03. Mod1- Uso de terminarl y git

ls - Listar directorio de donde estas
pwd - en que carpeta estas

Navegar entre carpetas:
1. Sabiendome la ruta
2. cd ~ - Volver a la carpeta raíz
3. cd (espacio) - Arrastramos la carpeta a la terminal

Como se crean archivos o carpetas:
mkdir css <- nombre de la carpeta que quieres crear

Para entrar en la carpeta:
cd (espacio) c(tabulador - te completa el nombre de la carpeta)

Para subir carpetas: cd..
Para limpiar consola: clear o cls
Crear achivos: touch index.html <- nombre del archivo
Borrar archivos: rm index.html <-nombre del archivo
Borrar carpetas: rmdir - Pero solo si no tiene contenido
Forzar borrado: rm -rf css <- nombre del archivo a Borrar
Abrir vscode: code .

GitKraken

Crear un repositorio:
File - Init repo - Local Only
Name: Prueba1Git
Initialize in (ruta de la carpeta)
Default branc name: main
"Si cambiásemos de nombre a la carpeta, solo hay que buscar la ruta de nuevo: File - Open repo - Open repository y buscamos la carpeta"

Nos vamos a vscode, abrimos la carpeta y creamos el documento de texto datos.texto

Nos vamos a GitKraken y vemos que tenemos un commit message y aparece un +1

Le damos al +1
En la ventana de la derecha (commits atomicos - puede haber 1 o 100 - y no entran dentro del control de versiones) le damos a "Stage all shanges" para que entren dentro del control de versiones y abajo escribimos el nombre del commit y su descripción.

El fichero está comiteado

Volvemos a vscode y seguimos completando el archivo - guardamos y volvemos a GitKraken

Aparece un icono de un lápiz en vez del + ya que solo has modificado el archivo

Pinchamos y en los commits atomicos está el archivo modificado, que, si lo abrimos, aparecerá en verde lo último que has modificado. (Los cambios de borrado aparecen en rojo)

Commiteamos el archivo modificado
Volvemos a vscode y añadimos una nueva línea
Commiteamos el archivo modificado

Imagina que nos hemos equivocado, lo que hay que hacer es volver tantos puntos atrás como necesitemos.

Nos vamos al punto anterior del que queremos modificar, botón derecho, Reset main to this commit,

Soft - keep all changes
Sirve para cambiar el commit message
Se va air al commit antrior, va a dejar el dato en la zona de stage (justo antes de crear el commit)

Mixed - keep working copy but reset index
datos.txt aparece en el commit atomico, justo antes de commitear. Lo utilizas cuando sabes donde tienes el error y solo tienes que cambiarlo en vscode.

Crear una rama nueva:

Se usa para no tocar en producción "main" lo que queremos arreglar y cuando arreglemos el problema, podremos volverlo a main.

En el último commmit del que vamos a partir, le damos al botón de Branch, se abre un cuadro de texto "entrer branch name" y hay que poner:
Feature + nombre (por ejemplo, feature-curriculum)

Las ramas de nueva funcionalidad: Feature
Solución de problemas: Hotfifix

Una vez puesto el nombre, le damos a intro y en la aprte de local debería de aparecernos la rama que has creado
(Para saber en qué rama estoy, en local, aparece en verde, todos los cambios que se produzcan a partir de ahora posicionados con la rama marcada en verde, se producirán en dicha rama)

Me cambio de rama, dentro de local, haciendo doble click en la rama a la que queremos ir.

Vamos a vscode a la carpeta del proyecto. Como podemos ver, si en GitKraken estamos en feature-feature_curriculum, en vscode, abajo a la izquierda (aparece el nombre de la rama donde nos encontramos)

Creamos un nuevo documento txt (curriculum.txt)

Ponemos: Educación: y Trabajos: y guardamos.
Vuelvo a GitKraken y lo commiteo
Y como vemos, se posiciona por encima del main que quiere decir que feature-curriculum tiene un commit de diferencia.
Volvemos a vscode al documento curriculum.txt, lo completamos y guardamos.

Volvemos a vscode al documento curriculum.txt, lo completamos y guardamos.

Vamos a Gitkraken y Commiteamos

La rama feature tiene dos commits de diferencia respecto a la rama main.

Suponemos que hemos acabado nuestra tarea, asi que hay que pasar el curriculum.txt de la rama feature-curriculum a la rama main.

Primero: Situarse en la rama de destino, en este caso main.
Segundo: situarte en el commit con los últimos cambios que queremos trasladar. Tercero: Botón derecho, Merch feature_curriculum into main. Ahora main está por enci-ma de feature-curriculum.
Cuarto: Borrar la rama feature-curriculum:
- En local, nos posicionamos sobre la rama main (en verde), botón derecho sobre el feature que queremos borrar y delete feature_curriculum
- Si, por el contrario, te encuentras en la rama feature-curriculum (en verde) y le das a borrar, te saldrá una alerta como que no puedes borrarla.

Imaginemos que hemos cometido un error, en este caso faltan las tildes, pero aún no lo sabemos y seguimos con las tareas.

Tarea nueva: Hacer una Lista

Nos posicionamos en el último commit y creamos una rama con el nombre feature_lista-tareas y en local, nos posicionamos en la nueva rama(en verde)

Vamos a vscode y creamos un fichero nuevo: listatareas.txt

Volvemos a GitKraken y commiteamos

Pero recordamos que en el fichero anterior había un problema (las tildes)

Así que vamos a GitKraken, nos posicionamos en main y volvemos a vscode para realizar el cambio (recuerda cerrar las carpetas en las que estabas trabajando en la feature, en este caso feature_listatareas, haberlos guardado y commiteado para no perderlos)

Así que me voy a GitKraken, me coloco en main que es donde está el problema, y creo una rama, en este caso un hotfix: hotfix_tildes.

Volvemos a vscode y solucionamos el problema y guardamos.

Volvemos a GitKraken, commiteamos y lo volcamos a main: nos vamos a la rama main, botón derecho en el commit, Merge hotfix_tildes into main, pero en la tarea nueva "fea-ture_listatareas" no se ha realizado el cambio (Cuando creamos un hotfix hay que vol-carlo en todas las ramas) Así que me coloco en la feature_listatareas, pongo el ratón sobre el hotfix y botón derecho, Merge hotfix_tildes into feature_listatareas y ya si puedo borrar el hotfix_tildes.

Volvemos a feature_listatareas en vscode y añadimos más campos. Volvemos a Gi-tKraken y commiteamos

Ahora creamos una nueva rama, nos posicionamos en mail y le damos a Branch y creamos feature_contactos.

Nos vamos a vscode y creamos el archivo de texto contactos.txt, volvemos a GitKraken y commiteamos.

Recordamos que lista tareas está terminada, así que la volcamos en main (nos posicio-namos en main(verde), me posiciono sobre el commit que quiero volcar, botón derecho Merch...) y borramos el feature listatareas.

Volvemos a posicionarnos en feature_contactos(verde)y volvemos a vscodepara termi-nar el documento, volvemos a GitKraken y commiteamos. Lo volcamos a main y borra-mos la rama feature_contactos.

Creación de carpeta, pero con línea de comandos:

Abrimos consola
Nos vamos a la carpeta: cd (y arrastramos la carpeta)
Nos indica que estamos en la rama main:
delic@DESKTOP-81KKPKM MINGW64 ~
$ cd "C:\\Users\\delic\\Desktop\\UNIR\\03. Mod1- Uso de terminarl y git\\Prueba1Git"

Creamos una nueva rama:
$ git branch feature_softskills

Para cambiar de rama:
$ git checkout feature_softskills

Nos vamos a vscode, a curriculum, añadimos las softskills y guardamos.

Tenemos un commit, así que hay que pasar de la zona de stage a la zona de unstage:
$ git add .

Ahora hay que hacer el commit: git commit -m "soft skills añadidas" <- texto del mensaje:
$ git commit -m "soft skills añadidas"
[feature_softskills 9e50cfe] soft skills añadidas
1 file changed, 5 insertions(+), 1 deletion(-)

Ver el árbol en la terminal: Se abre un terminal de texto y te enseña los commits:
$ git log

Pra salir del ($ git log) ponemos :q

Para resetear: $ git reset --hard 9e50cfe <- Numero del commit (id):
$ git reset --hard 9e50cfe
HEAD está ahora en 9e50cfe Merge Brand feature_contactos into main.

Volvemos a vscode y volvemos a añadir las softskills y guardamos.

Volvemos a hacer
$ git add .

$ git commit -m “soft skills añadidas”

Volcamos:
Cambiamos a la rama destino (main)
$ git checkout main

Volcamos el contenido de feature_softskills:
$ git merge feature_softskills

Borramos la rama feature_softskills: git branch (--delete o -d) feature_softskills
$ git branch --delete feature_softskills
Eliminada la rama feature_softskills

Stage line ----------

Hacer un Stage line de una parte del fichero (tienen que ser líneas conti-nuas, no separadas):
Entramos en el archivo desde GitKraken, antes de comitear, pulsamos en el archivo, nos aparecerá en verde lo que hemos añadido, y seleccionamos la parte a la que le queremos hacer el Stage File y botón derecho, Stage this line y se quedará una parte del fichero arriba y otra bajará para commitear.
Ej: Imagina que se acaba el día y no quiero comitearlo todo, así que elijo las partes del fichero que funcionan y las comiteo. Y el resto lo dejo.

Clase 3 - Repositorio remoto

Creamos otro proyecto: Otra carpeta: Prueba3Git

Abrimos GitKraken y terminal

Desde la terminal:

Me sitúo en la carpeta de trabajo:
$ cd (ruta de la carpeta arrastrando la carpeta)

Inicia un repositorio en esa carpeta
$ git init

Vamos a GitKraken y abrimos repositorio:
Open repository - Open a repository
Seleccionamos la carpeta y en el mensaje de alerta le damos a cancelar.

Creamos el commit inicial desde la terminal:

Abrimos vscode
$ code .

Creamos un index.htm y vamos a la terminal:
$ git status

No hay nada agregado al commit pero hay archivos sin seguimiento
(usa git add para hacerles el seguimiento.

$ git add .

$ git commit -m "git init proyect"

Vamos a GitKraken y abrimos el repositorio

Vamos al navegador y abrimos github y creamos nuestro repositorio remoto.
Para ello vamos al perfil - youtr repositories - y new y creamos un nuevo repositorio.
Lo nombramos igual que en local: Prueba3Git y create repository

Vamos a GitKraken y en Remote, add remote por url.
Pegamos la url que nos ha generado github y le ponemos el nombre origin y pulsamos add remote.

Ahora tendremos origin en remote y main en local

Ahora tenemos que subir la rama main a github por lo que hay que hacer un push, y le decimos que queremos subir main. y le damos a submit.

* Si al crear el main por la consola te lo crea con el nombre master, solo hay que cambiarlo desde GitKraken y hacer push. Desde la terminal:
$ git branch -m (main) <-- nombre al que queremos cambiar

Vamos a vscode y en el index.html creamos un html plano y una carpeta de css con un documento style.css. Guardamos y hacemos un commit.

Para unir la rama local y la remota hacemos un push.
.
Desde la terminal:

Lo subimos
$ git add .

Lo commiteamos
$ git commit -m "header iniciado"

Lo subimos al repositorio remoto
$ git push origin main

Simular un error:

Vamos a Github, abrimos el css, le ponemos un estilo al body y commiteamos los cam-bios.

```
body{
background-color: tomato;
font-size: 1em;
}
```

Volvemos a Gitkraken y hacemos un fetch (Hace una comprobación inicial, de si hay cambios en el servidor remoto - siempre hay que hacer un fetch antes de hacer un pull)

Si los cambios me convencen, me los descargo con pull.

Abrimos vscode para crear el conflicto y en el css escribimos una línea de codigo:

```
font-weight: bold;
```

Ahora volvemos a github y escribimos en el css una nueva línea y commiteamos:

```
line-height: 1.5px;
```

Hacemos un fetch para comprobar y un pull, te saldrá una alerta de conflicto.
Vamos a vscode y aparecen dos cuadros de texto: El que hay en remoto y el que tú has hecho en local. Ya solo tienes que decidir con cual te quedas o si aceptas los dos.

Aceptamos los dos cambios y nos vamos a GitKraken y hacemos un commit que se ha resuelto solo en local.

```
Mark all resolved
```

Si funciona correctamente le damos a push

Normalmente, en un flujo de trabajo hay varias ramas para trabajar. La rama main es la rama de producción y la rama devlope es donde vuelcan los desarrolladores sus features(cada tarea que tiene el desarrollador) o hotfix(corrección de errores)

Clase 3 - Repositorio remoto - Clonar repositorio

Desde la terminal:

Creamos una nueva carpeta: Prueba4Git y en la terminal llamamos a la carpeta:
$ cd "C:\\Users\\delic\\Desktop\\UNIR\\03. Mod1- Uso de terminarl y git\\Prueba4git"

Y hacemos un git clone de la anterior, para ello debemos ir a github, abrir el repositorio que queremos clonar y en el boton <>code copiamos la url. Recuerda, en la terminal se pegan las cosas con el botón derecho.

$ git clone https://github.com/AnaNunezRejon/Prueba3Git.git

Desde GitKraken: Clone repo, por url y elegimos la carpeta donde queremos clonarlo

---
Vamos a seguir con Prueba3Git para hacer nuestra primera actividad, así que abrimos el repositorio desde GitKraken.

Nos vamos a vscode y creamos el sistema de carpetas: js e images y añadimos una imagen a esa carpeta.

Para que no nos pasemos con el peso por si las imágenes pesan mucho, lo normal es ig-norar ciertos archivos para que no se suban al repositorio y compartirlos con el resto de desarrolladores (googedrive - metadatos - cdn)

La forma de hacerlo es la siguiente: nos vamos a vscode y a la altura el index.html creamos un archivo .gitignore y le decimos que ignore las imágenes escribiendo en el código o escribimos uno por uno los ficheros/archivos que queremos ignorar o si vamos a ignorar todas las imágenes, con poner images es suficiente. Si queremos ignorar todas las imá-ge-nes por extensión: (images/*.jpg) o (images/*.png)

Commiteamos el cambio y alineamos las ramas con un push.

Pero hemos ignorado las imágenes en la rama main y debemos ignorarla en la develop

Así que sobre el commit anterior, le damos a Reset - Mixed. Nos vamos a vscode y borramos el archivo .gitignore. Volvemos GitKraken, commiteamos, hacemos un push y forzamos.

Creamos una nueva branch con nombre develop y en la rama develop (ojo que este marcada en verde o que aparezca en la zona inferior izquierda del panel de vscode) creamos el .gitignore eliminando las imágenes (images) y hacemos el commit.

Todos los desarrolladores crean las ramas desde develop. Los hotfix se crean en main (porque son corrección de errores) y luego se vuelcan a develop.

Creamos una nueva branch sobre develop con el nombre feature-home y nos vamos a vscode a crear nuestros archivos y commiteamos.

Una vez todo creado todo y commiteado. Lo volcamos todo a develope.

Ahora nos encontramos en la rama feature-home por lo que primero tene-mos que pa-sarnos a la rama develop y luego hacemos un merch, nos siuamos en el último commit y botón derecho: Merche feature-home into develop. Borramos la feature. Subimos deve-lop a remote con un push.

De momento no podemos subir a main porque aún tenemos que crear la pestaña de contactos. Así que desde develop creamos una nueva feature-contacto

Desde la terminal:

Primero me cambio a la rama develop si no estoy en ella:
$ git checkout develop

Ahora creamos el nuevo feature:
$ git branch feature-contacto

Y nos cambiamos a la rama creada:
$ git checkout feature-contacto

Vamos a vscode, creamos el contacto.html y commiteamos. Cuando este todo bien:

Vamos a develop desde la terminal:
$ git checkout develop

Hacemos un merch de feature-contacto
$ git merge feature-contacto

Y un git push:
$ git push origin develop

Borramos feature-contacto:
$ git branch -d feature-contacto

Lo hacemos desde GitKraken

1. Hacemos un merch en local. Nos vamos a main y hacemos un merch de develop a main.
2. Y hacemos push (recuerda hacer un fech antes de hacer un puch)

Ahora nos vamos a github a la actividad, cogemos la url de la página y eso sería lo que entregaríamos.