

# Comenzamos

Composición de textos científicos

11 de septiembre de 2020

## 1. Datos generales

Nombre oficial de la asignatura: Composición de textos científicos con editores de libre distribución ( $\text{\LaTeX}$ ).

Profesor: Fernando Chamizo.

Horario: Viernes 13:30–15:30.

Modalidad de docencia: En *streaming* via TEAMS. Se intentará grabar todas las clases, las cuales quedarán almacenadas en Microsoft Stream hasta la fecha de caducidad impuesta por esta plataforma.

Página web del curso: <http://matematicas.uam.es/~fernando.chamizo/asignaturas/2021latex/2021latex.html>. Habrá periódicamente una copia en <https://moodle.uam.es/>. Moodle se usará sobre todo para las calificaciones y para entregar las tareas semanales.

## 2. Implementación del temario

El temario tal como aparece en la guía docente está, para mi gusto, demasiado dividido en temas específicos. El alumno medio que tengo en mente es alguien que quiere adquirir rápidamente los conocimientos suficientes para empezar a usarlos en la redacción de su TFG u otros trabajos. Por tanto mi idea es comenzar de lo fácil a lo difícil cubriendo lo antes posible todos los temas importantes de forma somera y utilizar el resto del curso para profundizar sobre ellos. En definitiva, se prevé tratar todos los temas de la guía docente, reduciendo quizá un poco la parte de programación en  $\text{\LaTeX}$ , pero no se guardará la ordenación allí propuesta. Con suerte habrá también tiempo para tratar algunos paquetes especiales o al menos para que formen parte de proyectos finales o tareas voluntarias.

## 3. Evaluación

El curso se evalúa mediante unas tareas semanales y un proyecto final. La calificación del curso viene dada por la fórmula

$$\text{Calificación} = \min(10, 0.8S + 0.2F)$$

Aquí  $F$  es la nota en el proyecto final y  $S$  es la calificación media de las tareas semanales con el añadido de unas posibles tareas extra voluntarias, con lo cual  $S$  podría superar el valor 10 y de ahí que se considere el mínimo. El proyecto final no es necesario para aprobar pues solo con las tareas semanales, sin extras, uno puede obtener hasta un 8. Mi idea es que haya suficientes tareas extra como para que cualquiera que desee evitar el proyecto final para estar menos agobiado al terminar el curso pueda hacerlo sin perder la oportunidad de una calificación alta.

#### 4. ¿Qué es L<sup>A</sup>T<sub>E</sub>X?

En los años 80 el famoso informático D. Knuth creó el sistema tipográfico T<sub>E</sub>X y L<sup>A</sup>T<sub>E</sub>X surgió de la mano de L. Lamport para facilitar su uso. En pocas palabras, sobre el papel L<sup>A</sup>T<sub>E</sub>X es T<sub>E</sub>X con nuevos comandos y entornos predefinidos y en principio uno podría reproducir el aspecto visual de cualquier documento L<sup>A</sup>T<sub>E</sub>X empleando T<sub>E</sub>X. En la práctica, muy pocos usuarios hoy en día sabrían desenvolverse con soltura utilizando solo T<sub>E</sub>X porque es demasiado primitivo, hay demasiadas cosas básicas que requerirían conocimientos profundos y una cantidad nada desdeñable de código. Ciertamente el hijo ha superado al padre.

Al principio, L<sup>A</sup>T<sub>E</sub>X era un poco rígido, si no te gustaba la forma en que se había diseñado el formato de los títulos de secciones o cosas similares la única forma era aguantarse o aprender mucho del funcionamiento interno. Una comunidad creciente de desarrolladores, una documentación muy extensa y la posibilidad de cargar paquetes para las cosas más peregrinas, han hecho que esto sea cosa del pasado.

Una pregunta natural es para qué quiero un L<sup>A</sup>T<sub>E</sub>X si ya tengo en mi ordenador algo como Word con dos ventajas fundamentales:

1. Está universalmente extendido. Casi todo ciudadano del siglo XXI con un ordenador podrá abrir un fichero `.docx` o similar.
2. El uso básico es prácticamente trivial. Cualquiera puede editar un documento y ponerse a escribir directamente. Es lo que motiva en los currículos el eufemismo “conocimientos a nivel de usuario”.

Incluso si uno odiara este producto, en muchos ámbitos uno “debe” saber algo de Word. A pesar de ser de pago, hasta la administración pública u organismos dependientes de ella fuerzan a veces a presentar documentación en Word sin alternativas.

En comparación, posiblemente pocos de los que acceden a nuestras redes sociales podrán compilar un fichero `.tex` y la prueba de que no es trivial es que exista un curso como este en una universidad prestigiosa. Lo primero queda rebajado con que siempre podremos crear un fichero PDF, que sí es

universal, de hecho no difundir el fichero fuente nos da cierta manera de demostrar nuestra autoría.

La gran ventaja de  $\text{\LaTeX}$  es obvia para cualquiera que haya pasado por el infierno de utilizar el editor de fórmulas de Word, una vez que uno tiene práctica todo es muchísimo más rápido y queda muchísimo más atractivo. La gran mayoría de las revistas de matemáticas y física, y seguramente de otras áreas científicas, ya solo admiten artículos en  $\text{\LaTeX}$ .

En el corazón de  $\text{\TeX}$  y por tanto de  $\text{\LaTeX}$  hay un complicado algoritmo para la división de palabras en líneas y párrafos con lo cual incluso en un texto sin fórmulas el resultado es más profesional, aunque quizá sea difícil que convenzas a algunos de ello.

Las diferencias más obvias de  $\text{\LaTeX}$  y Word, y lo que hace que la curva de aprendizaje del primero tenga una pendiente inicial alta, son que en  $\text{\LaTeX}$  uno utiliza comandos y no ve el resultado hasta que compila. Esto requiere para empezar cierta memorización de comandos y alguna habilidad para interpretar los posibles errores. Por ejemplo en  $\text{\LaTeX}$  usaremos el comando `\textbf{Texto}` para poner la palabra `Texto` en negrita y si se nos olvida la segunda llave es posible que nos diga que se niega a compilar con un aviso del tipo `File ended while scanning use of \textbf` que en este caso es medianamente orientativo pero otras veces no tanto.

## 5. Instalar o no instalar

Ya sabemos que  $\text{\LaTeX}$  requiere disponer  $\text{\TeX}$ . Es muy conveniente tener también un editor especialmente adaptado que complete comandos o nos señale errores obvios.

Hay dos posibilidades: instalar en nuestro ordenador una distribución de  $\text{\TeX}$  añadiendo previsiblemente un buen editor adaptado o usar  $\text{\LaTeX}$  online o, como parece que se dice más ahora, *en la nube*.

La primera posibilidad tiene los inconvenientes de que lleva un rato y ocupa una cantidad de memoria que no es despreciable para los que tenemos la mentalidad informática rúcana formada con el ZX Spectrum. Por otro lado, si uno va a usar  $\text{\LaTeX}$  a menudo, por ejemplo para escribir su TFG, es lo que a mí me parece más adecuado y así uno no depende de la conexión de red o de los posibles fallos en la plataforma donde tengan los ficheros. De nuevo quizá solo sean paranoias ligadas a un inicio en internet con Mosaic.

Si prefieres no instalar nada o esperar a probar un poco hasta que te decidas, la opción que actualmente tiene mayor difusión es Overleaf. En <https://www.overleaf.com/> uno puede registrarse gratuitamente y utilizar  $\text{\LaTeX}$  de forma muy similar a como si lo tuvieras instalado. Actualmente tengo una cuenta de Overleaf ligada a la universidad pero hace años la tuve de forma privada y prácticamente no me dieron lata con mensajes, más allá de notificarme rara vez alguna novedad.

En mi opinión es más conveniente instalar porque los editores son mejores y uno puede configurar lo que quiera pero todo el curso se puede seguir con Overleaf sin instalar nada. Mi única duda es si algún paquete que uno quiera usar en una tarea voluntaria puede faltar en Overleaf, yo creo que no pero no estoy totalmente seguro. En cualquier caso solo afectaría a trabajos voluntarios.

## 5.1. Instalar

Las dos distribuciones más conocidas de  $\text{T}_{\text{E}}\text{X}$  son `texlive` y `MiKTeX`. La segunda, hasta donde sé, no está disponible para Linux. Hay tantas versiones de los sistema operativos y mi experiencia es tan sesgada que seguramente te pueda brindar poca ayuda. Lo mejor que puedes hacer es buscar unas instrucciones de instalación recientes en la red. Por ejemplo siguiendo los enlaces <https://www.latex-project.org/get/>. En el blog de J.R. Berrendero hay un enlace a una guía, un poco antigua, para instalar MiKTeX con TeXmaker.

En breve, mi experiencia es que en Linux solo he tenido que marcar algo en la lista del gestor de paquetes y en Windows ejecutar un instalador. Lleva tiempo pero en casi todos los casos ha resultado trivial. En Windows el editor se instala después de que se ha instalado la distribución  $\text{T}_{\text{E}}\text{X}$ . Los tres editores más famosos son:

1. Kile (Linux)
2.  $\text{T}_{\text{E}}\text{X}$ maker (posiblemente el más usado)
3.  $\text{T}_{\text{E}}\text{X}$ studio (un proyecto que procede de  $\text{T}_{\text{E}}\text{X}$ studio)
4.  $\text{T}_{\text{E}}\text{X}$ nicCenter (nunca lo he usado)

Mi preferido es Kile. En teoría es posible usarlo también en Windows pero si quieres hacerlo hay que instalar algunas cosas para que simulen cierto entorno de escritorio de Linux y posiblemente no merezca la pena. Estos editores se instalan como cualquier otro programa. Al menos Kile no es muy pesado, ocupa poco en memoria.

No he incluido en la lista anterior LyX que parece que cuenta con sus adeptos porque tengo el prejuicio de que los editores WYSIWYM solo te retrasan en cuanto tienes un poco de práctica. Nunca he conocido a nadie que escriba rápido en  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  y use editores de ese tipo.

## 5.2. No instalar

Si quieres utilizar  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  online a través de Overleaf, debes registrarte en <https://www.overleaf.com/>. Una vez que lo hayas hecho cuando entres ya podrás usar  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .

A veces he visto vender la ventaja de Overleaf frente a instalar  $\text{\LaTeX}$  porque permite trabajar con diferentes versiones y coautores de forma automática. En este curso no vamos a utilizar esta funcionalidad porque el trabajo de cada estudiante debe ser personal. Por otro lado, en lo poco que la he usado yo en un proyecto, no he visto grandes diferencias a intercambiar ficheros por *email*. Supongo que con muchos coautores la cosa cambiaría.

Aunque no te registres en Overleaf podrás disfrutar de muchos ejemplos y documentación acerca de  $\text{\LaTeX}$  que ponen a disposición de cualquiera. El acceso a algunas plantillas está restringido a usuarios registrados.

## 6. Mi primer fichero $\text{\LaTeX}$

Como es lógico cada editor tiene su propia estructura y atajos pero lo básico es bastante intuitivo. Configurar todos los detalles requiere bucear un poco en la documentación. Vamos a ver tres situaciones representativas.

### 6.1. Con Kile

Dentro de **File**, o con el atajo, pinchamos en **New** donde nos aparece una lista de posibles plantillas. Elegimos por ejemplo **Article**. Podemos rellenar datos de esta plantilla o borrar casi todo hasta reducirla a

```
\documentclass[a4paper,10pt]{article}
\usepackage[utf8]{inputenc}

\begin{document}

\end{document}
```

Lo que escribamos entre `\begin{document}` y `\end{document}` será el texto. Para compilarlo y generar un fichero DVI el atajo de teclado es **Alt+2** y para verlo **Alt+3**. Si lo que queremos es un PDF, los análogos son **Alt+6** y **Alt+7**.

### 6.2. Con overleaf

Al entrar verás un botón con “Create first project” o una lista vacía de proyectos y un botón con “New Project”. Pinchando en él tienes las opciones *Blank Project*, *Example Project* y *Upload Project* cuyo nombre es suficientemente explicativo. Elijiendo la primera opción nos pedirá un nombre y tendremos un editor donde aparece la fuente  $\text{\LaTeX}$  a la izquierda y el resultado en PDF a la derecha. Después de hacer los cambios que queramos podemos compilar con el botón de “Recompile”. El fichero PDF resultante se puede descargar a través del icono correspondiente. Por cierto, el que está al lado es para ver el fichero `.log` describiendo todo el transcurso de la

compilación. El `stdout file` es la versión resumida que se vería en consola en otros editores.

### 6.3. Con TeXmaker y TeXstudio

Si abrimos un fichero nuevo estará en blanco. Lo mejor es usar el asistente donde podemos escoger algunos parámetros para hacer una plantilla. No te preocupes si ahora no te imaginas el efecto de casi ninguno de ellos, aceptando la opción por defecto tendrás una plantilla razonable.

Esencialmente la compilación va como con Kile pero hay que sustituir `Alt+n` por `Fn`. Por ejemplo, para compilar en PDF se usa la tecla especial de función F6.

Al abrir el PDF con acrobat reader, al menos con mis versiones no deja compilar de nuevo porque dice que el fichero está ocupado. Si no tenemos una vista por defecto que nos muestre el PDF, yo lo he solucionado utilizando el visor TexWork, incluido en MiKTeX para abrir el PDF.