

Informe: Sistema de Reservación de Vuelos

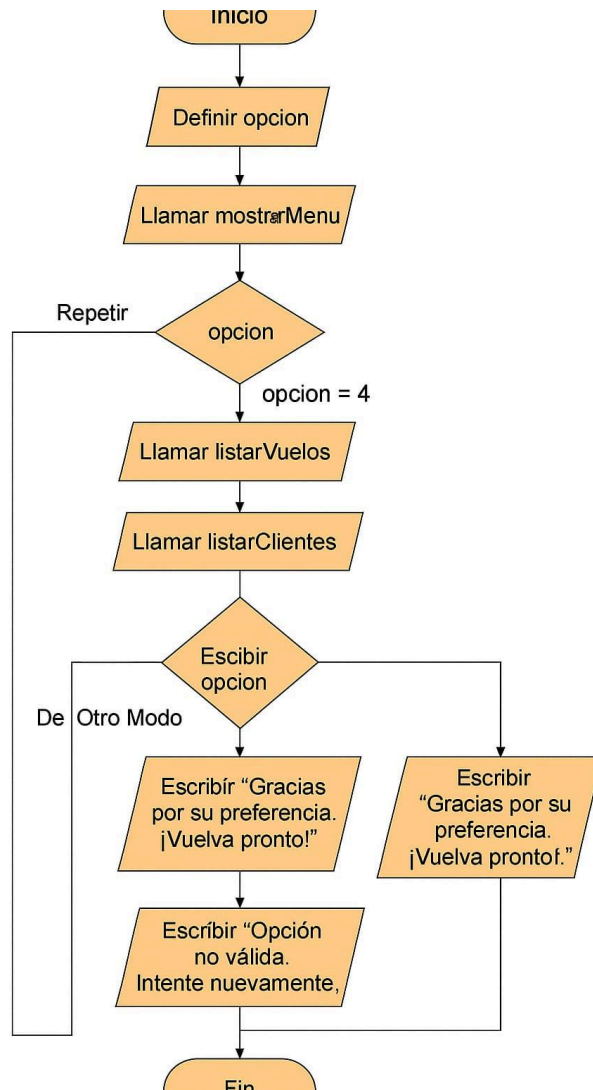
1. Análisis del Programa

Este programa simula un sistema básico de reservación de vuelos para una aerolínea ficticia llamada 'AIRLINE PREMIUM'. Permite al usuario realizar tres acciones principales: visualizar vuelos disponibles, comprar boletos y consultar la lista de clientes registrados.

El programa funciona utilizando estructuras para manejar información sobre vuelos y clientes, así como arreglos para almacenar múltiples registros. Se gestiona a través de un menú interactivo por consola.

- Datos de entrada:
 - ID del vuelo a comprar.
 - Nombre del cliente.
 - Correo electrónico del cliente.
- Datos de salida:
 - Lista de vuelos disponibles con detalles.
 - Confirmación de compra de boletos.
 - Lista de clientes frecuentes registrados.
- Restricciones:
 - Máximo 50 vuelos y 50 clientes.
 - No se permite la compra si no hay asientos disponibles.
 - No se acepta nombre o correo vacío.
 - No se permite seleccionar vuelos inexistentes.

2. Diagrama de Flujo



3. Pseudocódigo (Resumen)

// Definición de constantes

Const MAX_VUELOS <- 50

Const MAX_CLIENTES <- 50

// Definición de estructuras

Tipo Vuelo:

id: Entero

origen: Cadena

destino: Cadena

fechaVuelo: Cadena

asientos: Entero

precio: Real

FinTipo

Tipo Cliente:

nombre: Cadena

email: Cadena

idVuelo: Entero

pago: Real

FinTipo

// Variables globales

Definir vuelos[0..MAX_VUELOS-1] Como Vuelo

Definir totalVuelos Como Entero <- 0

Definir clientes[0..MAX_CLIENTES-1] Como Cliente

Definir totalClientes Como Entero <- 0

// Función principal

Proceso Principal

Definir opcion Como Entero

Llamar vuelosDisponibles()

Repetir

Llamar mostrarMenu()

Escribir "Elige una opción: "

Leer opcion

Segun opcion Hacer

1:

Llamar listarVuelos()

2:

Llamar comprarBoleto()

3:

Llamar listarClientes()

4:

Escribir "Gracias por su preferencia. ¡Vuelva pronto!"

De Otro Modo:

Escribir "Opción no válida. Intente nuevamente."

FinSegun

Hasta Que opcion = 4

FinProceso

// Carga de vuelos disponibles

SubProceso vuelosDisponibles()

vuelos[0] <- Nuevo Vuelo

vuelos[0].id <- 1

vuelos[0].origen <- "Ciudad de México"

vuelos[0].destino <- "Madrid"

```
vuelos[0].fechaVuelo <- "30/05/2025"
```

```
vuelos[0].asientos <- 10
```

```
vuelos[0].precio <- 830.0
```

```
vuelos[1] <- Nuevo Vuelo
```

```
vuelos[1].id <- 2
```

```
vuelos[1].origen <- "Ciudad de México"
```

```
vuelos[1].destino <- "Guadalajara"
```

```
vuelos[1].fechaVuelo <- "15/06/2025"
```

```
vuelos[1].asientos <- 15
```

```
vuelos[1].precio <- 380.50
```

```
vuelos[2] <- Nuevo Vuelo
```

```
vuelos[2].id <- 3
```

```
vuelos[2].origen <- "Ciudad de México"
```

```
vuelos[2].destino <- "Nueva York"
```

```
vuelos[2].fechaVuelo <- "20/07/2025"
```

```
vuelos[2].asientos <- 8
```

```
vuelos[2].precio <- 650.0
```

```
vuelos[3] <- Nuevo Vuelo
```

```
vuelos[3].id <- 4
```

```
vuelos[3].origen <- "Ciudad de México"
```

```
vuelos[3].destino <- "Can Cún"
```

```
vuelos[3].fechaVuelo <- "18/08/2025"
```

```
vuelos[3].asientos <- 4
```

```
vuelos[3].precio <- 420.0
```

```
totalVuelos <- 4
```

```
FinSubProceso
```

```
// Mostrar menú
```

```
SubProceso mostrarMenu()
```

```
    Escribir ""
```

```
    Escribir "=== AIRLINE PREMIUM ==="
```

```
    Escribir "1. Vuelos Disponibles"
```

```
    Escribir "2. Comprar Boleto"
```

```
    Escribir "3. Clientes Frecuentes"
```

```
    Escribir "4. Salir"
```

```
FinSubProceso
```

```
// Listar vuelos
```

```
SubProceso listarVuelos()
```

```
    Escribir ""
```

```
    Escribir "ID    Origen        Destino        Fecha    Asientos Precio"
```

```
    Para i <- 0 Hasta totalVuelos - 1 Con Paso 1
```

```
        Escribir vuelos[i].id, " ", vuelos[i].origen, " ", vuelos[i].destino, " ",  
vuelos[i].fechaVuelo, " ", vuelos[i].asientos, " ", "$", vuelos[i].precio
```

```
    FinPara
```

```
    Llamar pausa()
```

```
FinSubProceso
```

// Comprar boleto

SubProceso comprarBoleto()

Definir id, idx Como Entero

Definir nombre, email Como Cadena

Escribir "-- Comprar Boletos --"

Llamar listarVuelos()

Escribir "Ingresa el ID de vuelo a comprar: "

Leer id

idx <- buscarVueloPorID(id)

Si idx < 0 Entonces

Escribir "Error: El ID no existe."

Llamar pausa()

Sino

Si vuelos[idx].asientos <= 0 Entonces

Escribir "Lo sentimos, no hay asientos disponibles."

Llamar pausa()

Sino

Escribir "Nombre completo: "

Leer nombre

Escribir "Correo electrónico: "

Leer email

Si Longitud(nombre) = 0 O Longitud(email) = 0 Entonces

Escribir "Error: Nombre y correo no pueden estar vacíos."

Llamar pausa()

Sino

vuelos[idx].asientos <- vuelos[idx].asientos - 1

clientes[totalClientes].nombre <- nombre

clientes[totalClientes].email <- email

clientes[totalClientes].idVuelo <- id

clientes[totalClientes].pago <- vuelos[idx].precio

totalClientes <- totalClientes + 1

Escribir "¡Boleto comprado exitosamente!"

Escribir "Destino: ", vuelos[idx].destino

Escribir "Precio: \$", vuelos[idx].precio

Llamar pausa()

FinSi

FinSi

FinSi

FinSubProceso

// Listar clientes

SubProceso listarClientes()

Si totalClientes = 0 Entonces

Escribir "No hay clientes registrados aún."

Sino

Escribir ""

Escribir "Nombre Email Vuelo Pago"

Para i <- 0 Hasta totalClientes - 1 Con Paso 1


```
        Escribir clientes[i].nombre, " ", clientes[i].email, " ", clientes[i].idVuelo, " ", "$",  
clientes[i].pago
```

```
    FinPara
```

```
    FinSi
```

```
    Llamar pausa()
```

```
FinSubProceso
```

```
// Buscar vuelo por ID
```

```
Funcion buscarVueloPorID(id) : Entero
```

```
    Para i <- 0 Hasta totalVuelos - 1 Con Paso 1
```

```
        Si vuelos[i].id = id Entonces
```

```
            retornar i
```

```
        FinSi
```

```
    FinPara
```

```
    Retornar -1
```

```
FinFuncion
```

```
// Pausa
```

```
SubProceso pausa()
```

```
    Escribir ""
```

```
    Escribir "Presiona Enter para continuar..."
```

```
    Esperar Tecla
```

```
FinSubProceso
```

4. Pruebas de Escritorio

Caso	Entrada	Proceso	Salida Esperada	Observaciones
1	Opción 1	Listar vuelos	Lista completa de vuelos	Se muestran todos los vuelos disponibles.
2	Opción 2, ID=3, Nombre=Juan, Email=juan@mail.com	Compra válida	Confirmación y registro exitoso	Cliente registrado, asiento disminuye.
3	Opción 2, ID=999	ID inválido	Mensaje de error	No se encuentra vuelo, no se compra boleto.

5. Conclusiones

Prado vargas Manuel:

Con este proyecto logré aplicar de forma práctica los conceptos fundamentales de la programación estructurada, como el uso de estructuras de datos, procedimientos, funciones y manejo de arreglos. Desarrollé un sistema básico de gestión de vuelos que permite listar vuelos, comprar boletos y consultar a los clientes registrados. Esto me ayudó a fortalecer mi lógica de programación, mejorar mi organización del código y comprender mejor cómo diseñar un sistema que interactúe con el usuario de manera clara. Además, este proyecto representa una base sólida para seguir ampliándolo en el futuro con más funcionalidades como cancelaciones, historial de compras o integración con bases de datos. Me siento satisfecho con el resultado y con lo que aprendí durante su desarrollo.

Cabañas solis gerardo

Con este proyecto pude aprender los fundamentos básicos de la programación, a pesar de que en cierto momento se nos llegó a complicar lo pudimos solucionar de una manera adecuada solicitando la ayuda de la profesora, con este proyecto quedaron más claros los conceptos y el como se utilizan al programar ya que está parte es la que se me dificulta, me sentí un poco nervioso pues era la primera vez que hacía un programa desde ceros y no había tenido el acercamiento a la programación hasta ahora.

Perez Valverde Luis Esteban

Con el proyecto aplicamos todos los conocimientos dados durante el curso, teniendo en cuenta que solo son Fundamentos es impresionante lo que se puede hacer, el proyecto fue bastante complicado de hacer aunque muy gráficamente y entendimos como es que podemos aplicar la programación en un hábito más real.

Calderon Diaz Alcibiades Adrian

Con el proyecto final y, en general, con el curso de fundamentos de programación, pude aprender y aplicar de forma óptima las estructuras, los procedimientos y las funciones. En lo personal, hubo partes del curso que se me dificultaron, pero gracias a la ayuda y orientación de la profesora pude solucionarlas de la mejor manera. Al final, me siento satisfecho por lo aprendido con este proyecto, ya que también me sirvió para conocer mejor mis áreas de oportunidad y trabajar en ellas.

Referencias (formato APA 7):

1. PSeInt. (2023). PSeInt - Pseudo Interpreter. <http://pseint.sourceforge.net>
2. Joyanes Aguilar, L. (2011). Fundamentos de programación: algoritmos, estructura de datos y objetos (2.^a ed.). McGraw-Hill.
3. Pressman, R. S., & Maxim, B. R. (2015). Ingeniería de software: un enfoque práctico (8.^a ed.). McGraw-Hill.
4. Schneider, D., & Gersting, J. (2018). Invitation to Computer Science (8th ed.). Cengage Learning.
5. Forouzan, B. A., & Gilberg, R. F. (2013). Estructuras de datos: un enfoque orientado a objetos con Java (2.^a ed.). Cengage Learning.
6. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). Prentice Hall.
7. Deitel, P., & Deitel, H. (2015). C: How to Program (8th ed.). Pearson Education.
8. Malik, D. S. (2012). C Programming: From Problem Analysis to Program Design (7th ed.). Cengage
9. Tanenbaum, A. S., & Bos, H. (2015). Modern Operating Systems (4th ed.). Pearson.
10. McMillan, K. (2006). Data Structures and Problem Solving Using C (2nd ed.). Pearson.