Reporte de práctica de laboratorio

Práctica 10: Control de versiones (CodeCommit)

Fecha 01/04/2021

Maestría en Informática Aplicada

*Aplicaciones y Servicios en la Nube*

Prof. Mtro. Rodolfo Luthe Ríos

Roberto Leos

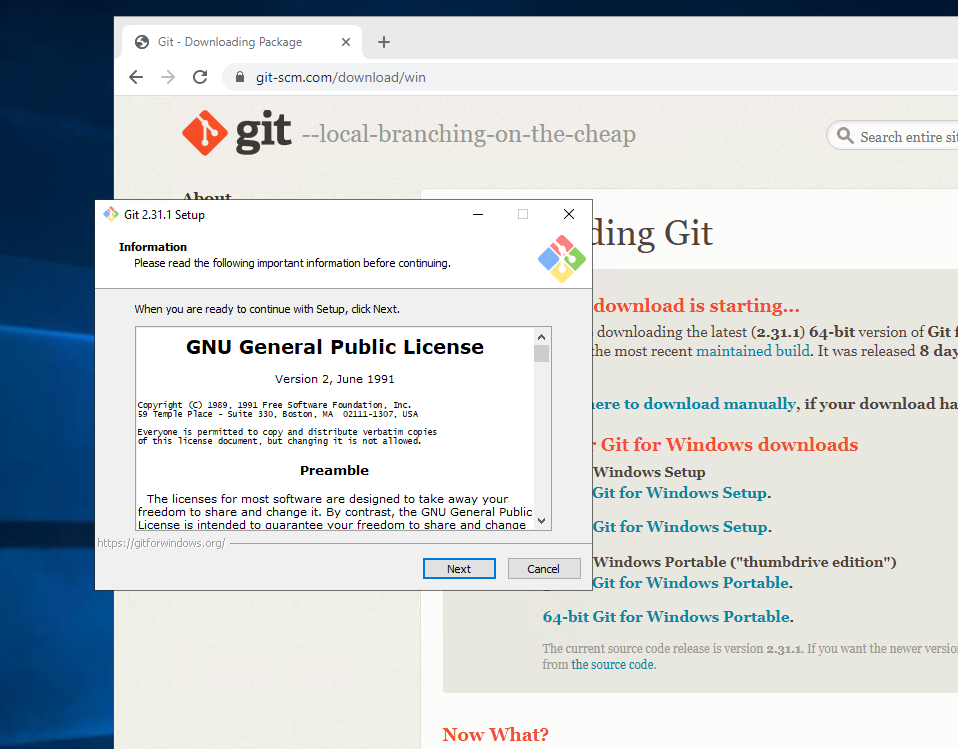
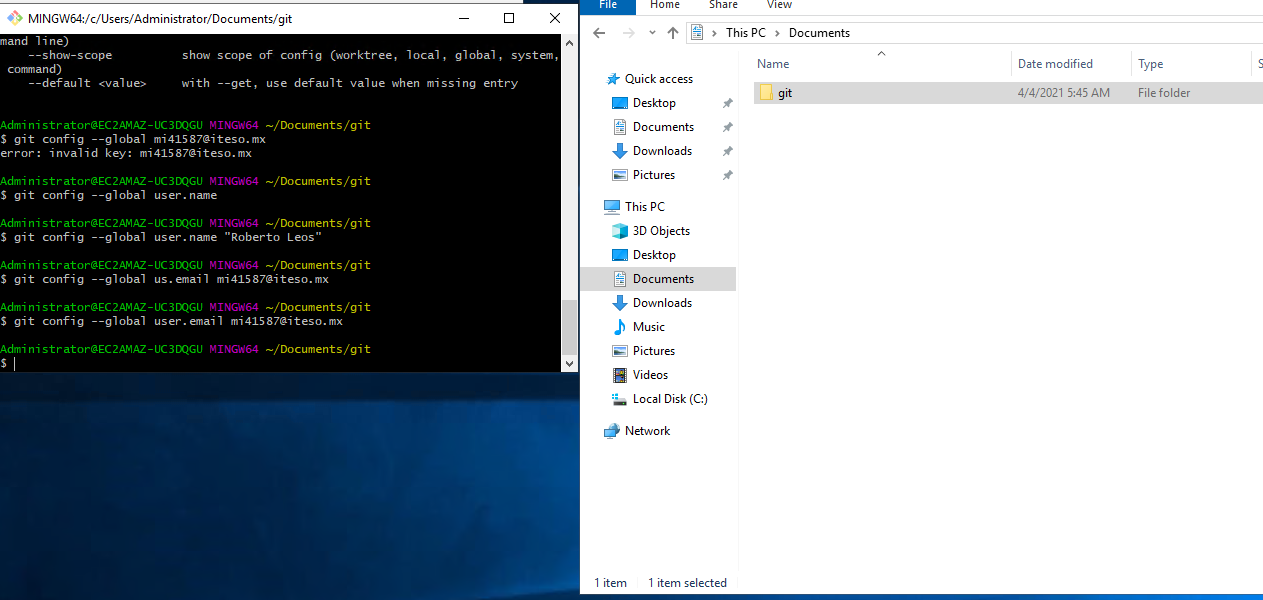
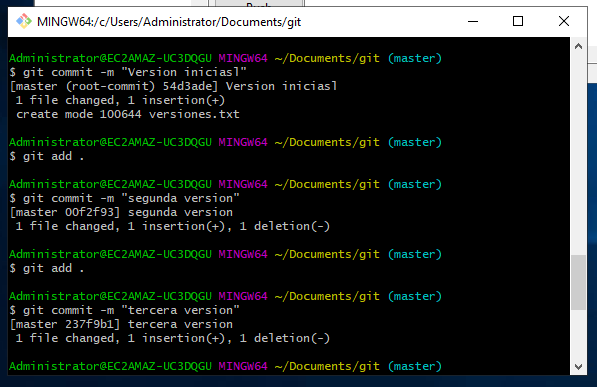
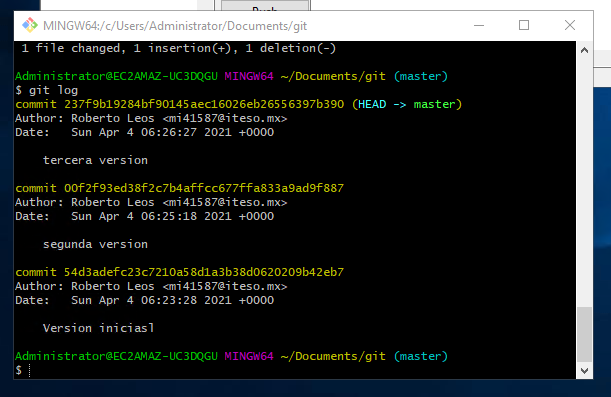
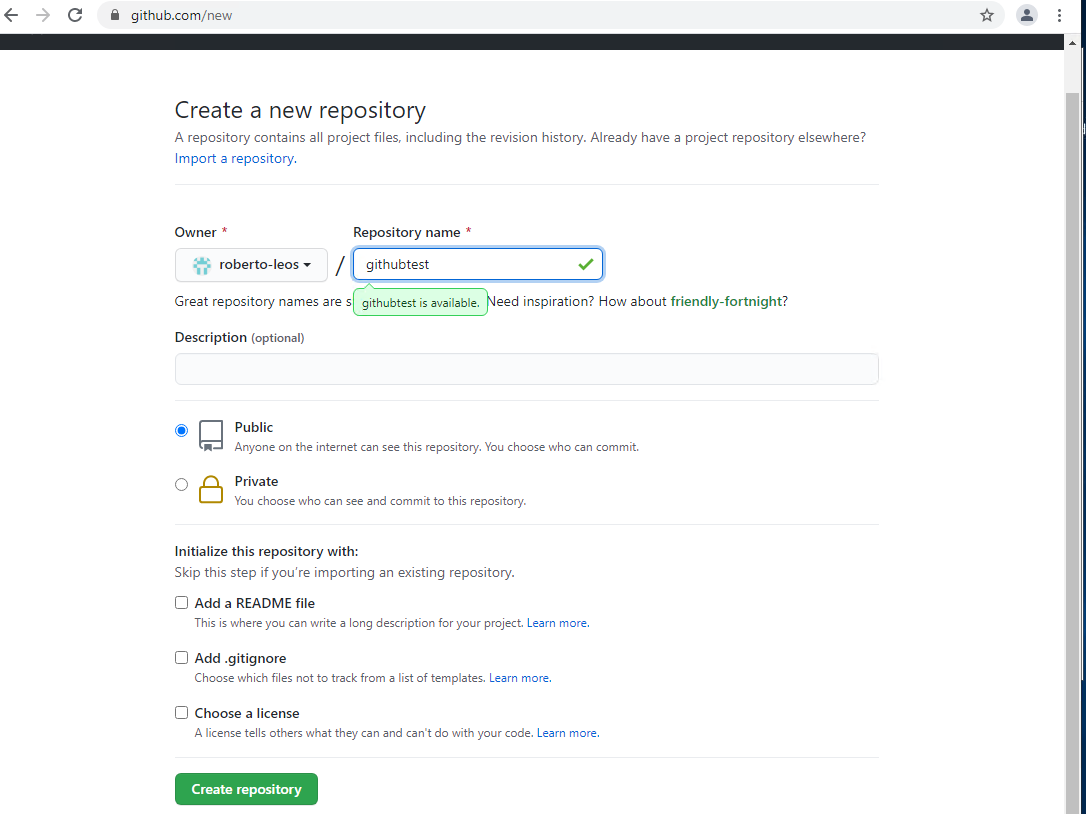
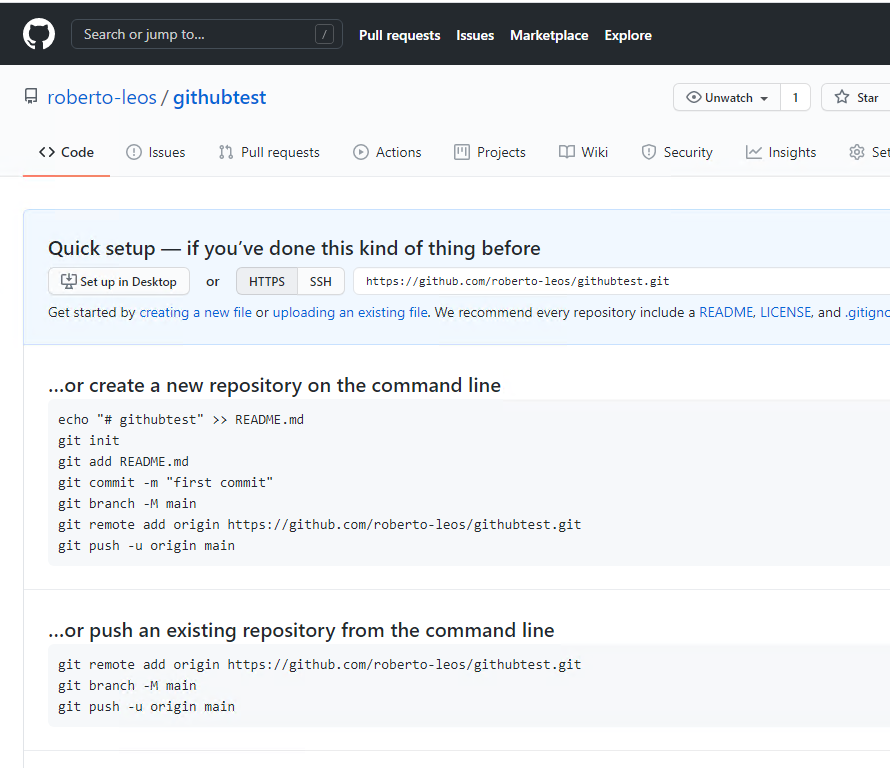
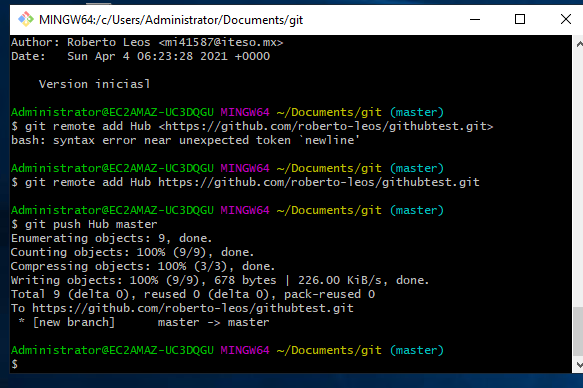
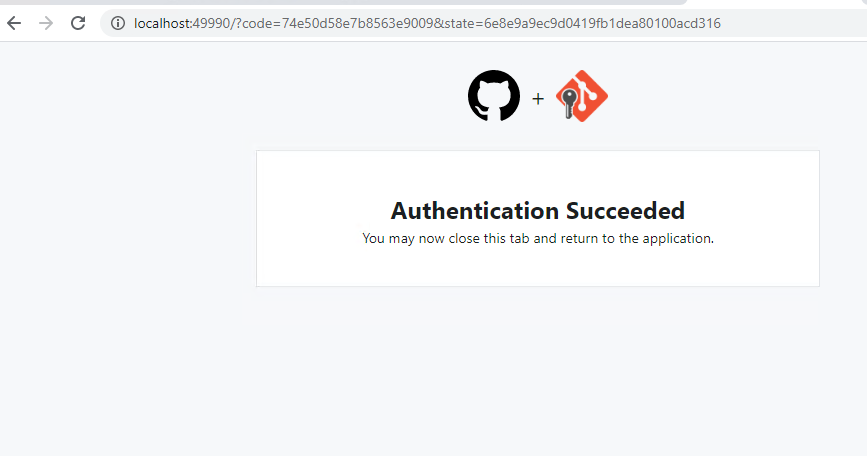
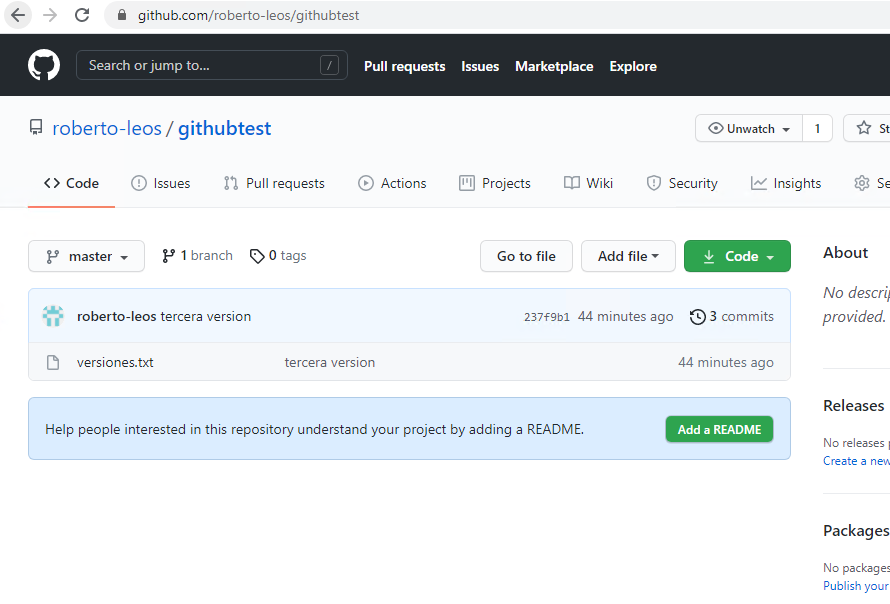
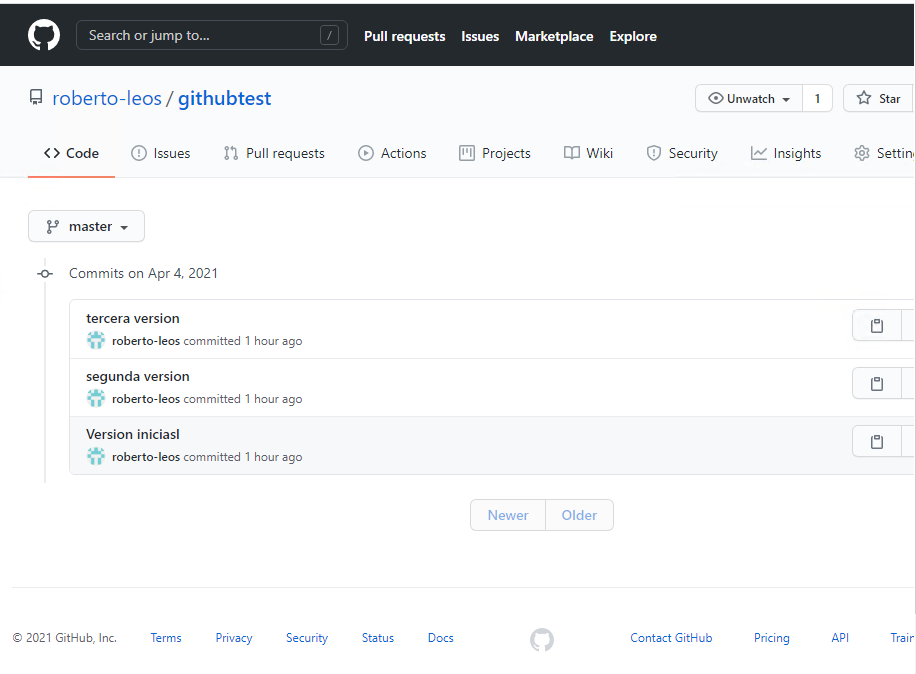
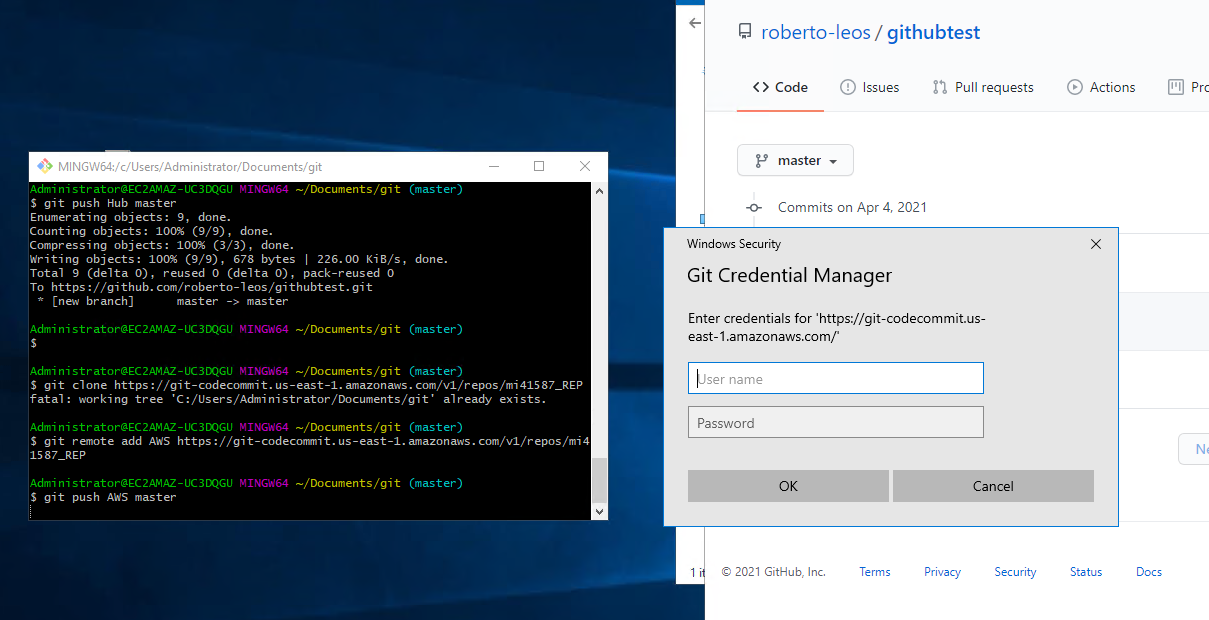
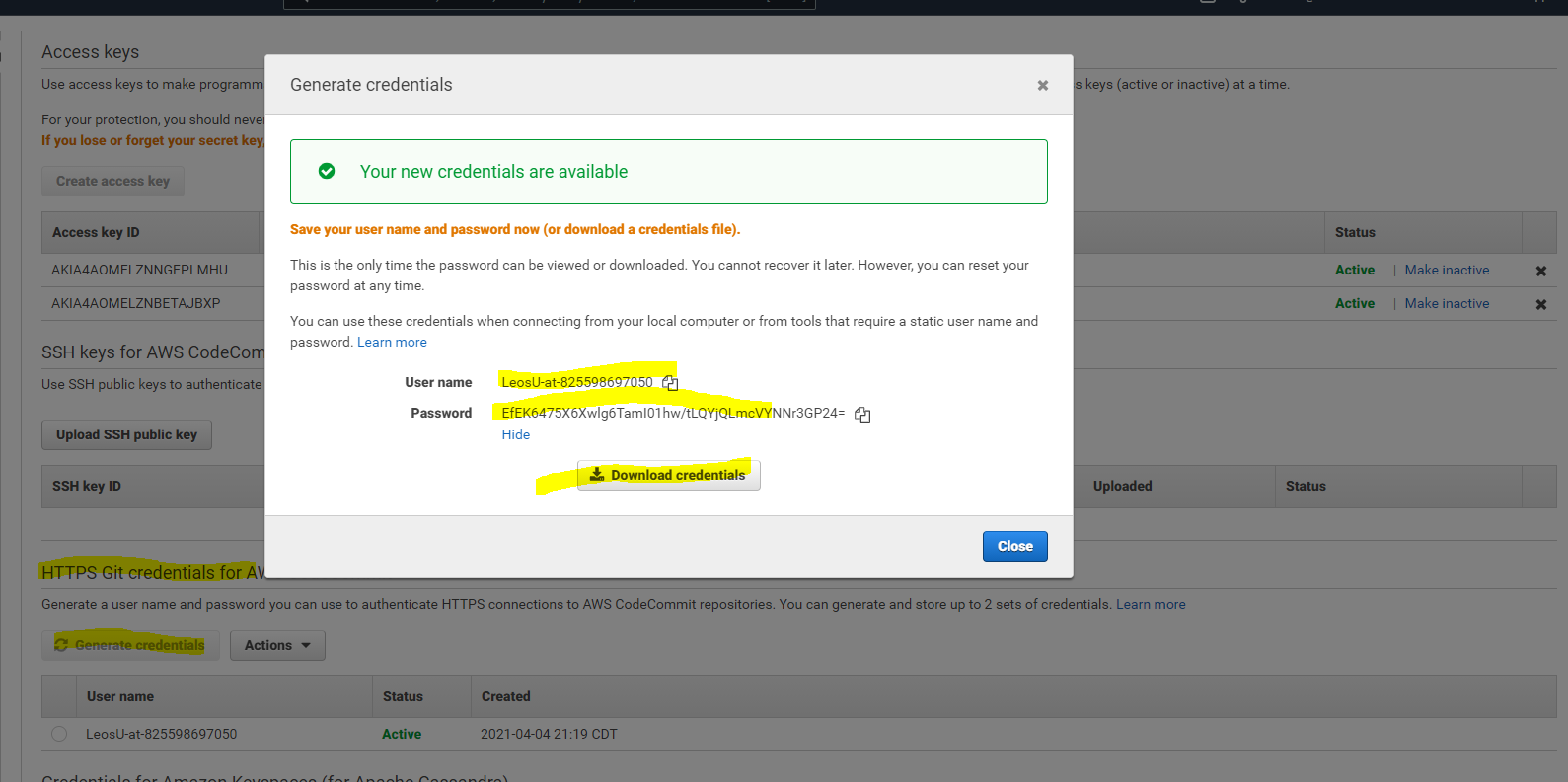
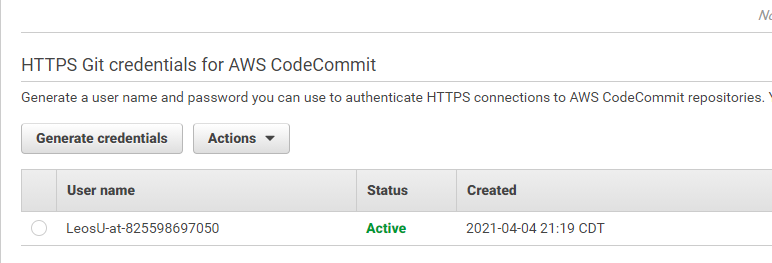
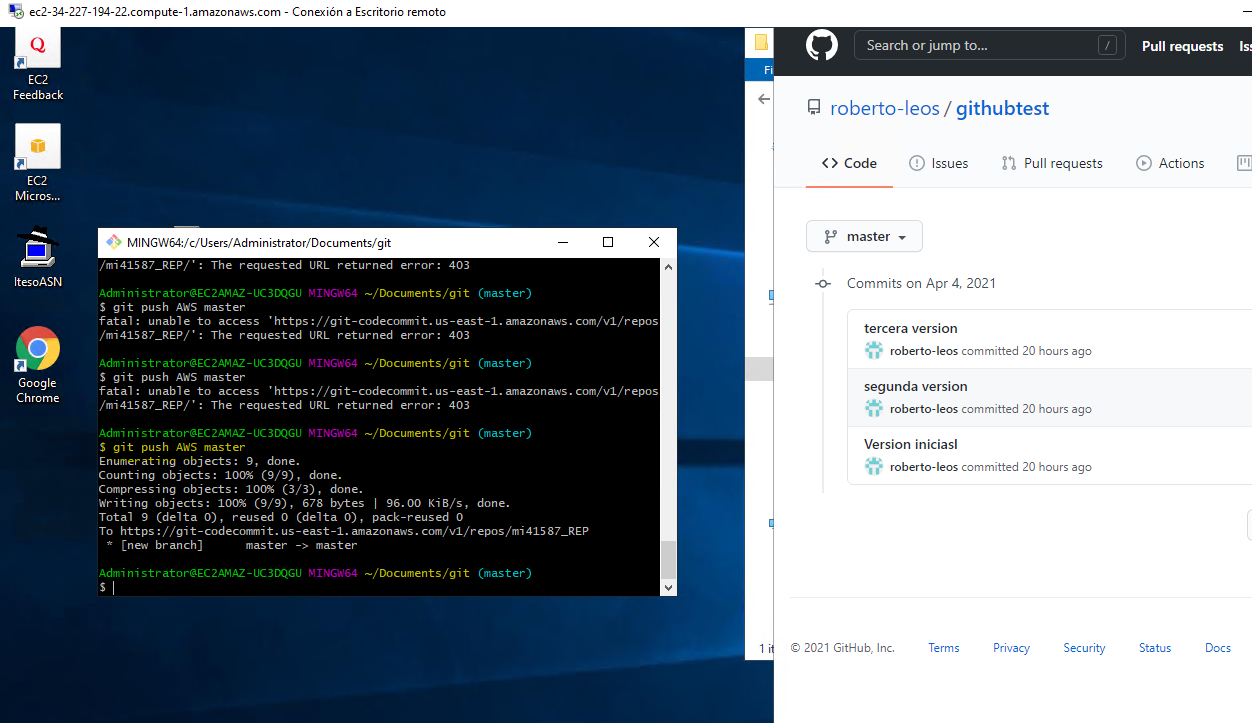
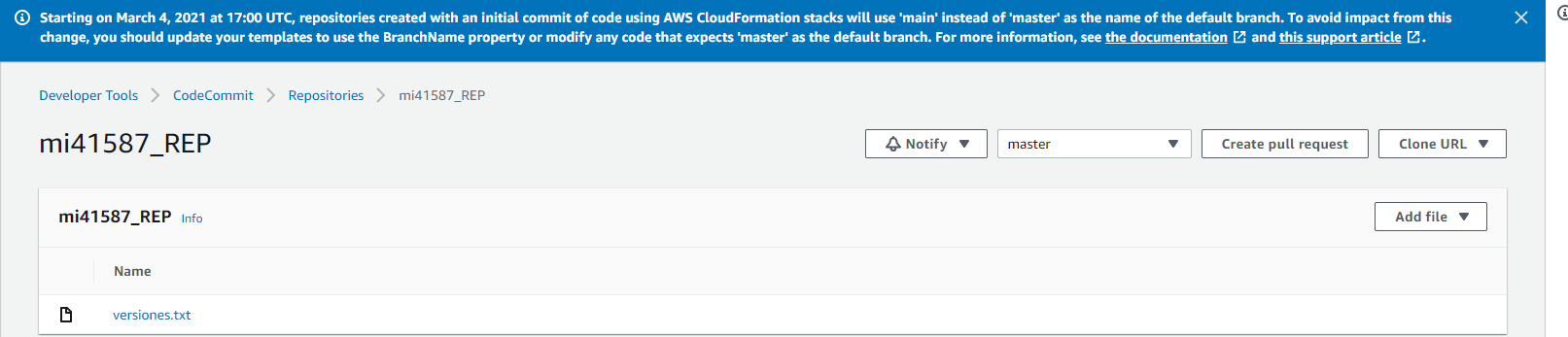
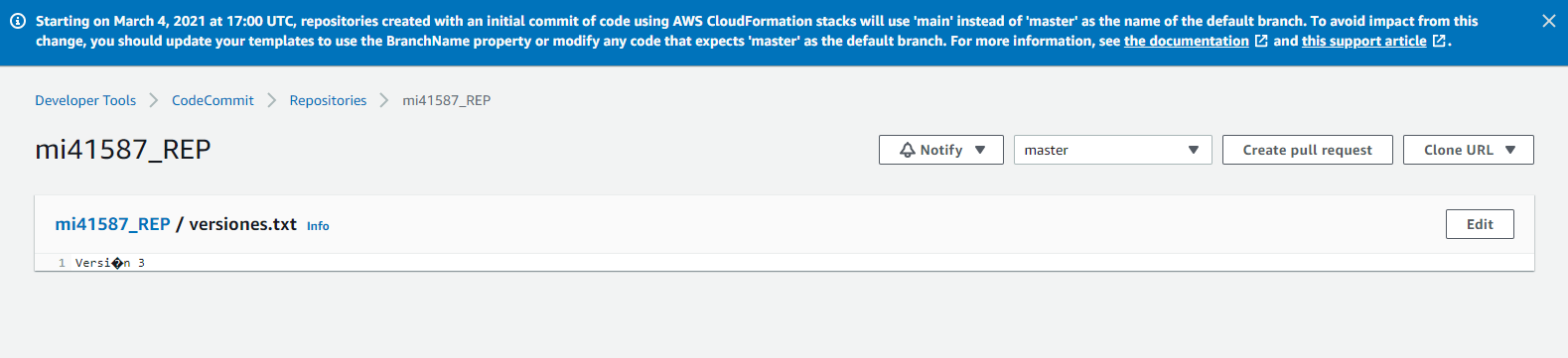
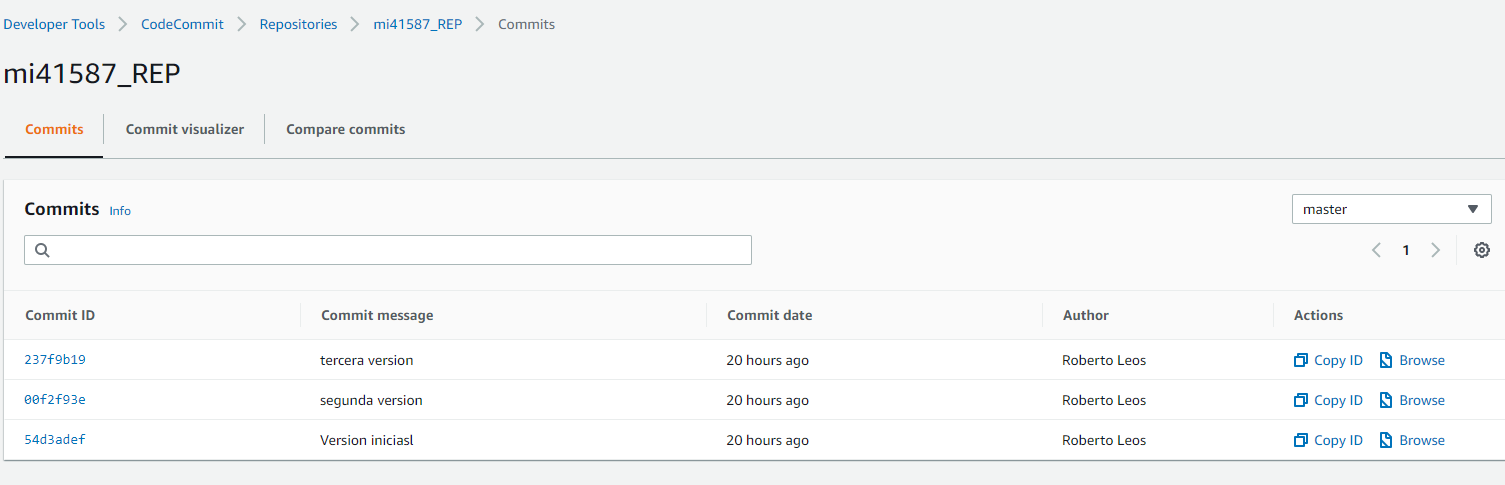
[mi41587@iteso.mx](mailto:mi41587@iteso.mx)

# Introducción

# Marco Teórico

# Diagrama

# Desarrollo de la Práctica.

1. Instalar cliente [git](https://git-scm.com)  
   
2. Vamos a configurar el git con cuenta del iteso y vamos a crear un folder nuevo en mis documentos llamada git a la cual dar botón derecho en el folder y seleccionar “git bash here”.comando “git config --global user.name "Roberto Leos" y git config --global user.email mi41587@iteso.mx”  
   
3. Se va crear un archivo .txt en el directorio git llamado versiones.txt con el contenido “versión 1”, luego daremos seguimiento al archivo desde el git dando el comando “git add .” desde la consola y luego poner un comentario “git commit -m “versión inicial”” vamos a repetir este paso en tres ocasiones cambiando el contenido del archivo “versiones.txt” y después consultaremos la historia con el comando “git log”  
     
   
4. Ahora crearemos una cuenta en GitHub  
   
5. Crearemos un repositorio  
   
6. Con el url que da el repositorio la vamos a ligar desde nuestra carpeta git usando el comando “git remote add Hub <https://github.com/roberto-leos/githubtest>” y luego “git push Hub master” con este va pedir dar permisos para hacer el link y veremos nuestro archivo “versiones.txt2 en el Github”  
     
     
   
7. Podemos ver el archivo con todos y la historia de los tres cambios que se hicieron.   
   
8. 
9. Siguiendo la guía “[setup for HTTPSUsers using Git credentials](https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-gc.html?icmpid=docs_acc_console_connect_np)” en al IAM después de agregar la política “AWSCodeCommitPowerUser” a tu usuario vamos a ir al tap “security credentials” y en la sección “HTTPS Git credentials for AWS CodeCommit” le damos al botón “Generate credentials”. Te va generar el usuario y password y lo podrás descargar en .csv   
     
   
10. Una vez generado el acceso podemos dar el “git remote add AWS <URL>” y después el “git push AWS master” el cual te pedirá las credenciales antes mencionadas y podrás ver el archivo “versiones.txt” en nuestro repositorio en AWS.   
      
      
      
    
11. Versiones del entregable de la practica 10 en donde por cada sección del documento vamos hacer add y commit para mostrar el log de cambios.
12. ¿Ves conveniente el uso de control de versiones para documentos que no son código?
    1. Si, por ejemplo, en documentación de proyectos, en áreas como control de documentos para los procedimientos. En general esto es muy útil para llevar una correcta documentación sobre cualquier cosa.
13. ¿Cómo comparas GitHub y CodeCommit? Los dos son implementaciones DVCS de git
    1. Los dos te dejan hacer cargas en la nube y compartir los archivos desde un repositorio, pero veo a codecommit mas con el proceso devops pues maneja AWS CodeStar, CodePipeline, CodeCommit, CodeBuild, CodeDeploy y en github solo encontré opciones de Kanban en los templates.

# Experimentos y Resultados.

1. [DetectLabels](https://docs.aws.amazon.com/rekognition/latest/dg/API_DetectLabels.html):

# Costo

# Conclusiones

# Bibliografía

|  |  |
| --- | --- |
| [1] | AWS, «AWS,» [En línea]. Available: https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html. |
| [2] | A. Rekognition, «AWS,» [En línea]. Available: https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html. |
| [3] | B. Copeland, «artificial intelligence (AI),» Britannica ACADEMIC, [En línea]. Available: https://academic-eb-com.ezproxy.iteso.mx/levels/collegiate/article/artificial-intelligence/9711#article-contributors. |