



Universidad Autónoma De Tamaulipas

Fundamentos de Programación

1-N

Tarea 06

Molina Meneses Diego

Bloque 1: Logistica y Reparto

Salto Caracol:

The screenshot displays two integrated development environments (IDEs) side-by-side, illustrating the implementation of a "Salto Caracol" (Frog Jump) algorithm.

PSeInt Environment (Top):

- Title Bar:** PSeInt
- Menu Bar:** Archivo, Editar, Configurar, Ejecutar, Ayuda
- Toolbar:** Includes icons for file operations, search, and execution.
- Code Editor:** Shows the pseudocode for the "SaltoCaracol" algorithm.

```
Algoritmo SaltoCaracol
    Definir H, S, R, dias Como Entero
    Escribir "Altura (H):"
    Leer H
    Escribir "Sube por dia (S):"
    Leer S
    Escribir "Resbala por noche (R):"
    Leer R
    Si S ≥ H Entonces
        dias ← 1
    SiNo
        dias ← (H - R - 1) / (S - R) + 1
    FinSi
    Escribir "Dias: ", dias
FinAlgoritmo
```
- Output Window:** Shows the execution log of the algorithm with user input and output.

```
PSelint - Ejecutando proceso SALTOCARACOL
*** Ejecución Iniciada. ***
Altura (H):
> 20
Sube por dia (S):
> 5
Resbala por noche (R):
> 4
Dias: 16
*** Ejecución Finalizada. ***
```

Eclipse IDE Environment (Bottom):

- Title Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help
- Toolbar:** Includes icons for file operations, search, and execution.
- Projects View:** Shows various Java projects and source files, with `SaltoCaracol.java` selected under `a2241330016_tarea06`.
- Code Editor:** Displays the Java code for the "SaltoCaracol" algorithm, mirroring the pseudocode from PSeInt.

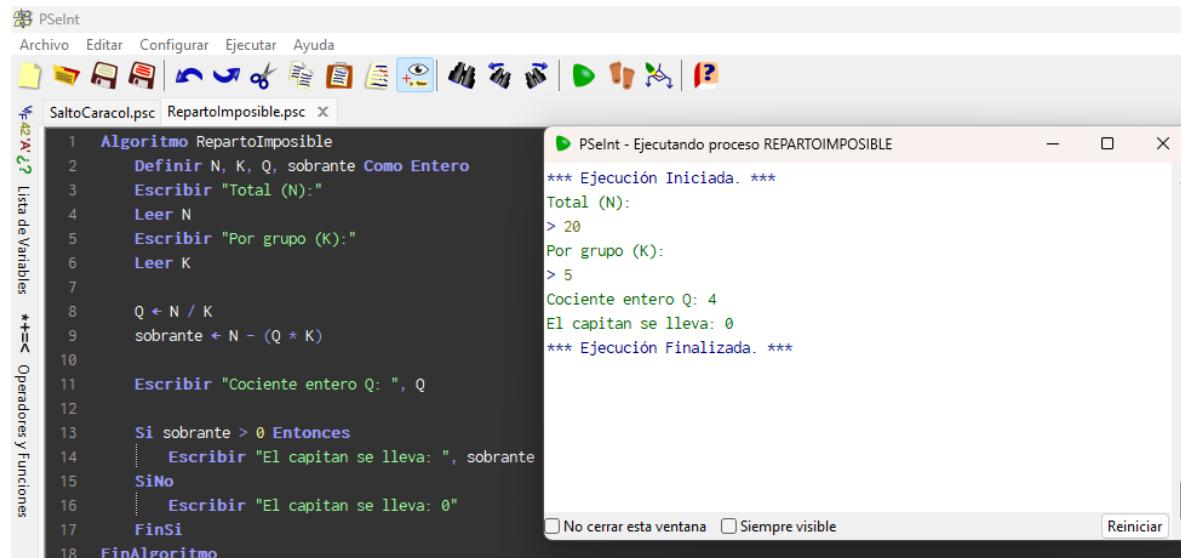
```
package a2241330016_tarea06;
import java.util.Scanner;
public class SaltoCaracol {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Altura (H): ");
        int H = sc.nextInt();
        System.out.print("Sube por dia (S): ");
        int S = sc.nextInt();
        System.out.print("Resbala por noche (R): ");
        int R = sc.nextInt();

        int dias;
        if (S >= H) {
            dias = 1;
        } else {
            dias = (H - R - 1) / (S - R) + 1;
        }
    }
}
```
- Output View:** Shows the terminal output of the Java application, matching the execution results from the PSeInt window.

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

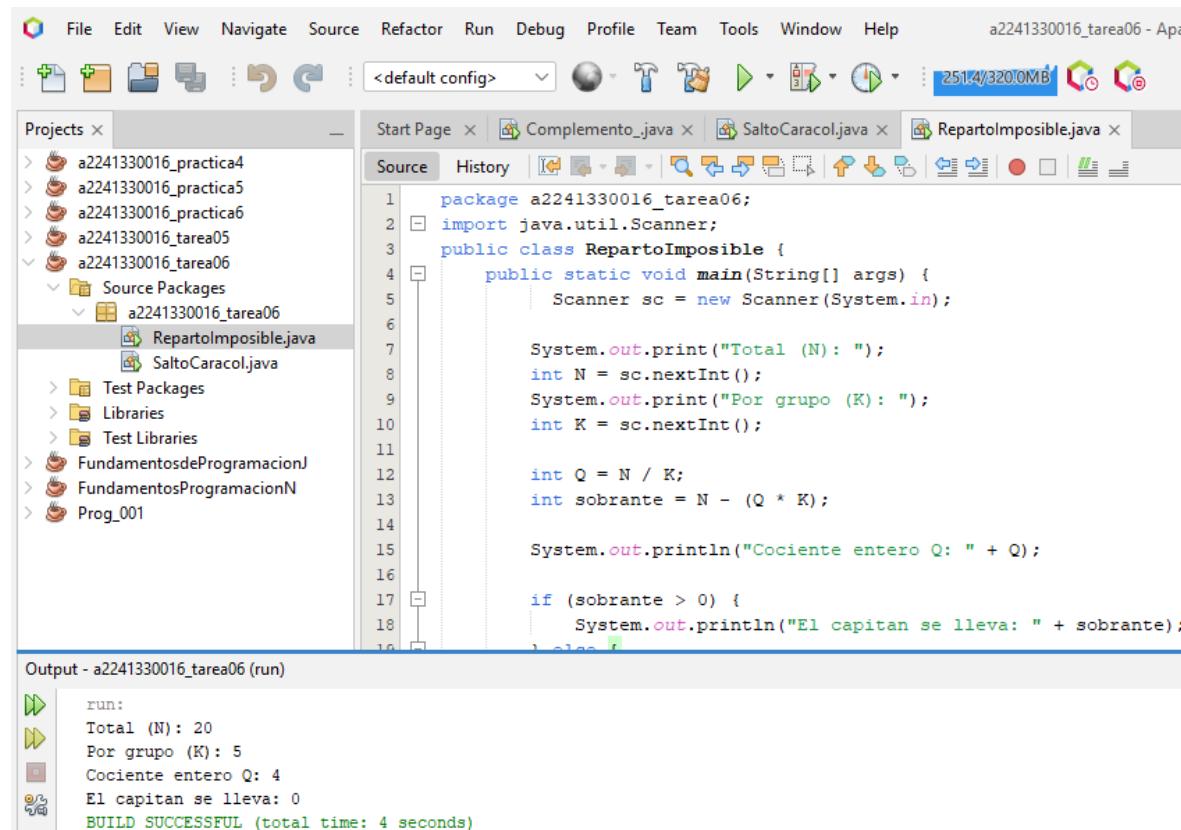
Reparto Imposible:



The screenshot shows the PSeInt IDE interface. On the left, the code editor displays the pseudocode for the 'RepartoImposible' algorithm. On the right, the terminal window shows the execution results.

```
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 capitán se lleva: ", sobrante
15  SiNo
16    Escribir "El capitán se lleva: 0"
17  FinSi
18 FinAlgoritmo
```

PSeInt - Ejecutando proceso REPARTOIMPOSIBLE
*** Ejecución Iniciada. ***
Total (N):
> 20
Por grupo (K):
> 5
Cociente entero Q: 4
El capitán se lleva: 0
*** Ejecución Finalizada. ***



The screenshot shows the Eclipse IDE interface. On the left, the 'Projects' view shows the Java project structure. On the right, the code editor displays the 'RepartoImposible.java' file.

```
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 capitán se lleva: " + sobrante);
19        } else {
20        }
21    }
22}
```

Output - a2241330016_tarea06 (run)
run:
Total (N): 20
Por grupo (K): 5
Cociente entero Q: 4
El capitán se lleva: 0
BUILD SUCCESSFUL (total time: 4 seconds)

Fabrica Tornillos:

PSelnt

Archivo Editar Configurar Ejecutar Ayuda

SaltoCaracol.psc RepartoImposible.psc FabricaTornillos.psc

```

1 Algoritmo FabricaTornillos
2   Definir X, cajas, sueltos Como Entero
3   Escribir "Tornillos totales (X):"
4   Leer X
5
6   cajas = trunc(X / 12)
7   sueltos = X - (cajas * 12)
8
9   Escribir "Cajas completas: ", cajas
10
11  Si sueltos > 0 Entonces
12    Escribir "Tornillos sueltos: ", sueltos
13  SiNo
14    Escribir "Tornillos sueltos: 0"
15  FinSi
16 FinAlgoritmo
17

```

PSelnt - Ejecutando proceso FABRICATORNILLOS
*** Ejecución Iniciada. ***
Tornillos totales (X):
> 150
Cajas completas: 12
Tornillos sueltos: 6
*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

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

Projects x Start Page x Complemento_java x SaltoCaracol.java x RepartoImposible.java x

Source History

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class FabricaTornillos {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Tornillos totales (X): ");
8         int X = sc.nextInt();
9
10        int cajas = X / 12;
11        int sueltos = X - (cajas * 12);
12
13        System.out.println("Cajas completas: " + cajas);
14
15        if (sueltos > 0) {
16            System.out.println("Tornillos sueltos: " + sueltos);
17        } else {
18            System.out.println("Tornillos sueltos: 0");
19

```

Output - a2241330016_tarea06 (run)

run:
Tornillos totales (X): 150
Cajas completas: 12
Tornillos sueltos: 6
BUILD SUCCESSFUL (total time: 4 seconds)

Cajero Automático:

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

SaltoCaracol.psc Repartolimposible.psc FabricaTornillos.psc CajeroAutomatico.psc

```

1 Algoritmo CajeroAutomatico
2   Definir Cantidad, entrega, perdida Como Entero
3   Escribir "Cantidad a retirar:"
4   Leer Cantidad
5
6   entrega ← (Cantidad / 100) * 100
7   perdida ← Cantidad - entrega
8
9   Escribir "Entrega del cajero: ", entrega
10
11  Si perdida > 0 Entonces
12    Escribir "Perdida del usuario: ", perdida
13  SiNo
14    Escribir "Perdida del usuario: 0"
15  FinSi
16 FinAlgoritmo
17

```

PSeInt - Ejecutando proceso CAJEROAUTOMATICO
*** Ejecución Iniciada. ***
Cantidad a retirar:
> 1500
Entrega del cajero: 1500
Perdida del usuario: 0
*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

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

Projects x Start Page x Complemento_java x SaltoCaracol.java x Repartolimposible.java x

Source History

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class CajeroAutomatico {
4   public static void main(String[] args) {
5     Scanner sc = new Scanner(System.in);
6
7     System.out.print("Cantidad a retirar: ");
8     int cantidad = sc.nextInt();
9
10    int entrega = (cantidad / 100) * 100;
11    int perdida = cantidad - entrega;
12
13    System.out.println("Entrega del cajero: " + entrega);
14
15    if (perdida > 0) {
16      System.out.println("Perdida del usuario: " + perdida);
17    } else {
18      System.out.println("Perdida del usuario: 0");
19

```

Output - a2241330016_tarea06 (run)

run:
Cantidad a retirar: 1500
Entrega del cajero: 1500
Perdida del usuario: 0
BUILD SUCCESSFUL (total time: 5 seconds)

Bloque 2: Tiempo y Conversiones

Reloj 24 Horas:

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

Reloj24Horas.psc

```

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

```

PSeInt - Ejecutando proceso RELOJ24HORAS

*** Ejecución Iniciada. ***

Total de minutos:
> 90
Hora resultante: 1

*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a22413300

Projects x Start Page x CajeroAutomatico.java x Reloj24Horas.java x

Source History

```

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
19

```

Output - a2241330016_tarea06 (run)

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

Convertidor Segundos:

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

Reloj24Horas.psc ConvertidorSegundos.psc X <sin_titulo>*

1 Algoritmo ConvertidorSegundos
2 Definir segund, horas, minutos, segundosFinal, resto Como Entero
3 Escribir "Segundos totales:"
4 Leer segund
5
6 horas ← trunc(segund / 3600)
7 resto ← segund - horas*3600
8
9 minutos ← resto / 60
10 segundosFinal ← resto - minutos*60
11
12 Si minutos ≥ 60 Entonces
13 minutos ← 59
14 FinSi
15
16 Si segundosFinal ≥ 60 Entonces
17 segundosFinal ← 59
18 FinSi
19
20 Escribir "Horas: ", horas
21 Escribir "Minutos: ", minutos
22 Escribir "Segundos: ", segundosFinal
23 FinAlgoritmo

PSeInt - Ejecutando proceso CONVERTIDORSEGUNDOS

*** Ejecución Iniciada. ***

Segundos totales:
> 1500
Horas: 0
Minutos: 25
Segundos: 0

*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

```

package a2241330016_tarea06;
import java.util.Scanner;
public class ConvertidorSegundos {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Segundos totales: ");
        int segundos = sc.nextInt();
        int horas = segundos / 3600;
        int resto = segundos - horas * 3600;
        int minutos = resto / 60;
        int segundosFinal = resto - minutos * 60;
        if (minutos >= 60) minutos = 59;
        if (segundosFinal >= 60) segundosFinal = 59;
        System.out.println("Horas: " + horas);
    }
}

```

Estacionamiento:

```

Algoritmo Estacionamiento
Definir minutos, horas, residuo Como Entero
Escribir "Tiempo en minutos:"
Leer minutos
horas ← trunc(minutos / 60)
residuo ← minutos - horas*60
Si residuo > 0 Entonces
    horas ← horas + 1
FinSi
Escribir "Horas a cobrar: ", horas
FinAlgoritmo

```

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class Estacionamiento {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Tiempo en minutos: ");
8         int minutos = sc.nextInt();
9
10        int horas = minutos / 60;
11        int residuo = minutos - horas * 60;
12
13        if (residuo > 0) {
14            horas++;
15        }
16
17        System.out.println("Horas a cobrar: " + horas);
18    }
19
20

```

Output - a2241330016_tarea06 (run)

```

run:
Tiempo en minutos: 30
Horas a cobrar: 1
BUILD SUCCESSFUL (total time: 15 seconds)

```

Bloque 3: Manipulación Numérica y Geometría

Invertir 2 Cifras:

PSelint

Archivo Editar Configurar Ejecutar Ayuda

Estacionamiento.psc Invertir2Cifras.psc

```

1 Algoritmo Invertir2Cifras
2   Definir N, D, U, invertido Como Entero
3   Escribir "Número de dos cifras:"
4   Leer N
5
6   D ← trunc(N / 10)
7   U ← N MOD 10
8
9   invertido ← U * 10 + D
10
11  Escribir "Número invertido: ", invertido
12 FinAlgoritmo
13
14

```

PSelint - Ejecutando proceso INVERTIR2CIFRAS

*** Ejecución Iniciada. ***

Número de dos cifras:
> 21

Número invertido: 12

*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

The screenshot shows an IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard icons for file operations like Open, Save, Find, etc.
- Project Explorer:** Shows a tree view of projects and source packages. The selected package is "a2241330016_tarea06" which contains several Java files: CajeroAutomatico.java, ConvertidorSegundos.java, Estacionamiento.java, FabricaTornillos.java, Invertir2Cifras.java (selected), Reloj24Horas.java, RepartolImposible.java, and SaltoCaracol.java.
- Code Editor:** Displays the Java code for "Invertir2Cifras.java". The code reads a two-digit number from standard input, separates it into tens (D) and units (U), and prints the reversed number (U * 10 + D).
- Output Window:** Shows the execution results:


```

run:
Número de dos cifras: 21
Número invertido: 12
BUILD SUCCESSFUL (total time: 3 seconds)

```

Área Círculo Cuadrado:

The screenshot shows the PSeInt environment with the following details:

- Toolbar:** Includes icons for opening files, saving, running, and executing processes.
- Project Explorer:** Shows two files: "Invertir2Cifras.psc" and "AreaCirculoEntero.psc".
- Pseudocode Editor:** Displays the pseudocode for "AreaCirculoEntero.psc":


```

Algoritmo AreaCirculoEntero
    Definir r, area Como Entero
    Escribir "Radio:"
    Leer r
    area ← 3 * r * r
    Si area > 78 Entonces
        area ← 78
    FinSi
    Escribir "Area aproximada: ", area
FinAlgoritmo

```
- Output Window:** Shows the execution of the process "AREACIRCULOENTERO" with the following output:


```

*** Ejecución Iniciada. ***
Radio:
> 11
Area aproximada: 78
*** Ejecución Finalizada. ***

```
- Status Bar:** Includes checkboxes for "No cerrar esta ventana" and "Siempre visible", and a "Reiniciar" button.

The screenshot shows an IDE interface with the following details:

- Projects:** A tree view showing several Java projects and their files. One project, "a2241330016_tarea06", is expanded, revealing files like AreaCirculoEntero.java, CajeroAutomatico.java, ConvertidorSegundos.java, Estacionamiento.java, FabricaTornillos.java, Invertir2Cifras.java, Reloj24Horas.java, Repartolimposible.java, and SaltoCaracol.java.
- Source Editor:** The main window displays the Java code for `AreaCirculoEntero`. The code calculates the area of a circle based on user input (radio) and prints the result (Area aproximada).
- Output:** The output window shows the execution results: "run:", "Radio: 11", "Area aproximada: 78", and "BUILD SUCCESSFUL (total time: 5 seconds)".

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class AreaCirculoEntero {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Radio: ");
8         int r = sc.nextInt();
9
10        int area = 3 * r * r;
11
12        if (area > 78) {
13            area = 78;
14        }
15
16        System.out.println("Area aproximada: " + area);
17    }
18}

```

Intercambio:

The screenshot shows the PSeint environment with the following details:

- Pseudocode Editor:** The left pane displays pseudocode for an "Intercambio" algorithm. It defines variables A and B, reads values, swaps them, and prints the result.
- Execution Window:** The right pane shows the execution process named "INTERCAMBIO". It logs the start, user inputs for values A and B, the swap operation, and the final output.

```

1 Algoritmo Intercambio
2   Definir A, B Como Entero
3   Escribir "Valor A:"
4   Leer A
5   Escribir "Valor B:"
6   Leer B
7
8   A ← A + B
9   B ← A - B
10  A ← A - B
11
12  Si A = (A) Entonces
13    Escribir "Intercambio realizado."
14  FinSi
15
16  Escribir "Nuevo A: ", A
17  Escribir "Nuevo B: ", B
18 FinAlgoritmo

```

PSeint - Ejecutando proceso INTERCAMBIO

*** Ejecución Iniciada. ***

Valor A:
> 5

Valor B:
> 7

Intercambio realizado.

Nuevo A: 7

Nuevo B: 5

*** Ejecución Finalizada. ***

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. Below the menu is a toolbar with various icons. The left side features a 'Projects' view with several Java projects listed under 'a2241330016_practica4' and 'a2241330016_tarea06'. The 'Source Packages' section for 'a2241330016_tarea06' contains files like AreaCirculoEntero.java, CajeroAutomatico.java, ConvertidorSegundos.java, Estacionamiento.java, FabricaTornillos.java, Intercambio.java, Invertir2Cifras.java, Reloj24Horas.java, RepartolImposible.java, and SaltoCaracol.java. The central workspace displays the source code for Intercambio.java:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class Intercambio {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Valor A: ");
8         int A = sc.nextInt();
9         System.out.print("Valor B: ");
10        int B = sc.nextInt();
11
12        A = A + B;
13        B = A - B;
14        A = A - B;
15
16        System.out.println("Nuevo A: " + A);
17        System.out.println("Nuevo B: " + B);
18    }
19

```

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

```

run:
Valor A: 5
Valor B: 7
Nuevo A: 7
Nuevo B: 5
BUILD SUCCESSFUL (total time: 8 seconds)

```

Hash Pobre:

The screenshot shows the PSelint tool interface. The top menu bar includes Archivo, Editar, Configurar, Ejecutar, and Ayuda. Below the menu is a toolbar with various icons. The left side has tabs for 'Intercambio.psc' and 'HashPobre.psc'. The 'HashPobre.psc' tab contains the following pseudocode:

```

1 Algoritmo HashPobre
2   Definir N, a, b, c, d, checksum Como Entero
3   Escribir "Número de 4 cifras:"
4   Leer N
5
6   a ← trunc(N / 1000)
7   b ← trunc(N / 100) MOD 10
8   c ← trunc(N / 10) MOD 10
9   d ← N MOD 10
10
11  checksum ← a + b + c + d
12
13  Escribir "Checksum: ", checksum
14 FinAlgoritmo
15
16

```

The right side shows the execution window titled 'PSelint - Ejecutando proceso HASHPOBRE' with the following output:

```

*** Ejecución Iniciada. ***
Número de 4 cifras:
> 2026
Checksum: 10
*** Ejecución Finalizada. ***

```

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

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a user ID a224. The toolbar contains various icons for file operations like Open, Save, and Build. The Projects tool window on the left lists several Java projects and packages, with 'a2241330016_tarea06' expanded to show its source files. The main code editor window displays the 'HashPobre.java' file:

```

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("Número 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

```

The Output tool window at the bottom shows the run results:

```

run:
Número 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 PSELNT environment. The top menu bar includes Archivo, Editar, Configurar, Ejecutar, and Ayuda. The toolbar has icons for file operations and execution. The main window displays a pseudocode editor with tabs for 'SubredesHosts.psc', 'SharingID.psc', 'CesarSimple.psc', and 'PropagacionPaquetes.psc'. The pseudocode for 'SubredesHosts.psc' is:

```

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
16
17

```

To the right, a terminal window titled 'PSELNT - Ejecutando proceso SUBREDESHOSTS' shows the execution results:

```

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

```

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

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects x Start Page x SubRedesHosts.java x SharingID.java x CesarS

Source History

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class SubRedesHosts {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Bits de host (N): ");
8         int N = sc.nextInt();
9
10        int total = (int) Math.pow(2, N);
11
12        int hosts;
13        if (total >= 2) {
14            hosts = total - 2;
15        } else {
16            hosts = 0;
17        }
18
19        System.out.println("Hosts disponibles: " + hosts);

```

Output - a2241330016_tarea06 (run)

run:
 Bits de host (N): 5
 Hosts disponibles: 30
 BUILD SUCCESSFUL (total time: 3 seconds)

SharingID:

PSelnt

Archivo Editar Configurar Ejecutar Ayuda

SubRedesHosts.psc SharingID.psc x CesarSimple.psc PropagacionPaquetes.psc

Algoritmo SharingID
 Definir ID, r Como Entero
 Escribir "ID:"
 Leer ID
 r ← ID MOD 3
 Si r = 0 Entonces
 Escribir "Servidor 0"
 FinSi
 Si r = 1 Entonces
 Escribir "Servidor 1"
 FinSi
 Si r = 2 Entonces
 Escribir "Servidor 2"
 FinSi
 FinAlgoritmo

PSelnt - Ejecutando proceso SHARINGID

*** Ejecución Iniciada. ***
 ID:
 > 10
 Servidor 1
 *** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

The screenshot shows an IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, and Project.
- Project Explorer:** Shows several projects under "a2241330016_practica5" and "a2241330016_tarea06".
- Code Editor:** The active tab is "SharingID.java" with the following code:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class SharingID {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("ID: ");
8         int ID = sc.nextInt();
9
10        int r = ID % 3;
11
12        if (r == 0) System.out.println("Servidor 0");
13        if (r == 1) System.out.println("Servidor 1");
14        if (r == 2) System.out.println("Servidor 2");
15    }
16 }
17 
```
- Output Window:** Shows the run command and the output "ID: 10 Servidor 1 BUILD SUCCESSFUL (total time: 3 seconds)".

Cesar Simple:

The screenshot shows the PSeint environment with the following details:

- Menu Bar:** Archivo, Editar, Configurar, Ejecutar, Ayuda.
- Toolbar:** Includes icons for file operations and execution.
- Script Editor:** The script is named "CesarSimple.psc" and contains pseudocode:

```

1 Algoritmo CesarSimple
2   Definir secreto, llave, cifra Como Entero
3   Escribir "Secreto (0-9):"
4   Leer secreto
5   Escribir "Llave (0-9):"
6   Leer llave
7
8   cifra ← secreto + llave
9
10  Si cifra ≥ 10 Entonces
11      cifra ← cifra - 10
12  FinSi
13
14  Escribir "Cifrado: ", cifra
15 FinAlgoritmo
16
17 
```
- Execution Window:** Shows the process "CESAR SIMPLE" running with the following interaction:

```

PSeint - Ejecutando proceso CESAR SIMPLE
*** Ejecución Iniciada. ***
Secreto (0-9):
> 8
Llave (0-9):
> 5
Cifrado: 3
*** Ejecución Finalizada. ***

```
- Status Bar:** Includes checkboxes for "No cerrar esta ventana" and "Siempre visible", and a "Reiniciar" button.

The screenshot shows an IDE interface with the following details:

- Projects:** A tree view showing several projects under "a2241330016_practica6". One project, "a2241330016_tarea06", is expanded, revealing files like AreaCirculoEntero.java, CajeroAutomatico.java, CesarSimple.java, etc.
- Source Editor:** Displays the Java code for "CesarSimple.java". The code implements a Caesar cipher where it adds a secret number (secreto) and a key (llave) to a character. If the result is greater than or equal to 10, it subtracts 10 to keep it within the 0-9 range.

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class CesarSimple {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Secreto (0-9): ");
8         int secreto = sc.nextInt();
9         System.out.print("Llave (0-9): ");
10        int llave = sc.nextInt();
11
12        int cifra = secreto + llave;
13
14        if (cifra >= 10) {
15            cifra = cifra - 10;
16        }
17
18        System.out.println("Cifrado: " + cifra);
19    }
}

```
- Output:** Shows the run results for "a2241330016_tarea06 (run)".

```

run:
Secreto (0-9): 8
Llave (0-9): 5
Cifrado: 3
BUILD SUCCESSFUL (total time: 2 minutes 34 seconds)

```

Propagación Paquetes:

The screenshot shows the PSELNT tool interface with the following details:

- Toolbar:** Includes icons for file operations, search, and execution.
- File List:** Shows four files: SubredesHosts.psc, SharingID.psc, CesarSimple.psc, and PropagacionPaquetes.psc.
- Code Editor:** Displays the pseudocode for the "PropagacionPaquetes" algorithm.

```

1 Algoritmo PropagacionPaquetes
2   Definir D, V, tiempo Como Entero
3   Escribir "Distancia (D):"
4   Leer D
5   Escribir "Velocidad (V):"
6   Leer V
7
8   tiempo ← trunc(D / V)
9
10  Si trunc(D MOD V) > 0 Entonces
11    ... tiempo ← tiempo + 1
12  FinSi
13
14  Escribir "Tiempo total: ", tiempo
15 FinAlgoritmo
16
17

```
- Execution Window:** Shows the output of the execution process "PROPAGACIONPAQUETES".

```

PSELNT - Ejecutando proceso PROPAGACIONPAQUETES
*** Ejecución Iniciada. ***
Distancia (D):
> 10
Velocidad (V):
> 3
Tiempo total: 4
*** Ejecución Finalizada. ***

```
- Status Bar:** Includes checkboxes for "No cerrar esta ventana" and "Siempre visible", and a "Reiniciar" button.

The screenshot shows a Java development environment with the following interface elements:

- Top Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Run.
- Project Explorer:** Shows a tree view of projects and source files. Projects include a2241330016_practica4, a2241330016_practica5, a2241330016_practica6, a2241330016_tarea05, and a2241330016_tarea06. The a2241330016_tarea06 project is expanded, showing its source packages and files: a2241330016_tarea06 (AreaCirculoEnterito.java, CajeroAutomatico.java, CesarSimple.java, ConvertidorSegundos.java, Estacionamiento.java, FabricaTornillos.java, HashPobre.java, Intercambio.java, Invertir2Cifras.java, PropagacionPaquetes.java, Reloj24Horas.java, RepartoImposible.java, SaltoCaracol.java).
- Code Editor:** Displays the `PropagacionPaquetes` class with its code. The code calculates time based on distance and velocity.
- Output Window:** Shows the run output for the `a2241330016_tarea06` project, displaying the input values (D: 10, V: 3) and the output (Tiempo total: 4), followed by a successful build message.

```

package a2241330016_tarea06;
import java.util.Scanner;
public class PropagacionPaquetes {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Distancia (D): ");
        int D = sc.nextInt();
        System.out.print("Velocidad (V): ");
        int V = sc.nextInt();
        int tiempo = D / V;
        if (D % V > 0) {
            tiempo = tiempo + 1;
        }
        System.out.println("Tiempo total: " + tiempo);
    }
}

```

Dígito Central:

The screenshot shows the PSeInt environment with the following components:

- PSeInt Menu:** Archivo, Editar, Configurar, Ejecutar, Ayuda.
- Toolbar:** Includes icons for file operations, search, and execution.
- Editor:** Displays pseudocode for an algorithm named `Algoritmo DígitoCentral`. The pseudocode defines a variable `N` as an integer, prompts for a number, reads it, and then checks if it's a 3-digit number. If so, it calculates the tens digit and prints it. Otherwise, it prints a message stating it's not a 3-digit number.
- Terminal Window:** Shows the execution of the pseudocode. It starts with the message "*** Ejecución Iniciada. ***", asks for a number (> 123), calculates the tens digit (Digitos centrales: 2), and ends with the message "*** Ejecución Finalizada. ***".

```

Algoritmo DígitoCentral
Definir N, decenas Como Entero
Escribir "Número:"
Leer N
Si N ≥ 100 Y N ≤ 999 Entonces
    decenas ← trunc(N / 10) MOD 10
    Escribir "Dígito central: ", decenas
SiNo
    Escribir "No es un número de 3 cifras."
FinSi
FinAlgoritmo

```

Screenshot of the Eclipse IDE interface showing the code editor and project explorer.

Projects view:

- a2241330016_practica4
- a2241330016_practica5
- a2241330016_practica6
- a2241330016_tarea05
- a2241330016_tarea06
 - Source Packages
 - a2241330016_tarea06
 - AreaCirculoEntero.java
 - CajeroAutomatico.java
 - CesarSimple.java
 - ConvertidorSegundos.java
 - DigitoCentral.java
 - Estacionamiento.java
 - FabricaTornillos.java
 - HashPobre.java
 - Intercambio.java
 - Invertir2Cifras.java
 - PropagacionPaquetes.java
 - Reloj24Horas.java
 - RepartoImposible.java

Code Editor view (DigitoCentral.java):

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class DigitoCentral {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Numero: ");
8         int N = sc.nextInt();
9
10        if (N >= 100 && N <= 999) {
11            int decenas = (N / 10) % 10;
12            System.out.println("Digito central: " + decenas);
13        } else {
14            System.out.println("No es un numero de 3 cifras.");
15        }
16    }
17 }
18 
```

Output view:

```

run:
Numero: 123
Digito central: 2
BUILD SUCCESSFUL (total time: 47 seconds)

```

Eliminar Extremos:

Screenshot of the PSeInt software interface showing the algorithm editor and execution window.

Algorithm Editor (Algoritmo EliminarExtremos):

```

1 Algoritmo EliminarExtremos
2   Definir N, centrales Como Entero
3   Escribir "Numero:"
4   Leer N
5
6   Si N ≥ 1000 Y N ≤ 9999 Entonces
7       centrales ← trunc(N / 10) MOD 100
8       Escribir "Cifras centrales: ", centrales
9   SiNo
10      Escribir "No es de 4 cifras."
11  FinSi
12 FinAlgoritmo
13
14 
```

Execution Window (PSeInt - Ejecutando proceso ELIMINAREXTREMOS):

```

*** Ejecución Iniciada. ***
Numero:
> 1234
Cifras centrales: 23
*** Ejecución Finalizada. ***

```

Screenshot of an IDE (likely Eclipse) showing a Java project named "a2241330016_tarea06". The "Source" tab is selected, displaying the code for the "EliminarExtremos.java" file.

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class EliminarExtremos {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Número: ");
8         int N = sc.nextInt();
9
10        if (N >= 1000 && N <= 9999) {
11            int centrales = (N / 10) % 100;
12            System.out.println("Cifras centrales: " + centrales);
13        } else {
14            System.out.println("No es de 4 cifras.");
15        }
16    }
17 }
18

```

The output window shows the results of running the program:

```

run:
Número: 1234
Cifras centrales: 23
BUILD SUCCESSFUL (total time: 4 seconds)

```

Suma Extremos:

Screenshot of the PSeint environment showing an algorithm named "Algoritmo SumaExtremos".

```

1 Algoritmo SumaExtremos
2   Definir N, centenas, unidades, suma Como Entero
3   Escribir "Número:"
4   Leer N
5
6   Si N > 0 Entonces
7       centenas ← trunc(N / 100)
8       unidades ← N MOD 10
9       suma ← centenas + unidades
10      Escribir "Suma de extremos: ", suma
11  SiNo
12      Escribir "El numero no es positivo."
13  FinSi
14 FinAlgoritmo
15
16

```

The status bar indicates the process is running:

PSeint - Ejecutando proceso SUMAEXTREMOS
*** Ejecución Iniciada. ***
Número:
> 11
Suma de extremos: 1
*** Ejecución Finalizada. ***

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

Projects x

- a2241330016_practica4
- a2241330016_practica5
- a2241330016_practica6
- a2241330016_tarea05
- a2241330016_tarea06
 - Source Packages
 - a2241330016_tarea06
 - AreaCirculoEntero.java
 - CajeroAutomatico.java
 - CesarSimple.java
 - ConvertidorSegundos.java
 - DigitoCentral.java
 - EliminarExtremos.java
 - Estacionamiento.java
 - FabricaTornillos.java
 - HashPobre.java
 - Intercambio.java
 - Invertir2Cifras.java
 - PropagacionPaquetes.java
 - Reloj24Horas.java

Start Page x DigitoCentral.java x EliminarExtremos.java x SumaExtremos.java

Source History

```
1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class SumaExtremos {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Numero: ");
8         int N = sc.nextInt();
9
10        if (N > 0) {
11            int centenas = N / 100;
12            int unidades = N % 10;
13            int suma = centenas + unidades;
14            System.out.println("Suma de extremos: " + suma);
15        } else {
16            System.out.println("El numero no es positivo.");
17        }
18    }
19 }
```

Output - a2241330016_tarea06 (run)

```
run:
Numero: 11
Suma de extremos: 1
BUILD SUCCESSFUL (total time: 4 seconds)
```

Primer Dígito:

The screenshot shows the PSelnt environment. The menu bar includes Archivo, Editar, Configurar, Ejecutar, and Ayuda. The toolbar contains icons for file operations like New, Open, Save, Print, and Help. The left sidebar has buttons for Listado de Variables and Operadores y Funciones. The main window displays a program in pseudocode:

```
1 Algoritmo PrimerDigito
2   Definir N, primer Como Entero
3   Escribir "Numero:"
4   Leer N
5
6   primer ← trunc(N / 100)
7
8   Si primer ≥ 1 Y primer ≤ 9 Entonces
9     Escribir "Primer digito: ", primer
10  SiNo
11    Escribir "No es valido."
12  FinSi
13 FinAlgoritmo
```

The right pane shows the execution results:

PSelnt - Ejecutando proceso PRIMERDIGITO
*** Ejecución Iniciada. ***
Número:
> 123
Primer digito: 1
*** Ejecución Finalizada. ***

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

Screenshot of the Eclipse IDE interface showing the code editor and project explorer.

Projects view:

- a2241330016_practica4
- a2241330016_practica5
- a2241330016_practica6
- a2241330016_tarea05
- a2241330016_tarea06
 - Source Packages
 - a2241330016_tarea06
 - AreaCirculoEntero.java
 - CajeroAutomatico.java
 - CesarSimple.java
 - ConvertidorSegundos.java
 - DigitoCentral.java
 - EliminarExtremos.java
 - Estacionamiento.java
 - FabricaTornillos.java
 - HashPobre.java
 - Intercambio.java
 - Invertir2Cifras.java
 - PrimerDigito.java
 - PropagacionPaquetes.java

Source tab in the code editor:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class PrimerDigito {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Numero: ");
8         int N = sc.nextInt();
9
10        int primer = N / 100;
11
12        if (primer >= 1 && primer <= 9) {
13            System.out.println("Primer digito: " + primer);
14        } else {
15            System.out.println("No es valido.");
16        }
17    }
18 }
```

Output - a2241330016_tarea06 (run)

```

run:
Número: 123
Primer digito: 1
BUILD SUCCESSFUL (total time: 3 seconds)

```

Formador:

Screenshot of the PSeint tool interface showing the algorithm editor and execution window.

Algoritmo Formador

```

1 Algoritmo Formador
2   Definir A, B, C, numero Como Entero
3   Escribir "A:"
4   Leer A
5   Escribir "B:"
6   Leer B
7   Escribir "C:"
8   Leer C
9
10  Si A ≥ 0 Y A ≤ 9 Entonces
11    Si B ≥ 0 Y B ≤ 9 Entonces
12      Si C ≥ 0 Y C ≤ 9 Entonces
13        numero ← A*100 + B*10 + C
14        Escribir "Número formado: ", numero
15        FinSi
16    FinSi
17  FinSi
18 FinAlgoritmo

```

PSeint - Ejecutando proceso FORMADOR

```

*** Ejecución Iniciada. ***
A:
> 1
B:
> 4
C:
> 6
Número formado: 146
*** Ejecución Finalizada. ***

```

No cerrar esta ventana Siempre visible Reiniciar

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a tab for 'a2241330016_tarea06 - Apache NetBeans'. The toolbar has icons for file operations like Open, Save, and Build. The left sidebar displays the 'Projects' view with several Java projects and source packages. The main workspace shows the 'Source' tab for the 'Formador.java' file, which contains the following Java code:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class Formador {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("A: ");
8         int A = sc.nextInt();
9         System.out.print("B: ");
10        int B = sc.nextInt();
11        System.out.print("C: ");
12        int C = sc.nextInt();
13
14        if (A >= 0 && A <= 9)
15            if (B >= 0 && B <= 9)
16                if (C >= 0 && C <= 9) {
17                    int numero = A * 100 + B * 10 + C;
18                    System.out.println("Número formado: " + numero);
19    }

```

The 'Output' window below shows the results of running the program with inputs A: 1, B: 4, and C: 6, resulting in the output 'Número formado: 146' and a 'BUILD SUCCESSFUL' message.

Bloque 6: Logística y Reparto:

Banquete:

The screenshot shows the PSeInt environment. The top menu bar includes Archivo, Editar, Configurar, Ejecutar, Ayuda. The toolbar includes icons for file operations and execution. The left sidebar shows 'Lista de Variables' and 'Operadores y Funciones'. The main workspace has tabs for 'Banquete.psc' and 'FilasCine.psc'. The 'Banquete.psc' tab contains the following pseudocode:

```

1 Algoritmo Banquete
2     Definir M, P, invitados, fuera Como Entero
3     Escribir "Kilos de comida (M):"
4     Leer M
5     Escribir "Platos servidos (P):"
6     Leer P
7
8     invitados ← trunc(M / 2)
9     fuera ← invitados - P
10
11    Si fuera > 0 Entonces
12        Escribir "Invitados fuera: ", fuera
13    SiNo
14        Escribir "Nadie queda fuera."
15    FinSi
16 FinAlgoritmo
17

```

To the right, a terminal window titled 'PSeInt - Ejecutando proceso BANQUETE' shows the execution results:

```

*** Ejecución Iniciada. ***
Kilos de comida (M):
> 11
Platos servidos (P):
> 7
Nadie queda fuera.
*** Ejecución Finalizada. ***

```

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

Selint File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help a2241330016_tar

Projects x Start Page x Formador.java x Banquete.java x

Source History

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class Banquete {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Kilos de comida (M): ");
8         int M = sc.nextInt();
9         System.out.print("Platos servidos (P): ");
10        int P = sc.nextInt();
11
12        int invitados = M / 2;
13        int fuera = invitados - P;
14
15        if (fueras > 0) {
16            System.out.println("Invitados fuera: " + fuera);
17        } else {
18            System.out.println("Nadie queda fuera.");
19

```

Output - a2241330016_tarea06 (run)

run:
 Kilos de comida (M): 11
 Platos servidos (P): 7
 Nadie queda fuera.
 BUILD SUCCESSFUL (total time: 4 seconds)

Filas Cine:

PSelint Archivo Editar Configurar Ejecutar Ayuda

Banquete.psc FilasCine.psc x Refrescos.psc RepartoCartas.psc SaltoGrillo.psc

Algoritmo FilasCine
 Definir N, fila Como Entero
 Escribir "Ticket:"
 Leer N
 Si N > 0 Entonces
 fila ← Trunc((N - 1) / 8) + 1
 Escribir "Fila: ", fila
 SiNo
 Escribir "Ticket invalido."
 FinSi
 FinAlgoritmo

PSelint - Ejecutando proceso FILASCINE
 *** Ejecución Iniciada. ***
 Ticket:
 > 2
 Fila: 1
 *** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class FilasCine {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Ticket: ");
8         int N = sc.nextInt();
9
10        if (N > 0) {
11            int fila = (N - 1) / 8 + 1;
12            System.out.println("Fila: " + fila);
13        } else {
14            System.out.println("Ticket invalido.");
15        }
16    }
17 }
18

```

Output - a2241330016_tarea06 (run)

```

run:
Ticket: 2
Fila: 1
BUILD SUCCESSFUL (total time: 5 seconds)

```

Refrescos:

PSEInt

Archivo Editar Configurar Ejecutar Ayuda

Banquete.psc FilasCine.psc Refrescos.psc RepartoCartas.psc SaltoGrillo.psc

Algoritmo Refrescos
Definir R, packs, sobrante Como Entero
Escribir "Cantidad R:"
Leer R

packs ← trunc(R / 6)
sobrante ← R - (packs * 6)

Si sobrante > 0 Entonces
Escribir "Sobran: ", sobrante
SiNo
Escribir "No sobran refrescos."
FinSi
FinAlgoritmo

Refrescos.psc

PSEInt - Ejecutando proceso REFRESOS
*** Ejecución Iniciada. ***
Cantidad R:
> 15
Sobran: 3
*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar

Screenshot of an IDE (likely NetBeans) showing Java code for a project named "a2241330016_tarea06".

Projects pane:

- a2241330016_practica4
- a2241330016_practica5
- a2241330016_practica6
- a2241330016_tarea05
- a2241330016_tarea06
- Source Packages
 - a2241330016_tarea06
 - AreaCirculoEntero.java
 - Banquete.java
 - CajeroAutomatico.java
 - CesarSimple.java
 - ConvertidorSegundos.java
 - DigitoCentral.java
 - EliminarExtremos.java
 - Estacionamiento.java
 - FabricaTornillos.java
 - FilasCine.java
 - Formador.java
 - HashPobre.java
 - Intercambio.java

Start Page pane:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class Refrescos {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Cantidad R: ");
8         int R = sc.nextInt();
9
10        int packs = R / 6;
11        int sobrante = R - (packs * 6);
12
13        if (sobrante > 0) {
14            System.out.println("Sobran: " + sobrante);
15        } else {
16            System.out.println("No sobran refrescos.");
17        }
18    }
19 }
```

Output pane:

```

run:
Cantidad R: 15
Sobran: 3
BUILD SUCCESSFUL (total time: 3 seconds)
```

Reparto Cartas:

Screenshot of the PSeint environment showing a pseudocode script for "Reparto Cartas".

PSeint window:

File Edit Configurar Ejecutar Ayuda

Buttons: Run, Stop, Save, Open, Print, Help.

Script tabs: Banquete.psc, FilasCine.psc, Refrescos.psc, RepartoCartas.psc*, SaltoGrillo.psc.

RepartoCartas.psc content:

```

1 Algoritmo RepartoCartas
2   Definir J, cartas Como Entero
3   Escribir "Jugadores:"
4   Leer J
5
6   Si J > 0 Entonces
7     cartas ← trunc(52 / J)
8     Escribir "Cartas por jugador: ", cartas
9   SiNo
10    Escribir "Número invalido de jugadores."
11  FinSi
12 FinAlgoritmo
13
14
```

Output window:

```

PSeint - Ejecutando proceso REPARTOCARTAS
*** Ejecución Iniciada. ***
Jugadores:
> 6
Cartas por jugador: 8
*** Ejecución Finalizada. ***
```

Checkboxes at the bottom: No cerrar esta ventana, Siempre visible, Reiniciar.

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer:** Projects x, showing several projects like a2241330016_practica4, a2241330016_practica5, a2241330016_practica6, a2241330016_tarea05, and a2241330016_tarea06. The a2241330016_tarea06 project is expanded, showing its source packages and files.
- Source Editor:** The current file is RepartoCartas.java. The code is as follows:


```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class RepartoCartas {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Jugadores: ");
8         int J = sc.nextInt();
9
10        if (J > 0) {
11            int cartas = 52 / J;
12            System.out.println("Cartas por jugador: " + cartas);
13        } else {
14            System.out.println("Número invalido de jugadores.");
15        }
16    }
17 }
18 
```
- Output Window:** Shows the run results:


```

run:
Jugadores: 6
Cartas por jugador: 8
BUILD SUCCESSFUL (total time: 2 seconds)
      
```

Salto Grillo:

The screenshot shows the PSelnt environment with the following details:

- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, Undo, Redo, Run, Stop, and Help.
- File Tabs:** Banquete.psc, FilasCine.psc, Refrescos.psc, RepartoCartas.psc, SaltoGrillo.psc.
- Pseudocode Editor:** The SaltoGrillo.psc tab is active, displaying the following pseudocode:


```

1 Algoritmo SaltoGrillo
2   Definir D, K, saltos Como Entero
3   Escribir "Distancia D:"
4   Leer D
5   Escribir "Salto K:"
6   Leer K
7
8   Si (D % K) = 0 Entonces
9     saltos ← D / K
10    Escribir "Saltos: ", saltos
11  SiNo
12    Escribir "D no es multiplo de K."
13  FinSi
14 FinAlgoritmo
15
16 
```
- Output Window:** Shows the execution results:


```

PSelnt - Ejecutando proceso SALTOGRILLO
*** Ejecución Iniciada. ***
Distancia D:
> 15
Salto K:
> 5
Saltos: 3
*** Ejecución Finalizada. ***
      
```
- Status Bar:** Includes checkboxes for "No cerrar esta ventana" and "Siempre visible", and a "Reiniciar" button.

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a user ID a2241330016. Below the menu is a toolbar with various icons. The left sidebar is titled 'Projects' and lists several Java projects: a2241330016_practica4, a2241330016_practica5, a2241330016_practica6, a2241330016_tarea05, and a2241330016_tarea06. The 'a2241330016_tarea06' project is expanded, showing its 'Source Packages' folder which contains a subfolder 'a2241330016_tarea06' holding ten Java files: AreaCirculoEntero.java, Banquete.java, CajeroAutomatico.java, CesarSimple.java, ConvertidorSegundos.java, DígitoCentral.java, EliminarExtremos.java, Estacionamiento.java, FabricaTornillos.java, FilasCine.java, Formador.java, HashPobre.java, and Intercambio.java. The main code editor window displays the following Java code:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class SaltoGrillo {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Distancia D: ");
8         int D = sc.nextInt();
9         System.out.print("Salto K: ");
10        int K = sc.nextInt();
11
12        if (D % K == 0) {
13            int saltos = D / K;
14            System.out.println("Saltos: " + saltos);
15        } else {
16            System.out.println("D no es multiplo de K.");
17        }
18    }
}

```

The output window below shows the results of running the program:

```

run:
Distancia D: 15
Salto K: 5
Saltos: 3
BUILD SUCCESSFUL (total time: 3 seconds)

```

Bloque 7: Tiempo y Conversiones

Días Semanas:

The screenshot shows the PSeint IDE interface. The top menu bar includes Archivo, Editar, Configurar, Ejecutar, Ayuda. Below the menu is a toolbar with various icons. The left sidebar has sections for 'Listado de Variables', 'Operadores y Funciones', and 'Algoritmo'. The main code editor window titled 'DiasSemanas.psc' contains the following pseudocode:

```

1 Algoritmo DiasSemanas
2   Definir X, semanas, resto Como Entero
3   Escribir "Cantidad de dias:"
4   Leer X
5
6   semanas ← Trunc(X / 7)
7   resto ← X % 7
8
9   Si semanas = 1 Entonces
10      Escribir semanas, " semana y ", resto, " dias"
11   SiNo
12      Escribir semanas, " semanas y ", resto, " dias"
13   FinSi
14 FinAlgoritmo
15

```

The right side of the interface shows a terminal window titled 'PSeint - Ejecutando proceso DIASSEMANAS' with the following output:

```

*** Ejecución Iniciada. ***
Cantidad de dias:
> 7
1 semana y 0 dias
*** Ejecución Finalizada. ***

```

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

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

Projects x Start Page x SaltoGrillo.java x DiasSemanas.java

Source History |

```

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
20 }
```

Output - a2241330016_tarea06 (run)

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

Horario Laboral:

PSelnt

Archivo Editar Configurar Ejecutar Ayuda

DiasSemanas.psc HorarioLaboral.psc x

Algoritmo HorarioLaboral
 Definir S, horas Como Entero
 Escribir "Segundos trabajados:"
 Leer S
 horas ← Trunc(S / 3600)
 Si horas ≥ 24 Entonces
 horas ← horas % 24
 FinSi
 Escribir "Horas completas: ", horas
FinAlgoritmo

PSelnt - Ejecutando proceso HORARIOLABORAL

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

No cerrar esta ventana Siempre visible Reiniciar

The screenshot shows an IDE interface with the following details:

- Projects View:** Shows several projects under "a2241330016_tarea06", including "Source Packages" which contains files like AreaCirculoEntero.java, Banquete.java, CajeroAutomatico.java, CesarSimple.java, ConvertidorSegundos.java, DiasSemanas.java, DígitoCentral.java, EliminarExtremos.java, Estacionamiento.java, FabricaTornillos.java, FilasCine.java, Formador.java, and HashPobre.java.
- Code Editor:** Displays the HorarioLaboral.java file with the following code:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class HorarioLaboral {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Segundos trabajados: ");
8         int S = sc.nextInt();
9
10        int horas = S / 3600;
11
12        if (horas >= 24) {
13            horas = horas % 24;
14        }
15
16        System.out.println("Horas completas: " + horas);
17    }
18}

```
- Output View:** Shows the execution results:

```

run:
Segundos trabajados: 15000
Horas completas: 4
BUILD SUCCESSFUL (total time: 4 seconds)

```

Reloj Arena:

The screenshot shows the PSeInt environment with the following components:

- PSeInt Menu:** Archivo, Editar, Configurar, Ejecutar, Ayuda.
- Toolbar:** Includes icons for opening files, saving, running, and executing processes.
- Project Manager:** Shows files: DiasSemanas.psc, HorarioLaboral.psc, RelojArena.psc.
- Code Editor:** Displays an algorithm in pseudocode:

```

1 Algoritmo RelojArena
2   Definir a, b, diferencia Como Entero
3
4   a ← 7
5   b ← 4
6
7   diferencia ← a - b
8
9   Si diferencia = 3 Entonces
10      Escribir "Proceso exitoso: ", diferencia, " minutos."
11   SiNo
12      Escribir "Resultado incorrecto."
13   FinSi
14 FinAlgoritmo
15
16

```
- Output Window:** Shows the execution log:

```

PSeInt - Ejecutando proceso RELOJARENA
*** Ejecución Iniciada ***
Proceso exitoso: 3 minutos.
*** Ejecución Finalizada ***

```
- Status Bar:** Includes checkboxes for "No cerrar esta ventana" and "Siempre visible", and a "Reiniciar" button.

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a status bar indicating memory usage (291.6/325.0MB) and several icons. The Projects tab on the left lists several Java projects under 'a2241330016_practica4' and 'a2241330016_tarea06'. The Source tab in the center displays the code for RelojArena.java:

```
1 package a2241330016_tarea06;
2 public class RelojArena {
3     public static void main(String[] args) {
4         int a = 7;
5         int b = 4;
6
7         int diferencia = a - b;
8
9         if (diferencia == 3) {
10             System.out.println("Proceso exitoso: " + diferencia + " minutos.");
11         } else {
12             System.out.println("Resultado incorrecto.");
13         }
14     }
15 }
16
17 }
```

The Output tab at the bottom shows the run results:

```
run:
Proceso exitoso: 3 minutos.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Años Bisiestos:

The screenshot shows the PSeint environment. The top menu bar includes Archivo, Editar, Configurar, Ejecutar, and Ayuda. The toolbar contains various icons for file operations. The central workspace shows pseudocode for 'Bisiestos_':

```
1 Algoritmo Bisiestos_
2   Definir i, bisiestos Como Entero
3   Escribir "Año final:"
4   Leer i
5
6   bisiestos ← Trunc(i / 4)
7
8   Si i = 0 Entonces
9     bisiestos ← 0
10  FinSi
11
12  Escribir "Años bisiestos: ", bisiestos
13 FinAlgoritmo
14
```

To the right, a terminal window titled 'PSeint - Ejecutando proceso BISIESTOS_' shows the execution results:

```
*** Ejecución Iniciada. ***
Año final:
> 2026
Años bisiestos: 506
*** Ejecución Finalizada. ***
```

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

The screenshot shows an IDE interface with the following details:

- Projects:** A tree view showing several projects: a2241330016_practica4, a2241330016_practica5, a2241330016_practica6, a2241330016_tarea05, and a2241330016_tarea06. The a2241330016_tarea06 project is expanded, showing its source packages and files.
- Source Editor:** The code for `Bisiestos_.java` is displayed. The code calculates leap years based on the input year `Y`.
- Output:** The output window shows the results of running the program with the year 2026, outputting 506 leap years.

```

package a2241330016_tarea06;
import java.util.Scanner;
public class Bisiestos_ {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Anio final: ");
        int Y = sc.nextInt();

        int bisiestos = Y / 4;

        if (Y == 0) {
            bisiestos = 0;
        }

        System.out.println("Anios bisiestos: " + bisiestos);
    }
}
  
```

Bloque 8: Geometría y Física

Perímetro Polígono:

The screenshot shows the PSelint environment with the following components:

- Toolbar:** Standard file operations like Open, Save, Print, etc.
- Menu Bar:** Archivo, Editar, Configurar, Ejecutar, Ayuda.
- Left Sidebar:** Includes a help icon, a list of variables, and an operators and functions section.
- Code Editor:** Displays pseudocode for calculating the perimeter of a polygon. It defines a variable `L` as the side length, checks if it's greater than zero, and calculates the perimeter as the sum of all sides.
- Output Window:** Shows the execution of the process `PERIMETROPOLIGONO`. It prompts for the side length, receives the value 5, calculates the perimeter as 30, and concludes the execution.

```

Algoritmo PerimetroPoligono
  Definir L, perimetro Como Real
  Escribir "Lado L:"
  Leer L
  Si L > 0 Entonces
    perimetro ← L + L + L + L + L + L
    Escribir "Perimetro: ", perimetro
  SiNo
    Escribir "El lado debe ser mayor que cero."
  FinSi
FinAlgoritmo
  
```

The screenshot shows an IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer:** Shows multiple projects under "a2241330016_tarea06".
- Source Editor:** The active file is `PerimetroPoligono.java`. The code is as follows:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class PerimetroPoligono {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Lado L: ");
8         double L = sc.nextDouble();
9
10        if (L > 0) {
11            double perimetro = L + L + L + L + L + L;
12            System.out.println("Perimetro: " + perimetro);
13        } else {
14            System.out.println("El lado debe ser mayor que cero.");
15        }
16    }
17 }

```

- Output Window:** Shows the run output for `PerimetroPoligono.java`.

Promedio Ponderado:

The screenshot shows the PSeInt environment with the following details:

- Toolbar:** Archivo, Editar, Configurar, Ejecutar, Ayuda.
- Pseudocode Editor:** The pseudocode is as follows:

```

1 Algoritmo PromedioPonderado
2   Definir N1, N2, N3, promedio Como Real
3   Escribir "Nota 1:"
4   Leer N1
5   Escribir "Nota 2:"
6   Leer N2
7   Escribir "Nota 3:"
8   Leer N3
9
10  promedio ← N1*0.2 + N2*0.3 + N3*0.5
11
12  Si promedio ≥ 6 Entonces
13    Escribir "Promedio satisfactorio: ", promedio
14  SiNo
15    Escribir "Promedio no satisfactorio: ", promedio
16  FinSi
17 FinAlgoritmo

```

- Execution Window:** Shows the output of the pseudocode execution.

The screenshot shows the NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a user ID a2241330016_tarea06 - Apache. The toolbar contains icons for file operations like New, Open, Save, and Build. The left sidebar displays the 'Projects' tree, which lists several Java files under 'a2241330016_tarea06'. The main workspace shows two tabs open: 'PerimetroPoligono.java' and 'PromedioPonderado.java'. The 'PromedioPonderado.java' tab is active, displaying the following Java code:

```
package a2241330016_tarea06;
import java.util.Scanner;
public class PromedioPonderado {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Nota 1: ");
        double N1 = sc.nextDouble();
        System.out.print("Nota 2: ");
        double N2 = sc.nextDouble();
        System.out.print("Nota 3: ");
        double N3 = sc.nextDouble();

        double promedio = N1*0.2 + N2*0.3 + N3*0.5;

        if (promedio >= 6) {
            System.out.println("Promedio satisfactorio: " + promedio);
        } else {
            System.out.println("Promedio no satisfactorio: " + promedio);
        }
    }
}
```

The bottom output window shows the run results:

```
run:
Nota 1: 6
Nota 2: 6
Nota 3: 6
Promedio satisfactorio: 6.0
BUILD SUCCESSFUL (total time: 5 seconds)
```

Costo Viaje:

The screenshot shows the PSelnt application window. The menu bar includes Archivo, Editar, Configurar, Ejecutar, and Ayuda. Below the menu is a toolbar with various icons for file operations like Open, Save, Print, and Cut/Paste. The main area has tabs for PerímetroPolígono.psc, PromedioPonderado.psc, and CostoViaje.psc. The current tab, CostoViaje.psc, contains the following pseudocode:

```
1 Algoritmo CostoViaje
2     Definir K, litros, costo Como Real
3     Escribir "Distancia K:"
4     Leer K
5
6     litros ← K / 12
7     costo ← litros * 20
8
9     Si K > 500 Entonces
10        costo ← costo + 100
11    FinSi
12
13    Escribir "Costo total: ", costo
14 FinAlgoritmo
```

To the left of the code editor is a sidebar with buttons for Variables, Operators, and Functions. The terminal window on the right shows the execution of the process COSTOVIAJE, starting with "Ejecución Iniciada.", followed by the input "Distancia K:", the output "> 60", and the final output "Costo total: 100". It concludes with "Ejecución Finalizada.".

Screenshot of an IDE (likely NetBeans) showing Java code for calculating travel costs.

Projects pane:

- a2241330016_practica4
- a2241330016_practica5
- a2241330016_practica6
- a2241330016_tarea05
- a2241330016_tarea06
 - Source Packages
 - a2241330016_tarea06
 - AreaCirculoEntero.java
 - Banquete.java
 - Bisietos_.java
 - CajeroAutomatico.java
 - CesarSimple.java
 - ConvertidorSegundos.java
 - CostoViaje.java
 - DiasSemanas.java
 - DigitoCentral.java
 - EliminarExtremos.java
 - Estacionamiento.java
 - FabricaTornillos.java
 - FilasCine.java

Source tab:

```

1 package a2241330016_tarea06;
2 import java.util.Scanner;
3 public class CostoViaje {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         System.out.print("Distancia K: ");
8         double K = sc.nextDouble();
9
10        double litros = K / 12;
11        double costo = litros * 20;
12
13        if (K > 500) {
14            costo += 100;
15        }
16
17        System.out.println("Costo total: " + costo);
18    }
19

```

Output - a2241330016_tarea06 (run)

```

run:
Distancia K: 60
Costo total: 100.0
BUILD SUCCESSFUL (total time: 6 seconds)

```

Escala Mapa:

Screenshot of PSelnt (a Pseudocode IDE) showing pseudocode for calculating map scale.

Archivos pane:

- PerimetroPoligono.psc
- PromedioPonderado.psc
- CostoViaje.psc
- EscalaMapa.psc

EscalaMapa.psc content:

```

1 Algoritmo EscalaMapa
2   Definir C, metros, km Como Real
3   Escribir "Centimetros en el mapa:"
4   Leer C
5
6   metros ← C * 500
7   km ← metros / 1000
8
9   Si km < 1 Entonces
10     Escribir "Distancia real: ", metros, " metros"
11   SiNo
12     Escribir "Distancia real: ", km, " km"
13   FinSi
14 FinAlgoritmo
15
16

```

Ejecución pane:

PSelnt - Ejecutando proceso ESCALAMAPA
*** Ejecución Iniciada. ***
Centimetros en el mapa:
> 150
Distancia real: 75 km
*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a tab for the current project: a2241330016_tarea06 - Apache Net.

The Projects view on the left lists several projects and source packages. Under 'a2241330016_tarea06' is a package named 'a2241330016_tarea06' which contains files like AreaCirculoEntero.java, Banquete.java, Bisieros.java, CajeroAutomatico.java, CesarSimple.java, ConvertidorSegundos.java, CostoViaje.java, DiasSemanas.java, DígitoCentral.java, EliminarExtremos.java, EscalaMapa.java (which is selected), Estacionamiento.java, and FabricaTornillos.java.

The Source tab in the center editor shows the code for 'EscalaMapa.java'. The code reads a centimeter value from standard input, converts it to meters (multiplied by 500), and then converts it to kilometers (divided by 1000). It then prints the original centimeters and the calculated kilometers.

```
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("Centímetros 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    }
}
```

The Output view at the bottom shows the run results:

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