ICONOS
Instituto de Investigación en Comunicación y Cultura

METODOLOGÍA DE LA PROGRAMACIÓN A TRAVÉS DE PSEUDOCÓDIGO Y SU RELACIÓN SINTÁCTICA CON EL LENGUAJE DE PROGRAMACIÓN PHP.

ANEXOS

QUE PARA OBTENER EL TITULO DE

MAESTRÍA EN COMUNICACIÓN CON MEDIOS VIRTUALES

PRESENTA:

ALEJANDRO GONZÁLEZ REYES

ASESOR:

MTRA. ROSELENA VARGAS VELASCO

MÉXICO, CDMX. JULIO 2015

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS DE LA SECRETARÍA DE EDUCACIÓN PÚBLICA SEGÚN ACUERDO NO. 2005625 CON FECHA 22 DE SEPTIEMBRE DE 2005. Y CLAVE 2012

ÍNDICE

ANEXO A 3

A1. INSTALACIÓN PSEINT 3
A2. INSTALACIÓN PHP 4
A3. PUBLICAR UN PROYECTO PHP MEDIANTE UN SERVIDOR WEB LOCAL 6

ANEXO B 12

B1. PALABRAS RESERVADAS DEL LENGUAJE DE PROGRAMACIÓN PHP 12
B2. PALABRAS RESERVADAS DEL PSEUDO-LENGUAJE PSEINT 13

ANEXO C 14

C1. ESTRUCTURAR CONDICIONES COMPLEJAS. 14
C2. DECLARA UNA ESTRUCTURA DE CONTROL COMO PARTE DE UN PROCESO INTERMEDIO PARA LA SOLUCIÓN DE UN PROBLEMA 18
C3. ENCADENAR ESTRUCTURAS DE SELECCIÓN "SI ENTONCES SINO" 22
C4. ANIDAR DIFERENTES ESTRUCTURAS DE SELECCIÓN 26
C5. ENCADENAR ESTRUCTURAS DE CONTROL 31
C6. RECORRER EL CONTENIDO INTERNO DE LAS ESTRUCTURAS DE REPETICIÓN EN FORMATO REGRESIVO 37
C7. DECLARAR DIFERENTES ESTRUCTURAS DE CONTROL (ENCADENADAS Y NO ENCADENADAS) EN UN MISMO PROYECTO 41
C8. REPETICIONES DEFINIDAS EN UNA ESTRUCTURA DEL TIPO "MIENTRAS" 47
C9. DECLARAR UNA ESTRUCTURA DEL TIPO "REPETIR MIENTRAS QUE" COMO PARTE DE UN PROCESO INTERMEDIO PARA LA SOLUCIÓN DE UN PROBLEMA 52

Anexo A

A1. Instalación PSeInt

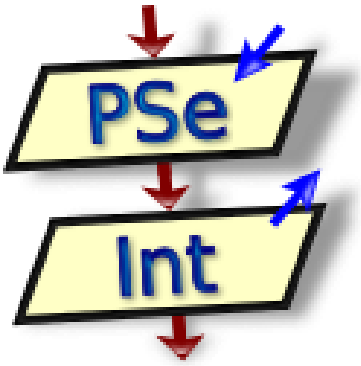


Imagen 1: Logotipo PSeInt - Fuente: http://pseint.sourceforge.net/

Antes de poder diseñar, probar y ejecutar sus Algoritmos mediante el uso de un ordenador, necesitará instalar el software necesario para poder interpretarlos. Este proceso es realmente sencillo en Windows, Mac OS, y las distribuciones Linux más populares como Ubuntu.

La mejor manera de instalar PSeInt en un ordenador depende de su correspondiente Sistema Operativo. Cuando sea posible, se recomienda descargar un instalador desde el sitio Web oficial de PSeInt en http://pseint.sourceforge.net/. Esta dirección electrónica siempre tiene la última versión del software que mantiene el equipo de desarrollo.

- **Instalar PSeInt en Windows:** Los usuarios de Windows deberían simplemente descargar el instalador en formato ejecutable (.exe) disponible desde el sitio Web del proyecto (al momento de escribir estas líneas de la investigación, PSeInt ofrece solo un instalador del programa compatible con procesadores de 32 bits. Sin embargo, puede ser utilizado sin problemas en ordenadores de 64 bits).

- **Instalar PSeInt en Mac OS:** La mejor opción para usuarios de Mac OS es simplemente descargar un instalador en formato "dmg o ppc" disponible desde el sitio Web de PSeInt.
- **Instalar PSeInt en Linux:** No existen instaladores para usuarios de Linux, pero puede descargar una carpeta comprimida en formato "tgz" con archivos previamente compilados (para plataformas de 32/64 bits) desde el sitio Web del proyecto. Dicho de otra forma, dentro de un entorno Linux, PSeInt tiene un comportamiento similar al de un programa portable (Para ejecutar PSeInt desde una distribución basada en Linux, es necesario descomprimir la carpeta que contiene los archivos compilados, acceder a su contenido, buscar el archivo "wxPSeInt" y hacer doble clic en el).

A2. Instalación PHP

Antes de poder escribir instrucciones que sean interpretadas por el Lenguaje de Programación PHP, es necesario instalar el software necesario en su ordenador para su correspondiente ejecución. A diferencia de otros programas, PHP no es un programa independiente, es decir, requiere previamente de la instalación de un programa del tipo servidor (por ejemplo, Apache) que será el encargado de servir los documentos HTML (formularios) al cliente (usuario) a través del uso de un navegador Web e invoque al intérprete de PHP, sólo en aquellos casos donde sea necesario procesar los datos enviados a través de uno o más controles de formulario, y por consiguiente, genere las correspondientes salidas de información; mismas que serán enviadas de vuelta al cliente gracias al programa de servidor.

Afortunadamente, en la red existen paquetes de instalación que incluyen todo lo necesario para preparar el entorno de trabajo con PHP en su ordenador, es decir, incorporan, instalan y configuran de manera automática el servidor de aplicaciones Web, el intérprete PHP y un gestor de Bases de Datos para persistir los datos de su aplicación. Ahora bien, con base en el Sistema Operativo del ordenador destino (donde se realizará la instalación), el nombre de estos paquetes tiende a conocerse por las siglas: **W**AMP (Windows), **L**AMP (Linux) y **M**AMP (Mac).



Imagen 2: Logotipo XAMPP - Fuente: https://upload.wikimedia.org/wikipedia/commons/0/03/Xampp_logo.svg

Si bien el presente apéndice de esta misma investigación no pretende ser una guía completa para la configuración y ejecución de cada uno de estos paquetes de instalación. Se propone el uso de **X**AMPP, una distribución gratuita y **multiplataforma** (de allí la sigla X) que contiene un servidor de aplicaciones Web (Apache), un gestor de Bases de Datos (MariaDB) y los intérpretes para lenguajes de programación PHP y Perl.

- **Instalar XAMPP en Windows:** Para usuarios con Sistema Operativo Windows se encuentra disponible un instalador en formato ejecutable en https://www.apachefriends.org/es/index.html, mismo que es compatible con arquitecturas de 32 o 64 bits.
- **Instalar XAMPP en Mac OS:** La forma más sencilla de ejecutar XAMPP sobre Mac OS, consiste en descargar el instalador en formado "dmg" disponible desde el sitio Web oficial del proyecto.
- **Instalar XAMPP en Linux:** Existe un instalador en formato "run" disponible para la mayoría de las distribuciones Linux tal y como se detalla en https://www.apachefriends.org/es/faq_linux.html. Por lo general, las distribuciones Linux incluyen el contenido de estos

paquetes (servidor Web, intérprete PHP y el gestor de Bases de Datos) por separado. Sin embargo, puede optar por instalar y delegar la configuración de cada uno de estos servicios a través del uso de un paquete general como es el caso de XAMPP.

A3. Publicar un proyecto PHP mediante un servidor Web local

Los pasos que se enumeran a continuación, asumen que se tiene instalado en su ordenador un servidor Web con soporte activado para PHP.

1. Ejecute un editor de textos e ingrese las etiquetas y contenidos necesarios que definirán el cuerpo de su documento HTML (formulario). Finalmente, guarde el archivo con algún nombre (por ejemplo: formHipotenusa**.html**) y sitúelo dentro del directorio **/htdocs** correspondiente a la carpeta de instalación[1] de **XAMPP**.

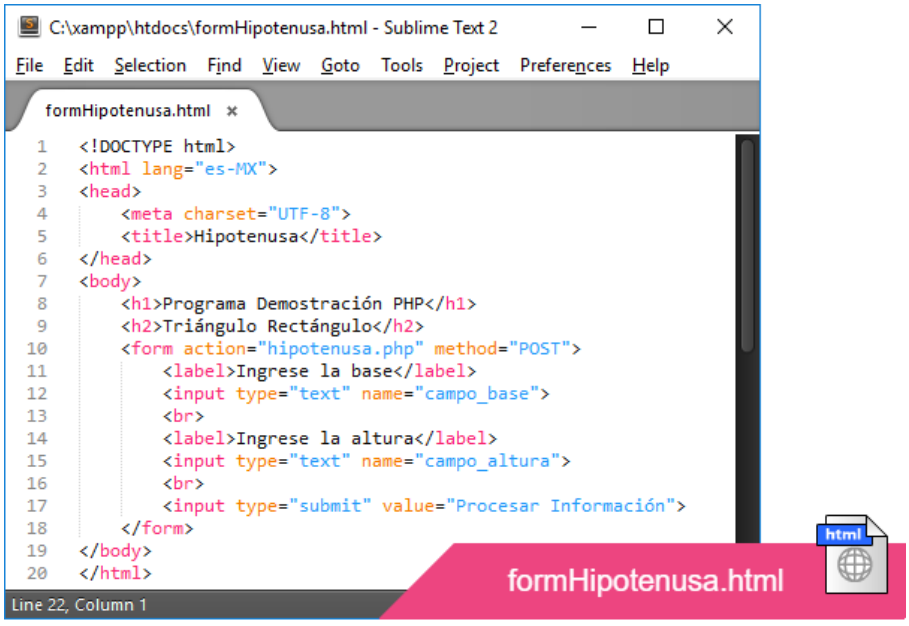


Imagen 3: Documento HTML - Fuente: Diseñada por el autor.

[1] Usuarios Windows: **C:\xampp\hdocs** - Usuarios MacOS: **/Applications/XAMPP/xamppfiles/htdocs** o **/Applications/XAMPP/htdocs** – Usuarios Linux: **/opt/lampp/htdocs**

2. Genere un nuevo documento con la ayuda del editor de textos e ingrese las instrucciones necesarias que definirán el cuerpo de su programa PHP. Finalmente, guarde el archivo con algún nombre que se corresponda con el valor situado dentro del atributo *"action"* de su formulario HTML anterior, (por ejemplo: hipotenusa**.php**) y sitúelo dentro del directorio **/htdocs** correspondiente a la carpeta de instalación de **XAMPP**.

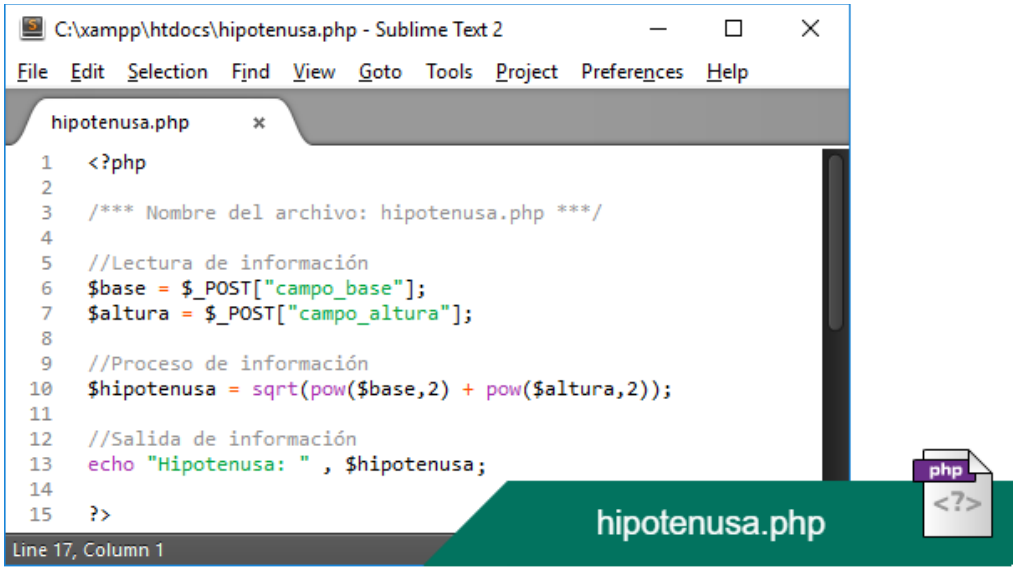


Imagen 4: Documento PHP - Fuente: Diseñada por el autor.

3. Con la ayuda de menú inicio o el lanzador de aplicaciones disponible en su Sistema Operativo. Localice y ejecute el software XAMPP, posteriormente, mediante el panel control[2] de dicha herramienta, inicie el servicio Apache.

[2] Para activar los servicios que provee XAMPP en ordenadores Linux, favor de ingresar el siguiente comando en la consola o terminal de su sistema: **sudo /opt/lampp/lampp start**
Para usuarios MacOS, basta con ejecutar la aplicación **manager-osx**, misma que se corresponde con el Panel de Control de XAMPP.

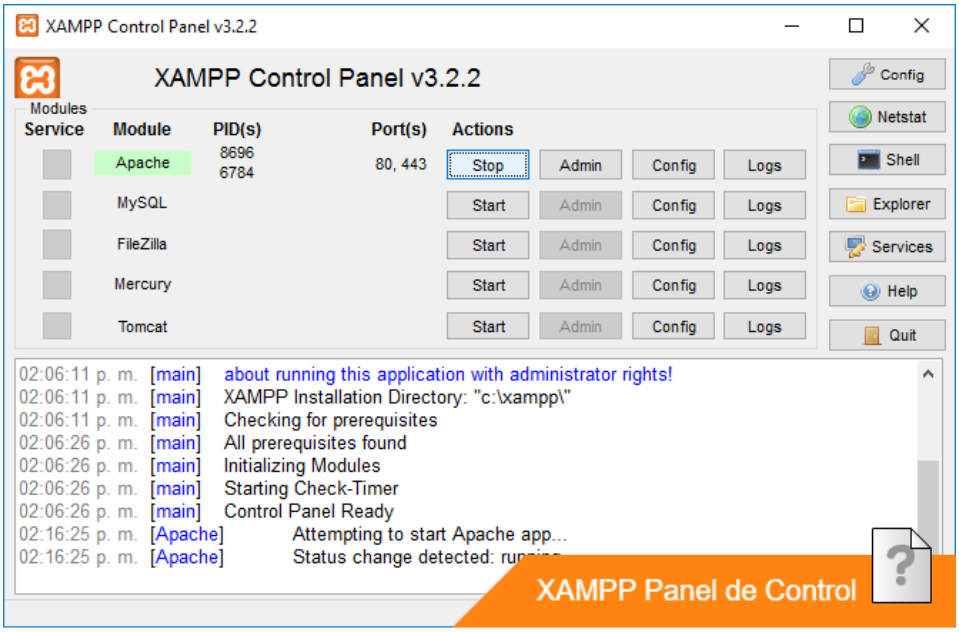


Imagen 5: XAMPP Panel de Control "Sistema Operativo Windows" - Fuente: Diseñada por el autor.

4. Ejecute un navegador Web (por ejemplo: Google Chrome o Mozilla Firefox) e ingrese en la barra de direcciones la siguiente URL:
   http://localhost/formHipotenusa.html
   Misma que se corresponde con el nombre del archivo donde se han definido previamente los controles de formulario HTML (paso 1).

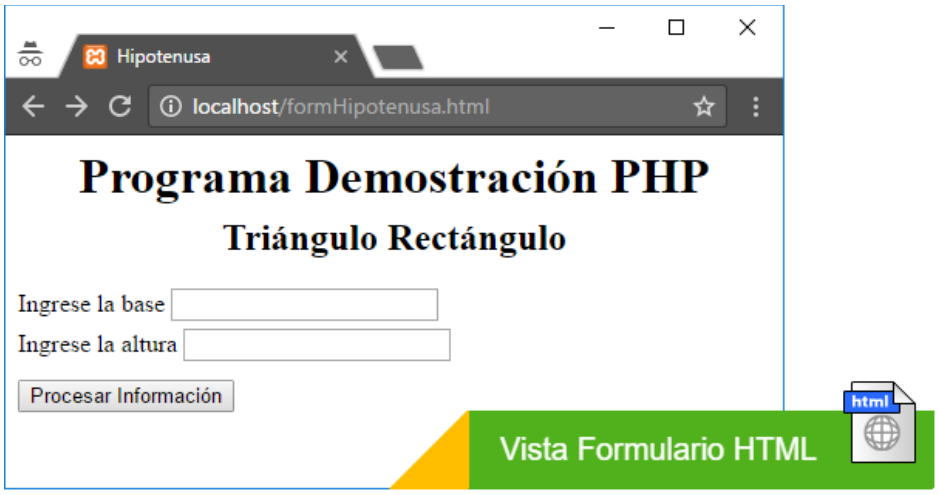


Imagen 6: Vista Formulario HTML - Fuente: Diseñada por el autor.

5. Ahora ingrese algunos datos válidos en los correspondientes campos de texto (por ejemplo: **25** de base y **78** de altura). Finalmente, presione el botón de tipo *"submit"* para enviar la información al archivo de programa PHP.

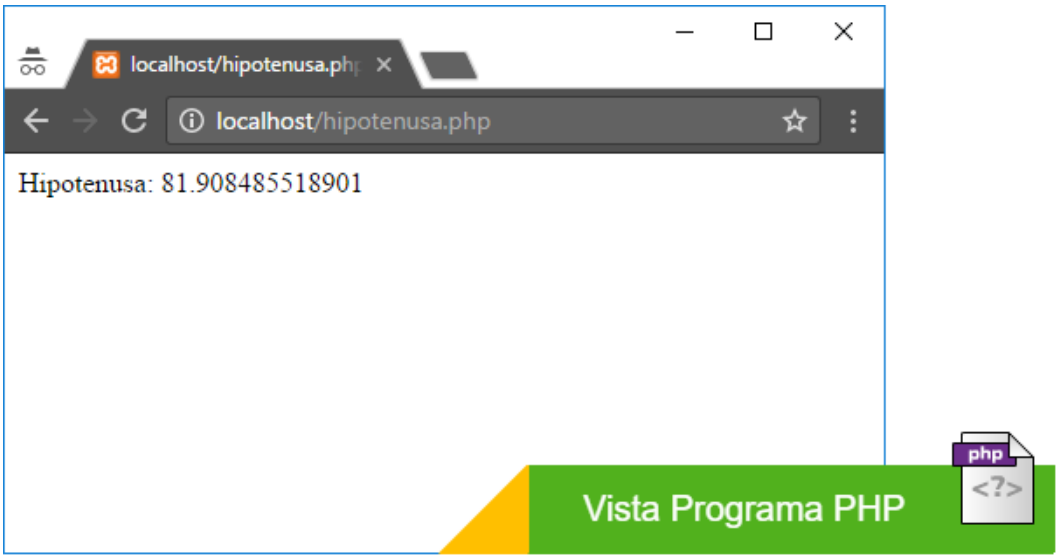


Imagen 7: Vista Programa PHP - Fuente: Diseñada por el autor.

Como se puede observar, si no existen errores de codificación en el archivo de programa PHP, éste mostrará los resultados esperados por cada cliente en la ventana del navegador Web. Sin embargo, en caso de existir errores en el código, el intérprete de PHP lanzará un mensaje de notificación con información concreta (en la mayoría de los casos) acerca del error.



Imagen 8: Notificación de errores de codificación PHP - Fuente: Diseñada por el autor.

La imagen anterior, exhibe uno de los escenarios más frecuentes que cometen los programadores PHP durante el desarrollo de nuevos proyectos "personales y profesionales", así que no debe existir preocupación o molestia al respecto (es una situación normal). Para realizar las correcciones correspondientes, simplemente edite el archivo de programa PHP con la ayuda de un editor de textos, localice la línea o instrucción que origina el error (en la mayoría de las ocasiones el error se localiza una línea antes según lo indicado por el intérprete de PHP), realice los cambios pertinentes, guarde cambios y finalmente, actualice la ventana de su navegador Web (tecla F5).

Si todo es correcto, los resultados esperados se deberían mostrar correctamente en la ventana del navegador. En caso contrario, repita los

pasos que se mencionan en el párrafo anterior; hasta conseguir ejecutar de forma satisfactoria su programa PHP.

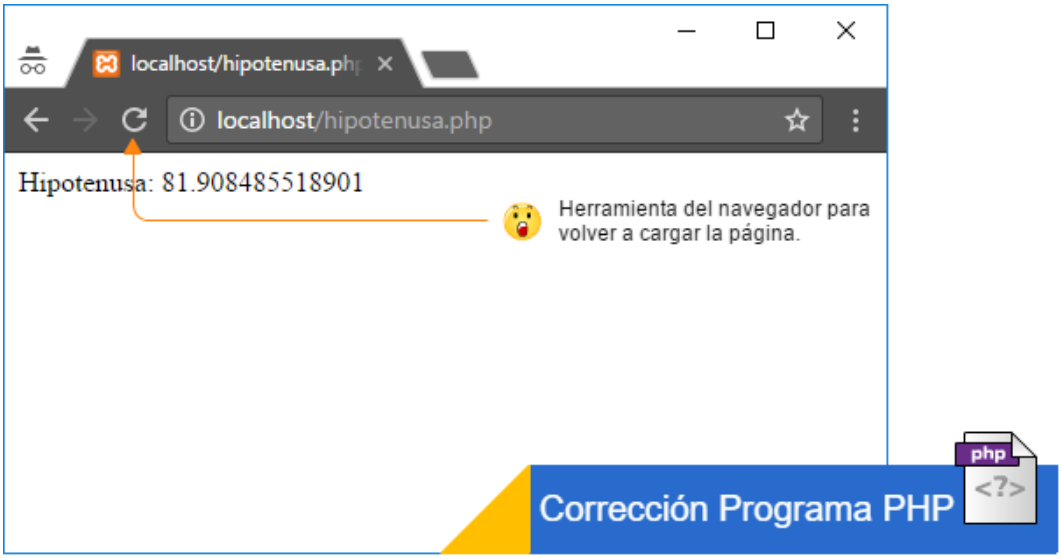


Imagen 9: Ejecución programa PHP - Fuente: Diseñada por el autor.

Llegado a este punto, se tiene una aplicación PHP completa (desde la perspectiva de esta investigación) ejecutándose correctamente a través de un servidor Web local. Sin embargo, es posible que, durante el seguimiento de esta breve guía tutorial haya experimentado algún tipo de problema en cuanto a permisos[3] de directorios (usuarios Linux y MacOS). Si este es su caso, favor de remitirse a la documentación de XAMPP para obtener mayor información al respecto.

[3] En la mayoría de las ocasiones este tipo de problemas se resuelve con otorgar los permisos necesarios (lectura, escritura y ejecución) al directorio htdocs de XAMPP, para ello, ejecute la consola o terminal de su Sistema Operativo e ingrese uno de los siguientes comandos:

Linux: **sudo chmod -R 777 /opt/lampp/htdocs**
MacOS: **sudo chmod -R 777 /Applications/XAMPP/xamppfiles/htdocs**
sudo chmod -R 777 /Applications/XAMPP/htdocs

Aviso importante: no ejecute el comando **chmod -R 777** en directorios alojados en servidores Web que se encuentren configurados para trabajar a nivel de producción.

Anexo B

B1. Palabras reservadas del Lenguaje de Programación PHP

El siguiente listado muestra algunos de los identificadores predefinidos comúnmente utilizados al momento de trabajar con PHP. Ninguno de estos identificadores se puede usar dentro de un *script* como identificador de variables, constantes y nombres de clases. Para mayor información sírvase en visitar el siguiente enlace: http://php.net/manual/es/reserved.php



PALABRAS RESERVADAS DEL LENGUAJE			
echo	print	array	function
break	continue	if	else
elseif	switch	case	for
while	do	foreach	default
and	or	xor	try
catch	class	public	private
protected	static	new	implements
final	global	interface	throw
require	var	use	extends
return	include	abstract	as
true	false	null	self
parent	int	float	string

Tabla 1: Listado de Palabras Reservadas - Lenguaje de Programación PHP - Fuente: Elaborada por el autor.

B2. Palabras reservadas del Pseudo-Lenguaje PSeInt

A continuación, se muestra un listado completo de los identificadores predefinidos por el Pseudolenguaje PSeInt. Ninguno de estos identificadores se puede utilizar como identificador de variables o constantes al momento de diseñar un Algoritmo con la ayuda de dicha herramienta. Para mayor información sírvase en visitar la documentación anexa en la instalación del programa (menú ayuda – Índice).



PALABRAS RESERVADAS DEL PSEUDO-LENGUAJE			
Proceso	FinProceso	Funcion	FinFuncion
SubProceso	FinSubProceso	Algoritmo	FinAlgoritmo
Imprimir	Mostrar	Escribir	Leer
Si	FinSi	Entonces	Sino
Segun	FinSegun	De Otro Modo	Repetir
Hasta Que	Mientras Que	Mientras	FinMientras
Para	FinPara	Con Paso	Hacer
Cada	De	Desde	VERDADERO
FALSO	Y	O	NO
Dimension	Por Valor	Por Referencia	Sin Saltar
Sin Bajar	Borrar Pantalla	Esperar Tecla	Esperar # Segundos
Esperar # Milisegundos	Limpiar Pantalla	MOD	PI
Definir	Como Caracter	Como Entero	Como Logico
Como Real	Opcion	Caso	Si Es
Es	Es Igual A	Es Distinto De	Es Mayor Que
Es Menor Que	Es Mayor O Igual A	Es Menor O Igual A	Es Cero
Es Positivo	Es Negativo	Es Par	Es Impar
Es Multiplo De	Es Divisible Por		

Tabla 2: Listado de Palabras Reservadas (actualizado al 05/Ene/2017) - Pseudolenguaje PSeInt – Fuente: Elaborada por el autor.

Anexo C

C1. Estructurar condiciones complejas.

Problema:

Desarrolle un programa de computadora (escrito mediante Pseudocódigo) para determinar el sueldo quincenal de un empleado partiendo de datos conocidos como son su salario diario y los días laborados (durante la quincena). Además, si el empleado percibe un sueldo quincenal menor o igual a $2,000 pesos y los días que ha laborado durante la quincena son superiores a 14; por políticas internas de la empresa, a éste empleado se le debe entregar un bono adicional "por concepto de productividad" de $500 pesos, los cuales deben ser sumados y girados a su correspondiente sueldo quincenal.

Solución PSeInt:

```
1  Proceso programaSueldo_SiEntonces
2      //Solicitud del salario diario y días trabajados
3      //por el empleado.
4      Imprimir "Ingrese salario empleado"
5      Leer salario
6      Imprimir "Ingrese dias laborados"
7      Leer diasLaborados
8      //Por políticas internas de la empresa, todo
9      //empleado tiene derecho a un bono de productividad.
10     //Sin embargo, para hacerse acreedor del mismo se
11     //debe cumplir con usa serie de requisitos.
12     bono = 0
13     //Determinar el sueldo quincenal del empleado
14     sueldo = salario * diasLaborados
15     //Para otorgar el bono de productividad de $500,
16     //es necesario que el trabajador perciba un sueldo
17     //menor o igual a $2,000 Y que los días trabajados
18     //sean superiores a 14.
19     Si sueldo <= 2000 Y diasLaborados > 14 Entonces
20         //sobre-escribe el valor actual del bono
21         bono = 500
22         Imprimir "Informe: Felicidades por tu bono"
23     FinSi
24     //Salida de información: se muestra por pantalla
25     //el pago final que debe girarse al empleado.
26     //Es decir, su correspondiente sueldo más su bono
27     //(que en este caso, puede ser de $0 o $500)
28     Imprimir "Sueldo Quincenal: $ " , (sueldo + bono)
29 FinProceso
```

Imagen 10: Pseudocódigo "Sueldo Quincenal" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Como se puede observar, la estructura de selección "**Si Entonces**" está conformada por las líneas de código 19 a 23. La condición se formula a partir de dos peguntas unidas por el operador lógico **Y**, la primera de ellas comprueba que el valor referenciado por la variable **sueldo** sea menor o igual a **2000** (que en este caso representa una cantidad monetaria), y la

segunda verifica que el contenido referenciado por la variable **diasLaborados** sea mayor a **14** (misma que representa el número de días trabajados por el empleado durante la quincena).

Una vez que se comprueban ambas condiciones, se verifica que el resultado final sea **VERDADERO** para proceder a ejecutar las instrucciones declaradas dentro del bloque de selección (para el caso que ocupa este ejemplo, al utilizar el operador lógico "**Y**" ambas condiciones deben satisfacerse para devolver un resultado verdadero), en caso contrario, dichas instrucciones son omitidas durante el proceso, lo que da a lugar que el flujo del programa continúe con normalidad y ejecute toda instrucción declarada inmediatamente después del bloque de selección, es decir, las líneas 28 y 29 correspondientes a este ejemplo (Recuerde que los comentarios son omitidos por PSeInt).

Capturas de Pantalla:

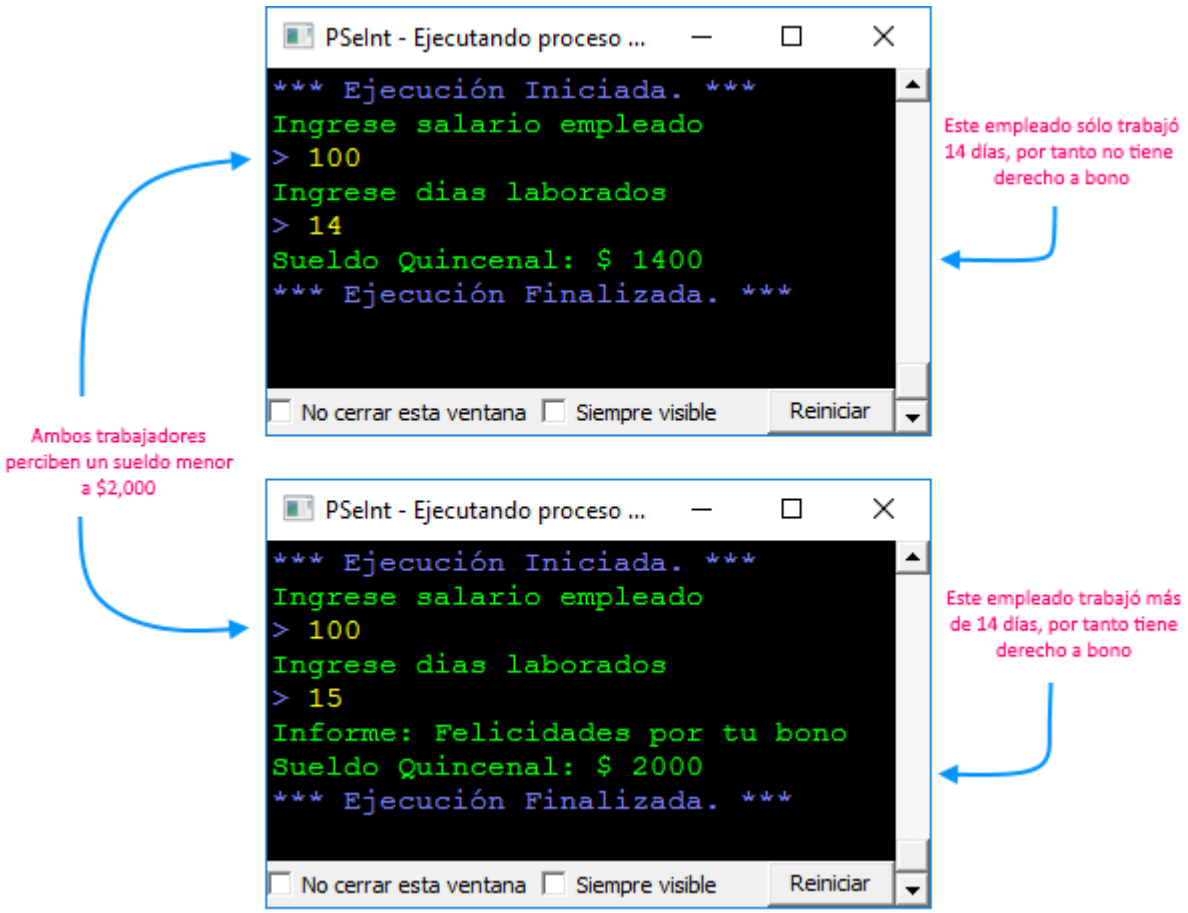


Imagen 11: Ejecución Pseudocódigo "Sueldo Quincenal" - Fuente: Diseñada por el autor.

C2. Declara una estructura de control como parte de un proceso intermedio para la solución de un problema

Problema:

Desarrolle un programa de computadora (escrito mediante Pseudocódigo) para determinar el pago que debe realizar un alumno por concepto de reinscripción escolar. Tome en cuenta que el costo por este concepto es de $3,600 pesos, menos un cierto porcentaje de descuento que se determina a partir de su promedio general y su correspondiente beca por aprovechamiento académico.

Si un alumno tiene un promedio general mayor o igual a 9 y actualmente cuenta con una beca, entonces el pago por concepto de reinscripción debe ser de $3,600 pesos menos un 30% de descuento; de lo contrario, el alumno pierde su beca (si es que la tiene) y debe pagar los $3,600 pesos menos un 4% de descuento.

Finalmente, el programa debe mostrar por pantalla un informe detallado (a manera de comprobante de pago) acerca del movimiento realizado, donde se refleje: el nombre del alumno, el costo real de la reinscripción ($3,600), el descuento aplicado, el total a pagar, así como un mensaje para conocer la situación actual de la beca escolar.

Solución PSeInt:

```
1  Proceso reinscripcionesEscolares
2      //---------- Datos de Entrada ------------
3      Imprimir "Ingrese nombre del alumno"
4      Leer nombre
5      Imprimir "Ingrese promedio general"
6      Leer promedio
7      Imprimir "¿El alumno tiene beca? (verdadero/falso)"
8      Leer beca
9      //-------------- Proceso ---------------
10     reinscripcion = 3600
11     //Si el promedio del alumno es mayor o igual a 9 Y a su vez
12     //tiene beca, se aplica un descuento del 30% en el pago de la
13     //reinscripción y por consiguiente sigue conservando su beca.
14     //En caso contrario, el alumno pierde su beca y el
15     //descuento de la reinscripción es de un 4%.
16     Si promedio >= 9 Y beca == VERDADERO Entonces
17         descuento = reinscripcion * 0.30
18     Sino
19         descuento = reinscripcion * 0.04
20         beca = FALSO
21     FinSi
22     pagoTotal = reinscripcion - descuento
23     //------------ Salida de Información -------------
24     Imprimir "NOMBRE DEL ALUMNO: " , nombre
25     Imprimir "CONCEPTO POR REINSCRIPCIÓN: $ " , reinscripcion
26     Imprimir "DESCUENTO: $ " , descuento
27     Imprimir "TOTAL A PAGAR: $ " , pagoTotal
28     Imprimir "RENOVACIÓN DE BECA: " , beca
29 FinProceso
```

Imagen 12: Pseudocódigo "Reinscripción Escolar" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Como se puede observar, se trata de un Pseudocódigo identificado bajo el nombre de **resincripcionesEscolares**, mismo que para poder trabajar correctamente, en las líneas 3-8 solicita al usuario el ingreso de ciertos datos como son: el nombre completo del alumno, su promedio general, y un valor booleano (**FALSO** o **VERDADERO**) para indicar si el alumno actualmente tiene una beca. Una vez que el programa cuenta con los

datos necesarios, éste procede a verificar el tipo de descuento a otorgar al alumno a partir de una estructura de selección "**Si Entonces Sino**" (líneas 16-21), misma que se encarga de comprobar (mediante una condición compuesta) que el promedio del alumno sea mayor o igual a **9** puntos **Y** que actualmente cuente con una beca académica (**VERDADERO**). En caso de que dicha condición se cumpla, el programa procede a determinar el 30% del costo real de la reinscripción; de lo contrario, el programa procede a cancelar la renovación de la beca de dicho alumno (si es que la tenía previamente) y determina solo el 4% del costo real de la reinscripción.

Sumado a lo expuesto, una vez que el programa conoce la cifra monetaria a descontar (30% o el 4% de los $3,600 pesos), la línea 22 se encarga de realizar dicho descuento y, por consiguiente, se muestra por pantalla un informe detallado (líneas 24-28) con el nombre del alumno, el descuento otorgado, el monto a pagar por concepto de reinscripción, y la situación actual de su beca para el siguiente ciclo escolar (si fue o no renovada).

Capturas de Pantalla:

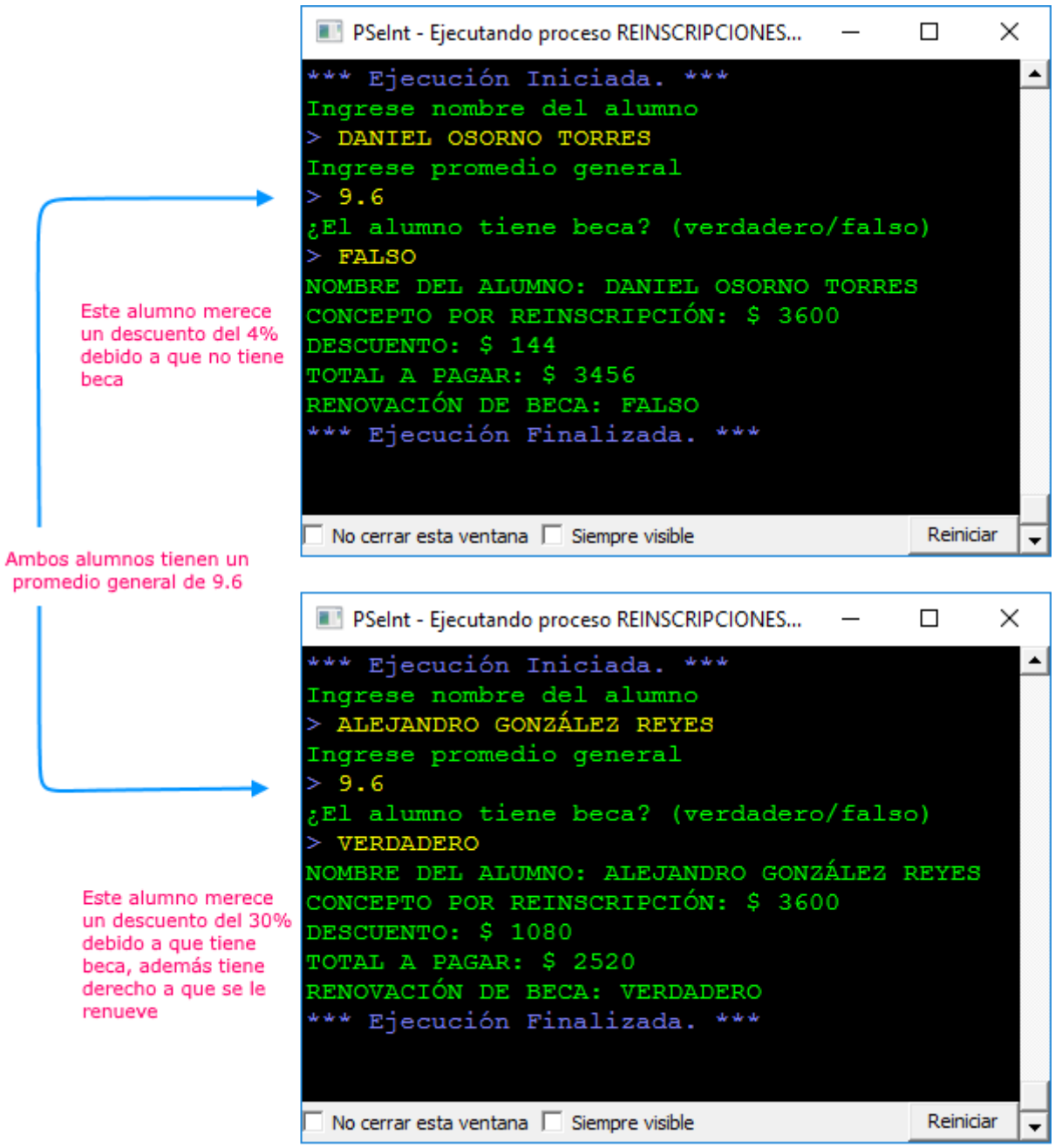


Imagen 13: Ejecución Pseudocódigo "Reinscripción Escolar" - Fuente: Diseñada por el autor.

C3. Encadenar estructuras de selección "Si Entonces Sino"

Problema:

Desarrolle un programa de computadora (escrito mediante Pseudocódigo) para determinar la situación académica que guarda un alumno con respecto a una determinada materia. Tome en cuenta que, para su correspondiente evaluación es necesario que éste presente dos exámenes parciales, mismos que deben ser promediados y valorados con base en las siguientes reglas:

- Si el promedio es inferior a 6.0, el alumno esta reprobado y por consiguiente debe presentar un examen de tipo extraordinario para su correspondiente regularización.
- Si el promedio oscila entre 6.0 y 7.9, el alumno no está reprobado, pero debe presentar un examen final de conocimientos para acreditar la materia correctamente.
- Si el promedio es mayor o igual a 8.0, el alumno acredita satisfactoriamente la materia (exenta) y por tanto no necesita presentar examen adicional alguno.

Finalmente, el programa debe mostrar por pantalla un informe detallado con los datos generales del alumno, el nombre de la materia, el promedio general y su correspondiente situación académica (materia acreditada, examen final, o examen extraordinario).

Solución PSeInt:

```
Proceso estructuraSiEntoncesAnidado
    //---------- Datos de Entrada -----------
    Imprimir "Ingrese nombre del Alumno"
    Leer nombreAlumno
    Imprimir "Ingrese nombre de la Materia"
    Leer materia
    Imprimir "Ingrese primera calificación parcial"
    Leer primerParcial
    Imprimir "Ingrese segunda calificación parcial"
    Leer segundoParcial
    //--------------- Proceso ---------------
    promedio = (primerParcial + segundoParcial) / 2
    //Comprobando si el promedio general de la materia es
    //aprobatorio, reprobatorio, o en su defecto el alumno
    //tiene que presentar un examen final.
    Si promedio < 6 Entonces
        //si el promedio es menor que 6, el alumno...
        mensaje = "Presenta examen extraordinario"
    Sino
        Si promedio >= 6 Y promedio < 8 Entonces
            //si el promedio esta entre 6 y 7.9, el alumno...
            mensaje = "Presenta examen final"
        Sino
            //si el promedio es de 8 en adelante, el alumno tiene...
            mensaje = "Materia acreditada"
        FinSi
    FinSi
    //---------- Salida de Información ------------
    Imprimir "NOMBRE ALUMNO: " , nombreAlumno
    Imprimir "NOMBRE MATERIA: " , materia
    Imprimir "PROMEDIO PARCIAL: " , promedio
    Imprimir "INFORME: " , mensaje
FinProceso
```

Imagen 14: Pseudocódigo "Evaluación Parcial" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Con la finalidad de resolver la problemática expuesta con anterioridad, el Pseudocódigo solicita al operador del programa el ingreso de ciertos datos de entrada, tales como: el nombre del alumno, la materia a evaluar, y las

dos calificaciones parciales que obtuvo "el alumno" a lo largo del ciclo escolar (líneas 3-10).

Sumado a lo expuesto, en la línea 12 se determina el promedio general de la materia a partir de los datos proporcionados por el usuario (calificaciones parciales). Luego, en la línea 16 se comprueba si el alumno ha reprobado la materia correspondiente (promedio menor que 6.0), de ser así, se almacena un mensaje con la leyenda "**Presenta examen extraordinario**"; de lo contrario, se vuelve a comprobar su promedio general mediante el uso de otra estructura selectiva (anidación). En este caso, se verifica que el alumno cuente con un promedio cuyo valor que se encuentre dentro del rango de los **6.0** a **7.9** puntos (línea 20), de ser así, se crea la cadena de texto "**Presenta examen final**" y se vincula a la variable "**mensaje**"; de lo contrario (al no cumplirse las condiciones anteriores), el alumno se encuentra debidamente acreditado en la materia y se procede a generar el mensaje de texto correspondiente (línea 25).

Finalmente, en las líneas 29-32 se emite un informe por pantalla que permite conocer a detalle el nombre completo del alumno, la materia evaluada, el promedio general obtenido, y la situación académica del alumno correspondiente a dicha materia.

Capturas de Pantalla:

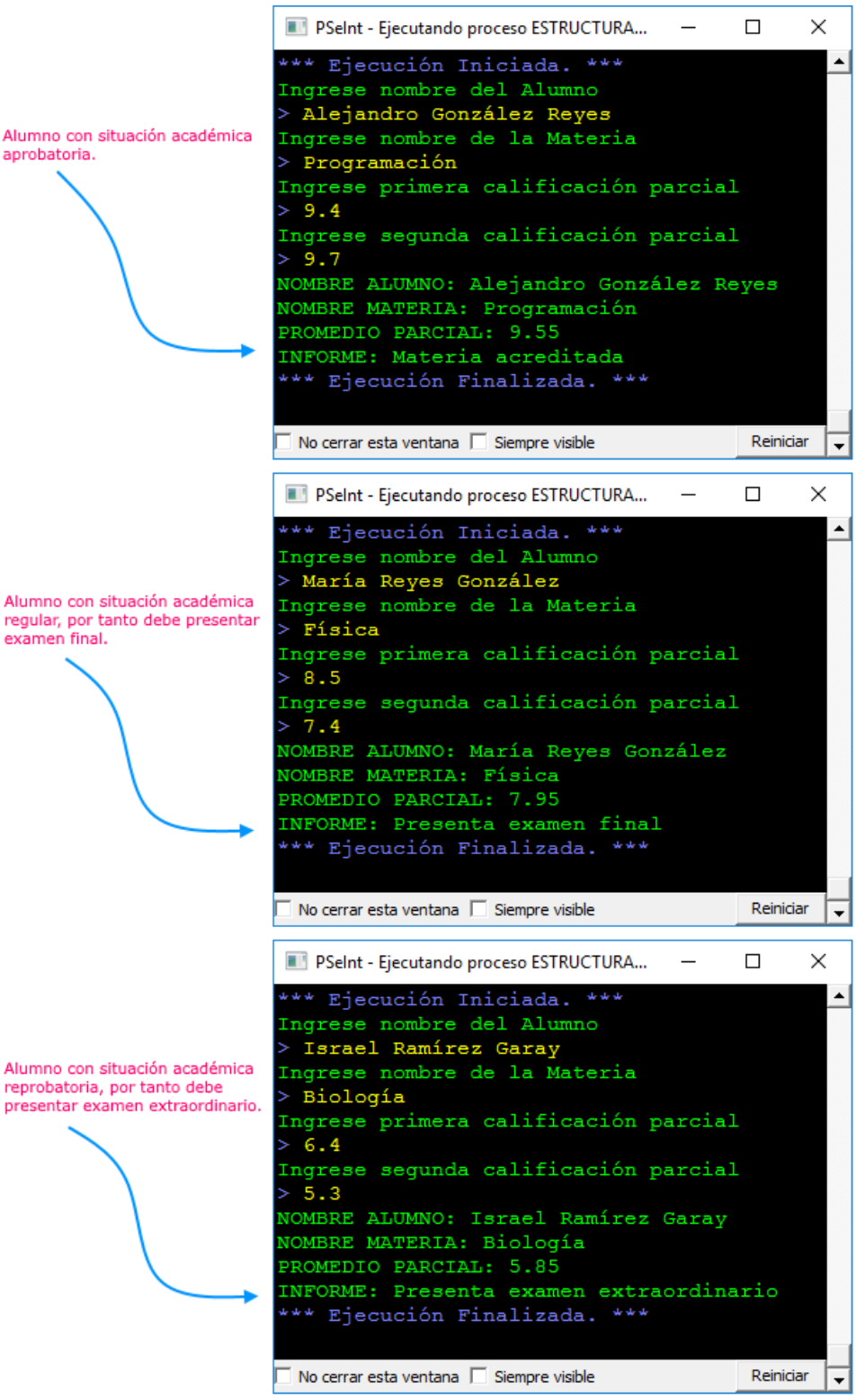


Imagen 15: Ejecución Pseudocódigo "Evaluación Parcial" - Fuente: Diseñada por el autor.

C4. Anidar diferentes estructuras de selección

Problema:

La empresa "Banco Mexicano" le pide a usted desarrollar un programa de computadora (escrito mediante Pseudocódigo) para el depósito de dineros girados en efectivo y en cheque. Mismo que a partir de datos conocidos como son: el nombre del cliente, el número de cuenta beneficiada, la cantidad monetaria a depositar, y el modo de depósito bancario (efectivo o cheque); determine si la operación bancaria es candidata o no al cobro de un determinado impuesto, mismo que se determina a partir de los siguientes puntos:

- Si el deposito es mediante cheque: no se cobra impuesto alguno, pero el programa debe solicitar adicionalmente al usuario, el número de cheque depositado para poder transferir los fondos monetarios a la cuenta beneficiada
- Si el deposito es en efectivo: por instrucciones del Gobierno Federal se debe retener un impuesto del 10% por concepto de IDE (impuesto sobre depósitos en efectivo), pero siempre y cuando la cantidad depositada exceda a los $15,000; en caso contrario, el deposito pasa integro a la cuenta beneficiada.

Es importante hacer notar que el impuesto (IDE) es aplicable sólo sobre la cantidad excedente, es decir, para un depósito de $16,500 el impuesto del 10% aplica sobre los $1,500 y no sobre los $16,500;

Finalmente, el programa debe mostrar por pantalla un informe detallado (comprobante) con los datos generales de la operación bancaria.

Solución PSeInt:

```
Proceso estructurasSeleccionFusionadas
    Imprimir "***********************************************"
    Imprimir "  BANCO MEXICANO - SUCURSAL SAN MIGUEL ALMAYA  "
    Imprimir "***********************************************"
    Imprimir "INGRESE NOMBRE DEL CLIENTE"
    Leer nombreCliente
    Imprimir "INGRESE NÚMERO DE CUENTA BENEFICIADA"
    Leer numeroCuenta
    Imprimir "INGRESE CANTIDAD A DEPOSITAR EN LA CUENTA"
    Leer cantidadDepositada
    Imprimir "SELECCIONE EL MODO DE DEPOSITO BANCARIO"
    Imprimir "1 = CHEQUE"
    Imprimir "2 = EFECTIVO"
    Leer modoDeposito
    //Comprobando el modo de deposito seleccionado por el usuario
    Segun modoDeposito Hacer
        1:
            Imprimir "INGRESE EL NÚMERO DE CHEQUE ASOCIADO"
            Leer numeroCheque
            Imprimir "-------------- Ticket ------------------"
            Imprimir "CUENTA BENEFICIADA: ", numeroCuenta
            Imprimir "CLIENTE: ", nombreCliente
            Imprimir "DEPOSITO GIRADO EN: CHEQUE"
            Imprimir "MONTO TRANSFERIDO: $ ", cantidadDepositada
            Imprimir "Gracias por su preferencia..."
        2:
            Si cantidadDepositada <= 15000 Entonces
                cantidadSujetaIDE = 0
                ide = 0
            Sino
                //Obteniendo el excedente
                cantidadSujetaIDE = cantidadDepositada - 15000
                //Determinando el impuesto sobre el excedente
                ide = (cantidadSujetaIDE * 0.10)
                //Reteniendo impuesto sobre depositos en efectivo
                cantidadDepositada = cantidadDepositada - ide
            FinSi
            Imprimir "-------------- Ticket ------------------"
            Imprimir "CUENTA BENEFICIADA: ", numeroCuenta
            Imprimir "CLIENTE: ", nombreCliente
            Imprimir "DEPOSITO GIRADO EN: EFECTIVO"
            Imprimir "MONTO TRANSFERIDO: $ ", cantidadDepositada
            Imprimir "CANTIDAD SUJETA A IDE: $ ", cantidadSujetaIDE
            Imprimir "IDE: $ ", ide
            Imprimir "Gracias por su preferencia..."
        De Otro Modo:
            Imprimir "Opción incorrecta, intente nuevamente."
    FinSegun
FinProceso
```

Imagen 16: Pseudocódigo "Depósitos Bancarios" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Las primeras tres líneas de código crean un título a manera de encabezado para el programa. Luego, las líneas 5 a 10 se encargan de solicitar los datos de entrada al usuario, tales como: el nombre completo del cliente, el número de cuenta bancaria beneficiada, y la cantidad a depositar. Posteriormente, las siguientes tres líneas (11-13) muestran un menú de opciones a el usuario para que seleccione la modalidad de depósito a realizar (efectivo o cheque), y por consiguiente la opción seleccionada queda vinculada a la variable **modoDeposito** (línea 14).

Ahora bien, para comprobar el modo de depósito seleccionado se hace uso de una estructura de selección múltiple, misma en la que aparecen dos casos disponibles (**1** para cheque, **2** para efectivo) y una opción por defecto (en caso de equivocación por parte del usuario). Si el usuario selecciona la opción **1** (modo cheque), se ejecuta el bloque de instrucciones (líneas 18-25) que solicitan el ingreso del número de cheque asociado e imprimen el comprobante de depósito girado. Sin embargo, si el usuario selecciona la opción número **2** (efectivo). Se procede a comparar que el monto depositado no exceda a los **$15,000**; de ser así, no se aplica impuesto alguno y por consiguiente la cantidad depositada pasa íntegra a la cuenta beneficiada. Pero, de no cumplirse esta condición (de allí la presencia de la estructura "**Si Entonces Sino**"); entonces se tiene que determinar el excedente (línea 32), calcular su correspondiente **10%** (impuesto a retener, línea 34), así como a reajustar la cantidad depositada (impuesto debidamente restado del depósito original).

Finalmente, dentro del mismo bloque de instrucciones correspondientes al caso **2** (efectivo), se genera el ticket con las leyendas pertinentes, mismas que informan si hubo o no impuesto aplicado durante la transacción.

Capturas de Pantalla:

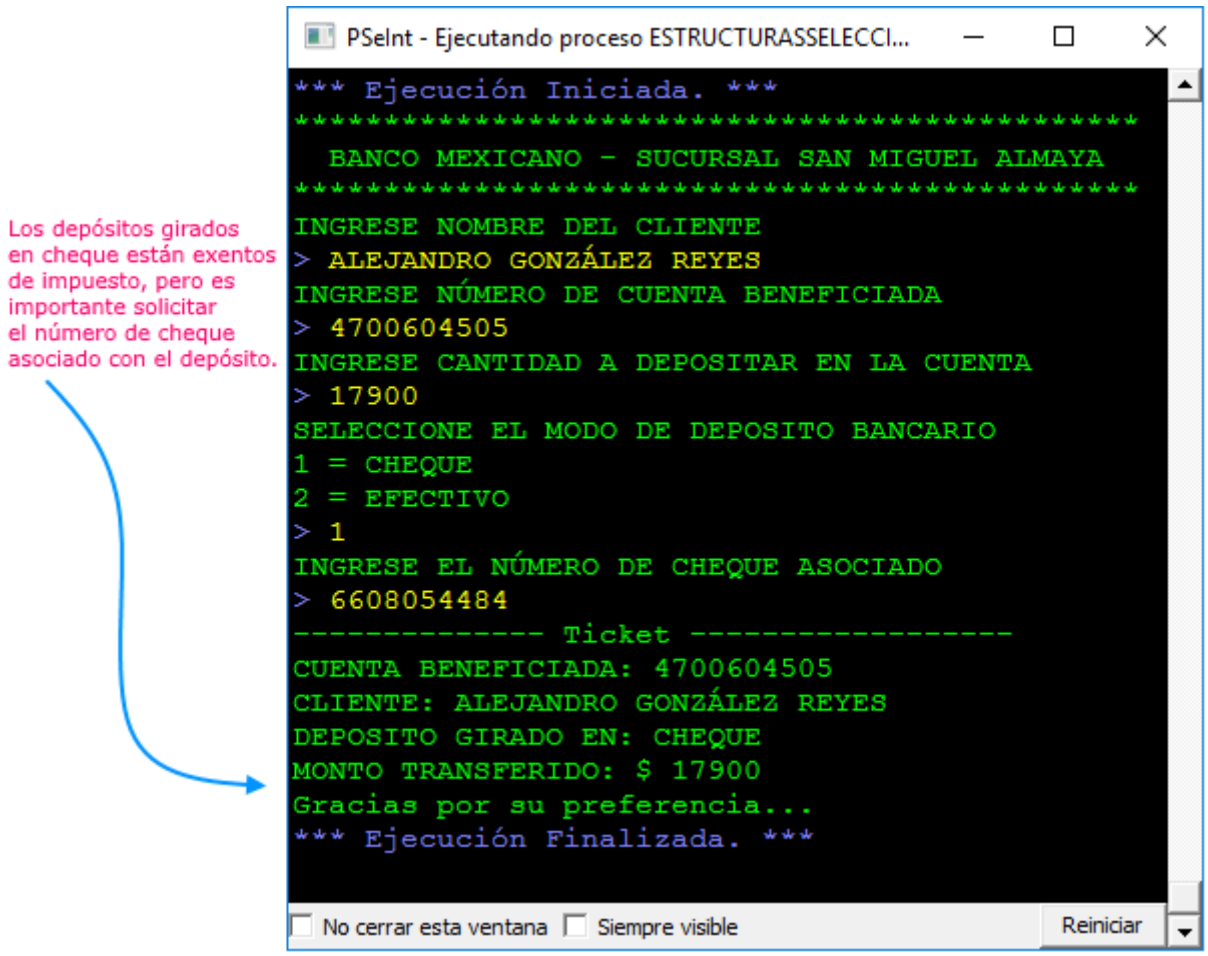


Imagen 17: Ejecución Pseudocódigo "Depósitos Bancarios" - Modo Cheque - Fuente: Diseñada por el autor.

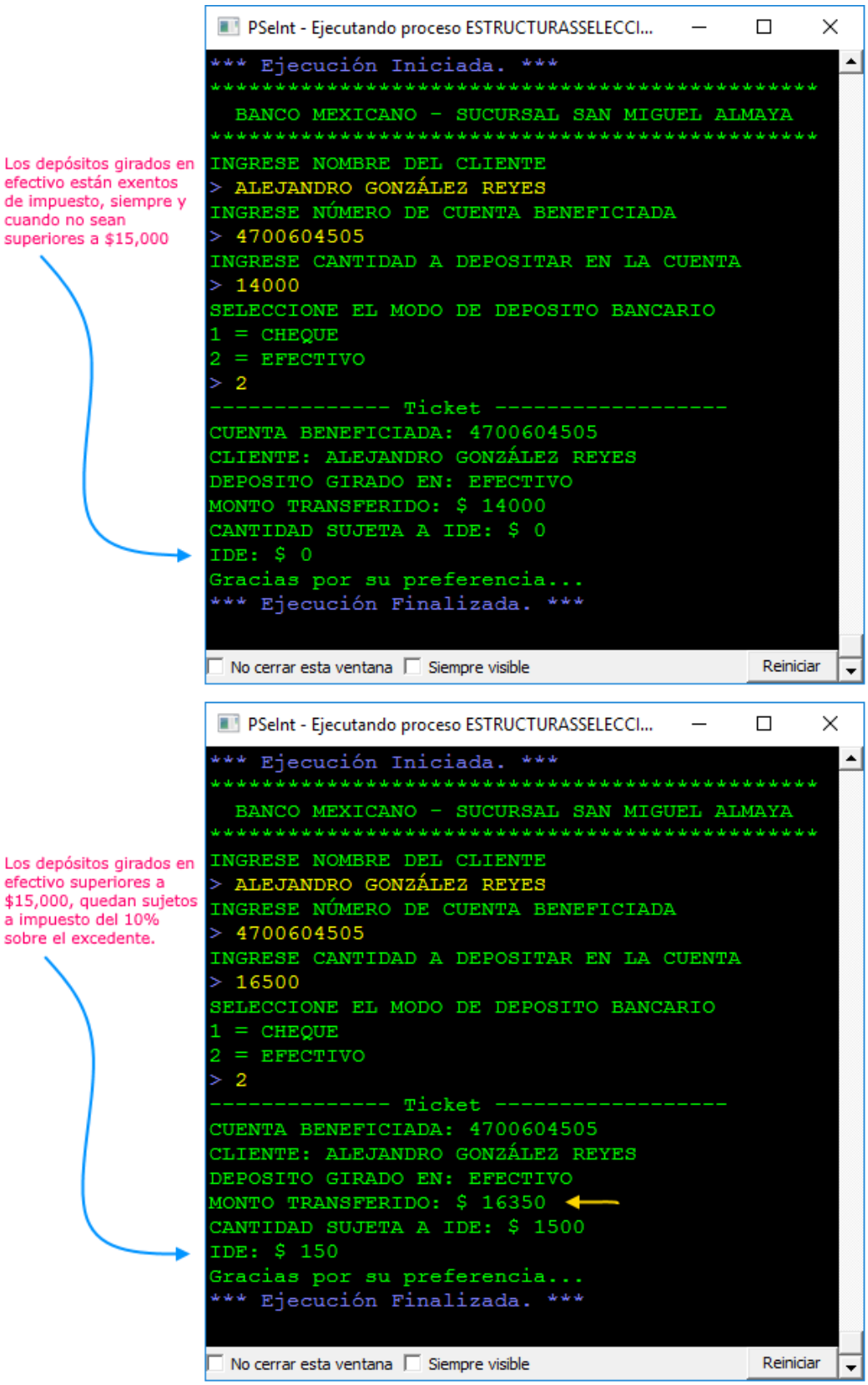


Imagen 18: Ejecución Pseudocódigo "Depósitos Bancarios" - Modo Efectivo - Fuente: Diseñada por el autor.

C5. Encadenar estructuras de control

Problema:

El Consejo de Participación Ciudadana le pide a usted, desarrollar un programa de computadora (escrito mediante Pseudocódigo) para el control de pago de impuestos pertenecientes a la comunidad rural de San Miguel Almaya.

Dichos impuestos reciben el nombre de faenas, y su imposición está relacionada con diferentes trabajos comunitarios (recolección de basura, pavimentación de calles, mantenimiento de la red de agua potable y drenaje, etc.) que deben llevarse a cabo durante todo el año dentro de dicha comunidad. Además, como cualquier otro trámite administrativo, el precio de las faenas está sujeto a cambios sin previo aviso al inicio de cada periodo (su precio se determina con base en un mes, pero el pago debe realizarse por año); por lo que el Consejo de Participación Ciudadana debe realizar un cálculo previo antes de comenzar a cobrar, ya que el precio de cada faena está relacionado con la publicación de una tarifa anual que impide a su vez que los ciudadanos realicen sus pagos por adelantado.

A su vez, estos impuestos sólo son aplicables a personas que han cumplido la mayoría de edad, y su cálculo debe estar determinado a partir de los 18 años en adelante, es decir, si un ciudadano tiene actualmente 22 años y por motivos desconocidos no ha cumplido con el pago de dicho puesto, éste solamente puede pagar como máximo 5 años (18, 19, 20, 21, 22) o una fracción de estos (uno o dos años).

Finalmente, el programa debe mostrar un informe por pantalla donde indique claramente el nombre del ciudadano, el total a pagar por concepto de faenas, el número de faenas pagadas, así como el último año de pago.

Solución PSeInt:

```
Proceso combinacionEstructurasSeleccionRepeticion
    Imprimir "--- CONSEJO DE PARTICIPACIÓN CIUDADANA - COPACI ---"
    Imprimir ""
    ///--- Datos de Entrada
    Imprimir "Ingrese nombre completo del ciudadano"
    Leer nombreCiudadano
    Imprimir "Ingrese edad del ciudadano"
    Leer edadCiudadano
    Imprimir "Ingrese número de años a pagar"
    Leer numeroPagos
    Imprimir "Ingrese último año de pago"
    Leer ultimoPago
    Imprimir "Ingrese año actual"
    Leer anioActual
    ///--- Proceso
    //Verificando si el ciudadano es mayor de edad
    Si edadCiudadano >= 18 Entonces
        //Verificando si los pagos por girar exceden al año actual
        Si (ultimoPago + numeroPagos) > anioActual Entonces
            Imprimir "INFORME: No se pueden girar pagos por adelantado"
        Sino
            pagoTotal = 0
            //Solicitando al operador del sistema el precio por faena estipulado
            //para cada año correspondiente
            Para contador = (ultimoPago + 1) Hasta (ultimoPago + numeroPagos) Hacer
                Imprimir "Ingrese precio por faena año " , contador
                Leer precioFaena
                //Acumulando el precio total de las faenas para este año
                pagoTotal = pagoTotal + (precioFaena * 12)
            FinPara
            ///--- Salida de Información
            Imprimir ""
            Imprimir "Ciudadano: " , nombreCiudadano
            Imprimir "Total a Pagar: $ " , pagoTotal
            Imprimir "Número de Faenas: " , (numeroPagos * 12)
            Imprimir "Pago Girado Hasta: " , (ultimoPago + numeroPagos)
        FinSi
    Sino
        Imprimir "INFORME: Usted no tiene mayoría de edad"
    FinSi
FinProceso
```

Imagen 19: Pseudocódigo "COPACI" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Las líneas 5-14 del Pseudocódigo anterior se encargan de solicitar al operador del sistema el nombre del ciudadano, su edad, el número de pagos que desea realizar (es decir, ¿cuántos años de faenas desea pagar?), el último pago realizado en ventanilla (dicho de otra forma, ¿hasta qué año tiene al corriente el pago de faenas?), así como el año actual (para verificar más adelante que el ciudadano no realice pagos por adelantado).

Una vez que el programa cuenta con los datos necesarios para poder operar, la estructura de selección declarada en las líneas 17-40 verifica que el ciudadano cumpla con la mayoría de edad (es decir, tenga 18 años o más) para proceder con el trámite de pago de impuestos; en caso contrario, se emite un informe con el error acontecido.

Ahora bien, en caso de que el ciudadano tenga la mayoría de edad, nuevamente se procede a verificar (mediante una estructura de selección declarada en las líneas 19-37) si el número de pagos exceden al máximo permitido, en caso de ser así, se emite un informe que especifica que es imposible hacer pagos girados por adelantado (que superen al año actual); de lo contrario, se comienza a solicitar al operador del sistema el precio por faena acordado para cada año correspondiente. En este sentido, resulta lógico hacer uso de una estructura de repetición (líneas 25-30) debido a que las operaciones son las mismas (solicitar el precio por faena al usuario), pero los datos con los que se trabaja pueden ser diferentes (precio de faena diferente para cada año).

Sumado a lo expuesto, para determinar el pago generalizado de dichos impuestos, la línea de código 29 declara el uso de un acumulador para

acarrear el precio de las faenas pertenecientes a un periodo (un ciclo en la estructura de repetición) y sumarlo a los siguientes.

Finalmente, se emite un comprobante de pago mismo que específica el nombre completo del ciudadano, el pago total a realizar por concepto de impuestos, el número de faenas cubiertas (expresada en meses), así como el último año de pago.

Capturas de Pantalla:

El ciudadano decide pagar 4 años de faenas, por tanto, su expediente queda cubierto hasta el año 2007

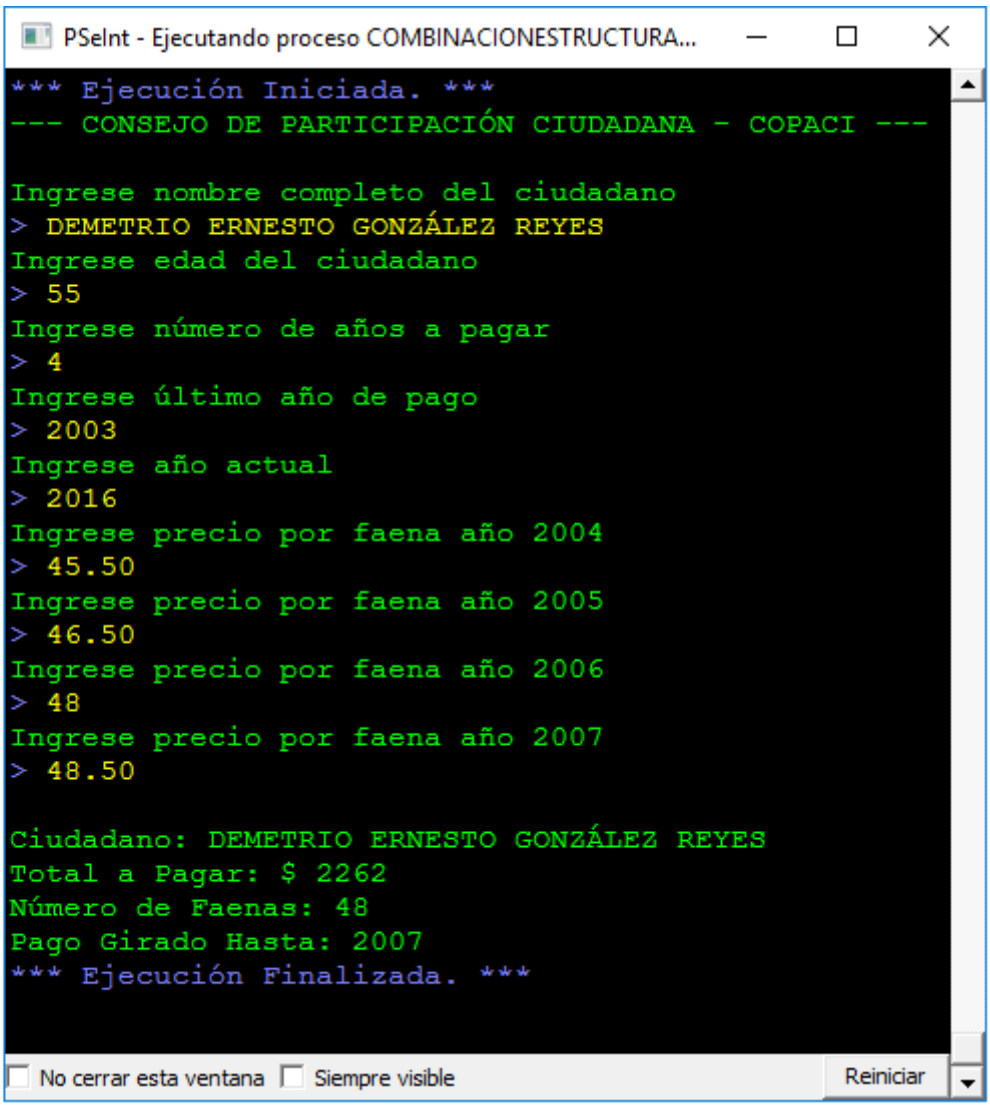
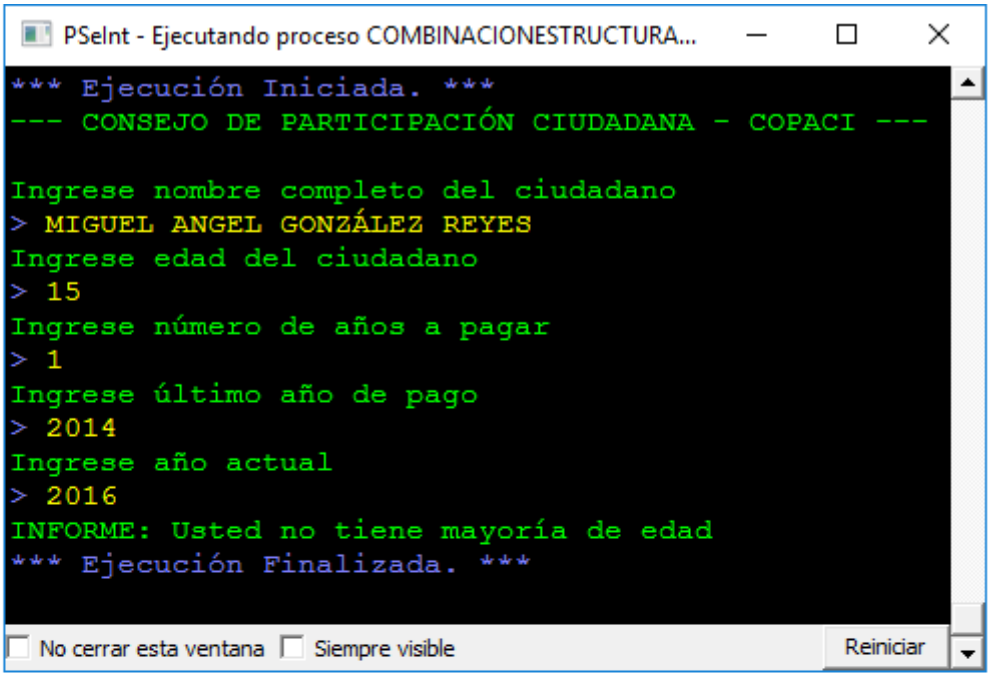


Imagen 20: Ejecución Pseudocódigo "COPACI" - Pagos normales - Fuente: Diseñada por el autor.

El ciudadano no cumple con la mayoría de edad para hacer frente a dicha obligación



El ciudadano excede el límite permitido de faenas pagadas, de aceptar dicho pago, el ciudadano cubriría hasta el año 2019, aspecto que no está permitido

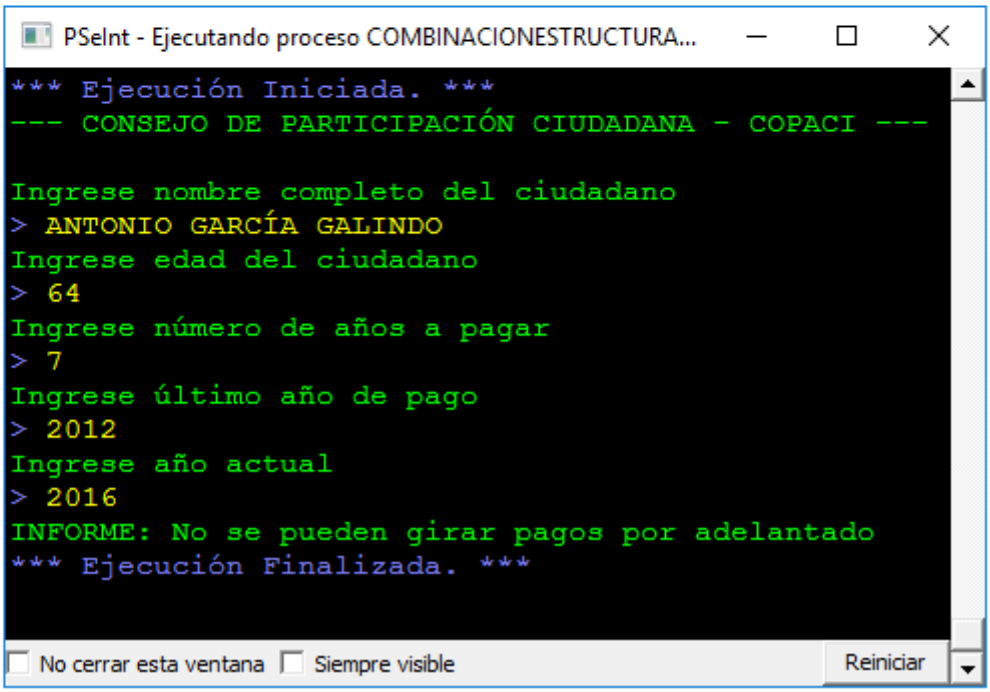


Imagen 21: Ejecución Pseudocódigo "COPACI" - Ciudadanos menores de edad y Pagos por adelantado - Fuente: Diseñada por el autor.

C6. Recorrer el contenido interno de las estructuras de repetición en formato regresivo

Problema:

Desarrolle un programa de computadora (escrito mediante Pseudocódigo) para llevar a cabo una cuenta regresiva con duración de 10 segundos. Durante el procesamiento, es importante que se notifique al usuario acerca del tiempo (en segundos) que falta antes de que el programa finalice; de lo contario, es decir, si el tiempo ha expirado, se debe emitir un mensaje acerca de lo sucedido (El tiempo se ha agotado).

De acuerdo con la problemática expuesta, se trata de una estructura de repetición que debe ejecutar un conjunto de instrucciones basadas en el tiempo, es decir, cada segundo. Además, el recorrido se debe llevar a cabo de manera inversa, de modo que su inicio debe ser a partir del segundo 10 y finalizar en el segundo 0. Para ello, es importante declarar un valor de paso personalizado que permita satisfacer dichas necesidades, es decir, para el caso de éste ejemplo, su valor debe ser de -1, puesto que, por cada iteración, el tiempo "se reduce en un segundo" y el programa finaliza cuando éste llega a cero.

Solución PSeInt:

```
1  Proceso repeticionInversa
2      Imprimir "--- PROGRAMA CUENTA REGRESIVA ---"
3      Imprimir ""
4      //Ciclo que se repite 10 veces en forma inversa,
5      //es decir, el valor de la variable tiempo reduce en 1
6      //por cada repetición hasta llegar a 0.
7      Para tiempo = 10 Hasta 0 Con Paso -1 Hacer
8          //Si hay tiempo disponible se muestra por pantalla
9          Si tiempo > 0 Entonces
10             Imprimir "Faltan: " , tiempo , " seg."
11             //Instrucción para detener el flujo del programa
12             //por un segudo, después de eso el programa
13             //continúa con normalidad
14             Esperar 1 Segundo
15         Sino
16             //En caso de no existir tiempo...
17             Imprimir "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
18             Imprimir "El tiempo se ha agotado..."
19             Imprimir "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
20         FinSi
21     FinPara
22 FinProceso
```

Imagen 22: Pseudocódigo "Cuenta Regresiva" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Como se puede observar, las líneas 7 a 21 representan una estructura de repetición donde la variable de control "**tiempo**" se inicializa en **10** unidades, mismas que representan los segundos para nuestro programa de cuenta regresiva. Además, se indica que las instrucciones declaradas en el cuerpo de la estructura (líneas 8-21) se deben repetir hasta que la variable de control obtenga un valor de **0**. Para ello, se hace imprescindible la declaración de un valor de paso que permita reducir en cada iteración el valor de la variable de control, el cual permita pasar de las **10** unidades originales a **0** unidades finales.

Sumado a lo expuesto, en las líneas de código 9 a 20 se lleva a cabo una comprobación del valor actual referenciado en la variable de control "**tiempo**", en caso de que éste sea superior a **0** (es decir, si aún hay tiempo) se imprime por pantalla el tiempo que falta para que éste finalice (línea 10) y, en seguida, se procede a detener por un segundo (línea 14) el flujo del programa (de no hacerlo el programa funciona correctamente, pero la cuenta regresiva no estaría relacionada con base en el tiempo, lo que da a lugar que su impresión en pantalla sea de forma inmediata); en caso contrario, se imprime un mensaje al usuario donde se le informa que el tiempo se ha agotado (línea 18) y, por consiguiente el programa finaliza con la ejecución de su tarea.

Llegado a este punto, es importante mencionar que la instrucción declarada en la línea 14, "**Esperar 1 Segundo**", puede personalizarse para pausar la ejecución del Pseudocódigo durante un intervalo de tiempo predefinido. Es decir, después de la palabra clave **Esperar**, se puede indicar la longitud y la unidad de tiempo de dicho intervalo. Por ejemplo:

- **Esperar** 3 **Segundos**
- **Esperar** 400 **Milisegundos**

Captura de Pantalla:

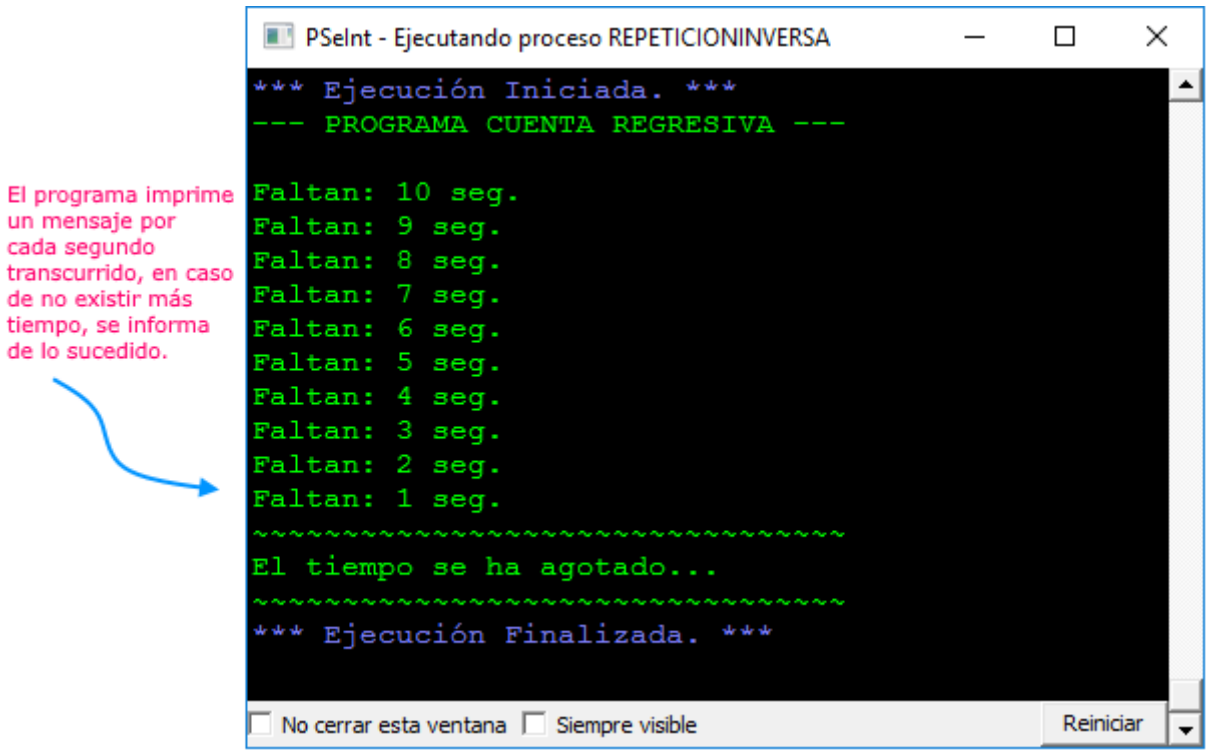


Imagen 23: Ejecución Pseudocódigo "Cuenta Regresiva" - Fuente: Diseñada por el autor.

C7. Declarar diferentes estructuras de control (encadenadas y no encadenadas) en un mismo proyecto

Problema:

Desarrolle un programa de computadora (escrito en Pseudocódigo) para determinar el monto total de ventas realizadas por un empleado a lo largo de 15 días. En caso de que sus ventas sean nulas, es decir $0.00, se debe mostrar un mensaje por pantalla que informe al empleado sobre su situación laboral (en este caso, que pronto será despedido); de lo contrario, se debe calcular el promedio de ventas totales para verificar si tiene derecho a que se le otorgue un bono de productividad sobre su pago quincenal (que es de $2,000 fijos). Para ello, por políticas internas de la empresa se establece que un empleado es acreedor a un bono quincenal de $1,800 sólo si sus ventas por día son superiores a $4,000.

Finalmente, el programa debe generar un informe detallado (por pantalla) con el nombre del empleado, las ventas totales, el promedio general de ventas, el bono de productividad y el pago quincenal.

Con base en la problemática expuesta, se hace evidente la necesidad de declarar tres estructuras de control para la solución del problema. Una estructura de repetición "Mientras" para la solicitud de las ventas (puesto que se desconoce en un primer momento el número de ventas realizadas por el empleado durante la quincena). Una estructura de selección "Si Entonces Sino" para determinar si el empleado tiene continuidad dentro de la empresa y una estructura de selección "Si Entonces" para verificar si el empleado tiene derecho o no a su correspondiente bono de productividad.

Solución PSeInt:

```
Proceso controlDePagosEmpleados
    Imprimir "SUCURSAL EL BUEN VESTIR S.A. DE C.V."
    Imprimir "      MÓDULO - PAGO DE SUELDOS      "
    Imprimir ""
    ///--- Datos de Entrada
    Imprimir "Ingrese nombre del empleado"
    Leer nombreEmpleado
    //Declaración de acumulador
    ventasTotales = 0
    //Primer cuestionamiento: se asigna de respuesta por defecto
    //para entrar a la estructura de repetición Mientras, es decir,
    //no se le pregunta al usuario si desea o no entrar
    registrarVenta = "si"
    ///--- Proceso de Información
    Imprimir ""
    //Repetir mientras el usuario desee registrar una venta
    Mientras registrarVenta == "si" O registrarVenta == "SI" Hacer
        Imprimir "Ingrese venta"
        Leer ventaParcial
        ventasTotales = ventasTotales + ventaParcial
        //Segundo cuestionamiento: si el usuario desea registrar
        //otra venta para el mismo empleado, el ciclo se repite,
        //de lo contrario, sale del ciclo.
        Imprimir "¿Desea agregar otra venta? (si/no)"
        leer registrarVenta
    FinMientras
    //Si las ventas totales son cero, entonces el empleado registrado
    //no vendió nada durante la quincena. Por tanto se le tiene que
    //despedir.
    Si ventasTotales == 0 Entonces
        Imprimir ""
        Imprimir "Lo siento, en breve usted será despedido."
    Sino
        //En caso de haber vendido algo durante dicho periodo...
        promedioVentas = ventasTotales / 15
        bonoProductividad = 0
        //Verificamos si el promedio de ventas exceden a los $4000, para
        //obsequiarle un bono de productividad adicional a su pago quincenal
        Si promedioVentas > 4000 Entonces
            bonoProductividad = 1800
        FinSi
        ///--- Salida de Información
        Imprimir ""
        Imprimir "-------- COMPROBANTE DE PAGO DE SUELDOS --------"
        Imprimir "Nombre del Empleado:   " , nombreEmpleado
        Imprimir "Ventas Totales:        $ " , ventasTotales
        Imprimir "Promedio de Ventas:    $ " , promedioVentas
        Imprimir "Bono de Productividad: $ " , bonoProductividad
        Imprimir "Pago Quincenal         $ " , (2000 + bonoProductividad)
    FinSi
FinProceso
```

Imagen 24: Pseudocódigo "Sueldos y Salarios" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

En el ejemplo anterior, las líneas 2-4 hacen referencia a información que se imprime como parte de la cabecera del programa. Las líneas 6-7 solicitan al operador del sistema el nombre del empleado para el cual se debe girar dicho pago quincenal. La línea de código número 9 inicializa la variable que fungirá como acumulador dentro de la estructura de repetición (líneas 17-26), es decir, su labor consiste en hacer referencia a la sumatoria de las ventas realizada por el empleado durante la quincena.

Sumado a lo expuesto, en la línea de código 13 se lleva a cabo la asignación de un valor por defecto "**si**" en la variable "**registrarVenta**"**,** la cual tiene como propósito evitar que se pregunte innecesariamente al usuario si desea registrar las ventas del empleado (primer cuestionamiento). De esta forma se valida automáticamente la condición declarada en la estructura de repetición (línea 17) y, por consiguiente, se logra ejecutar la primera iteración.

Una vez situados en este punto, las líneas 18-19 solicitan al usuario el ingreso de una venta realizada por el empleado, en seguida se acumula dicho valor en las ventas totales (línea 20) y, finalmente antes de terminar el ciclo, se cuestiona al usuario si desea agregar otra venta (líneas 24-25). Con base en la respuesta seleccionada "**si** o **no**", las acciones declaradas dentro de la estructura repetitiva deben o no volver a ejecutarse.

Ahora bien, una vez terminado el registro de ventas (finalización de la estructura Mientras). La estructura de selección declarada en las líneas 30-50 verifica si el empleado no ha realizado venta alguna durante la quincena, en caso de ser así, éste prácticamente ha quedado despedido (líneas 31-32); de lo contrario, se analiza su promedio de ventas (línea

35) y se coteja con la política de bonos de productividad establecidos por la propia empresa (líneas 39-41), misma que establece si el promedio de ventas por día es superior a **$4,000**, el empleado se hace acreedor a un bono adicional de **$1,800** (línea 40) sobre su pago quincenal.

Finalmente, las líneas de código 43-49 muestran por pantalla un informe detallado relativo al pago generado.

Capturas de Pantalla:

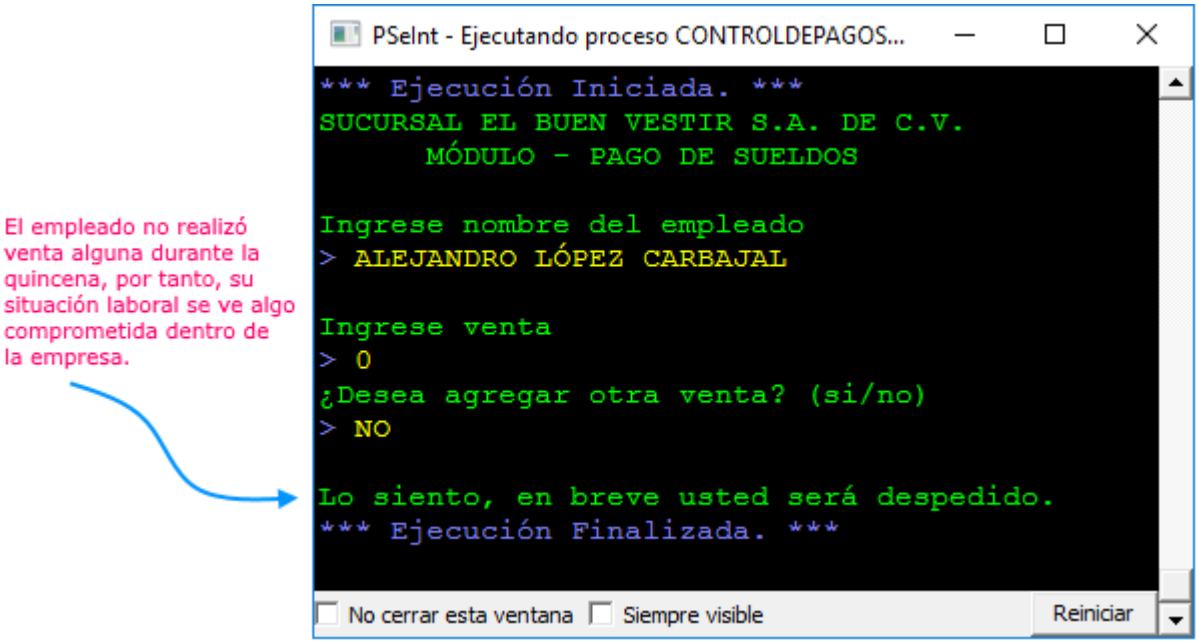


Imagen 25: Ejecución Pseudocódigo "Sueldos y Salarios" - Ventas nulas - Fuente: Diseñada por el autor.

El empleado realizó algunas ventas durante la quincena, sin embargo, el promedio de estas es insuficiente para ser acreedor a un bono de productividad.

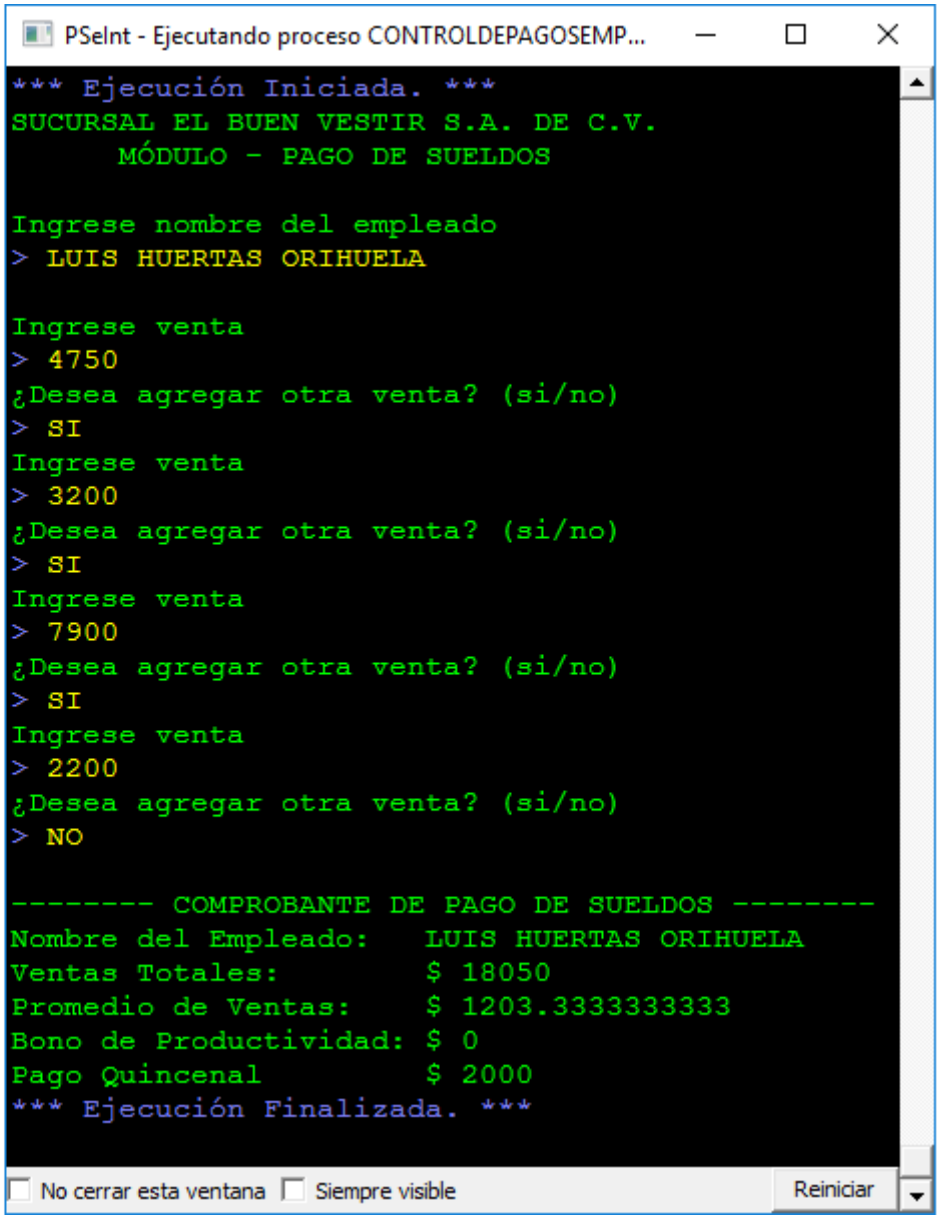


Imagen 26: Ejecución Pseudocódigo "Sueldos y Salarios" - Bono de productividad nulo - Fuente: Diseñada por el autor.

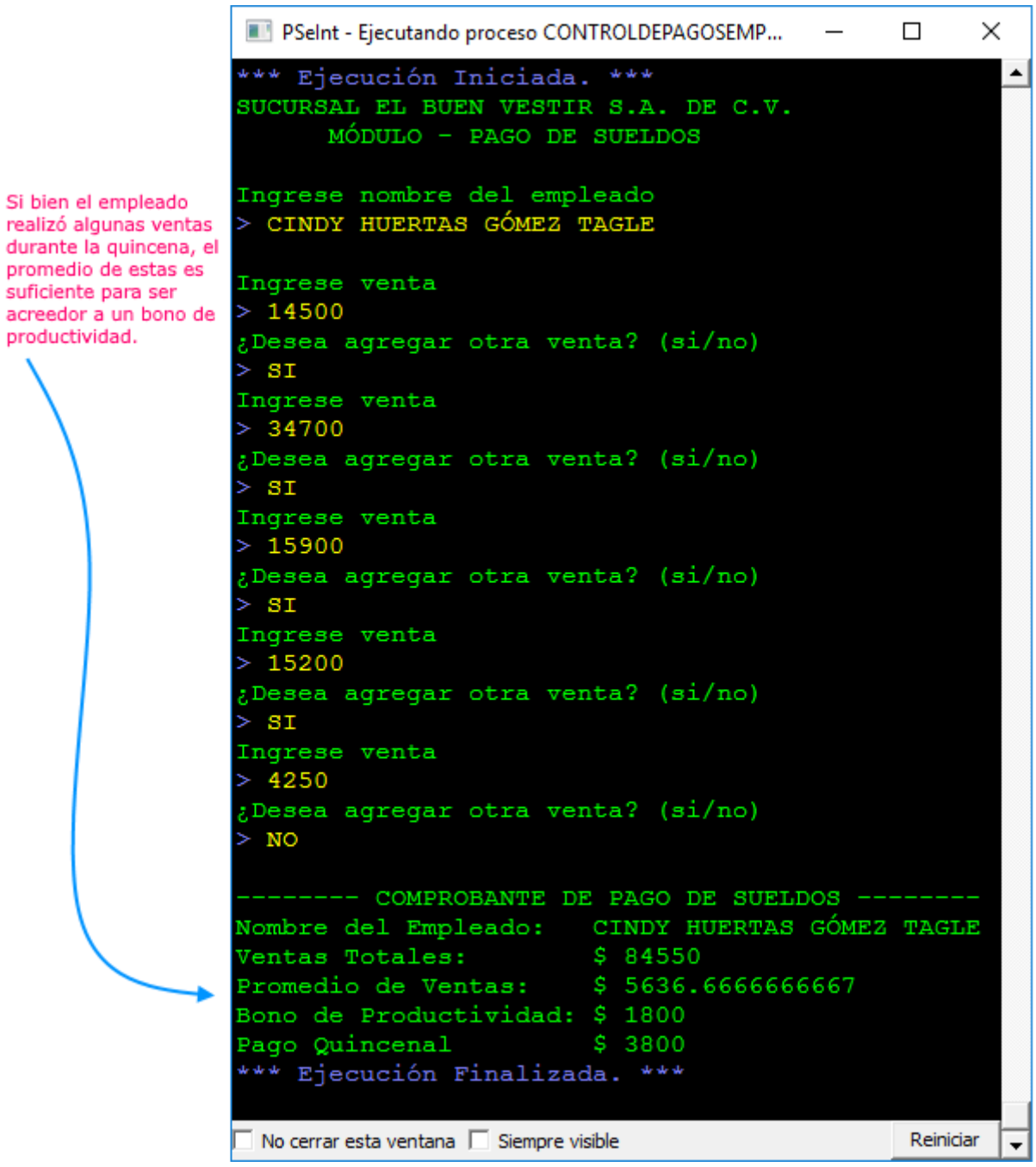


Imagen 27: Ejecución Pseudocódigo "Sueldos y Salarios" - Bono de productividad incluido - Fuente: Diseñada por el autor.

C8. Repeticiones definidas en una estructura del tipo "Mientras"

Problema:

Desarrolle un programa de computadora (escrito mediante Pseudocódigo) para determinar el factorial de un número proporcionado por el usuario. Entiéndase por factorial de un número como el producto de todos los números enteros positivos desde N hasta 1. Por ejemplo:

- **Factorial de 5:** 5 x 4 x 3 x 2 x 1 = 120
- **Factorial de 7:** 7 x 6 x 5 x 4 x 3 x 2 x 1 = 5040
- **Factorial de 3:** 3 x 2 x 1 = 6
- **Factorial 1:** 1

Además, por convención matemática el valor de 0 factorial debe definirse como 1 (Factorial de 0 = 1) y para el caso de números negativos, éste no tiene sentido.

Solución PSeInt:

```
1  Proceso programaFactorial
2      Imprimir "---- PROGRAMA FACTORIAL DE UN NÚMERO ----"
3      Imprimir ""
4      ///--- Datos de Entrada
5      Imprimir "Ingrese Número"
6      Leer numeroUsuario
7      Imprimir ""
8      ///--- Proceso de Información
9      //En caso de que el número seleccionado sea negativo...
10     Si numeroUsuario < 0 Entonces
11         Imprimir "Informe: "
12         Imprimir "Imposible obtener el factorial de un número negativo"
13     Sino
14         //En caso de que el número sea 0, entonces el factorial es 1
15         Si numeroUsuario == 0 Entonces
16             Imprimir "El Factorial es: 1"
17         Sino
18             //Para un número positivo diferente de 0, se obtiene en
19             //primer lugar el número positivo anterior a este...
20             contador = (numeroUsuario - 1)
21             //Iteramos siempre y cuando el valor de éste sea superior a 0
22             Mientras contador > 0 Hacer
23                 //Proceso de multiplicación -- Acumulando resultado factorial
24                 numeroUsuario = numeroUsuario * contador
25                 //Variable de cuenta regresiva --- (multiplicador) N x 3 x 2 x 1
26                 contador = contador - 1
27             FinMientras
28             Imprimir "El Factorial es: " , numeroUsuario
29         FinSi
30     FinSi
31 FinProceso
```

Imagen 28: Pseudocódigo "Factorial" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Observe que las líneas 5-6 se encargan de solicitar al usuario el ingreso de un número para determinar su correspondiente factorial. Luego la estructura de selección declarada en las líneas 10-30, verifica que dicho número sea inferior a **cero**, es decir, negativo. En caso de ser así, el programa finaliza con la leyenda "**Imposible obtener el factorial de un numero negativo**"; en caso contrario, se ejecuta el bloque de instrucciones declarado en las líneas 14-29, mismo que implementa

nuevamente una estructura de selección. Sin embargo, esta se encarga de verificar que el número proporcionado por el usuario contenga el valor de **cero**, en caso afirmativo, imprime un mensaje de notificación al usuario e informa de que "**El factorial de cero es 1**"; en caso contrario, se procede a ejecutar el bloque de instrucciones declarado en las líneas 18-28.

Sumado a lo expuesto, si por algún motivo durante el proceso de ejecución el flujo del programa se encuentra dentro de las líneas 18-28, se debe a que el usuario ha ingresado un valor numérico superior a **cero**. En este sentido, se debe proceder a multiplicar todos los números positivos que se encuentran desde el número proporcionado por el usuario (en la definición anterior este valor sería N) hasta llegar a **1**.

Con base en los supuestos anteriores, resulta lógico conocer en primer lugar el valor numérico inmediato anterior al proporcionado (es decir, si el usuario ha decidido obtener el factorial de 5, entonces el valor inmediato anterior sería 4. Para 9 factorial, este valor sería 8, y así sucesivamente), después de todo este será el primer valor con el cual se tendrá que comenzar a multiplicar (ver línea 20).

Ahora bien, el proceso de multiplicación en formato regresivo, sólo es válido siempre y cuando el último operando sea **1**, es por ello que en la declaración de la estructura de repetición (líneas 22-27) se incluye una condicional, la cual específica que las instrucciones declaradas dentro de su cuerpo (líneas 23-26) sólo deben ejecutarse si el valor asignado en la variable **"contador"** es superior a **cero**.

Bajo este mismo orden de ideas, para almacenar y acarrear el producto de los operandos obtenido en cada iteración, se hace uso de un acumulador, el cual durante la primera iteración hace referencia al valor original

proporcionado por el usuario (línea 24). Además, para prevenir la presencia de un ciclo infinito durante la ejecución, antes de finalizar un ciclo completo, se modifica el valor de la variable **"contador"** (línea 26). Esto permite que se lleve a cabo la multiplicación en forma regresiva (lógica factorial), de modo que sea posible detener toda acción una vez que su valor ha llegado a **1** (si el valor llega a cero, las acciones declaradas en la estructura de repetición no se vuelven a ejecutar).

Finalmente, en la línea de código número 28 se informa al usuario del resultado obtenido, es decir, el factorial del número ingresado.

Capturas de Pantalla:

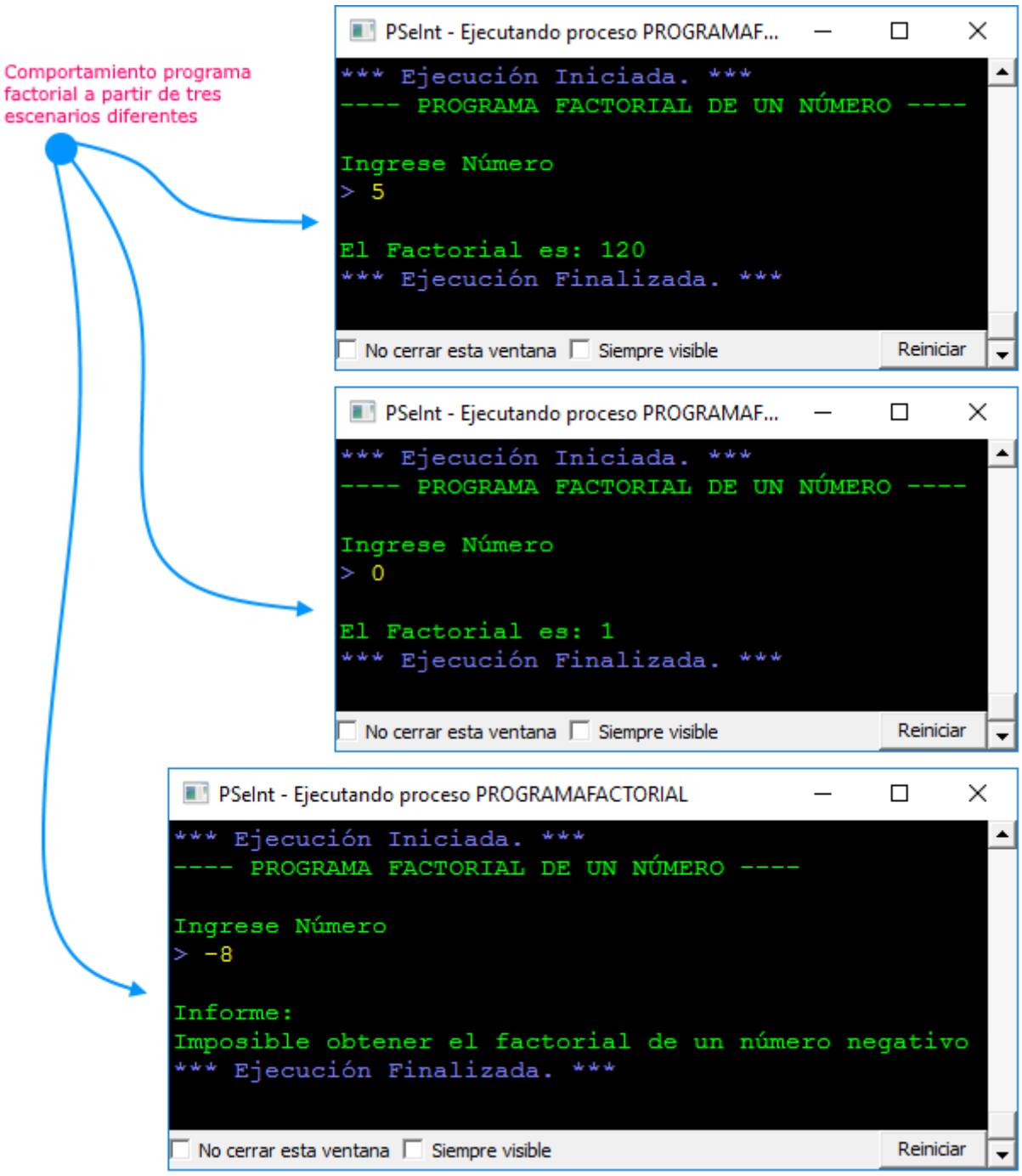


Imagen 29: Ejecución Pseudocódigo "Factorial" - Fuente: Diseñada por el autor.

C9. Declarar una estructura del tipo "Repetir Mientras Que" como parte de un proceso intermedio para la solución de un problema

Problema:

Desarrolle un programa de computadora (escrito mediante Pseudocódigo) que solicite una cantidad indeterminada de números aleatorios al usuario, y proceda a determinar: la suma, promedio y valor máximo de aquellos valores que por su naturaleza puedan ser considerados como números pares.

Finalmente, en caso de que el usuario ingrese un valor numérico de cero, el programa debe terminar la solicitud de información e imprimir por pantalla los resultados correspondientes.

Solución PSeInt:

```
1  Proceso programaNumerosPares
2      Imprimir "-----------------------------------------------------"
3      Imprimir "PROGRAMA NÚMEROS PARES - SUMA | PROMEDIO | PAR MAYOR"
4      Imprimir "-----------------------------------------------------"
5      Imprimir "0) Salir"
6      Imprimir ""
7      sumaPares = 0
8      contarPares = 0
9      numMayor = 0
10     mostrarInforme = FALSO
11     Repetir
12         Imprimir "Ingrese número:" Sin Saltar
13         leer numUsuario
14         Si (numUsuario % 2) == 0 Y numUsuario != 0 Entonces
15             sumaPares = sumaPares + numUsuario
16             Si numUsuario > numMayor Entonces
17                 numMayor = numUsuario
18             FinSi
19             contarPares = contarPares + 1
20             mostrarInforme = VERDADERO
21         FinSi
22     Mientras Que numUsuario != 0
23     Imprimir "Procesando información, espere un momento..."
24     Esperar 2 segundos
25     Imprimir ""
26     Si mostrarInforme == VERDADERO Entonces
27         Imprimir "Suma Números Pares:     " , sumaPares
28         Imprimir "Promedio Números Pares: " , (sumaPares / contarPares)
29         Imprimir "Número Par Mayor:       " , numMayor
30     Sino
31         Imprimir "Informe: Usted no registró números pares."
32     FinSi
33 FinProceso
```

Imagen 30: Pseudocódigo "Reporte Números Pares" - Fuente: Diseñada por el autor.

Explicación Pseudocódigo:

Del ejemplo anterior, se tiene que las líneas 2-6 imprimen información de cabecera referente a las capacidades del programa. Posteriormente, para almacenar la suma de los números pares y la cantidad de éstos (necesario para obtener el promedio) se hace indispensable contar con la declaración

de algunas variables cuyo funcionamiento interno sirva como el de un acumulador (líneas 7-8). Además, para determinar el número par mayor ingresado por el usuario, es necesario que se le compare constantemente con un valor base considerado por defecto (línea 9). Dicho de otra forma, si el valor numérico ingresado (por el usuario) es mayor al valor tomado como base, el valor base se sobre-escribe (por el valor numérico ingresado), en caso contrario, su valor no se modifica.

Llegado a este punto, surge la siguiente incógnita: ¿qué pasa si el usuario jamás ingresa un valor numérico par diferente de cero durante el proceso? Para responder a esta situación, se ha declarado la instrucción de la línea 10, la cual permite verificar si es pertinente o no imprimir los resultados en pantalla (evita que se consuman recursos innecesarios).

Sumado a lo expuesto, para solicitar el ingreso de los valores numéricos al usuario de forma indeterminada, se hace uso de una estructura de repetición del tipo "**Repetir Mientras Que**" (líneas 11-22), misma que finaliza cuando el valor de la variable de control **"numUsuario"** es idéntico a **cero** (línea 22). Internamente, las líneas 12-13 se encargan de solicitar un número por cada iteración, el cual más adelante (líneas 14-21) es verificado para comprobar si es de naturaleza par (sólo si es divisible entre 2, es decir, si el residuo o sobrante de la división entre 2 es cero) **Y** a su vez es diferente de **cero** (si bien el número cero puede ser considerado como par, no tiene caso contabilizarlo, ya que es el número seleccionado para finalizar el proceso).

Ahora bien, en caso de que el valor ingresado sea considerado como número par, éste es considerado por el acumulador de números pares (línea 15). En seguida, se procede a verificar si éste nuevo número es mayor al anterior registrado (líneas 16-18), y se contabiliza para que más

adelante sea posible obtener su correspondiente promedio (línea 19). Finalmente, al existir evidencia de un número par registrado, se indica que debe mostrarse un informe detallado (línea 20) con los resultados obtenidos al finalizar todo el proceso.

Bajo este mismo orden de ideas, si el valor ingresado por el usuario es un **cero** o un **número impar**, la estructura selectiva lo ignora, y delega el compromiso a la estructura repetitiva para que verifique si es pertinente repetir nuevamente el proceso (acciones declaradas en el cuerpo del ciclo) o en su defecto finalizarlo (línea 22).

Una vez finalizadas las tareas de solicitud de información, es decir, cuando el ciclo termina. El flujo del programa se detiene por dos segundos antes de mostrar el informe de resultados correspondientes (línea 24) al usuario. Pasado este tiempo de espera, las líneas 26-32 se encargan de comprobar si es necesario mostrar un informe detallado con los resultados obtenidos (si se registraron números pares), o en su defecto, se debe mostrar un aviso con los hechos acontecidos (en caso de que no se haya registrado un valor numérico par).

Seguramente en este punto del proceso, se esté preguntando ¿Qué función desempeña la instrucción "**Sin Saltar**" declarada en la línea 12 del Pseudocódigo anterior? Bueno, la explicación es sencilla. Por defecto PSeInt asigna una línea nueva para cada acción de lectura o escritura. Sin embargo, en ocasiones este tipo de comportamiento no es el adecuado para la presentación de información en nuestros programas. En este sentido, al declarar las palabras clave "**Sin Saltar**" se instruye al programa para que la próxima acción de lectura o escritura continúe en la misma línea de salida donde se quedó.

Capturas de Pantalla:

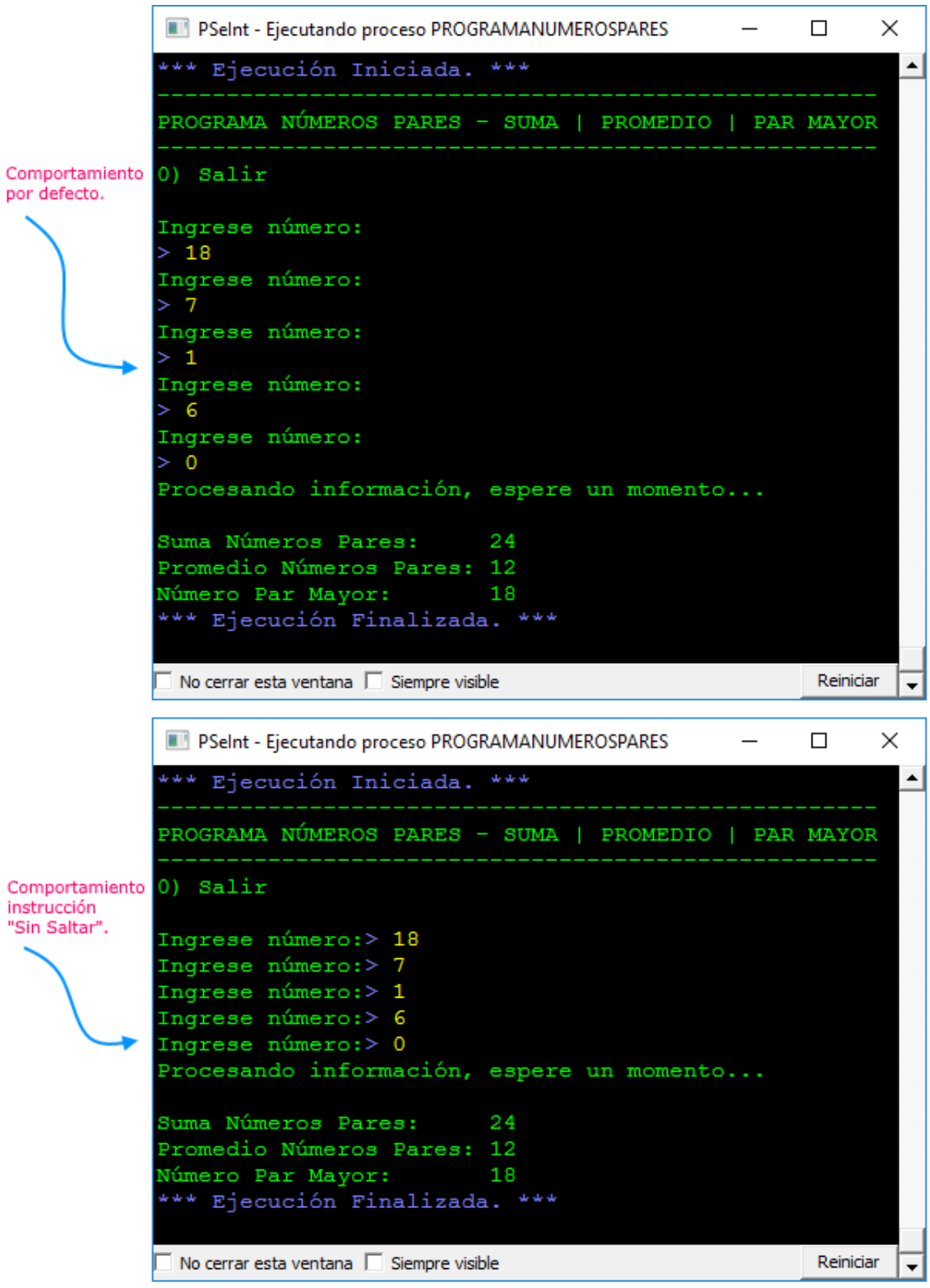


Imagen 31: Ejecución Pseudocódigo "Reportes Números Pares" - Fuente: Diseñada por el autor.