Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Análisis y Diseño de Sistemas 1
Inga. Allan Alberto Morataya Ramos
Aux. Cesar Alejandro Sazo Quisquinay

Practica No. 1
Versionamiento

Contenido

Objetivos
Generales

- Conocer y entender los elementos que integran un sistema de control de versiones.
- Comprender la importancia del trabajo en equipo

Especificos

- Conocer y entender los elementos que integran un sistema de control de versiones.
- Entender cómo trabaja, almacena y modela la información Git.
- Utilizar un repositorio para guardar la información.
- Aplicar al menos 5 acciones básicas y 5 avanzadas sobre un sistema de control de versiones

Descripción

La práctica de laboratorio consiste en crear un repositorio en un servidor en la nube, se debe crear distintas ramas para llevar el control del proyecto a desarrollar y realizar las diferentes acciones para agregar funcionalidades, reparar bug y hacer entregas usando **gitflow.**

Se debe asignar el rol de responsable de integración a un integrante del grupo, esta persona será la encargada de llevar el control de la rama reléase y el control del repositorio para verificar el correcto uso de los demás integrantes en la creación de branch, commits y merges. Se debe notificar la persona encargada a través de un correo con asunto [AyD1]P1_Grupo#.

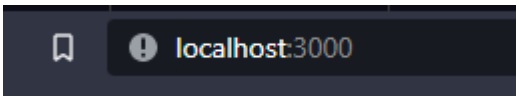
Se debe crear un repositorio en un proveedor de servicio en la nube a elección del grupo de trabajo (gitlab, github, bitbucket,etc.), el nombre del repositorio debe ser practica1_grupo#.

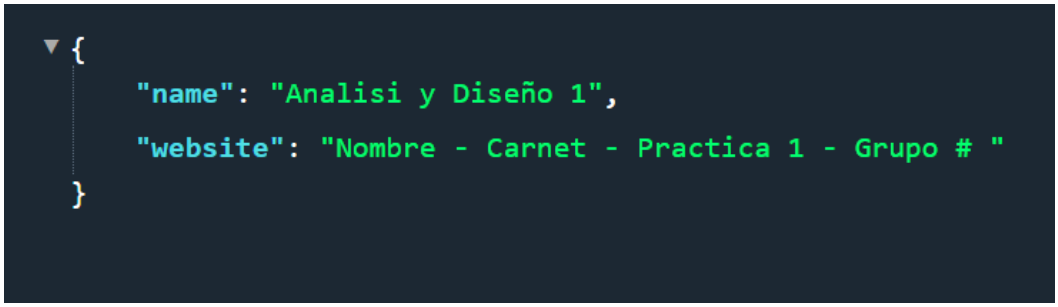
Se debe configurar los repositorios remotos y locales de cada uno de los integrantes del grupo, cada integrante del grupo deberá presentar la configuración del repositorio en su computadora al momento de la calificación.

Ramas

Master

Se debe crear la rama **master**, en esta rama se debe crear un proyecto de Nodejs el cual tenga como petición de tipo GET con la ruta raíz (http://localhost:3000/) la información de cada uno de los integrantes del grupo como se detalla en la imagen. A su vez se debe crear la rama **develop**, en la que se agruparan todas las funcionalidades a lo largo del desarrollo de la práctica.

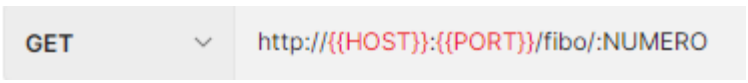




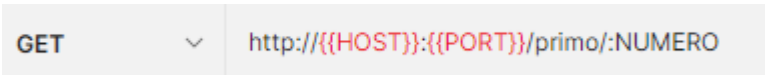
Features

Se debe crear una rama con nombre del siguiente reléase seguido de un guion y el número de carné del estudiante (ejemplo: “f/1.0.0-202100000”). En esta rama se debe crear una petición de tipo GET, la cual retorne “palíndromo valido”, “palíndromo invalido” y evalué que una palabra es palíndromo. Se debe hacer un merge a la rama master y crear una rama con nombre de la versión del reléase (ejemplo “1.0.0”), también se debe crear el tag correspondiente.

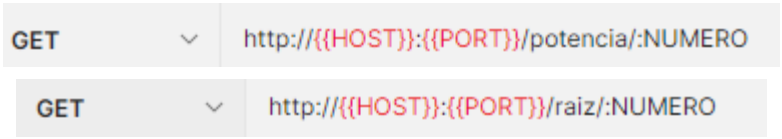
Se debe crear una rama con nombre del siguiente reléase seguido de un guion y el número de carné del estudiante (ejemplo: “f/1.1.0-202100000”). Se debe crear una petición de tipo GET, la cual retorna el resultado de la función del cálculo de Fibonnacci de un número. Se debe hacer un merge a la rama master y crear una rama con nombre de la versión del reléase, también se debe crear el tag correspondiente.



A partir de la rama anterior debe crear otra rama con nombre del siguiente reléase seguido de un guión y el número de carné del estudiante. Esta vez otro desarrollador, sobre el mismo archivo debe implementar una petición de tipo GET para verificar si un número es primo o no. Se debe hacer un merge a la rama anterior y posteriormente a la rama master y crear una rama con nombre de la versión del reléase, también se debe crear el tag correspondiente.



Se debe crear una rama con nombre del siguiente reléase seguido de un guion y el número de carné del estudiantes. Dos desarrolladores deben agregar las peticiones de tipo GET que son **potencia al cubo** y **raíz cúbica**. Se debe hacer un merge a la rama master y crear una rama con nombre de la versión del reléase, también se debe crear el tag correspondiente.



En la rama del reléase anterior se agrega una interfaz gráfica desplegada en el puerto 4000 para esta interfaz se puede implementar cualquier herramienta de FrontEnd (Angular, React, Vue) esta interfaz debe ser agradable y sencilla de comprender para el usuario implementando cada una de las peticiones antes desarrolladas. Se debe hacer un merge a la rama master y crear una rama con nombre de la versión del reléase, también se debe crear el tag correspondiente.

Hot Fix

En la rama del reléase anterior se agrega las peticiones tipo GET multiplicación y división. Se debe hacer un merge a la rama master y crear una rama con nombre de la versión del reléase, también se debe crear el tag correspondiente. En la peticion de división no se ha agregado la funcionalidad de no dividir entre cero. Esto se hará creando una rama de hotfix llamada “VersionActual/hotfix_Division”. Se debe hacer un merge a la rama master y crear una rama con nombre de la versión del reléase, también se debe crear el tag correspondiente.

BugFix’s

Se creará una rama de bugFix por cada estudiante, agregándole una funcionalidad extra a cada función.

Se debe crear un Workflow de cómo quedarían las ramas con sus respectivos nombres y describiendo las acciones que se realizaron incluyendo el número de carné del estudiante que lo realizó. Se recomienda utilizar Draw.io este diagrama deberá estar incluido en un manual técnico incluyendo la descripción de las acciones realizadas en el repositorio.

Gitflow - Workflow

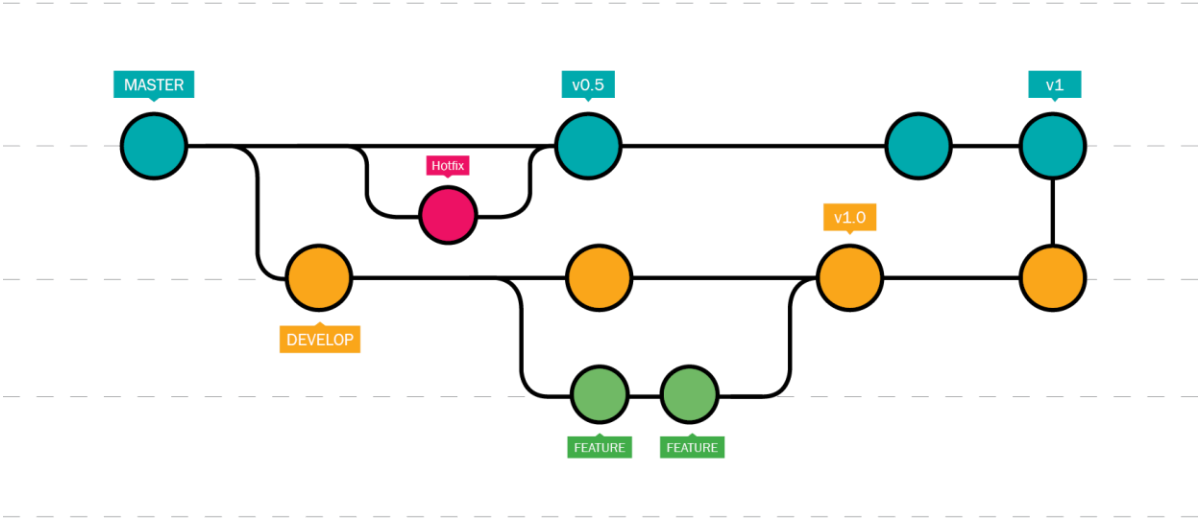


Imagen 1: Ejemplo de flujo de ramas.

Consideraciones a tomar en cuenta

- Se designa un responsable quien deberá hacer la integración. Así mismo deberá hacer el merge al Master y crear los branch de reléase y tag.

- Se deberá trabajar con los grupos formados en la clase.

- Agregar el siguiente usuario de github: @mdmata gitlab: @cesar-sazo como colaborador para revisión de commit trabajados.

- El lenguaje en el cual están desarrolladas las funciones es a elección del grupo de trabajo de la clase.

- Cada “commit” debe tener un mensaje describiendo la acción, al final de cada mensaje debe ir incluido el carné.

- Dudas y preguntas serán respondidas en el correo electrónico o foros de UEDI.

- Se realizarán las acciones básicas y avanzadas al momento de la calificación.

- Copias totales y parciales tendrá una nota de 0 y será reportada a la escuela de sistemas.

- Se debe enviar el manual de técnico un día antes de la calificación.

- Los commit deben iniciar al menos 10 días antes de la entrega como mínimo para no recibir penalización.

- Entregables:
    - Documento PDF con nombre: [AyDS1]Practica1_#grupo.pdf
    - Entregar vía UEDI (una persona por grupo).

Fecha de entrega: Miercoles, 23 de febrero de 2022; a las 23:59.
Fecha y hora de calificación: Jueves, 24 de febrero de 2022, el horario está pendiente.
Lugar: Google Meet.