Guía práctica de sintaxis Typst

Referencia rápida para fichas, tablas, imágenes y maquetación

SOLO CON LINUX - Documentación Técnica
Fecha: 23/11/2026
Versión documento: 1.3

Guía práctica de sintaxis Typst

Referencia rápida para fichas, tablas, imágenes y maquetación

Autor	`SOLO CON LINUX`
Fecha	`23/11/2026`
Versión Documento	`1.3`

1 Formato básico

En Typst el marcado es directo y se combina con funciones precedidas por `#`:

- **Cursiva** con guiones bajos: `_texto en cursiva_` se obtiene: *texto en cursiva*
- **Negrita** con asteriscos simples: `*texto en negrita*` se obtiene: **texto en negrita**
- **Subrayado** con la función `#underline[texto subrayado]` se obtiene: texto subrayado
- **Tachado** con la función `#strike[texto tachado]` se obtines:~~texto tachado~~
- **Espaciado** se puede forzar un espaciado de la medida deseada entre un elemento y el siguiente con la función `#v()`:

```
... Texto A ...
#v(0.4cm) // espaciado a dejar antes del siguiente elemento
... Texto B ...
```

… Texto A …

… Texto B …

- **Forzar salto de línea** se puede realizar mediante el caracter `\`:

```
Esto es una línea... \ y esto es otra
```

Esto es una línea…
y esto es otra

- **Enlaces externos** con `#link("url")[texto visible]` o escribiendo la URL directa: `#link("https://typst.app/docs")[Documentación oficial de Typst]`.
- **Comentarios** se usan dobles barras:

```
\\ Comentario
\\ Se ignora por Typst
```

- **Símbolos matemáticos** se escriben entre signos de dólar `$ ... $`:
  - **En línea**: la siguiente fórmula

    ```
    $E = m c^2$ o la raíz $sqrt(x^2 + y^2)$
    ```

    se vería en la misma línea como $E = mc^2$ o la raíz $\sqrt{x^2 + y^2}$.
  - **En bloque** (con espacios dentro de `$`) si escribimos:

    ```
    $ sum_(i=1)^n x_i = alpha + beta - infinity $.
    ```

    El resultado que se genera es:

$$\sum_{i=1}^{n} x_i = \alpha + \beta - \infty$$

1.1 Listas

1.1.1 Lista normal

Se utiliza el símbolo de guión `-` para crear la lista:

```
- Elemento
- Elemento
```

- Elemento
- Elemento

1.1.2 Lista numerada

Se escribe el número, seguido de un punto y el texto de la lista. Sse pueden saltar numeros a la hora de generar la lista numerada

```
1. Primer elemento
2. Segundo elemento
9. Noveno (permite saltar numeros)
```

1. Primer elemento
2. Segundo elemento
9. Noveno (permite saltar numeros)

1.1.3 Mezcla de Listas

Se pueden mezclar ambos tipos de listas, tan solo hay que *identar* el elemento de la lista:

```
// Numerados con varios elementos
1. Elemento uno
  - Elemento de uno
    - Sub-elemento
    - Sub-elemento
2. Elemento dos
  - Elemento de dos
```

1. Elemento uno
   - Elemento de uno:
     - Sub-elemento
     - Sub-elemento
2. Elemento dos
   - Elemento de dos

```
// Lista con elementos numerados
- Lista de elementos:
  1. Uno
  2. Dos
  3. Tres
```

- Lista de elementos:
  1. Uno
  2. Dos
  3. Tres

1.2 Líneas de separación

- **Líneas de separación normal**

Pondemos usar `#line()` para forzar una línea, pero además podemos personalizarla:

```
// Linea de separación (recomendable forzar siempre longitud a 100%)
#line(length: 100%,)
```

- **Línea de separación personalizada**

```
// Ajuste posición dejando 10% a cada lado (centrada)
#line(
  start: (10%, 0pt),
  end: (90%, 0pt),
  stroke: 3pt + luma(160),
)
```

- **Personalización con inclinación**

```
// Linea de separación personalizada con inclinación
#line(
  length: 16cm,
  angle: 2deg,
  stroke: 1pt + luma(140)
)
```

1.3 Lorem ipsum

Typst tiene la función `#lorem()` que permite generar texto de relleno, se suele usar para ver como va a quedar el documento que estás preparando:

```
#lorem(90)
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis.

1.4 Código y Bloques de código

- **Código en línea** con acento grave: `variable_x` (se escribe `variable_x`).
- **Bloque de Código** encerrado entre triple acento grave:

````
```
#let sumar(a, b) = a + b"
```
````

- **Bloque de Código** usando la función `#raw`:

```
#raw(
  lang: "typst",
  block: true,
  "#let sumar(a, b) = a + b",
)
```

En ambos casos el código final se veria así:

```
#let sumar(a, b) = a + b
```

Hay dos comportamientos que conviene vigilar al generar el PDF con bloques de código largos:

- *Líneas muy largas* (desbordamiento horizontal): Por defecto, un bloque de código (raw / triple acento grave) no corta automáticamente las líneas largas si no se le indica. Si una línea es muy ancha, puede salirse del margen derecho en el PDF en vez de envolver hacia abajo.
- Cortes de página en medio de un bloque: Si un bloque cruza de una página a otra, Typst lo divide entre páginas; si quieres evitar cortes feos o controlar cómo se ajusta, puedes configurar raw o envolverlo en un block(…).

1.4.1 Asegurar el código para que se vea bien en el PDF

Puedes **ajustar el estilo global** de los bloques de código `raw` al principio de tu documento con una regla `#show`:

```
// Hace que las líneas largas dentro de bloques de código hagan salto de línea
#show raw.where(block: true): set block(width: 100%, breakable: true)
#show raw.where(block: true): it => block(
  width: 100%,
  fill: luma(245),
  inset: 8pt,
  radius: 4pt,
  text(size: 8.5pt)[
    #set par(justify: false)
    #box(width: 100%)[
      #show raw.line: it => {
        // Permite ajuste de línea en líneas largas si lo combinas con `raw`
        it
      }
      #it
    ]
  ]
)
```

En la práctica, para la mayoría de casos en Typst, basta con esto para que los bloques de código largos sean legibles y no se salgan del ancho de página:

```
#set raw(lang: "bash") // O el lenguaje que corresponda

#show raw.where(block: true): it => block(
  width: 100%,
  fill: luma(245),
  inset: 8pt,
  radius: 4pt,
  breakable: true,
  text(size: 8.5pt, it),
)
```

- Con `breakable: true`: el bloque puede continuar en la siguiente página si es muy largo, en lugar de quedarse colgado o desbordar el pie de página.
- Con `width: 100% y text(size: 8.5pt, ...)`: evitas que líneas extensas se vean demasiado apretadas o se salgan del área útil de la página.
- Si una línea tiene cadenas muy largas sin espacios (como URLs o hashes), en Typst lo más limpio es acortarlas o partirlas manualmente en el código fuente, ya que la tipografía de código suele tratarse de forma literal.

Ajuste con borde

Podemos forzar a que en el documento el codigo quede diferenciado del texto normal.

```
#show raw.where(block: true): it => block(
  width: 100%,
  fill: luma(246),              // Gris muy suave para ahorrar tinta al imprimir
  stroke: 0.75pt + luma(195),   // Borde gris discreto
  radius: 5pt,                   // Esquinas redondeadas
  inset: (x: 11pt, y: 9pt),      // Espacio interior cómodo para leer
  breakable: true,               // Permite continuar en otra página si es largo
  {
    set text(
      fill: black,               // Texto siempre en negro puro para máxima nitidez
      size: 8.8pt,
    )
    it
  }
)
```

2 Tablas, anchos y alineación

En `table(...)`, el parámetro `columns` define cada columna:

- `auto`: se ajusta al contenido.
- `1fr` (o `2fr`): reparte el espacio restante de forma proporcional.
- Unidades fijas (`3cm`, `40mm`, `120pt`): ancho exacto.
- Porcentajes (`25%`): porcentaje del ancho disponible de la tabla.

Con `align` puedes controlar la alineación de celdas o columnas (`left`, `center`, `right`, `horizon` para centrar verticalmente):

Código	Descripción	Alineación	Ancho
`3cm`	Columna con medida fija en centímetros	Izquierda	3.0 cm
`1fr`	Ocupa el espacio libre restante	Izquierda	Flexible
`center`	Texto centrado horizontalmente	Centro	2.6 cm
`right`	Valores numéricos alineados a la derecha	Derecha	2.2 cm

La tabla anterior se ha creado con el siguiente código:

```
#table(
  columns: (2.8cm, 1fr, 2.6cm, 2.2cm),
  inset: 7pt,
  stroke: 0.6pt + luma(170),
  fill: (x, y) => if y == 0 { luma(230) },
  align: (left, left, center, right),
  [*Código*], [*Descripción*], [*Alineación*], [*Ancho*],
  [`3cm`], [Columna con medida fija en centímetros], [Izquierda], [3.0 cm],
  [`1fr`], [Ocupa el espacio libre restante], [Izquierda], [Flexible],
  [`center`], [Texto centrado horizontalmente], [Centro], [2.6 cm],
  [`right`], [Valores numéricos alineados a la derecha], [Derecha], [2.2 cm],
)
```

3 Escritura en Columnas

Podemos hacer que Typst nos gestione de forma automatica la escritura en columnas:

```
#columns(2, gutter: 14pt)[
  *Escribir a dos columnas*

  Este texto llenará primero la columna izquierda y, al llegar abajo,
  continuará automáticamente en la columna derecha.

  #colbreak() // Opcional: fuerza el salto manual a la siguiente columna

  Aquí empieza el contenido tras el salto de columna.
  #lorem(15)_
]
```

Escribir a dos columnas

Este texto llenará primero la columna izquierda y, al llegar abajo, continuará automáticamente en la columna derecha.

Aquí empieza el contenido tras el salto de columna:

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

4 Imágenes: tamaños y disposición múltiple

Para insertar una imagen se usa `image("ruta.png", ...)`.

Los parámetros más útiles para controlar el tamaño son:

- `width` (ancho) y `height` (alto): admiten `mm`, `cm`, `pt`, `%` o `fr`.
- `fit`:
  - `"cover"` (llena el marco recortando)
  - `"contain"` (encaja entera dentro del marco)
  - `"stretch"` (estira sin respetar proporción).

4.1 Imagen con tamaño definido

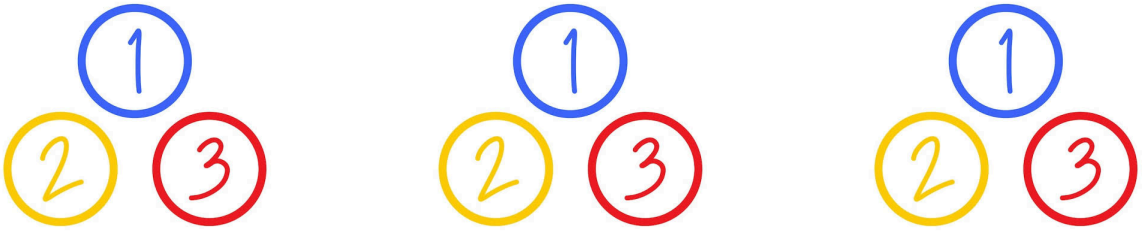
```
#image("ejemplo.png", width: 6cm, fit: "contain")
```



4.2 Varias imágenes juntas

Para colocar varias imágenes una al lado de otra con tamaños controlados, lo más limpio es usar `grid(...)` o `stack(dir: ltr, ...)`:

```
#grid(
  columns: (1fr, 1fr, 1fr),
  column-gutter: 8pt,
  image("img1.png", width: 80%, height: 3.5cm, fit: "cover"),
  image("img2.png", width: 80%, height: 3.5cm, fit: "cover"),
  image("img3.png", width: 80%, height: 3.5cm, fit: "cover"),
)
```



4.3 Imagen con texto al pie y Enlace a la Figura

Si se usa la función `#figure()` Se pueden incluir texto al pie de las imágenes, mediante `caption:`

Y además se puede usar `<nombre-enlace-foto>` para incluir un enlace a una foto determinada desde cualquier otra parte del documento:

```
#figure(image("ejemplo.png", width: 6cm), caption: [Imagen de ejemplo]) <fig-ejemplo>
```



Figura 1: Imagen de ejemplo

5 Enlaces internos en el documento

Puedes poner una etiqueta `<etiqueta>` detrás de cualquier encabezado, figura, tabla o bloque, y saltar a ella con cualquiera de estas opciones:

- `#link(<etiqueta>)[texto]`
- `#ref(<etiqueta>)`

5.1 Usando `link`

5.1.1 Crear enlace

Para crear un enlace se usa en cualquier elemento:

```
<nombre-enlace>`
```

5.1.2 Acceder al enlace

Para acceder a elemento se usa `#link()` Si quisieramos ir a la sección de tablas y columnas usariamos:

```
#link(<sec-tablas>)[tablas y columnas]
```

5.2 Usando `#ref`

Podemos crear una referencia automática a un enlace que se usará posteriormente.

5.2.1 Crear la referencia

Para crear una referencia sobre un enlace:

```
#ref(<sec-imagenes>)
```

5.3 Ir al la referencia creada

Usaremos `@sec-imagenes` para ir la sección Sección 4.

```
// Localiza y crea el nombre y numero de sección de forma automática
@sec-imagenes.
```

Podemos tambien usarlo para acceder a las Figuras, por ejemplo: ver Figura 1

```
// Localiza y crea el nombre de la figura automaticamente
@fig-ejemplo
```

6 Anotaciones y pies de página

6.1 Notas al pie (`#footnote`)

Typst gestiona automáticamente la numeración[1] y coloca el texto al pie de la página.

Este dato técnico incluye una aclaración adicional[2].

```
Typst gestiona automáticamente la numeración#footnote[Typs realiza autonumeración] y
coloca el texto al pie de la página.
Este dato técnico incluye una aclaración adicional#footnote[En el pie de la pagina
actual aparecerá todo el texto que quieras poner].
```

6.2 Anotaciones o cajas de aviso en el texto

Typst no tiene un comando único llamado «anotación», pero es habitual crear cajas de nota con `block(...)` o `rect(...)` para avisos o comentarios en fichas:

> **Nota:** Usa cajas con borde izquierdo o `callout` propio en tu plantilla para destacar advertencias, tolerancias o revisiones dentro de cada ficha.

La nota anterior se creó con un `bloque` y el siguiente código de `Typst`:

```
#block(
  width: 100%,
  inset: 10pt,
  radius: 4pt,
  fill: luma(245),
  stroke: (left: 3pt + navy),
)[
  *Nota:* Usa cajas con borde izquierdo o `callout` propio en tu plantilla para
destacar advertencias, tolerancias o revisiones dentro de cada ficha.
]
```

[1] Typs realiza autonumeración
[2] En el pie de la pagina actual aparecerá todo el texto que quieras poner

7 Portadas de página

7.1 Portada Estandard

El código a usar para generar la misma portada de este documento es:

```
#page(
  paper: "a4",
  margin: 2.5cm,
  numbering: none,
  header: none,
  footer: none,
)[
  #v(2cm)
  #text(size: 11pt, tracking: 2pt, fill: luma(110))[SOLO CON LINUX - DOCUMENTACIÓN
TÉCNICA]
  #v(0.4cm)
  #line(length: 100%, stroke: 1.5pt + navy)
  #v(0.8cm)
  #text(size: 26pt, weight: "bold", fill: navy)[Guía práctica de sintaxis Typst]
  #v(0.4cm)
  #text(size: 15pt, fill: luma(80))[Referencia rápida para fichas, tablas, imágenes y
maquetación]

  #v(1fr) // Empuja el bloque inferior hacia el fondo de la portada

  #table(
    columns: (5.0cm, 1fr),
    stroke: 0.5pt + luma(200),
    inset: 8pt,
    [*Autor*], [*`SOLO CON LINUX`*],
    [*Fecha*], [*`23/11/2026`*],
    [*Versión Documento*], [*`1.3`*],
  )
]

#pagebreak()
```

7.2 Portada con imagen de fondo

El código a usar para generar la misma portada de este documento es:

```
#page(
  paper: "a4",
  margin: 2cm,
  numbering: none,
  header: none,
  footer: none,
  background: image("fondo-portada.jpg", width: 100%, height: 100%, fit: "cover"),
)[
  #v(10cm)
  #block(
    width: 100%,
    inset: 20pt,
    radius: 6pt,
    fill: white.transparentize(12%),
    stroke: 1pt + luma(180),
  )[
    #text(size: 24pt, weight: "bold")[Guía práctica de sintaxis Typst]
    #v(0.5cm)
    #text(size: 15pt, fill: luma(80))[Referencia rápida para fichas, tablas, imágenes
y maquetación]
    #text(size: 12pt)[

      *`SOLO CON LINUX - Documentación Técnica`* \
      *Fecha: `23/11/2026`* \
      *Versión documento: `1.3`*
    ]
  ]
]

#pagebreak()
```

8 Configuración del Documento

Podemos definir como se quiere el formato de todo el documento definiendo el formato `page` (página).

En este documento el formato aplicado ha sido:

```
#set page(
  paper: "a4",
  margin: (top: 2.2cm, bottom: 2.2cm, left: 2cm, right: 2cm),
  numbering: "1 / 1",
  header: [
    #set text(size: 8.5pt, fill: luma(110))
    #grid(
      columns: (1fr, auto),
      [Mini guía de sintaxis Typst],
      [Referencia rápida],
    )
    #line(length: 100%, stroke: 0.5pt + luma(200))
  ],
  footer: context [
    #set text(size: 8.5pt, fill: luma(110))
    #line(length: 100%, stroke: 0.5pt + luma(200))
    #v(4pt)
    #grid(
      columns: (1fr, auto),
      [Documentación interna],
      [Página #counter(page).display("1 de 1")],
    )
  ],
)

#show raw.where(block: true): it => block(
  width: 100%,
  fill: luma(246),              // Gris muy suave para ahorrar tinta al imprimir
  stroke: 0.75pt + luma(195),   // Borde gris discreto
  radius: 5pt,                   // Esquinas redondeadas
  inset: (x: 11pt, y: 9pt),      // Espacio interior cómodo para leer
  breakable: true,               // Permite continuar en otra página si es largo
  {
    set text(
      fill: black,               // Texto siempre en negro puro para máxima nitidez
      size: 8.8pt,
    )
    it
  }
)

#set text(font: "Liberation Sans", size: 10.5pt, lang: "es")
#set par(justify: true, leading: 0.65em)
#set heading(numbering: "1.1")
```

9 Instalación de Typst y Extras necesarios

9.1 Typst (Compilador)

La instalación no puede ser más sencilla, tan solo hay que descargar el fichero que contiene el binario de Typst y luego copiarlo a una ruta de nuestro `PATH` de ejecución de programas.

1. Descargar desde GitHub la ultima release disponible.

Descargar Typst (`https://github.com/typst/typst/releases`)



Figura 2: Fichero comprimido. Release de Typst

2. Extraer el fichero `.xz` descargado
3. Copiar el fichero `typst`
   - Solo para tu usuario:

```
cp typst ~/.local/bin/`
```

   - Para todos los usuarios:

```
sudo cp typst /usr/local/bin/
```

9.2 TinyMist (Plugin/Integrador)

Tinymist es una herramienta que permite mostrar una previsualización de todos los que se está escribiendo en los ficheros `.typ`

Al igual que hicimos con `typst` desde el GitHub de este proyecto nos iremos a los «releases» y nos bajaremos directamente el binario de `tinymist` y lo dejaremos en una ruta de ejecución.

1. Descargar la ultima releas desde GitHub

Descargar TinyMist (`https://github.com/Myriad-Dreamin/tinymist/releases`)

2. Copiar el binario a la ruta de ejecución:
   - Solo para nuestro usuario:

```
# Solo para nuestro usuario
cp tinymist-linux-x64 ~/.local/bin/tinymist
```

   - Para todos los usuarios:

```
# Para todos los usuarios
sudo cp tinymist-linux-x64 /usr/local/bin/tinymist
```

10 Texpile (Editor IDE)

Texpile es un editor ligero y fácil de usar para edición de ficheros Typst (`.typ`) con previsualización y que permite además directamente el fichero final del documento.

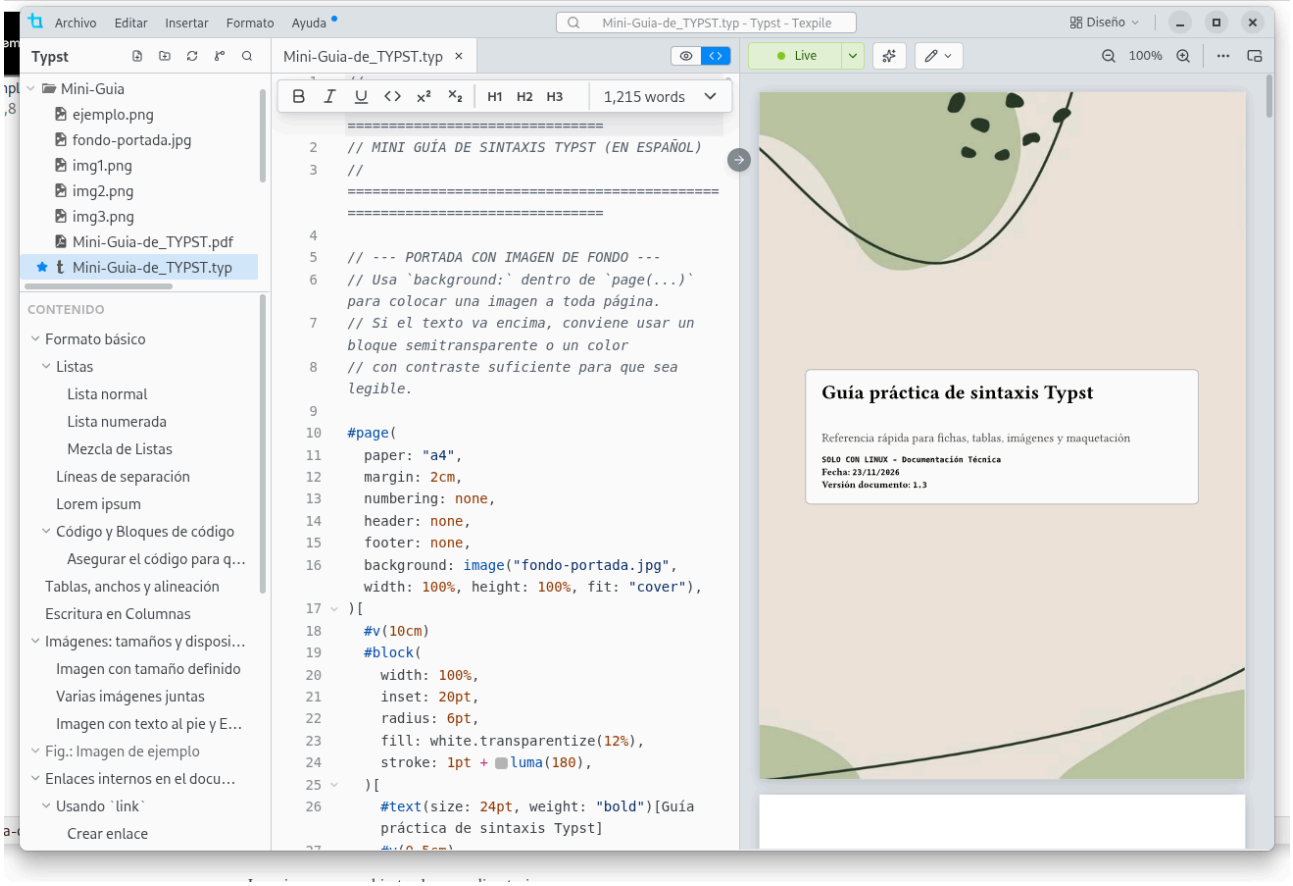


Figura 3: Texpile en modo edicion y previsualización

La instalación de Texpile es realmente sencilla, podemos descargar y usar el fichero `.AppImage`, o usar el fichero `.deb` que hay disponible en el sitio oficial:

Descargar Texpile (`https://texpile.com/download`)

10.1 AppImage

Si obtamos por usar el fichero `.AppImage` deberemos darle permisos de ejecución y crear un acceso directo para poder usarlo cuando queramos.

```
chmod +x Texpile.AppImage
```

10.2 Fichero .deb

Si escogemos usar el fichero `.deb` la integración con nuestro escritorio será completa. Para ello simplemente ejecutaremos desde el directorio desde donde lo hemos descargado:

```
sudo apt install ./texpile-2.0.1.deb
```

11 Creación y Uso de Índices

Para crear indices se usa la función `#outline` y se separa el indice para que ocupe una o varias paginas independientes mediante `#pagebreak()`:

```
#pagebreak()

#outline(
  title: [Contenido],
  depth: 2,
  indent: 1.2em,
)

#pagebreak()
```

Contenido

1 Formato básico . . . 3
1.1 Listas . . . 4
1.2 Líneas de separación . . . 5
1.3 Lorem ipsum . . . 5
1.4 Código y Bloques de código . . . 6
2 Tablas, anchos y alineación . . . 9
3 Escritura en Columnas . . . 9
4 Imágenes: tamaños y disposición múltiple . . . 10
4.1 Imagen con tamaño definido . . . 10
4.2 Varias imágenes juntas . . . 10
4.3 Imagen con texto al pie y Enlace a la Figura . . . 11
5 Enlaces internos en el documento . . . 12
5.1 Usando `link` . . . 12
5.2 Usando `#ref` . . . 12
5.3 Ir al la referencia creada . . . 12
6 Anotaciones y pies de página . . . 13
6.1 Notas al pie (`#footnote`) . . . 13
6.2 Anotaciones o cajas de aviso en el texto . . . 13
7 Portadas de página . . . 14
7.1 Portada Estandard . . . 14
7.2 Portada con imagen de fondo . . . 15
8 Configuración del Documento . . . 16
9 Instalación de Typst y Extras necesarios . . . 17
9.1 Typst (Compilador) . . . 17
9.2 TinyMist (Plugin/Integrador) . . . 17
10 Texpile (Editor IDE) . . . 18
10.1 AppImage . . . 18
10.2 Fichero .deb . . . 18
11 Creación y Uso de Índices . . . 19