



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Mecatrónica  
Microprocesadores y microcontroladores

## **Tarea #2**

Profesor:  
Rodolfo Piedra

Estudiantes:  
Natalia Gil Vilorio  
Jervis Villalobos Villegas

II Semestre 2019

## Preguntas teóricas

### 1. Que es Git?

GIT es un software de control de versiones, fue diseñado con un énfasis en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Git proporciona las herramientas para desarrollar un proyecto en equipo de manera inteligente y rápida. Un control de versiones es un sistema que guarda los cambios en un archivo o conjunto de archivos de forma que se pueda acceder a versiones específicas posteriormente. Las características principales de GIT incluyen:

- Rapidez en la gestión de ramas, un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
- Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
- Gestión eficiente de proyectos grandes.
- Realmacenamiento periódico en paquetes.

### 2. Que es Github?

GitHub es una plataforma de **desarrollo colaborativo de software** para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, o forma privada, con una cuenta pagada. GitHub sirve para alojar un repositorio de código y brinda **herramientas** útiles para el **trabajo en equipo**, dentro de un proyecto.

### 3. Que es un branch?

La traducción literal sería: *rama*. Es decir, dentro de nuestro sistema de control de versiones podemos ver el histórico de cambios como si de un árbol se tratara. De esta forma podemos ir abriendo ramas que parten bien de la rama principal (master) o de otra rama (branch).

### 4. Que es un commit?

Un commit es cada modificación que se realiza al repositorio. Un objeto commit almacena metadatos por cada cambio introducido al repositorio incluyendo el autor, la fecha y el mensaje. Cada commit apunta a un objeto del árbol que captura en una “fotografía” (snapshot), el estado del repositorio en el momento que el commit se realiza. Además, la acción commit se refiere a guardar o subir los archivos a un repositorio remoto (una

actualización de los cambios) también puede hacerse al local. El comando `git commit` guarda los cambios hechos a los archivos en el repositorio.

5. Que se entiende cuando se dice que un archivo está “staged”?

Git tiene tres estados principales en los que se pueden encontrar tus archivos: confirmado (committed), modificado (modified), y preparado (staged). Preparado significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.

6. Que hace el comando `git checkout`?

El comando `checkout` se puede usar para crear ramas o cambiar entre ellas. Por ejemplo, el siguiente comando crea una nueva y se cambia a ella:

```
command git checkout -b <branch-name>
```

Para cambiar de una rama a otra solo usa:

```
git checkout <branch-name>
```

7. Que hace el comando `git stash`?\*

Guarda las modificaciones locales en temporal y vuelve el directorio de trabajo a un estado inicial (como si no se hubiese hecho ningún cambio en la rama).

8. Que hace el comando `git add`?\*

Este comando puede ser usado para agregar archivos al index. Por ejemplo, el siguiente comando agrega un nombre de archivo `temp.txt` en el directorio local del index:

```
git checkout <branch-name>
```

9. Que es Pytest?

Pytest es un marco de pruebas que permite escribir códigos de prueba usando Python. Se caracteriza por su simplicidad y fácil sintaxis, puede correr pruebas en paralelo, puede correr una prueba específica o un subconjunto de pruebas y detecta automáticamente las funciones de prueba.

10. Bajo el contexto de pytest. Que es un “assert”?\*

Pytest le permite usar “assert” de python para verificar las expectativas y los valores en las pruebas de Python.

#### 11. Que es flake8?

Flake8 es una biblioteca de Python que envuelve PyFlakes, pycodestyle y el script McCabe de Ned Batchelder. Es un gran juego de herramientas para verificar su base de código contra el estilo de codificación (PEP8), errores de programación (como "biblioteca importada pero no utilizada" y "Nombre indefinido") y para verificar la complejidad ciclomática.

#### Referencias

- Bustos, G. (2018). Comandos básicos de GIT. Retrieved 13 September 2019, from <https://www.hostinger.es/tutoriales/comandos-de-git>
- Chacón, S., & Straub, B. (2014). *Pro Git*. Berkeley, CA: Apress.
- Loeliger, J., & McCullough, M. (2012). *Version control with Git*. Sebastopol, Calif: O'Reilly.
- Oliveira, B. (2018). *Pytest Quick Start Guide*. [S.l.]: Packt Publishing.
- Pérez García, A. (2011). Trabajando con GIT, introducción al uso de los branch y git-completion.bash - Adictos al trabajo. Retrieved 13 September 2019, from <https://www.adictosaltrabajo.com/2011/06/27/git-branch-bash/>
- Software Freedom Conservancy. (2014). Git - Fundamentos de Git. Retrieved 13 September 2019, from <https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>