4.1.2 el modelo no es exacto con respecto a la realidad. Además se realizaron simplificaciones para la simulación, como por ejemplo utilizar un único índice de absorción, etc. Se presenta en la Figura 4.13 el resultado simulado en trabajos previos. Se destaca una gran semejanza entre ambos resultados.

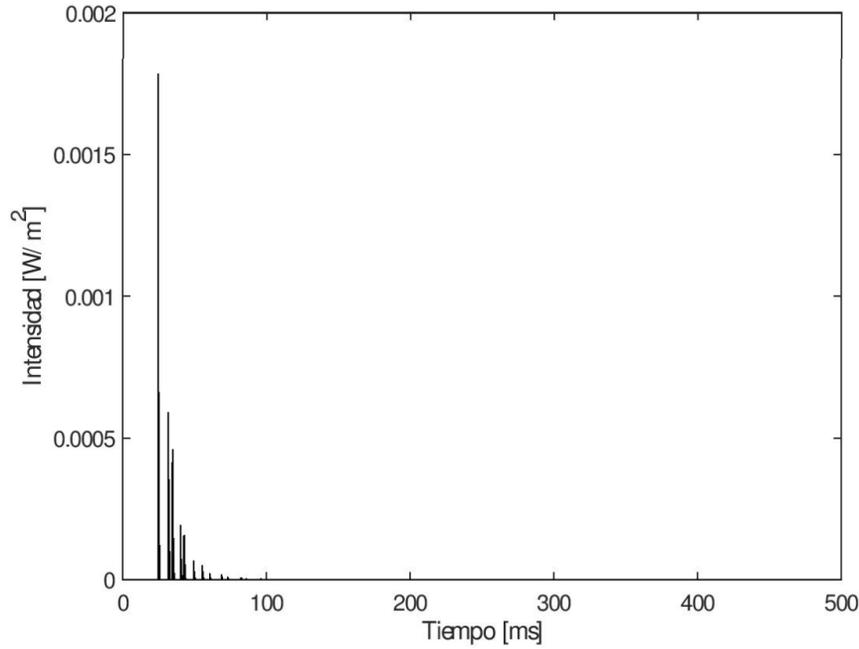


Figura 4.13: IR simulado obtenido con  $10^7$  rayos y parámetros idénticos. Simulación extraída de (Satut, 2022).

Continuando con la experimentación sobre el impacto de la cantidad de rayos sobre la solución, se presenta la Figura 4.14, que contiene el resultado de la simulación al aumentar la cantidad de rayos a  $10^8$ . No se observan grandes diferencias con respecto a la simulación realizada con  $10^7$  rayos, por lo que similarmente a la escena anterior, se ha llegado a una cantidad de rayos donde aumentos no producen mejoras cualitativas en términos de exactitud.

Usando el mismo criterio de la escena anterior, se procede a investigar la relación entre los primeros picos del IR para poder establecer un posible valor para la absorción que brinde resultados más exactos. Nuevamente, a partir de la medición presentada en la Figura 4.10 se puede calcular la relación entre el primer pico y el segundo, donde el primero corresponde con impactos directos y el segundo pico con impactos de caminos con un rebote. Dado que la relación entre el segundo y tercer pico no resulta exacta con respecto a la realidad (donde los picos de los resultados simulados son similares en intensidad mientras que en la medida los valores son más desiguales) no se lo contemplara para estimar la absorción.

La relación entre las intensidades esta dada por:

$$\frac{\text{Valor máximo del primer pico}}{\text{Valor máximo del segundo pico}} = \frac{93,92}{50,52} \approx 1,86$$

A partir de los resultados de la Sección 4.2.1 se realiza la ejecución de 100 simulaciones con  $10^7$  rayos y un coeficiente de absorción en 0 para obtener el valor máximo del segundo pico sin reducir su intensidad mediante la absorción. El valor es de aproximadamente 0,0013, y se utiliza para realizar el siguiente cálculo:

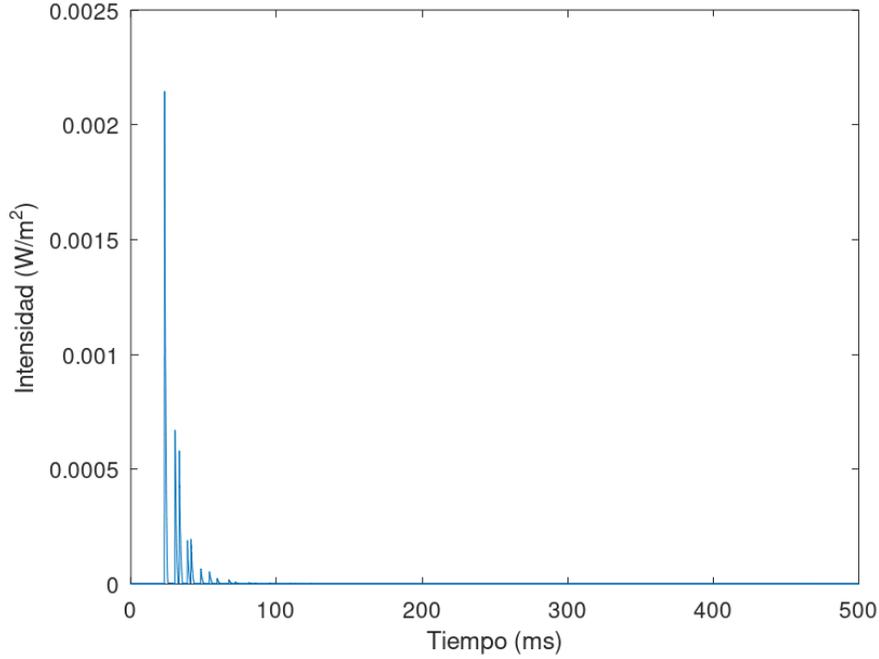


Figura 4.14: IR simulado obtenido con  $10^8$  rayos.

$$\frac{\mu_1}{(1-x)*\mu_2} = \frac{0,002149}{((1-x)*0,0013)} = 1,86 \rightarrow x \approx 0,111$$

Nuevamente,  $x$  representa el coeficiente de absorción  $\mu_1$  es la media para el máximo del primer pico, y  $\mu_2$  es la media para el máximo del segundo pico. Los resultados al usar este valor de coeficiente se presentan en la Figura 4.15. Aunque comparando con la medida original de la Figura 4.10 se aprecia visiblemente que la relación de intensidad entre los dos primeros picos es más exacta, la relación entre los picos siguientes no se asemeja al resultado esperado. Se destaca como la simulación presenta una pérdida más lenta de intensidad, donde los valores posteriores a los tres primeros picos tienen una intensidad considerablemente mayor a la deseada.

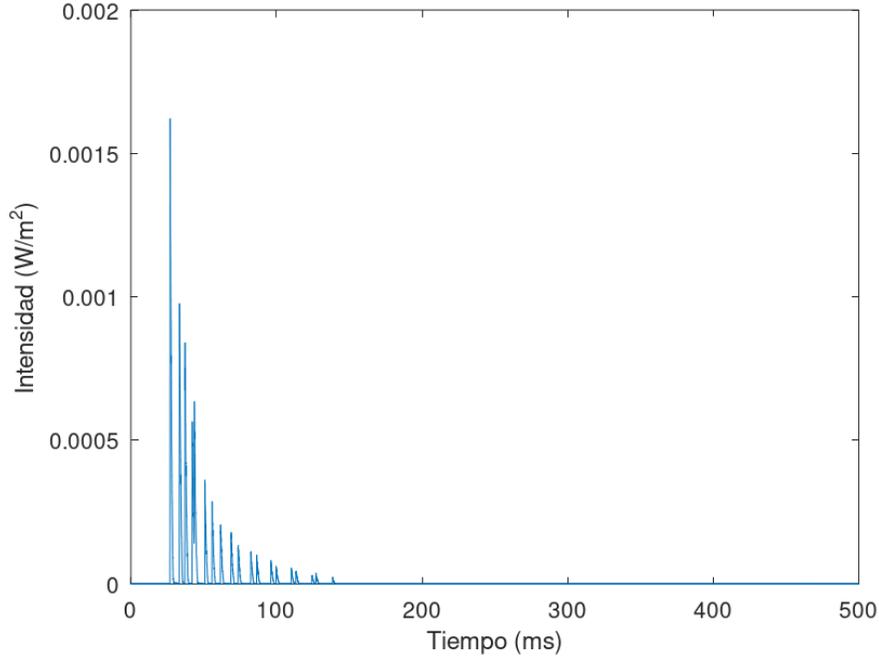


Figura 4.15: IR simulado obtenido con  $10^7$  rayos, absorción de 0,111.

### Escena 3D

Se presenta en la Figura 4.16 el IR generado a partir de la medición real y en la Figura 4.17 la distribución de los últimos rebotes que generaron un impacto. Como puede verse en esta escena, debido a las dos paredes introducidas a la escena la cantidad de caminos aumenta en gran medida. En la medición se observan dos picos iniciales de intensidad similar. Estos picos aparecen alrededor de los 30,00 ms (correspondientes a impactos directos que recorren una distancia aproximada de 10,3 m) y 36,64 ms. Posteriormente, se identifican varios picos adicionales, siendo los dos siguientes a los 38,95 ms y 41,68 ms. Sin embargo, debido a la ausencia de una medida cuantitativa para identificar los picos de intensidad, la complejidad de la gráfica dificulta distinguir más picos después de estos.

Nuevamente se comienza a experimentar con  $10^7$  rayos, correspondiente a la menor cantidad de rayos que produce resultados precisos (ver Sección 4.2.1). El coeficiente de absorción se mantiene en 0,5 al igual que al inicio de la experimentación en otras escenas. El resultado del primer experimento se aprecia en la Figura 4.18.

El tiempo para el primer pico es de 27,0 ms, presentando una diferencia de 3 ms con respecto al tiempo original. Esta medida corresponde a un camino directo de aproximadamente 9,3 m, lo que es acorde a lo esperado según las dimensiones de la escena virtual y el tamaño del receptor. El siguiente pico se produce en el instante 33,43 ms y también presenta una diferencia temporal cercana a 3 ms. Los dos siguientes se producen en los tiempos 34,93 ms y 37,31 ms, que corresponden con una diferencia temporal mayor.

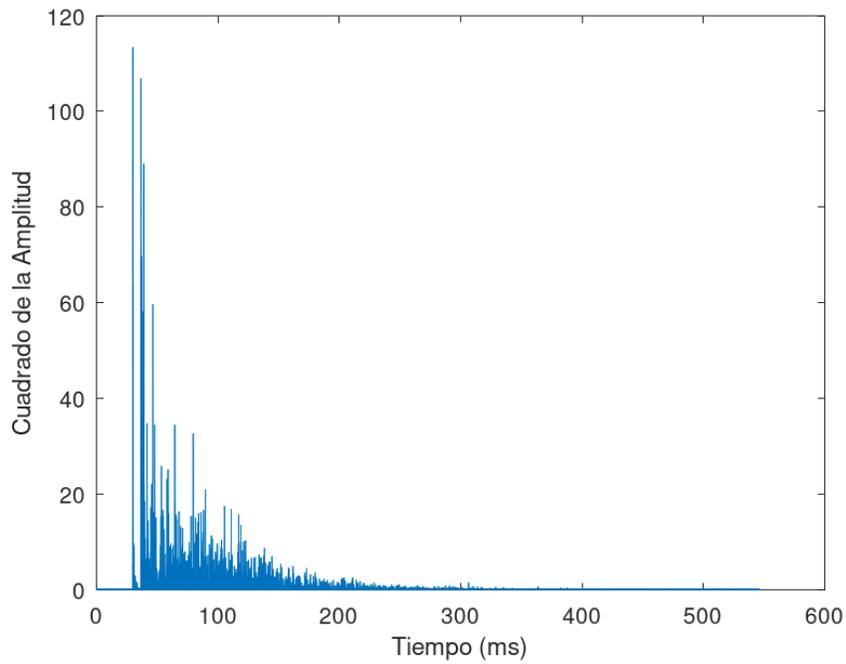


Figura 4.16: Respuesta al Impulso medida en la escena 3D. Generado a partir de los resultados de (Bort y Beckers, 2021). Se presentan únicamente los primeros *samples* de este IR para apreciar el detalle.

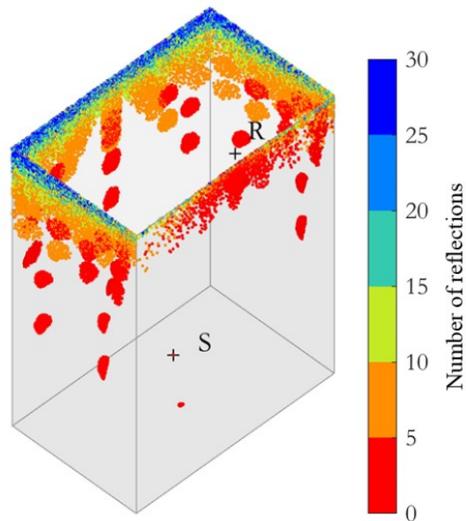


Figura 4.17: Rebotes en la escena 3D, se colorea el último rebote en la propagación de un rayo antes que este alcance al oyente. Extraído de (Bort y Beckers, 2021).

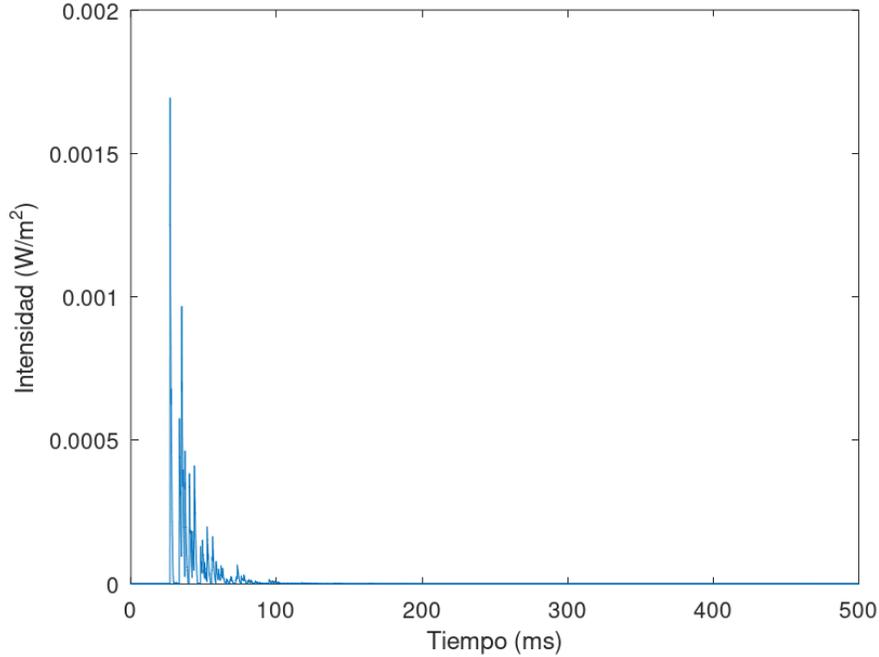


Figura 4.18: IR simulado obtenido con  $10^7$  rayos.

También se observan diferencias en cuanto a la relación de intensidad entre los distintos picos de intensidad, se puede apreciar en la medida original que los tres primeros picos son decrecientes en intensidad mientras que en el resultado el tercer pico es mayor al segundo.

Se procede a aumentar la cantidad de rayos, el resultado se aprecia en la Figura 4.19.

Los tiempos para los picos no presentan cambios para este experimento, son consistentes con los del anterior. Similarmente a simulaciones previas en otras escenas, aumentar la cantidad de rayos pasado el punto de  $10^7$  no produce diferencias apreciables sobre el IR resultante.

Aunque los resultados son precisos en si mismos, se encuentra que presentan grandes diferencias y por lo tanto no se corresponden con la realidad. Esto parece deberse a las diferencias entre la escena real y la modelada (ver Sección 4.1.2). Principalmente se observan diferencias en la rápida caída de la intensidad luego del primer pico, defecto que puede ser alterado al modificar la absorción utilizada.

Analizando los resultados de (Satut, 2022), se presenta que para esta escena una absorción de 0,1 resulta en una simulación cualitativamente similar a lo deseado. En la Figura 4.20 se muestra el resultado obtenido con este parámetro, aunque es destacable como este experimento produce una mayor reverberación, se destaca en gran medida como este valor de absorción produce un resultado que tampoco es exacto en comparación a la medida original. Todo esto se destaca fundamentalmente en el valor

de intensidad para el tercer pico, que resulta mayor que el primero. En la Figura 4.21 se ve el resultado obtenido previamente, que claramente presenta el mismo fenómeno.

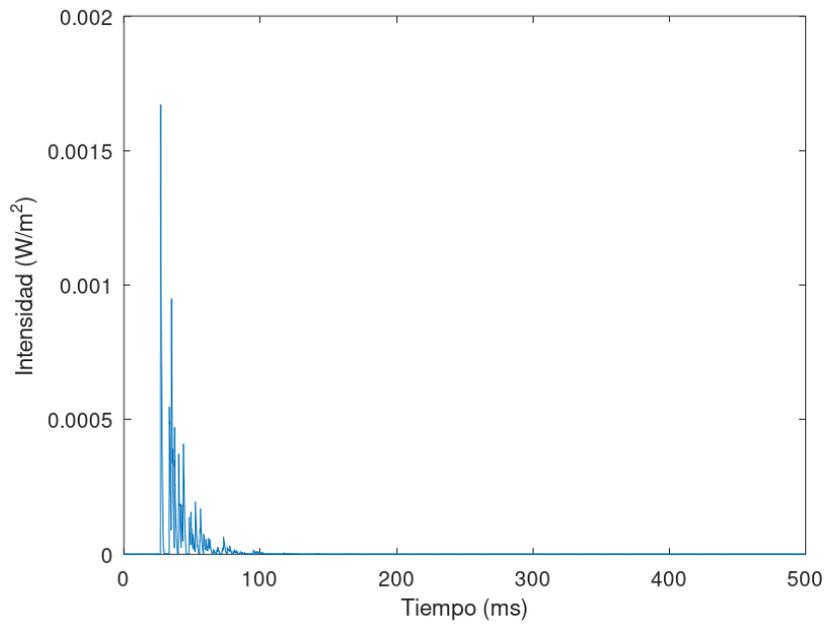


Figura 4.19: IR simulado obtenido con  $10^8$  rayos.

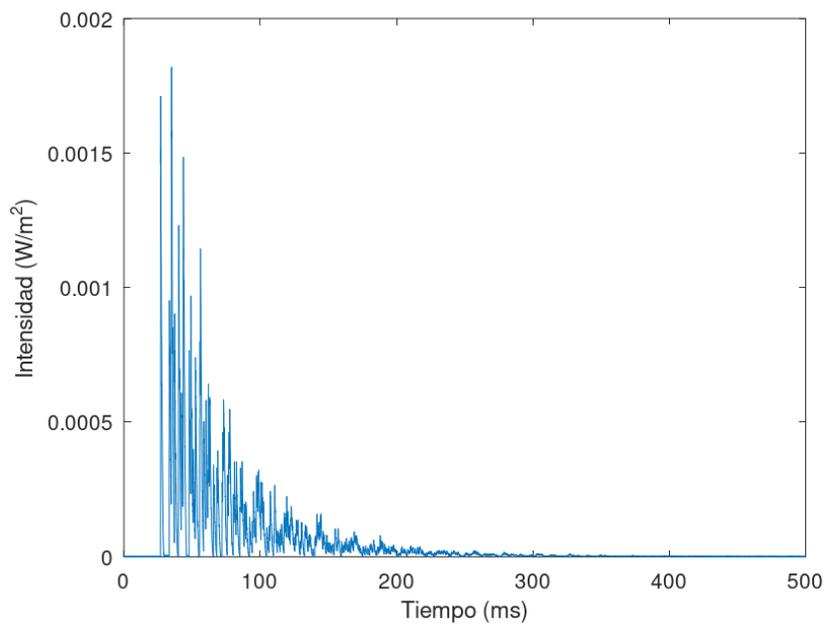


Figura 4.20: IR simulado obtenido con  $10^7$  rayos y absorción 0,1.

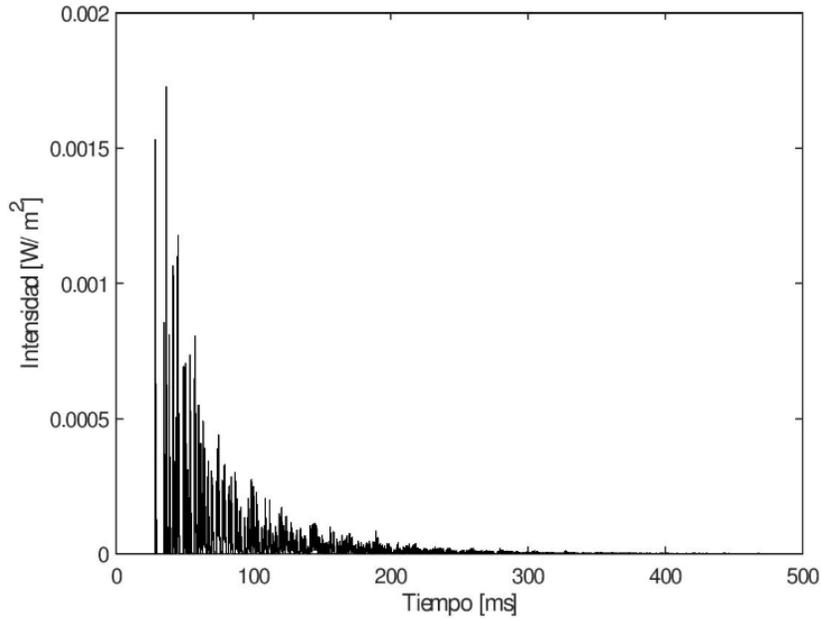


Figura 4.21: IR simulado obtenido con  $10^7$  rayos, absorción 0,1 y límite de 30 rebotes. Simulación extraída de (Satut, 2022).

Determinar un valor adecuado para la absorción resulta un problema más complejo en comparación a escenas anteriores. Se decidió ignorar el segundo pico (el menor de los tres primeros) y utilizar el tercero para el cálculo, de no ser así se especula que el resto del IR incrementaría demasiado. También se puede apreciar en el resultado de la Figura B.3 del Anexo B que los primeros picos se ven afectados por caminos de uno y dos rebotes, por lo que se realizaron pruebas adicionales para averiguar la intensidad aportada al tercer pico por los caminos de distintos rebotes. La ecuación utilizada fue la siguiente.

$$\frac{\mu_1}{(1-x)*\alpha_3+(1-x)^2*\beta_3} = 1,061 \rightarrow x \approx 0,221$$

Donde

- $\mu_1 = 0,001711$  es el máximo del primer pico
- $x$  es el coeficiente de absorción a utilizar.
- 1,061 es la relación entre el valor máximo del primer y tercer pico.
- $\alpha_3 = 0,001785$  es la intensidad aportada al tercer pico por los caminos de un rebote, utilizando absorción 0.
- $\beta_3 = 0,000338$  es la intensidad aportada al tercer pico por los caminos de dos rebotes, utilizando absorción 0.

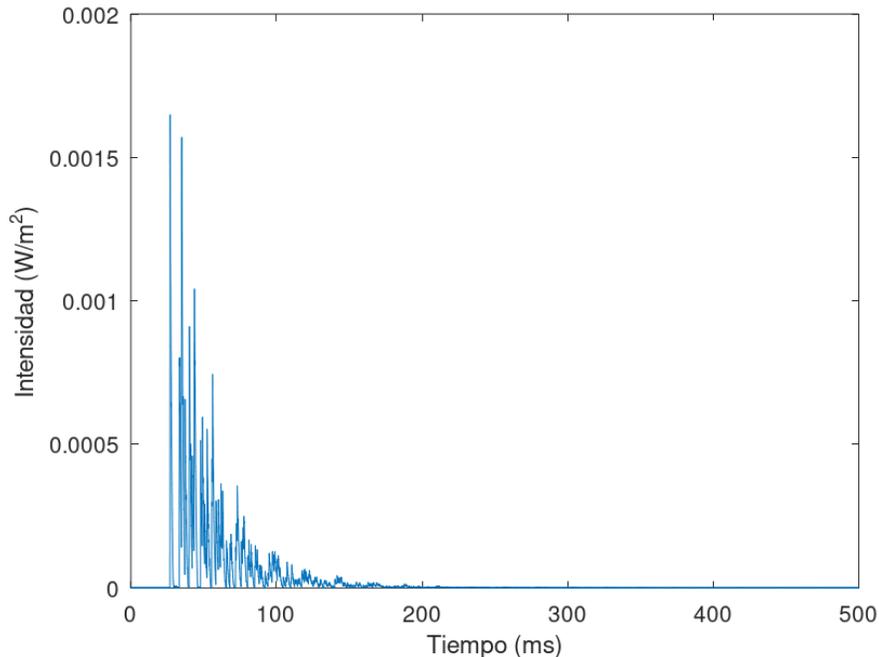


Figura 4.22: IR simulado obtenido con  $10^7$  rayos y absorción 0,221.

Se presenta la simulación de la Figura 4.22. Si el único criterio a utilizar es la relación de intensidad entre los primeros tres picos, el coeficiente de absorción adoptado en esta prueba resulta en el IR más exacto hasta el momento. Pero, si se toma un criterio más general, resulta importante considerar que la reverberación ha disminuido mucho respecto a la medición (4.16), ya que se observan picos pasados los 300 ms, cosa que no se aprecia en la Figura 4.22. A su vez, observando la Figura 4.20, se observan picos del mismo orden de magnitud a los de la Figura 4.16 en los 300 ms. Por esta razón, se determina que la absorción de 0,1 refleja un poco mejor la realidad.

En el caso de la escena 3D es posible determinar que debido a factores externos a las capacidades de la simulación (como por ejemplo, la escena virtual utilizada simplifica demasiado la realidad), no es posible brindar simulaciones exactas en todos los aspectos del IR. Por lo tanto, se encuentra con un caso donde es necesario tomar una decisión sobre cuáles aspectos del IR se valora más, la exactitud en los primeros impactos o en la reverberación.

### 4.2.3. Pruebas de rendimiento

A continuación se presentan los resultados de las pruebas de rendimiento realizadas. Las mismas se ejecutaron en cada uno de los ambientes de prueba registrados en la Sección 4.1.3. Estas pruebas se enfocan en la medición de tiempos de ejecución de los procesos más significativos del sistema, el *path tracing* y la convolución. Cada medición

se realizó 5 veces y se promediaron los resultados para generar las siguientes tablas. En cuanto a los parámetros utilizados, la cantidad de rayos fue la principal variable a estudiar y se fue modificando a lo largo de cada prueba. Por otro lado, se fijó en 100 la cantidad máxima de rebotes, mientras que el factor de absorción resulta irrelevante para el rendimiento.

En la tabla 4.5 se aprecian los tiempos de cada sistema para realizar el *path tracing* de un IR de dos segundos con variada cantidad de rayos dentro de la escena 3D.

Cantidad de rayos para <i>path tracing</i>	Tiempos para ambiente de prueba 1 (ms)	Tiempos para ambiente de prueba 2 (ms)	Tiempos para ambiente de prueba 3 (ms)
$10^3$	0,393	7,319	8,212
$10^6$	27,047	352,088	32,890
$10^7$	263,307	3250,910	248,610
$10^8$	2416,302	32047,840	2189,650
$10^9$	24042,140	339115,600	23181,740

Tabla 4.5: Resultados de los tiempos de ejecución para el trazado de caminos en la escena 3D.

Como puede apreciarse, el ambiente 2, que tiene la única GPU que no posee la arquitectura RTX, es el más lento en cuanto al trazado de rayos. Otra observación es que entre los experimentos con  $10^6$  y  $10^9$  rayos, los tiempos parecen aumentar linealmente, por un factor de 10 en cada ocasión, coincidiendo con el aumento en cantidad de rayos.

En la tabla 4.6 se presentan los tiempos de ejecución para una cantidad de rayos fija a lo largo de distintas escenas. Como puede apreciarse, el aumento en la complejidad de la escena causa más rebotes en el trazado de caminos, lo que impacta directamente en los tiempos de ejecución. Se puede observar como los tiempos, incluso para el ambiente más lento, son mejores que los reportados por Satut (Satut, 2022).

En la tabla 4.7 se miden los tiempos de ejecución para la función que realiza la convolución de un IR de dos segundos con un archivo de audio de 8 segundos, a 16 kHz. Además, se registra el tiempo total del proceso de convolución, compuesto por la función mencionada junto con operaciones de inicialización, transferencias de memoria y otros procesos auxiliares.

### 4.3. Simulaciones en otros escenarios

Hasta ahora solo se ha experimentado con las escenas sobre las que se tienen IRs de referencia, sin embargo, aunque no se posea una forma cuantitativa de verificar los

Escena	Tiempos para ambiente de prueba 1 (ms)	Tiempos para ambiente de prueba 2 (ms)	Tiempos para ambiente de prueba 3 (ms)	Tiempos reportados en (Satut, 2022) (ms)
1D	173,813	1314,350	156,778	3107
2D	160,513	1488,940	195,259	5227
3D	263,307	3250,910	248,610	12347

Tabla 4.6: Resultados de los tiempos de ejecución para el trazado de caminos en diversas escenas con  $10^7$  rayos.

	Ambiente de prueba 1 (ms)	Ambiente de prueba 2 (ms)	Ambiente de prueba 3 (ms)
<b>Función de convolución (ms)</b>	4,249	93,477	56,781
<b>Proceso completo de convolución (ms)</b>	6,389	98,558	62,286

Tabla 4.7: Resultados de los tiempos de ejecución para la convolución.

resultados también es de interés observar los resultados de la solución en escenas más complejas.

### 4.3.1. Conference Room

Se trabajó con la escena “*Conference Room*” tomada de (McGuire, 2017), la misma modela una sala de conferencias cerrada y amueblada, se modificó la escena eliminando las puertas para crear una abertura por la que rayos puedan escapar. La Figura 4.23 ilustra la escena.

Se le asignó un material a cada elemento de la escena para luego asignarle una correspondiente absorción, los datos exactos se ven en la Tabla 4.8. Las absorciones son correspondientes a la absorción del material real frente a sonidos de 1 KHz de frecuencia, extraídos de (Everest, 2001). Para aquellos materiales que no se encontró una absorción, se utiliza un valor de 0,1.

Los resultados se aprecian en la Figura 4.24. Como puede apreciarse, la forma del IR es similar a los anteriores, con una caída rápida en los primeros 100 ms. El IR es repentino dada la proximidad entre el emisor y el receptor (5,4 m) y una vez recibido el primer impacto, se nota una gran densidad de impactos en los *samples* posteriores (en los siguientes 2 segundos posteriores al primer impacto, solo 62 de los 32000 *samples*

Elemento de la escena	Material elegido	Absorción correspondiente
Paredes	Yeso	0,03
Estructura de madera	Madera contrachapada	0,09
Pizarrones	Madera contrachapada	0,09
Cortinas	Algodón	0,57
Suelo	Alfombra sobre cemento	0,37
Techo	Cemento	0,02

Tabla 4.8

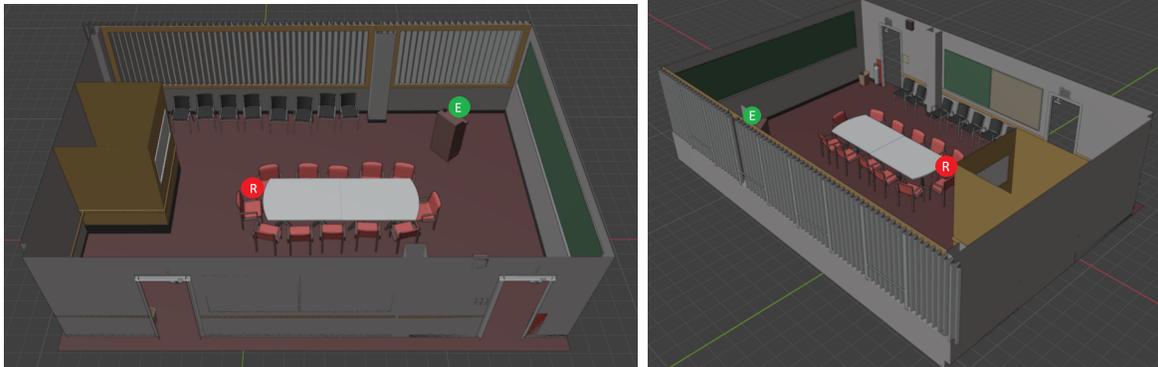


Figura 4.23: Capturas de la escena tridimensional con las posiciones para el emisor y receptor, el techo de la escena se removió temporalmente para tomar estas imágenes.

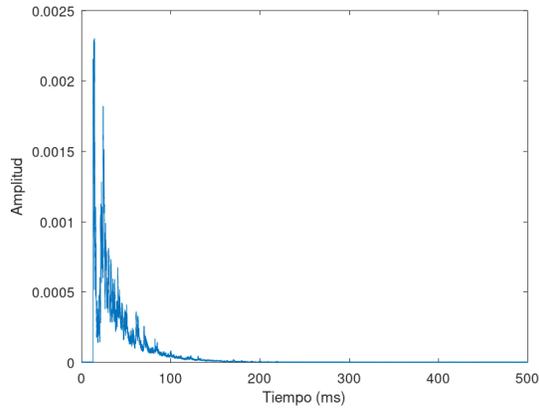
tienen intensidad 0).

El resultado es un IR mucho más detallado que los de las escenas anteriores, sin embargo al carecer de IRs de referencia no se puede profundizar en gran forma en los resultados.

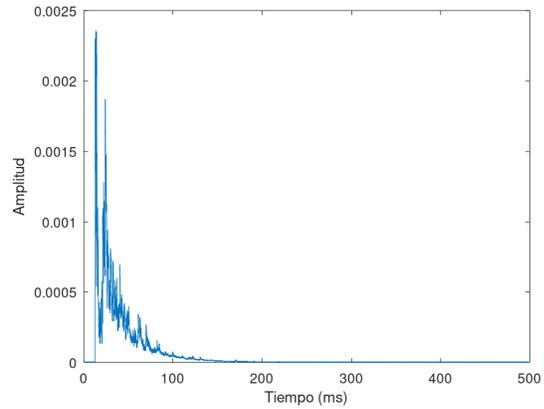
### 4.3.2. Caja

Para investigar los resultados sobre un caso extremo de reverberación se trabajó en la escena denominada “Caja”, la misma es una caja cúbica de 50 m de lado. El emisor y el receptor se posicionan en el interior de la caja para generar un ambiente donde el sonido pueda propagarse largas distancias numerosas veces para luego impactar con el receptor, y de esta forma generar una gran cantidad de impactos tardíos y un gran efecto de reverberación. Para ayudar con este fin, se utiliza un valor bajo de absorción (0,1).

Dado que estamos en una escena donde un rayo puede tener un muy largo trayecto antes de impactar al receptor, puede suceder que dicho impacto ocurra fuera de la ventana de un IR elegido, por ejemplo, si se elige un IR de 2 segundos, un rayo podría recorrer como máximo 686 m. Por esta razón se experimenta también con variar el largo del mismo.



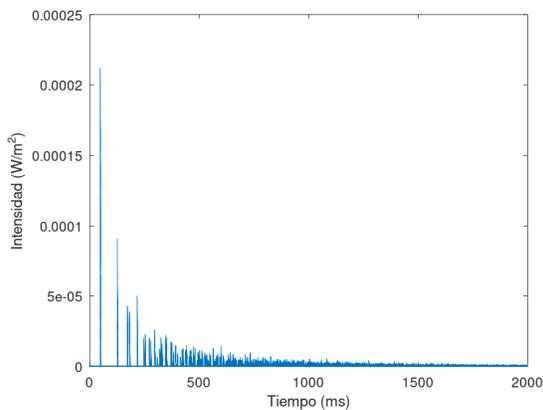
(a) IR simulado.  $10^7$  rayos.



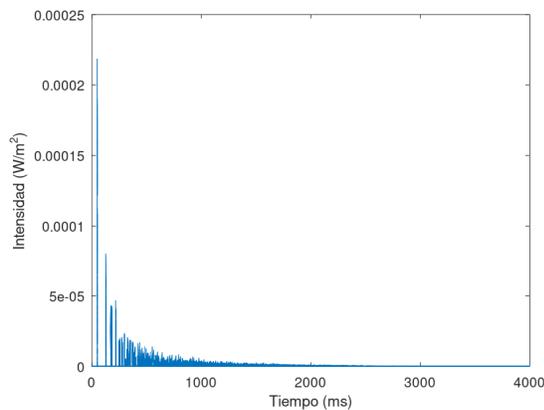
(b) IR simulado.  $10^8$  rayos.

Figura 4.24: IRs para escena Conference Room, límite de 30 rebotes.

Se posicionó al emisor cerca del centro de la caja y al receptor a 17,3 m del mismo. Los resultados de los primeros experimentos se pueden ver en la Figura 4.25. Se puede observar que los picos no son tan intensos en comparación a experimentos de escenas anteriores. Esto indica una menor concentración de impactos de rayos en un solo instante de tiempo dado, y esto es coherente con el incremento de distancia entre el emisor y el receptor.



(a) IR simulado. IR de 2 segundos.



(b) IR simulado. IR de 4 segundos.

Figura 4.25: IRs para la escena Caja,  $10^7$  rayos, absorción de 0,1, sin límite de rebotes.

Ambos IRs presentan un IR más extenso en el tiempo en comparación a experimentos anteriores, donde solamente 500 ms de *samples* eran suficientes para capturar los impactos. Debido a esto se considera que la reverberación especulada está presente en los resultados.

Para el experimento de la Figura 4.25b, existen valores para el IR mayores a 0

incluso en el *sample* final, como se aprecia en la Figura 4.26, indicando que aún se podría extender más este IR si se deseara registrar más impactos.

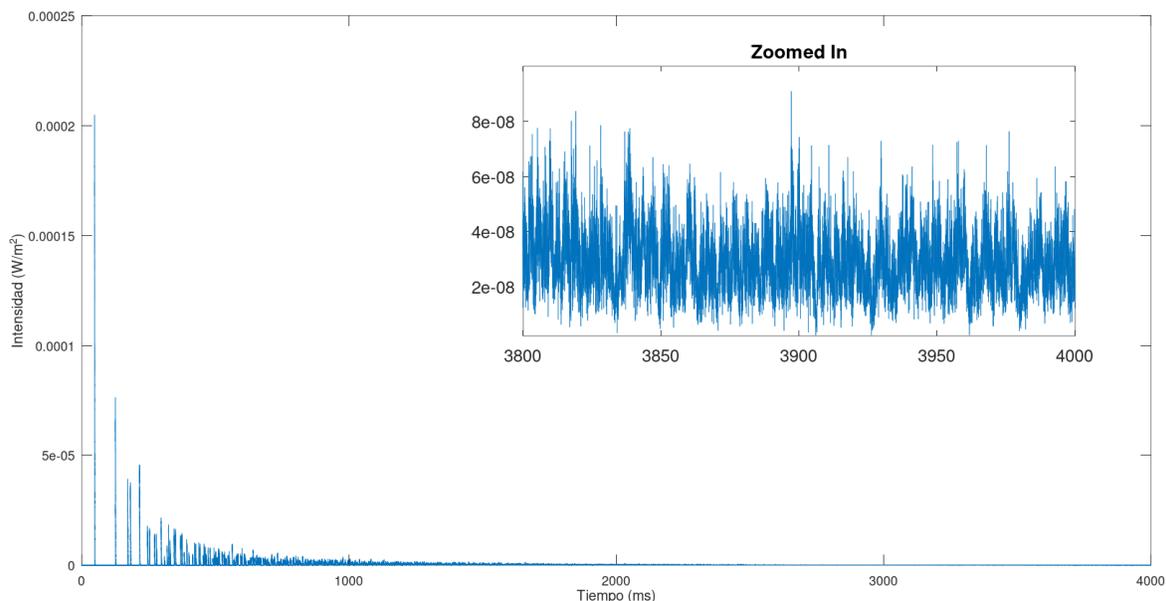


Figura 4.26: IR de la Figura 4.25b, se agrega un zoom sobre los últimos 200 ms para apreciar la existencia de datos hasta el último momento del IR.

## 4.4. Análisis de la auralización

En esta sección se presentan distintos ejemplos de resultados de convoluciones en función de distintas configuraciones que la solución permite, como por ejemplo, las modalidades de entrada (audio pregrabado, audio en vivo) y las modalidades de salida (mono, estéreo).

### 4.4.1. Convolución de audio pregrabado

Para este tipo de entrada se eligió utilizar un audio sencillo para poder notar fácilmente los cambios generados por la convolución de forma visual. Se utilizó el sonido de una claqueta<sup>1</sup>, la duración del audio es de 2,5 segundos a un *sample rate* de 48 KHz. Se puede visualizar la forma del audio en la Figura 4.27 donde se caracteriza por un fuerte ruido repentino y una rápida caída en la amplitud.

---

<sup>1</sup>Utensilio compuesto de dos planchas de madera, unidas por una bisagra, que se hacen chocar ante la cámara para sincronizar sonido e imagen.

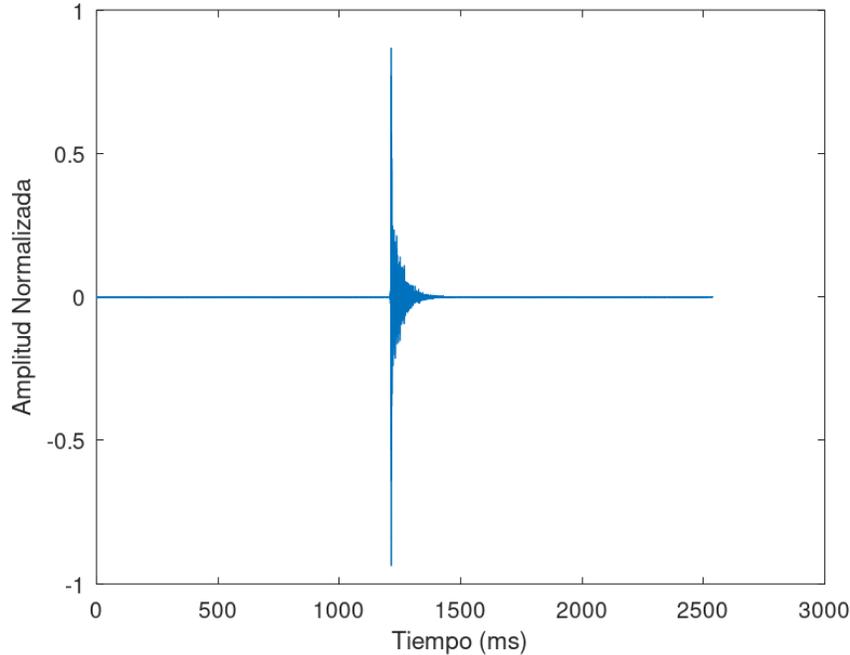


Figura 4.27: Forma de onda para el audio de la claqueta utilizado.

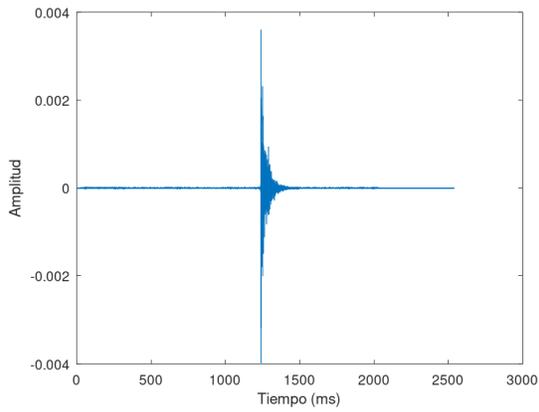
Se procede a convolucionar este audio con cada escena mencionada hasta ahora. Para la escena 1D en la Figura 4.28a, no se ven grandes cambios con respecto a la onda original aunque sí se aprecia la existencia de una leve reverberación, identificable por el “ruido” del audio entre el segundo 1,5 y 2. Para la escena 2D en la Figura 4.28b el resultado es similar.

Para la escena 3D, el resultado de la convolución se puede ver en la Figura 4.29a. Se aprecia como el audio comienza a extenderse, ya que la caída no es tan veloz. El resultado de la convolución en la escena “Conference Room” se puede ver en la Figura 4.29b, la forma resulta bastante similar, se destacan diferencias en los valores de la amplitud.

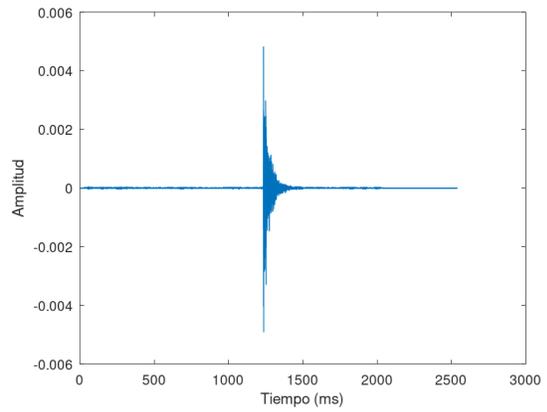
Finalmente, se puede ver la convolución del audio con la escena “Caja” en la Figura 4.30, donde la reverberación es mucho más evidente. También se aprecian la amplitud es mejor a todas las convoluciones hasta ahora, factor atribuido a la poca intensidad del IR de esta escena.

#### 4.4.2. Convolución estéreo de audio pregrabado

Para poder observar mejor la funcionalidad del modo estero se creó una escena con forma de caja a la cual le faltan 2 caras como se muestra en las Figuras 4.31a, 4.31b, y 4.31c, de forma que existieran varios caminos de los rayos desde el emisor al receptor. El receptor se perfiló de manera que un hemisferio estuviera más expuesto a los rayos que el otro.

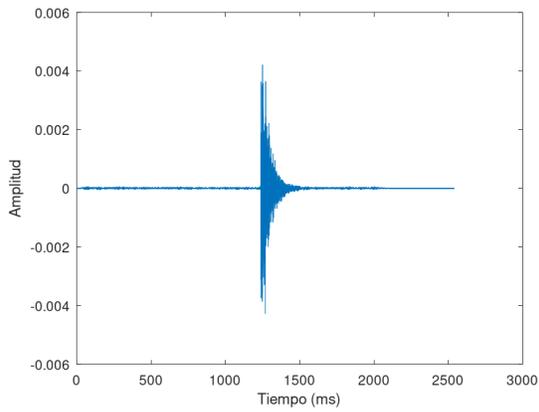


(a) Convolución del audio en la escena 1D. Absorción de  $0,6 \cdot 10^7$  rayos, 3 rebotes máximos.

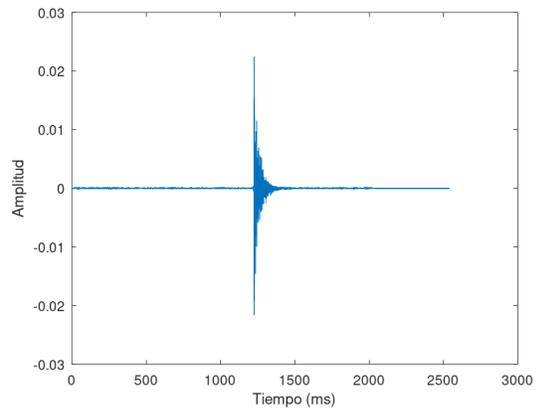


(b) Convolución del audio en la escena 2D. Absorción de  $0,111 \cdot 10^7$  rayos, 30 rebotes máximos.

Figura 4.28



(a) Convolución en la escena 3D. Absorción de  $0,22 \cdot 10^7$  rayos, 30 rebotes máximos.



(b) Convolución en la escena "Conference Room". Parámetros idénticos a los usados en ese experimento.

Figura 4.29

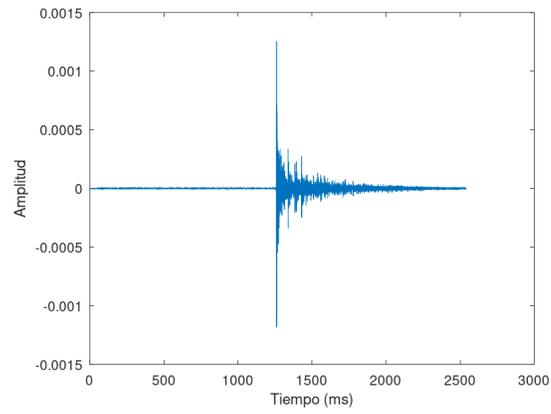
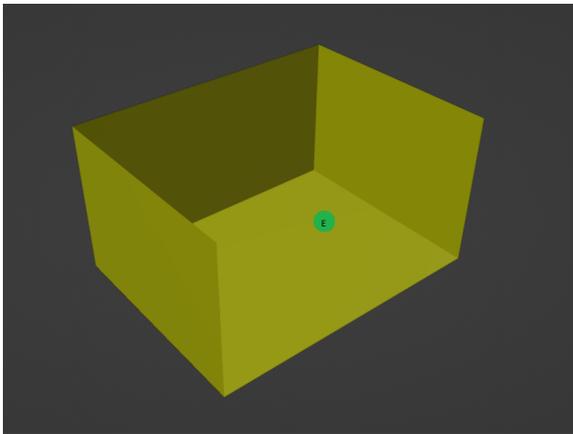
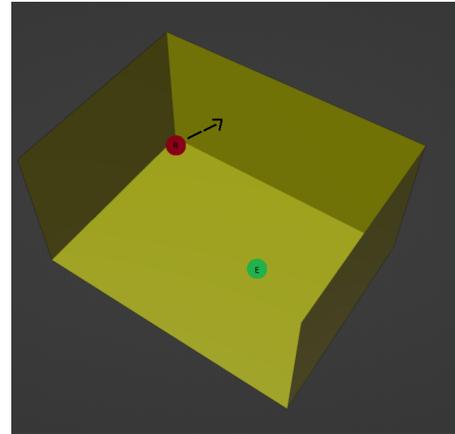


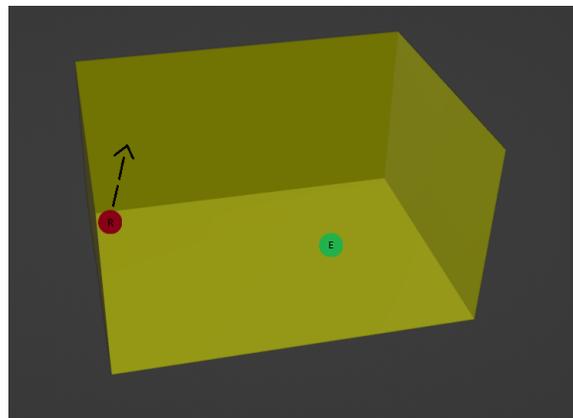
Figura 4.30: Convolución en la escena “Caja”. Parámetros idénticos a los usados en ese experimento.



(a) Perspectiva 1.



(b) Perspectiva 2.



(c) Perspectiva 3.

Figura 4.31: Captura de la escena utilizada, la flecha ilustra la orientación del receptor.

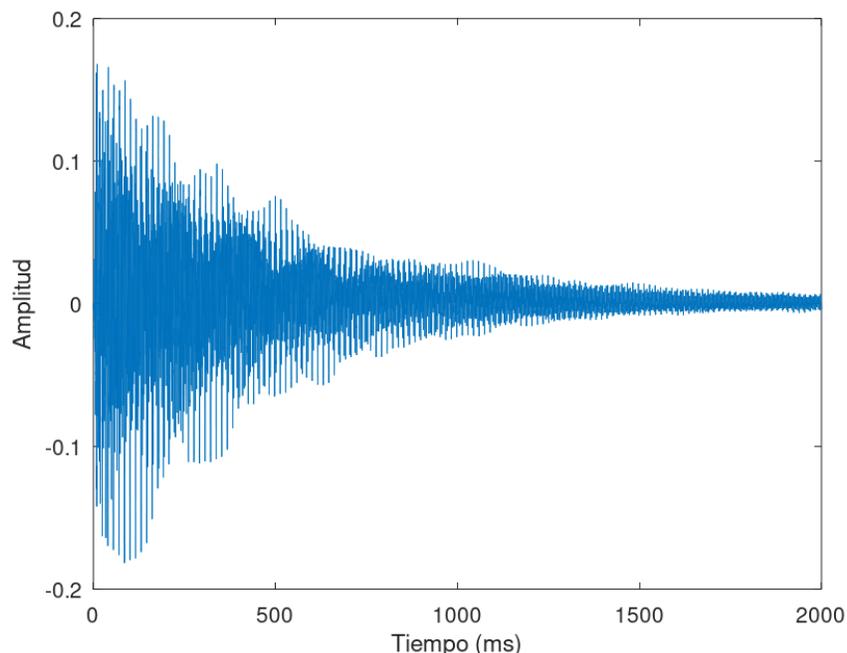


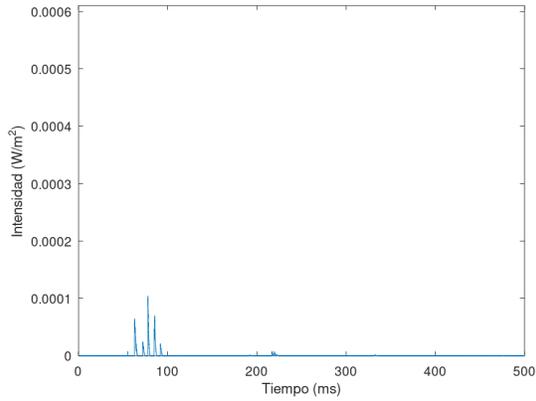
Figura 4.32: Audio utilizado para la convolución, grabación de una nota de piano.

En la Figura 4.32 se puede ver representado el audio sin ninguna alteración que se utilizó para este experimento, de esta forma se podrá observar con mayor facilidad como se ve afectado por cada uno de los IRs y como se puede ver una diferencia en ambos resultados.

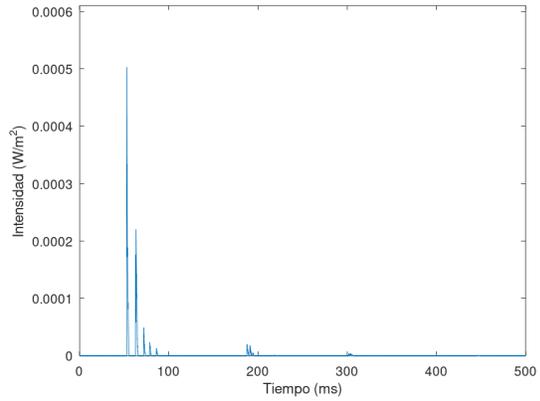
En la Figura 4.33a se puede ver el IR resultado del hemisferio izquierdo del receptor, que se encuentra enfocado hacia una pared del lado contrario al emisor. Por otro lado, en la Figura 4.33b se aprecia el IR resultado del hemisferio derecho del receptor, que estaba directamente enfocado hacia el emisor. Se puede identificar también la presencia de la HRTF ya que los impactos directos recibidos por el hemisferio derecho en el milisegundo 44,28 se reflejan en el IR del hemisferio izquierdo en la forma de un sutil pico (el primero) en el milisegundo 44,29.

En las Figuras 4.34a y 4.34b se pueden ver los resultados de realizar las convoluciones del audio con los respectivos IRs. En la convolución del lado izquierdo se aprecia una gran disminución en la intensidad del sonido, a diferencia de la convolución del lado derecho, que posee una mayor amplitud a lo largo de todo el audio.

Por este motivo se nota una diferencia auditiva considerable al momento de escuchar el audio generado, dado que se distingue claramente que el origen del sonido se encuentra del lado derecho.

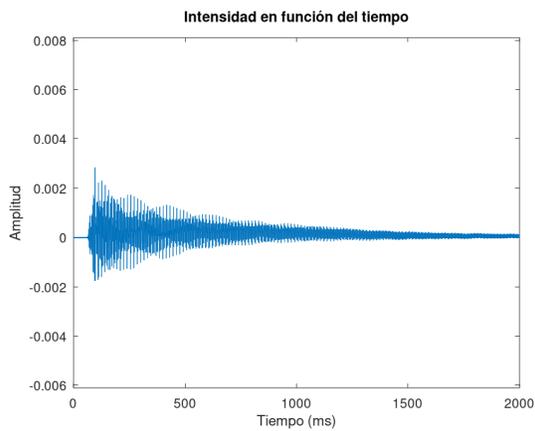


(a) IR izquierdo obtenido en la escena.

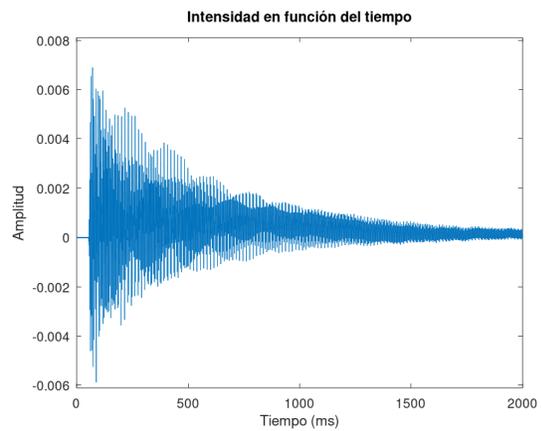


(b) IR derecho obtenido en la escena.

Figura 4.33: IRs generados con  $10^7$  rayos, absorción de 0,5 y límite de 100 rebotes máximos.



(a) Convolución izquierda.



(b) Convolución derecha.

Figura 4.34: Convoluciones resultantes del experimento de la Sección 4.4.2.

### 4.4.3. Entrada por micrófono

Al igual que (Satut, 2022), se perciben los efectos de la convolución claramente al usar el micrófono. Una vez carga la escena virtual (mientras se realiza el trazado de rayos) no se notan grandes *delays* en la entrada, que es esperable dado que los tiempos de convolución son menores a una décima de segundo para todos los sistemas (ver Sección 4.2.3).

Puede apreciarse también cómo muchos sonidos ambientales son percibidos por el micrófono y auralizados. Normalmente, esta clase de sonidos son filtrados por aplicaciones que utilizan esta entrada de sonido; sin embargo, no es el caso de esta implementación. Esto causa que el emisor de sonido sea una fuente tanto de la voz emitida, como de los sonidos ambientales percibidos, lo que de cierto modo disminuye la inmersión causada por la direccionalidad del audio, ya que en una escena real, distintos tipos de sonidos tendrían diversos orígenes. Además, a medida que se disminuye el coeficiente de absorción, puede percibirse un ruido de fondo con mayor intensidad que opaca el sonido de interés. Esto es de esperarse y puede mitigarse usando un micrófono mejor calibrado de forma de no detectar el sonido del ambiente.

### 4.4.4. Auralización en tiempo real

Para los sistemas 1 y 3 (ver Sección 4.2.3), al utilizar  $10^7$  rayos la auralización es ininterrumpida al recorrer la escena. Para el sistema 2 resulta necesario utilizar una cantidad menor de rayos para obtener una reproducción con transiciones suaves entre convoluciones, en este caso  $10^6$ .

En cuanto a los efectos de la auralización, al moverse por una escena abierta como la 1D o 2D uno puede notar como disminuye el volumen del sonido mientras se aleja del emisor, el fenómeno opuesto ocurre al acercarse. También se puede notar silencio cuando el emisor resulta bloqueado por una pared, y cuando está parcialmente obstruido, por ejemplo ocultando su mitad izquierda, se puede apreciar como se lo escucha por el canal derecho. De forma similar, al rotar la cámara el efecto estéreo está fuertemente presente.

Se puede identificar fácilmente el momento que se realiza una nueva convolución y la salida de sonido cambia, especialmente al desplazarse por la escena cerca del emisor. Este efecto disminuye al alejarse del emisor.



# Capítulo 5

## Conclusiones y Trabajo Futuro

En este capítulo se sintetizan los resultados obtenidos a lo largo del proyecto.

### 5.1. Conclusiones

El presente trabajo ha permitido alcanzar los objetivos originalmente planteados, desarrollando un renderizador acústico con capacidad de auralización en tiempo real en escenas virtuales tridimensionales. Los experimentos realizados han demostrado que, bajo una configuración apropiada, el sistema es preciso y capaz de representar la realidad de manera exacta, logrando una simulación acústica precisa y eficiente.

A lo largo del desarrollo del proyecto, se tomaron decisiones estratégicas respecto a qué aspectos de la propagación del sonido considerar. Se decidió no incluir fenómenos como la difracción y la transmisión, enfocándose exclusivamente en la reflexión del sonido. A pesar de esta omisión, no se ha percibido un impacto negativo significativo en la calidad de la simulación, lo que valida la decisión de simplificar el modelo para optimizar la implementación y el rendimiento.

Comparando nuestro trabajo con el producto resultante de (Satut, 2022), se observa una evolución gracias a la cantidad de funcionalidades añadidas y las mejoras de rendimiento registradas. Los experimentos indican que, aunque los resultados son similares, las diferencias de tiempo entre los distintos sistemas son sustanciales, destacándose la superioridad de la GPU para este tipo de procesamiento.

En cuanto a los parámetros de configuración, se ha determinado que el uso de  $10^7$  rayos es ideal para las escenas estudiadas. Auditivamente, no se percibe una diferencia notoria al aumentar este número, pero sí se nota una degradación de la calidad al disminuirlo.

En términos de rendimiento, el trazado de rayos es el principal factor a considerar. Por lo tanto, dadas las particularidades del sistema que se esté utilizando, resulta esencial evaluar los valores para los parámetros de trazado de rayos, para encontrar un balance óptimo entre la calidad de la simulación y el tiempo de cálculo requerido.

Finalmente, resulta esencial no simplificar demasiado la realidad para obtener si-

mulaciones de alta exactitud. Los principales factores a considerar son la fidelidad con respecto a la realidad del modelo virtual donde se esté trabajando, y además evitar utilizar un factor de absorción global en lo posible, tomándose el trabajo de etiquetar los materiales adecuadamente con sus correspondientes absorciones.

En el contexto metodológico, el desarrollo del sistema se llevó a cabo de manera incremental y modular, construyendo sobre una base sólida y ampliando gradualmente las funcionalidades. Inicialmente, se implementó un receptor único y luego se avanzó hacia configuraciones estéreo. Se comenzó con la convolución de audio pregrabado y posteriormente se añadió la capacidad de procesar audio en vivo. Además, la implementación inicial no incluía renderización en tiempo real, que fue incorporada en etapas posteriores. Esta metodología de desarrollo progresivo permitió asegurar la estabilidad y funcionalidad del sistema en cada etapa antes de proceder con adiciones más complejas.

En conclusión, se ha cumplido con los objetivos propuestos. Además, también ha demostrado ser una plataforma capaz para la auralización en tiempo real. Las decisiones tomadas durante el desarrollo han sido validadas por los resultados obtenidos, y el resultado proporciona una base sólida para futuros desarrollos y mejoras en la simulación acústica.

## 5.2. Trabajo Futuro

A continuación, se listan posibles mejoras a realizar con el propósito de continuar la implementación del programa generado y generar un mejor resultado.

### 5.2.1. Mejorar modelado de la propagación

Una opción clara para mejoras es continuar añadiendo aspectos más complejos de la interacción del sonido con el ambiente, con el fin de añadir realismo y nuevas facetas a la simulación. Los puntos específicos de la interacción que se decidieron no modelar son la transmisión, la refracción y la difracción. Cada uno presenta distintos desafíos: para modelar la refracción, se necesita investigar cómo representar ambientes con distintas densidades y la propagación del sonido a través de ellos; para la transmisión, esto implicaría introducir nuevos rayos de sonido con cada impacto en superficies que permitan la transmisión, que implica un aumento exponencial en la cantidad de rayos; y para la difracción, aunque los modelos complejos pueden representarla con precisión, el principal desafío es mantener un modelo simplificado y que simule de forma eficaz y veloz este fenómeno.

Otro aspecto significativo que fue simplificado en el modelo es la homogeneización de la interacción del sonido con el ambiente según la frecuencia de la onda. En el modelo actual, se asume que el comportamiento del sonido al propagarse es independiente de su frecuencia, y esto no refleja completamente la realidad física donde diferentes frecuencias interactúan de manera diversa con el entorno. Como se exploró en la Sección [2.1.3](#), la absorción del sonido es mayor para frecuencias altas, mientras que la difracción

es más pronunciada en frecuencias bajas debido a sus longitudes de onda más largas, y las propiedades de reflexión varían según la frecuencia, con superficies rugosas dispersando más las ondas altas. Incorporar estas complejidades en el modelo mejorará significativamente la fidelidad y el realismo de la simulación acústica, proporcionando una herramienta más robusta y precisa para el análisis del comportamiento del sonido en diversos entornos.

### 5.2.2. Múltiples emisores

Agregar múltiples emisores al sistema de simulación acústica sería altamente deseable ya que permitiría modelar de manera más realista una amplia gama de escenarios del mundo real. En muchas situaciones prácticas, el sonido no proviene de una única fuente, sino de múltiples fuentes distribuidas en el espacio. Por ejemplo, en una sala de conciertos, los sonidos de los instrumentos musicales provienen de distintas ubicaciones en el escenario. Modelar múltiples emisores permitiría simular cómo se mezcla y se propaga el sonido desde cada instrumento hasta los oyentes, ofreciendo una representación más precisa de la experiencia acústica.

Otro caso relevante es el entorno urbano, donde el ruido ambiental se compone de múltiples fuentes como el tránsito, construcciones y las actividades humanas. La capacidad de incluir múltiples emisores en la simulación permitiría evaluar el impacto del ruido en diferentes ubicaciones, diseñar estrategias de mitigación y planificar mejor la acústica urbana.

Agregar esta funcionalidad implica resolver la superposición de múltiples señales en una única salida de audio. La mezcla de estas señales puede ser constructiva o destructiva y debe resolverse de manera precisa para garantizar que la interacción de las distintas fuentes de sonido se represente de manera fiel y coherente.

### 5.2.3. Canales de salida parametrizables

Utilizar más de dos canales de audio ofrece una experiencia auditiva más inmersiva en comparación, y este puede ser un aspecto deseable a desarrollar. Dado el equipamiento adecuado para reproducir audios con múltiples canales de salida, el desafío de parametrizar la cantidad de canales de salida resulta accesible.

Suponiendo que el sistema sea capaz de generar múltiples IRs, uno por canal de salida, el módulo auralizador puede realizar la convolución de cada IR en paralelo si se aprovechan al máximo las funcionalidades de CUDA. El ajuste principal consiste en la generación de un objeto receptor capaz de separar los impactos por canal. Como se mencionó en la Sección 3.3.2, el receptor utilizado es un objeto estático dividido en dos mitades. Un posible enfoque para agregar múltiples canales sería generar un objeto estático fragmentado en la cantidad de canales de salida deseados en tiempo de ejecución.



# Bibliografía

- Abel, Jonathan S. et al. (2010). “Estimating Room Impulse Responses from Recorded Balloon Pops”. En: *Audio Engineering Society*.
- Altavilla, Dave (2018). “Nvidia And Microsoft Lay Foundation For Photorealistic Gaming With Real-Time Ray Tracing”. En: *Forbes*.
- Arvo, James et al. (2019). “An Introduction to Ray Tracing”. En: *Academic Press*.
- Ballou, Glen M. (2008). “Handbook for Sound Engineer”. En: *Elsevier*.
- Barron, Mike (1995). “Interpretation of Early Decay Times in Concert Auditoria”. En: *Acta Acustica united with Acustica* 81, págs. 320-331.
- Beranek, Leo L. (1954). “Acoustics”. En: *McGraw-Hill*.
- Berg, Richard E. (2024). “Noise.” En: *Encyclopedia Britannica*. URL: <https://www.britannica.com/science/sound-physics/Noise> (visitado 05-08-2024).
- Bort, Inès de y Benoit Beckers (2021). “Impact of multiple reflections on urban acoustics”. En: *Journal of Physics: Conference Series* 2042 012052.
- Caulfield, Brian (2018). *What’s the Difference Between Ray Tracing and Rasterization?* URL: <https://blogs.nvidia.com/blog/whats-difference-between-ray-tracing-rasterization/> (visitado 05-08-2024).
- Chang, Chun-Fa, Kuan-Wei Chen y Chin-Chien Chuang (2015). “Performance comparison of rasterization-based graphics pipeline and ray tracing on GPU shaders”. En: *IEEE*.
- Christen, Martin (2005). “Ray Tracing on GPU”. En: *University of Applied Sciences Basel (FHBB)*.
- Claus, Christensen, Koutsouris George y Rindel Jens (2013). “The ISO 3382 parameters: Can we simulate them? Can we measure them?” En: *Akustika* 20.

- Congress, Library Of (s.f.). *Wavefront OBJ File Format*. URL: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml> (visitado 05-08-2024).
- Corporation, IBM y Microsoft Corporation (1991). *Multimedia Programming Interface and Data Specifications 1.0*. URL: <https://www.aelius.com/njh/wavemetatools/doc/riffmci.pdf> (visitado 05-08-2024).
- Everest, F. Alton (2001). “The Master Handbook of Acoustics”. En: *McGraw-Hill*.
- Farber, Rob (2011). “CUDA Application Design and Development”. En: *Elsevier*.
- Fletcher, Harvey y W. A. Munson (1933). “Loudness, Its Definition, Measurement and Calculation”. En: *Journal of the Acoustical Society of America* 5.
- Gamma, Erich et al. (1994). “Patrones de Diseño. Elementos de Software Orientado a Objetos Reutilizables”. En: *Addison Wesley*.
- Gardner, Bill y Keith Martin (1994). “HRTF Measurements of a KEMAR Dummy-Head Microphone”. En: *MIT Media Lab Perceptual Computing - Technical Report 280*.
- Gerzon. M and Ambisonics at Oxford (s.f.). *What is Ambisonics?* URL: <https://intothesoundfield.music.ox.ac.uk/what-is-ambisonics> (visitado 05-08-2024).
- Gomes, Jonas y Luiz Velho (1999). “From Fourier Analysis to Wavelets - Course Notes – SIGGRAPH 99”. En: *Instituto de Matemática Pura e Aplicada, IMPA*.
- Google (s.f.). *Resonance Audio - Fundamental Concepts*. URL: <https://resonance-audio.github.io/resonance-audio/discover/concepts.html> (visitado 05-08-2024).
- Heckbert, Paul (1998). “Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm”. En: URL: <https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf> (visitado 05-08-2024).
- International Organization for Standardization (1997). *ISO 10843:1997 - Acoustics, Methods for the description and physical measurement of single impulses or series of impulses*.
- International Organization for Standardization (2008). *ISO 3382-2. Acoustics — Measurement of room acoustic parameters — Part 2: Reverberation time in ordinary rooms*.
- Jordan, Vilhelm Lassen (1970). “Acoustical Criteria for Auditoriums and Their Relation to Model Techniques”. En: *Acoustical Society of America* 47, 408–412.

- JSON (s.f.). *Introducing JSON*. URL: <https://www.json.org/json-en.html> (visitado 05-08-2024).
- Kajiya, James T. (1986). “The Rendering Equation”. En: *Association for Computing Machinery* 20, 143–150.
- L., Wightman F. y Kistler D. J (1989). “Headphone Simulation of Free-Field Listening. I: Stimulus Synthesis”. En: *Journal of the Audio Engineering Society* 85(2), 858–867.
- Lavry Engineering Inc. (2012). *The Optimal Sample Rate for Quality Audio*. URL: [https://www.lavryengineering.com/pdfs/lavry-white-paper-the\\_optimal\\_sample\\_rate\\_for\\_quality\\_audio.pdf](https://www.lavryengineering.com/pdfs/lavry-white-paper-the_optimal_sample_rate_for_quality_audio.pdf) (visitado 05-08-2024).
- Lee, Jin-Hee, Su-Jeong Shin y Cynthia Istook (2006). “Analysis of Human Head Shapes in the United States”. En: *International Journal of Human Ecology* 7, págs. 77-83.
- Ludvigsen, H. y A. C. Elster (2010). “Real-Time Ray Tracing Using Nvidia OptiX”. En: *Norwegian University of Science and Technology*.
- Mauskopf, Eric (s.f.). “Resonance Audio”. En: *Google* (). URL: <https://blog.google/products/google-ar-vr/resonance-audio-multi-platform-spatial-audio-scale/> (visitado 05-08-2024).
- McGuire, Morgan (2017). *Computer Graphics Archive*. <https://casual-effects.com/data>. URL: <https://casual-effects.com/data> (visitado 05-08-2024).
- Möser, Michael y José Luis Barros (2009). “Ingeniería Acústica, Teoría y Aplicaciones”. En: *Springer Berlin, Heidelberg*.
- NVIDIA (s.f.[a]). *CUDA Toolkit*. URL: <https://developer.nvidia.com/cuda-toolkit> (visitado 05-08-2024).
- NVIDIA (s.f.[b]). *cuFFT API Reference*. URL: <https://docs.nvidia.com/cuda/cufft/index.html> (visitado 05-08-2024).
- NVIDIA (s.f.[c]). *NVIDIA OptiX 7.7 – Programming Guide*. URL: <https://raytracing-docs.nvidia.com/optix7/guide/index.html> (visitado 05-08-2024).
- NVIDIA (s.f.[d]). *NVIDIA OptiX™ Ray Tracing Engine*. URL: <https://developer.nvidia.com/rtx/ray-tracing/optix> (visitado 05-08-2024).
- NVIDIA (s.f.[e]). *NVIDIA RTX™ platform*. URL: <https://developer.nvidia.com/rtx#:~:text=RTX%2Dcapable%20GPUs%20include%20dedicated,games%20and%20other%20graphics%20applications> (visitado 05-08-2024).

- NVIDIA (s.f.[f]). *VRWorks Audio SDK in-depth*. URL: <https://developer.nvidia.com/vrworks-audio-sdk-depth> (visitado 05-08-2024).
- Pharr, Matt, Wenzel Jakob y Greg Humphreys (2023). *Physically Based Rendering: From Theory To Implementation*. Fourth. The MIT Press.
- Purcell, Timothy J. et al. (2005). “Ray Tracing on Programmable Graphics Hardware”. En: *Stanford University*.
- Satut, Camilo (2022). “Simulación del sonido con trazado de rayos”. En: *FIng, Udelar*.
- Savioja, Lauri (2000). “Modeling Techniques for Virtual Acoustics”. En: *Helsinki University of Technology*.
- Scavone, Gary P. (s.f.). *The RtAudio Home Page*. URL: <https://www.music.mcgill.ca/~gary/rtaudio> (visitado 05-08-2024).
- Schröder, Dirk (2011). “Physically Based Real-Time Auralization of Interactive Virtual Environments”. En: *Rheinisch-Westfälische Technische Hochschule Aachen*.
- Siltanen, Samuel, Tapio Lokki y Lauri Savioja (2012). “Rays or Waves? Understanding the Strengths and Weaknesses of Computational Room Acoustics Modeling Techniques”. En: *Proceedings of the International Symposium on Room Acoustics*.
- Smith, Steven W. (1997). “The Scientist and Engineer’s Guide to Digital Signal Processing”. En: *California Technical Publishing*. Chapter 18: FFT Convolution.
- Stark, Adam (s.f.). *Audio File*. URL: <https://github.com/adamstark/AudioFile> (visitado 05-08-2024).
- Stevens, Francis y Damian Murphy (2014). “Spatial impulse response measurement in an urban environment”. En.
- The Khronos Group Inc (s.f.). *OpenGL*. URL: <https://www.opengl.org/> (visitado 05-08-2024).
- Thomas, Reuben (2016). “Wayverb”. En: *University of Huddersfield*. URL: <https://reuk.github.io/wayverb/> (visitado 05-08-2024).
- Tsingos, Nicolas et al. (2002). “Validation of Acoustical Simulations in the Bell Labs Box”. En: *Princeton University*.
- Valve Software (s.f.). “Steam Audio”. En: (). URL: <https://valvesoftware.github.io/steam-audio/> (visitado 05-08-2024).

- Weaver, J. (1989). "Theory of Discrete and Continuous Fourier Transform". En: *Wiley Sons*.
- Wu, Fukun y Changwen Zheng (2014). "Simulation of Wave Effects Based on Ray Tracing". En: *IEEE*.
- Yang, L.N. y B.M. Shield (2000). "Development of a Ray Tracing Computer Model for the Prediction of the Sound Field in Long Enclosures". En: *Journal of Sound and Vibration* 229, págs. 133-146.
- Yang, Zhijian y Romit Roy Choudhury (2021). "Personalizing Head Related Transfer Functions for Earables". En: *University of Illinois at Urbana Champaign*.



# Anexo A

## Archivo de configuración

Archivo de configuración de ejemplo para la escena 1D.

---

```
1
2 {
3     "renderer_parameters": {
4         "initial_volume": 1,
5         "ir_length_in_seconds": 2,
6         "width": 1366,
7         "height": 768,
8         "write_first_ir_to_file": true,
9         "write_first_output_to_file": false,
10        "re_render_distance_threshold": 2,
11        "re_render_angle_threshold": 5
12    },
13    "scene_parameters": {
14        "mono": false,
15        "audio_file_path": "../assets/sound_samples/Clap.wav",
16        "scene_file_path": "../assets/models/1D_U.obj",
17        "initial_receiver_pos": {
18            "x": -2.5,
19            "y": 9.9,
20            "z": 0.0
21        },
22        "initial_emitter_pos": {
23            "x": 0.0,
24            "y": 0.0,
25            "z": 0.0
26        }
27    },
28    "pathtracer_parameters": {
```

```

29         "base_power": 100,
30         "rays": {
31             "x": 100,
32             "y": 100,
33             "z": 100
34         },
35         "ray_energy_threshold": 0.0,
36         "ray_max_bounces": 100,
37         "hrtf_absorption_rate": 0.9,
38         "materials": [
39             {
40                 "name": "plywood",
41                 "mat_absorption": 0.09
42             }
43         ]
44     }
45 }

```

---

Notas sobre los valores y sus significados:

- *initial\_volume* indica el volumen inicial de la aplicación. El volumen se puede alterar durante la ejecución (ver Sección 3.3.3)
- *write\_first\_ir\_to\_file* y *write\_first\_output\_to\_file* son herramientas para la experimentación, permiten guardar en un archivo sus buffers respectivos de forma automática. Esto también se puede hacer a demanda durante la ejecución (ver Sección 3.3.3).
- *re\_render\_distance\_threshold* es la distancia a recorrer dentro de la escena por el receptor para volver a realizar la simulación y auralización.
- *re\_render\_angle\_threshold* es similar, indicando cuantos grados debe de rotar la cámara para volver a simular y auralizar.
- *mono* indica si la auralización es mono o estéreo.
- *base\_power* es el nivel de potencia sonora que será distribuido entre todos los rayos.
- *ray\_energy\_threshold* es la intensidad mínima que debe de tener un rayo o se descarta.
- *ray\_max\_bounces* rebotes máximos que debe de tener un rayo o se descarta.
- *hrtf\_absorption\_rate* es el factor de absorción para simular HRTF.

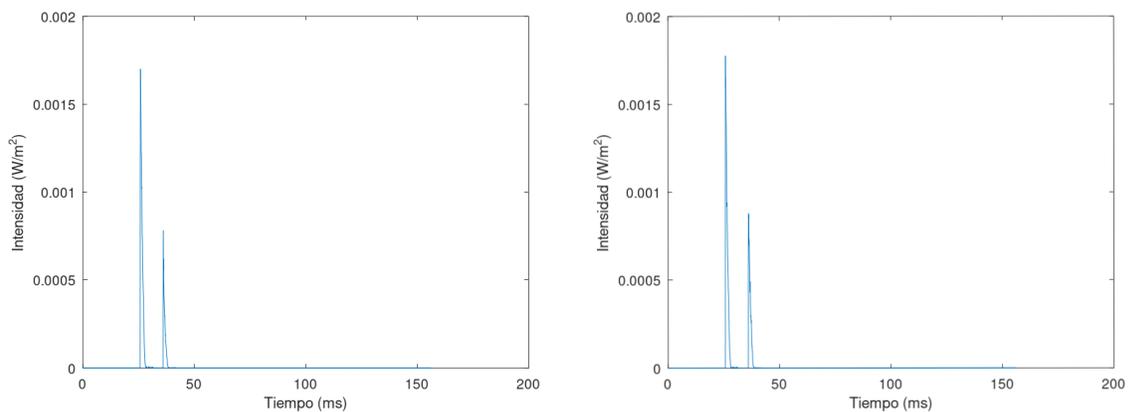
# Anexo B

## Experimentos extra

### B.1. Experimentos con limite de rebotes

Todos los experimentos realizados con  $10^7$  rayos y absorción 0. En la Figura B.1 y B.2 se aprecia claramente como los máximos no se ven afectados por la cantidad de rebotes, aunque si se presentan diferencia en la forma del pico.

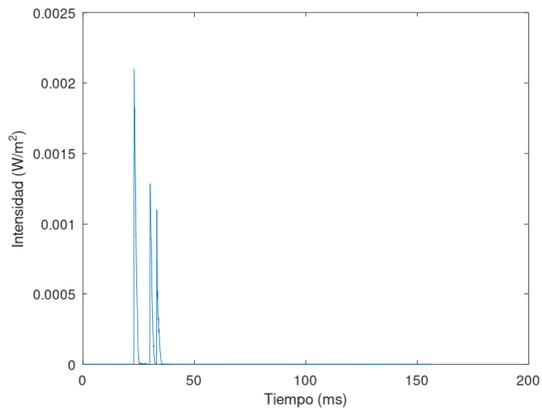
En la Figura B.3 claramente se aprecia como el tercer pico aumenta en intensidad al tener caminos de más rebotes.



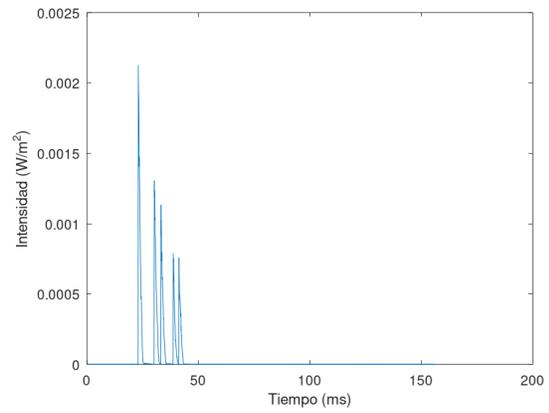
(a) IR por caminos de 1 o menos rebotes.

(b) IR por caminos de 2 o menos rebotes.

Figura B.1: Simulación en la escena 1D.

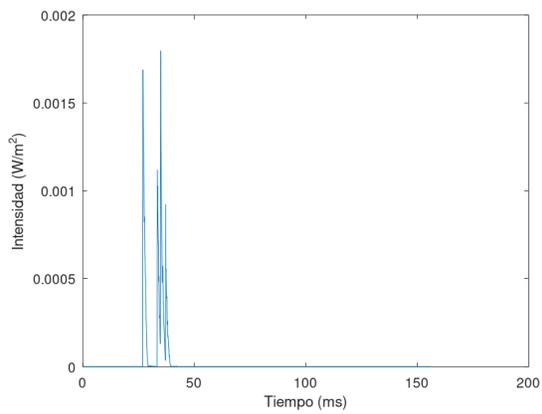


(a) IR por caminos de 1 o menos rebotes.

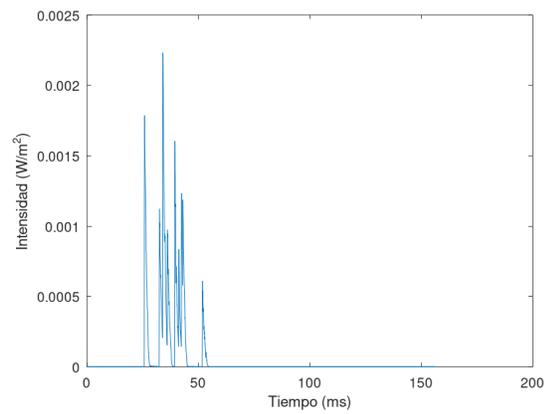


(b) IR por caminos de 2 o menos rebotes.

Figura B.2: Simulación en la escena 2D.



(a) IR por caminos de 1 o menos rebotes.



(b) IR por caminos de 2 o menos rebotes.

Figura B.3: Simulación en la escena 3D.