

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA
GESTION II/2021



PRÁCTICA GENERAL
LAB – 121 ALGORITMOS Y PROGRAMACIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS

1.

TABLET
-idTablet: int -Fabrica: string -modelo: string -precio: double -almacenamiento: int
+Tablet() +Tablet(id,fab,mod,pre,alm) +...()

- Crear 2 constructores
- Instanciar n objetos de tipo Tablet
- Mostrar a las tablets que provienen de la misma fabrica
- Mostrar el precio de la tablet mas cara, acontinuacion todos los id, de las tablets con ese precio.
- Debido a una actualizacion del Fabricante X, incrementar el precio de todos los dispositivos de Fabrica X, con almacenamiento Y en un 15%.

2.

ALBAÑIL
-nombre: string -apellido: string -ci: int -fechaNac: string -salario: double -especialidad: string -nroHerramientas: int -herramienta [50]: string -precio [50]: double

(Ej: especialidad = "techado", "obra fina",...)

- Instanciar 3 albaniles de forma distinta
- Mostrar a los 3 albaniles
- Mostrar el nombre y apellido del albanil que gana mas.
- Para dos albaniles que tienen la misma especialidad, mostrar el albanil que tiene mas dinero invertido en sus herramientas.
- Mostrar el nombre, apellido, ci, salario todos los albaniles, tal que dia+mes+anio de su nacimiento sea un numero primo.

3.

HOSPITAL
-nombreHospital: string
-direccion: string
-telefono: int
-nroPacientes: int
+paciente[50][3]: string
-nroDoctores: int
-doctor[50][4]: string

paciente[i][0] = nombre

paciente[i][1] = ci

paciente[i][2] = enfermedad

paciente = {nombrePaciente,ci,enfermedad}

doctor[i][0] = nombre

doctor[i][1] = ci

doctor[i][2] = especialidad

doctor[i][3] = sueldo

doctor = {nombreDoctor,ci, especialidad, sueldo}

- Instanciar 2 Hospitales de formas distintas
- Mostrar a los hospitales tal que su nombre sea un palindromo
- Mostrar al hospital que tienen mas pacientes con la enfermedad "COVID-19"
- Para un hospital mostrar cuantos doctores tienen la especialidad de Y (Ej "NUTRICION")
- Mostrar a los doctores que ganan mas en el hospital con nombre X
- Debido a que el hospital de nombre X ha tenido complicaciones por el desorden de sus pacientes, te piden: Ordenar a los pacientes por su CI
- Para todos los hospitales, incrementar el sueldo en un Y% ($1 \leq y \leq 100$) de todos los medicos, que tienen la especialidad de "TERAPIA INTENSIVA"

POLIMORFISMO

1.

FECHA
+dia: int
+mes: int
+anio: int
+Fecha()
+Fecha(d,m,a)
+...()

// puedes suponer que todos los meses tienen 30 días (aunque no es lo ideal)

- Instanciar 2 fechas de forma distinta
- Mostrar la fecha en el formato "DD/MM/AAAA"
- sobrecargar el operador ++ para avanzar una fecha en 1 día
- sobrecargar el operador -- para retroceder una fecha en 1 día
- sobrecargar un operador para verificar si dos fechas son iguales
- sobrecargar el operador < para verificar si una fecha es menor a otra, (ej: 30/05/2021 es menor que 01/06/2021)

2.

GUARDERIA
-nombre: string
-direccion: string
-telefono: int
-nroNineras: int
-ninera[50][4]: string
+...()

- Instanciar 2 guarderías de forma distinta y mostrarlas
- Sobrecargar el operador ++ para leer los datos de una guardería
- Sobrecargar el método mostrar de 3 formas distintas
- Sobrecargar el operador + para incrementar el salario de todas las niñas de una guardería en un Y%
- Sobrecargar el operador -- para mostrar a las niñas con nombres palíndromos (EJ: ANA, ANINA, HANNAH,...)
- Sobrecargar el operador * para mostrar la guardería con mayor grado de paciencia (se define el grado de paciencia de una guardería, como la sumatoria de todos los grados de paciencia de sus niñas)
- Cambiar el número de teléfono de una guardería con el valor X sobrecargando el operador %

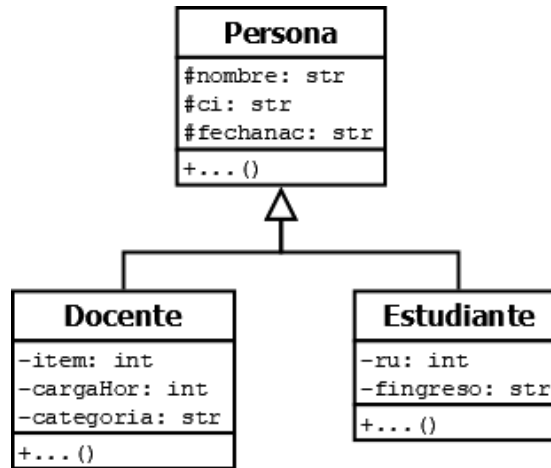
3.

PASTELERIA	PERSONA
-nombre: string	-nombre: string
-telefono: int	-ci: int
-nroRepartidores: int	-edad: int
-repartidor[50][3]: string	+...()
+nroProducto: int	
-producto[50]: string	
-costo[50]: double	
+...()	

- Instanciar 2 Pastelerias
- Instanciar 3 Personas de forma distinta
- Sobrecargar el operador ++ para leer una pasteleria
- Sobrecargar el metodo mostrar() de la pasteleria de 3 formas distintas
- sobrecarga el operador ++ para mostrar una Persona
- Sobrecarga una funcion de 2 formas para poder modificar el ci de una persona
- Sobrecarga un operador y haz otra funcion, para añadir una Persona p como repartidor a una pasteleria
- Sobrecarga el operador True para verificar si una pasteleria tiene servicio de delivery
- Dado una Persona p y una Pasteleria, sobrecarga el operador - para despedir a la persona p como repartidor y posteriormente mostrar la pasteleria, en caso de no haber trabajado alli, mostrar un mensaje "ESTA PERSONA NO TRABAJA ALLI."
- Sobrecarga un operador para incrementar el costo de sus productos con precio menor a 50 bs en un monto de X bs
- Debido a que en el catalogo de postres que tiene la pasteleria es dificil poder localizar un determinado producto, te piden ordenar los productos que tiene, por su nombre de manera ascendente.
- Debido a que el dia de hoy les fue bien en sus ventas, el dueño decide hacer una fiesta con un poco de licor, pero como tiene empleados menores de edad y no debe comprar para todos(ya que el dueño es muy responsable), te pide que muestres un listado de todos los repartidores que son mayores de edad (nombre, edad).

HERENCIA

1. Sea el siguiente diagrama UML:

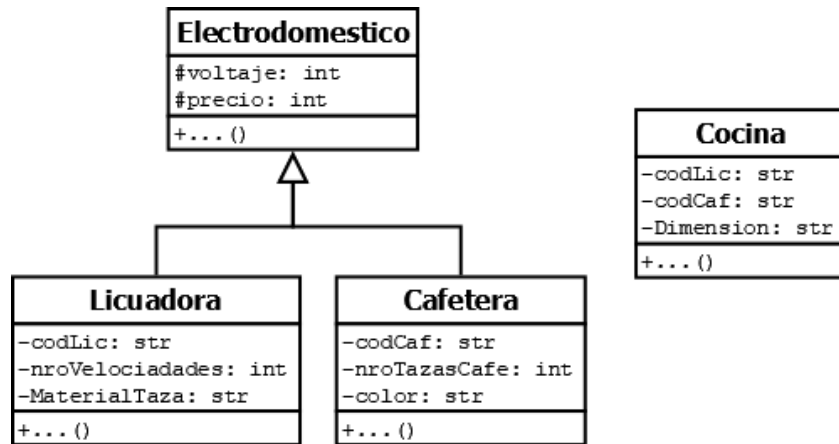


Resolver:

- Implementar todas las clases y sus métodos principales y con al menos 3 constructores distintos.
 - Crear al menos dos objetos docentes y dos objetos estudiantes.
 - Realizar un método que verifique si un docente y un estudiante son la misma persona.
 - Mostrar si los objetos estudiantes son mayores de edad.
 - Realizar un método que con los dos objetos docente mostrar la persona que tiene mayor carga horaria.
2. Una compañía editorial produce tanto libros impresos como audio-libros en discos compactos. Diseñe una clase denominada PUBLICACION que almacene el título y el precio de una publicación. A partir de esta clase, derive dos clases: LIBRO a la cual le agregue el número de páginas y CD, a la cual le agregue el tiempo de reproducción en minutos. Cada una de las clases debe tener propiedades para acceder a sus respectivos datos.

Realizar:

- Elabore un diagrama de clases UML indicando las relaciones de herencia.
 - Implementar los métodos para obtener el título y el precio en las clases LIBRO y CD.
 - Calcular el precio total por la venta de 2 Libros y 1 CD.
 - Verificar si el libro también es un audio libro.
3. Sea el siguiente diagrama:

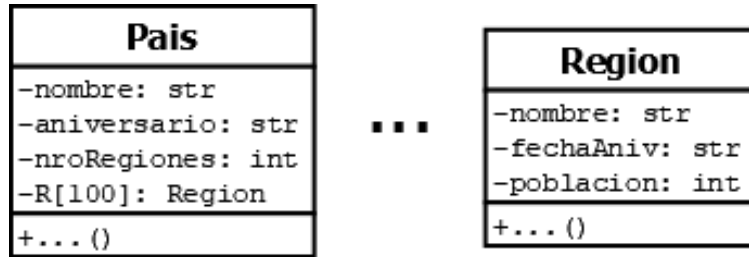


Resolver:

- Instanciar los Objetos: 3 Licuadoras, 2 Cafeteras, 1 Cocinas.
- Verificar si la Licuadora x pertenece a la cocina.
- Mostrar el número de tazas de café de la cafetera de la cocina.
- Verificar si la licuadora y la cafetera de una cocina tienen el mismo voltaje.

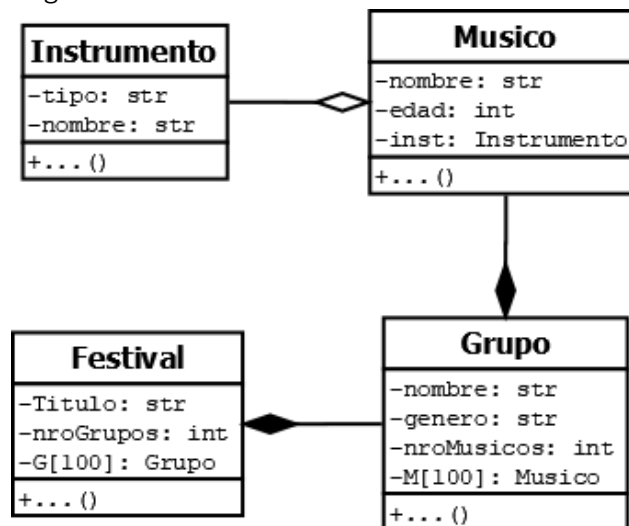
AGREGACIÓN Y COMPOSICIÓN

1. Complete el siguiente diagrama con la relación que corresponde.



Resolver:

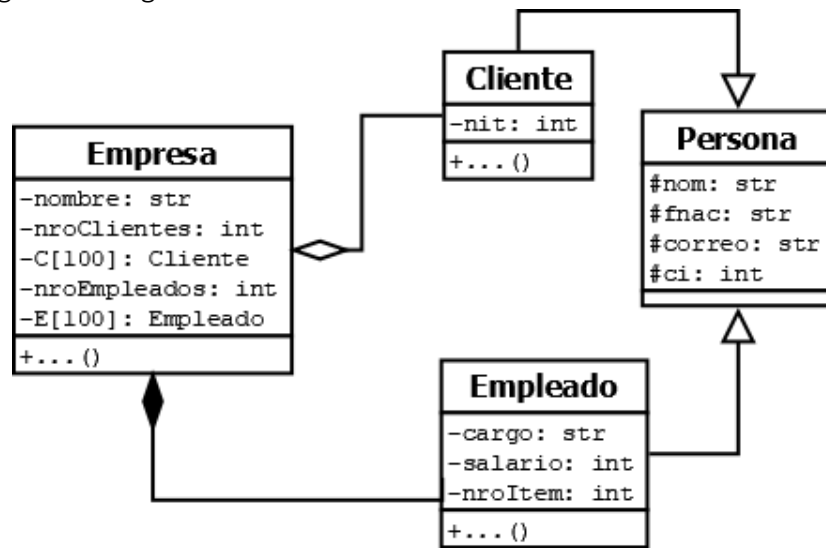
- Crear las clases y instanciar al menos dos objetos país.
 - Realizar un método que muestre solo las fechas de aniversarios de todas las regiones.
 - Mostrar la región que tenga mas habitantes.
 - Mostrar la población total de alguno de los objetos país.
2. Dado el siguiente diagrama:



Resolver:

- Crear todas las clases con sus métodos utilizados.
- Determinar el nombre de los músicos que tocan los instrumentos de tipo Viento
- Determinar si en el festival existe un grupo de genero Cumbia
- Contar cuántos grupos son del género Rock
- Ordenar el festival dependiendo de la cantidad de músicos (ascendentemente).
- Verificar si tocara el Grupo X

3. Dado el siguiente diagrama:

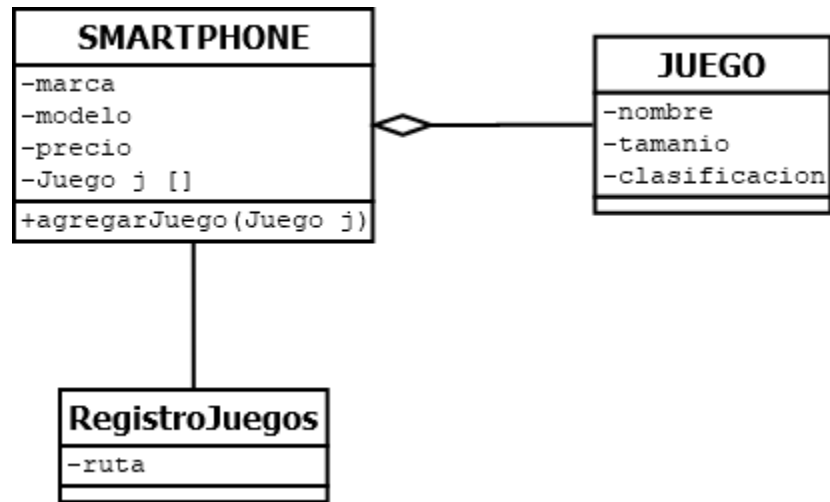


Resolver:

- Crear todas las clases y sus métodos utilizados.
- Eliminar al empleado con `ci x` de la lista de empleados de la empresa.
- Ordenar a todos los empleados según su salario.
- Eliminar al cliente que es cliente y empleado en la empresa.

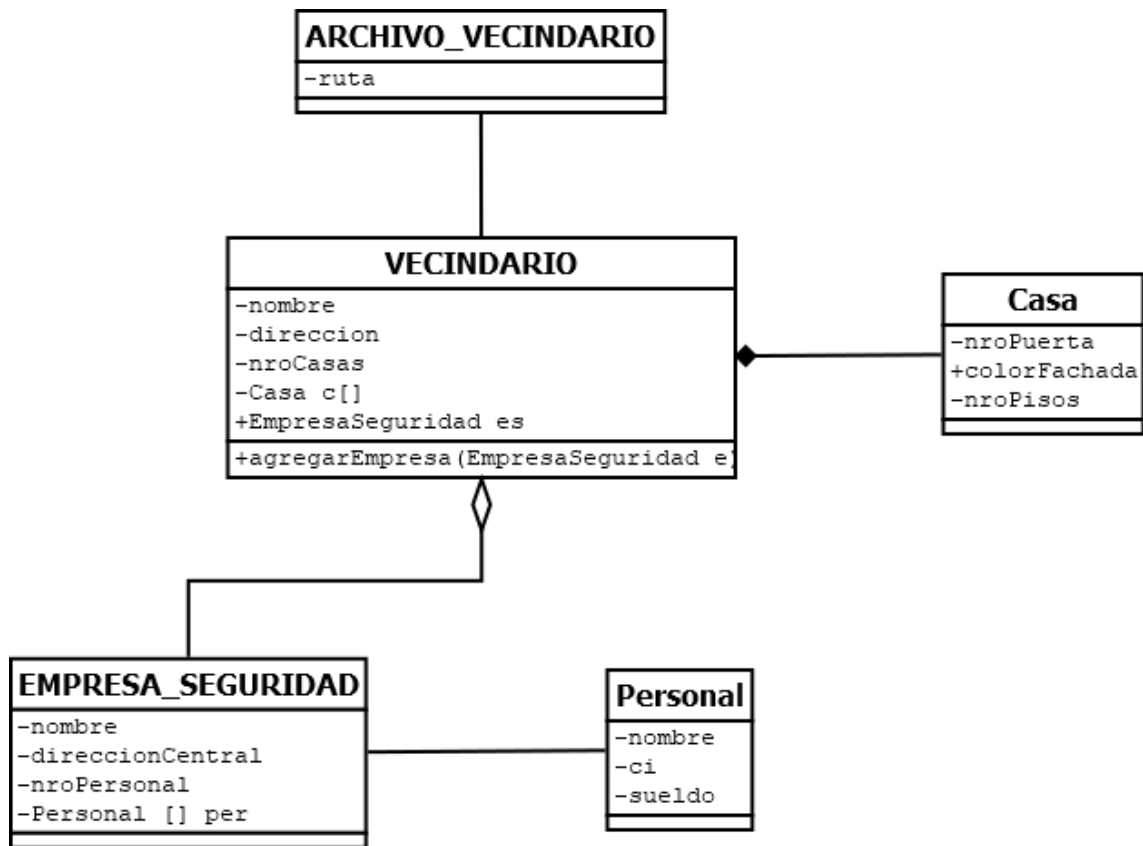
PERSISTENCIA

1.



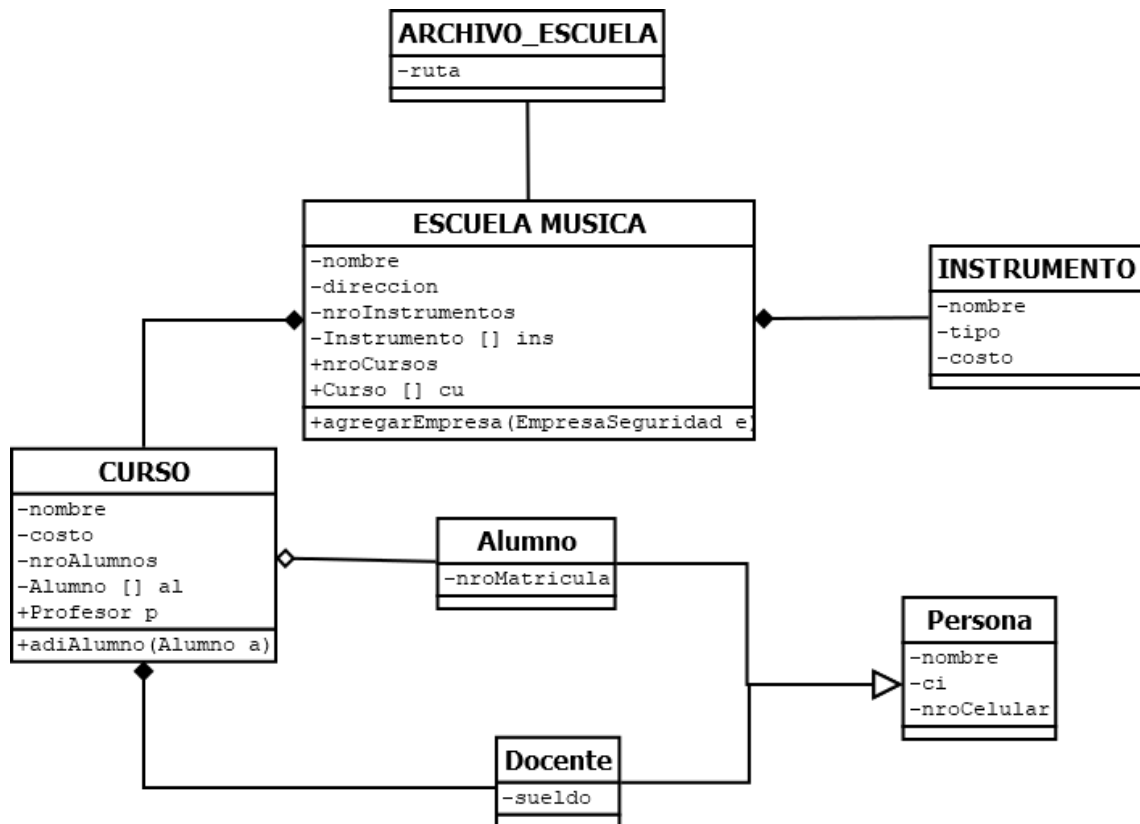
- a) Crear, Leer y Mostrar un Registro de Juegos
- b) Agregar un juego a todos los smartphones
- c) Adicionar un Smartphone al archivo de Juegos
- d) Disminuir el precio de los celulares Samsung en un 5% de su precio actual
- e) Mostrar los Smartphones con menor precio
- f) Mostrar los Smartphones que tengan mas juegos

2.



- Crear, Leer y Mostrar un Archivo_Vecindario
- Mostrar todo los vecindarios que tienen X personal de seguridad , en su vecindario
- Mostrar las casas donde el numero de puerta es un numero perfecto (se dice que x es un numero perfecto si la suma de sus divisores de x excepto el mismo es x)
- Modificar las casas en las cuales el nroPisos sea un número primo circular, (se dice que un número es primo circular si al poner el ultimo dígito como primero, sigue siendo primo el numero)
Ejemplo el 311 es primo circular
- Eliminar a los vecindarios que tengan personal nombre X, ademas que tengas nroCasas entre el rango [l, r], y en su ci tenga un dígito primo.

3.



- Crear, Leer y Mostrar el Archivo Escuela
- Mostrar a las escuelas que tengan mas instrumentos diferentes
- De cada escuela de música mostrar el curso donde tengan alumnos con el nombre mas largo.
- Inscribir al Alumno X en el curso Y de la escuela de música Z
- Mostrar las escuelas que tienen un docente que gana menos de 3000 bs
- Ordenar a las escuelas de acuerdo a la mayor cantidad de estudiantes
- Eliminar el curso que tenga menos de X alumnos y el profesor tenga nombre Z
- Modificar el nroCelular del docente que dicta un curso donde la cantidad de alumnos sea un numero Fibonacci
- Ordenar a las escuelas de acuerdo a los docentes que ganan mas sueldo
- Mostrar cuanto gana cada escuela
- De cada curso ordenar a los estudiantes por el nombre (orden lexicográfico) si dos estudiantes tienen el mismo nombre ordenar por el número de celular (de manera decreciente)

GENERICIDAD

1.

Usuario
-codigo: string
-nombre: string
-contraseña: T

- Crear 2 usuarios donde la contraseña sea de tipo **string** y mostrar sus datos
- Crear 2 usuarios donde la contraseña sea de tipo **int** y mostrar sus datos
- Crear un vector de Usuarios donde la contraseña es de tipo **int**, luego debe ordenar en forma descendente respecto al nombre, si tienen nombres iguales, ordenar de acuerdo a la contraseña.
- En el vector de Usuarios eliminar los que tengan el mismo nombre.

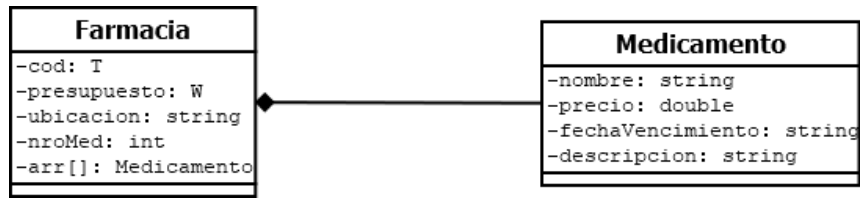
2.

Jugador
-nombre: string
-apellido: string
-sueldo: W
-titulo[]: int
-posicion: string

Dónde: titulo[] : están los años donde un jugador gano un título.

- Crear un jugador donde el sueldo es de tipo **int** y mostrar
- Crear un jugador donde el sueldo es de tipo **double** y mostrar
- Crear un vector de jugadores donde el sueldo sea de tipo **int**.
- Mostrar a(los) jugador(es) que tengan el sueldo máximo.
- Mostrar los nombres de los jugadores que hayan ganado un título en el año X, y sean delanteros.
- Eliminar a los jugadores que tengan el sueldo máximo y si han ganado algún título entre los años X e Y, donde no siempre se cumple que $X \leq Y$.

3.



- a) Crear una Farmacia donde el cod es de tipo **int** y el presupuesto es de tipo **double** y mostrar.
- b) Mostrar los medicamentos que ya vencieron en el día X, mes Y y año Z.
- c) Crear un array de Farmacias donde cod es de tipo **string** y presupuesto es de tipo **int**.
- d) Mostrar a las farmacias donde el nombre de algún medicamento sea palíndromo.
- e) Mostrar a las farmacias que tengan el mayor presupuesto.
- f) Ordenar las farmacias ascendentemente de acuerdo al código que tienen.
- g) Eliminar las farmacias que tengan código Z y el presupuesto sea un número primo.
- h) De cada farmacia calcular el precio total de todos los medicamentos que tienen sin que estos hayan vencido.
- i) De cada farmacia eliminar los medicamentos que tienen una fecha no válida (tomando en cuenta el calendario y años bisiestos), si todos los medicamentos tienen fechas no válidas debe agregar X medicamentos en dicha farmacia.
- j) De cada farmacia ordenar los medicamentos de la siguiente manera:
 - a. Los primeros elementos del vector están ordenados por precio (de mayor a menor) pero siempre y cuando el año de la fecha de vencimiento sea un año bisiesto
 - b. Los últimos elementos del vector están ordenados por precio (de menor a mayor) siempre y cuando el año de la fecha de vencimiento no sea un año bisiesto