

1. Iniciar una sesión de trabajo en GNU-Linux.
2. Abra una terminal.
3. Muestre el árbol de directorios de su HOME (`tree`).
4. Sitúese en la **Carpeta de Proyecto** de la asignatura Técnicas Experimentales esto es en el directorio *TE* (`cd TE`).
5. Muestre el contenido del directorio actual (`ls -la`).

Los temas de trabajo en equipo son:

- Grupo de prácticas 1 (miércoles 12:00 - 14:00): Búsqueda de raíces
- Grupo de prácticas 2 (viernes 8:30 - 10:30) : Series numéricas
- Grupo de prácticas 3 (jueves 11:00 - 13:00) : Cálculo de probabilidades

Cada equipo ha de elegir un **Coordinador del Equipo de Trabajo**.

El **Nombre del Equipo de Trabajo** será:

EQUIPO_*j*_*k*

donde:

- *j* puede tomar los valores 1,2 o 3, dependiendo del grupo de prácticas.
- *k* puede tomar los valores en el rango A..J, dependiendo del equipo.

En los ejercicios siguientes el coordinador creará la estructura del ‘directorio de trabajo del equipo’. Transformará el directorio de trabajo local en un ‘repositorio git’. Creará un ‘repositorio en *github*’ con el nombre del equipo. Dará permisos de escritura en el repositorio de ‘*github*’ a todos los miembros del equipo y al profesor de prácticas. Los miembros del equipo ‘clonarán’ el repositorio de *github*. Todos los miembros del equipo, realizarán un cambio y lo incorporarán al repositorio compartido.

El **Coordinador del Equipo de Trabajo** ha de realizar los ejercicios del 6 al 23.

6. Cree un nuevo directorio con el **Nombre del Equipo de Trabajo** (`mkdir`). Este será el **directorio de trabajo** durante la realización de esta práctica.
7. Sitúese en el directorio de trabajo y cree la estructura de directorios que le permita tener subcarpetas para el código y los documentos, es decir:
 - Un subdirectorio *src*
 - Un subdirectorio *docs*

8. Cree el fichero `README.md` en el directorio de trabajo. Este fichero ha de contener:
 - El nombre del equipo.
 - El tema del trabajo.
 - El caso de estudio.
 - El nombre del coordinador y el de los miembros del equipo.
9. Cree un fichero `solucion.py` en el directorio `src`. Este fichero ha de contener el código Python que resuelve el problema que se ha asignado al equipo.
(`echo "#! /usr/bin/python" > src/solucion.py`)
10. Descargue del Aula Virtual de la asignatura el fichero `estiloTE.tgz` en el directorio `docs` y descomprímalo. Este directorio ha de contener el código `LATEX` con el informe final y las transparencias del tema que se ha asignado al equipo.
(`tar -zxvf estiloTE.tgz`)
11. Situado en el directorio de trabajo, inicialícelo para que sea un repositorio `git`.
(`git init`)
12. Compruebe que se crea el directorio `.git` (`ls -la`).
13. Compruebe el estado del repositorio `git` local (`git status`).
14. Añada todos los ficheros y subdirectorios del directorio actual al *índice del repositorio git*.
(`git add .`)
15. Confirme (*commit*) los cambios del índice en el repositorio `git` local.
(`git commit -m "Creando el proyecto del trabajo en equipo"`)
16. Muestre el fichero con los registros realizados en el repositorio hasta el momento. (`git log`)
17. Compruebe el estado del repositorio `git` local (`git status`).
18. Cree un repositorio en *GitHub*
 - a) En la barra de usuario, en la esquina superior derecha de la página, haga clic en el icono de “Crear un repositorio nuevo” (*Create a New Repo*).
 - b) Introduzca el **Nombre del equipo de trabajo**.
 - c) Seleccione que quiere hacer el repositorio público.
 - d) **No** seleccione la casilla de crear el fichero `README.md`.
 - e) Pulse el botón para crear el repositorio (*Create repository*)
19. Cree un enlace al repositorio remoto con nombre corto *origin*
(`git remote add origin git@github.com:XXX/YYY.git`)

20. Empuje los cambios en el repositorio remoto denominado *origin*.
(`git push -u origin master`)
21. Muestre los detalles del repositorio remoto denominado *origin*. (`git remote show origin`)
22. Muestre los repositorios remotos que están definidos. (`git remote -v`)
23. Añada como colaboradores al resto de miembros del equipo en el repositorio *GitHub*
- a) Abra en el navegador el repositorio *Github* del trabajo en equipo.
 - b) En la el marco de la derecha, seleccione el menú de configuración del repositorio “*Settings*”.
 - c) En la el marco de la izquierda, seleccione del menú de opciones “*Collaborators*”.
 - d) En el campo de texto que aparece, introduzca el nombre de usuario de un compañero de equipo: `alu01000YYYYY`.
 - e) Pulse el botón para añadir al colaborador (*Add*)
 - f) Repita los pasos 23d) y 23e) por cada miembro del equipo y su profesor de prácticas.

Los **Miembros del Equipo** han de realizar los ejercicios del 24 al 27.

24. Sitúese en el directorio *TE* y clone el repositorio remoto que ha creado el coordinador del equipo.
`$git clone git@github.com:alu01000XXX/YYYY.git`
25. Compruebe que tiene la misma estructura de directorios y ficheros que ha creado el coordinador del equipo. (`ls -la`)
26. Muestre los detalles del repositorio remoto denominado *origin*. (`git remote show origin`)
27. Muestre los repositorios remotos que están definidos. (`git remote -v`)

Los siguientes ejercicios los han de realizar todos los miembros del equipo.

Se ha de llegar a un acuerdo sobre quién realiza cada tarea y quién la guarda en el repositorio remoto para evitar **conflictos** de versiones con *git*.

Cuando se le indique que copie un texto, NO lo corte para luego pegarlo en el editor, porque se añaden caracteres extraños e invisibles que le impedirán compilar.

28. Con el editor que prefiera añada el siguiente texto en el fichero denominado `cap1.tex` situado en el directorio *tex* (sin incluir el número de línea):

```

1 Si simplemente se desea escribir texto normal en LaTeX,
2 sin complicadas fórmulas matemáticas o efectos especiales
3 como cambios de fuente, entonces simplemente tiene que escribir
4 en español normalmente.
5 Si desea cambiar de párrafo ha de dejar una línea en blanco o bien
6 utilizar el comando \par
7 No es necesario preocuparse de la sangría de los párrafos:
8 todos los párrafos se sangrarán automáticamente con la excepción
9 del primer párrafo de una sección.
10 Se ha de distinguir entre la comilla simple 'izquierda'
11 y la comilla simple 'derecha' cuando se escribe en el ordenador.
12 En el caso de que se quieran utilizar comillas dobles se han de
13 escribir dos caracteres 'comilla simple' seguidos, esto es,
14 ''comillas dobles''.
15 También se ha de tener cuidado con los guiones: se utiliza un único
16 guión para la separación de sílabas, mientras que se utilizan
17 tres guiones seguidos para producir un guión de los que se usan
18 como signo de puntuación --- como en esta oración.

```

29. Compile el fichero *memte.tex* (`make`).
30. Visualice el fichero *memte.pdf* (`okular memte.pdf`).
31. Añada todos los ficheros y subdirectorios del directorio *mem* al *índice del repositorio git*.
(`git add .`)
32. Registre los cambios del índice en el repositorio `git` local.
(`git commit -m "Motivacion y objetivos"`)
33. ¿Qué ocurre con los párrafos? Utilizando la instrucción $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ `\par` separe los dos primeros párrafos. Separe el resto de párrafos usando líneas en blanco.
34. Compile el fichero *memte.tex* y visualice el fichero *memte.pdf*.
35. Muestre el estado del repositorio local (`git status`).
36. Muestre las diferencias entre los ficheros sin registrar y los del último registro (`git diff`).
37. Registre los cambios en el repositorio. (`git commit -a -m "Parrafos en capitulo 1"`)
38. Muestre la historia de los distintos registros en el repositorio (`git log`).
39. Añada el siguiente texto en el fichero denominado *cap2.tex* situado en el directorio *tex* (sin incluir el número de línea):

```

1 En \LaTeX{}~\cite[Lam:86] es sencillo escribir expresiones
2 matemáticas como  $\sum_{i=1}^{10} x_i^3$ 
3 y deben ser escritas entre dos símbolos  $\$$ .
4 Los superíndices se obtienen con el símbolo  $\{$ , y
5 los subíndices con el símbolo  $\_$ .
6 Por ejemplo:  $x^2 \times y^{\alpha + \beta}$ .
7 También se pueden escribir fórmulas centradas:
8 
$$h^2 = a^2 + b^2$$


```

40. Compile el fichero *memte.tex*. ¿Qué indica el compilador? ¿Un error o un aviso?
41. Visualice el fichero *memte.pdf*. ¿Qué ocurre con la cita de la línea 1?
42. Sustituya la cita de la línea 1 por la etiqueta del primer libro que aparece en el fichero de bibliografía *references.bib*.
43. Compile el fichero *memte.tex*. ¿Qué indica el compilador?
44. Compile de nuevo el fichero *memte.tex*. ¿Qué indica el compilador? ¿Cuántas veces es necesario compilar para resolver las citas y las referencias?
45. Visualice el fichero *memte.pdf*. ¿Qué ocurre con la cita de la línea 1?
46. Registre los cambios en el repositorio. (`git commit -a -m "Citas a la bibliografía"`)
47. Muestre la historia de los distintos registros en el repositorio (`git log`).
48. Añada el siguiente texto en el fichero denominado `table.tex` situado en el directorio *tables* (sin incluir el número de línea):

```

1 \begin{table}[!ht]
2 \begin{center}
3 \begin{tabular}{|l|c|c|}
4 \hline
5 Nombre & Edad & Nota \\ \hline
6 Pepe & 24 & 10 \\ \hline
7 Juan & 19 & 8 \\ \hline
8 Luis & 21 & 9 \\ \hline
9 \end{tabular}
10 \end{center}
11 \caption{Mi primer cuadro de datos}
12 \label{tab}
13 \end{table}

```

49. Compile el fichero *memte.tex* y visualice el fichero *memte.pdf*.
50. Muestre el estado del repositorio local (`git status`).
51. Muestre las diferencias entre los ficheros sin registrar y los del último registro (`git diff`).
52. Registre los cambios en el repositorio. (`git commit -a -m "Tabla"`)
53. Muestre la historia de los distintos registros en el repositorio (`git log`).
54. Guarde en el directorio *images*, el fichero *imagen1.eps* disponible en el aula virtual.
55. Añada el siguiente texto en el fichero denominado `cap3.tex` situado en el directorio *tex* (sin incluir el número de línea):

```

1 \begin{figure}[!th]
2 \begin{center}
3 \includegraphics[width=0.75\textwidth]{images/imagen1.eps}
4 \caption{Ejemplo de figura con gráfico}
5 \label{fig}
6 \end{center}
7 \end{figure}

```

56. Compile el fichero *memte.tex* y visualice el fichero *memte.pdf*.
57. Muestre el estado del repositorio local (`git status`).
58. Muestre las diferencias entre los ficheros sin registrar y los del último registro (`git diff`).
59. Registre los cambios en el repositorio. (`git commit -a -m "Gráfico"`)
60. Muestre la historia de los distintos registros en el repositorio (`git log`).
61. Todos los miembros del equipo deben subir la dirección del repositorio github en la tarea habilitada en el campus virtual.
62. Cierre la sesión.