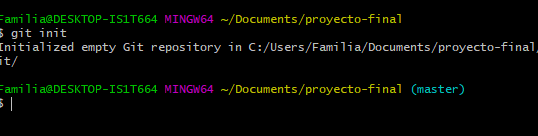
Objetivo principal: Aplicar los comandos necesarios para el control de versiones en el árbol de movimientos en un proyecto administrado por la herramienta GIT.

Siga las instrucciones detenidamente, digite el comando empleado según se le solicite.

1. Cree una carpeta con el nombre proyecto-final en el escritorio
2. Copie y pegue el contenido de la carpeta capturelife en proyecto-final
3. Cree un proyecto git en dicha carpeta

**Comando empleado:**



1. Agregue los archivos al stage y realice el primer commit con el mensaje “Sitio web original”.
2. Agregue solamente la carpeta images a la carpeta proyecto-final, que está en la carpeta recursos, y seguidamente agregue al *stage* solamente la carpeta mediante el comando.

**Comando empleado:**

1. Realice un commit con el mensaje “Se agregaron las imágenes al proyecto”.
2. Cree un archivo llamado README.md, y digite en él:

#Notas

Proyecto realizado para la empresa capturelife.

1. Agregue al stage el archivo y realice un commit con el mensaje “Archivo README.md creado”.
2. Después del commit, digite en el archivo README.md, la siguiente descripción:

Proyecto de un sitio web para fotógrafos.

No realice ningún commit, guarde el documento solamente

1. Retrocede a los últimos cambios del commit con “checkout”.

**Comando empleado:**

git checkout - -.

1. Realice un log para visualizar los commits.

**Comando empleado:**

1. Agregue el código que se encuentra en el bloc de notas de la carpeta recursos llamado: **menu**, dentro de todas las páginas del sitio web en la sección señalada entre comentarios como

<!- - menu -- > <!- - /menu - - >

1. Agregue el código del menú, verifique que se pueda visualizar en todas las páginas del sitio web.
2. Verifique el estado del proyecto, agregue al stage los cambios y realice un commit con el siguiente mensaje (tal como está): “Se agregó el menú en…”
3. Utilice el comando apropiado para corregir el mensaje anterior, con el siguiente mensaje “Se agregó el menú en todas las páginas del sitio”.

**Comando empleado:**

1. Agregar la carpeta icon-fonts dentro de la carpeta proyecto-final
2. Comprobar su estado

**Comando empleado:**

1. Agregar el formulario de contacto que se encuentra en el bloc de notas llamado **form** dentro de la carpeta recursos, en la página contact.html en la sección comentada:

<!-- contact -->

<!-- /contact -->

1. Aplicar un comando que permita agregar al stage y un commit a la vez con el siguiente mensaje: “Formulario de contacto listo”.

**Comando empleado:**

1. Aplicar un comando para eliminar el último commit pero no el código.

**Comando empleado:**

1. Realizar una modificación al formulario de contacto cambiar el color el botón “SEND MESSAGE” por #df2b2b.
2. Agregar al stage el cambio y aplicar un commit con el mensaje del “Formulario de contacto listo”.
3. Aplicar un comando para eliminar todos los cambios desde el código hasta los commits a partir del segundo commit es decir “Se agregaron a las imágenes al proyecto”.

**Comando empleado:**

1. Verificar para ver si se aplicó correctamente los cambios.
2. Recuperar el proyecto hasta el último commit, es decir: “Formulario de contacto listo”.

**Comando empleado:**

1. Crear una rama llamada colores-oscuros.

**Comando empleado:**

1. Moverse a la rama colores-oscuros.

**Comando empleado:**

1. Modificar los colores del sitio web en la rama colores-oscuros, en las siguientes secciones:
2. Botones
3. Footer
4. Copyright
5. Iconos de servicios en el index.html
6. Iconos de Redes sociales
7. Enlaces

\*Colores a su preferencia

1. Guardar cambios, guardarlos a los stage y agregar el siguiente mensaje en el commit “Se realizaron ajustes en colores oscuros”.
2. Combinar la rama de colores oscuros con la master

**Comando empleado:**

1. Eliminar la rama colores oscuros.

**Comando empleado:**

1. Crear la rama llamada widgets-menu y moverse a ella a la vez.

**Comando empleado:**

**Agregar widgets en todas las páginas**

1. Dentro de la rama widgets-menu agregar el código que está en el bloc de notas widget-search, dentro de los comentarios

<! - - widget-search - - > <! - - /widget-search - - >

1. Agregar el cambio al stage y realizar un commit con el siguiente mensaje “Se agregó el widget de búsqueda”.
2. Dentro de la rama widgets-menu agregar el código que está en el bloc de notas widget-minipost, dentro de los comentarios

<! - - widget-minipost - - > <! - - /widget-minipost - - >

1. Agregar el cambio al stage y realizar un commit con el siguiente mensaje “Se agregó el widget de minipost”.
2. Los commits anteriores unirlos en un solo con el mensaje “Widgets del menu”.

**Comando empleado:**

1. Salirse de la rama e ir a la rama master
2. En la rama master unir los cambios de widgets-menu

**Comando empleado:**

1. Crear una página llamada login.html que contenga un formulario con usuario y contraseña.
2. Guardar en el stash

**Comando empleado:**

1. Realizar un cambio en contact.html, agregando un campo de texto llamado, last name
2. Agregar al stage y realizar un commit con el mensaje “Se agrega campo de texto, de emergencia en el formulario”.
3. Recuperar el login.html

**Comando empleado:**

1. Agregar al stage y realizar un commit con el mensaje “Se crea login.html”.
2. Eliminar la rama widgets-menu

**Comando empleado:**

1. Aplicar una versión 1.0 en este último commit.

**Comando empleado:**

**GIT HUB**

1. Crear un repositorio con el nombre proyecto-final

**Comando empleado:**

1. Subir todo el proyecto a GIT HUB

**Comando empleado:**

1. Generar un cambio desde GIT HUB, en el index.html, en lugar del texto “Hi, I’m Editorial by HTML5 UP” a “Capture Life”, genere un mensaje “Se realizó un cambio desde Github”
2. Bajar esos cambios a GIT repositorio local

**Comando empleado:**

**Fin del proyecto.**