Reporte de práctica de laboratorio

Control de versiones (CodeCommit)

19 Febrero 2020

Maestría en Informática Aplicada

*Aplicaciones y Servicios en la Nube*

Prof. Mtro. Rodolfo Luthe Ríos

Jesús Alberto Figueroa Curiel

mi713781@sanmina.com

El reporte debe ser breve, se busca calidad no cantidad

# Introducción

En esta práctica se pretende ver los siguientes objetivos:

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

# Marco Teórico

En esta sección se debe presentar una pequeña investigación bibliográfica sobre los conceptos principales de la práctica. Esta sección debe estar correctamente citada en formato IEEE. (aproximadamente 1 cuartilla)

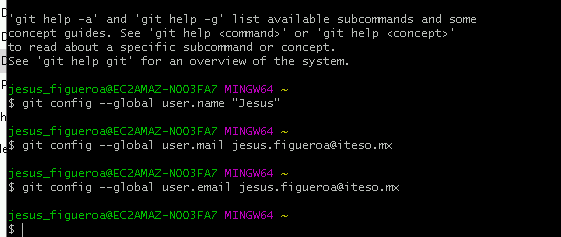
# Diagrama

Diagrama de la arquitectura implementada.

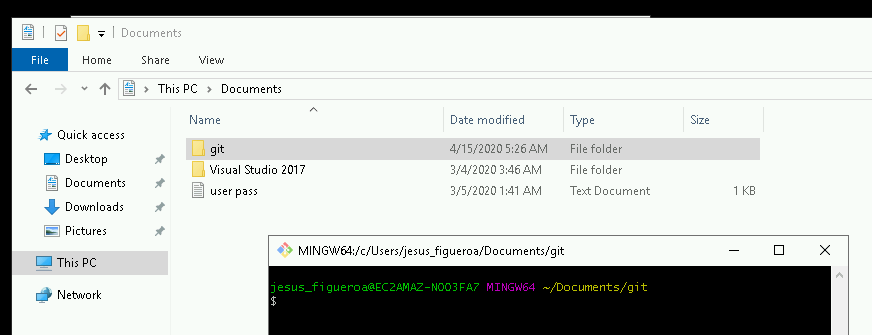
# Desarrollo de la Práctica.

**Configurar repositorio local**

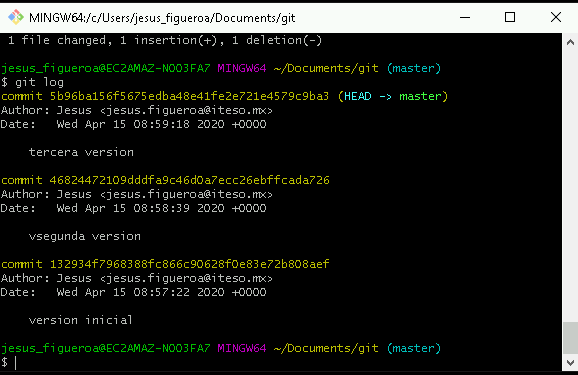
1. Instalar el cliente de git en una máquina
   1. [https://git-scm.com](https://git-scm.com/)
2. Configurar git con su cuenta de iteso a. git config --global user.name "John Doe" b. git config --global user.email [johndoe@example.com](mailto:johndoe@example.com)



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”) i. git init

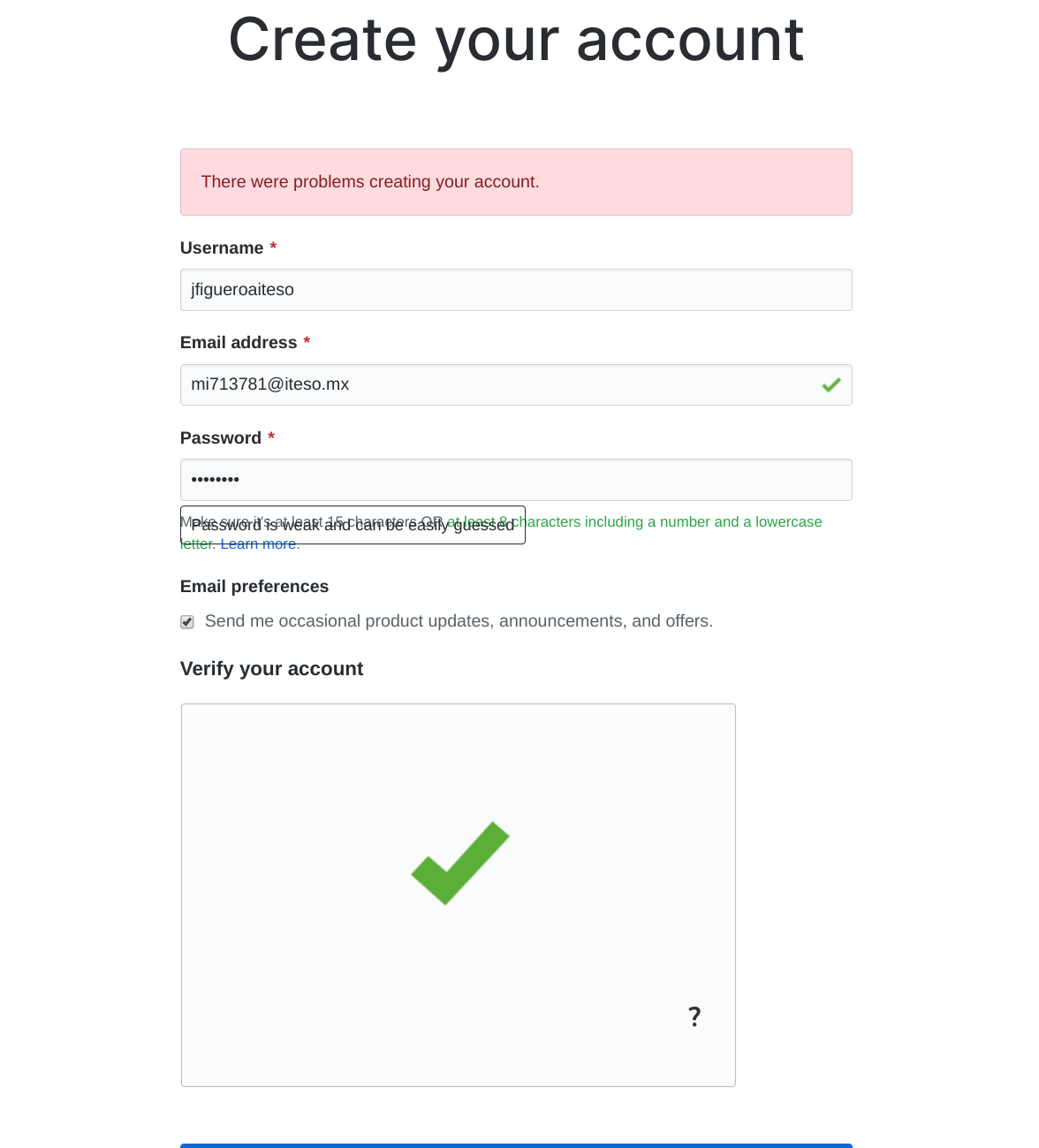


1. Control de versiones
   1. Crear en el directorio git un archivo: versiones.txt con el contenido: Versión 1
   2. Añadir el archivo al seguimiento de git, desde la consola en la carpeta: git add . i. Es importante el “.” para incluir todos los archivos en el tracking
   3. Hacer un commit de los cambios: git commit –poner mensaje versión inicial
   4. 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”
   5. Hacer lo mismo para la Versión 3
   6. Consultar historial de vesiones: git log

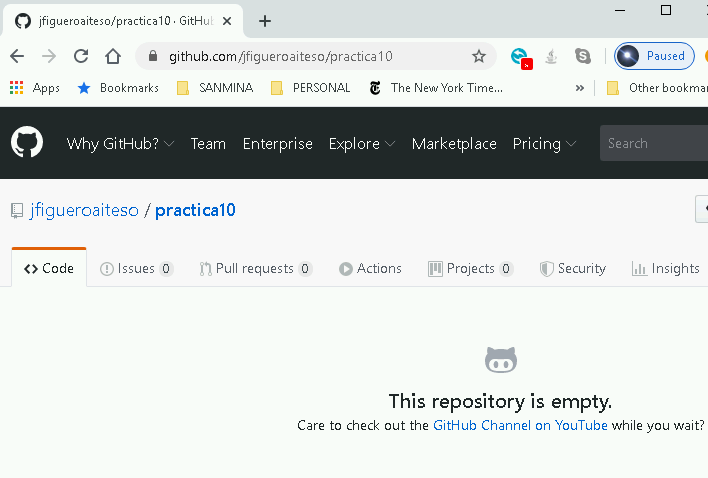


**Configurar repositorio en GitHub**

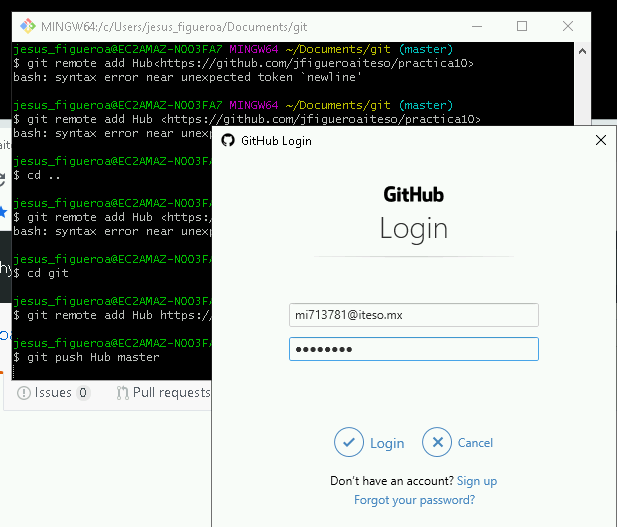
1. Crear una cuenta con su correo de iteso en GitHub



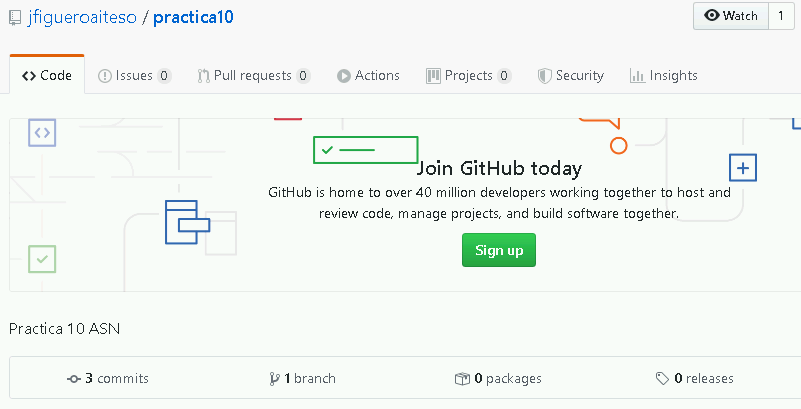
1. Crear un repositorio en GitHub
2. Consultar el URL del repositorio



1. Añadir al repositorio local el repostorio de GitHub, desde la carpeta de git:
   1. git remote add Hub “url”
   2. git push Hub master



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

*Aparecen los 3 commit y 1 branch*

**Configurar repositorio en AWS CodeCommit**

1. Crear credemnnciales para CodeCommit
   1. Desde la consola de IAM y seleccionar el usuario deseado

*AKIAZFU5IOO3OFAQB35H*

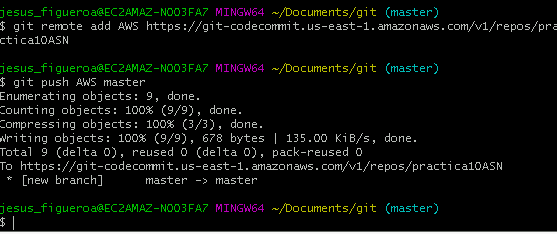
*/8jbwdQYzmKLD6gkFGMFCLtHVEbmDv8i/A/QylW3*

* 1. En Security Credentials generar para: HTTPS para CodeCommit

*adminfigueroa-at-630616257462*

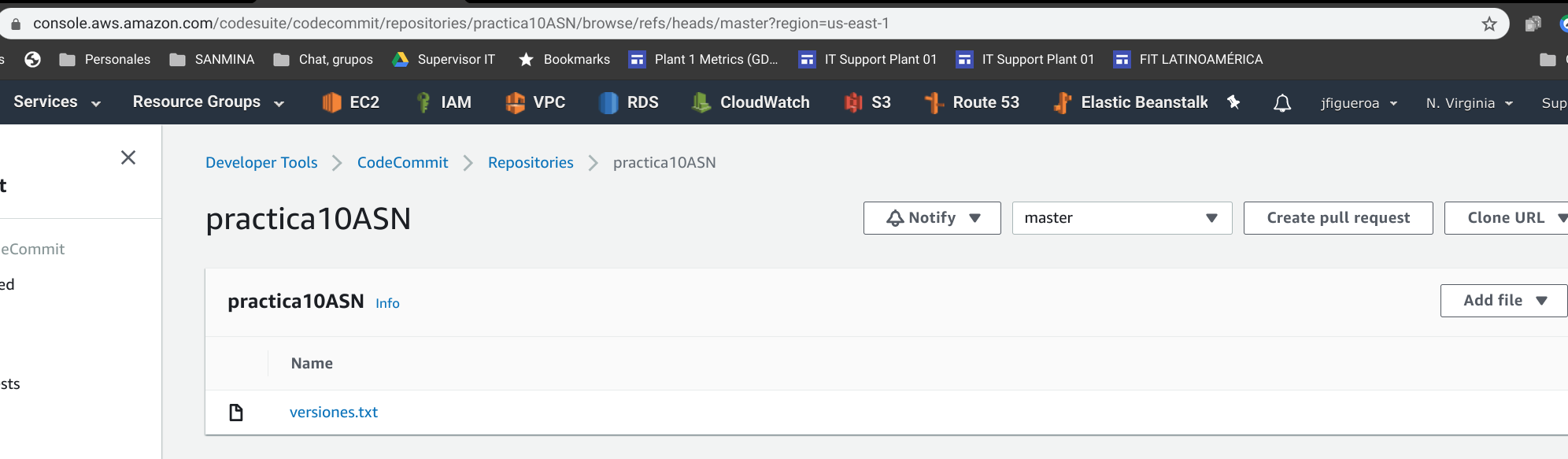
*R2UFHK7WD6k9pnzKA5DVcbpKYIi/Cyz7pEz/YcGSttg=*

1. Desde la consola de CodeCommit crear un repositorio
2. Consultar el URL del repositorio
3. Añadir al repositorio local el repostorio de CodeCommit, desde la carpeta de git:
   1. git remote add AWS “url”
   2. git push AWS master



1. Consultar el repositorio de CodeCommit

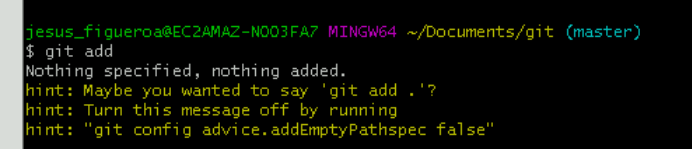
*Aparece el file de versiones*



**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, 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

Se tuvo problema cuando se estaba guardando el documento de la práctica ya que no reflejaba los commit, la solución fue que se tenía que estar guardando el documento para que refleja un cambio:</ht

# Experimentos y Resultados.

Los resultados están en el apartado de desarrollo de práctica.

*¿Ves conveniente el uso de control de versiones para documentos que no son código?*

Si lo veo conveniente principalmente en el área de desarrollo de software ya que ahí se manejan continuamente diferentes tipos de versiones

• *¿Cómo comparas GitHub y CodeCommit? Los dos son implementaciones DVCS de git*

LGithub es Open Source por lo que llega a tener el tema de confidencialidad, aunque si usa seguridad muchas veces para empresas este tema de “Open” causa dudas respecto al cuidado y protección de la información.

CodeCommit es una herramienta que viene dentro de AWS, por lo que tiene un respaldo y un soporte más comercial

• *Evidencia de la práctica*

coloca en la parte de desarrollo de la misma

# Costo

Se encontró que CodeCommit no genera costo siempre y cuando se usen menos de 5 usuarios, después de 5, empieza a costar $1usd :

“Esta oferta de la [capa gratuita de AWS](https://aws.amazon.com/es/free/) para AWS CodeCommit se encuentra disponible para clientes de AWS nuevos y existentes de forma indefinida, es decir, no vence al finalizar el período estándar de 12 meses de la capa gratuita.”</h

# Conclusiones

Las conclusiones deberán ser un trabajo de reflexión que nos presente el conocimiento adquirido desde los experimentos, los resultados y el marco teórico presentado. No son conclusiones válidas: “¡aprendí mucho!”, “me gustó mucho la práctica”, “todo funcionó correctamente”. (1 o 2 párrafos).

# Bibliografia

En formato de la IEEE