

## Rúbrica Prueba - Programación avanzada en Python

Definir una clase y sus atributos	0,25 Puntos	0,5 Puntos	0,75 Puntos	1 Punto
	Define al menos 2 clases de las requeridas (Campaña, Anuncios, Video, Display y Social)	Define al menos 3 clases de las requeridas (Campaña, Anuncios, Video, Display y Social).  Define atributos de clase (FORMATO y SUB_TIPOS) en al menos una de las clases requeridas (Video, Display o Social), o en todas ellas pero de forma incorrecta.	Define clases Campaña, Anuncios, Video, Display y Social.  Define atributos de clase (FORMATO y SUB_TIPOS) en al menos dos de las clases requeridas (Video, Display o Social), o en todas ellas pero de forma incorrecta.	Define clases Campaña, Anuncios, Video, Display y Social.  Define atributo de clase FORMATO con valores "Video", "Display" y "Social" en las clases Video, Display y Social respectivamente.  Define atributo de clase SUB_TIPOS con valores ("instream", "outstream"), ("traditional", "native") y ("facebook", "linkedin") en las clases Video, Display y Social respectivamente.
Sobrecargar funciones especiales para las clases	0,25 Puntos	0.5 Puntos	1 Punto	2 Puntos
	Sobrecarga función especial __init__ correctamente (incluyendo todos los parámetros requeridos, y asignando en atributos de instancia respectivos) al menos en 1 clase de las requeridas (Campaña, Anuncio, Video), o en más clases pero incorrectamente, o sobrecarga función especial __str__ en clase Campaña retornando una cadena de texto pero sin toda la información solicitada.	Sobrecarga función especial __init__ correctamente (incluyendo todos los parámetros requeridos, y asignando en atributos de instancia respectivos) en al menos 1 clase de las requeridas (Campaña, Anuncio, Video), o en más clases pero incorrectamente.  Sobrecarga función especial __str__ en clase Campaña retornando una cadena de texto pero sin toda la información solicitada.	Sobrecarga función especial __init__ correctamente (incluyendo todos los parámetros requeridos, y asignando en atributos de instancia respectivos) en 2 clases de las requeridas (Campaña, Anuncio, Video), o en todas las clases pero incorrectamente.  Sobrecarga función especial __str__ en clase Campaña para retornar una cadena de texto que incluya nombre, fechas, y cantidades de cada tipo de anuncio que tiene la campaña instanciada.	Sobrecarga función especial __init__ en clase Campaña recibiendo por parámetro los valores de nombre, fecha_inicio, fecha_termino y anuncios (estructura de datos), y asignando en atributos de instancia respectivos.  Sobrecarga función especial __init__ en clase Anuncio recibiendo por parámetro los valores de ancho, alto, url_archivo y url_clic, y asignando en atributos de instancia respectivos.  Sobrecarga función especial __init__ en clase Video recibiendo por parámetro los valores de url_archivo, url_clic y duración, y asignando en atributos de instancia respectivos.

				Sobrecarga función especial __str__ en clase Campaña para retornar una cadena de texto que incluya nombre, fechas, y cantidades de cada tipo de anuncio que tiene la campaña instanciada.
<b>Implementar una especificación técnica en programación orientada al objeto</b>	<b>0,5 Puntos</b>	<b>1 Punto</b>	<b>2 Puntos</b>	<b>3 Puntos</b>
	<p>Aplica encapsulamiento correctamente al menos 1 de las clases requeridas (Campaña, Anuncio, Video), o en más pero de forma incorrecta.</p>	<p>Implementa polimorfismo sobrescribiendo uno de los métodos requeridos (comprimir_anuncio, redimensionar_anuncio) en 1 de las clases requeridas (Video, Display y Social), implementando en cada caso lo solicitado.</p> <p>Aplica encapsulamiento correctamente en 1 de las clases requeridas (Campaña, Anuncio, Video), o en más pero de forma incorrecta.</p> <p>Aplica herencia para definir al menos 1 de las clases requeridas (Video, Display y Social) haciendo que hereden de Anuncio.</p>	<p>Aplica abstracción para definir clase abstracta anuncio y métodos abstractos comprimir_anuncio y redimensionar_anuncio.</p> <p>Implementa polimorfismo sobrescribiendo métodos comprimir_anuncio y redimensionar_anuncio en dos de las clases requeridas (Video, Display y Social), implementando en cada caso lo solicitado.</p> <p>Aplica encapsulamiento correctamente en 2 de las clases requeridas (Campaña, Anuncio y Video), o en todas ellas pero de forma incorrecta.</p> <p>Aplica herencia para definir dos de las clases requeridas (Video, Display y Social) haciendo que hereden de Anuncio.</p> <p>Sobrescribe correctamente constructor de la clase Video, haciendo llamado a constructor de clase padre dentro de él.</p>	<p>Aplica abstracción para definir clase abstracta Anuncio y métodos abstractos comprimir_anuncio y redimensionar_anuncio.</p> <p>Aplica herencia para definir clases Video, Display y Social, haciendo que las tres hereden de Anuncio.</p> <p>Implementa polimorfismo sobrescribiendo métodos comprimir_anuncio y redimensionar_anuncio en clases Video, Display y Social, implementando en cada caso lo solicitado.</p> <p>Aplica encapsulamiento para atributos de instancia nombre, fecha_inicio, fecha_termino y anuncios de clase Campaña, definiendo sus getter y setter.</p> <p>Aplica encapsulamiento para atributos de instancia ancho, alto, url_archivo, url_click y sub_tipo de clase Campaña, definiendo sus getter y setter.</p> <p>Aplica encapsulamiento para atributo de instancia duracion de la clase Video, definiendo su getter y setter.</p> <p>Sobrescribe setters de ancho y alto en Video de forma que no se modifique valores de instancia actual.</p>

				Sobrescribe correctamente constructor de la clase Video, haciendo llamado a constructor de clase padre dentro de él.
<b>Utilizar los principios de colaboración y composición para resolver un problema complejo por etapas, usando la orientación al objeto</b>	0,25 Puntos	0,5 Puntos	1 Punto	2 Puntos
	Aplica colaboración de alguna de las clases que heredan de Anuncio (Video, Display y Social) con la clase Anuncio para definir método estático que muestra parte de los formatos y/o sub tipos disponibles.	Aplica composición para definir atributo privado de instancia "anuncios" en clase Campaña, pero no maneja correctamente la creación de instancias según los parámetros recibidos en el constructor, o aplica colaboración de las clases Video, Display y Social con la clase Anuncio para definir método estático que muestra todos los formatos y sub tipos disponibles.	Aplica composición para definir atributo privado de instancia "anuncios" en clase Campaña, pero no maneja correctamente la creación de instancias según los parámetros recibidos en el constructor.  Aplica colaboración de las clases Video, Display y Social con la clase Anuncio para definir método estático que muestra todos los formatos y sub tipos disponibles.	Aplica composición para definir atributo privado de instancia "anuncios" en clase Campaña; Crea instancias de anuncios dentro del constructor, al momento de añadirlas a la lista de anuncios.  Aplica colaboración de las clases Video, Display y Social con la clase Anuncio para definir método estático que muestra todos los formatos y sub tipos disponibles.
<b>Utilizar flujos de excepción y escritura de archivos</b>	0,25 Puntos	0,5 Puntos	1 Punto	2 Puntos
	Define correctamente excepciones Error y al menos 1 de las excepciones solicitadas (LargoExcedidoError y SubTipoInvalidoError).	Define correctamente excepciones Error y al menos 1 de las excepciones solicitadas (LargoExcedidoError y SubTipoInvalidoError).  Lanza correctamente al menos una de las excepciones solicitadas (LargoExcedidoError y SubTipoInvalidoError)	Define correctamente excepciones Error, LargoExcedidoError y SubTipoInvalidoError.  Lanza correctamente excepción LargoExcedidoError en setter de atributo nombre de clase Campaña, en caso de que el largo supere los 200 caracteres.  Lanza correctamente excepción SubTipoInvalidoError en setter de atributo sub_tipo de clase Anuncio, según el tipo de la instancia que hace uso del método y el valor que se quiere asignar al atributo.	Define correctamente excepciones Error, LargoExcedidoError y SubTipoInvalidoError.  Lanza correctamente excepción LargoExcedidoError en setter de atributo nombre de clase Campaña, en caso de que el largo supere los 200 caracteres.  Lanza correctamente excepción SubTipoInvalidoError en setter de atributo sub_tipo de clase Anuncio, según el tipo de la instancia que hace uso del método y el valor que se quiere asignar al atributo.  Añade correctamente mensaje de excepción lanzada en un archivo .log.