Contenido

[Git Workflow 1](#_Toc512702257)

[Conceptos 1](#_Toc512702258)

[Comandos 2](#_Toc512702259)

[Git Tags 3](#_Toc512702260)

[Git Bare 3](#_Toc512702261)

[Procesos de repositories “Forked”(Para proyectos de terceros) 3](#_Toc512702262)

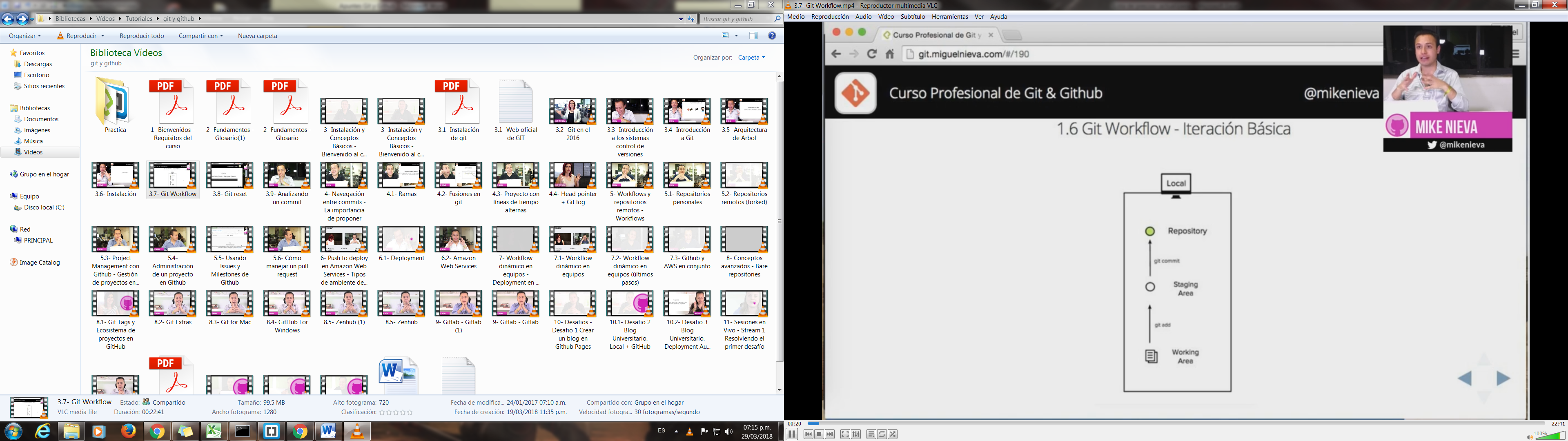
[Git Extra 4](#_Toc512702263)

[Git Pages 4](#_Toc512702264)

[Git Diff 4](#_Toc512702265)

[Git Stash 4](#_Toc512702266)

# Git Workflow



# Conceptos

**Ramas:** Son líneas alternativas al proyecto original, en las que se pueden trabajar sin afectar al proyecto master, se puede tener varias ramas dentro de un proyecto

**Fusiones:** Es la unión de una rama con el proyecto master. Al unirlo lo hace automáticamente siempre y cuando no se haya trabajado en la misma línea de código de un mismo archivo, cuando sucede esto el líder de proyecto debe de decidir manualmente cual es la línea mas conveniente y hacer el cambio

**Rebase:** es colocar las ramas adelante del master poniéndolo todo en línea recta, para después mover el master al final de la recta

# Comandos

**Git init:** Agrega el repositorio a git para iniciar

**Git status:** Te da un status de los archivos modificados o los que no se al almacenado en el repositorio

**Git add –A:** Prepara el repositorio para ser almacenado

**Git commit –m “Se pone que se editó”:** Se lanza el repositorio y lo almacena

**Git log:** Muestra los commits que se han realizado. Se le pueden agregar mas comandos para que las búsquedas de los commits sean mas fáciles de buscar

**Git reset –hard (id commit):** Regresa a un punto de control atrás en los commits, eliminando todo los cambios que se hicieron (Nota: es importante respaldar los ID que se van a eliminar)

**Git reset –mixed (id commit):** regresa a un punto de control atras pero sin eliminar los cambios en el workArea, este nos permite incorporar todos los commits que se eliminaron en unos solo, sin embargo se necesita seleccionar los archivos que se van a agregar al stanging área para hacer el commit

**Git reset –soft (id commit):** Nos permite hacer lo mismo que el mixed pero aquí automáticamente se agregan los cambios al stanging area

**Git checkout (id commit):** Nos permite retornar a un punto de control sin eliminar nada, y de ahí podemos crear un punto paralelo al que ya se tiene

**Git checkout –b (Nombre de la rama):** Crea una nueva rama

**Git checkout master:** Nos movemos a la rama master

**Git merge (Nombre de la rama):** Une la rama con el master

**Git remote add origin (Direccion ssh del repositorio git):** Conecta con el servidor remoto

**Git push origin (Rama a enviar):** Envia los datos al servidor remoto

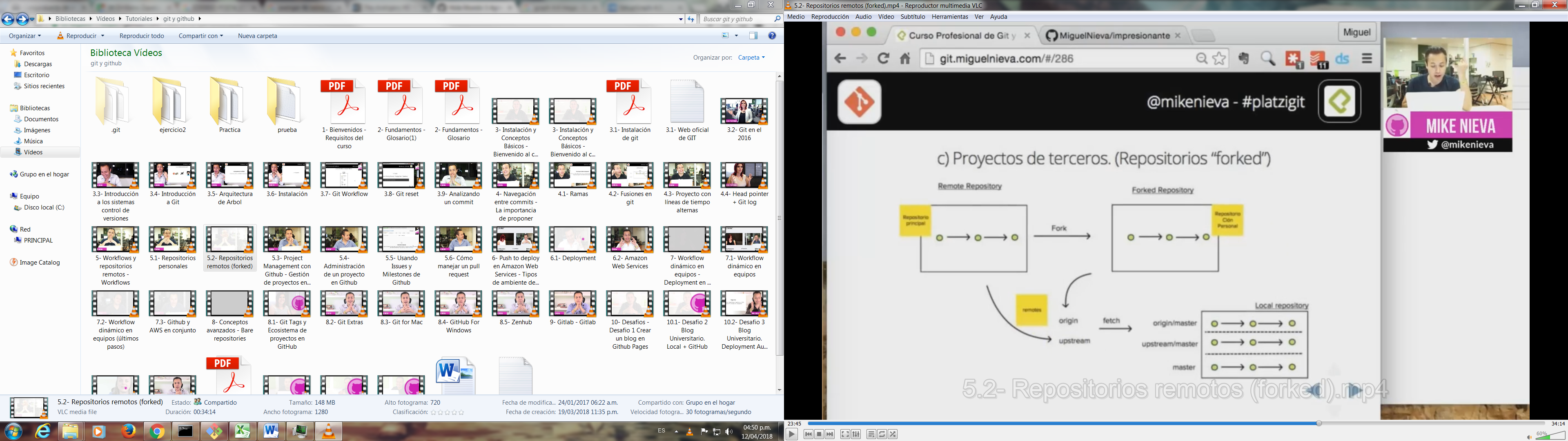
**Git fetch origin:** Trae los cambios del servidor remoto a la rama origin/master

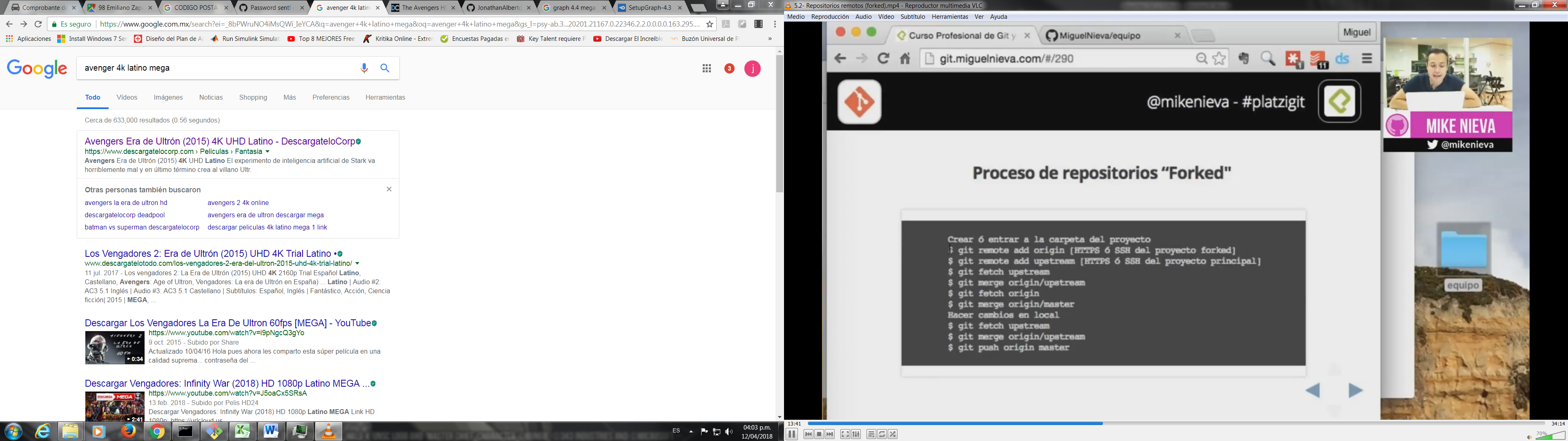
**Superlog:** git config --global alias.superlog "log --graph --abbrev-commit --decorate --date=relative --format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset) %C(white)%s%C(reset) %C(dim white)- %an%C(reset)%C(bold yellow)%d%C(reset)' --all"

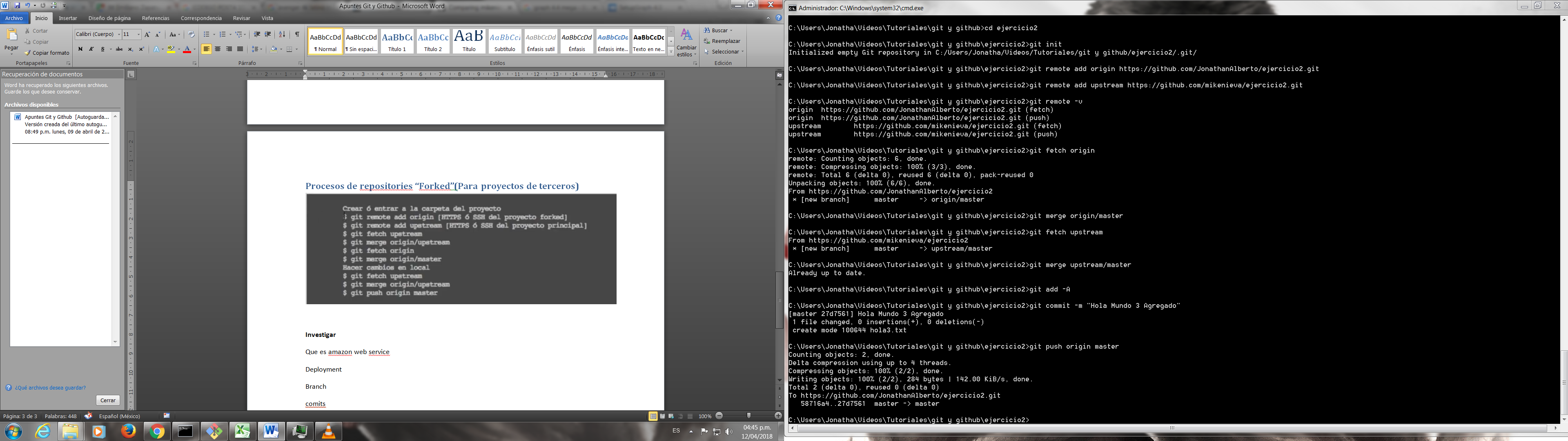
# Git Tags

# Git Bare

# Procesos de repositories “Forked”(Para proyectos de terceros)







# Git Extra

# Git Pages

Podemos publicar nuestro proyecto web utilizando github pages, para esto

# Git Diff

# Git Stash

**Investigar**

Que es amazon web service

Deployment

Branch

comits

feactures

bug fixed