

# Construcción de software

## Control de versiones

**Instrucciones:** Desarrolle el tutorial consignado en el anexo.

### I. Propósito

Desarrolle el tutorial sobre control de versiones en Git y GitHub.

## ANEXO

### SISTEMA DE CONTROL DE VERSIONES

#### REPOSITORIOS REMOTOS - OPERACIONES BÁSICAS

##### Crear un repositorio local con git

1. Cree la carpeta /recetario en el directorio de su preferencia.
2. Desde Git Bash o la terminal ubíquese en la carpeta y ejecute el siguiente comando.

```
cd recetario
```

3. Cree un repositorio mediante el comando init.

```
git init
git config --global user.email "dgamarra@continental.edu.pe"
git config --global user.name "dgamarra"
```

##### Crear un repositorio remoto en GitHub

Para realizar el tutorial debe contar con una cuenta en GitHub. Realice los siguientes pasos:

1. Inicie sesión en GitHub.
2. Haga clic en el botón New.
3. Ingrese el nombre del repositorio: recetario. Seleccione el perfil de seguridad de acuerdo con el contexto del proyecto que va a trabajar y haga clic en el botón Create repository.

Con lo anterior habrá creado un repositorio remoto, ahora copie su URL para empezar a trabajar en él.

```
#Sincronizar el repositorio remoto
git remote add origin https://github.com/dgamarra/recetario.git
git branch -M main
git push -u origin main
```

## CICLO DE TRABAJO ENTRE EL REPOSITORIO LOCAL Y EL REPOSITORIO REMOTO

### PASO 1. AGREGAR LÍNEA BASE DEL PROYECTO

El punto de partida para este tutorial es un proyecto web que contiene diferentes recetas de cocina hispanoamericana. Lo que se hará a continuación será realizar cambios a ese proyecto desde el repositorio local y sincronizarlos con el repositorio remoto entendiendo la dinámica del ciclo de trabajo.

Realice los siguientes pasos:

1. Copie el contenido de la carpeta /recetario en el directorio de su repositorio local.
2. Desde la consola de Git Bash o la terminal ubíquese en la carpeta del proyecto. Consulte el estado del repositorio.

```
git status
```

Note que los archivos están en la copia de trabajo local, pero aún no han sido incluidos en el área de preparación.

3. Añada las carpetas de recetas e imágenes al área de preparación.

```
git add .
```

4. Consulte nuevamente el estado de su repositorio e identifique los cambios.

```
git status
```

Ahora los archivos han sido etiquetados como new file, es decir que se encuentran por primera vez en el área de preparación y están pendientes de ser enviados al repositorio.

5. Utilice el comando commit para añadir los archivos que están en el área de preparación al repositorio.

```
git commit -m "Se añade código base"
```

Como respuesta Git especifica cuántos archivos han sido cambiados y cuántas líneas de código han sido insertadas o eliminadas.

En este momento, ya cuenta con el código base del proyecto en el repositorio local y está listo para actualizar el repositorio remoto.

6. Sincronice los cambios con el repositorio remoto.

```
git push -u origin main
```

La opción -u mantiene un registro de la rama a la que se hace push, lo anterior con el fin de evitar conflictos o ausencia de información al hacer pull.

Al ejecutar el comando git push por primera vez, es necesaria la autenticación en GitHub con el usuario y contraseña definidos al crear la cuenta.

Después de autenticarse, Git sincroniza todo lo existente en el repositorio local con el repositorio remoto.

Note que se obtiene como resultado el porcentaje de archivos enviados y no enviados. Además de las ramas de origen y destino.

7. Verifique los cambios en GitHub.

Ahora el código base del proyecto se encuentra en el repositorio remoto y es posible visualizarlo, navegar en él y obtener información relevante de todo el historial desde cualquier parte del mundo.

Deténgase un poco para navegar en las diferentes opciones de GitHub.

## PASO 2. MODIFICAR ARCHIVOS DEL REPOSITORIO

Siga los siguientes pasos para modificar una receta del repositorio local:

1. Se ha identificado que la imagen en la receta Arroz con coco y camarón fresco (Colombia) no corresponde al plato.

Ubíquese en el siguiente directorio desde su copia de trabajo local: /recetas/Colombia/arroz\_coco.html y corrija la ruta de la imagen por: ../../imagenes/arroz-coco.jpg

2. Consulte el estado del repositorio.

```
git status
```

Ahora el archivo `/recetas/Colombia/arroz_coco.html` ha sido etiquetado como `modified` (en color rojo), es decir que Git identificó cambios en el archivo de la copia de trabajo local que aún no han sido movidos al área de preparación.

3. Agregue los cambios al área de preparación.

```
git add recetas/Colombia/arroz_coco.html
```

4. Consulte nuevamente el estado del repositorio para verificar los cambios.

```
git status
```

El archivo `/recetas/Colombia/arroz_coco.html` ha sido etiquetado como `modified` (en color verde), es decir que los cambios ya fueron movidos al área de preparación y están pendientes de guardarse en el repositorio.

5. Una vez esté seguro del cambio, confírmelo en el repositorio.

```
git commit -m "Se modifica ruta de imagen en receta"
```

6. Consulte el historial de cambios.

```
git log
```

Verifique los 2 `commit` realizados hasta el momento: su identificador, fecha de creación, autor y mensajes de confirmación.

Note que el primer `commit` tiene la etiqueta (`repo_remoto/master`) mientras que el segundo tiene la etiqueta (`HEAD -> master`), esto quiere decir que el primero se encuentra sincronizado con el repositorio remoto mientras que el segundo sólo está en el repositorio local.

7. Utilice el comando `diff` para consultar las diferencias entre la rama `master` del repositorio local y la rama `master` del repositorio remoto.

```
git diff remotes/origin/main main
```

Como puede observar, este comando permite identificar el directorio del archivo modificado y las líneas específicas que fueron cambiadas, en este caso la línea roja representa lo que está en el repositorio local y la verde lo que está en el repositorio remoto.

8. Sincronice los cambios con el repositorio remoto.

```
git push
```

Ahora el repositorio remoto ha sido actualizado con los últimos cambios.

9. Consulte el historial de cambios.

```
git log
```

Ahora las etiquetas (`HEAD -> master`) y (`repo_remoto/master`) se encuentran en el segundo `commit`, esto significa que el repositorio local y el remoto están actualizados.

### PASO 3. AGREGAR ARCHIVOS AL REPOSITORIO

Después de agregar el código base y modificar un archivo, es hora de agregar una nueva receta al proyecto.

Para realizar esta parte del tutorial tenga en cuenta los archivos en la carpeta: `/files_to_add`.

Siga los siguientes pasos:

1. Copie y pegue la carpeta `/files_to_add/Mexico` en el directorio `/recetas` de su copia de trabajo local.
2. Copie y pegue la imagen `/files_to_add/imagenes/pozole.jpg` en la carpeta `/imagenes` de su copia de trabajo local.
3. Verifique el estado del repositorio.

```
git status
```

4. Agregue los cambios al área de preparación.

```
git add .
```

5. Confirme los cambios en el repositorio local.

```
git commit -m "Se incluye receta mexicana"
```

6. Sincronice los cambios con el repositorio remoto.

```
git push
```

7. Consulte el historial de cambios.

```
git log
```

## PASO 4. DEVOLVERSE A UNA VERSIÓN ANTERIOR

Para devolverse a una versión anterior utilice el comando `reset`. Dicho comando restablece el apuntador de la versión actual (HEAD) a la especificada.

Es posible usarlo de diferentes formas, en este tutorial aprenderá dos:

- La primera consiste en usar el id del commit al cual se desea regresar:

```
git reset --hard <id_commit>
```

- Y la segunda, especificando el número de commits que se desea devolver:

```
git reset --hard HEAD~<numero_de_commit_atras>
```

Realice los siguientes pasos para devolverse a la versión inicial (primer commit):

1. Teniendo en cuenta el historial de commit consultado en la sección anterior, utilice el comando de su preferencia.

```
git reset --hard HEAD~2
```

Note que se ha devuelto 2 commit hasta llegar al primero.

2. Consulte el historial de commit.

```
git log
```

Al consultar el log, podrá verificar que los demás commit fueron eliminados. Sin embargo, estos cambios sólo están en el repositorio local (HEAD -> master).

3. Actualice los cambios en el repositorio remoto.

```
git push -f origin main
```

La opción `-f` o `--force` sobre escribe toda la estructura y secuencia de commits en el repositorio remoto. Lo recomendable es asegurarse de hacer pull y estabilizar el código en el repositorio local antes de hacer push. Dado que Git encontrará diferencias en cuanto a la posición del HEAD en el repositorio local y el repositorio remoto, es probable que rechace el push. Para evitarlo force la operación agregando `-f` y la rama de destino.

4. Consulte nuevamente el historial de commit.

```
git log
```

Como puede observar tanto el repositorio local como el repositorio remoto se han devuelto a la primera versión.

## REPOSITORIOS REMOTOS - CONFLICTOS

### CONFLICTO: SITUACIÓN 1

Esta situación de conflicto se provocará al modificar diferentes partes del mismo archivo y tratar de sincronizar los cambios con el repositorio remoto.

Realice los siguientes pasos:

1. Clone el repositorio en un directorio diferente al usado en la parte 1 del tutorial (repositorio 2).  
Note que al terminar la clonación se habrá creado la carpeta del repositorio local (2) en el directorio seleccionado.

2. Ubíquese en el directorio del repositorio (2).

```
cd receta
```

3. Modifique la receta Arroz con coco y camarón fresco (Colombia) en la línea 14 cambiando el número de personas a 10.
4. Guarde los cambios en el repositorio local (2).

```
git status
git add .
git status
git commit -m "Se ajusta número de personas en receta"
```

5. Actualice el repositorio remoto.

```
git push
```

Ahora desde el repositorio creado en la primera parte del tutorial (repositorio 1), modifique la misma receta del punto 3 en la línea 33 cambiando Ajiaco por Ajiaco Santafereno.

6. Guarde los cambios en el repositorio local (1).

```
git status
git add .
git status
git commit -m "Se ajusta nombre de url para próxima receta"
```

7. Actualice el repositorio remoto.

```
git push
```

Note que Git ha rechazado el push al repositorio remoto. Esto se debe a que el mismo archivo fue modificado desde repositorios diferentes y la última versión del repositorio remoto no coincide con la versión anterior del repositorio local.

Para solucionarlo es necesario actualizar el repositorio local y después enviar los cambios al repositorio remoto.

```
git pull
```

Una vez ejecutado el comando git pull, se hace automáticamente una mezcla entre el repositorio local y el repositorio remoto. A este proceso se le conoce como merge.

El merge es un commit dentro del historial de cambios y por esto requiere de un mensaje. Para identificar el por qué fue necesario realizarlo. Guarde los cambios y cierre el editor.

8. Actualice el repositorio remoto.

```
git push
```

Con lo anterior habrá solucionado el conflicto generado al modificar diferentes partes del mismo archivo. Continúe con el tutorial para aprender a solucionar conflictos generados por modificar la misma parte del mismo archivo.

## CONFLICTO: SITUACIÓN 2

Esta situación de conflicto se provocará al modificar la misma parte del mismo archivo y tratar de sincronizar los cambios con el repositorio remoto.

Realice los siguientes pasos:

1. Clone el repositorio en un directorio diferente al usado en la primera parte del tutorial (repositorio 2).

Note que al terminar la clonación se habrá creado la carpeta del repositorio local (2) en el directorio seleccionado.

- Ubíquese en el directorio del repositorio (2).

```
cd recetaario
git pull
```

Modifique la receta Arroz con coco y camarón fresco (Colombia) en la línea 8 cambiando el name de la imagen por arroz\_coco.

- Guarde los cambios en el repositorio local (2).

```
git status
git add .
git status
git commit -m "Se modifica name de imagen"
```

- Actualice el repositorio remoto.

```
git push
```

Ahora deberá tener un commit adicional en el repositorio remoto. Verifíquelo en GitHub. Desde el repositorio creado en la primera parte del tutorial (repositorio 1), modifique la misma receta del punto 3 en la línea 8 cambiando el id de la imagen por Imagen1.

- Guarde los cambios en el repositorio local (1).

```
git status
git add .
git status
git commit -m "Se modifica id de imagen"
```

- Actualice el repositorio remoto.

```
git push
```

Note que Git ha rechazado el push al repositorio remoto. Esto se debe a que existen cambios en él que aún no han sido sincronizados con el repositorio local (1). Para resolverlo, haga pull del repositorio remoto.

```
git pull
```

Como se observa en la imagen anterior, Git ha identificado un conflicto entre la versión del repositorio local (1) y la versión del repositorio remoto, esto se debe a que el mismo archivo fue modificado desde diferentes repositorios (1 y 2).

- Resuelva el conflicto.

Abra el archivo que generó el conflicto con su IDE preferido y corrija manualmente.

Las líneas de código que están después de la etiqueta HEAD corresponden a lo que está en el repositorio local (1) y lo que está después de ===== corresponde a lo que está en el repositorio remoto.

Es su decisión si se queda con lo que está en uno, en el otro o en ambos. Sólo tenga en cuenta eliminar las etiquetas <<<<<< HEAD, ===== y >>>>>>. Ya que con eso Git identifica que el conflicto ha sido solucionado.

- Guarde los cambios en el repositorio local (1) y actualice el repositorio remoto.

```
git status
git add .
git status
git commit -m "Se soluciono el conflicto"
git push
```

- Actualice repositorio 2.

```
git pull
```

Note que Git incorpora los cambios al repositorio local 2, muestra los archivos modificados y el número de líneas insertadas o eliminadas.

Es importante recordar que el conflicto anterior se podría haber evitado actualizando el repositorio local 1 antes de realizar los cambios. Esto se considera una buena práctica en cuanto al manejo de repositorios compartidos.