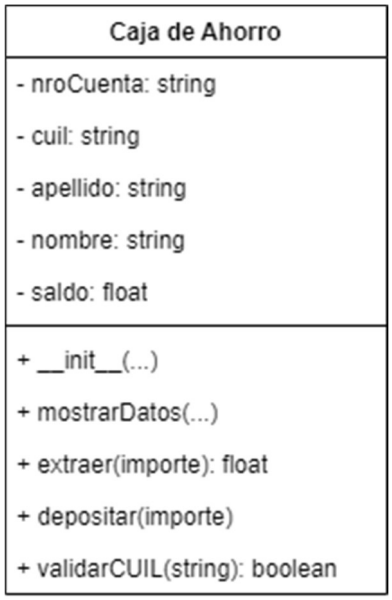
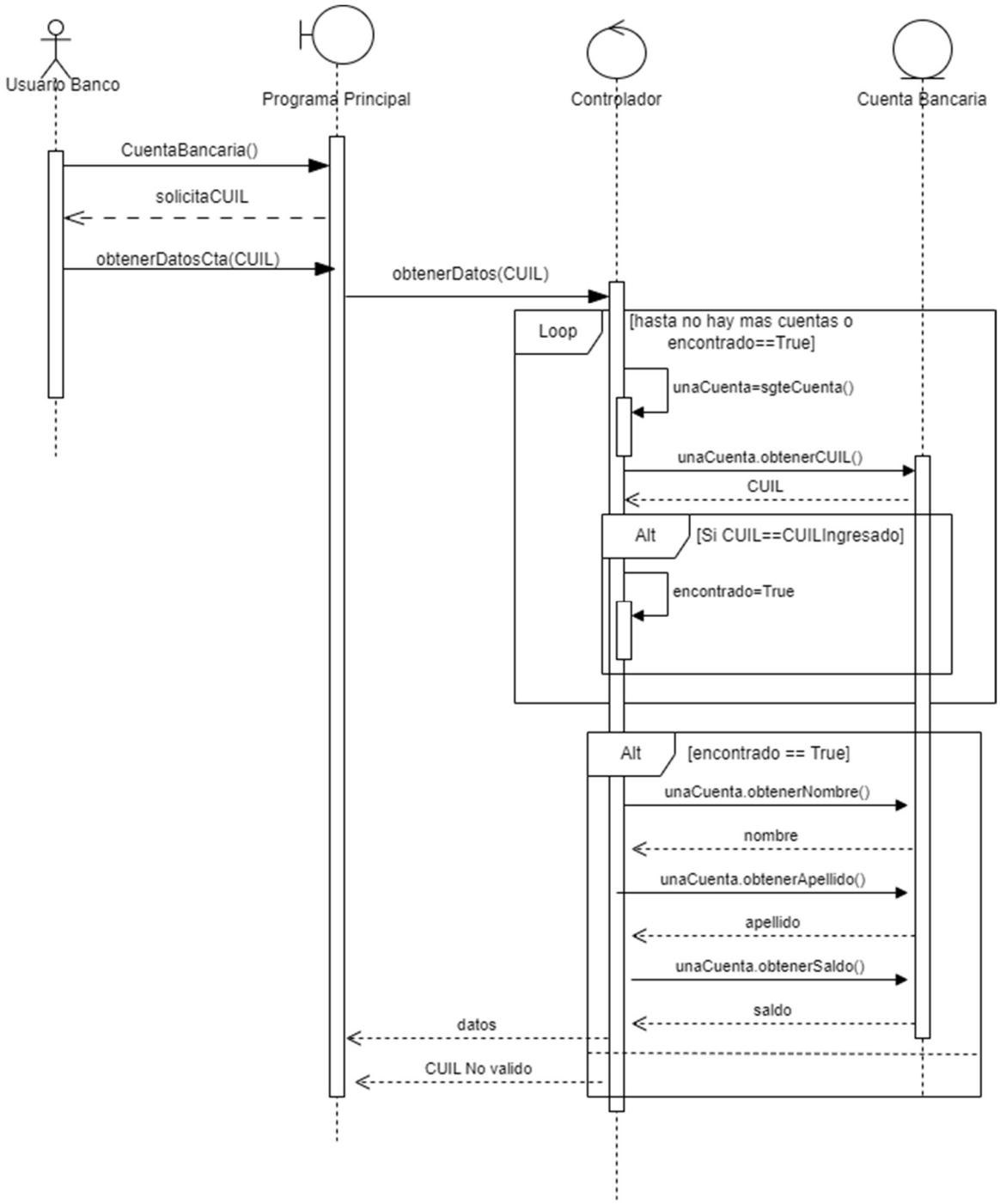
Práctico 2 – Unidad 2
Programación Orientada a Objetos

Ejercicio 1
Dada la siguiente clase en UML:



Codifique la clase utilizando el lenguaje Python, usando módulos.
Escriba una función, test, que lea desde teclado los datos, 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 extraer(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 hacer la operación de extracción.
El método depositar(importe), debe verificar que el importe sea positivo.
Validar el CUIL, según lo expresado en Anexo I.

Ejercicio 2
Codificar en un módulo una clase contenedor, basada en una lista de Python, para almacenar objetos de la clase Caja de Ahorro, desarrollada en el módulo del ejercicio 1.
La clase contendor deberá tener los métodos necesarios que resuelvan el siguiente diagrama de secuencia:



Notas:

La clase contenedora, debe contener un método que permita agregar a la lista objetos de la clase Cuenta Bancaria.

Agregue los métodos necesarios para resolver el diagrama de secuencia planteado, en las clases identificadas.

Anexo I

¿Qué es el CUIL/T?
El **CUIL/T** es el **Código Único de Identificación Laboral** – **Código Único de Identificación Tributaria**. El mismo consta de 11 (once) números. Los 10 (diez) primeros (2 + 8) conforman el **Código de Identificación** y el último conforma el **Digito de Verificación**.
Para obtener los elementos mencionados en el párrafo anterior se aplica el siguiente algoritmo matemático:
XY – 12345678 – Z
XY: Indican el Tipo (Hombre, Mujer, Sociedad o Empresa)
12345678: Número de DNI
Z: Dígito de Verificación
Se determina XY de la siguiente manera:
Hombre = 20
Mujer = 27
Empresa o Sociedad = 30
Se multiplica XY 12345678 por un número de forma separada:
Dado XY = 20, a modo de ejemplo.
2 * 5 = 10
0 * 4 = 0
1 * 3 = 3
2 * 2 = 4
3 * 7 = 21
4 * 6 = 24
5 * 5 = 25
6 * 4 = 24
7 * 3 = 21
8 * 2 = 16
Ahora se suman los resultados de las multiplicaciones como se muestra a continuación:
10 + 0 + 3 + 4 + 21 + 24 + 25 + 24 + 21 + 16 = 148
El resultado calculado en el paso anterior se divide por 11 (once) y se obtiene el resto de dicha división.
148 / 11 = 13 (División Entera)
Resto: 148 – (13 * 11) = 5
Una vez determinado el resto se aplican las siguientes reglas:
Si el resto es igual a 0 (cero), entonces Z (Dígito de Verificación) es igual a 0 (cero).
Si el resto es igual a 1 (uno) ocurre lo siguiente:
• Si es Hombre, entonces Z (Dígito de Verificación) es igual a 9 (nueve) y XY es igual a 23 (veintitrés).
• Si es Mujer, entonces Z (Dígito de Verificación) es igual a 4 (cuatro) y XY es igual a 23 (veintitrés).
• En cualquier otro caso Z (Dígito de Verificación) es igual a 11 (once) menos el resto del cociente.
Resto = 5
11 – 5 = 6
Z = 6
CUIL: 20 – 12345678 – 6

Ejercicio 3

La cadena de farmacias “FarmaCiud” lo contrata a usted como programador para resolver la siguiente situación problemática.
Para cada una de sus 5 sucursales necesita almacenar la facturación que hacen durante los 7 días de la semana para presentarla a la Obra Social Provincia.
Para ello, el analista funcional le solicita a usted que codifique los siguientes requerimientos:
Codificar un Gestor de ventas, que, por reglas de negocio, debe ser un arreglo bidimensional de 5 filas 7 columnas (utilice una lista bidimensional para resolverlo). El Gestor de ventas, se inicializa en todas sus posiciones en 0.

Codificar un menú de opciones que resuelva los siguientes requerimientos funcionales:
a) Leer por teclado, día de la semana, número de sucursal e importe de una factura, acumularlo para ese día y esa sucursal.
b) Leer por teclado una sucursal, calcular el total de facturación para esa sucursal.
c) Leer por teclado un día de la semana, obtener la sucursal que más facturó para ese día.
d) Calcular la sucursal con menos facturación durante toda la semana.
e) Calcular el total facturado por todas las sucursales durante toda la semana.

Ejercicio 4
La empresa “RapiPedido”, lo contrata para desarrollar un sistema que debe procesar los pedidos que llevan las motos que dispone para el delibery de comida.
Cada moto registra: patente, marca, nombre y apellido del conductor, kilometraje.
De cada pedido se registran: patente de la moto que lo tuvo asignado, identificador de pedido, comidas pedidas, tiempo estimado de entrega (en minutos), tiempo real de entrega (se inicializa en cero), precio del pedido.

El sistema debe proveer un menú de opciones que permita:
1. Leer los datos de las motos, desde un archivo denominado “datosMotos.csv” y cargarlos en un Gestor de Motos.
2. Leer los datos de los pedidos, desde un archivo denominado “datosPedidos.csv”, y cargarlos en el Gestor de Pedidos.
3. Cargar nuevos pedidos, leer los datos por teclado, y asignar a una moto el pedido, al solicitar la patente de la moto para asignarla, validar que la moto existe.
4. Leer por teclado número de patente, identificador de pedido, y tiempo real de entrega, modificar en el Gestor de Pedidos, el tiempo real de entrega para ese pedido.
4. Leer por una patente de una moto, mostrar los datos del conductor y el tiempo promedio real de entrega de los pedidos que hizo.
5. Generar un listado para cada moto, para el pago de comisiones a los conductores de las motos:

Patente de Moto: xxxx-xxx
Conductor: xxxxxxxxxxxxxxxxxxxxxxxxxx

Identificador de pedido	Tiempo estimado	Tiempo real	Precio
xxx	xxx	xxx	$ xxxx.xx
xxx	xxx	xxx	$ xxxx.xx
xxx	xxx	xxx	$ xxxx.xx
Total:			$ xxxx.xx

Comisión: $ xxxx.xx (20% del total)

Regla de negocio:
Los archivos no presentan un orden en particular
Ordenar de menor a mayor el Gestor de pedidos por número de patente, de modo que permita resolver eficientemente las búsquedas. Pare ello debe sobrecargar el operador < (__lt__).