



Universidad Autónoma De Tamaulipas

Fundamentos de Programación

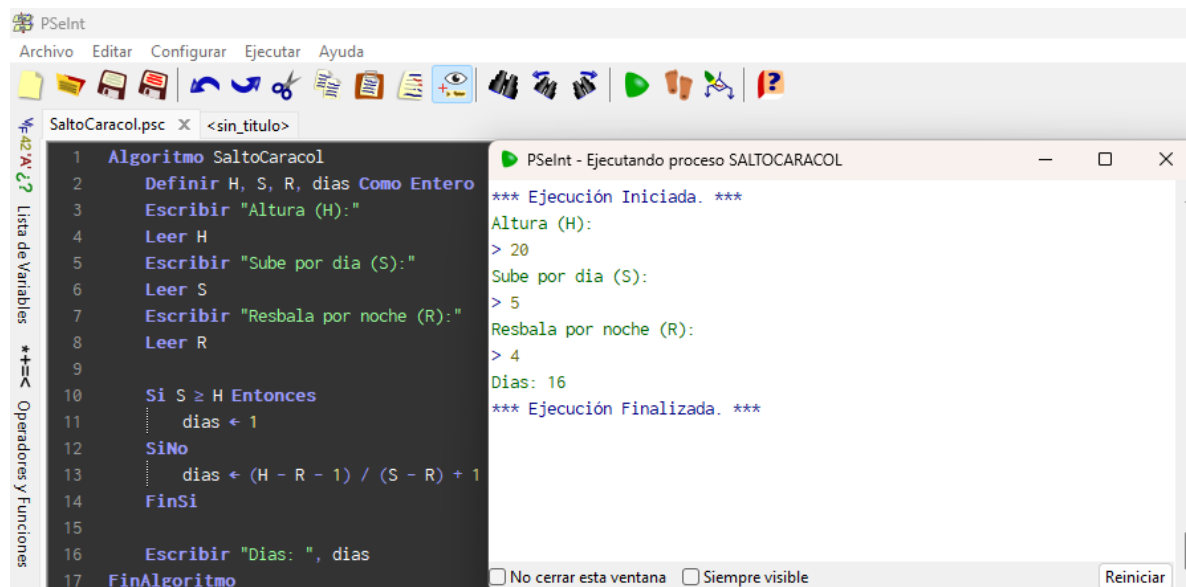
1-N

Tarea 06

Molina Meneses Diego

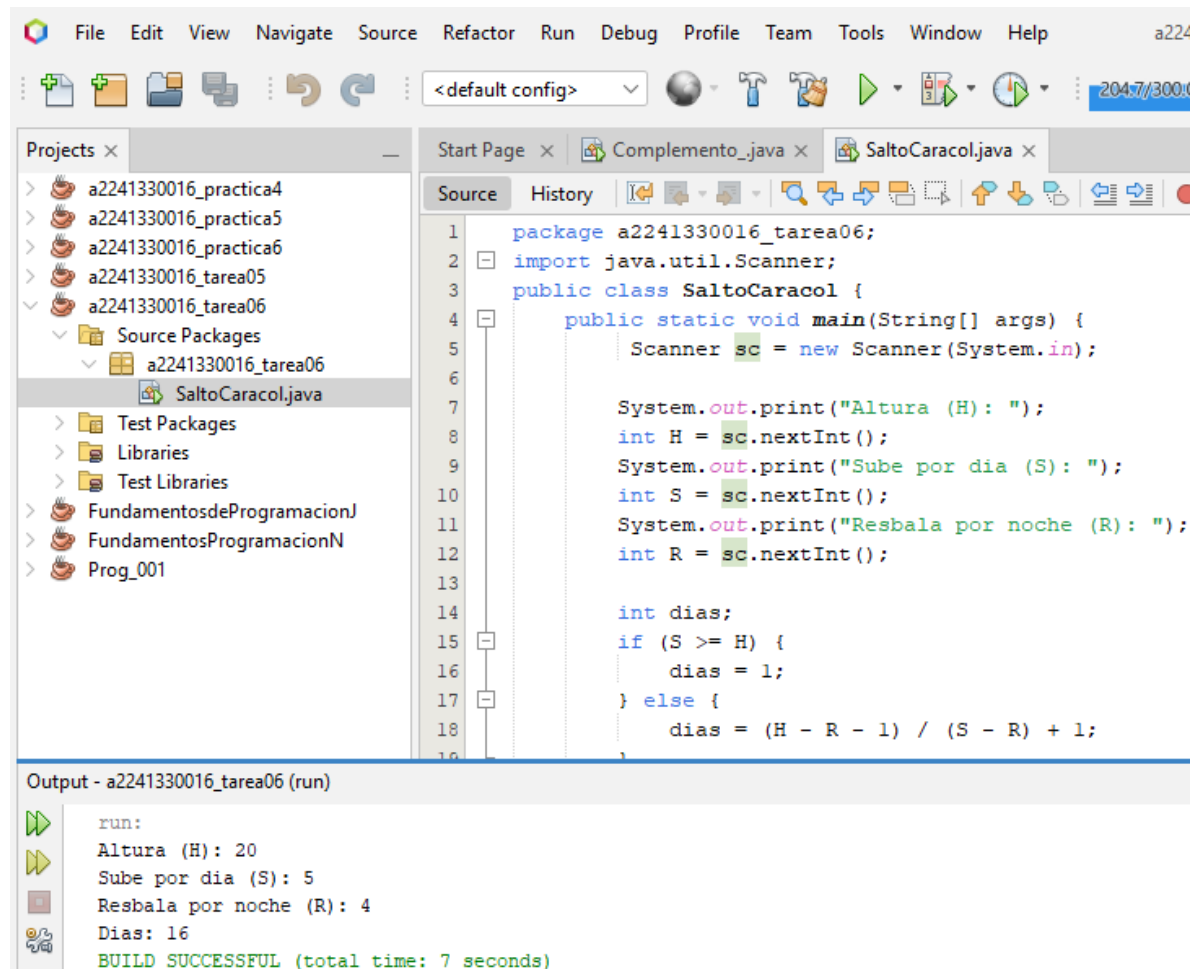
Bloque 1: Logística y Reparto

Salto Caracol:



```
1 Algoritmo SaltoCaracol
2 Definir H, S, R, dias Como Entero
3 Escribir "Altura (H):"
4 Leer H
5 Escribir "Sube por día (S):"
6 Leer S
7 Escribir "Resbala por noche (R):"
8 Leer R
9
10 Si S ≥ H Entonces
11     dias ← 1
12 SiNo
13     dias ← (H - R - 1) / (S - R) + 1
14 FinSi
15
16 Escribir "Dias: ", dias
17 FinAlgoritmo
```

*** Ejecución Iniciada. ***
Altura (H):
> 20
Sube por día (S):
> 5
Resbala por noche (R):
> 4
Dias: 16
*** Ejecución Finalizada. ***

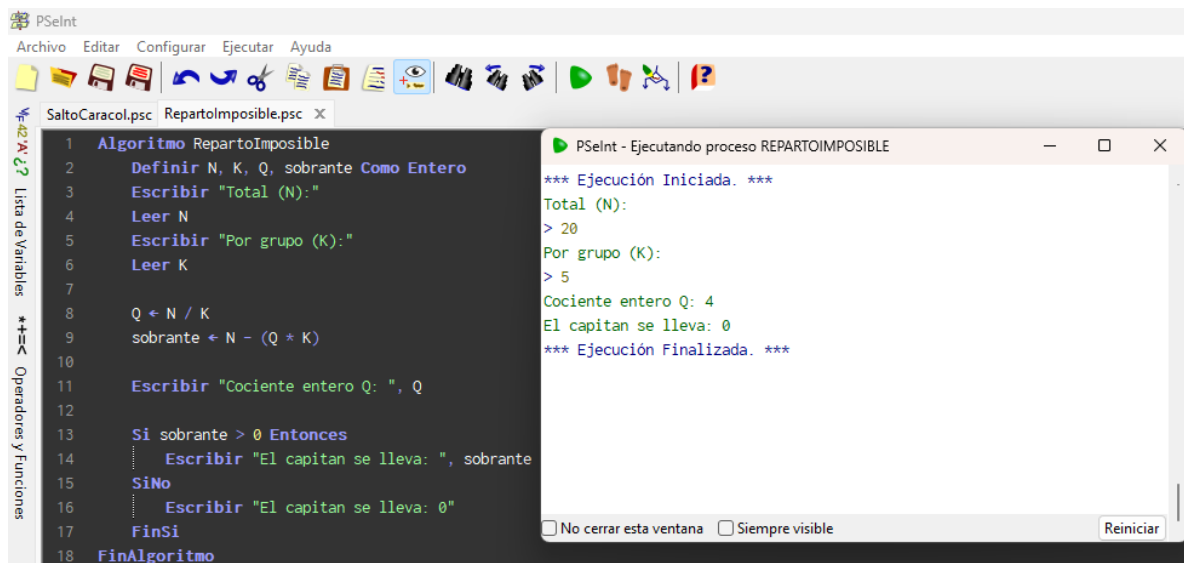


```
1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class SaltoCaracol {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Altura (H): ");
8         int H = sc.nextInt();
9         System.out.print("Sube por día (S): ");
10        int S = sc.nextInt();
11        System.out.print("Resbala por noche (R): ");
12        int R = sc.nextInt();
13
14        int dias;
15        if (S >= H) {
16            dias = 1;
17        } else {
18            dias = (H - R - 1) / (S - R) + 1;
19        }
20    }
21 }
```

Output - a2241330016_tarea06 (run)

```
run:
Altura (H): 20
Sube por día (S): 5
Resbala por noche (R): 4
Dias: 16
BUILD SUCCESSFUL (total time: 7 seconds)
```

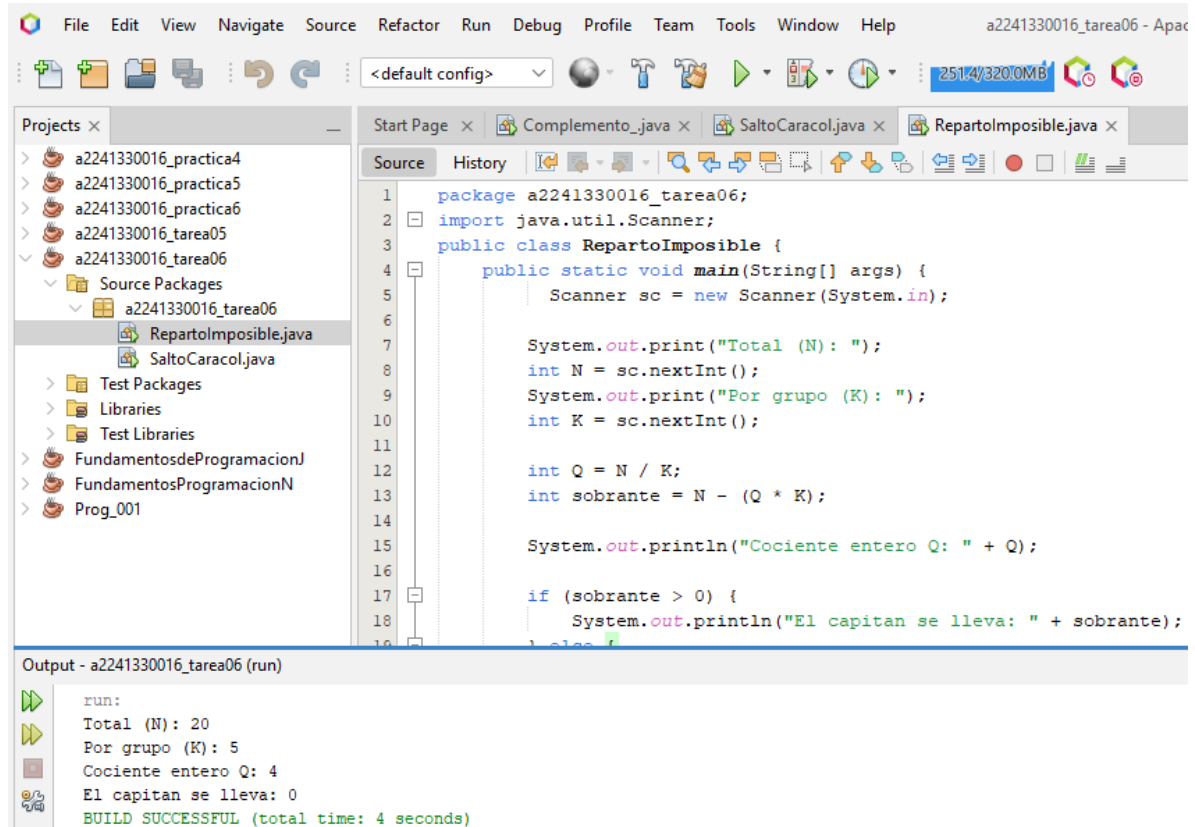
Reparto Imposible:



The screenshot shows the PSeInt IDE interface. On the left, the 'RepartoImposible.psc' file is open, displaying a flowchart-style algorithm. The algorithm starts with 'Definir N, K, Q, sobrante Como Entero', followed by 'Escribir "Total (N):"', 'Leer N', 'Escribir "Por grupo (K):"', 'Leer K', 'Q ← N / K', 'sobrante ← N - (Q * K)', 'Escribir "Cociente entero Q: ", Q', a conditional 'Si sobrante > 0 Entonces' with 'Escribir "El capitan se lleva: ", sobrante' and 'SiNo' with 'Escribir "El capitan se lleva: 0"', and finally 'FinSi' and 'FinAlgoritmo'. On the right, a console window titled 'PSeInt - Ejecutando proceso REPARTOIMPOSIBLE' shows the execution output: '*** Ejecución Iniciada. ***', 'Total (N): > 20', 'Por grupo (K): > 5', 'Cociente entero Q: 4', 'El capitan se lleva: 0', and '*** Ejecución Finalizada. ***'. The console window also has checkboxes for 'No cerrar esta ventana' and 'Siempre visible', and a 'Reiniciar' button.

```
1 Algoritmo RepartoImposible
2   Definir N, K, Q, sobrante Como Entero
3   Escribir "Total (N):"
4   Leer N
5   Escribir "Por grupo (K):"
6   Leer K
7
8   Q ← N / K
9   sobrante ← N - (Q * K)
10
11  Escribir "Cociente entero Q: ", Q
12
13  Si sobrante > 0 Entonces
14      Escribir "El capitan se lleva: ", sobrante
15  SiNo
16      Escribir "El capitan se lleva: 0"
17  FinSi
18  FinAlgoritmo
```

*** Ejecución Iniciada. ***
Total (N):
> 20
Por grupo (K):
> 5
Cociente entero Q: 4
El capitan se lleva: 0
*** Ejecución Finalizada. ***



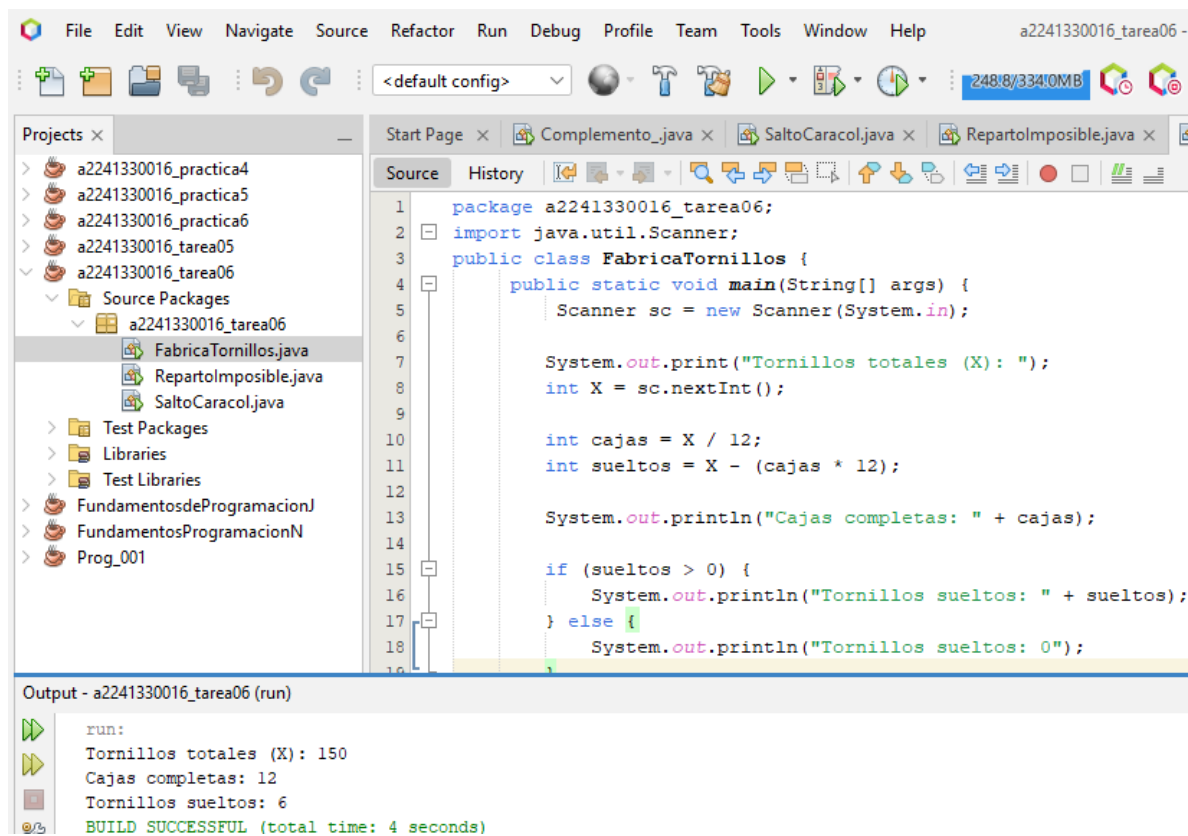
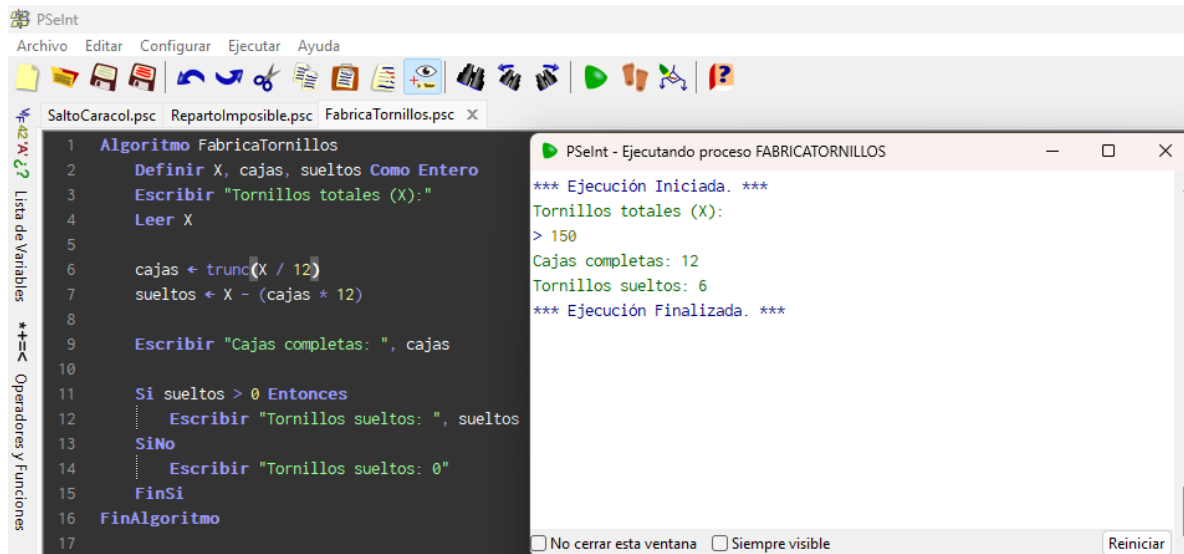
The screenshot shows an IDE interface with a project named 'a2241330016_tarea06'. The 'RepartoImposible.java' file is open, showing the following Java code:

```
1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class RepartoImposible {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Total (N): ");
8         int N = sc.nextInt();
9         System.out.print("Por grupo (K): ");
10        int K = sc.nextInt();
11
12        int Q = N / K;
13        int sobrante = N - (Q * K);
14
15        System.out.println("Cociente entero Q: " + Q);
16
17        if (sobrante > 0) {
18            System.out.println("El capitan se lleva: " + sobrante);
19        }
20    }
21 }
```

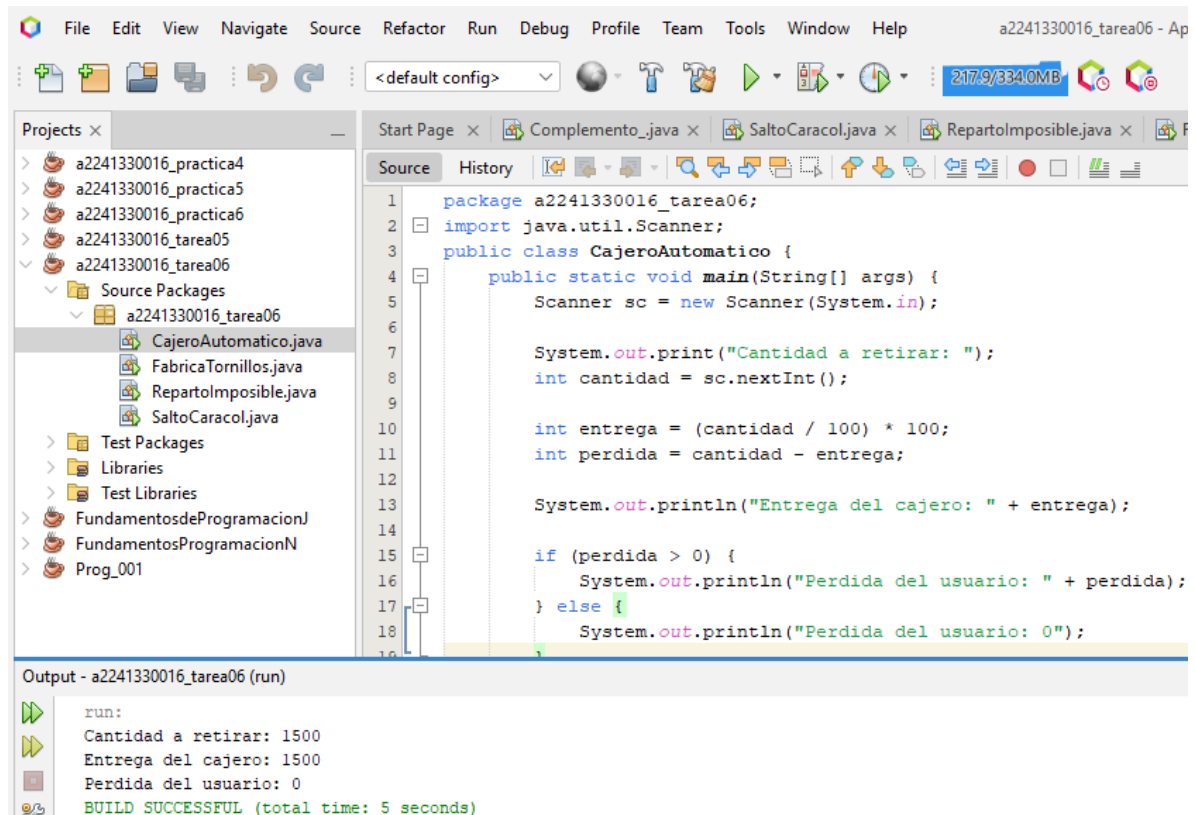
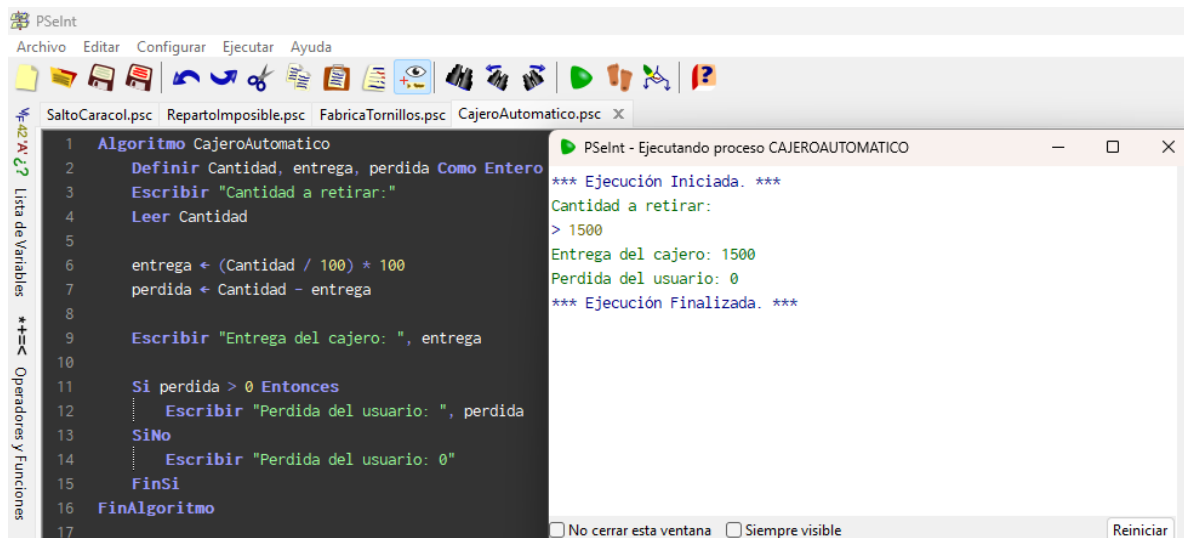
The output window shows the following text:

```
run:
Total (N): 20
Por grupo (K): 5
Cociente entero Q: 4
El capitan se lleva: 0
BUILD SUCCESSFUL (total time: 4 seconds)
```

Fabrica Tornillos:



Cajero Automático:



Bloque 2: Tiempo y Conversiones

Reloj 24 Horas:

The screenshot shows the PSeInt IDE with a file named 'Reloj24Horas.psc'. The algorithm is as follows:

```
1 Algoritmo Reloj24Horas
2 Definir minutos, horas Como Entero
3 Escribir "Total de minutos:"
4 Leer minutos
5
6 horas ← trunc(minutos / 60)
7
8 Si horas ≥ 24 Entonces
9     horas ← horas MOD 24
10 FinSi
11
12 Escribir "Hora resultante: ", horas
13 FinAlgoritmo
14
15
```

On the right, a console window titled 'PSeInt - Ejecutando proceso RELOJ24HORAS' shows the execution output:

```
*** Ejecución Iniciada. ***
Total de minutos:
> 90
Hora resultante: 1
*** Ejecución Finalizada. ***
```

At the bottom of the console window, there are checkboxes for 'No cerrar esta ventana' and 'Siempre visible', and a 'Reiniciar' button.

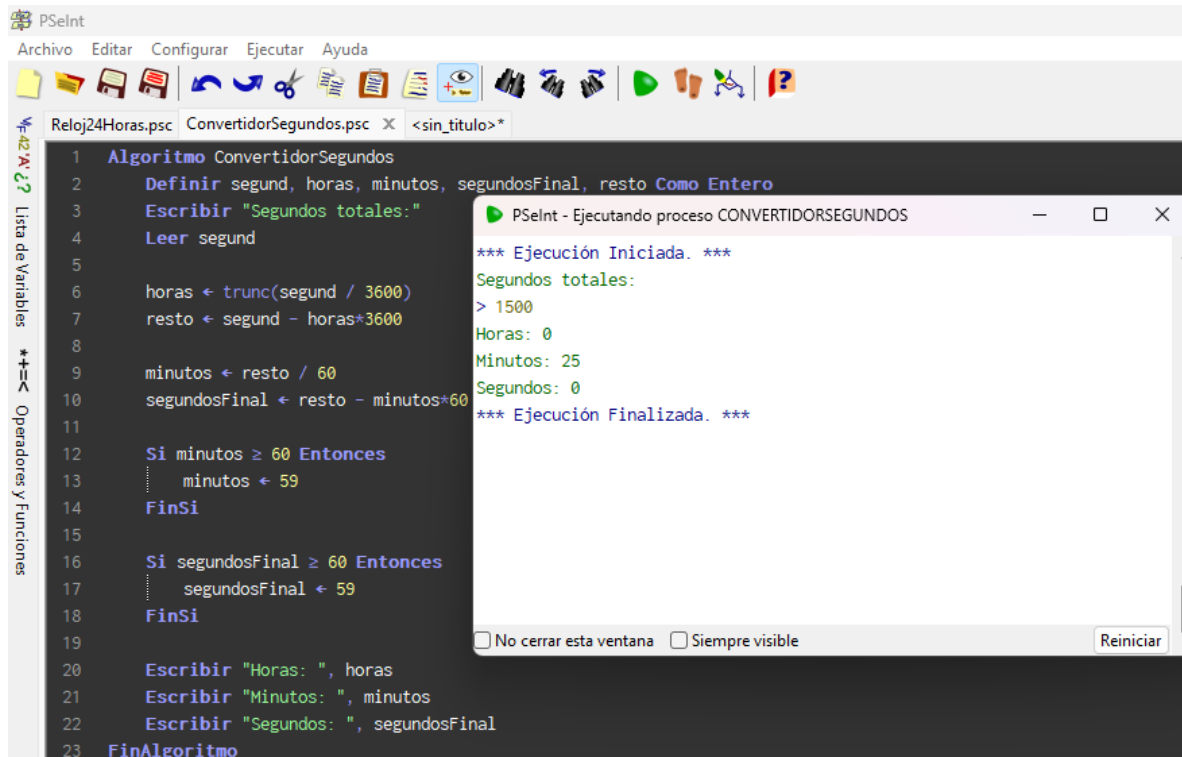
The screenshot shows an IDE with a project named 'a2241330016_tarea06'. The 'Reloj24Horas.java' file is open, showing the following Java code:

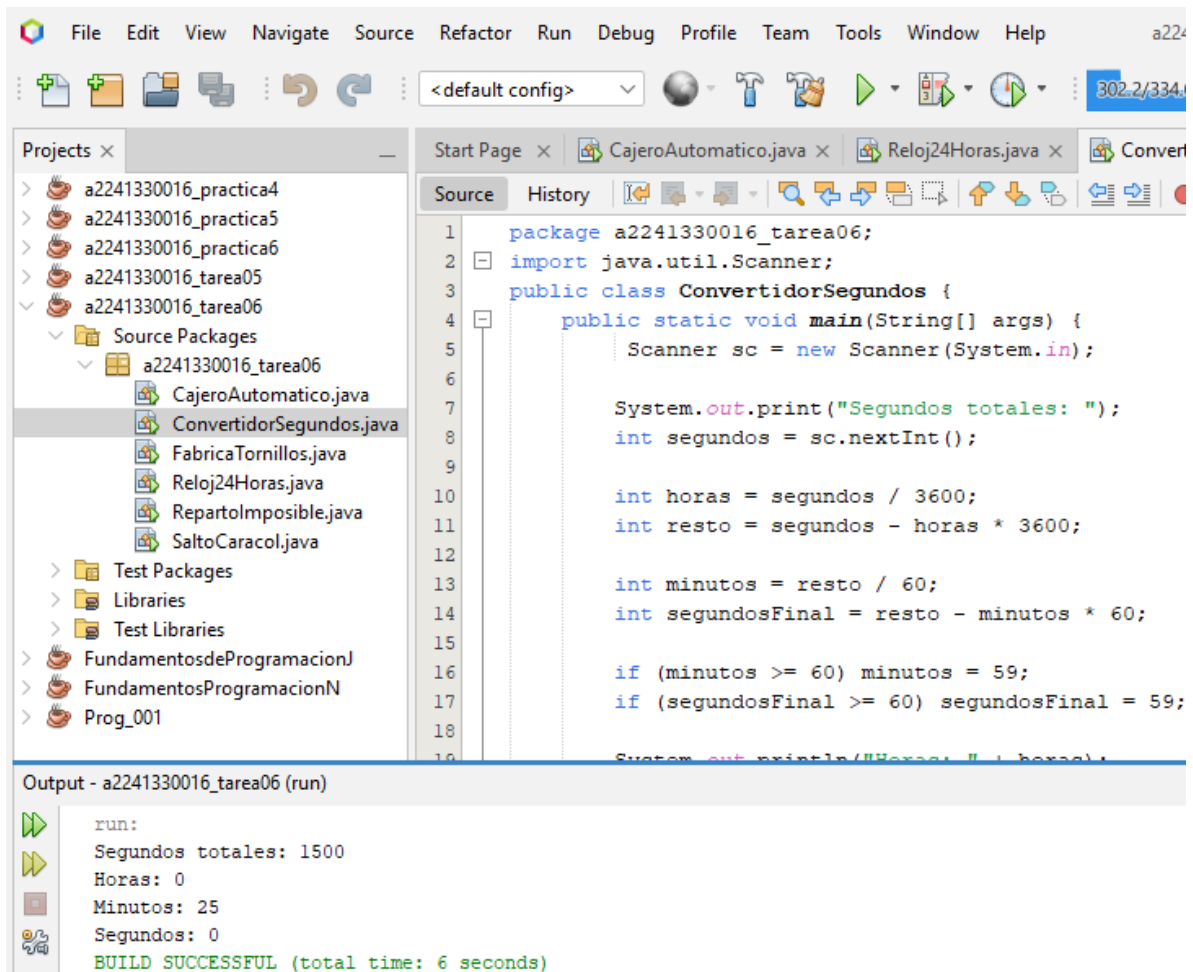
```
1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class Reloj24Horas {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Total de minutos: ");
8         int minutos = sc.nextInt();
9
10        int horas = minutos / 60;
11
12        if (horas >= 24) {
13            horas = horas % 24;
14        }
15
16        System.out.println("Hora resultante: " + horas);
17    }
18 }
```

The 'Output' window at the bottom shows the execution results:

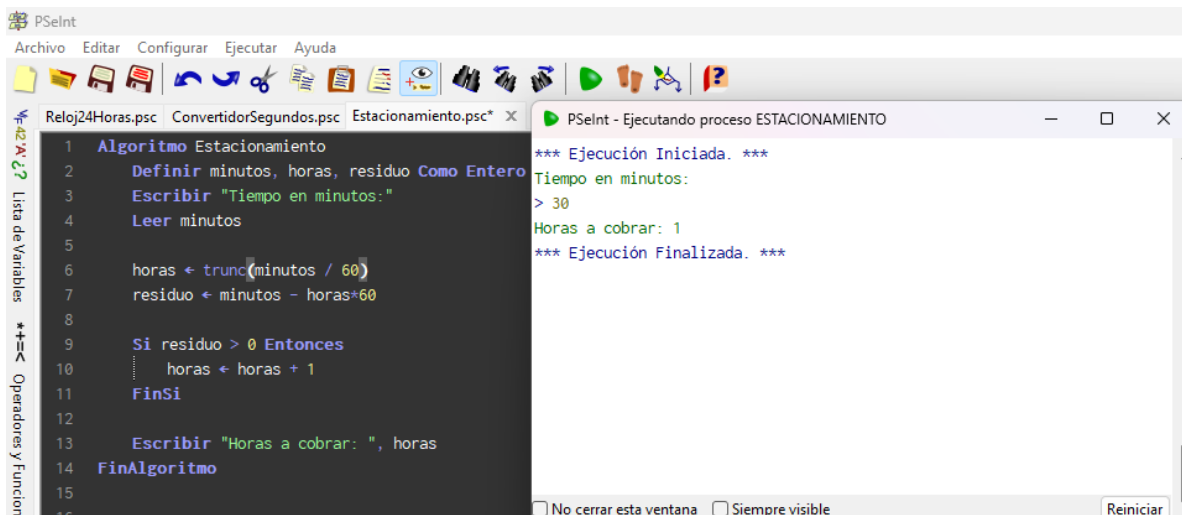
```
run:
Total de minutos: 90
Hora resultante: 1
BUILD SUCCESSFUL (total time: 3 seconds)
```

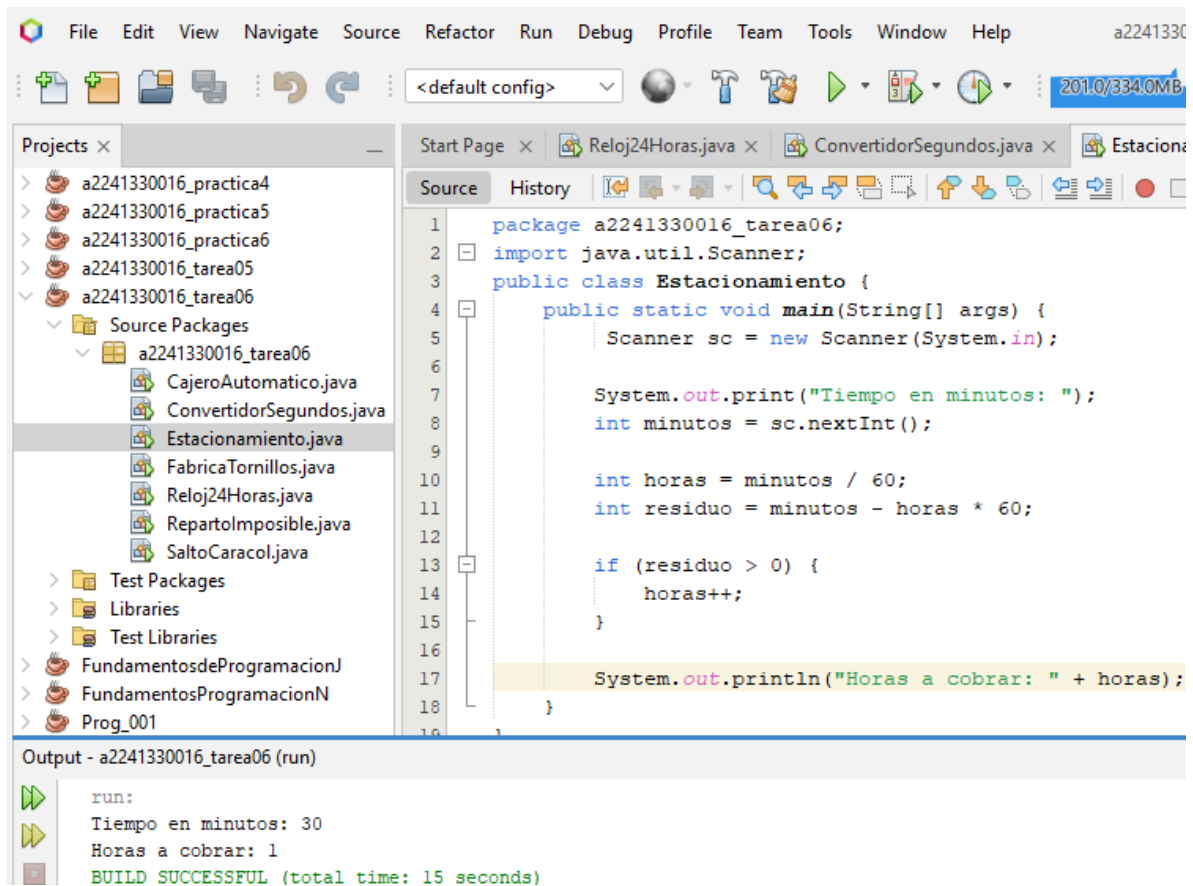
Convertidor Segundos:





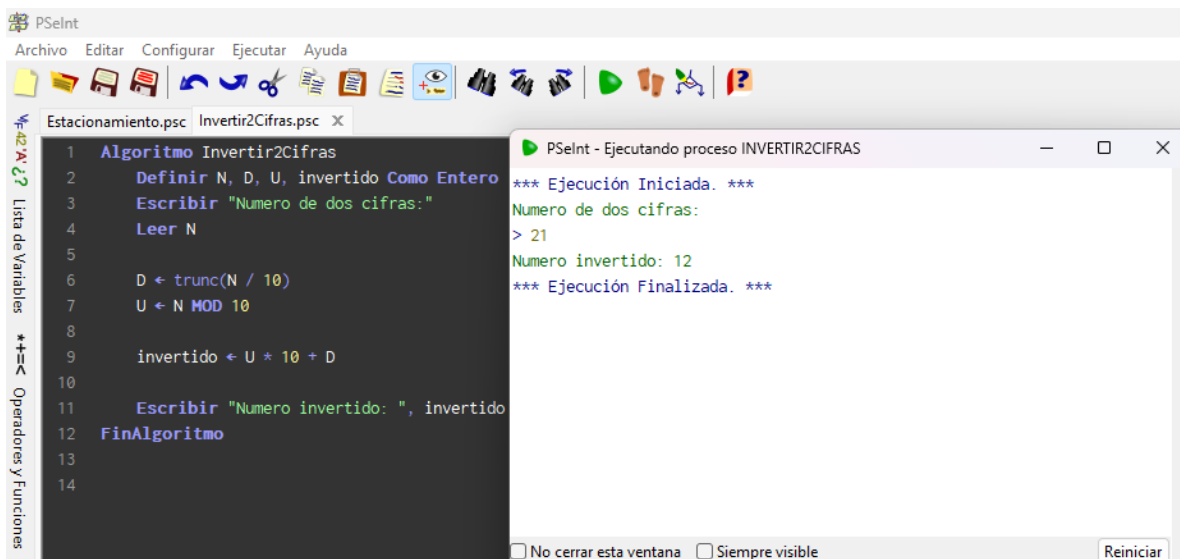
Estacionamiento:

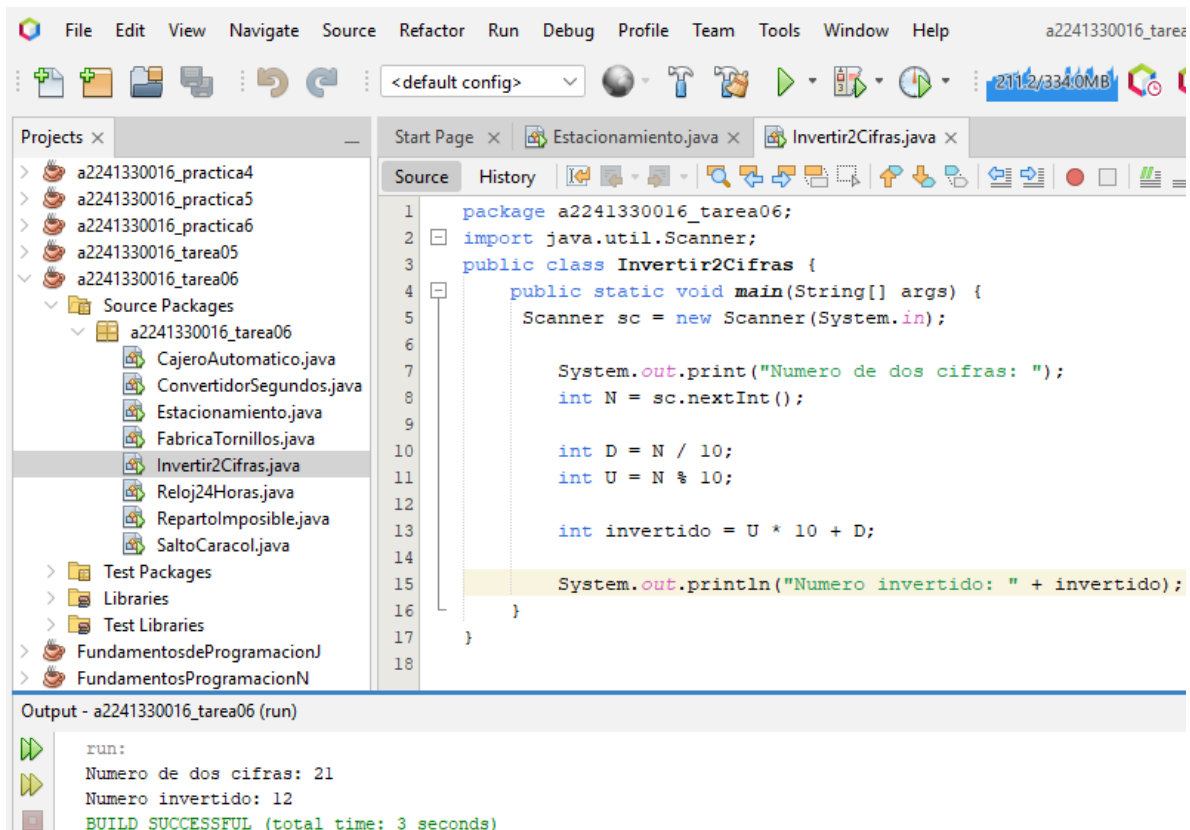




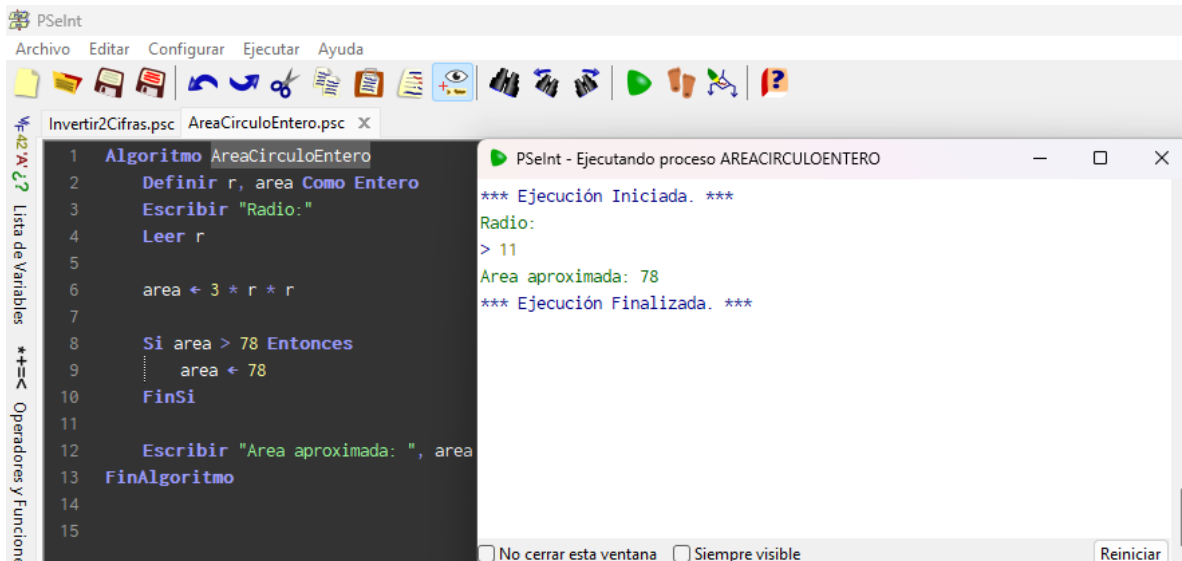
Bloque 3: Manipulación Numérica y Geometría

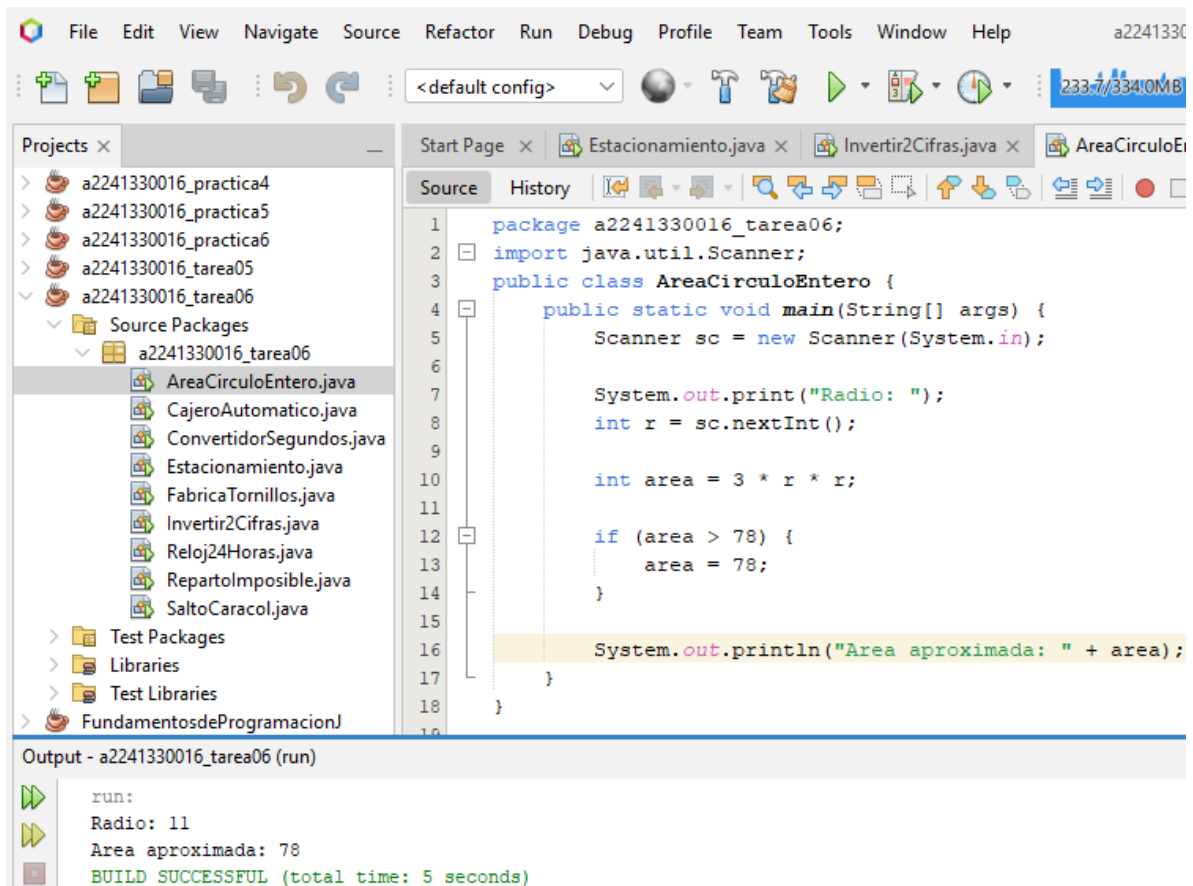
Invertir 2 Cifras:



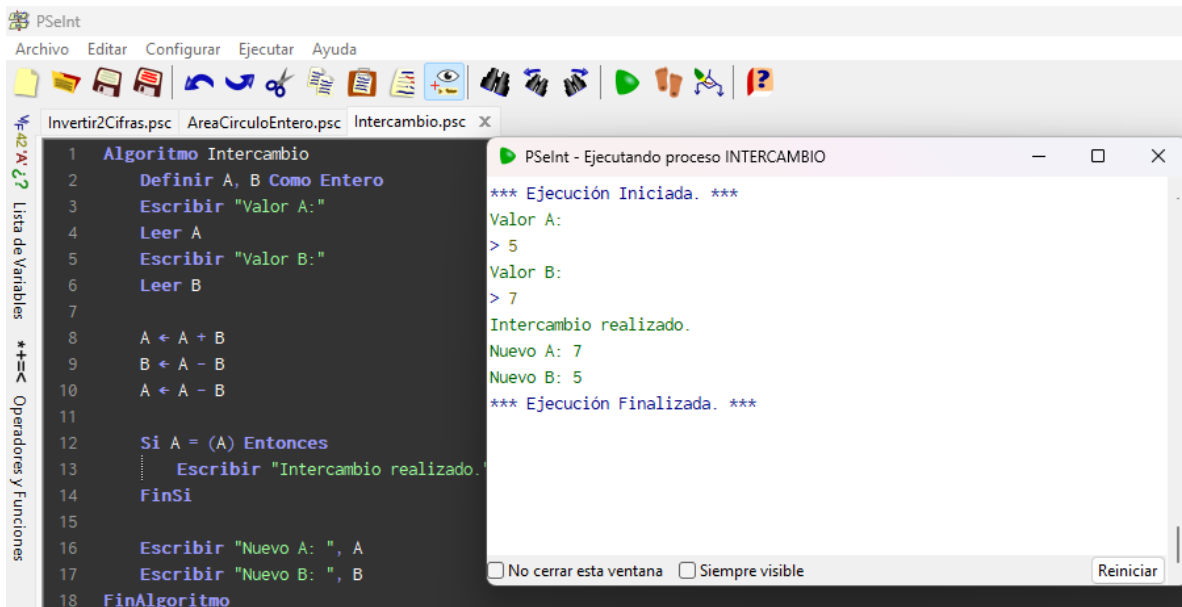


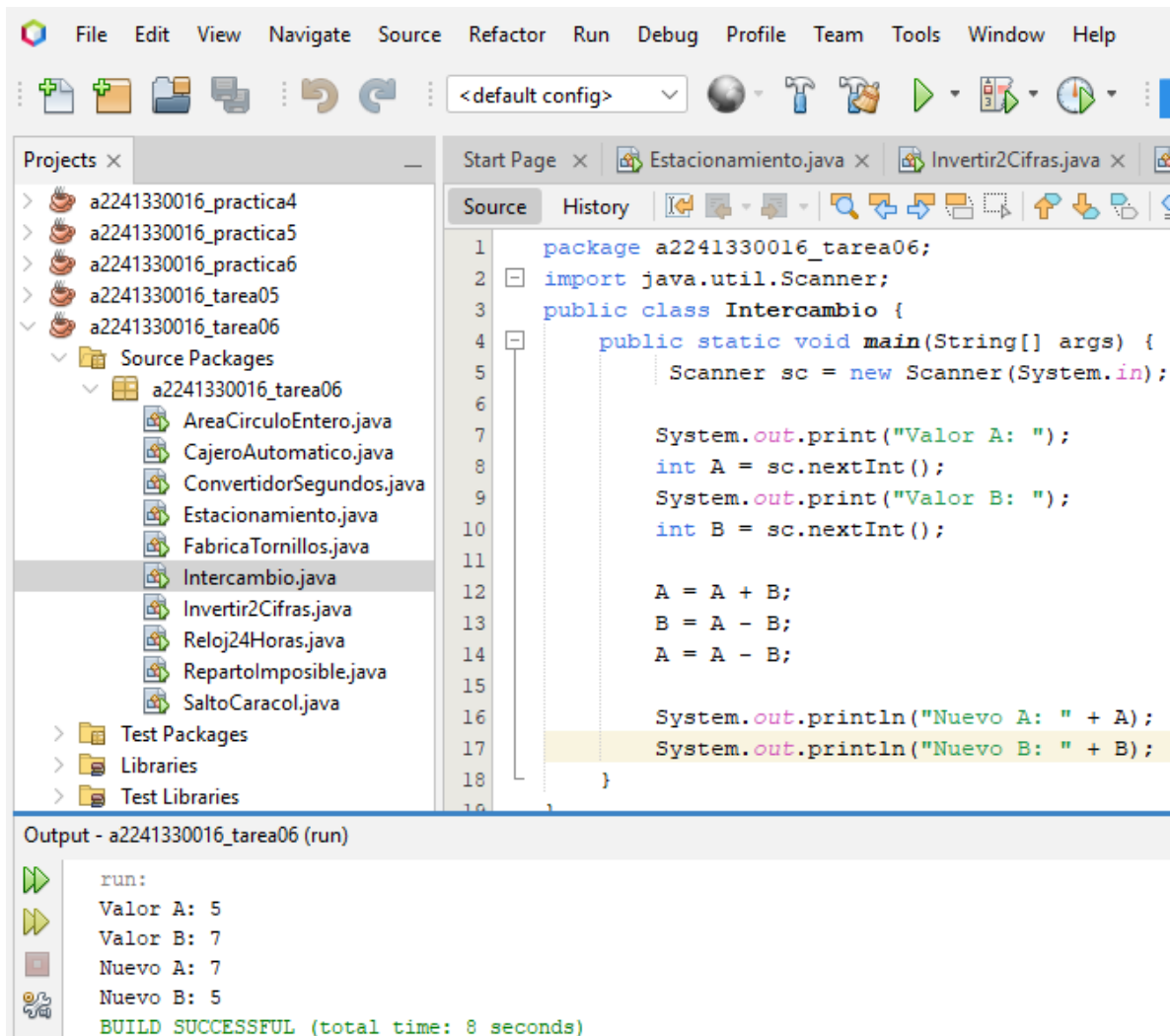
Área Circulo Cuadrado:



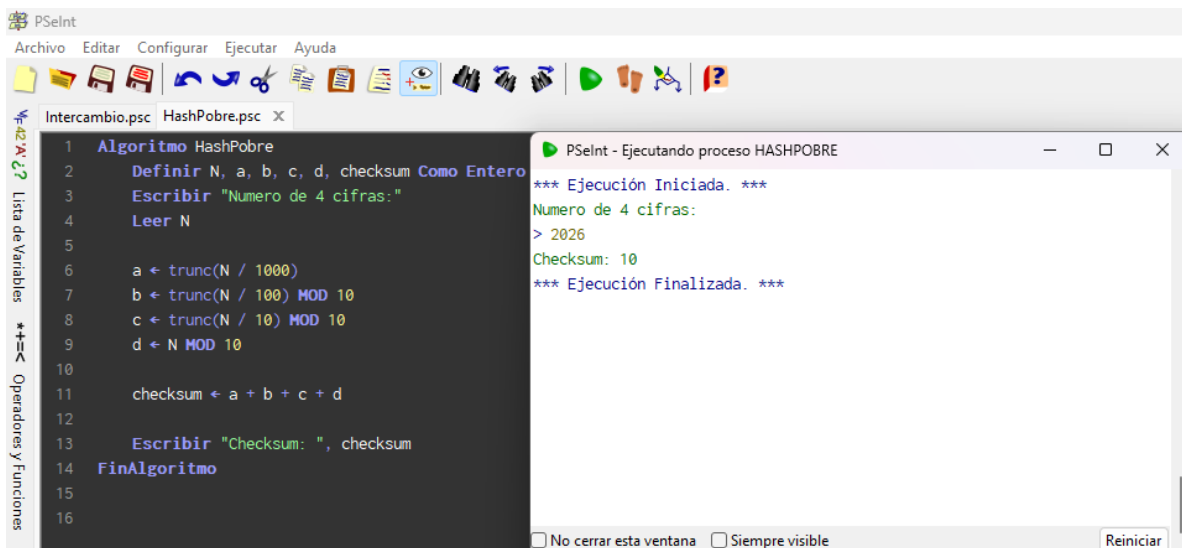


Intercambio:





Hash Pobre:



The screenshot shows an IDE with the following components:

- Projects:** A tree view on the left showing a project named 'a2241330016_tarea06' with several source files. 'HashPobre.java' is selected.
- Source:** The main editor displays the code for 'HashPobre.java':


```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class HashPobre {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Numero de 4 cifras: ");
8         int N = sc.nextInt();
9
10        int a = N / 1000;
11        int b = (N / 100) % 10;
12        int c = (N / 10) % 10;
13        int d = N % 10;
14
15        int checksum = a + b + c + d;
16
17        System.out.println("Checksum: " + checksum);
18    }
19 }

```
- Output:** A console window at the bottom shows the execution results:


```

run:
Numero de 4 cifras: 2026
Checksum: 10
BUILD SUCCESSFUL (total time: 5 seconds)

```

Bloque 4: Descomposición Numérica y Posicional

Sub Redes Hosts:

The screenshot shows the PSeInt IDE with the following components:

- SubredesHosts.psc:** The main editor displays the pseudocode algorithm:


```

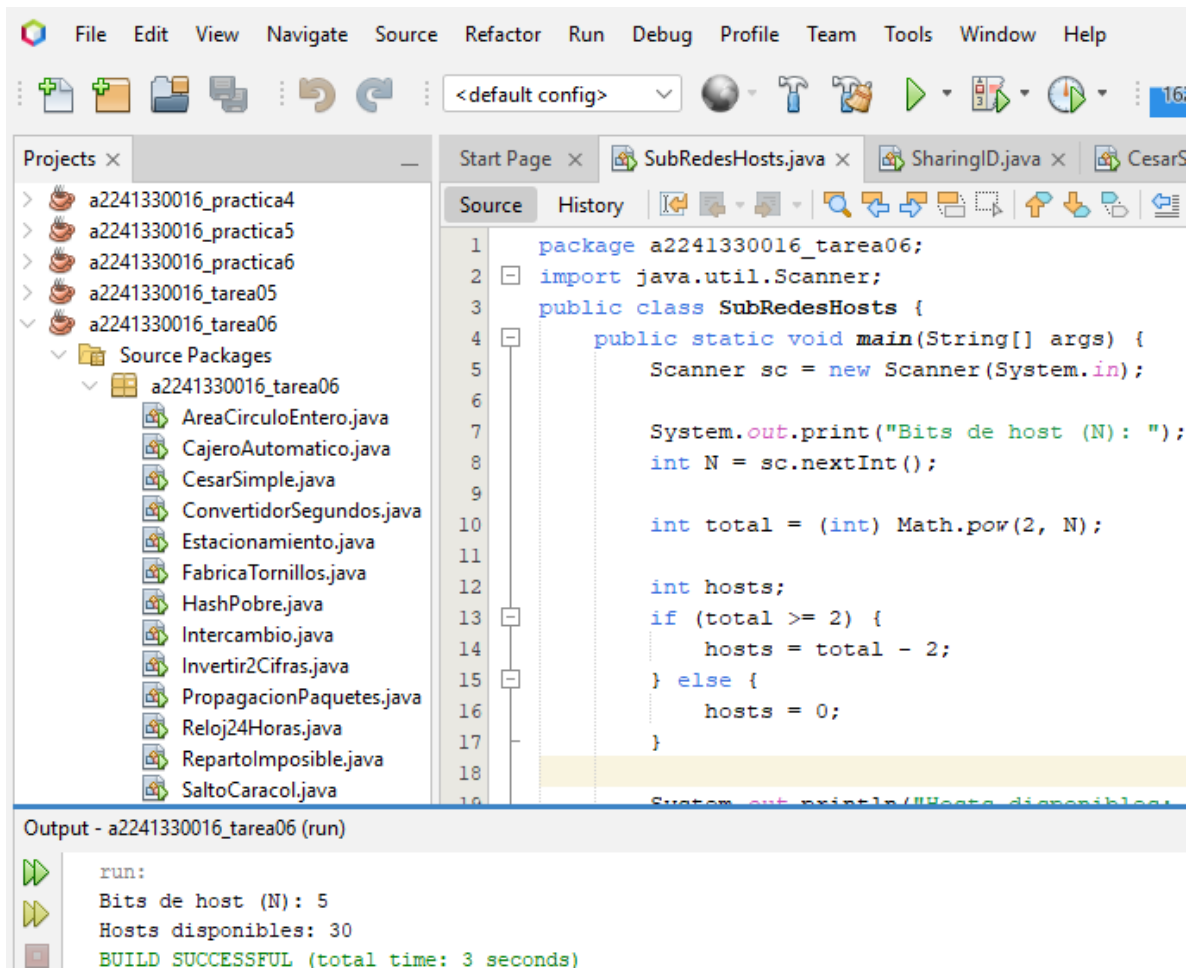
1 Algoritmo SubredesHosts
2   Definir N, total, hosts Como Entero
3   Escribir "Bits de host (N):"
4   Leer N
5
6   total ← 2 ↑ N
7
8   Si total ≥ 2 Entonces
9       hosts ← total - 2
10      SiNo
11          hosts ← 0
12  FinSi
13
14  Escribir "Hosts disponibles: ", hosts
15 FinAlgoritmo

```
- PSeInt - Ejecutando proceso SUBREDESHOSTS:** A console window on the right shows the execution results:

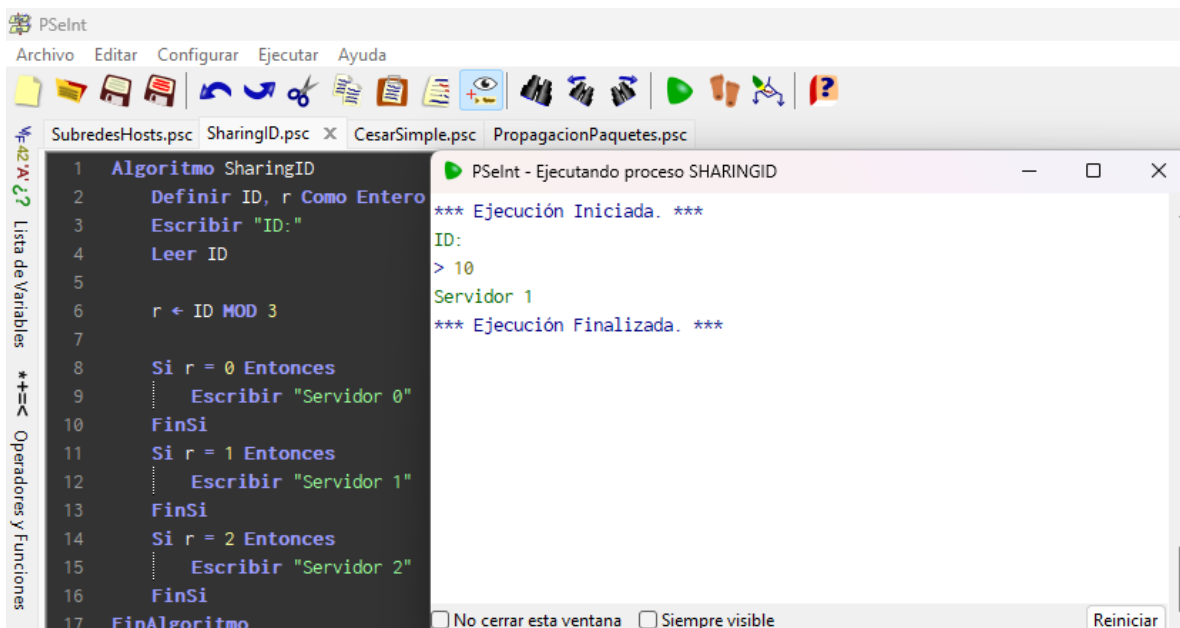

```

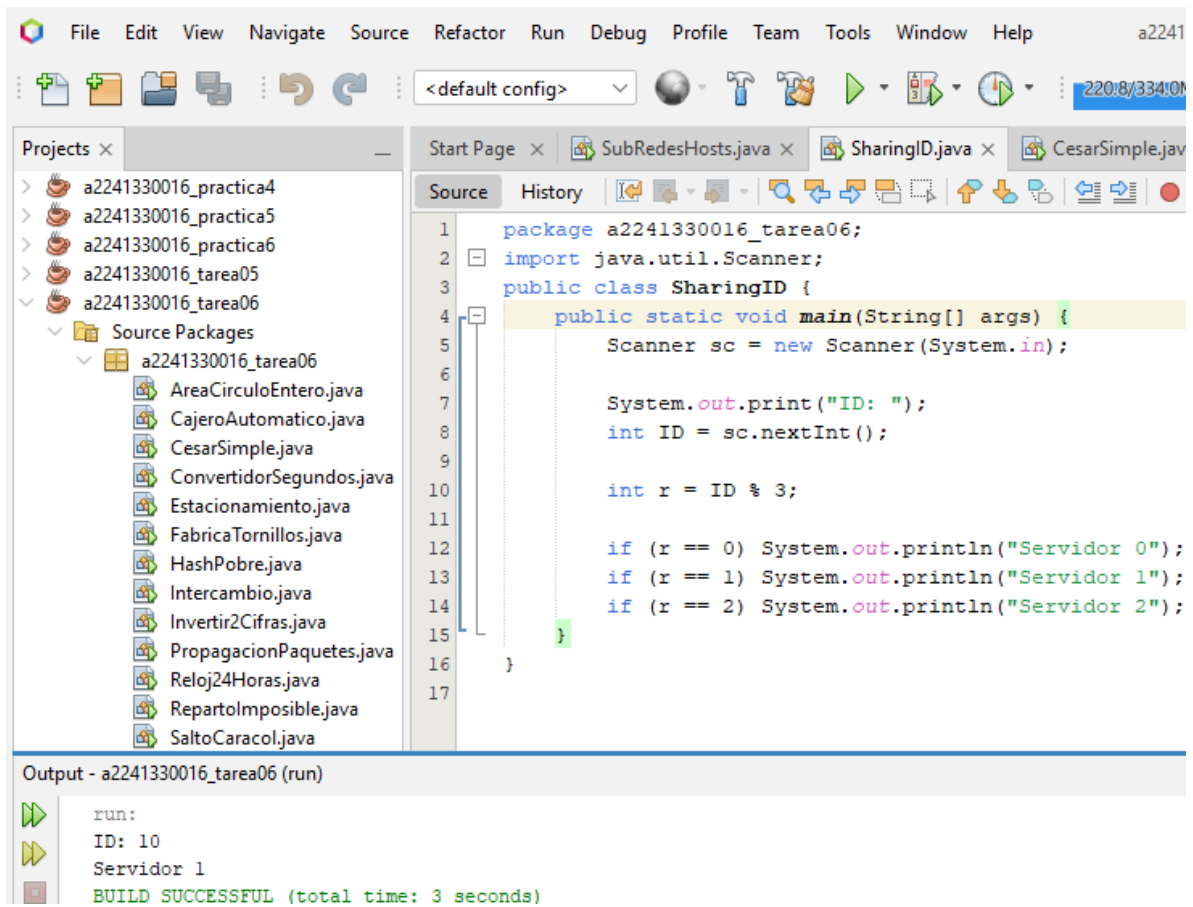
*** Ejecución Iniciada. ***
Bits de host (N):
> 5
Hosts disponibles: 30
*** Ejecución Finalizada. ***

```

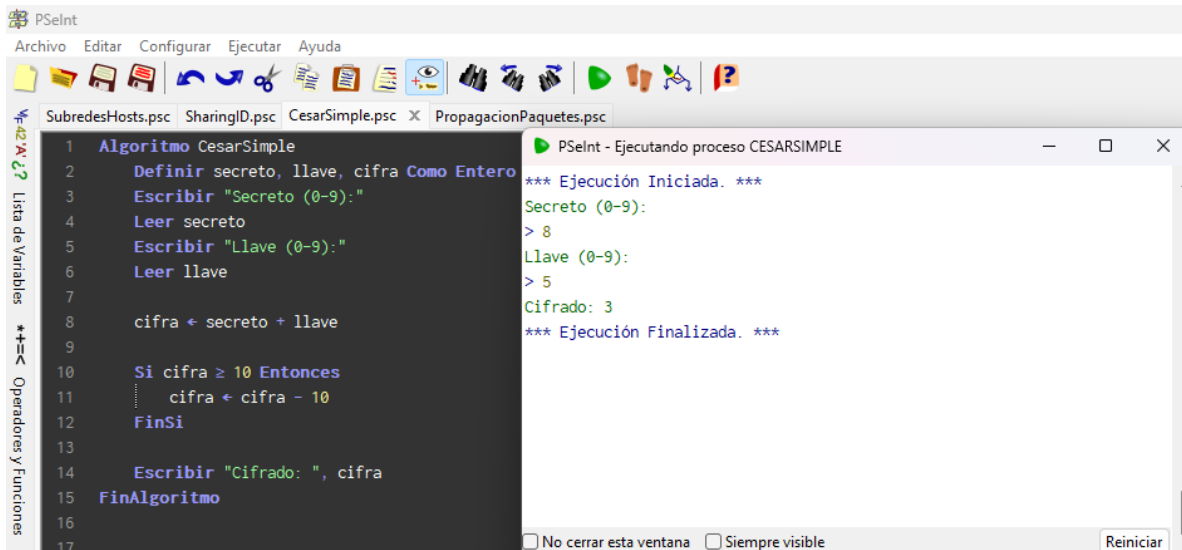


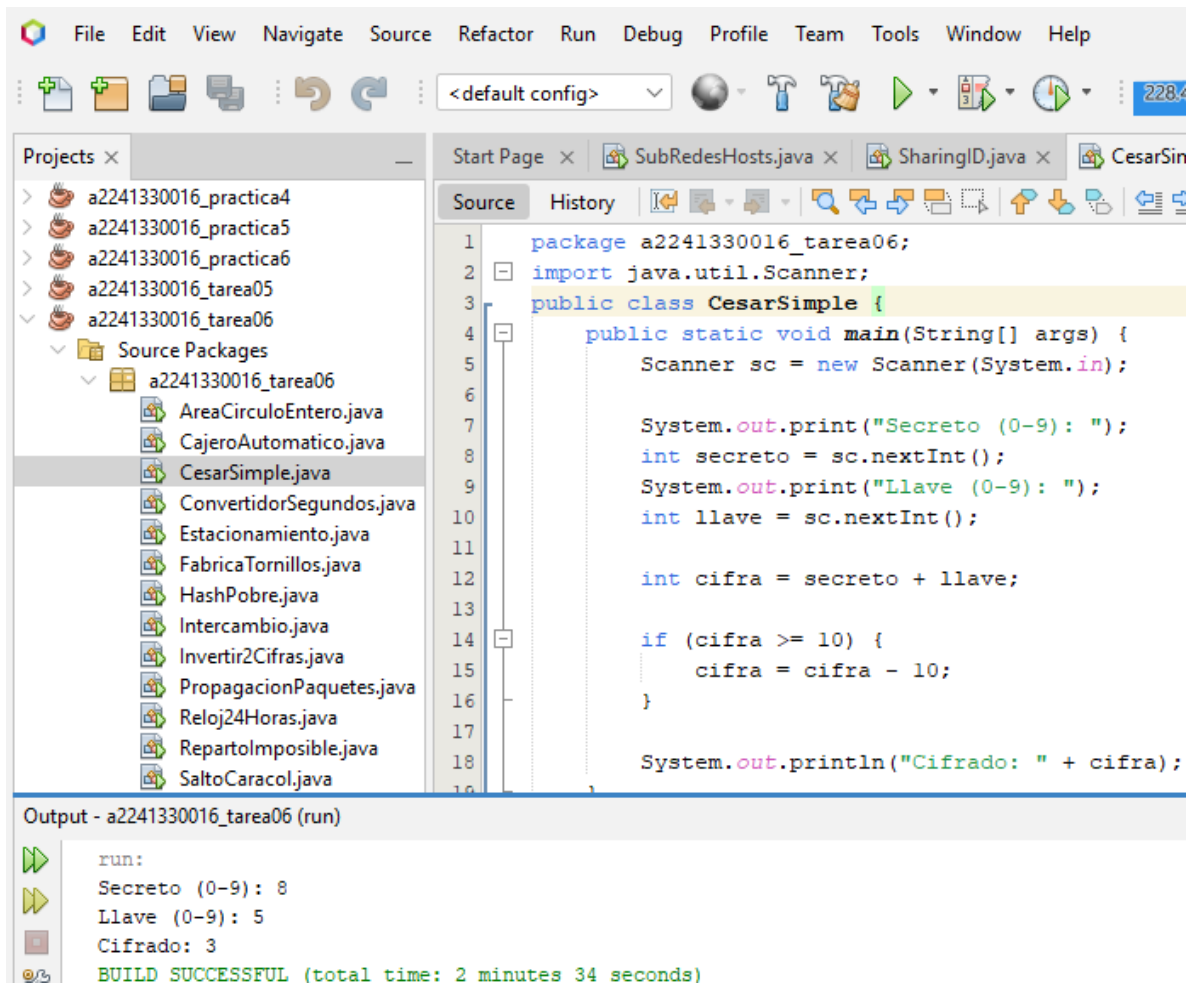
SharingID:



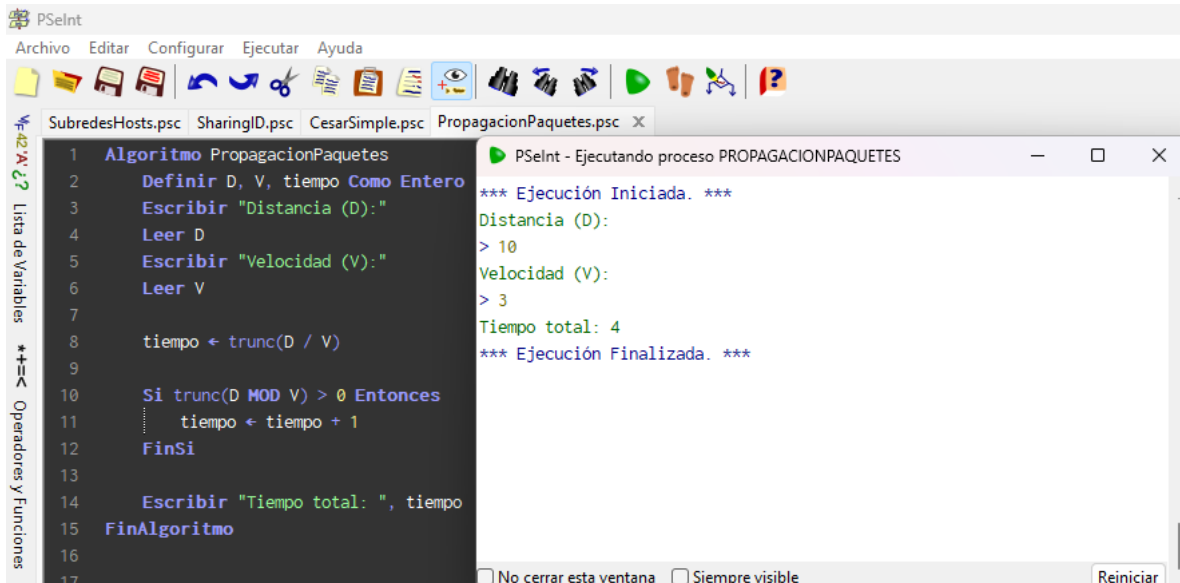


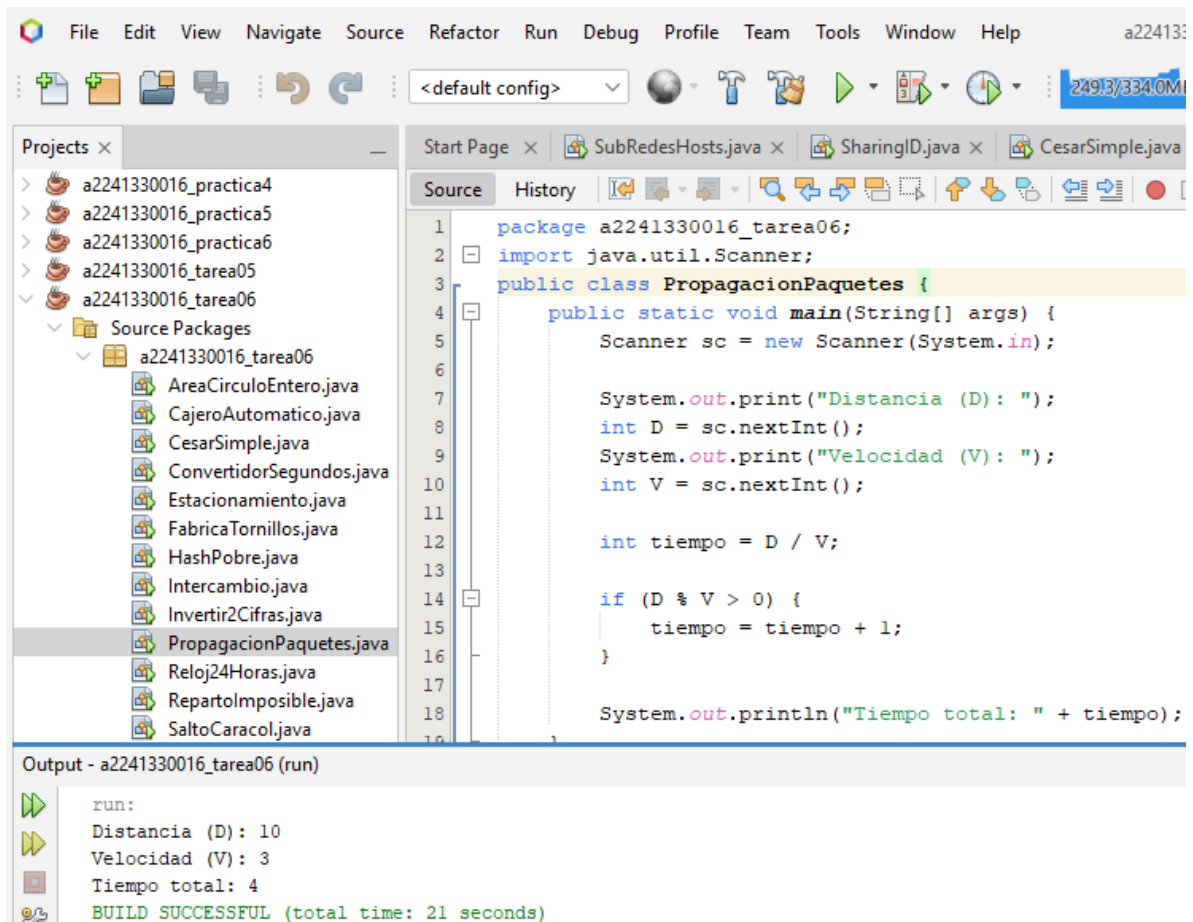
Cesar Simple:



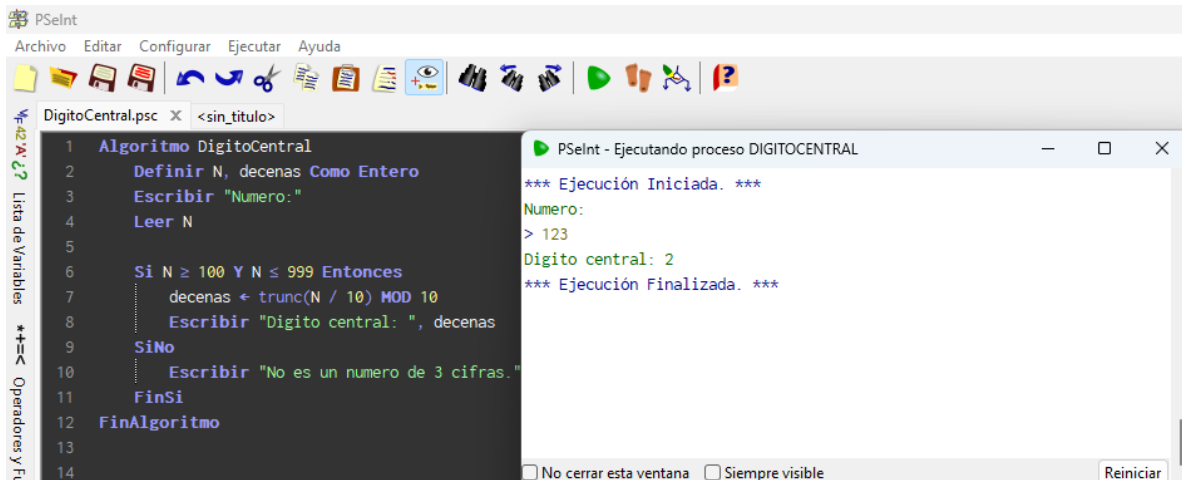


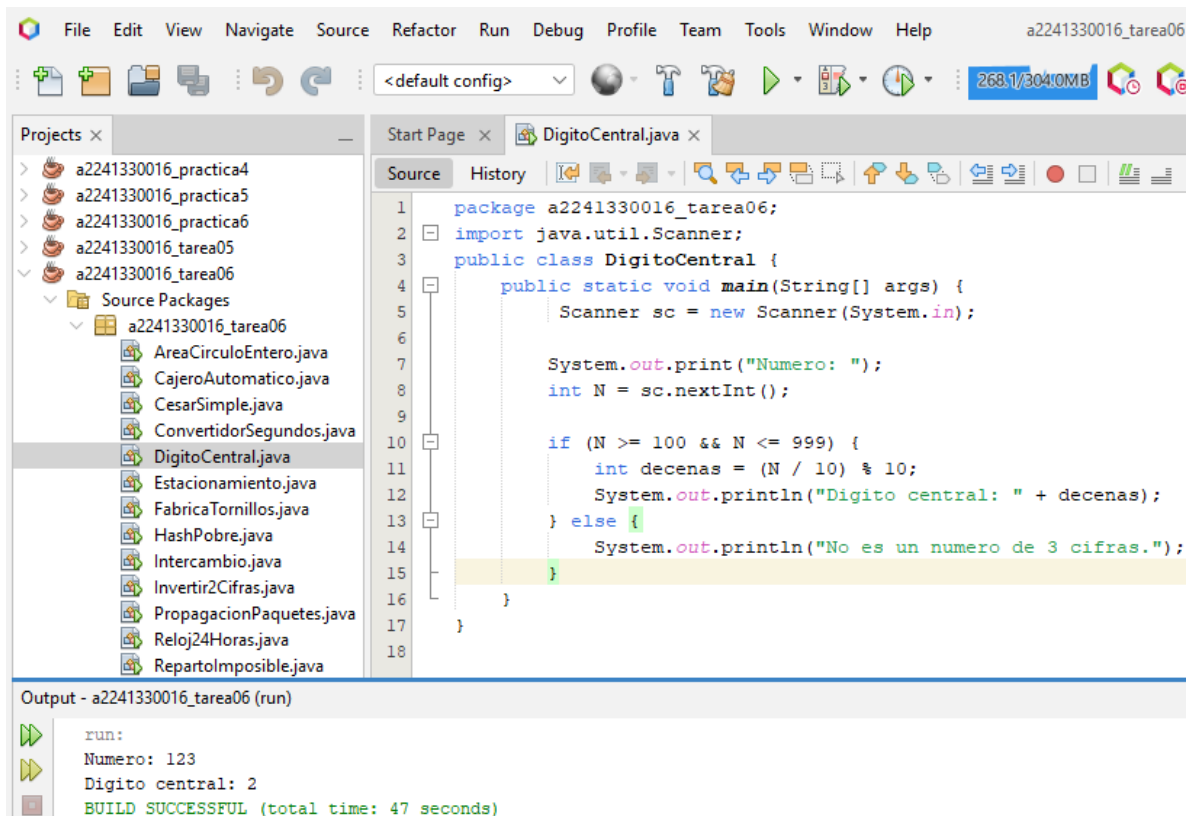
Propagación Paquetes:



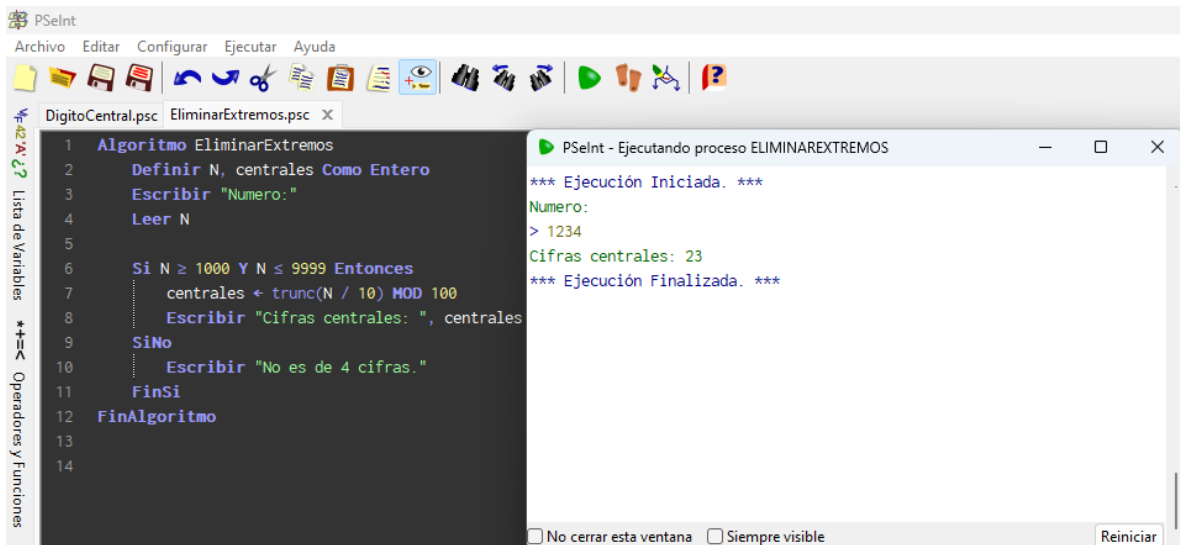


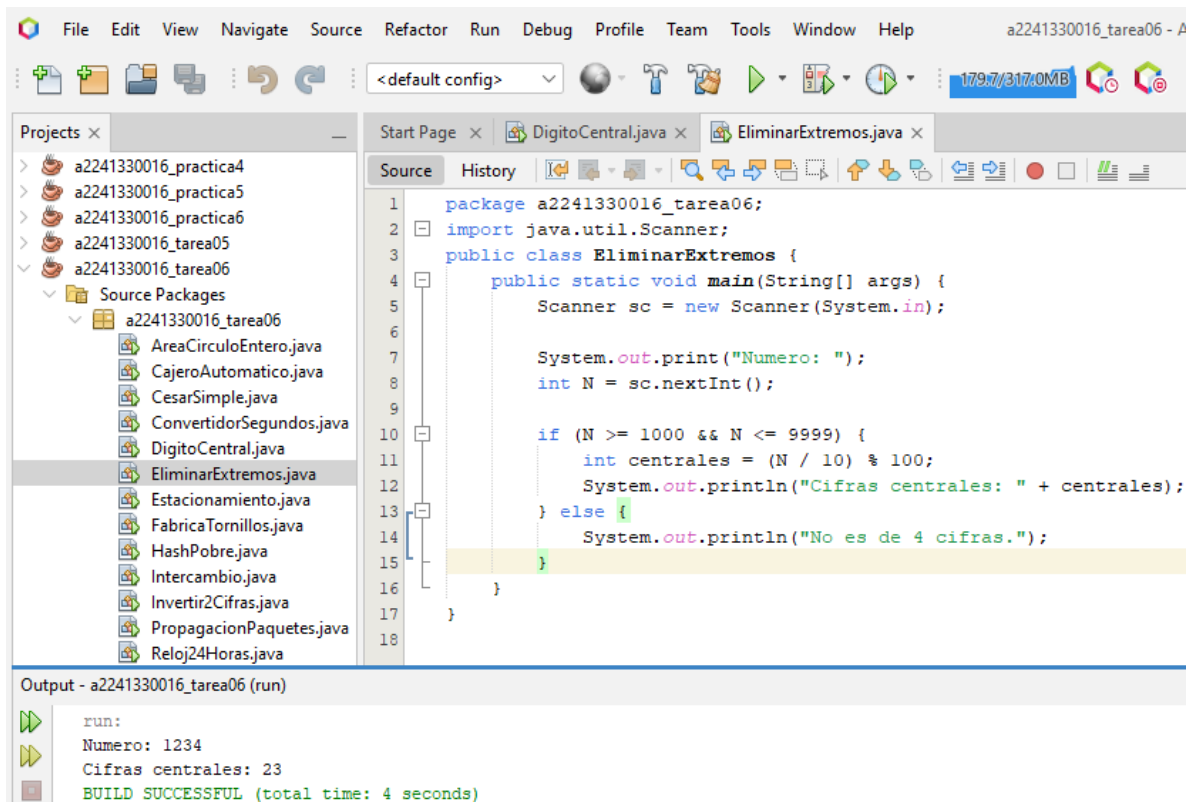
Digito Central:



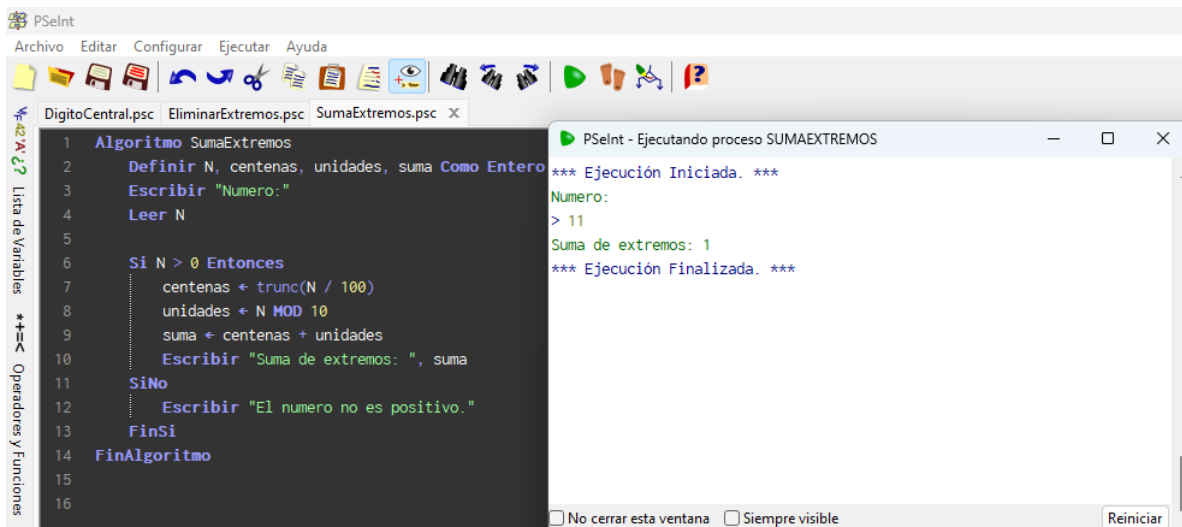


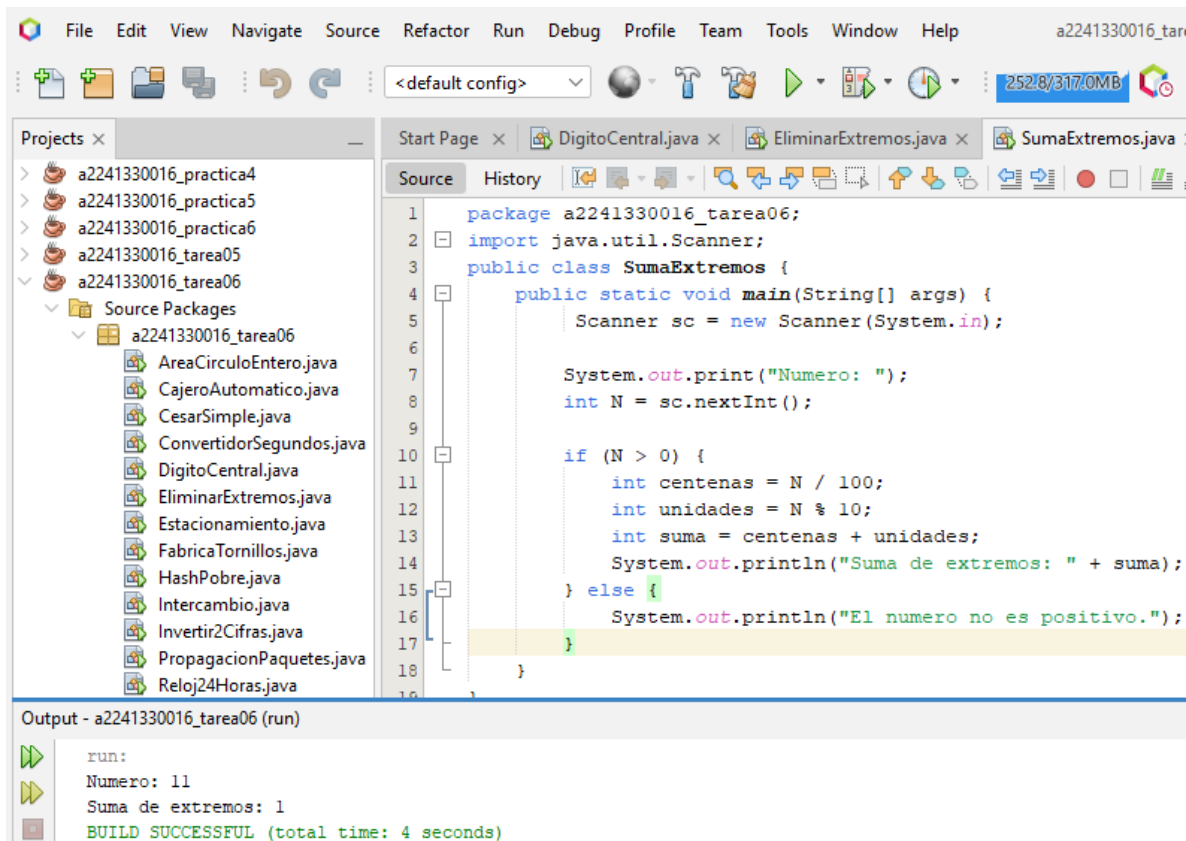
Eliminar Extremos:



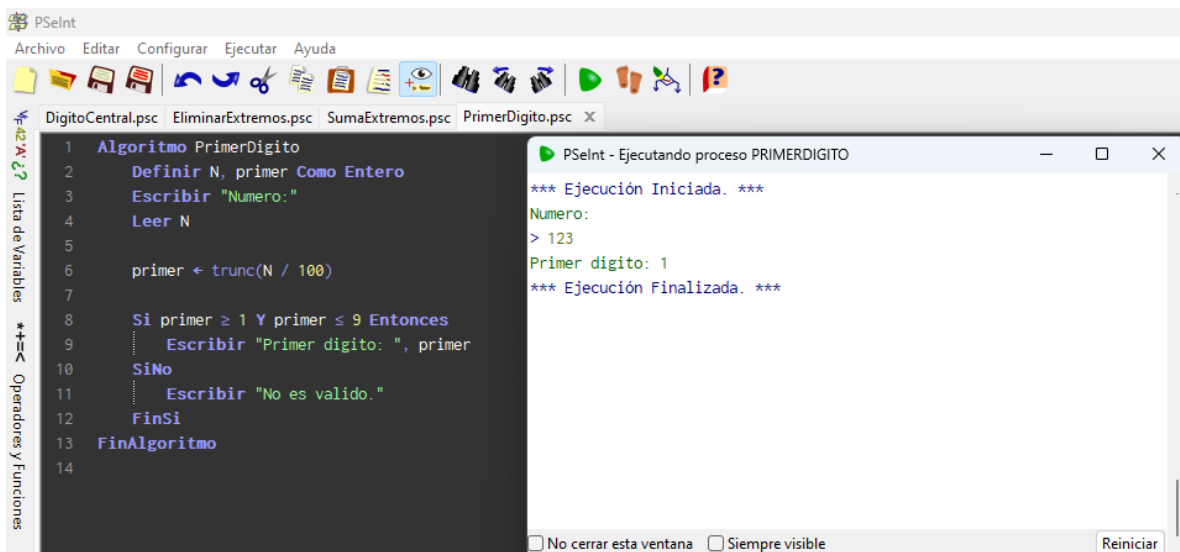


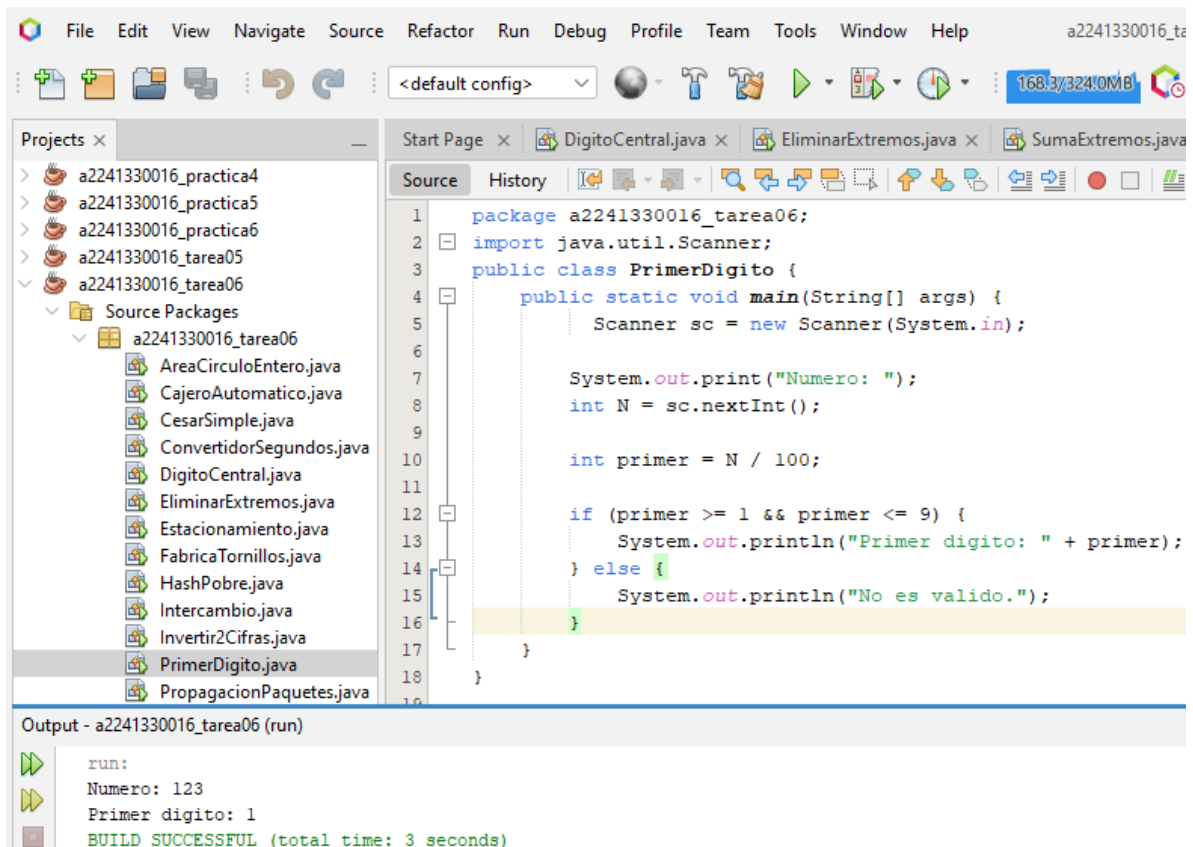
Suma Extremos:



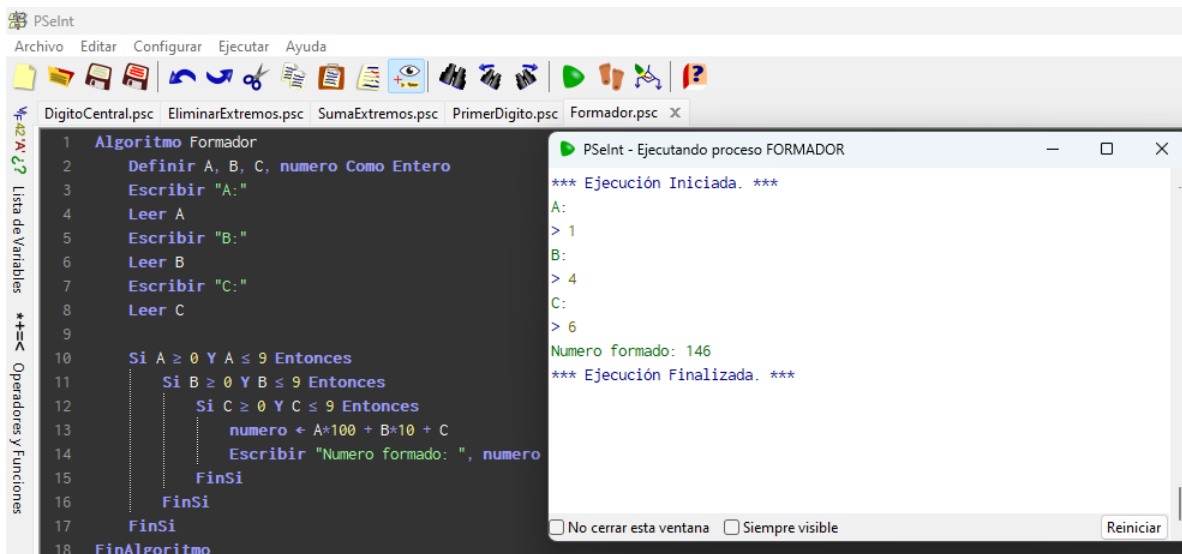


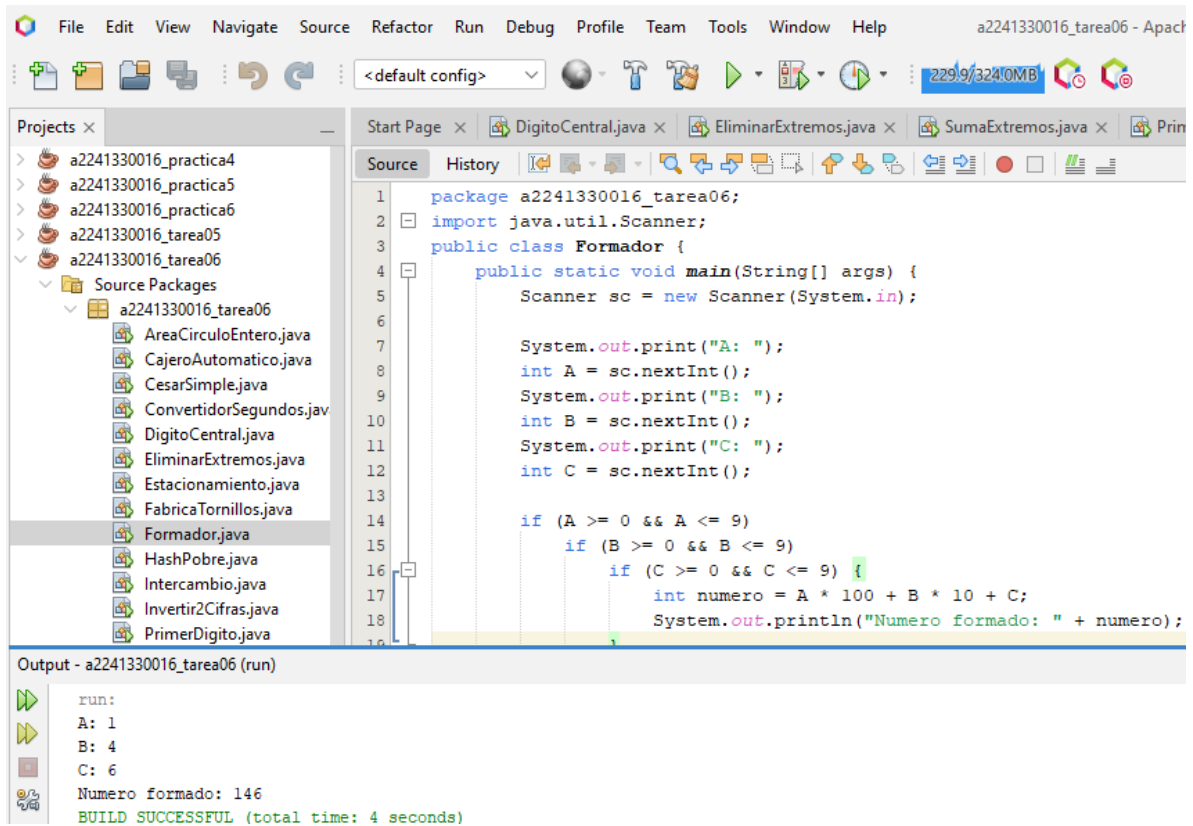
Primer Dígito:





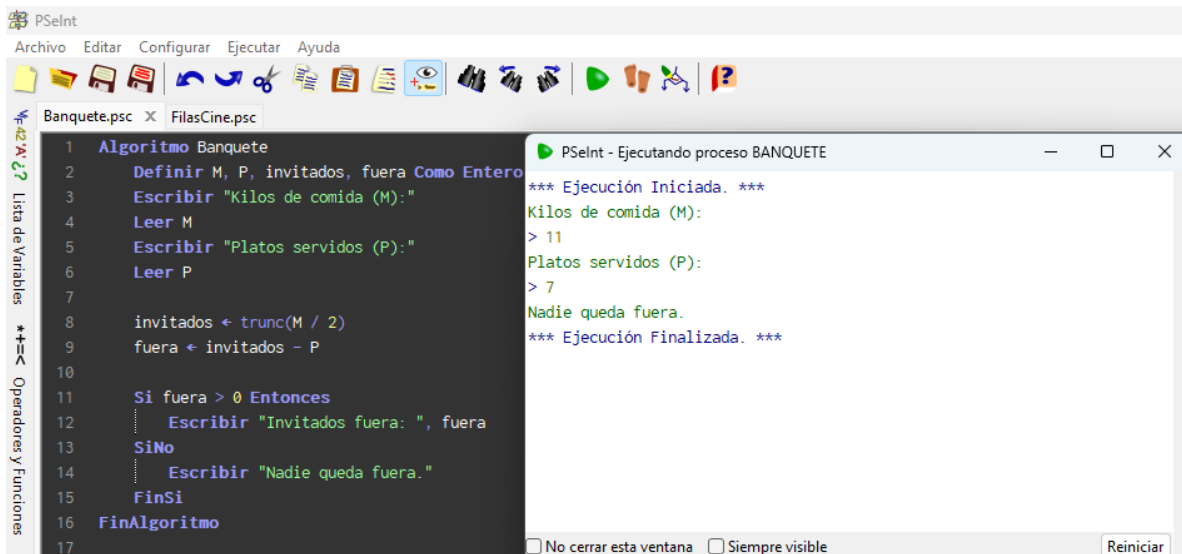
Formador:

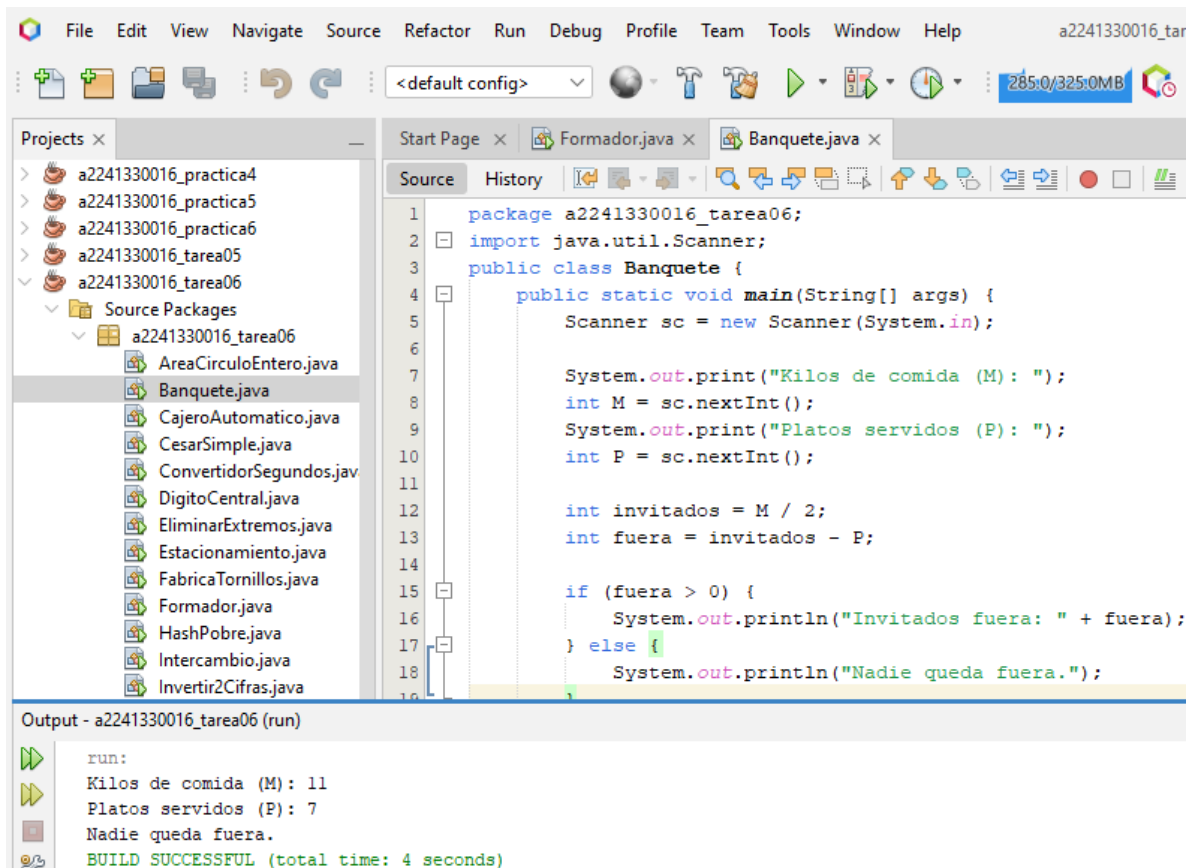




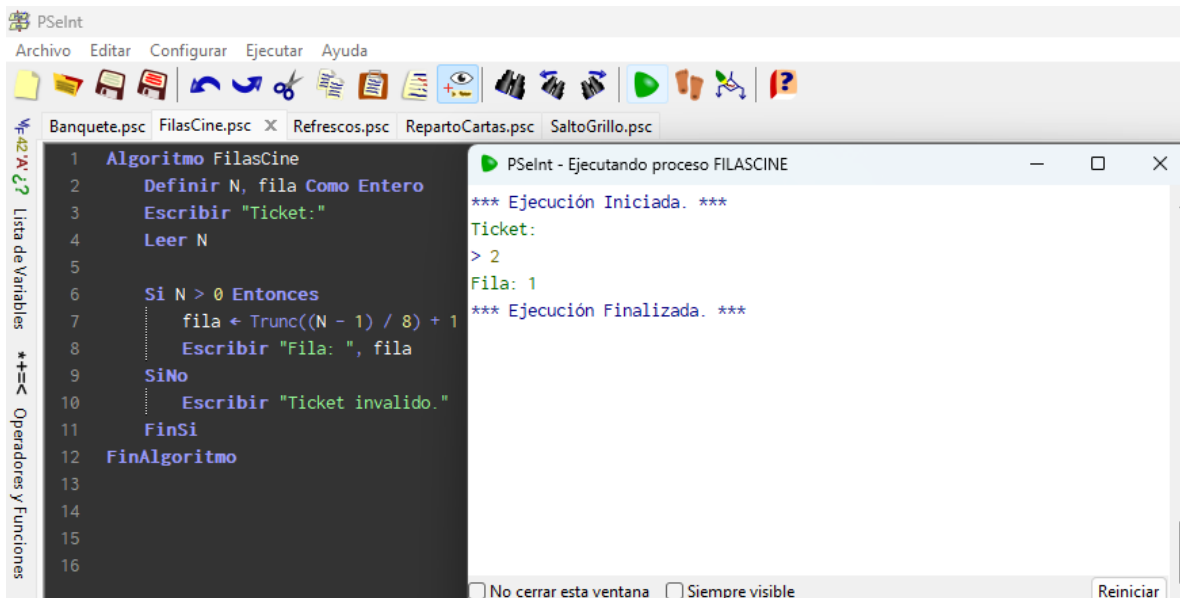
Bloque 6: Logística y Reparto:

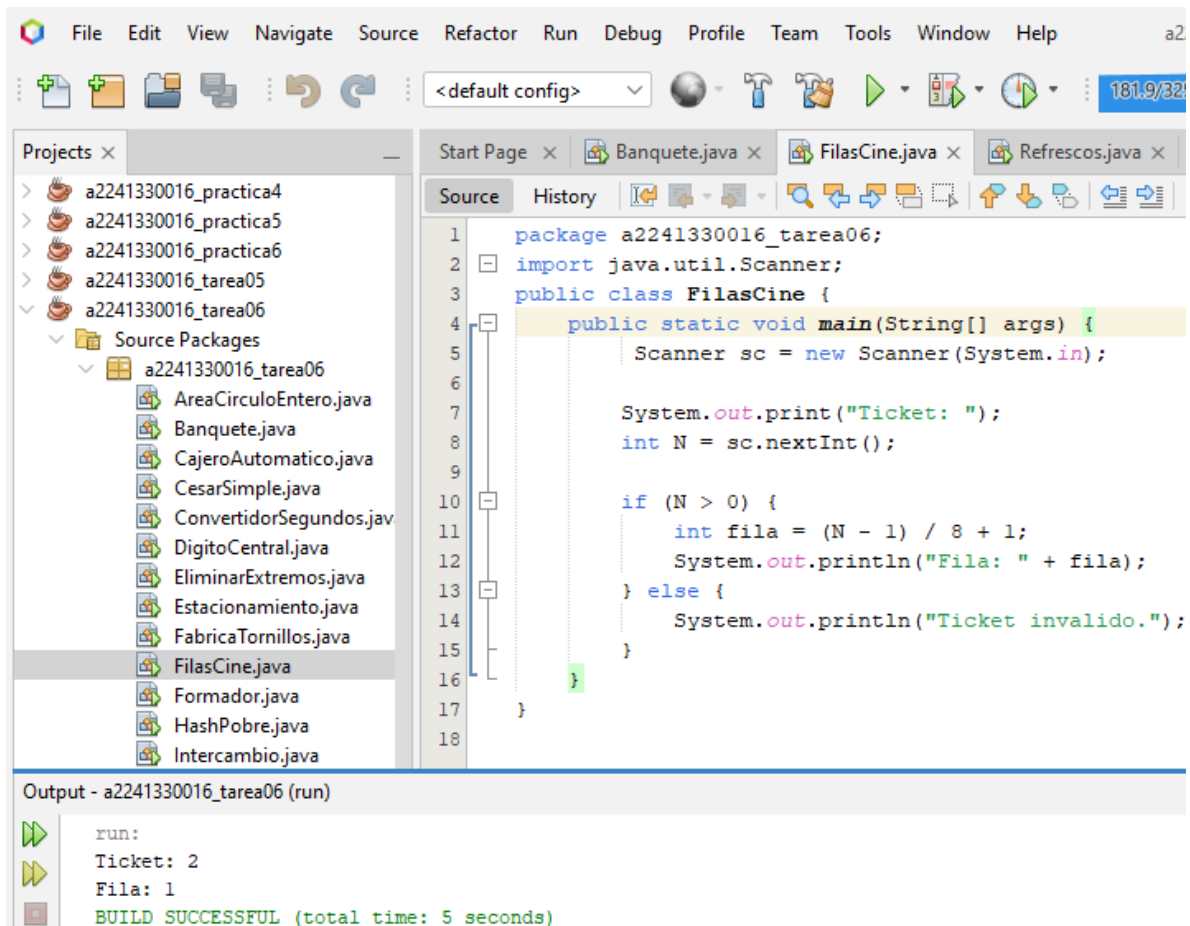
Banquete:



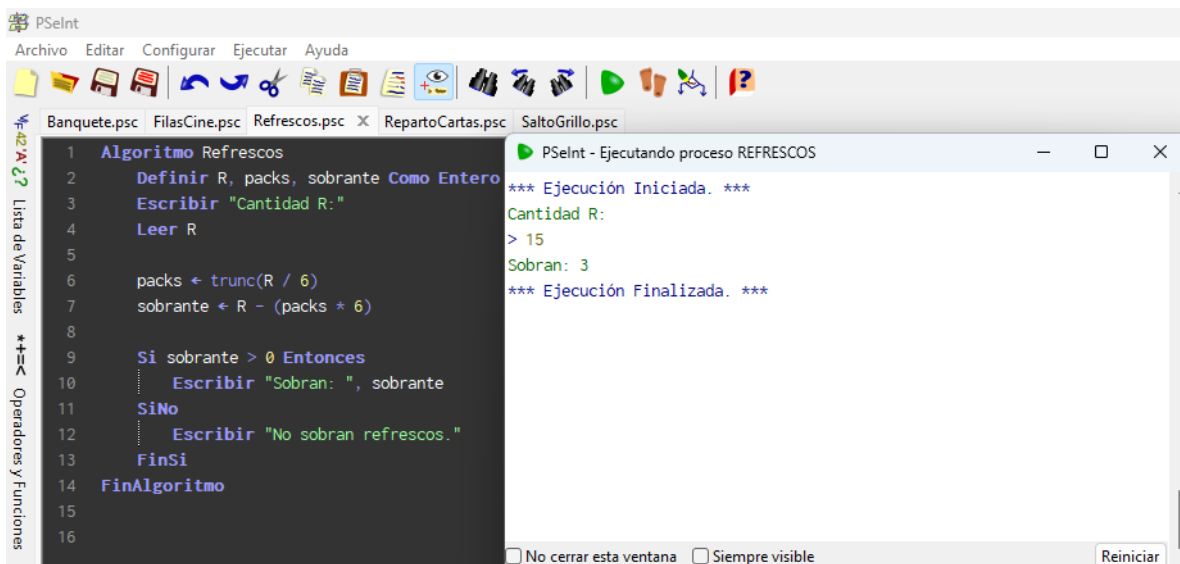


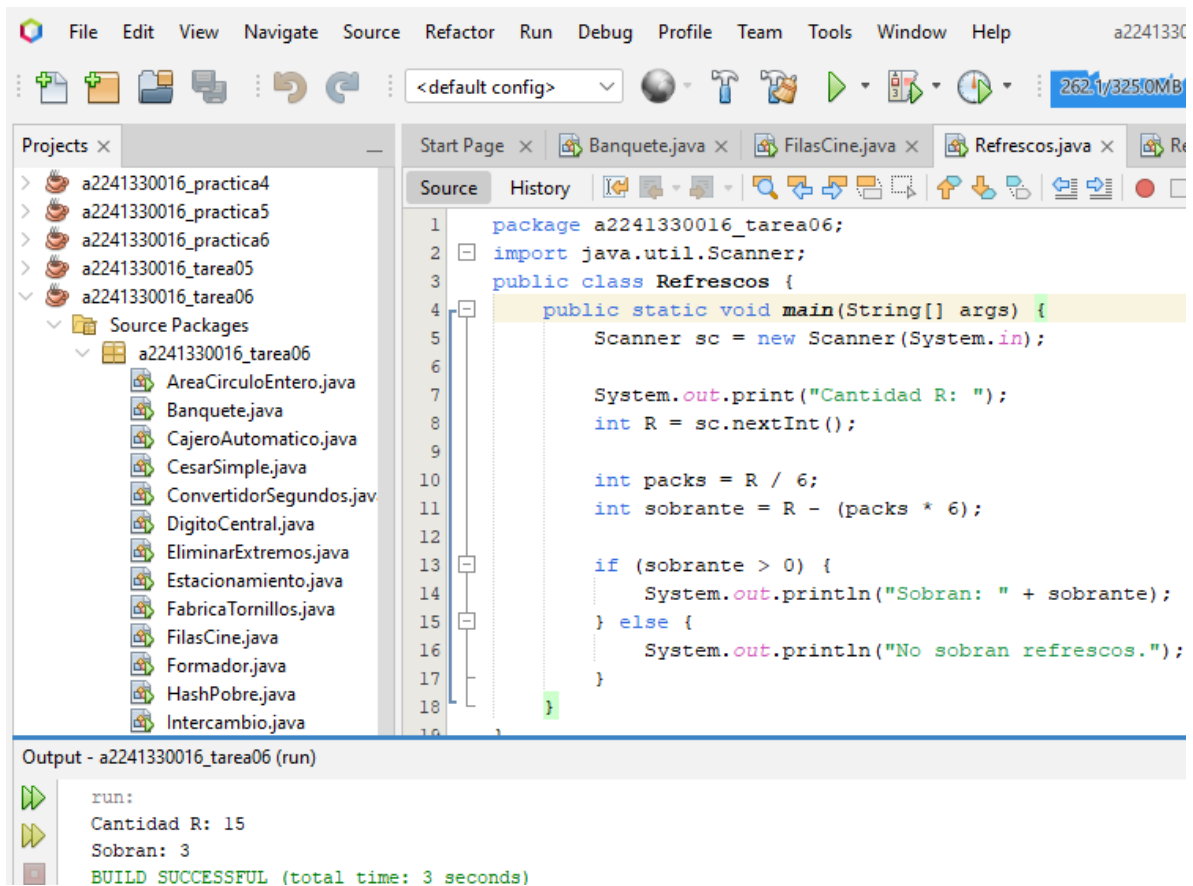
Filas Cine:



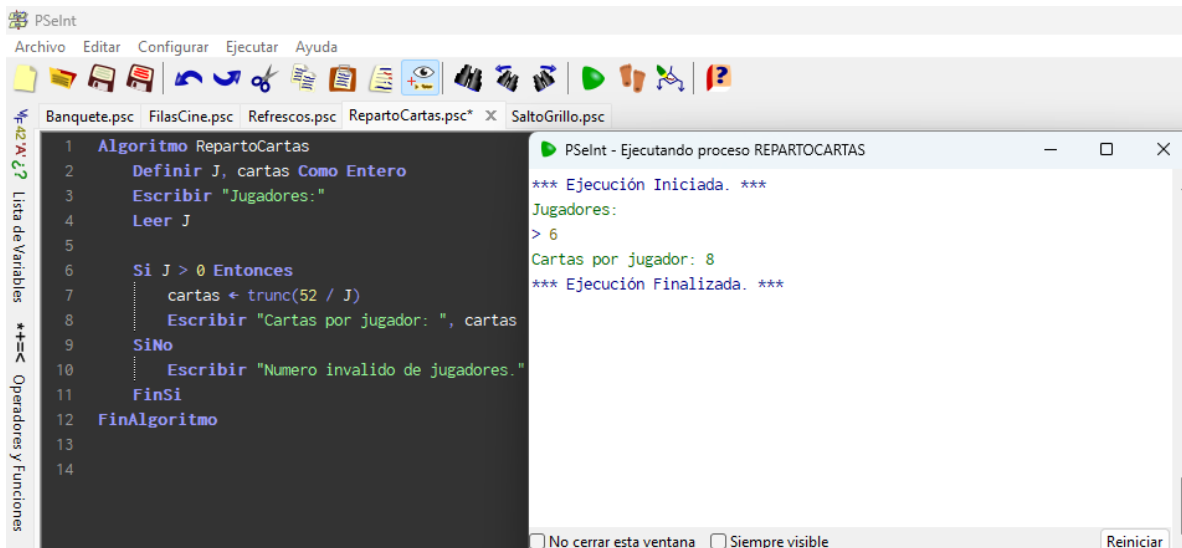


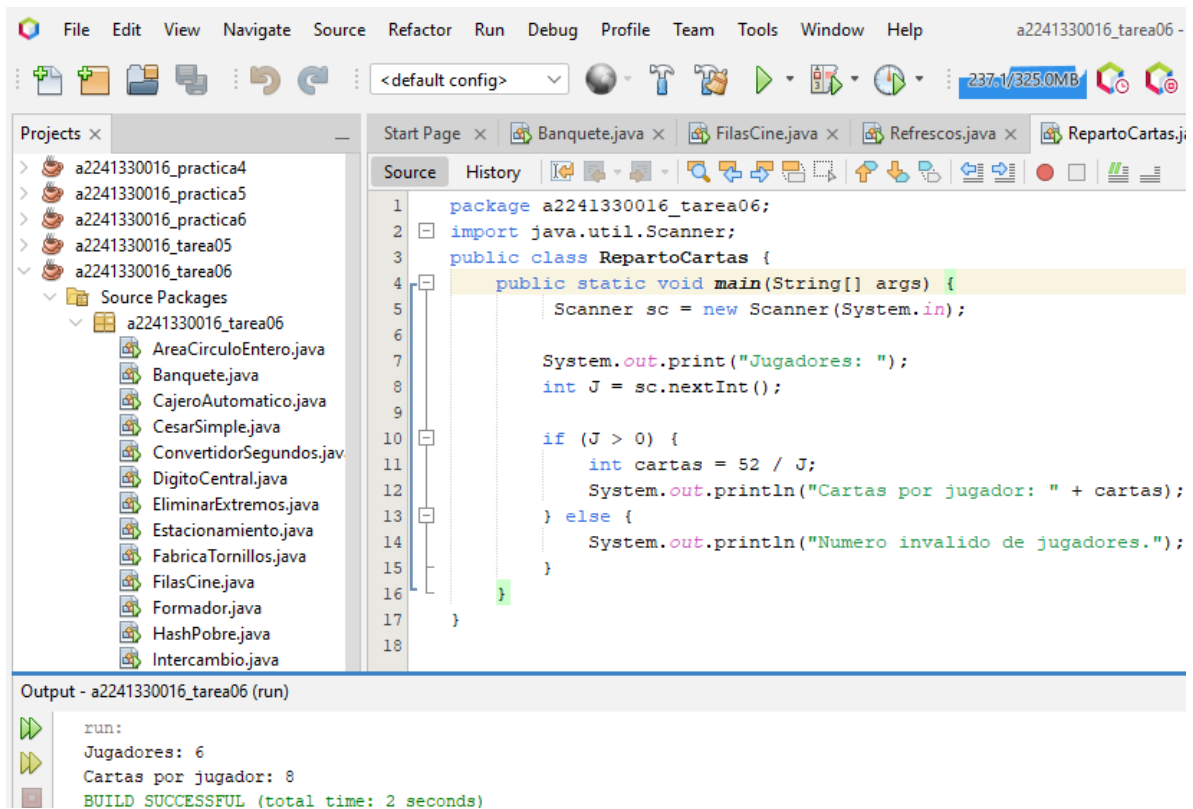
Refrescos:



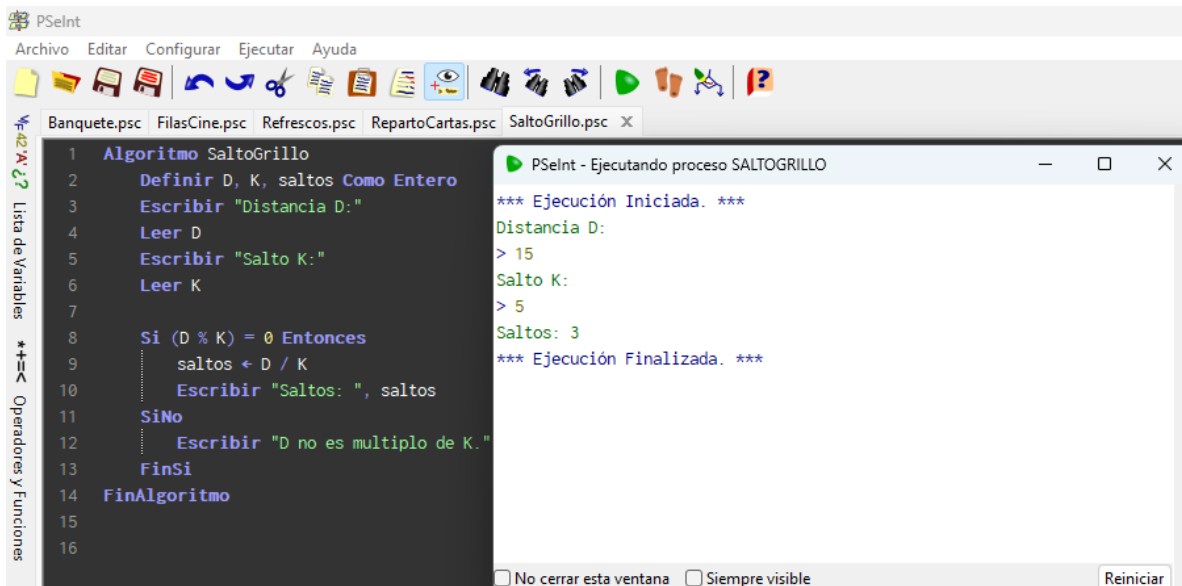


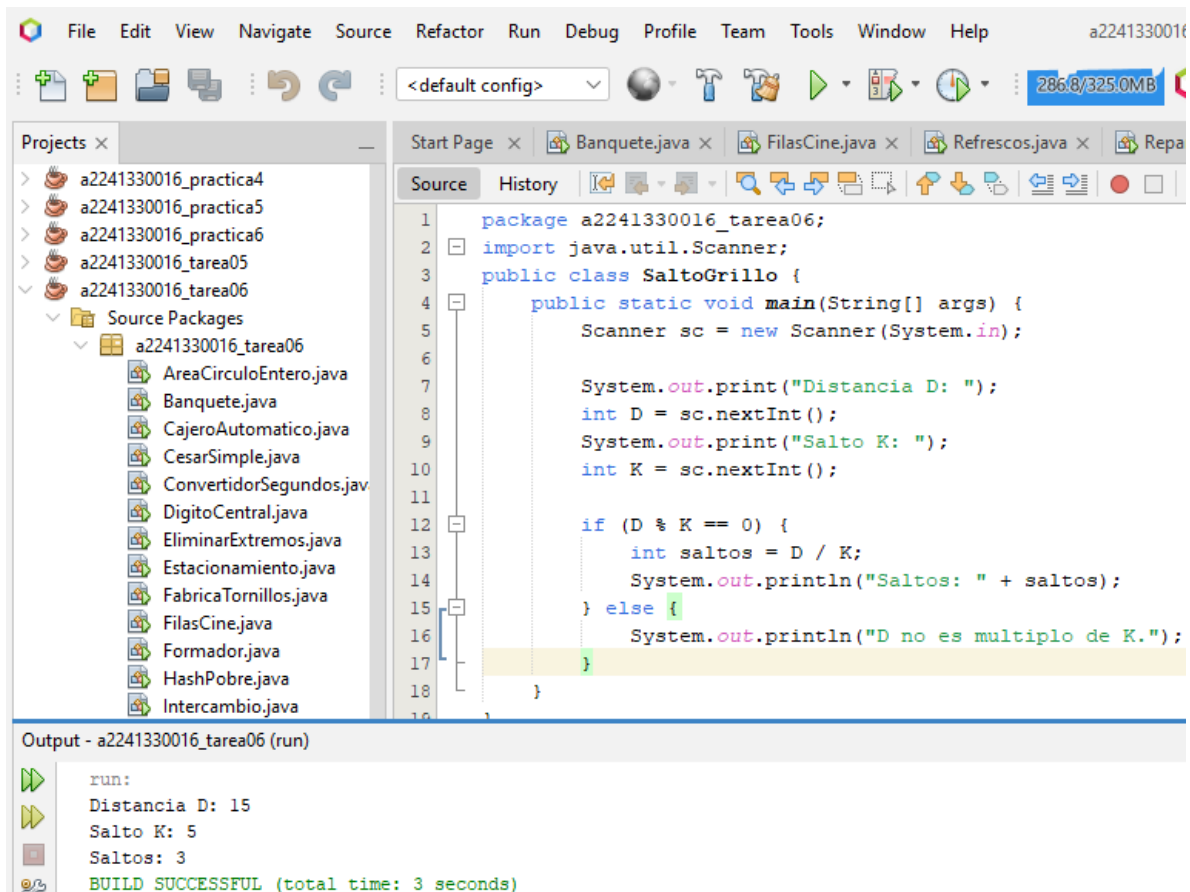
Reparto Cartas:





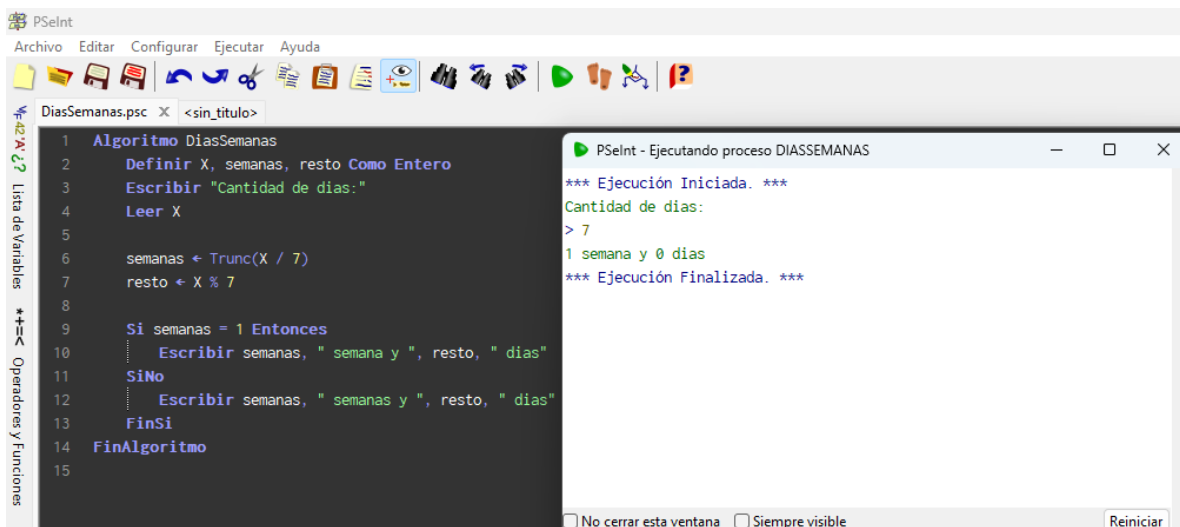
Salto Grillo:





Bloque 7: Tiempo y Conversiones

Días Semanas:



The screenshot shows an IDE window titled "a2241330016_tarea06 - Apache NetBeans". The "Projects" pane on the left shows a project structure with "Source Packages" containing several Java files, including "DiasSemanas.java". The "Source" pane displays the following Java code:

```
1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class DiasSemanas {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Cantidad de dias: ");
8         int X = sc.nextInt();
9
10        int semanas = X / 7;
11        int resto = X % 7;
12
13        if (semanas == 1) {
14            System.out.println(semanas + " semana y " + resto + " dias");
15        } else {
16            System.out.println(semanas + " semanas y " + resto + " dias");
17        }
18    }
19 }
```

The "Output" pane at the bottom shows the execution results:

```
run:
Cantidad de dias: 7
1 semana y 0 dias
BUILD SUCCESSFUL (total time: 2 seconds)
```

Horario Laboral:

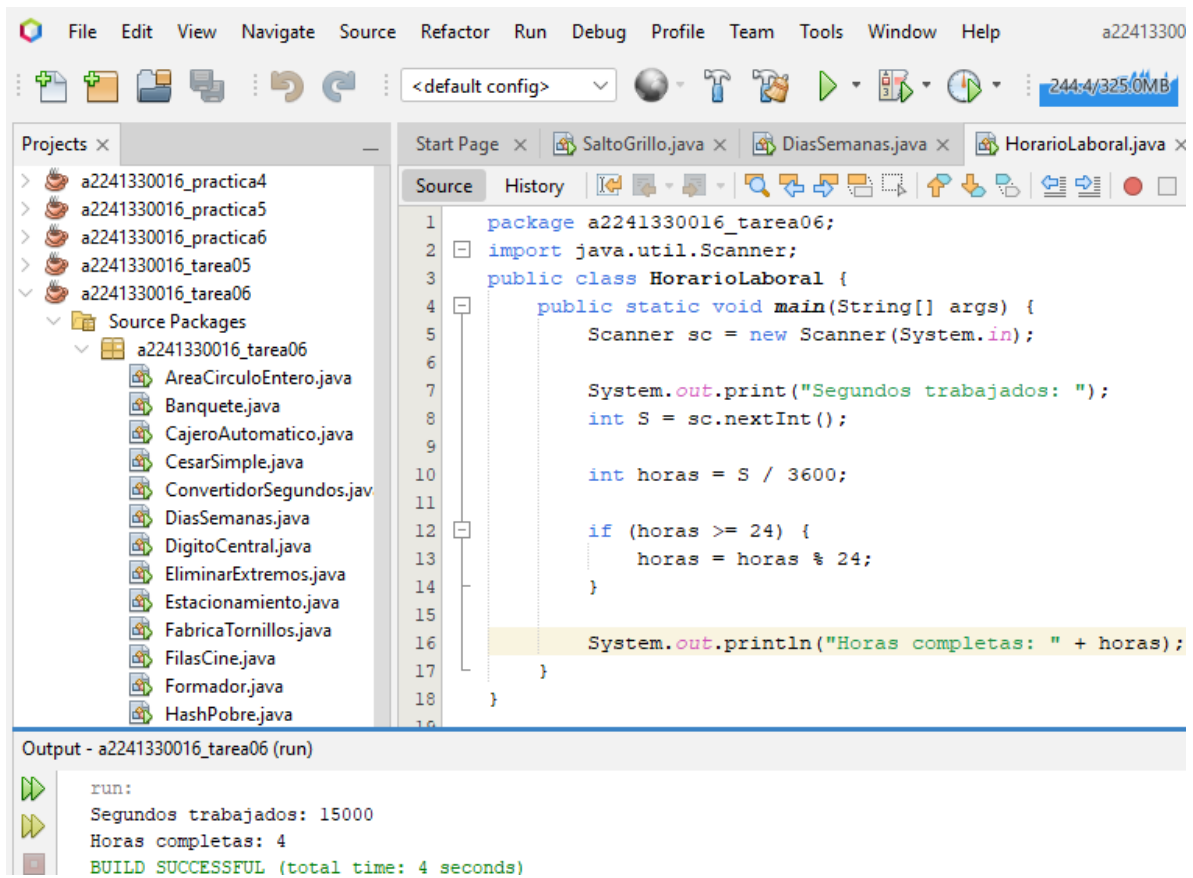
The screenshot shows the PSeInt IDE. The "HorarioLaboral.psc" file contains the following Pascal code:

```
1 Algoritmo HorarioLaboral
2   Definir S, horas Como Entero
3   Escribir "Segundos trabajados:"
4   Leer S
5
6   horas ← Trunc(S / 3600)
7
8   Si horas ≥ 24 Entonces
9       horas ← horas % 24
10  FinSi
11
12  Escribir "Horas completas: ", horas
13 FinAlgoritmo
```

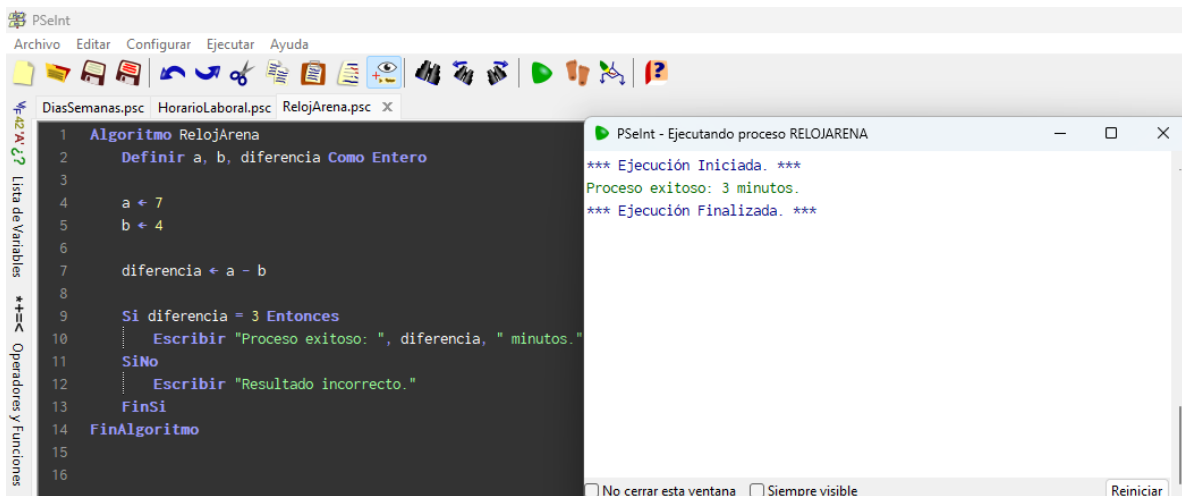
The "PSeInt - Ejecutando proceso HORARIOLABORAL" window shows the execution output:

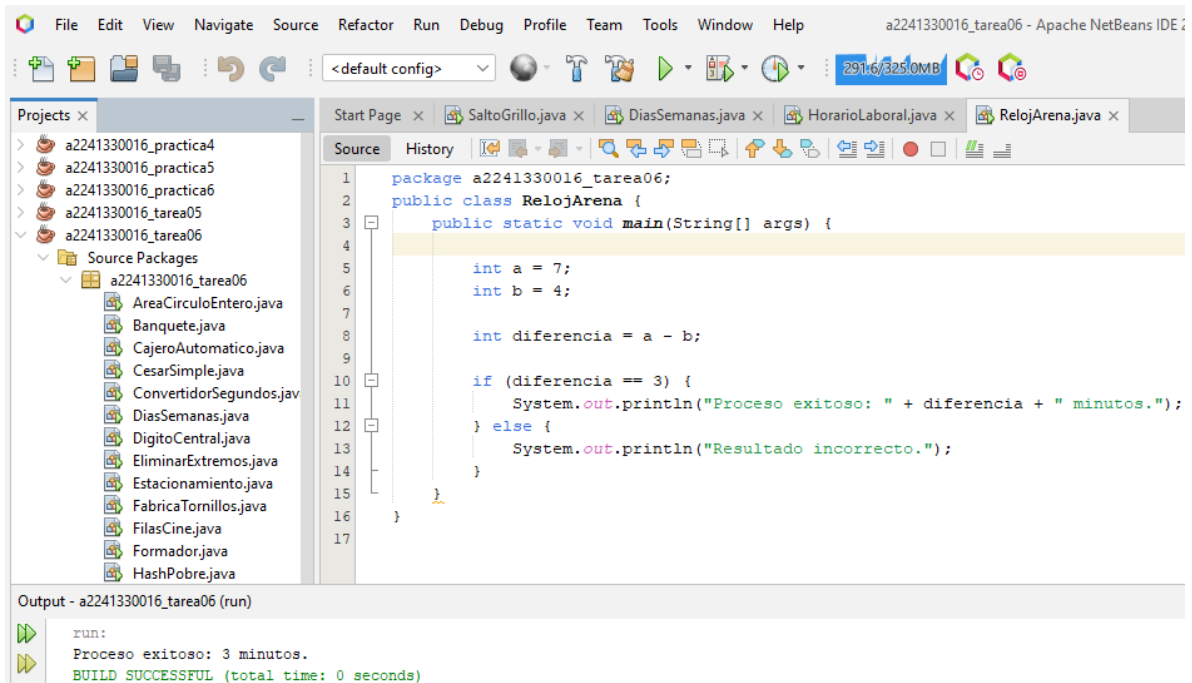
```
*** Ejecución Iniciada. ***
Segundos trabajados:
> 15000
Horas completas: 4
*** Ejecución Finalizada. ***
```

At the bottom, there are checkboxes for "No cerrar esta ventana" and "Siempre visible", and a "Reiniciar" button.

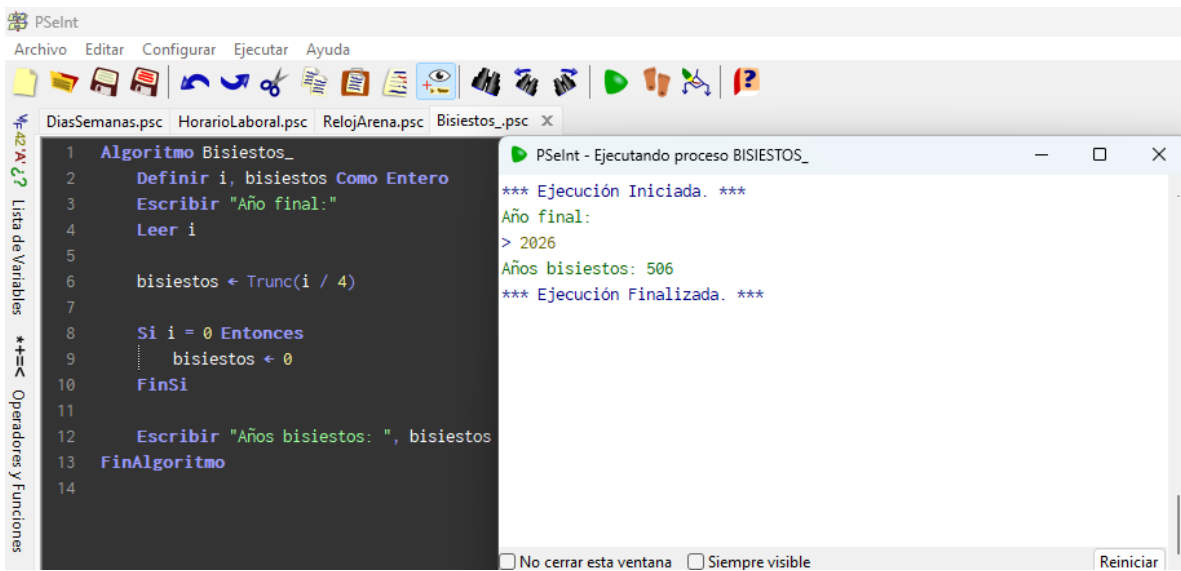


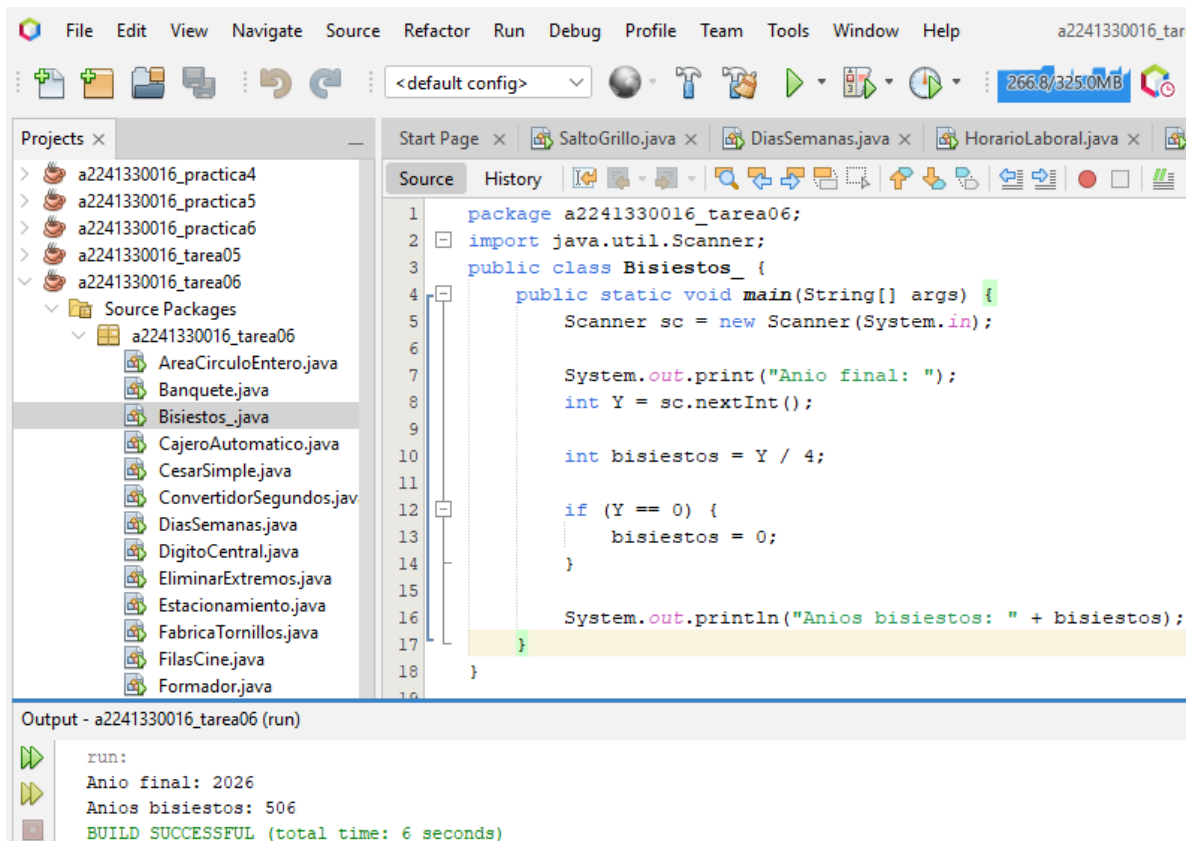
Reloj Arena:





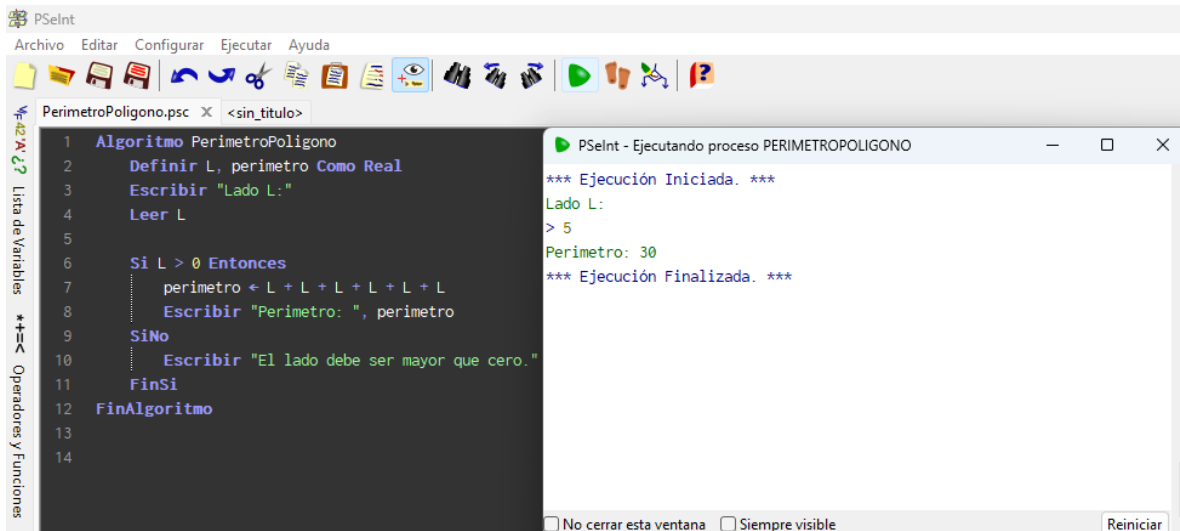
Años Bisiestos:

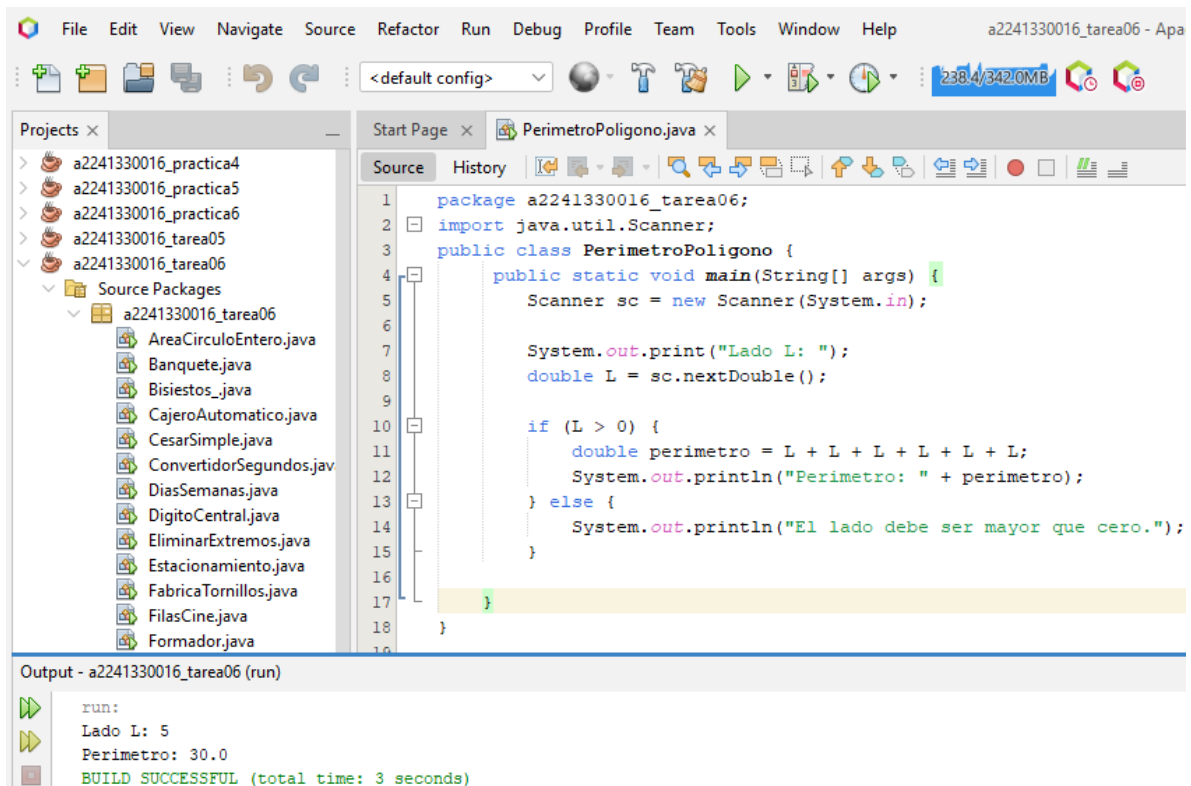




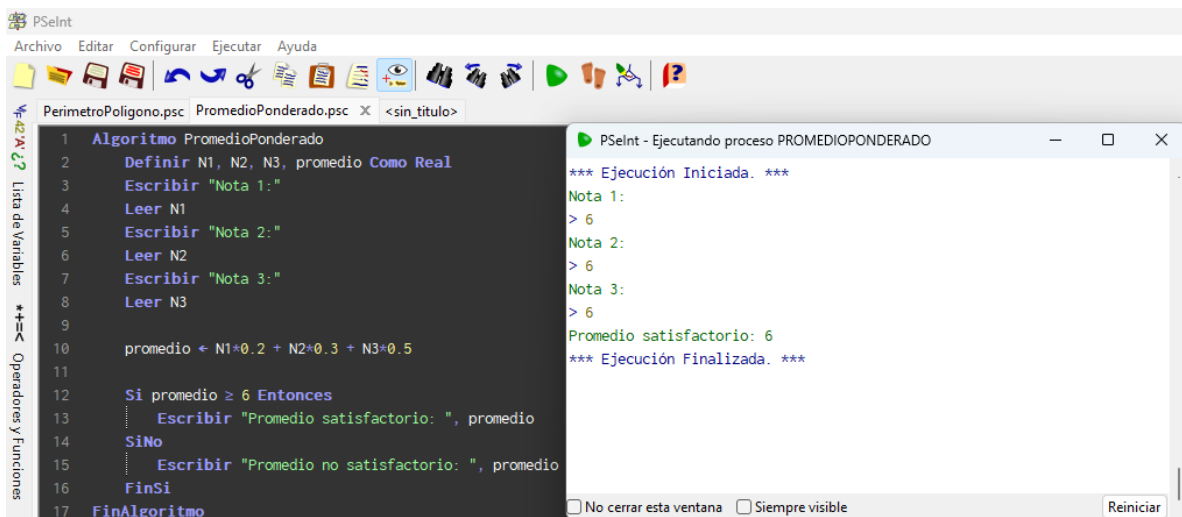
Bloque 8: Geometría y Física

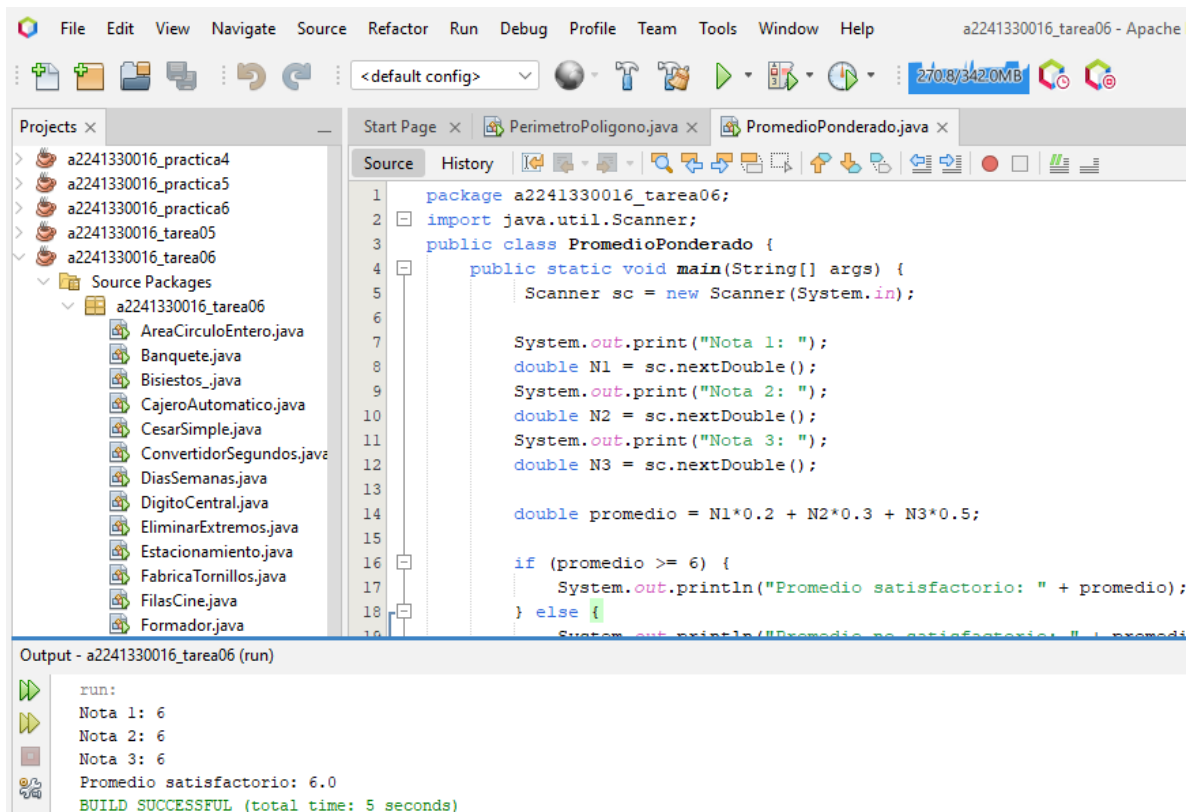
Perímetro Polígono:



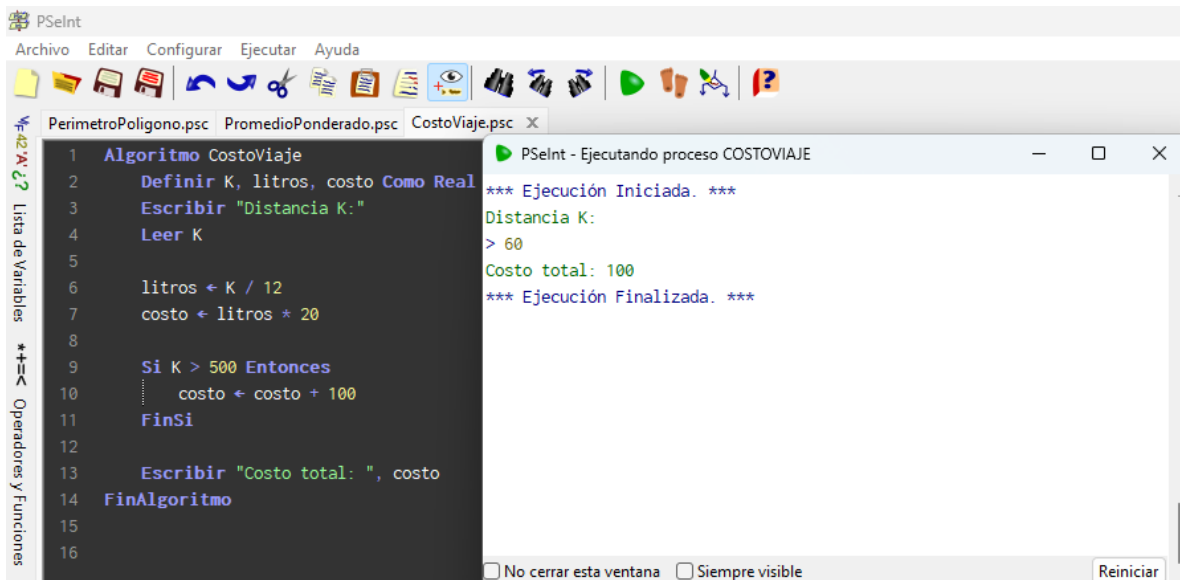


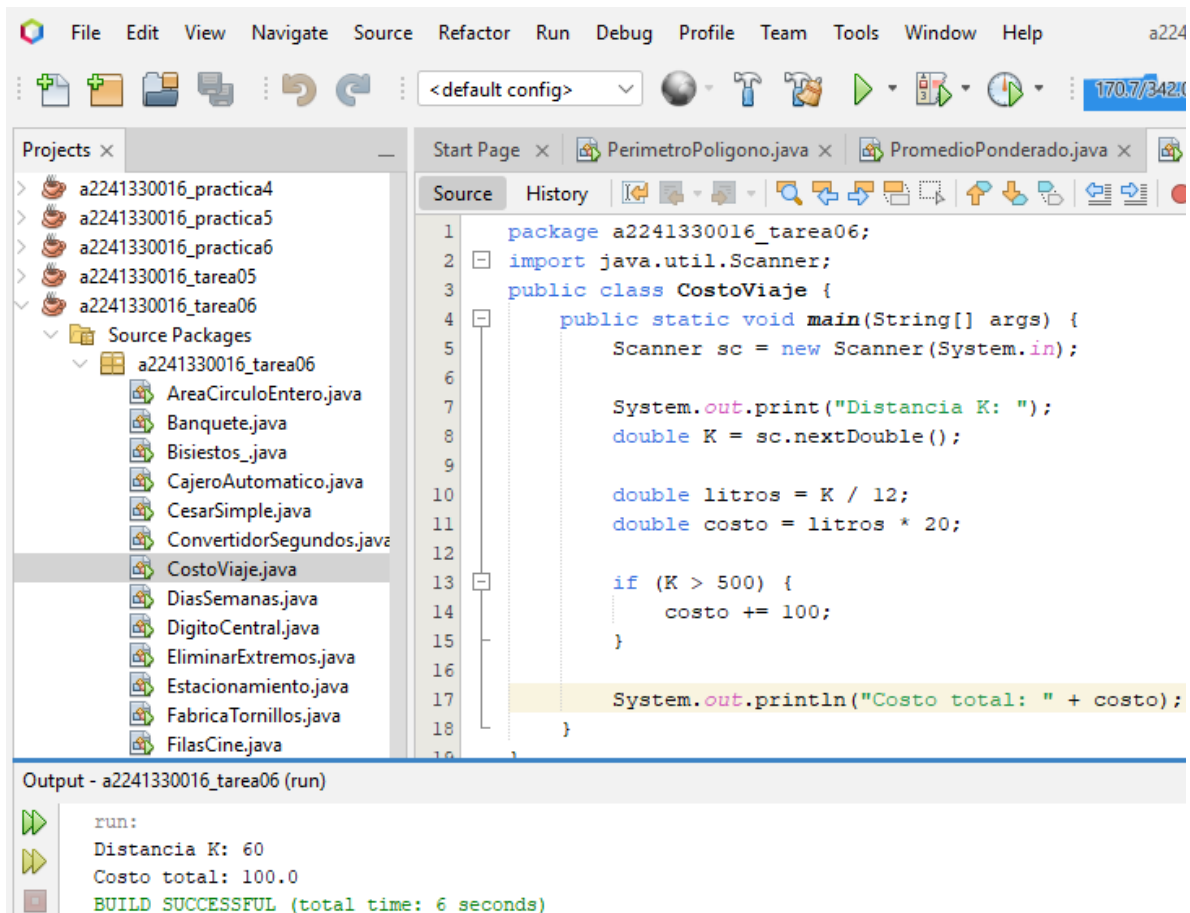
Promedio Ponderado:



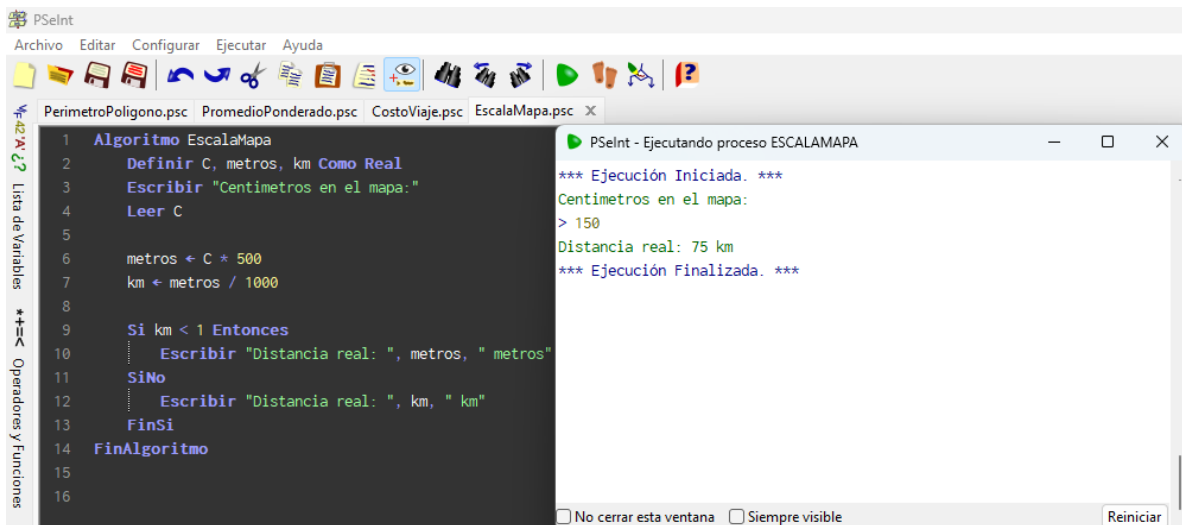


Costo Viaje:





Escala Mapa:



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a2241330016_tarea06 - Apache Net

206.4/342.0MB

Projects ×

- > a2241330016_practica4
- > a2241330016_practica5
- > a2241330016_practica6
- > a2241330016_tarea05
- ▼ a2241330016_tarea06
 - Source Packages
 - AreaCirculoEntero.java
 - Banquete.java
 - Bisiestos_java
 - CajeroAutomatico.java
 - CesarSimple.java
 - ConvertidorSegundos.java
 - CostoViaje.java
 - DiasSemanas.java
 - DigitoCentral.java
 - EliminarExtremos.java
 - EscalaMapa.java
 - Estacionamiento.java
 - FabricaTornillos.java

Start Page × PerimetroPoligono.java × PromedioPonderado.java × CostoViaje.java × Escal

Source History

```
1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class EscalaMapa {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Centimetros en el mapa: ");
8         double C = sc.nextDouble();
9
10        double metros = C * 500;
11        double km = metros / 1000;
12
13        if (km < 1) {
14            System.out.println("Distancia real: " + metros + " metros");
15        } else {
16            System.out.println("Distancia real: " + km + " km");
17        }
18    }
19 }
```

Output - a2241330016_tarea06 (run)

run:
Centimetros en el mapa: 150
Distancia real: 75.0 km
BUILD SUCCESSFUL (total time: 6 seconds)