

ALUMNO
GERARDO RAMOS CARACAS

DOCENTE
ROGELIO ENRIQUE TELONA TORRES

GRUPO
110A

MATERIA
FUNDAMENTOS DE PROGRAMACIÓN

**FUNDAMENTOS DE
PROGRAMACIÓN
ORIENTADA A OBJETOS**

INTRODUCCIÓN

La Programación Orientada a Objetos (POO) es un paradigma de programación que se basa en el uso de “objetos” para diseñar aplicaciones y programas. Estos objetos son instancias de “clases”, que definen las propiedades y comportamientos de los objetos.

Desarrollaremos a través de esta investigación la adquisición de conocimientos básicos sobre conceptos fundamentales de la Programación Orientada a Objetos desde su Evolución, Conceptos fundamentales, Lenguajes orientados a objetos, Relaciones entre clases y objetos, Papel de clases y objetos en el análisis y el diseño, así como los Entornos de programación.

1.1 Evolución de la programación

Para hablar del origen de la programación nos tenemos que remontar hasta 1801, año en el que Joseph Marie Jacquard inventó el primer sistema de programación: un programa introducido en un telar que leía el código y permitía tejer unas figuras determinadas.

En 1833, treinta años después, Charles Babbage crea la conocida como máquina diferencial, capaz de resolver no solo sumas y restas, sino también ecuaciones.

Fue en 1957 cuando el conocido científico John W. Backus creó Fortran, el primer lenguaje de programación conocido que tenía como fin realizar cálculos numéricos y científicos de manera eficiente.

Más tarde, en 1964, apareció BASIC, una herramienta de apoyo enfocada a la educación que continúa utilizándose hoy en día en algunas herramientas.

En la década de 1970 surgieron varios lenguajes de programación importantes que tuvieron un impacto significativo en el desarrollo de software:

- Pascal: un lenguaje de programación que surgió para enseñar a los alumnos y que se sigue utilizando hoy en día.
- Lenguaje C: inventado por Dennis Ritchie en 1972. Se sigue utilizando hoy en día.
- Lenguaje C++: aparece en 1979 con el fin de ampliar el lenguaje C.
- HTML y JavaScript (1990s): Con la llegada de Internet, lenguajes como HTML y JavaScript se volvieron esenciales para el desarrollo web3.
- Python (1991): Creado por Guido van Rossum, Python es conocido por su simplicidad y legibilidad, y se utiliza en una amplia variedad de aplicaciones, desde desarrollo web hasta inteligencia artificial3.

1.2. Conceptos fundamentales de la programación orientada a objetos

- **Clase:** Es una plantilla o modelo que define las propiedades (atributos) y comportamientos (métodos) que los objetos creados a partir de ella tendrán. Piensa en una clase como un plano para construir objetos.
- **Objeto:** Es una instancia de una clase. Cada objeto tiene su propio estado y comportamiento, definido por su clase.
- **Encapsulación:** Este principio agrupa datos y métodos que manipulan esos datos dentro de una unidad llamada clase. La encapsulación ayuda a ocultar los detalles internos de un objeto y solo expone lo necesario, protegiendo así los datos de modificaciones no autorizadas.
- **Herencia:** Permite a una clase heredar propiedades y métodos de otra clase, promoviendo la reutilización del código. En Java, por ejemplo, se utiliza la palabra clave `extends` para crear una clase que hereda de otra.
- **Polimorfismo:** Este principio permite que los objetos tomen muchas formas. En la práctica, significa que una sola interfaz puede representar diferentes tipos de objetos. El polimorfismo permite realizar una sola acción de diferentes maneras.
- **Abstracción:** Consiste en ocultar los detalles complejos y mostrar solo lo esencial. La abstracción ayuda a reducir la complejidad y a aumentar la eficiencia del programa.

1.3. Lenguajes orientados a objetos

1. **Java:** Uno de los lenguajes más utilizados en el mundo, especialmente en aplicaciones empresariales y desarrollo de Android.
2. **C++:** Una extensión del lenguaje C que incluye características de POO. Es ampliamente utilizado en sistemas operativos, videojuegos y aplicaciones de alto rendimiento.
3. **Python:** Conocido por su simplicidad y legibilidad, Python es muy popular en desarrollo web, ciencia de datos, inteligencia artificial y más.
4. **C:** Desarrollado por Microsoft, es utilizado principalmente en aplicaciones de Windows y desarrollo de videojuegos con Unity.
5. **Ruby:** Conocido por su elegancia y simplicidad, es popular en el desarrollo web, especialmente con el framework Ruby on Rails.
6. **PHP:** Utilizado principalmente para el desarrollo web, PHP soporta POO y es muy común en aplicaciones web dinámicas.
7. **Objective-C:** Utilizado principalmente para el desarrollo de aplicaciones en macOS y iOS antes de la introducción de Swift.
8. **Swift:** El lenguaje moderno de Apple para el desarrollo de aplicaciones en iOS y macOS, diseñado para ser seguro y eficiente.
9. **JavaScript:** Aunque originalmente no era un lenguaje orientado a objetos, ha evolucionado para soportar POO, especialmente con la introducción de ES6.
10. **Perl:** Un lenguaje de propósito general que soporta POO y es conocido por su capacidad de procesamiento de texto.
11. **Smalltalk:** Uno de los primeros lenguajes de programación orientados a objetos, influyó en muchos lenguajes modernos.
12. **Visual Basic .NET:** Utilizado principalmente para aplicaciones de Windows, es parte del framework .NET de Microsoft.

1.4. Relaciones entre clases y objetos

La resolución de un problema exige la colaboración de objetos.

1. Esto exige que los agentes se conozcan
2. El conocimiento entre objetos se realiza mediante el establecimiento de relaciones.
3. Las relaciones se pueden establecer entre clases o entre objetos.

Además, a nivel de objetos, podemos encontrar dos tipos de relaciones:

- **Persistentes:** recogen caminos de comunicación entre objetos que se almacenan de algún modo y que por tanto pueden ser reutilizados en cualquier momento.
- **No persistentes:** recogen caminos de comunicación entre objetos que desaparecen tras ser utilizados.

Existen tres tipos básicos de relaciones entre clases:

1. Generalización/Especialización

Denota una relación «es un» (is a). Por ejemplo, una rosa es un tipo de flor, lo que quiere decir que una rosa es una subclase especializada de una clase más general, la de las flores.

2. Asociación

Denota alguna dependencia semántica entre clases de otro modo independientes, como entre las mariposas y las flores. Un ejemplo más: las rosas y las velas son clases claramente independientes, pero ambas representan cosas que podrían utilizarse para decorar la mesa de una cena.

3. Relación entre objetos/Relaciones entre clases

Relación todo/parte (whole/part)

Denota una relación «parte de» (parí of). Así, un pétalo no es un tipo de flor; es una parte de una flor.

Las relaciones permiten a un objeto relacionarse con los demás que forman parte de la misma organización.

Hay 3 tipos fundamentales de relaciones:

- Relaciones jerárquicas, de herencia o de des generalización:
- Relaciones de pertenencia o de agregación
- Relaciones de asociación

1.5. Papel de clases y objetos en el análisis y el diseño

El Diseño Orientado a Objetos (DO) difiere considerablemente del diseño estructurado ya que en Diseño Orientado a Objetos (DOO) no se realiza un problema en términos de tareas (subrutinas) ni en términos de datos, sino se analiza el problema como un sistema de objetos que interactúan entre sí.

Un problema desarrollado con técnicas orientadas a objetos requiere, en primer lugar, saber cuáles son los objetos del programa. Como tales objetos son instancias de clases, la primera etapa en el desarrollo orientado a objetos requiere de la identificación de dichas clases (atributos y comportamiento), así como las relaciones entre estas y su posterior implementación en un lenguaje de programación.

Existen numerosos métodos de diseño orientado a objetos, pero en general como ocurre en cualquier proyecto estructurado, un proyecto software Orientado a Objetos se compone de las siguientes etapas:

- Análisis Orientado a Objetos (AOO).
- Diseño Orientado a Objetos (DOO).
- Programación Orientada a Objetos (POO)

1.6. Entornos de programación

Los entornos de programación, también conocidos como Entornos de Desarrollo Integrados (IDEs, por sus siglas en inglés), son herramientas de software diseñadas para proporcionar a los desarrolladores un entorno de trabajo completo y unificado para la creación, edición, depuración y gestión de software. Aquí te dejo una descripción de sus componentes y funciones principales:

Componentes de un IDE

- **Editor de Código:**

Es el componente principal donde los desarrolladores escriben el código fuente. Los editores de código suelen incluir características como resaltado de sintaxis, autocompletado y numeración de líneas para facilitar la escritura y lectura del código.

- **Compilador/Intérprete:**

Convierte el código fuente escrito en un lenguaje de programación a un lenguaje máquina que la computadora puede entender. Los compiladores traducen todo el código de una vez, mientras que los intérpretes lo hacen línea por línea.

- **Depurador:**

Herramienta que ayuda a los desarrolladores a encontrar y corregir errores en el código. Permite ejecutar el programa paso a paso, inspeccionar variables y ver el flujo de ejecución.

- **Enlazador:**

Combina varios módulos de código objeto en un solo ejecutable o biblioteca. Es esencial para programas grandes que se dividen en múltiples archivos.

- **Gestor de Proyectos:**

Facilita la organización y gestión de los archivos y recursos del proyecto. Permite a los desarrolladores manejar múltiples archivos y directorios de manera eficiente.

- **Visual Studio:**

Desarrollado por Microsoft, es uno de los IDEs más completos y utilizados, especialmente para desarrollo en .NET y C++1.

Los entornos de programación son esenciales para los desarrolladores, ya que proporcionan todas las herramientas necesarias en un solo lugar, mejorando la eficiencia y productividad.

REFERENCIAS

- [1] P. Gómez, “Historia de la programación: ¿qué es y cómo ha evolucionado con los años?”, DevCamp, 08-jun-2023. [En línea]. Disponible en: <https://devcamp.es/historia-de-la-programacion-que-es-y-como-ha-evolucionado-con-los-anos/>. [Consultado: 22-sep-2024].
- [2] G. Marker, “Lenguajes de programación orientados a objetos”, Tecnología + Informática, 07-nov-2022. [En línea]. Disponible en: <https://www.tecnologia-informatica.com/lenguajes-de-programacion-orientada-a-objetos/>. [Consultado: 22-sep-2024].
- [3] “CONCEPTOS BÁSICOS de la programación orientada a objetos”, Rua.ua.es. [En línea]. Disponible en: <https://rua.ua.es/dspace/bitstream/10045/18799/1/UD2-ConceptosBasicos.pdf>. [Consultado: 22-sep-2024].
- [4] “Papel de clases y objetos en el análisis y el diseño”, Marcoteorico.com. [En línea]. Disponible en: <https://www.marcoteorico.com/curso/51/fundamentos-de-programacion/393/papel-de-clases-y-objetos-en-el-analisis-y-el-diseno>. [Consultado: 22-sep-2024].



CONCLUSIÓN

El desarrollo de Software con base en la Programación Orientada a Objetos ha sido un salto tecnológico en cuestión de flexibilidad, mantenimiento y optimización de código, se utiliza en tantas áreas que parece fundamental para la labor cotidiana, desde sus notables interfaces de desarrollo hasta pasar por su desarrollo de Software empresarial, su aplicación en desarrollo web o móvil e incluso motores gráficos para videojuegos.

Curso: [Fundamentos de Programación](#)

Tarea: [Investigación Unidad 1](#) 

Valor 30

[Ver todos los envíos](#)



Gerardo Ramos Caracas

241u0354@alumno.itssat.edu.mx

Fecha de entrega: 15 de septiembre ...



Cambiar usuario



2 de 4 [Reiniciar preferencias de tabla](#)



Página 1 de 11



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 6 días después de la fecha límite

Los estudiantes pueden editar este envío

 [Investigación Programación Orientada a Objetos.pdf](#) 21 de septiembre de 2024, 22:40

▶ [Comentarios \(0\)](#)

Calificación

Calificación:

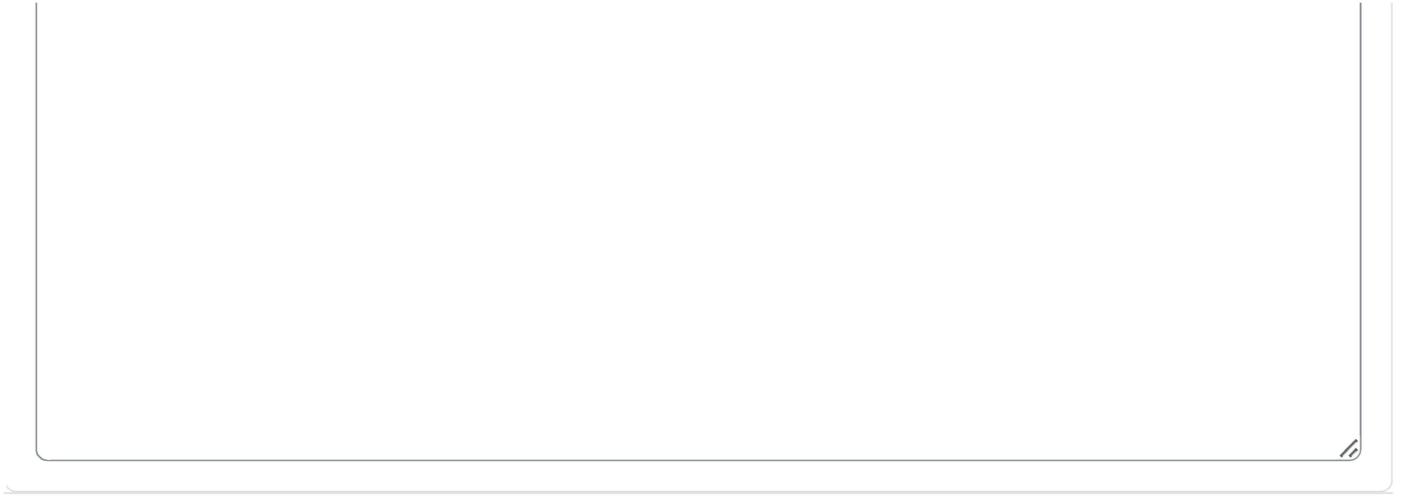
Hoja de presentación	No contien todos los datos 0 puntos	Datos incompletos 1 puntos	Completo 2 puntos	
Introducción	No contiene 0 puntos	Muy pequeña 2.5 puntos	Completa 5 puntos	
Contenido	No cubre los temas 0 puntos	La mitad de los temas 6 puntos	Completo 13 puntos	
Referencias IEEE	No contiene 0 puntos	Una o no tiene el formato 2 puntos	Más de una y formato correcto 4 puntos	
Conclusión	No contien 0 puntos	Muy pequeña 2.5 puntos	Completa 5 puntos	
Archivo PDF	Sin formato 0 puntos	Correcto 1 puntos		

Calificación actual en el libro

30.00

Comentarios de retroalimentación

↓
A ▾
B
I
☰
☷
☰
☷
🔗
🔄
🖼️
📄
🎤
📹
📄
H&P
🕒
📱



Notificar a estudiantes [?](#)

GUARDAR CAMBIOS

GUARDAR Y MOSTRAR SIGUIENTE

REINICIAR

ALUMNO
GERARDO RAMOS CARACAS

DOCENTE
ROGELIO ENRIQUE TELONA TORRES

GRUPO
110A

MATERIA
FUNDAMENTOS DE PROGRAMACIÓN

EVIDENCIA DE PRÁCTICAS
UNIDAD 1

ÍNDICE

Prácticas

Ejercicio 1. Cómo Imprimir	3
Ejercicio 2. Imprimir una Pirámide	3
Ejercicio 3. Asignación de Variables.....	4
Ejercicio 4. Suma de Variables.....	4
Ejercicio 5. Operaciones Varias.....	5
Ejercicio 6. Perímetro y Área de un Triángulo	5
Ejercicio 7. Calcular con Segundos	6
Ejercicio 8. Calcular Salario Final	7
Ejercicio 9. Arreglos con Nombres	8

Ejercicio 1. Cómo Imprimir

The screenshot shows the PSeInt IDE with a program named 'aprender_imprimir'. The code in the editor is as follows:

```

1 Algoritmo aprender_imprimir
2
3     // Gerardo Ramos Caracas
4     // 110A
5
6     Escribir "Gerardo Ramos Caracas";
7     Escribir "Ingeniería Informática";
8     Escribir "05/09/2024";
9
10 FinAlgoritmo
11

```

An execution window titled 'PSeInt - Ejecutando proceso APRENDER_IMPRIMIR' is open, displaying the output:

```

*** Ejecución Iniciada. ***
Gerardo Ramos Caracas
Ingeniería Informática
05/09/2024
*** Ejecución Finalizada. ***

```

Below the execution window, there are checkboxes for 'No cerrar esta ventana' and 'Siempre visible', and a 'Reiniciar' button.

La ejecución ha finalizado sin errores.

Ejercicio 2. Imprimir una Pirámide

The screenshot shows the PSeInt IDE with a program named 'piramide'. The code in the editor is as follows:

```

1 Proceso piramide
2
3     // Gerardo Ramos Caracas
4     // 110A
5
6     Escribir " *";
7     Escribir " * *";
8     Escribir " * * *";
9     Escribir " * * * *";
10    Escribir " * * * * *";
11
12 FinProceso
13

```

An execution window titled 'PSeInt - Ejecutando proceso PIRAMIDE' is open, displaying the output:

```

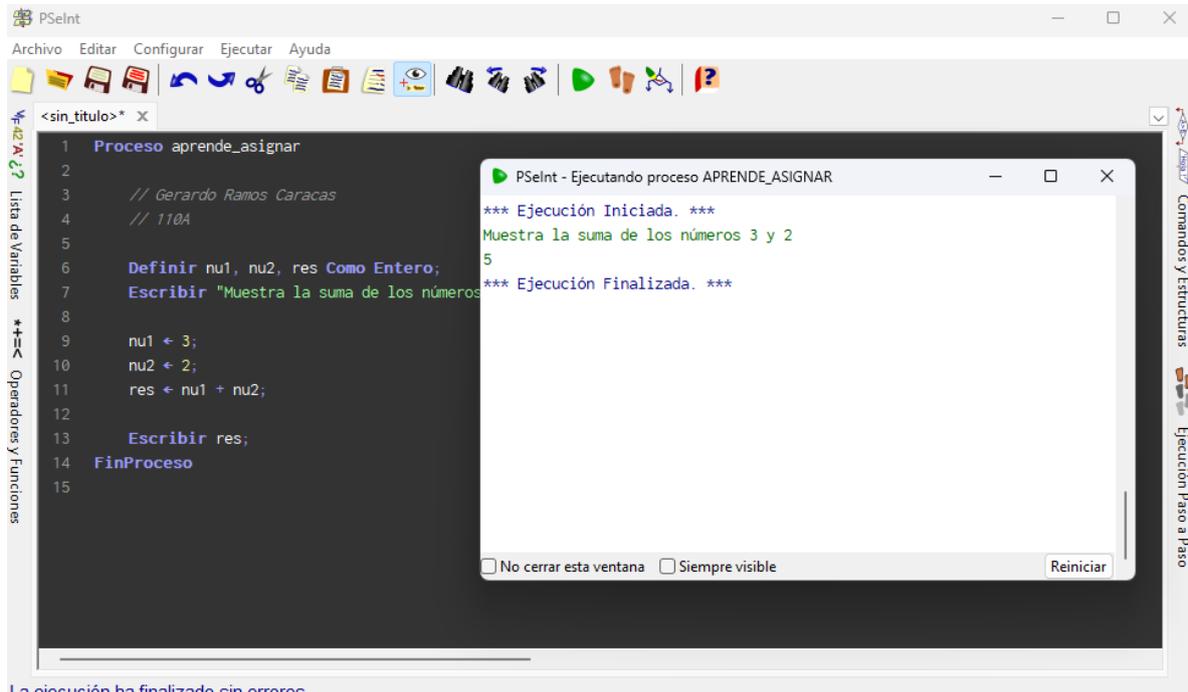
*** Ejecución Iniciada. ***
 *
 * *
 * * *
 * * * *
 * * * * *
*** Ejecución Finalizada. ***

```

Below the execution window, there are checkboxes for 'No cerrar esta ventana' and 'Siempre visible', and a 'Reiniciar' button.

La ejecución ha finalizado sin errores.

Ejercicio 3. Asignación de Variables



The screenshot shows the PSeInt IDE with a program named 'aprende_asignar' being executed. The code in the editor is as follows:

```

1 Proceso aprende_asignar
2
3 // Gerardo Ramos Caracas
4 // 110A
5
6 Definir nu1, nu2, res Como Entero;
7 Escribir "Muestra la suma de los números";
8
9 nu1 ← 3;
10 nu2 ← 2;
11 res ← nu1 + nu2;
12
13 Escribir res;
14 FinProceso
15
    
```

The execution window shows the following output:

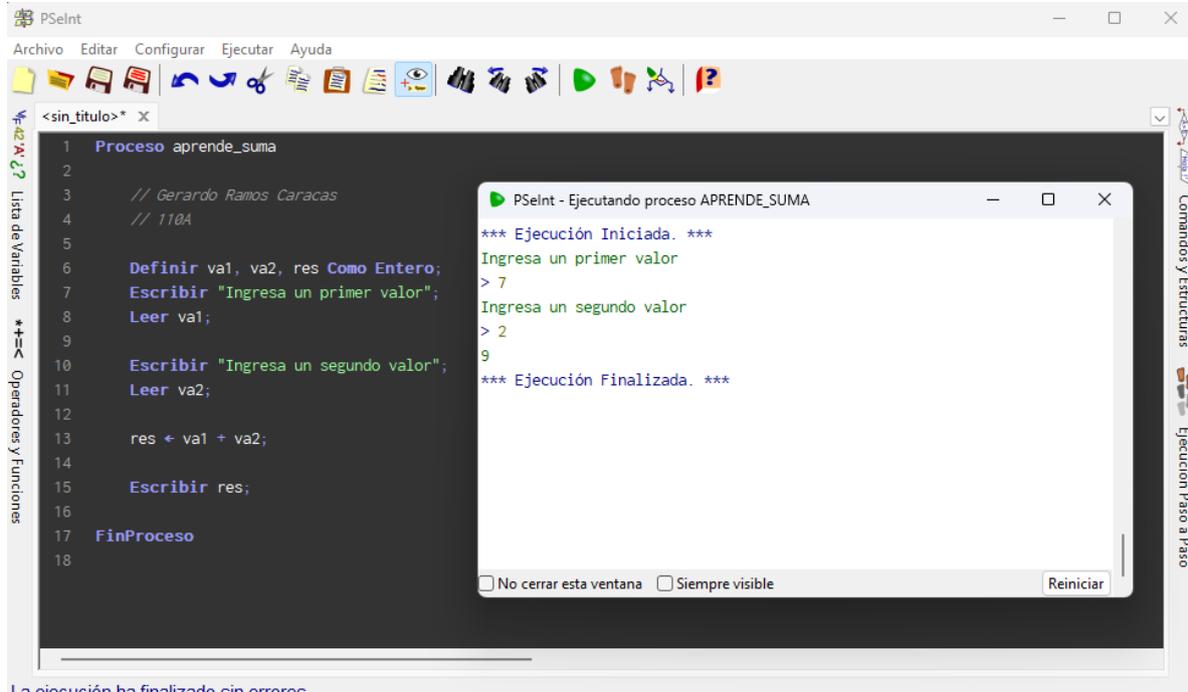
```

PSeInt - Ejecutando proceso APRENDE_ASIGNAR
*** Ejecución Iniciada. ***
Muestra la suma de los números 3 y 2
5
*** Ejecución Finalizada. ***
    
```

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

La ejecución ha finalizado sin errores.

Ejercicio 4. Suma de Variables



The screenshot shows the PSeInt IDE with a program named 'aprende_suma' being executed. The code in the editor is as follows:

```

1 Proceso aprende_suma
2
3 // Gerardo Ramos Caracas
4 // 110A
5
6 Definir va1, va2, res Como Entero;
7 Escribir "Ingresa un primer valor";
8 Leer va1;
9
10 Escribir "Ingresa un segundo valor";
11 Leer va2;
12
13 res ← va1 + va2;
14
15 Escribir res;
16
17 FinProceso
18
    
```

The execution window shows the following output:

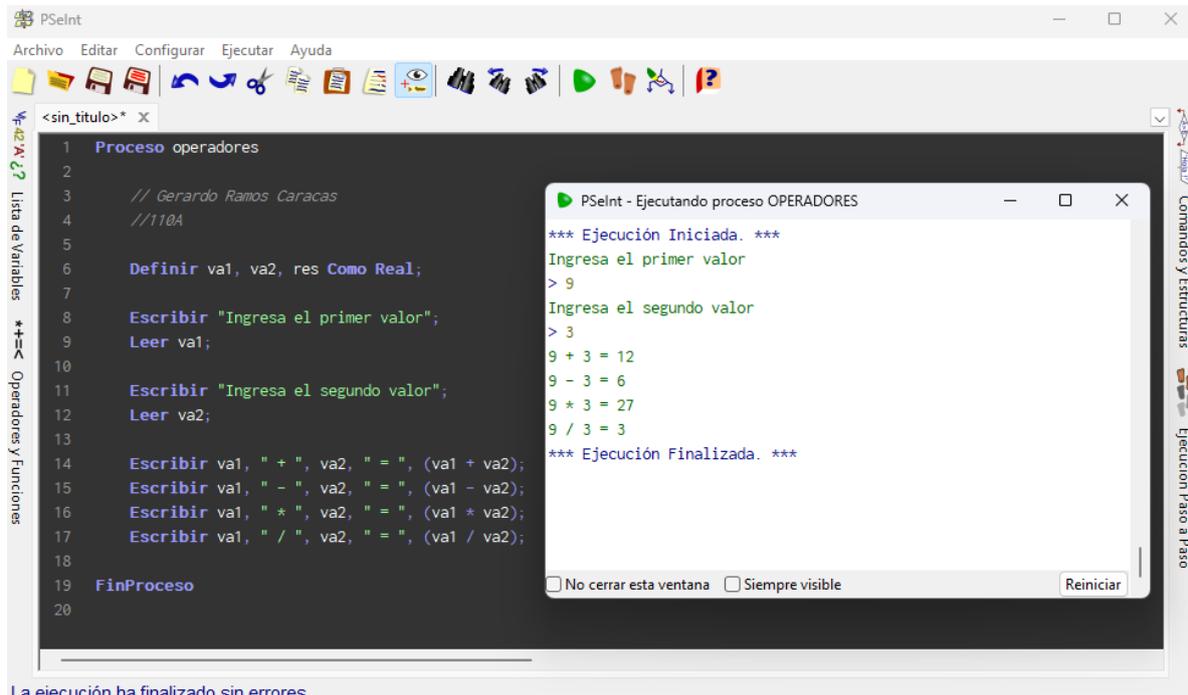
```

PSeInt - Ejecutando proceso APRENDE_SUMA
*** Ejecución Iniciada. ***
Ingresa un primer valor
> 7
Ingresa un segundo valor
> 2
9
*** Ejecución Finalizada. ***
    
```

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

La ejecución ha finalizado sin errores.

Ejercicio 5. Operaciones Varias



```

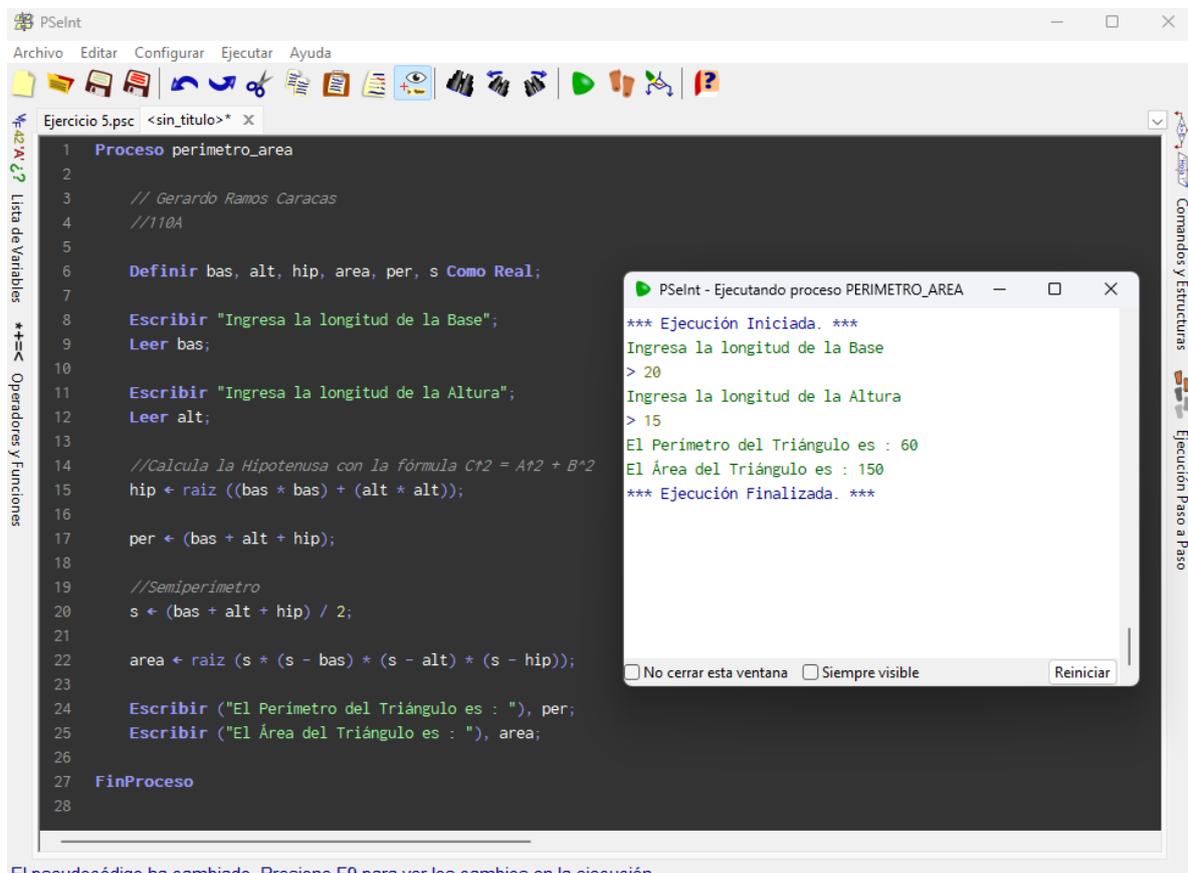
1 Proceso operadores
2
3 // Gerardo Ramos Caracas
4 //110A
5
6 Definir va1, va2, res Como Real;
7
8 Escribir "Ingresa el primer valor";
9 Leer va1;
10
11 Escribir "Ingresa el segundo valor";
12 Leer va2;
13
14 Escribir va1, " + ", va2, " = ", (va1 + va2);
15 Escribir va1, " - ", va2, " = ", (va1 - va2);
16 Escribir va1, " * ", va2, " = ", (va1 * va2);
17 Escribir va1, " / ", va2, " = ", (va1 / va2);
18
19 FinProceso
20
    
```

```

*** Ejecución Iniciada. ***
Ingresa el primer valor
> 9
Ingresa el segundo valor
> 3
9 + 3 = 12
9 - 3 = 6
9 * 3 = 27
9 / 3 = 3
*** Ejecución Finalizada. ***
    
```

La ejecución ha finalizado sin errores.

Ejercicio 6. Perímetro y Área de un Triángulo



```

1 Proceso perimetro_area
2
3 // Gerardo Ramos Caracas
4 //110A
5
6 Definir bas, alt, hip, area, per, s Como Real;
7
8 Escribir "Ingresa la longitud de la Base";
9 Leer bas;
10
11 Escribir "Ingresa la longitud de la Altura";
12 Leer alt;
13
14 //Calcula la Hipotenusa con la fórmula C^2 = A^2 + B^2
15 hip ← raiz ((bas * bas) + (alt * alt));
16
17 per ← (bas + alt + hip);
18
19 //Semiperimetro
20 s ← (bas + alt + hip) / 2;
21
22 area ← raiz (s * (s - bas) * (s - alt) * (s - hip));
23
24 Escribir ("El Perímetro del Triángulo es : "), per;
25 Escribir ("El Área del Triángulo es : "), area;
26
27 FinProceso
28
    
```

```

*** Ejecución Iniciada. ***
Ingresa la longitud de la Base
> 20
Ingresa la longitud de la Altura
> 15
El Perímetro del Triángulo es : 60
El Área del Triángulo es : 150
*** Ejecución Finalizada. ***
    
```

El pseudocódigo ha cambiado. Presione F9 para ver los cambios en la ejecución.

Ejercicio 7. Calcular con Segundos

The screenshot shows the PSeInt IDE with a Pascal program named 'calcula_tiempo.psc'. The program logic is as follows:

```
1 Proceso calcula_tiempo
2
3 // Gerardo Ramos Caracas
4 // 110A
5
6 Definir seg, min, hrs, dias Como Entero;
7
8 min ← 0; hrs ← 0; dias ← 0;
9
10 Escribir "Ingrese cantidad total de Segundos";
11 Leer seg;
12
13 // Conteo total de Horas
14 hrs ← trunc (seg / 3600) ;
15 Mientras seg ≥ 3600 Hacer
16     seg ← seg - 3600;
17 FinMientras
18
19 // Conteo total de Minutos
20 Si seg ≥ 60 Entonces
21     min ← trunc (seg / 60);
22
23     Mientras seg ≥ 60 Hacer
24         seg ← seg - 60;
25     FinMientras
26 FinSi
27
28 // Conteo total de Dias
29 Si hrs ≥ 24 Entonces
30     dias ← trunc (hrs / 24);
31
32     Mientras hrs ≥ 24 Hacer
33         hrs ← hrs - 24;
34     FinMientras
35 FinSi
36
37 Escribir dias, " : ", hrs, " : ", min, " : ", seg;
38
39
40 FinProceso
41
```

An execution window titled 'PSeInt - Ejecutando proceso CALCULA_TIEMPO' is overlaid on the code, showing the following output:

```
*** Ejecución Iniciada. ***
Ingrese cantidad total de Segundos
> 134578
1 : 13 : 22 : 58
*** Ejecución Finalizada. ***
```

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

Below the IDE window, the text reads: "La ejecución ha finalizado sin errores".

Ejercicio 8. Calcular Salario Final

The screenshot shows the PSeInt IDE interface. The main editor window displays the following Pascal code:

```
1 Proceso salario_final
2
3 // Gerardo Ramos Caracas
4 // 18/09/2024 110A
5
6 Definir nom Como Caracter;;
7 Definir sal, mon, ven Como Real;
8
9 Escribir "Ingrese nombre del Vendedor";
10 Leer nom;
11
12 Escribir "Ingrese Salario Fijo mensual del Vendedor";
13 Leer sal;
14
15 Escribir "Ingrese el monto de las Ventas realizadas del mes";
16 Leer ven;
17
18 mon ← sal + (ven * 0.15);
19
20 Escribir ("TOTAL = R$ "), mon;
21
22 FinProceso
23
```

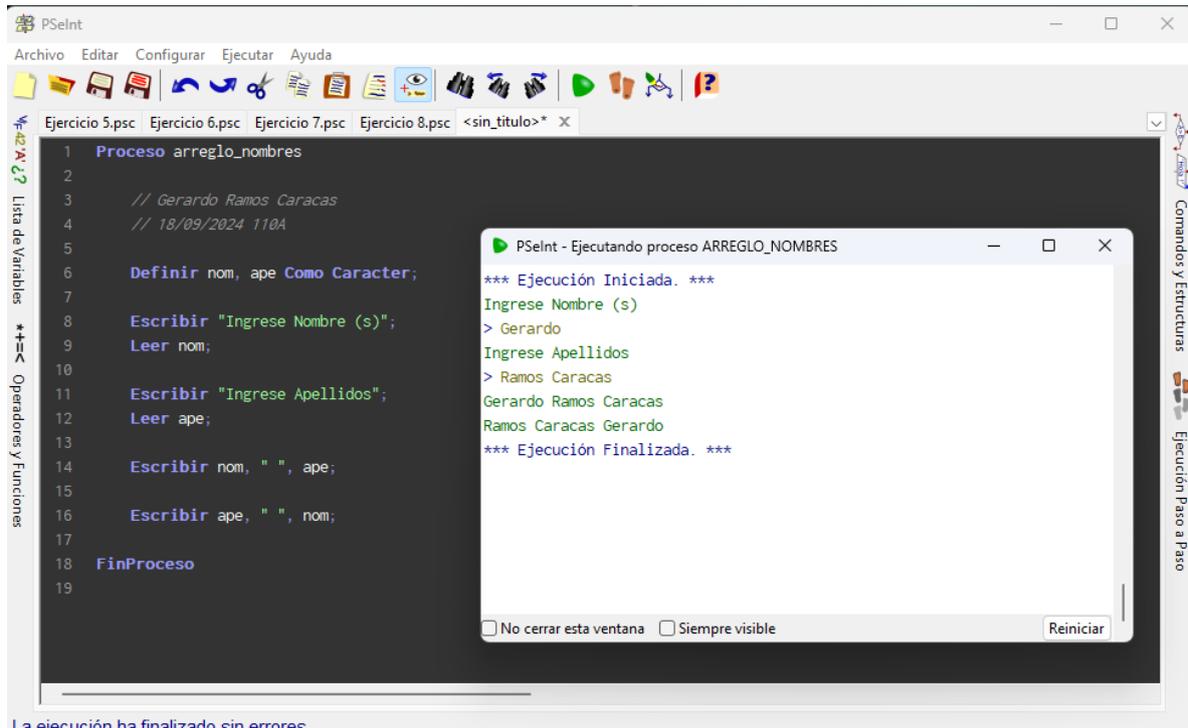
An execution window titled "PSeInt - Ejecutando proceso SALARIO_FINAL" is open, showing the following output:

```
*** Ejecución Iniciada. ***
Ingrese nombre del Vendedor
> Gerardo
Ingrese Salario Fijo mensual del Vendedor
> 500
Ingrese el monto de las Ventas realizadas del mes
> 1230
TOTAL = R$ 684.5
*** Ejecución Finalizada. ***
```

At the bottom of the execution window, there are checkboxes for "No cerrar esta ventana" (unchecked) and "Siempre visible" (checked), along with a "Reiniciar" button.

La ejecución ha finalizado sin errores.

Ejercicio 9. Arreglos con Nombres



The screenshot shows the PSeInt IDE interface. The main window displays a C program named 'Proceso arreglo_nombres'. The code includes comments for the author 'Gerardo Ramos Caracas' and the date '18/09/2024 110A'. The program defines a character array 'nom' and prompts the user to enter a name and surnames. It then prints the name and surnames concatenated with spaces. The execution window shows the program running successfully, with the user input 'Gerardo Ramos Caracas' and the output 'Gerardo Ramos Caracas Ramos Caracas Gerardo'.

```
1 Proceso arreglo_nombres
2
3 // Gerardo Ramos Caracas
4 // 18/09/2024 110A
5
6 Definir nom, ape Como Caracter;
7
8 Escribir "Ingrese Nombre (s)";
9 Leer nom;
10
11 Escribir "Ingrese Apellidos";
12 Leer ape;
13
14 Escribir nom, " ", ape;
15
16 Escribir ape, " ", nom;
17
18 FinProceso
19
```

*** Ejecución Iniciada. ***
Ingrese Nombre (s)
> Gerardo
Ingrese Apellidos
> Ramos Caracas
Gerardo Ramos Caracas
Ramos Caracas Gerardo
*** Ejecución Finalizada. ***

La ejecución ha finalizado sin errores.

Curso: [Fundamentos de Programación](#)

Tarea: [Practicas Unidad 1](#) 

Valor 30

[Ver todos los envíos](#)



Gerardo Ramos Caracas

241u0354@alumno.itssat.edu.mx

Fecha de entrega: 21 de septiembre ...



Cambiar usuario



1 de 1 [Reiniciar preferencias de tabla](#)



◀ Página 1 de 8 ▶



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 2 días 1 hora antes de la fecha límite

Los estudiantes pueden editar este envío

 [Evidencia Prácticas de Clase.pdf](#) 19 de septiembre de 2024, 21:10

▶ [Comentarios \(0\)](#)

Calificación

Calificación:

Practicas

Hoja de presentación	No contien todos los datos 0 puntos	Datos incompletos 2 puntos	Completo 4 puntos		
Indice	No contiene 0 puntos	Contiene 2 puntos			
Practicas	No contiene 0 puntos	Parcialmente 5 puntos	Mitad 10 puntos	Casi el 100% 15 puntos	Todas 20 puntos
Conclusión	No contiene 0 puntos	Pequeña 1 puntos	Completa, falta coherencia 2 puntos	Completa y coherente 4 puntos	

Calificación actual en el libro

26.00

Comentarios de retroalimentación

↕
A ▾
B
I
☰
☰
☰
☰
🔗
🔄
🖼️
📄
🎤
📹
📄
H-P
🕒
🔍

Notificar a estudiantes ?

GUARDAR CAMBIOS

GUARDAR Y MOSTRAR SIGUIENTE

REINICIAR

ALUMNO
GERARDO RAMOS CARACAS

DOCENTE
ROGELIO ENRIQUE TELONA TORRES

GRUPO
110A

MATERIA
FUNDAMENTOS DE PROGRAMACIÓN

EVIDENCIA DE EXAMEN
UNIDAD 1

<sin_titulo>* [X]

```
1  Proceso cociente_residuo
2
3      // Exámen Unidad 1
4      // Gerardo Ramos Caracas
5      // 110A 19/09/2024
6
7  Definir c, r Como Real;
8  Definir div, dno como Entero;
9
10 Escribir "Ingrese Divisor";
11 Leer div;
12
13 Escribir "Ingrese Dividendo";
14 Leer dno;
15
16 c ← trunc (div / dno);
17
18 r ← (div mod dno);
19
20 Escribir div, " / ", dno;
21
22 Escribir ("Cociente = "), c;
23
24 Escribir ("Resto = "), r;
25
26
27 FinProceso
28
```

PSeInt - Ejecutando proceso COCIENTE_R

```
*** Ejecución Iniciada. ***
Ingrese Divisor
> 10
Ingrese Dividendo
> 3
10 / 3
Cociente = 3
Resto = 1
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible

Reiniciar

La ejecución ha finalizado sin errores.

Curso: [Fundamentos de Programación](#)

Tarea: [Examen Unidad 1](#) ⚙️

Valor 40

[Ver todos los envíos](#)



Gerardo Ramos Caracas

241u0354@alumno.itssat.edu.mx

Fecha de entrega: 21 de septiembre ...



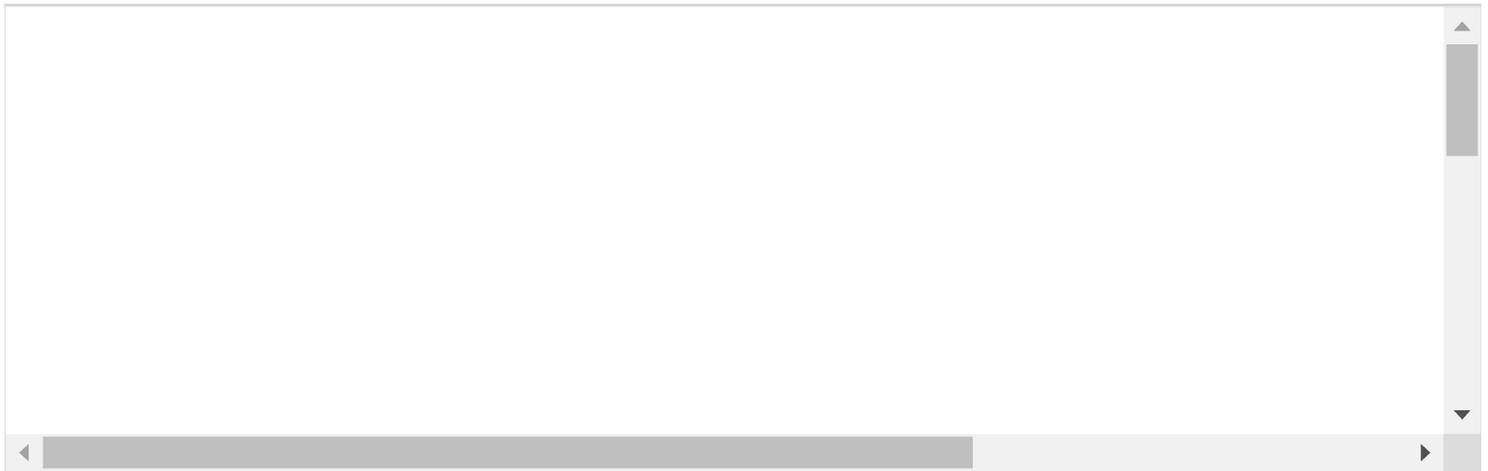
Cambiar usuario



2 de 2 [Reiniciar preferencias de tabla](#)



◀️ Página 1 de 2 ▶️



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 2 días 2 horas antes de la fecha límite

Los estudiantes pueden editar este envío

[Evidencia Captura de Examen.pdf](#) 19 de septiembre de 2024, 20:25

▶ [Comentarios \(0\)](#)

Calificación

Calificación:

Practicas

PDF	Otro formato 0 puntos	Con formato 1 puntos					
Reporte con capturas	No envió 0 puntos	Si envió 4 puntos					
Funcionamiento de código	No funciona 0 puntos	Soluciona 1 prueba 7 puntos	Soluciona 2 pruebas 14 puntos	Soluciona 3 pruebas 21 puntos	Soluciona 4 pruebas 28 puntos	Soluciona 5 pruebas 35 puntos	

Calificación actual en el libro

40.00

Comentarios de retroalimentación

↓
A ▾
B
I
☰
☰
☰
☰
🔗
🔄
🖼️
📄
🎤
📹
📄
H-P
🌐
🔍

Notificar a estudiantes ?

GUARDAR CAMBIOS

GUARDAR Y MOSTRAR SIGUIENTE

REINICIAR