Reporte de práctica de laboratorio

Control de versiones (CodeCommit)

03/28/2020

Maestría en Sistemas Computacioneales

*Aplicaciones y Servicios en la Nube*

Prof. Mtro. Rodolfo Luthe Ríos

Carlos Alberto Cordero Robles

ms197686@iteso.mx

# Introducción

Esta práctica está diseñada para crear y manejar un repositorio usando la herramienta de control de versiones de GIT.

Hay varios proveedores de servicio que permiten almacenar y administrar el repositorio en nube, en esta práctica se trabajara con dos github y codecommit. GitHub es un proveedor ya conocido con una interfaz web y opciones muy trabajadas por muchos desarrolladores. CodeCommit es particularmente de AWS y aunque su alcance no es tan amplio como el de GitHub su diferenciador es que puede interactuar junto con los otros servicios de AWS.

# Marco Teórico

Control de versiones

Son sistemas que ayudan al equipo de desarrollo a manejar los cambios en el código. Las herramientas de control de versiones llevan un registro de cada modificación en una base de datos y si por ejemplo se encuentra algún error en el funcionamiento del código se puede analizar dicho registro para saber a partir de que cambio se hizo presente dicho error y corregirlo sin afectar el resto del trabajo del equipo. Con estas herramientas el equipo puede paralelizar su trabajo ya que cada desarrollador puede realizar sus cambios en el conocido “Branch” y posterior mente integrar los cambios a un “branch” que será común para todos.

URL

Son las siglas para Uniform Resource Locator. Son una serie de caracteres que sirven para indentificar un recurso en internet. También puede ser visto como una dirección a dicho recurso que va a estar disponible de manera virtual en internet.

GIT

Git es una herramienta de open source para el control de versiones diseñado para manejar toda clase de proyectos que involucren código. Resulta ser una herramienta bastante rápida y eficiente. Git es un sistema d eocntrol de versiones distribuido y a diferencia de los sistemas centralizados, los sistemas distribuidos tienen un manejo remoto y un manejo local de los archivos y branches y es necesario actualizar o sincronizar los cambios locales con los remotos.

Git fue desarrollado por Linus Torblads (desarrollador de Linux) y fue diseñado para el manejo de un gran número de archivos de código.