

# Ingeniería de Software I

## Práctica #2: Arreglos Vectores

### Objetivo

Verificar el uso de las estructuras de datos tipo arreglo, como alternativa para representación de una colección de variables del mismo tipo, que se operan con la misma semántica.

### Descripción

La práctica trata sobre el cálculo de impuestos anual a pagar por una persona física; omitiendo para sencillez de la práctica las consideraciones respecto a los pagos mensuales de ISR e IVA que se deberían tomar en cuenta para dicho cálculo.

Partiendo de la práctica #1, el concepto de “Ingreso” se aplica ahora a la cantidad acumulada de ingresos durante los 12 meses del año fiscal; igualmente aplica lo anterior para el concepto de “Gasto”

Para una utilización cómoda de la aplicación que ha de capturar los ingresos y gastos de cualquier mes (no de todos los meses a la vez y con posibilidad de modificar la información de cualquier mes) se hará uso de un menú con opciones.

#### CÁLCULO DE IMPUESTOS ANUAL

Menú principal:

1. Establecer mes para captura (mes actual es Enero)
2. Captura de ingresos
3. Captura de gastos
4. Mostrar lista de ingresos anual
5. Mostrar lista de gastos anual
6. Cálculo de impuestos anual
7. Salir

Opción:

Al elegir la opción #1 se muestre un listado de meses como el siguiente:

```
Establecer mes para captura
1) Enero
2) Febrero
3) Marzo
4) Abril
5) Mayo
6) Junio
7) Julio
8) Agosto
9) Septiembre
10) Octubre
11) Noviembre
12) Diciembre
Elige el mes (1-12): 9
Se ha establecido el mes de captura en Septiembre
Presione entrar para continuar . . .
```

...y una vez elegido el mes, se muestre un mensaje que corrobore el mes elegido seguido de una pausa.

Al elegir la opción #2 se vea como lo siguiente:

```
CÁLCULO DE IMPUESTOS ANUAL

Menú principal:
1. Establecer mes para captura (mes actual es Septiembre)
2. Captura de ingresos
3. Captura de gastos
4. Mostrar lista de ingresos anual
5. Mostrar lista de gastos anual
6. Cálculo de impuestos anual
7. Salir

Opción: 2

Captura de ingresos
Dame el ingreso del mes de Septiembre: 10000
Presione entrar para continuar . . .
```

...y una vez capturado el ingreso del mes en cuestión (reemplazando el valor almacenado anterior) se muestre una “pausa” (esperar por la tecla “entrar”).

Para la opción #3 funcionaría similar a la #2.

Al elegir la opción #4 se vea al menos tan claro como lo siguiente:

```
Mostrar lista de ingresos anual
Enero = 0.00
Febrero = 0.00
Marzo = 10000.00
Abril = 0.00
Mayo = 0.00
Junio = 0.00
Julio = 0.00
Agosto = 0.00
Septiembre = 10000.00
Octubre = 0.00
Noviembre = 0.00
Diciembre = 0.00
Presione entrar para continuar . . .
```

Para la opción #5 funcionaría similar a la #4.

Para la opción #6, suponiendo la suma de ingresos anuales por \$20000 y gastos anuales por \$2000, se vería como lo siguiente:

```
Cálculo de impuestos anual

***Tabla Recibo de Honorarios***
Ingresos                20000.00
(+) IVA                  3200.00
(=) Subtotal            23200.00
(-) Retención ISR       2000.00
(-) Retención IVA       2000.00
(=) Total               19200.00
***Tabla Ganancias***
Ingresos                20000.00
(-) Gastos              2000.00
(=) Ganancia Bruta      18000.00
(-) ISR                 1980.00
(=) Ganancia Neta       16020.00
***Tabla ISR***
ISR                     1980.00
(-) ISR Retenido        2000.00
(=) ISR a Pagar         -20.00
***Tabla IVA***
IVA                     3200.00
(-) Gastos IVA          320.00
(-) Retención IVA       2000.00
(=) IVA a Pagar         880.00
Presione entrar para continuar . . .
```

Al elegir la opción #7 del menú, la aplicación termine sin “pausa” alguna.

### Requerimientos Generales

1. Entregar archivo(s) fuente para aplicación de consola que cumpla(n) con los siguientes requerimientos.
2. Cumplir con los requerimientos generales de la Practical, salvo los cambios que sean absolutamente necesarios para cumplir con los siguientes requerimientos.
3. Declarar y definir después de las librerías y antes que todo lo demás, todas las constantes simbólicas que se necesiten.
4. A continuación de la declaración de variables, se encuentre la definición en tiempo de compilación para una estructura arreglo de cadenas para los nombres de los 12 meses del año.
5. Se utilice una estructura de datos arreglo para almacenar en ella los *ingresos* de cada uno de los meses del año; de la misma forma sea para los *gastos* mensuales; ambos arreglos sean independientes.
6. Imprimir en consola el título de la aplicación y a continuación un menú con las siete opciones citadas en la descripción.
7. Escribir comentarios en el código fuente destacando cada una de las siguientes partes del programa:
  - a) Declaración y definición de constantes
  - b) Declaración de variables para entrada de datos
  - c) Declaración de variables para cómputo
  - d) Entrada de datos
  - e) Cálculo de impuestos
  - f) Salida de resultados
8. Las partes del programa citadas anteriormente deben ser exclusivas, es decir, ningún cómputo del cálculo de impuestos debe llevarse a cabo en la sección de entrada de datos (captura de información) ni en la sección de salida de datos (presentación de resultados).
9. Para cada dato requerido del usuario imprimir la solicitud de información correspondiente (printf, cout, etc.) y a continuación esperar por la entrada de datos (scanf, cin, etc.) y solicitándole al usuario un dato a la vez.
10. Se realice un mostrado cíclico del menú mientras no se elija la opción #7 del menú.
11. La opción #1 del menú permita establecer el valor de una variable para el *mes de trabajo* a considerar para las opciones #2 y #3 del menú, mostrando a Enero como el mes #1.
12. Validar que el mes de trabajo siempre sea un mes válido. Validar es impedir que el programa utilice valores no válidos.
13. Al mostrar el menú, para la opción #1 se muestre el mes de Enero como mes a considerar para las opciones #2 y #3 del menú, y cada que cambie el mes, se muestre el *mes de trabajo* seleccionado.
14. La opción #2 del menú permita capturar/reemplazar (no acumulación, sino reemplazo) el ingreso de sólo el *mes de trabajo*.
15. La opción #3 del menú permita capturar/reemplazar (no acumulación, sino reemplazo) el gasto de sólo el *mes de trabajo*.

16. Durante la captura de un ingreso o un gasto, imprimir el mes para el cual se hará la captura antes de solicitar al usuario el monto.
17. La opción #4 muestre la lista de ingresos de todos los meses del año.
18. La opción #5 muestre la lista de gastos de todos los meses del año.
19. Para cada ingreso o gasto no definido por el usuario se muestre un valor de cero.
20. Al elegir la opción #6, definir una variable *ingresoTotal* en base a la suma acumulada de los ingresos mensuales según el punto 6 anterior; de la misma forma sea para una variable *gastoTotal*.
21. Al elegir la opción #6, definir las demás variables según práctica #1 en base las variables *ingresoTotal* y *gastoTotal*.
22. Al elegir la opción #6, imprimir en la consola las 4 tablas citadas en la descripción de la práctica #1 de forma que sus contenidos sean claros a la vista y los cálculos sean correctos para cualquier valor almacenable en tiempo de ejecución para las variables *ingresoTotal* y *gastoTotal*.
23. Todos los valores capturados y resultados impresos contemplen centavos; al imprimir los resultados estos se muestren alineados por el punto decimal (esto último hacerlo al menos en Lenguajes C/C++).
24. Al terminar de ejecutar una opción del menú, excepto “Salir”, hacer una “pausa” mostrando el mensaje “Presione entrar para continuar” y a continuación esperar por la tecla “entrar”.
25. Si el usuario elige una opción no contemplada en el menú se imprima “Opción no válida”, mostrar el mensaje “Presione entrar para continuar” y esperar por la tecla “entrar”
26. La opción “Salir” del menú funcione como se describió evitando usar la sentencia return o la instrucción exit() dentro del bucle que muestra el menú
27. El programa termine solo si se elige la opción “Salir” del menú
28. Para poder visualizar todo lo impreso por la aplicación, desde el título del programa o en su caso, desde el título de la opción de menú hasta lo último impreso para dicha opción, imprimir de tal manera que no se requiera redimensionar el tamaño por defecto de la consola (25 filas por 80 columnas) ni usar las barras de desplazamiento.
29. INDISPENSABLE PARA DERECHO A CALIFICACIÓN MAYOR A 5/100: A excepción de constantes simbólicas, todos los requerimientos se cubran usando solo una subrutina (main es una subrutina) y todas las variables se declaren dentro de main.

### Requerimientos de Archivo

30. Cumplir con los requerimientos de archivo de la práctica anterior.
31. Usar un nombre de archivo diferente para cada reporte de cálculo de impuestos que se solicite, donde al final del nombre de archivo se indique fecha y hora en formato “AAAA\_MM\_DD\_HH\_MM\_SS”. Conservar todos los archivos generados.

### Requerimientos para Puntos Extras

32. Entregar en el mismo comprimido carpetas diferentes para cada lenguaje:
  - a) Una nombrada “C” con los códigos fuente en C (ANSI);

- b) Una nombrada “C++” con los códigos fuente en C++;
- c) Una nombrada “Java” con los códigos fuente en Java;
- d) Una nombrada “C#” con los códigos fuente en C#;
- e) Una nombrada “English” con los códigos fuente escritos en su totalidad en inglés (solo para el lenguaje de programación de su preferencia).

...las primeras opciones se redacten absolutamente en español y la última absolutamente en inglés. Si incluye 2 carpetas con lenguajes OO, uno de ellos puede redactarse en inglés y sin necesidad de entregar la versión en español para ese lenguaje.

33. Para cada lenguaje cumplir con los requerimientos de archivo.

### **Criterios de Evaluación**

- Los establecidos en las “Reglas de Operación y Evaluación” del curso.
- Cumplir con la fecha límite de entrega citada en el Excel de Actividades.
- Calificación en base a cobertura de requerimientos.
- Cumplir con Requerimientos de Valor Agregado en Código Fuente (hasta el req. “Z”).
- Entrega en al menos un lenguaje: C++ (ANSI) o Java o C#.
- Es indispensable la entrega de un programa con variables, comentarios e impresiones a consola completamente en idioma Español.
- Entregar el programa preferentemente después de haber recibido retroalimentación de la práctica anterior, ya sea a través del Excel Evaluación o de sus compañer@s de equipo; lo anterior dado que para la evaluación, todo requerimiento de la práctica anterior se cuenta como un requerimiento más de esta práctica.