

# Guión presentación de Typst

## Introducción

¡Bienvenidos a mi presentación sobre Typst! ¿Alguna vez has trabajado con LaTeX y te has sentido frustrado por su complejidad y lentitud? Si es así, entonces Typst podría ser la solución que estás buscando. Typst es un lenguaje de tipografía de código abierto que ofrece una sintaxis sencilla y moderna para escribir documentos de alta calidad tipográfica.

Desde su inicio en 2019, el equipo detrás de Typst ha estado comprometido con la colaboración abierta y la comunidad de código abierto. El lenguaje fue creado por un grupo de desarrolladores que estaban insatisfechos con LaTeX y buscaban una alternativa más rápida y fácil de usar. En marzo del 2023 se abrió la beta abierta y desde entonces ha crecido hasta llegar a más de 17.000 estrellas en GitHub y una comunidad de Discord con más de 2.700 miembros.

En esta presentación, exploraremos cómo Typst se compara con otros lenguajes de marcado similares, y las características y ventajas que ofrece. También veremos herramientas utilizadas para crear archivos de Typst, como por ejemplo VSCode o su App Web, para crear presentaciones, informes, artículos y más. Cabe mencionar que su aplicación web nos permite trabajar de forma colaborativa en tiempo real.

En resumen, Typst es una opción excelente para aquellos que buscan una manera sencilla y moderna de escribir documentos con formato profesional, y que valoran la colaboración abierta y la comunidad de código abierto. ¡Así que vamos a adentrarnos en el mundo de Typst y ver cómo puede mejorar nuestra forma de escribir documentos!

## Problemáticas que soluciona Typst

- **Compilación rápida a PDF:** Una de las problemáticas que resuelve Typst es la compilación lenta que se puede experimentar al utilizar LaTeX. La compilación en Typst es mucho más rápida, lo que significa que los usuarios pueden ahorrar tiempo y aumentar su productividad. Además, Typst tiene una función de previsualización en tiempo real, lo que permite a los usuarios ver cómo se verá el documento final mientras lo están editando, sin tener que compilar todo el documento cada vez que realizan un cambio.
- **Trabajo colaborativo:** Typst permite el trabajo colaborativo en tiempo real. Los usuarios pueden trabajar en el mismo documento simultáneamente, lo que facilita la colaboración en equipo. Esto es particularmente útil para proyectos en los que varios autores necesitan trabajar juntos en un documento, ya sea para una presentación, un informe o un artículo.
- **Creación sencilla de macros y templates:** Typst tiene una sintaxis clara y sencilla, lo que facilita la creación de macros y templates personalizados. Esto significa que los usuarios pueden personalizar sus documentos de acuerdo a sus necesidades y preferencias, lo que les permite ahorrar tiempo y aumentar su eficiencia. Además, los usuarios pueden compartir sus macros y templates con la comunidad, lo que fomenta la colaboración y el intercambio de conocimientos.
- **Más intuitivo de aprender que LaTeX:** Typst es más fácil de aprender que LaTeX, ya que su sintaxis es más sencilla y moderna. Esto significa que los usuarios pueden comenzar a trabajar en documentos rápidamente, sin tener que invertir mucho tiempo en aprender la sintaxis y la estructura de LaTeX. Además, Typst tiene una documentación clara y completa, lo que facilita la comprensión de su funcionamiento y características.

## Ventajas en comparación a

LaTeX

- **Previsualización en tiempo real:** Typst tiene una función de previsualización en tiempo real, lo que permite a los usuarios ver cómo se verá el documento final mientras lo están editando, sin tener que compilar todo el documento cada vez que realizan un cambio.
- **Mejores mensajes de errores:** Typst tiene mensajes de error más claros y fáciles de entender que LaTeX, lo que facilita la solución de problemas y la depuración de código.
- **Creación de macros más amigables para programadores:** Typst tiene una sintaxis más sencilla y moderna que LaTeX, lo que facilita la creación de macros y templates personalizados. Esto significa que los usuarios pueden personalizar sus documentos de acuerdo a sus necesidades y preferencias, lo que les permite ahorrar tiempo y aumentar su eficiencia. Además, los usuarios pueden compartir sus macros y templates con la comunidad, lo que fomenta la colaboración y el intercambio de conocimientos.

## Word

- **Fácil modo colaborativo:** Typst permite el trabajo colaborativo en tiempo real. Los usuarios pueden trabajar en el mismo documento simultáneamente, lo que facilita la colaboración en equipo. Esto es particularmente útil para proyectos en los que varios autores necesitan trabajar juntos en un documento, ya sea para una presentación, un informe o un artículo.
- **Puedes cambiar el formato de tu documento sin tener que hacerlo manualmente en todo el documento:** por ejemplo, si decides que quieres cambiar el tipo de letra o tamaño de fuente en todo el documento, en lugar de tener que ir página por página y cambiar el formato manualmente, puedes simplemente cambiar la configuración de formato en el archivo de Typst y automáticamente actualizará en todo el documento.

Esta funcionalidad es muy útil cuando se trabaja con documentos largos o complejos, ya que permite ahorrar tiempo y esfuerzo en el proceso de edición y formateo del documento. En contraste, con Word a menudo es necesario ir página por página y hacer cambios manuales de formato, lo que puede ser tedioso y propenso a errores.

## Google Docs:

- **Documentos más profesionales**
- **Mayor cantidad de opciones de formato:** como por ejemplo el modo matemático, que permite escribir ecuaciones matemáticas de forma sencilla y rápida.

## Open Source

Ser open source significa que el código fuente de Typst está disponible públicamente y puede ser modificado, mejorado y distribuido libremente por cualquiera que tenga la habilidad técnica y los recursos necesarios. Esta característica es una gran ventaja para Typst por varias razones:

- **Mejora constante:** Al ser open source, Typst se beneficia de la colaboración abierta de la comunidad de desarrolladores y usuarios que pueden sugerir mejoras, solucionar problemas y hacer contribuciones al proyecto. Esto permite una mejora constante del lenguaje y su adaptación a las necesidades de la comunidad.
- **Flexibilidad:** Al ser open source, los usuarios tienen la libertad de modificar el código fuente de Typst para adaptarlo a sus necesidades específicas. Esto significa que los usuarios pueden crear versiones personalizadas de Typst con funcionalidades y características que no están disponibles en la versión estándar.
- **Transparencia:** Al ser open source, los usuarios pueden examinar el código fuente de Typst y entender cómo funciona internamente. Esto aumenta la confianza en la seguridad y privacidad del lenguaje.

- Gratuito: Al ser open source, Typst es gratuito y de libre distribución. Esto hace que sea accesible para cualquier persona que quiera utilizarlo sin costos adicionales.

En resumen, la naturaleza open source de Typst permite que la comunidad de desarrolladores y usuarios trabajen juntos para mejorar el lenguaje, crear versiones personalizadas y aumentar la transparencia y accesibilidad del lenguaje.

## Cómo ocupar Typst

Hay dos opciones, la primera es descargar el programa y la segunda es ocuparlo en línea.

### Descargar el CLI

Se puede descargar el binario de Typst desde la página oficial de Typst. El binario está disponible para Windows, Mac y Linux. Una vez descargado, se puede instalar el programa en el sistema operativo correspondiente.

### Comandos

Una vez instalado, se puede utilizar Typst desde la terminal con los siguientes comandos:

- `typst compile <nombre>`: compila el archivo de Typst con el nombre especificado.
- `typst watch <nombre>`: compila el archivo de Typst con el nombre especificado y lo actualiza automáticamente cada vez que se guardan cambios en el archivo.

Yo utilizo el comando `typst watch <nombre>` junto con `zathura` para ver el documento en tiempo real.

### Ocuparlo en línea

Se puede ocupar Typst en línea desde la página oficial de [Typst](#). Para ello, es necesario crear una cuenta de usuario y luego iniciar sesión en la página web. Una vez iniciada la sesión, se puede comenzar a utilizar Typst en línea.

## Sintaxis y modos de Typst

Typst se basa en tres modos principales: el **modo markup** o bloques de contenido, el **modo código** o de funciones y el **modo matemático**.

El de markup es el modo por defecto, y se utiliza para escribir texto normal, con algunas características que enriquecen el texto al estilo de Markdown. Y el modo código se utiliza para escribir funciones y macros, que permiten modificar bloques de contenido y darle formato al documento.

**= Este es un título**

Hola, este es un párrafo normal.

- item 1
- item 2
  - + subitem 1
  - + subitem 2

**== Subtítulo**

`#lorem(10)`

**Este es un título**

Hola, este es un párrafo normal.

- item 1
- item 2
  1. subitem 1
  2. subitem 2

**Subtítulo**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Por defecto se utiliza el modo markup.

La función `#lorem(n)`: genera un párrafo de texto lorem ipsum de n palabras.

Para desambiguar que queremos entrar al modo markup se embuelve en `[]`.

## Modo matemático

Para entrar en modo matemático se utiliza \$ al igual que en Markdown o LaTeX.

### = Ecuación

La mitad de \$1\$ se escribe \$1 / 2\$.

Para crear bloques de ecuaciones se hace con un espacio en blanco antes y después de la ecuación.

### = Ecuación

La mitad de \$1\$ se escribe:

\$ 1 / 2 \$

## Ecuación

La mitad de 1 se escribe  $\frac{1}{2}$ .

## Ecuación

La mitad de 1 se escribe:

$$\frac{1}{2}$$

## Sintaxis de símbolos intuitiva

La sintaxis de Typst es intuitiva y fácil de aprender. Esto significa que los usuarios pueden escribir ecuaciones de forma rápida y eficiente, lo que aumenta su productividad y eficiencia.

### = Ecuación

```
$ A = pi r^2 $  
$ "area" = pi dot "radius"^2 $  
$ cal(A) :=  
  { x in RR | x "is natural" } $  
#let x = 5  
$ #x < 17 $
```

## Ecuación

$$A = \pi r^2$$

$$\text{area} = \pi \cdot \text{radius}^2$$

$$\mathcal{A} := \{x \in \mathbb{R} \mid x \text{ is natural}\}$$

$$5 < 17$$

## Alinear ecuaciones

Para alinear ecuaciones se utiliza & para indicar el punto de alineación. Y se dan saltos de línea con \.

```
$ sum_(k=0)^n k  
  &= 1 + ... + n \  
  &= (n(n+1)) / 2 $
```

$$\sum_{k=0}^n k = 1 + \dots + n$$
$$= \frac{n(n+1)}{2}$$

## Modo código

Vamos a ver algunas funciones y macros que permiten darle formato al documento. De todos modos hay una lista más extensa de funcionalidades.

Normalmente sigue el patrón:

```
#función(argumentos)[  
  contenido  
]
```

El contenido es un bloque de contenido, y por ende puede tener más funciones y macros dentro.

El # se utiliza para desambiguar el modo código del modo markup, cuando ya se está en el modo código y se quiere ocupar de nuevo una función no es necesario volverla a ocupar.

## Cambiar el formato de un párrafo

```
#par(justify: true)[  
  = Título  
  #lorem(40).  
]
```

## Título

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut

labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequi doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere..

### Cambiar el formato utilizando #set

Esto sólo cambia el formato de todo el bloque de contenido que se le pasa como argumento. Para hacer que este cambio sea permanente se debe utilizar la macro #set.

```
#set par(justify: true)
```

```
= Título  
#lorem(40).
```

### Título

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequi doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere..

### Función #show

Esta función permite reemplazar el contenido de un bloque de contenido por otro.

```
#show "Ping": "Pong"
```

Pong

Ping

Podemos recuperar el contenido original con la siguiente sintaxis:

```
#show "Hola": name => [  
  #name...  
]
```

Hola...

Hola

También se puede ocupar para afectar a todos los bloques de contenido del documento.

```
#show: rest => text(  
  [#rest...]  
)
```

Hola que tal  
Nooooo  
Bla bla bla

Hola que tal\  
Bla bla bla

### Modo sexo

...