

Trabajo Práctico Nro. 2 - 2025

Algoritmos y Estructuras de Datos



Índice

Introducción	2
Desarrollo	2
Modelo de datos y estructuras sugeridas para almacenar la información en memoria.	9

Introducción

En este TP2 se introduce el concepto de variables del tipo array (unidimensionales y bidimensionales), junto con Procedimientos y Funciones con pasaje de parámetros.

Deben realizar la codificación en lenguaje Python de lo requerido.

Recordar indicar claramente la declarativa de variables utilizadas en cada procedimiento o función, como también en el programa principal, en especial, las referidas a las variables del tipo Array (dimensiones, tamaño de cada arreglo y su tipo de datos).

Tener en cuenta al implementar los Arrays en Python deben respetar el marco teórico de la cátedra, es decir, deben tener dimensión y tamaño fijo (no modificable en tiempo de ejecución del programa) y mismo tipo de dato de cada elemento del Array.

Antes de comenzar con este TP, por favor leer el documento Enunciado General de manera completa, para saber de qué va el negocio, y entender el modelo de datos requerido. Es obligatoria también la lectura de las preguntas frecuentes, para evitar prácticas que resten puntos al momento de la entrega.

Desarrollo

Deberán realizar lo siguiente:

Al iniciar el programa se debe visualizar un menú con los siguientes puntos:

1. **Registrarme**
2. **Iniciar Sesión (login en el sistema)**

Una vez que inició sesión se deberá:

Hacer un menú para cada tipo de usuario ("administrador", "ceo", y "usuario").

- Pedir un nombre de usuario y contraseña al operador. Verificar que exista el usuario y su clave guardado en el sistema, y de acuerdo al tipo de usuario que está ingresando, presentar el menú correspondiente.
- Reusar el control de claves desarrollado en el TP1, permitiendo hasta 3 intentos como máximo.
- Deberán tener precargados, al comienzo del programa, usando arrays, los siguientes datos referidos a la entidad **USUARIOS**:

Código	Usuario	Clave	Tipo
0	admin@ventaspasajes777.com	admin123	administrador
1	ceo1@ventaspasajes777.com	ceo123	ceo
2	ceo2@ventaspasajes777.com	ceo456	ceo
.	_(máximo 5 ceos porque son 5 aerolíneas)
6	usuario1@ventaspasajes777.com	usuario123	usuario
7	Usuario2@ventaspasajes777.com	usuario456	usuario
8			
9			

Esta precarga de datos de usuarios hacerla en un procedimiento de nombre “CargaUsuarios()”

Suponer un máximo de 10 usuarios en total (un administrador, 5 ceos y 2 usuarios comunes y 2 para el registro de nuevos usuarios)

Para el caso del “Administrador”

○ Si el usuario que ingresa (por medio de su nombre de usuario y su clave) coincide con un usuario y la respectiva clave guardada, y el mismo es del tipo **Administrador**, entonces se deberá mostrar el menú completo correspondiente a un usuario de ese tipo:

1. **Gestión de Aerolíneas**
2. Aprobar/Denegar Promociones
3. Gestión de Novedades
4. Reportes
5. Salir

Y luego, si el operador (en este caso el Administrador), elige la opción: **1. Gestión de Aerolíneas**, se abrirá un nuevo menú de opciones (submenú), con:

1. Gestión de Aerolíneas
 - a) **Crear Aerolínea**
 - b) **Modificar Aerolínea**
 - c) Eliminar Aerolínea
 - d) **Volver**

Lo mismo sucede al elegir la opción 3 y la opción 4

3. Gestión de Novedades
 - a) Crear Novedad
 - b) Modificar Novedad
 - c) Eliminar Novedad
 - d) Ver Novedades
 - e) Volver

4. Reportes
 - a) Reporte de Ventas (reservas con estado “confirmada”)
 - b) Reporte de Vuelos
 - c) Reporte de Usuarios
 - d) Volver

En todos los casos, la opción “**Volver**”, abandona la pantalla del submenú y vuelve al menú principal.

Con la opción **5. Salir** del menú principal, se abandona el sistema.

Para el caso de los “ceo” de las aerolíneas

○ Si el usuario que ingresa (por medio de su nombre de usuario y su clave) coincide con un usuario y la respectiva clave guardada, y el mismo es del tipo “**ceo**”, deberán mostrar el menú completo correspondiente a un usuario de ese tipo:

1. Gestión de Vuelos
 - a. Crear Vuelo
 - b. Modificar Vuelo
 - c. Eliminar Vuelo
 - d. Volver
2. Gestión de Promociones
 - a. Crear Promoción
 - b. Modificar Promoción
 - c. Eliminar Promoción
 - d. Volver
3. Reportes
 - a. Reporte de ventas de mi Aerolínea
 - b. Reporte de ocupación de Vuelos de mi Aerolínea
 - c. Volver
4. Cerrar Sesión

Para el caso de los “Usuarios”

○ Si el usuario que ingresa (por medio de su nombre de usuario y su clave) coincide con un usuario y la respectiva clave guardada, y el mismo es del tipo “**usuario**”, deberán mostrar el menú completo correspondiente a un usuario de ese tipo:

1. **Buscar Vuelos**
2. **Buscar asientos**
3. Reservar Vuelos
4. Gestionar Reservas
 - a. Consultar Reservas
 - b. Cancelar/Confirmar Reservas
 - c. Volver
5. Ver Historial de Compras (reservas con estado “confirmada”)
6. Ver Novedades
7. Cerrar Sesión

Nota 1: en todos los casos donde no haya que desarrollar el punto de menú se deberá mostrar al momento de ingresar al mismo la leyenda “En construcción...”. Este texto debería tener algún formato especial como ser de algún color específico, contener algún diseño con bordes o formato de tabla (investigar como realizarlo en Python)

Nota 2: al momento de cerrar sesión en el sistema, se deberá volver a pedir login sin perder en ningún caso los datos ingresados (ya que trabajamos con array en memoria RAM).

- **Requerimiento N° 1:** vamos a trabajar programando la opción **1. Gestión de Vuelos**

Para el usuario del tipo “**ceo**”, desarrollar todos los puntos del menú de **Gestión de Vuelos**:

- a) **Crear Vuelo**
- b) **Modificar Vuelo**
- c) **Eliminar Vuelo**
- d) **Volver**

Cada vez que se ingresa a Crear, Modificar o Eliminar Vuelos, preguntar al operador si desea ver los vuelos ya cargados hasta el momento, si es así, mostrarlos.

- **Crear Vuelos:**

No se sabe cuántos vuelos se podrán crear cada vez que se ingresa a esta opción (proponer fin de datos acorde), pero se sabe que no serán más de 20.

De cada “vuelo” se deberán ingresar: el código de la aerolínea (validar mediante una función BuscarAerolinea() con su correspondiente parámetro) que la misma se encuentre previamente cargada en el arreglo de aerolíneas que estará en memoria al momento de realizar correctamente el login al sistema, el origen del vuelo (por ejemplo ‘ROSARIO’), el destino del vuelo (por ejemplo ‘BARILOCHE’), y la fecha de salida y hora de salida. Por último, se cargará el precio del vuelo (en otro arreglo).

Nota: para simplificar, asumimos que todos los vuelos utilizan el mismo modelo de avión, por lo que serán siempre 240 asientos disponibles.

Luego, en base al código de vuelo, se deberán inicializar los estados de todos los asientos para dicho número de vuelo. Esto se realiza de forma aleatoria, o sea, todos los asientos del vuelo en esta etapa del proyecto estarán todos cargados con algunos de los 3 estados posibles, “L” de libre, “O” de ocupado, y “R” de reservado. Utilizar la “matriz de asientos” disponible en la memoria del sistema. (ver imagen de estructura sugerida en la sección de “Modelo de datos y estructuras sugeridas para almacenar la información en memoria”). Esta inicialización es sólo a modo de ejemplo para simular como sería el comportamiento real en la selección de asientos a la hora de viajar.

Al finalizar la carga, y mediante un procedimiento “ListarVuelosAerolineas ()” mostrar la cantidad de vuelos “vigentes” por aerolíneas, ordenados de manera descendente por cantidad, para poder visualizar rápidamente la aerolínea con menor y mayor cantidad de vuelos cargados. Nota 1: para realizar los cálculos de este punto se puede utilizar un arreglo auxiliar sin inconvenientes. Nota 2: se asume que el sistema manejará un máximo de 5 aerolíneas.

=====

REPORTE DE VUELOS VIGENTES POR AEROLÍNEA

=====

POSICIÓN	AEROLÍNEA	CANTIDAD DE VUELOS
0	Aerolíneas Argentinas	5
1	LATAM Airlines	3
2	Flybondi	2
3	GOL	1
4	Iberia	0

TOTAL DE VUELOS VIGENTES: 11

Aerolínea con MAYOR cantidad de vuelos: Aerolíneas Argentinas (5)

Aerolínea con MENOR cantidad de vuelos: Iberia (0)

Notas:

- Un vuelo se considera “vigente” cuando su fecha de salida es superior a la fecha del sistema.

○ **Modificar Vuelo:**

Ingresar un código de vuelo (validar que exista). Permitir modificar todos los campos o atributos del vuelo, excepto su código. Tener en cuenta las mismas validaciones que en el menú “Crear Vuelo”. También se puede modificar su precio si lo desea.

○ **Eliminar Vuelo:**

Ingresar un código de vuelo (validar que exista). Preguntar confirmación antes de eliminar el vuelo. Eliminaremos un vuelo de forma lógica, sin perder la información, esto es guardando en un nuevo atributo denominado **estado**, que puede tomar los valores “A” (activo) o “B” (baja). Determinar cómo incorporar ese nuevo atributo en las estructuras de datos que están utilizando.

Por lo cual:

- Al crear un vuelo, a la columna **estado** se le debe asignar el valor “A” (activo).
- Para modificar un vuelo, el mismo debe estar activo (si está dado de baja preguntar si se lo desea activar, si es así, cambiar “B” por “A”).
- Para dar de baja un vuelo, el mismo debe estar activo.

- **Requerimiento N° 2:** vamos a trabajar programando la opción 1. **Buscar vuelos** del perfil “usuario”.

- **Buscar vuelos:** cuando el usuario ingrese a este punto de menú se le solicitará la Fecha salida. El sistema deberá buscar en el arreglo de “vuelos” todos aquellos que estén vigentes. El reporte generado por el sistema será similar al siguiente:

=====

LISTADO DE VUELOS DISPONIBLES EN EL SISTEMA

=====

CÓDIGO vuelo	AEROLÍNEA	ORIGEN	DESTINO	FECHA	HORA	PRECIO
1	Aerolíneas Arg	Buenos Aires	Madrid	15/07/2025	08:30	\$1.101.150
8	LATAM	Santiago	Lima	16/07/2025	10:15	\$550.000
13	Sky Airline	Miami	Nueva York	18/08/2025	14:45	\$800.220
17	GOL	Río Janeiro	Buenos Aires	20/07/2025	22:30	\$670.000
18	LATAM	Lima	Nueva York	22/08/2025	20:30	\$1670.000

Total de vuelos: 5

- **Requerimiento N° 3:** vamos a trabajar programando la opción 2. **Buscar asientos** del perfil “usuario”.

Cuando un usuario ingrese a este punto de menú, el sistema solicitará el código de vuelo, y validará por medio de una función que el vuelo exista en memoria y esté vigente (para esto último verificará que la fecha de salida del vuelo sea *mayor* a la fecha del sistema). Si todo está correcto, deberá mostrar en pantalla por cada fila del avión, todos los asientos junto con su posible estado cargado inicialmente para dicho vuelo en la matriz de asientos. Este menú deberá ser iterativo, proponer un fin de datos. En esta etapa del trabajo NO se procederá con la confirmación del asiento, simplemente se muestra en pantalla los asientos y cuando el usuario presione una tecla se sale del menú.

Observaciones:

- 1) La matriz de asientos de todos los vuelos (ver representación gráfica sugerida al final de este documento) tendrá asignada para cada vuelo 40 filas. O sea, desde la fila 0 a la 39 corresponderá a los asientos del primero vuelo, desde la fila 40 a la fila 79 serán los asientos del segundo vuelo y así sucesivamente hasta llegar al máximo de vuelos permitidos en la matriz (20).
- 2) Todos los vuelos en la matriz tendrán la misma capacidad: 40 filas y 6 asientos por filas (A, B, C, Pasillo, D, E, F). Total, de 240 asientos por vuelo.
- 3) Cada asiento tendrá 3 estados posibles: L (Libre), O (Ocupado), y R (Reservado)
- 4) El pasillo no podrá ocuparse por ningún pasajero.

Notas:

- Todas las opciones de los diferentes menús que no fueron desarrolladas aún, mostrarán la leyenda “En construcción ...”.
- **Importante!!** Respetar los principios de la programación estructurada modular descendente.
Por lo tanto, no se admitirán sentencias como: **break**, **exit()**, **while true ...**

Modelo de datos y estructuras sugeridas para almacenar la información en memoria.

Del modelo completo descrito en la presentación general del TP, en este TP2 vamos a trabajar con las siguientes entidades:

USUARIOS

```
codUsuario (int)
nombreUsuario (string(100))
clave Usuario (string(8))
tipoUsuario ('administrador', 'ceo de
aerolinea', 'usuario') (string(20))
emailUsuario (string(100))
telefonoUsuario (string(20))
```

VUELOS

```
codVuelo (int)
codAerolinea (string [5])
origenVuelo (string(50))
destinoVuelo (string(50))
fechaSalidaVuelo (string (10))
horaSalidaVuelo (string (5))
precioVuelo (decimal)
asientosDisponibles (int)
```

AEROLINEAS

```
codAerolinea (string [5])
nombreAerolinea (string(100))
codigoIATA (string(3))
descripcionAerolinea (string(200))
codPais (string(3))
```

VUELOS					
	1	2	3	4	5
	Codigo Aerolínea	origenVuelo	destinoVuelo	fechaSalidaVuelo	horaSalidaVuelo
0					
1					
2					
3					
4					
5					
...					
19					

`vuelos=[[" "]*5 for i in range(20)]`

PRECIO VUELOS	
	precioVuelo
0	
1	
2	
3	
4	
5	
...	
19	

`precio vuelos = [0.0] * 20`

MATRIZ DE ASIENTOS (única para todos los vuelos)								
	0	1	2	3	4	5	6	
	A	B	C	PASILLO	D	E	F	
0								VUELO 1
1								
2								
...								
39								
40								VUELO 2
41								
42								
...								
79								
80								VUELO 3
81								
82								
...								
119								
120								VUELO N
121								
...								
799								