

### Examen Recuperación Recortado - Crea una aplicación con las siguientes características:

- Patrón de diseño MVC (modelo-vista-controlador)
- Uso de paquetes python para controlar lo que se exporta e importa con imports desde distintos sitios de la aplicación.
- Uso de CustomTkinter y/o Tkinter para la GUI
- BD en Sqlite3
- Hay que indicar el tipo de los argumentos en la funciones, así como el tipo devuelto.

### Clases de la Aplicación(Hay que usar constructor cooperativo):

- Clase Abstracta Artículos:
  - Un título.
    - Debes crear usar una función en un archivo [utilidades.py](#) para formatear líneas. Estas deben tener estar tener la primera letra en mayúsculas y no tener espacios en blanco al principio y al final.
  - Un año.
    - El año debe estar comprendido entre el 1500 y el año actual.
  - Fecha de Adquisición.
- Clase Abstracta Prestable
  - Número de días máximo de un préstamo, que en principio será de 31 días. Atributo de clase con el mismo valor para todos los préstamos.
  - Booleano prestado que diga si está prestado o no.
  - Numero\_usuario: número que identifica al usuario que tiene el libro prestado. Si no lo tiene nadie será null.
  - fecha\_prestamo: fecha en la que se prestó el libro. En caso de no estar prestado se pondrá a null.
  - Tendrán un atributo para el cálculo automático de si ha pasado el tiempo máximo de préstamo de un libro.
    - ```
from datetime import date
hoy = date.today()
diferencia = abs((hoy - fecha_bd).days)
```
- Libros
  - Para la BD, consideraremos que solo existe 1 ejemplar de cada libro.
  - Heredan de Artículos y de Prestables.
  - ISBN\_10: hay que validarlo cuando se va a introducir un valor. Para ello, se realizará una validación matemática: se multiplican los dígitos de izquierda a derecha por 10, 9, 8, 7, 6, 5, 4, 3, 2 y 1, respectivamente. La suma de estos productos debe ser múltiplo de 11.
- En la carpeta vista se pondrá una o varias pantallas para la Interfaz gráfica:
  - Se ha de ver el catálogo en una TreeView.
    - Se podrá filtrar algún campo usando una función que se llame buscar\_items - devuelve los artículos encontrados que cumplan las condiciones. Podrá recibir:

- Cero, uno o varios títulos a buscar
  - Cero uno o varios filtros. Pueden ser: autor=", año=2020, disponible=True, etc
- Prestar libro
  - Controlar la disponibilidad de los libros para realizar un préstamo.
- Crear libro
- Eliminar libro
- Editar libro
- Botón que exporta a csv aquellos libros que el plazo máximo de préstamo ha cumplido.
- En el main.py
  - Se creará la BD si no existe.
    - Se crearán las tablas y se realizará una carga inicial de datos.
  - Se ejecutará la pantalla principal