**Soluciona el siguiente cuestionario.**

César González

**Respuestas en negrita y subrayado**

1. ¿Qué comando se utiliza para clonar un repositorio remoto?

a) git init

**b) git clone**

c) git pull

d) git push

2. ¿Cuál es el propósito del comando `git pull`?

a) Subir cambios al repositorio remoto.

b) Clonar un repositorio.

**c) Descargar y fusionar cambios desde un repositorio remoto.**

d) Crear una nueva rama.

3. ¿Cuál es el comando para crear una nueva rama en Git?

a) git branch

**b) git checkout**

c) git commit

d) git merge

4. ¿Qué hace el comando `git status`?

a) Crea una nueva confirmación.

**b) Muestra el estado de los archivos en el repositorio.**

c) Fusiona ramas.

d) Elimina archivos del repositorio.

5. ¿Cómo se deshacen los cambios no confirmados en Git?

**a) git revert**

b) git reset

c) git checkout

d) git commit

\*\*Verdadero/Falso:\*\*

6. En Git, cada confirmación se identifica por un hash único.

- **Verdadero** / Falso

7. Git es un sistema de control de versiones centralizado.

- **Verdadero** / Falso

8. El comando `git merge` se utiliza para crear una nueva rama.

- Verdadero / **Falso**

9. El comando `git stash` se utiliza para descartar permanentemente los cambios locales.

- Verdadero / **Falso**

10. Git es únicamente útil para programadores y no tiene aplicaciones en otros campos.

- Verdadero / **Falso**

\*\*Preguntas Abiertas:\*\*

11. Describe el proceso de resolución de conflictos en Git cuando se fusionan dos ramas.

***R: La rama quedaría en estado mergeging:***

***- Se identifica los archivos que tienen conflictos.***

***- Se edita los archivos con una herramienta de edición de texto (Visual Code, Notepad, etc) y se selecciona los cambios que sí deben permanecer luego del merge. Los cambios que permanezcan pueden ser uno o todos, esto lo debe definir el arquitecto o responsables del desarrollo.***

***- Luego de seleccionados los cambios que permanecen, se añaden los cambios (git add), y se crea una nueva confirmación (git commit). Esta última es la que se sube a la rama resultado (ejemplo: release).***

12. ¿Por qué es importante crear confirmaciones (`commits`) atómicas y descriptivas en Git?

**Para poder identificar fácilmente cambios específicos entre el conjunto de todos los cambios aceptados. Facilita la identificación de errores.**

13. Explica el concepto de ramificación (`branching`) en Git y cómo puede ser beneficioso en el desarrollo de software.

**R: Permite separar los desarrollos y dividir el trabajo entre programadores.**

14. ¿Qué es un repositorio remoto en Git y cómo se conecta a un repositorio local?

**R: repositorio remoto es el lugar donde residen los cambios/proyectos/código fuente en un servidor o nube. Un repositorio local es una copia de este repositorio remoto y permite al desarrollador dueño de dicho repositorio local hacer cambios y pruebas lcales antes de unirlos a los elementos en el repositorio remoto.**

15. Menciona algunos posibles casos de uso para el archivo `.gitignore`.

**R: Por ejemplo:**

* **Archivos de configuración, que pueden ser únicos para cada repositorio local y no deben ser modificados en el repositorio remoto.**
* **Archivos de prueba o compilados que son generados en el repositorio local y no son necesarios en el repositorio remoto.**

16. ¿Qué es un conflicto de fusión (`merge conflict`) y cómo se resuelve?

**R: Cuando al unir 2 o más cambios generados por desarrollos realizados a elementos comunes y la herramienta git no puede determinar cuál de los 2 cambios debe permanecer al final del proceso. Aquí se genera un conflicto, que debe ser resuelto por la persona responsable o responsables del código fuente.**

**Se resuelve siguiendo los pasos:**

***- Se identifica los archivos que tienen conflictos.***

***- Se edita los archivos con una herramienta de edición de texto (Visual Code, Notepad, etc) y se selecciona los cambios que sí deben permanecer luego del merge. Los cambios que permanezcan pueden ser uno o todos, esto lo debe definir el arquitecto o responsables del desarrollo.***

***- Luego de seleccionados los cambios que permanecen, se añaden los cambios (git add), y se crea una nueva confirmación (git commit). Esta última es la que se sube a la rama resultado (ejemplo: release).***

17. ¿Cómo se crea una etiqueta (`tag`) en Git y en qué escenarios se suele utilizar?

**R: Se con el comando add tag y se utiliza para marcar versiones finales de cambios dentro de un grupo de cambios en un repositorio.**

18. Describe el flujo de trabajo básico en Git, desde la creación de un repositorio hasta la colaboración en equipo.

**R: Para un flujo de trabajo básico, se define las siguientes ramas:**

* **Master/main: rama principal que representa el código fuente existente en producción.**
* **Reléase: rama que contiene el código estable que será impactado contra master y donde convergen todos los desarrollos de un proyecto.**
* **Feature: rama de nuevas características individuales que serán agrupadas posteriormente una reléase para su validación.**
* **Development: rama copia de master donde se agregan todos los cambios con el propósito de ser probada en ambientes pre-productivos y pruebas.**

**El flujo iniciaría con una rama feature donde los programadores agregan sus cambios para hacer pruebas unitarias. Estos cambios son impactados contra development para las pruebas de certificación del aplicativo. Los cambios por proyectos son agrupados en los releases previo a la liberación a producción. Por último, los cambios certificados son combinados en la rama main/master y productiva.**

19. ¿Cuál es el propósito del comando `git rebase` y cuáles son sus implicaciones?

**R: funciona para mover un commit de manera que se inserte antes o al final de una serie de cambios durante el proceso de merge. Con esto se logra evitar que salten conflictos al momento de mezclar cambios y modificar el orden de los cambios al momento de leer los logs.**

20. Explica qué es el flujo de trabajo Gitflow y cómo se estructuran las ramas en este enfoque.

**R: Flujo de trabajo en Git es determinado por los responsables del repositorio y sirve para llevar un procedo ordenado y control de los cambios que son enviados a las ramas principales. Las ramas se definen de acuerdo con su funcionalidad dentro del flujo: como ramas productivas, de trabajo o de nuevas funcionalidades.**