

---

# MÓDULO 1 - PRODUCCIÓN DE TEXTOS E HIPERTEXTOS

---

## MÓDULO TYHM

### **Berard, Louise**

Técnicas y Herramientas Modernas  
Facultad de Ingeniería, Uncuyo  
Mendoza, Argentina  
louise.berard28@gmail.com

### **Ibañez, Celina**

Técnicas y Herramientas Modernas  
Facultad de Ingeniería, Uncuyo  
Mendoza, Argentina  
mcelinaibanez@gmail.com

### **Mobilia, Pilar**

Técnicas y Herramientas Modernas  
Facultad de Ingeniería, Uncuyo  
Mendoza, Argentina  
pilarmobilia@gmail.com

### **Torresi, Carla**

Técnicas y Herramientas Modernas  
Facultad de Ingeniería, Uncuyo  
Mendoza, Argentina  
torresi.carla16@gmail.com

### **Valli, Karima**

Técnicas y Herramientas Modernas  
Facultad de Ingeniería, Uncuyo  
Mendoza, Argentina  
karimavallillalalen@gmail.com

### **Zanella, Bernardita**

Técnicas y Herramientas Modernas  
Facultad de Ingeniería, Uncuyo  
Mendoza, Argentina  
bernizanella2014@gmail.com

Año 2024

## **Abstract**

Este informe proporciona una guía integral para la creación de documentos y presentaciones académicas utilizando herramientas como R Markdown, LaTeX y HTML. Se incluyen ejemplos prácticos y el uso de la plantilla proporcionada por la cátedra para facilitar el aprendizaje y la aplicación de estas tecnologías en la elaboración de contenidos educativos y profesionales.

**Keywords** HTML · Mark Down · LaTeX · Plantilla TyHM

## **1 Introduction**

En la era digital actual, la habilidad para presentar información de manera clara y efectiva es más importante que nunca. Este informe se centra en tres herramientas esenciales para la creación de documentos y presentaciones académicas: HTML, LaTeX y Markdown. Cada una de estas herramientas ofrece ventajas únicas para la estructuración y el diseño de contenido que son indispensables para estudiantes y profesionales por igual.

A lo largo de este informe, destacaremos los aspectos más importantes de HTML, que es el lenguaje de marcado estándar para la creación de páginas web, permitiendo la integración de texto, imágenes y otros contenidos en un formato estructurado y accesible. También exploraremos LaTeX, un sistema de composición de alta calidad que facilita la creación de documentos científicos y técnicos con una presentación impecable. Por último, pero no menos importante, abordaremos Markdown, un lenguaje de marcado ligero que permite convertir texto en HTML de manera eficiente y sin complicaciones.

Para asegurar la coherencia y la calidad en la presentación de nuestros trabajos, utilizaremos la plantilla proporcionada por la cátedra. Esta plantilla no solo nos ayudará a mantener un estándar en la estructuración de nuestros documentos, sino que también nos permitirá enfocarnos en el contenido, sabiendo que el formato ya ha sido cuidadosamente diseñado para nosotros.

```
# Título 1
## Título 2
### Título 3
#### Título 4
##### Título 5
```

Figure 1: Títulos MD

```
Esto es el primer párrafo.

Este es el segundo párrafo.
```

Figure 2: Párrafos MD

Con estas herramientas y la plantilla a nuestra disposición, estamos listos para adentrarnos en el mundo del diseño y la presentación de documentos, mejorando así nuestras habilidades y nuestra capacidad para comunicar nuestras ideas de manera efectiva.

## 2 Mark Down

Para comenzar a utilizar Mark Down hay una serie de pasos que debemos seguir. Comenzamos eligiendo un editor de texto. Podemos usar cualquier editor de texto para escribir en Markdown, como Bloc de Notas (Windows), TextEdit (Mac), Visual Studio Code, Atom o cualquier otro editor. Luego, procedemos a crear nuestro archivo, y a la hora de guardarlo, debemos hacerlo bajo la extensión .md.

Finalmente, para visualizar el contenido realizado, utilizamos un visor de Markdown en línea o un procesador de Markdown integrado en algunos editores de texto. Esto nos permitirá ver cómo se verá el contenido una vez que se convierta a HTML u otro formato.

### 2.1 Títulos

En Markdown, un lenguaje de marcado ligero, podemos añadir títulos utilizando el símbolo de numeral (.). La cantidad de numerales indica el nivel de jerarquía del título. Por ejemplo (Títulos MD):

El número de numerales determina el tamaño y la jerarquía del título. Cuantos mas numerales, menor es el nivel de jerarquía del título. Por ejemplo, Título 1 es el título principal, mientras que Título 5 es un título de quinto nivel, que suele ser el más pequeño y menos prominente.

Algunas implementaciones de Markdown también permiten usar guiones o signos de igual para títulos de diferentes niveles, pero los numerales son la forma más común y recomendada de agregar títulos en Markdown.

### 2.2 Párrafos

En Markdown, podemos separar párrafos simplemente dejando una línea en blanco entre ellos. Por ejemplo (Párrafos MD):

En este ejemplo, hay una línea en blanco entre los dos párrafos, lo que indica que son párrafos separados. Cuando se renderiza este texto en Markdown, los dos párrafos se mostrarán correctamente separados. Es una forma simple y natural de crear párrafos en Markdown.

### 2.3 Negrita

En Markdown, podemos poner palabras en negrita utilizando asteriscos o guiones bajos alrededor de la palabra o frase que deseamos destacar. Por ejemplo (Negrita MD):

Ambas formas producirán el mismo resultado. Simplemente tenemos que asegurarnos de colocar los asteriscos o guiones bajos al principio y al final de la palabra o frase que deseamos resaltar en negrita.

### 2.4 Viñetas

En Markdown, podemos crear listas con viñetas utilizando asteriscos (\*), guiones (-), o signos más (+). Por ejemplo (Viñetas MD):

```
**Esto es texto en negrita**
..Esto también es texto en negrita..
```

Figure 3: Negrita MD

```
* Elemento 1
* Elemento 2
* Elemento 3
```

Figure 4: Viñetas MD

Este código generará una lista con viñetas. También podemos utilizar guiones o signos más para lograr el mismo resultado. Markdown es flexible en cuanto a qué tipo de viñetas preferimos usar, así que podemos elegir la que nos resulte más cómoda.

## 2.5 Imagen

En Markdown, podemos insertar imágenes utilizando la siguiente sintaxis (Imagen1 MD):

Donde: 'Texto alternativo' es el texto que se muestra si la imagen no se puede cargar. 'Ruta de la imagen' es la URL o la ruta relativa de la imagen que deseas insertar. Por ejemplo (Imagen2 MD):

Esto mostrará la imagen con el texto alternativo "Logo de Markdown". Es importante asegurarse de que la ruta de la imagen sea correcta y accesible desde el contexto donde se está visualizando el documento Markdown.

## 2.6 Tabla

En Markdown, podemos crear una tabla utilizando barras verticales para separar las columnas y guiones para separar la primera fila (que generalmente se utiliza como encabezado) de las demás filas. Por ejemplo (Tabla1 MD):

Esto creará una tabla como esta (Tabla2 MD):

Cada línea representa una fila en la tabla. Las barras verticales separan las celdas dentro de una fila, y los guiones en la segunda fila (que sirve como separador) indican la alineación de las columnas (por defecto, las columnas están alineadas a la izquierda). Podemos ajustar la alineación agregando dos puntos : antes o después de los guiones. Por ejemplo, :— indica que la columna está alineada a la izquierda, :—: indica que está centrada, y —: indica que está alineada a la derecha.

## 2.7 Hipervínculo

En Markdown, podemos crear hipervínculos utilizando la siguiente sintaxis (Hiper1 MD):

Donde: 'Texto del enlace' es el texto que se mostrará como enlace. 'URL del enlace' es la dirección web a la que se dirige el enlace. Por ejemplo (Hiper2 MD):

Esto creará un enlace con el texto "Visita Google" que, al hacer clic en él, llevará al usuario a la página web de Google.

```
![[Texto alternativo](ruta_de_la_imagen)]
```

Figure 5: Imagen1 MD



Figure 6: Imagen2 MD

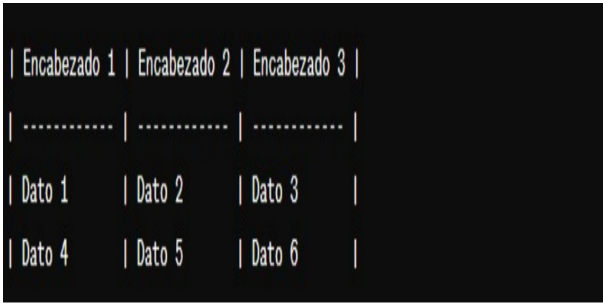


Figure 7: Tabla1 MD

Encabezado1	Encabezado2	Encabezado3
Dato1	Dato2	Dato3
Dato4	Dato5	Dato6

Figure 8: Tabla2 MD

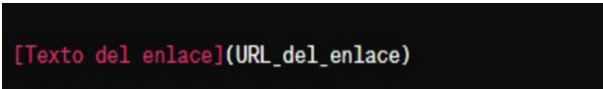


Figure 9: Hiper1 MD

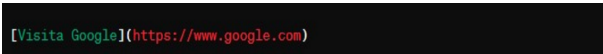


Figure 10: Hiper2 MD

```

<h1> título principal </h1>
<h2> subtítulo </h2>
<h3> subtítulo secundario </h3>

```

Figure 11: Títulos HTML

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Títulos en HTML</title>
</head>
<body>
  <h1>Este es un Título de Primer Nivel (h1)</h1>
  <h2>Este es un Título de Segundo Nivel (h2)</h2>
  <h3>Este es un Título de Tercer Nivel (h3)</h3>
  <h4>Este es un Título de Cuarto Nivel (h4)</h4>
  <h5>Este es un Título de Quinto Nivel (h5)</h5>
  <h6>Este es un Título de Sexto Nivel (h6)</h6>
</body>
</html>

```

Figure 12: Títulos2 HTML

### 3 HTML Language

Para comenzar a utilizar HTML hay una serie de pasos que debemos seguir.

Comenzamos eligiendo un editor de texto. Podemos usar cualquier editor de texto para escribir en Markdown, como Bloc de Notas (Windows), TextEdit (Mac), Visual Studio Code, Atom o cualquier otro editor.

Luego, procedemos a crear nuestro archivo, y a la hora de guardarlo, debemos hacerlo bajo la extensión .html.

Finalmente, para visualizar el contenido realizado, guardamos los cambios en nuestro archivo HTML y lo abrimos en un navegador web para ver cómo se ve nuestra página. Podemos abrirlo directamente desde el navegador haciendo clic en "Abrir archivo" y seleccionando nuestro archivo HTML, o simplemente arrastrando el archivo al navegador.

#### 3.1 Títulos

(Títulos HTML)

En HTML, podemos utilizar etiquetas específicas para crear títulos y encabezados en tu página web. Las etiquetas de encabezado van desde `h1` hasta `h6`, donde `h1` representa el título principal y `h6` el encabezado de menor nivel. Por ejemplo (Títulos2 HTML):

Cuando visualicemos esta página en un navegador, cada título se mostrará con un tamaño y un estilo de fuente diferentes, donde el título `h1` será el más grande y más prominente, y el título `h6` será el más pequeño y menos prominente. Estas etiquetas nos permiten organizar visualmente el contenido de nuestra página web de manera jerárquica.

#### 3.2 Párrafos

En HTML, los párrafos se crean utilizando la etiqueta `p`. Por ejemplo (Párrafos HTML):

Cuando visualizamos esta página en un navegador, veremos el texto dentro de la etiqueta `p` formateado como un párrafo, con un espacio adicional antes y después del texto para dar una apariencia visualmente separada de otros elementos en la página. Podemos utilizar la etiqueta `p` tantas veces como necesitemos para separar nuestro contenido en párrafos distintos.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Párrafos en HTML</title>
</head>
<body>
  <p>Este es un párrafo de ejemplo. En HTML, los párrafos se crean utilizando la et
</body>
</html>

```

Figure 13: Párrafos HTML

```
<b> texto que quiero poner en negrita </b>
```

Figure 14: Negrita1 HTML

### 3.3 Negrita

Para poner negrita se coloca (Negrita1 HTML)

Para aplicar negrita a texto en HTML, podemos utilizar la etiqueta `<strong>` o la etiqueta `<b>`. Ambas etiquetas se utilizan para indicar que el texto contenido dentro de ellas debe mostrarse en negrita. Por ejemplo (Negrita2 HTML):

Ambas etiquetas `<strong>` y `<b>` se renderizarán de la misma manera en la mayoría de los navegadores web modernos. Sin embargo, la etiqueta `<strong>` tiene un significado semántico adicional, ya que indica que el texto es importante o relevante para el contenido, mientras que la etiqueta `<b>` simplemente indica que el texto debe mostrarse en negrita sin implicar ningún significado adicional.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Negrita en HTML</title>
</head>
<body>
  <p>Este es un ejemplo de <strong>texto en negrita</strong> utilizando la etiquet
  <p>Este es otro ejemplo de <b>texto en negrita</b> utilizando la etiqueta <b>
</body>
</html>

```

Figure 15: Negrita2 HTML

```
<ul> texto en una viñeta </ul>
```

Figure 16: Viñetas1 HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Viñetas en HTML</title>
</head>
<body>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
  </ul>
```

Figure 17: Viñetas2 HTML

Por lo tanto, es una buena práctica utilizar la etiqueta `<strong>` cuando desees enfatizar texto de manera significativa.

### 3.4 Viñetas

Para agregar viñetas debemos poner tantas veces como viñetas querramos. (Viñetas1 HTML)

En HTML, podemos crear listas con viñetas utilizando la etiqueta `<ul>` (unordered list) para crear una lista sin orden específico y la etiqueta `<li>` (list item) para cada elemento de la lista. Por ejemplo (Viñetas2 HTML):

Cuando visualicemos esta página en un navegador, veremos una lista con viñetas donde cada elemento está marcado con un punto, un guion o cualquier otro tipo de viñeta predeterminada, dependiendo del estilo del navegador. La etiqueta `<ul>` indica que es una lista sin orden específico, y la etiqueta `<li>` se utiliza para cada elemento de la lista. Podemos agregar tantos elementos como deseemos, simplemente agregando más etiquetas `<li>` dentro de la etiqueta `<ul>`.

### 3.5 Imagen

Para colocar una imagen (Imagen1 HTML):

Para insertar imágenes en HTML, podemos utilizar la etiqueta `<img>`. Esta etiqueta no tiene una etiqueta de cierre, y en su lugar, utiliza atributos para especificar la ubicación y el tamaño de la imagen, así como otros detalles opcionales. Por ejemplo (Imagen2 HTML):

En este ejemplo: El atributo `src` especifica la ruta de la imagen. Podemos proporcionar una URL absoluta si la imagen está alojada en un servidor remoto, o una ruta relativa si la imagen está en el mismo directorio que nuestro archivo HTML. El atributo `alt` proporciona un texto alternativo que se muestra si la imagen no se

```

```

Figure 18: Imagen1 HTML

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Imagen en HTML</title>
</head>
<body>
  
</body>
</html>

```

Figure 19: Imagen2 HTML

```

<tr> agrega fila </tr>
<td> agrega columna </td>

```

Figure 20: Tabla HTML

puede cargar o para usuarios con discapacidad visual. Es importante proporcionar una descripción útil y descriptiva para mejorar la accesibilidad de nuestro sitio web. Podemos ajustar otros atributos, como width y height, para controlar el tamaño de la imagen, así como title para proporcionar un texto emergente cuando se pasa el cursor sobre la imagen.

### 3.6 Tabla

Para armar una tabla debemos colocar dos comandos, uno para filas y otro para columnas. (Tabla HTML)

Para crear una tabla en HTML, podemos utilizar las siguientes etiquetas: `<table>`: Define una tabla en HTML. `<tr>`: Define una fila en la tabla. `<th>`: Define una celda de encabezado en la tabla (opcional). `<td>`: Define una celda de datos en la tabla.

### 3.7 Hipervínculo

Para agregar un hipervínculo (Hipervínculo1 HTML)

Para crear hipervínculos en HTML, podemos utilizar la etiqueta `<a>` (del inglés "anchor"), que significa "ancla". Esta etiqueta requiere un atributo href, que especifica la URL a la que apunta el enlace. Por ejemplo (Hipervínculo2 HTML):

En este ejemplo, `<a href="https://www.ejemplo.com">` indica que el texto "Visita Ejemplo" será un enlace que dirige al usuario a la URL especificada (`https://www.ejemplo.com`) cuando hagan clic en él.

Además del atributo href, podemos incluir otros atributos en la etiqueta `<a>` para personalizar el comportamiento y la apariencia del enlace. Por ejemplo, el atributo tar-

```

<a href="insertar link"> Escribir nombre pag

```

Figure 21: Hipervínculo1 HTML



```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Hipervínculo en HTML</title>
</head>
<body>
  <a href="https://www.ejemplo.com">Visita Ejemplo</a>
</body>
</html>

```

Figure 22: Hipervínculo2 HTML

get="blank"abriraelnlaceenunanuevapestañoaventanadelnavegador, yelatributotitle = "Descripción "proporcionaraunttextoemergentecuandosecoloqueelcursorsobreelenlace.

## 4 LaTeX

Para comenzar a utilizar LaTeX hay una serie de pasos que debemos seguir. Comenzamos eligiendo un editor de texto. Podemos utilizar editores especializados en LaTeX como TeXShop (para macOS), TeXworks, Texmaker, TexStudio o Overleaf (una plataforma en línea). Todos estos editores nos permiten escribir y compilar documentos LaTeX de manera efectiva.

Luego, procedemos a crear nuestro archivo, y a la hora de guardarlo, debemos hacerlo bajo la extensión .tex. Finalmente, utilizamos la opción de compilación del editor para compilar nuestro documento LaTeX. Esto generará un archivo de salida (generalmente en formato PDF) con el documento formateado según las especificaciones de LaTeX.

### 4.1 Introducción

Para comenzar a escribir un documento en LaTeX debemos colocar: "documentclassarticle". Antes de cada comando, como el anterior, debemos colocar una barra inversa invertida.

Antes de comenzar a escribir el cuerpo del documento, es decir, antes de "begindocument", debemos especificar los paquetes que vamos a utilizar, a continuación se detallan los mismos:

- Para viñetas: "usepackageenumitem"
- Para hyperlinks: "usepackagehyperref"
- Para tablas: "usepackageetabularx"
- Para imágenes: "usepackagegraphicx"

Luego de haber enunciado los paquetes, debemos colocar el título del texto con "titlenombre del título". Abajo de este podemos colocar los autores del documento con "authornombre de los autores", donde los nombres se pueden colocar usando simplemente comas o se pueden colocar uno debajo del otro de la siguiente manera: "authorLouise Berard, Bernardita Zanella, etc", colocando 2 barras inversas invertidas entre cada nombre. Debajo podemos colocar la fecha con el siguiente comando: "dateMarzo 2024".

A continuación se comenzará el cuerpo del documento con "begindocument" y se finalizará el mismo con "enddocument". Y luego dentro de estos dos comandos, se colocan los que vamos a describir a continuación para agregar viñetas, títulos y subtítulos, tablas, hyperlinks (tanto links de una página web cómo puede ser la de GitHub como nuestros correos electrónicos), imágenes y texto en negrita.

```

43 \begin{enumerate}
44   \item
45   \item
46   \item
47 \end{enumerate}

```

Figure 23: Viñeta1 LT

```

55 \begin{itemize}
56   \item Primer elemento
57   \item Segundo elemento
58   \item Tercer elemento
59 \end{itemize}

```

Figure 24: Viñeta2 LT

## 4.2 Viñeta

Para enumerar con números, se utiliza el entorno "enumerate". Dentro de este entorno, cada ítem se inicia con el comando "item", seguido del texto que se desea enumerar. Por ejemplo (Viñeta1 LT):

Para utilizar viñetas en lugar de números, se puede usar el entorno "itemize". Dentro de este entorno, cada ítem se inicia con el comando "item", seguido del texto que se desea incluir con viñetas. Por ejemplo (Viñeta2 LT):

Además, en el entorno "itemize", se puede cambiar el tipo de viñeta. Por defecto, se utilizan viñetas redondas, pero se pueden cambiar a viñetas cuadradas, rombos, etc., utilizando el comando "item[]" y especificando el tipo de viñeta entre corchetes.

## 4.3 Tabla

Una tabla en LaTeX se crea utilizando el entorno "table". Dentro de este entorno, usamos el entorno "tabular" para crear la tabla en sí. Primero, se especifica la estructura de la tabla dentro de llaves, donde se indica el formato de cada columna. Por ejemplo, si queremos tres columnas centradas y separadas con barras verticales, colocamos entre 2 llaves, primero una barra vertical, luego una "c", y repetimos esto dos veces más, completándose las tres columnas, y antes de cerrar con la llave colocamos otra barra vertical.

Luego, dentro de la tabla, escribimos los datos línea por línea. Cada línea se separa con el comando "dos barras inversas invertidas". Los elementos de cada columna se separan por el símbolo "et". Para dibujar líneas horizontales en la tabla, usamos el comando "hline" después de cada línea. Finalmente, fuera del entorno "tabular", usamos "caption" para agregar un título a la tabla, y "label" para poder hacer referencia a la tabla en el texto utilizando "ref".

(Tabla LT)

## 4.4 Títulos

Para crear títulos y subtítulos en LaTeX, se puede usar los comandos "section", "subsection" y "subsubsection".

1. "section" se utiliza para crear un título principal.
2. "subsection" se utiliza para crear un subtítulo dentro

```

26 \begin{table}[htbp]
27 \begin{center} % Center the table horizontally
28 \begin{tabular}{|c|c|c|c|}
29 \hline
30 Nombre & correo electrónico & Foto & GitHub \\
31 \hline
32 Louise Berard & louise.berard28@gmail.com & Foto & GitHub \\
33 \hline
34 blabla&blabla&blabla&blabla \\
35 \hline
36 blabla&blabla&blabla&blabla \\
37 \hline
38 \end{tabular}
39 \end{center}
40 \end{table}

```

Figure 25: Tabla LT

```

21 \section{Título Principal}
22 \subsection{Subtítulo}
23 \subsubsection{Sub-subtítulo}
24

```

Figure 26: Títulos1 LT

de una sección. 3. “subsubsection” se utiliza para crear un subtítulo dentro de un subtítulo. Por ejemplo (Títulos1 LT):

Que se puede observar de la siguiente manera (Títulos2 LT):

## 4.5 Hyperlinks

Al agregar el paquete “hyperref” antes de “begin{document}” con “usepackage{hyperref}”, se puede usar tanto “url” como “href”. 1. “url” se utiliza para mostrar URLs sin convertirlas en enlaces activos. 2. “href” se utiliza para crear enlaces activos a URLs o direcciones de correo electrónico, permitiendo personalizar el texto del enlace mostrado.

```

2. Título Principal
2.1. Subtítulo
2.1.1. Sub-subtítulo

```

Figure 27: Títulos2 LT

```

64 \begin{figure}
65   \centering
66   \includegraphics[width=0.75\linewidth]{PDP.jpg}
67   \caption{Enter Caption}
68   \label{fig:enter-label}
69 \end{figure}

```

Figure 28: Imagen LT

## 4.6 Negrita

El comando “textbf” encierra el texto que deseas en negrita. Cuando compilas el documento LaTeX, se verá que el texto entre los corchetes aparecerá en negrita en el PDF resultante.

## 4.7 Imagen

Para agregar una imagen a un documento LaTeX, primero se necesita asegurar de tener la imagen en un formato compatible, como JPEG, PNG o PDF. Luego, dentro del documento LaTeX, se debe cargar el paquete graphicx en el preámbulo con el comando “usepackagegraphicx”. Este paquete permite la inclusión de gráficos en tu documento.

Después de cargar el paquete graphicx, se puede agregar la imagen utilizando el comando “includegraphics”.

Dentro de los corchetes, se debe especificar la ruta de la imagen. Si la imagen está en la misma carpeta que tu archivo LaTeX, solo se necesita especificar el nombre del archivo. Por ejemplo (Imagen LT):

1. “[width=0.75linewidth]” se utiliza para especificar el ancho de la imagen en relación con el ancho de la línea de texto en el documento. En este caso, “0.75linewidth” significa que la imagen ocupará el 75% del ancho de la línea de texto.
2. “captionEnter Caption”: con este comando se agrega una leyenda o título a tu figura o tabla en tu documento LaTeX.
3. “labelfig:enter-label”: este comando asigna una etiqueta a tu figura o tabla para que puedas referenciarla fácilmente en otras partes del documento.

## 4.8 Párrafos

Para separar en párrafos se usan las 2 barras inversas invertidas.

## 5 GitHub

Para utilizar GitHub, seguimos estos pasos:

- 1) Creamos una cuenta: Vamos al sitio web de GitHub (github.com) y hacemos clic en “Sign up” para crear una cuenta. Proporcionamos nuestra dirección de correo electrónico, nombre de usuario y contraseña, y seguimos los pasos para completar el proceso de registro.
- 2) Confirmamos nuestra dirección de correo electrónico: Después de registrarnos, recibiremos un correo electrónico de GitHub para confirmar nuestra dirección de correo electrónico. Seguimos las instrucciones en el correo electrónico para confirmar nuestra cuenta.
- 3) Exploramos la interfaz de usuario: Una vez que hayamos iniciado sesión en nuestra cuenta de GitHub, nos tomamos un tiempo para explorar la interfaz de usuario. Nos familiarizamos con las pestañas de perfil, repositorios, notificaciones y configuración.
- 4) Creamos un nuevo repositorio: Para comenzar a trabajar en un proyecto, podemos crear un nuevo repositorio. Hacemos clic en el botón “New” en la página principal de GitHub, proporcionamos un nombre para nuestro repositorio, una descripción opcional y elegimos si el repositorio será público o privado.

- 5) Clonamos un repositorio existente (opcional): Si estamos interesado en contribuir a un proyecto existente en GitHub, podemos clonar el repositorio a nuestra computadora local. Para hacer esto, navegamos al repositorio en GitHub, hacemos clic en el botón "Code" y copiamos la URL del repositorio. Luego, en nuestra terminal o línea de comandos, utilizamos el comando `git clone` seguido de la URL del repositorio.
- 6) Aprendemos a utilizar Git: GitHub utiliza Git, un sistema de control de versiones, para gestionar los cambios en los repositorios. Aprendemos los conceptos básicos de Git, como `commit`, `push`, `pull` y `merge`, para colaborar de manera efectiva en proyectos en GitHub.
- 7) Contribuimos a proyectos existentes (opcional): Exploramos proyectos en GitHub que nos interesen y consideramos contribuir con código, reportes de errores (issues), o mejoras (pull requests). Aprendemos sobre los procesos de contribución específicos de cada proyecto.
- 8) Utilizamos las funciones adicionales: GitHub ofrece muchas otras funciones útiles, como la gestión de proyectos, integración continua, acciones, wikis, y más. Exploramos estas funciones según nuestras necesidades y objetivos.

Con estos pasos, estaremos listos para comenzar a utilizar GitHub para colaborar en proyectos, compartir código, y contribuir a la comunidad de desarrollo de software.

## 6 Template TyHM

La Plantilla de TyHM es una herramienta esencial para la presentación de trabajos en la cátedra, basada en el estilo arXiv de la Universidad de Cornell y compatible con r-markdown template arXiv. Para utilizarla correctamente, se deben seguir los siguientes pasos:

1. **Descarga de la Plantilla:** La plantilla se descarga automáticamente con el nombre `Template_TyHM_2024`. Se busca el archivo en la carpeta de descargas y se procede al siguiente paso.
2. **Descompresión de la Plantilla:** Se hace clic derecho sobre el archivo descargado y se selecciona la opción para descomprimir. Esto creará una carpeta con todos los archivos necesarios para trabajar.
3. **Preparación en R Studio (Posit Cloud):**
  - Se abre [R Studio en Posit Cloud], que te permite trabajar directamente desde el navegador sin necesidad de instalaciones complicadas.
  - Se crea un nuevo espacio de trabajo y se sube la carpeta de la plantilla descomprimida.
4. **Trabajar con el Archivo `ArXiv_con_codigo.Rdm`:** Dentro de R Studio, se localiza el archivo `ArXiv_con_codigo.Rdm` en la carpeta de la plantilla que se subió. Este archivo es el que se utilizará para redactar y compilar los trabajos.
5. **Edición y Compilación:**
  - Se abre el archivo `ArXiv_con_codigo.Rdm` para comenzar a editar el documento.
  - Se inserta el contenido académico, código y cualquier otro elemento necesario para el trabajo.
  - Se utiliza las funciones de r-markdown para formatear el documento según las normas de la cátedra y el estilo arXiv.
  - Se compila el documento para generar una versión en PDF o el formato que requiera la cátedra.
6. **Entrega del Trabajo:** Una vez que se haya terminado de editar y de compilar el trabajo, se guarda y sube a la plataforma que la cátedra designó para la entrega de trabajos. Esta plataforma es GitHub, que fue explicado en el apartado anterior), en la cual se arma un repositorio por módulo, y en cada uno se suben los archivos correspondientes. Luego se sube el link correspondiente al módulo específico de GitHub al aula abierta de la cátedra.