



INSTITUTO TECNOLÓGICO SUPERIOR QUITO METROPOLITANO

CARRERA TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE

Resolución Caso de Estudio Empresarial Previo a la Aprobación de la Asignatura: LENGUAJE DE PROGRAMACION.

Resolución Estudio de Caso: Micelanía de Ejercicios.

Danny Favian Ledesma Valencia



RESUMEN:

NetBeans es un entorno de desarrollo integrado libre, orientado principalmente al desarrollo de aplicaciones Java.

La plataforma NetBeans permite el desarrollo de aplicaciones estructuradas mediante un conjunto de componentes denominados "módulos". Cada uno de estos módulos sería un archivo Java conteniendo un conjunto de clases que interactuarán con las APIs de NetBeans. El objetivo de esta arquitectura es favorecer el desarrollo de funcionalidades de forma independiente y la reutilización de componentes.

Las principales características de NetBeans son las siguientes:

Proporciona una base modular y extensible para el desarrollo de aplicaciones, la Plataforma NetBeans.

Esta plataforma incluye servicios para el control del interfaz de usuario, la configuración, el almacenamiento, las ventanas, etc.

El IDE está desarrollado con la misma metodología modular, por lo que puede extenderse incluyendo módulos con funcionalidades determinadas.

Aunque está ideado para el desarrollo Java, permite el desarrollo en otros lenguajes, como PHP o Python, mediante paquetes adicionales.

En resumen, NetBeans es un IDE versátil y de código abierto que brinda a los desarrolladores herramientas poderosas para escribir, depurar y gestionar proyectos de software en diversos lenguajes de programación, con un enfoque particular en Java y tecnologías relacionadas.

.



ABSTRACT:

NetBeans is a free integrated development environment, mainly oriented to the development of Java applications.

The NetBeans platform allows the development of structured applications through a set of components called "modules". Each of these modules would be a Java file containing a set of classes that will interact with the NetBeans APIs. The objective of this architecture is to favor the development of functionalities independently and the reuse of components.

The main features of NetBeans are the following:

It provides a modular and extensible foundation for application development, the NetBeans Platform.

This platform includes services to control the user interface, configuration, storage, windows, etc.

The IDE is developed with the same modular methodology, so it can be extended by including modules with certain functionalities.

Although it is designed for Java development, it allows development in other languages, such as PHP or Python, through additional packages.

In short, NetBeans is a versatile, open source IDE that provides developers with powerful tools for writing, debugging, and managing software projects in various programming languages, with a particular focus on Java and related technologies.



Tabla de contenido

INTRODUCCIÓN O RESUMEN	1
OBJETIVO GENERAL:	2
MARCO TEÓRICO:	3
METODOLOGÍA:	4
ANÁLISIS DE RESULTADOS	5
CONCLUSIONES6	
RECOMENDACIONES	7



Tabla de los gráficos.

Paso 1: Crear Base de Datos.	1
Paso 2: Crear Tablas Dinámicas	2
Paso 3: Definir la Hoja del DashBoard	3
Paso 4: Crear Gráficos Dinámicos	
Paso 5: Tarjetas o Tarjetones	5
Paso 6: Insertar Segmentación de Datos y Escala de tiempo	6
Paso 7: Diseño al DashBoard	7



INTRODUCCIÓN O RESUMEN:

Java es un lenguaje de programación versátil y ampliamente utilizado que ha dejado una huella significativa en la industria del software desde su creación en la década de 1990. Desarrollado por Sun Microsystems, y ahora mantenido por Oracle Corporation, Java se ha convertido en una herramienta fundamental en la construcción de aplicaciones y sistemas en una variedad de plataformas y entornos.

Lo que hace que Java destaque es su enfoque en la portabilidad, la seguridad y la facilidad de uso. A través del concepto "Write Once, Run Anywhere" (Escribe una vez, ejecuta en cualquier lugar), Java permite que el código fuente escrito en este lenguaje se ejecute en diferentes sistemas operativos y arquitecturas de hardware sin necesidad de modificaciones significativas.

Java se ha utilizado ampliamente en una variedad de campos, desde desarrollo de aplicaciones de escritorio hasta aplicaciones web, móviles y empresariales de gran envergadura. Su diseño orientado a objetos, su sistema de manejo de excepciones y su enfoque en la seguridad han contribuido a su popularidad y a su capacidad para abordar una amplia gama de desafíos de programación.

Algunos de los puntos destacados de Java incluyen:

Orientación a objetos: Java es un lenguaje de programación orientado a objetos, lo que significa que se basa en la organización de datos y funcionalidades en objetos reutilizables, lo que promueve la modularidad y el diseño eficiente del software.

Portabilidad: Gracias a la Máquina Virtual de Java (JVM), el código Java puede ser compilado en un formato intermedio que se ejecuta en diferentes plataformas, lo que brinda una notable portabilidad.

Seguridad: Java ha sido diseñado con un enfoque en la seguridad, lo que permite la ejecución segura de aplicaciones en entornos controlados y no confiables, como navegadores web.

Librerías estándar: Java proporciona una amplia colección de bibliotecas estándar (API)



que facilitan tareas comunes de programación, desde manipulación de cadenas hasta entrada/salida y comunicación en red.

Comunidad activa: La comunidad de desarrollo de Java es vasta y activa, lo que ha llevado a la creación de una gran cantidad de recursos, tutoriales y herramientas para ayudar a los programadores a aprovechar al máximo el lenguaje.

En resumen, Java es un lenguaje de programación influyente que ha dejado una marca indeleble en la industria del software debido a su portabilidad, seguridad y versatilidad. Desde aplicaciones de escritorio hasta sistemas empresariales y aplicaciones móviles, Java continúa siendo una opción confiable para desarrolladores que buscan crear software robusto y eficiente.

OBJETIVO GENERAL:

El objetivo general de Java como lenguaje de programación es proporcionar un entorno confiable, portátil, seguro y de alto rendimiento para el desarrollo de una amplia gama de aplicaciones de software.

MARCO TEÓRICO:

El marco teórico de NetBeans aborda los conceptos y fundamentos clave detrás de este entorno de desarrollo integrado (IDE) de código abierto.

IDE y Desarrollo de Software:

Explicación del concepto de un Entorno de Desarrollo Integrado (IDE) y su importancia en la programación de software.

Discusión sobre cómo un IDE simplifica el proceso de desarrollo al proporcionar herramientas, editores de código, depuración y otras funcionalidades en un solo lugar.

Historia y Evolución de NetBeans:

Orígenes de NetBeans en Sun Microsystems y su desarrollo inicial.

Transición de NetBeans a un proyecto de código abierto bajo la Apache Software Fundación.

Descripción de las versiones clave y las características introducidas en cada una.

Arquitectura y Componentes de NetBeans:

Desglose de la arquitectura de NetBeans, incluyendo su diseño modular y basado en plugin.

Explicación de los componentes principales, como el sistema de ventana, el gestor de proyectos y el editor de código.



Introducción a la estructura de los plugins y cómo permiten la extensibilidad del IDE.

Soporte de Lenguajes y Tecnologías:

Enumeración de los lenguajes de programación compatibles con NetBeans, con un enfoque en Java.

Descripción de cómo NetBeans proporciona soporte para otros lenguajes como HTML, PHP, C/C++, etc.

Mención de las tecnologías y frameworks que NetBeans puede integrar, como Java EE (Enterprise Edition) y herramientas de desarrollo web.

Características y Herramientas Clave:

Resumen de las características esenciales, como el editor de código con resaltado de sintaxis y autocompletado.

Descripción de las capacidades de depuración y perfilado que ayudan a identificar y resolver errores en el código.

Detalles sobre el generador de GUI para el diseño de interfaces gráficas de usuario.

Gestión de Proyectos y Control de Versiones:

Explicación de cómo NetBeans facilita la creación, configuración y gestión de proyectos.

Introducción a la integración de sistemas de control de versiones como Git y Subversion en el IDE.

Comunidad y Recursos:

Descripción de la comunidad de usuarios y desarrolladores de NetBeans.

Mención de recursos útiles como tutoriales, documentación y foros de soporte.

Ejemplos de proyectos notables desarrollados con NetBeans.

Perspectivas Futuras y Desarrollos Continuos:

Discusión sobre las tendencias actuales y futuras en el desarrollo de software y cómo NetBeans se adapta a ellas.

Mención de planes futuros de desarrollo y mejoras para el IDE.

METODOLOGÍA:

NetBeans se lo realizo los ejercicios que se presentan a continuación:



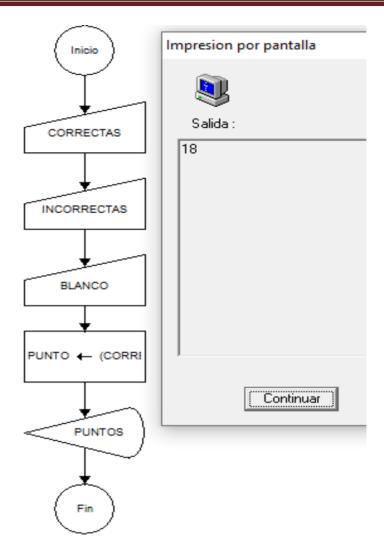
Actividad 1: Resuelva los siguientes ejercicios utilizando PseInt y DFD, debe proporcionar una captura de pantalla del código y la ejecución.

Ejercicio 1 (1 Punto)

Elabore un algoritmo que solicite el número de respuestas correctas, incorrectas y en blanco, correspondientes a un postulante para ingresar en la carrera de ingeniería. AL finalizar, debe mostrar el puntaje final, considerando que por cada respuesta correcta tendrá 4 puntos, por cada respuesta incorrecta tendrá -1 y por cada respuesta en blanco tendrá 0.

```
PSeInt - Ejecutando proceso EJERCICIO3
Algoritmo EJERCICIO3
                                                                 *** Ejecución Iniciada. ***
   DEFINIR CORRECTAS, INCORRECTAS, BLANCO, PUNTOS COMO ENTEROS;
                                                                INGRESE EL NUMERO DE RESPUESTAS CORRECTAS
   ESCRIBIR "INGRESE EL NUMERO DE RESPUESTAS CORRECTAS";
                                                                > 5
   LEER CORRECTAS;
                                                                INGRESE EL NUMERO DE RESPUESTAS INCORRECTAS
   ESCRIBIR "INGRESE EL NUMERO DE RESPUESTAS INCORRECTAS";
                                                                > 2
   LEER INCORRECTAS;
                                                                INGRESE EL NUMERO DE RESPUESTAS BLANCO
   ESCRIBIR "INGRESE EL NUMERO DE RESPUESTAS BLANCO";
   LEER BLANCO;
                                                                PUNTAJE FINAL ES18
   PUNTOS=(CORRECTAS*4)+(INCORRECTAS*(-1));
                                                                *** Ejecución Finalizada. ***
   ESCRIBIR "PUNTAJE FINAL ES", PUNTOS;
FinAlgoritmo
```







Ejercicio 2 (1 Punto)

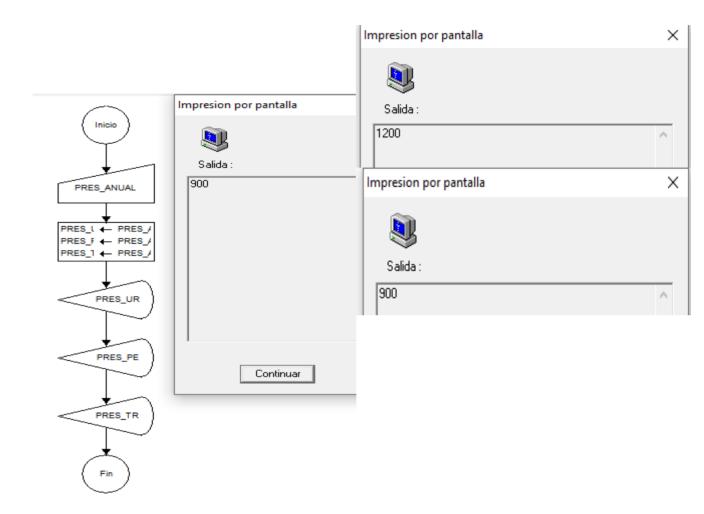
En un hospital hay 3 áreas: Urgencias, Pediatría y Traumatología. El presupuesto anual del hospital se reparte de la siguiente manera:

- Urgencias: 30% Pediatría: 40%
- Traumatología 30%

Elabora un algoritmo que permita obtener la cantidad de dinero que recibirá cada área para cualquier monto presupuestal.

```
Algoritmo EJERCICIO5
      Definir TOTAL Como Real;
      Escribir "ESCRIBA EL PRESUPUESTO ANUAL";
      leer TOTAL:
     Escribir "A LA AREA DE URGENCIAS LE CORRESPONDE $", total * 0.30;
      Escribir "A LA AREA DE PEDIATRIA LE CORRESPONDE $", total *0.40;
      Escribir "A LA AREA DE TRAUMATOLOGIA LE CORRESPONDE $".total *0.30
                                                                 ×
PSeInt - Ejecutando proceso EJERCICIO5
*** Ejecución Iniciada. ***
ESCRIBA EL PRESUPUESTO ANUAL
> 3000
A LA AREA DE URGENCIAS LE CORRESPONDE $900
A LA AREA DE PEDIATRIA LE CORRESPONDE $1200
A LA AREA DE TRAUMATOLOGIA LE CORRESPONDE $900
*** Ejecución Finalizada. ***
```







Actividad 2: Resuelva los siguientes ejercicios utilizando JAVA, debe proporcionar una captura de pantalla del código y su ejecución desde consola.

Ejercicio 1 (0.50 Puntos)

Desarrolle un programa que permita calcular el cambio de moneda de dólares a euros.

```
public class CambioMoneda {
   public static void main(String[] args) {
        Scanner sc = new Scanner(source: System.in);
        System.out.print(s: "Ingresa la cantidad en dolares: ");
        double dolares = sc.nextDouble();
        // Tasa de cambio actual (Dólares a Euros)
        double tasaCambio = 0.85;
        double euros = dolares * tasaCambio;
        System.out.println(dolares + " dolares son equivalentes a " + euros + " euros.");
}
```

```
er2.CambioMoneda  euros  euros  

TALLER2 (run) ×

run:
Ingresa la cantidad en dolares: 50

50.0 dolares son equivalentes a 42.5 euros.
BUILD SUCCESSFUL (total time: 16 seconds)
```

Ejercicio 2 (0.50 Puntos)

Elabore un programa que lea una distancia en metros y la transforme a pies y pulgadas



```
public class ConversionDistancia {
   public static void main(String[] args) {
        Scanner sc = new Scanner(source: System.in);
        System.out.print(s: "Ingresa la distancia en metros: ");
        double metros = sc.nextDouble();
        // 1 metro = 3.28084 pies
        double pies = metros * 3.28084;
        // 1 pie = 12 pulgadas
        double pulgadas = pies * 12;
        System.out.println(metros + " metros son equivalentes a " + pies + " pies.");
        System.out.println(metros + " metros son equivalentes a " + pulgadas + " pulgadas.");
    }
}
```

```
er2.ConversionDistancia  pulgadas  un:

TALLER2 (run) ×

run:

Ingresa la distancia en metros: 2
2.0 metros son equivalentes a 6.56168 pies.
2.0 metros son equivalentes a 78.74016 pulgadas.

BUILD SUCCESSFUL (total time: 28 seconds)
```

Ejercicio 3 (0.50 Puntos)

Elabore un programa que permita transformar el peso en kilos de una persona en gramos, libras y toneladas.

```
public class ConversionPeso {
    public static void main(String[] args) {
        Scanner sc = new Scanner(source: System.in);
        System.out.print(s: "Ingresa el peso en kilogramos: ");
        double kilos = sc.nextDouble();
        // Conversiones
        double gramos = kilos * 1000;
        double libras = kilos * 2.20462;
        double toneladas = kilos * 0.001;
        System.out.println(kilos + " kilogramos son equivalentes a:");
        System.out.println(gramos + " gramos");
        System.out.println(libras + " libras");
        System.out.println(toneladas + " toneladas");
    }
}
```



Ejercicio 4 (0.50 Puntos)

Elabore un programa que permita leer la temperatura en grados Celsius, la transforme y muestre los resultados en grados Fahrenheit.

Actividad 3: Resuelva los siguientes ejercicios utilizando JAVA, debe proporcionar una captura de pantalla del código y su ejecución desde consola.

Ejercicio 1 (1 Punto)

BUILD SUCCESSFUL (total time: 16 seconds)

45.0 grados Celsius son equivalentes a 113.0 grados Fahrenheit.

En una librería se hacen los siguientes descuentos:

Si el cliente compra menos de 5 libros, se le da un descuento del 10% sobre la compra; Si el número de libros es mayor o igual a 5, pero menos de 10, se le otorga 20% de descuento; Y si son 10 o más libros, se le da 40% de descuento. Haga un programa que determine cuánto debe pagar un cliente si el valor de cada libro es de \$80.



```
public class CalculoDescuentoLibros {
     public static void main(String[] args) {
         Scanner sc = new Scanner (source: System.in);
         final double PRECIO POR LIBRO = 80.0;
         System.out.print(s: "Ingresa la cantidad de libros que el cliente compro: ")
         int cantidadLibros = sc.nextInt();
         double subtotal = cantidadLibros * PRECIO_POR_LIBRO;
         double descuento = 0.0;
         if (cantidadLibros < 5) {
             descuento = subtotal * 0.10;
         } else if (cantidadLibros < 10) {
             descuento = subtotal * 0.20;
         } else {
             descuento = subtotal * 0.40;
         double totalPagar = subtotal - descuento;
         System.out.println("Subtotal: $" + subtotal);
         System.out.println("Descuento: $" + descuento);
         System.out.println("Total a pagar: $" + totalPagar);
er2.CalculoDescuentoLibros > 0 main >
ALLER2 (run) #4 × TALLER2 (run) #5 ×
Ingresa la cantidad de libros que el cliente compro: 5
Subtotal: $400.0
Descuento: $80.0
Total a pagar: $320.0
BUILD SUCCESSFUL (total time: 25 seconds)
```

Ejercicio 2 (1 Punto)

Cierta universidad tiene un programa para estimular a los estudiantes con buen rendimiento académico:

- Si el promedio es de 9,5 o más y el alumno es de pregrado, entonces cursará 30 créditos y se le hará un 25% de descuento.
- Si el promedio es mayor o igual a 7,0 pero menor que 9,5 y el alumno es de pregrado, entonces cursará 25 créditos y se le hará un 10% de descuento.
- Si el promedio es mayor que 6,0 y menor que 7,0 y es de pregrado, cursará 20 créditos y no tendrá ningún descuento.
- Si el promedio es mayor o igual a 5,0 y menor que 6,0 y es de pregrado, cursará 15 créditos y no tendrá descuento.
- Si el promedio es menor de 5,0 y es de pregrado, no podrá matricularse.
- Si el promedio es mayor o igual a 8,5 y es de posgrado, cursará 20 créditos y se le hará un 20% de descuento.
- Si el promedio es menor de 8,5 y es de posgrado cursará 15 créditos y no tendrá descuento.

Haga un algoritmo que determine cuánto debe pagar un estudiante y cuántos créditos registra si el valor de cada crédito es de \$500 para pregrado y \$300 para posgrado.



Total a pagar: \$4800.0

BUILD SUCCESSFUL (total time: 41 seconds)

```
public class CalculoPagoCreditos {
        public static void main(String[] args) {
            Scanner sc = new Scanner (source: System.in);
            final double VALOR CREDITO PREGRADO = 500.0;
            final double VALOR CREDITO POSGRADO = 300.0;
            System.out.print(s: "Ingresa el promedio academico del estudiante: ");
            double promedio = sc.nextDouble();
            System.out.print(s: "Es el estudiante de pregrado (S/N): ");
            char esPregrado = sc.next().charAt(index: 0);
            int creditos = 0;
            double descuento = 0.0;
            if (esPregrado == 'S' || esPregrado == 's') {
                 if (promedio >= 9.5) {
                  creditos = 30;
                  descuento = 0.25;
              } else if (promedio >= 7.0) {
                  creditos = 25;
                  descuento = 0.10;
              } else if (promedio > 6.0) {
                  creditos = 20;
               else if (promedio >= 5.0) {
                  creditos = 15:
              } else {
                  System.out.println(x: "El estudiante no puede matricularse.");
                  return:
             if (promedio >= 8.5) {
                 creditos = 20;
                 descuento = 0.20;
              } else if (promedio < 8.5) {
                 creditos = 15;
        double valorCredito = (esPregrado == 'S' || esPregrado == 's') ? VALOR CREDITO PREGRADO : VALOR CREDITO F
        double totalPagar = creditos * valorCredito * (1 - descuento):
        System.out.println("Cantidad de creditos: " + creditos);
        System.out.println("Descuento aplicado: " + (descuento * 100) + "%");
        System.out.println("Total a pagar: $" + totalPagar);
Ier2.CalculoPagoCreditos > 🌗 main >
- TALLER2 (run) #6 ×
Ingresa el promedio academico del estudiante: 18
Es el estudiante de pregrado (S/N): 16
Cantidad de creditos: 20
Descuento aplicado: 20.0%
```



Actividad 4: Resuelva los siguientes ejercicios utilizando JAVA, debe proporcionar una captura de pantalla del código y su ejecución desde consola.

Ejercicio 1 (0.5 Puntos)

Elabore un programa que haga el siguiente censo para una empresa de transporte:

- Número de vehículos cuyo modelo sea anterior a 1990.
- Número de vehículos cuyo modelo sea de 1990 o posterior y cuya capacidad sea menor de 40 pasajeros.
- Número de buses cuyo modelo sea posterior a 1990 con capacidad mayor de 40 pasajeros.
- Número de buses cuyo modelo sea anterior a 1990 con capacidad mayor de 40 pasajeros.
- Número de busetas con capacidad menor de 40 pasajeros.
- El total de vehículos de la empresa.

Por cada vehículo la empresa tiene la siguiente información:

- Tipo de vehículo (1: buseta; 2: bus)
- Modelo del vehículo
- Capacidad del vehículo

```
public class CensoEmpresaTransporte {
    public static void main(String[] args) {
       Scanner sc = new Scanner(source: System.in);
       int numVehiculosAntes1990 = 0;
       int numVehiculos1990oDespuesMenos40 = 0;
       int numBuses1990oDespuesMayor40 = 0;
       int numBusesAntes1990Mayor40 = 0;
       int numBusetasMenos40 = 0;
       int totalVehiculos = 0;
       System.out.print(s: "Ingresa la cantidad de vehiculos en la empresa: ");
       int cantidadVehiculos = sc.nextInt();
        for (int i = 1; i <= cantidadVehiculos; i++) {
           System.out.println("Datos del vehiculo #" + i + ":");
            System.out.print(s: "Tipo de vehiculo (1: buseta; 2: bus): ");
           int tipoVehiculo = sc.nextInt();
           System.out.print(s: "Modelo del vehiculo: ");
           int modelo = sc.nextInt();
           System.out.print(s: "Capacidad del vehiculo: ");
            int capacidad = sc.nextInt();
```



```
if (modelo < 1990) {
       numVehiculosAntes1990++;
    if (modelo >= 1990 && capacidad < 40) {
       numVehiculos1990oDespuesMenos40++;
    if (modelo >= 1990 && tipoVehiculo == 2 && capacidad > 40) {
       numBuses1990oDespuesMayor40++;
    if (modelo < 1990 && tipoVehiculo == 2 && capacidad > 40) {
       numBusesAntes1990Mayor40++;
    if (tipoVehiculo == 1 && capacidad < 40) {
        numBusetasMenos40++;
    totalVehiculos++;
System.out.println(x: "Resultado del censo:");
System.out.println("Numero de vehiculos cuyo modelo es anterior a 1990: | + numVehiculosAntes1990);
 System.out.println("Numero de vehiculos cuyo modelo es de 1990 o posterior y capacidad menor a 40 pasajer
 System.out.println("Numero de buses cuyo modelo es de 1990 o posterior y capacidad mayor a 40 pasajeros:
 System.out.println("Numero de buses cuyo modelo es anterior a 1990 y capacidad mayor a 40 pasajeros: " +
 System.out.println("Numero de busetas con capacidad menor a 40 pasajeros: " + numBusetasMenos40);
 System.out.println("Total de vehiculos en la empresa: " + totalVehiculos);
```

```
er2.CensoEmpresaTransporte >> 0 main >> for (int i = 1; i <= cantidadVehiculos; i++) >> capacidad >>
- TALLER2 (run) #7 ×
Ingresa la cantidad de vehiculos en la empresa: 2
Datos del vehiculo #1:
Tipo de vehiculo (1: buseta; 2: bus): 1
Modelo del vehiculo: 1990
Capacidad del vehiculo: 36
Datos del vehiculo #2:
Tipo de vehiculo (1: buseta; 2: bus): 2
Modelo del vehiculo: 1990
Capacidad del vehiculo: 40
Resultado del censo:
Numero de vehiculos cuyo modelo es anterior a 1990: 0
Numero de vehiculos cuyo modelo es de 1990 o posterior y capacidad menor a 40 pasajeros: 1
Numero de buses cuyo modelo es de 1990 o posterior y capacidad mayor a 40 pasajeros: 0
Numero de buses cuyo modelo es anterior a 1990 y capacidad mayor a 40 pasajeros: 0
Numero de busetas con capacidad menor a 40 pasajeros: 1
Total de vehiculos en la empresa: 2
BUILD SUCCESSFUL (total time: 42 seconds)
```



Ejercicio 2 (0.5 Puntos)

Un hotel presta cuatro clases de servicios. Por cada servicio que presta se tienen los siguientes datos: clase de servicio prestado (valores del 1 al 4), jornada en la que se prestó el servicio (M: mañana; T: tarde) y valor del servicio. Al final del día se quiere determinar el valor producido por cada clase de servicio, el número de veces que se prestó cada servicio, el servicio que más veces se prestó y si este se prestó más en la mañana o en la tarde.

```
public class ServicioPrestaHotel {
    public static void main(String[] args) {
       int[][] clasesServicio = new int[4][3];
       // [valor total, veces prestado manana, veces prestado tarde]
       Scanner scanner = new Scanner (source: System.in);
       System.out.print(s: "Ingrese el numero de servicios prestados: ");
       int numServicios = scanner.nextInt();
        for (int i = 0; i < numServicios; i++) {
            System.out.print(s: "Ingrese la clase de servicio (1 al 4): ");
            int clase = scanner.nextInt();
         System.out.print(s: "Ingrese la jornada (M: Manana; T: Tarde): ");
            char jornada = scanner.next().charAt(index: 0);
           System.out.print(s: "Ingrese el valor del servicio: ");
           double valor = scanner.nextDouble();
           int index = clase - 1;
           clasesServicio[index][0] += valor;
           if (jornada == 'M' || jornada == 'm') {
               clasesServicio[index][1]++;
           } else if (jornada == 'T' || jornada == 't') {
               clasesServicio[index][2]++;
       // Encontrar la clase de servicio que se prestó más veces
       int[] servicioMasPrestado = clasesServicio[0];
        for (int i = 1; i < clasesServicio.length; i++) {
            if ((clasesServicio[i][1] + clasesServicio[i][2]) > (servicioMasPrestado[1] + servicioMasPrestado[2])) {
                servicioMasPrestado = clasesServicio[i];
        }
        // Determinar si el servicio más prestado se hizo más en la mañana o en la tarde
        String momentoMasFrecuente = (servicioMasPrestado[1] > servicioMasPrestado[2]) ? "mañana" : "tarde";
        // Imprimir los resultados
        System.out.println(x: "\nValor producido por cada clase de servicio:");
        for (int i = 0; i < clasesServicio.length; i++) {</pre>
            System.out.println("Clase " + (i + 1) + ": $" + clasesServicio[i][0]);
        System.out.println(x: "\nNumero de veces que se presto cada servicio:");
        for (int i = 0; i < clasesServicio.length; i++) {</pre>
            System.out.println("Clase " + (i + 1) + ": " + (clasesServicio[i][1] + clasesServicio[i][2]) + " veces");
        System.out.println(x: "\nServicio mas prestado:");
        System.out.println("Clase " + (getIndex(arr:servicioMasPrestado, arr2D: clasesServicio) + 1) +
                ", se presto " + (servicioMasPrestado[1] + servicioMasPrestado[2]) +
                " veces, mas frecuentemente en la " + momentoMasFrecuente + ".");
    // Función para encontrar el índice de un arreglo bidimensional
    public static int getIndex(int[] arr, int[][] arr2D) {
        for (int i = 0; i < arr2D.length; i++) {</pre>
            if (arr == arr2D[i]) {
                return i:
        return -1:
```



Output - TALLER2 (run) #4 × run: Ingrese el numero de servicios prestados: 10 Ingrese la clase de servicio (1 al 4): 3 Ingrese la jornada (M: Manana; T: Tarde): T Ingrese el valor del servicio:\$ 20 Ingrese la clase de servicio (1 al 4): 2 Ingrese la jornada (M: Manana; T: Tarde): M Ingrese el valor del servicio:\$ 25 Ingrese la clase de servicio (1 al 4): 1 Ingrese la jornada (M: Manana; T: Tarde): M Ingrese el valor del servicio:\$ 25 Ingrese la clase de servicio (1 al 4): 4 Ingrese la jornada (M: Manana; T: Tarde): T Ingrese el valor del servicio:\$ 20 Ingrese la clase de servicio (1 al 4): 1 Ingrese la jornada (M: Manana; T: Tarde): T Ingrese el valor del servicio:\$ 35 Ingrese la clase de servicio (1 al 4): 2 Ingrese la jornada (M: Manana; T: Tarde): 1

Ingrese el valor del servicio:\$ 2

Ingrese el valor del servicio:\$ 23
Ingrese la clase de servicio (1 al 4): 1
Ingrese la jornada (M: Manana; T: Tarde):

Ingrese la clase de servicio (1 al 4): 1
Ingrese la jornada (M: Manana; T: Tarde): T



Actividad 5

Ejercicio 1 (2 Puntos)

Escribir un programa en Java que simule un sistema de calificaciones de un curso de matemáticas. El programa debe permitir al usuario ingresar las notas de los estudiantes y realizar diversas operaciones usando matrices, estructuras de decisión, selección, repetición, acumuladores y contadores.

El programa debe cumplir con las siguientes especificaciones:

- 1. El usuario podrá ingresar el número de estudiantes (n) que desea calificar, donde n debe ser mayor a 0.
- 2. El programa deberá solicitar las notas de cada estudiante y almacenarlas en una matriz de tamaño n x 3, donde cada fila representará a un estudiante y las tres columnas corresponderán a las notas de tres evaluaciones (por ejemplo, parciales).
- 3. Luego de ingresar las notas, el programa deberá calcular y mostrar el promedio de cada estudiante y determinar si el estudiante aprobó o reprobó el curso. Un estudiante se considera aprobado si su promedio es igual o mayor a 60 puntos.
- 4. Además, el programa deberá determinar y mostrar el promedio general del curso.
- 5. Se debe contar la cantidad de estudiantes que aprobaron el curso y aquellos que lo reprobaron.
- 6. El programa debe encontrar y mostrar la nota más alta y la nota más baja de cada evaluación (parcial) y el número de estudiante al que corresponde cada una de ellas.
- 7. El programa debe contar y mostrar cuántos estudiantes obtuvieron una calificación perfecta de 100 en al menos una de las evaluaciones.
- 8. Finalmente, el programa debe permitir repetir el proceso para calificar otro curso o finalizar la ejecución.

Nota: Considerar que las notas ingresadas por el usuario están en el rango de 0 a 100, y el promedio y demás cálculos se realizarán usando números decimales con dos cifras de precisión.



```
public class SistemaCalificaciones {
    public static void main(String[] args) {
        Scanner scanner = new Scanner (source: System.in);
        int numEstudiantes:
        do {
           System.out.print(s: "Ingrese el numero de estudiantes (n > 0): ");
           numEstudiantes = scanner.nextInt();
        } while (numEstudiantes <= 0);
        double[][] notas = new double[numEstudiantes][3];
        double[] promedios = new double[numEstudiantes];
        int aprobados = 0;
        int reprobados = 0;
        int perfectos = 0;
        for (int i = 0; i < numEstudiantes; i++) {
            System.out.println("Ingrese las notas del estudiante " + (i + 1) + ":");
            for (int j = 0; j < 3; j++) {
                System.out.print("Nota " + (j + 1) + ": ");
                notas[i][j] = scanner.nextDouble();
               promedios[i] += notas[i][j];
            promedios[i] /= 3;
            if (promedios[i] >= 60) {
               aprobados++;
            } else {
               reprobados++;
```

```
for (int j = 0; j < 3; j++) {
        if (notas[i][j] == 100) {
           perfectos++;
            break;
    1
double promedioGeneral = 0;
for (double promedio : promedios) {
    promedioGeneral += promedio;
promedioGeneral /= numEstudiantes;
System.out.println(x: "\nResultados:");
System.out.println("Promedio general del curso: " + String.format(format: "%.2f", args: promedioGeneral));
System.out.println("Estudiantes aprobados: " + aprobados);
System.out.println("Estudiantes reprobados: " + reprobados);
System.out.println("Estudiantes con calificación perfecta: " + perfectos);
double[] maxNotas = new double[3];
double[] minNotas = new double[3];
int[] estudianteMax = new int[3];
int[] estudianteMin = new int[3];
for (int j = 0; j < 3; j++) {
   maxNotas[j] = notas[0][j];
    minNotas[j] = notas[0][j];
```



```
estudianteMax[j] = 1;
estudianteMin[j] = 1;

for (int i = 1; i < numEstudiantes; i++) {
    if (notas[i][j] > maxNotas[j]) {
        maxNotas[j] = notas[i][j];
        estudianteMax[j] = i + 1;
    }

    if (notas[i][j] < minNotas[j]) {
        minNotas[j] = notas[i][j];
        estudianteMin[j] = i + 1;
    }
}

for (int j = 0; j < 3; j++) {
    System.out.println("Nota mas alta en evaluacion " + (j + 1) + ": " + maxNotas[j] + " (Estudiante " + estus to the standard of the standard
```

```
Output - TALLER2 (run) ×
\square
      Ingrese el numero de estudiantes (n > 0): 2
\square
      Ingrese las notas del estudiante 1:
Nota 1: 16
     Nota 2: 18
%
      Nota 3: 13
      Ingrese las notas del estudiante 2:
      Nota 1: 17
      Nota 2: 18
      Nota 3: 20
      Resultados:
      Promedio general del curso: 17,00
      Estudiantes aprobados: 0
      Estudiantes reprobados: 2
      Estudiantes con calificacion perfecta: 0
      Nota mas alta en evaluacion 1: 17.0 (Estudiante 2)
      Nota mas baja en evaluacion 1: 16.0 (Estudiante 1)
      Nota mas alta en evaluacion 2: 18.0 (Estudiante 1)
      Nota mas baja en evaluacion 2: 18.0 (Estudiante 1)
      Nota mas alta en evaluacion 3: 20.0 (Estudiante 2)
      Nota mas baja en evaluacion 3: 13.0 (Estudiante 1)
      BUILD SUCCESSFUL (total time: 29 seconds)
```



Ejercicio 2 (1 Punto)

Elabore un programa que permita ingresar información personal de los empleados de una importante empresa, para ello se debe considerar:

- El programa debe admitir el ingreso de N personas
- El programa debe permitir eliminar un registro
- El programa debe imprimir las personas registradas
- El programa debe utilizar ArrayList como estructura de datos.

```
import java.util.ArrayList;
import java.util.Scanner;class Empleado {
   String nombre;
   int edad;
   String puesto;
   public Empleado(String nombre, int edad, String puesto) {
       this.nombre = nombre;
       this.edad = edad;
       this.puesto = puesto;
   }
   @Override
   public String toString() {
       return "Nombre: " + nombre + ", Edad: " + edad + ", Puesto: " + puesto;
public class ProgramaEmpleados {
   public static void main(String[] args) {
       Scanner scanner = new Scanner(source: System.in);
       ArrayList<Empleado> empleados = new ArrayList<>();
       int opcion;
       do {
           System.out.println(x: "\nMenu:");
           System.out.println(x: "1. Agregar empleado");
           System.out.println(x: "2. Eliminar empleado");
```



```
System.out.println(x: "3. Mostrar empleados");
           System.out.println(x: "4. Salir");
           System.out.print(s: "Seleccione una opcion: ");
           opcion = scanner.nextInt();
           scanner.nextLine();
// Consumir el salto de línea después de leer un número
           switch (opcion) {
               case 1:
                   System.out.print(s: "Ingrese el nombre del empleado: ");
                    String nombre = scanner.nextLine();
                    System.out.print(s: "Ingrese la edad del empleado: ");
                   int edad = scanner.nextInt():
                   scanner.nextLine();
// Consumir el salto de línea después de leer un número
                   System.out.print(s: "Ingrese el puesto del empleado: ");
                    String puesto = scanner.nextLine();
                    Empleado empleado = new Empleado(nombre, edad, puesto);
                    empleados.add(e: empleado);
                    System.out.println(x: "Empleado agregado correctamente.");
                   break;
                case 2:
                    if (!empleados.isEmpty()) {
                        System.out.println(x: "Lista de empleados:");
                        for (int i = 0; i < empleados.size(); i++) {</pre>
                            System.out.println((i + 1) + ". " + empleados.get(index: i));
                       System.out.print(s: "Ingrese el numero de empleado a eliminar: ");
                        int indiceEliminar = scanner.nextInt();
                        if (indiceEliminar > 0 && indiceEliminar <= empleados.size()) {</pre>
                        empleados.remove(indiceEliminar - 1);
                             System.out.println(x: "Empleado eliminado correctamente.");
                          } else {
                             System.out.println(x: "Numero de empleado no valido.");
                      1 else {
                         System.out.println(x: "No hay empleados registrados.");
                     break:
                 case 3:
                      if (!empleados.isEmpty()) {
                         System.out.println(x: "Lista de empleados:");
                          for (int i = 0; i < empleados.size(); i++) {</pre>
                              System.out.println((i + 1) + ". " + empleados.get(index: i));
                      } else {
                         System.out.println(x: "No hay empleados registrados.");
                     break;
                 case 4:
                     System.out.println(x: "Saliendo del programa...");
                     System.out.println(x: "Opcion no valida.");
         } while (opcion != 4);
         scanner.close():
```



```
Output - TALLER2 (run) #3 ×
Menu:
      1. Agregar empleado
2. Eliminar empleado
<u>0</u>€
      3. Mostrar empleados
       4. Salir
       Seleccione una opcion: 1
       Ingrese el nombre del empleado: Luis
Ingrese la edad del empleado: 35
       Ingrese el puesto del empleado: Financiero
       Empleado agregado correctamente.
       1. Agregar empleado
       2. Eliminar empleado
       3. Mostrar empleados
       4. Salir
       Seleccione una opcion: 3
       Lista de empleados:
       1. Nombre: Luis. Edad: 35. Puesto: Financiero

    Agregar empleado
    Eliminar empleado

       3. Mostrar empleados
       4. Salir
       Seleccione una opcion:
```

CONCLUSIONES:

Java NetBeans es un lenguaje de programación de alto nivel y de propósito general, que ofrece una gran cantidad de beneficios para los desarrolladores.

Dispone de un gran número de bibliotecas y frameworks para el desarrollo de aplicaciones empresariales y de dispositivos móviles.

RECOMENDACIONES:

La recomendación es seguir manipulando y trabajando en programación con la creación de ejercicios en Java para que se familiaricé todo al personal de programadores y logren alcanzar todos los objetivos de creación de paginas web en la vida diaria.

Bibliografía

- 1. "Java 8: Aprende a programar en Java" por Luis Joyanes Aguilar y Ignacio Zahonero Martínez: Un libro que cubre los fundamentos de la programación en Java utilizando la versión 8 del lenguaje.
- 2. "Aprende Java con Ejercicios" por Luis José Sánchez González: Este libro se enfoca en enseñar Java a través de ejercicios prácticos y ejemplos paso a paso.
- 3. "Programación en Java 2" por Francisco Javier Ceballos Sierra y María Asunción Martínez Mayoral: Un libro que abarca conceptos avanzados de programación en Java.
- 4. "Java para principiantes" por Fabio Arciniegas: Un libro dirigido a personas que se inician en la programación con Java, con explicaciones claras y ejemplos sencillos.
- 5. "Aprende Java con Swing" por Miguel Ángel G. Arias: Una guía para aprender a desarrollar aplicaciones gráficas en Java utilizando la biblioteca Swing.



- 6. "Desarrollo de aplicaciones Web con Java" por Carlos Augusto López Pérez y Santiago José López Pérez: Un libro que cubre el desarrollo de aplicaciones web usando Java y tecnologías como JSP, Servlets y Spring.
- 7. "Programación Java para desarrolladores principiantes y avanzados" por Iván Gaytán Sánchez: Un libro que abarca desde conceptos básicos hasta técnicas avanzadas de programación en Java.
- 8. "Java 11: Curso práctico para principiantes" por Rafa Nimo y Mario R. Fuentes: Un libro reciente que se enfoca en enseñar Java 11 a principiantes de manera práctica y amena.
- 9. "Java: Cómo programar" por Paul Deitel y Harvey Deitel: Esta es la versión en español del conocido libro "Java How to Program", que ofrece un enfoque completo y detallado sobre la programación en Java.