Apuntes de Typst

Carlos E. Tafur Egido
Artos Institute
tung@artos.edu

Eugene Deklan
Honduras State
e.deklan@hstate.hn

Abstract
Apuntes del *software* de composición tipográfica Typst.

Índice

Introducción

```rust
pub fn main() {
    println!("Hello, world!");
}
```

También, se puede presentar código Python:

```python
print("Hello, world!")
```

El símbolo de conjunto vacío es: $\emptyset$. Si se encuentra en modo matemático, no habría que precederlo de `#sym`.

Desde hace tiempo, vengo pensando en Typst como una alternativa moderna a TeX y LaTeX. He probado varios lenguajes pero no terminaba de convencerme ninguno, pues, en la mayoría de los casos, se trataba de lenguajes con poca expresividad y recursos.

Muchos de estos son lo que se suele llamar lenguajes de marcado ligero, como es el caso del famoso Markdown o de otros con algo más de riqueza, como pueden ser reStructuredText o Djot. En cualquier caso, estos lenguajes tienen bastantes limitaciones en cuanto a las opciones que proporcionan.

No digo que esto tenga que ser malo para todo el mundo. Simplemente, lo es para mí, que deseo hacer un uso algo intensivo de la tipografía TKTK.

Hay quien no requiere de funcionalidades avanzadas en este sentido. Por ejemplo, alguien que escriba un libro de historia.

También, hay quien cree que un documento no debería ser programable. TKTK.

Hacks

Creo que Typst tiene un fallo de diseño en la gestión de las etiquetas de las ecuaciones. No es tan grave, pero sí que es molesto. En principio, la única forma que se tiene de gestionar las etiquetas es activando la numeración para todas las ecuaciones que se muestren. Esto es algo molesto y hay quien prefiere que se numeren solo las ecuaciones a las que se hace referencia. Para esto, se puede hacer lo que explican en este enlace.

Paquetes y plantillas

Al igual que LaTeX, Typst cuenta con un repositorio con paquetes (*packages*), incluyendo también a plantillas (*templates*), con el que podrá lograr fácilmente textos muy avanzados, con gráficos, etc. El repositorio oficial de Typst recibe el nombre de [Typst Universe][].

[Typst Universe]: https://typst.app/universe/

Al contrario de lo que sucede con TeX y sus formatos, crear un paquete básico o una plantilla para Typst no es complicado.

La explicación de cómo crear un paquete para su uso interno en su sistema o bien promocionarlo (*submit*) en Typst Universe se encuentra en el [repositorio][repo-packages] oficial de Typst a este respecto.

[repo-packages]: https://github.com/typst/packages

La primera plantilla que hice la publiqué en un repositorio de GitHub y la añadía a mis documentos de Typst mediante un submódulo de Git. Lo cierto es que era bastante engorro. Es mejor que TKTK.

Lo primero que le recomendaría es que crease el paquete de forma local en su sistema y, una vez que lo tenga bastante depurado, lo «suba» a Typst Universe. Para esto, debe tener en cuenta lo que se explica en la parte de [Local packages][]. Tenga en cuenta que, aunque no exista el directorio de Typst donde ahí se indica, puede crearlo usted mismo y el propio Typst reconocerá ese paquete local.

[Local packages]: https://github.com/typst/packages?tab=readme-ov-file#local-packages

Algo que debe tener es un archivo *typst.toml* en el directorio raíz del paquete. Puede ver la [documentación sobre este archivo][typst-toml-docs]. Tenga en cuenta que, si se trata de uina plantilla (*template*), será algo diferente, tal y como se explica ahí.

[typst-toml-docs]: https://github.com/typst/packages/blob/main/docs/manifest.md

Si finalmente decide «subir» el paquete a Typst Universe, puede consultar en la documentación oficial [cómo se hace][typst-submit-universe].

[typst-submit-universe]: https://github.com/typst/packages/blob/main/docs/README.md#package-submission-guidelines

Otros

HOLA ESTO

Ahora no.

uno	dos	tres	cuatro	cinco
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25
6	12	18	24	30
7	14	21	28	35
8	16	24	32	40
9	18	27	36	45

Algo que es muy útil es consultar, en la documentación oficial, los símbolos existentes. Se pueden consultar aquí.

Entornos tipo teorema

En realidad, no solo es para los entornos de tipo teorema, sino que serviría para cualquier tipo.

Aunque existen *plug-ins* que parece que lo hacen muy bien, Typst es tan bueno que es muy fácil crearlos por uno mismo. Puede ver cómo se crean en algunos ejemplos que ponen en la entrada *counter*. Por ejemplo, en la sección *How to step*.

Código en el Texto

Puede consultar, en la documentación oficial, la sección titulada link(«https://typst.app/docs/reference/scripting/»)[*Scripting*].

Es algo extraño el uso del símbolo #, pero permite hacer muy cómoda la creación de lo que en el mundo TeX llaman entornos.

Fuentes de información y de consulta

Algo que parece que están haciendo muy bien los creadores de Typst es propiciar que toda la actividad relativa al proyecto pase por ellos. Así, se tiene, por ejemplo, el [foro oficial del proyecto][foro-oficial], hospedado bajo su misma web. En el fondo lo que usa es Discourse, con lo que se ve muy moderno y con gran usabilidad.

[foro-oficial]: https://forum.typst.app/

También, a la hora de crear documentación en los módulos Typst, existe un paquete llamado [Tidy][] con el que podrá hacerlo en Typst.

[Tidy]: https://typst.app/universe/package/tidy

Carencias

Typst en un proyecto que aún no lleva muchos años en fase de desarrollo, con lo que es normal que tenga algunas carencias. Aun así, creo que actualmente merece mucho la pena pasarse de LaTeX a Typst.

En lo que respecta a las cosas que echo en falta, por ejemplo, se tiene que la adaptación al español aún no está muy lograda, pero creo que no tardará en llegar.

EE

Algo que estaría muy bien es que, en el texto que se pone dentro del modo matemáticas, se usase la misma fuente tipográfica que tiene el resto del texto normal, es decir, el del cuerpo de texto.

Hay un *issue* de GitHub sobre esta cuestión, solo que aún están debatiendo sobre si es recomendable que tenga este comportamiento o no.

Coeficientes binómicos

Algunas identidades

Una identidad sobre los coeficientes binómicos que es muy útil es la siguiente.

Simetría en los Coeficientes Binómicos

Dados $n, k \in \mathbb{N}$, se cumple

$$\binom{n}{k} = \binom{n}{n-k}$$

Se podría demostrar mediante su expresión como fracción de factoriales, pero más elegante sería dar una definición combinatoria. En este caso, es bastante sencilla.

Recuerde que el coeficiente binómico

$$\binom{n}{k}$$

indica las distintas selecciones de k elementos de un conjunto de n elementos, sin tener en cuenta el orden. Pero, si se fija, al seleccionar k elementos, también está seleccionando los $n-k$ elementos restantes. Es decir, los elementos que deshecha es como si también los seleccionara, por lo que el número de selecciones posibles sería igual que el número de las distintas formas de dechechar elementos.

Ahora, vamos a ver otra identidad que es muy útil.

Identidad de Pascal Dados $k, n \in \mathbb{N}$ siendo $1 \leq k \leq n-1$. Se cumple

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

La justificación de los valores entre los que se pueden mover las variables que aparecen, lo que debe tener en cuenta es que, en el desarrollo del coeficiente binómico no aparezca un factorial de un número negativo. Sí puede aparecer $0!$, que, como dijimos, vale 1.

También hay quien la llama Regla de Pascal o Fórmula de Pascal.

Se pueden dar varias demostraciones de este hecho. Una muy directa y sencilla es mediante manipulaciones algebraicas de esos coeficientes binómicos. Es bastante fácil. También se puede dar una demostración combinatoria que es más creativa y entretenida. Sería la siguiente.

Sea S un conjunto de n elementos. Designaremos por S_k al conjunto de todos sus subconjuntos de tamaño k. Se podría expresar como $S_k = \mathcal{P}_k(S)$. Como ya sabemos por combinatoria, se tiene que

$$|S_k| = \binom{n}{k}$$

Lo que pretendemos hacer a continuación será, a partir de la selección de un elemento arbitrario e de S, formar una partición de S_k en dos conjuntos de conjuntos de tamaño k tales que los de uno contengan siempre a e y los del otro no lo contengan nunca.

Partimos de un conjunto

$$D = S \setminus \{e\}$$

que, como es evidente, tiene por tamaño $n-1$.

A partir de este, generamos el conjunto D_k, formado por todos los subconjuntos de D de tamaño k. Se tiene, por lo que ya sabemos de combinatoria, que

$$|D_k| = \binom{n-1}{k}$$

Ahora, en lugar de centrarnos en D_k, lo hacemos en D_{k-1}, es decir, el conjunto de todos los subconjuntos de D de tamaño $k-1$. Este tiene por tamaño

$$|D_{k-1}| = \binom{n-1}{k-1}$$

A partir de este, vamos a generar un conjunto F_k simplemente uniendo cada uno de sus elementos, que son conjuntos de tamaño $k-1$, con $\{e\}$. Por tanto, los conjuntos que constituyen F_k tienen tamaño k. Alternativamente, podíamos haber dado la definición en forma simbólica siguiente:

$$F_k = \{M \cup \{e\} \mid M \in D_{k-1}\}$$

El tamaño de F_k será, entonces, el mismo que el de D_{k-1}, que es

$$|F_k| = |D_{k-1}| = \binom{n-1}{k-1}$$

Si se fija, todos pares de conjuntos de D_k y F_k son disjuntos, puesto que todos los conjuntos que constituyen a F_k contienen al elemento e, cosa que no sucede para ninguno de los que constituyen a D_k. Además, entre ambos, forman todos los subconjuntos posibles de k elementos de S, es decir, S_k. Es decir,

$$S_k = D_k \cup F_k \quad \text{y} \quad D_k \cap F_k = \emptyset$$

Debido a esto, D_k y F_k forman una partición de S_k, tal y como pretendíamos. Por tanto, se tiene que

$$|S_k| = |D_k| + |F_k|$$

o, lo que es lo mismo,

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Si se fija, la fórmula anterior nos sirve como definición recursiva del coeficiente binómico. TKTK.

Vamos ahora a hacer una prueba para comprobar si se puede ampliar el signo de llaves:

$$n! = \begin{cases} 1 & \text{si } n = 0 \\ (n-1)!n & \text{si } n \neq 0 \end{cases}$$