

# Taller de $\text{\LaTeX}$

ComCom

Departamento de Computación  
FCEyN UBA

Segundo Cuatrimestre 2025



¿Qué *no* es  $\text{\LaTeX}$ ?

- Un procesador de documentos o texto.
- Un lenguaje de programación (solamente).

## ¿Qué es $\text{\LaTeX}$ ?

- Es una herramienta o sistema de preparación de documentos.
- Se usa comunmente en ambientes académicos o profesionales.
- La idea es escribir código que después se compila, y produce nuestro documento.



**Figura:** Dibujo del león mascota de CTAN por Duane Bibby.

## ¿Por qué querría usar $\text{\LaTeX}$ ?

- Permite formatear fórmulas matemáticas complejas con facilidad.
- Permite crear referencias y mantenerlas coherentes a través del texto.
- Extremadamente customizable y orientado a la comunidad.
- Es de uso libre y gratuito (sin que se entrenen IAs con su uso).

# Introducción

Y lo mas importante. . .

[SUBMISSION](#)[ATTENDANCE](#)[PROGRAM](#)[SPONSORS & EXPO](#)[ORGANIZERS](#)

## Submission Guidelines

[TIMELINE](#)[CALL FOR PAPERS](#)[AUTHOR  
GUIDELINES](#)[Policies](#)[Submission Guidelines](#)[Rebuttal Instructions](#)[Author FAQs](#)

All submissions will be handled electronically via the conference's CMT Website. By submitting a paper, the authors agree to the policies stipulated in this website. The paper submission deadline is March 8, 2023. Supplementary material can be submitted until March 15, 2023.

Papers are limited to eight pages, including figures and tables, in the ICCV style. Additional pages containing only cited references are allowed. Appendices may not be a part of the additional pages. Appendices are counted towards the eight-page limit or should be moved to the supplementary document. Please refer to the following files for detailed formatting instructions:

- Example submission paper with detailed instructions [PDE](#)
- LaTeX Templates (zip): [iccv2023AuthorKit.zip](#)

Papers that are not properly anonymized, or do not use the template, or have more than eight pages (excluding references) will be considered for rejection without review.

**Figura:** En el proceso de presentación de un paper, muchas veces es obligatorio entregarlo con un formato específico. Para asegurarlo, se provee un template de  $\text{\LaTeX}$ .

En la facultad, se usa  $\text{\LaTeX}$  para:

- Los exámenes y guías prácticas de las materias.
- Las diapositivas usadas en las clases.
- Armar informes, TPs, presentaciones.
- Escribir apuntes y notas.
- Diapositivas para alguna presentación.
- ...

## ¿Cómo usamos $\text{\LaTeX}$ ?

Hay dos maneras principales de usar  $\text{\LaTeX}$ .

- En el navegador (Overleaf).
  - **Pros:** Es extremadamente fácil de empezar a usar y colaborar en equipos.
  - **Cons:** Tiene una versión gratis, pero con el paso del tiempo fueron sacándole features y haciéndolas pagas. Tiene una configuración limitada.
- Instalación local (MiKTeX, TeX Live).
  - **Pros:** Se puede usar con los editores de texto o entornos de preferencia, sincronizar con Git y configurar a medida.
  - **Cons:** Puede ser difícil de instalar y troubleshootear.

# Comenzando

## ¿Cómo funciona?

- Se escribe tu documento en texto plano con comandos que describen su estructura y significado.
- El programa de latex procesa tu texto y comandos y produce un documento formateado.

El universo (que otros llaman la `\emph{Biblioteca}`) `\`  
se compone de un número `\textbf{indefinido}`, `\`  
y tal vez infinito, de galerías `\underline{hexagonales}`.

El universo (que otros llaman la *Biblioteca*)  
se compone de un número **indefinido**,  
y tal vez infinito, de galerías hexagonales.



# Comenzando

## Otros ejemplos

```
\begin{itemize}
\item Mate
\item Café
\item Harina
\item Palmitos
\end{itemize}
```

- Mate
- Café
- Harina
- Palmitos

```
\begin{equation}
\alpha + \beta + 1
\end{equation}
```

$$\alpha + \beta + 1 \quad (1)$$

# Comenzando

## Cambiando la actitud

- Usar  $\text{\LaTeX}$  nos provee una separación entre el **contenido** del documento y el **estilo**. Se usan comandos para describir el 'qué', en vez de 'como se ve' (nos enfocamos en el contenido y  $\text{\LaTeX}$  se encarga del resto).
- Como contraejemplo, un editor de texto plano, o un programa Word pertenecen a una forma de mostrar información llamada "What You See Is Whay You Get" (WYSIWYG).

# Comenzando

## Cambiando la actitud

- Una vez escrito el contenido del documento, el estilo se puede cambiar con facilidad. No hay necesidad de reformatear las cosas a mano.
- Se puede también escribir un archivo de estilo propio (o descargarlo de internet), y aplicarlo a los documentos que quisiéramos.

# Comenzando

Hola mundo!

Como es habitual con muchos lenguajes, veamos cómo hacer nuestro "Hola mundo!":

hola\_mundo.tex

```
1 \documentclass{article}
2
3 \begin{document}
4
5 Hola mundo! %esto es un
   comentario
6
7 \end{document}
```

hola\_mundo.pdf

Hola mundo!

# Comenzando

Hola mundo!

Como es habitual con muchos lenguajes, veamos cómo hacer nuestro "Hola mundo!":

hola\_mundo.tex

```
1 \documentclass{article}
2
3 \begin{document}
4
5 Hola mundo! %esto es un
   comentario
6
7 \end{document}
```

hola\_mundo.pdf

Hola mundo!

Tal vez parece mucho texto de  $\text{\LaTeX}$  para que el resultado sea sólo una línea. Pero veamos cual es el rol de cada parte del documento.

# Comenzando

Hola mundo!

hola\_mundo.tex

```
1 \documentclass{article}
2
3 \begin{document}
4
5 Hola mundo! %esto es un
   comentario
6
7 \end{document}
```

- Los comandos comienzan con un `\`.
- Todo documento comienza con un *documentclass*. Define la estructura general del documento. Esto incluye la fuente, el tamaño de la letra, el tamaño del papel, entre otros.
- El símbolo de porcentaje empieza un *comentario* –  $\text{\LaTeX}$  va a ignorar el resto de la línea. `%` y `\` son caracteres especiales.

# Formateando texto

- Todo el cuerpo del documento va dentro del `\begin{document}` y `\end{document}`.
- Casi siempre se puede escribir texto normalmente.

Las palabras se separan  
por uno o más espacios.

Los párrafos por una o  
mas líneas en blanco.

Las palabras se separan por  
uno o más espacios.

Los párrafos por una o mas  
líneas en blanco.

- Los espacios de más se colapsan en el resultado final.

El        universo        se compone  
de un                    número(`\ldots`)

El universo se compone de un  
número(...)

- Hay algunos caracteres con significados especiales en  $\text{\LaTeX}$ . Si los escribís, te va a dar un error. Tenés que escaparlos con un `\`.

`\$ \% \& \# !`

`$ \% \& \# !`

# Ejercicio

Escribamos en  $\text{\LaTeX}$  este mismo texto.

¡Llegaron los resultados de la carrera!

1. En primer lugar, y recibiendo un premio de \$0 llega Alice.
2. En segundo lugar, a nada de ser ganar el lugar #1, llega Bob.
3. Por último, pero incrementando su participación en carreras desde la última temporada en un 120 %, llega Charlie.
4. ...

Hacer click para abrir un template en Overleaf



- Veamos por qué el \$ es un caracter especial.

```
% :(
```

```
Sean a y b enteros positivos.
```

```
Sea  $c = a - b + 1$ .
```

```
% :)
```

```
Sean  $a$  y  $b$  enteros positivos.
```

```
Sea  $c = a - b + 1$ .
```

Sean  $a$  y  $b$  enteros positivos.

Sea  $c = a - b + 1$ .

Sean  $a$  y  $b$  enteros positivos.

Sea  $c = a - b + 1$ .

- Se usan siempre de a pares - uno para comenzar el modo matemático y otro para salir.
- $\LaTeX$  maneja los espacios automáticamente.

```
Sea  $y=mx+b$  \ldots
```

```
Sea  $y = m x + b$  \ldots
```

Sea  $y = mx + b \dots$

Sea  $y = mx + b \dots$

- Si tu ecuación es muy importante, podés mostrarla dentro del entorno *equation*.

Las raices de una ecuación cuadrática son

```
\begin{equation}
```

```
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
```

```
\end{equation}
```

donde  $a$ ,  $b$  y  $c$  son  $\ldots$

Las raices de una ecuación cuadrática son

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2)$$

donde  $a$ ,  $b$  y  $c$  son  $\ldots$

- Cuidado:  $\text{\LaTeX}$  generalmente ignora los espacios en matemática, pero no soporta inserción de líneas en blanco.

# Ejemplos

## Armando un TP

¿Qué podríamos querer hacer?

segundoTP.tex

```
1 \documentclass{article}
2 \title{Tremendo TP}
3 \author{Nuestro grupo}
4 \date{23:55 (nos sobraron 5)}
5
6 \begin{document}
7 \maketitle
8
9 En este trabajo práctico nos proponemos resolver ...
10
11 \end{document}
```

# Ejemplos

Armando un TP

segundoTP.pdf

Tremendo TP

Nuestro grupo

23:55 (nos sobraron 5)

En este trabajo práctico nos proponemos resolver ...

A su vez, se puede incluir un subtítulo, omitir algunos de los campos, o también usar funciones como `\today` para que la fecha se actualice automáticamente con cada compilación.

# Ejemplos

## Presentaciones

¿Y si queremos armar slides para una presentación?

```
1 \documentclass{beamer}
2 \begin{document}
3
4 \begin{frame}[containsverbatim]
5 \frametitle{Ejemplos}
6 \framesubtitle{Presentaciones}
7
8 ¿Y si queremos armar slides para una presentación?
9
10 ...
11
12 \end{frame}
13
14 \end{document}
```

# Ejemplos

## Paquetes

En muchas ocasiones, vamos a querer funcionalidades, entornos, estilos, fuentes, etc. que para utilizarse requieren que usemos el **paquete** adecuado para ello.

Por ejemplo, nos gustaría poder incluir hipervínculos en nuestro documento, de forma que en el contenido del mismo utilice enlaces para acceder a otros documentos o sitios web.

# Ejemplos

## Paquetes

Por defecto, tenemos la posibilidad de indicar que algo es una URL.

Por ejemplo:

(...) para más información, consultar la web del departamento de  
computación <https://www.dc.uba.ar>.

# Ejemplos

## Paquetes

Por defecto, tenemos la posibilidad de indicar que algo es una URL.

Por ejemplo:

(...) para más información, consultar la web del departamento de computación <https://www.dc.uba.ar>.

Mientras que utilizando el paquete `hyperref` podemos escribir lo siguiente

(...) para más información, consultar la web del [departamento de computación](#).

Lo cual resulta más prolijo para enlazar URLs, secciones de un informe, referencias, etc.



# Código

Un escenario común, es el querer incluir código en nuestro documento con el **entorno** *lstlisting* y el **paquete** *listings*. Por ejemplo:

```
\begin{lstlisting}[language=c]
int sumar(int x, int y) {
    return x + y;
}
\end{lstlisting}
```

```
1 int sumar(int x, int y) {
2     return x + y;
3 }
```

# Ejercicio

Escribamos en  $\text{\LaTeX}$  este mismo texto.

(...) teniendo en cuenta esto, podemos definir el **teorema de Bayes** como:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

para utilizar esto, queremos primero implementar la función `probaCondicional` de la siguiente manera

```
1 def probaCondicional(A, B):  
2     numerador    = proba(interseccion(A, B))  
3     denominador = proba(B)  
4     return division(numerador, denominador)
```

# Ejercicio

A su vez, quizás queremos incluir gráficos en algún informe o presentación. Esto lo podemos hacer con el **entorno** `figure` y el **comando** `\includegraphics`. De esta forma:

```
1 \begin{figure}[h]
2   \centering
3   \includegraphics
4     {../images/lion.png}
5   \caption{Titulo.}
6 \end{figure}
```



**Figura:** Titulo.

# Ejercicio

## Pseudocódigo

Por último, queremos poder tomar un paso atrás y escribir en abstracto un pseudocódigo, más allá del lenguaje en el que lo implementemos.

Para ello, podemos utilizar el **paquete** `algpseudocode` con el **entorno** `algorithmic`. Y dentro del mismo, incorporamos el pseudocódigo de nuestro programa (noten el uso de mayúsculas para cada comando).

# Ejercicio

## Pseudocódigo

Por último, queremos poder tomar un paso atrás y escribir en abstracto un pseudocódigo, más allá del lenguaje en el que lo implementemos.

Para ello, podemos utilizar el paquete `algpseudocode` con el entorno `algorithmic`. Y dentro del mismo, incorporamos el pseudocódigo de nuestro programa (noten el uso de mayúsculas para cada comando).

Un bloque *function*:

```
\begin{algorithmic}
\Function{nombre}{ $p_1, p_2, \dots$ }
  \State  $\textit{suma} \leftarrow 2$ 
\EndFunction
\end{algorithmic}
```

# Ejercicio

## Pseudocódigo

### Un condicional *if-else*:

```
\If{a}  
  \State b  
\Else  
  \State c  
\EndIf
```

### Ciclo *for*:

```
\For{ $i \in 1, \dots, n$ }  
  \State  $a \leftarrow a + i$   
\EndFor
```

### Retorno:

```
\State \Return res
```

# Ejercicio

## Pseudocódigo

Con ello, escribamos el siguiente pseudocódigo:

```
function SUMARTUPLAS(tuplas)  
  suma  $\leftarrow$  (0, 0)  
  if tupla  $\neq$  {} then  
    for  $i = 1, \dots, |tuplas|$  do  
       $suma_1 \leftarrow tupla[i]_1$   
       $suma_2 \leftarrow tupla[i]_2$   
    end for  
  end if  
  return suma  
end function
```

Fijense que algunos símbolos necesitan ser escapados o escribirse en modo matemático

Hay distintas fuentes que podemos consultar para saber más de  $\text{\LaTeX}$  o sobre paquetes particulares. Algunas de ellas son:

- La documentación de [Overleaf](#).
- [CTAN](#), un repositorio de conocimiento sobre múltiples paquetes (por ejemplo aquellos usados en esta clase).
- $\text{\LaTeX}$  [Stack Exchange](#), que sirve como foro para dudas diversas.
- ...



# ¿Preguntas?

*i i i i i i i i i i ? ? ? ? ? ? ? ?*