



## **Programacion 3**

### **Tarea GIT**

**Calificación:** la que indica la plataforma.

#### **Punto1:**

#### **Desarrolla el siguiente Cuestionario**

1. ¿Qué es Git?
2. ¿Cuál es el propósito del comando git init en Git?
3. ¿Qué representa una rama en Git y cómo se utiliza?
4. ¿Cómo puedo determinar en qué rama estoy actualmente en Git?
5. ¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado?
6. ¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan?
7. ¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?

## Punto 2:

### Desarrolle un ejercicio práctico en Azure Devops o GitHub con las siguientes características

#### Link con respuestas:

- Utiliza OneDrive para la entrega de la teoría o sube el documento al repositorio del proyecto.
- Comparte el documento y asegúrate de configurarlo como público o proporciona acceso al profesor y al monitor. (Si das los accesos no se corregirá)

#### Link del proyecto en GitHub o Azure Devops:

#### Tareas o requerimientos (Seguir los pasos indicados):

Crea issues para cada tarea.

En el ejemplo explicado verá que la estructura que debe tener el proyecto será la siguientes:

Branch- **main** -> Es la rama principal o rama productiva.

**QA** - > Esta es la rama utilizada para el entorno de prueba o entorno de QA donde los equipo de calidad son los responsable de hacer toda la prueba necesaria del código en este ambiente.

#### Nota:

En algunos lugares la rama **Main/Master** es **QA**, ya que se utilizan tags para crear la versión productiva del proyecto cuando se terminan las pruebas.

**Dev** -> Rama donde los programadores suben su código para hacer prueba.

#### Funcionalidades ( Feature branches)

1. Ramas para desarrollar nuevas características.
2. Fusionadas en la rama principal después de completar.

feature/nueva-funcionalidad

#### Correcciones Rápidas (Hotfix Branches):

1. Ramas para corregir errores críticos.
2. Fusionadas en la rama principal y en las ramas de funcionalidades activas.

hotfix/error-crítico

**Pr** = Pull request (Este es el nombre que recibe en GITHUB)

**Mr** = Merge Request (Este es el nombre que recibe en azure devops)

**Crear MR o PR:**

Una vez termine de realizar el feature debe de subir el cambio al Dev y luego a QA y luego a Main, se debe de crear un pull request en Github para simular el proceso real que sucede cuando hacemos un feature.

**Importante:**

- Asegúrate de que el repositorio no esté vacío.
- Asegúrate de crear los pull request para ver la evidencia.
- Solo se permite la entrega del código en github o Azure devops
- Verifica que el repositorio sea público o tenga acceso al profesor y monitor.
- Recuerda configurar el documento como público en Google Docs o dar acceso en Word.

**Email de los calificadores:**

Azure Devops/Github/JIRA: **20186927@itla.edu.do**

Azure Devops/Github/JIRA: **ktejada@itla.edu.do**

**Link de ejemplo hecho por el monitor:**

Plataforma Youtube: **{Aqui colocare un video de explicacion}**

**Recordatorios**

- 1- Acceso a los link al profesor y monitor con los link indicado.
- 2- No se puede enviar tarea en ningún momento por email solo por la plataforma.
- 3- Cualquier dudas se hará únicamente por correo al profesor y monitor.
- 4- Respetar las fechas acordadas.
- 5- La nota será 0 si no cumplen con el requerimiento del paso 1, 2, 3 y 4.

## Criterio de evaluación

Permiso al repositorio	10%
Cumplir con la estructura indicada	60%
Crear los PR o MR	30%
En caso de no dar el permiso completo	-100%
Plagio	-100%