

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

PRACTICA DE CAMPO 3”

Autores:

Palomino Ramos, Renzo Alexander N00408148

Curso:

Técnicas de programación orientada a objetos

Docente:

Torres Rodríguez, Martin Eduardo

Repositorio:

https://github.com/re286estudentupn-svg/practica_de_campo_3.git

Lima – Perú

2025-2

1. RESUMEN

Se desarrollaron cinco programas en Java que aplican sobrecarga de métodos, manejo de excepciones y uso de colecciones. El control de versiones se realizó con Git y GitHub en el repositorio pública practica_de_campo_3. Se documentan los requerimientos de PC, el procedimiento seguido y el diagrama de causa-efecto de los problemas más frecuentes al gestionar repositorios.

2. OBJETIVO

Aplicar buenas prácticas de programación en Java y control de versiones con GitHub para resolver ejercicios individuales, documentar el entorno utilizado y presentar evidencia del proceso.

3. ALCANCE

El trabajo comprende la implementación y prueba de cinco ejercicios en Java, la carga del código fuente al repositorio, y la elaboración de un breve informe con requerimientos, formato de PC y análisis Ishikawa. No se incluyen pruebas automatizadas ni empaquetado del software.

4. REQUERIMIENTOS DEL PC UTILIZADO

Hardware mínimo: procesador de dos núcleos, 4 GB de RAM y 2 GB libres en disco.

Software: Windows 10 o superior, Java JDK 8 o superior, Git y un IDE (Eclipse o VS Code).

Conectividad: acceso a Internet para clonar, hacer commits y push a GitHub.

5. FORMATO DE PC (LLENADO)

Nombre del equipo: [ej. ALEX-LAP]

Procesador: [ej. Intel Core i5-8250U]

Memoria RAM: [ej. 8 GB]

Almacenamiento: [ej. 256 GB SSD]

Sistema operativo: [ej. Windows 10 Pro 22H2]

Versión de Java: [resultado de java -version]

IDE usado: [Eclipse 2024-xx o VS Code]

Versión de Git: [resultado de git --version]

Observaciones: [breve nota, opcional]

6. PROCEDIMIENTO

7. Creación del repositorio en GitHub con el nombre practica_de_campo_3.

8. Inicialización del repositorio local, vinculación remota y primer commit.

9. Desarrollo de los cinco programas en Java con comentarios explicativos.

10. Pruebas básicas desde consola y desde el IDE.

11. Commit con el mensaje Practica - Sobrecarga, Manejo de Errores y Colecciones y push a la rama main.

12. DESCRIPCIÓN DEL SOFTWARE ENTREGADO

CalculadoraPregunta1.java

Implementa sobrecarga de métodos sumar con diferentes firmas: dos enteros, tres enteros y dos doubles. Incluye demostración en main.

DivisionSeguraPregunta2.java

Solicita dos enteros y realiza la división controlando ArithmeticException por división entre cero y entradas no válidas.

ListaEstudiantesPregunta3.java

Usa ArrayList para agregar al menos cinco nombres, mostrar la lista, eliminar el tercer elemento y volver a mostrar el resultado.

InventarioPregunta4.java

Clase Inventario con métodos sobrecargados para agregar productos por nombre, por nombre-precio y por nombre-precio-cantidad. Valida que precio y cantidad no sean negativos. Usa ArrayList.

NumeroNegativoPregunta5.java

Lee una cadena, intenta convertirla a entero y maneja NumberFormatException. Lanza y captura la excepción personalizada NumeroNegativoException si el número es negativo.

8. REPOSITORIO DE CÓDIGO

Enlace: https://github.com/re286estudentupn-svg/practica_de_campo_3.git

Rama principal: main

Mensajes de commit: se usó un mensaje claro y único para la entrega principal.

9. DIAGRAMA DE ISHIKAWA

Problema central: errores al gestionar repositorios y versiones del código.

Categorías analizadas: métodos, herramientas, conocimiento, comunicación, tiempo y práctica individual.

Conclusión del análisis: los errores se concentran en la falta de convenciones, el uso incorrecto de comandos Git y la baja revisión previa antes de subir cambios.

[insertar la imagen del diagrama]

10. CONCLUSIONES

11. La sobrecarga de métodos facilita interfaces claras cuando cambian los tipos o la cantidad de parámetros.

12. El manejo de excepciones mejora la robustez ante entradas no válidas y operaciones críticas como la división.

13. Las colecciones, en particular ArrayList, simplifican el manejo dinámico de datos.

14. El uso de Git y GitHub permite evidenciar el trabajo realizado y mantener un historial claro de cambios.

15. El análisis Ishikawa ayuda a identificar causas frecuentes de errores de control de versiones y orienta acciones de mejora.

```
Administrador: Símbolo del sistema
C:\Windows\system32>cd C:\Users\ALEX\Desktop
C:\Users\ALEX\Desktop>mkdir practica_de_campo_3
C:\Users\ALEX\Desktop>cd practica_de_campo_3
C:\Users\ALEX\Desktop\practica_de_campo_3>git init
Initialized empty Git repository in C:/Users/ALEX/Desktop/practica_de_campo_3/.git/
C:\Users\ALEX\Desktop\practica_de_campo_3>echo "# practica_de_campo_3" >> README.md
C:\Users\ALEX\Desktop\practica_de_campo_3>echo // Pregunta 1 > CalculadoraPregunta1.java
C:\Users\ALEX\Desktop\practica_de_campo_3>echo // Pregunta 2 > DivisionSeguraPregunta2.java
C:\Users\ALEX\Desktop\practica_de_campo_3>echo // Pregunta 3 > ListaEstudiantesPregunta3.java
C:\Users\ALEX\Desktop\practica_de_campo_3>echo // Pregunta 4 > InventarioPregunta4.java
C:\Users\ALEX\Desktop\practica_de_campo_3>echo // Pregunta 5 > NumeroNegativoPregunta5.java
C:\Users\ALEX\Desktop\practica_de_campo_3>notepad CalculadoraPregunta1.java
C:\Users\ALEX\Desktop\practica_de_campo_3>notepad DivisionSeguraPregunta2.java
C:\Users\ALEX\Desktop\practica_de_campo_3>notepad ListaEstudiantesPregunta3.java
C:\Users\ALEX\Desktop\practica_de_campo_3>notepad InventarioPregunta4.java
C:\Users\ALEX\Desktop\practica_de_campo_3>notepad NumeroNegativoPregunta5.java
```

```

C:\Users\ALEX\Desktop\practica_de_campo_3>git add .

C:\Users\ALEX\Desktop\practica_de_campo_3>git commit -m "Practica - Sobrecarga, Manejo de Errores y Colecciones"
[master (root-commit) e84ea7b] Practica - Sobrecarga, Manejo de Errores y Colecciones
 6 files changed, 241 insertions(+)
 create mode 100644 CalculadoraPregunta1.java
 create mode 100644 DivisionSeguraPregunta2.java
 create mode 100644 InventarioPregunta4.java
 create mode 100644 ListaEstudiantesPregunta3.java
 create mode 100644 NumeroNegativoPregunta5.java
 create mode 100644 README.md

C:\Users\ALEX\Desktop\practica_de_campo_3>git remote add origin https://github.com/re286studentupn-svg/practica_de_campo_3.git

C:\Users\ALEX\Desktop\practica_de_campo_3>git branch -M main

C:\Users\ALEX\Desktop\practica_de_campo_3>git push -u origin main
remote: Repository not found.
fatal: repository 'https://github.com/re286studentupn-svg/practica_de_campo_3.git/' not found

C:\Users\ALEX\Desktop\practica_de_campo_3>git remote remove origin

C:\Users\ALEX\Desktop\practica_de_campo_3>git remote add origin https://github.com/re286studentupn-svg/practica_de_campo_3.git

C:\Users\ALEX\Desktop\practica_de_campo_3>git branch -M main

remote: Repository not found.
fatal: repository 'https://github.com/re286studentupn-svg/practica_de_campo_3.git/' not found

C:\Users\ALEX\Desktop\practica_de_campo_3>C:\Users\ALEX\Desktop\practica_de_campo_3>
La sintaxis del comando no es correcta.

C:\Users\ALEX\Desktop\practica_de_campo_3>git remote remove origin

C:\Users\ALEX\Desktop\practica_de_campo_3>git remote add origin https://github.com/re286studentupn-svg/practica_de_campo_3.git

C:\Users\ALEX\Desktop\practica_de_campo_3>git branch -M main

C:\Users\ALEX\Desktop\practica_de_campo_3>git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 3.57 KiB | 1.79 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/re286studentupn-svg/practica_de_campo_3.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\ALEX\Desktop\practica_de_campo_3>cd C:\Users\ALEX\Desktop\practica_de_campo_3

C:\Users\ALEX\Desktop\practica_de_campo_3>git status
On branch main
Your branch is up to date with 'origin/main'.

```

Palomino R.

