

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE CANTABRIA**



***Proyecto Fin de Carrera***

**SIMULACIÓN DE SENSORES DE FIBRA  
ÓPTICA CON BLENDER**  
(Optical fiber sensors simulation using  
Blender)

Para acceder al Título de

**INGENIERO TÉCNICO DE TELECOMUNICACIÓN**

Autor: Eduardo Ruiz Terrazas

Septiembre – 2014



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

**INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN**  
*CALIFICACIÓN DEL PROYECTO FIN DE CARRERA*

**Realizado por: Eduardo Ruiz Terrazas**

**Director del PFC: Adolfo Cobo García**

**Título:** “*Simulación de sensores de fibra óptica con Blender*”

**Title:** “*Optical fiber sensors simulation using Blender*”

**Presentado a examen el día: 18 de Septiembre de 2014**

*para acceder al Título de*

**INGENIERO TÉCNICO DE TELECOMUNICACIÓN,  
ESPECIALIDAD EN SISTEMAS ELECTRÓNICOS**

*Composición del Tribunal:*

*Presidente: Lomer Barboza, Mauro*

*Secretario: Cobo García, Adolfo*

*Vocal: Agüero Calvo, Ramón*

*Este Tribunal ha resuelto otorgar la calificación de:*

.....

*Fdo.: El Presidente*

*Fdo.: El Secretario*

*Fdo.: El Vocal*

*Fdo.: El Director del PFC  
(sólo si es distinto del Secretario)*

*Vº Bº del Subdirector*

*Proyecto Fin de Carrera Nº  
(a asignar por Secretaría)*

*Trabajo realizado en parte con los medios aportados por el proyecto SENSORES DE FIBRA  
ÓPTICA PARA SEGURIDAD Y PROTECCIÓN (TEC2013-47264-C2-1-R).*

## *Agradecimientos*

*A mi familia, por la preocupación que han tenido durante tanto tiempo para la finalización de éste.*

*A mis amigos, por tan buenos ratos y darme los ánimos necesarios para continuar cuando más lo necesitaba.*

*A Javi y mis primos Olga y Salva por tan inestimable ayuda a costa de su tiempo por un bien para mí.*

*Y especialmente a Adolfo, por su incalculable apoyo e infinita paciencia en que siempre me mostró su lado bueno aún habiéndole fallado en tan numerosas ocasiones.*

***El hombre nunca sabe de lo que es capaz  
hasta que lo intenta.  
(Charles Dickens)***

***No es más grande aquel que nunca falla,  
sino aquel que nunca se da por vencido.  
(Anónimo)***

# ÍNDICE

---

<b>CAPÍTULO 1: INTRODUCCIÓN</b>	<b>1</b>
<b>1.1. CONTEXTO DEL TRABAJO</b>	<b>1</b>
<b>1.2. MOTIVACIÓN DEL PROYECTO</b>	<b>1</b>
<b>1.3. OBJETIVOS DEL PROYECTO</b>	<b>2</b>
<b>1.4. ORGANIZACIÓN DEL DOCUMENTO</b>	<b>4</b>
<b>CAPÍTULO 2: BLENDER</b>	<b>5</b>
<b>2.1. INTRODUCCIÓN E HISTORIA</b>	<b>5</b>
<b>2.2. INTERFAZ DE BLENDER</b>	<b>6</b>
<b>2.3. CARACTERÍSTICAS</b>	<b>8</b>
<b>2.4. PROYECTOS Y DISEÑOS REALIZADOS EN BLENDER</b>	<b>13</b>
<b>2.5. BLENDER PARA SIMULAR SENSORES DE FIBRA ÓPTICA</b>	<b>21</b>
<b>CAPÍTULO 3: PRUEBAS REALIZADAS</b>	<b>22</b>
<b>3.1. FIBRA ÓPTICA: PARÁMETROS Y COMPORTAMIENTO</b>	<b>22</b>
<b>3.2. BLENDER: IDEAS Y DISEÑOS</b>	<b>25</b>
<b>CAPÍTULO 4: ESTIMACIÓN LUMINOSA UTILIZANDO MATLAB</b>	<b>30</b>
<b>4.1. INTRODUCCIÓN</b>	<b>30</b>
<b>4.2. OBJETIVOS A LOGRAR CON LA AYUDA DE MATLAB</b>	<b>30</b>
<b>4.3. DEFINICIÓN DEL CÓDIGO EN MATLAB PARA CÁLCULO DE INTENSIDAD LUMINOSA TOTAL</b>	<b>30</b>
<b>4.4. VARIACIÓN DE LUZ RECIBIDA EN FUNCIÓN DE LA POSICIÓN DE LA FUENTE</b>	<b>32</b>
<b>4.5. COMPROBAR LA PROPAGACIÓN DE LUZ HACIA LA SALIDA SIMULANDO EL CASO DEL LABORATORIO</b>	<b>34</b>
<b>CAPÍTULO 5: RESULTADOS</b>	<b>36</b>
<b>5.1. INTRODUCCIÓN</b>	<b>36</b>
<b>5.2. MODELADO DE FIBRA DE CRISTAL EN BLENDER</b>	<b>36</b>
<b>5.3. OBTENCIÓN DE ESTIMACIÓN LUMINOSA POR MATLAB</b>	<b>40</b>
<b>CAPÍTULO 6: INTRODUCCIÓN</b>	<b>43</b>
<b>6.1. INTRODUCCIÓN</b>	<b>43</b>
<b>6.2. RESUMEN</b>	<b>43</b>
<b>6.3. CONCLUSIONES</b>	<b>43</b>
<b>6.4. LÍNEAS FUTURAS</b>	<b>44</b>
<b>CAPÍTULO 7: BIBLIOGRAFÍA</b>	<b>45</b>



# Capítulo 1.- Introducción

---

## ❖ 1.1. Contexto:

A menudo son incontables las ocasiones en que uno mismo se ha cuestionado si habrá algún tipo de aplicación o software que puede servirle de ayuda para un tema en concreto, recrear una idea, etc., o si existirá alguna opción mejor que la que pueda estar utilizando habitualmente dado que ésta no es capaz de cumplir las con expectativas y/o tareas deseadas. A parte de esto, destaca por encima aquellas opciones que resulten accesibles a cualquier público y a ser posibles, gratuitas.

En el campo óptico y fotónico también se plantea mucho esta cuestión dado que los materiales que se precisan para testear las distintas ideas, teorías o hipótesis, no están siempre al alcance de todos y por lo general tienden a ser costosos. Contar con un software que sea capaz de simular un trabajo de laboratorio, bajo un previo y estricto estudio de su viabilidad, sin la necesidad de recurrir al material y la incertidumbre de si llegará a dar resultado o no, resulta una aportación fundamental y considerablemente importante.

El inconveniente que se encuentra actualmente en cuanto a dar con un software adecuado para tratar estos temas, es que las opciones existentes son muy limitadas y realmente costosas, pues puede verse como ejemplo que adquirir una licencia del software ZEMAX, programa diseñado especialmente para ámbito óptico, su cuantía ronda los 3.600\$.

Al verse esta gran dificultad para acceder a estas opciones, surge entonces la incertidumbre de si es posible que algún otro tipo de software pueda llegar a emular las simulaciones complejas que puede encontrarse en programas profesionales específicos para ello e incluso, por qué no, si es posible que pueda ir un poco más allá y llegar a solventar casos que no han conseguido resolverse con algunas de estas herramientas profesionales.

Por ello, conociendo que el fundamento básico que rige estos campos es la luz, con observar el potencial en auge que está suponiendo el mundo de la animación y el diseño en 3D en el día de hoy, sería un hecho realmente a estudiar por trabajar enormemente con ella. En el caso de darse esto y se observa que esta opción es viable, constituiría una gran alternativa a tener en cuenta contribuyendo enormemente en la ayuda de futuros proyectos de laboratorio a la hora de buscar un programa acorde a las necesidades que se buscan.

## ❖ 1.2. Motivación de este proyecto:

La motivación fundamental de este proyecto proviene de la cuestión de verificar si podría llegar a ser posible simular un trabajo de laboratorio, tomando como referencia un caso concreto en el que se observan una ligera serie de hechos ilógicos entre lo práctico y teórico, a través de un software de entorno gráfico 3D y gratuito, permitiendo esclarecer así una solución al respecto. Este trabajo en particular consiste en que al incidir con un láser de gran potencia sobre un material, en el instante preciso de la ruptura, se desprende un haz luminoso el cual se quiere recoger con la ayuda de una espiral cilíndrica de fibra óptica de plástico a través de sus paredes internas para su estudio. Gracias a esta idea práctica, se consigue recoger una parte casi inapreciable de intensidad luminosa. Sin embargo, según el aspecto teórico esto no podría ser posible dado que la luz recibida al final de la fibra utilizando este proceso, debería ser nula.

En el caso de que pudiese llegarse a recrear la situación descrita y se observarse que sucede tal y como el caso práctico, confirmándolo bajo un previo estudio, se demostraría que gracias a esta herramienta podrían llegarse a simular situaciones de laboratorio aportando soluciones al respecto de forma concisa, donde éstas sólo podían verse en softwares de uso profesional y en ocasiones, ni con ello.

Partiendo de este punto y que en el trabajo se están utilizando elementos ópticos, en particular la fibra, es necesario hallar un programa que sea capaz de trabajar exhaustivamente con la luz, con el cristal como material y por encima de todo, con los fenómenos y principios producidos por esta en los materiales.

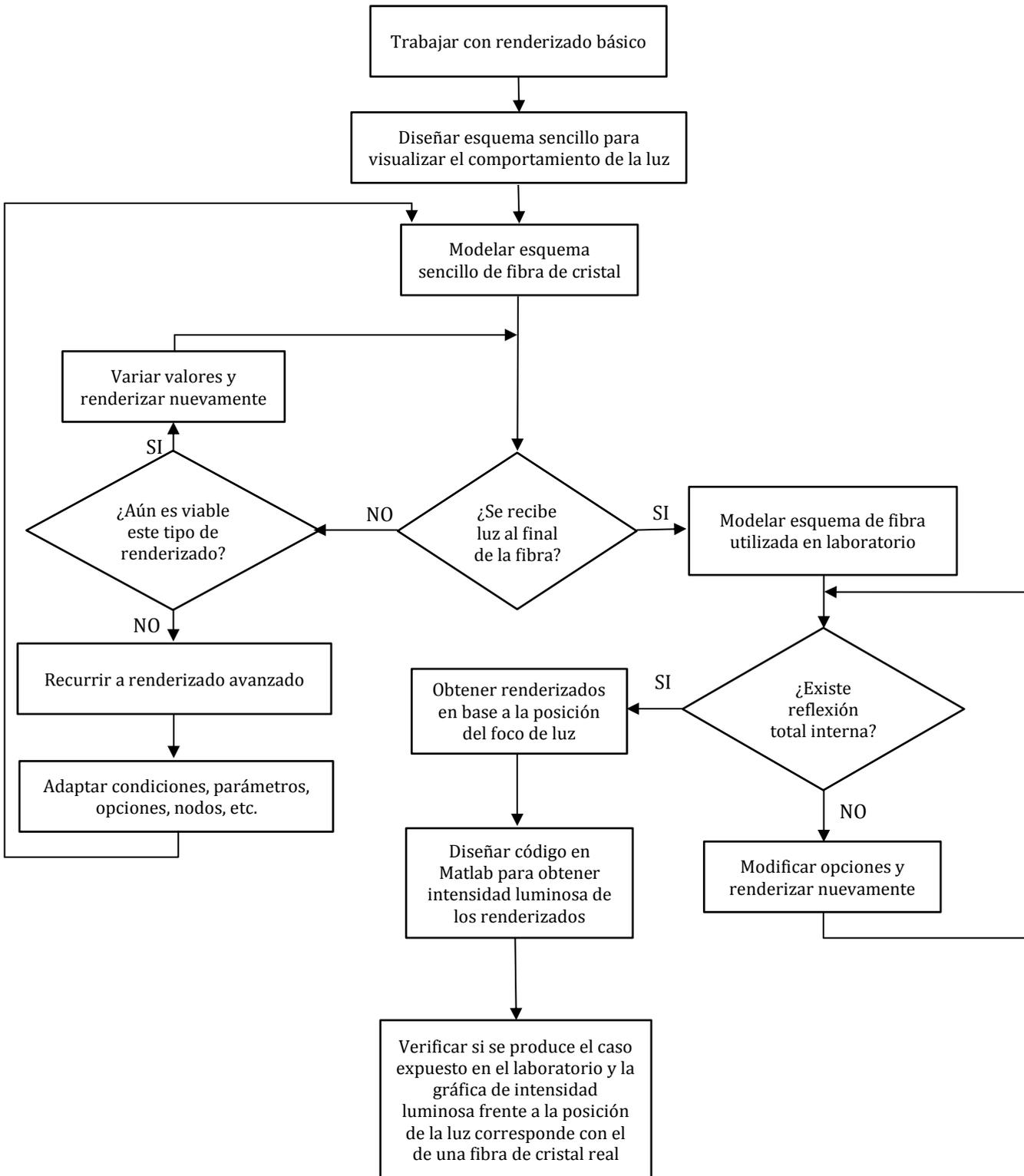
Planteadas todas las incógnitas, requisitos necesarios y los objetivos a cumplir, se establece el programa **Blender Render** como la herramienta para llevar a cabo este proyecto y que llegue a ser capaz de cumplir estas expectativas.

### ❖ 1.3. Objetivos

El objetivo principal de este proyecto, como se ha planteado, es el de lograr verificar si por medio de la herramienta de diseño gráfico **Blender**, es posible simular una propuesta de laboratorio y demostrar si sería un aporte a tomar en cuenta para futuras ayudas en trabajos de laboratorio. Una vez sentadas las bases de la idea a realizar, se enumeran a continuación los objetivos a cumplir:

- Estudiar las propiedades del programa y ejemplos de este para observar las posibles funcionalidades y ver hasta qué punto alcanzan sus resultados.
- Conocer y aprender el programa en su entorno básico para dar así un manejo más ameno a la hora de ir trabajando con él.
- Entender cómo llega a trabajar la luz dentro del programa.
- Diseñar y modelar la fibra óptica con los parámetros apropiados para su posterior aplicación de la luz.
- Conocer los aspectos más avanzados del programa en el caso de que las herramientas básicas no permitan lograr el objetivo deseado, para que así sea posible recurrir a opciones alternativas y/o más complejas y llevar a cabo nuestra idea.
- Verificar tras el modelado y añadido de una fuente de luz en uno de los extremos de la fibra, que se cumplen fenómenos obligados como la reflexión total interna permitiendo que la luz sea capaz de recorrer el interior de la fibra y llegar hasta el extremo opuesto demostrando que nuestro diseño es correcto.
- Demostrar que las variaciones en la fuente de luz en cuanto a intensidad, desplazamientos, etc. producen una respuesta significativa en el extremo saliente a la hora de observar la luz recibida.
- Realizar capturas de los distintos renderizados (jerga utilizada para definir la simulación visual del diseño) aplicando desplazamientos a la fuente de luz.
- Calcular la cantidad exacta de radiación luminosa recibida por medio de la implementación de un código en Matlab a partir de las imágenes renderizadas.
- Obtener con ello una gráfica de la radiación luminosidad recibida en función de la posición de la fuente de luz con respecto a la entrada de la fibra.
- Verificar y demostrar si el programa es apto simular tareas planteadas en el laboratorio al comprobar, si la situación presentada en el programa ocurre de la misma forma que en trabajo del laboratorio ayudando en caso afirmativo notablemente en ocasiones futuras.

En vista de los objetivos listados, se establece el siguiente diagrama de flujo como el método de trabajo a seguir:



## ❖ 1.4. Organización del documento

La memoria de este documento se estructura de la siguiente manera:

- *Capítulo 1: se realiza una presentación al problema existente, tanto a las limitaciones que pueden encontrarse a la hora de acceder a softwares de tipo profesional como al dilema ocasionado durante un trabajo de laboratorio. Se añade a su vez un listado de objetivos a cumplir para permitir así esclarecer la viabilidad de este programa.*
- *Capítulo 2: se ofrece una introducción al programa Blender junto con una detallada explicación de las principales características de éste. Como añadido, se muestran una serie de ejemplos y trabajos elaborados bajo esta herramienta de diseño para una mejor visualización de las capacidades del programa.*
- *Capítulo 3: se describe inicialmente el qué es una fibra óptica, cómo es su comportamiento y sus distintos modos de funcionamiento. A su vez, se plantean las diversas ideas y modelos para la obtención del diseño de fibra óptica que se pretende conseguir a lo largo de este proyecto.*
- *Capítulo 4: se habla sobre el procesado de imágenes mediante el uso de herramientas del programa Matlab y se crea un código para realizar la estimación de radiación luminosa generada por las imágenes que nos devuelven las renderizaciones de Blender, además de esclarecer la viabilidad de nuestro diseño.*
- *Capítulo 5: dentro de este capítulo se recogen todos los resultados logrados tanto en los diseños realizados bajo Blender, como la obtención de medidas bajo código en Matlab.*
- *Capítulo 6: finalmente, se presenta un resumen del trabajo realizado a lo largo de todo este proyecto, así como las conclusiones a las que se han llegado durante este desarrollo y las posibles líneas futuras de trabajo.*
- *Capítulo 7: recoge de manera informativa, la bibliografía de las principales fuentes a las que se ha accedido para la obtención de la información necesaria para llevar a cabo este proyecto.*

# Capítulo 2.- Blender

## ❖ 2.1. Introducción e historia

*Blender Render es un programa de diseño gráfico en 3D especializado en animación, iluminación, texturización, modelación y edición digital tanto en fotografía, vídeos, etc.*

*Además es considerada una herramienta realmente distinguida para la creación de videojuegos dado que posee un motor de juegos interno.*

*El programa a día de hoy se distribuye de forma libre con compatibilidad para todas las plataformas y puede encontrarse en casi todos los idiomas.*

*Su origen se remonta a 1988 cuando Ton Roosendaal co-fundó el estudio de animación Holandes, **NeoGeo**.*



*Tras casi una década después, en 1995, se vio que la herramienta utilizada era demasiado voluminosa y comenzaba a quedarse obsoleta por lo que se optó por reescribirla desde el principio. Esta nueva reescritura dio paso a lo que se conoce actualmente, como el software de creación y animación en 3D "**Blender**".*

*A medida que fueron transcurriendo los años, se realizaron diversas mejoras y se pensó que esta herramienta podría llegar a utilizarse fuera del ámbito de **NeoGeo** y que podría ser de gran ayuda en otras casas y/o empresas. Se decidió entonces dar paso a la creación de una nueva empresa por parte de su autor conocida como **NaN** (Not a Number). En la bases de esta empresa, se encontraba el deseo de que pudiese llegarse a distribuir un paquete de herramientas gratuito de creación de gráficos en 3D que fuese compacta y multiplataforma. Esto al fin y al cabo, era una idea que iba en contra de los matices comerciales de por aquel entonces, ya que la mayor parte de los programas que se podían llegar a adquirir de diseño gráfico, suponían un gasto de miles de dólares para el usuario.*

*Debido a las dificultades por las que atravesó la compañía, tanto en cuestiones económicas como en lo referente en ventas, las inversiones que recibía se vieron realmente afectadas deteniendo de esta forma el desarrollo de **Blender**.*

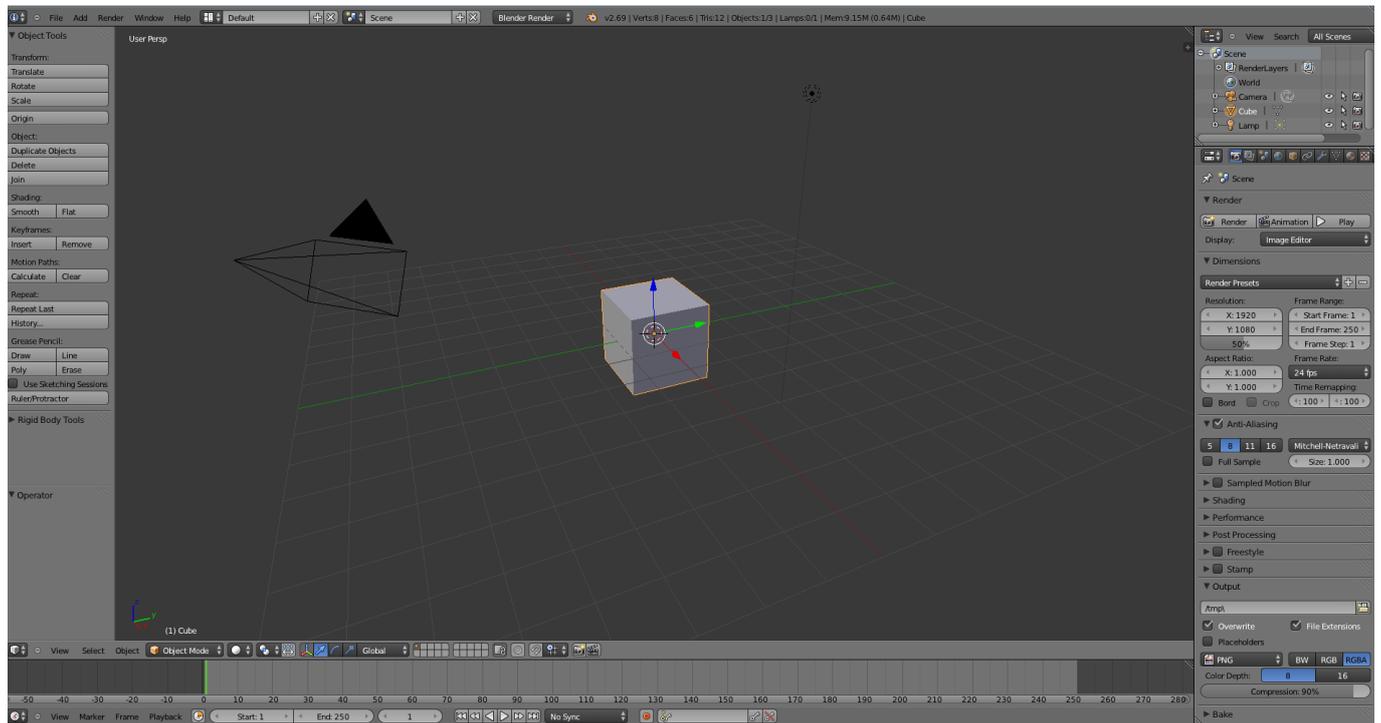
*Ton Roosendaal no quiso permitir que **Blender** cayese en el olvido, y relanzar una nueva compañía con los medios de que disponía no era una opción factible, por lo que en marzo de 2002 decidió lanzar y fundar la organización no lucrativa, **Blender Foundation**, cuyo objetivo primordial fue la continuación y desarrollo de Blender como un proyecto de código abierto basado en una comunidad de usuarios, bajo el sobrenombre de campaña "Liberar a Blender". Esta debía recoger una suma de 100.000€ para la compra de derechos del código fuente y propiedad intelectual a los inversores de **NaN**. La cifra se alcanzó en tan sólo siete semanas.*

*El 13 de octubre de 2002, fue liberado al mundo bajo los términos de la Licencia Pública General de GNU (GPL) y continua hasta el día de hoy desarrollándose y mejorándose gracias a la participación de usuarios de todas partes del mundo bajo la supervisión de su creador, Ton Roosendaal.*

*Actualmente es incontable la cantidad de foros, videoblogs, páginas y demás, en que usuarios de cualquier rincón del planeta comparten sus proyectos a otros usuarios para mostrar sus progresos, solicitar ayuda, compartir opiniones, programar addons entre algunos, etc. permitiendo que la comunidad de Blender siga expandiéndose de una forma desmesurada y realmente amistosa.*

## ❖ 2.2. Interfaz de Blender

La interfaz de Blender resulta a primera vista intuitiva, presentando un plano de cuadrículas y un cubo por defecto con el que uno puede realizar sus primeras pruebas tales como: movimiento y rotaciones de la cámara, desplazamientos del objetos, etc. Todos en base a movimientos y clicks del ratón.



Realizando un vistazo rápido a lo que se puede observar, en el panel izquierdo se encuentran las **herramientas de edición del objeto**, las cuales permiten entre otras muchas desde duplicar, trozear y escalar un objeto, hasta fusionar varios en uno sólo.



En el panel derecho se muestra las capas en que está organizado, o compuesto, nuestro proyecto y las distintas **opciones de caracterización** del entorno, imagen, objetos, etc. Dentro estas opciones se encuentran los parámetros que definen cada **elemento seleccionado del diseño**. Por ejemplo, seleccionando el opción de la luz, se permite establecer unos valores o propiedades a la fuente de luz **seleccionada**. Al igual que entrando en la opción de propiedades del objeto, se definen los parámetros que caracterizarán a este, tales como la textura, luminosidad, transparencia, color, etc.



En el panel superior se encuentran las opciones propias de Blender dentro de las cuales, se puede acceder a la configuración del programa, a la adición de elementos al diseño o la personalización de la visualización de la en pantalla entre otras. A la derecha de estas, se encuentra la definición de las capas y una de las más importantes de todo Blender: **la selección del tipo de renderizado**.



Por último, en la parte inferior se muestra el menú de **opciones del mapa de trabajo**, entre las que se puede acceder a los tipos de selección de objetos, el tipo de modo con el que se va a trabajar (esculpido, visualización, pintado...), control de los planos, etc.

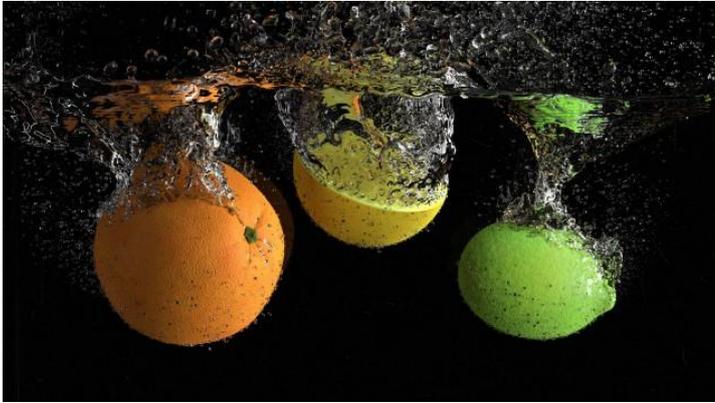
Bajo esto, se encuentra la línea temporal de controlador de secuencias y captación de frames junto con las opciones correspondientes al campo de la **animación**.



## ❖ 2.3. Características

**Blender** cuenta que con una serie de características que le han convertido en una de las herramientas más potentes del mercado a la hora de dar con un software que abarque diseño gráfico y animación en 3D. Algunas de estas características se listan y describen a continuación:

### Renderizado foto-realista: Cycles



· Nuevo tipo de renderizado conocido como “Cycles” o ciclos, que ofrece resultados áltamente sorprendentes y ultra-realistas a las renderizaciones que se realiza. Este renderizado está caracterizado la:

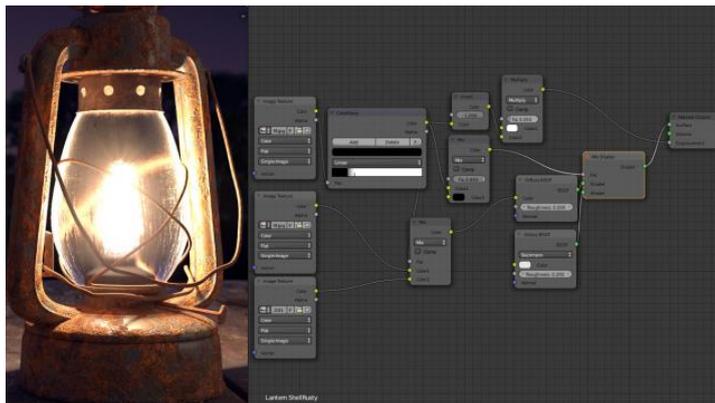
- Utilización de memoria y procesador del equipo más la tarjeta gráfica instalada a la hora de utilizar el motor gráfico.
- Gran ayuda con iluminaciones de altos rangos dinámicos (HDR).
- Visualización en tiempo real del renderizado a medida que el usuario realiza cambios en su trabajo evitando la necesidad de estar realizando la simulación de éste para observar los nuevos cambios.

### Modelado rápido



- Amplia gama de herramientas de modelado que permite crear, editar y transformar cualquier modelo con rapidez.
- Decenas de accesos directos para un manejo más cómodo y rápido a la hora de modelar.
- Fácil elaboración de polígonos de N-lados.
- Desplazamiento sencillo de trazos, unión y disolución de estos.
- Establecimiento de medidas precisas con uso de cuadrículas.
- Uso de scripts en código python para añadir herramientas personalizadas y addons.

### Materiales realistas

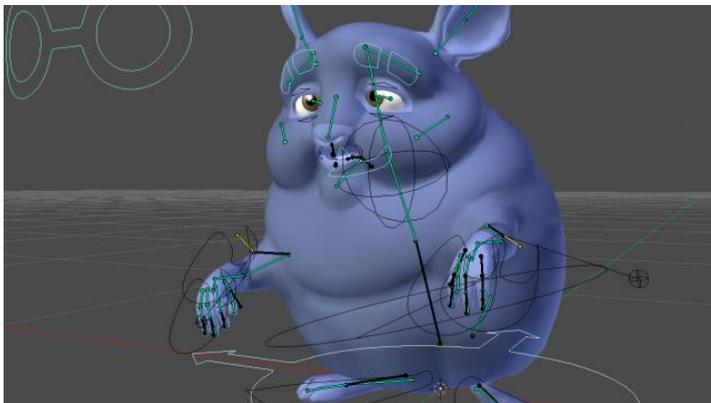


- Posibilidad infinita a la hora de crear materiales.
- Uso de nodos para una personalización más compleja del material.
- Adición de texturas físicamente precisas como el cristal, materiales traslucidos, etc.
- Opción de lenguaje de texturas de código abierto (OSL) para codificar materiales propiamente únicos.



### **Modelado rápido con cadenas de huesos (esqueleto virtual de un cuerpo)**

- Transformación de diseño de un modelo en personaje real con sencillos pasos.
- Extenso conjunto de herramientas que permiten:
  - Definición del exterior del modelo, esqueleto y automatización de la piel.
  - Estilizado y pintado del peso.
  - Función espejo.
  - Capas para zonas de huesos del cuerpo y grupos de colores para una fácil organización.
  - Trazos de curvas para los huesos para un sencillo manejo y control de estos.



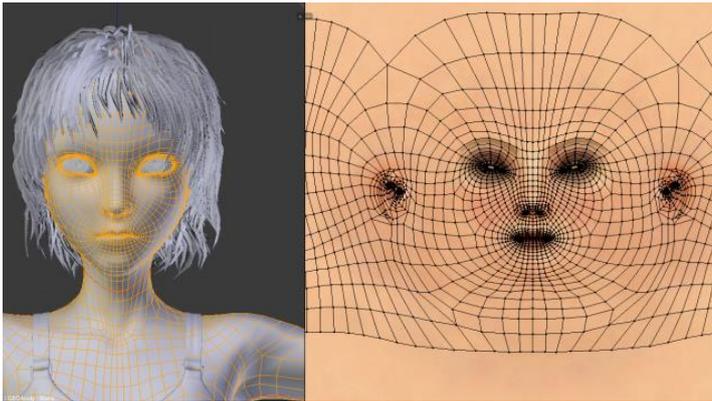
### **Set de herramientas de animación**

- Gran facilidad para convertir un personaje en una animación.
- Disposición de trayectos o caminos para simulación del personaje caminando.
- Editor animado de poses del personaje.
- Herramienta de animaciones no lineales para movimientos independientes.
- Opción de cinemática inversa para intuición de puntos finales de un movimiento.
- Sincronización de sonidos a un movimiento o punto de este en concreto.



### **Esculpido**

- Veinte tipos diferentes de pincel.
- Soporte de múltiples resoluciones en el esculpido.
- Topología dinámica de esculpido.
- Función espejo para esculpido.



### **Mapeado rápido de texturas**

- Desglose de una textura en mapa de cuadrículas a modo de plano para un fácil pintado o utilización de imágenes sobre esta.
- Distintos modos de desglose: cubo, cilindro, esfera y proyecciones de la cámara.
- Opciones de selección de desglose tales como realizarlo en base a un ángulo dado.
- Múltiples mapeados en capas.



### **Compositor completo**

- Modelado de cualquier tipo de objeto, idea o diseño sin necesidad de recurrir a terceros programas.
- Extensa librería de nodos para creación de efectos de cámara, grados de colores, viñetas, etc.
- Organización en capas de los renderizados.
- Múltiples mapeados en capas.
- Compatibilidad con ficheros de imagen y vídeo para modificación y trabajo de estos a los que pueden incluirse animaciones sobre estas.
- Habilidad para renderizar en múltiples capas ficheros en formato HDR.
- Capacidad de trabajar en múltiples tareas.



### **Simulaciones realistas**

- Herramientas de simulación de entre las que se incluyen:
  - Efectos reales de agua y fluidos.
  - Ondulaciones de humo producido por las llamas e interacción con el entorno.
  - Oscilación de cabellos con el mecer del viento e interacción con las colisiones.
  - Realismo del comportamiento de prendas de vestir ante movimientos, entorno, etc.
  - Aplicaciones de física realista haciendo que los objetos actúen a las colisiones y puedan llegarse a destruir
  - Creación de partículas para añadir efectos tales como lluvia, chispas, metralla, etc.

### Creación de juegos

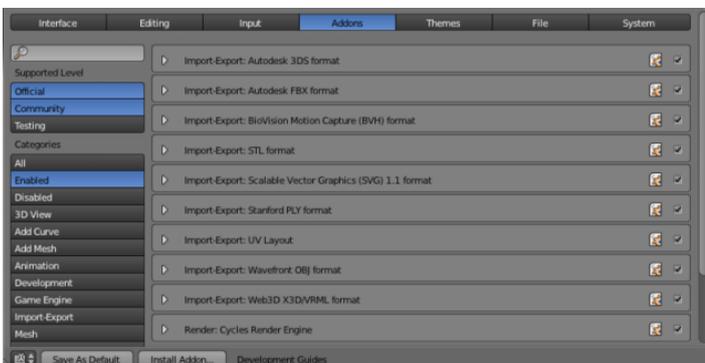


- Pack completo de herramientas para la creación de juegos 3D.
- Capacidad para portar modelos de cualquier otro motor de juegos de terceros a Blender.
- Creación o codificación de una lógica propia para el juego.
- Integración completa de física realista de balas.
- Codificación de scripts API en python para control avanzado y diseño de IA.
- Soporte tanto para todas las iluminaciones dinámicas de gráficos OpenGL, sombras de caracteres animados, y materiales animados como para el mapeado normal o Parallax (objetos que se mueven a distantes velocidades permitiendo aparentar una animación; ej: animación en 3D que puede verse sin necesidad de gafas).
- Reproducción del juego dentro de Blender sin necesidad de compilación o pre-procesado.
- Librerías de audio OpenAL para uso de sonido envolvente 3D.



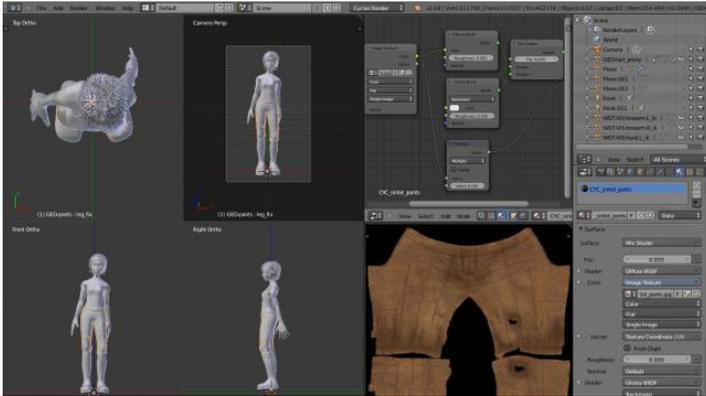
### Seguimiento de cámara y objetos

- Completa gama de opciones que elimina el recurrir a otros programas.
- Seguimiento manual y automático.
- Reconstrucción del seguimiento realizado.
- Visualización en tiempo real del seguimiento a pie o de la escena en 3D.
- Soporte para seguimiento de puntos fijos (Tripod tracking) o conjunto de puntos móviles definidos por el usuario (Planar tracking).



### Librería de extensiones

- Extensa librería de extensiones que el usuario puede encender o apagar a su elección. Algunas de estas extensiones incluyen:
  - Generador de árboles, terreno, hiedra y nubes.
  - Fracturación de objetos.
  - Herramientas de impresión en 3D
  - Rigidez en sistemas de cadenas de huesos.
  - Soporte de importación y exportación de formatos AfterEffects, DirectX, motor de juegos de Unreal, etc.



### Interfaz sencilla

- Personalización total de la interfaz a gusto del usuario, incluyendo el uso de scripts python.
- Posibilidad de división de ventanas.
- Consistencia de la interfaz personalizada en todas las plataformas al utilizar un fichero de un equipo a otro, p.e.
- Extensa librería de extensiones que el usuario puede encender o apagar a su elección. Algunas de estas extensiones incluyen:
  - Bloqueo de las ventanas emergentes de Windows.
  - Nitidez de texto.



### Formatos de fichero

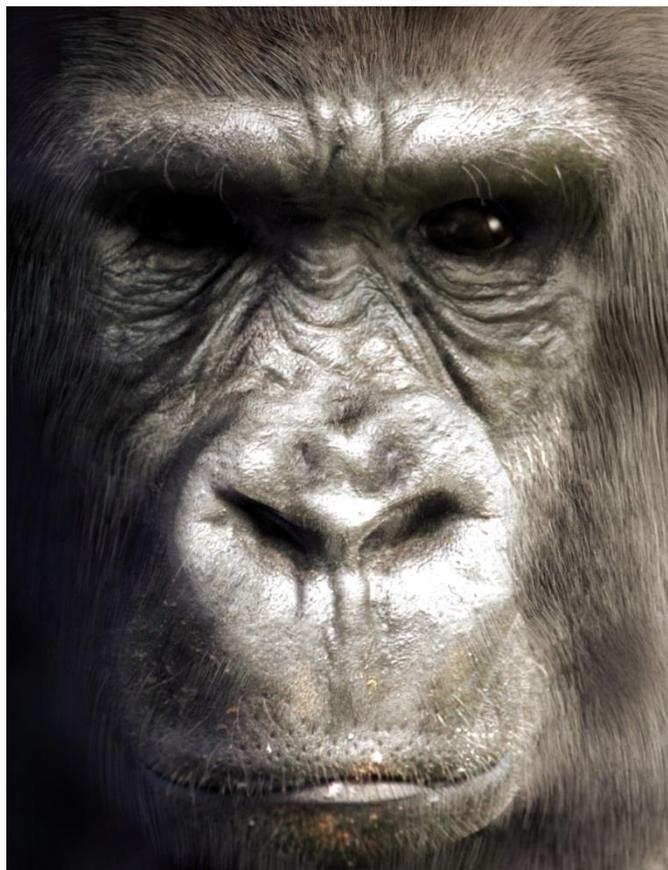
- Alta compatibilidad con los distintos tipos formatos de ficheros a la hora de importar o exportar un archivo:
  - Imagen: JPEG, JPEG2000, PNG, TARGA, OpenEXR, DPX, Cineon, Radiance HDR, SGI Iris, TIFF
  - Vídeo: AVI, MPEG and Quicktime (on OSX).
  - 3D: 3D Studio (3DS), COLLADA (DAE), Filmbox (FBX), Autodesk (DXF), Wavefront (OBJ), DirectX (x), Lightwave (LWO), Motion Capture (BVH), SVG, Stanford PLY, STL, VRML, VRML97, X3D.

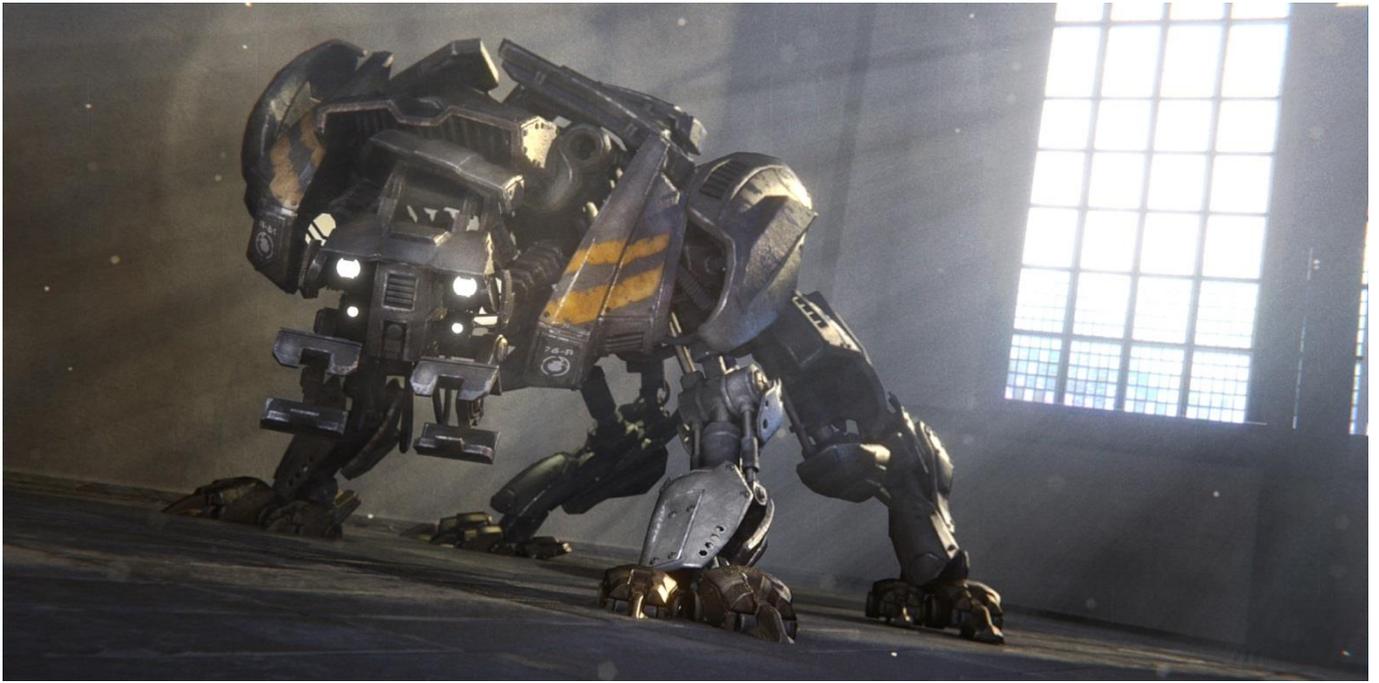
## ❖ 2.4. Proyectos y diseños realizados con Blender

Una vez conocidas las principales características de Blender, el modo más rápido para llegar a comprender hasta qué punto es capaz de llegar el potencial de este, es siempre por medio de imágenes.

### ✚ Imágenes estáticas realistas y ultra-calidad

A través de las siguientes imágenes puede observarse unos pocos ejemplos de lo que es capaz de crearse por medio de Blender y el detalle casi real con el que son capaces de diseñarse algunos modelos:





🌈 *Diseño de interiores, exteriores e iluminación*

*Recrear un espacio interior y aparentar que es una fotografía real y no el modelado de un programa de diseño gráfico, es realmente posible gracias a las herramientas de diseño realista y sus opciones de iluminación.*

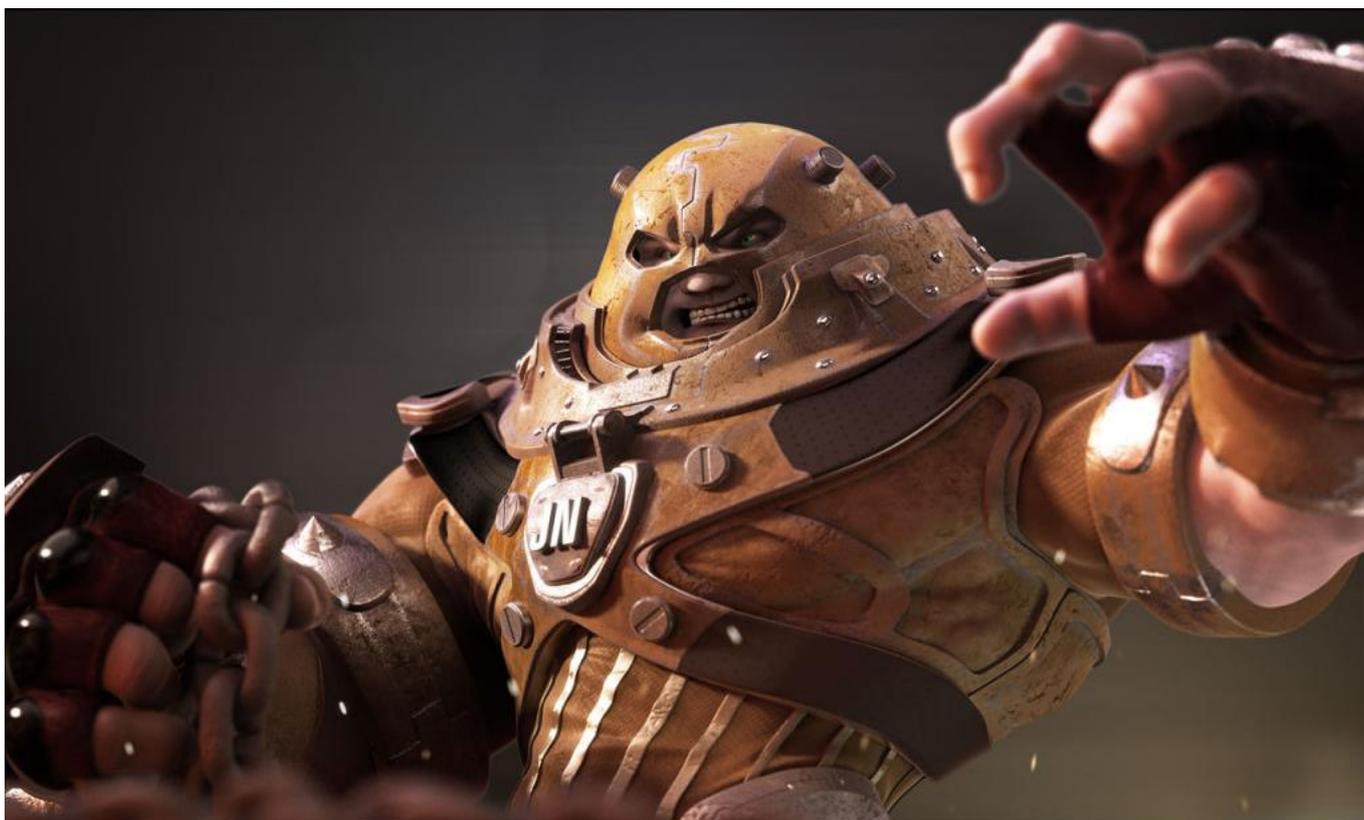


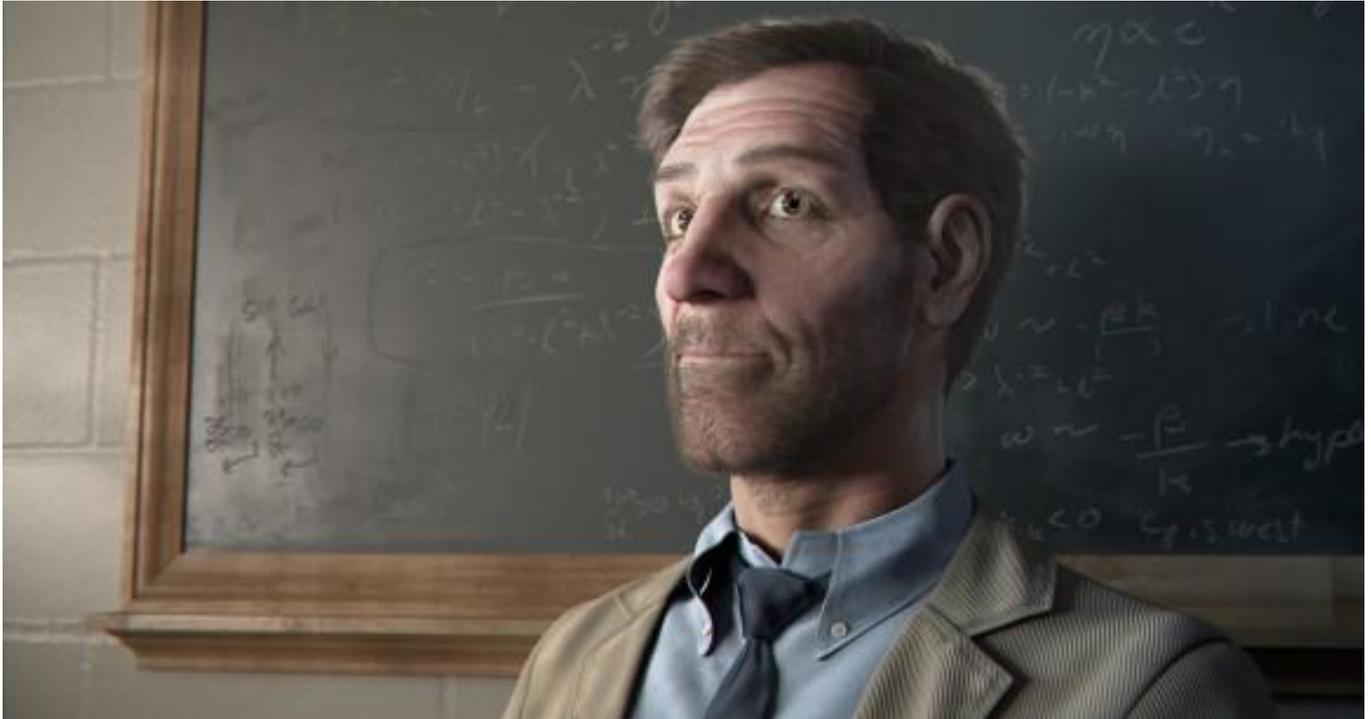


## 🌈 Definición de personajes e incorporación a la animación

Como se ha mencionado dentro de las características, el modelado de personajes no resulta tarea compleja gracias a que cuenta con un sencillo set de herramientas para la creación de estos, como utilizar un esqueleto virtual o cadenas de huesos como base al que méramente hay que añadirle una “cobertura corpórea”.

La altísima definición de estos y el sorprendente resultado de cómo puede quedar dentro de una animación, salta a la vista:





Actualmente se está trabajando en diversos proyectos abiertos de animación entre los que destacan la pura animación y incorporación de animación a grabaciones reales:

Dos de estos proyectos más famosos de hoy en día, en los que se puede observar un ejemplo de cada, uno son:

- *Sintel*: <http://www.sintel.org/>



➤ Tears of Steel: <http://mango.blender.org/>



### 🎮 Diseño de videojuegos

Por último, el añadido de un set de herramientas para la creación de videojuegos ha convertido a Blender en una de las opciones más potentes del mercado, pues viendo hasta qué punto es capaz de llegar el nivel detalle, no es de extrañar que los resultados sean realmente sorprendentes, como puede verse en las creaciones de algunos usuarios tales como "Al grito de Guerra" que puede observarse en la última imagen:





## ❖ 2.5. Blender para simular sensores de fibra óptica

A primera vista, resulta una idea innovadora tratar de simular y operar con elementos ópticos a través de un programa gráfico de diseño en 3D. Es lógico pensar de antemano que no llegue a ser capaz de trabajar de la misma forma a cómo lo haría uno de ámbito profesional específico de ello, pero al fin y al cabo ese es el objetivo de este proyecto: intentar demostrar que es posible reflejar un caso de la vida real. Puede parecer algo inalcanzable pero no por ello ha de ser imposible.

Sin tratar con el programa, se prevén de antemano dos puntos clave o dificultades con los que se ha de lidiar y que en el caso de que estos no lleguen a solventarse, limitarían enormemente las posibilidades de éxito de este proyecto, determinando de que el programa es apto o no para realizar este tipo de simulaciones. Estos puntos clave son:

- Que los materiales modelados no se asemejen realmente a lo que puede ser una fibra de cristal real.
- Que la luz no llegue a comportarse adecuadamente en el núcleo/cubierta provocando que en la fibra modelada no llegue a producirse la reflexión total interna.

De nada serviría trabajar con un material distinto al cristal aunque simule el comportamiento de una fibra, pues al fin y al cabo no podrían recrearse situaciones reales. A su vez tampoco, valdría que la luz atravesase los materiales sin reflejarse adecuadamente en estos pues haría que, la luz observada en los extremos de la fibra, sea considerablemente inferior o incluso nula.

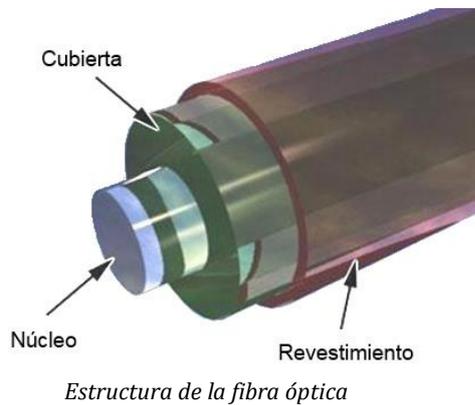
## Capítulo 3.- Pruebas realizadas

### ❖ 3.1. Fibra óptica: parámetros y comportamiento

Antes de proceder a realizar diseño alguno, se ha de conocer cómo se comporta una fibra óptica de cristal, los principios o fenómenos que se desarrollan en ella y ante todo, los parámetros y/o medidas precisas que la definen pues son estos los que consiguen que su funcionamiento sea el adecuado.

#### ➤ Composición

Una fibra óptica está compuesta por un filamento de sílice, cuarzo o plástico de un determinado diámetro y otro de mayor diámetro que recubre a este primero. Al filamento interior se le denomina **núcleo** y al externo **cubierta**. Finalmente, se le añade un revestimiento a modo de funda para proteger ambas partes internas.



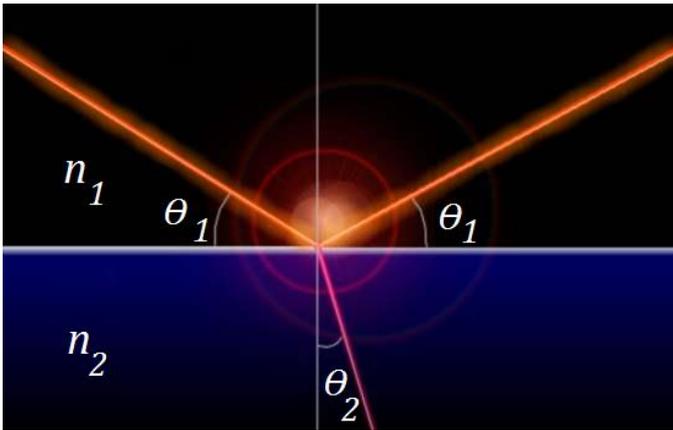
Cada uno de estos filamentos está elaborado con un determinado índice de refracción de la luz siendo el del núcleo mayor que el de la cubierta (generalmente 1.65 para el núcleo y 1.55 para la cubierta).

Cuando la luz incide sobre el borde de una superficie que limita con otra de menor índice de refracción, esta se refleja cuanto mayor sea el ángulo de incidencia y la diferencia entre los índices. Al darse el caso opuesto, que la luz proceda de un medio con un índice menor, como puede ser el aire cuyo valor se estima en 1, es posible que el medio con mayor índice recoja este haz de luz y permita su propagación por él.

#### ➤ Reflexión Total Interna (TIR)

Cuando un haz luminoso incide sobre un medio con un índice de refracción menor ( $n_2$ ) que el medio en el que se encuentra ( $n_1$ ), cabe la posibilidad de que la luz se refracte y parte de esta viaje por este nuevo medio. En el caso de que la refracción sea inexistente y la luz permanece retenida en el medio en que se encuentra, se produce entonces el llamado fenómeno de la **Reflexión Total Interna**.

Para un entendimiento más cómodo y visual de este fenómeno, se añade la imagen siguiente en la que se puede apreciar: el haz incidente, su reflejo, su refracción hacia el otro medio y los distintos parámetros que caracterizan este fenómeno.



$n_1$  = índice de refracción del medio en que se encuentra el haz de luz.

$n_2$  = índice de refracción del medio sobre el que incide el haz de luz.

$\theta_1$  = ángulo de incidencia y reflejo del haz de luz.

$\theta_2$  = ángulo del haz de luz refractado.

Todos estos parámetros al fin y al cabo tienen un denominador común en cuanto a su aplicación: la **Ley de Snell**. A través de esta fórmula es posible conocer el ángulo de luz que se refracta hacia el medio con menor índice de refracción. La fórmula descrita, es la siguiente:

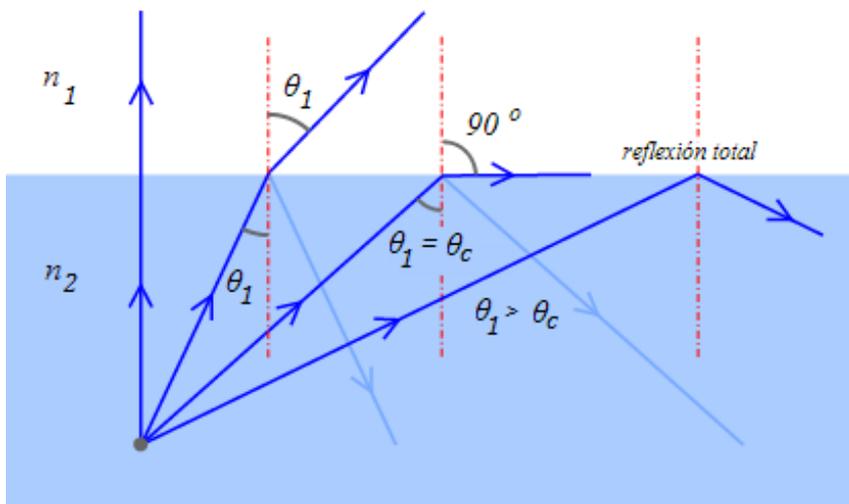
$$n_1 \cdot \text{sen}(\theta_1) = n_2 \cdot \text{sen}(\theta_2)$$

Fórmula de la Ley de Snell

Si  $n_1 > n_2$ , entonces  $\theta_2 < \theta_1$ . Esto significa que cuanto más aumenta  $\theta_1$ , más fácil resulta que  $\theta_2$  llegue a los  $90^\circ$  antes que  $\theta_1$ . Cuando  $\theta_2 = 90^\circ$  ( $\pi/2$ ), el haz de luz sale paralelo a la superficie de separación de entre los dos medios. Alcanzado este punto, a medida que se siga aumentando  $\theta_1$  (dado que  $\theta_2$  no puede ser mayor que  $90^\circ$ ) no se produce transmisión hacia el otro medio por lo que la luz se refleja totalmente. En este punto es donde se hace mención al **Ángulo crítico**. Cuando el ángulo de incidencia  $\theta_1$  es mayor que el ángulo crítico  $\theta_c$ , se produce entonces la reflexión total interna.

$$\theta_c = \text{arcsen}(n_2 / n_1)$$

Fórmula del ángulo crítico



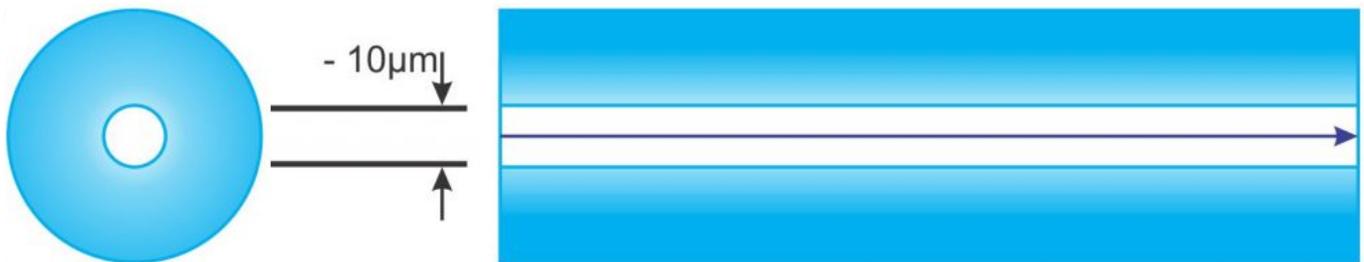
➤ *Modos de propagación de la luz*

En función de la modalidad de propagación de la luz de la fibra, pueden llegar a distinguirse dos tipos:

- Fibra monomodo

Este tipo de fibras sólo permiten un único modo de propagación haciendo que la transmisión sea paralela al eje de la fibra. Para lograr esto es preciso reducir el diámetro del núcleo de forma considerable, hasta alcanzar medidas en torno a las 8-10 micras.

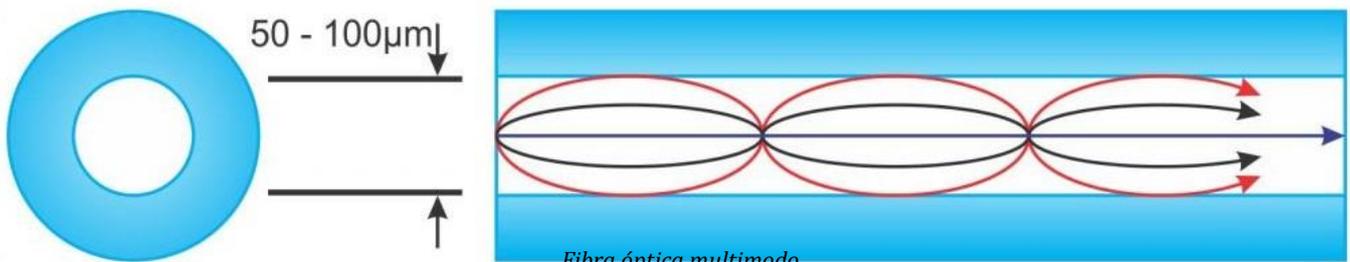
Las fibras monomodo se utilizan habitualmente para largas distancias pudiendo alcanzar varios miles de km dado que al transmitirse únicamente un modo de luz, se sufre menos el fenómeno de la dispersión (efecto que produce un haz de luz cuando atraviesa un material). A su vez, la tasa de información que puede transmitirse es mucho mayor al verse una menor atenuación con la distancia.



Fibra óptica monomodo

- Fibra multimodo

Este tipo de fibras permiten propagar más de un modo de luz, pudiendo transmitir hasta más de mil de estos. Para permitir semejante propagación, es necesario que el diámetro del núcleo sea mucho mayor, entre 50 y 100 micras. Debido al gran tamaño de este núcleo, resulta más fácil de conectar a otros componentes y cuenta con una mayor tolerancia a componentes de menor precisión. Se utiliza comúnmente en aplicaciones de corta distancia, no superiores a los 2km.



Fibra óptica multimodo

En cuanto al índice de refracción, el núcleo cuenta con un valor mayor que la cubierta, solo que ha de ser del mismo orden de magnitud. En función de tipo de índice de refracción del núcleo, se destacan dos tipos de fibras multimodo:

- Índice gradual: el índice de refracción en este tipo de fibra no es constante dado que decrece a medida que se acerca a la cubierta. La banda de paso puede llegar hasta los 500 MHz/km. La principal ventaja de este tipo de fibras es que permite reducir la dispersión que sufren los modos que se propagan a través de esta.
- Escalonado o de salto de índice: el índice en este tipo de fibra se mantiene constante en toda la sección cilíndrica provocando que del núcleo a la cubierta haya un “salto” de un valor a otro; de ahí su nombre. La banda de paso llega hasta los 40MHz/km. En este caso la dispersión modal es mayor.

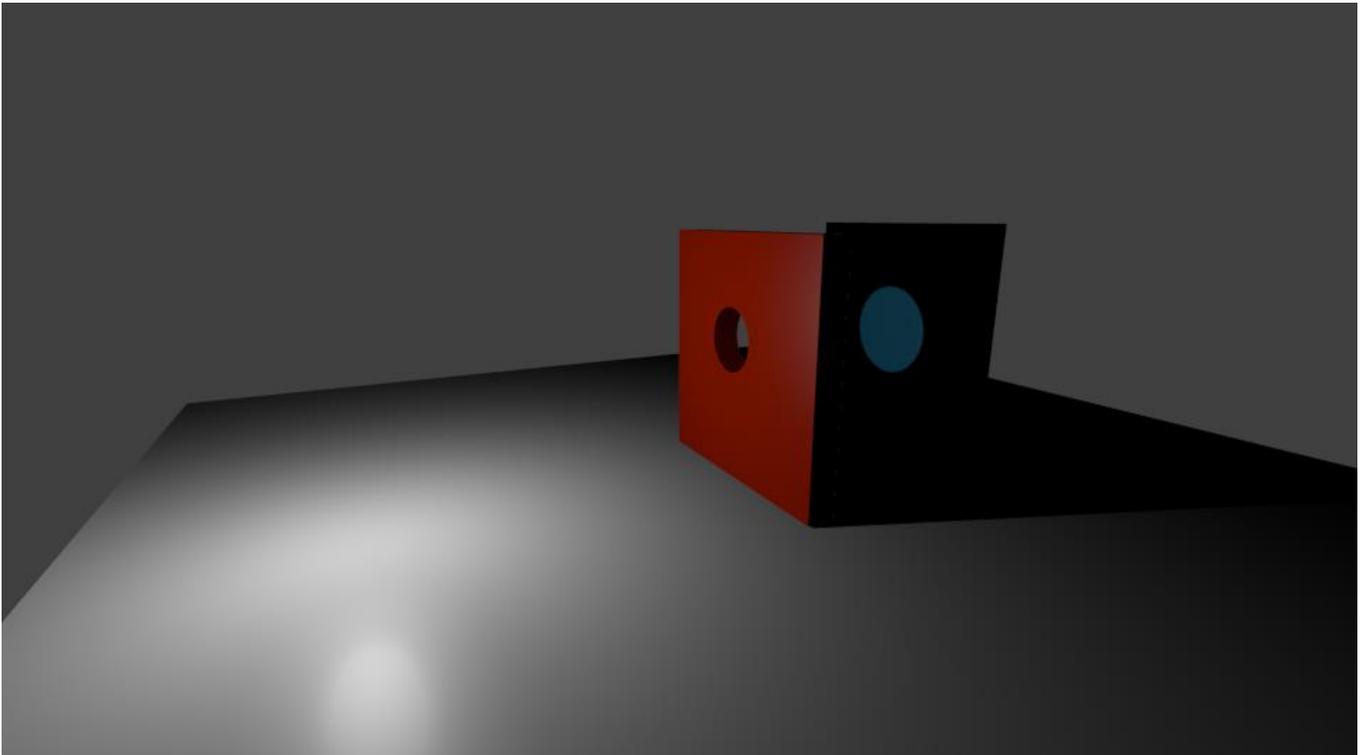
Para lograr los objetivos de este trabajo, es importante que el software de simulación sea capaz de modelar correctamente la estructura de la fibra óptica, en nuestro caso, fibras multimodo con perfil de salto de índice.

## ❖ 3.2. Blender: Ideas y diseños

### ➤ Render básico

Como inicio y para dar a conocer el funcionamiento básico de la luz, se trabajó primeramente en un diseño sencillo con dos paneles rectangulares seguidos uno detrás de otro, a los que a uno de ellos se le realizó orificio a modo de ventana para que pudiese pasar la luz y observar cómo se producía el traspaso de un material a otro.

Este primer diseño no supuso dificultad alguna y se pudo visualizar cómo se generaba sombra en la totalidad del material, por encontrarse tapado por uno delantero, salvo en una pequeña zona por la que la luz había conseguido adentrarse gracias al orificio del primer material.



Traspaso de luz de un elemento a otro por medio de un orificio.

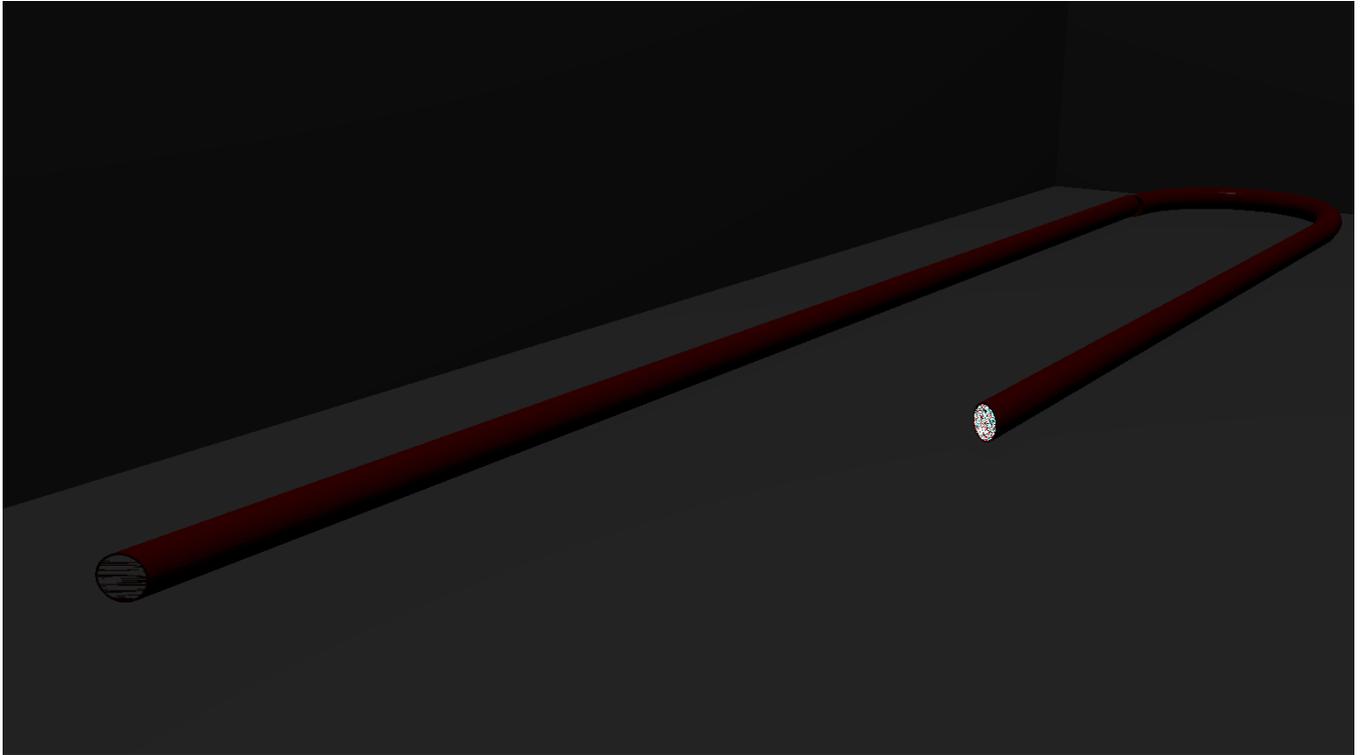
Visto esto, se propuso crear un cilindro doble sin ningún grosor en particular, simulando una fibra, para entender cómo comportaría la luz dentro de éste.

Se realizó el diseño de un material interior que se asemejara al cristal estableciendo entre otros un parámetro **alpha** adecuado, el cual recrea la transparencia del material y la cantidad de luz que es capaz de atravesar a este. Se le indicó un índice de refracción de 1.6 y que tuviese la propiedad de reflejar la iluminación a modo de espejo, por lo que al esconder la

fuelle de luz en el interior de este, se observaría el comportamiento de las paredes internas a la hora de funcionar con la iluminación.

Posteriormente se intentaría trabajar con la cubierta para observar si sería posible el diseño de una fibra óptica real y que pudiese darse el caso de la reflexión total interna.

El resultado que se observó para este primer diseño fue el siguiente:

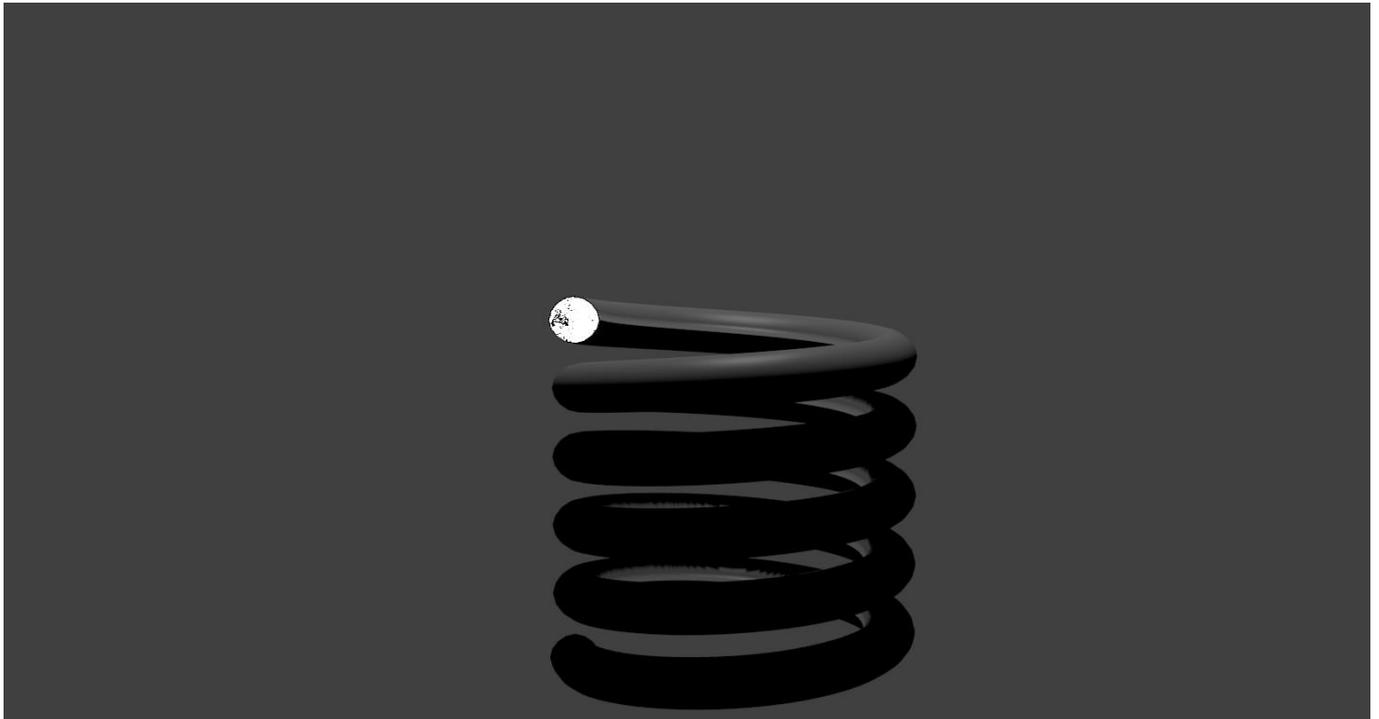


*Reflejo de luz por el interior de un cilindro simulando cristal estando la fuente de luz oculta en el extremo izquierdo de la fibra.*

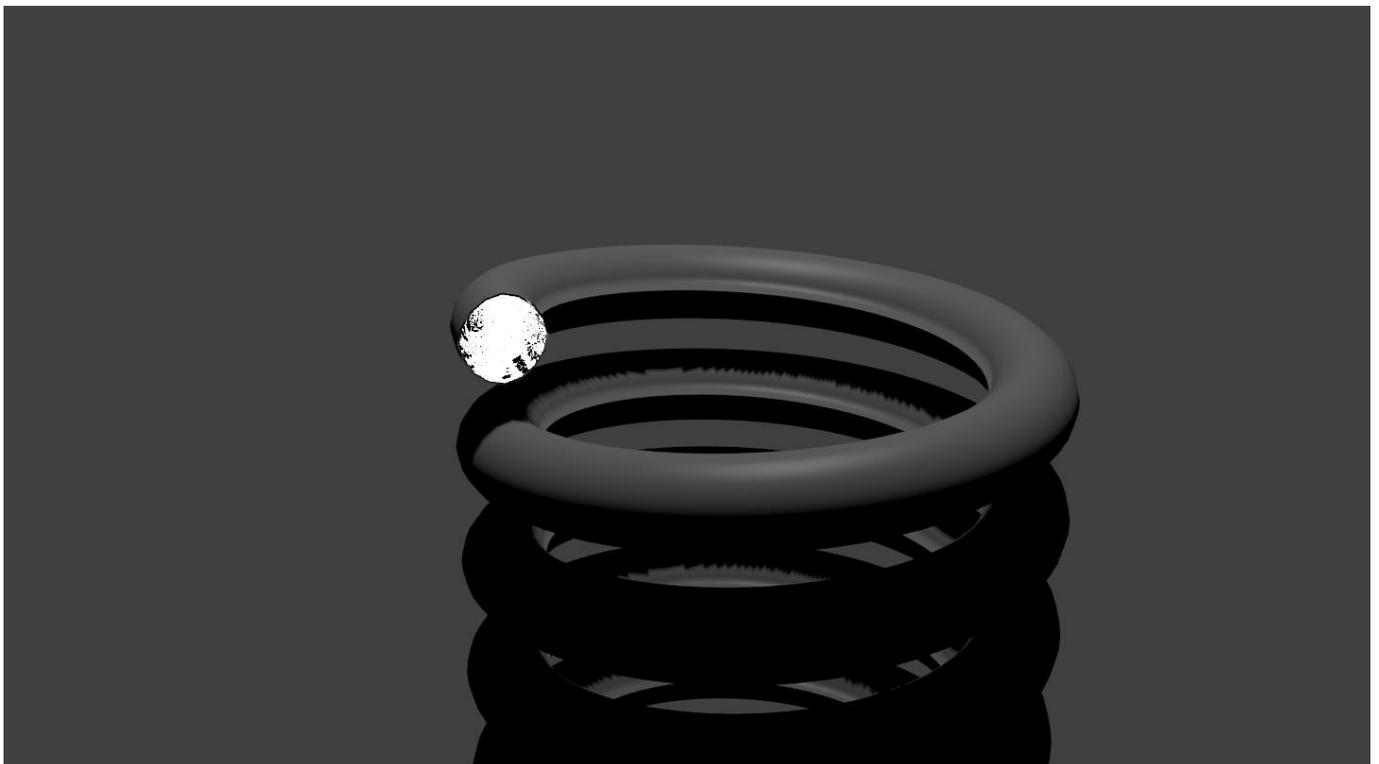
Como primera impresión, el resultado daba buenas expectativas dado que sentaba las bases de un buen inicio, pero no reflejaba verdaderamente el comportamiento de una fibra real. La cubierta al fin y al cabo era de un material opaco, el cilindro interior si fuese de cristal realmente, debía reflejar la luz sin la necesidad de habilitar una opción de espejo y los diámetros del núcleo y cubierta debían ser cantidades acordes a las cifras utilizadas hoy en día. De esta forma podría producirse la reflexión total interna posibilitando que saliese mucha más luz por el otro extremo.

Sin embargo, una parte de la idea fundamental que ha regido todo este proyecto siempre ha sido esclarecer si es realmente posible diseñar una fibra en hélice y que circule luz por el interior de esta, por lo que antes de ir un paso más allá se probó esta idea siguiendo el modelo planteado en el diseño anterior. Se modificaron diversos parámetros tales como la potencia luminosa de la fuente, los diámetros de las fibras, y el valor de  $\alpha$  del cilindro interior entre otros. La fuente de luz se situó dentro del cilindro interior en el tramo intermedio de la espiral para poder observar la luz saliente en ambos extremos de la fibra.

Para sorpresa, el resultado fue más satisfactorio tal y como puede verse en las imágenes siguientes pero seguía dándose el problema comentado en el diseño anterior.



*Vista genérica del diseño de la fibra*



*Vista del extremo superior de la fibra*



Vista del extremo inferior de la fibra

Viendo las posibilidades que se lograban alcanzar con esta idea, se puso en marcha el paso siguiente de lograr esto mismo pero con condiciones más apropiadas tales como utilizar núcleo y cubierta de cristal para ir perfeccionando el diseño y lograr en su punto final el resultado tan anhelado.

### ❖ Problemas y dificultades del renderizado básico

El modelado de la fibra con cubierta y núcleo de cristal no supuso problema alguno en cuanto a su diseño, sin embargo sembró la semilla de lo que pudo ser una incontable serie de dificultades a la hora de trabajar con ello.

Al intentar renderizar el diseño en cuestión, comenzaron a contabilizarse las horas en cuanto al tiempo de espera frente a lo que debía ser una cuantía aceptable de minutos (10min). Tras muchos cambios y búsquedas de soluciones para aligerar semejante carga, se vio que el renderizado básico no era capaz de trabajar con semejantes especificaciones. Se procedió a tratar con el renderizado avanzado: **cycles**.

### ❖ Renderizado avanzado: Cycles

Uno de los primeros pasos que se dieron para fijar un inicio, fueron buscar ejemplos e información sobre posibles diseños de fibra óptica por parte de otros usuarios para:

- 1º Llegar a comprender de un modo más técnico como trabajaban e interferían en el diseño muchas de las nuevas opciones con las que contaba este renderizado tras la lectura de varios manuales y tutoriales.
- 2º Ver si algún usuario había conseguido llegar a modelar algo similar.

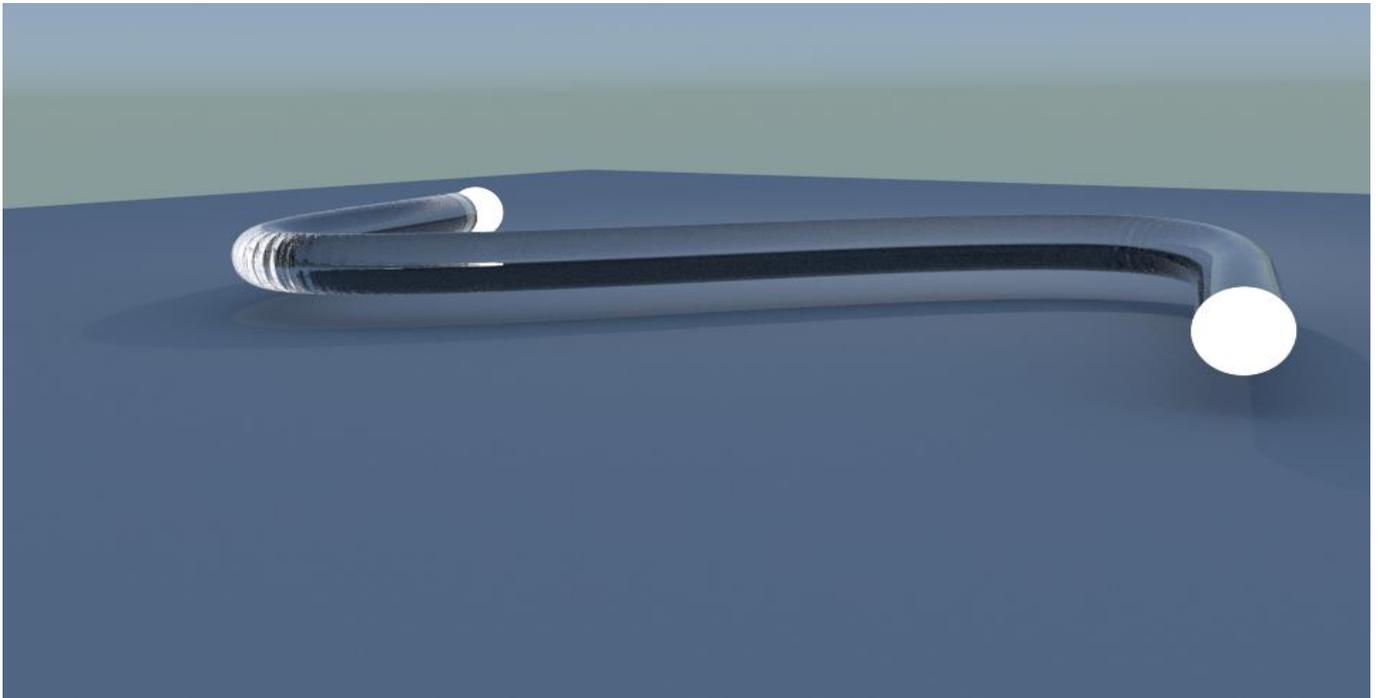
Dicha búsqueda aportó grandes noticias y una muy mala. Las buenas eran que muchos usuarios habían conseguido recrear fibras ópticas visualmente reales en las que podía observarse como la luz viajaba desde un punto inicial a uno final por medio de lo que parecía una fibra de cristal o plástico. La mala es que los diseños que modelaban los usuarios, eran puramente “fakes” pues ninguno recreaba una fibra real. Se establecía por medio de opciones que la luz recorriese un cuerpo de extremo a extremo, se diseñaban “pelos” para simular lámparas de fibra a los cuales no se les daba ningún

parámetro en concreto como el índice de refracción o se utilizaban meramente “tubos de cristal” para estos propósitos sin el añadido de una cubierta, entre otros muchos casos. Estos detalles rompen con la idea propuesta que es demostrar que en una fibra modelada, se cumpla como es debido, el fenómeno de la reflexión total interna, hecho que a la vista parecía cumplirse ya que estaba trucado (“fake”).

En vista de esto, se procedió a modelar la misma idea planteada en el diseño anterior con la diferencia de contar con otras herramientas. Este nuevo renderizado permite modelar elementos con una calidad gráfica mucho mayor que el utilizado en los primeros diseños. Muchas de las opciones existentes en el primer renderizado desaparecen tales como la personalización de parámetros para la fuente o la caracterización del material al mínimo detalle entre otras. A la hora de definir un material se abre una lista de elementos predefinidos que imitan y/o simulan el comportamiento de estos en la realidad.

Como punto inicial se diseñó una fibra curvada en dos tramos para observar el comportamiento de esta ante la exposición de una fuente de luz esférica a la entrada de la fibra. Las medidas utilizadas para el núcleo-cubierta de la fibra fueron 50/125  $\mu\text{m}$  y de 100  $\mu\text{m}$  de diámetro para la fuente de luz. El índice de refracción para la cubierta fue de 1.5 mientras que el índice del núcleo fue de 1.6. Con la definición de estos parámetros y el funcionamiento interno del programa a la hora de trabajar con un elemento de tipo “cristal” se pudo ver que el comportamiento de la fibra modelada era correcto,

El resultado final del modelado con este nuevo renderizado puede observarse en la siguiente imagen.



Intensidad luminosa recibida con fuente de en uno de los extremos de la fibra.

Modelada ésta, quedó ceñirse a la idea original: diseñar una fibra de cristal en hélice en que pudiese observarse un comportamiento similar o igual al de una fibra real y visualizar al aplicar distintas fuentes de luz, si la intensidad luminosa recibida al final de ésta se corresponde con los estudios realizados existentes.

Tras la modificación de diversos parámetros y activación de varias opciones logró obtenerse el resultado deseado, dejando únicamente por realizar el trabajo citado en el párrafo anterior.

Todos estos resultados pueden verse de forma más explícita en los puntos correspondientes del capítulo 5.

# Capítulo 4.- Estimación luminosa utilizando Matlab

## ❖ 4.1. Introducción

Dentro de este capítulo se hablará del uso de Matlab para la obtención de la intensidad luminosa total recibida en uno de los extremos de la fibra. Para esto, se facilitará a un código escrito, las capturas de imagen de los distintos renderizados realizados para su posterior cálculo.

Matlab, de entre sus muchas funciones, cuenta con un paquete de herramientas para el tratado y procesado de imágenes. Gracias a estas herramientas, es posible realizar ediciones en la imagen, restaurar porciones ruidosas o degradadas, analizar texturas, extraer información, etc.

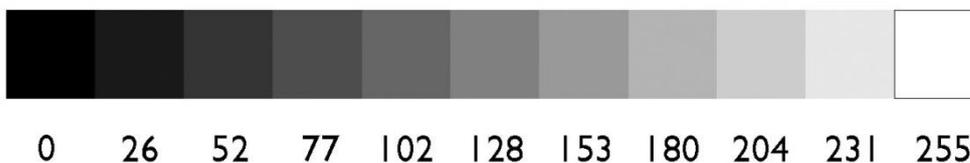
## ❖ 4.2. Objetivos a lograr con la ayuda de Matlab

Realizado el modelado de la fibra en el capítulo anterior y comprobado que existe reflexión total interna, se tratarán dos puntos importantes en este capítulo para esclarecer si el comportamiento de la fibra diseñada logra asemejarse al de una real.

- Observar las variaciones luminosas que se producen ante los desplazamientos de la fuente de luz.
- Demostrar que se recibe luz (aunque resulte inapreciable) en los extremos de la fibra cuando la fuente de luz incide sobre las paredes interiores. Objetivo que ha recogido todo este proyecto. En el caso de darse afirmativamente este punto, la fibra modelada llegaría a emular el caso citado del laboratorio.

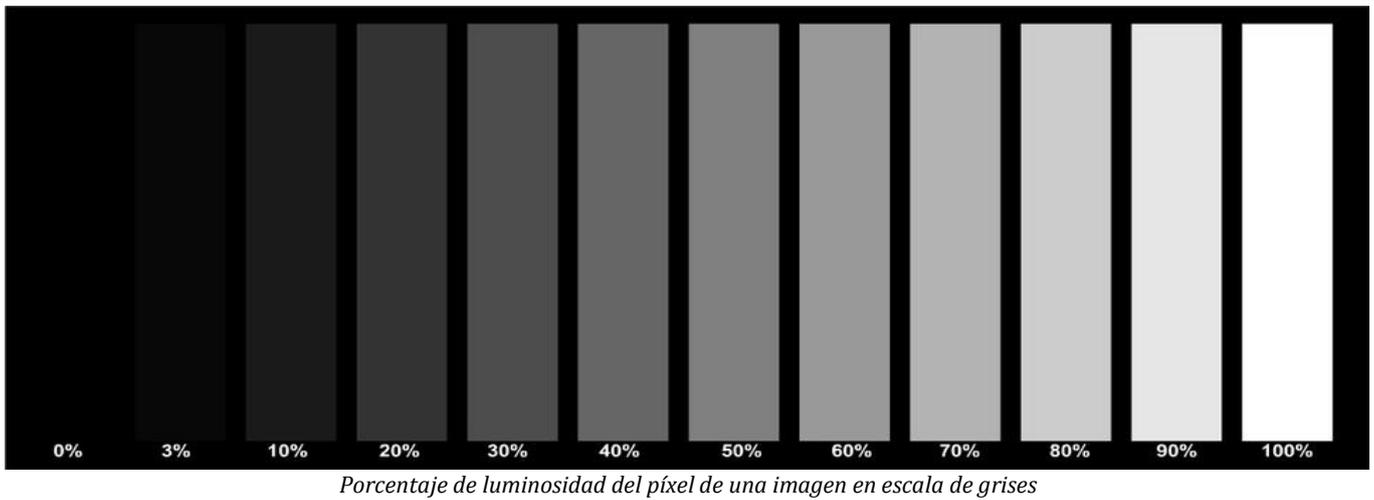
## ❖ 4.3. Definición del código en Matlab para cálculo de intensidad luminosa total

Matlab al igual que muchos otros programas de procesado de imágenes, tratan la luminosidad de un pixel dentro de una escala de grises, como un valor entre 0 y 255, siendo 0 el negro y 255 el blanco.



Escala gradual del valor de luminosidad en una escala de grises

Matlab, a la hora de trabajar con estos valores cuando se trata de intensidad luminosa, utiliza en su lugar con una escala de 0 a 1, representando desde el 0% al 100% de luminosidad del respectivo pixel.



Para obtener entonces la intensidad luminosa total de la imagen facilitada será preciso extraer el valor de intensidad luminosa de cada uno de los píxeles presentes en la imagen y realizar la suma de todos estos.

A continuación, se añade el código utilizado para el cálculo de este valor:

```
%% CÁLCULO DE INTENSIDAD LUMINOSA DE UNA IMAGEN

% Reseteo de parámetros
close all; clear all; clc;

% Lectura y obtención de la matrix de píxeles en valores RGB
imagen=imread('nombre del fichero de la imagen.png');

En Matlab una imagen en formato de color RGB, se representa por tres matrices
bidimensionales correspondientes a los planos R, G, B.

% Visualización de la imagen
imshow(imagen)

% Verificación de las dimensiones de la imagen
[m,n,p]=size(imagen);

Permite conocer las dimensiones de la imagen y el número de matrices RGB.

% Obtención de la matrix de píxeles en valores HSI
hsi=rgb2hsv(imagen);

Realiza el cambio a valores HSI para tratar con los valores adecuados.

%% Obtención de valores HSI de la imagen
H=hsi(:,:,1); % Tono
S=hsi(:,:,2); % Saturación
I=hsi(:,:,3); % Intensidad*

Obtiene cada matrix por separado dado que el parámetro hsi sigue siendo una matrix
de submatrices.

%% Obtención de la intensidad luminosa total de la imagen
suma=sum(sum(I))

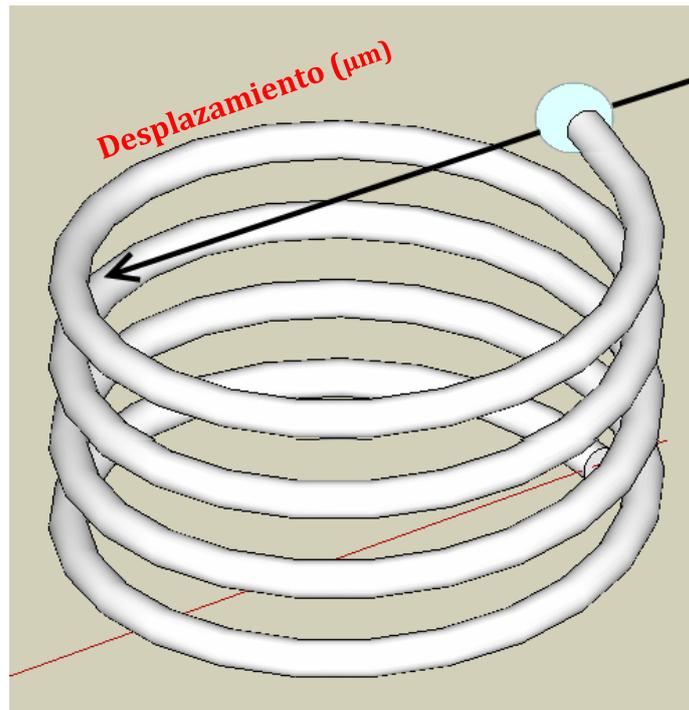
Calcula la suma de todos los elementos de la matrix I dando como resultado la
intensidad luminosa total de la imagen
```

En el próximo apartado se abordará el primer punto descrito y la resolución de este.

#### ❖ 4.4. Variación de luz recibida en función de la posición de la fuente

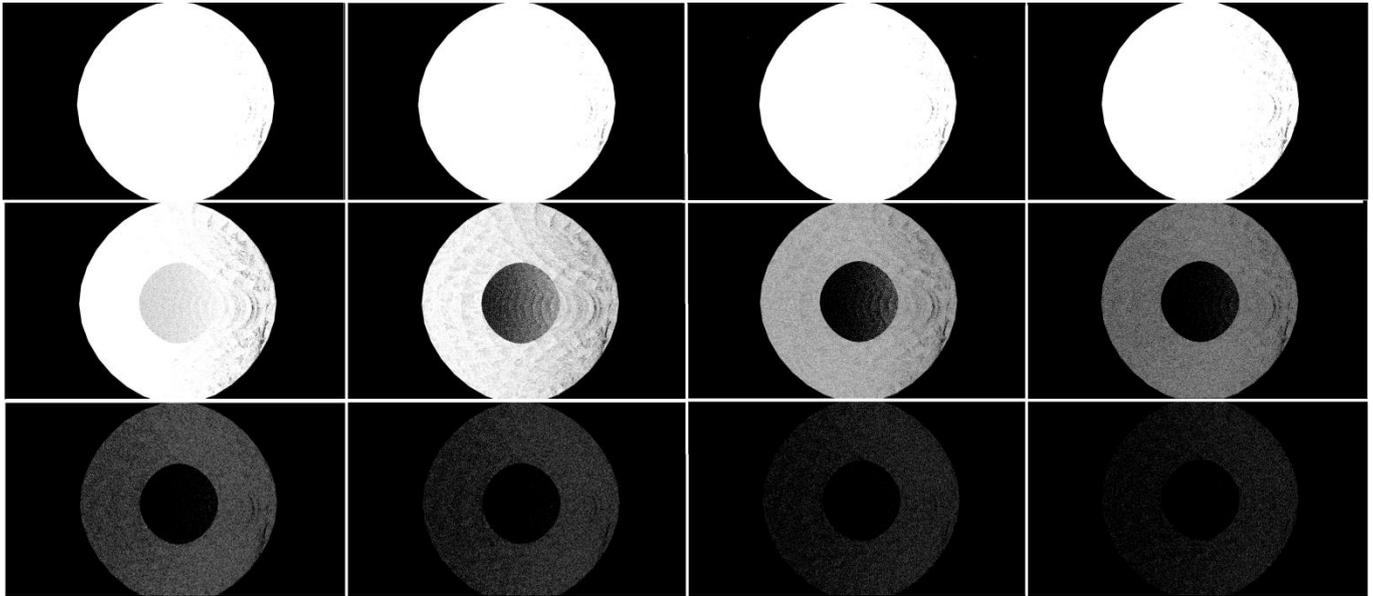
Descrito el programa, se procede a facilitarle a Matlab las imágenes a analizar a través de la segunda línea de código. Como primer paso se realiza una captura del renderizado en la salida de la fibra cuando la fuente de luz se encuentra en la propia entrada de esta. Esta circunstancia marcará el valor máximo de intensidad luminosa que se podrá observar en la salida.

Obtenida esta primera de captura de pantalla, se establece realizar desplazamientos de  $20\ \mu\text{m}$  a la fuente de luz a lo largo del eje horizontal para observar las variaciones que se producen en la salida de la fibra realizando las capturas oportunas. De esta forma, se recoge el incremento o decremento de la radiación luminosa con respecto al desplazamiento realizado.



Esquema de desplazamiento de la fuente emisora.

Realizando cada una de las capturas al renderizado obtenido, se consigue la siguiente serie de imágenes en la que se aprecia el decaimiento de la luminosidad a medida que se aleja la fuente.

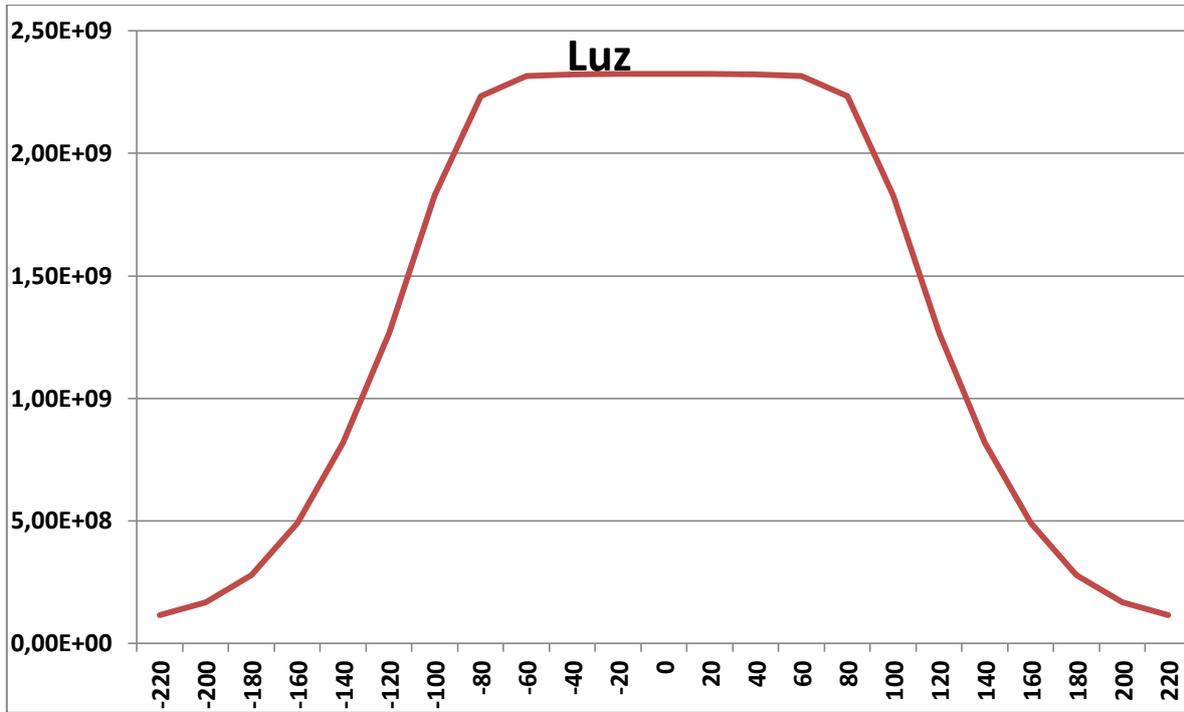


Capturas realizadas de los renderizados en función del desplazamiento de la fuente.

Para establecer una estimación, si se pasan cada una de estas imágenes al código escrito en Matlab, se obtienen una serie de valores que determinan la luz percibida al final de la fibra. Estos, no poseen una unidad de medida en sí (se establece una unidad arbitraria) pero sirven para cuantificar de forma relativa esta radiación luminosa. Tomando como punto de referencia la entrada de la fibra (desplazamiento 0  $\mu\text{m}$ ), a medida que se han ido realizando los desplazamientos, se obtiene lo siguiente:

Desplazamiento ( $\mu\text{m}$ )	Luz recibida (u.a.)	Desplazamiento ( $\mu\text{m}$ )	Luz recibida (u.a.)
-200	1,67E+08	20	2,32E+09
-180	2,79E+08	40	2,32E+09
-160	4,92E+08	60	2,32E+09
-140	8,20E+08	80	2,23E+09
-120	1,27E+09	100	1,83E+09
-100	1,83E+09	120	1,27E+09
-80	2,23E+09	140	8,20E+08
-60	2,32E+09	160	4,92E+08
-40	2,32E+09	180	2,79E+08
-20	2,32E+09	200	1,67E+08
0	2,33E+09	220	1,17E+08

Pasando estos valores a una gráfica:



Como puede observarse, existe un tramo relativamente ancho de intensidad luminosa recibida en torno a la posición inicial dado que la fuente emisora es una esfera de diámetro  $100\ \mu\text{m}$ . Al tratarse de una esfera la apertura numérica de ésta es considerablemente elevada lo que provoca que puedan darse haces de luz incidentes en la entrada de la fibra aun estando a una gran distancia.

En el siguiente apartado se quiere dar respuesta al objetivo que se ha perseguido a lo largo de todo este proyecto.

#### ❖ 4.5. Comprobar la propagación de luz hacia la salida simulando el caso del laboratorio

Llegado a este punto, se pretende comprobar si la fibra modelada a través de Blender ha logrado comportarse y recrear exactamente el mismo caso ocurrido durante el trabajo de laboratorio.

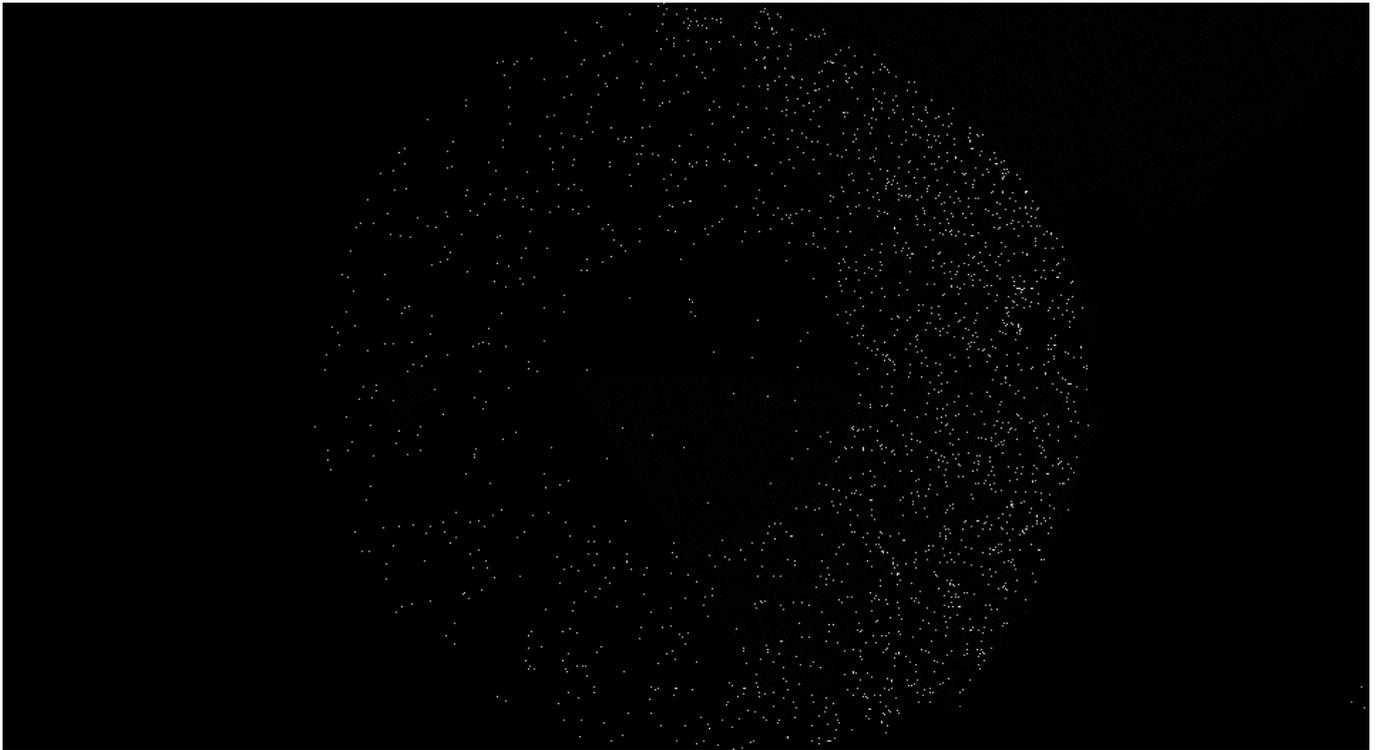
Según este trabajo de laboratorio, la fibra utilizada para captar el haz de luz a través de las paredes internas de la hélice, conseguía propagar una inapreciable cantidad de luz hacia las salidas de la fibra. Según el aspecto teórico, esto no podría llegar a darse, mas sin embargo se produce.

Se realiza entonces las renderizaciones necesarias para obtener las capturas a pasar al programa de Matlab.

Una de ellas, situando la fuente de luz a una distancia altamente considerable, por tener una referencia de un valor aproximado a 0. Al pasar esta captura al código se obtiene como cantidad de luz recibida: 0.0157

La segunda, situando la fuente de luz en el interior de la hélice.

La captura resultante de este segundo caso es la siguiente. La imagen ha sido editada dándole mayor intensidad a los píxeles dado que, como hemos mencionado anteriormente, la luz recibida al final de la fibra es prácticamente inapreciable. De este modo, puede visualizarse de una forma más nítida.



*Pasando esta imagen al código de Matlab se obtiene, que la intensidad luminosa total que se puede observar en la salida es:*

*7.018.745*

*A raíz de este resultado, se deduce que se percibe luz al final de la fibra, produciéndose el mismo caso que el ocurrido durante el trabajo de laboratorio.*

*Tras este estudio y, en base a este resultado, se llega a la conclusión de que sí es posible plantear, recrear o trabajar con elementos ópticos utilizando un programa de diseño en 3D sin la necesidad de recurrir a un software de ámbito profesional.*

## Capítulo 5.- Resultados

---

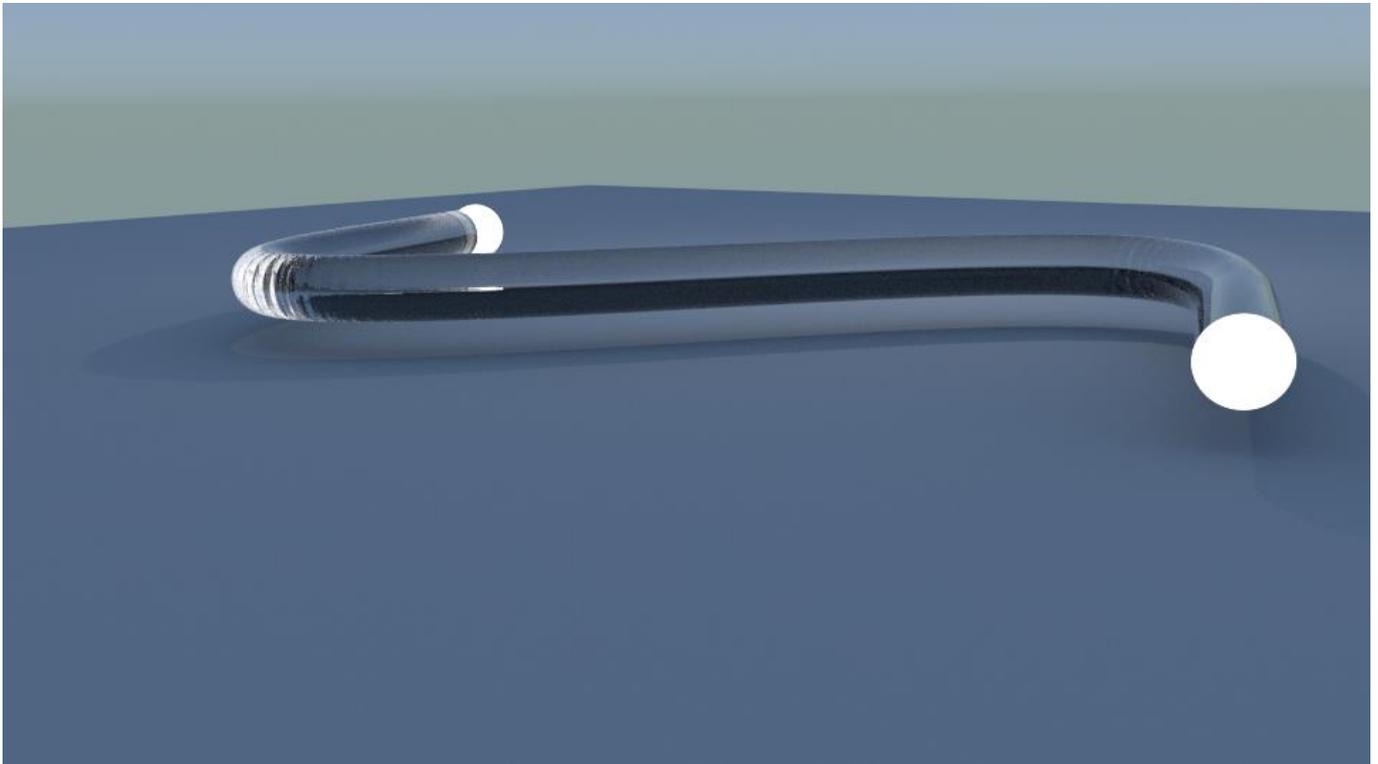
### ❖ 5.1. Introducción

Dentro de este capítulo se recogen los resultados obtenidos hasta ahora tanto en el modelado de la fibra de cristal multimodo de salto de índice por medio de Blender, como el cálculo de intensidad luminosa total por el código de Matlab.

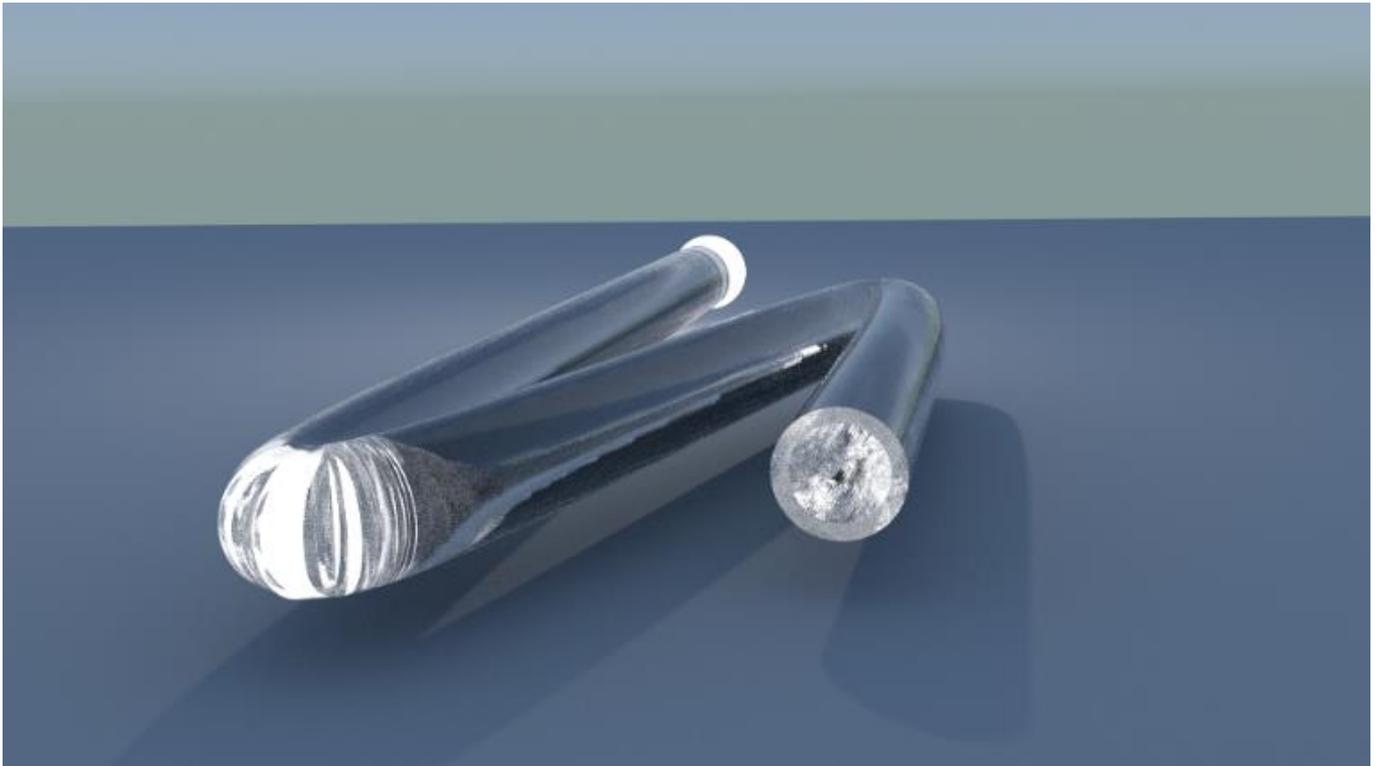
### ❖ 5.2. Modelado de fibra de cristal en Blender

Gracias a la utilización de uno de los renderizados avanzados con los que cuenta Blender, cycles, ha podido lograrse el punto que más se ha perseguido en este proyecto: llegar a modelar una fibra multimodo de cristal de salto de índice en la que se produce el fenómeno de reflexión total interna.

Las imágenes siguientes muestran el modelado de una fibra de cristal con medidas 50/125  $\mu\text{m}$  curvada en dos tramos que se utilizó antes de proceder al diseño en hélice para visualizar, como primer paso, la atenuación producida a medida que aumentaban los radios de curvatura. Se aprecia que a mayor radio de curvatura, menor es la intensidad luminosa recibida en el extremo saliente de la fibra correspondiéndose con su comportamiento real.

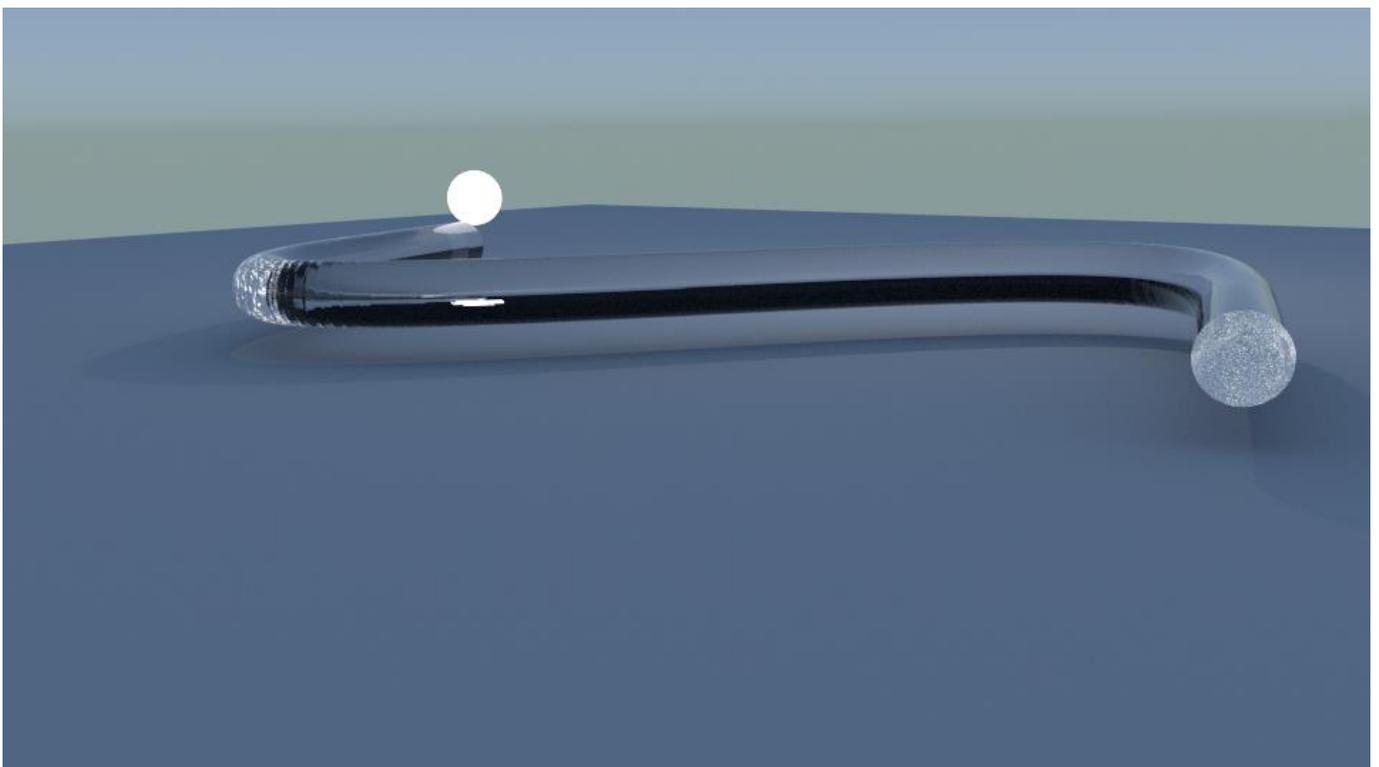


Luz percibida a la salida a al curvar en dos tramos la fibra modelada.

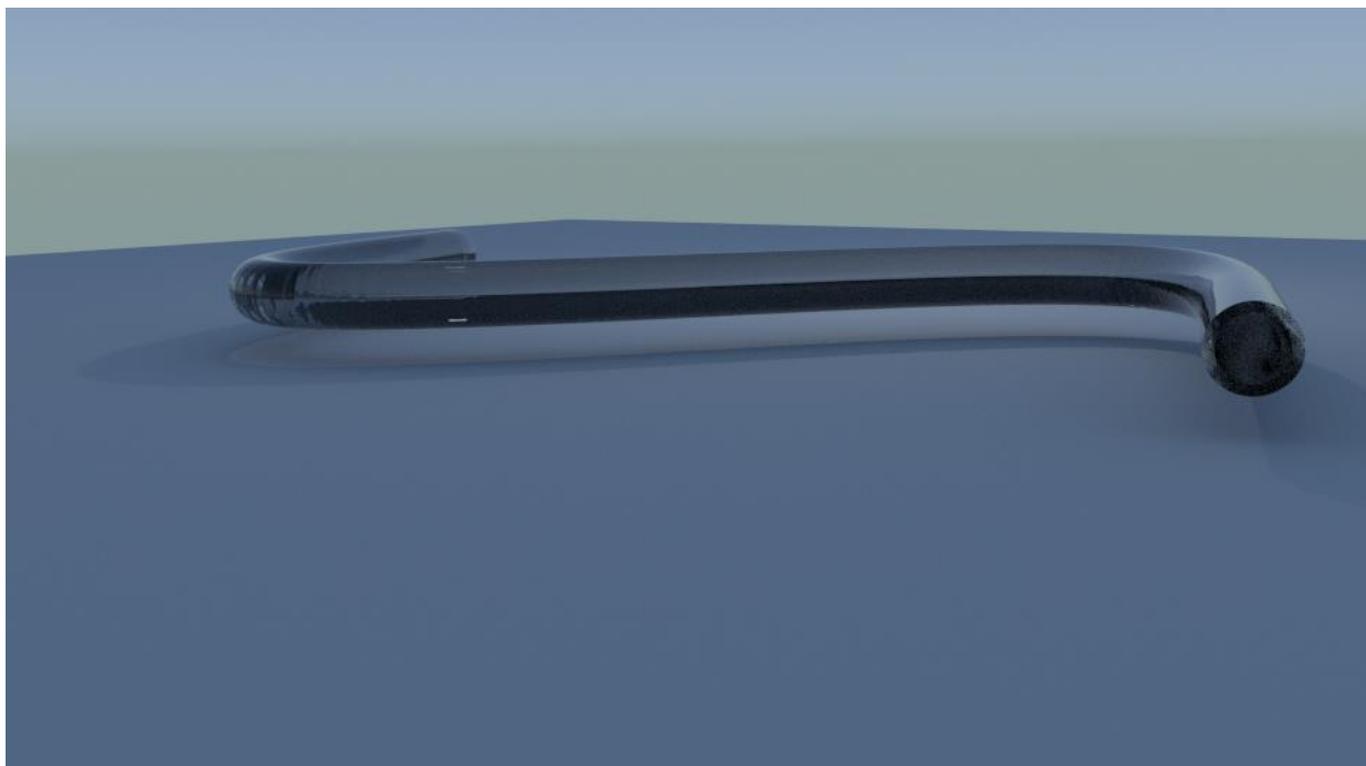


*Luz percibida a la salida al incrementar el radio de curvatura.*

*Se da paso entonces a observar la variación de intensidad luminosa recibida en función del desplazamiento de la fuente de luz.*

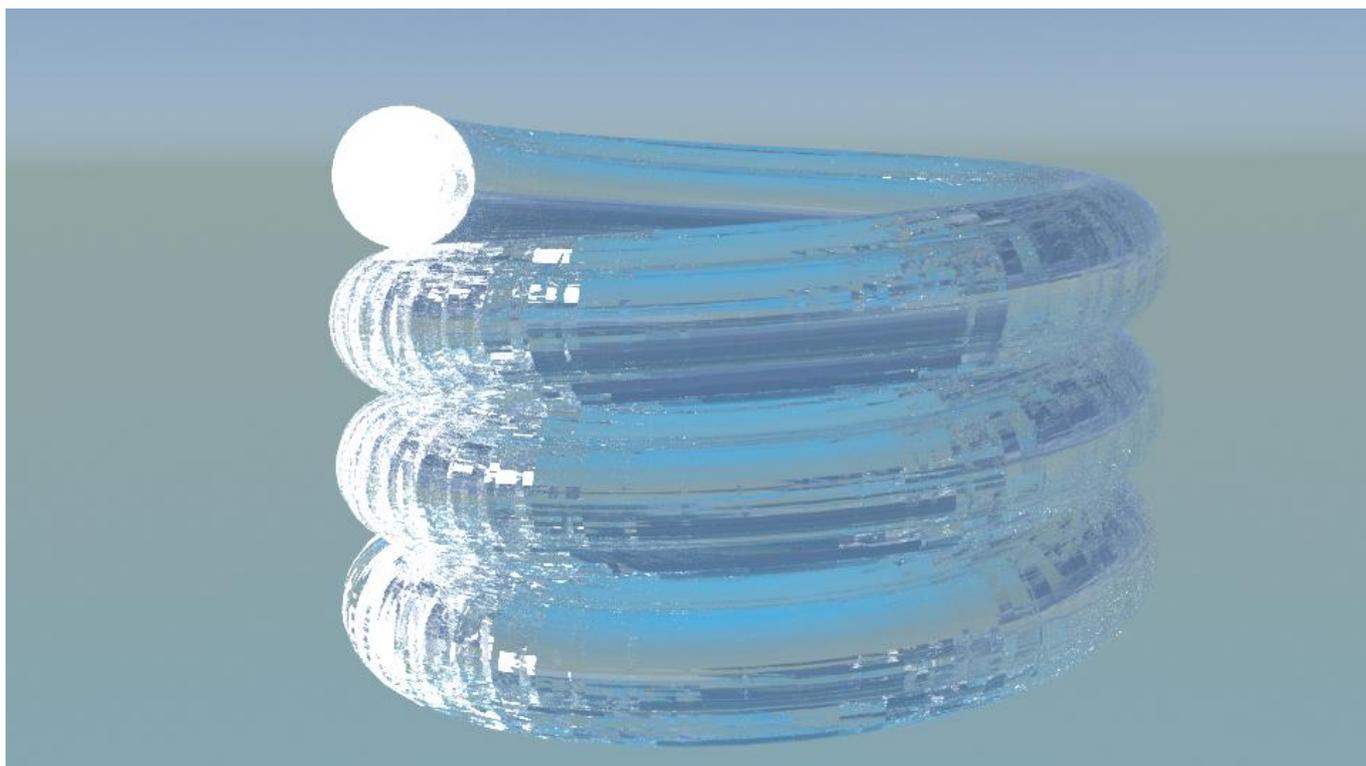


*Luz percibida a la salida con fuente de luz desplazada una corta distancia.*



*Luz percibida a la salida con la fuente emisora una gran distancia.*

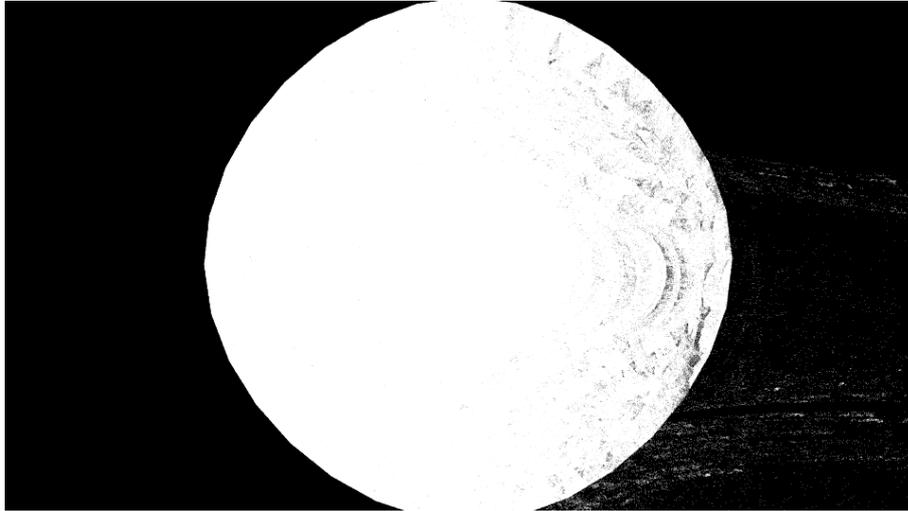
*Cuanta mayor es la distancia de la fuente de luz con respecto a la entrada de la fibra, menor es la recepción de haces de luz por parte de esta (influyendo factores tales como la luz ambiente, etc.) provocando que propague una menor cantidad de intensidad luminosa. Conseguido esto se utiliza el mismo tipo de fibra para su diseño en hélice y ver así su comportamiento en si captura luz o no. La imagen inferior muestra el resultado de ello.*



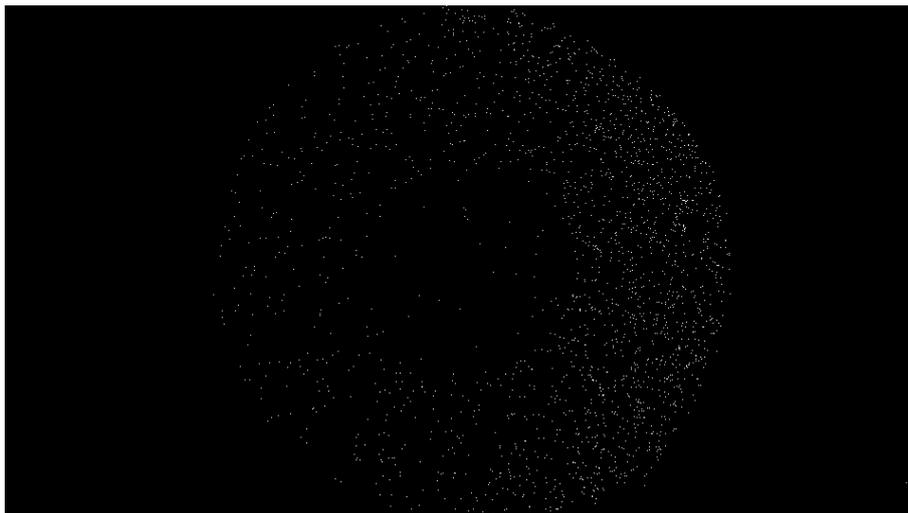
*Modelado de fibra multimodo de salto de índice en hélice con fuente situada en la entrada inferior.*

Retirada la luz ambiente, se comprueba si el caso de laboratorio es capaz de lograrse en la fibra modelada. Para esto, como se describió en el capítulo 3, se sitúa una fuente de luz en uno de los extremos de la fibra y se observa la intensidad luminosa recibida en el extremo saliente. A su vez se realiza una captura del renderizado obtenido para su posterior utilización en la estimación de radiación luminosa por medio de Matlab.

Tomada esta, se posiciona entonces la fuente de luz en el interior de la hélice para comprobar si los haces de luz incidentes sobre la cubierta de la fibra son capaces de propagarse por el núcleo de la fibra hasta el extremo saliente. Nuevamente se realiza una captura del renderizado obtenido. Cabe añadir que en el caso de la fuente de luz situada en el entorno interior de la hélice, la luz que se captura es prácticamente inapreciable. Por ello para una más cómoda visualización dentro de este apartado, se ha modificado la captura de imagen realizada otorgando mayor intensidad a los haces recogidos. De esta forma podrán llegar a verse con menor dificultad. Los resultados recogidos se añaden a continuación.



*Percepción luminosa recogida con fuente de luz en entrada de la fibra.*



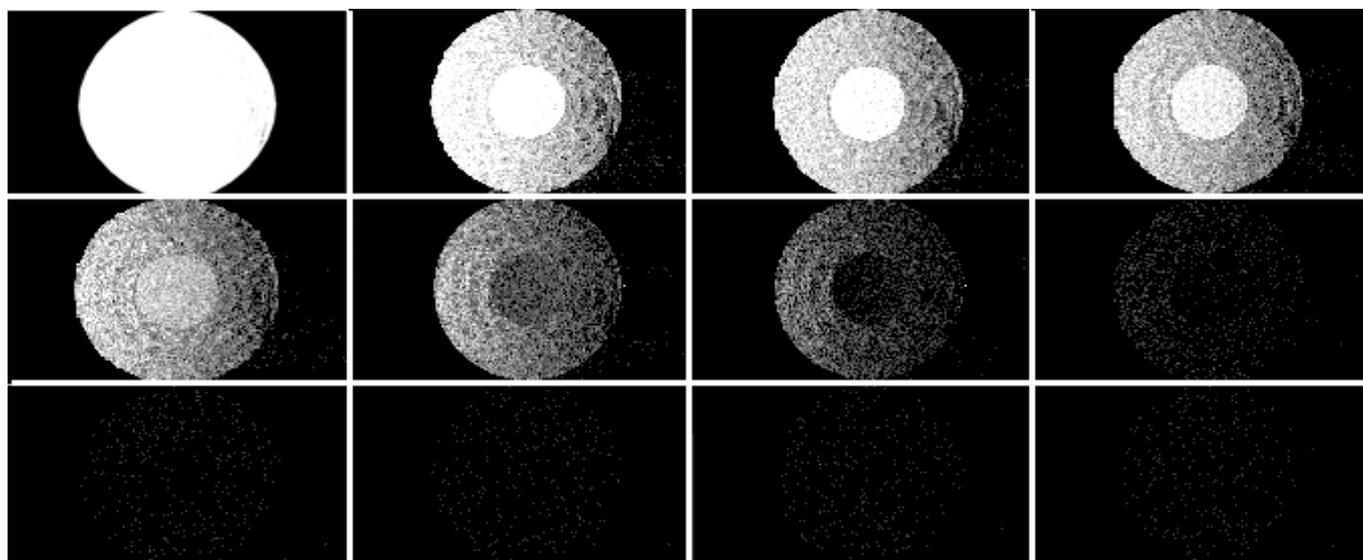
*Percepción luminosa recogida con fuente de luz en el interior de la hélice.*

Con ello, queda demostrado que es posible modelar elementos ópticos con Blender en los que llegan a cumplirse principios físicos propios de estos tales como transmisión de la luz, atenuación, refracción, reflexión, etc. y se producen fenómenos como la reflexión total interna entre otros.

❖ 5.3. Obtención de estimación luminosa por Matlab

Para comprobar de forma más exacta que la fibra modelada trabaja adecuadamente, se ha sustituido la fuente en esfera, por una fibra de cristal con dimensiones 100/125  $\mu\text{m}$  la cual actuará como emisor. De este modo, resulta más cómodo realizar la comparativa con las mediciones experimentales de una fibra convencional de hoy en día, como por ejemplo para esta comparación, las medidas de acoplo de una fibra de 100  $\mu\text{m}$  de diámetro de Ocean Optics.

Al igual que en lo descrito en el apartado 4 del capítulo 4, se realizan las capturas de los renderizados resultantes en función de la posición de la fuente emisora.

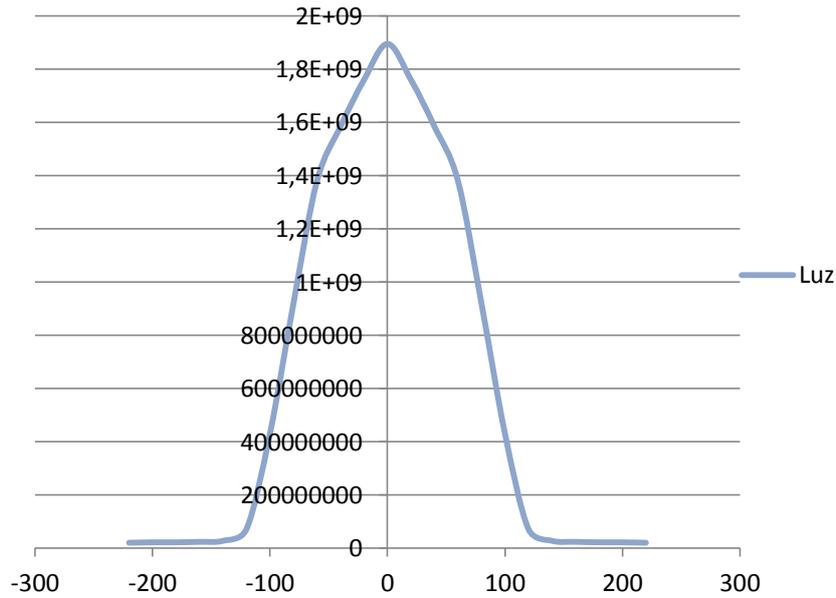


Capturas realizadas de los renderizados en función del desplazamiento ante la emisión de luz de una fibra de cristal.

Conseguidas las capturas de los renderizados, se pasan estas al código de Matlab para obtener una estimación de la radiación luminosa percibida a la salida de la fibra.

Desplazamiento ( $\mu\text{m}$ )	Luz recibida (u.a.)	Desplazamiento ( $\mu\text{m}$ )	Luz recibida (u.a.)
-200	2,27E+07	20	1,76E+09
-180	2,28E+07	40	1,58E+09
-160	2,46E+07	60	1,38E+09
-140	2,84E+07	80	9,21E+08
-120	7,27E+07	100	4,31E+08
-100	4,31E+08	120	7,27E+07
-80	9,21E+08	140	2,84E+07
-60	1,38E+09	160	2,46E+07
-40	1,58E+09	180	2,28E+07
-20	1,76E+09	200	2,27E+07
0	1,89E+09	220	2,12E+07

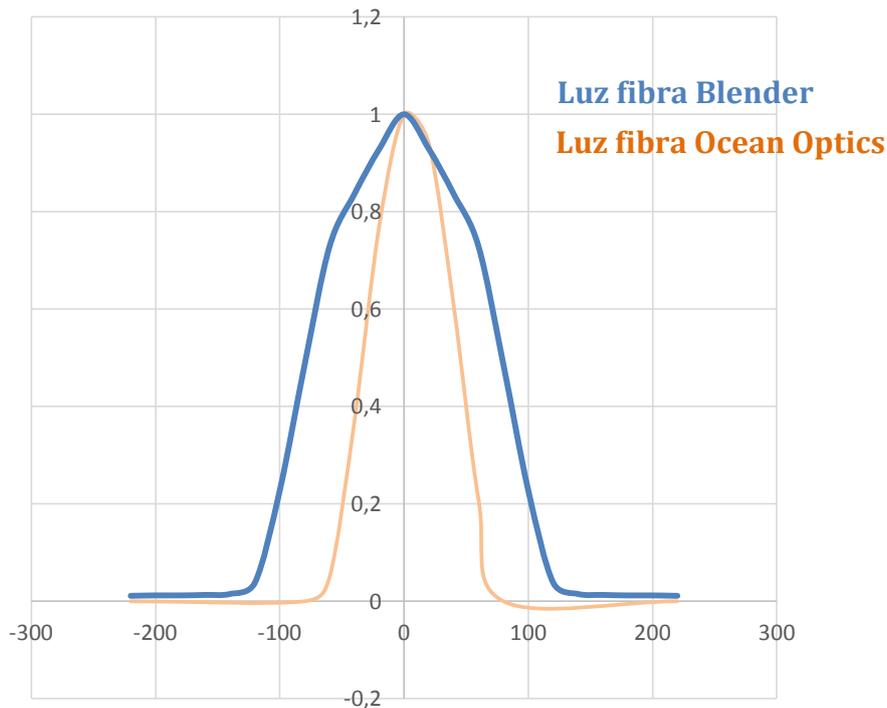
Representados gráficamente estos valores:



Gráfica resultante de los valores de luz percibida al final de la fibra

Como puede apreciarse, al ser una fibra óptica la fuente emisora, la apertura numérica de ésta es considerablemente mucho menor que en el caso visto de la esfera, por lo que las mediciones obtenidas se aproximan mucho más a las de una fibra convencional.

Realizando entonces una comparativa, a partir de las gráficas, entre las medidas experimentales de la fibra de 100  $\mu\text{m}$  de diámetro de Ocean Optics y las últimas obtenidas en nuestro ensayo, se puede apreciar lo siguiente:



Comparativa de luz recibida a la salida entre los dos modelos de fibra

Observando esta última figura se aprecia que la gráfica de la fibra modelada en nuestro proyecto es más ancha que la experimental, debido a que los parámetros ópticos no son idénticos.

*A su vez del fabricante (Ocean Optics) sólo se conoce el tamaño del núcleo y el de la cubierta, por lo que al faltar entre otros, los índices de refracción, podría explicarse significativamente esta diferencia entre la simulación y las medidas.*

*En vista de los resultados obtenidos y tras la realización de nuestro estudio, se ha demostrado que la fibra modelada en hélice a través de Blender es capaz de capturar luz cuando recibe ésta a través de la cubierta tal y como sucede en el caso del laboratorio y que la percepción de luz en la salida en función del desplazamiento de la fuente, se comporta de la misma forma que lo haría una fibra multimodo convencional.*

# Capítulo 6.-Resumen, conclusiones y líneas futuras

---

## ❖ 6.1. Introducción

*En el presente proyecto se ha estudiado y propuesto la idea de ver si sería posible diseñar y modelar en un programa de diseño y animación en 3D un elemento óptico, en este caso una fibra óptica de cristal en hélice, que sea capaz de emular las condiciones normales y principios físicos de esta tomando como referencia un caso particular de laboratorio.*

*Anteriormente se intentó realizar este tipo de diseño pero las herramientas de que se disponían por aquel entonces no eran lo suficientemente capaces de emular correctamente el comportamiento que se veía en un suceso real.*

*El por qué de este interés reside en que los programas de uso profesional de hoy en día, no resultan accesibles a cualquier tipo de público debido a licencias costosas. En el caso de que un programa de fuente libre y gratuito llegue a comportarse como uno de estos, la ayuda que proporcionaría a los usuarios a la hora de trabajar con elementos ópticos sería abismal ya que dispondrían de una herramienta de fácil acceso y que es capaz de recrear hechos que en la teoría no serían viables y que sí se dan en la práctica.*

*A lo largo de este capítulo se menciona de forma resumida todo el trabajo realizado, las conclusiones que se han ido obteniendo a medida que se elaboraba este y las opciones de pensamiento futuras que sería de gran interés estudiar para conocer más a fondo tanto el campo óptico como el desarrollo y modelado de elementos ópticos bajo un programa de diseño en 3D.*

## ❖ 6.2. Resumen

*Tras un minucioso estudio de las capacidades de Blender y observar hasta qué punto era capaz de trabajar, se diseñó y modeló una fibra óptica multimodo de cristal con el deseo de que llegase a comportarse de la misma forma que lo haría una fibra real.*

*Tras varios diseños, se logró gracias al modelado en un renderizado avanzado, cycles, que se viesen fenómenos tales como la reflexión total interna, la atenuación producida por incrementos en el radio de curvatura de algunos tramos o la influencia que tiene la posición de una fuente de luz a la hora de la propagación de haces luz.*

*El diseño en hélice reveló que la fibra modelada actuaba de la misma forma que la fibra utilizada en el caso particular del laboratorio, viéndose que Blender era capaz de recrear situaciones reales en donde el campo teórico negaba que esto pudiese darse.*

*Se elaboró entonces un código en Matlab para conocer la intensidad luminosa que se recibía en el extremo saliente de la fibra, cuantificándola con una medida relativa, para así demostrar de una forma más correcta que el modelado realizado recreaba fielmente el comportamiento de una fibra real.*

*A la vista de los resultados, se verificó que el modelado realizado era correcto, por lo que se había puesto el punto y final al determinar que el programa de diseño gráfico Blender **SÍ** es capaz de recrear fielmente un elemento óptico y que su comportamiento consigue acercarse realmente al que puede verse en una fibra en la realidad.*

## ❖ 6.3. Conclusiones

*En un principio se desconocía si esta idea daría resultado dado que el intento anterior bajo emulación con POV-RAY no tuvo éxito, pero al ser un programa más avanzado era posible que esto pudiese darse.*

*La modelación de estos diseños en Blender y su posterior renderización es un ámbito muy atrayente dentro del terreno óptico, ya que este tipo de programas se suponen fotorealistas y que son capaces de simular la luz perfectamente.*

Se vio que el renderizado básico de Blender permite modelar cualquier tipo de objeto o elemento con sencillez tras un largo estudio, pero a la hora de trabajar con necesidades más complejas, este renderizado no era capaz de soportar tales solicitudes.

El renderizado avanzado cycles en cambio, es capaz de moldearse y recrear fielmente cualquier exigencia que le solicitemos. A su vez, los resultados obtenidos son mucho más detallados, más realista y con una mejor calidad gráfica a la hora de visualizarlos.

Tras el modelado de la fibra se pudo observar que los resultados obtenidos eran bastante lógicos y acordes a cómo trabaja una fibra óptica real:

- **A medida que el radio de curvatura en el tramo de una fibra aumenta, menor es la capacidad de transmisión de luz haciendo que se propague una menor cantidad de esta.**
- **El porcentaje de luz capturada por la cubierta de la fibra es muy bajo pero no es nulo.**
- **Si el radio de curvatura de los tramo de la fibra no es superior al radio de curvatura máximo, el cual viene determinado por las medidas del núcleo cubierta, no se produce reflexión alguna hacia el exterior y se da entonces el fenómeno de la reflexión total interna.**
- **Cuanto más alejada se encuentra una fuente de luz de la boca o entrada de una fibra, menor es la recepción de esta por lo que la intensidad luminosa que recibe en el extremo saliente de la fibra es a su vez menor y por ello la posición óptima para la máxima captura de luz es en línea con la fibra y lo más cerca posible de ésta.**

### ❖ 6.4. Líneas futuras

La verificación de que Blender es capaz lograr resultados totalmente realistas dentro del ámbito óptico y que ha sido capaz de recrear un caso particular de laboratorio donde otros programas no lo han conseguido, ha sentado las bases de que es posible utilizar este tipo de herramientas para trabajar y tratar con elementos ópticos y que sobretodo, es una herramienta se encuentra al alcance de cualquiera al tratarse de un software de código libre y abierto.

En el caso de querer profundizar más y observar hasta qué punto sería capaz Blender de alcanzar resultados todavía más precisos, se podría llegar a estudiar y utilizar renderizados todavía más complejos destinados específicamente para un caso en concreto, como por ejemplo la luz.

El proyecto realizado al fin y al cabo no es más que una pequeña parte de un análisis mucho mayor y viendo las inmensas posibilidades que abarca y ofrece Blender, se podría llegar a probar otro tipo de simulaciones para comprobar si a su vez consigue recrear fielmente un caso real. Incluso observar de antemano, si un planteamiento consigue verse en Blender como algo viable antes de llevarlo a la práctica en un laboratorio por ejemplo, ahorrando en el proceso un importe considerable si anteriormente se compraba material para realizar esta prueba.

## Capítulo 7.- Referencias bibliográficas

---

### **Fibra óptica**

- 1) ADOLFO COBO GARCÍA. *Introducción a la fibra óptica y sus aspectos fundamentales. Apuntes de la asignatura de Sistemas de Comunicaciones Ópticas. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Universidad de Cantabria.*
- 2) *Teoría de fibra óptica.* Artículo de Wikipedia. [http://es.wikipedia.org/wiki/Fibra\\_%C3%B3ptica](http://es.wikipedia.org/wiki/Fibra_%C3%B3ptica)
- 3) ÓSCAR M. SANTA CRUZ. *Introducción a la fibra óptica. Ejemplar de distribución gratuita compilado anexo y redactado.*
- 4) *Reflexión total interna.* Artículo de Wikipedia. [http://es.wikipedia.org/wiki/Reflexi%C3%B3n\\_interna\\_total](http://es.wikipedia.org/wiki/Reflexi%C3%B3n_interna_total)

### **Blender: Funcionamiento, modelado de cristal y comportamiento de la luz**

- 5) *Página oficial. Descargas, manuales, tutoriales...* <http://www.blender.org/>
- 6) TONY MULLEN. *Introducing character animation with Blender.*
- 7) TON ROSENDALL, STEFANO SELLERI. *Guía de Blender 2.3.*
- 8) JOAQUÍN HERRERA GOÁS. *Guía de iniciación a Blender. Guía de iniciación para Blender 2.5.* [http://joaclintistgud.files.wordpress.com/2009/11/guia\\_blender\\_25.pdf](http://joaclintistgud.files.wordpress.com/2009/11/guia_blender_25.pdf)
- 9) *Tutoriales de Blender para los distintos campos y versiones.* [http://wiki.blender.org/index.php/Main\\_Page](http://wiki.blender.org/index.php/Main_Page)
- 10) *Cuestiones de Blender. Dudas y respuestas planteadas por usuarios.* <http://blenderartists.org/forum/>
- 11) AARON W. POWELL. *Blender 2.5 Lighting and Rendering.*

### **Procesado de imagen con Matlab**

- 12) *Página oficial.* <http://www.mathworks.es/products/matlab/>
- 13) *Procesamiento de imágenes con Matlab.* <http://es.slideshare.net/lonely113/procesamiento-digital-de-imagenes-con-matlab>
- 14) *Introducción a procesamiento digital de imágenes con Matlab.* [http://www.unioviedo.es/compnum/laboratorios\\_web/laborat03\\_intro\\_imagen/laborat03.html](http://www.unioviedo.es/compnum/laboratorios_web/laborat03_intro_imagen/laborat03.html)
- 15) *Procesamiento de imágenes con Matlab.* [http://asignatura.us.es/imagendigital/Matlab\\_PID\\_1314.pdf](http://asignatura.us.es/imagendigital/Matlab_PID_1314.pdf)
- 16) CRISTINA S. ROCHA. *Imágenes en Matlab.* <http://www-2.dc.uba.ar/materias/t1/material/Imagenes.pdf>