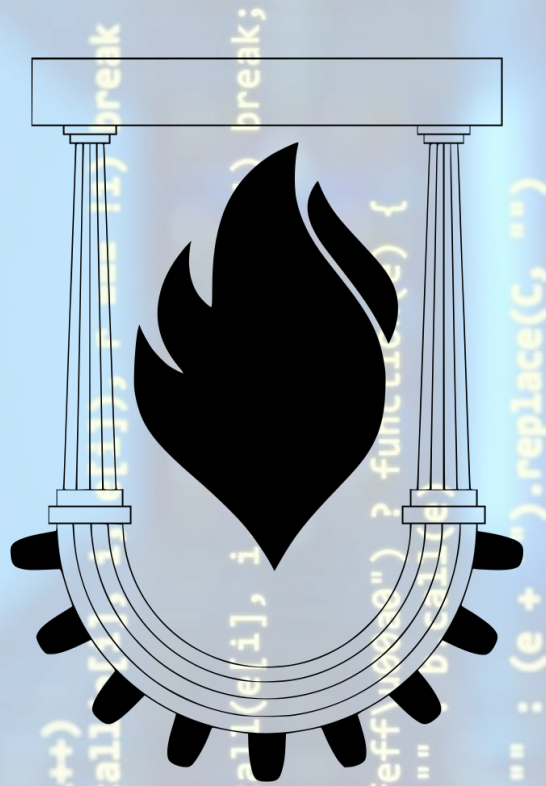




*Universidad Nacional de  
Lomas de Zamora*



# Fundamentos de Programación Informática



Reválida 2021-2022



## Reválida

### Condiciones de entrega:

- Se debe entregar la carpeta completa del proyecto de Code::blocks comprimida y el nombre del archivo debe ser apellido + guión medio + nombre + número de documento todo en minúsculas. Ejemplo: nerea-selene37535140.
- Se debe poner un rótulo con comentarios /\* \*/ indicando NOMBRE, APELLIDO Y DNI.

### Condiciones De Aprobación:

- El programa debe compilar sin errores.
- El programa debe funcionar sin modificaciones.
- Se debe cumplir la consigna dada respetando el orden de ingreso de datos, orden de aviones del menú y los mensajes que deben ser impresos al usuario.
- Presentación del trabajo y defensa oral e individual de cada uno de manera PRESENCIAL.



### Consigna:

Un aeropuerto desea desarrollar de 0 su software de aviones, pasajeros y viajes, para esto determina lo siguiente:

Un vector de aviones de 20 estructuras.

Un vector de pasajeros de 20 estructuras.

Un vector de viajes de 20 estructuras.

El vector de viajes relaciona el de aviones y los pasajeros con el mismo.

### Estructuras:

#### pasajeros

- Nombre y apellido (string 40 caracteres)
- Numero de DNI (int)
- ID (int (Se debe cargar automáticamente al crear nuevos pasajeros, comenzando de 0))

#### aviones

- Código de avión (string 20 caracteres)
- Cantidad de asientos (int)
- Descripción (string 200 caracteres)
- ID (int (Se debe cargar automáticamente al crear aviones, comenzando de 0))

#### viajes

- Código de avión (string 20 caracteres)
- origen (string 20 caracteres)
- destino (string 20 caracteres)
- Vector pasajeros (vector de int con todos los ID de los pasajeros que viajarán)



Cada estructura se guardará en su propio header y fuente (pasajeros.h / pasajeros.c / aviones.h / aviones.c / viajes.h / viajes.c) y las funciones que correspondan a cada estructura deberán realizarse en la fuente correspondiente.

Se pide crear un menú con las siguientes opciones:

1. Registrar pasajero: Esta opción debe permitir el ingreso del nombre y apellido del pasajero, así como su número de DNI y guardarlo en una posición de su vector determinado. Si el vector se llena de pasajeros se debe imprimir un mensaje diciendo que no hay espacios disponibles. **El ingreso debe ser de a un pasajero. El id debe cargarse automáticamente comenzando desde 0.**
2. Mostrar pasajeros: Esta opción debe mostrar una lista de los pasajeros creados. **Esta opción no debe funcionar si no hay pasajeros creados.**
3. Ordenar pasajeros por nombre: Esta opción debe ordenar el vector de pasajeros sin mostrarlo. Una vez ordenado el vector, emitir el mensaje "Se han ordenado los pasajeros por nombre". **Esta opción no debe funcionar si no hay pasajeros creados.**  
**Si su dni TERMINA en par, ordene por nombre alfabéticamente (A-Z)**  
**Si su dni TERMINA en impar, ordene por nombre inversamente (Z-A)**



4. Ordenar pasajeros por ID: Esta opción debe ordenar el vector de pasajeros según su ID sin mostrarlo. Una vez ordenado el vector emitir el mensaje "Se han ordenado los pasajeros por ID". **Esta opción no debe funcionar si no hay pasajeros creados.**  
**Si su dni COMIENZA en par, ordene por id de menor a mayor (0-9)**  
**Si su dni COMIENZA en impar, ordene por id de mayor a menor (9-0)**
  
5. Registrar aviones: Esta opción debe permitir el ingreso del código del avión, la cantidad de asientos y la descripción y guardarlo en una posición de su vector determinado. Si el vector se llena de aviones se debe imprimir un mensaje diciendo que no hay espacios disponibles. **El ingreso debe ser de a una carga de aviones. Si se vuelve a registrar aviones y el código del avión es el mismo, se debe actualizar la cantidad y descripción del avión con el último valor.**
  
6. Mostrar aviones: Esta opción debe mostrar una lista de los aviones creados. **Esta opción no debe funcionar si no hay aviones creados.**
  
7. Quitar aviones: Se debe dar un código de avión. Se deberá buscar en el vector de aviones ese avión, y si se encuentra, se debe quitar el mismo del vector. **Esta opción no debe funcionar si no hay aviones creados.**  
**Si el SEGUNDO número de su dni (-x.---.---) es par, busque por código.**  
**Si el SEGUNDO número de su dni (-x.---.---) es impar, busque por id.**



8. Registrar viaje: Esta opción debe permitir el ingreso de un viaje, consistente en “código de avión” y “pasajero”. Si no existe el código de avión en el vector de aviones, deberá mostrar un mensaje de error. Si existe, se deben mostrar los pasajeros creados para seleccionar uno y el programa registrar el viaje.

El ingreso debe ser de a un viaje, y no se pueden registrar viajes si no hay aviones creados.

Si cuando cargo un viaje de un avión, ya hay un viaje anterior del mismo avión del mismo avión, debo mostrar un mensaje de error.

9. Mostrar viajes: Esta opción debe mostrar una lista de los viajes creados. Esta opción no debe funcionar si no hay viajes creados.

10. Liquidar viaje de avión: Esta opción debe mostrar una lista de los viajes creados para seleccionar. Se deberá seleccionar uno y luego el programa deberá borrar todos los pasajeros del vector de pasajeros, cuyos ID coincidan con los ID del viaje a liquidar. Esta opción no debe funcionar si no hay aviones, pasajeros o viajes creados.

11. Salir: Esta opción finaliza el programa