Reporte de práctica de laboratorio

*Control de versiones (CodeCommit)*

*04 de Abril del 2021*

Maestría en Sistemas Computacioneales

*Aplicaciones y Servicios en la Nube*

Prof. Mtro. Rodolfo Luthe Ríos

Priscila Campos Velázquez

[ms727309@iteso.mx](mailto:mi11729@iteso.mx)

# Introducción

* ¿Ves conveniente el uso de control de versiones para documentos que no son código?
* ¿Cómo comparas GitHub y CodeCommit? Los dos son implementaciones DVCS de git

Los objetivos de esta practica:

* Utilizar un cliente de control de versiones
* Configurar servicios de control de versions centralizados
* Controlar las versiones de un documento

Las actividades principales de la practica fueron:

* Configurar un repositorio local
* Crear un repositorio en GitHub
* Crear un reporsitorio en CodeCommit
* Controlar las versiones de un documento

# Marco Teórico

Los conceptos más destacados para la buena comprensión de la practica fueron:

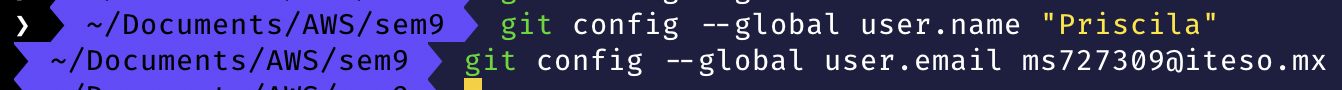
* ***DVCS (Distributed Version Control System):***El DVCS
* ***GitHub:*** xxxx
* ***CodeCommit:*** xxxx

# Diagrama

# Desarrollo de la Práctica.

**Configurar repositorio local**

* 1. Instalar el cliente de git en una máquina.
     1. https://git-scm.com
  2. Configurar git con su cuenta de iteso.
     1. git config --global user.name "John Doe".
     2. git config --global user.email johndoe@example.com.

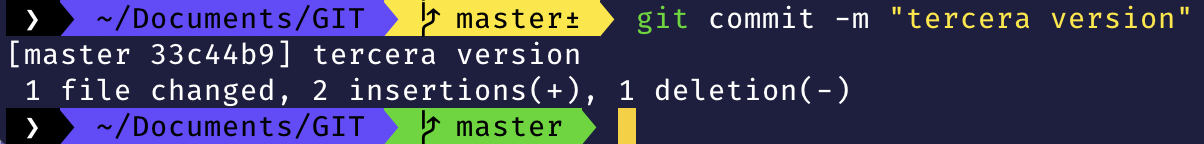


* 1. Configurar el repositorio.
     1. Crear en Mis Documentos una carpeta llamada git.
     2. Desde consola cambiarse a ese directorio (o en Windows click derecho sobre la carpeta y “Git Bash here”), git init.
  2. Control de versiones:
     1. Crear en el directorio git un archivo: versiones.txt con el contenido: Versión 1

A picture containing text

Description automatically generated

* + 1. Añadir el archivo al seguimiento de git, desde la consola en la carpeta: git add .
       1. Es importante el “.” para incluir todos los archivos en el tracking
    2. Hacer un commit de los cambios: git commit –poner mensaje versión inicial.Text

       Description automatically generated with medium confidence
    3. Editar el archivo y cambiar el contenido a: Versión 2.
       1. Hacer add y commit de los cambios:
          1. git add .
          2. git commit -m “segunda version”
    4. Hacer lo mismo para la Versión 3
    5. Consultar historial de vesiones: git log. Text

       Description automatically generated

**Configurar repositorio en GitHub**

* 1. Crear una cuenta con su correo de iteso en GitHub Graphical user interface, text, application, email

     Description automatically generated
  2. Crear un repositorio en GitHub
  3. Consultar el URL del repositorio Graphical user interface, text, application, email

     Description automatically generated
  4. Añadir al repositorio local el repositorio de GitHub, desde la carpeta de git:
     1. git remote add Hub <URL>
     2. git push Hub master

Graphical user interface, text

Description automatically generated

* 1. Refrescar la pantalla del repositorio en GitHub ¿Qué cambio observan?

Graphical user interface, application, Teams

Description automatically generated

**Configurar repositorio en AWS CodeCommit**

* 1. Crear credemnnciales para CodeCommit.
     1. Desde la consola de IAM y seleccionar el usuario deseado.
     2. En Security Credentials generar para: HTTPS para CodeCommit.

Application

Description automatically generated with medium confidence

* 1. Desde la consola de CodeCommit crear un repositorio.Graphical user interface, text, application

     Description automatically generated
  2. Consultar el URL del repositorio.

Graphical user interface, text, application

Description automatically generated

* 1. Añadir al repositorio local el repostorio de CodeCommit, desde la carpeta de git:
     1. git remote add AWS <URL>.
     2. git push AWS master.

Text

Description automatically generated

* 1. Consultar el repositorio de CodeCommit.

Graphical user interface, text, application, Word

Description automatically generated

**Controlar las versiones del entregable de la práctica 10**

* 1. Crear el documento del reporte, y añadirlo para tracking.
  2. Hacer add y commit al terminar cada sección del reporte -Marco Teorico, Diagrama, etc- con el comentario correspondiente.
  3. Mostrar log de los cambios al documento.
  4. Subir PDF del reporte a Moodle, compartir el URL del documento en CodeCommit.

# Problemas y Soluciones

Cuando intenté hacer el push AWS master, marcaba error 403. Se solucionó agregando al usuario el permiso de AWSCodeCommitFullAccess.



Graphical user interface, text, application, email

Description automatically generated

# Costo

xxx

# Conclusiones

xxx

# Bibliografia

[1]