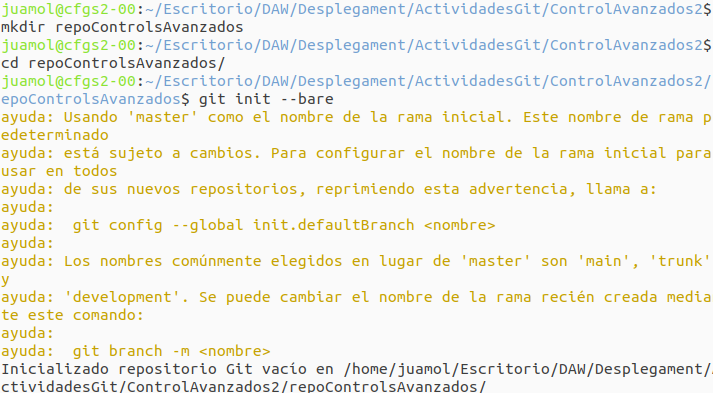
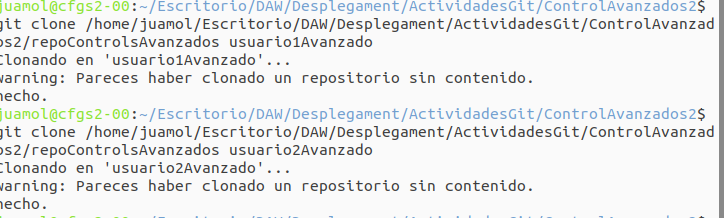
**Actividad Git 4**

**1. Crea un repositorio en local con la opción --bare (repoControlsAvanzados).**

****

**2. Simula que dos usuarios están haciendo cambios sobre este mismo repositorio**

**realizando dos clonaciones en carpetas llamadas usuario1Avanzado y usuario2Avanzado.**

****

**3. El usuario 1 se encargará de subir inicialmente al repositorio el siguiente contenido:**

**a. Archivo README.md en el que se incluye un texto descriptivo cualquiera. Por ejemplo, "Repositorio sobre el que realizar ejercicios de controles avanzados de git".**

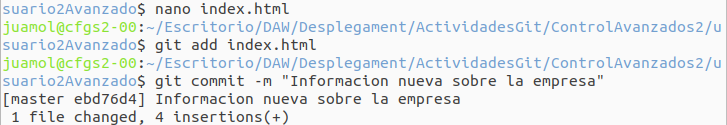
****

**b. Archivo index.html que contendrá una cabecera con dos enlaces que serán:**

**I. Inicio**

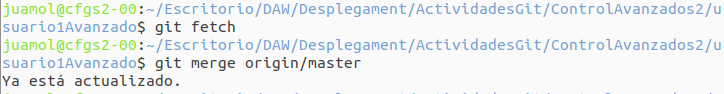
**II. Contacto**

**c. El segundo usuario modificará el archivo index.html para añadir información relativa a una empresa ficticia (por ejemplo, puede incluir un logotipo y un eslogan) y subirá los cambios al repositorio remoto.**

****

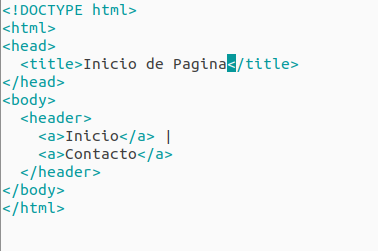
**nanod. El primer usuario creará el archivo contacte.html que contendrá información relativa a los datos de contacto de la empresa ficticia e intentará subir los cambios al repositorio remoto. Debería ser advertido en ese punto que no está actualizado.**

**e. Llegado a este punto, el primer usuario actualizará su repositorio con los cambios del segundo usuario utilizando la estrategia git fetch + git merge.**

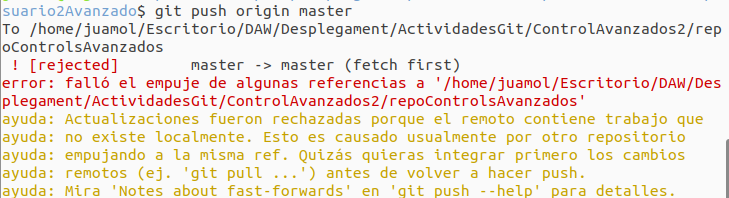
****

**4. Simula que hay conflictos a resolver en un archivo del proyecto. Por eso:**

**a. El primer usuario modificará el archivo index.html estableciendo el título de la página como "Inicio de la página" y subirá sus cambios al repositorio remoto.**

****

**b. El segundo usuario, sin haberse actualizado previamente el contenido del repositorio, realizará el mismo cambio que el anterior usuario. En este caso estableciendo el título de la página como "Inicio de Nombre\_empresa". Después de realizar el cambio, intentará subirlo al servidor remoto. En ese momento, git debería quejarse porque no puede resolver los conflictos. Resuélvelos manualmente y súbelos al servidor remoto.**

****

**5. Simula que se desea crear una nueva funcionalidad sobre la página que se está desarrollando. Por eso, el primer usuario creará una nueva rama partiendo de la rama master. Esta rama se llamará clientesPage\_branch y sobre ella se realizarán los cambios necesarios para poder añadir al proyecto un tercer enlace (titulado "Nuestros clientes") en la cabecera que muestre el contenido de la nueva página clients.html. Una vez terminada esta funcionalidad, realiza el merge de la rama creada a master para que los cambios estén en la rama principal del proyecto.**

**6. Etiqueta el proyecto con lo siguiente:**

**a. v1.0 -> Commit asociado a la funcionalidad de la página Inicio.**

**b. v2.0 -> Commit asociado a la funcionalidad de la página Contacto.**

**c. v3.0 -> Commit asociado a la funcionalidad de la página Clientes.**

**7. Subir las etiquetas creadas en el ejercicio anterior al servidor remoto.**