



Curso: Fundamentos de Programación

Tarea: Examen Unidad 1

Fecha de vencimiento: Tuesday, 19 de September de 2023, 22:30



Leonardo Linarez Utrera

leonardoo.utrera@gmail.com

Valor: 40%

Cambiar usuario

16 de 33



Página 1 de 1



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 1 hora 44 minutos antes

El estudiante puede editar esta entrega

 [Captura de pantalla 2023-09-19 190047.pdf](#)

► Comentarios (0)

Calificación

Calificación:

Practicas

PDF	Otro formato 0 puntos	Con formato 1 puntos					<input type="text"/>
Reporte con capturas	No envió 0 puntos	Si envió 4 puntos					<input type="text"/>
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	<input type="text"/>

Calificación actual en el libro de calificaciones

40,00

Comentarios de retroalimentación



Notificar a los estudiantes

Guardar cambios

Reiniciar

Curso: Fundamentos de Programación

Tarea: Investigación Unidad 1

Fecha de vencimiento: Friday, 15 de September de 2023, 23:00



Leonardo Linarez Utrera

leonardoo.utrera@gmail.com

Valor: 30%

Cambiar usuario

16 de 33



Navigation toolbar with icons for back, forward, search, list, highlight, hand, mouse, underline, slash, square, circle, eraser, redaction, pin, and close.

Página 1 de 6

Entrega

Enviado para calificar

Calificado

La tarea fue enviada 1 hora 1 minutos antes

El estudiante puede editar esta entrega

 [investigacion-1.pdf](#)

► Comentarios (0)

Calificación

Calificación:

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

Calificación actual en el libro de calificaciones

30,00

Comentarios de retroalimentación



Notificar a los estudiantes

Guardar cambios

Reiniciar



Nombre del alumno: Leonardo Linarez Utrera

Materia: Fundamentos de Programación

Maestro: Rogelio Enrique Telona Torres

Actividad: Investigación sobre el temario de fundamentos de programación

Primer semestre, Ingeniería en Informática

Introducción

En esta investigación, se verá más a fondo el concepto de fundamentos de programación orientada a objetos, al igual que los subtemas que este conlleva para ser procesada, en esta investigación conocerá los procesos de una programación orientada a objetos y para que funciona cada uno de estos procesos con el fin de comprenderla mejor

Fundamentos de programación orientada a objetos

Es el más usado porque cada uno de estos elementos que necesita el sistema o la aplicación, como el carrito, el producto, el usuario, entre otros, es un objeto en este paradigma. Estos objetos tienen sus propios datos y su propio comportamiento

Evolución de la programación

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.

Conceptos fundamentales de la programación orientada a objetos

En este paradigma, los programas se modelan en torno a objetos que aglutinan toda la funcionalidad relacionada con ellos. De este modo en lugar de crear una serie de funciones sin conexión alguna entre ellas, en POO se crean clases, que representan entidades que quieres manejar en tu programa. Por ejemplo, facturas, líneas de factura, clientes, coches... o cualquier entidad que necesites gestionar conceptualmente. En ese sentido, la POO gira más en torno a los datos que en torno a la lógica. Existen un conjunto de ideas fundamentales que forman los cimientos del desarrollo de software. A estos 4 conceptos que vamos a ver les llamamos los 4 pilares de la programación orientada a objetos.

Esto no quiere decir que fuera de estos 4 pilares no existan otras ideas igual de importantes, sin embargo, estos 4 pilares representan la base de ideas más avanzadas, por lo que es crucial entenderlos.

Estos pilares son: abstracción, encapsulamiento, herencia y polimorfismo.

Relación entre clase y objetos

Las clases y los objetos son los conceptos más importantes de la Programación Orientada por Objetos (POO), y están fuertemente relacionados. Los objetos se crean a partir de clases, y las clases sirven como plantillas para crear objetos. A los objetos también se les llama instancias de clase.

Diseño orientado a objetos

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.

Entorno de programación

Un entorno de programación es un programa o conjunto de programas que engloban todas las tareas necesarias para el desarrollo de un programa o aplicación. Estas tareas son básicamente las siguientes:

- Edición del programa.
- Compilación y enlazado.
- Ejecución.
- Depuración.

Hay quien además incluye la creación de documentación complementaria que facilita el mantenimiento del programa dentro de estas funciones.

Este tipo de entornos incorporan numerosas herramientas, utilidades, aplicaciones ya desarrolladas, ejemplos, tutoriales, etc. Todas ellas encaminadas a facilitar y mejorar el desarrollo.

Editores

El primer elemento necesario para el desarrollo de un programa es un editor de texto. Un editor es un programa que nos permite escribir (editar) las instrucciones del programa y posteriormente guardar el programa en un fichero en un soporte de almacenamiento. Cualquier editor de texto se puede utilizar para editar programas con la única precaución de que a la hora de guardar, salvar o almacenar el programa sólo se almacene el texto sin opciones de formato: negrita, estilos, itálica, etc.

Procesadores del lenguaje: traductores, compiladores e intérpretes

Una vez editado nuestro programa es necesario que este sea procesado y transformado en órdenes que puedan ser ejecutadas por el ordenador. Estas órdenes por tanto deben estar en el único lenguaje que la máquina entiende: el código máquina. Para ello son necesarios los procesadores de lenguaje cuyo concepto es muy amplio. Dentro de los procesadores de lenguaje destacan los traductores, los compiladores y los intérpretes.

Un **compilador** es un programa cuyo cometido es realizar la conversión de un programa escrito en un lenguaje de programación a su correspondiente equivalente en lenguaje máquina. El resultado que devuelve un compilador es un programa que ya puede ser ejecutado por el ordenador destino sin la necesidad de que el compilador esté presente. Por ejemplo, el lenguaje Pascal o el lenguaje C son lenguajes de programación que necesitan ser compilados. Cuando la conversión se realiza entre el lenguaje ensamblador

(*Assembly*) y el código máquina, el compilador recibe el nombre específico de **Ensamblador** (*Assembler*).

Un **intérprete** es un programa que convierte línea por línea el programa escrito en un lenguaje de programación y que a medida que realiza la conversión ejecuta las instrucciones. Evidentemente el intérprete no devuelve nada ya que la ejecución se realiza de forma simultánea. Por este motivo, el intérprete debe estar presente durante la ejecución. Lenguajes de programación que tradicionalmente son interpretados son el LISP y el BASIC.

Un **traductor** es el nombre que reciben aquellos procesadores de lenguaje que convierten programas de unos lenguajes a otros, pero no generan código máquina. Por ejemplo, hay traductores de Pascal a C y viceversa.

Hay otros lenguajes de programación que combinan ambas estrategias como por ejemplo sucede con el lenguaje de programación Java. Para este lenguaje existen traductores que generan un programa en un código denominado intermedio que luego será ejecutado a través de un intérprete que recibe en este caso el nombre de máquina virtual Java.

Enlazadores

Por simplificación y para facilitar la comprensión de los conceptos anteriores se ha señalado que los compiladores y los ensambladores (caso particular de compilador) generan código máquina que puede ser ejecutado por el ordenador. Sin embargo, esto no es totalmente cierto ya que hay una etapa de enlazado que debe ser realizada por otro programa denominado **enlazador** (*linker*). Lo habitual es que durante la escritura de un programa sea necesario utilizar otros subprogramas en forma de bibliotecas de funciones o bien que el propio programa esté formado realmente por varios programas almacenados en diferentes ficheros. Esta situación hace que durante la compilación de cada módulo no se conozca con exactitud la ubicación de las instrucciones del resto de programas o bibliotecas de funciones. El papel del enlazador es unir en un único fichero ejecutable el resultado de todas las compilaciones, así como las bibliotecas estáticas de funciones.

Depuradores

Una vez editado y compilado el programa es necesario ejecutarlo (*run* en inglés), pero es habitual que durante el desarrollo de una aplicación que generen ficheros ejecutables que, aunque sean correctos desde un punto de vista sintáctico no realicen lo que realmente se espera de ellos por lo que se consideran que no funcionan correctamente. Los **depuradores** (*debuggers*) son capaces de ejecutar el programa paso a paso incluyendo además un conjunto de facilidades que permiten observar el valor de las variables y estructuras de datos permitiendo así una mejor localización de errores no evidentes

Conclusión

En esta investigación se vieron los temas relacionados con los fundamentos de programación orientada a objetos, los subtemas de cómo fue que evoluciono la programación, sus conceptos principales, el principal uso de cada una de las ramas de esta, su entorno en cuanto a la programación, entre otros, que en esta investigación se hizo con el fin de poder hacer comprender cada uno de los conceptos explicados con el fin de conocer un poco más acerca de la programación orientada a objetos

Fuentes bibliográficas (En formato IEEE)

“¿Qué es la Programación Orientada a Objetos (POO)?” EDteam - En español nadie te explica mejor. Accedido el 15 de septiembre de 2023. [En línea]. Disponible: <https://ed.team/blog/que-es-la-programacion-orientada-a-objetos-poo>

“Historia de la programación: ¿qué es y cómo ha evolucionado con los años? - DevCamp”. DevCamp. Accedido el 15 de septiembre de 2023. [En línea]. Disponible: <https://devcamp.es/historia-de-la-programacion-que-es-y-como-ha-evolucionado-con-los-anos/>

“Los 4 pilares de la Programación Orientada a Objetos”. Felipe Gavilán. Accedido el 15 de septiembre de 2023. [En línea]. Disponible: <https://gavilanch.wordpress.com/2018/07/05/los-4-pilares-de-la-programacion-orientada-a-objetos/>

“Clases y Objetos - Programación Orientada por Objetos en Ruby”. Inicio - Ruby on Rails 5. Accedido el 15 de septiembre de 2023. [En línea]. Disponible: <https://makeitrealcamp.gitbook.io/programacion-orientada-por-objetos-en-ruby/clases-y-objetos>

“Papel de clases y objetos en el análisis y el diseño | MarcoTeorico.com”. MarcoTeorico.com. Accedido el 15 de septiembre de 2023. [En línea]. Disponible: <https://www.marcoteorico.com/curso/51/fundamentos-de-programacion/393/papel-de-clases-y-objetos-en-el-analisis-y-el-diseno>

“Lenguajes de Programación: Entornos de programación”. Accedido el 15 de septiembre de 2023. [En línea]. Disponible: http://descargas.pntic.mec.es/mentor/visitas/nav_Inici_Progr/lenguajes/len20.html



Nombre del alumno: Leonardo Linarez Utrera

Materia: fundamentos de programación

Maestro: Rogelio Enrique Telona Torres

Actividad: ejercicios hechos en clase junto a ejercicios extra de programación en Pseint

Primer semestre, grupo 110A

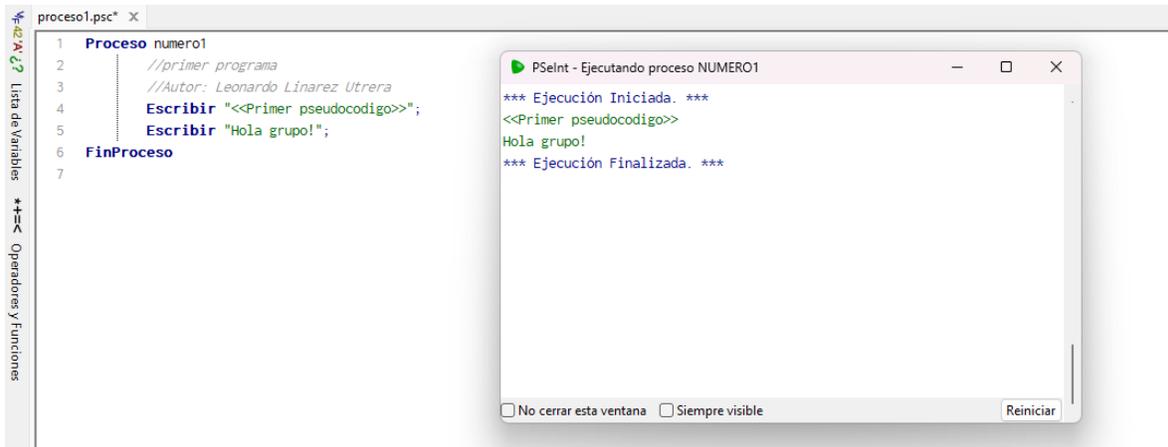
Contenido

Ejercicios en clase	3
Ejercicio 1	3
Ejercicio 2	3
Ejercicio 3	3
Ejercicio 4	4
Ejercicios extra	4
Ejercicio1	4
Ejercicio 2	5
Conclusión	7

Ejercicios en clase

Ejercicio 1

Primer programa

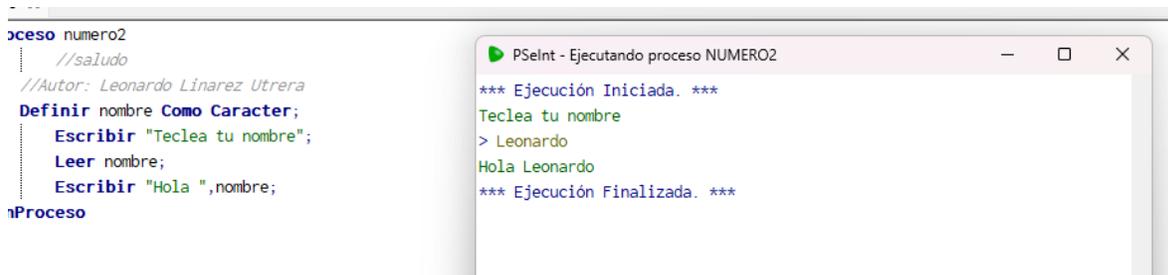


```
proceso1.psc x
1 Proceso numero1
2 //primer programa
3 //Autor: Leonardo Linarez Utrera
4 Escribir "<<Primer pseudocodigo>>";
5 Escribir "Hola grupo!";
6 FinProceso
7
```

```
*** Ejecución Iniciada. ***
<<Primer pseudocodigo>>
Hola grupo!
*** Ejecución Finalizada. ***
```

Ejercicio 2

Hacer un saludo

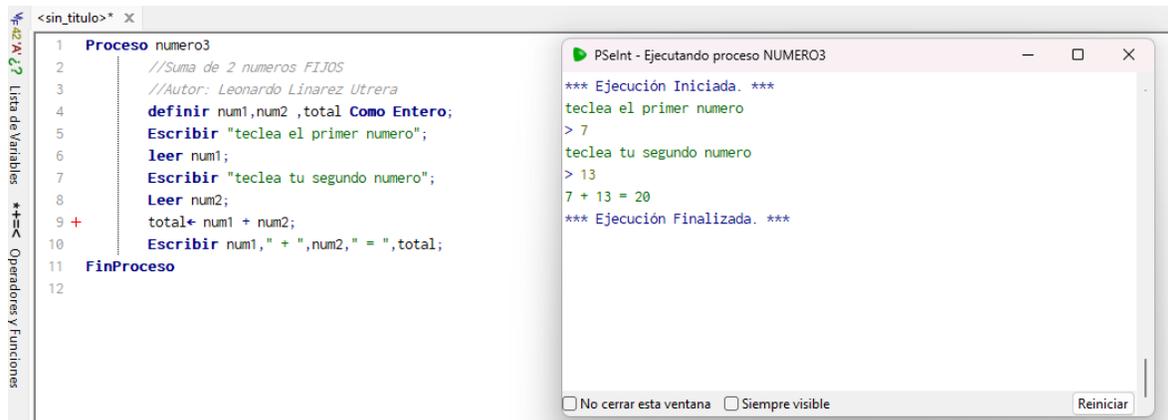


```
proceso numero2
//saludo
//Autor: Leonardo Linarez Utrera
Definir nombre Como Caracter;
Escribir "Teclea tu nombre";
Leer nombre;
Escribir "Hola ", nombre;
FinProceso
```

```
*** Ejecución Iniciada. ***
Teclea tu nombre
> Leonardo
Hola Leonardo
*** Ejecución Finalizada. ***
```

Ejercicio 3

Suma de dos números fijos

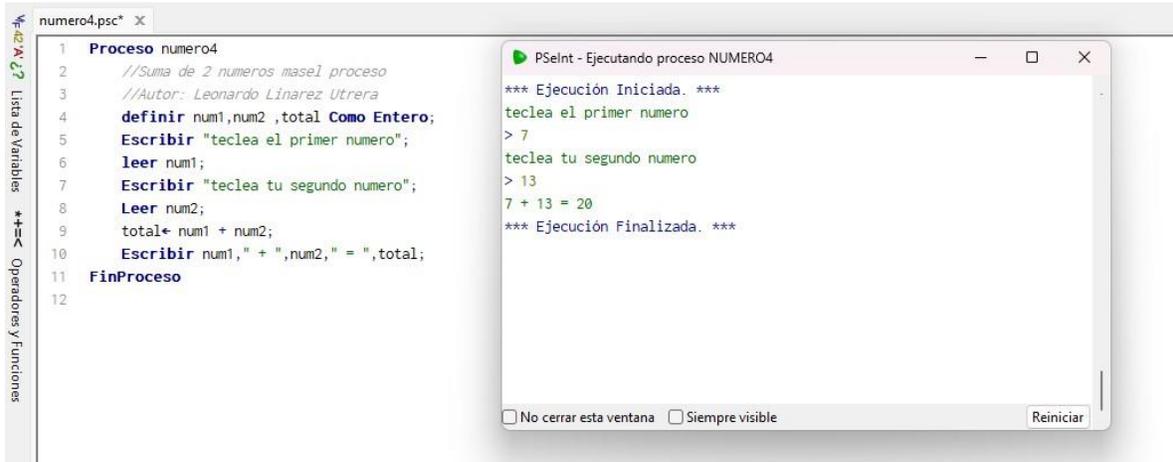


```
<sin_titulo> x
1 Proceso numero3
2 //Suma de 2 numeros FIJOS
3 //Autor: Leonardo Linarez Utrera
4 definir num1,num2 ,total Como Entero;
5 Escribir "teclea el primer numero";
6 leer num1;
7 Escribir "teclea tu segundo numero";
8 leer num2;
9 total<- num1 + num2;
10 Escribir num1, " + ", num2, " = ", total;
11 FinProceso
12
```

```
*** Ejecución Iniciada. ***
teclea el primer numero
> 7
teclea tu segundo numero
> 13
7 + 13 = 20
*** Ejecución Finalizada. ***
```

Ejercicio 4

Suma de dos números fijos integrando el proceso



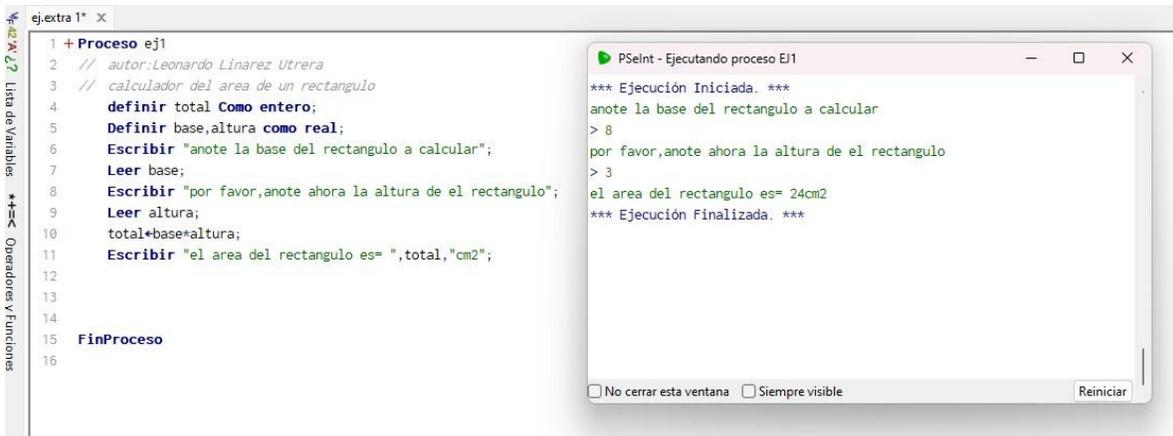
```
1 Proceso numero4
2 //Suma de 2 numeros mas el proceso
3 //Autor: Leonardo Linarez Utrera
4 definir num1,num2 ,total Como Entero;
5 Escribir "teclea el primer numero";
6 leer num1;
7 Escribir "teclea tu segundo numero";
8 Leer num2;
9 total← num1 + num2;
10 Escribir num1," + ",num2," = ",total;
11 FinProceso
12
```

```
*** Ejecución Iniciada. ***
teclea el primer numero
> 7
teclea tu segundo numero
> 13
7 + 13 = 20
*** Ejecución Finalizada. ***
```

Ejercicios extra

Ejercicio 1

Calculador de área de un rectángulo utilizando la formula de Base x Altura

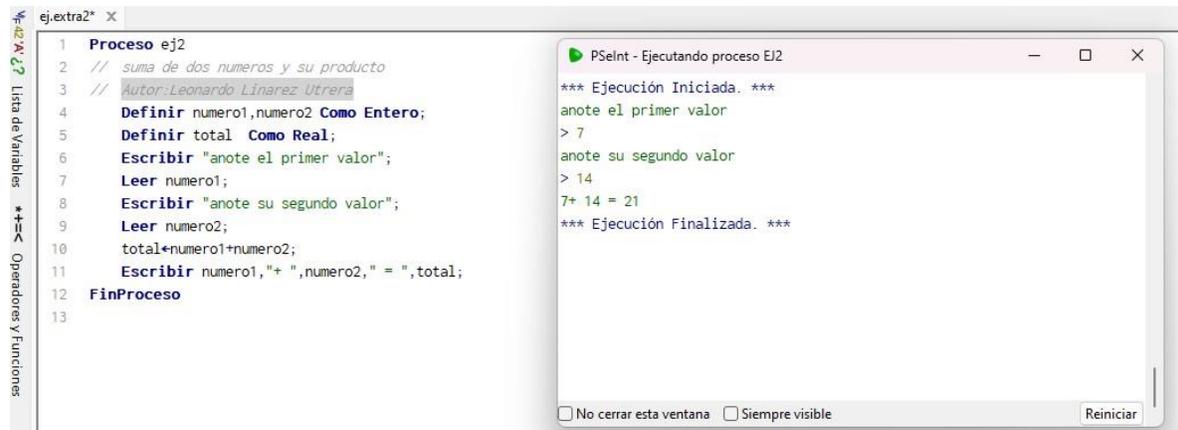


```
1 + Proceso ej1
2 // autor:Leonardo Linarez Utrera
3 // calculador del area de un rectangulo
4 definir total Como entero;
5 Definir base,altura como real;
6 Escribir "anote la base del rectangulo a calcular";
7 Leer base;
8 Escribir "por favor,anote ahora la altura de el rectangulo";
9 Leer altura;
10 total←base*altura;
11 Escribir "el area del rectangulo es= ",total,"cm2";
12
13
14
15 FinProceso
16
```

```
*** Ejecución Iniciada. ***
anote la base del rectangulo a calcular
> 8
por favor,anote ahora la altura de el rectangulo
> 3
el area del rectangulo es= 24cm2
*** Ejecución Finalizada. ***
```

Ejercicio 2

Ejercicio de suma de dos números integrando el proceso



The image shows a code editor window titled 'ej.extra2*' and a terminal window titled 'PSelnt - Ejecutando proceso EJ2'. The code in the editor is as follows:

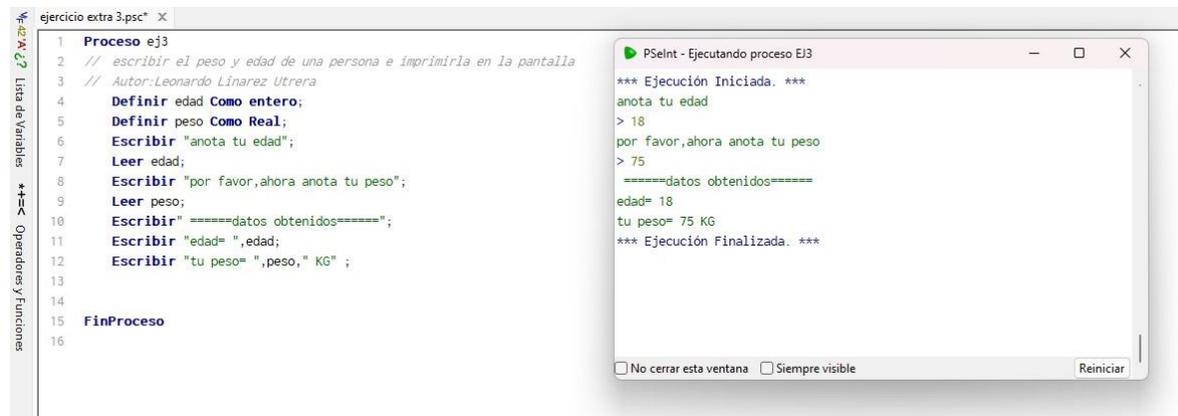
```
1 Proceso ej2
2 // suma de dos numeros y su producto
3 // Autor:Leonardo Linarez Utrera
4 Definir numero1,numero2 Como Entero;
5 Definir total Como Real;
6 Escribir "anote el primer valor";
7 Leer numero1;
8 Escribir "anote su segundo valor";
9 Leer numero2;
10 total=numero1+numero2;
11 Escribir numero1,"+ ",numero2," = ",total;
12 FinProceso
13
```

The terminal output is:

```
*** Ejecución Iniciada. ***
anote el primer valor
> 7
anote su segundo valor
> 14
7+ 14 = 21
*** Ejecución Finalizada. ***
```

Ejercicio 3

Ejercicio sobre anotar datos(peso, edad) para posteriormente reflejarse en la pantalla



The image shows a code editor window titled 'ejercicio extra 3.psc*' and a terminal window titled 'PSelnt - Ejecutando proceso EJ3'. The code in the editor is as follows:

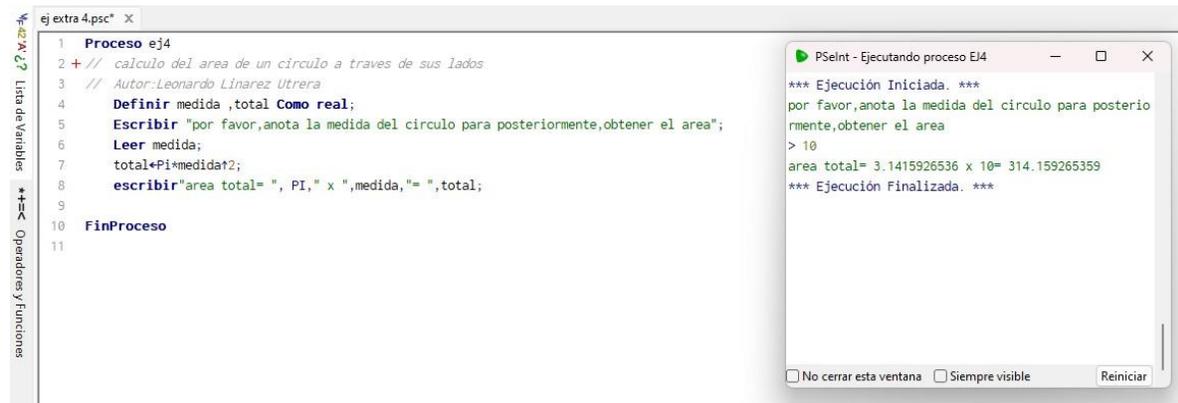
```
1 Proceso ej3
2 // escribir el peso y edad de una persona e imprimirla en la pantalla
3 // Autor:Leonardo Linarez Utrera
4 Definir edad Como entero;
5 Definir peso Como Real;
6 Escribir "anota tu edad";
7 Leer edad;
8 Escribir "por favor,ahora anota tu peso";
9 Leer peso;
10 Escribir "====datos obtenidos====";
11 Escribir "edad= ",edad;
12 Escribir "tu peso= ",peso," KG" ;
13
14
15 FinProceso
16
```

The terminal output is:

```
*** Ejecución Iniciada. ***
anota tu edad
> 18
por favor,ahora anota tu peso
> 75
====datos obtenidos====
edad= 18
tu peso= 75 KG
*** Ejecución Finalizada. ***
```

Ejercicio 4

Ejercicio de calculo de el área de un circulo utilizando la formula de $\text{Pi} \cdot \text{R}^2$

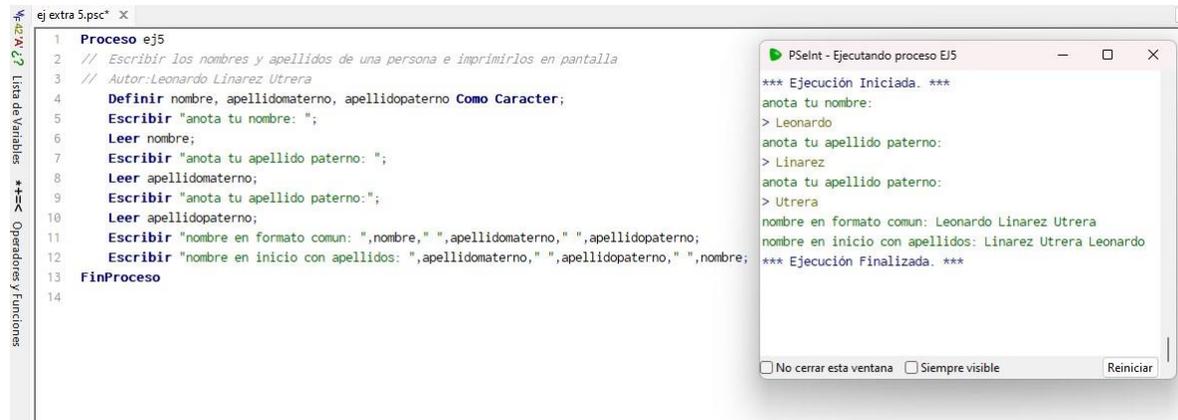


```
ej extra 4.psc X
1 Proceso ej4
2 + // calculo del area de un circulo a traves de sus lados
3 // Autor:Leonardo Linarez Utrera
4 Definir medida ,total Como real;
5 Escribir "por favor,anota la medida del circulo para posteriormente,obtener el area";
6 Leer medida;
7 total+Pi*medida^2;
8 escribir"area total= ", PI," x ",medida," = ",total;
9
10 FinProceso
11
```

```
PSeInt - Ejecutando proceso E4
*** Ejecución Iniciada. ***
por favor,anota la medida del circulo para posteriormente,obtener el area
> 10
area total= 3.1415926536 x 10= 314.159265359
*** Ejecución Finalizada. ***
 No cerrar esta ventana  Siempre visible 
```

Ejercicio5

Ejercicio que consiste en anotar tu nombre completo y que el pseudocodigo lo transcriba de dos maneras diferentes



```
ej extra 5.psc X
1 Proceso ej5
2 // Escribir los nombres y apellidos de una persona e imprimirlos en pantalla
3 // Autor:Leonardo Linarez Utrera
4 Definir nombre, apellidomaterno, apellidopaterno Como Caracter;
5 Escribir "anota tu nombre: ";
6 Leer nombre;
7 Escribir "anota tu apellido paterno: ";
8 Leer apellidomaterno;
9 Escribir "anota tu apellido paterno: ";
10 Leer apellidopaterno;
11 Escribir "nombre en formato comun: ",nombre," ",apellidomaterno," ",apellidopaterno;
12 Escribir "nombre en inicio con apellidos: ",apellidomaterno," ",apellidopaterno," ",nombre;
13 FinProceso
14
```

```
PSeInt - Ejecutando proceso EJ5
*** Ejecución Iniciada. ***
anota tu nombre:
> Leonardo
anota tu apellido paterno:
> Linarez
anota tu apellido paterno:
> Utrera
nombre en formato comun: Leonardo Linarez Utrera
nombre en inicio con apellidos: Linarez Utrera Leonardo
*** Ejecución Finalizada. ***
 No cerrar esta ventana  Siempre visible 
```

Ejercicio 6

Pseudocódigo que lee dos números agregados por el usuario, para en la parte final hallar el número mayor entre estas dos



The image shows a screenshot of a pseudocode editor window titled 'ej extra 6.psc' and a separate window titled 'PSeInt - Ejecutando proceso EJ6'. The editor window contains the following pseudocode:

```
1 Proceso ej6
2 // proyectar en pantalla cual es el numero mayor de 2 numeros elegidos por el usuario
3 // Autor:Leonardo Linares Utrera
4 Definir numero1,numero2 Como Entero;
5 Escribir "anota tu primer valor numerico";
6 Leer numero1;
7 Escribir "anota tu segundo valor numerico";
8 Leer numero2;
9 si numero1<numero2 Entonces
10 | Escribir "tu numero mayor es ",numero2;
11 | SiNo
12 | | Escribir "tu numero mayor es ",numero1;
13 | FinSi
14 FinProceso
15
```

The execution window shows the following output:

```
*** Ejecución Iniciada. ***
anota tu primer valor numerico
> 14
anota tu segundo valor numerico
> 7
tu numero mayor es 14
*** Ejecución Finalizada. ***
```

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

Conclusión

Para mí, ver y utilizar los códigos principales establecidos por el profesor me ha ayudado bastante a poder comprender donde y como se pueden usar estos mismos códigos para obtener diferentes resultados, ya que como se muestra en los ejercicios, hay varias veces una misma palabra<<escribir>>, con el fin de poder proyectar algún numero o letra en la pantalla, que sinceramente no hubiera podido llegar a entender sin explicaciones y ejercicios para reforzar lo aprendido.

La programación en informática es una de las principales materias para poder entender el sentido y las actividades hechas, ya que, como se había mencionado antes, para poder resolver un problema con códigos, se necesita entender a lo que se refieren los códigos y palabras, consiguiendo así un código como los mostrados en este documento.

Curso: Fundamentos de Programación

Tarea: Practicas Unidad 1

Fecha de vencimiento: Thursday, 28 de September de 2023, 23:00



Leonardo Linarez Utrera

leonardoo.utrera@gmail.com

Valor: 30%

Cambiar usuario

16 de 33



Página 1 de 7



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 1 hora 25 minutos antes

El estudiante puede editar esta entrega

 [practicass de clase inf.pdf](#)

► 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 de calificaciones

23,00

Comentarios de retroalimentación



sin entrega

Notificar a los estudiantes

Guardar cambios

Reiniciar