Comenzamos

Composición de textos científicos

15 de septiembre de 2023

1. Datos generales

Nombre oficial de la asignatura: Composición de textos científicos con editores de libre distribución (LaTeX).

Profesor: Fernando Chamizo.

Horario: Viernes 13:30–15:30.

Aula: Módulo 17, 101.

Página web del curso: `http://matematicas.uam.es/~fernando.chamizo/`. Moodle `https://moodle.uam.es/` se usará para las calificaciones y para entregar las tareas semanales.

2. Desarrollo del temario

A mi juicio, el temario aparece en la guía docente 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. Se prevé tratar todos los temas de la guía docente, reduciendo quizá un poco la parte de programación en 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

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

Aquí S es la calificación media de las tareas semanales, F es la nota en el proyecto final y E es la contribución de las tareas voluntarias. Tanto S

como F están sobre 10 en la fórmula anterior, por tanto el proyecto final no es necesario para aprobar, pues se puede alcanzar hasta un 8 sin él y sin ninguna tarea extra. Por otro lado, 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.

En condiciones ideales, dependiendo de las capacidades de cada uno, el tiempo de clase dedicado a que los alumnos practiquen puede ser suficiente para resolver las tareas, o al menos las menos exigentes.

4. ¿Qué es LaTeX?

En los años 80 el famoso informático D. Knuth creó el sistema tipográfico TeX y LaTeX surgió de la mano de L. Lamport para facilitar su uso. En pocas palabras, sobre el papel LaTeX es TeX con nuevos comandos y entornos predefinidos y en principio uno podría reproducir el aspecto visual de cualquier documento LaTeX empleando TeX. En la práctica, muy pocos usuarios hoy en día sabrían desenvolverse con soltura utilizando solo TeX 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, LaTeX 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, lo que era casi como volver a TeX. 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 LaTeX 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 abrir un documento y ponerse a escribir directamente. Es la excusa para incluir 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. Hasta la administración pública u organismos dependientes de ella fuerzan a veces a presentar documentación en Word sin alternativas, lo cual es llamativo siendo un *software* de pago.

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 este curso en una universidad como la UAM que aparece en los

rankings de centros de prestigio. 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 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 con LaTeX 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 LaTeX.

En el corazón de TeX y por tanto de LaTeX hay un complejo algoritmo para la división de palabras en líneas y párrafos. Gracias a ello, 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 LaTeX y Word, y lo que hace que la curva de aprendizaje del primero tenga una pendiente inicial alta, son que en 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 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 LaTeX requiere disponer TeX. Es muy conveniente tener también un editor especialmente adaptado que complete comandos o nos señale errores obvios. La docencia de esta asignatura tiene lugar en un aula que dispone de ordenadores bajo el sistema operativo Linux (Debian) y con el editor, o entorno de desarrollo, TeXstudio para LaTeX.

Sin embargo más de una vez querremos trabajar en casa o en cualquier otro lugar con nuestro portátil. Si estamos escribiendo nuestro TFG, sería un poco ridículo depender del horario del aula. Hay dos posibilidades: instalar en nuestro ordenador una distribución de TeX y un buen editor adaptado o usar 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 viejos que tenemos la mentalidad informática rácana (formada con el ZX Spectrum y el Amstrad CPC 6128). Por otro lado, si uno va a usar LaTeX a menudo, por ejemplo para escribir su TFG, es lo personalmente 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 LaTeX de forma muy similar a como si lo tuvieras instalado. Tengo una cuenta de Overleaf que apenas uso y prácticamente no me han dado nunca lata con mensajes, más allá de que me notificaron al principio 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 TeX 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 debe instalar después de que se ha instalado la distribución TeX. Los tres editores más famosos son:

1. Kile (Linux)
2. TeXmaker (posiblemente el más usado)
3. TeXstudio (un proyecto que procede de TeXmaker)
4. TeXnicCenter (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 LaTeX y use editores de ese tipo.

5.2. No instalar

Si quieres utilizar LaTeX online a través de Overleaf, debes registrarte en `https://www.overleaf.com/`. Una vez que lo hayas hecho cuando entres ya podrás usar LaTeX.

A veces he visto vender la ventaja de Overleaf frente a instalar 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, no he visto grandes diferencias a intercambiar ficheros por *email*. Supongo que con muchos coautores la cosa cambiaría. Automáticamente guarda las versiones y el historial de los accesos de cada autor.

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

6. Mi primer fichero 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. Eligiendo la primera opción nos pedirá un nombre y tendremos un editor donde aparece la fuente 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, al menos en las versiones de TeXmaker que yo conocía, 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. Por alguna razón en la versión de los ordenadores del aula la tecla para compilar es F5. En cualquier caso, si uno duda, siempre hay un menú con pestañas y seguramente algunos iconos que harán difícil que te pierdas.

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.

7. Dos posibles problemas

En nuestro primer fichero escrito en LaTeX hay dos cosas que inicialmente nos pueden dar dolores de cabeza. Lo mismo con tu editor o con el texto que has escrito, no te percatas de ello, pero es importante que no te sorprendan cuando cambies de ordenador o recibas un fichero de otra persona.

La primera no es algo de LaTeX sino un problema informático general, que los viejos usuarios de Linux conocen bien. Diferentes sistemas utilizan diferentes codificaciones, esto es, diferentes maneras de asignar bytes a los caracteres. Por ejemplo que `O` (la letra o mayúscula) se represente por `4F` en hexadecimal es bastante universal, pero cómo indicar `Ó` (el mismo símbolo acentuado) es algo que habitualmente difiere en Linux o Mac y Windows, lo cual nos puede machacar todos los acentos de un texto. Además dentro de cada uno de estos sistemas operativos uno puede tener editores que trabajen con una u otra codificación. Dentro de un fichero LaTeX, la codificación se puede especificar con el paquete `inputenc` (*input encoding*) que apareció en la cabecera generada con Kile y debe coincidir con la del texto y la del

editor que manejemos. Dos posibilidades son:

`\usepackage[utf8]{inputenc}` y `\usepackage[latin1]{inputenc}`

que corresponden a las dos codificaciones principales usadas en gran parte del mundo o al menos en la que nos resulta más cercana.

La segunda cosa que nos puede dar disgustos es que incluso con la codificación adecuada, los acentos podrían todavía dar problemas o los comandos no responden en el lenguaje que nos gustaría. Por ejemplo, el comando `\today` escribe la fecha de hoy y lo hará en inglés aunque nuestro texto esté en castellano. Evidentemente, LaTeX no puede adivinar en qué idioma queremos escribir y de hecho en el origen de los tiempos solo se podía hacer en inglés y los acentos se lograban con comandos. Afortunadamente, existe el paquete `babel` que admite muchísimos idiomas, entre los que se encuentra, por supuesto, el español. Ya veremos más sobre la internacionalización. Ahora basta con saber que seguramente nos guste incluir muy arriba en nuestra cabecera, antes de la codificación, `\usepackage[spanish]{babel}`. Por ejemplo, un posible código fuente mínimo para escribir en español a partir de lo que nos da como plantilla Overleaf quitando todas los cosas que no usamos es:

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

\begin{document}
\end{document}
```

Recuerda que el texto va entre `\begin{document}` y `\end{document}`. Si quieres prueba el efecto de los comandos `\today` y `\LaTeX`. Seguramente no funcionarán como tú esperabas cuando están dentro de una frase, por ejemplo cuando escribes:

```
\begin{document}
Aprenderé \LaTeX muy
rápido.
\end{document}
```

Si no quieres esperar a otra clase para saber la solución, enciérralos entre llaves o escribe `{}` tras ellos. También te sorprenderá que no te haga mucho caso en los espaciamientos entre palabras, ni respete los fines de línea (como en el ejemplo anterior) o las líneas en blanco. Esto que parece una desventaja, es en realidad una de las grandes fortalezas del corazón de LaTeX, un maravilloso algoritmo que pone los espacios adecuados para que el resultado sea armonioso.

Una cosa más, sobre todo para los que hacéis el TFG en matemáticas. Si lo visto te sabe a poco y quieres empezar a compilar ficheros con una cabecera bastante complicada, por ejemplo la plantilla que te ofrecen para el TFG, lo mismo ves errores muy raros. Es posible que provengan de paquetes de los que no dispones. El remedio expeditivo es instalarlos. En el caso de la plantilla del TFG para matemáticos hay algún paquete un poco inusual y, en general, innecesario (a no ser que hagas cosas avanzadas). Si ese fuera el caso, simplemente precede el `\usepackage` correspondiente por un %, lo que hará que el compilador se olvide de esa línea.

8. Un vistazo más a la cabecera

La cabecera de un documento es lo que hay antes de `\begin{document}`. En nuestros primeros ejemplos, podemos usar algo tan breve como

```
\documentclass{article}
\usepackage[utf8]{inputenc}
```

De hecho esto es lo que pone Overleaf como dos primeras líneas al crear un fichero nuevo. La primera indica que se toman los parámetros por defecto de un formato llamado `article`. Con otros editores se especifican algunos de estos parámetros entre corchetes. Por ejemplo

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

que indica que el formato del papel es A4 y que las fuentes son de tamaño diez puntos. Podemos jugar a cambiar `10pt` por `11pt` o `12pt`, o `a4paper` por `letterpaper` (los folios usados en EE.UU.) y ver los resultados.

La segunda línea indica que la codificación es UTF8 con un parámetro que se pasa al paquete `inputenc`. Parámetros aparte, un paquete es algo que se carga con `\usepackage{...}` y que extiende de cierta forma las capacidades de LaTeX. Hay algunos muy específicos y otros que hacen cosas muy tontas. Por ejemplo, incluyendo `\usepackage{lipsum}` en la cabecera dispondremos de la instrucción `\lipsum` que genera el típico texto *Lorem ipsum* usado en pruebas de imprenta, con las variantes `\lipsum[n-m]` que dan los párrafos entre `n` y `m`. Por ejemplo `\lipsum[11-11]`, que se puede abreviar con `\lipsum[11]`, produce:

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

En matemáticas casi todo el mundo carga tres paquetes de la AMS (*American Mathematical Society*) incluso si no los usa y así lo haremos nosotros. Nuestras cabeceras incluirán

```
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
```

En la práctica, casi ningún matemático tenemos mucha idea de qué comandos dejarían de funcionar si no los cargásemos y la realidad es que en textos simples no necesitaríamos ninguno. Es más una costumbre y una precaución. Un físico o un químico seguro que hará lo mismo con otros paquetes.