

# L<sup>A</sup>T<sub>E</sub>X para la tarea de mañana

## Clase #1: Base

Amado Alberto Cabrera Estrada  
[amadocab.268@gmail.com](mailto:amadocab.268@gmail.com)\*

Junio 2022

### Resumen

En esta clase se explicará la sintaxis básica de L<sup>A</sup>T<sub>E</sub>X y conceptos relevantes para la escritura de nuestros documentos. Además de eso, se hará un repaso del preámbulo básico que se usará para maquetar nuestros documentos explicando uno por uno los comandos que lo componen.

## Índice

<b>1. Partes de un archivo L<sup>A</sup>T<sub>E</sub>X</b>	<b>2</b>
1.1. Estructura . . . . .	2
1.2. Clases y paquetes . . . . .	2
1.2.1. Clases . . . . .	2
1.2.2. Paquetes . . . . .	2
<b>2. Comandos y <i>environments</i></b>	<b>3</b>
2.1. Comandos . . . . .	3
2.2. <i>Environments</i> . . . . .	4
2.3. Más estructuras . . . . .	4
<b>3. Preámbulo base</b>	<b>4</b>
3.1. <code>article</code> . . . . .	5
3.2. <code>inputenc</code> . . . . .	5
3.3. <code>fontenc</code> . . . . .	5
3.4. <code>babel</code> . . . . .	5
<b>4. Más información</b>	<b>6</b>

---

\*Para sugerencias, correcciones o comentarios

# 1. Partes de un archivo $\text{\LaTeX}$

## 1.1. Estructura

Un archivo de  $\text{\LaTeX}$  (`.tex`) se compone de dos partes principales, el **preámbulo** y el **cuerpo** del documento. El preámbulo, tal como su nombre indica, es la parte preliminar al cuerpo y donde escribe toda la configuración que será posteriormente aplicada al documento. Por otro, lado el cuerpo es donde se coloca todo el contenido que será presentado en el documento y es a este al cual se le aplican las configuraciones del preámbulo.

En la imagen de abajo se puede ver un pequeño ejemplo de como se ve generalmente un archivo de  $\text{\LaTeX}$ . Donde se marca la parte correspondiente al preámbulo y la correspondiente al cuerpo.

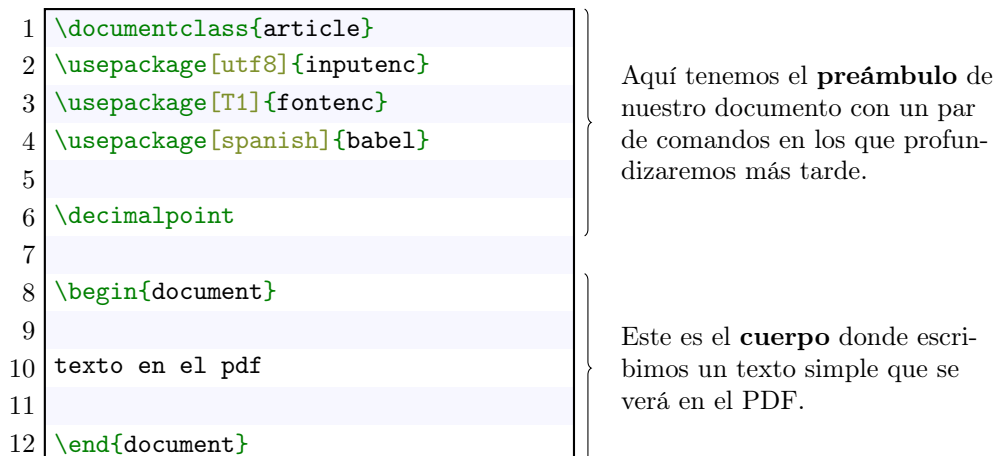


Figura 1: Estructura de un `.tex`

## 1.2. Clases y paquetes

Dentro de  $\text{\LaTeX}$  hay dos conceptos muy importantes a entender para conocer bien el funcionamiento de este. Estos son los conceptos de **clase** (`.cls`) y **paquete** (`.sty`). Fundamentalmente hay que saber que estos dos son archivos que contienen un conjunto de configuraciones predeterminadas o nuevas funcionalidades que nosotros podemos añadir a nuestros documentos. Se puede acceder a dichas funcionalidades simplemente llamándolas desde el preámbulo.

### 1.2.1. Clases

Las clases pueden entenderse como plantillas básicas sobre las que se construye un documento y estas son las principales encargadas de su configuración.  $\text{\LaTeX}$  nos provee unas clases básicas para hacer nuestros documentos, estas son `book`, `report`, `article`, `letter` y `slides`. Estas se pueden llamar por el comando:

```
\documentclass{<nombre de la clase>}
```

### 1.2.2. Paquetes

Los paquetes, por otro lado, suelen añadir nuevas funcionalidades o facilidades a nuestros documentos, estas *-idealmente-* funcionan con cualquier tipo de clase y paquete. Se puede llamar a las clases por medio del comando:

```
\usepackage{<nombre del paquete>}
```

Para más información pueden visitar este artículo de  $\text{\TeX}$ <sub>FAQ</sub><sup>[1]</sup>.

## 2. Comandos y *environments*

En  $\text{\LaTeX}$  existen dos unidades básicas que nos permiten llamar todo tipo de funcionalidades en nuestros documentos, los comandos y los *environments* -aunque este segundo tiene traducción literal se preferirá el término en inglés-.

### 2.1. Comandos

Un comando en  $\text{\LaTeX}$  es equivalente a lo que en otros lenguajes de programación llamaríamos una función y llaman algún tipo de funcionalidad especial. Más adelante en nuestro curso aprenderemos a crear nuestros propios comandos para poder abreviar el código que utilizamos en la maquetación de nuestros documentos.

La estructura de un comando en  $\text{\LaTeX}$  consiste en una barra invertida y una secuencia de letras precedida por los argumentos que recibe (si es que lo hace) el comando.

Los argumentos de los comandos pueden ser de dos tipos, **opcionales** y **obligatorios**. Los argumentos opcionales se escriben entre «[]» (corchetes) y -convencionalmente- solo puede ser uno, los argumentos obligatorios se colocan entre «{}» (llaves) y pueden ser hasta 9 (u 8 en presencia de un argumento opcional).

Ejemplos:

**Comando sin argumentos:**

```
\palabra
```

Un ejemplo de esto es:

```
\LaTeX
```

**Comando solo con argumento opcional:**

```
\palabra[arg]
```

Un ejemplo de esto es:

```
\kant[1] % Del paquete kantlipsum
```

**Comando solo con argumento obligatorio:**  
extensible hasta 9 pares de llaves

```
\palabra{arg}
```

Un ejemplo de esto es:  
(con un solo argumento obligatorio)

```
\textbf{texto}
```

**Comando con cada tipo de argumento:**  
extensible hasta 8 pares de llaves

```
\palabra[arg1]{arg2}
```

Un ejemplo de esto es:  
(con un solo argumento obligatorio)

```
\sqrt[n]{expr}
```

De manera adicional a esto existen comandos que aceptan además un «\*» (asterisco) luego de la palabra que los compone, estos son llamados la «versión estrellada» de un comando y suele cambiar ligeramente el comportamiento del comando al cual es aplicado. Este estilo de comando estrellado puede combinarse con cualquier ejemplo de los anteriormente mencionados.

Ejemplo:

**Comando estrellado:**

```
\palabra*{arg}
```

Un ejemplo de esto es:

```
\section*{expr}
```

Cabe mencionar que los comandos de ejemplo utilizados para mostrar la forma general de estos serán estudiados más adelante en el curso y el lector no debe preocuparse por ellos, si no hasta después.

## 2.2. *Environments*

Los *environments* no tienen un análogo tan claro<sup>1</sup> en otros lenguajes de programación, la mejor manera en la cual podemos describirlos es como un conjunto de configuraciones que se aplican a una parte delimitada de nuestro documento. Más adelante en el curso también aprenderemos a definirlos.

La estructura de un *environment* de L<sup>A</sup>T<sub>E</sub>X consiste en dos comandos, uno de inicio y otro de cierre, que envuelven la sección del archivo a la cual se le aplicarán nuestras definiciones. La manera de aplicar argumentos a nuestros *environments* es exactamente la misma que para nuestros comandos, estos argumentos se pasan al comando que da inicio a nuestro *environment*. (Cabe mencionar que el cuerpo de nuestro documento está delimitado por un *environment*, como se ve en la figura 1)

Ejemplos de *environments*:

**Environment sin argumentos:**

```
\begin{nombre}  
contenido  
\end{nombre}
```

Un ejemplo de esto es:

```
\begin{center}  
Parte del documento que queremos centrar  
\end{center}
```

**Environment con todos los argumentos:**

```
\begin{nombre}[opt]{mand}  
contenido  
\end{nombre}
```

Un ejemplo de esto es:

```
\begin{minipage}[h]{0.9\textwidth}  
Formateado especial  
\end{minipage}
```

Los *environments* al igual que los comandos tienen su versión estrellada, en estos el asterisco se pone en el nombre del *environment* tanto al iniciarlo como al cerrarlo.

Ejemplo:

**Environment con todos los argumentos:**

```
\begin{nombre*}[opt]{mand}  
contenido  
\end{nombre*}
```

Un ejemplo de esto es:

```
\begin{equation*}  
ecuaciones % Del paquete amsmath  
\end{equation*}
```

Al igual que antes el lector no debe preocuparse por los ejemplos utilizados que solo muestran la forma general de los *environments*.

## 2.3. Más estructuras

Pese a que nosotros estamos simplificando nuestro conocimiento de L<sup>A</sup>T<sub>E</sub>X únicamente a los comandos y *environments*, es necesario mencionar que L<sup>A</sup>T<sub>E</sub>X está compuesto por muchísimas cosas más, como: contadores, tipos de pegamento, *cat codes*, estructuras de control, bucles, etc. En nuestro curso estos no serán temas tratados, pero es importante conocerlos por si algún día el lector quiere profundizar en alguno de los temas.

## 3. Preámbulo base

Cuando uno inicia en L<sup>A</sup>T<sub>E</sub>X es común que se provea una plantilla sobre la cual se pueda iniciar a escribir sin ninguna dificultad, pero el desconocimiento de lo que se está usando puede llevar a documentos extremadamente grandes y poco funcionales. Con este motivo en el curso iremos construyendo el preámbulo poco a poco y entendiendo que cosas se añaden y por que.

<sup>1</sup>Para aquellos familiarizados con el encapsulamiento, los *environments* proveen de este a comandos que afectan de manera general al contenido del documento.

Para iniciar el curso les proveeré de un preámbulo fundamental para escribir documentos en español, este es la configuración mínima necesaria para iniciar.

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[spanish]{babel}
\decimalpoint
```

Ahora entenderemos qué hace cada una de estas instrucciones en nuestro documento.

### 3.1. article

**article** es una de las clases base de  $\text{\LaTeX}$ , en esta se definen una gran cantidad de comandos, *environments* y otras configuraciones necesarias para maquetar un documento.

### 3.2. inputenc

El paquete **inputenc** sirve para decirle al procesador de  $\text{\LaTeX}$  cuál es la codificación en la que se encuentra nuestro archivo `.tex`. En nuestro preámbulo le estamos diciendo a  $\text{\LaTeX}$  con el argumento opcional `utf8` que es esa nuestra codificación -Esto ya no es necesario, pues  $\text{\TeX}$  por defecto ya asume esa configuración, sin embargo es buena práctica hacerlo explícito en el archivo-.

### 3.3. fontenc

**fontenc** funciona de manera muy similar a **inputenc**, pero en vez de especificar la codificación del texto en un archivo de texto plano (el `.tex`); este se preocupa de la codificación de la fuente del archivo de salida de  $\text{\LaTeX}$  (usualmente `.pdf`).

El tema de la codificación de las fuentes no es tan trivial, por eso preparé un ejemplo con el que pueden verificar por su propia cuenta la diferencia.

Este es un texto de prueba que tiene como función demostrar que usar ciertos caracteres especiales trae consigo ciertos problemas no perceptibles a simple vista. Con palabras como canción, garçon, grüß gott, entre otros.

Sin **fontenc**, T1

Este es un texto de prueba que tiene como función demostrar que usar ciertos caracteres especiales trae consigo ciertos problemas no perceptibles a simple vista. Con palabras como canción, garçon, grüß gott, entre otros.

Con **fontenc**, T1

Figura 2: Ejemplos del uso de **fontenc**

Pese a que externamente no parezca que haya ninguna diferencia entre los dos lados (y de hecho esta no es visible), basta con seleccionar cada uno de los dos lados y pegarlos en un editor de texto y se podrá ver cómo los caracteres del lado que usa **fontenc** se escriben correctamente y el resto no.

### 3.4. babel

Por último, **babel** es un paquete que se encarga de manejar los idiomas: activa caracteres que usualmente darían error en un archivo, ayuda a separar las palabras correctamente cuando estas se cortan al final de una línea y traduce las etiquetas que están escritas en inglés por defecto. Ya que algunos países de habla hispana utilizan la «*,*» (*coma*) para separar los decimales de los enteros, este paquete hace ese cambio de manera automática; para evitar esto puede usarse el comando `\decimalpoint`.

## 4. Más información

Por ahora solo he hecho una recopilación muy pequeña de paquetes y clases que se pueden usar en L<sup>A</sup>T<sub>E</sub>X, sin embargo en internet existe una web oficial, la *Comprehensive T<sub>E</sub>X Archive Network* (CTAN), donde se publica todo lo relacionado a T<sub>E</sub>X y familia (entre ellos paquetes, clases, etc).

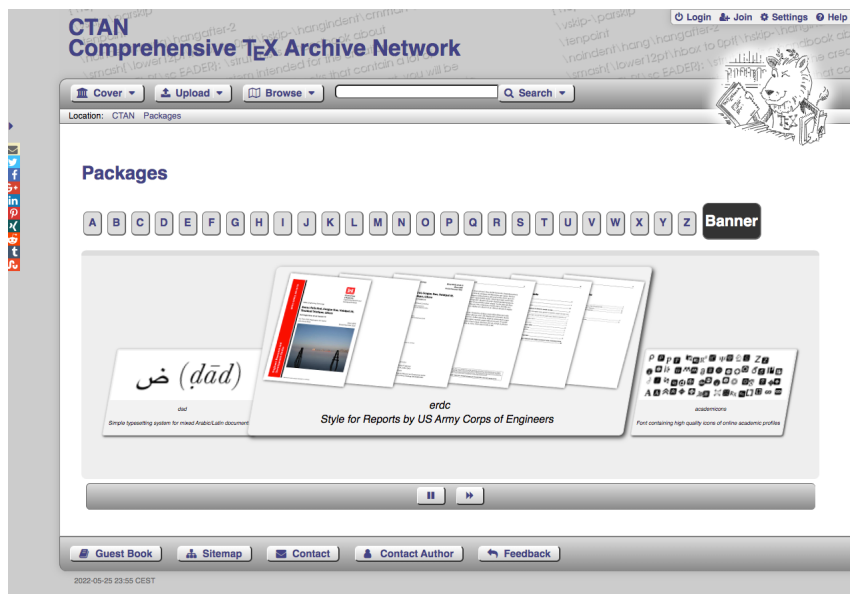


Figura 3: Página de [ctan.org](https://ctan.org)

Les recomiendo ampliamente la lectura de las documentaciones oficiales de los paquetes que quieran utilizar (de tenerla). Durante el curso procuraré facilitar las documentaciones oficiales de los paquetes que usemos al momento de necesitarlos. Para no abrumar durante esta primera primera clase no coloqué las documentaciones, pero para los curiosos a continuación les facilito la referencia.

- `article`<sup>[3]</sup>
- `inputenc`<sup>[5]</sup>
- `fontenc`<sup>[4]</sup>
- `babel`<sup>[2]</sup>

## Referencias

- [1] TeX FAQ, ed. *What are LaTeX classes and packages?* [Online; accessed 23-May-2022]. 2022. URL: <https://texfaq.org/FAQ-clsvpkg>.
- [2] Javier Bezos López. *babel – Multilingual support for L<sup>A</sup>T<sub>E</sub>X, LuaL<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, and Plain T<sub>E</sub>X*. URL: <https://ctan.org/pkg/babel>.
- [3] The L<sup>A</sup>T<sub>E</sub>X Team. *article – Default class for composing an article*. 2021. URL: <https://ctan.org/pkg/article>.
- [4] The L<sup>A</sup>T<sub>E</sub>X Team. *fontenc – Standard package for selecting font encodings*. URL: <https://ctan.org/pkg/fontenc>.
- [5] The L<sup>A</sup>T<sub>E</sub>X Team. *inputenc – Accept different input encodings*. 2021. URL: <https://ctan.org/pkg/inputenc>.