

# Introducción a la edición de textos científicos con L<sup>A</sup>T<sub>E</sub>X

Daniel López Avellaneda ([iesmarsarena.org](http://iesmarsarena.org))

José Manuel Calahorra García ([iesmarsarena.org](http://iesmarsarena.org))

# L<sup>A</sup>T<sub>E</sub>X

Manual para el curso online organizado por:

CEP Indalo

<http://www.cepindalo.es>

Octubre 2009 - Enero 2010

# Índice

<b>1. Formatos gráficos</b>	<b>2</b>
1.1. Mapas de bits . . . . .	2
1.2. Gráficos vectoriales . . . . .	3
<b>2. Tipos de gráficos en <math>\LaTeX</math></b>	<b>4</b>
<b>3. El paquete <code>graphicx</code></b>	<b>5</b>
3.1. Orden <code>\includegraphics</code> . . . . .	5
3.2. Orden <code>\scalebox</code> . . . . .	8
3.3. Orden <code>\resizebox</code> . . . . .	10
3.4. Orden <code>\reflectbox</code> . . . . .	11
3.5. Orden <code>\rotatebox</code> . . . . .	12
<b>4. El entorno <code>figure</code></b>	<b>13</b>
<b>5. Otros paquetes gráficos</b>	<b>14</b>

# Gráficos en $\text{\LaTeX}$

## 1 Formatos gráficos

Recordemos algunos conceptos sobre los formatos gráficos. Básicamente existen dos grupos: gráficos de mapa de bits y gráficos vectoriales.

### 1.1 Mapas de bits

*Los gráficos de mapa de bits* se basan en una matriz o tabla de puntos o pixels, donde cada punto tiene un determinado color, por tanto son los más sencillos de almacenar y de reproducir. La desventaja es que si modificamos el tamaño de la imagen pierden calidad. Entre ellos destacamos:

- bmp** Formato de MS-DOS de Micro\$oft que aún siguen usando las últimas versiones de Windows. No tiene ningún tipo de compresión. Ocupan excesivo espacio.
- jpg** También llamado jpeg. Usa un algoritmo de compresión bastante eficiente y basado en que el número de colores que distingue el ojo humano es limitado y por tanto, si dos puntos son de colores distintos pero de diferencia inapreciable para el ojo humano, jpeg los dibuja del mismo color (así no tiene que usar tantos millones de colores distintos). Suele ser el formato usado por cámaras fotográficas. Es el formato que debemos usar para fotos.
- gif** Usa un algoritmo de compresión propietario que puede dar lugar a problemas de patentes (por lo que no aconsejo su uso). Usa como máximo 256 colores, maneja transparencia y puede ser un formato apropiado para pequeños iconos.
- png** Formato bajo licencia libre, que surgió como alternativa a los problemas de patentes del formato gif. Maneja transparencia (algunos navegadores de baja calidad no son capaces de reproducirlo, como versiones antiguas de Explorer). Es el formato que mejor se lleva con  $\text{\LaTeX}$ . Es el formato ideal para guardar imágenes procedentes de capturas de ventanas o pantalla. Su principal virtud es el algoritmo de compresión sin pérdida de calidad.

## 1.2 Gráficos vectoriales

Los gráficos vectoriales están formados por una serie de instrucciones para el ordenador (en lugar de una matriz de puntos), por lo que podemos modificar su tamaño, ampliarlos o reducirlos todo lo que queramos, sin ninguna pérdida de calidad. Las coordenadas de los puntos están dadas de forma relativa, por ello el cambiar el tamaño de la imagen es, para el ordenador, un sencillo ejercicio de multiplicar y/o dividir. Algunos formatos de gráficos vectoriales son:

**SVG** Scalable Vector Graphics es quizás, la mejor opción para gráficos escalables. El formato SVG ya fue recomendado como estándar en 2001 por el W3C (organismo internacional que dicta los estándares para la web). La mayoría de navegadores, como suele ser normal, siguen los estándares web (Firefox lo incluye desde la versión 1.5, Opera desde la versión 8, Amaya lo incluye desde hace muchos años). También suele ser normal que Explorer no cumpla los estándares web y no incorpora SVG, aunque permite ver las imágenes SVG mediante un plug-in (la suma de esos pequeños detalles suele ser la causa de la lentitud del navegador de Micro\$oft). Los gráficos SVG se pueden crear con el programa **Inkscape** (software libre que suele ir incluido en la mayoría de distribuciones LiNuX).

**PS** PostScript es un lenguaje de descripción de páginas usado por muchas impresoras. También ha sido usado tradicionalmente en talleres de impresión profesional, como formato de transporte de gráficos.

**EPS** PostScript encapsulado es un PS con algunas restricciones que facilitan su inclusión como parte de un archivo PostScript

**PDF** El formato PDF también sirve par almacenar gráficos vectoriales, aunque es un formato más complejo (gráficos vectoriales + mapas de bits + texto)

**OTROS** Existen bastantes formatos propietarios de gráficos vectoriales, como los wmf de Windows, los de Corel Draw, programas de CAD, etc.

## 2 Tipos de gráficos en $\LaTeX$

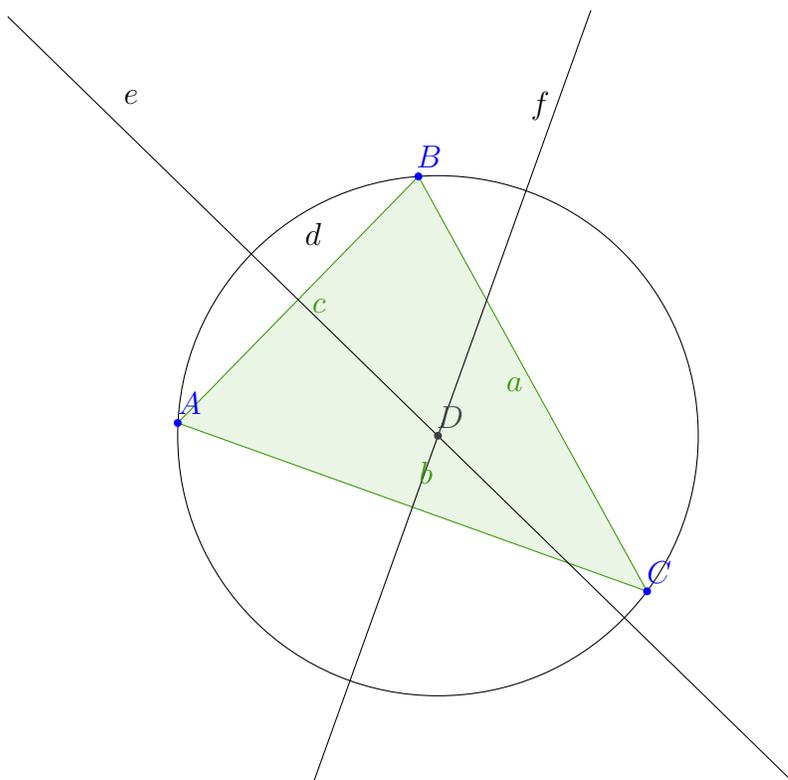
Los formatos de imagen más apropiados para  $\LaTeX$  son **PS** (PostScript) y **EPS** (Encapsulated PostScript). Sin embargo, si el destino de nuestros documentos es convertirlos a PDF (recomendado en este curso) mediante PdfLatex, los formatos apropiados son **PNG**, **JPG**, **GIF** o **PDF**.

Desde el punto de vista de  $\LaTeX$ , debemos distinguir dos tipos principales de gráficos que podemos poner en nuestros documentos (en cuanto a la forma de incluirlos):

1. Imágenes ya existentes (creadas con otros programas, escaneadas, etc.). Es la forma clásica de incluir imágenes en los procesadores de textos tradicionales.



2. Gráficos creados con  $\LaTeX$ . En el propio documento le indicamos en código  $\LaTeX$  el gráfico que queremos crear.



## 3 El paquete graphicx

El paquete *graphicx* es uno de los más usados en  $\text{\LaTeX}$  para incluir gráficos. Para usarlo, tenemos que incluirlo en el preámbulo:

```
\usepackage{graphicx}
```

El paquete *graphicx* incluye comandos para gráficos (incluir gráficos, escalarlos, etc.) y comandos que actúan sobre cualquier objeto, no necesariamente una imagen (podemos rotar, escalar, reflejar, etc. un objeto: texto, fórmulas, tablas, etc.). Las principales ordenes (o comandos) de este paquete son:

- `\includegraphics`
- `\scalebox`
- `\resizebox`
- `\reflectbox`
- `\rotatebox`

### 3.1 Orden `\includegraphics`

Usaremos `\includegraphics` para insertar una imagen disponible en nuestro PC. La sintaxis del comando es la siguiente:

```
\includegraphics[parámetros opcionales]{ruta/imagen}
```

Observe que la orden anterior tiene una parte entre corchetes [lo que hay entre corchetes siempre es opcional en  $\text{\LaTeX}$ ], y una parte entre llaves {lo que hay entre llaves suele ser obligatorio}

- La ruta hacia la carpeta donde se encuentre la imagen es relativa (partiendo desde la carpeta donde tengamos el fichero `.tex`). Si la imagen se encuentra en la misma carpeta que el fichero `.tex` no es necesario poner la ruta. Si el fichero `.tex` se encuentra en la ruta `/home/pepe/curso_latex/fichero.tex` y la imagen (de nombre `img.png`<sup>1</sup>) se encuentra en la ruta `/home/pepe/curso_latex/images/img.png` pondríamos

```
\includegraphics{images/img.png}2
```

<sup>1</sup>Debe observar que un foco de problemas en el funcionamiento de sus programas suele ser la manera de nombrar ficheros y carpetas: un nombre de fichero o carpeta no debe incluir Mayúsculas, ni espacios en blanco, ni caracteres extraños (debería restringirse a letras minúsculas, números y guiones como caracteres de separación). Ejemplo: la carpeta "Mis Exámenes" será un foco de problemas, pero no lo será si su nombre es "mis\_examenes"

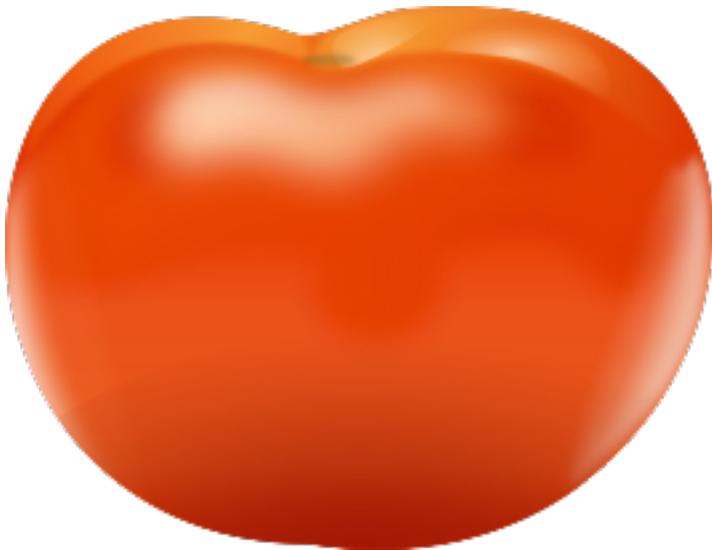
<sup>2</sup>Hay Sistemas Operativos (como Windows) donde las rutas se escriben al revés: se usa la barra invertida (`\`). Esto a veces ocasiona problemas cuando se pone información en Internet, donde las rutas son del tipo "servidor/carpeta/subcarpeta/fichero" (se usa la barra normal `/`)

- Parámetros opcionales. Nos permiten escalar la imagen, rotarla, etc.
  - **scale**: escala (redimensiona) la imagen. Un valor “0.8” reducirá la imagen a un 80 %, mientras que un valor “2” la aumentará al doble.
  - **angle**: especifica el ángulo de rotación
  - **width**: ajusta el gráfico al ancho especificado
  - **height**: ajusta el gráfico al alto especificado

Veamos algunos ejemplos:

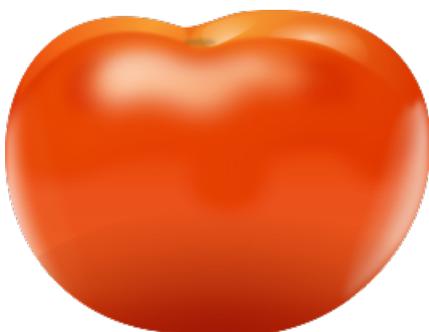
En primer lugar un gráfico normal (sin parámetros)

```
\includegraphics{images/tomate.png}
```



Ahora lo reducimos al 60 %

```
\includegraphics[scale=0.6]{images/tomate.png}
```



Ajustamos el ancho a 3 cm (L<sup>A</sup>T<sub>E</sub>X calculará el porcentaje conveniente para reducirlo a un ancho de 3 cm)

```
\includegraphics[width=3cm]{images/tomate.png}
```



Vamos a rotarlo 45 grados (dejando la escala al 60%)

```
\includegraphics[scale=0.6,angle=45]{images/tomate.png}
```



Ahora dejamos la imagen normal, pero lo recortamos

```
\includegraphics[bb=0cm 0cm 6cm 4cm,clip]{images/tomate.png}
```



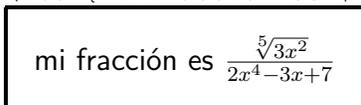
## 3.2 Orden `\scalebox`

Con el comando `\scalebox` podemos escalar un objeto (un texto, una tabla, ..). Su sintaxis es la siguiente:

```
\scalebox{horizontal}[vertical]{objeto}
```

Usaremos un texto recuadrado, que sin escalar aparece como:

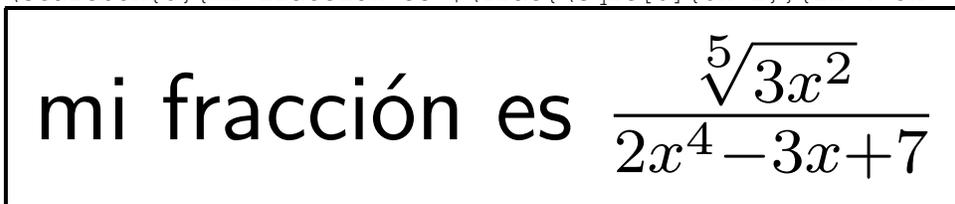
```
\fbox{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}
```



$$\text{mi fracción es } \frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$$

Ahora lo escalamos con el factor 3 (triple)

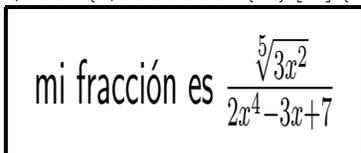
```
\scalebox{3}{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}
```



$$\text{mi fracción es } \frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$$

Podemos escalar también en vertical:

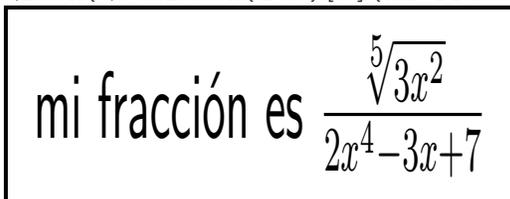
```
\fbox{\scalebox{1}[2]{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}}
```



$$\text{mi fracción es } \frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$$

Si usamos ambas escalas (vertical y horizontal):

```
\fbox{\scalebox{1.5}[3]{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}}
```



$$\text{mi fracción es } \frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$$

En lugar de texto, podemos escalar otro objetos (como una tabla).

En primer lugar veamos la tabla normal:

```

\begin{tabular}{|c|c|}
\hline esto es & una\tabularnewline
\hline \hline tabla & xxxxxxxxxxx\tabularnewline
\hline
\end{tabular}

```

esto es	una
tabla	xxxxxxxxxxx

Ahora escalamos la tabla a un factor 3.5

```

\scalebox{3.5}{
\begin{tabular}{|c|c|}
\hline esto es & una\tabularnewline
\hline \hline tabla & xxxxxxxxxxx\tabularnewline
\hline \end{tabular}
}

```

esto es	una
tabla	xxxxxxxxxxx

Ahora usamos escala horizontal y vertical

```

\scalebox{1.5}[4]{
\begin{tabular}{|c|c|}
\hline esto es & una\tabularnewline
\hline \hline tabla & xxxxxxxxxxx\tabularnewline
\hline \end{tabular}
}

```

esto es	una
tabla	xxxxxxxxxxx

### 3.3 Orden `\resizebox`

Con `\resizebox` podemos ajustar un objeto a unas determinadas dimensiones (ancho y alto).  $\LaTeX$  calculará el porcentaje de escalado para ajustar a esas dimensiones.

```
\resizebox{long-hor}{long-vert}{objeto}
```

Usaremos un texto recuadrado, que sin redimensionar aparece así:

```
\fbox{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}
```

```
\fbox{\resizebox{6cm}{1.5cm}{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}}
```

Con un ancho de 9cm y manteniendo la proporción (!)

```
\fbox{\scalebox{9cm}{!}{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}}
```

El triple de alto (`3\height`):

```
\fbox{\resizebox{\width}{3\height}{mi fracción es $\frac{\sqrt[5]{3x^2}}{2x^4-3x+7}$}}
```

### 3.4 Orden `\reflectbox`

Permite reflejar el texto:

```
{\Large\reflectbox{6cm}{TEXTO REFLEJADO}}
```

TEXTO REFLEJADO

En realidad, es lo mismo que

```
\scalebox{-1}[1]{texto}
```

Si ponemos:

```
\scalebox{-1}[5]{TEXTO REFLEJADO}
```

TEXTO REFLEJADO

Una tabla reflejada

ebcds	12345	12345
ATLHOH	AJOH	ebcds
0087a	087a	87a

## 3.5 Orden `\rotatebox`

Se puede rotar un objeto un ángulo (entre 360 y -360 grados)

```
\rotatebox{ángulo}{objeto}
```

Podemos rotar una tabla, e incluso una celdilla de una tabla. Una tabla normal:

```
\begin{tabular}{|c|c|c|}
\hline & \rotatebox{60}{Enero} & Febrero \\
\hline Faltas & 12 & 3 \\
\hline \end{tabular}
```

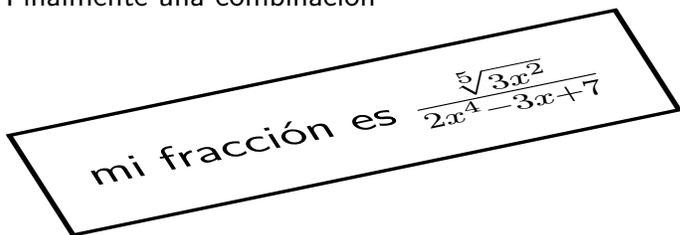
	Enero	Febrero
Faltas	12	3

Ahora rotamos celdas:

```
\begin{tabular}{|c|c|c|}
\hline & \rotatebox{60}{Enero} & \rotatebox{60}{Febrero} \\
\hline Faltas & 12 & 3 \\
\hline \end{tabular}
```

	Enero	Febrero
Faltas	12	3

Finalmente una combinación



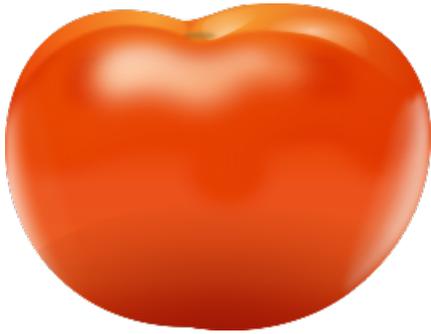


Figura 1: Imagen vectorial de un tomate

## 4 El entorno figure

Cuando insertamos una imagen o tabla al final de una página, puede que no haya espacio suficiente y entonces  $\LaTeX$  pasará esa imagen o tabla a la página siguiente, quedándose la página actual con un espacio en blanco que puede resultar algo feo en publicaciones profesionales.

$\LaTeX$  soluciona este tema con el entorno figure, que permite incluir gráficos o tablas “flotantes”, es decir no se sitúan en un lugar concreto del documento, sino que pueden “flotar” y situarse en una página anterior, en la página siguiente, etc. ( $\LaTeX$  calcula automáticamente el lugar más apropiado para situar ese objeto).

Este sistema de flotantes tiene partidarios y detractores. Particularmente no soy amante de este sistema, entre otras cosas porque debemos olvidarnos de expresiones del tipo “en la imagen puede observar ...” (puesto que la imagen puede estar dos páginas antes).

El entorno figure se usa de la forma siguiente:

```
\begin{figure}
\includegraphics[scale=0.6]{images/tomate.png}
\caption{Imagen vectorial de un tomate}
\label{etiqueta-tomate}
\end{figure}
```

Posteriormente podemos referirnos a esa imagen usando un referencia cruzada:

Como podemos ver en la figura 1 de la página 13, los tomates son de color tomate.

## 5 Otros paquetes gráficos

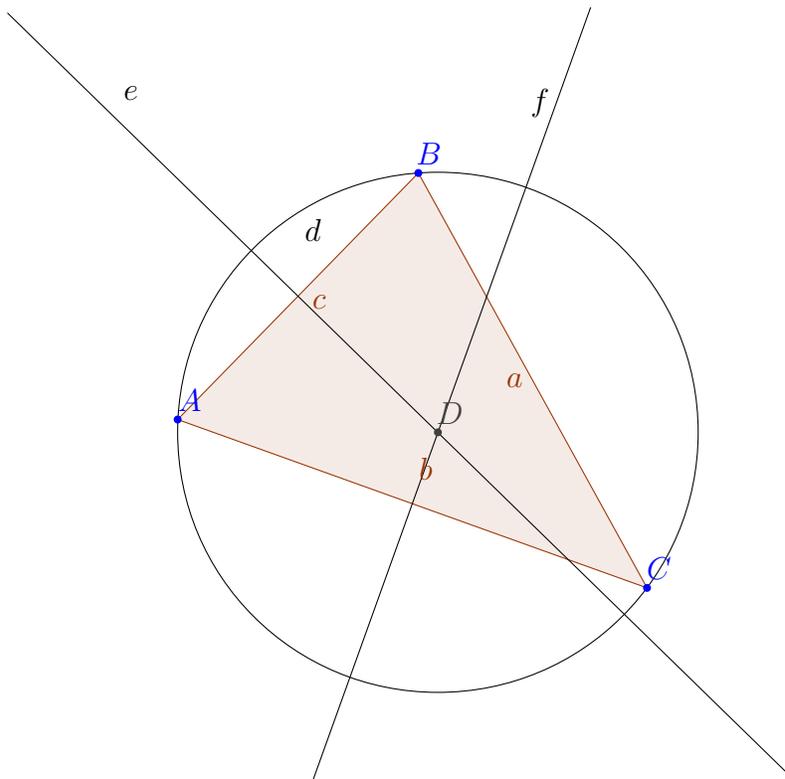
Existen muchos paquetes gráficos con los que se puede hacer de todo. podemos considerarlos apropiados para usuarios avanzados de  $\text{\LaTeX}$  (no podemos meterlos en un curso de introducción a  $\text{\LaTeX}$ ). Algunos de los más usados son `xymatrix`, `pstricks`, `pgf`, `tiz`, etc.

No obstante veamos un par de ejemplos:

El siguiente código

```
\definecolor{uququq}{rgb}{0.25,0.25,0.25}
\definecolor{zzttqq}{rgb}{0.6,0.2,0}
\definecolor{qqqqff}{rgb}{0,0,1}
\begin{tikzpicture}[line cap=round,line join=round,>=triangle 45,x=1.0cm,y=1.0cm]
\clip(-4.04,-2.16) rectangle (8.62,8.1);
\fill[color=zzttqq,fill=zzttqq,fill opacity=0.1] (-1.78,2.62) -- (1.42,5.9) -- (4.46,0.38) -- (-1.78,2.62);
\draw [color=zzttqq] (-1.78,2.62)-- (1.42,5.9);
\draw [color=zzttqq] (1.42,5.9)-- (4.46,0.38);
\draw [color=zzttqq] (4.46,0.38)-- (-1.78,2.62);
\draw [domain=-4.04:8.62] plot (\x, {(-13.4--3.2*\x)/-3.28});
\draw [domain=-4.04:8.62] plot (\x, {(-5-6.24*\x)/-2.24});
\draw(1.68,2.45) circle (3.46cm);
\fill [color=qqqqff] (-1.78,2.62) circle (1.5pt);
\draw[color=qqqqff] (-1.62,2.88) node {$A$};
\fill [color=qqqqff] (1.42,5.9) circle (1.5pt);
\draw[color=qqqqff] (1.56,6.16) node {$B$};
\fill [color=qqqqff] (4.46,0.38) circle (1.5pt);
\draw[color=qqqqff] (4.62,0.64) node {$C$};
\draw[color=zzttqq] (0.1,4.18) node {$c$};
\draw[color=zzttqq] (2.7,3.14) node {$a$};
\draw[color=zzttqq] (1.52,1.96) node {$b$};
\draw[color=black] (-2.4,6.96) node {$e$};
\draw[color=black] (3.04,6.84) node {$f$};
\fill [color=uququq] (1.68,2.45) circle (1.5pt);
\draw[color=uququq] (1.84,2.7) node {$D$};
\draw[color=black] (0.02,5.14) node {$d$};
\end{tikzpicture}
```

produce la siguiente imagen:



En un segundo ejemplo podemos ver una estructura molecular (demandada por un compañero en uno de los foros del curso)

```
\begin{tikzpicture}[level distance=4em , ultra thick]
\node {C}
[grow=down]
child [grow=up] {node {H}}
child [grow=left] {node {H}}
child [grow=down] {node {H}}
child [grow=right]
{node {C}
child[grow=up] {node {H}}
child[grow=right]{node {H}}
child[grow=down] {node {H}}
edge from parent[double distance=3pt]
coordinate (wrong)
};
\end{tikzpicture}
```

