

## Trabajo Práctico N° 1 para entregar.

lunes, 4 abril, 23:59

Evento de curso

Importante: Deben enviar el link a la resolución en su repositorio en GitHub del ejercicio.

### Ejercicio 1

La empresa de Transporte de Pasajeros “Viaje Feliz” quiere registrar la información referente a sus viajes. De cada viaje se precisa almacenar el código del mismo, destino, cantidad máxima de pasajeros y los pasajeros del viaje.

Realice la implementación de la clase Viaje e implemente los métodos necesarios para modificar los atributos de dicha clase (incluso los datos de los pasajeros). Utilice un array que almacene la información correspondiente a los pasajeros. Cada pasajero es un array asociativo con las claves “nombre”, “apellido” y “numero de documento”.

Implementar un script testViaje.php que cree una instancia de la clase Viaje y presente un menú que permita cargar la información del viaje, modificar y ver sus datos.

### Resolucion

#### Clase Viaje

Proposito: Resolver el enunciado del ejercicio 1 para entregar solicitado por la catedra.

Variables de instancia: codigo, destino, cantMaxPasajeros, colPasajeros

#### Metodos de instancia

Metodos de acceso a las variables de instancia:

setters y getters (uno para cada variable de instancia).

Metodos de resolucion:

**\_\_toString() RETORNA string:** Redefinicion del metodo heredado de Object a fin de mostrar los datos almacenados en el objeto de clase Viaje en forma de cadena de caracteres.

**mostrarPasajeros() RETORNA string:** Metodo que devuelve una cadena de caracteres con la informacion de todos los pasajeros del viaje.

**disponibilidad() RETORNA int:** Metodo que devuelve la cantidad de lugares disponibles en el viaje.

**existeViaje(string unViaje) RETORNA boolean:** Metodo que recibe un codigo de viaje por parametro y responde si existe.

**existeLugar() RETORNA boolean:** Metodo responde si existen lugares disponibles en el viaje.

**existeDni(string unDni) RETORNA boolean:** Metodo que recibe un dni por parametro y responde si el pasajero ya existe en el viaje.

**agregarPasajero(string nombre,apellido,dni) RETONA void:** Metodo que recibe un nombre, un apellido y un dni por parametro e intenta agregar un nuevo pasajero al viaje. El metodo asume que el pasajero no existe previamente (no verifica existencia por dni).

**modificarPasajero(string unDni, nuevoNombre, nuevoApellido, nuevoDni): RETORNA void:** Metodo que recibe un dni, un nuevoNombre, un nuevoApellido y un nuevoDni y modifica los datos del pasajero del array coleccion de pasajeros. El metodo asume que el existe un pasajero con este dni. Trabaja en conjunto con el metodo privado indexDni(unDni).

**eliminarPasajero(string dni) RETORNA void:** Metodo que recibe un dni y eliminar el pasajero del array coleccion de pasajeros. El metodo asume que el existe un pasajero con este dni. Trabaja en conjunto con el metodo privado indexDni(unDni). El metodo utiliza **array\_splice()** para la eliminaci3n.

**indexDni(unDni): RETORNA int:** Metodo (privado) que recibe un dni y devuelve el valor del indice del array multidimensional colPasajeros que contiene al array asociativo correspondiente al pasajero cuyo dni coincide.

## Clase TestViaje

Proposito: probar el funcionamiento de la clase Viaje.

Variables de instancia: misViajes

Observaciones: misViajes es un array con objetos precargados de tipo Viaje.

### Metodos de instancia

**existeViaje (array misViajes, string unCodigo) RETORNA boolean:** Metodo recibe un array de viajes y un codigo e intenta determinar si el viaje existe.  
recuperarViaje (array misViajes, string unCodigo)

**crearViaje(array misViajes): REOTRNA array:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta agregar un nuevo viaje a misViajes si el codigo ingresado por el usuario no existe.

**mostrarViajes (array misViajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta mostrar informacion completa de todos los viajes (codigo, destino, cantMaxPasajeros y lista de Pasajeros con nombre, apellido y dni).

**mostrarDatosViaje (array misViajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta mostrar informacion de un viaje en particular si el codigo de viaje ingresado por el usuario existe.

**cargarPasajero (array misViajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta cargar un pasajero a un viaje si el codigo de viaje ingresado por el usuario existe y el DNI del pasajero en este viaje no existe.

**mostrarPasajeros (array misViajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta mostrar los datos de los pasajeros de un viaje si el codigo de viaje ingresado por el usuario existe.

**mostrarDisponibilidad (array misVajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta mostrar la cantidad de lugares disponibles de un viaje si el codigo de viaje ingresado por el usuario existe.

**modificarDatosViaje (array misViajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta modificar los datos de un viaje si el codigo de viaje ingresado por el usuario existe. Muestra por pantalla los valores originales y modificados por pantalla.

**eliminarPasajerosPorDni(array misViajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta eliminar un pasajero del viaje si el dni ingresado por el usuario existe en el viaje.

**modificarDatosPasajero(array misViajes): REOTRNA void:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta modificar los datos de un pasajero si el codigo de viaje y el dni ingresados por el usuario existen.

**eliminarViaje(array misViajes): RETORNA array:** Metodo que recibe un arreglo de objetos tipo Viaje e intenta eliminar un viaje si el codigo de viaje ingresado por el usuario existe. El metodo utiliza **array\_splice()** para la eliminaci3n.

**seleccionarOpcion() RETORNA int:** Metodo que muestra un menu de opciones y devuelve un valor entero ingresado por el usuario.

Representacion grafica del array multidimensional colPasajeros y el array asociativo pasajero.

