|  |  |  |
| --- | --- | --- |
|  | **UNIVERSIDAD NACIONAL DE JOSE C. PAZ** | |
| **Departamento de Industria e Innovación Tecnológica** | |
| **Carrera:** Analista Programador Universitario | |
| **Asignatura:** Opt1 - **Mesa:** Primer Parcial | |
| **Alumno: Mauricio Germán Benitez** | **Fecha:** 06/05/2024 |

### Recomendaciones

Requisitos para aprobar: Para que el parcial esté aprobado deberá tener correctamente desarrollado/ contestado el **60%** del examen.

1. Lea todo el parcial antes de comenzar a responder.

**TEORÍA**

**A -Explique la diferencia entre git add . y git add archivo.**

git add . guarda en el repositorio local los cambios de todos los archivos. En cambio git add archivo guarda solamente los cambios del archivo señalado.

**B -Explique la diferencia entre git checkout -b rama y git checkout rama**

*git checkout –b rama* crea y se posiciona sobre la rama. En cambio *git checkout rama*, solamente se posiciona sobre esa rama existente.

**C -Describa brevemente el propósito de los comandos push y pull en Git.**

push 🡪 sirve para subir los cambios al repositorio. Se usa después de comitear los cambios.

pulll 🡪 sirve para traer los cambios desde el repositorio remoto al local. Se usa cuando nuestro repositorio local no está actualizado con los cambios del repositorio remoto.

**D -¿Que es un hilo?**

Un hilo es un flujo de control dentro de un programa. El uso de hilos permite ejecutar varias tareas de forma simultánea. Se usan para aprovechar mejor el hardware, mejorar la eficiencia del programa y crear aplicaciones mas fáciles de extender. Los hilos funcionan bajo el control de un proceso. Un proceso puede tener uno o varios hilos.

**E - ¿Cuáles son los motivos de que un hilo este en estado not running?**

Un hilo puede estar en estado not running si fue fue detenido intencionalmente de forma temporal. Esto puede producirse por ejemplo, porque fué suspendido, o fue pasado a estado de reposo.

Un hilo en estado de not running puede volver al estado running, a diferencia de lo que pasa con el estado dead, donde el hilo termina su ciclo.

**F-**



**Afirmaciones:**

Afirmación 1: El hilo se detiene después de 5 segundos de ejecución.

FALSO: el Thread se detiene por cinco segundos.

Afirmación 2: Después de detener el hilo, se crea un nuevo hilo y se reinicia la ejecución.

VERDADERO: se crea un nuevo hilo reasignándolo a la variable *hilo* y se lo ejecuta.

Indica si cada afirmación es verdadera o falsa según el comportamiento observado del programa.

Justifica tus respuestas con base en la ejecución del código.

**PRÁCTICA**

**REPOSITORIO DEL PARCIAL**

<https://github.com/MauroMauro/primer-parcial.git>

**1 -Inicie un nuevo repositorio Git llamado "mi\_proyecto".**

Ingresar las credenciales de mail y usuario de github en el bash.

Ubicarse en la carpeta creada para el repositorio y correr git init “primer-parcial”

Generar cambios y guardarlos en el repositorio remoto con git add . && git commit –m “titulo del commit” && git push origin master

**2 -Clone el repositorio "mi\_proyecto"**

Desde la web de github acceder al repositorio creado, copiar la url, ubicarse con el bash en la carpeta donde se quiera clonar el repositorio, correr

*git clone* <https://github.com/MauroMauro/primer-parcial.git>

**3 -Cree un archivo llamado "archivo.txt" y agregue algunas líneas de texto.**

**4 -Agregue "archivo.txt" al área de preparación (staging area).**

Ubicarse en la carpeta del repositorio y correr git add “archivo.txt”

**5 -Realice un commit con el mensaje "Agregando archivo.txt".**

git commit –m "Agregando archivo.txt"

**6 -Suba los cambios al repositorio**

git push origin master

**7 -Cree y cambie a una nueva rama llamada "nueva-caracteristica".**

git checkout –b nueva-caracteristica

**8 -Modifique "archivo.txt" agregando más texto.**

**9 -Agregue los cambios al área de preparación y realice un commit**

**con el mensaje "Agregando nueva característica".**

$ git commit -m "agregando nueva caracteristica"

**10 -Suba los cambios al repositorio**

git push origin nueva-caracteristica

**11 -Cambie de regreso a la rama principal (master/main).**

git checkout main

**12 -Fusione la rama "nueva-caracteristica" con la rama principal.**

git merge nueva-caracteristica

**13 -Resuelva cualquier conflicto que pueda surgir durante la fusión.**

**15-El objetivo de esta tarea es implementar un programa en Java que utilice hilos para leer un archivo de texto de manera concurrente.**

Implementación del Programa Java:

Deben crear un programa Java que utilice hilos para leer el archivo de texto proporcionado.

El programa debe crear al menos un hilo para llevar a cabo la lectura del archivo.

Mientras el hilo o los hilos leen el archivo, el hilo principal debe realizar otras operaciones, como mostrar un mensaje en la consola o realizar cálculos simples.

Firma…32013385 Mauricio Germán Benítez