



# Tecnicatura Universitaria en Programación

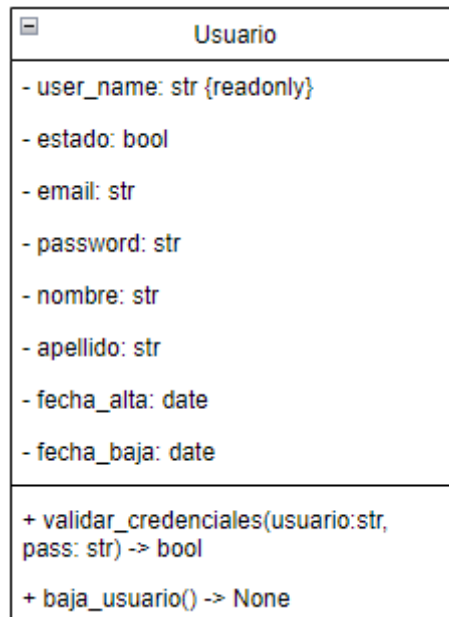
## Programación II

### PRÁCTICA: Unidad 3 y Unidad 4

#### Práctica 1

##### Ejercicio 1

Codificar en python la siguiente clase representada en UML.



##### Requerimientos:

Codificar todos los atributos privados con @property.

Para crear (instanciar) un usuario:

Se ingresa el user\_name y email. Se asigna en fecha\_alta la fecha del día y estado True.

Luego se le solicita al usuario que ingrese su contraseña.

Una vez creado el usuario al mismo no se le puede cambiar el user\_name(sólo lectura).

Luego de creado el usuario, se pueden ingresar los demás datos del usuario.

Al dar de baja el usuario, el mismo no se elimina, si no que su estado cambia a False y se le asigna en fecha\_baja la fecha del día.

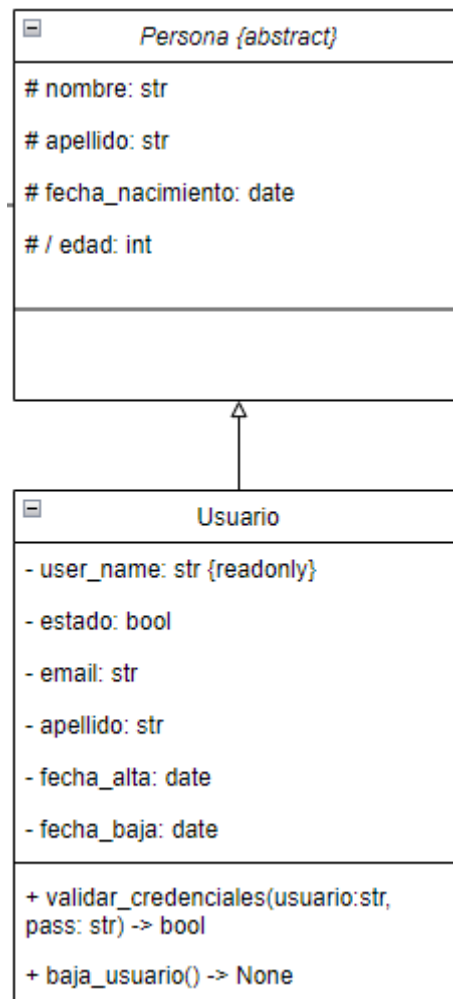
Al validar las credenciales, se retorna verdadero si el usuario y pass ingresado coincide con el user\_name y password el usuario; en caso contrario se retorna False.

Utilizar los siguientes imports y funciones/métodos.

```
from datetime import date  
  
x = date.today()
```

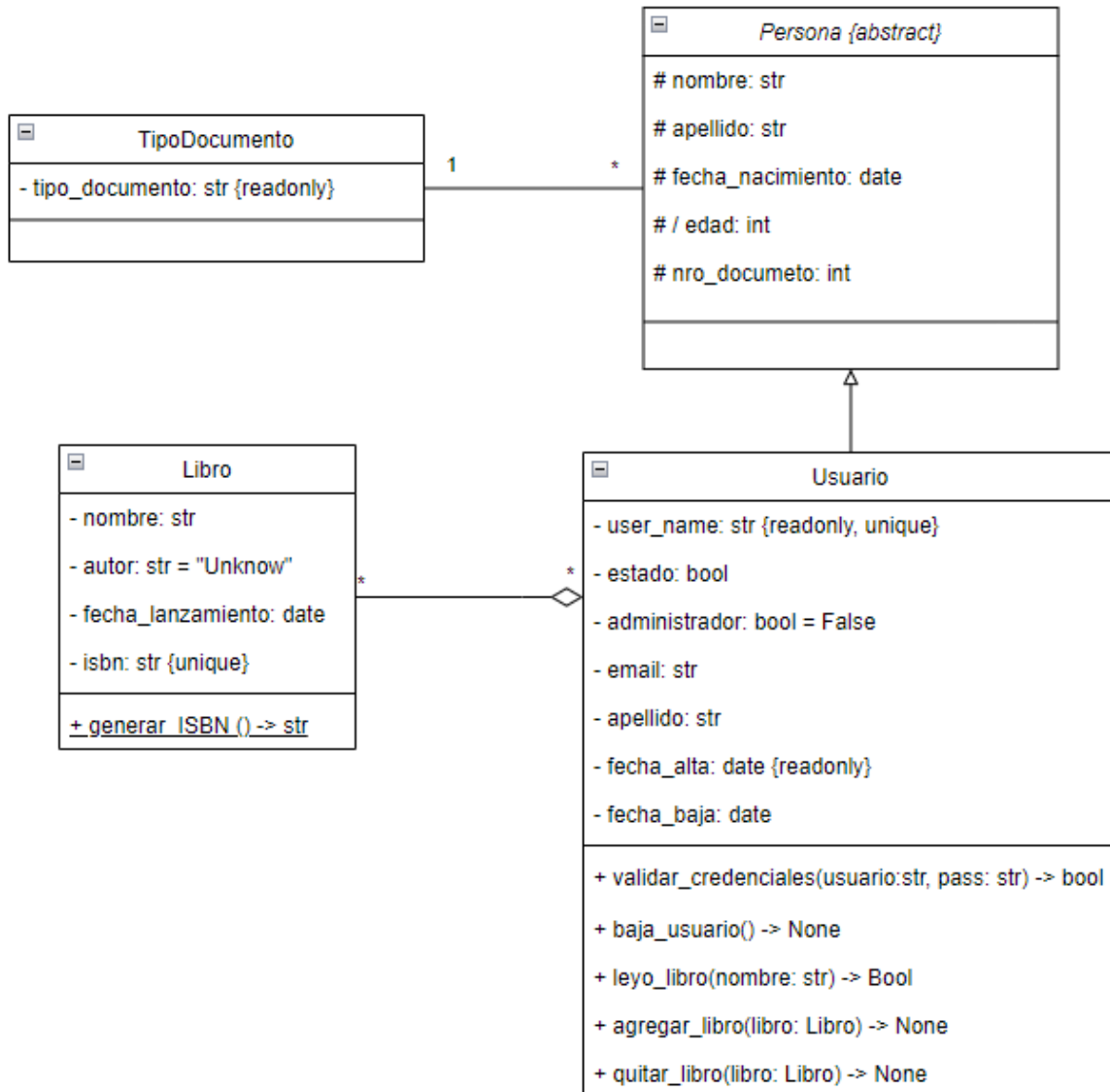
## Ejercicio 2

Adaptar el código anterior para respetar el siguiente diagrama en UML. Aplican los requerimientos del ejercicio 1.



## Ejercicio 3

Codificar en python la siguiente clase representada en UML.



### Requerimientos:

Codificar todos los atributos privados con @property.

Para crear (instanciar) un usuario:

Se ingresa el user\_name y email. El user\_name es único. No se puede crear dos usuario con el mismo user\_name.

Se asigna en fecha\_alta la fecha del día y estado True.

Por default un usuario no es administrador. Al dar de alta un nuevo usuario se pregunta si el mismo es un usuario o no administrador y se almacena la respuesta True/False en el atributo administrador.

Luego se le solicita al usuario que ingrese su contraseña 2 veces, la misma debe coincidir y tener longitud entre 8 y 10 caracteres y tener al menos un carácter numérico.

Una vez creado el usuario al mismo no se le puede cambiar el user\_name(sólo lectura).  
Luego de creado el usuario, se pueden ingresar los demás datos del usuario.

Para seleccionar el TipoDocumento del usuario, se dispone de una lista precargada de objetos TipoDocumento.

Al dar de baja el usuario, el mismo no se elimina, si no que su estado cambia a False y se le asigna en fecha\_baja la fecha del día.

Al agregar un libro, se busca el mismo por ISBN, si se encuentra se agrega a la colección, en caso contrario se muestra cartel aclaratorio.

Al quitar un libro de la colección, se busca el mismo por ISBN y si está dentro de la colección del usuario el mismo se quita, en caso contrario se muestra cartel aclaratorio.

Se puede consultar si un usuario leyó un libro ingresando el nombre del mismo y retornando verdadero o falso si el mismo está en su lista de libros.

Para seleccionar los libros del usuario se dispone de una lista precargada de objetos Libro.

Al validar las credenciales, se retorna verdadero si el usuario y pass ingresado coincide con el user\_name y password el usuario; en caso contrario se retorna False.

Un Libro se da de alta con nombre, autor y fecha de lanzamiento, y se le asigna un ISBN. Si no se conoce el autor, al mismo se le asigna "Unknown". El ISBN es único para cada libro.

Al ingresar al sistema se debe mostrar un menú en bucle con las siguientes opciones:

Menú:

1. Ingresar como usuario
2. Ingresar como administrador
3. Ver libros del usuario...  
*Se busca un usuario por user\_name y se muestran todos los libros que se encuentran en su colección. De no encontrarlo se muestra cartel aclaratorio.*
4. Ver usuario del libro...  
*Se busca un libro por ISBN y se muestran todos los usuarios que leyeron el libro. De no encontrar el libro se muestra cartel aclaratorio.*
5. Salir

Para ingresar como administrador se debe utilizar las credenciales user\_name = "admin" password = "admin". Una vez ingresado se recupera el siguiente menú:

1. Nuevo usuario
2. Nuevo libro
3. Cerrar sesión

Al ingresar como usuario se debe utilizar las credenciales de un usuario ya registrado, si las credenciales son correctas se recupera al usuario y se muestra el siguiente menú.

1. Agregar libro a mi colección
2. Quitar libro de mi colección
3. Cerrar sesión



La opción cerrar sesión regresa el menú para ingresar cómo usuario o administrador.

La opción salir finaliza el bucle.

app.py

```
from usuario import *
from libro import *
from datetime import date

libros = [Libro("El Principito","Antoine de Saint-Exupéry",date(1943,4,6)),
Libro("Juan Salvador Gaviota","Richard Bach",date(1970,1,1))]

administrador = Usuario(usuario='admin',contraseña='admin'
,estado=True,alta=date(2023,6,20),admin=True)

def menu_principal():
    print("1 - Ingresar como usuario")
    print("2 - Ingresar como administrador")
    print("3 - Ver libros del usuario...")
    print("4 - Ver usuario del libro...")
    print("5 - Salir")

def menu_administrador():
    print("1 - Nuevo usuario")
    print("2 - Nuevo libro")
    print("3 - Cerrar sesión")

def menu_usuario():
    print("1 - Agregar libro a mi coleccion")
    print("2 - Quitar libro de mi coleccion")
    print("3 - Cerrar sesión")

while True:
    menu_principal()
    opt = input("Ingrese una opción")
    if opt == "1":
        pass
    if opt == "2":
        pass
    if opt == "3":
        pass
    if opt == "4":
        pass
    if opt == "5":
```

```
print("Saludos!.")  
break
```

## Versiones

Fecha	Versión
02/10/2023	1.0
03/10/2023	1.1 - agrego {readonly} en fecha_alta

## Autores

María Mercedes Valoni