



Creación de libros interactivos iCartesiLibri

Joel Espinosa Longi

iCartesiLibri

Creación de libros interactivos iCartesiLibri

Joel Espinosa Longi
Universidad Nacional Autónoma de México



Título:

Creación de libros interactivos iCartesiLibri

Autor:

Joel Espinosa Longi

Código JavaScript para el libro: [Joel Espinosa Longi](#), [IMATE](#), UNAM.

Diseño de portada: Joel Espinosa Longi

Imágenes al inicio de los capítulos, generadas con el modelo: [FLUX.1\[dev\]](#)

Fuentes: [Cabin](#), [PaytoneOne](#) y [Courier Prime](#)

Fórmulas matemáticas: [K^AT_EX](#)

Proyecto iCartesiLibri

<https://proyectodescartes.org/iCartesiLibri/index.htm>

<https://prometeo.matem.unam.mx/recursos/VariosNiveles/iCartesiLibri/>



Esta obra está bajo una licencia [Creative Commons 4.0 internacional: Reconocimiento-No Comercial-Compartir Igual](#).



Tabla de contenido

- Prefacio 10
- 1. Introducción 12
 - 1.1 Libros digitales con tecnologías web 14
 - 1.2 El proyecto iCartesiLibri 19
- 2. Nociones básicas de HTML 23
 - 2.1 ¿Qué es HTML? 23
 - 2.2 Modelo de Objetos del Documento (DOM) 26
 - 2.3 El elemento <head> 27
 - 2.3.1 Etiqueta <title> 27
 - 2.3.2 Etiqueta <meta> 28
 - 2.4 El elemento <body> 31
 - 2.4.1 Etiqueta <p> 31

2.4.2	Etiquetas , <i>, , 	32
2.4.3	Etiqueta 	32
2.4.4	Etiquetas <h1>, <h2>, <h3> y <h4>	33
2.4.5	Etiquetas , y 	33
2.4.6	Etiqueta <a>	35
2.4.7	Etiqueta <base>	37
2.4.8	Etiqueta 	38
2.4.9	Etiquetas <audio> y <video>	39
2.4.10	Etiquetas <figure> y <figcaption>	40
2.4.11	Etiqueta <div>	41
2.4.12	Etiqueta de comentarios	42
2.4.13	Etiqueta 	42
2.4.14	Entidades HTML	43
2.4.15	Etiquetas <sup> y <sub>	44
2.4.16	Etiqueta <table>	44
2.5	Cómo comenzar a utilizar HTML	46
2.5.1	Editores de texto plano o sin formato	47
3.	Nociones básicas de CSS	51
3.1	¿Qué es CSS?	51
3.2	Reglas CSS en línea	52
3.3	Reglas CSS por bloque	54
3.3.1	Selectores CSS	54
3.4	Algunos atributos CSS	59

3.4.1 color	59
3.4.2 font-size	64
3.4.3 background-color	64
3.4.4 border	64
3.4.5 border-width	64
3.4.6 borde-style	64
3.4.7 border-color	65
3.4.8 border-radius	65
3.4.9 font-family	65
3.4.10 text-align	66
3.4.11 background	66
3.4.12 background-color	66
3.4.13 background-image	66
3.4.14 background-position	67
3.4.15 background-repeat	67
3.4.16 background-size	68
3.5 Posición y tamaños de los elementos	69
3.5.1 Unidades de medida en CSS	70
Unidades absolutas	70
Unidades relativas	70
3.5.2 padding	72
3.5.3 border	72
3.5.4 margin	72

3.5.5 box-shadow	77
3.5.6 text-shadow	79
3.5.7 position	80
3.5.8 display	81
4. Estructura de un libro iCartesiLibri	84
4.1 Plantilla básica	84
4.1.1 El directorio book/	85
4.1.2 El directorio extra/	86
4.1.3 El directorio img/	87
4.1.4 Licencia de uso	87
4.1.5 Archivo index.html	89
4.2 Una revisión al código de la plantilla básica	90
4.2.1 Contenido del archivo index.html	92
DOCTYPE	92
Etiquetas <meta>	93
El título de la pestaña del navegador	94
Precarga de imágenes	95
Variables de configuración del libro	96
Inclusión de los archivos del libro	97
Inclusión de los archivos de personalización del libro	97
Pantalla de carga	98
El contenedor del libro	101
Los botones de navegación entre páginas	101

El botón de la tabla de contenido	101
El botón de configuración y herramientas	102
El contenedor de las páginas	103
Dimensiones predeterminadas de las páginas	103
Página cero	105
Portada	106
Formatos de imágenes	107
Parte trasera de la portada	110
Portadilla	111
Portada interior y derechos de autor	112
Atribución al núcleo de presentación iCartesiLibri	112
Tabla de contenido	112
Entradas de la tabla de contenido	113
El prefacio	114
Página de inicio del capítulo	115
Secciones y subsecciones	117
Bibliografía	118
El final del libro	120
4.3 Elementos ofrecidos para los libros iCartesiLibri	120
4.3.1 Pie de página	120
4.3.2 Notas	123
4.3.3 Bloques autonumerados	125
4.3.4 Figuras autonumeradas	127

4.3.5 Referencias a bloques autonumerados y figuras	129
4.3.6 Referencias a páginas	130
4.3.7 Elementos interactivos	132
4.3.8 Audio	138
4.3.9 Video	140
4.3.10 Imágenes	145
4.3.11 Información emergente o Tooltips	148
5. Personalización del libro	150
5.1 Cambio en las dimensiones de las páginas	151
5.2 Cambio en los márgenes de las páginas	154
5.3 Modificar la tipografía utilizada	155
5.4 Estilo de la tabla de contenido	169
5.4.1 Flexbox	174
5.4.2 Ejemplos de estilos	180
5.5 Variables de colores y modo oscuro	189
5.6 Botón de herramientas y configuración	192
5.6.1 Tema de colores claros y oscuros	192
5.6.2 Inclusión de herramientas externas	194
5.7 Estilos de comienzo de capítulo	205
5.7.1 Portada de capítulos de la plantilla básica	205
5.7.2 Portada de capítulos con una imagen en la misma ... página del contenido	207
5.7.3 Inicio de capítulos puramente textual	212
5.7.4 Imagen para el borde del título del capítulo	217

5.7.5 Inicio de capítulo con una tabla de contenido ... específica para el capítulo	220
5.8 Estilo en la numeración de páginas	222
5.9 Generación de un PDF del libro	226
6. Expresiones matemáticas	230
6.1 ¿Qué es $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$?	231
6.1.1 Modo matemático de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$	232
6.1.1.1 Símbolos	235
6.1.1.2 Operadores	236
6.1.1.3 Potencias e índices	239
6.1.1.4 Fracciones	240
6.1.1.5 Coeficientes binomiales	242
6.1.1.6 Raíces	243
6.1.1.7 Sumas, integrales y operaciones similares	244
6.1.1.8 Delimitadores	247
6.1.1.9 Entornos	250
6.1.1.10 Texto dentro del modo matemático	256
6.2 Biblioteca $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$	258
6.3 Biblioteca MathJax	263
7. Referencia rápida de elementos	266
8. Bibliografía	271



Prefacio

Este trabajo tiene como objetivo principal presentar los conocimientos necesarios para la creación de libros digitales interactivos, en especial aquellos que forman parte del proyecto iCartesiLibri. Su contenido será actualizado constantemente con respecto a las nuevas funcionalidades y correcciones incorporadas al núcleo de presentación de libros tipo iCartesiLibri, sirviendo así como una referencia o documentación exhaustiva que se irá extendiendo con el tiempo.

En el capítulo 1 se ofrece una breve introducción sobre la naturaleza de los libros digitales interactivos y su contexto dentro del proyecto iCartesiLibri. Se explora la importancia de estos recursos educativos y las ventajas que ofrecen, además de guiar al lector en cómo comenzar a crear documentos con estas características innovadoras.

Los capítulos 2 y 3 se dedican a explorar las nociones básicas de HTML y CSS, componentes fundamentales para la construcción de libros iCartesiLibri. Estos capítulos están diseñados para servir como base para aquellos lectores sin conocimientos previos sobre el funcionamiento de estas tecnologías web, proporcionando la información necesaria para comprender su función; sin embargo, los lectores familiarizados con HTML y CSS pueden omitir estos capítulos sin perder información relevante para la creación de libros digitales interactivos tipo iCartesiLibri.

El capítulo 4 explora la estructura interna de un libro iCartesiLibri, describiendo los archivos que lo componen y su contenido, también se presentan los diversos elementos que el núcleo de presentación incorpora a HTML con el objetivo de facilitar la creación de libros. En este capítulo se pueden encontrar las últimas actualizaciones y nuevas funcionalidades incorporadas al núcleo de presentación de libros tipo iCartesiLibri.

En el Capítulo 5 se exploran diversas formas de personalizar la apariencia del libro interactivo a través de ejemplos de reglas CSS que permiten lograr efectos visuales y maquetaciones particulares. También se proporcionan algunas plantillas para ofrecer opciones flexibles al momento de iniciar la creación de un libro interactivo.

Por último, en el capítulo 6, se presentan algunos mecanismos para incluir fórmulas matemáticas mediante las bibliotecas $KATEX$ o $MathJax$, enriqueciendo así el contenido del libro digital interactivo.



1. Introducción

Los libros digitales, electrónicos o también conocidos como *eBooks* han existido desde hace muchos años, y se trata de documentos digitales que utilizan texto e imágenes para presentar su contenido.

Surgen de la necesidad de aprovechar las pantallas de dispositivos electrónicos para presentar la información de obras impresas, como libros o revistas, ofreciendo una alternativa a los medios físicos tradicionales. Debido a su similitud con los medios impresos, inicialmente los eBooks solo proporcionan texto e imágenes para su contenido, además de mantener la disposición o estructura visual (*layout*) de sus contrapartes físicas. Los clásicos documentos en PDF representan un ejemplo antiquísimo de eBook, utilizado desde hace más de treinta años (1993).

Los eBooks tradicionales se basan en la reproducción digital de un libro impreso, con un enfoque principal en el texto y la estructura visual. Y si bien los visualizadores de eBooks incorporan algunos componentes interactivos como los marca páginas digitales (*bookmarks*), formularios para introducir información, hipervínculos y anotaciones en las páginas, estos elementos de interacción suelen ser limitados.

Con el avance tecnológico, los libros digitales han incorporado más características que sus contrapartes físicas no pueden proporcionar, por ejemplo, pueden incluir elementos multimedia como audio y video, así como objetos interactivos sofisticados. Estos cambios han dado origen a lo que se conoce como *libros digitales interactivos*, con el objetivo de hacer que el lector participe activamente en la experiencia de lectura, complementando la información textual con elementos multimedia e interactivos.

Una desventaja que han acarreado los libros digitales interactivos es que sus funcionalidades avanzadas requieren una aplicación dedicada para su visualización. Por ejemplo, el formato ePub ofrece múltiples características para la presentación de eBooks, permitiendo incluir diversos elementos multimedia y objetos interactivos programados con JavaScript, sin embargo, este formato presenta el problema que no todos los usuarios potenciales cuentan con un visor de archivos compatible con este formato, siendo necesario instalar una aplicación adicional, y además, la mayoría de los visores existentes varían en el soporte de las diversas características del formato, lo que causa que el acceso a libros en ePub no sea homogéneo para todos los lectores.

De hecho, esto no es algo que ocurra únicamente con los archivos ePub o con los libros interactivos, es una tendencia usual cuando se introducen nuevos formatos, por ejemplo ha ocurrido lo mismo con los formatos de imágenes, audio y video; no obstante, por lo general surgen algunos formatos que se estandarizan y son soportados por la mayoría de sistemas y aplicaciones, en el caso de los libros digitales sería el PDF, el cual no es muy interactivo que digamos, aunque la mayoría de los sistemas lo soporta.

1.1 Libros digitales con tecnologías web

Las tecnologías web como HTML, CSS y JavaScript han evolucionado para incorporar las características necesarias para la presentación de libros digitales interactivos, soportando diversos elementos multimedia dentro de una página web, así como la implementación de interacciones sofisticadas programadas mediante JavaScript. La falta de un visor general para libros interactivos y la omnipresencia de los navegadores de internet han convertido a estas tecnologías en el foco de atención para la creación y presentación de libros digitales interactivos.

Actualmente, son comunes los sitios web que ofrecen servicios para crear libros digitales a partir de documentos en PDF o DOCX, haciendo una conversión del documento a una versión que utiliza tecnologías web; así mismo dentro de los servicios de estos sitios se ofrece el alojamiento del eBook, el cual se puede compartir por medio de un vínculo para que los lectores accedan a él desde cualquier lugar. Algunos ejemplos de estos sitios son: [Issuu](#) (Figura 1.1), [PubHTML5](#) (Figura 1.2), [flippingbook](#) (Figura 1.3); existiendo muchos otros más.

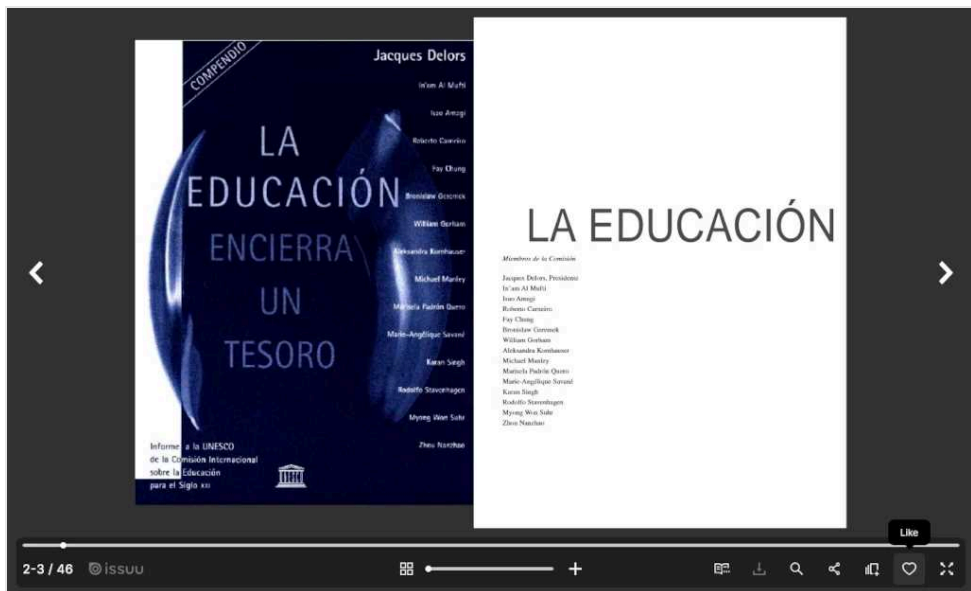


Figura 1.1. Ejemplo de libro digital en el sitio [Issuu](https://www.issuu.com).

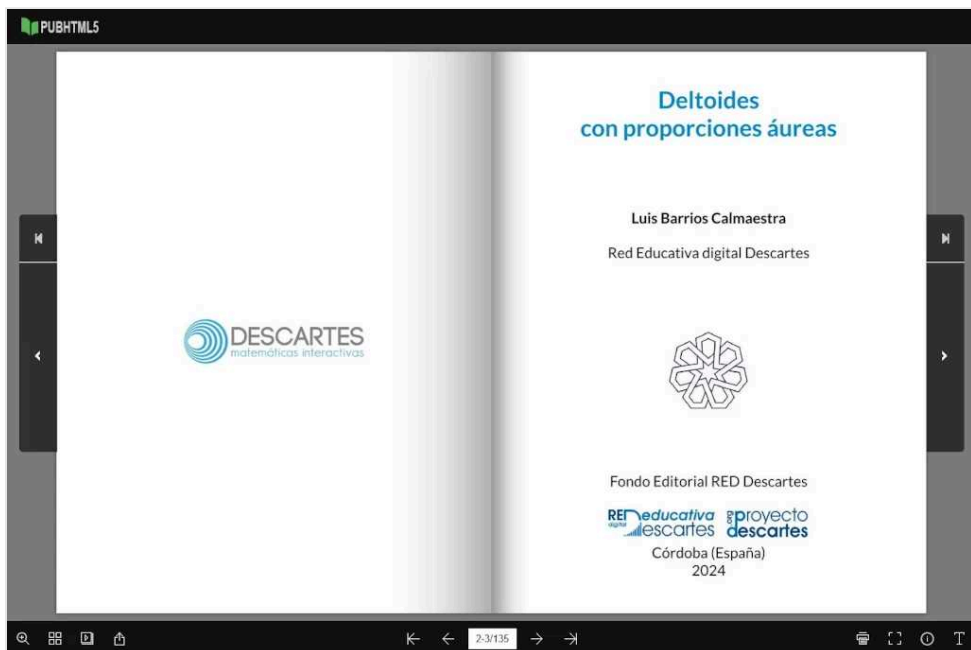


Figura 1.2. Ejemplo de libro digital en el sitio [PubHTML5](https://www.pubhtml5.com).



Figura 1.3. Ejemplo de libro digital en el portal [flippingbook](#).

Si bien, servicios como estos permiten crear versiones web de documentos y compartirlos fácilmente, estas versiones no son consideradas propiamente libros digitales interactivos, ya que las interacciones que proporcionan se limitan a botones de navegación para cambiar de página, marca páginas, hipervínculos y anotaciones; esto se debe a que se basan en documentos sin interacción, como lo son los PDF o DOCX. De hecho, en la actualidad, la mayoría de los navegadores de internet pueden abrir y mostrar directamente documentos PDF, lo que reduce aún más las bondades que ofrecen estas versiones web de los documentos.

También existen herramientas que permiten crear libros interactivos, donde cada capítulo o sección se corresponde con una página web individual, interconectadas mediante hipervínculos. En estos libros digitales, las tecnologías web se utilizan para presentar el contenido de manera flexible y dinámica, sin la intención de replicar la estructura visual de un libro tradicional, por lo que el texto se despliega en pantallas, permitiendo al lector visualizar el contenido de cada página mediante el desplazamiento (o *scroll*) vertical, en lugar de una navegación paginada como en los libros impresos. Algunos ejemplos de estas herramientas son: [The Magic Book Project](#) (Figura 1.4); [BookDown](#) (Figura 1.5); [PreTeXt](#) (Figura 1.6); entre otras. Además, existen soluciones personalizadas desarrolladas por los propios autores, como por ejemplo las utilizadas en: “[Interactive Linear Algebra](#)”, “[Collision Detection](#)”, “[CK-12 Conceptos de Álgebra 2 con Trigonometría - en Español](#)”, por mencionar algunos.

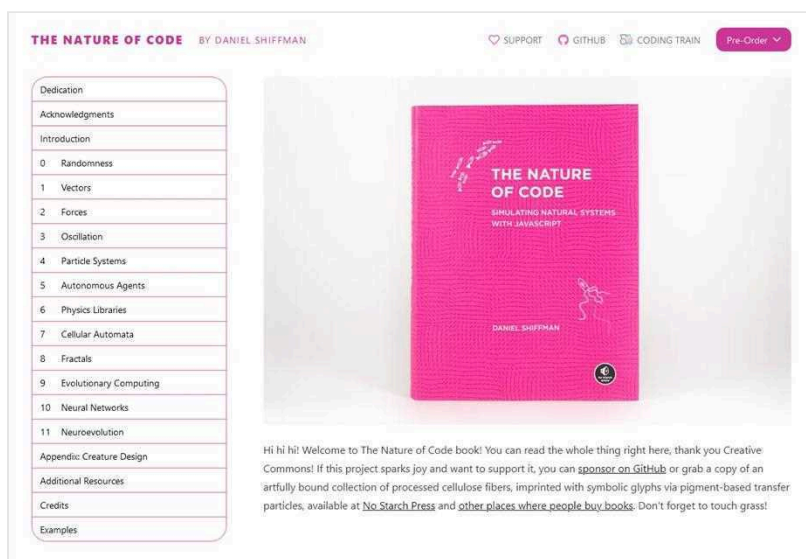


Figura 1.4. Libro digital interactivo “[The Nature of Code](#)” hecho con The Magic Book Project.

Applied Data Visualization

🔍 🏠 📄 📄 📄

- 1 About
- 2 Introduction
- 3 Quality of graphs
- 4 Grammar of graphics & Ggplot
- 5 Visualizing descriptive statistics
- 6 Visualizing statistical models
- 7 Visualizing geographic data
- 8 Interactive data visualization (IDV)
- 9 Plotly (IDV)
- 10 Shiny (IDV)
- 11 Visualizing text data
- 12 Animations & movies
- 13 Academic posters
- 14 Appendix A: Data visualization examples
- 15 Appendix B: AI & LLMs & data visualization
- 16 Appendix C: Data examples & more stuff
- 17 Appendix D (IN GERMAN): Data report (term paper)

About

This website serves as slides and script for the GESIS workshop *Applied Data Visualization*, for the seminar *Datenvisualisierung für Politikwissenschaftler* (Universität Freiburg) as well as for workshops in other, shorter formats taught by [Paul C. Bauer](#).¹ Original material is licensed under a [Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](#) license. Where I draw on other authors material (I do so extensively) other licenses may apply. Please note the references in the syllabus as well as the citations and links in the script. For potential future versions of this material see www.paulcbauer.de. If you have feedback or discover errors/dead links please let me know (mailto:paulcbauer.de). And make sure to [install all the necessary packages](#) before you use the script.

- Current version of website can be found at <https://bookdown.org/paulcbauer/applied-data-visualization/>.
- Interested in a data visualization workshop at your institution/company? Please contact me at mail@paulcbauer.de.

1 Versions: Workshops & Seminars

- Seminar at University of Freiburg (18h teaching hours in total):

On this page

- 1 Versions: Workshops & Seminars
- 2 Instructor: About me
- 3 Your turn
- 4 Script & material
- 5 Online vs. offline
- 6 Strategy & Goals/Learning outcomes
- 7 Software we will use
- 8 Structure (1)
- 9 Structure (2)
- 10 Helpful resources

🔗 Edit this page
Report an issue

Figura 1.5. Libro digital interactivo “[Applied Data Visualization](#)” hecho con BookDown.

A Second Course in Linear Algebra

Robert A. Beezer

- Contents
- Front Matter
- 1 The Fundamentals
- 2 Matrix Decompositions
- 3 Canonical Forms
- 4 Applications
- 5 Topics
- A GNU Free Documentation License

Authored in PreTeXt

POWERED BY
MathJax

< Prev
^ Up
Next >

A Second Course in Linear Algebra

Robert A. Beezer
Department of Mathematics and Computer Science
University of Puget Sound
beezer@pugetsound.edu

DRAFT March 9, 2020 DRAFT

Colophon ▶

Acknowledgements ▶

Figura 1.6. Libro digital interactivo “[A Second Course in Linear Algebra](#)” hecho con PreTeXt.

Este tipo de libros son interesantes porque representan realmente lo que puede llegar a ser un libro digital interactivo, en el sentido de emplear texto, imágenes y elementos interactivos para presentar y potenciar su contenido.

1.2 El proyecto iCartesiLibri

El proyecto iCartesiLibri originado por la Red Educativa Digital Descartes (España y Colombia) fue concebido con la intención de crear libros digitales interactivos que presenten una estructura similar a la de un libro físico, pero complementados con videos y recursos interactivos creados con la herramienta de autor DescartesJS. El libro “[DescartesJS - Nivel I](#)” (Figura 1.7) es un ejemplo de libro digital interactivo desarrollado para el proyecto iCartesiLibri.

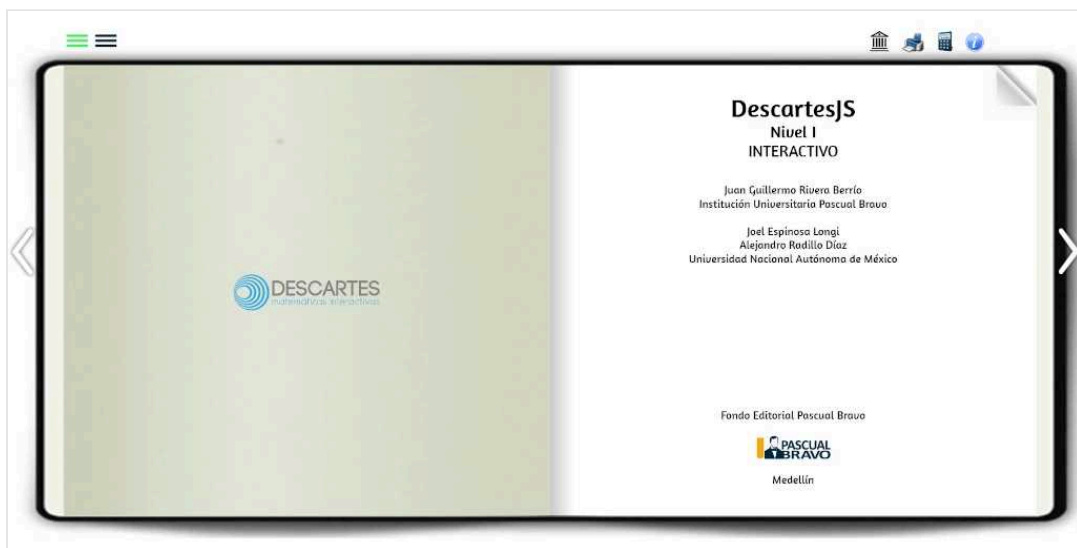


Figura 1.7. Primera edición del libro “DescartesJS - Nivel I”.

Paralelamente, dentro del grupo Descartes del Instituto de Matemáticas de la UNAM, también se busca la creación de libros interactivos, en particular se comienza el desarrollo de herramientas para presentar el libro de Geometría Visual del Dr. Javier Bracho y el Dr. José Luis Abreu. Este esfuerzo dio origen al proyecto [D.I.Books](#) (*Digital Interactive Books*), que consiste en un visor y editor de libros interactivos integrado con la herramienta de Geometría Proyectiva: [ProGeo3D](#); el cual es desarrollado como una escena sofisticada de DescartesJS.

A raíz del interés y la necesidad de una herramienta independiente para crear y presentar libros interactivos que mantengan la estructura visual de un libro físico, y que no este integrada a una herramienta adicional (como ProGeo3D o DescartesJS), se inició el desarrollo de un nuevo sistema que aproveche las características que proporcionan las tecnologías web.

Mediante código en HTML, CSS y JavaScript se creó un núcleo para la presentación de libros interactivos que permite la navegación paginada de un documento en HTML, ofreciendo diversos elementos adicionales como una tabla de contenido, notas al pie de página, referencias, autonumeración de páginas, entre otras cosas más. Además, al utilizar tecnologías web se pueden incluir directamente elementos multimedia como imágenes, audios y videos, así mismo, al no tener integrada o depender de una herramienta adicional es posible incluir objetos interactivos desarrollados con diversas herramientas, como [DescartesJS](#), [GeoGebra](#), [p5.js](#), [three.js](#), o programados directamente con JavaScript. Esta flexibilidad aumenta las posibilidades de creación de objetos interactivos para los libros digitales interactivos.

Las últimas versiones de los libros del proyecto [iCartesiLibri](#) integran actualmente el [núcleo de presentación](#) mencionado anteriormente. Por ello, en este documento se referirá a esta forma de libros como “libros interactivos tipo iCartesiLibri” o simplemente “libros iCartesiLibri”. Hay que destacar que, en el año 2024, el proyecto cuenta con poco más de 100 libros interactivos que abarcan temas como aritmética, álgebra, geometría, cálculo, química, física, inteligencia artificial, entre otros.

En la siguiente imagen se muestra la segunda edición del libro “[DescartesJS - Nivel I](#)” la cual utiliza el nuevo núcleo de presentación de libros tipo iCartesiLibri.



Figura 1.8. Segunda edición del libro “DescartesJS Nivel I”.

Debido a que los libros tipo iCartesiLibri se construyen directamente escribiendo código HTML y CSS, similar a la construcción de un página web estándar, es fundamental comprender las características y conceptos de estos dos lenguajes para la creación de este tipo de libros interactivos.

También es prioritario enfatizar que la producción de estos libros interactivos, se asemeja más a la creación de diapositivas tipo PowerPoint que a la edición de documentos en un procesador de textos como Word o los creados con L^AT_EX; esto se debe a que el contenido que aparece en cada página del libro se determina individualmente, principalmente por la manera en que funcionan las tecnologías web, ya que estas no ofrecen mecanismos para auto paginar el contenido, siendo el autor quien se encarga de dividir y dosificar en páginas el material que se presenta.



2. Nociones básicas de HTML

2.1 ¿Qué es HTML?

HTML (HyperText Markup Language) es un lenguaje de marcado utilizado para crear páginas web. Los lenguajes de marcado consisten en un conjunto de reglas sintácticas que permiten incorporar, por medio de etiquetas, información de presentación o estructura a un documento puramente textual; debido al uso de etiquetas se les suele denominar también lenguajes de etiquetado.

Un ejemplo sencillo de lenguaje de marcado es el usado en aplicaciones de mensajería como WhatsApp o Discord, para especificar estilos como negrita y cursiva. Esto se logra utilizando una sintaxis sencilla: un texto acotado por asteriscos (**un texto en negrita**) representa un texto en negrita, mientras que un texto delimitado por guiones bajos (*_un texto en cursiva_*) muestra un texto en itálica o cursiva.

En el ejemplo anterior, el asterisco y el guión bajo actúan como marcas o etiquetas textuales que proporcionan información adicional sobre el estilo del texto que delimitan.

Las etiquetas utilizadas en HTML son más sofisticadas que las vistas en el ejemplo anterior, ya que permiten incorporar más detalles (o propiedades) sobre los elementos marcados, ofreciendo una manera de especificar el contenido y la estructura de un documento web, delimitando elementos como títulos, párrafos, imágenes, videos, botones, entre muchos otros. Esta capacidad, combinada con otros lenguajes web como CSS, para la configuración de la apariencia visual, y JavaScript, para agregar interactividad, constituyen los pilares fundamentales de las páginas de internet, y en particular, serán utilizadas para crear libros digitales interactivos tipo iCartesiLibri.

La forma que tienen las marcas o etiquetas de HTML es la siguiente `<etiqueta>contenido</etiqueta>`, es decir, el tipo o nombre de la etiqueta está acotado por un menor y un mayor que (`<` y `>`), siendo este nombre específico para cada elemento y sus posibles valores están definidos por el lenguaje. Asimismo, se tiene una etiqueta que abre o delimita inicialmente el contenido (`<etiqueta>`) y una etiqueta con el mismo nombre y una diagonal al comienzo (`/`) que cierra el marcado del contenido (`</etiqueta>`); todo el contenido que se encuentra acotado por este par de elementos es efectivamente marcado o etiquetado. Por ejemplo, HTML proporciona una etiqueta para identificar un párrafo de texto, siendo `<p></p>`, que permite agrupar una porción de texto etiquetándola para que forme una misma unidad de contenido.

El siguiente código muestra cómo se crea un párrafo en HTML:

```
<p>El siguiente fragmento de texto está escrito en lenguaje HTML y representa un párrafo.</p>
```

El contenido o texto acotado por la etiqueta `<p>` es considerado un párrafo y un navegador de internet al leer el código determina la forma en la que debe mostrarse dicho párrafo; aunque claro, esta representación visual puede ser configurada para conseguir un estilo visual propio.

Gracias a la versatilidad de los lenguajes de marcado, es posible anidar diversas etiquetas, es decir, pueden existir etiquetas dentro de otras etiquetas. Por ejemplo:

```
<p>Este es el contenido de un párrafo en HTML y el siguiente texto <b>dentro del párrafo está en negrita</b>.</p>
```

En el código anterior, se observa una etiqueta `<p>` que delimita un párrafo, y dentro de su contenido, aparece una etiqueta `` que marca una porción más pequeña del contenido del párrafo y representa un texto en negrita (*bold*). Esta anidación de etiquetas, es decir, la inclusión de etiquetas dentro de otras, crea una estructura jerárquica entre los diversos elementos, lo que se utiliza para construir un árbol. Aunque no se trata de un árbol físico como un pino o un abedul, más bien se refiere a una estructura similar a la utilizada para representar un árbol genealógico, es decir, una estructura jerárquica que relaciona progenitores con descendientes.

2.2 Modelo de Objetos del Documento (DOM)

Dentro de un documento HTML, las etiquetas construyen la estructura jerárquica conocida como el Modelo de Objetos del Documento (*Document Object Model*, **DOM**), que es la forma computacional en la que se representa un documento HTML. El DOM además de contener la representación de una página web, también permite a lenguajes de programación como JavaScript interactuar y manipular sus elementos.

Un documento HTML siempre está constituido por un elemento inicial, a saber `<html></html>`, el cual es considerado como la raíz de la estructura jerárquica y se presenta como el ancestro común de cualquier otro elemento dentro del documento. Generalmente, la raíz contiene dos descendientes `<head></head>` y `<body></body>`, que se encargan de contener la información del documento. Esta estructura jerárquica puede ser visualizada de la siguiente manera:

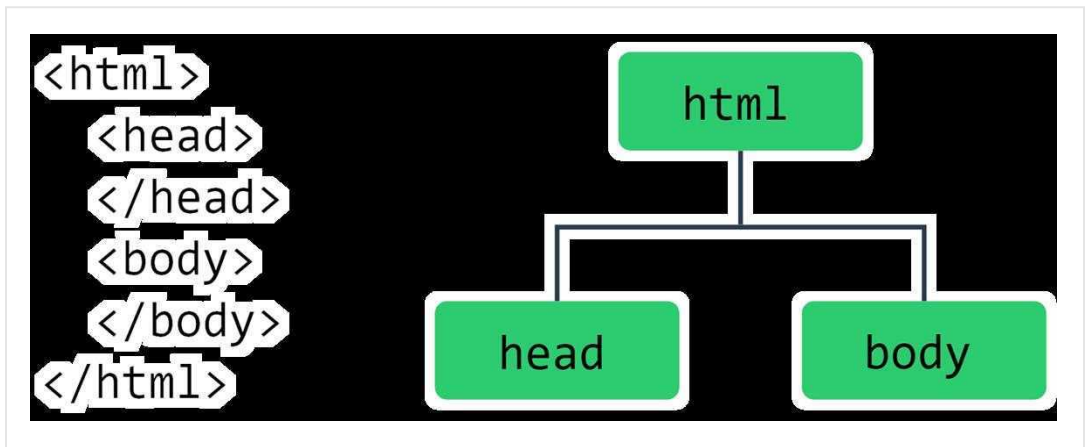


Figura 2.1. A la izquierda se encuentra el código HTML y a la derecha una representación visual del DOM.

En el esquema anterior las etiquetas se representan mediante rectángulos (verdes) y las relaciones dentro de la estructura jerárquica son representadas por líneas que unen los rectángulos (o nodos del árbol). Un nodo es progenitor de otro si está relacionado por una línea y se encuentra arriba de él. Por otro lado, un nodo es descendiente si está relacionado por una línea y se encuentra debajo. Al ver el esquema es posible notar las semejanzas con la estructura de un árbol genealógico.

2.3 El elemento `<head>`

La etiqueta `<head></head>` corresponde a la cabecera del documento y en su jerarquía se utilizan etiquetas específicas para incluir metadatos de la página y elementos que controlan la apariencia, como lo son las hojas de estilos. También se permite enlazar archivos JavaScript para incorporar algún tipo de interactividad en la página. Por lo general, los elementos que forman parte de la cabecera no son visibles en un navegador, a excepción de la etiqueta `<title></title>`.

2.3.1 Etiqueta `<title>`

El texto mostrado en la pestaña de cada página abierta en un navegador de internet es determinado por el elemento `<title></title>`, que especifica el contenido que se visualiza (**Figura 2.2**). Si esta etiqueta no forma parte del documento, se muestra en la pestaña la dirección de la página visualizada. Es recomendable siempre incluir esta etiqueta como descendiente del elemento `<head>`.



Figura 2.2. Nombre de la pestaña del navegador especificado como: `<title>HTML - Wikipedia, la enciclopedia libre</title>`.

2.3.2 Etiqueta `<meta>`

Las etiquetas de metadatos de la cabecera son utilizadas para proporcionar datos como el autor, las palabras clave (que permiten buscar el documento en internet), el tamaño de la ventana en el navegador de un dispositivo móvil. Sin embargo, quizá la más importante sea la etiqueta de metadatos que indica al navegador cuál es la codificación utilizada para interpretar y desplegar el texto.

Esta etiqueta es: `<meta charset="UTF-8">`, y es utilizada para que el navegador pueda representar correctamente los caracteres del texto del documento. En el pasado era muy común encontrar sitios que mostraban caracteres incorrectos, por ejemplo, letras acentuadas o la letra ñ:



Figura 2.3. Ejemplo de texto incorrectamente representado a causa de una mala codificación.

Esto se produce debido a que la codificación utilizada por defecto por el navegador es la misma del sistema operativo en que se ejecuta. En sistemas que utilizan un idioma diferente al español, puede suceder que la codificación del navegador no considere caracteres acentuados, dando lugar a la aparición de símbolos extraños en el texto. La inclusión de `<meta charset="UTF-8">` al inicio del documento indica al navegador que el contenido utiliza caracteres codificados bajo la especificación UTF-8, que en términos sencillos significa que se pueden representar prácticamente cualquier símbolo textual. De esta manera, independientemente del idioma del sistema que utilice el usuario para acceder a la página web, la información será mostrada tal y como fue originalmente creada.

La codificación de caracteres es un tema más complejo que la simplificación previamente presentada, ya que se debe asegurar que la codificación interna del documento coincida con la utilizada en la etiqueta `<meta>`. A pesar de ello, en la mayoría de los casos, la codificación UTF-8 es utilizada directamente por los editores de texto modernos, por lo que basta con incluir la etiqueta `<meta charset="UTF-8">` como el primer elemento dentro de la etiqueta `<head>` y el contenido debería mostrarse correctamente.

En la etiqueta anterior, se observa que `charset="UTF-8"` aparece dentro del nombre de la etiqueta `<meta>` separados únicamente por un espacio, es decir, esta información no está delimitada por una etiqueta que abre y otra que cierra.

De esto se derivan dos aspectos importantes:

Primero, al igual que existen etiquetas que pueden tener descendientes (elementos que están acotados por marcas que abren y cierran), también existen etiquetas que no pueden contener descendientes, ya que se auto cierran.

En cuanto a la sintaxis de este tipo de etiquetas, a veces se suelen escribir con una diagonal al final (antes del mayor que), por ejemplo `<meta charset="UTF-8" />`, con la finalidad de indicar que la propia etiqueta se cierra. Este detalle se omite frecuentemente al escribir HTML, ya que los navegadores son relativamente permisivos en cuanto a la sintaxis del etiquetado se refiere. Sin embargo, en un documento bien estructurado, sí es necesario incluir la diagonal al final de los elementos de este tipo; por ejemplo, en lenguaje de marcado como XML o SVG, la falta de esta diagonal puede causar un error.

Segundo, se pueden incluir atributos o parámetros que proporcionan información adicional a la etiqueta, en el caso de la etiqueta `<meta>`, el atributo es `charset` y se le asigna (=) el valor de `"UTF-8"`. Esta característica permite una amplia personalización en la forma en que se etiqueta un documento HTML, ya que permite incluir información extra sobre el elemento marcado. La manera de utilizarlo es: `nombre_atributo="valor"`, donde el nombre del atributo (propiedad o parámetro) es una cadena de caracteres sin espacios, seguido de un signo igual y, finalmente, su valor se encuentra delimitado por comillas dobles (") o sencillas (').

A continuación, se presentan un conjunto de elementos HTML utilizados dentro de la etiqueta `<body>`, los cuales permiten estructurar, especificar y construir el contenido de un documento.

Una lista con todos los elementos del lenguaje HTML se encuentra en: <https://developer.mozilla.org/es/docs/Web/HTML/Element>. Se aconseja tener a la mano este vínculo como una tabla de referencia, ya que en este libro interactivo solo se explicarán las etiquetas útiles para la construcción de libros iCartesiLibri.

2.4 El elemento `<body>`

En el elemento `<body></body>` se incluyen los elementos que constituyen el contenido visible del documento, es decir, texto, imágenes, vídeos, enlaces y cualquier otro elemento mostrado al usuario en la página.

2.4.1 Etiqueta `<p>`

Un párrafo es delimitado por la etiqueta `<p>`, como se presentó al inicio del capítulo, y constituye uno de los bloques fundamentales de un documento HTML. Este conjunto de caracteres, que agrupa oraciones relacionadas entre sí para desarrollar una idea específica o tema, es considerado la unidad fundamental de texto. Se le puede visualizar como un pequeño bloque de construcción dentro de un texto más extenso, con la función de organizar la información de manera clara y coherente.

Al visualizar un párrafo en un navegador de internet, generalmente se incorpora un salto de línea para separar el contenido entre los diferentes párrafos que conforman el texto. Esta práctica facilita la legibilidad y comprensión del documento al distinguir cada párrafo como una unidad conceptual independiente y guiar la atención del lector de forma natural a través del texto.

2.4.2 Etiquetas ``, `<i>`, ``, ``

Dentro de un párrafo, se pueden marcar porciones de texto para indicar que tienen un significado semántico o estilo específico, por ejemplo, para enfatizar o resaltar una palabra, se puede utilizar la etiqueta `` para indicar el uso de negritas, o `<i>` para indicar el uso de itálicas o cursivas. Existen también las etiquetas `` y ``, que visualmente presentan el mismo resultado que `` e `<i>`, respectivamente; aunque esta representación puede variar según el navegador. Sin embargo, algunos consideran que el uso de las etiquetas `` e `<i>` rompe el principio de separar el contenido de su representación visual. Debido a que estas etiquetas, además de marcar, se asocian con un comportamiento visual, se suele preferir utilizar `` o ``, ya que proporcionan una mejor caracterización semántica. Si bien este detalle no es relevante para la mayoría de los autores, es un detalle que suele aparecer al profundizar en la información sobre HTML y su uso.

2.4.3 Etiqueta ``

Para los textos, se dispone de la etiqueta ``, que actúa como una etiqueta genérica para delimitar partes del texto, y por sí misma, no proporciona ningún tipo de información semántica adicional; su función principal es agrupar caracteres y es al autor al que corresponde determinar el comportamiento o uso de dicha agrupación. Esta característica resulta útil, por ejemplo, cuando se busca aplicar un estilo visual específico a una palabra dentro del texto, lo que cobra más sentido cuando se combina el uso de HTML con CSS, como se explicará posteriormente.

2.4.4 Etiquetas <h1>, <h2>, <h3> y <h4>

Asimismo, para el texto se encuentran etiquetas que permiten diferenciar distintos niveles de encabezado, es decir, el texto que suele aparecer al comienzo de un bloque de texto que identifica el capítulo o sección en la que se encuentra el texto. Estas etiquetas son: <h1>, <h2>, <h3>, <h4> <h5> y <h6>, donde la h proviene de *header*, y el número indica el nivel de importancia, siendo el <h1> el encabezado con mayor relevancia, generalmente empleado para el título de un capítulo; <h2> se suele usar para las secciones, <h3> para las subsecciones y así sucesivamente.

2.4.5 Etiquetas , y

También se encuentran etiquetas que permiten crear listas de texto, tanto las clásicas numeradas como con viñetas. Para ello, existen los elementos , que proviene de *ordered list* y que representa una lista autonumerada, y de *unordered list*, con la cual se puede crear una lista con viñetas o no numerada. Estas etiquetas indican que su contenido corresponde a una lista, y para especificar qué elementos forman parte de la lista, se utilizan conjuntamente con los elementos (*list item*) para marcar cada elemento. Por ejemplo:

```
<ol>
  <li>Primer elemento de la lista</li>
  <li>Segundo elemento de la lista</li>
  <li>Tercer elemento de la lista</li>
</ol>
```

Representa la siguiente lista numerada:

Resultado

1. Primer elemento de la lista
2. Segundo elemento de la lista
3. Tercer elemento de la lista

Mientras que el código:

```
<ul>  
  <li>Primer elemento de la lista</li>  
  <li>Segundo elemento de la lista</li>  
  <li>Tercer elemento de la lista</li>  
</ul>
```

Representa la siguiente lista con viñetas:

Resultado

- Primer elemento de la lista
- Segundo elemento de la lista
- Tercer elemento de la lista

Hay que mencionar que las listas pueden ser anidadas dentro de otras listas, creando así una estructura jerárquica en los elementos listados. Y más adelante se muestra la manera en que pueden ser configuradas, para presentar numeraciones o viñetas diferentes.

2.4.6 Etiqueta <a>

Parte fundamental de los documentos HTML, son los hipervínculos, la **H** en HTML. Los hipervínculos, también conocidos como hiperenlaces, enlaces o *links*, son elementos que permiten crear un puente de unión a otros elementos, ya sea dentro del mismo documento o referenciando un documento externo. La etiqueta `<a>` que proviene de *anchor* (ancla), se utiliza para este fin. Esta etiqueta no posee una representación visual propia y, en su lugar, se emplea para englobar otros contenidos, como textos o imágenes; y en particular al etiquetar un texto es usual que un hipervínculo se indique mediante el dibujo de una línea debajo del texto y el uso de un color diferente al del texto general. Por ejemplo:

```
<a href="https://descartes.matem.unam.mx/">vínculo  
a DescartesJS</a>
```

Muestra lo siguiente:

Resultado

[vínculo a DescartesJS.](https://descartes.matem.unam.mx/)

El parámetro `href` que proviene de *hiperreferencia*, proporciona la ruta o ubicación a la que se enlaza. Así mismo, al reemplazar el texto “vínculo a DescartesJS” por una imagen, por ejemplo:

```
<a href="https://descartes.matem.unam.mx/"></a>
```



En este caso, toda la imagen (`logo_descartes.svg`) se convierte en un enlace a la página de DescartesJS y no se suele añadir ningún estilo visual adicional a la imagen para indicar que es un hipervínculo. Hay que comentar que, aunque se habla de un estilo visual, cuando se trata únicamente de HTML, este no posee una representación visual en sí mismo, sino que al momento de visualizar un documento HTML en un navegador de internet es cuando se utilizan estilos visuales para representar el contenido de la página web.

Así como se pueden crear hipervínculos que contengan texto e imágenes, es posible hacer lo mismo con prácticamente cualquier elemento de HTML.

Al pulsar un enlace, el comportamiento predeterminado consiste en reemplazar la página web actual con el contenido del documento referenciado en `href`. Este proceso ocasiona una pérdida momentánea del documento que se observa, siendo reemplazado por uno nuevo, requiriéndose el uso de los controles de navegación del navegador para regresar. En muchos casos, se busca que el vínculo se presente en una ventana nueva, sin perder el documento original, para ello, la etiqueta `<a>` permite especificar, mediante su parámetro `target`, la ubicación donde se mostrará el contenido.

Los valores permitidos para este parámetro son los siguientes:

- `_self`, carga el contenido en la misma ventana, reemplazando el documento inicial. Este es el valor por omisión, que determina el comportamiento predeterminado.
- `_blank`, indica al navegador que el enlace se debe mostrar en una nueva ventana o pestaña, según la configuración del navegador y del usuario.
- `_parent`, cuando la página está contenida dentro de un `<iframe>`, este valor indica que el vínculo se mostrará en la ventana que contiene dicho `<iframe>`.
- `_top`, similar a `parent`, pero busca el primer elemento progenitor que no sea un `<iframe>` en la jerarquía.

Por lo tanto, si se desea mostrar el enlace en una nueva ventana (o pestaña), es necesario incluir el parámetro `target` con el valor `"_blank"`. Un ejemplo de esto sería:

```
<a href="https://descartes.matem.unam.mx/">vínculo  
a DescartesJS</a>
```

2.4.7 Etiqueta `<base>`

Cuando un documento HTML contiene numerosos hipervínculos que utilizan el atributo `target` para modificar su comportamiento predeterminado, es decir, cuando no utilizan `"_self"` la repetición del parámetro `target` en cada hipervínculo se vuelve necesaria.

Para simplificar esta situación, existe la etiqueta `<base>`, que entre sus funciones, permite asignar un objetivo común a todos los hipervínculos de un documento. Por ejemplo:

```
<base>
```

Hace que todos los enlaces se muestren en una nueva pestaña, de esta manera, el objetivo predeterminado (`target`) se define solo en un lugar, aplicándose a todos los enlaces del documento.

La etiqueta `<base>` no tiene representación visual y, por lo tanto, se recomienda incluirla dentro de la sección `<head>`.

2.4.8 Etiqueta ``

Otro elemento que no puede faltar en un documento HTML son las imágenes, las cuales son incorporadas por medio de la etiqueta que se auto cierra ``, junto con el parámetro o atributo `src` (de *source*) para indicar la ruta del archivo de la imagen. Por ejemplo:

```

```

Determina una imagen que se encuentra en la ubicación `https://descartes.matem.unam.mx/img/logo_descartes.svg`. De nuevo, se observa cómo el uso de atributos dentro de las etiquetas proporciona más información sobre el elemento; en este caso, se expresa que el documento incluye una imagen, y con el atributo `src` se indica cuál es dicha imagen.

Asimismo, mediante parámetros adicionales se puede configurar el ancho y alto de la imagen, entre otras características más, por ejemplo el siguiente código:

```

```

indica la misma imagen que antes, pero en esta ocasión el atributo `width="100px"` determina que la imagen ocupa un ancho de cien píxeles y `height="80px"` indica que el alto es de ochenta píxeles.

2.4.9 Etiquetas `<audio>` y `<video>`

De forma similar, otros elementos multimedia como audios y videos pueden incluirse; sin embargo, estas etiquetas requieren de elementos que especifiquen la ubicación del archivo multimedia a presentar.

Por ejemplo, para representar un audio y un video, se utiliza código como el siguiente:

```
<audio>  
  <source src="musica.mp3" type="audio/mpeg">  
</audio>
```

```
<video>  
  <source src="pelicula.mp4" type="video/mp4">  
</video>
```

Con lo visto anteriormente, debe ser fácil entender qué es lo que ocurre con el código de las etiquetas anteriores, donde se tienen los elementos `<audio>` y `<video>`, que a su vez tienen como descendiente un elemento de tipo `<source>` para especificar la ubicación (`src`) y el formato (`type`) del archivo multimedia correspondiente.

La razón por la que se utiliza un elemento de tipo `<source>`, a diferencia de las imágenes, es que en los audios y videos se pueden especificar múltiples archivos en diferentes formatos, para que el navegador utilice el más adecuado. En particular, para el elemento `<video>`, se puede incluir un elemento de tipo `<track>` para especificar un archivo de subtítulos.

2.4.10 Etiquetas `<figure>` y `<figcaption>`

Al incorporar elementos como ilustraciones, infografías, fotografías, gráficas, mapas o cualquier elemento que aporte información visual, comúnmente denominado “figura” en un documento, HTML ofrece la etiqueta `<figure></figure>`.

Es importante destacar que una figura por sí misma no ofrece una representación gráfica; es su contenido lo que constituye la figura. Generalmente, las figuras se acompañan de una leyenda, que es un breve texto situado arriba o abajo de la figura y que describe elementos importantes de la misma. Para este fin HTML ofrece la etiqueta `<figcaption>`.

A continuación se presenta un ejemplo de figura en HTML:

```
<figure>
  
  <figcaption>Logo de iCartesiLibri.</figcaption>
</figure>
```

Lo que muestra lo siguiente:



2.4.11 Etiqueta `<div>`

Otra etiqueta que es primordial para estructurar documentos HTML es `<div></div>`, que se deriva de **divisor**, la cual no tiene una representación visual directa, sino que sirve para dividir el documento en secciones arbitrarias. Generalmente, los elementos `<div>` se utilizan como contenedores que agrupan múltiples elementos, por ejemplo, en los libros interactivos iCartesiLibri, una página se define como un `<div>` y todo el contenido incluido dentro de este elemento se considera parte del contenido de dicha página.

De igual manera, los `<div>` pueden usarse para crear cajas de texto que delimiten ejercicios, ejemplos o cualquier otro elemento que contribuya a la organización y estructura del libro.

2.4.12 Etiqueta de comentarios

En la autoría de libros digitales interactivos, los comentarios constituyen un elemento útil. Estos consisten en líneas de texto que son ignoradas por el navegador, ya que su propósito es proporcionar información, documentación o comentarios únicamente para el desarrollador, sin ser visibles para el lector. La creación de comentarios se realiza mediante el uso de `<!--` al inicio y `-->` al final. Por ejemplo:

```
<!-- Todo este texto es ignorado por el navegador y  
no es representado.
```

```
Sin importar que el comentario abarque varias  
líneas.-->
```

2.4.13 Etiqueta `
`

En el texto en HTML, los espacios en blanco (o tabuladores) múltiples son ignorados, y cuando se utilizan varios, solo se representa uno de ellos. De igual manera, los saltos de línea son completamente ignorados, por esta razón, para separar bloques de texto delimitados por saltos de línea, es recomendable utilizar párrafos. Aún así, existen formas adicionales que permite incluir múltiples saltos de línea, así como tantos espacios en blanco como sea necesario.

Para forzar un salto de línea, se utiliza la etiqueta de cierre automático o vacía, `
`, derivada de la palabra *break*. Su función es análoga a la obtenida al presionar la tecla “Enter” al escribir un documento en algún procesador de texto.

2.4.14 Entidades HTML

Para evitar que los espacios en blanco colapsen en un solo espacio, es necesario utilizar entidades HTML. Estas entidades son representaciones de caracteres especiales que no pueden escribirse directamente en el código HTML. Por ejemplo, las etiquetas HTML utilizan los caracteres `<` y `>`, y si se quieren mostrar en pantalla estos caracteres no se pueden usar directamente en el texto de un documento HTML, ya que inmediatamente delimitan una etiqueta, por lo que se deben representar mediante las entidades `<` para `<` y `>` para `>`. La estructura general de una entidad HTML es la siguiente: el carácter ampersand (`&`) al inicio, el valor de la entidad en el interior y un punto y coma (`;`) al final.

Para incluir un espacio en blanco que no se elimine, se utiliza la entidad ` ` proveniente del término *non-breaking space* o espacio irrompible. Por ejemplo el siguiente código usa algunas entidades HTML:

```
<p>Este es un párrafo con &nbsp;&nbsp;&nbsp;&nbsp; cuatro  
espacios en blanco, uno antes de las dos entidades  
y uno después. Y además se utilizan los caracteres  
de menor que (&lt;) y mayor que (&gt;).</p>
```

Resultado

Este es un párrafo con cuatro espacios en blanco, uno antes de las dos entidades y uno después. Y además se utilizan los caracteres de menor que (<) y mayor que (>).

La especificación oficial con todas las entidades HTML se encuentra en el siguiente vínculo: <https://html.spec.whatwg.org/multipage/named-characters.html#named-character-references>. Asimismo, la página <https://www.toptal.com/designers/htmlarrows/> también proporciona una referencia a los símbolos que pueden ser representados mediante entidades.

2.4.15 Etiquetas `<sup>` y `<sub>`

Dos etiquetas adicionales que ofrece HTML, son: `<sup>`, para la creación de superíndices, y `<sub>`, para la creación de subíndices. Por ejemplo:

```
<p>Este es un texto normal seguido de <sup>un texto en superíndice</sup> y después aparece <sub>un texto como subíndice</sub>.</p>
```

Lo que genera lo siguiente:

Resultado
Este es un texto normal seguido de ^{un texto en superíndice} y después aparece _{un texto como subíndice}

2.4.16 Etiqueta `<table>`

Para finalizar, una herramienta útil para organizar y estructurar información son las tablas, las cuales están compuestas por un conjunto de celdas ordenadas mediante filas o grupos de filas. Las tablas permiten mostrar datos de forma clara, facilitando la búsqueda y relación entre sus datos, gracias a que se crean asociaciones visuales entre las filas y columnas que las conforman.

En HTML, una tabla se inicia con la etiqueta `<table>` y dentro de esta, los datos se construyen fila por fila. Las filas son etiquetadas mediante `<tr>` proveniente de *table row* (fila de la tabla) y dentro de cada fila para especificar una celda, se utiliza la etiqueta `<td>` de *table data* (dato de la tabla). En el caso que se quieran destacar los encabezados de una tabla, se utiliza la etiqueta `<th>` que viene de *table header* (cabecera de la tabla), en lugar de utilizar `<td>`. Ejemplo:

```
<table>
  <tr>
    <th>Primera cabecera</th>
    <th>Segunda cabecera</th>
    <th>Tercera cabecera</th>
  </tr>
  <tr>
    <td>Primer valor de la primera fila</td>
    <td>Segundo valor de la primera fila</td>
    <td>Tercer valor de la primera fila</td>
  </tr>
  <tr>
    <td>Primer valor de la segunda fila</td>
    <td>Segundo valor de la segunda fila</td>
    <td>Tercer valor de la segunda fila</td>
  </tr>
  <tr>
    <td>Primer valor de la tercera fila</td>
    <td>Segundo valor de la tercera fila</td>
    <td>Tercer valor de la tercera fila</td>
  </tr>
</table>
```

Con el código anterior se crea un tabla con tres columnas y tres filas más su cabecera (cuatro filas en total):

Primera cabecera	Segunda cabecera	Tercera cabecera
Primer valor de la primera fila	Segundo valor de la primera fila	Tercer valor de la primera fila
Primer valor de la segunda fila	Segundo valor de la segunda fila	Tercer valor de la segunda fila
Primer valor de la tercera fila	Segundo valor de la tercera fila	Tercer valor de la tercera fila

Si bien la apariencia de la tabla es simple, el lenguaje HTML solo se preocupa por el mecanismo para organizar y especificar la información. Como se verá en el siguiente capítulo, los estilos CSS son los encargados de controlar la apariencia de los elementos HTML del documento.

2.5 Cómo comenzar a utilizar HTML

El lenguaje HTML, como puede observarse, no es muy difícil de entender, sin embargo, requiere conocer las etiquetas que lo componen y desarrollar familiaridad para utilizarlo de forma efectiva en la construcción de un documento. Con este primer acercamiento se busca reducir la brecha en la familiaridad con el lenguaje, permitiendo así que el autor dé el primer paso hacia la creación de libros interactivos con esta tecnología web.

Además de los conocimientos sobre lenguaje, se requieren dos herramientas o aplicaciones de computo. Una de ellas es un navegador web, que es la aplicación encargada de interpretar y presentar el resultado final del documento escrito en HTML.

La mayoría de los sistemas operativos ya tienen instalado un navegador de internet, como por ejemplo Microsoft Edge, Safari, Chromium, Firefox o Google Chrome. Se busca que la apariencia de los libros interactivos se mantenga uniforme en cualquier navegador web moderno, gracias al trabajo del núcleo de presentación de libros iCartesiLibri; de esta manera, el autor puede usar el navegador que más le convenga.

La segunda herramienta es una aplicación que permite escribir y editar el código HTML necesario para el libro, dado que HTML es un formato puramente textual se vuelve necesario utilizar una aplicación específica para editar su contenido.

2.5.1 Editores de texto plano o sin formato

Existen dos tipos principales de editores de texto: los editores de texto enriquecido y los de text plano (o sin formato). La diferencia fundamental radica en su capacidad para aplicar directamente un formato o estilo visual al texto.

Los editores de texto enriquecidos permiten aplicar diversos formatos al texto, como negrita, cursiva, colores, tamaños de fuente y las listas; esta capacidad hace que sean ideales para la creación de documentos con formato visual complejo y variado. Al utilizar un editor de texto enriquecido, el documento se presenta de la misma forma en que finalmente se visualizará, lo que facilita la creación de contenido profesional. La mayoría de los lectores estarán familiarizados con procesadores de texto como Microsoft Word, ejemplo claro de editor de texto enriquecido, donde todos los estilos visuales son representados directamente en el documento.

Por otro lado, los editores de texto plano (o sin formato), tal y como su nombre indica, se limitan a mostrar el texto tal cual fue escrito. No permiten el uso de estilos como negrita, cursiva, colores o las listas. Son ideales para escribir código fuente de aplicaciones y archivos donde no es importante la presentación visual inmediata. Su simplicidad hace que sean fáciles de usar y, además, los archivos creados con estos editores son portables, es decir, son compatibles con una gran variedad de sistemas y editores, ya que se tratan puramente de texto.

HTML, como lenguaje puramente textual, es leído e interpretado por un navegador de internet, siendo este último el encargado de proporcionar la representación visual de los estilos indicados por las etiquetas del documento. Para construir un archivo en HTML se requiere utilizar un editor de texto plano, es decir, uno que no incorpore directamente estilos visuales y que ofrezca la posibilidad de manipular directamente el contenido textual.

Al igual que los navegadores, la mayoría de los sistemas operativos ofrecen actualmente algún editor de texto plano ya instalado. Por ejemplo, MacOS proporciona **TextEdit**, el cual permite editar archivos en texto plano y además, admite la edición de algunos formatos de texto enriquecido. Linux, según la distribución utilizada, puede tener instalado algún editor como **Emacs**, **vi**, **nano** o **Kate**; en general, todas las distribuciones incluyen un editor como parte del sistema. Por último, para Windows se dispone del **Bloc de Notas**, y si bien la aplicación **WordPad** puede editar archivos de texto plano, no ofrece ninguna ventaja adicional y, al contrario, al mostrar opciones para agregar estilos, puede convertir fácilmente el formato plano a uno enriquecido con estilos.

Expuesto lo anterior, se puede comenzar la creación de un documento HTML sin necesidad de instalar herramientas adicionales, aún así, los editores de texto plano incluidos con los sistemas operativos Windows y MacOS suelen ser simples y carecen de las características o herramientas que faciliten la edición de código HTML. Por ello, se recomienda la instalación de un editor de código, ya que estos ofrecen ventajas como el coloreado de texto o resaltado de sintaxis para mejorar la legibilidad del código al separar visualmente diferentes partes del contenido; el autocompletado o autoinserción de elementos para ahorrar tiempo y reducir errores en la programación, como el cierre automático de una etiqueta HTML al momento de abrirla; búsqueda y reemplazo avanzados para trabajar con patrones complejos, por medio de expresiones regulares; soporte para múltiples lenguajes de programación (no solo HTML); entre otras opciones que facilitan la edición.

Adicionalmente, se pueden añadir algunas características a los editores de código mediante la inclusión de extensiones o complementos adicionales (*plugins*), lo cual aumenta las características básicas y ofrece una amplia personalización, permitiendo por ejemplo, la posibilidad de agregar un corrector ortográfico, un servidor local para pruebas y muchas opciones más.

Algunos editores de código recomendados son:

- **Notepad++** es un editor de texto gratuito para Windows, bastante competente y capaz. <https://notepad-plus-plus.org/>
- **Sublime Text** es un editor de código muy popular debido a su ligereza y velocidad de operación. Disponible para Windows, MacOS y Linux. <https://www.sublimetext.com/>

- **Kate** es un editor originario del mundo de Linux, que está ganando popularidad. También se encuentra disponible en Windows, MacOS y Linux. <https://kate-editor.org/>
- **Visual Studio Code** es un editor desarrollado por Microsoft, que ha sido adoptado por diversos programadores debido a su potencia, flexibilidad y amplia gama de extensiones. A pesar de estar desarrollado por Microsoft, está disponible para Windows, MacOS y Linux. <https://code.visualstudio.com/>

En situaciones donde el autor no tenga los permisos necesarios para instalar programas adicionales en su equipo, varios editores de código ofrecen versiones portables; estas se pueden ejecutar sin necesidad de una instalación, y generalmente basta con descomprimir un archivo y ejecutarlos desde ahí. Algunos ejemplos son [Notepad++ portable](#) o [SublimeText portable](#). Además, Visual Studio Code dispone de una versión online accesible a través del sitio <https://vscode.dev/>, esta versión funciona en cualquier navegador moderno y permite editar archivos locales en la computadora, sin necesidad de enviarlos a algún servidor.

Estas herramientas aunque puedan parecer intimidantes al principio, con el uso es posible adquirir familiaridad y soltura, aprovechando cada vez más las diversas características y ventajas que proporcionan. Aún así, cada autor debe decidir con que herramienta se siente más cómodo para el desarrollo del libro interactivo, y la única forma de tomar una decisión adecuada es probando las diferentes opciones disponibles; siendo las anteriores, solo una pequeña muestra de las múltiples herramientas existentes.



3. Nociones básicas de CSS

El lenguaje HTML es utilizado para dar estructura a un documento, es decir, permite especificar el tipo de información que presenta el texto y así diferenciar cada parte del documento. Un archivo `.html` que contenga únicamente código HTML, puede ser visualizado en un navegador de internet, presentando un formato visual predeterminado. Este formato, generalmente considerado aburrido y plano, depende del navegador y sistema operativo en el que se visualice. En términos estrictos, un documento puramente en HTML no posee información sobre la representación visual de sus elementos, al menos no con las versiones modernas de HTML.

3.1 ¿Qué es CSS?

Para conferirle un estilo visual propio a un documento HTML, se utiliza otro lenguaje denominado CSS (*Cascading Style Sheets*), o hojas de estilo en cascada.

El concepto de “hoja de estilo” se define como un conjunto de reglas que determinan las características visuales de una etiqueta. Estas reglas se establecen mediante atributos específicos que controlan el estilo del elemento, incluyendo aspectos como el color, el tamaño y la presencia o ausencia de bordes, entre otros atributos más.

Por otra parte, el término “en cascada” hace referencia al proceso por el cual los estilos se aplican desde los elementos progenitores hacia sus descendientes, similar a una cascada fluyendo de arriba hacia abajo. Este comportamiento indica que los estilos se heredan a través de la jerarquía del documento HTML.

La sintaxis para especificar un atributo de estilo sigue un formato sencillo, `nombre_del_atributo : valor_del_atributo;`, donde el `nombre_del_atributo` es generalmente una palabra determinada por el propio lenguaje, seguido por dos puntos (:), y `valor_del_atributo` es un valor o conjunto de valores asociados al atributo.

Las reglas CSS pueden especificarse de dos maneras: como un atributo de un elemento HTML o como un bloque independiente.

3.2 Reglas CSS en línea

Cuando se especifican como un atributo, se utiliza la propiedad `style` del elemento HTML, por ejemplo:

```
<p style="color:red;">un texto rojo</p>
```

En el ejemplo, `style` es una propiedad o atributo asociado con el elemento `<p>` que se está definiendo, y su valor es la regla `color:red;`, que indica al navegador que utilice el color rojo (`red`) para dibujar el texto del párrafo. Si se desea especificar múltiples reglas CSS dentro del estilo de un elemento, estas deben separarse por un punto y coma (`;`), por ejemplo:

```
<p style="color:red; font-size:12px; border:2px solid black;">un texto rojo con un borde negro</p>
```

Aquí el párrafo tiene tres reglas CSS asociadas, a saber: `color:red` para el color del texto del párrafo, `font-size:12px` que determina el tamaño de la fuente tipográfica o tipo de letra siendo en este caso de doce píxeles, y `border:2px solid black` especifica un borde de 2 píxeles de ancho, sólido y de color negro alrededor de la caja que contiene el párrafo.

Resultado

un texto rojo con un borde negro

Es importante mencionar que las reglas especificadas de esta manera están acotadas por comillas dobles (`"`) o comillas sencillas (`'`), ya que representan el valor asociado al parámetro `style` del elemento. Así mismo, estas reglas solo afectan al elemento al que se le asignan, y no a todos los elementos de un tipo, por ejemplo, en un documento solo el párrafo con las reglas anteriores se presentará de color rojo con un borde negro, mientras que todos los demás párrafos tendrán su propia representación visual. Es por eso que a esta forma de especificar estilos se le conoce como: “estilos en línea” o *inline*.

3.3 Reglas CSS por bloque

La aplicación de estilos de esta manera puede volverse repetitiva si existen muchos (digamos cientos de) elementos que utilizan el mismo conjunto de reglas. Por esa razón, existe una forma más general de especificar reglas CSS, con la finalidad de aplicarlas a múltiples elementos simultáneamente. Esta forma se conoce como reglas CSS por bloque. La forma que tienen estas reglas es:

```
selector {  
    nombre_del_atributo_1 : valor_del_atributo_1;  
    nombre_del_atributo_2 : valor_del_atributo_2;  
    nombre_del_atributo_3 : valor_del_atributo_3;  
}
```

Como se observa, un bloque de reglas está delimitado por llaves (`{` y `}`), lo que equivale a las comillas en las reglas definidas como atributo. El `selector`, por su parte, constituye una forma de determinar a qué etiquetas se aplica el conjunto de reglas especificado. Si bien existen diversas formas de construir selectores, en este apartado solo se examinarán las más comunes.

3.3.1 Selectores CSS

Una forma de especificar el selector de un bloque de reglas es mediante el nombre de una etiqueta HTML, por ejemplo:

```
p {  
    color:red;  
}
```


Es un bloque de reglas que se aplica a todos los párrafos del documento, es decir, todos los elementos de tipo `<p>`. De esta manera, las reglas pueden aplicarse a cualquier elemento que pueda etiquetarse directamente con HTML, en lugar de hacerlo elemento por elemento.

Otra forma de determinar el `selector` es por medio de lo que se conoce como una clase CSS. Esta clase se refiere a la asociación de reglas a los elementos HTML que posean un atributo `class` con un valor dado por el autor. Por ejemplo, si se tiene el siguiente elemento HTML:

```
<p class="especial">este es un párrafo especial</p>
```

Se puede asociar el siguiente bloque de reglas CSS:

```
.especial {  
  color: blue;  
}
```

Es importante notar que el selector en este caso inicia con un punto (.) seguido del mismo valor utilizado en el atributo `class` del párrafo, esto permite que los elementos marcados con la misma clase utilicen el mismo estilo visual.

El mismo estilo puede ser utilizado en diferentes tipos de etiquetas utilizando una clase CSS, siempre y cuando compartan el mismo valor de `class`.

Similar a lo anterior, pueden aplicarse estilos solo a un elemento con un identificador específico, es decir, una etiqueta con el atributo `id`. Por ejemplo, al párrafo:

```
<p id="parrafo_unico">este es un párrafo único</p>
```

Se le puede asociar el siguiente bloque de reglas CSS:

```
#parrafo_unico { color: pink; }
```

En este caso, el selector inicia con el símbolo de almohadilla, gato, numeral o *hashtag* (`#`). Es importante mencionar que los identificadores deben ser únicos y no repetirse dentro del documento, por lo que este bloque de reglas se aplica únicamente a un elemento específico.

Asimismo, debe resaltarse que tanto el valor de la clase como el del identificador no pueden contener espacios en blanco, por eso, en los ejemplos anteriores se utiliza el guión bajo (`_`) para separar las palabras. Generalmente, es buena práctica solo usar caracteres alfanuméricos (letras y números), incluido el guión bajo, para estos identificadores.

Utilizando la coma (`,`) como separador, es posible asignar un bloque de reglas CSS a múltiples selectores. Por ejemplo:

```
.clase1, .clase2, .clase3 {  
  color: black;  
  background-color: pink;  
}
```

El código anterior aplica un conjunto de reglas idénticas a las tres clases (`clase1`, `clase2` y `clase3`), a saber, el color del texto en negro (`color: black`) y el color de fondo rosa (`background-color: pink`).

Para lograr mayor precisión en el estilo, se pueden utilizar los conocimientos sobre la estructura jerárquica del documento HTML, y ser más específicos en la construcción del selector. Por ejemplo, si se desea dar formato a todos los elementos etiquetados con `` dentro de un párrafo con `class="resaltar"`, algo como lo siguiente:

```
<p class="resaltar">Este párrafo tiene información
<span>importante</span>.</p>
```

Se puede utilizar el siguiente bloque de reglas CSS:

```
.resaltar span {
  color: red;
  background-color: yellow;
}
```

En este selector, `.resaltar span`, primero se selecciona el elemento con la clase `resaltar`, y luego, mediante un espacio en blanco, se especifica que se deben aplicar las reglas a los elementos `span` que son descendientes de un elemento con la clase `resaltar`. Es importante destacar que la relación jerárquica en los selectores se especifica del ancestro al descendiente. En este caso, las reglas se aplican a los elementos `` que están dentro de un elemento con la clase `resaltar`.

Actualmente, no es posible crear un selector para ancestros específicos de un descendiente debido a la naturaleza en “cascada” de los estilos CSS, que se aplican de arriba hacia abajo (de ancestros a descendientes) y no al revés.

Dónde se almacenan las reglas CSS por bloque

Estas reglas por bloque pueden almacenarse de dos formas diferentes. Una es dentro de una etiqueta `<style>` que sea parte del propio documento HTML, usualmente como descendiente del elemento `<head>`. La otra forma es mediante un archivo externo.

En el primer caso, las reglas se pueden definir dentro del elemento `<head>` de la siguiente manera:

```
<head>
  <style>
    p {
      color: black;
      font-size: 18px;
    }

    p b {
      color: red;
    }
  </style>
</head>
```

Con lo visto previamente, debe ser claro lo que significan estas reglas.

En el segundo caso, es decir, al utilizar un archivo externo, el contenido de la etiqueta `<style>` se coloca directamente en un archivo, usualmente con la extensión `.css`. Este archivo se vincula desde el elemento `<head>` mediante una etiqueta `<link>`, por ejemplo:

```
<head>
  <link rel="stylesheet" href="style.css"
  type="text/css">
</head>
```

El atributo `rel` (de *relationship*) es utilizado por la etiqueta `<link>` para especificar el tipo de elemento al que vincula; en este caso, una hoja de estilo (`stylesheet`). El atributo `href` es similar al utilizado por los hipervínculos (`<a>`) y determina la ruta donde se encuentra el archivo con los estilos. Finalmente `type` especifica la forma en que el documento está codificado, en este caso del tipo textural y en particular es un CSS (`text/css`).

3.4 Algunos atributos CSS

Una vez presentada la forma en la que se escriben y construyen las reglas CSS, es posible describir algunos de los atributos a utilizar para configurar la presentación del documento HTML. Previamente se han presentado las siguientes propiedades:

3.4.1 color

Esta propiedad permite determinar el color del texto del elemento.

Existen varias formas que se utilizan para especificar colores en CSS. En ejemplos pasados, se han utilizado valores como `red` o `black`. Esta es una forma de referenciar los colores por nombres, teniendo una lista de nombres de colores soportados por la mayoría de navegadores. Se tiene una lista con aproximadamente 150 colores, soportados por los navegadores modernos.

aliceblue	antiquewhite	aqua	aquamarine
azure	beige	bisque	black
blanchedalmond	blue	blueviolet	brown
burlywood	cadetblue	chartreuse	chocolate
coral	cornflowerblue	cornsilk	crimson
cyan	darkblue	darkcyan	darkgoldenrod
darkgray	darkgreen	darkgrey	darkkhaki
darkmagenta	darkolivegreen	darkorange	darkorchid
darkred	darksalmon	darkseagreen	darkslateblue
darkslategray	darkslategrey	darkturquoise	darkviolet
deeppink	deepskyblue	dimgray	dimgrey
dodgerblue	firebrick	floralwhite	forestgreen
fuchsia	gainsboro	ghostwhite	gold
goldenrod	gray	green	greenyellow
grey	honeydew	hotpink	indianred
indigo	ivory	khaki	lavender

lavenderblush	lawngreen	lemonchiffon	lightblue
lightcoral	lightcyan	lightgoldenrodyellow	lightgray
lightgreen	lightgrey	lightpink	lightsalmon
lightseagreen	lightskyblue	lightslategray	lightslategrey
lightsteelblue	lightyellow	lime	limegreen
linen	magenta	maroon	mediumaquamarine
mediumblue	mediumorchid	mediumpurple	mediumseagreen
mediumslateblue	mediumspringgreen	mediumturquoise	mediumvioletred
midnightblue	mintcream	mistyrose	moccasin
navajowhite	navy	oldlace	olive
olivedrab	orange	orangered	orchid
palegoldenrod	palegreen	paleturquoise	palevioletred
papayawhip	peachpuff	peru	pink
plum	powderblue	purple	rebeccapurple
red	rosybrown	royalblue	saddlebrown
salmon	sandybrown	seagreen	seashell
sienna	silver	skyblue	slateblue
slategray	slategrey	snow	springgreen
steelblue	tan	teal	thistle
tomato	turquoise	violet	wheat
white	whitesmoke	yellow	yellowgreen

Otra manera de especificar colores es por medio del modelo de color RGB, el cual, es un modelo aditivo de color que utiliza combinaciones de tres colores primarias, el rojo (R), el verde (G) y el azul (B). La intensidad de cada color primario se expresa como un valor entero entre 0 y 255. Por ejemplo, el color rojo puro se representaría como (255, 0, 0), mientras que el blanco se representaría como (255, 255, 255) y el negro como (0, 0, 0).

Para expresar un color en el sistema RGB, existen dos métodos a disposición.

El primero implica el uso de la palabra “rgb”, seguida por paréntesis que contienen los valores numéricos correspondientes a cada componente del color. Por ejemplo: `rgb(255,0,0)` equivale al color `red` (rojo); `rgb(0,255,0)` equivale al color `green` (verde); y `rgb(0,0,255)` equivale al color `blue` (azul).

El segundo método es más compacto, aunque implica expresar los valores de cada componente mediante su representación hexadecimal.

Por ejemplo, para especificar el color `darkorange` (un color naranja oscuro), cuyos valores RGB son (255, 140, 0), se debe seguir un proceso. Primero, cada componente debe expresarse en hexadecimal, se convierte 255 a `ff`, 140 a `8c` y 0 a `00`, es importante que cada componente tenga dos dígitos. Con esta información, el color `rgb(255,140,0)` se expresa en hexadecimal como `#ff8c00`. Este código se conforma por el signo de número (`#`), seguido de los dos dígitos hexadecimales del componente rojo, luego los dos dígitos del componente verde y finalmente los correspondientes al azul.

La cantidad de opacidad de un color puede ser especificada utilizando la palabra clave “rgba”, en lugar de solo “rgb”. Con esto se adiciona el canal alfa del color, permitiendo especificar un valor entre 0 y 1, donde 0 representa transparencia completa y 1, opacidad total. Por ejemplo, `rgba(255, 0, 0, 0.5)` genera un color rojo medianamente transparente.

Además del modelo de colores RGB, es posible usar el modelo HSL, *Hue* (tono), *Saturation* (saturación) y *Lightness* (brillo), que ofrece una forma más intuitiva de definir colores que el modelo RGB. El tono, representado como un ángulo en el círculo cromático (por ejemplo, rojo = 0°, verde = 120°, azul = 240°). La saturación, indica la intensidad del color, del 0% (gris) al 100% (color puro). Mientras que el brillo, determina la luminosidad del color, del 0% (negro) al 100% (blanco).

HSL permite que se creen armonías de colores fácilmente al mantener el tono constante y ajustar brillo y saturación.

La forma en que se especifica un color es mediante la palabra clave “hsl”. Por ejemplo, el color `darkorange` en este modelo sería especificado como `hsl(33, 100%, 50%)`, lo que significa que tiene un tono de 33°, una saturación del cien por ciento y un brillo del cincuenta por ciento.

Asimismo, la versión “hsla” se utiliza para construir un color en este modelo, añadiendo la opacidad como una cuarta componente. De forma similar a lo explicado con “rgba”, esta versión también permite modificar el nivel de opacidad del color.

3.4.2 font-size

Especifica el tamaño de la fuente tipográfica, refiriéndose sencillamente a la altura de las letras.

3.4.3 background-color

Indica el color del fondo del elemento.

3.4.4 border

Esta propiedad abreviada se refiere a una regla que especifica varias propiedades al mismo tiempo. Generalmente, cuando una propiedad CSS recibe varios valores (separados por espacios), suele tratarse de una propiedad **abreviada**. En este caso, `border` especifica simultáneamente las propiedades `border-width`, `border-style` y `border-color`.

3.4.5 border-width

Esta propiedad configura el ancho del borde del elemento.

3.4.6 border-style

La propiedad `border-style` determina el estilo del borde. Sus valores permitidos son:

- `none` o `hidden` para indicar que no tiene un borde visible,
- `dotted` para crear un borde punteado,
- `dashed` para un borde punteado con segmentos de línea,

- `solid` para dibujar el borde de forma continua o sólida,
- `double` para dibujar dos líneas sólidas como borde,
- `groove`, `ridge`, `inset` y `outset` crean un borde con un estilo tridimensional, cada uno con características visuales propias.

3.4.7 border-color

Esta propiedad especifica el color del borde.

Otras propiedades útiles son:

3.4.8 border-radius

Esta propiedad permite especificar el radio de un círculo que se coloca en las esquinas de un elemento, permitiendo así la creación de esquinas redondeadas en elementos que dibujan un borde, o en imágenes.

3.4.9 font-family

Esta propiedad permite seleccionar la fuente tipográfica que utiliza el elemento. Generalmente, se pueden utilizar los valores `serif`, `sans-serif` y `monospace` para utilizar las tipografías de ese tipo proporcionadas por el propio navegador. De igual manera se puede seleccionar una tipografía por nombre, colocando su nombre entre comillas (por ejemplo, `"Arial"`), en este caso, el sistema busca si dicha tipografía está disponible en el sistema operativo del usuario; si la encuentra, se utiliza; caso contrario, se utiliza una tipografía genérica.

Más adelante en la parte de personalización del libro, se proporcionará más información sobre tipografías, así como el mecanismo utilizado para incluir tipografías externas y garantizar su uso aunque el lector no tenga el tipo de letra instalado en su sistema.

3.4.10 text-align

Esta propiedad determina la alineación horizontal del texto, es decir, si el texto se presenta alineado a la izquierda (`left`), a la derecha (`right`), centrado (`center`), o agregando espacio entre las palabras para que los bordes de cada línea de texto estén alineados a los márgenes, es decir, justificado (`justify`).

3.4.11 background

Esta propiedad abreviada engloba múltiples propiedades que permiten configurar el comportamiento de la apariencia del fondo de un elemento. Entre las más utilizadas se encuentran las propiedades `background-color` y `background-image`.

3.4.12 background-color

Esta propiedad especifica un color sólido que se utiliza para cubrir toda el área del fondo del elemento.

3.4.13 background-image

Esta propiedad define una o varias imágenes utilizadas para dibujar el fondo.

En caso de utilizar múltiples imágenes, estas deben estar separadas por comas. Las imágenes deben incluirse mediante una `url`, siguiendo la siguiente sintaxis: `background-image : url('imagen1.png');`. Es importante destacar que se puede especificar tanto el color como la imagen del fondo, siendo aplicado primero el color y luego la imagen sobre él.

Asociados a la imagen de fondo se tienen algunas propiedades adicionales para su configuración:

3.4.14 `background-position`

Esta propiedad determina la posición en la que se coloca una imagen de fondo. Se pueden utilizar valores como: `top` (arriba), `bottom` (abajo), `left` (izquierda), `right` (derecha) o `center` (centrada) para ubicar la imagen. También es posible especificar la posición mediante números, por ejemplo: `background-position: 25px 150px;` muestra la imagen veinticinco píxeles a la izquierda y ciento cincuenta píxeles hacia abajo.

3.4.15 `background-repeat`

La propiedad `background-repeat` determina si la imagen de fondo se repite o no tanto en dirección vertical como horizontal. Esta característica es útil cuando se incluye una imagen de fondo más pequeña que el elemento, permitiendo crear de forma sencilla un patrón repetitivo. Esta propiedad acepta uno o dos valores, al utilizar un valor, se configura simultáneamente la repetición horizontal y vertical; cuando se utilizan dos valores, el primero configura la repetición horizontal y el segundo la repetición vertical. Los valores aceptados son:

- `repeat`: La imagen se repite tantas veces como sea necesario para cubrir completamente el elemento.
- `no-repeat`: Se evita que la imagen se repita.
- `space`: La imagen se repite procurando evitar que alguna de las repeticiones se corte debido a que no cabe completamente en el elemento. En este caso, en lugar de cortar la imagen, se agrega un espacio entre las repeticiones.

3.4.16 `background-size`

La propiedad `background-size` configura el tamaño de la imagen de fondo. Es posible utilizar un par de palabras clave:

- `cover`: La imagen se agranda con la intención de cubrir completamente el elemento, sin modificar la proporción de la imagen, aunque esto puede implicar que parte de la imagen se recorte para lograr cubrir por completo el área del elemento.
- `contain`: La imagen se agranda tanto como sea posible para cubrir el elemento, pero evitando cortar o distorsionar su proporción, manteniéndose siempre dentro del área del elemento.

También es posible proporcionar dos valores numéricos que determinan el tamaño de la imagen, siendo el primer valor el ancho y el segundo el alto. Por ejemplo, `background-size: 20px 100px`, escala la imagen de fondo a un tamaño de veinte píxeles de ancho por cien de alto.

3.5 Posición y tamaños de los elementos

Antes de continuar con otros elementos de CSS, es importante mencionar algunas características que presentan los elementos HTML. Estas características se refieren al tamaño y al lugar que ocupan los elementos en el documento, así como a las unidades que pueden utilizarse para los elementos que requieren valores numéricos.

Los navegadores de internet son (de forma súper simplificada) aplicaciones especializadas en interpretar código HTML y CSS para crear una representación visual de un documento, para ello, los elementos que tienen una representación gráfica, como los textos, imágenes y demás elementos multimedia, deben ocupar un cierto espacio o área dentro de la ventana del navegador. Para ello, los elementos que tienen una representación gráfica, como los textos, imágenes y demás elementos multimedia, deben ocupar un cierto espacio o área dentro de la ventana del navegador. El espacio que ocupan estos elementos es determinado por un conjunto de reglas internas del navegador, muchas de las cuales pueden ser configuradas con atributos o propiedades CSS.

Por ejemplo, recordando lo visto anteriormente sobre las imágenes, los atributos `width` y `height` se utilizan para especificar el ancho y el alto, respectivamente. Estos valores también pueden ser configurados por medio de reglas CSS en lugar de atributos. En cuanto a los textos, el tamaño de la letra (`font-size`) se utiliza para configurar la altura de la tipografía usada, y con esta información, el navegador calcula internamente el área que debe cubrir cada bloque de texto.

3.5.1 Unidades de medida en CSS

Considerando lo anterior, es importante conocer cuáles son las unidades utilizadas por un navegador para determinar los tamaños de los elementos. Estas se dividen en dos grandes grupos: las unidades absolutas y las relativas.

Unidades absolutas

Las unidades absolutas son estáticas y dependen únicamente del tamaño que ocupa un píxel en la pantalla. El píxel (`px`) es la unidad principal y más utilizada. Otras unidades absolutas incluyen el centímetro (`cm`), equivalente aproximadamente a 37.8 píxeles, el milímetro (`mm`) alrededor de 3.78 píxeles, la pulgada (`in`) aproximadamente 96 píxeles y el punto (`pt`) que equivale a 1/72 pulgadas.

Es importante mencionar que estas unidades con equivalente físico no se utilizan con frecuencia, ya que el tamaño físico de un píxel varía según la pantalla. Por ejemplo, un centímetro especificado por CSS en una pantalla pequeña (como la de un teléfono) no abarcará la misma longitud que un centímetro en el mundo real. Debido a esta inconsistencia, se recomienda evitar su uso en el diseño de páginas web, y utilizar directamente píxeles en caso de requerirse una unidad absoluta.

Unidades relativas

Las unidades relativas en CSS dependen del tamaño asignado a otras longitudes o atributos, así como de las dimensiones de los ancestros de un elemento.

El porcentaje (%) requiere que el ancestro directo del elemento tenga una dimensión absoluta calculada para determinar su porcentaje. Por ejemplo, si un elemento tiene un ancho de 100px y dentro de él se encuentra otro con un ancho del 50%, el ancho final de este último será de 50px. Las unidades `vw` y `vh` son porcentajes del tamaño de la ventana del navegador, por ejemplo, si la ventana tiene un tamaño de 1024x768, 10vw es igual a 102.4px, y 10vh a 76.8px.

La unidad `em` corresponde al tamaño de la tipografía (el valor especificado con `font-size`) del elemento progenitor, entonces por ejemplo, si un elemento utiliza `font-size: 20px` y tiene un descendiente que especifica un ancho de 10em, el valor final del ancho sería de $10 \times 20 = 200$ px. De forma similar, la unidad `rem` también utiliza el valor de `font-size`, pero este valor se obtiene directamente del nodo raíz del documento (`<html>`). La unidad `ch` corresponde al ancho que tiene el número cero (0) con el tamaño de letra especificado, este valor depende también de la tipografía utilizada, ya que una misma letra (en este caso, número) ocupa diferentes tamaños dependiendo de la fuente y estilo tipográfico.

Al interpretar el código HTML y CSS, los navegadores calculan los tamaños de todos los elementos y construyen cajas a su alrededor para acomodar cada elemento en su lugar correspondiente. Este proceso se conoce como modelo de caja (o *Box Model*) y representa estas cajas que envuelven el contenido de un elemento HTML. Las dimensiones de la caja que rodean a un elemento son inicialmente determinadas por el tamaño propio del contenido del elemento, en otras palabras, el tamaño inicial de la caja coincide con el tamaño abarcado por el elemento.

Adicionalmente a este tamaño, tres propiedades se utilizan para ampliar o extender el área ocupada por la caja, esto valores son `padding`, `border` y `margin`.

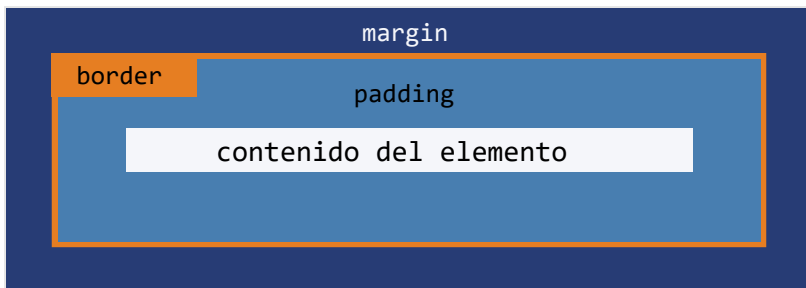


Figura 3.1. Propiedades que afectan el área que ocupa un elementos.

3.5.2 padding

El `padding` es un margen creado entre el contenido del elemento y un potencial borde, ya sea que este borde se dibuje o no; este atributo permite crear una separación (o relleno) interna dentro del elemento.

3.5.3 border

La propiedad `border`, vista previamente, corresponde a un borde alrededor del propio elemento, independientemente de si se dibuja o no. El borde suele corresponder al límite que puede abarcar el contenido de una etiqueta.

3.5.4 margin

La propiedad `margin` proporciona un margen externo, alrededor del borde del elemento, y es útil para crear una separación entre elementos cercanos, manteniendo así un espacio entre ellos.

Por ejemplo, la siguiente regla:

```
p {  
  margin: 100px;  
  border: 8px solid purple;  
  padding: 20px;  
}
```

Aplica un estilo a todos los elementos de tipo `<p>` para que presenten un margen de cien píxeles, un borde de ocho píxeles de ancho dibujado como una línea sólida de color púrpura, y una separación interna de veinte píxeles. Dando resultado a algo como lo siguiente:

Resultado

Este es un párrafo estilado con un margen, un borde y una separación o *padding*.

El ejemplo anterior presenta algunas características importantes que deben ser consideradas. Principalmente, se debe notar la diferencia entre las propiedades `margin` y `padding`, ya que, mientras `margin` separa un elemento del contenido circundante, `padding` crea una separación interna entre el contenido y borde del elemento. En ocasiones, puede resultar difícil distinguir ambas propiedades, especialmente cuando el elemento no posee un borde visible, ya que tanto `margin` como `padding` contribuyen a crear una separación, pero esta separación es conceptualmente diferente, y debe comprenderse su diferencia para utilizarlas de forma efectiva al momento de estilar un elemento.

Es importante destacar que las propiedades aplican el valor asociado a los cuatro posibles lados de la caja (arriba, abajo, izquierda y derecha). Por ejemplo, en el caso de `border`, se puede observar que hay un borde de 8 píxeles en la parte superior, otro de igual tamaño en la parte inferior y de igual manera para los laterales.

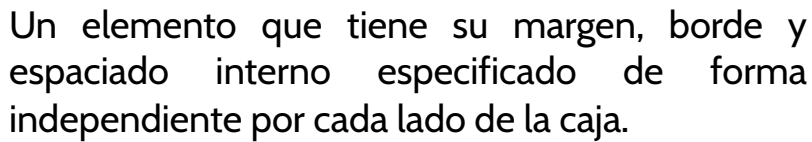
Tanto `margin` como `padding` (así como otras propiedades CSS) pueden recibir uno, dos o cuatro valores para especificar por separado los valores asignados a los cuatro lados de la caja, por otra parte, la propiedad `border` no admite múltiples valores ya que es un propiedad abreviada, en su lugar, se pueden proporcionar varios valores a las propiedades `border-width`, `border-style` y `border-color`, o utilizar las propiedades abreviadas (similar a `border`) `border-left`, `border-right`, `border-top` y `border-bottom` para configurar el borde de cada lado del elemento.

Por ejemplo las siguientes reglas CSS:

```
p {  
  margin: 10px 20px;  
  padding: 5px 20px 15px 50px;  
  border-top: 10px ridge blue;  
  border-right: 5px dashed pink;  
  border-bottom: 1px double green;  
  border-left: 8px solid red;  
}
```

Muestran el siguiente elemento:

Resultado



Un elemento que tiene su margen, borde y espaciado interno especificado de forma independiente por cada lado de la caja.

De forma equivalente el siguiente conjunto de reglas definen el mismo estilo:

```
p {  
  margin: 10px 20px;  
  padding: 5px 20px 15px 50px;  
  border-width: 10px 5px 1px 8px;  
  border-style: ridge dashed double solid;  
  border-color: blue pink green red;  
}
```

Es importante notar que cuando se utilizan dos valores al especificar un tamaño, el primer valor define el tamaño de arriba y abajo, mientras que el segundo valor define los tamaños de la izquierda y derecha, es decir, los valores se configuran por pares. Por ejemplo, `margin: 10px 20px`, significa que el margen de arriba y de abajo tienen un tamaño de 10 píxeles y los márgenes izquierdo y derecho tienen un valor de 20 píxeles.

Cuando se utilizan cuatro valores, el orden de especificación es el siguiente: primero el valor de arriba, luego el de la derecha, posteriormente el de abajo y por último el de la izquierda. Imaginando el orden como las horas de un reloj de manecillas, el primer valor correspondería a las 12, el segundo a las 3, el tercero a las 6 y el cuarto a las 9. Por ejemplo: `border-color: blue pink green red`, significa que el borde de arriba tiene el color azul (`blue`), el borde de la derecha rosa (`pink`), el borde de abajo es verde (`green`) y por último el borde izquierdo es rojo (`red`).

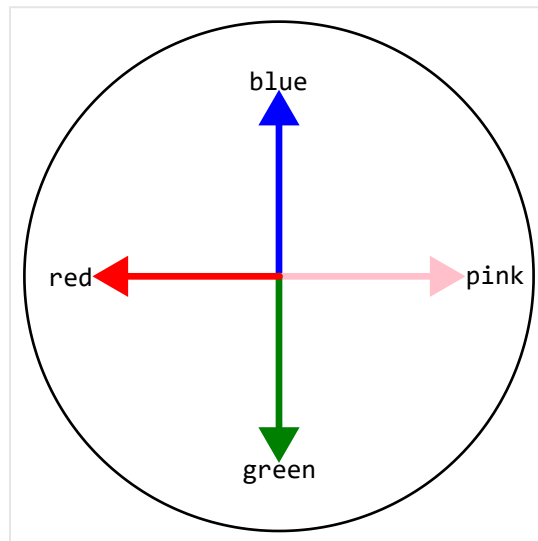


Figura 3.2. Imagen de reloj que muestra el orden en que se especifican 4 valores en la regla: `border-color: blue pink green red`.

3.5.5 box-shadow

Tras conocer las cajas que envuelven a los elementos HTML, es posible hablar de una propiedad que permite crear una sombra al rededor de dicha caja.

Esta propiedad, denominada `box-shadow`, puede recibir dos, tres o cuatro valores numéricos, así como el color de la sombra, y opcionalmente se puede usar la palabra clave `inset` para indicar que la sombra se dibuja dentro de la caja y no fuera de ella.

Los valores numéricos utilizados tienen significados específicos: dos valores determinan un desplazamiento en la dirección horizontal (el primer valor) y un desplazamiento en la dirección vertical (el segundo valor). Este desplazamiento corresponde a la dirección en la que se proyecta la sombra. El tercer parámetro numérico indica el grado de enborronamiento de la sombra, mientras que el cuarto parámetro determina la extensión o dispersión de la sombra.

A continuación, se presenta un ejemplo:

```
p {  
    box-shadow: 4px 10px 2px 5px coral;  
}
```

El estilo anterior indica que la sombra es desplazada cuatro píxeles en la horizontal y diez píxeles en la vertical. Además, se emborrona con un radio de dos píxeles; si se usa un valor de 0px entonces la sombra no se verá borrosa, y entre más grande sea el tercer parámetro, más borrosa se percibe la sombra.

Posteriormente se tiene una dispersión o extensión de cinco píxeles, lo que hace que la sombra se vea más grande de lo que realmente es, ya que este valor incrementa el tamaño de la sombra. Por último, se utiliza un color `coral` para dibujar la sombra.

Dando como resultado lo siguiente:

Resultado

Un párrafo con una sombra a su alrededor.



El atributo opcional `inset` puede utilizarse para crear un efecto tridimensional en un elemento. Por ejemplo:

```
p {  
  box-shadow: 0px 0px 10px coral inset;  
}
```

Crear un párrafo que se ve de la siguiente manera:

Resultado

Un párrafo con una sombra en su interior.



En el ejemplo anterior, no hay ningún desplazamiento horizontal ni vertical, por lo que se utilizan el valor de `0px`. La sombra se emborrona con un radio de diez píxeles. Y al utilizar la palabra clave `inset`, la sombra se dibuja en el interior del elemento en lugar del exterior.

Resultado

Este es un párrafo cuyo texto tiene una sombra gris.

3.5.7 position

Durante la construcción de la representación visual de un documento HTML, como se mencionó anteriormente, el navegador construye cajas que envuelven el contenido y las coloca de forma consecutiva conforme van apareciendo en el documento. Este es el flujo predeterminado en el que se muestra una página HTML: una disposición lineal que va de arriba hacia abajo, presentando los elementos HTML en el orden en que fueron definidos.

Debido a que este es el comportamiento predeterminado para acomodar los elementos del documento, generalmente los sitios o páginas web se presentan en una área vertical larga y la visualización del contenido completo se realiza mediante un desplazamiento vertical.

La propiedad CSS `position`, permite modificar la forma en que los elementos se colocan dentro del flujo normal de presentación, abriendo la posibilidad de ubicar un elemento en una posición específica. Esta propiedad puede recibir valores diferentes que afectan dónde y cómo se colocan los elementos, siendo estos:

- `static`: Este valor es el predeterminado para todos los elementos y indica al navegador que utilice el orden usual (de arriba hacia abajo y de izquierda a derecha) para colocarlos.

Los siguientes valores de `position` al ser utilizados permiten utilizar las propiedades `left`, `right`, `top` y `bottom`, para realizar desplazamientos a la izquierda, derecha, arriba y abajo.

- **relative**: Al utilizar este valor, el elemento se coloca de la misma forma que lo haría con **static**. Sin embargo, al usar las propiedades de desplazamiento, se puede mover el elemento de forma relativa a su posición inicial.
- **absolute** Este valor retira al elemento del flujo normal en el que se encuentra y lo coloca respecto al primer ancestro que tenga una posición relativa. En caso de que no exista dicho ancestro, se toma como referencia el nodo raíz del documento. Los desplazamientos en este caso son absolutos respecto al ancestro.
- **fixed**: Al igual que **absolute**, este valor retira al elemento del flujo normal. Sin embargo, su posición y desplazamientos se determinan respecto a la pantalla.

3.5.8 display

Finalmente, se debe mencionar que, al igual que a las imágenes, se puede asignar un tamaño a otros elementos mediante las propiedades **width** y **height**, para el ancho y el alto respectivamente. Es importante destacar que HTML, a través de la propiedad CSS **display**, distingue entre los elementos que forman bloques, por ejemplo los párrafos o los encabezados; de los elementos que se presentan dentro de una línea o *inline*, como los hipervínculos, los elementos para indicar negrita o itálica. De forma simplificada, la distinción principal radica en que los elementos de bloque se separan por saltos de línea, mientras que los elementos en línea no lo hacen; y adicionalmente, solo a los elementos de tipo bloque se les permite asignar su ancho y su alto, mientras que los elementos en línea, obtienen su tamaño de su contenido.

La propiedad `display` no solo se encarga de la representación externa de un elemento, es decir, su relación con los elementos adyacentes, sino que además determina la forma en que sus descendientes se acomodan y visualizan. Sus valores más usuales son:

- `block` lo que indica que elemento incorpora saltos de línea en la región circundante a manera de separación.
- `inline` especifica que el elemento puede ser parte de una línea sin afectar el flujo del texto.
- `none` causa que el elemento y todos sus descendientes sean ignorados y no se representen, como si estos no formaran parte del documento.
- `flex` activa un mecanismo para acomodar los descendientes del elemento, de tal manera que se puedan alinear y acomodar de manera más flexible que el flujo normalmente usado.
- `grid` similar a `flex` este parámetro modifica el acomodamiento de los descendientes del elemento, pensado para crear de forma sencilla acomodados en forma de cuadrícula.

Para cerrar este capítulo, es importante destacar, que la dificultad de los estilos CSS no reside en la complejidad de su lenguaje, sino en la gran cantidad de atributos y posibles valores que ofrece, con más de cien propiedades que pueden ser especificadas. Todas las posibles propiedades del lenguaje CSS pueden consultarse en el sitio: <https://developer.mozilla.org/es/docs/Web/CSS/Reference>. Y en cada nueva versión de CSS, se van incorporando nuevas reglas, propiedades y mecanismos al lenguaje.

Además, existen formas más sofisticadas y complejas de crear selectores para los bloques de estilo, como los pseudo selectores, que permiten hacer referencia a elementos creados dinámicamente por el navegador, o apuntar a un descendiente ubicado en una posición jerárquica determinada, entre otras funcionalidades. Un listado completo de todos los posibles selectores disponibles se puede encontrar en https://developer.mozilla.org/es/docs/Web/CSS/CSS_selectors.

Por fortuna, para iniciar la creación de un libro iCartesiLibri, no se necesitan reglas sofisticada de CSS, ya que el núcleo de presentación de libros iCartesiLibri proporciona un conjunto de estilos básicos, que pueden ser utilizados para comenzar a trabajar en la creación del libro. Estos se explican en el siguiente capítulo.



4. Estructura de un libro iCartesiLibri

A partir de lo visto hasta ahora y utilizando una plantilla básica que contenga los archivos del núcleo de presentación de libros tipo iCartesiLibri, se puede empezar a crear un libro interactivo. En este apartado se examinará la estructura general de un libro iCartesiLibri, centrándose en los archivos del núcleo de presentación y su contenido.

4.1 Plantilla básica

Las plantillas de uso para los libros iCartesiLibri se encuentran alojadas en el repositorio de código https://github.com/jlongi/libro_interactivo, desde donde pueden ser descargadas. Asimismo, se proporciona el siguiente vínculo directo para descargar la plantilla básica: https://github.com/jlongi/libro_interactivo/releases/latest/download/libro_interactivo_plantilla.zip. Su contenido será detallado a continuación.

4.1.1 El directorio `book/`

El funcionamiento principal del libro tipo `iCartesiLibri` está determinado por el código del archivo `book-min.js` contenido en el directorio `book/`. Este archivo conjunta de forma comprimida tres archivos esenciales:

- `book.js`: Este archivo, escrito en JavaScript, proporciona la interacción básica con los elementos del libro. Se encarga de la autonumeración de páginas, secciones y figuras, además de crear y agregar elementos adicionales de forma transparente para el autor, mejorando así la operatividad del libro.
- `style.css`: Este archivo define las reglas CSS fundamentales para la presentación del libro, determinando dimensiones, posiciones e iconos. Complementa el código JavaScript de `book.js` para proporcionar características como el escalado automático al tamaño de la ventana del navegador y la posibilidad de mostrar elementos interactivos o imágenes a pantalla completa.
- `icons.svg`: Este archivo contiene una imagen vectorial que alberga los iconos utilizados en la presentación del libro tipo `iCartesiLibri`, incluyendo los botones de configuración, tabla de contenido y expansión a pantalla completa de los interactivos.

La funcionalidad que ofrece `book-min.js` corresponde a lo que se menciona o nombra como **el núcleo de presentación de libros `iCartesiLibri`**. Y es el responsable de que el documento HTML adquiera la apariencia de un libro digital interactivo, acomodando las páginas y ofreciendo la navegación correspondiente.

Este archivo es modificado cuando se corrigen errores o se añaden nuevas funcionalidades al núcleo de presentación. Para aprovechar estas mejoras en el futuro, se recomienda no modificar el archivo `book/book-min.js`. Si bien es posible personalizar la apariencia del libro, esto se realiza de forma externa mediante otro archivo de estilos proporcionado para su modificación por parte del autor. La personalización en las plantillas se encuentra en el directorio `extra/`, aunque se puede especificar en otra ubicación según las preferencias del autor.

4.1.2 El directorio `extra/`

Los archivos incluidos en la plantilla básica dentro del directorio `extra/` son dos archivos CSS y un directorio; a saber se proporciona el archivo `style.css` desde donde se recomienda crear el conjunto de reglas CSS que el autor necesite para presentar el contenido del libro con un estilo visual personalizado.

Asimismo, se incluye el subdirectorio `fonts/` con la tipografía [Lato](#) del conjunto de fuentes tipográficas que ofrece [Google Fonts](#), que en combinación con el archivo `fonts.css` permite utilizar el tipo de letra incluido, asegurando que los lectores vean el contenido del libro con la tipografía seleccionada, independientemente de si su sistema (o navegador) proporciona dicha tipografía. El mecanismo utilizado para incluir tipografías se explicará a detalle más adelante; sin embargo, hay que mencionar que en el directorio `fonts/` por lo general suelen colocarse otros directorios con el nombre del tipo de letra a utilizar, y dentro de estos se encuentran los archivos propios de las tipografías.

4.1.3 El directorio `img/`

Además de los dos directorios mencionados anteriormente, se encuentra el directorio `img/`, encargado de contener las imágenes requeridas por el libro interactivo. Al igual que con el directorio `extra/`, el nombre de esta carpeta contenedora de imágenes puede ser diferente; los nombres de estos directorios son solo sugerencias proporcionadas por las plantillas y su propósito es mantener la organización del material necesario para la creación de un libro interactivo. La carpeta `img/` contiene otra carpeta llamada `general/` que ofrece algunas imágenes de uso general para el libro, aunque no son necesarias y el autor puede prescindir o modificarlas si lo desea.

Las imágenes incluidas son el logotipo utilizado para la licencia del libro (`by-nc-sa.svg`) siendo [Creative Commons 4.0 internacional: Reconocimiento-No Comercial-Compartir Igual](#). El logo del proyecto iCartesiLibri (`icartesilibri_logo.svg`), en caso de querer publicar el libro en dicho proyecto y dos imágenes una para la portada del libro (`portada.svg`) y otra para utilizarse al comienzo de cada capítulo (`portada_capitulos.svg`).

4.1.4 Licencia de uso

Respecto a la licencia de los libros tipo iCartesiLibri, se debe mencionar que el código del núcleo del libro está sujeto a una licencia [Atribución-CompartirIgual 4.0 Internacional](#). Esta licencia permite compartir y adaptar el código sin restricciones, siempre y cuando se atribuya al autor original y las modificaciones se compartan bajo la misma licencia.

Es importante destacar que esta licencia solo aplica al código del núcleo de presentación, es decir, el archivo [book/book-min.js](#). El contenido del libro, así como los elementos multimedia o interactivos, no están obligados a utilizar esta licencia.

La plantilla y la mayoría de los libros de tipo iCartesiLibri están bajo una licencia Creative Commons 4.0 Internacional: Reconocimiento-No Comercial-Compartir Igual. Por ello, se incluye la imagen [img/general/by-nc-sa.svg](#) correspondiente. Esta licencia permite compartir y adaptar el contenido del libro siempre que se atribuya al autor (o autores), no se utilice con fines comerciales y el nuevo material derivado se comparta bajo la misma licencia. La utilización de esta licencia se debe al objetivo básico de los libros interactivos tipo iCartesiLibri: *ofrecer contenido de calidad de forma abierta y gratuita*. Y dicha licencia permite la creación de obras derivadas mientras protege los derechos de autor de las mismas, además de limitar que cualquier trabajo u obra derivada no pueda utilizarse comercialmente.

Dicho lo anterior, se debe destacar que el autor puede utilizar la licencia que mejor le convenga para publicar su trabajo. Por ejemplo, la licencia [Atribución-Reconocimiento-NoComercial-SinDerivados 4.0 Internacional](#) permite compartir libremente el libro, pero se permite la creación de contenido derivado del original y mucho menos utilizar el trabajo de forma comercial.

Se recomienda revisar información sobre las licencias Creative Commons para determinar cuál se ajusta mejor a las necesidades del autor. En la página relativa en Wikipedia (https://es.wikipedia.org/wiki/Licencias_Creative_Commons) se presenta una tabla que muestra lo que permite cada tipo de licencia.

4.1.5 Archivo `index.html`

El archivo `index.html`, considerado uno de los más importantes para el autor del libro interactivo, es utilizado para enlazar y utilizar todo lo proporcionado anteriormente, así como para almacenar el contenido del libro. En este archivo se escribe el código HTML que compone el documento, además, es éste el archivo al que accederá el navegador de internet para presentar el libro digital interactivo al lector o usuario final.

Para facilitar la creación de libros interactivos, la organización adecuada de sus archivos es fundamental. La gestión de un gran número de archivos, como imágenes, videos o recursos interactivos, puede presentar complejidad y confusiones. Se recomienda la separación de los diferentes tipos de elementos utilizados en directorios distintos, por ejemplo, los videos podrían almacenarse en un directorio denominado `videos/` y los elementos interactivos en un directorio llamado `interactivos/` (o simplemente `inter/`). Si bien el nombre específico de los directorios no es determinante, se sugiere utilizar nombres significativos para simplificar su organización.

Adicionalmente, es necesario tener en cuenta las particularidades de los sistemas operativos en relación con el nombre de los directorios y archivos. En Windows, por ejemplo, los nombres de archivo no distinguen entre el uso de mayúsculas y minúsculas, `portada.png`, `PORTADA.png`, `Portada.PNG` (así como cualquier variación en mayúsculas y minúsculas) se refieren al mismo archivo. A pesar de que un libro pueda desarrollarse exclusivamente en Windows, es importante considerar que el lector podría acceder a él localmente desde un sistema operativo diferente.

Además, muchos servidores de internet donde el libro podría alojarse pueden utilizar algún sistema operativo basado en Linux, por lo tanto, al subir el libro a internet, las referencias a archivos que no respeten la distinción entre mayúsculas y minúsculas pueden impedir que el archivo sea encontrado y, como consecuencia dicho archivo se vuelva inaccesible.

4.2 Una revisión al código de la plantilla básica

En el siguiente interactivo se presenta la plantilla básica del libro iCartesiLibri (**Interactivo 4.1**), es posible interactuar directamente con ella y ver algunos de los elementos principales que ofrece.



Interactivo 4.1. Plantilla básica.

Para evitar complicaciones y la necesidad de crear reglas CSS para obtener un estilo visual particular, el núcleo de presentación de libros tipo iCartesiLibri ofrece un conjunto mínimo de estilos, los cuales pueden utilizarse como punto de partida para comenzar a escribir un libro interactivo, obteniendo un estilo visual sencillo pero funcional. De hecho, algunos de los primeros libros del proyecto iCartesiLibri utilizan directamente este estilo básico, incorporando solo algunos colores y estilos adicionales, algunos ejemplos son:

- [Máquinas Térmicas](#),
- [Teoría de la relatividad](#),
- [HCI - Interfaces Humano-Computador orientado a la Inclusión Social](#),
- [Fundamentos de Programación](#),
- [Diseño de bases de datos](#),
- [Mantenimiento preventivo y predictivo de hardware](#) y
- [Experiencia de curso Introducción a la Ingeniería](#).

Asimismo, la presentación del documento puede personalizarse según las necesidades del documento, por ejemplo, utilizando el mismo núcleo de presentación de iCartesiLibri, se publica la [Revista Digital RED Descartes](#), que presenta un diseño distinto al de un simple libro interactivo.

Cabe destacar que la principal limitación para lograr estilos visuales personalizados reside en el correcto uso de las reglas de estilo CSS, ya que los estilos visuales complejos generalmente requieren un mayor esfuerzo para estilar cada elemento según los requerimientos del autor.

4.2.1 Contenido del archivo `index.html`

En esta sección se presenta el código del archivo `index.html`, el cual está incluido en la plantilla básica, de esta manera, se podrá estudiar la estructura que posee un libro típico de iCartesiLibri.

DOCTYPE

Al abrir el archivo, lo primero que se encontrará es la línea:

```
<!DOCTYPE html>
```

Esta primera línea indica al navegador la versión de HTML que utiliza el documento, con la finalidad de que los distintos navegadores interpreten correctamente el contenido. Antes de la versión 5 de HTML, la declaración sobre la versión de HTML usada en el documento era más extensa, por ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

En algunos sitios web antiguos, se encuentran declaraciones similares a la anterior; estas declaraciones se encargan de indicar al navegador de internet algún modo para presentar el documento. Esta práctica era necesaria principalmente porque los navegadores en el pasado no cumplían con un conjunto mínimo de características estandarizadas de HTML. Por fortuna, a partir de la versión 5 de HTML todos los navegadores se han ido apegando a los estándares y ya no son necesarios los modos especiales para presentar un documento.

Para garantizar que el navegador utilice la interpretación de HTML5, se recomienda siempre incluir la línea `<!DOCTYPE html>`, de lo contrario, la versión queda a consideración del navegador y puede ocurrir que el contenido no se muestre como se tiene planeado.

Etiquetas `<meta>`

A continuación, dentro de la etiqueta `<head>`, se encuentran las siguientes etiquetas de tipo `<meta>`:

```
<meta charset="UTF-8">
```

Esta etiqueta, explicada en el capítulo sobre HTML, se encarga de indicarle al navegador que el documento utiliza la codificación de caracteres `UTF-8`.

```
<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,minimum-scale=1,user-scalable=yes,minimal-ui">
```

La etiqueta anterior determina opciones para dispositivos móviles, respecto al *viewport* o ventana de visualización. El ancho de la página web se establece igual al ancho de la pantalla del dispositivo mediante `width=device-width`, con el objetivo de que el contenido se adapte al tamaño del dispositivo que visualiza el libro.

El nivel de zoom inicial se fija en 1 por medio de `initial-scale=1`, es decir, la página se muestra inicialmente sin alterar su escala.

Las propiedades `maximum-scale=1` y `minimum-scale=1` definen que el nivel máximo y mínimo del zoom que el usuario puede aplicar es 1, lo que indica que se mantendrá la escala original y no se permiten cambios.

Por último, con `minimal-ui` se solicita al navegador que la interfaz de usuario sea mínima, usualmente ocultando la barra de navegación y otros elementos. Sin embargo, la implementación de esta acción depende de la consideración de cada navegador y puede ser ignorada.

```
<meta name="mobile-web-app-capable" content="yes">  
<meta name="apple-mobile-web-app-capable" content="yes">
```

En algunas ocasiones, el usuario puede agregar una página web a la pantalla de inicio de su dispositivo móvil, entonces, las dos etiquetas anteriores indican que la página web es capaz de actuar como una aplicación web móvil (siendo la segunda etiqueta usada para los dispositivos de Apple). Al usar estas etiquetas, el navegador puede presentar la página de manera que parezca una aplicación nativa, omitiendo la barra de direcciones y otros elementos del navegador, ofreciendo una experiencia de pantalla completa.

El título de la pestaña del navegador

Tal como se explicó en el apartado sobre HTML, con `<title>Libro digital interactivo</title>` se especifica el título de la página, en este caso “Libro digital interactivo”. Esta sería la primera línea que el autor debe modificar al crear su propio libro, cambiando el contenido para que refleje el título del libro interactivo.

Precarga de imágenes

Posteriormente, se utiliza una etiqueta de tipo `<link>` para precargar la imagen del logotipo del proyecto iCartesiLibri (ubicada en `img/general/icartesilibri_logo.svg`), la cual será utilizada en la pantalla de carga del libro.

Si un autor desea modificar la imagen mostrada en el cargador, se recomienda que modifique la ruta de esta etiqueta para precargar la imagen adecuada. Más adelante se muestra dónde se incluye y utiliza la imagen del cargador; mientras que aquí con la etiqueta `<meta>` solo se indica que la imagen debe precargarse.

```
<link rel="preload" as="image"
href="img/general/icartesilibri_logo.svg">
```

En el capítulo sobre CSS, se presentó la forma de incluir reglas por bloque mediante el elemento `<link>`, este elemento indica al navegador que se va a vincular un recurso externo. En particular, el recurso externo es una imagen, como se especifica con el atributo `as="image"`, esta información ayuda al navegador a identificar cómo debe tratar el recurso y optimizar su descarga. El atributo `rel="preload"` especifica que la intención del enlace es precargar el recurso, es decir, indica al navegador que debe descargar el recurso lo antes posible, incluso antes de que se necesite en la página. Finalmente, el atributo `href` determina la URL del recurso que se quiere precargar, en este caso, la imagen utilizada por la pantalla de carga, la cual debe estar disponible lo más pronto posible para su uso y presentación.

Variables de configuración del libro

Luego se incluye un poco de código de JavaScript con el que se asignan valores a la variable `window.book_config` con la que se configuran algunos comportamientos del libro:

```
<script>
window.book_config = {
  auto_numerate_sections_and_figures: true,
  open_interactives_fullscreen: true,
}
</script>
```

El parámetro `auto_numerate_sections_and_figures` indica si los capítulos, secciones y figuras se numeran automáticamente. Si se establece en `true`, las etiquetas de encabezado (`<h1>`, `<h2>`, `<h3>`; y `<h4>`) serán utilizadas para la auto-numeración de estos elementos. Sin embargo, si se asigna el valor `false`, la numeración deberá ser agregada manualmente por el autor. Esta opción se ofrece ya que en algunos casos, el autor puede requerir que la numeración se realice bajo criterios personalizados.

El parámetro `open_interactives_fullscreen` controla el comportamiento al presionar el botón de expansión de los interactivos (aquel que se muestra en la esquina superior derecha de un interactivo). Cuando se establece en `true`, los interactivos se muestran a pantalla completa; si se configura como `false`, una nueva ventana o pestaña del navegador será abierta para mostrar el interactivo.

Inclusión de los archivos del libro

A continuación se presenta una línea encargada de incluir el archivo `book/book-min.js` el cual es fundamental para aprovechar las características que ofrecen los libros iCartesiLibri. Si el archivo del núcleo de presentación se colocan en otro directorio, debe ajustarse la ruta para que contenga la ubicación correcta.

```
<script type="text/javascript" src="book/book-min.js" charset="utf-8"></script>
```

El archivo `book/book-min.js`, como se mencionó anteriormente, es un archivo minimizado (comprimido) que contiene imágenes, reglas CSS y código JavaScript integrados. Su uso se recomienda, ya que reduce el tamaño de los archivos necesarios para el correcto funcionamiento del libro interactivo; además, al contar con un único archivo, se simplifica la realización de futuras actualizaciones del núcleo de presentación, pues solo es necesario reemplazar este archivo para realizar la actualización.

Inclusión de los archivos de personalización del libro

Posteriormente, se incluyen las hojas de estilos que se proporcionan al autor para configurar y estilar la presentación del libro: el archivo `extra/fonts.css` para incluir las tipografías y el archivo `extra/style.css` para los estilos propios del libro, que en la plantilla básica se ofrece un conjunto mínimo de estilos iniciales.

```
<link rel="stylesheet" href="extra/fonts.css"
type="text/css">
```

```
<link rel="stylesheet" href="extra/style.css"
type="text/css">
```

Con esto, se completa el contenido de la cabecera (<head>) del documento. Como se puede observar, simplemente se incluyen y configuran algunas de las características del libro interactivo. A continuación, se examinarán los elementos que forman parte de la etiqueta <body>, que será la zona que el autor debe modificar para incluir todo el contenido del libro.

Pantalla de carga

El primer descendiente de <body> corresponde al contenedor <div id="book_loader_container">, que será el encargado de albergar los elementos visibles en la pantalla de carga del libro (Figura 4.1). Estos elementos son: <div id="book_loader"></div> que representa el círculo que gira durante el proceso de carga; y una imagen (que es la misma precargada anteriormente) la cual se muestra a la derecha del círculo giratorio, con un ancho equivalente al veinte por ciento del ancho de la pantalla.

```
<div id="book_loader_container">
  <div id="book_loader"></div>
  
</div>
```



Figura 4.1. Pantalla de carga predeterminada.

Cualquier elemento dentro de `book_loader_container` será mostrado en la pantalla de carga del libro, solo es importante que no se modifique el valor del `id` de este contenedor, ya que el núcleo de presentación se encarga de ocultar este elemento cuando termina la carga del libro.

La pantalla de carga puede ser personalizada reemplazando la imagen que se presenta, o mediante reglas CSS que se apliquen a los identificadores de los elementos, por ejemplo:

```
#book_loader_container {  
    background-color: #130f40;  
}  
#book_loader {  
    border-color: #7ed6df;  
    border-top-color: #22a6b3;  
}
```

Con estas reglas se obtiene una pantalla de carga como la mostrada en la **Figura 4.2**, donde el color del fondo y del círculo de carga son diferentes a los predeterminados.



Figura 4.2. Pantalla de carga estilizada.

El bloque de estilos anterior se recomienda incluirlo en el archivo `extra/style.css`, aunque también puede ser especificado directamente en los elementos HTML mediante su propiedad `style`. Observando el código, el color del círculo de carga está determinado por el color del borde del contenedor `book_loader`. Esto se debe a que el círculo es dibujado haciendo que el `<div>` tenga las esquinas redondeadas al 50%, y lo que se observa es simplemente el borde de este contenedor dando vueltas. Con esta información, también se puede modificar el grosor del círculo cambiando el valor de `border-width` del elemento. De esta manera, se da el primer paso para personalizar el libro interactivo. Sin embargo, se sugiere no utilizar muchos elementos o reglas complicadas para simplificar la presentación de la pantalla de carga.

El contenedor del libro

El siguiente descendiente del elemento `<body>` corresponde a `<div id="book_container">`, encargado de almacenar todos los elementos principales del libro. Estos incluyen los botones de navegación y todas las páginas (alojadas dentro de su propio contenedor).

Los botones de navegación entre páginas

Los botones laterales, identificados por la imagen de flechas que se localizan a los lados de la ventana del navegador, permiten avanzar o retroceder entre las páginas (**Figura 4.3**), y están determinados por los siguientes dos elementos:

```
<div id="btn_back_page"></div>  
<div id="btn_next_page"></div>
```



Figura 4.3. Botones de navegación de páginas.

El botón de la tabla de contenido

Le sigue el botón situado en la esquina superior izquierda (**Figura 4.4**), asociado con la presentación de la tabla de contenido, identificado por el siguiente elemento:

```
<button id="go_to_table_of_content"></button>
```



Figura 4.4. Botón asociado a la tabla de contenido.

Si el elemento `<div id="table_of_content"></div>` está presente en el documento, al pulsar el botón de la tabla de contenido se mostrará una copia de las entradas de la tabla de contenido del libro; esta copia será accesible desde cualquier página del documento. En caso contrario, al pulsar el botón se cambiará a la primera página que contenga un elemento con la clase `toc_link`, que, como se verá más adelante, es el mecanismo utilizado para incluir las entradas de la tabla de contenido.

El botón de configuración y herramientas

En la esquina superior derecha (**Figura 4.5**) se encuentra el botón de configuración y herramientas. Este botón está concebido para permitir al autor incluir herramientas personalizadas, accesibles desde cualquier página del libro, y generalmente, aquí se incluye un switch para cambiar entre el tema claro y oscuro del libro.

```
<button id="btn_config"></button>
```



Figura 4.5. Botón para mostrar las herramientas.

Las herramientas personalizadas se incluyen en el elemento `<div id="config_options">`, en la plantilla, este elemento muestra solamente el switch del tema de color. Posteriormente, se describirá la forma en que se pueden incluir herramientas externas adicionales.


```
<div id="config_options">  
  <div id="dark_light_mode"></div>  
</div>
```

El contenedor de las páginas

El último descendiente del contenedor `book_container` es un elemento que contiene todas las páginas del documento. Se menciona que este elemento solo debe contener páginas; cualquier otro elemento incluido puede arruinar el flujo de presentación del libro, ya que puede alterar la posición de las páginas y desfasarlas.

```
<div id="pages_container">
```

Como se insinuó en el capítulo sobre HTML, cada página que constituye el libro es construida por medio de un `<div>` que utiliza la clase `page`; se recomienda no modificar directamente las dimensiones definidas por esta clase, para evitar que las páginas se acomoden y presenten incorrectamente.

Dimensiones predeterminadas de las páginas

Todas las páginas comparten el mismo tamaño: 640 píxeles de ancho por 825 píxeles de alto, así como los mismos márgenes: 50 píxeles para los márgenes izquierdo, derecho y superior, y 20 píxeles para el margen inferior (**Figura 4.6**). Es importante mencionar que el margen inferior no es estricto ya que los párrafos pueden superarlo y salirse, aún así, el margen inferior se utiliza para ubicar la numeración de la página, así como determinar la posición de los pies de página.

Con esta información se puede deducir que el ancho del texto dentro de una página es de 540 píxeles (640 - (2 × 50)); y la altura será de 755 píxeles (825 - 50 - 20).

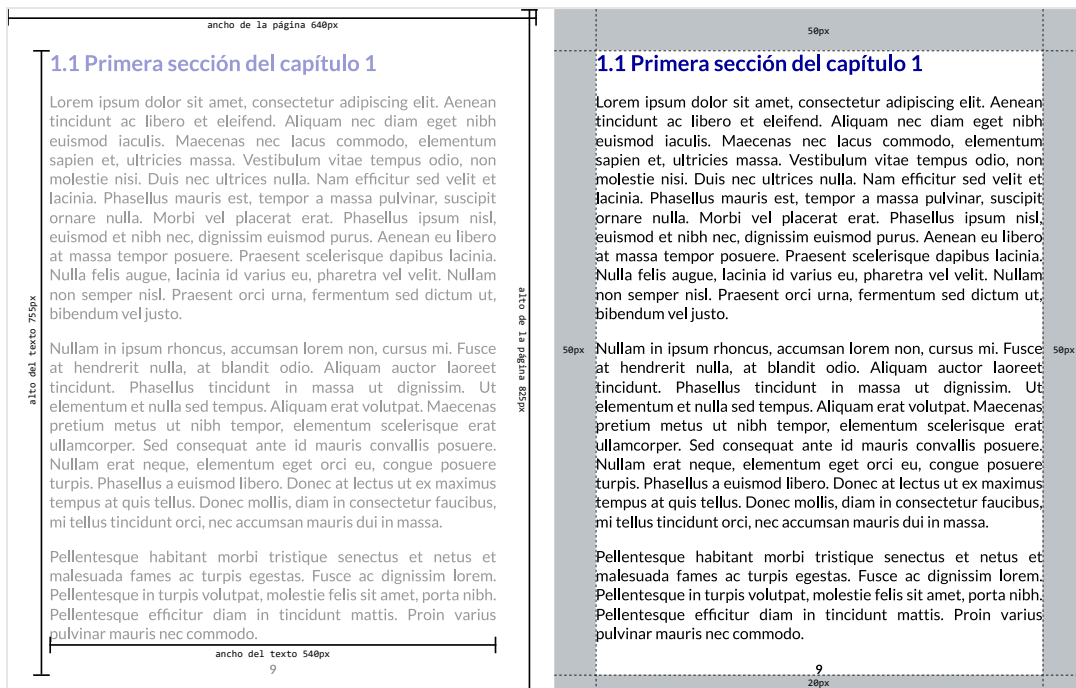


Figura 4.6. A la izquierda se muestran las dimensiones estándar de una página. A la derecha se muestran en gris los márgenes.

También hay que decir que estas dimensiones son los valores utilizados internamente, independiente del tamaño de la pantalla (o ventana del navegador) en que se visualice el libro, ya que parte de las funciones del núcleo de presentación es escalar su contenido para cubrir de la mejor manera el área ofrecida por el navegador. Asimismo, las dimensiones fueron seleccionadas para que con una tipografía de 18 píxeles se tengan líneas de texto con alrededor de 60 caracteres, ayudando a que la lectura sea más agradable y fácil para el lector.

El código HTML utilizado para especificar una página es:

```
<div class="page">
  <!-- Aquí va el contenido de la página -->
</div>
```

Todos los elementos HTML presentados previamente pueden utilizarse dentro de las páginas. Por ejemplo, para bloques de texto, se utilizan párrafos (`<p>`) que abarcan todo el ancho de la pantalla y agregan un salto de línea del mismo tamaño de la tipografía. Al incluir imágenes, se recomienda siempre agregar un valor para el ancho por medio de porcentajes, de manera que cubran una porción (o la totalidad) del ancho de la página; si no se incluye un valor para el ancho, las imágenes utilizan sus propias dimensiones en píxeles, las cuales pueden ser mayores al ancho del texto (540 píxeles), con lo cual puede ocurrir que las imágenes se salgan de los márgenes o incluso de la propia página.

Página cero

Continuando con la revisión del código de la plantilla, se observa que el primer descendiente del contenedor de páginas (`pages_container`) es una página vacía, de la siguiente forma:

```
<div class="page" style="background-
color:transparent;"></div>
```

Esta sería la página cero y su inclusión es necesaria para alinear la posición de presentación del resto de las páginas. Generalmente, no posee contenido, ya que al visualizar el libro en modo de una sola página, esta no es visible.

Portada

La primera página con contenido se utiliza habitualmente para incluir la portada del documento, en la plantilla se lleva a cabo de la siguiente manera:

```
<div class="page invertcolor" style="margin:0;
padding:0; background-size:contain; background-
image:url(img/general/portada.svg);">
  <div style="position:absolute; top:595px;
left:57px; right:0; color:#0089cd; font-size:63px;
text-align:center;">Título del libro</div>
  <div style="position:absolute; left:57px;
right:0; bottom:45px; text-align:center;
color:#0089cd; font-size:40px;">Autor o
editorial</div>
</div>
```

En el código anterior, se observa que el elemento `<div>` cuenta con dos valores en su clase, `page` e `invertcolor`, separados por un espacio. La clase `page` es esencial para todas y cada una de las páginas, mientras que `invertcolor` se puede incluir en cualquier elemento y ayuda a invertir los colores del mismo al visualizar el libro en modo oscuro.

Posteriormente, después de las clases, se encuentra un estilo en línea, que elimina los márgenes y espacios de la página, además de configurar una imagen de fondo (correspondiente a la imagen de portada `url(img/general/portada.svg)`), de tal manera que la imagen se muestre contenida en el área que abarca la página (`background-size: content`).

Como descendientes, cuenta con dos elementos colocados con una posición absoluta (respecto a la página), el primero contiene el título del libro y el segundo agrega el autor (o autores) así como la editorial; la razón de colocarlos de forma absoluta y con estilos en línea es para facilitar el estilo personalizado del libro, sin la necesidad de crear una regla para estos elementos que son únicos. Con lo visto en las secciones anteriores, debe ser claro cómo funcionan las reglas CSS utilizadas en la portada.

Formatos de imágenes

Aprovechando que se habla sobre la imagen de la portada, es importante mencionar los formatos de imagen, los cuales se clasifican principalmente en dos tipos: basadas en mapas de bits y vectoriales.

Las imágenes basadas en mapas de bits se componen de una cuadrícula organizada de píxeles, donde cada píxel representa un color dentro de la imagen. La calidad de estas imágenes está determinada directamente por la cantidad de bits utilizados por cada píxel; a mayor número de bits, mayor calidad, pero también se incrementa el espacio de almacenamiento necesario.

Este tipo de imagen es utilizado ampliamente para fotografías debido a su capacidad de representar una amplia gama de colores y tonos. No obstante, su tamaño puede aumentar considerablemente entre más detalles presente. Para abordar esta problemática, diversos formatos de compresión como PNG, JPG y WebP han sido desarrollados, los cuales reducen el tamaño del archivo sin comprometer significativamente la calidad visual.

Es importante destacar que las imágenes basadas en mapas de bits poseen una resolución fija, lo cual implica que al ampliarse, pueden experimentar artefactos visuales indeseables como pixelado o borronamiento.

Por otra parte, las imágenes vectoriales son creadas mediante el uso de constructores geométricos con los que se generan círculos, rectángulos, líneas, curvas, y otros elementos más. A diferencia de las imágenes de mapas de bits, que se basan en píxeles para su representación.

La construcción particular de las imágenes vectoriales les otorga una característica fundamental: no pierden calidad al modificar el tamaño en que se presentan, ya que no están definidas por una resolución fija en píxeles. Esta característica permite que se puedan escalar sin pérdida de nitidez, lo cual las convierte en elementos ideales para gráficos sencillos como logotipos, iconos e ilustraciones que requieren escalabilidad. Sin embargo, esta misma fortaleza se vuelve una debilidad en el momento de querer representar muchos detalles, como los que se encuentran en una fotografía, lo que puede resultar en archivos de gran tamaño de almacenamiento debido a que se requiere un gran número de elementos geométricos para definir todos los posibles detalles de una fotografía.

Los formatos de imágenes de mapas de bits soportados en internet incluyen PNG, GIF, JPG (a veces llamada JPEG), WebP y AVIF, entre otros menos populares. El único formato vectorial soportado por los navegadores es el SVG, que, como se puede observar, se utiliza en varias de las imágenes de la plantilla.

A continuación, se presenta la comparación entre las versiones de mapa de bits y vectorial del logo del proyecto iCartesiLibri (**Figura 4.7**).

La versión de mapa de bits, tiene una resolución de 635×145 píxeles y utiliza el formato WebP, mientras que la versión vectorial utiliza el formato SVG; ambas ocupan aproximadamente 7KB (siete kilobytes) de espacio de almacenamiento. A pesar de parecer casi idénticas a simple vista, al visualizarlas a pantalla completa mediante un clic sobre cada una de ellas, se aprecia un emborronamiento en la imagen de mapa de bits, por otro lado, la versión vectorial se percibe nítida y detallada, algo especialmente notorio en pantallas de alta resolución.



Figura 4.7. Comparación en el logo del proyecto iCartesiLibri, en la parte superior se observa la versión en mapa de bits y en la parte inferior en versión vectorial.

Parte trasera de la portada

La siguiente página, que representa la parte trasera de la portada, usualmente solo contiene una imagen relacionada con el proyecto o editorial del libro. En el caso de la plantilla básica (y la mayoría de los libros del proyecto iCartesiLibri), se coloca un hipervínculo con el logotipo de iCartesiLibri.

```
<div class="page" style="display:flex; align-
items:center; justify-content:center;"></code>
  <a style="width:50%"
href="https://prometeo.matem.unam.mx/recursos/Vario
sNiveles/iCartesiLibri/"></a>
</div>
```

El código anterior muestra que en esta página se utiliza un estilo que permite centrar el único elemento que contiene (un vínculo con una imagen), en el área que ocupa la página. Esto se logra utilizando `display:flex`, que como se mencionó con anterioridad activa un modelo de caja flexible (o *flexbox*), diseñado para organizar y alinear elementos dentro de un contenedor de manera eficiente y adaptable, a diferencia del modelo de caja estándar.

Con el atributo `align-items` se determina la forma en que se acomodan los elementos verticalmente, mientras que con `justify-content` se indica cómo se acomodan los elementos horizontalmente. En este caso, ambos atributos tienen el valor de `center`, lo que hace que el elemento se coloque centrado en ambas direcciones respecto al área de la página.

Portadilla

La siguiente parte es considerada una especie de “portadilla”, en ella, se presenta una portada más sencilla donde se indica de forma simplificada, el título, el autor y la editorial, así como cualquier otro dato relevante.

```
<div class="page" init-page-num="true" num="">  
  <p style="text-align:center; font-size:50px;  
margin-top:100px;">Título del libro</p>  
  <p style="margin-top:200px; font-  
size:25px;">Nombre del o los autores</p>  
</div>
```

En el código de esta página hay que resaltar los parámetros del elemento HTML, específicamente, el parámetro `init-page-num="true"` que se utiliza para indicar que la numeración de las páginas comienza a partir de esta página. Cualquier página que aparezca antes de la página con este parámetro no se considera numerada. El comportamiento predeterminado hace que las páginas numeradas utilicen números romanos, lo cual puede modificarse mediante otro parámetro, como se indicará más adelante.

El parámetro `num` indica que la página no mostrará un número en su parte inferior, siempre que se desee una página sin numeración, este último parámetro debe incluirse. El valor que se le asigna no es importante, solo se requiere que aparezca el atributo para no mostrar la numeración.

Portada interior y derechos de autor

Tras la portadilla, generalmente se incorpora una página que funciona como portada interior y contiene información de derechos de autor. En esta página, se incluye el nombre de la obra, el autor (o autores), y además se incluye la atribución al código del núcleo del libro, se otorgan los créditos a las tipografías utilizadas y, si es necesario, se acreditan las imágenes de portada. También se suelen incluir los datos editoriales, la atribución al proyecto iCartesiLibri en caso de que el libro forme parte de este proyecto, el ISBN y la licencia bajo la cual se publica el libro. Si bien el formato de esta página es libre, el incluido en la plantilla es el formato típico utilizado en los libros del proyecto iCartesiLibri.

Atribución al núcleo de presentación iCartesiLibri

Recordando lo visto al hablar sobre la licencia del núcleo de presentación del libro, es necesario incluir la atribución al programador, lo cual se recomienda realizar mediante un elemento `<div>` o `<p>` con el siguiente contenido:

```
Código JavaScript para el libro: <a
href="https://github.com/jlongi/libro_interactivo">
Joel Espinosa Longi</a>, <a
href="https://www.matem.unam.mx/">IMATE</a>, UNAM.
```

Tabla de contenido

En la plantilla (y en muchos libros iCartesiLibri), la página siguiente se designa para la tabla de contenido, índice o sumario. Esta página permite listar los capítulos y secciones que conforman el libro.

Es importante mencionar que el núcleo de presentación no exige iniciar con la tabla de contenido; se puede comenzar, por ejemplo, con los agradecimientos, el prólogo o el prefacio, según la preferencia del autor.

Entradas de la tabla de contenido

La construcción de la tabla de contenido se realiza mediante la inclusión de referencias a las diversas partes de forma manual, utilizando código como el siguiente:

```
<div class="toc_link" href="#IDENTIFICADOR"></div>
```

Donde `IDENTIFICADOR` corresponde al valor del parámetro `id` de un elemento dentro del documento que se quiera referenciar. El signo de número (`#`) debe incluirse al inicio del valor del parámetro `href`, pero no en el parámetro `id` del elemento referenciado. Por ejemplo, si se desea agregar a la tabla de contenido una referencia al prefacio, definido por el siguiente elemento:

```
<h2 id="prefacio">Prefacio</h2>
```

Se utiliza el código:

```
<div class="toc_link" href="#prefacio"></div>
```

Es importante tener en cuenta que el texto incluido en la entrada de la tabla de contenido será el mismo que contenga el elemento referenciado. En el ejemplo anterior, se incluirá el texto “Prefacio”, que es el contenido del elemento `<h2>`.

Para facilitar la autoría, se recomienda que los identificadores contengan información relevante para permitir una fácil identificación. Asimismo, se enfatiza que los identificadores deben ser únicos y no deben incluir espacios, recomendando utilizar caracteres alfanuméricos incluyendo el guión bajo en caso de querer separar palabras dentro del identificador.

La creación de elementos de tipo `<div>` con la clase `toc_link` es un paso importante, ya que permite que el núcleo de presentación busque en el documento todos los elementos con dicha clase, y les asigne como descendientes tres elementos que se corresponden con: el texto contenido en el elemento referenciado por la entrada en la tabla de contenido; la secuencia de puntos intermedios; y el número de página donde se encuentra la referencia.

El prefacio

Tras la tabla de contenidos, por lo general, se agrega el prefacio, donde el contenido propio del libro puede comenzar a escribirse. Un detalle importante de esta página es que, como se mencionó anteriormente, la numeración se inicia con una representación de números romanos, para comenzar a utilizar números arábigos, se debe establecer el parámetro `num-type-arabic="true"`, desde el momento en que aparece este parámetro, todas las siguientes páginas utilizarán números arábigos. Por ejemplo:

```
<div class="page" num-type-arabic="true">  
  ...  
</div>
```

Página de inicio del capítulo

Muchos de los libros del proyecto iCartesiLibri utilizan una página completa con una imagen de fondo y el título del capítulo correspondiente, lo cual se incorpora mediante código de la siguiente forma:

```
<div class="page invertcolor" num=""  
style="margin:0; padding:0; background-  
size:contain; background-  
image:url(img/general/portada_capitulos.svg);">  
  <div class="chap_text">Capítulo I</div>  
  <h1 class="chap_name" id="capitulo_1">Título del  
capítulo 1</h1>  
</div>
```

El estilo de la página debe ser claro a estas alturas, ya que es similar al usado en la portada.

La página contiene dos elementos: el primero es un `<div>` con la clase `chap_text` que muestra el texto “Capítulo” y de forma manual la numeración de dicho capítulo (“I”).

Luego se encuentra un elemento `<h1>` con la clase `chap_name`, el cual lleva el nombre o título del capítulo. Este elemento, si se observa bien, tiene un `id="capitulo_1"`, que será utilizado para referenciarlo en la tabla de contenidos. Es importante indicar que el elemento `<h1>` y la clase `chap_name` son utilizados para iniciar el contador de capítulos y ambos son necesarios para la auto numeración.

Las reglas que definen estas dos clases se encuentran en el archivo [extra/style.css](#) y para el caso de la plantilla, tienen el siguiente conjunto de reglas:

```
.chap_text {  
    position: absolute;  
    right: 50px;  
    top: 200px;  
    color: #0089cd;  
    font-size: 90px;  
    text-shadow: 1px 2px rgba(0,0,0,0.15);  
}
```

```
.chap_name {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    position: absolute;  
    left: 58px;  
    width: 582px;  
    top: 395px;  
    height: 170px;  
    color: #0089cd;  
    font-size: 45px;  
    text-shadow: 1px 2px rgba(0,0,0,0.15);  
    padding: 10px;  
    text-align: center;  
}
```

Con todo lo explicado en las secciones previas, debe ser posible entender la función de cada una de las reglas anteriores. De forma general, se colocan los textos con posiciones absolutas respecto a la página, configurando el color y tamaño de la letra, así como el tamaño del contenedor de la caja que lo contiene, y adicionalmente se configura una sobra proyectada por el texto.

En caso de no querer una página con imagen de fondo al inicio de un capítulo, se puede quitar el estilo de la página y los textos pueden acomodarse directamente con el flujo del documento, es decir, eliminar la regla `position: absolute`. Algunos ejemplos de plantillas con estilos diversos para el inicio de los capítulos serán proporcionados más adelante, como la utilizada en esta obra, la cual emplea imágenes en la parte superior del comienzo del capítulo, pero estas no abarcan toda la página.

Secciones y subsecciones

Las siguientes páginas de la plantilla simplemente muestran cómo incluir una sección:

```
<h2 id="cap1_sec1">Primera sección del capítulo 1</h2>
```

Así como una subsección:

```
<h3 id="cap1_sec2_subsec1">Primera subsección del la segunda sección del capítulo 1</h3>
```

Además, se incluye un segundo capítulo, siendo igual al primero, agregado únicamente para facilitar el copiado y pegado al momento de construir un libro.

Bibliografía

En la penúltima página se incluye un ejemplo de una entrada bibliográfica, la cual se agrega de la siguiente forma:

```
<bibitem id="id_del_libro_1">  
  El o los autores. <b>Título del libro  
  referenciado</b>. Datos de publicación.  
</bibitem>
```

Y se ve de la siguiente manera:

[1] El o los autores. **Título del libro referenciado.** Datos de publicación.

El número que aparece al inicio se agrega automáticamente y corresponde al orden en que se incluye la entrada bibliográfica.

Si se desea emplear un texto en lugar del número, como por ejemplo, para seguir alguna norma específica, se puede utilizar el parámetro `label` dentro de la entrada bibliográfica:

```
<bibitem id="id_del_libro_1" label="bib_libro_1">  
  El o los autores. <b>Título del libro  
  referenciado</b>. Datos de publicación.  
</bibitem>
```

Lo que se muestra de la siguiente forma:

[bib_libro_1] El o los autores. **Título del libro referenciado.**
Datos de publicación.

Como se puede notar, para incluir un elemento a la bibliografía se utiliza la etiqueta `<bibitem>` con un identificador con el cual se puede hacer referencia a dicha entrada bibliográfica. Asimismo, el contenido de la etiqueta corresponde a la información de la entrada, la cual puede formatearse al gusto y requerimientos del autor.

Es importante mencionar que la etiqueta `<bibitem>` no forma parte del lenguaje HTML; está incluida como parte del núcleo de presentación del libro, en otras palabras, no es una etiqueta estándar. Si se utiliza en un documento cualquiera de HTML, no realiza ninguna acción específica ni tiene una representación predeterminada, de hecho, dependiendo del navegador, puede interpretarse como una etiqueta de bloque o en línea, por lo tanto, se desaconseja su uso fuera del contexto de los libros iCartesiLibri.

Para hacer una referencia a una entrada de la bibliografía, se utiliza la etiqueta `<bibref>` con el parámetro `ref_id` tomando el valor del identificador de la entrada que se quiere referenciar. Por ejemplo, para crear una referencia al elemento con `id="id_del_libro_1"`, se utilizaría el siguiente código:

```
<bibref ref_id="id_del_libro_1"></bibref>
```

La etiqueta `bibref` tampoco forma parte del HTML estándar.

El final del libro

La última página incluida en la plantilla básica y en general en los libros iCartesiLibri se presenta como una página con el logotipo del proyecto al que pertenece el libro, y su propósito es marcar el final del documento. En caso de requerirlo el autor, la última página puede ser la contraportada del libro, incluyendo una breve sinopsis, biografía del autor o información relevante respecto al libro.

4.3 Elementos ofrecidos para los libros iCartesiLibri

El núcleo de presentación iCartesiLibri, similar a lo descrito con `bibitem` y `bibref`, incorpora algunas etiquetas HTML adicionales, y agrega algunos comportamientos a elementos ya existentes, con el objetivo de simplificar y mejorar la construcción de libros interactivos. Los elementos adicionales incluidos se presentan en esta sección.

4.3.1 Pie de página

El elemento `<footer>` es utilizado en documentos HTML y generalmente se ubica en la parte inferior de una página web. Su función principal es agrupar información adicional como derechos de autor, información de contacto, el mapa del sitio, entre otros. El uso de este elemento contribuye a estructurar visualmente la página web, mejorando su lectura y navegación tanto para usuarios como para motores de búsqueda.

En una página de un libro, lo equivalente al elemento `<footer>` son los pies de página, estos se encuentran también en la parte inferior de la página y brindan información relevante que complementa el texto principal, como, notas aclaratorias, referencias, créditos de imágenes y detalles complementarios sobre el contenido de la página; contribuyendo a la organización y estética del libro.

En un libro, una nota a pie de página está compuesta por dos elementos (**Figura 4.8**).

- El primero es un número o símbolo que relaciona la nota con el contenido de un párrafo y se encuentra ubicado dentro del flujo del texto del párrafo.
- El segundo elemento es el bloque de texto que contiene la información de la nota y que se ubica en la parte inferior de la página, mostrando el mismo número o símbolo de la referencia.

Este es un párrafo que incluye una nota a pie de página¹, es importante ver que el flujo del texto no se ve interrumpido.

¹ Este es el contenido de la nota a pie de página.

Figura 4.8. Ejemplo de una nota a pie de página. El número 1 a manera de superíndice es la forma en la que se vincula la referencia con la nota.

Para mejorar la visibilidad del texto de la nota a pie de página, una línea horizontal se suele incluir con el propósito de dividir visualmente el contenido.

En los libros iCartesiLibri, se ofrece un mecanismo para la inclusión de notas a pie de página, agregando un elemento con una clase especial con el contenido de la nota directamente en el texto del párrafo, el cual es reemplazado por el núcleo de presentación con el número de referencia y su contenido se coloca en la parte inferior de la página. Tanto los números de referencia como la reubicación del texto son realizados automáticamente y de forma transparente para el autor, debido a esto, no se puede utilizar un elemento de tipo `<footer>`, ya que su inclusión como descendiente de un párrafo (`<p>`) divide el párrafo con saltos de línea, modificando así el flujo del texto. En su lugar, se utiliza un elemento `` con la clase `footnote`. Por ejemplo, para obtener el resultado mostrado en la imagen anterior (**Figura 4.8**), se utiliza el siguiente código:

```
<p>Este es un párrafo que incluye una nota a pie de
página<span class="footnote">Este es el contenido
de la nota a pie de página.</span>, es importante
ver que el flujo del texto no se ve interrumpido.
</p>
```

El contenido del elemento `` con la clase `footnote`, tal como se observa en el código HTML anterior, corresponde al texto que aparece en la parte inferior de la página. Se observa también que el número `1` se incluye automáticamente para vincular la nota al pie de página y la posición de su inclusión dentro del párrafo; este número se autoincrementa con cada nota que se añada a lo largo del libro. La clase `footnote` es especial porque no solo determina el estilo CSS de la nota al pie de página, sino que además el núcleo de presentación la trata de forma singular al reemplazar su localización en el documento HTML y gestionar el contador asociado.

Es importante tener en cuenta que en un documento HTML que no utilice el núcleo de presentación iCartesiLibri, este elemento no produce el mismo resultado, ya que se comporta simplemente como un ``, y el contenido se incrusta directamente.

Si el contenido de la nota al pie de página requiere saltos de línea, la etiqueta `
` debe ser utilizada dentro del contenido del `` para su inclusión.

4.3.2 Notas

Para facilitar la comprensión y destacar información importante, los libros iCartesiLibri, al igual que las notas al pie de página, ofrecen una clase específica: `note`. Esta clase puede utilizarse para indicar que un bloque de texto constituye una nota o contiene información relevante para el lector.

En casos donde la nota se limita a un único párrafo, se sugiere emplear directamente un elemento `<p class="note">`, por ejemplo:

```
<p class="note">Esta es una nota que abarca solo un párrafo.</p>
```

Lo que da el siguiente resultado:

Resultado

Esta es una nota que abarca solo un párrafo.

Si por el contrario la nota abarca varios párrafos (o etiquetas HTML), se recomienda utilizar un contenedor `<div class="note">` e incorporar dentro de él todos aquellos elementos que forman parte de la nota. Por ejemplo:

```
<div class="note">
  <p>Esta es una nota que abarca más que un solo
  párrafo, y su información se encuentra contenida en
  un contenedor.</p>
  <p>Este es el segundo párrafo de la nota.</p>
  <p>Y un tercer párrafo.</p>
</div>
```

Con el código anterior se obtiene lo siguiente:

Resultado

Esta es una nota que abarca más que un solo párrafo, y su información se encuentra contenida en un contenedor.

Este es el segundo párrafo de la nota.

Y un tercer párrafo.

El estilo predeterminado de la clase `note` está definido para incluir una franja azul en el lado izquierdo, mediante el uso del atributo `border-left`. La tipografía se reduce a un 90% de su tamaño original (`font-size:90%`) y se le aplica un relleno lateral (`padding`) de un caracter y medio (`1.5ch`). Este estilo puede ser modificado según el gusto del autor sin ningún tipo de restricción.

4.3.3 Bloques autonumerados

Para crear bloques de elementos autoenumerados, como definiciones, teoremas, lemas, ejercicios o ejemplos, se utiliza una clase especial, similar a la nota anterior. Esta clase permite la inclusión de elementos que se quieran enumerar a lo largo de la presentación del libro, para ello, un contenedor `<div>` con la clase `num_block` se define para este propósito, y se le incluye el atributo `prefix` que proporciona el nombre utilizado para presentar el texto; este nombre coincide también con el contador que registra la cantidad de elementos creados con ese mismo prefijo.

```
<div class="num_block" prefix="Definición">
  Esta es la primera definición de mi libro.
</div>
```

El código anterior, construye el siguiente bloque autonumerado.

Resultado

Definición 4.1. Esta es la primera definición de mi libro.

En el ejemplo anterior se observa el texto “**Definición 4.1.**”, el cual emplea el valor del atributo `prefix="Definición"` junto con el número del capítulo al cual pertenece, a continuación, y separando por un punto, se añade el número correspondiente a la numeración propia del bloque. Si posteriormente se introduce otro bloque con el mismo prefijo, el último número presentado mostrará el siguiente consecutivo en la secuencia. Tras la cabecera o nombre del bloque, se presenta finalmente el contenido que lo define.

La clase `num_block` se caracteriza por incluir un borde izquierdo con el estilo `border-left: 4px solid #16a085` e incorporar un relleno a la izquierda mediante `padding-left: 1.5ch`. De forma automática, la primera parte del bloque es creada utilizando un elemento de tipo `` con la clase `num_block_prefix`, la cual se ofrece al autor para que pueda modificar su apariencia. Por defecto, esta clase aplica las reglas `font-weight: bold;` y `font-style: italic`, lo que indica que el texto se muestra en negrita e itálica, en caso de no desear este formato, los valores `bold` e `italic` pueden ser reemplazados por el valor `normal`.

Si se busca que el contenido de la definición se muestre en una línea aparte, basta con incluir un párrafo (`<p>`) con el contenido.

Por ejemplo:

```
<div class="num_block" prefix="Definición">
  <p>Esta es la segunda definición de mi libro y su
  contenido se encuentra separado del texto creado
  automáticamente con el valor del prefijo.</p>
</div>
```

Se crea el elemento:

Resultado

Definición 4.2.

Esta es la segunda definición de mi libro y su contenido se encuentra separado del texto creado automáticamente con el valor del prefijo.

4.3.4 Figuras autonumeradas

Al hablar de elementos autonumerados, las figuras (`<figure>`) de los libros iCartesiLibri tienen el comportamiento de incluir automáticamente su numeración cuando presentan un elemento del tipo `<figcaption>`. En el caso que una figura no incluya este tipo de elemento, esta no se enumera. Por ejemplo:

```
<figure>
  
  <figcaption>La leyenda o título correspondiente a
  la figura.</figcaption>
</figure>
```

Muestra la siguiente figura:

Resultado



Figura 4.9. La leyenda o título correspondiente a la figura.

En el ejemplo anterior, de nuevo, la primera parte “Figura 4.9.” es creada automáticamente por medio de un elemento tipo `` con la clase `figcaption_prefix` y similar a lo explicado en los bloques autonumerados, los elementos `<figcaption>`, pueden incluir el parámetro `prefix`, para indicar el tipo de figura que se está incluyendo.

El valor por omisión de `prefix` es “Figura”, que es el texto que se presenta al incluir el elemento; y los números incluidos corresponden otra vez al número del capítulo seguido del número de figura (con el prefijo indicado).

Si, por ejemplo, se busca incluir una tabla dentro de una figura, que también presente una leyenda o título, se puede asignar el valor “Tabla” a la propiedad `prefix`, como se muestra a continuación:

```
<figure>
  <table>
    <tr>
      <td>Título 1</td>
      <td>Título 2</td>
    </tr>
    <tr>
      <td>Dato 1</td>
      <td>Dato 2</td>
    </tr>
  </table>
  <figcaption prefix="Tabla">La leyenda o título
  correspondiente a la tabla.</figcaption>
</figure>
```

Con lo que se representa la siguiente tabla:

Resultado

Título 1	Título 2
Dato 1	Dato 2

Tabla 4.1. La leyenda o título correspondiente a la tabla.

Se observa ahora el texto automático “**Tabla 4.1.**” que utiliza el valor asignado a `prefix`, así como el número del capítulo y el número de tabla. Esta funcionalidad permite indicar diferentes tipos de figuras, según su contenido, confiando en que el núcleo de presentación iCartesiLibri se encargue de llevar los contadores de cada uno de los prefijos utilizados.

Hay que remarcar que en documentos que no utilicen el núcleo de presentación iCartesiLibri, la numeración de las figuras no está presente, ya que ésta no es una característica propia de HTML.

4.3.5 Referencias a bloques autonumerados y figuras

Un comportamiento adicional proporcionado por los libros iCartesiLibri, es la posibilidad de incluir referencias a los bloques autonumerados y las figuras. Para esto, se define el elemento `<ref>`, que utiliza el valor de su parámetro `ref_id` para incluir la referencia a un elemento con un identificador dado. Por ejemplo, para incluir una referencia a la tabla anterior, considerando que la figura tiene el identificador `<figure id="ejemplo_tabla">`, se escribiría algo como:

```
<p>Dentro del contenido de este párrafo se incluye una referencia a la <ref ref_id="ejemplo_tabla"> </ref>.</p>
```

Con lo que se crea el siguiente párrafo:

Resultado

Dentro del contenido de este párrafo se incluye una referencia a la **Tabla 4.1.**

Debe mencionarse que es indispensable que el elemento referenciado cuente con un identificador, pues de lo contrario, ninguna referencia será incluida. Asimismo, para las figuras, se requiere que éstas cuenten con un elemento `<figcaption>` a fin de realizar la referencia.

4.3.6 Referencias a páginas

Hablando de referencias, ya se ha visto con anterioridad cómo incluir una referencia a alguna entrada de la bibliografía (`<bibref>`) y ahora a elementos construidos como bloques autonumerados y figuras.

Adicionalmente, los libros iCartesiLibri incorporan un mecanismo que permite la inclusión de una referencia a una página, utilizando como punto de anclaje cualquier identificador de algún elemento que forme parte de la página a referenciar.

Por ejemplo para incluir una referencia al “Prefacio”, se utiliza el siguiente código:

```
<pageref ref_id="prefacio"></pageref>
```

Lo anterior simplemente muestra el número de la página donde se encuentra el “Prefacio”, e incluye un hipervínculo local que dirige a dicha página.

Resultado

[10](#)

El elemento `<pageref>` ofrece algunos parámetros para configurar la presentación de su contenido. El parámetro `prefix` puede ser utilizado para incluir un texto antes del número de página, y de manera similar, el atributo `postfix` se utiliza para un texto después del número de página. El parámetro `digits` controla el número de dígitos usados para representar el número de página. Por ejemplo:

```
<pageref ref_id="prefacio" prefix="En la página"
postfix="se encuentra el Prefacio del libro."
digits="3"></pageref>
```

Lo que muestra:

Resultado

[En la página 010 se encuentra el Prefacio del libro.](#)

Así mismo, vinculadas a estos elementos se ofrecen reglas CSS para cambiar el estilo con el que se muestra cada parte del texto. En particular, las clases `pageref_prefix`, `pageref_postfix` y `pageref_number` se proporcionan para estilar el texto prefijo, postfijo y número de página, respectivamente.

Para finalizar, si se desea que el número de página no aparezca en la referencia y, en su lugar, se muestre un texto particular, solamente se debe colocar el texto deseado como contenido del elemento `<pageref>`, por ejemplo:

```
<pageref ref_id="prefacio">como se muestra en el
Prefacio</pageref>
```

Proporciona el siguiente resultado:

Resultado

[como se muestra en el Prefacio](#)

4.3.7 Elementos interactivos

Hasta ahora se han presentado principalmente funcionalidades relacionadas con texto, lo cual permite la creación de un libro digital. Sin embargo, el punto fuerte del proyecto iCartesiLibri reside en la adición de componentes interactivos a estos libros digitales.

En el contexto de una página web, un componente o recurso interactivo es aquel elemento que ofrece la posibilidad de que el usuario realice una acción y, a cambio, la página genere una reacción, es por ello, que muchos consideran que un hipervínculo es un elemento interactivo, ya que al pulsar sobre él se muestra el recurso al que apunta. En el contexto de libros interactivos, esta definición debe extenderse, ya que los recursos interactivos no solo deben reaccionar a las acciones del usuario, sino también tener una intencionalidad didáctica y complementar o contextualizar el contenido circundante.

Desde un punto de vista técnico y en el contexto de las tecnologías web, un recurso interactivo se construye utilizando HTML para la estructura del documento, CSS para la presentación y JavaScript para el manejo de las interacciones (aunque a veces se pueden utilizar lenguajes que se ejecutan en el servidor). La creación de recursos interactivos utilizando directamente estas tecnologías puede resultar complicada debido a la necesidad de dominar tres lenguajes diferentes.

No obstante, existen diversas herramientas que permiten la creación de recursos interactivos con tecnologías web sin necesidad de ser expertos en dichas tecnologías. Solo por mencionar un par, DescartesJS y GeoGebra son dos herramientas populares que ofrecen características diversas para la construcción de recursos interactivos manteniendo aislados del usuario los conceptos y nociones de HTML, CSS y JavaScript.

Lo importante a considerar es que los recursos interactivos son páginas web por sí mismos, y para incluirlos en un libro iCartesiLibri se necesita un mecanismo que permita incluir páginas web externas. Esta funcionalidad es proporcionada por la etiqueta `<iframe>`, que viene de *Inline frame* y ofrece un marco o región donde presentar un documento HTML externo; similar al elemento `` que permite incrustar en una página una imagen externa por medio de su atributo `src`, los `<iframe>` hacen lo mismo pero con páginas web. De hecho, es posible incrustar en un `<iframe>` cualquier elemento que un navegador pueda interpretar, aunque para los propósitos de incluir interactivos se utilizarán solo páginas web (.html).

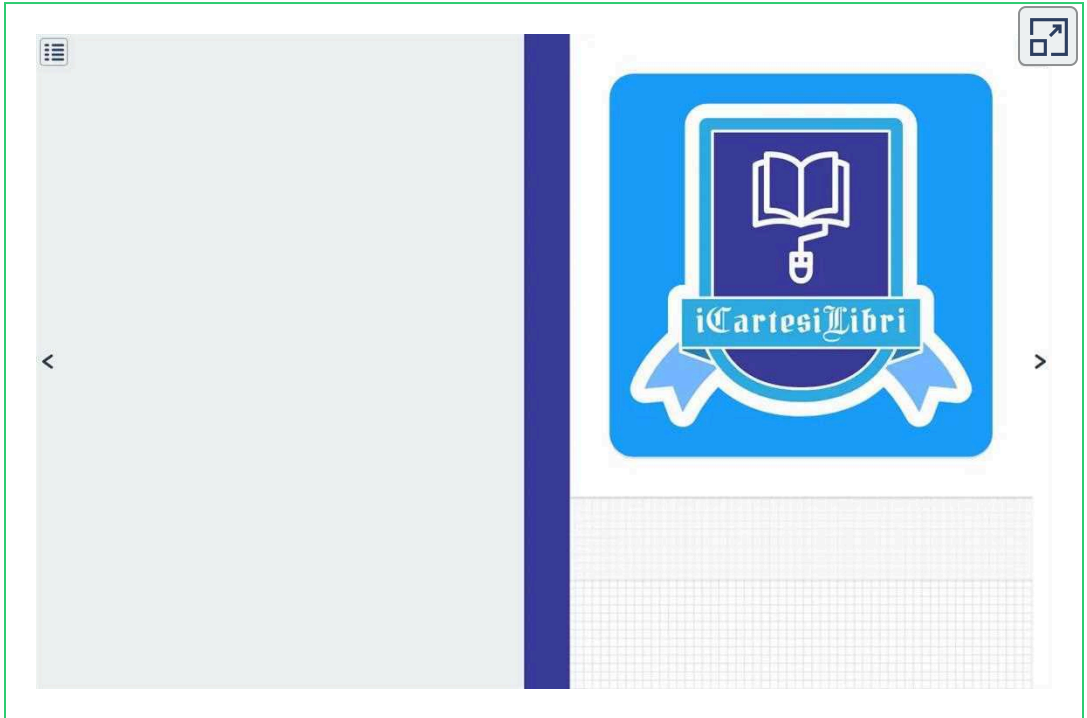
La creación de un libro iCartesiLibri incluyendo interactivos directamente con `<iframe>` es posible; pero el núcleo de presentación iCartesiLibri proporciona la clase `interactive` que al utilizarse dentro de una etiqueta `<div>` facilita la inclusión de un recurso interactivo con la funcionalidad adicional de incluir un botón para mostrar el recurso interactivo a pantalla completa o en una ventana nueva, según se haya configurado el parámetro `open_interactives_fullscreen`.

El elemento `<div class="interactive">` se encarga internamente de construir todos los elementos necesarios para la presentación del recurso interactivo, se crea el botón para maximizar el interactivo, el `<iframe>` correspondiente y las reglas CSS necesarias. Independientemente de si se escoge utilizar directamente un `<iframe>` o un `<div>` con la clase `interactive`, las dimensiones del elemento interactivo deben ser proporcionadas por el autor. Esto se debe a que, a diferencia de una imagen, los `<iframe>` no conocen directamente su tamaño, ya que dependen de un documento externo que puede o no tener unas dimensiones asignadas.

Para incluir un recurso interactivo utilizando un elemento `<div class="interactive">`, se necesita proporcionar el ancho (`width`) y alto (`height`) como atributos de la etiqueta. Estos valores son necesarios para garantizar que la relación de aspecto y las dimensiones del `<iframe>` construido automáticamente sean las correctas para presentar el recurso interactivo. Además, la ubicación del interactivo a mostrar se especifica en el parámetro `src`. El elemento `<div class="interactive">` puede recibir también el parámetro `scale`, que especifica el porcentaje del ancho del texto que el interactivo debe cubrir.

Por ejemplo, para incluir la plantilla básica con dimensiones de 1280px de ancho, dos páginas de 640px por cada página y 825px de alto:

```
<div      class="interactive"      width="1280px"      height="825px"
src="inter/plantilla_basica/index.html"></div>
```

Por ejemplo, el mismo interactivo con 50% del ancho del texto:

```
<div class="interactive" width="1280px" height="825px" scale="50%"  
src="inter/plantilla_basica/index.html"></div>
```

Con lo que se obtiene:



Para incluir la plantilla básica mediante un `<iframe>`, se presentan algunas diferencias con el método anterior, ya que las dimensiones se especifican de forma diferente para forzar que las proporciones se mantengan. Por ejemplo:

```
<iframe src="inter/plantilla_basica/index.html" style="width:100%; height:calc((var(--text_width) / 1280) * 825);"></iframe>
```

Lo anterior crea lo siguiente:

Al observar el estilo del `<iframe>`, se observa que el ancho tiene un valor del `100%`, este porcentaje se especifica respecto al ancho del texto; para el alto, se utiliza una expresión que puede parecer compleja: `calc((var(--text_width) / 1280) * 825)`, pero es más sencilla de lo que aparenta.

La función `calc` de CSS permite realizar cálculos aritméticos para determinar algunas dimensiones, se utilizan los operadores comunes, es decir, `+`, `-`, `/` y `*`, para la suma, resta, división y multiplicación, respectivamente. La función `var` de CSS permite acceder al valor de alguna variable definida dentro de las reglas de estilo, en particular la variable `--text_width` la crea el núcleo de presentación a partir del valor del ancho de la página y los márgenes asignados. Entonces el alto del `<iframe>` se determina como: $(\text{ancho del texto} / \text{ancho de la escena interactiva}) \times \text{alto de la escena interactiva}$.

Ahora bien, si se busca que el interactivo cubra el 50% del ancho del texto, basta con indicar este porcentaje en el parámetro `width` y multiplicar por 0.5 el ancho del texto en el cálculo del alto. Por ejemplo:

```
<iframe src="inter/plantilla_basica/index.html" style="margin:0 auto; width:50%; height:calc((0.5 * var(--text_width) / 1280) * 825)"></iframe>
```

El estilo `margin:0 auto;` permite que el `<iframe>` se centre horizontalmente.

Los mecanismos descritos previamente permiten la presentación tanto de elementos locales como de direcciones externas en Internet dentro del libro interactivo. Por ejemplo, en el siguiente interactivo se muestra el sitio web de la herramienta de autor DescartesJS:

```
<div class="interactive"
src="https://descartes.matem.unam.mx/"
width="1280px" height="400px"></div>
```



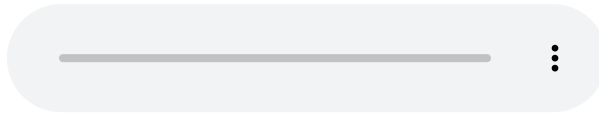
Es importante considerar que los interactivos, independientemente de la tecnología utilizada en su creación, deben diseñarse para escalar y ajustarse al área del elemento `<iframe>` que los contiene. De no ser así, pueden aparecer cortados en el libro (o al mostrarlos a pantalla completa), lo cual impediría una correcta presentación.

4.3.8 Audio

Como se vio en el apartado de HTML, elementos multimedia como audios y videos, se pueden incluir directamente dentro del libro a través del lenguaje HTML, mediante el uso de las etiquetas correspondientes.

Por ejemplo para incluir un audio dentro del libro, puede utilizar código como el siguiente:

```
<audio controls="" preload="none">
  <source src="inter/strong-hit-36455.mp3"
  type="audio/mpeg">
</audio>
```



Audio 4.1.
Sound Effect from [Pixabay](#).

Es importante incluir el parámetro `controls=""` para asegurar que la interfaz de usuario del reproductor de audio sea visible. Sin su inclusión, no será posible reproducir el video directamente, a menos que se incluya código JavaScript para su manipulación indirecta.

El parámetro `preload` con el valor `"none"` indica al navegador que no precargue el audio, esta acción es esencial para optimizar tanto el rendimiento como la experiencia del usuario, pues al evitar la descarga previa se ahorra ancho de banda del servidor y del usuario. Además, se reduce el tiempo de espera inicial en la carga del libro, especialmente cuando existen muchos archivos multimedia, y de esta manera, se evita dar la impresión al usuario de que hay algún problema con el libro. El valor predeterminado para `preload` es `auto`, lo que le sugiere al navegador que intente precargar el contenido del audio, sin embargo, esta acción está sujeta a la implementación específica de cada navegador.

Adicionalmente, la interfaz de reproducción presenta opciones limitadas para su personalización, ya que solo permite cambiar su ancho, y esta modificación únicamente puede realizarse mediante una regla CSS, por ejemplo: `style="width:60%"` causa que el ancho del reproductor sea del 60% de su ancestro. Intentar hacer lo mismo con la altura (`height`) provoca que la interfaz del reproductor de audio se desplace, como si de un margen se tratara, pero no aumenta la altura de los controles de reproducción. Asimismo, dependiendo del navegador la representación de la interfaz puede variar, detalle que es necesario tener en cuenta, ya que la interfaz puede abarcar una mayor o menor altura, o usar colores diferentes:

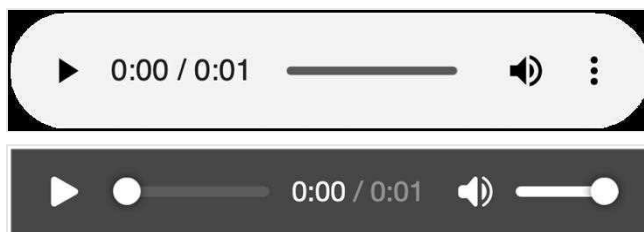


Figura 4.10. En la parte superior se muestra el reproductor de audio en navegadores como Google Chrome o Microsoft Edge; mientras que en la parte inferior se muestra el reproductor en Mozilla Firefox.

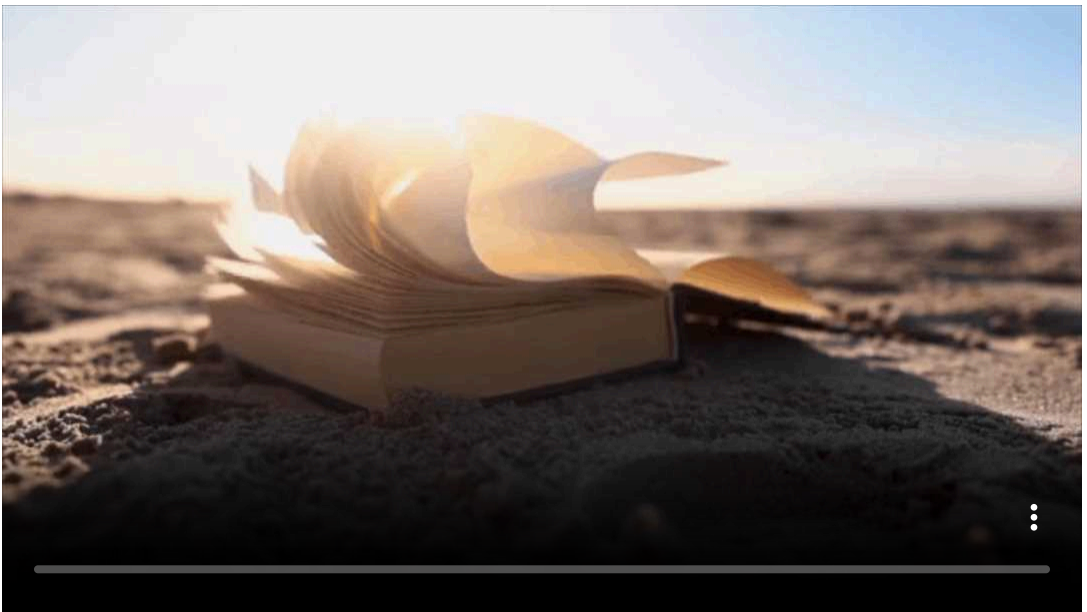
4.3.9 Video

Para el video, similar a lo que ocurre con los audio, se debe incluir el atributo `controls=""` para visualizar la interfaz de usuario, igualmente, se recomienda utilizar `preload="none"` para evitar que el navegador precargue el contenido del video, y que solo comience la descarga al pulsar el botón de reproducción. Esta recomendación es más relevante en el caso de los videos, dado que suelen ocupar mayor espacio de almacenamiento que los audios.

Un ejemplo para incluir un video sería:

```
<video controls="" preload="none" width="100%"  
poster="inter/185092-874643408_tiny_poster.png">  
  <source src="inter/185092-874643408_tiny.mp4"  
  type="video/mp4">  
</video>
```

Con lo que se obtiene el siguiente video:



Video 4.1. Video by [Stefanie Jockschat](#) from [Pixabay](#)

Para controlar el ancho de los videos, se recomienda emplear el atributo `width`, aunque también es posible hacerlo mediante reglas CSS, como por ejemplo `style="width:100%"`, utilizado en el código anterior.

En videos al igual que los audios, tampoco se puede estilar el reproductor, y aunque aquí si se permite especificar el alto con el atributo `height`, esto no se recomienda debido a que puede generar barras negras arriba y abajo, o en los costados del video, dependiendo de si el alto es mayor o menor al del propio video.

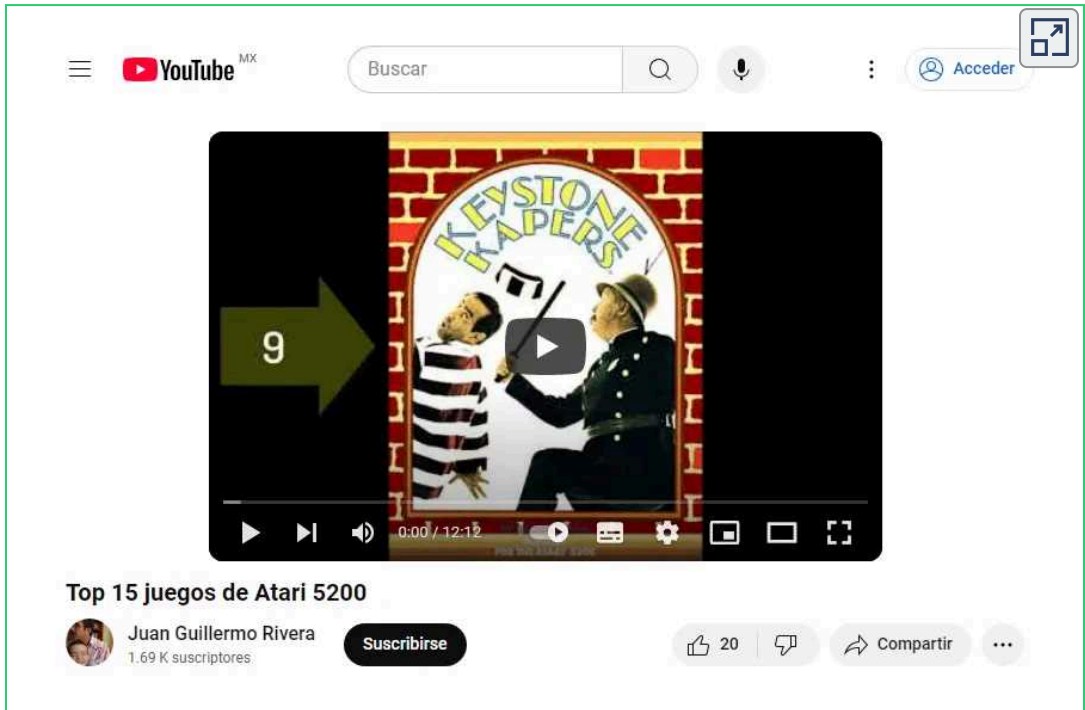
Otro atributo importante para los videos es `poster`, el cual indica la ubicación de una imagen que se mostrará como previsualización del video, evitando así una pantalla vacía mientras se carga.

La inclusión de elementos multimedia locales se recomienda mediante el uso de las etiquetas `<audio>` y `<video>`. Estos elementos se utilizan cuando se dispone del archivo multimedia correspondiente y estos se distribuye junto con los archivos del libro.

Para incorporar elementos multimedia externos, se puede emplear el elemento `<iframe>` o `<div class="interactive">`, no obstante, es necesario seguir los lineamientos establecidos por los sitios externos para su inclusión. Por ejemplo, si se desea incluir un video de YouTube como <https://www.youtube.com/watch?v=qJ1FWE0yoRg>, y se utiliza este enlace directamente como valor de `src` del `<div class="interactive">` (o del `<iframe>`), lo que se obtiene es un error, ya que YouTube bloquea la inclusión de sus videos en páginas externas mediante el uso de este vínculo general.

```
<div class="interactive"
src="https://www.youtube.com/watch?v=qJ1FWE0yoRg"
width="851" height="542"></div>
```


Dependiendo del navegador se puede observar un error al intentar cargar la página, o un mensaje de que el archivo no se encontró.



Para incluir un video de YouTube es necesario seguir sus lineamientos (https://developers.google.com/youtube/player_parameters?hl=es) y utilizar los vínculos con la forma adecuada. De forma simplificada, los vínculos para incluir un video de YouTube consisten en sustituir `watch?v=` por `embed/`, del ejemplo anterior se tiene entonces que `https://www.youtube.com/watch?v=qJ1FWE0yoRg` pasa a ser `https://www.youtube.com/embed/qJ1FWE0yoRg`, entonces el código correcto es:

```
<div class="interactive"
src="https://www.youtube.com/embed/qJ1FWE0yoRg"
width="851" height="542"></div>
```

Que en esta ocasión agrega correctamente el video al libro.



En el enlace sobre la manera de incluir videos en YouTube (https://developers.google.com/youtube/player_parameters?hl=es), se explican los pasos para incluir un video de YouTube en una página web, así como diversos parámetros adicionales que permiten modificar aspectos específicos del reproductor. Estos parámetros pueden utilizarse para controlar diferentes funciones, por ejemplo: el parámetro `start` permite establecer el tiempo inicial de reproducción, mientras que `controls=0` oculta los controles del reproductor. El parámetro `mute=1` inicia la reproducción silenciada. Además, `disablekb=1` deshabilita el uso del teclado para controlar la reproducción, y `autoplay=1` permite un inicio automático de la misma.

Es importante tener en cuenta que, aunque se utilicen los vínculos correctos, no todos los videos de YouTube pueden ser incluidos dentro de un `<iframe>`, esta posibilidad depende únicamente de los permisos otorgados por el creador del video. Por lo tanto, se recomienda que los videos externos incluidos en los libros interactivos provengan de un canal propio o de un canal confiable que garantice su disponibilidad a largo plazo, de lo contrario, podrían dejar de funcionar sin previo aviso.

4.3.10 Imágenes

En relación con las imágenes, el núcleo de presentación ofrece funciones que se pueden asociar a una imagen para su ejecución al ser pulsada.

Una función disponible es `openImage(this)`, que permite mostrar la imagen a pantalla completa o en una nueva pestaña del navegador según la configuración especificada en el parámetro `open_interactives_fullscreen`, previamente mencionado. Por ejemplo:

```

```

Al utilizar `onclick="openImage(this)"` se indica al navegador que, al recibir un clic en la imagen (`onclick`), ejecute la función `openImage(this)`, esta función se encarga de abrir (o mostrar a pantalla completa) la imagen y esta se encuentra definida por el núcleo de presentación iCartesiLibri.



Si en lugar de lo anterior, se desea abrir un vínculo externo al hacer clic en una imagen, se proporciona la función `openInteractive`. Esta función también se asigna al evento `onclick` de la imagen y recibe tres argumentos: `openInteractive(ubicación, ancho, alto)`, donde `ubicación` es la dirección del recurso que se quiere abrir, `ancho` y `alto` son las dimensiones en píxeles de la ventana donde se presenta el recurso. Por ejemplo:

```

```

Genera la siguiente imagen:



Con lo anterior, al pulsar sobre la imagen, se abre una nueva ventana con un tamaño de 800×600 píxeles, mostrando el sitio: <https://prometeo.matem.unam.mx/recursos/VariosNiveles/iCartesiLibri/>. Es importante notar que los argumentos de la función están separados por comas (,) y que la dirección del recurso que se desea mostrar está acotada por medio de comillas simples ('). En este caso, se recomienda utilizar la clase `image_link` en la imagen, la cual agrega un botón en la esquina superior derecha, permitiendo al usuario saber que la imagen abre un link externo, además modifica el cursor del ratón para indicar que se puede dar clic sobre ella, sin embargo, es indispensable mencionar que esta clase no es necesaria para que la función `openInteractive` se ejecute correctamente, solo es una característica estética.

En el caso de que no sea necesario abrir una nueva ventana con dimensiones específicas, es posible utilizar directamente una etiqueta `<a>` que contenga la imagen:

```
<a  
href="https://prometeo.matem.unam.mx/recursos/VariosNiveles/iCartesiLibri/"></a>
```

Resultado



The image shows the rendered result of the HTML code above. It features a logo for 'iCartesiLibri'. The letter 'i' is green and has a decorative, calligraphic style. The letters 'CartesiLibri' are blue and also have a decorative, calligraphic style. The entire logo is displayed within a white rectangular frame.

En los navegadores modernos, se ha incorporado una funcionalidad útil en las imágenes: la capacidad de descargarlas solo cuando son visibles para el usuario. Generalmente, al introducir una etiqueta `` cuando el navegador construye el DOM y encuentra estas etiquetas, comienza a descargar su contenido, sin embargo, en un libro interactivo, el lector no siempre necesita que todas las imágenes estén descargadas para empezar a leer el contenido del libro. Por lo tanto, se recomienda cargar los recursos solo cuando sean necesarios para el lector. Para ello, se puede incorporar el parámetro `loading="lazy"` a las imágenes, este parámetro indica al navegador que no debe descargar la imagen inmediatamente, sino hasta que sea visible. Un ejemplo:

```

```

Se recomienda incluir siempre el parámetro `loading="lazy"` para agilizar la carga de un libro y permitir al lector acceder al contenido lo más pronto posible, sin tener que esperar a que se carguen imágenes que posiblemente no vea.

4.3.11 Información emergente o *Tooltips*

Un *tooltip* es una pequeña ventana emergente que aparece cuando el cursor del ratón pasa por encima de un elemento en una interfaz de usuario, proporcionando información adicional sobre ese elemento, son ampliamente utilizados para dar contexto, definir términos, mostrar valores o guiar al usuario en su uso, los *tooltips* son breves, aparecen cerca del elemento desencadenante y desaparecen automáticamente después de un tiempo o al mover el cursor fuera de ellos.

El núcleo de presentación permite crear estos elementos por medio del uso de un elemento de tipo `` con la clase `tooltip_container` que engloba el contenido que puede detonar el comportamiento de mostrar el tooltip, y dentro de esta etiqueta se debe incluir otro `` con la clase `tooltip` que contiene el contenido propio de la información emergente.

```
El texto de un párrafo con una <span
class="tooltip_container">información
emergente<span class="tooltip">Este es el texto del
tooltip, usualmente debe ser un contenido breve.
</span></span>, continuación del texto del párrafo.
```

Lo que crea lo siguiente:

Resultado

El texto de un párrafo con una información emergente, continuación del texto del párrafo.

El núcleo de presentación determina la posición en la que se mostrarán los *tooltips* para que aparezcan lo más cerca posible del contenido y de forma legible, si los *tooltips* contienen demasiado contenido, pueden aparecer mal colocados o incluso salir de la página. En dispositivos móviles, donde no hay cursor, los *tooltips* pueden mostrarse al pulsar sobre el elemento asociado, permitiendo así su funcionalidad en estos dispositivos.



5. Personalización del libro

En este capítulo se describen las diferentes formas en que diversas partes del libro pueden particularizarse. Esta personalización puede incluir aspectos como la tabla de contenido, la numeración de páginas y el uso efectivo de los temas de colores claro y oscuro, entre otras opciones.

Se destaca que gran parte de la personalización del libro interactivo es realizada a través de reglas CSS aplicadas a los elementos que lo componen.

Anteriormente se indicó que el núcleo de presentación iCartesiLibri incorpora el archivo `style.css` dentro del contenido de `book/book-min.js`, el cual incorpora un conjunto de estilos básicos para el funcionamiento del libro interactivo, en particular se incluye un número de variables CSS, las cuales pueden ser modificadas externamente, preferentemente en el archivo `extra/style.css`.

5.1 Cambio en las dimensiones de las páginas

El estilo de las dimensiones del libro se determina mediante las siguientes reglas:

```
:root {  
  --pages_container_width: 1280px;  
  --pages_container_height: 825px;  
  --page_width: 640px;  
}
```

La primera línea `:root` es un selector especial que apunta al nodo raíz del documento. Esta selección permite que cualquier descendiente del documento acceda a la información de las dimensiones, aprovechando el comportamiento en cascada de los estilos.

Las propiedades con notación iniciada por dos guiones (`--`) corresponden a variables (o propiedades) CSS del usuario, y permiten almacenar valores para luego ser reutilizados a lo largo del documento. Estas propiedades personalizadas se definen únicamente para los nodos en que se establecen y su descendencia, por ello se suelen agregar al nodo raíz del documento (`:root`) o en `<body>`.

Entonces, el código anterior define tres variables: el ancho y el alto del contenedor de las páginas, así como el ancho de la página. El núcleo de presentación opera bajo el supuesto de que el alto de la página es igual al alto del contenedor; por lo tanto, no se define una variable para el alto de la página.

Los libros tipo iCartesiLibri están diseñados para que su presentación se adapte al tamaño de la ventana de visualización. Si el ancho de la ventana es mayor que su altura, dos páginas se muestran simultáneamente; por esta razón, el ancho de cada página es la mitad del ancho del contenedor de páginas. Por el contrario, si el ancho de la ventana es menor que su altura, únicamente una página se visualiza. Este ajuste en las dimensiones de la ventana permite una mejor visualización en diferentes tamaños de pantalla y, aunque no es perfecta, aumenta la versatilidad del uso de los libros interactivos.

Dicho lo anterior, es posible modificar el comportamiento de mostrar una o dos páginas, simplemente proporcionando un valor al ancho de la página (`--page_width`) mayor al alto del contenedor (`--pages_container_height`), en este caso el núcleo de presentación no intenta mostrar dos páginas simultáneamente y siempre muestra una única página.



Interactivo 5.1. Esta plantilla puede descargarse pulsando [aquí](#).

El ejemplo anterior se logra mediante las siguientes reglas, que deben colocarse al principio del archivo `extra/style.css`:

```
:root {  
  --pages_container_width: 1800px;  
  --pages_container_height: 900px;  
  --page_width: 1800px;  
}
```

De esta manera, al utilizar el mismo valor para el ancho del contenedor y el ancho de la página, se consigue cubrir por completo el contenedor y mostrar siempre una sola página. Este comportamiento es útil en casos donde se desee realizar, por ejemplo, una presentación de diapositivas tipo PowerPoint, pero aprovechando la creación con HTML y agregando elementos multimedia e interactivos.

Del ejemplo anterior (de presentación tipo PowerPoint) se observa que los botones adicionales del libro no se muestran, esta característica se consigue comentando o eliminando las líneas que incorporan los elementos con identificadores: `btn_back_page` y `btn_next_page` para la navegación lateral entre páginas; `go_to_table_of_content` y `table_of_content` para el botón de la tabla de contenido; así como `btn_config`, `config_options` y `dark_light_mode`, para el botón de herramientas. La navegación entre páginas se puede realizar mediante un deslizamiento lateral con el dedo en un dispositivo táctil o con el ratón. Igualmente, el cambio de página se puede efectuar utilizando las flechas izquierda y derecha del teclado para retroceder o avanzar respectivamente.

Algo a tener en cuenta es que, si bien el autor cuenta con la libertad de personalizar las dimensiones de su libro interactivo, se recomienda que estas no sean extremas para evitar romper la presentación de las páginas. En particular, se sugiere que el ancho de cada página sea igual al ancho del contenedor o la mitad de éste, para utilizar de forma efectiva el área total de presentación.

5.2 Cambio en los márgenes de las páginas

Existen algunas variables adicionales que se pueden modificar correspondientes a los márgenes de las páginas, determinados por: `--page-left-margin`, `--page-right-margin`, `--page-top-margin` y `--page-bottom-margin`, que representan el margen izquierdo, derecho, superior e inferior respectivamente. Asimismo, se puede configurar el margen interior que se utiliza entre el borde y un interactivo mediante la variable `--interactive-margin`. Al igual que antes, se recomienda asignar valores a estas variables dentro del elemento `:root` al inicio del archivo `extra/style.css`.

Es importante tener en cuenta que, las dimensiones y márgenes especificados de forma predeterminada en el núcleo de presentación iCartesiLibri, fueron seleccionados para cubrir de la mejor manera diversos tipos de dispositivos de forma general. Y respecto al texto, se busca que con una tipografía de 18 píxeles, cada línea de texto tenga cerca de 60 caracteres, con la finalidad de facilitar la lectura. Si se modifican los valores por omisión, el autor debe verificar que el libro se visualice correctamente en diferentes dispositivos, y en caso contrario, es necesario que ajuste los tamaños para garantizar una óptima legibilidad.

5.3 Modificar la tipografía utilizada

En la sección dedicada a la descripción del contenido de la plantilla básica, se menciona que, en el directorio `extra/`, junto al archivo `style.css`, se incluye el archivo `fonts.css` y un directorio llamado `fonts/`. Las siguientes reglas CSS se encuentran dentro del contenido del archivo `extra/fonts.css`:

```
/* Lato */
@font-face {
  font-family: 'Lato';
  src: url("fonts/Lato/Lato-Regular.woff2") format("woff2");
  font-weight: normal;
  font-style: normal;
}

@font-face {
  font-family: 'Lato';
  src: url("fonts/Lato/Lato-Italic.woff2") format("woff2");
  font-weight: normal;
  font-style: italic;
}

@font-face {
  font-family: 'Lato';
  src: url("fonts/Lato/Lato-Bold.woff2") format("woff2");
  font-weight: bold;
  font-style: normal;
}

@font-face {
  font-family: 'Lato';
  src: url("fonts/Lato/Lato-BoldItalic.woff2") format("woff2");
  font-weight: bold;
  font-style: italic;
}
```

Como se observa, se definen cuatro bloques de reglas que inician con `@font-face`. Estas reglas se utilizan para determinar los archivos fuente utilizados para una tipografía dada, en este caso, se proporciona el nombre 'Lato' como el valor de la propiedad `font-family`. El uso del mismo nombre de tipografía en los cuatro bloques se debe a que se están definiendo los cuatro posibles estilos tipográficos disponibles para este tipo de letra:

- **Letra regular:** El archivo `src: url("fonts/Lato/Lato-Regular.woff2")` se utiliza con los valores `normal` en las propiedades `font-weight` y `font-style`.
- **Cursiva:** El archivo `src: url("fonts/Lato/Lato-Italic.woff2")` se utiliza con el valor `italic` en la propiedad `font-style` y `normal` en `font-weight`.
- **Negrita:** El archivo `src: url("fonts/Lato/Lato-Bold.woff2")` se utiliza con el valor `bold` en la propiedad `font-weight` y `normal` en `font-style`.
- **Cursiva negrita:** El archivo `src: url("fonts/Lato/Lato-BoldItalic.woff2")` se utiliza con el valor `italic` en la propiedad `font-style` y `bold` en `font-weight`.

En caso de que la tipografía no disponga de todos los estilos, solo se requiere incluir los bloques de reglas correspondientes a los archivos y el estilo deseado. Hay que notar que los archivos fuente de la tipografía se encuentran en el directorio `extra/fonts/`, pero debido a que las reglas están almacenadas en el archivo `extra/style.css` la ruta de los archivos no incluye el directorio `extra/`, y empiezan directamente con `fonts/`.

Algo que puede parecer extraño es el formato de las tipografías, `format("woff2")`, existen varios formatos soportados por los navegadores, sin embargo, algunos de ellos son específicos del sistema operativo y otros del navegador donde se visualiza el contenido. Los archivos `.woff2` (**Web Open Font Format**, versión 2) son un formato de archivo utilizado para almacenar fuentes tipográficas en una forma optimizada para su uso en la web, ofrecen una buena balance entre tamaño del archivo y calidad de visualización, además de ser compatibles con la mayoría de los navegadores modernos.

Al intentar utilizar una tipografía externa, deben tenerse en cuenta las licencias de uso de la fuente, ya que la conversión al formato `.woff2` no implica necesariamente el permiso para su uso, ya que existen licencias que restringen su empleo en páginas web; en este sentido es necesario tener cuidado con el uso de cualquier tipografía encontrada en internet, ya que no todas permiten su libre utilización. Se aconseja usar tipografías que cuenten con la *SIL Open Font License*, la cual permite el uso de las tipografías de forma libre y gratuita.

Se recomienda el uso de fuentes tipográficas provenientes de [Google Fonts](#), un sitio que alberga cientos de tipografías diseñadas por artistas y diseñadores internacionales, las cuales cuentan con licencias permisivas para su uso en proyectos web, impresos o digitales. La fuente tipográfica “Lato”, empleada en la plantilla básica y en múltiples libros del proyecto iCartesiLibri, es un ejemplo de una fuente obtenida de Google Fonts bajo una licencia abierta. El catálogo de este sitio ofrece una extensa colección de estilos tipográficos, desde clásicos hasta modernos, serif, sans-serif, script, etc., permitiendo que cualquier proyecto encuentre la opción ideal para su diseño.

El único inconveniente de Google Fonts es que, desde su sitio web, solo se puede descargar el formato `.ttf`, por lo que se requiere utilizar un sitio adicional para convertir las tipografías al formato `.woff2`. A continuación, se recomiendan algunos sitios que permiten realizar esta conversión:

- <https://cloudconvert.com/ttf-to-woff2>,
- <https://anyconv.com/ttf-to-woff2-converter/> y
- <https://everythingfonts.com/ttf-to-woff2>.

También existe el sitio <https://gwfh.mranftl.com/>, el cual ofrece un buscador de tipografías de Google Fonts y permite descargar directamente la versión `.woff2` de la tipografía.

La elección entre tipografías serif y sans serif es crucial para transmitir la apariencia adecuada que se busca con un libro. Las tipografías serif o con *serifas*, caracterizadas por las pequeñas extensiones en las letras, evocan tradición, elegancia y sofisticación. Su diseño, con remates en forma de adornos, las hace apropiadas para textos largos, documentos formales y libros. Por otro lado, las tipografías sans serif (sin serifas) ofrecen un aspecto moderno, limpio y directo. Su simplicidad las convierte en una opción ideal para títulos llamativos, diseño web, carteles, logotipos e interfaces de usuario. La mejor elección depende del contexto específico. Se recomienda considerar: el propósito del texto o diseño; el formato de presentación (impreso o digital); y el público objetivo al que va dirigido el documento.

A continuación se proporcionan algunas versiones de la plantilla básica que incluyen diversas tipografías populares.

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.1. La plantilla con la tipografía [HankenGrotesk](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.2. La plantilla con la tipografía [Jost](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.3. La plantilla con la tipografía [Amaranth](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.4. La plantilla con la tipografía [ComicNeue](#) puede descargarse [aquí](#).

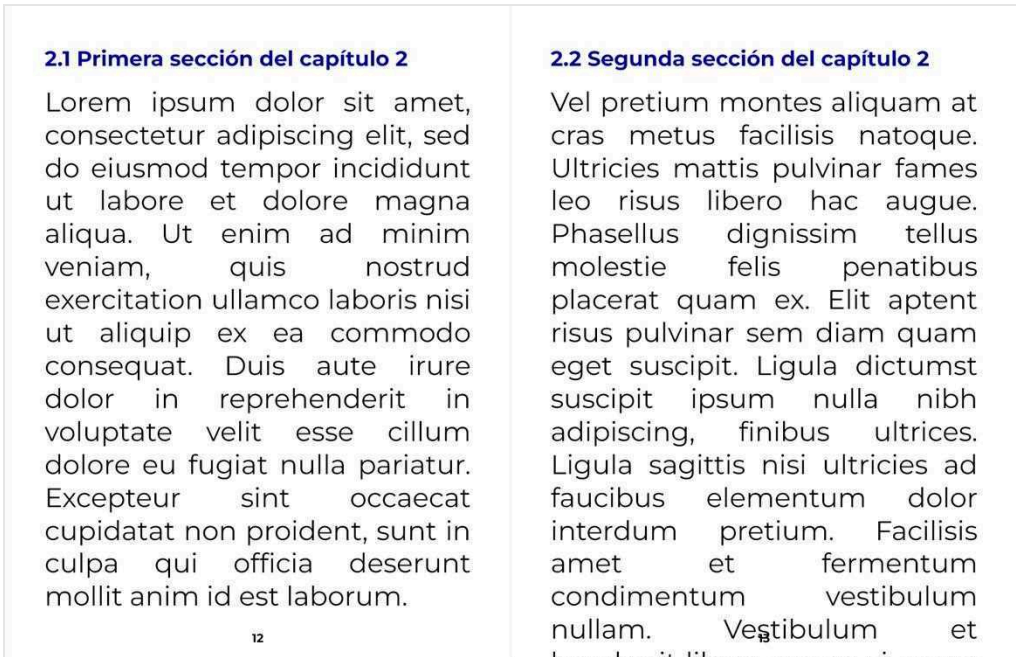


Figura 5.5. La plantilla con la tipografía [Montserrat](#) puede descargarse [aquí](#).

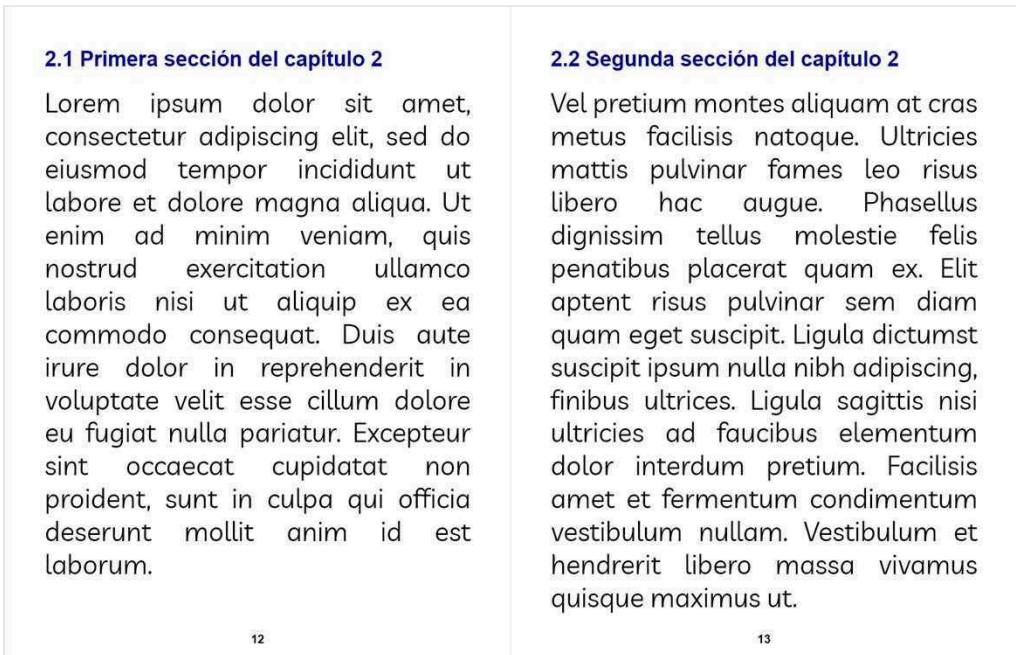


Figura 5.6. La plantilla con la tipografía [Livvic](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.7. La plantilla con la tipografía [Cabin](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.8. La plantilla con la tipografía [Lexend](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus, ut.

Figura 5.9. La plantilla con la tipografía [Nunito](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.10. La plantilla con la tipografía [Afacad](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit

Figura 5.11. La plantilla con la tipografía [Poppins](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit

Figura 5.12. La plantilla con la tipografía [WorkSans](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.13. La plantilla con la tipografía [SourceSans3](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa

Figura 5.14. La plantilla con la tipografía [SourceSerif4](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.15. La plantilla con la tipografía [CrimsonPro](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus,ut.

Figura 5.16. La plantilla con la tipografía [Aleo](#) puede descargarse [aquí](#).

<p>2.1 Primera sección del capítulo 2</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p> <p>12</p>	<p>2.2 Segunda sección del capítulo 2</p> <p>Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.</p> <p>13</p>
---	--

Figura 5.17. La plantilla con la tipografía [CormorantGaramond](#) puede descargarse [aquí](#).

<p>2.1 Primera sección del capítulo 2</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p> <p>12</p>	<p>2.2 Segunda sección del capítulo 2</p> <p>Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.</p> <p>13</p>
---	--

Figura 5.18. La plantilla con la tipografía [Lora](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus ut.

13

Figura 5.19. La plantilla con la tipografía [STIXTwoText](#) puede descargarse [aquí](#).

2.1 Primera sección del capítulo 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

12

2.2 Segunda sección del capítulo 2

Vel pretium montes aliquam at cras metus facilisis natoque. Ultricies mattis pulvinar fames leo risus libero hac augue. Phasellus dignissim tellus molestie felis penatibus placerat quam ex. Elit aptent risus pulvinar sem diam quam eget suscipit. Ligula dictumst suscipit ipsum nulla nibh adipiscing, finibus ultrices. Ligula sagittis nisi ultricies ad faucibus elementum dolor interdum pretium. Facilisis amet et fermentum condimentum vestibulum nullam. Vestibulum et hendrerit libero massa vivamus quisque maximus,ut.

Figura 5.20. La plantilla con la tipografía [Vollkorn](#) puede descargarse [aquí](#).

Las imágenes de las plantillas anteriores muestran que, aunque en todos los ejemplos se utilice el mismo valor para el tamaño de la tipografía (`font-size`), dependiendo de la fuente empleada, el tamaño del texto puede variar considerablemente, tanto en ancho como en alto. Por lo tanto, se recomienda probar diferentes tamaños para asegurar una correcta visualización del texto, evitando que sea demasiado grande o demasiado pequeño.

Como comentario final, en caso de requerir la incorporación de múltiples tipografías en un libro, se debe copiar el bloque de reglas existente y modificar el nombre de la fuente. Asimismo, se especificará la ubicación de los archivos fuente correspondientes a cada tipografía. Se recomienda no incluir demasiadas tipografías en un mismo libro para mantener la cohesión visual y evitar distraer innecesariamente al lector, sin embargo, es posible combinar diversos estilos en una misma obra, por ejemplo, pueden usarse una tipografía sans serif para los encabezados y serif para el contenido del libro, o viceversa. En última instancia, la elección del estilo tipográfico que mejor se adapte al aspecto visual del libro corresponde al autor.

5.4 Estilo de la tabla de contenido

El estilo visual de los elementos de la tabla de contenido puede ser ajustado dependiendo del público al que vaya dirigido el libro interactivo. Se mencionó anteriormente, en el capítulo dedicado a la estructura de un libro iCartesiLibri, sobre la tabla de contenido y la forma de incluir entradas en ella, la cual se realiza con:

```
<div class="toc_link" href="#IDENTIFICADOR"></div>
```

Se observa que la entrada de la tabla de contenido es una etiqueta de tipo `<div>` que no contiene ningún elemento; el núcleo de presentación agrega posteriormente tres elementos `` como descendientes, es decir, se construye algo de la siguiente forma:

```
<div class="toc_link" href="#IDENTIFICADOR">  
  <span class="toc_name">Título</span>  
  <span class="toc_dots"></span>  
  <span class="toc_number">#</span>  
</div>
```

El contenido del elemento referido por el `IDENTIFICADOR` se encuentra en el primer `` con la clase `toc_name`. El segundo elemento con la clase `toc_dots` se expande para cubrir el ancho existente entre el título de la entrada y el número. Por último, el tercer `` con la clase `toc_number` contiene el número de página que corresponde al elemento referido. El estilo de la entrada de la tabla de contenido es controlado por estas tres clases, permitiendo así la aplicación de estilos personalizados en ella.

Adicional a las clases anteriores, dependiendo del nivel en la jerarquía de las cabeceras, se asigna la variable `level`. Los capítulos reciben un valor de `level=0`, los elementos `<h2>` adquieren un valor de `level=1`, los `<h3>` tienen asignado un valor de `level=2` y así sucesivamente. Este parámetro permite asignar a cada nivel de la jerarquía una sangría (o *padding*) del lado izquierdo. Entonces por ejemplo, una entrada de una subsección sería de la siguiente forma:

```
<div class="toc_link" level="2"> ... </div>
```

El nivel de la jerarquía se calcula automáticamente y su valor no necesita ser especificado manualmente; en caso de hacerlo, el valor será sobrescrito por el núcleo de presentación. Las reglas asociadas a los niveles son las siguientes:

```
.toc_link[level="0"] {  
  font-weight: bold;  
}  
.toc_link[level="1"] {  
  padding-left: 1em;  
}  
.toc_link[level="2"] {  
  padding-left: 2em;  
}  
.toc_link[level="3"] {  
  padding-left: 3em;  
}  
.toc_link[level="4"] {  
  padding-left: 4em;  
}
```

Las reglas de estilo son sencillas; en el nivel cero, no se agrega sangría y el texto se muestra en negrita (`font-weight: bold`). En los demás niveles, se añade un espacio mediante un *padding* izquierdo que aumenta según el nivel; y en caso de no requerir el espacio a la izquierda, se puede asignar `padding-left:0`; en cada nivel. Esta forma de aplicar estilos por nivel resulta útil porque permite, por ejemplo, asignar colores o tamaños de letra específicos a las referencias de capítulos, secciones y subsecciones.

De las reglas anteriores, la forma del selector puede resultar extraña a primera vista, ya que utiliza una notación nueva. Por ejemplo, la regla `.toc_link[level="0"]` selecciona todos los elementos con la clase `toc_link` que poseen el atributo `level` con un valor de `0`. Esta forma permite una selección más precisa de variaciones de elementos, lo cual facilita la construcción de reglas más específicas.

El resultado de las reglas predeterminadas presenta una tabla de contenido como la siguiente:

Tabla de contenido

Prefacio	5
1. Título del capítulo 1	7
1.1 Primera sección del capítulo 1	9
1.2 Segunda sección del capítulo 1	9
1.2.1 Primera subsección de la segunda sección del..... capítulo 1	9
1.2.2 Segunda subsección de la segunda sección del..... capítulo 1	9

El estilo por omisión se caracteriza por su simplicidad y funcionalidad, permitiendo una relación visual sencilla entre el nombre del capítulo o sección y el número de página correspondiente. El núcleo de presentación establece todas las reglas CSS y el comportamiento, y la mayoría de los libros del proyecto iCartesiLibri mantienen el estilo predeterminado de la tabla de contenido.

Para darle personalidad a un libro interactivo, existen varias formas de crear estilos únicos para la tabla de contenido; sin embargo, en la mayoría de los casos se requiere la construcción de reglas CSS sofisticadas para especificar correctamente las diversas características. A continuación, se presenta la forma en que están estiladas las entradas de la tabla de contenido por defecto, y se proponen algunos ejemplos de estilos personalizados.

Como se presentó anteriormente, el modelo de caja (*Box Model*), presente en HTML, define para cada elemento una caja que engloba su contenido; esta caja incluye relleno (*padding*), margen y borde, componentes esenciales para determinar el tamaño, apariencia y posición del elemento en la página web. Por defecto, los elementos se posicionan secuencialmente según su aparición en el código HTML; siendo que los elementos de tipo bloque se muestran uno debajo del otro, mientras que los elementos en línea (*inline*) se disponen de izquierda a derecha.

La disposición de los elementos bajo el modelo estándar de posicionamiento puede ser modificada mediante desplazamientos absolutos o relativos, sin embargo, esta técnica puede resultar compleja al diseñar disposiciones precisas, como las que requieren seguir una rejilla o organizar elementos en filas y columnas. Por ello, HTML ofrece algunos sistemas de diseño avanzados que proporcionan mayor flexibilidad en la construcción de disposiciones de elementos; estos sistemas están pensados principalmente para diseños fluidos (o responsivos), es decir aquellos que se adaptan automáticamente al tamaño de pantalla del dispositivo en el que se visualizan, lo que se conoce como el modelo de caja flexible o *Flexbox*, el cual pueden ser utilizado para crear diseños precisos y complejos con facilidad.

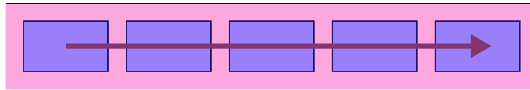
5.4.1 Flexbox

En la revisión del contenido del archivo `index.html` de la plantilla básica, se explicó sobre la “Parte trasera de la portada”; esta página contiene una imagen con el logotipo del proyecto iCartesiLibri, la cual utiliza el modelo *flexbox* con la propiedad CSS `display: flex` para centrarse tanto vertical como horizontalmente.

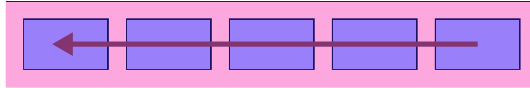
Las entradas de la tabla de contenido emplean el modelo *flexbox* para colocar sus elementos, debido a que este modelo permite, entre otras cosas, alinear sus elementos de forma fácil y particularmente hacer que los puntos que unen el título con el número de página se extiendan para cubrir el área requerida.

El modelo *flexbox* está diseñado para ser utilizado dentro de un contenedor, generalmente un elemento `<div>`, con la finalidad de organizar y acomodar los elementos que contiene; esto se logra asignando al contenedor el atributo `display: flex`, lo cual habilita una serie de atributos CSS específicos para configurar el comportamiento de los elementos dentro del contenedor. Un concepto fundamental a comprender sobre el modelo *flexbox* es que se establece un eje principal que determina la dirección en la que se posicionan los elementos dentro del contenedor, es decir, cada elemento dentro de un contenedor flexible se acomoda en la dirección especificada por el eje principal, ya sea creando renglones o columnas en su disposición. Este eje principal puede configurarse mediante la propiedad `flex-direction` del contenedor flexible; y los valores aceptados son `row`, `row-reverse`, `column` y `column-reverse`, en la **Figura 5.21** se observa cómo se acomodan los elementos dentro del contenedor bajo cada uno de estos valores.

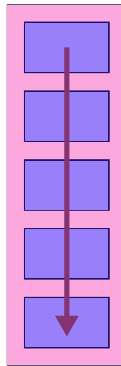
a) `flex-direction: row;`



b) `flex-direction: row-reverse;`



c) `flex-direction: column;`



d) `flex-direction: column-reverse;`

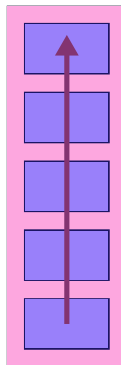


Figura 5.21. Propiedad `flex-direction` con los valores: a) `row`, b) `row-reverse`, c) `column` y d) `column-reverse`.

El comportamiento predeterminado de los descendientes directos de un contenedor flexible es distribuirse en una única fila (`row` y `row-reverse`) o en una sola columna (`column` o `column-reverse`), sin embargo, este comportamiento puede modificarse mediante el atributo `flex-wrap`, que permite los siguientes valores: `nowrap` permite mantener el comportamiento predeterminado e intentar acomodar todos los elementos de forma consecutiva; por otro lado, el valor `wrap` determina que, cuando el contenedor se quede sin espacio, se genere una nueva fila o renglón; y finalmente, el valor `wrap-reverse` es similar a `wrap` pero invierte la dirección donde se coloca la nueva fila o columna.

La propiedad `justify-content` de un contenedor flexible es fundamental para controlar la distribución de los elementos a lo largo del eje principal. Su función principal es distribuir el espacio libre restante cuando los elementos dentro del contenedor hayan alcanzado su tamaño máximo o no puedan expandirse más. El valor predeterminado, `flex-start`, coloca los elementos apilados al inicio del eje principal, por otro lado, `flex-end` ubica los elementos al final del eje principal, mientras que `center` los distribuye de forma centrada.

Además, existen tres valores adicionales que permiten distribuir el espacio: el valor `space-between` provoca que el primer elemento se alinee al comienzo del contenedor y el último al final, distribuyendo el espacio restante de manera equitativa entre el resto de elementos; con `space-around`, se agrega un espacio del mismo tamaño a los lados de los elementos en la dirección del eje principal; y `space-evenly` crea una separación equidistante entre todos los elementos, incluyendo los bordes del contenedor.

En la siguiente imagen se muestra la influencia del parámetro `justify-content`, con una dirección de una fila (`flex-direction:row`).

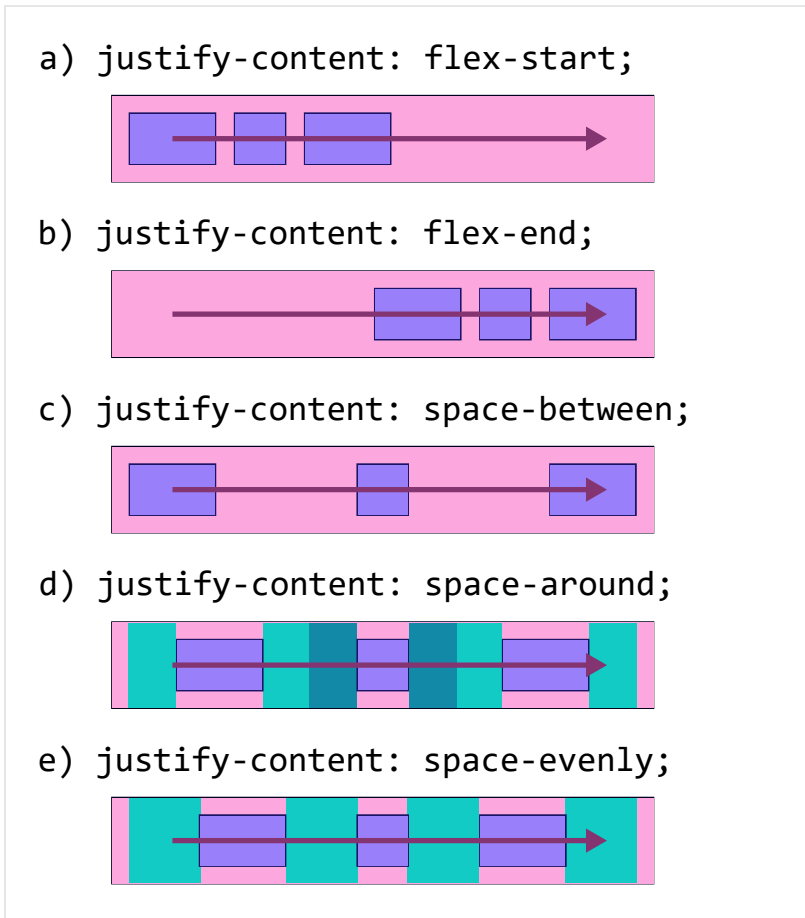


Figura 5.22. Propiedad `justify-content` con los valores: **a)** `flex-start`, **b)** `flex-end`, **c)** `space-between`, **d)** `space-around` y **e)** `space-evenly`.

Es importante entender que la distribución del espacio libre determinada por `justify-content` se realiza en el eje principal, mientras que la alineación en la dirección del eje perpendicular (o trasversal) se controla mediante otra propiedad: `align-items`.

En un contenedor flexible, la propiedad `align-items` define cómo se disponen los elementos respecto al eje transversal, es decir, el eje perpendicular a la dirección principal. Dicha propiedad acepta valores como: `flex-start`, donde los elementos se alinean al inicio; `flex-end`, donde se alinean al final; y `center`, donde se alinean al centro del eje transversal. El valor predeterminado de `align-items` es `stretch`, que estira los elementos para que ocupen todo el espacio disponible en el eje transversal.

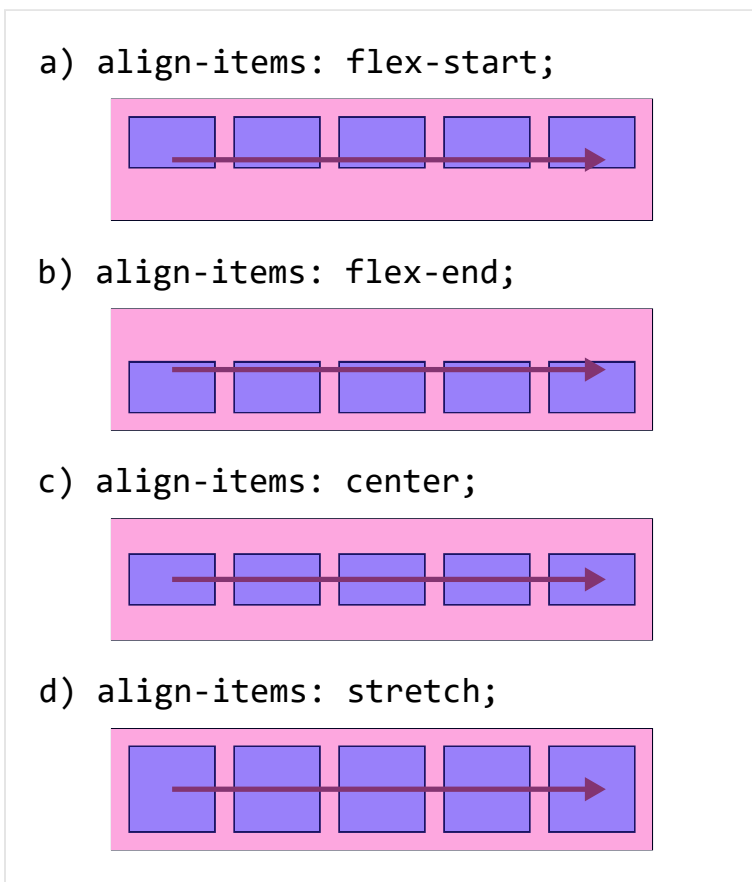


Figura 5.23. Propiedad `align-items` con los valores: a) `flex-start`, b) `flex-end`, c) `center` y d) `stretch`.

Por último, la propiedad `gap` se encarga de agregar explícitamente un espacio entre los elementos de un contenedor flexible, este espaciado únicamente se aplica a los elementos que no se encuentran en los bordes exteriores, es decir, al espacio interior. La propiedad recibe un par de valores numéricos donde el primero determina el espacio entre filas y el segundo el espacio entre columnas.

Las propiedades antes mencionadas se asocian a un contenedor flexible, que utiliza la propiedad `display: flex`. Adicionalmente a estas propiedades en el contenedor, se agregan algunas propiedades a sus descendientes para configurar aún más la forma en que se distribuyen dentro de él; entre estas propiedades, la más importante es `flex-grow`, la cual determina cómo un elemento crece dentro de su contenedor y acepta un valor numérico que representa una proporción que indica qué parte del espacio disponible debe ocupar el elemento, ofreciendo así un control preciso sobre la distribución del espacio ocupado.

Con toda esta información, las reglas para la elaboración de la tabla de contenido deben ser claras:

```
.toc_link {
  display: flex;
  justify-content: space-between;
  flex-wrap: nowrap;
  align-items: flex-start;
  gap: 0.2em;
}
```

Las reglas anteriores establecen que, al no especificarse el valor de `flex-direction`, su valor por omisión es `row`, distribuyendo los tres elementos de una entrada de la tabla de contenido de forma horizontal. La regla `justify-content: space-between` coloca el primer elemento (el título de la entrada) y el último elemento (el número de página) pegados al borde del contenedor, mientras que los puntos intermedios se acomodan con espacios iguales a los lados. La propiedad `flex-wrap: nowrap` obliga a que los tres elementos formen un solo renglón. Por otro lado, `align-items: flex-start` alinea los elementos al inicio del eje transversal; dado que la dirección es horizontal, esto provoca que se alineen verticalmente en particular en la parte superior del contenedor. Finalmente, `gap: 0.2em` agrega un espacio equivalente al veinte por ciento del tamaño de la tipografía entre cada elemento dentro del contenedor.

5.4.2 Ejemplos de estilos

Toda la explicación anterior fue necesaria para una correcta comprensión y se realizó intentando evitar la percepción de que las reglas CSS que las estilan son el resultado de un proceso complejo e inexplicable. El conocimiento del comportamiento de los contenedores flexibles facilita, a su vez, el diseño de otros elementos dentro del libro interactivo, como el centrado de imágenes en sus respectivos contenedores o la creación simplificada de estructuras tabuladas u ordenadas por celdas.

En el primer ejemplo de estilos, la información de la tabla de contenido será presentada de la siguiente manera: primero se mostrará el número de página seguido de los puntos que la conectan y finalmente el título de la entrada (**Figura 5.24**).

Tabla de contenido

5	Prefacio
7	1. Título del capítulo 1
9	1.1 Primera sección del capítulo 1
9	1.2 Segunda sección del capítulo 1
9	1.2.1 Primera subsección de la segunda sección del capítulo 1
9	1.2.2 Segunda subsección de la segunda sección del capítulo 1

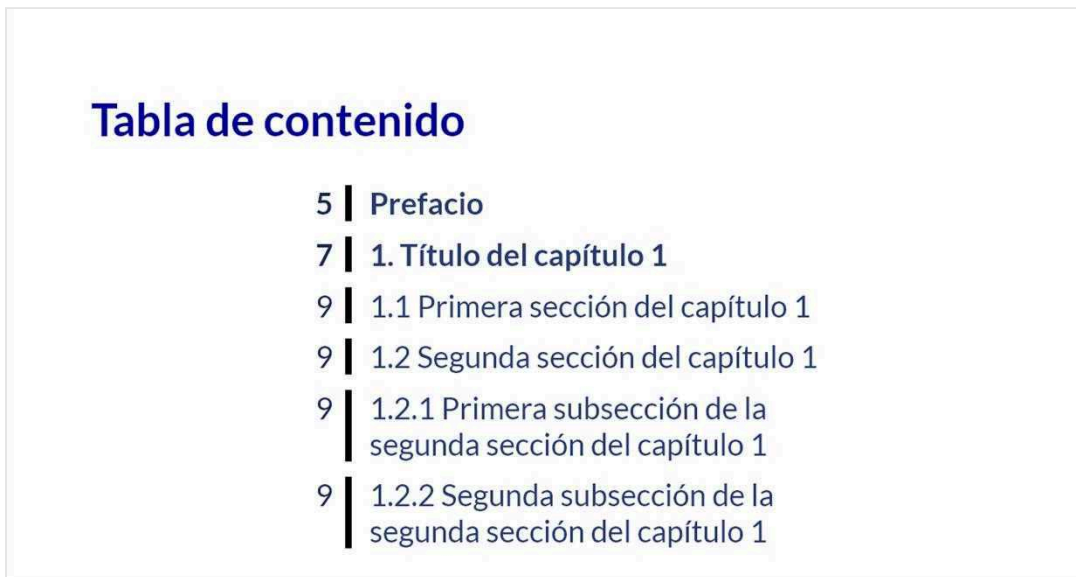
Figura 5.24. Ejemplo de tabla de contenido donde se muestran los elementos de la entrada en el orden invertido, primero el número de la página, luego los puntos y al final el título.

Las reglas utilizadas en este ejemplo son sencillas, se modifica únicamente la dirección del eje principal con `flex-direction: row-reverse` y se elimina el espacio en blanco a la izquierda de cada nivel con `padding-left: 0`.

```
.toc_link {  
    flex-direction: row-reverse;  
}  
  
.toc_link[level="1"], .toc_link[level="2"],  
.toc_link[level="3"], .toc_link[level="4"],  
.toc_link[level="5"] {  
    padding-left: 0;  
}
```

El espacio entre el título y el número de página es cubierto por el elemento ``, para lograr esto, el núcleo de presentación utiliza: `flex-grow: 1`.

Basándose en el conocimiento de la estructura del documento y en el cambio en la dirección en que se acomodan los elementos, se puede crear un estilo con mayor personalidad, agregando reglas CSS para los descendientes de `toc_link`. Por ejemplo, se puede implementar un estilo que oculte los puntos entre los elementos y añada una barra vertical como separador entre el número de página y el título:



The image shows a table of contents with a vertical bar separating the page numbers from the titles. The titles are left-aligned, and the page numbers are right-aligned relative to the vertical bar.

5		Prefacio
7		1. Título del capítulo 1
9		1.1 Primera sección del capítulo 1
9		1.2 Segunda sección del capítulo 1
9		1.2.1 Primera subsección de la segunda sección del capítulo 1
9		1.2.2 Segunda subsección de la segunda sección del capítulo 1

Figura 5.25. Ejemplo de tabla de contenido donde los números de página se encuentran a la izquierda, separados por una línea del título.

Se utiliza el siguiente conjunto de reglas:

```
.toc_link {  
  flex-direction: row-reverse;  
  width: 350px;  
  margin: 0 auto;  
}
```


Las reglas CSS anteriores modifican la dirección de los elementos y establecen un ancho de 350 píxeles para el contenedor. El margen superior e inferior se establece en cero píxeles, mientras que se asigna un margen automático (`auto`) tanto a la izquierda como a la derecha. Se debe tener en cuenta que la aplicación del margen automático solo es efectiva para elementos de tipo bloque.

```
.toc_link .toc_name {  
  width: 300px;  
  border-left: 4px solid black;  
  padding-left: 1ch;  
  text-align: left;  
}
```

Al elemento `toc_name` se le asigna un ancho de 300 píxeles, un borde izquierdo de 4 píxeles de ancho que representa la barra de separación entre el número y el título, además se añade un espacio en blanco equivalente al ancho de un carácter entre la barra y el texto, finalmente el texto se alinea a la izquierda.

```
.toc_link .toc_dots {  
  width: 0;  
  flex-grow: unset;  
  border: none;  
}
```

A los puntos se les asigna un ancho de cero, se previene que se ensanche con `flex-grow: unset` y se elimina el estilo del borde.

```
.toc_link .toc_number {  
  width: 50px;  
  text-align: right;  
}
```

En las reglas del número de página simplemente se le asigna un ancho de 50 píxeles y se alinea a la derecha.

La razón por la que todos los elementos tienen asignado un valor para el ancho, es para garantizar que todas las entradas de la tabla de contenido estén alineadas, independientemente de la longitud de los títulos; cabe destacar que el ancho del contenedor es de 350 píxeles, resultado de la suma del ancho de sus descendientes: 300 + 0 + 50 píxeles. La línea que separa el número de página del título se genera utilizando el borde del título, asegurando que esta línea tenga la misma altura que el contenido textual. Si se desea mostrar una separación como una línea vertical continua, basta con eliminar el espaciado agregando la regla `padding: 0` en el bloque de reglas de `toc_link`.

Tabla de contenido

5	Prefacio
7	1. Título del capítulo 1
9	1.1 Primera sección del capítulo 1
9	1.2 Segunda sección del capítulo 1
9	1.2.1 Primera subsección de la segunda sección del capítulo 1
9	1.2.2 Segunda subsección de la segunda sección del capítulo 1

Los ejemplos, a pesar de su sencillez, demuestran la versatilidad y facilidad con que se pueden crear representaciones personalizadas en la tabla de contenido. Aunque se debe considerar que dependiendo del diseño requerido, el conjunto de reglas puede ser más sofisticado, incluyendo colores, imágenes u otros elementos. A continuación, se presenta otro ejemplo de tabla de contenido:

Tabla de contenido

Prefacio	5
1. Título del capítulo 1	7
1.1 Primera sección del capítulo 1	9
1.2 Segunda sección del capítulo 1	9
1.2.1 Primera subsección de la segunda sección del capítulo 1	9
1.2.2 Segunda subsección de la segunda sección del capítulo 1	9

El diseño de la tabla de contenido anterior se caracteriza por mantener el orden de izquierda a derecha, presentando bordes con esquinas redondeadas alrededor de los títulos y los números de página. Además, los puntos se sustituyen por líneas para unir los elementos, los cuales se centran verticalmente respecto al contenedor. Para lograr esto, se utilizan las siguientes reglas:

```

.toc_link {
  width: 350px;
  margin: 0 auto;
  align-items: center;
  gap: 0;
}
.toc_link .toc_name {
  width: 300px;
  text-align: center;
  border: 3px solid var(--text-color);
  border-radius: 5px;
  padding: 4px;
}
.toc_link .toc_dots {
  width: 10px;
  height: 1px;
  flex-grow: unset;
  border: 3px solid var(--text-color);
}
.toc_link .toc_number {
  width: 40px;
  height: 40px;
  border-radius: 20px;
  border: 3px solid var(--text-color);
  font-size: 20px;
  text-align: center;
  line-height: 35px;
}

```

Similar al ejemplo anterior, todos los elementos reciben un ancho para que se alineen correctamente.

En `toc_link`, la regla `align-items: center` se utiliza para centrar verticalmente a sus descendientes. La mayoría de las reglas utilizadas han sido explicadas previamente, con excepción de la regla `line-height`, la cual se emplea para determinar la altura en que se colocan las letras dentro de la línea de texto, esta regla permite que los números de página se acomoden verticalmente de forma correcta.

Algo que también puede notarse aunque pudiera resultar inusual, es que se utiliza la expresión `var(--text-color)` para definir el color del borde de las cajas que albergan el contenido y la línea que las conecta. La función de la expresión `var(--ID)`, radica en recuperar el valor asociado a una variable personalizada de CSS y asignarlo a alguna propiedad específica. Los tamaños de la página, como se mencionó anteriormente, se definen mediante variables CSS; y para su utilización en otras reglas de estilo, es necesario recurrir a `var`, ya que no permiten un uso directo. En la siguiente sección se presentan todas las variables internas del núcleo de presentación iCartesiLibri asociadas a los colores predeterminados.

Siguiendo el mismo principio utilizado para generar la tabla de contenidos, se puede elaborar una lista ordenada de elementos, conectados mediante puntos. Por ejemplo:

	Resultado
Tabletas entregadas	5,300
Útiles entregados	12,700
Computadoras actualizadas	51,500

A continuación se presenta el código HTML del ejemplo anterior:

```
<div class="renglon_punteado">
  <span>Tabletas entregadas</span>
  <span class="puntos"></span>
  <span>5,300</span>
</div>
<div class="renglon_punteado">
  <span>Útiles entregados</span>
  <span class="puntos"></span>
  <span>12,700</span>
</div>
<div class="renglon_punteado">
  <span>Computadoras actualizadas</span>
  <span class="puntos"></span>
  <span>51,500</span>
</div>
```

Y las reglas CSS correspondientes:

```
.renglon_punteado {
  display: flex;
}

.puntos {
  margin: 0 2px;
  flex: 1;
  border-bottom: 2px dotted red;
  height: 1em;
}
```

5.5 Variables de colores y modo oscuro

El núcleo de presentación proporciona un conjunto de variables CSS que definen los colores predeterminados, similares a las que sirven para especificar las dimensiones de las páginas y sus márgenes, y se utilizan para establecer valores una sola vez y reutilizarlos en múltiples reglas. La aplicación de variables CSS permite mantener la consistencia en el diseño; al cambiar el valor de una variable en un solo lugar, todos los elementos que dependen de esa información se actualizan automáticamente.

El propósito de las variables CSS del núcleo de presentación es brindar a los autores la posibilidad de personalizar los colores de su obra, manteniendo las reglas predeterminadas del libro. Esta funcionalidad permite una modificación sencilla y sin necesidad de crear o editar directamente las reglas CSS; basta con asignar un valor diferente a alguna de las variables de colores para que los cambios se reflejen de inmediato en la presentación.

```
body { --text-color: #ff0000; }  
body.dark { --text-color: #0000ff; }
```

Como se puede observar, las variables se asocian a `body` para el modo claro y a `body.dark` para el modo oscuro. La clase `dark` es agregada automáticamente por el núcleo de presentación cuando se pulsa el switch de cambio de tema ubicado en el área de las herramientas. Hay que mencionar que aunque los nombres de las variables están en inglés, es fácil comprender qué elementos son afectados por cada una de ellas.

Variables de colores para el modo claro:

```
body {
  --book-background-color: #ecf0f1;
  --page_color: #ffffff;
  --text-color: #000000;
  --toc-over-color: #cfe9fc;
  --toc-name-color: #273c75;
  --toc-dots-color: #273c75;
  --toc-number-color: #192a56;
  --toc-selected-color: #273c75;
  --toc-selected-name-color: #ffffff;
  --toc-selected-dots-color: #ffffff;
  --toc-selected-number-color: #ffffff;
  --h1-color: #0089cd;
  --h2-color: #000099;
  --h3-color: #000099;
  --h4-color: #000099;
  --h5-color: #000099;
  --link-text-color: #3867d6;
  --note-color: #3867d6;
  --note-text-color: #1e272e;
  --tooltip-color: #ecf0f1;
  --tooltip-text-color: #130f40;
  --tooltip-border-color: rgba(0,0,0,0.5);
  --interactive-color: #ffffff;
  --interactive-border-color: #2ecc71;
  --image-border-color: #2ecc71;
  --page-separation-color: #e5e5e5;
  --bibliography-color: #0089cd;
  --figcaption-text: #34495e;
  --popup-bib-info-background: #c7ecee;
  --button-border: #8e9091;
  --switch-background-color: #ffffff;
}
```


Variables de colores para el modo oscuro:

```
body.dark {
  --book-background-color: #34495e;
  --page_color: #1e1e1e;
  --text-color: #d4d4d4;
  --toc-over-color: #34495e;
  --toc-name-color: #dddddd;
  --toc-dots-color: #dddddd;
  --toc-number-color: #bbbbbb;
  --toc-selected-color: #dddddd;
  --toc-selected-name-color: #000000;
  --toc-selected-dots-color: #000000;
  --toc-selected-number-color: #000000;
  --h1-color: #358cd6;
  --h2-color: #4fb8fe;
  --h3-color: #4fb8fe;
  --h4-color: #4fb8fe;
  --h5-color: #4fb8fe;
  --link-text-color: #c39178;
  --note-color: #3867d6;
  --note-text-color: #d4d4d4;
  --tooltip-color: #30336b;
  --tooltip-text-color: #dff9fb;
  --tooltip-border-color: rgba(255,255,255,0.5);
  --interactive-color: #2c3e50;
  --interactive-border-color: #ecf0f1;
  --image-border-color: #ecf0f1;
  --page-separation-color: #4b4b4b;
  --bibliography-color: #48dbfb;
  --figcaption-text: #bcbbbc;
  --popup-bib-info-background: #30336b;
  --button-border: #85929e;
  --switch-background-color: #2d3436;
}
```

5.6 Botón de herramientas y configuración

El botón de herramientas, mencionado previamente, se encuentra en la esquina superior derecha, con forma de engrane; su función es mostrar y ocultar una pequeña área donde pueden incluirse múltiples herramientas. La plantilla básica incluye, por defecto, un switch para cambiar entre un tema de colores claro y uno oscuro; los valores de los colores utilizados por defecto se determinan a través de las variables CSS, presentadas en la sección anterior.

5.6.1 Tema de colores claros y oscuros

Respecto al cambio de temas de colores, cabe mencionar que el autor no está obligado a proporcionar la variación de colores para el tema oscuro; de hecho, si no le interesa ofrecer esta variante, se recomienda no incluir el switch bajo el botón de herramientas. Sin embargo, si el autor decide incluir la variación de colores, es importante que las reglas inicien con el selector `body.dark`, ya que el núcleo de presentación se encarga de asignar la clase `dark`, cuando se pulsa el switch de cambio de tema.

Hay que considerar que para cada regla escrita para el tema claro, también debe incluirse una para el tema oscuro. Por ejemplo, si se define la siguiente regla para la clase `resaltado` que en el modo claro asigna un color azul al texto y un fondo rosa:

```
.resaltado {  
  color: blue;  
  background-color: pink;  
}
```

Se tendría que construir una regla similar para el modo oscuro, por ejemplo, que asigne un color púrpura al texto y un color de fondo amarillo:

```
body.dark .resaltado {  
  color: purple;  
  background-color: yellow;  
}
```

Se hace énfasis nuevamente en que `body.dark` debe estar al principio del selector para garantizar su correcto funcionamiento al cambiar al modo oscuro; y se recomienda probar los cambios de tema para asegurar que los colores seleccionados presenten el contraste adecuado para una buena lectura.

Finalmente, hay que mencionar que si se aplica un estilo directamente a una etiqueta mediante su atributo `style`, dicho estilo será visible independientemente del tema de colores seleccionado. De igual manera, un comportamiento predeterminado en el modo oscuro es que tanto los elementos interactivos como las imágenes incluidas directamente en una etiqueta ``, ajustan su contraste y brillo para verse menos luminosas con el tema de color oscuro. En caso de no desear este comportamiento, se puede remover el filtro que modifica el brillo y el contraste agregando el siguiente bloque de reglas CSS:

```
body.dark img, body.dark iframe {  
  filter: none;  
}
```

5.6.2 Inclusión de herramientas externas

El libro interactivo, aunque solo proporcione inicialmente el selector de cambio de tema de color, permite la inclusión de otros botones para agregar múltiples herramientas personalizadas. Entre estas, se pueden incluir calculadoras, graficadoras e incluso juegos pequeños.

En el siguiente interactivo se muestra una plantilla de libro que incorpora la calculadora gráfica Desmo (<https://www.desmos.com/calculator>) la cual se presenta en una nueva ventana al pulsar el botón correspondiente dentro del área de herramientas.



Interactivo 5.2. Plantilla básica que incorpora un botón adicional para mostrar la calculadora gráfica del sitio Desmos. Puede descargarse [aquí](#).

El icono utilizado para la calculadora fue obtenido de [Material Symbols](#), distribuidos gratuitamente bajo una licencia Apache 2.0.

Para incorporar una herramienta se deben seguir algunos pasos, que deben resultar sencillos con todos los conocimientos previamente presentados. Primero, el libro interactivo debe contar con el elemento `<div id="config_options">` y dentro de este elemento, como descendiente, se agrega un nuevo botón `<button>`. Para el ejemplo de la calculadora gráfica Desmos, se utiliza el siguiente código:

```
<div id="config_options">
  <div id="dark_light_mode"></div>
  <div style="text-align:center;">
    <button id="tool_desmos"
onclick="openInteractive('https://www.desmos.com/ca
lculator','800','600')"></button>
  </div>
</div>
```

El código anterior muestra una etiqueta `<div style="text-align:center;">` que funciona como contenedor para los botones que abren las herramientas, y además incluye un estilo que se encarga de centrar el nuevo botón agregado, aunque también es posible alinearlos a la izquierda o la derecha según las necesidades del autor. El botón `<button id="tool_calculadora">` tiene asignado un identificador para facilitar su personalización por medio de reglas CSS, así mismo, tiene asociada una función a su evento de clic, `onclick="openInteractive('https://www.desmos.com/calculator','800','600')"`; esta función se explicó en el capítulo de la estructura de un libro iCartesiLibri (en el apartado de imágenes), y se utiliza para abrir una ventana del navegador con una dirección y dimensiones específicas.

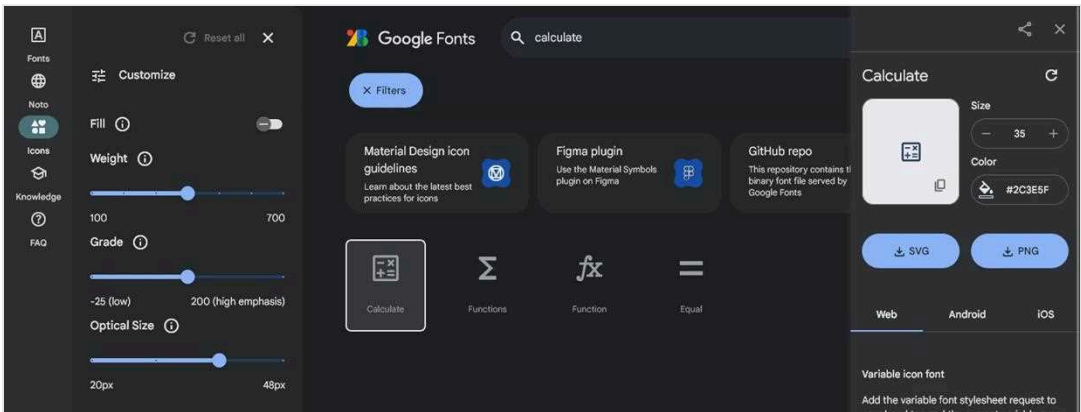
En el ejemplo, la dirección es `'https://www.desmos.com/calculator'`, que corresponde a la ubicación de la calculadora gráfica Desmos; y la ventana donde se muestra tiene 800 píxeles de ancho por 600 píxeles de alto. La función `openInteractive` será responsable de presentar o abrir todas las herramientas personalizadas del libro interactivo, accesibles desde sitios externos como en este ejemplo, o como interactivos locales a los archivos del libro; respecto a las herramientas locales, se recomienda ubicar sus archivos fuente en directorios independientes dentro del directorio `tools/`, para diferenciarlas de las escenas interactivas que forman parte del propio contenido del libro.

Con el botón en su lugar, se crean dos reglas CSS para asignar imágenes al botón de la calculadora, una para el tema claro y otra para el tema oscuro (asumiendo la utilización del cambio de tema de color). La ubicación de las imágenes debe considerar que estas se especifican respecto a la ubicación del archivo de estilos CSS; en este caso, los estilos se encuentran en el archivo `extra/style.css` por lo que la dirección de los iconos debe especificarse respecto al directorio `extra/`.

```
button#tool_desmos {
  width: 35px;
  height: 35px;
  background-image: url("iconos/icono_claro.svg");
}
body.dark button#tool_desmos {
  background-image: url("iconos/icono_oscuro.svg");
}
```

Para mantener los tamaños predeterminados de los botones en el libro interactivo, se recomienda utilizar un botón con 35 píxeles de ancho y alto, como se muestra en las reglas CSS anteriores.

Los iconos del botón fueron obtenidos del sitio <https://fonts.google.com/icons>, donde se encuentran iconos proporcionados por Google bajo la licencia Apache 2.0; esta licencia permite el uso gratuito de los iconos siempre y cuando se incorpore el archivo de texto de la licencia junto a los archivos utilizados (la cual se puede descargar de [aquí](#)). El sitio facilita la obtención de iconos para el libro interactivo gracias a su amplia selección, la posibilidad de descargarlos en formato PNG o SVG, y la opción de personalizar las dimensiones y colores. Como nota, el color de relleno para los iconos en modo claro es #2c3e5f, mientras que para el modo oscuro es #ecf0f1.



Adicionalmente, se puede descargar una tipografía con todos los iconos desde el mismo sitio, aunque se desaconseja su uso a menos que se empleen muchos iconos dentro del libro; esto debido al tamaño considerable de la misma, que podría no justificar su incorporación en el caso de necesitar solo unos pocos iconos.

El área de herramientas está diseñada como una zona para aislar las diversas herramientas que un libro interactivo pueda requerir. En algunos casos, solo se necesita un botón para mostrar una herramienta única, sin necesidad del área de herramientas completa; para este caso, se pueden incluir botones independientes, similares al botón de la tabla de contenido y al de herramientas, que se muestren siempre visibles y se dibujen por encima de las páginas del libro.

En la siguiente plantilla de libro, se observa que el botón de herramientas ha sido reemplazado por el botón de la calculadora Desmos.



Interactivo 5.3. Plantilla básica que incorpora un botón adicional para mostrar la calculadora gráfica del sitio Desmos. Puede descargarse [aquí](#). El icono utilizado para la calculadora fue obtenido de [Material Symbols](#), distribuidos gratuitamente bajo una licencia Apache 2.0.

Para lograr la inclusión de un botón como el presentado en la plantilla anterior, se requiere que el botón sea un descendiente directo del elemento `<div id="book_container">`. Además, debe posicionarse mediante desplazamientos absolutos y aplicarse más estilos, por ejemplo:

```
button#tool_desmos {
  position: absolute;
  width: 35px;
  height: 35px;
  right: 5px;
  top: 5px;
  border: 1px solid var(--button-border);
  border-radius: 4px;
  background-repeat: no-repeat;
  background-size: 95%;
  background-position: center;
  z-index: 1;
  cursor: pointer;
  background-image: url("iconos/icono_claro.svg");
}
body.dark button#tool_desmos {
  background-image: url("iconos/icono_oscuro.svg");
}
```

De las reglas anteriores, solo hay que mencionar que la propiedad `z-index` se utiliza para determinar la profundidad (componente Z) del elemento, en este caso, se requiere asignarle un valor de 1 para que se dibuje encima de las páginas. Por otro lado la propiedad `cursor` controla la representación del apuntador del ratón, siendo en este caso `pointer` para indicar que el elemento puede recibir un clic.

Se recomienda evitar incluir demasiados botones de este estilo para no saturar la interfaz del libro. En caso de necesitar muchas herramientas, se sugiere aprovechar el área proporcionada por el botón de herramientas y configuración.

Como nota final, las herramientas pueden ser incorporadas por un autor experimentado en JavaScript de diversas maneras, mediante la provisión de su propia implementación para presentarlas; esto implica implementar funciones personalizadas para asociarlas con los botones. A modo de ejemplo, se presenta una plantilla de libro que muestra un botón que no abre una herramienta, sino que reproduce o pausa un audio al pulsarlo.



Interactivo 5.4. Plantilla básica que incorpora un botón para reproducir un audio, puede descargarse [aquí](#).

Los iconos utilizados para el audio fueron obtenidos de [Material Symbols](#), distribuidos gratuitamente bajo una licencia Apache 2.0. Music by [Amaksi](#) from [Pixabay](#).

Para lograr la reproducción y pausa del audio, se agrega un botón con el siguiente código HTML:

```
<button id="btn_bg_music" class="bg_music_play"
onclick="this.classList.toggle('bg_music_play');
let myAudio = document.getElementById('bg_music');
myAudio.paused ? myAudio.play() :
myAudio.pause();">
  <audio id="bg_music" loop src="music/night-
detective-226857.mp3"></audio>
</button>
```

Con lo anterior se agrega un botón identificado como `btn_bg_music` y que tiene la clase `bg_music_play`. El evento `onclick` del botón especifica código JavaScript que controla la reproducción de un audio que es descendiente directo del botón, y que tiene el identificador `bg_music`; el audio tiene el atributo `loop` para repetirse indefinidamente. Si bien el presente libro no tiene como objetivo explicar el funcionamiento de JavaScript, se verá cómo este código determina el comportamiento del botón; primero presentando de forma más clara las tres líneas de código que lo componen:

```
01 | this.classList.toggle('bg_music_play');
02 | let myAudio =
    | document.getElementById('bg_music');
03 | myAudio.paused ? myAudio.play() :
    | myAudio.pause();
```

La variable `this` en el código es una referencia al botón de reproducción y pausa, es decir, el elemento que contiene el código JavaScript.

En la primera línea, `this.classList`, accede a la lista de clases asociadas con el botón. Inicialmente, el botón posee la clase `bg_music_play`, la cual determina la imagen de reproducción (play); la función `toggle('bg_music_play')` se encarga de alternar la presencia de esta clase en la lista. En otras palabras: si la clase `bg_music_play` está presente en el botón, la función la elimina; si no está presente, la agrega; esta alternancia se utiliza para cambiar las imágenes de los iconos del botón cada vez que se pulsa, con la ayuda de reglas CSS.

En la segunda línea, con el código `let myAudio = document.getElementById('bg_music')` se crea la variable `myAudio` y se le asigna una referencia al elemento de audio que se quiere reproducir, la cual se obtiene al buscar en el documento un elemento con el identificador `bg_music`.

Con la referencia al audio, en la tercera línea, se verifica si el audio está pausado a través del atributo `paused`. Si se obtiene un valor verdadero (el audio no se está reproduciendo), la reproducción se inicia utilizando `myAudio.play()`. Si, por el contrario, el audio ya se está reproduciendo, este se detiene con `myAudio.pause()`.

Como se puede notar, aunque el comportamiento descrito por el código JavaScript no es complejo de entender, requiere conocimientos adicionales a HTML y CSS, así como habilidades de programación para realizar comportamientos personalizados. Sin embargo, la estructura básica del núcleo de presentación no restringe las posibilidades, sino que ofrece un marco de trabajo sobre el cual construir una interfaz interactiva para el desarrollo de un libro interactivo personalizado.

Para completar el ejemplo, únicamente falta analizar el código CSS correspondiente:

```
button#btn_bg_music {  
  position: absolute;  
  width: 35px;  
  height: 35px;  
  left: 45px;  
  top: 5px;  
  border: 1px solid var(--button-border);  
  border-radius: 4px;  
  background-repeat: no-repeat;  
  background-size: 95%;  
  background-position: center;  
  z-index: 1;  
  cursor: pointer;  
  background-image:  
  url("iconos/bgmusic_pause.svg");  
}
```

```
body.dark button#btn_bg_music {  
  background-image:  
  url("iconos/bgmusic_pause_dark.svg");  
}  
button#btn_bg_music.bg_music_play {  
  background-image: url("iconos/bgmusic_play.svg");  
}  
body.dark button#btn_bg_music.bg_music_play {  
  background-image:  
  url("iconos/bgmusic_play_dark.svg");  
}
```

Los estilos CSS para el botón de reproducción son los mismos que en el ejemplo anterior, es decir, se determinan unas dimensiones de 35 píxeles de ancho por 35 píxeles de alto, y se establece un posicionamiento absoluto. En esta ocasión, se ha modificado la posición para ubicar el botón a la derecha del botón de tabla de contenido, es decir, 45 píxeles a la izquierda del borde de su contenedor. Además, se incluyen todos los estilos necesarios para configurar el borde, así como aquellos necesarios para mostrar y asociar la imagen correspondiente tanto para el tema claro como para el oscuro.

Las dos últimas reglas son las encargadas de cambiar el icono cuando se pulsa sobre el botón, como se puede observar en el selector: `button#btn_bg_music.bg_music_play`, se busca un elemento del tipo botón (`button`) con el identificador `btn_bg_music` y que además tenga la clase `bg_music_play`. Es esta última parte del selector la que permite el cambio, cuando JavaScript alterna la aparición de la clase `bg_music_play`, cada vez que el botón es pulsado.

La presentación anterior, aunque quizá no sea útil para la mayoría de los autores de libros interactivos, fue diseñada con el objetivo de ilustrar cómo, a pesar de su aparente complejidad, gran parte de los conceptos utilizados han sido previamente abordados en este libro, con la excepción, claro está, del uso de JavaScript. Se espera que el lector y potencial autor, motivado por esta información, supere cualquier reticencia que pueda impedirle adentrarse en el aprendizaje del lenguaje web más demandante pero a la vez más gratificante: JavaScript.

5.7 Estilos de comienzo de capítulo

Diversas reglas CSS, aprendidas a lo largo de este libro, pueden aplicarse para lograr variaciones en la presentación de la primera página de cada capítulo. Para ello, es fundamental tener en cuenta el comportamiento de los títulos de capítulos, específicamente los elementos `<h1>` con la clase `chap_name`, ya que estos elementos no incluyen la numeración del capítulo en su título, aunque sí la presentan al ser referenciados en la tabla de contenido. Esta implementación permite incluir la numeración del capítulo por separado, ofreciendo así una mayor posibilidad de personalización de la portada de los capítulos.

5.7.1 Portada de capítulos de la plantilla básica

En la revisión del contenido del archivo `index.html` de la plantilla básica, se habló sobre la página de inicio del capítulo (**Figura 5.26**). Esta presenta una imagen de fondo que ocupa toda la página y en la parte superior un texto grande que dice “Capítulo I” y debajo, el título propio del capítulo “Título del capítulo 1”. Este tipo de portada de capítulo tiene una estructura de código HTML como la siguiente:

```
<div class="page invertcolor" num=""
style="margin:0; padding:0; background-
size:contain; background-
image:url(img/general/portada_capitulos.svg);">
  <div class="chap_text">Capítulo I</div>
  <h1 class="chap_name" id="capitulo_1">Título del
capítulo 1</h1>
</div>
```

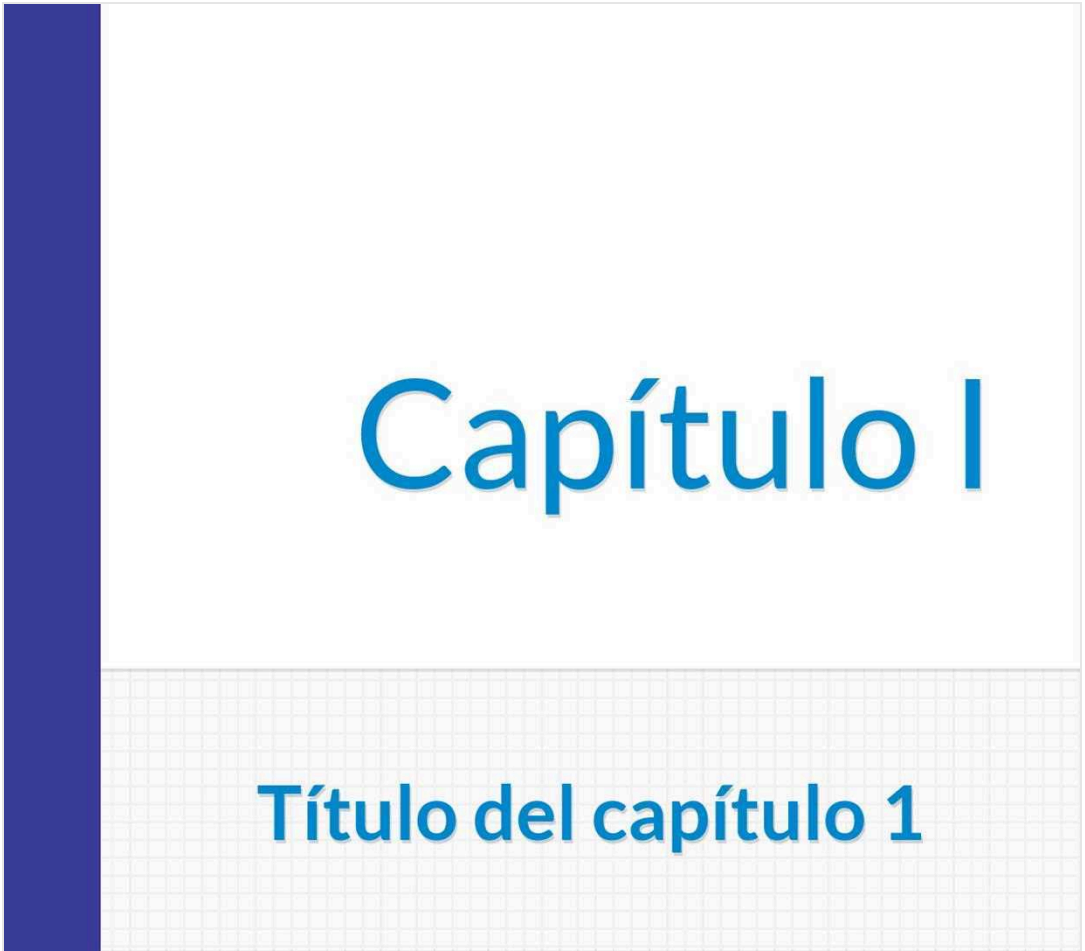


Figura 5.26. Portada al inicio del primer capítulo de la plantilla básica.

En el código anterior, el texto asociado con el elemento de clase `chap_name` aparece directamente, sin incluir el número del capítulo en su inicio, mientras que el número del capítulo se agrega explícitamente a través de un elemento con la clase `chap_text`. De estos dos elementos, solo es requerido el `<h1 class="chap_name">` ya que es el encargado de incrementar el contador de capítulos, mientras que el otro es meramente decorativo para lograr el estilo deseado.

Hay que resaltar que esta presentación no es obligatoria para los autores de libros iCartesiLibri, quienes pueden optar por presentar la información de su libro de manera diferente, y a continuación se presentan algunas variaciones al estilo básico.

5.7.2 Portada de capítulos con una imagen en la misma página del contenido

En este libro, los capítulos inician con una imagen en la parte superior, seguida de una franja sobre la que se muestra el título del capítulo, y el contenido del capítulo comienza inmediatamente después (Figura 5.27).



Figura 5.27. Portada al inicio del primer capítulo utilizada en este libro.

Como se observa, en la portada se muestra el número correspondiente al capítulo actual (en la figura sería el 1.) antes del título del capítulo. Este número puede ser incluido de varias maneras, por ejemplo, agregándolo manualmente fuera del elemento con la clase `chap_name`, o utilizando una regla CSS que utilice el valor del atributo `prefix`, el cual corresponde al número del capítulo y que incluye automáticamente el núcleo de presentación. En el presente libro, se realizó mediante el uso de un pseudoselector para agregar un elemento al vuelo en el documento con el contenido del atributo `prefix`. A continuación se muestra el código HTML del inicio de portada correspondiente:

```
<div class="page" style="padding-top:400px">
  <div class="chap_img" style="background-
image:url('img/Cap1/portada_introduccion.webp');">
</div>
  <h1 class="chap_name"
id="capitulo_1">Introducción</h1>
  <p>Los libros digitales, ...</p>
</div>
```

Del código HTML anterior, se observa que un contenedor con la clase `chap_img` es creado y se le asigna la imagen particular que se muestra al inicio del capítulo, por otra parte, el título se incluye como un elemento con la clase `chap_name` y un identificador para hacer referencia al elemento en la tabla de contenido. Cabe destacar que en ningún momento del código se incluye explícitamente el número de capítulo actual, ya que este valor lo calcula internamente el núcleo de presentación y lo asocia con el atributo `prefix` del elemento con clase `chap_name`.

Mientras que el código CSS correspondiente es:

```
.chap_img {  
  position: absolute;  
  left: 0;  
  top: 0;  
  width: 640px;  
  height: 360px;  
  background-size: cover;  
}
```

```
.chap_name {  
  position: absolute;  
  right: 0;  
  bottom: 485px;  
  color: #0089cd;  
  font-size: 26px;  
  font-family: "PaytoneOne", sans-serif;  
  box-shadow: 0px 4px 4px rgba(0,0,0,0.35);  
  background-color: rgb(255, 255, 255, 0.9);  
  padding: 8px var(--page-right-margin) 8px 20px;  
  border-radius: 10px 0 0 10px;  
  margin: 0;  
}
```

```
.chap_name::before {  
  content: attr(prefix) ". ";  
}
```

Las reglas del primer bloque determinan el comportamiento de la imagen de fondo con la clase `chap_img`, las cuales deben ser claras en este punto, ya que son similares a las utilizadas en la imagen de portada del libro.

Las reglas del segundo bloque estilizan el elemento con la clase `chap_name`, que corresponde a los títulos de los capítulos. Todas las reglas utilizadas en este bloque ya fueron explicadas con anterioridad.

El tercer y último bloque de reglas CSS es el más interesante, ya que utiliza dos conceptos nuevos. Uno de ellos es el pseudoselector `::before`; este se denomina pseudoselector porque el elemento al que apunta realmente no existe dentro del documento HTML, por ejemplo, `p::before` selecciona primero todos los elementos de tipo párrafo y luego a cada uno de ellos se les agrega un nuevo elemento inicialmente vacío de contenido que será mostrado antes del contenido textual del párrafo. En las reglas de la portada de capítulo, primero se seleccionan todos los elementos con la clase `chap_name` (los títulos de los capítulos) y, una vez seleccionados, se les agrega un elemento antes (`::before`) que tiene como contenido el valor del atributo `prefix`, siendo este valor el que es asignado con el atributo `content` del pseudoelemento, y es necesario utilizar la expresión `attr(prefix)` para obtener el valor de dicho atributo. Finalmente, este valor se concatena con el texto dentro de las comillas dobles, correspondientes a un punto seguido de un espacio. Es en este último bloque de estilos donde puede configurarse la apariencia del texto utilizado para mostrar el número del capítulo.

En CSS, se pueden utilizar diversos pseudoselectores para seleccionar elementos que no se definen explícitamente en el documento, así como aquellos que presentan un estado específico o una posición determinada en el árbol del documento.

Los pseudoselectores de contenido, como `::before` y `::after`, permiten insertar contenido nuevo antes o después de un elemento, respectivamente. Este contenido puede ser texto o imágenes y se utilizan para añadir adornos gráficos, prefijos o sufijos a elementos existentes sin modificar su estructura original.

Los pseudoselectores de estado, como `:hover`, seleccionan un elemento cuando el cursor del ratón pasa sobre él, permitiendo cambiar su estilo para indicar interactividad o dar énfasis a elementos importantes; el pseudoselector `:active` selecciona un elemento cuando se hace clic en él y se mantiene pulsado el botón del ratón, siendo útil para mostrar un estado activo o indicar la confirmación de una acción. Por otro lado, `:focus` selecciona un elemento que tiene el foco de atención, como un campo de texto activo, permitiendo destacarlo visualmente o indicar su estado.

Los pseudoselectores de posición, como `:first-child` y `:last-child`, seleccionan el primer o último descendiente de un elemento, respectivamente, permitiendo aplicar estilos específicos a estos elementos, diferenciándolos del resto de sus hermanos. Asimismo, `:nth-child(n)` selecciona un descendiente con una posición específica, donde `n` se refiere al número del elemento dentro de su ancestro directo, lo que permite seleccionar elementos de forma precisa basada en su orden dentro del árbol DOM.

Existen una gran variedad de pseudoselectores y pseudoclasses que pueden consultarse en los siguientes vínculos: <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-elements> y <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>.

5.7.3 Inicio de capítulos puramente textual

En contraste con los ejemplos previos, se pueden elaborar inicios de capítulos más sobrios, donde simplemente se presenta un bloque de texto que indica el número del capítulo y su título, sin ornamentación excesiva. Por ejemplo:

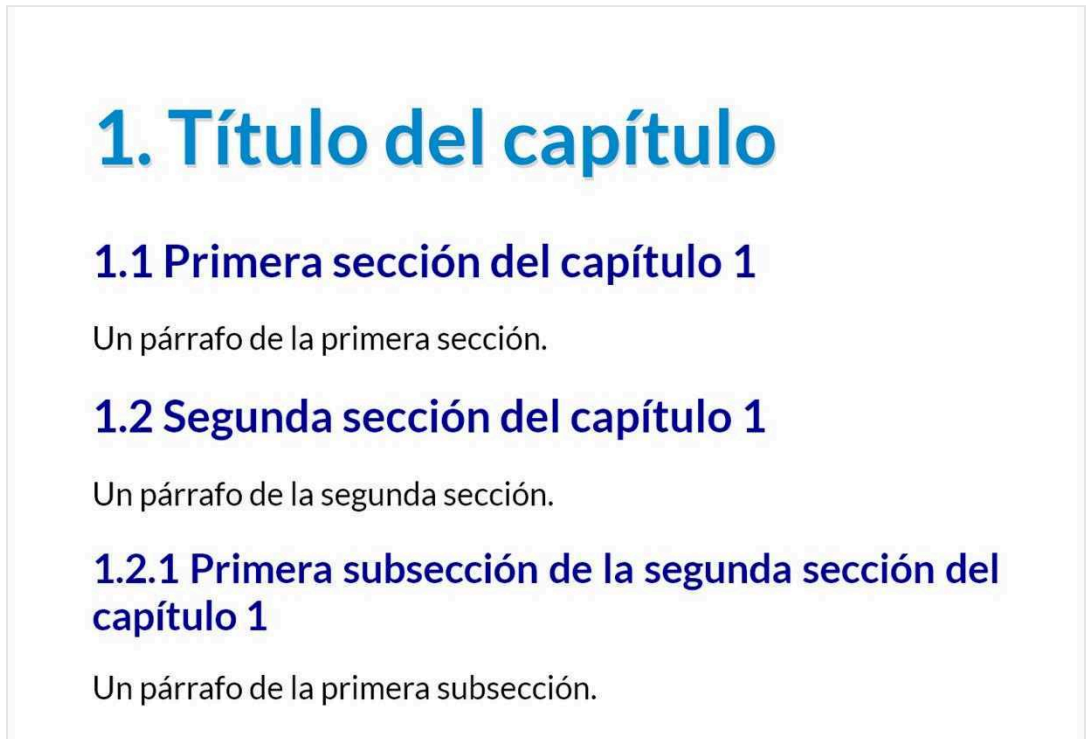


Figura 5.28. Inicio de capítulo con solo texto.

El código HTML para este y los siguientes ejemplos se presenta de la siguiente manera:

```
<h1 class="chap_name" id="capitulo_1">Título del capítulo</h1>
```

Las reglas CSS son las responsables de crear las diversas representaciones, siendo en este caso simplemente:

```
.chap_name {  
    color: #0089cd;  
    font-size: 45px;  
    text-shadow: 1px 2px rgba(0,0,0,0.15);  
}  
  
.chap_name::before {  
    content: attr(prefix) ". ";  
}
```

Es posible que el ejemplo anterior sea excesivamente simple, pero las técnicas vistas en secciones previas pueden ser aplicadas para darle un poco más de estilo. Por ejemplo, se puede enmarcar o encerrar el número del capítulo en un rectángulo, o separarlo mediante una línea, utilizando el borde de los elementos.

El siguiente ejemplo de inicio de capítulo muestra el número del capítulo por encima del título del capítulo, con un tamaño de letra grande y con un círculo que lo envuelve, así mismo se agrega una línea horizontal para dividir el título del capítulo con el inicio del contenido, tal como se observa en la **Figura 5.29**.

1

Título del capítulo

1.1 Primera sección del capítulo 1

Un párrafo de la primera sección.

1.2 Segunda sección del capítulo 1

Un párrafo de la segunda sección.

1.2.1 Primera subsección de la segunda sección del capítulo 1

Un párrafo de la primera subsección.

Figura 5.29. Inicio de capítulo con solo texto y colores de fondo.

Las reglas CSS para el ejemplo anterior son:

```
.chap_name {  
  color: #0089cd;  
  font-size: 30px;  
  text-shadow: 1px 2px rgba(0,0,0,0.15);  
  text-align: center;  
  padding-bottom: 10px;  
  border-bottom: 5px solid aqua;  
}
```



```
.chap_name::before {  
  content: attr(prefix);  
  font-size: 80px;  
  display: block;  
  width: 100px;  
  height: 100px;  
  margin: 0 auto;  
  border-radius: 50%;  
  background-color: aqua;  
}
```

Otro ejemplo sería:

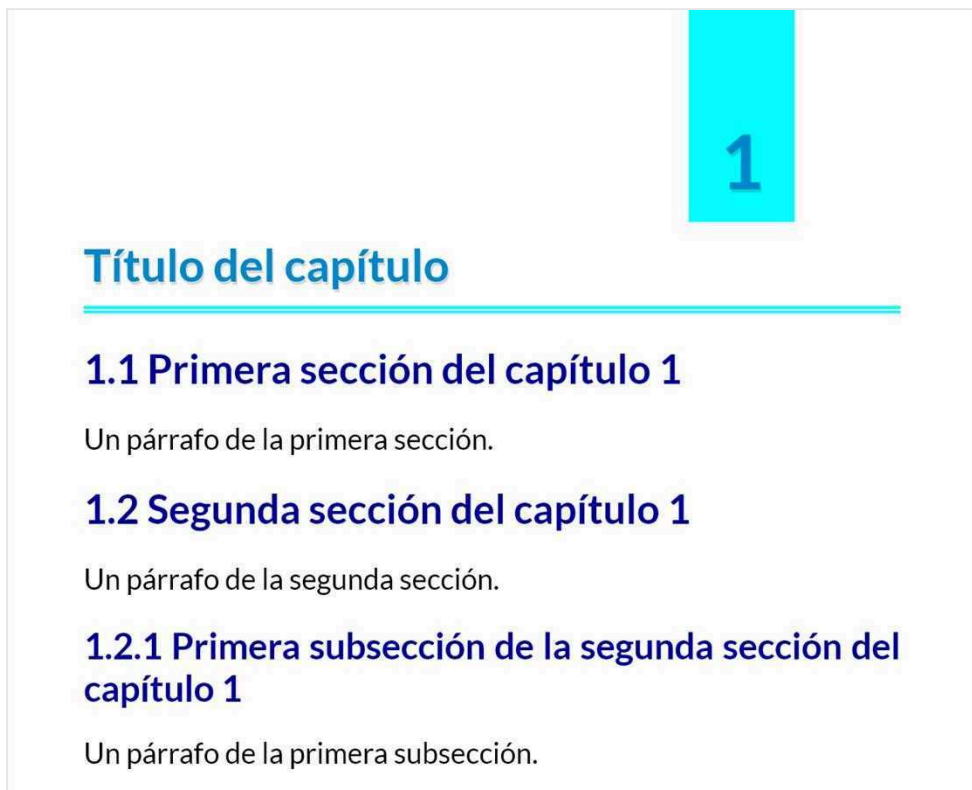


Figura 5.30. Inicio de capítulo con solo texto y colores de fondo.

```
.chap_name::before {
  color: #0089cd;
  font-size: 30px;
  text-shadow: 1px 2px rgba(0,0,0,0.15);
  padding-top: 100px;
  padding-bottom: 10px;
  border-bottom: 5px double aqua;
}
```

```
.chap_name::before {
  content: attr(prefix);
  font-size: 50px;
  display: block;
  position: absolute;
  top: 0;
  right: 120px;
  text-align: center;
  width: 70px;
  height: 140px;
  padding-top: 70px;
  background-color: aqua;
}
```

Al revisar todas las reglas CSS anteriores, se observa que se utilizan propiedades básicas para configurar la apariencia de los elementos. Estas propiedades permiten proporcionar personalidad a los libros, haciendo que la presentación del contenido sea menos árida y más atractiva para los lectores. Es importante recordar que, en caso de utilizar el tema oscuro, deben crearse reglas con `body.dark` al inicio de los selectores y con los colores seleccionados para el tema oscuro.

5.7.4 Imagen para el borde del título del capítulo

El borde de un elemento, generalmente un contenedor, puede ser decorado con una imagen mediante CSS, esta característica se utiliza comúnmente para mejorar la presentación de cajas de texto, como por ejemplo, para adornar los títulos de capítulos.

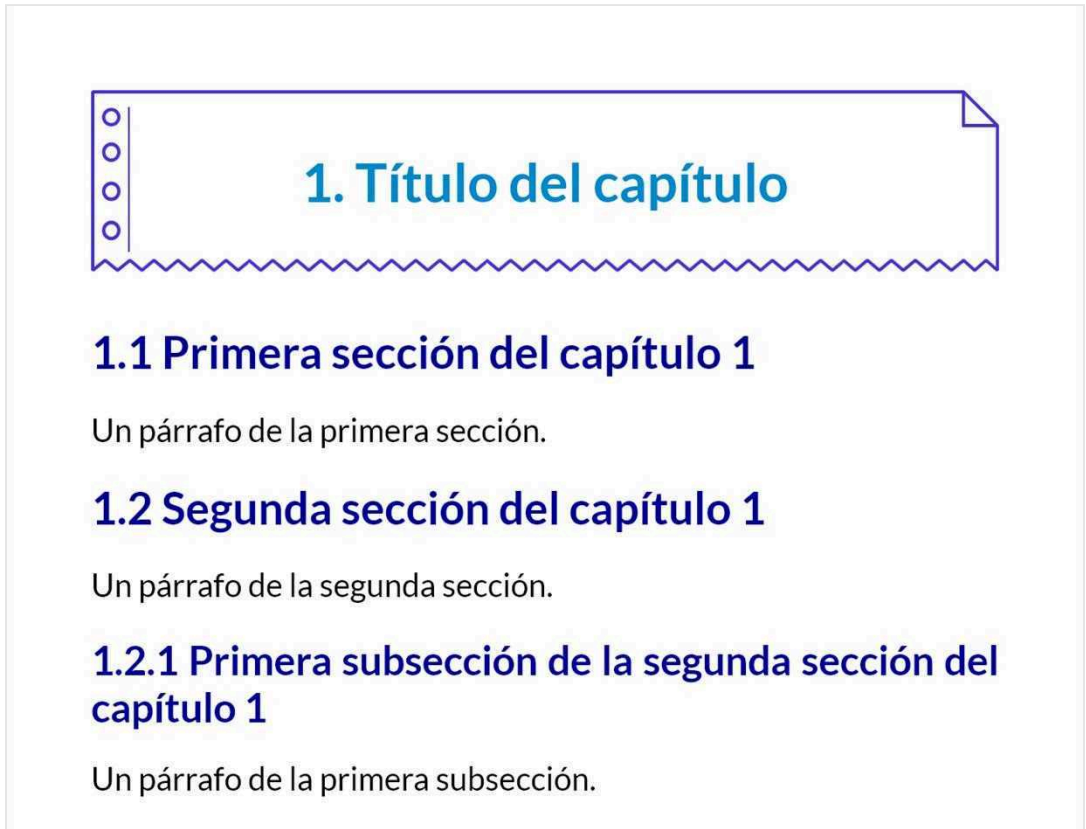


Figura 5.31. Inicio de capítulo con una imagen para decorar el borde del título del capítulo.

A diferencia de utilizar una imagen de fondo en el contenedor, esta técnica está pensada principalmente para decorar las orillas del elemento, ofreciendo la posibilidad de tener cajas de texto de diferentes tamaños con la misma imagen para su borde.

Las reglas CSS para obtener el resultado anterior, son:

```
.chap_name {
  border-image: url('img_borde.svg') 22 24 16 23
  fill / 22px 22px 16px 23px;
  border-image-repeat: round round;
  width: 100%;
  padding: 35px;
  text-align: center;
  margin-bottom: 1em;
  font-size: 32px;
}
.chap_name::before {
  content: attr(prefix) ". ";
}
```

Las nuevas propiedades utilizadas son `border-image` y `border-image-repeat`. La propiedad `border-image`, primero, toma como valor un URL que especifica la ruta de la imagen a utilizar (en este caso, `url('img_borde.svg')`). A continuación, se establecen cuatro valores numéricos que determinan los cortes en la imagen, siguiendo el orden arriba, derecha, abajo e izquierda. Es posible incluir el valor `fill` para rellenar el centro de la imagen o si no se usa, dejarlo transparente. Finalmente, una diagonal (`/`) separa el tamaño de los bordes del contenedor, con valores numéricos que corresponden al orden arriba, derecha, abajo e izquierda. La propiedad `border-image-repeat`, por su parte, recibe dos valores que dictan si la imagen se repite horizontal y verticalmente; el valor `round` asegura una repetición con un número entero de veces, mientras que `repeat` permite repeticiones sin necesidad de ser enteras, y `stretch` estira la imagen para cubrir el borde.

La siguiente imagen muestra y permite analizar los valores de corte utilizados por `border-image` en el ejemplo anterior:

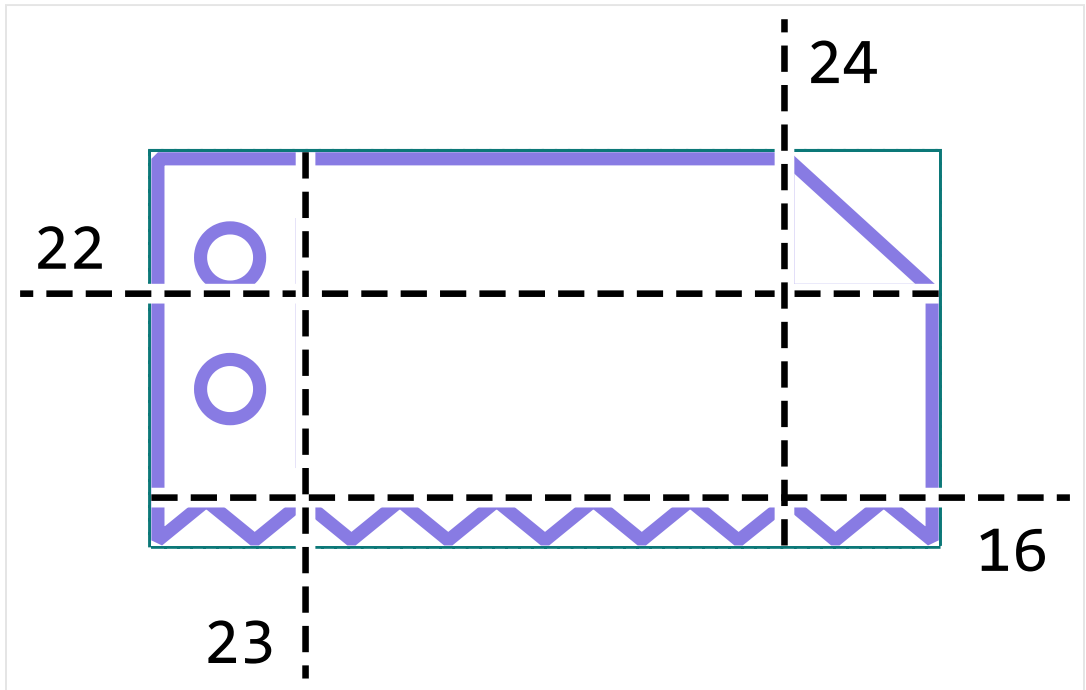


Figura 5.32. Valores de corte de 22 arriba, 24 derecha, 16 abajo y 23 izquierda. La imagen original, puede ser descargada de [aquí](#).

Observando la figura, se determina que estos valores delimitan nueve regiones: cuatro esquinas, cuatro lados y el centro. Las esquinas de la imagen se colocan directamente en las esquinas del contenedor, mientras que los lados pueden repetirse o estirarse según el valor asignado. El centro de la imagen puede ser transparente o sólido, dependiendo de la preferencia.

Como se mencionó previamente, esta técnica se utiliza para agregar bordes más estilizados a los contenedores mediante imágenes y puede emplearse en diversos elementos a lo largo del libro, más allá de la presentación del título.

5.7.5 Inicio de capítulo con una tabla de contenido específica para el capítulo

En algunos casos, se busca incluir una tabla de contenido completa al inicio del libro y, adicionalmente, al inicio de cada capítulo una pequeña tabla de contenido con las entradas correspondientes a ese capítulo (ver **Figura 5.33**).

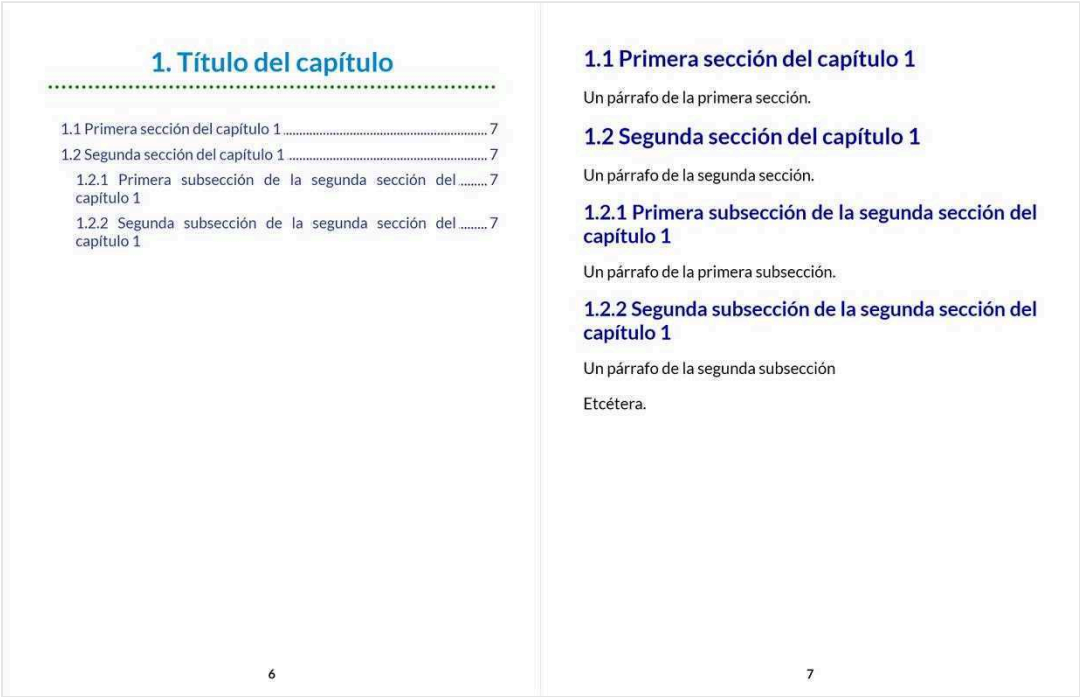


Figura 5.33. Mini tabla de contenido al inicio del capítulo.

Sin embargo, la utilización directa de los elementos previamente vistos para generar estas entradas puede causar un problema en la tabla de contenido externa, accesible mediante el botón situado en la esquina superior izquierda, ya que se producen repeticiones de las entradas (ver **Figura 5.34**).

Prefacio	5
1. Título del capítulo	6
1.1 Primera sección del capítulo 1	7
1.2 Segunda sección del capítulo 1	7
1.2.1 Primera subsección de la segunda sección del capítulo 1	7
1.2.2 Segunda subsección de la segunda sección del capítulo 1	7
2. Título del capítulo 2	9
2.1 Primera sección del capítulo 2	11
2.2 Segunda sección del capítulo 2	11
2.2.1 Primera subsección de la segunda sección del capítulo 2	11
2.2.2 Segunda subsección de la segunda sección del capítulo 2	11
1.1 Primera sección del capítulo 1	7
1.2 Segunda sección del capítulo 1	7
1.2.1 Primera subsección de la segunda sección del capítulo 1	7
1.2.2 Segunda subsección de la segunda sección del capítulo 1	7

Figura 5.34. Entradas duplicadas en la parte inferior de la tabla de contenido externa.

Para solucionar este problema de duplicación de entradas, a los elementos con clase `toc_link` agregados al inicio de cada capítulo, se les puede asignar el parámetro `add_toc="no"`; esta acción previene la inclusión de la entrada en la tabla de contenido externa, manteniendo el comportamiento deseado. De esta manera, se logra la funcionalidad requerida únicamente mediante la adición de elementos HTML sin necesidad de reglas CSS adicionales.

Por ejemplo:

```
<div class="page">
  <h1 class="chap_name" id="capitulo_1">Título del
  capítulo</h1>
  <div class="toc_link" add_toc="no"
  href="#cap1_sec1"></div>
  <div class="toc_link" add_toc="no"
  href="#cap1_sec2"></div>
  <div class="toc_link" add_toc="no"
  href="#cap1_sec2_subsec1"></div>
  <div class="toc_link" add_toc="no"
  href="#cap1_sec2_subsec2"></div>
</div>
```

Se espera que, con todo lo presentado en esta sección, se muestren las posibilidades de personalización que se pueden lograr al inicio de un capítulo. Algunas de las propiedades CSS presentadas pueden ser útiles en otros contextos, como los pseudoselectores `::before` y `::after`, así como la imagen asociada al borde de un contenedor.

5.8 Estilo en la numeración de páginas

En los libros interactivos iCartesiLibri, la numeración de las páginas es un elemento presente en prácticamente todos ellos. Esta se muestra de forma predeterminada en la parte inferior de cada página, alineada con el margen inferior y centrada. Los números de página son agregados por el núcleo de presentación mediante un elemento de tipo `<div>` con la clase `page_number`; dentro de este elemento se incluye otro de tipo `` que contiene el número correspondiente de la página.

Las reglas CSS para la numeración de páginas son las siguientes:

```
.page_number {  
  position: absolute;  
  left: 0;  
  right: 0;  
  bottom: var(--page-bottom-margin);  
  text-align: center;  
  font-weight: bold;  
  font-size: 80%;  
  line-height: 1.2em;  
}
```

Del bloque de reglas anterior se observa que la disposición de los números se determina mediante una posición absoluta en relación a la página que lo contiene. Así mismo, el centrado del número está controlado por la alineación del texto.

Con esta información, se pueden crear reglas fácilmente para colocar el número de página en la parte superior de la página e incluir una línea horizontal (ver **Figura 5.35**).



Figura 5.35. Numeración de las páginas en la parte superior de la página.

Para lograr esto se pueden usar las siguientes reglas:

```
.page_number {  
  bottom: initial;  
  top: 15px;  
  left: var(--page-left-margin);  
  right: var(--page-right-margin);  
  border-bottom: 2px solid var(--text-color);  
}
```

En las reglas anteriores, lo único destacable es el valor `initial` asignado al valor de `bottom`, este valor es necesario para reiniciar el valor por omisión de esta propiedad, y que el posicionamiento se realice correctamente.

Un estilo más interesante para la numeración sería mostrar los números a la izquierda en las páginas pares y a la derecha en las impares (ver **Figura 5.36**). Para esto, es necesario utilizar pseudoselectores, en particular aquel que permite seleccionar elementos en base a su posición consecutiva de un conjunto de elementos previamente seleccionados.

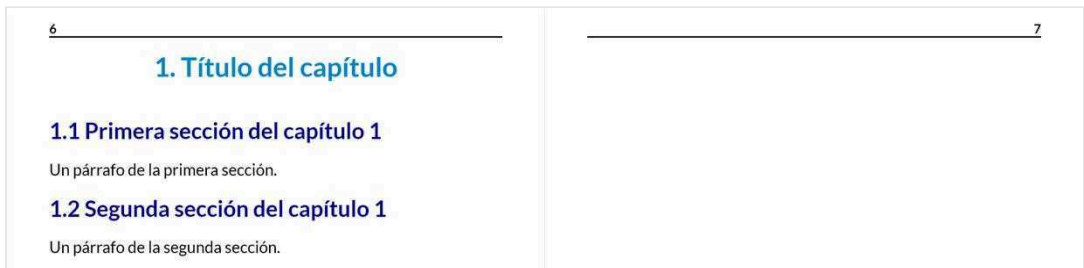


Figura 5.36. Numeración de las páginas en la parte superior de la página, alineada a la izquierda para páginas pares y a la derecha para páginas impares.

El pseudoselector `:nth-child(N)` es utilizado para seleccionar un elemento en base a su posición dentro de un conjunto de elementos. Este pseudoselector puede recibir un valor numérico para `N`, permitiendo la selección del elemento con el índice indicado. Además, se pueden utilizar las palabras clave `odd` para seleccionar los elementos impares y `even` para los elementos pares.

El funcionamiento de este selector implica primero la selección de un conjunto de elementos; en el ejemplo, se seleccionan todas las páginas (elementos con la clase `page`). Una vez seleccionados estos elementos, el pseudoselector `:nth-child` se utiliza para elegir los elementos impares o pares, según el caso. Y finalmente, se seleccionan los elementos con la clase `page_number`, para asignar su alineación. Este proceso puede verse ilustrado en el siguiente código:

```
.page_number {
  bottom: initial;
  top: 15px;
  left: var(--page-left-margin);
  right: var(--page-right-margin);
  border-bottom: 2px solid var(--text-color);
}
.page:nth-child(odd) .page_number {
  text-align: left;
}
.page:nth-child(even) .page_number {
  text-align: right;
}
```

5.9 Generación de un PDF del libro

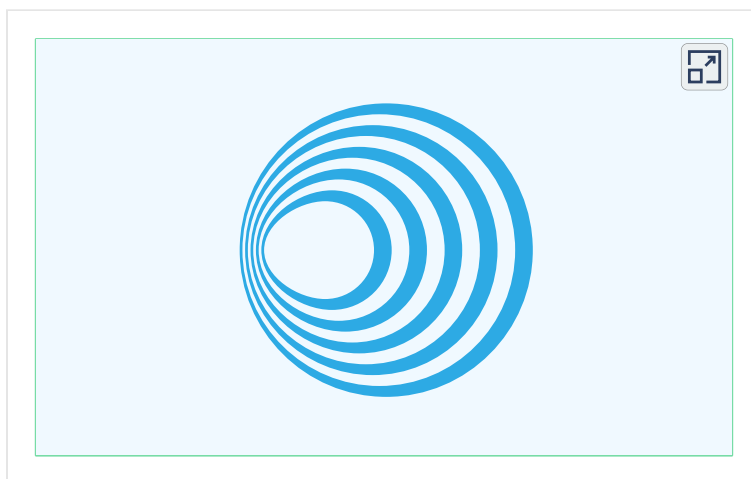
El núcleo de presentación de libros iCartesiLibri ofrece un conjunto de reglas CSS diseñadas para configurar y estilar la impresión del documento. Estas reglas definen parámetros importantes como el tamaño de la página, los márgenes y ocultan elementos no deseados durante la impresión, como las flechas de navegación, los botones de la tabla de contenido y la de herramientas. Su objetivo es garantizar una representación impresa limpia y precisa del contenido del libro.

Sin embargo, la implementación de estas reglas puede presentar dificultades debido a que la mayoría de los navegadores no las respetan completamente, esto puede resultar en una impresión incorrecta del libro. Afortunadamente, algunos navegadores basados en Chromium, como Google Chrome, Microsoft Edge y versiones actualizadas de Safari de Apple, ofrecen soporte para las reglas CSS necesarias para una impresión correcta del libro.

Esta capacidad permite generar un archivo PDF que represente fielmente la versión estática del libro interactivo utilizando las opciones de impresión integradas en estos navegadores. Para ello, se debe ejecutar la acción de “Imprimir” desde el navegador y, en la opción “impresora” o “destino”, seleccionar el valor “Guardar como PDF”. Se recomienda activar la opción “Gráficos de fondo” para asegurar que las imágenes empleadas mediante reglas `background-image:url(...)` sean visibles en el PDF generado. Además, se aconseja evitar modificar los márgenes y utilizar la opción “Ninguno” o “Predeterminados” para aplicar los valores de margen internos definidos en el núcleo del libro interactivo.



Para una correcta generación del PDF, se deben considerar ciertas recomendaciones que permitan aprovechar al máximo las características del núcleo del libro. Por ejemplo, al imprimir un libro a PDF, puede suceder que en lugar del interactivo se visualice una imagen con el logo de la herramienta DescartesJS.



El comportamiento descrito anteriormente puede modificarse agregando al elemento que incluye el interactivo el parámetro `poster`; este parámetro permite especificar la ruta de una imagen que representará al interactivo durante la impresión. Por ejemplo:

```
<div class="interactive" width="819" height="588"
poster="inter/posters/imagen1.png"
src="inter/interactivo1.html"></div>
```

Con esta inclusión, al imprimir el libro, el logo de DescartesJS será reemplazado por la imagen especificada en el parámetro `poster`. Además, el código del libro genera automáticamente un hipervínculo al recurso interactivo utilizando la dirección especificada en el parámetro `src`. Es importante destacar que para que los vínculos se generen correctamente, el libro debe estar alojado en un sitio web en el momento de realizar la impresión; esta condición es necesaria porque desde un PDF no es posible incluir vínculos a archivos locales.

Por lo tanto, si el libro se encuentra alojado en el siguiente sitio:

```
https://prometeo.matem.unam.mx/recursos/VariosNiveles/i
CartesiLibri/recursos/mi_libro_interactivo/index.html
```

Y contiene un interactivo con el valor `inter/escena1/index.html` en el parámetro `src`, el hipervínculo que se crea al imprimir y generar el PDF será:

```
https://prometeo.matem.unam.mx/recursos/VariosNiveles/i
CartesiLibri/recursos/mi_libro_interactivo/inter/escena
1/index.html
```

Además de lo mencionado anteriormente, el libro genera automáticamente algunos hipervínculos internos, como los que corresponden a los elementos de la tabla de contenido. Sin embargo, es necesario tener en cuenta que esta funcionalidad solo funciona cuando los elementos de la tabla de contenido utilizan el parámetro `href` para hacer referencia al identificador (`id`) de algún elemento dentro del libro. En caso contrario, el enlace interno no puede ser construido.



6. Expresiones matemáticas

La representación precisa de fórmulas matemáticas en libros científicos, especialmente en áreas como las matemáticas, física o química, presenta un desafío considerable; para abordar este problema, HTML ofrece una solución mediante **MathML**, un conjunto de etiquetas que permiten la representación directa de expresiones matemáticas utilizando código HTML. Sin embargo, a pesar de esta posibilidad, la visualización de estas expresiones matemáticas en navegadores web se encuentra limitada, lo que da lugar a variaciones en la apariencia de las fórmulas según el navegador utilizado. Un ejemplo de una expresión en MathML es:

```
<math><mfrac><mn>1</mn><msqrt><mn>2</mn></msqrt></mfrac></math>
```

Con lo que se obtiene la siguiente expresión: $\frac{1}{\sqrt{2}}$

Debido a las diferencias en el soporte para MathML, se vuelve imprescindible el uso de bibliotecas adicionales que proporcionen las herramientas necesarias para representar fórmulas matemáticas de forma confiable. En este contexto, es esencial destacar a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, el cual actúa como un lenguaje y sistema de creación de documentos ampliamente difundido dentro del campo científico, destacado por su capacidad para expresar y mostrar fórmulas matemáticas con una presentación profesional y de alta calidad. A pesar de que no se puede acceder directamente al sistema proporcionado por $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ desde HTML, algunos proyectos buscan replicar la visualización de fórmulas matemáticas; entre estos proyectos se encuentran $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ y MathJax.

6.1 ¿Qué es $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$?

$\text{T}_{\text{E}}\text{X}$ es un sistema de composición tipográfica desarrollado por Donald Knuth con el objetivo de generar textos académicos capaces de representar fórmulas matemáticas complejas de manera profesional. Por otro lado, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ es un conjunto de macros construidos sobre $\text{T}_{\text{E}}\text{X}$ para simplificar su; y se considera una de las versiones más utilizadas para la creación de documentos científicos.

De forma simplificada, puede entenderse a $\text{T}_{\text{E}}\text{X}$ como dos grandes elementos. Primero, similar a lo que ofrecen las etiquetas HTML, proporciona un lenguaje de marcado que permite escribir documentos en texto plano agregando información de estilo mediante comandos. Segundo, es una aplicación que genera un documento estático (como un PDF) a partir de la información proporcionada en su lenguaje de marcado, de cierta forma similar a lo que realiza un navegador al interpretar HTML.

Las marcas o etiquetas dentro de un documento \LaTeX se estructuran como `\comando{valor}`; a diferencia del lenguaje HTML, estas marcas en \LaTeX especifican comandos o funciones que se ejecutan durante la compilación y generación del documento final. Los comandos se identifican por una diagonal (`\`) seguida del nombre del comando, y su alcance generalmente queda delimitado por llaves (`{` y `}`) o, si no se incluyen, se aplican al texto que precede al comando. Por ejemplo, para indicar que un elemento debe ser presentado en negrita en HTML, se utiliza la etiqueta ``, mientras que en \LaTeX se emplea el comando `\textbf{ ... }` donde todo el texto contenido entre las llaves será mostrado en negrita.

Aunque \TeX y \LaTeX ofrecen un sistema completo para el desarrollo de documentos con su propio lenguaje de marcado, en páginas web se prioriza la implementación de los comandos y funcionalidades utilizados para representar fórmulas matemáticas, mientras que el lenguaje HTML se mantiene como la norma para especificar el contenido del documento.

6.1.1 Modo matemático de \LaTeX

Para representar fórmulas matemáticas dentro de un documento \LaTeX , se requiere emplear un ambiente especial conocido como el modo matemático, donde el texto contenido es acomodado y ajustado de forma diferente al flujo normal. Este modo es fundamental para incluir expresiones y ecuaciones matemáticas en documentos \LaTeX así como en libros tipo iCartesiLibri, y para su utilización efectiva, es crucial comprender sus características y las opciones que ofrece.

En el modo matemático, se ofrecen dos modalidades de presentación: una en línea, que permite incluir fórmulas dentro del texto sin interrumpir el flujo, y otra en modo presentación, que se caracteriza por dibujar las fórmulas con mayor tamaño para enfatizar la información y presentarla en su propia línea, rompiendo los párrafos con saltos de línea.

Para indicar el uso del modo matemático con la modalidad en línea, se utilizan los delimitadores `\(la expresión va aquí \)` o `$ la expresión va aquí $`, es decir, la expresión matemática se encuentra acotada por paréntesis con diagonales al inicio o por un signo de dolar al inicio y otro al final.

Por ejemplo, el siguiente código:

```
<p>Este es un párrafo con dos expresiones matemáticas en línea, una es:  $(x^2 + y^2 = 1)$ , y la otra es:  $\frac{1}{2}$ </p>
```

Genera el siguiente resultado:

Resultado

Este es un párrafo con dos expresiones matemáticas en línea, una es: $x^2 + y^2 = 1$ y la otra es $\frac{1}{2}$

Para utilizar el modo matemático en modalidad de presentación, se utilizan los delimitadores `\[la expresión va aquí \]` o `$$ la expresión va aquí $$`, es decir, la expresión matemática esta acotada por corchetes con sus respectivas diagonales al inicio o por dos signos de dolar al inicio y otros dos al final.

Por ejemplo:

```
<p>Este es un párrafo con dos expresiones matemáticas en modo presentación, una es:  $[x^2 + y^2 = 1]$ , y la otra es:  $\frac{1}{2}$ </p>
```

Con lo anterior, se obtiene:

Resultado

Este es un párrafo con dos expresiones matemáticas en modo presentación, una es:

$$x^2 + y^2 = 1$$

, y la otra es:

$$\frac{1}{2}$$

Como se observa, la modalidad de presentación rompe los párrafos para mostrar la expresión en su propia línea de texto; así mismo la fracción se despliega con un tamaño mayor ya que no tiene la restricción de mostrarse dentro del texto de una misma línea. Es importante entender esta sutil diferencia para utilizar de forma efectiva el modo matemático proporcionado por $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

Dentro del modo matemático, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ofrece un conjunto de comandos para especificar expresiones matemáticas; aunque el conjunto de comandos proporcionado por $\text{T}_{\text{E}}\text{X}$ o $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ es mucho más amplio que los comandos soportados por las soluciones que se pueden utilizar en un documento HTML, por tanto en los siguientes apartados se presentan las funcionalidades que se pueden utilizar dentro de las páginas web.

6.1.1.1 Símbolos

En matemáticas, se utiliza un conjunto de símbolos básicos como: $+$, $-$, $*$, $/$, $($, $)$, $[$, $]$, los cuales pueden ser utilizados directamente dentro del modo matemático. En el contexto de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, se pueden usar los símbolos $<$ y $>$; sin embargo, es importante recordar su uso en HTML puede generar problemas, por lo que generalmente se utilizan las entidades HTML correspondientes (`<` y `>`). Lamentablemente, el símbolo ampersand (`&`) tiene un significado dentro del modo matemático, por lo que usar entidades HTML dentro de este contexto puede causar conflictos. Por fortuna, el lenguaje $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, proporciona comandos para incluir símbolos no disponibles en un teclado, como letras griegas, y en particular los símbolos menor y mayor que, se representan mediante los comandos `\lt` y `\gt`, que es el mecanismo recomendado para expresiones matemáticas que los requieran.

Al igual que las entidades HTML, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ proporciona una amplia variedad de símbolos a través de su extenso conjunto de comandos. Para consultar la lista completa de símbolos disponibles se recomienda visitar el siguiente enlace: https://ia.wikipedia.org/wiki/Wikipedia:LaTeX_symbols. La mayoría de estos símbolos son compatibles con bibliotecas como $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ y MathJax para navegadores web, aunque se sugiere verificar su disponibilidad en las tablas de elementos soportados de cada biblioteca.

Es importante tener en cuenta que, al igual que las entidades HTML, los símbolos en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ se identifican mediante sus nombres en inglés. Por ejemplo, para incluir la letra griega alfa se utiliza el comando `\alpha` y para su versión en mayúscula se emplea `\Alpha`.

Ejemplo:

```
<p>Este es un párrafo que utiliza una expresión matemática en línea con algunos símbolos definidos por  $\LaTeX$ :  $(\alpha \lt \beta)$ .</p>
```

Resultado

Este es un párrafo que utiliza una expresión matemática en línea con algunos símbolos definidos por \LaTeX : $\alpha < \beta$.

Algunos símbolos útiles son:

\leq \geq \approx \equiv
 \in \notin \subset \neq

6.1.1.2 Operadores

Los operadores suelen ser funciones de una sola palabra, como el módulo, el logaritmo las funciones trigonométricas y el límite. Para este tipo de funciones, \LaTeX ofrece también un conjunto de comandos para su representación.

Se ofrecen dos comandos para el módulo: \bmod , que simplemente agrega la palabra `mod` a la expresión, y \pmod , que automáticamente agrega paréntesis alrededor. Ejemplo:

```
<p>Este es un ejemplo con  $\bmod$ :  $a \bmod b$ ; y este es con  $\pmod$ :  $a \pmod b$ .</p>
```

Resultado

Este es un ejemplo con \bmod : $a \bmod b$; y este es con \pmod : $a \pmod b$.

Para el logaritmo base 10 se dispone del comando `\log`. Por ejemplo:

```
<p>Este es el  $\log x$ .</p>
```

Resultado

Este es el $\log x$.

Estos comandos quizá parezcan superfluos a primera vista, sin embargo, es importante tener en cuenta que, dentro del modo matemático, el texto se comporta de forma distinta; ya que inicialmente se presenta en cursiva y además los espacios en blanco son eliminados debido a que los algoritmos internos son los encargados de determinar la mejor forma de presentar la expresión matemática. Por lo tanto, utilizar simplemente:

```
<p> $\log x$ </p>
```

Genera un resultado diferente a la representación esperada:

Resultado

$\log x$.

El logaritmo natural se representa con `\ln`. Por ejemplo:

```
<p>Este es el  $\ln y$ .</p>
```

Resultado

Este es el $\ln y$.

Las funciones trigonométricas se definen por medio de los comandos: `\sin`, `\cos`, `\tan`, `\ctg`, `\sec` y `\csc`; para el seno, coseno, tangente, cotangente, secante y cosecante, respectivamente.

```
<p> $\sin \alpha$ ,  $\cos \beta$ ,  $\tan \gamma$ ,  $\ctg \delta$ ,  $\sec \eta$ ,  $\csc \theta$ .</p>
```

Resultado

$\sin \alpha$, $\cos \beta$, $\tan \gamma$, $\ctg \delta$, $\sec \eta$, $\csc \theta$.

De forma similar existen los operadores asociados a las funciones trigonométricas inversas: `\arcsin`, `\arccos`, `\arctan` y `\arcctg`; para el arco seno, arco coseno, arco tangente y arco cotangente, respectivamente.

```
<p> $\arcsin \alpha$ ,  $\arccos \beta$ ,  $\arctan \gamma$ ,  $\arcctg \delta$ .</p>
```

Resultado

$\arcsin \alpha$, $\arccos \beta$, $\arctan \gamma$, $\arcctg \delta$.

Para el límite se utiliza el comando `\lim`, y adicionalmente, se puede usar el comando `\limits` para especificar los valores a los que tiende el límite. Por ejemplo:

```
<p>$\lim e^{-x}$</p>
```

```
<p>$\lim \limits_{x \to \infty} e^{-x} = 0$</p>
```

Resultado

$$\lim e^{-x}$$

$$\lim_{x \rightarrow \infty} e^{-x} = 0$$

Como se observa, es necesario utilizar `\limits` para colocar correctamente los valores del límite. En el segundo ejemplo hay que notar la presencia del guión bajo (`_`) después del comando `\limits` y del acento circunflejo (`^`) después de la letra `e`.

6.1.1.3 Potencias e índices

En un contexto matemático, los superíndices se asocian con las potencias, mientras que los subíndices se asocian simplemente con los índices de expresiones. En este caso, como se observó en el ejemplo de límites, el operador guión bajo (`_`) se utiliza para determinar los índices de una expresión; por eso, en el caso del límite, se utiliza después del comando `\limits` para indicar el índice correspondiente, mientras que el operador del acento circunflejo (`^`) se corresponde con los exponentes.

Cuando el índice o el exponente solo es un caracter, las llaves pueden omitirse; si estos requieren de varios caracteres, su utilización es obligatoria, de lo contrario, solo se tomará en cuenta el primer caracter. Ejemplo:

```
<p>Estas expresiones se muestran correctamente:  
$x^2$, $x_0$.</p>
```

```
<p>En contraste, al no utilizar llaves, estas  
expresiones solo consideran el primer caracter:  
$x^{-3}$, $x_{-1}$, y deberían ser $x^{\{-3\}}$, $x_{\{-1\}}$.  
</p>
```

Resultado

Estas expresiones se muestran correctamente: x^2 , x_0 .

En contraste, al no utilizar llaves, estas expresiones solo consideran el primer caracter: x^{-3} , x_{-1} , y deberían ser $x^{\{-3\}}$, $x_{\{-1\}}$.

6.1.1.4 Fracciones

Las fracciones se pueden especificar en \LaTeX mediante expresiones del tipo `\frac{numerador}{denominador}`. El comando `\frac` indica que se trata de una fracción, y entre las llaves se colocan la expresión correspondiente al numerador y la del denominador, respectivamente. Por ejemplo:

```
<p>$\frac{1}{2}$ o $\frac{\frac{1}{2}}{\frac{x^2}{y_0^3}}$</p>
```

Lo que genera:

$$\frac{1}{2} \circ \frac{\frac{1}{2}}{\frac{x^2}{y_0^3}}$$

Cuando se necesitan fracciones complejas que abarcan una área vertical mayor, como la segunda fracción del ejemplo anterior, se recomienda usar el modo de presentación para mejorar su legibilidad. Por ejemplo:

```
##\frac{\frac{1}{2}}{\frac{x^2}{y_0^3}}##
```

Despliega las siguientes fracciones:

$$\frac{\frac{1}{2}}{\frac{x^2}{y_0^3}}$$

Aun así, si se desea que la fracción sea más grande y forme parte de un párrafo, se puede utilizar el comando `\dfrac` el cual intenta representar la fracción en modo de presentación pero sin romper el párrafo. Por ejemplo:

```
<p>Este es un párrafo con una fracción usando
\frac: $\frac{\frac{1}{2}}{\frac{x^2}{y_0^3}}$; y
así mismo se muestra la misma fracción utilizando
\dfrac: $\dfrac{\frac{1}{2}}{\frac{x^2}{y_0^3}}$, y
si bien el párrafo no se separa por saltos de
línea, esta forma de especificar fracciones agrega
espacios verticales entre las líneas del párrafo.
</p>
```

Este es un párrafo con una fracción usando `\frac`: $\frac{\frac{1}{2}}{\frac{x^2}{y^3}}$; y así mismo se muestra la misma fracción utilizando `\dfrac`: $\frac{\frac{1}{2}}{\frac{x^2}{y^3}}$, y si bien el párrafo no se separa por saltos de línea, esta forma de especificar fracciones agrega espacios verticales entre las líneas del párrafo.

En definitiva, la elección del comando específico para representar las expresiones matemáticas depende de la decisión del autor, teniendo en cuenta el contexto de la expresión dentro del texto.

6.1.1.5 Coeficientes binomiales

Los coeficientes binomiales, también conocidos como números combinatorios, se representan generalmente como un par de números enteros colocados uno encima del otro y acotados por paréntesis. Para su representación en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, se utiliza el comando `\binom`. Por ejemplo:

```
<p>\$ \binom{n}{k} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{k \times (k-1) \times \dots \times 1} \$</p>
```

$$\binom{n}{k} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{k \times (k-1) \times \dots \times 1}$$

Al igual que las fracciones, los coeficientes binomiales se muestran de mayor tamaño en modo presentación:

```
$$\binom{n}{k} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{k \times (k-1) \times \dots \times 1}$$
```

Resultado

$$\binom{n}{k} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{k \times (k-1) \times \dots \times 1}$$

Asimismo, existe la variante `\dbinom` que permite mostrar el binomio en modo presentación sin interrumpir el párrafo que lo contiene.

```
<p>$$\dbinom{n}{k} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{k \times (k-1) \times \dots \times 1}$$</p>
```

Resultado

$$\binom{n}{k} = \frac{n \times (n-1) \times \dots \times (n-k+1)}{k \times (k-1) \times \dots \times 1}$$

6.1.1.6 Raíces

El comando para especificar raíces es: `\sqrt`, y la expresión a la que se le aplica la raíz está acotada por llaves, por ejemplo:

```
<p>$$\sqrt{x^2+y^2+z^2}$$</p>
```

Resultado

$$\sqrt{x^2 + y^2 + z^2}$$

Para especificar el índice de la raíz, se utiliza un parámetro opcional incluido entre corchetes antes de la expresión a la que se aplica la raíz. Por ejemplo, para una raíz cúbica, se escribe:

```
<p>\sqrt[3]{\dfrac{a+b+c}{d+e}}</p>
```

Resultado

$$\sqrt[3]{\frac{a+b+c}{d+e}}$$

La raíz se adapta automáticamente a la expresión a la que se aplica, por lo que este comando no dispone de una variante para el modo presentación.

6.1.1.7 Sumas, integrales y operaciones similares

Para agregar una suma, se dispone del comando `\sum`, al cual se pueden especificar sus límites mediante el guión bajo (`_`) y un acento circunflejo (`^`), similar a lo indicado para los límites.

Por ejemplo:

```
<p>\sum_{i=0}^{n+1} s_i</p>
```

Muestra la siguiente suma en línea:

Resultado

$$\sum_{i=0}^{n+1} s_i$$

Y con:

```
$$\sum_{i=0}^{n+1} s_i$$
```

Se muestra la siguiente suma en modo presentación:

Resultado

$$\sum_{i=0}^{n+1} s_i$$

Se observa que el modo presentación muestra la expresión en un tamaño mayor, pero en este caso, además, coloca los límites de la suma en una posición diferente. Para mostrar una suma en formato grande en una línea, no existe un comando como el de las fracciones o los binomios; para lograrlo, se necesita utilizar antes de la suma el comando `\displaystyle`. Por ejemplo:

```
<p>Este párrafo contiene una suma con  
representación grande, $$\displaystyle  
\sum_{i=0}^{n+1} s_i$$, donde al usar el comando  
\displaystyle, el párrafo no se parte con saltos de  
línea.</p>
```

Resultado

Este párrafo contiene una suma con representación grande,
$$\sum_{i=0}^{n+1} s_i$$
, donde al usar el comando `\displaystyle`, el párrafo no se
parte con saltos de línea.

De forma similar funcionan las integrales, pero con el comando `\int`. Ejemplo:

```
<p>$$\int_{i=0}^{n+1} s_i$</p>
```

Resultado

$$\int_{i=0}^{n+1} s_i$$

Y:

```
$$\int_{i=0}^{n+1} s_i$$
```

Resultado

$$\int_{i=0}^{n+1} s_i$$

Al igual que las sumas e integrales, existen varios comandos para agregar operadores grandes, los cuales funcionan de la misma manera; por ello, solo se enlistan a continuación:

`\prod` \prod

`\coprod` \coprod

`\bigoplus` \bigoplus

`\bigotimes` \bigotimes

`\bigodot` \bigodot

`\bigsqcup` \bigsqcup

`\bigcup` \bigcup

`\bigcap` \bigcap

`\bigvee` \bigvee

`\bigwedge` \bigwedge

`\iint` \iint

`\iiint` \iiint

`\oint` \oint

`\oiint` \oiint

`\oiiint` \oiiint

6.1.1.8 Delimitadores

Los delimitadores son símbolos utilizados en matemáticas para marcar el inicio y el final de grupos de elementos o expresiones. Paréntesis, corchetes y llaves desempeñan un papel crucial en la definición de estructura y orden de operaciones; estos, por ser los más comunes, se pueden escribir directamente. Algunos ejemplos son:

```
<p>$( a ), [ b ], \{ c \}, | d |, || e ||$</p>
```

Resultado

$(a), [b], \{c\}, |d|, ||e||$

Además de los delimitadores mencionados anteriormente, \LaTeX incorpora comandos para delimitadores especializados como:

```
<p>$\langle f \rangle, \lceil g \rceil, \lfloor h \rfloor, \llbracket i \rrbracket, \lBrace j \rBrace$</p>
```

Resultado

$\langle f \rangle, \lceil g \rceil, \lfloor h \rfloor, \llbracket i \rrbracket, \lBrace j \rBrace$

Al utilizar estos delimitadores directamente con expresiones que abarcan un área vertical grande (como fracciones, integrales u otros operadores), su representación puede resultar pequeña en comparación.

Por ejemplo, en la siguiente integral, los paréntesis son demasiado pequeños para cubrir completamente la expresión:

```
$$n (\int_{i=0}^n s_i)$$
```

$$n\left(\int_{i=0}^n s_i\right)$$

Para ajustar la altura de algunos delimitadores, \LaTeX ofrece comandos que permiten determinar su tamaño de forma manual. Estos comandos son: `\big`, `\Big`, `\bigg`, `\Bigg`, seguidas de un delimitador. Por ejemplo:

```
<p>$( \big( \Big( \bigg( \Bigg( a \Bigg) \bigg)
\Big) \big) )$,
$[ \big[ \Big[ \bigg[ \Bigg[ a \Bigg] \bigg] \Big]
\big] ]$,
${ \big\{ \Big\{ \bigg\{ \Bigg\{ a \Bigg\} \bigg\}
\Big\} \big\} \} }$,
$| \big| \Big| \bigg| \Bigg| a \Bigg| \bigg| \Big|
\big| |$,
$\| \big\| \Big\| \bigg\| \Bigg\| a \Bigg\| \bigg\|
\Big\| \big\| \|$,
$\lang \big\lang \Big\lang \bigg\lang \Bigg\lang a
\Bigg\rang \bigg\rang \Big\rang \big\rang \rang$,
$\lceil \big\lceil \Big\lceil \bigg\lceil \Bigg\lceil a
\Bigg\rceil \bigg\rceil \Big\rceil \big\rceil \rceil$,
$\lfloor \big\lfloor \Big\lfloor \bigg\lfloor \Bigg\lfloor a
\Bigg\rfloor \bigg\rfloor \Big\rfloor \big\rfloor
\Big\rfloor \big\rfloor \rfloor$</p>
```

$$\begin{array}{ccc}
 (((((a))))), & [[[[[a]]]], & {{{{\{a\}}}}, \\
 |||||a|||, & \langle\langle\langle a \rangle\rangle\rangle, & \lceil\lceil\lceil a \rceil\rceil\rceil, \\
 \lfloor\lfloor\lfloor a \rfloor\rfloor\rfloor, & &
 \end{array}$$

Además de especificar el tamaño manualmente, existen comandos como `\left` y `\right` que intentan ajustar automáticamente la altura de los delimitadores para cubrir lo mejor posible las expresiones. Por ejemplo:

```

<p>$\left( \dfrac{1}{2} \right),
\left[ \dfrac{1}{2} \right],
\left\{ \dfrac{1}{2} \right\},
\left| \dfrac{1}{2} \right|,
\left\| \dfrac{1}{2} \right\|,
\left\langle \dfrac{1}{2} \right\rangle,
\left\lceil \dfrac{1}{2} \right\rceil,
\left\lfloor \dfrac{1}{2} \right\rfloor$</p>

```

$$\left(\frac{1}{2}\right), \left[\frac{1}{2}\right], \left\{\frac{1}{2}\right\}, \left|\frac{1}{2}\right|, \left\|\frac{1}{2}\right\|, \left\langle\frac{1}{2}\right\rangle, \left\lceil\frac{1}{2}\right\rceil, \left\lfloor\frac{1}{2}\right\rfloor$$

6.1.1.9 Entornos

\LaTeX ofrece una variedad de entornos especiales, diseñados para formatear expresiones matemáticas con precisión y profesionalismo dentro de un documento, estos entornos pueden considerarse como marcos o plantillas que definen una estructura específica para la visualización de expresiones como matrices, arreglos o ecuaciones. Cada entorno posee características particulares de representación y usos específicos, y la forma de delimitar los entornos en \LaTeX se realiza mediante los comandos `\begin{ENTORNO}` y `\end{ENTORNO}`, donde `ENTORNO` se reemplaza por el nombre del entorno a utilizar.

Existen diversos entornos para especificar matrices, diferenciándose principalmente por el delimitador empleado. El entorno más sencillo es `matrix`, y en estos entornos, cada fila de la matriz se define en una línea independiente y el comando `\\` se utiliza para indicar un nuevo renglón. Los elementos dentro de cada fila se separan mediante el símbolo ampersand (`&`). Por ejemplo:

```
<p>$\begin{matrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{matrix}$</p>
```

Resultado

$$\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix}$$

El entorno `matrix` únicamente acomoda los elementos de la matriz sin incluir ningún delimitador, como se observa en el ejemplo anterior. La matriz en cuestión contiene tres renglones y el comando `\\` indica el paso al siguiente renglón. Además, se puede notar la presencia del símbolo `&`, lo cual, como se comentó, puede generar problemas con ciertas entidades HTML dentro de las expresiones matemáticas. Aparte del entorno `matrix`, existen otros entornos para matrices: `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` y `Vmatrix`, cada uno de estos entornos proporciona un delimitador específico para la matriz, como paréntesis, corchetes o barras verticales.

```
<p>\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix},
\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},
\begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Bmatrix},
\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix},
\begin{Vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Vmatrix}</p>
```

Resultado

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Bmatrix},$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \begin{Vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Vmatrix}$$

Se proporciona otro entorno que permite expresar diversos casos en una expresión, donde las posibles expresiones se engloban mediante llaves, utilizando los entornos `cases` para la llave izquierda y `rcases` para la llave derecha. A continuación, se presentan algunos ejemplos:

```
<p>$x = \begin{cases}
0 & \text{si } y = 0 \\
1 & \text{si } y > 0 \\
-1 & \text{si } y < 0
\end{cases}$</p>
```

Resultado

$$x = \begin{cases} 0 & \text{si } y = 0 \\ 1 & \text{si } y > 0 \\ -1 & \text{si } y < 0 \end{cases}$$

```
<p>$\begin{rcases}
0 & \text{si } y = 0 \\
1 & \text{si } y > 0 \\
-1 & \text{si } y < 0
\end{rcases} \implies x$</p>
```

Resultado

$$\left. \begin{array}{l} 0 \quad \text{si } y = 0 \\ 1 \quad \text{si } y > 0 \\ -1 \quad \text{si } y < 0 \end{array} \right\} \implies x$$

El entorno `equation` permite la representación de ecuaciones y su numeración automática. Este entorno se distingue de los anteriores por la necesidad de utilizar el modo de presentación para su uso. Un ejemplo de su aplicación se muestra a continuación:

```
$$\begin{equation} x^2 + y^2 = z^2 \end{equation}$$
```

Resultado

$$x^2 + y^2 = z^2 \quad (1)$$

La numeración automática de las ecuaciones se muestra a la derecha de la misma, asignándole un número correspondiente. Si se desea evitar esta numeración, se puede emplear un asterisco al final del nombre del entorno (`equation*`). Ejemplo:

```
$$\begin{equation*} x^2 + y^2 = z^2 \end{equation*}$$
```

Resultado

$$x^2 + y^2 = z^2$$

Para añadir una etiqueta personalizada a una ecuación, se puede utilizar el comando `\tag{...}` dentro de la ecuación:

```
$$\begin{equation} x^2 + y^2 = z^2 \tag{Mi ecuación 1} \end{equation}$$
```

Resultado

$$x^2 + y^2 = z^2 \quad (\text{Mi ecuación 1})$$

Al intentar separar ecuaciones dentro del entorno `equation` por medio de saltos de línea, no se obtiene el resultado esperado, ya que las ecuaciones se muestran en una única línea. Por ejemplo:

```
$$\begin{equation} 2x + 3y = 5 \\ 15x - 7y = -10 \end{equation}$$
```

Genera la siguiente representación:

Resultado

$$2x + 3y = 5 \quad 15x - 7y = -10 \quad (1)$$

Para obtener una separación correcta, se necesita utilizar el entorno adicional `split`, dentro del entorno `equation`, a fin de que los saltos de línea no sean ignorados:

```
$$\begin{equation} \begin{split} 2x + 3y = 5 \\ 15x - 7y = -10 \end{split} \end{equation}$$
```

Resultado

$$\begin{aligned} 2x + 3y &= 5 \\ 15x - 7y &= -10 \end{aligned} \quad (1)$$

Si se desea alinear las ecuaciones respecto al signo igual, se puede utilizar el ampersand (`&`) antes del igual, de esta manera, se indica el lugar donde debe ocurrir la alineación de cada línea, por ejemplo:


```


$$\begin{aligned} 2x + 3y &= 5 \\ 15x - 7y &= -10 \end{aligned}$$


```

Resultado

$$\begin{aligned} 2x + 3y &= 5 \\ 15x - 7y &= -10 \end{aligned} \tag{1}$$

Es importante tener en cuenta que los espacios en blanco y los saltos de línea adicionales son sugerencias que el modo matemático puede o no tomar en cuenta, ya que este intenta acomodar el texto basándose en reglas internas preestablecidas con el objetivo de generar el mejor acomodo posible para la representación de las expresiones.

Si se busca alinear ecuaciones y además enumerar cada una de forma independiente, se puede utilizar el entorno `align`, por ejemplo:

```


$$\begin{aligned} 2x + 3y &= 5 \\ 15x - 7y &= -10 \end{aligned}$$


```

Resultado

$$\begin{aligned} 2x + 3y &= 5 & (1) \\ 15x - 7y &= -10 & (1) \end{aligned}$$

Para alinear ecuaciones sin utilizar la numeración, se puede emplear la variante con asterisco: `align*`.

El entorno `align` puede ser concebido como una matriz, donde el símbolo ampersand (& actúa como un separador, que permite alinear varias ecuaciones, por ejemplo, en relación con el signo de igualdad (=):

```
$$\begin{align*}a&=b & c&=d & e&=f \\x^2&=y^2 & 3z^3&=w^5 & x&=w \\ \alpha&=0 & \beta&=\pi & \delta&=\frac{1}{2}\pi\end{align*}$$
```

Resultado

$$\begin{array}{ccc} a = b & c = d & e = f \\ x^2 = y^2 & 3z^3 = w^5 & x = w \\ \alpha = 0 & \beta = \pi & \delta = \frac{1}{2}\pi \end{array}$$

6.1.1.10 Texto dentro del modo matemático

Como se ha mencionado previamente, el texto en modo matemático se presenta con cursiva y los espacios suelen omitirse; esto puede llevar a errores en la inclusión de fragmentos de texto dentro de expresiones matemáticas. Por ejemplo:

```
<p>$Juanito comió \frac{1}{4} de pizza.$</p>
```

Con lo que se obtiene lo siguiente:

Resultado

Juanitocomió $\frac{1}{4}$ **depizza.**

Para solucionar este comportamiento, \LaTeX ofrece varios comandos que permiten la inclusión de texto con distintos formatos. El comando `\text{...}` se utiliza para un texto regular, `\textit{...}` para un texto en cursiva y `\textbf{...}` para texto en negrita. Ejemplo:

```
<p>$\text{Juanito comió } \frac{1}{4}\text{bf{ de  
pizza}.$</p>
```

Resultado

Juanito comió $\frac{1}{4}$ **de pizza.**

Para representar conjuntos de números como los naturales, los reales o los complejos, se suele utilizar una tipografía específica, la cual se puede obtener utilizando el comando `\mathbb` seguido de una letra. A continuación, se muestra un ejemplo:

```
<p>$\mathbb{N}$, $\mathbb{R}$, $\mathbb{Q}$,  
$\mathbb{I}$, $\mathbb{C}$</p>
```

Resultado

$\mathbb{N}, \mathbb{R}, \mathbb{Q}, \mathbb{I}, \mathbb{C}$

El texto presentado con anterioridad solo constituye una parte de lo que $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ofrece, ya que este sistema, como se mencionó previamente, es un lenguaje completo con numerosas opciones que, generalmente, no se incluyen en las bibliotecas disponibles para HTML. De estas bibliotecas, las dos más relevantes son $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ y MathJax, las cuales pueden incorporarse a cualquier sitio web de forma sencilla.

En las secciones siguientes se expondrá brevemente en qué consiste cada biblioteca, así como los archivos que la componen y cómo pueden agregarse a los libros iCartesiLibri.

6.2 Biblioteca $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ es una biblioteca escrita en el lenguaje JavaScript que se diseñó para dibujar expresiones matemáticas en formato $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ dentro de un navegador web. Desarrollada por Khan Academy como alternativa a MathJax, $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ destaca por su rapidez, bajo espacio de almacenamiento requerido y la ausencia de dependencia con MathML. Su funcionamiento se basa en la construcción de elementos HTML y reglas CSS para representar la fórmulas matemáticas, garantizando así la compatibilidad entre diferentes navegadores.

El sitio oficial de $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$, ubicado en <https://katex.org/>, proporciona acceso a su documentación, siendo la sección sobre funciones soportadas (<https://katex.org/docs/supported>) la más relevante, ya que en esta sección se enumeran las características que implementa y que pueden utilizarse sin problemas dentro de documentos HTML.

El código fuente de $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ se puede acceder desde el repositorio oficial alojado en GitHub (<https://github.com/KaTeX/KaTeX/releases>). La última versión disponible al momento de redactar este documento es la `v0.16.11`, y los archivos de esta versión se encuentran disponibles para su descarga en un archivo comprimido con formato zip (<https://github.com/KaTeX/KaTeX/releases/download/v0.16.11/katex.zip>).

El archivo zip contiene un directorio llamado `contrib/`, el cual alberga archivos adicionales que facilitan la utilización de $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ y amplían sus funcionalidades. Por ejemplo, el archivo `contrib/auto-render.js` se puede incluir para convertir todas las expresiones en $\text{T}_{\text{E}}\text{X}$ dentro de un documento HTML a una representación visualizable por un navegador web. La inclusión del archivo `contrib/copy-tex.js` permite copiar el código el código $\text{T}_{\text{E}}\text{X}$ de las expresiones matemáticas desde el navegador. Además, `contrib/mhchem.js` ofrece la posibilidad de extender las expresiones soportadas, principalmente para representar fórmulas químicas. Se recomienda utilizar las versiones minimizadas (con extensión `.min.js`) de estos archivos, ya que ocupan menos espacio de almacenamiento.

También se encuentra el directorio `fonts/` que contiene un conjunto de tipografías matemáticas que son necesarias para la correcta representación de las expresiones matemáticas. Estas tipografías están disponibles en varios formatos dentro del directorio, pero para visualizar el libro en navegadores modernos, solo se requieren las versiones en `woff2`.

Además de estos dos directorios, se incluyen los archivos `katex.css` (y `katex.min.css`), que contienen las reglas CSS necesarias para la presentación visual. Del mismo modo, se encuentran presentes los archivos `katex.js` (y `katex.min.js`), que contienen el código JavaScript encargado de realizar todos los cálculos necesarios para colocar los elementos de las expresiones en sus posiciones correctas. Al igual que con los archivos ubicados en `contrib/`, se recomienda utilizar la versión minimizada de estos archivos.

Para usar la biblioteca $\text{KAT}_{\text{E}}\text{X}$ dentro de un libro iCartesiLibri, se recomienda descomprimir el archivo zip en el directorio `extra/`, de esta manera, se obtiene el directorio `extra/katex/` con todos los archivos necesarios, y se pueden borrar los archivos no minimizados y las tipografías con formato diferente a `woff2`.

Además de incluir estos archivos, es necesario realizar algunos cambios en el archivo `index.html` para hacer referencia a los archivos de la biblioteca. Estos cambios se deben realizar dentro de la etiqueta `<head>`, de la siguiente manera:

```
01 | <link rel="stylesheet" type="text/css"
    | href="extra/katex/katex.min.css"/>
02 | <script src="extra/katex/katex.min.js">
    | </script>
03 | <script src="extra/katex/contrib/copy-
    | tex.min.js"></script>
04 | <script src="extra/katex/contrib/auto-
    | render.min.js"></script>
```

La primera línea se encarga de incluir el archivo de reglas CSS y la segunda de incluir el código JavaScript.

Se incluye la tercera línea para permitir que el lector copie las expresiones matemáticas en formato $\text{T}_{\text{E}}\text{X}$ y la cuarta línea es necesaria para que las expresiones se desplieguen de forma automática.

Si se decide no colocar los archivos dentro del directorio `extra/`, es necesario ajustar las rutas anteriores para que indiquen correctamente a la ubicación de los archivos.

En el caso de que el libro utilice fórmulas químicas, se recomienda también incluir la siguiente línea:

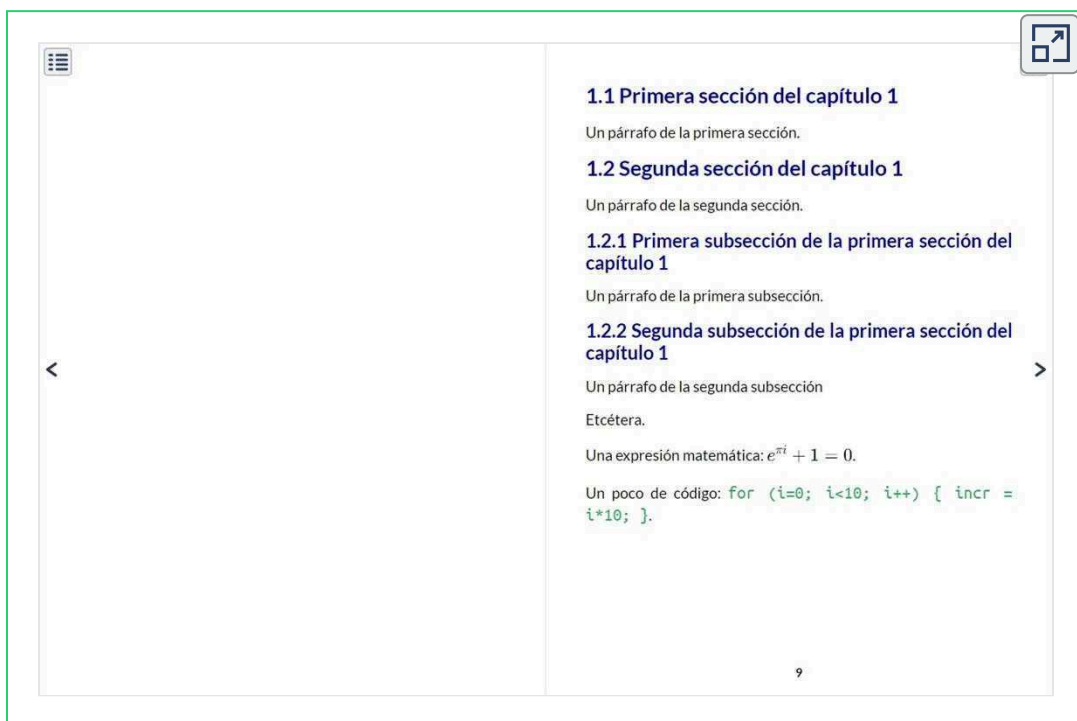
```
<script src="extra/katex/contrib/mhchem.min.js">
</script>
```

Adicionalmente a las inclusiones anteriores, se requiere incorporar un pequeño bloque de código JavaScript para que se ejecute la interpretación de fórmulas escritas en $\text{T}_{\text{E}}\text{X}$ dentro del documento. Esta tarea se realiza con el siguiente código:

```
<script>
document.addEventListener("DOMContentLoaded",
function() {
  renderMathInElement(document.body, {
    delimiters: [ {left: "$$", right: "$$",
display: true}, {left: "$", right: "$", display:
false}, {left: "\\[", right: "\\]", display: true},
{left: "\\(", right: "\\)", display: false} ]
  });
});
</script>
```

El código JavaScript anterior, indica que al momento de terminar de cargar el documento (`DOMContentLoaded`), se buscarán los textos con los delimitadores específicos de $\text{T}_{\text{E}}\text{X}$, y con esa información se genera la representación de todas las expresiones matemáticas del documento.

Para facilitar su uso, se proporciona una plantilla que ya incluye todos los archivos necesarios para que $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ funcione en un libro iCartesiLibri, la cual puede descargarse desde [aquí](#).



The image shows a screenshot of a web page template for a book. The page is divided into two main sections. The left section is a table of contents with a vertical scrollbar and navigation arrows. The right section contains the main content of the page, which is a sample chapter structure. The content includes section headers, paragraphs, and a code block. At the bottom of the page, the number '9' is visible.

1.1 Primera sección del capítulo 1
Un párrafo de la primera sección.

1.2 Segunda sección del capítulo 1
Un párrafo de la segunda sección.

1.2.1 Primera subsección de la primera sección del capítulo 1
Un párrafo de la primera subsección.

1.2.2 Segunda subsección de la primera sección del capítulo 1
Un párrafo de la segunda subsección

Etcétera.

Una expresión matemática: $e^{\pi i} + 1 = 0$.

Un poco de código: `for (i=0; i<10; i++) { incr = i*10; }.`

9

Interactivo 6.1. Plantilla básica con $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

6.3 Biblioteca MathJax

MathJax es una biblioteca JavaScript diseñada para la visualización de notación matemática en páginas web, es compatible con lenguajes de representación de fórmulas matemáticas comunes como \LaTeX , MathML y AsciiMath. Su amplia compatibilidad abarca diversos navegadores web, permitiendo una mayor riqueza en la representación y visualización de fórmulas matemáticas, independientemente del lenguaje o navegador utilizado. A pesar de su amplia compatibilidad, MathJax suele ser un poco más pesada y con un mayor tiempo de ejecución que \KATEX , lo cual se vuelve evidente en sitios con una gran cantidad de fórmulas matemáticas.

El sitio web oficial de MathJax, accesible a través de <https://www.mathjax.org/>, proporciona una guía completa sobre su configuración y uso. La documentación específica para el lenguaje \LaTeX se encuentra disponible en <https://docs.mathjax.org/en/latest/input/te/x/index.html>. Si bien el contenido de este capítulo permite un comienzo práctico con \LaTeX , se recomienda consultar la documentación oficial para explorar todas las funcionalidades que ofrece MathJax. Asimismo, para los interesados, la documentación para el uso de MathML está disponible en <https://docs.mathjax.org/en/latest/input/mathml.html> y para AsciiMath <https://docs.mathjax.org/en/latest/input/asciimath.html>.

El código de MathJax puede ser descargado desde el repositorio oficial, siendo la última versión estable la `v3.2.2`. Esta versión se puede descargar en formato zip desde: <https://github.com/mathjax/MathJax/archive/refs/tags/3.2.2.zip>.

El contenido del archivo zip de MathJax contiene varios archivos; sin embargo, los archivos necesarios para su uso se encuentran dentro del directorio `es5/`. En este directorio se hallan diversos archivos JavaScript, cada uno diseñado para ser utilizado con uno de los lenguajes soportados ($\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, MathML o AsciiMath), por lo que solo es necesario incluir uno de estos archivos; en el caso específico de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, el archivo `es5/tex-cthtml.js` debe ser incluido.

Para utilizar esta biblioteca en un libro tipo iCartesiLibri, se requiere incluir dos `<script>`, uno para la configuración:

```
<script>
  MathJax = { tex: { inlineMath: [['$', '$'], ['\(', '\)']] } };
</script>
```

E inmediatamente después, se debe incluir el siguiente `<script>` para incorporar el archivo adecuado.

```
<script type="text/javascript"
src="extra/MathJax/es5/tex-cthtml.js"></script>
```

Con ambas inclusiones dentro de la etiqueta `<head>`, ya es posible utilizar fórmulas matemáticas en $\text{T}_{\text{E}}\text{X}$ dentro del libro interactivo.

Se proporciona una plantilla que ya incluye todos los archivos necesarios para utilizar MathJax en un libro iCartesiLibri, la cual puede obtenerse desde [aquí](#).

1.1 Primera sección del capítulo 1
Un párrafo de la primera sección.

1.2 Segunda sección del capítulo 1
Un párrafo de la segunda sección.

1.2.1 Primera subsección de la primera sección del capítulo 1
Un párrafo de la primera subsección.

1.2.2 Segunda subsección de la primera sección del capítulo 1
Un párrafo de la segunda subsección
Etcétera.
Una expresión matemática: $e^{\pi i} + 1 = 0$.
Un poco de código: `for (i=0; i<10; i++) { incr = i*10; }.`

9

Interactivo 6.2. Plantilla básica con MathJax.

El autor debe elegir una biblioteca adecuada para representar expresiones matemáticas según sus necesidades específicas, ya que tanto $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ como MathJax tienen características únicas. Aunque $\text{K}^{\text{A}}\text{T}_{\text{E}}\text{X}$ suele ser más rápida y ligera que MathJax en algunos casos, esta última ofrece mayor compatibilidad con comandos $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Así mismo si se busca escribir formulas matemáticas utilizando MathML o AsciiMath, MathJax tiene un excelente soporte para estos lenguajes, por lo que es la opción indicada.

Para un uso efectivo, es crucial consultar la documentación de cada biblioteca para comprender sus limitaciones y posibilidades.



7. Referencia rápida de elementos

Esta sección presenta el código necesario para incorporar todos los elementos que forman parte del núcleo de presentación de libros interactivos tipo iCartesiLibri. Se dirige a usuarios con conocimientos previos sobre iCartesiLibri que buscan incluir elementos específicos en sus libros y desean obtener únicamente el código pertinente para su uso.

Página sin numeración visible

```
<div class="page" num=""> </div>
```

Entrada en la tabla de contenido

```
<div class="toc_link"  
href="#ID_DE_UN_ELEMENTO_DENTRO_DEL_LIBRO"></div>
```

Nota

```
<div class="note">CONTENIDO_DE_LA_NOTA</div>
```

Nota al pie de página

```
<span  
class="footnote">CONTENIDO_DE_LA_NOTA_AL_PIE_DE_PÁG  
INA</span>
```

Recurso interactivo

```
<div class="interactive" width="ANCHO"  
height="ALTO" scale="100%" src="RUTA_DEL_RECURSO"  
poster="RUTA_DE_UNA_IMAGEN_USADA_PARA_EL_PDF">  
</div>
```

Video de YouTube

```
<div class="interactive" width="ANCHO"  
height="ALTO"  
src="https://www.youtube.com/embed/IDENTIFICADOR_DE  
L_VIDEO_EN_YOUTUBE"></div<
```

Video

```
<video preload="none" width="100%" controls=""  
poster="RUTA_DE_UNA_IMAGEN_DE_PREVISUALIZACIÓN_DEL_  
VIDEO">  
  <source src="RUTA_AL_ARCHIVO_DE_VIDEO"  
  type="video/mp4">  
</video>
```

Imagen

```

```

Imagen centrada y que se muestra completa al pulsar sobre ella

```

```

Imagen que abre una nueva ventana con el contenido de un sitio

```

```

Figura con una descripción

```
<figure>
  
  <figcaption>DESCRIPCIÓN_DE_LA_FIGURA</figcaption>
</figure>
```

Figura con una descripción y un prefijo diferente al texto "Figura"

```
<figure
id="IDENTIFICADOR_EN_CASO_DE_HACER_REFERENCIA_A_LA_
FIGURA">
  ELEMENTOS_DE_LA_FIGURA
  <figcaption
prefix="Tabla">DESCRIPCIÓN_DE_LA_FIGURA</figcaption
>
</figure>
```

Bloques autonumerados

```
<div class="num_block"
prefix="PREFIJO_PARA_LA_AUTO_NUMERACIÓN">CONTENIDO_
DEL_BLOQUE_AUTONUMERADO</div>
```

Referencia a figuras o bloques autonumerados

```
<ref ref_id="ID_DEL_ELEMENTO_A_REFERENCIAR"></ref>
```

Referencia a una página, mostrando solo el número de página

```
<pageref
ref_id="ID_DE_ALGÚN_ELEMENTO_DENTRO_DE_UNA_PÁGINA">
</pageref>
```

Referencia a una página, con un texto prefijo y uno sufijo

```
<pageref  
ref_id="ID_DE_ALGÚN_ELEMENTO_DENTRO_DE_UNA_PÁGINA"  
prefix="TEXTO_ANTES_DEL_NÚMERO_DE_PÁGINA"  
postfix="TEXTO_DESPUÉS_DEL_NÚMERO_DE_PÁGINA"  
digits="CANTIDAD_DE_DIGITOS_PARA_EL_NÚMERO_DE_PÁGINA"  
></pageref>
```

Referencia a una página con texto genérico, sin número de página

```
<pageref  
ref_id="ID_DE_ALGÚN_ELEMENTO_DENTRO_DE_UNA_PÁGINA">  
TEXTO_ARBITRARIO_PARA_LA_REFERENCIA</pageref>
```

Entrada bibliográfica

```
<bibitem id="ID_DE_LA_ENTRADA_BIBLIOGRÁFICA">  
  DATOS_DE_LA_ENTRADA_BIBLIOGRÁFICA  
</bibitem>
```

Referencia a una entrada bibliográfica

```
<bibref ref_id="ID_DE_UNA_ENTRADA_BIBLIOGRÁFICA">  
</bibref>
```




8. Bibliografía

- [1] <https://developer.mozilla.org/es/docs/Web/HTML/Element>
- [2] <https://developer.mozilla.org/es/docs/Web/CSS/Reference>
- [3] Juan Guillermo Rivera, Joel Espinosa. **“Diseño de libros interactivos”**. Red Educativa Digital Descartes Córdoba (España). Segunda Edición, 2022.
- [4] Adam Freeman. **“The Definitive Guide to HTML5”**. Apress, 2012.
- [5] Eric A. Meyer, Estelle Weyl. **“CSS: The Definitive Guide”**. O'Reilly Media, 2023.
- [6] Anselm Bradford, Paul Haine. **“HTML5 Mastery: Semantics, Standards, and Styling”**. Apress, 2011.

- [7] Jon Duckett. **“HTML y CSS. Diseño y Construcción de Sitios Web”**. ANAYA MULTIMEDIA, 2024.
- [8] Àlex Bartrolí, Jordi Collell, Anna Ferry. **“HTML5 y CSS3”**. Universitat Oberta de Catalunya.
- [9] Jürgen Wolf. **“HTML and CSS: The Comprehensive Guide”**. Rheinwerk Computing, 2023.

iCartesiLibri