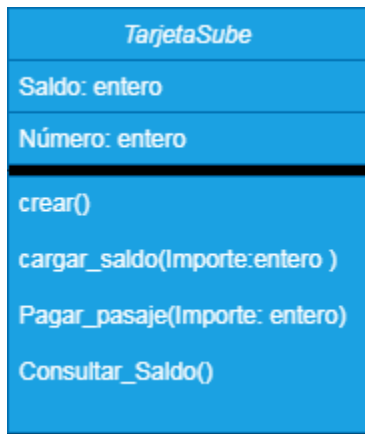


## PROGRAMACION ORIENTADA A OBJETOS

### Practico Nº 2 Unidad 2

#### **Ejercicio 1**

Dada la siguiente clase en UML:



Codifique la clase "Tarjeta Sube" utilizando el lenguaje Python, usando módulos.

Escriba una función *test()*, que lea desde teclado los datos (los datos los lee dentro de la función), para crear 3 objetos de la clase, verificando el funcionamiento correcto de todos los métodos.

En otro módulo, hacer el programa principal que invoque a la función *test*.

#### Reglas de negocio:

El método *Pagar\_pasaje(Importe)*, debe verificar si hay saldo suficiente, en caso afirmativo llevar a cabo la operación e informar el nuevo saldo, y en caso negativo, devolver un valor negativo que se deberá chequear para saber si se pudo realizar el pago.

El método *cargar\_saldo(importe)*, debe verificar que el importe sea positivo.

**Lea atentamente el enunciado, identifique las clases y los gestores/manejadores.**

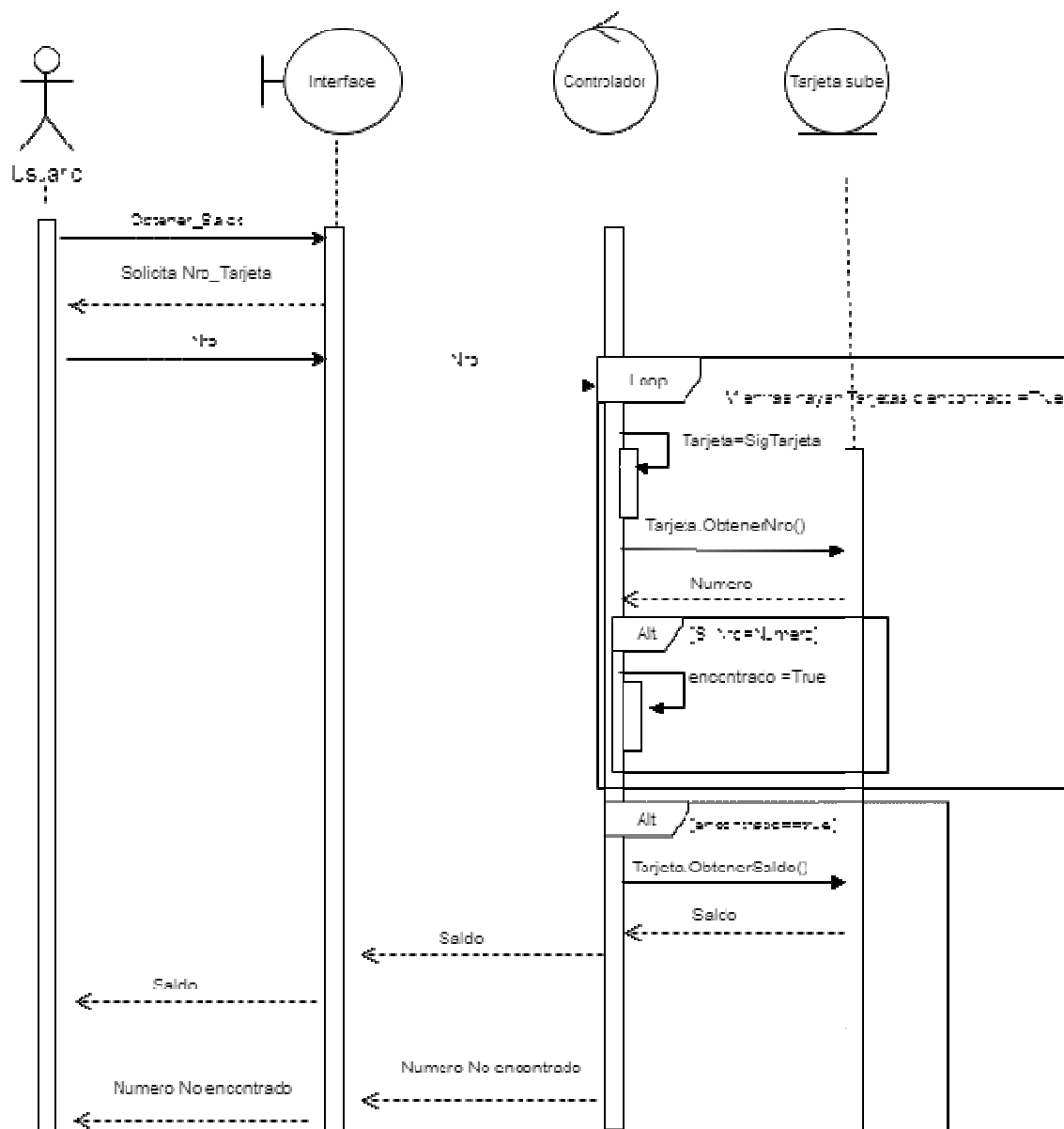
**Escriba el código de cada una de las clases y de los gestores/manejadores, uno por cada módulo.**

**NO SE PERMITE sacar de los gestores/manejadores sublistas para resolver lo solicitado.**

#### **Ejercicio 2**

Codificar en un módulo una clase Controlador, basada en una lista de Python, para almacenar objetos de la clase Tarjeta Sube, desarrollada en el módulo del ejercicio 1. La clase contendor deberá tener los métodos necesarios que resuelvan las siguientes funcionalidades:

1. Agregar tarjetas Sube a la lista, validando que sea un objeto de la clase Tarjeta Sube.
2. Recorrer el contenedor y mostrar el número de las Tarjetas Sube cuyo saldo sea negativo.
3. Agregue los métodos necesarios para resolver el siguiente diagrama de secuencia.



### **Ejercicio Nº 3** (arreglo bidimensional)

La Policía de tránsito de la provincia, necesita una aplicación que maneje la información de los accidentes de tránsito ocurridos en la provincia en el año anterior, discriminados por departamentos y por mes.

Para ello cuenta con el Archivo: "Departamentos.csv" los datos que posee el archivo son los el identificador de departamento que es un numero (1..19) y el nombre del departamento.

El analista funcional del sistema le encarga a usted que desarrolle un programa que lleve a cabo las siguientes funcionalidades

1. Definir la clase departamento.
2. Defina una clase "Accidente" con un atributo que sea un arreglo bidimensional que permita registrar la cantidad de accidentes mensuales ocurridos en cada uno de los 19 departamentos el año anterior. El ingreso es por teclado, se ingresa número de departamento, número de mes y la cantidad de accidentes.
3. Definir una clase Manejador de departamento (usando una lista de Python) para almacenar las instancias de departamento desde el archivo "Departamento.csv"
4. Presente un menú de opciones permita realizar las siguientes tareas
  - a. Dado un numero de mes, mostrar para cada uno de los Departamentos nombre del departamento y el total de accidentes ocurridos en el mes dado.
  - b. Dado un mes, mostrar nombre de departamento y cantidad de accidentes, para el departamento que tuvo la mayor cantidad de accidentes.
  - c. Dado el nombre de un departamento indicar la cantidad total de accidentes ocurridos el año anterior.
  - d. Mostrar los datos registrados con el siguiente formato. La fila "TOTAL" se debe mostrar el total de accidentes del mes.

Departamento	1	2	3	4	...	9	10	11	12
Xxxx	Xx	xxx	xxx	xxx		xxx	Xxx	xxx	xxx
Xxxx	xx	xxx	xxx	xxx		xxx	Xxx	xxx	xxx

#### **Ejercicio 4**

La Universidad Nacional de San Juan necesita una aplicación que maneje la información de las distintas carreras de grado que ofrece a través de las cinco Facultades que la conforman.

Para ello cuenta con dos archivos Facultades.csv y Carreras.csv, separados por punto y coma.

Los datos almacenados en el archivo Facultades son: código de Facultad, Nombre, dirección, localidad, teléfono de contacto.

Los datos almacenados en el archivo Carreras: código de carrera, nombre, fecha de Inicio, duración, título que otorga y código de la facultad en la que se dicta.

Usted es el programador del sistema, y se le pide un programa para procesar ambos archivos, que permita leer los datos de los Facultad y los Datos de Carreras y almacenarlos en la colección arreglo numpy.

Para cargar los datos en memoria, deberá crear una clase ManejadorFacultad, y una clase ManejadordeCarrera

El analista funcional del sistema le encarga a usted que desarrolle un programa que lleve a cabo las siguientes funcionalidades:

- a) Cargar los datos de las Carreras desde el archivo **Carreras.csv**.
- b) Cargar los datos de las Facultades desde del archivo **Facultades.csv**.
- c) Dado el Nombre de una Carrera, mostrar el nombre de la Facultad en la que se dicta.

- d) Para todas las facultades calcular y mostrar la cantidad de carreras que se dictan en cada una de ellas.
- e) Dado el nombre de una Facultad, generar un listado ordenado alfabéticamente, que muestre: nombre y duración de las carreras que en ella se dictan. Para ello, el analista le solicita que en la clase que representa a las carreras, sobrecargue el operador `__lt__`.

### **Ejercicio 5**

La Secretaría de Bienestar Social de la Universidad Nacional de San Juan, tiene un archivo con los beneficiarios de becas, "beneficiarios.csv" y un archivo con las becas otorgadas "becas.csv".

De los beneficiarios de becas, el archivo "beneficiarios.csv", registra: dni, nombre, apellido, carrera que cursa, sigla de Facultad, año que cursa, promedio, idBeca.

De las becas otorgadas, el archivo "becas.csv", registra: idBeca, tipo de beca ('F'- Fotocopia, 'T'-Transporte, 'A'-Ayuda Económica), importe de la beca.

Nota: los archivos no presentan ningún orden en particular, y el separador utilizado es el ";".

El analista de sistemas de la Secretaría le solicita a usted un programa en Python, para procesar ambos archivos.

El programa debe:

1. Crear las clases Beca y Beneficiario. Los datos de los archivos representan el estado de los objetos pertenecientes a estas clases.
2. Cargar los datos de los Beneficiarios en un ManejadorBeneficiarios, leyendo los datos del archivo "beneficiarios.csv"
3. Cargar los datos de las Becas en un ManejadorBecas, leyendo los datos del archivo "becas.csv".

A través de un menú de opciones, llevar a cabo las siguientes funcionalidades:

- a. Leer por teclado un tipo de Beca, e informar los beneficiarios de dicha beca y el importe total que debe disponer la Secretaría para el pago de dicha Beca.
- b. Leer por teclado un dni, informar si el beneficiario tiene más de una beca, mostrando nombre y apellido.
- c. Listar los beneficiarios, ordenados de mayor a menor por Facultad.

**Regla de negocio: para resolver este último punto, el analista le solicita que sobrecargue el operador `>`.**

- d. Listar nombre, apellido y promedio de los estudiantes, que poseyendo un promedio mayor que 8, no poseen beca de ayuda económica.

### **Ejercicio 6**

La obra social de la UNSJ(DAMSU) posee un archivo con los datos personales de los agentes que trabajan para la UNSJ: llamado **pacientes.csv** y un archivo con las atenciones realizadas a sus pacientes durante el mes de abril llamado **atenciones.csv**

El archivo de pacientes registra: dni, Nombre, edad, y la unidad en la que presta servicio (FCEfyN, FASCO, etc.)

El archivo atenciones registra: Dni, fecha de atención, importe de atención.

Nota: los archivos no presentan ningún orden en particular, y el separador utilizado es el “;”.

El analista de sistemas de la Obra Social le solicita a usted un programa en Python, para procesar ambos archivos.

El programa debe:

1. Crear las clases Paciente y Atenciones. Los datos de los archivos representan el estado de los objetos pertenecientes a estas clases.
2. Leer los datos de los Pacientes desde el archivo “Pacientes.csv” y cargarlos en un ManejadorPacientes implementado usando una lista de Python.
3. Leer los datos de las atenciones desde el archivo “Atenciones.csv” y cargarlos en un ManejadorAtenciones implementado usando un arreglo numpy.
4. A través de un menú de opciones, llevar a cabo las siguientes funcionalidades:
  - a. Leer por teclado una fecha, e informar las atenciones realizadas en dicha fecha y el importe total que debe disponer la UNSJ para el pago a la obra social en esa fecha.
  - b. Leer por teclado un dni, e informar Nombre y apellido, y cantidad de atenciones que tuvo.
  - c. Listar nombre, apellido de los pacientes que no tuvieron ninguna atención
  - d. Listar los Pacientes, ordenados por Apellido, de menor a mayor por unidad.

**Regla de negocio: para resolver este último punto, el analista le solicita que sobrecargue el operador “<”.**