

Columna1	Puntuación	1	0,75	0,5	0,25	0
Diseño POO	1	Las clases están bien diseñadas, correctamente implementadas y siguen los principios de la POO.	Las clases están bien implementadas y siguen los principios de la POO, pero pueden mejorar en su diseño.	Las clases están implementadas, pero pueden tener algunos errores o no siguen completamente los principios de la POO.	Las clases tienen serias deficiencias en su implementación o no siguen los principios básicos de la POO.	Las clases no están implementadas correctamente o no siguen los principios básicos de la POO.
Modelo-Vista-Controlador	0,5	El modelo MVC está correctamente implementado y sigue los principios del patrón de diseño.	El modelo MVC está implementado correctamente y sigue los principios del patrón de diseño, pero unos pocos métodos están donde no corresponde.	El modelo MVC está implementado adecuadamente, pero tiene algunas debilidades en su estructura.	El modelo MVC tiene serias deficiencias en su implementación o no sigue completamente los principios del patrón de diseño.	El modelo MVC no está implementado correctamente o no sigue los principios del patrón de diseño.
Atributos y métodos de clase	0,5	Utiliza atributos y métodos de clase o estáticos correctamente, únicamente cuando es necesario.	Algunos de los atributos y métodos que deberían ser de objeto están definidos como estáticos o de clase, pero la mayoría están bien definidos.	Utiliza atributos y métodos estáticos o de clase más de lo necesario, pero existen bastantes atributos y métodos de objeto cuando es necesario.	Casi todos los atributos y métodos son estáticos o de clase.	Define todos los atributos y métodos como de clase o estáticos.
Control de la visibilidad de atributos	0,5	Utiliza _ y __ para controlar la visibilidad cuando es oportuno y los decoradores @property y @setter para su acceso	Utiliza _ y __ para controlar la visibilidad cuando es oportuno y los decoradores @property y @setter para su acceso para casi todos los atributos que deben ser privados.	Utiliza _ y __ para controlar la visibilidad, pero no utiliza decoradores.	Utiliza _ y __ para controlar la visibilidad pero solo de unos pocos atributos que deben ser privados.	Todos los atributos son públicos
Subdivisión en métodos	0,5	El código está correctamente subdividido en métodos y estos responden al principio de "caja negra" y recibe y devuelve los datos adecuadamente.	El código está correctamente subdividido en métodos y estos responden al principio de "caja negra" y recibe y devuelve los datos adecuadamente, salvo casos puntuales.	El código está subdividido en métodos pero algunos de ellos son demasiado largos.	El código está subdividido en métodos pero todos ellos son demasiado largos.	El código no está dividido en métodos.
Bajo nivel de acoplamiento	0,5	Cada elemento del programa (clase, método, función...) está bien definido y es independiente de los demás.	La mayoría de los elementos del programa son independientes, aunque alguno de ellos dependen unos de otros y no son reutilizables.	La mayoría de los elementos son independientes entre sí, pero existen bastantes elementos que están acoplados entre sí y no pueden reutilizarse.	Solo algunos de los elementos del programa son independientes y pueden reutilizarse en otros proyectos.	Los elementos del programa están muy acoplados entre sí, es imposible reutilizarlos para otros proyectos.
Funcionamiento correcto	1	Todas las especificaciones funcionan correctamente.	Las especificaciones funcionan correctamente, aunque existen pequeños fallos.	La mayoría de las especificaciones funcionan correctamente.	Una pequeña parte de las especificaciones funcionan correctamente.	No funciona.
Uso de cadenas	0,75	Utiliza las cadenas para almacenar información, las recorre y analiza para modificarlas o buscar información.		Utiliza las cadenas y las modifica o analiza de forma sencilla.		No utiliza cadenas.
Uso de listas	0,75	Utiliza las listas para almacenar información, las recorre y analiza para modificarlas o buscar información.		Utiliza las listas y las modifica o analiza de forma sencilla.		No utiliza listas.
Uso de diccionarios	0,5	Utiliza los diccionarios para el almacenaje de datos y para facilitar la lógica del programa.		Utiliza diccionarios simples para consulta de información.		No utiliza diccionarios.
Acceso a ficheros	0,5	Utiliza acceso a ficheros para lectura y escritura de datos en el proyecto.		Utiliza acceso a ficheros solo para lectura o escritura de datos.		No utiliza acceso a ficheros.
Excepciones	0,5	Utiliza control de excepciones para gestionar las eventualidades del código. Lanza excepciones cuando se producen errores y crea excepciones personalizadas para las eventualidades que se pueden producir en el proyecto.	Utiliza control de excepciones para gestionar las eventualidades del código. Lanza excepciones cuando se producen errores.	Utiliza control de excepciones para gestionar las eventualidades del código.	Utiliza en muy pocos casos control de excepciones.	No utiliza control de excepciones.
Grado de dificultad	1	El programa implementa funcionalidades extra con respecto a las establecidas, utiliza librerías, estructuras de datos y elementos del lenguaje avanzados.		El programa cumple los mínimos establecidos en el enunciado.		El programa es muy sencillo.
Código limpio y comentarios	1	Utiliza los principios de código limpio. Nombra las variables, objetos, métodos, etc. utilizando las convenciones de Python, utiliza las reglas de estilo del lenguaje, comenta el código cuando es necesario.				No utiliza los principios de código limpio. No nombra las variables, objetos, métodos, etc. utilizando las convenciones de Python, ni utiliza las reglas de estilo del lenguaje, no comenta el código cuando es necesario.
Type hints	0,5	Se utilizan type hints para la definición de los atributos de las funciones y su valor devuelto; para las variables y para los atributos de una clase.	Utiliza type hints en 2 de las 3 situaciones indicadas.	Utiliza type hints en 1 de las 3 situaciones indicadas.	Apenas utiliza type hints	No se utilizan type hints.