

Primer Parcial Febrero 2023

Para evaluar este examen se tendrá en cuenta tanto el correcto funcionamiento de los ejercicios, como la eficiencia de los mismos, por esa razón un ejercicio podrá obtener la máxima nota siempre y cuando tenga en cuenta ambas premisas.

Los ejercicios deben estar resueltos en una página HTML para demostrar su correcto funcionamiento, pudiéndose utilizar la consola, además se debe utilizar una codificación estricta de JavaScript. **Si un ejercicio no está demostrado, el ejercicio se puntuará con un máximo de la mitad de la nota de dicho ejercicio.**

En cada ejercicio se deberá utilizar los métodos de los diferentes objetos del API de JavaScript para su resolución. En el caso de implementar funcionalidad que ya exista en el API, el ejercicio se verá penalizado por dicha circunstancia.

Por último, no se pueden presentar errores en las pruebas de los ejercicios. Errores de sintaxis o no captura de excepciones implicarán la reducción de la nota o la pérdida total de la puntuación.

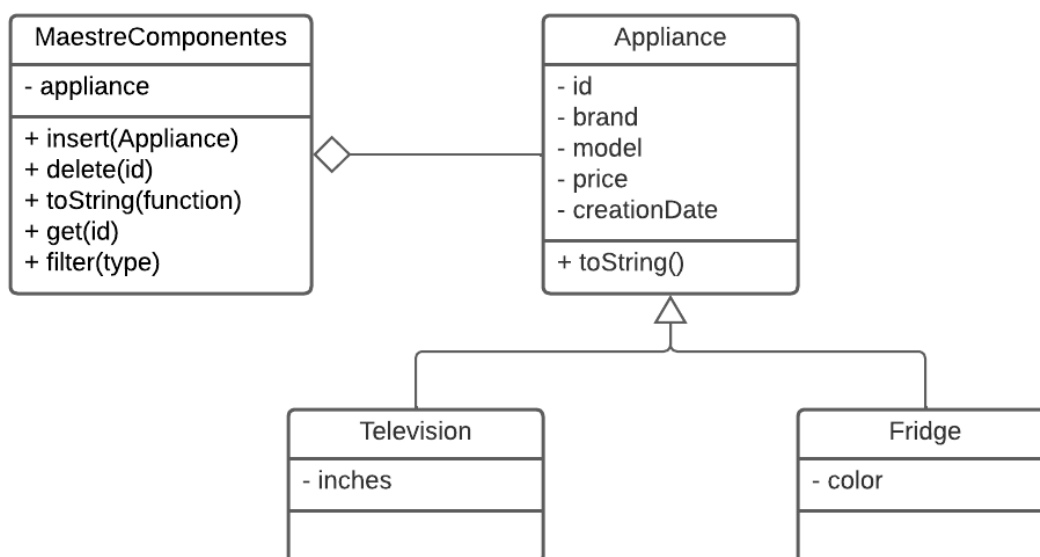
Instrucciones de entrega

Crea una carpeta con tu nombre, ejemplo "LizanoMontalvoPablo", y dentro de ella crea una carpeta por cada ejercicio, ejemplo "ejercicio1", "ejercicio2", etc, las cuales deberán contener la resolución de cada ejercicio. Empaqueta la carpeta principal en un ZIP y súbelo a Delphos.

Si lo ves necesario, explica el ejercicio en un párrafo de la página HTML que hayas creado.

1. Ejercicio único

La tienda de electrodomésticos MaestreComponentes necesita una aplicación para gestionar sus productos en venta para la campaña de rebajas, para ello debemos construir una estructura de objetos como demuestra el siguiente diagrama de clases.



En la siguiente tabla describimos la estructura del objeto **Appliance** que representará los electrodomésticos. La clase debe ser **abstracta**.

Descripción del objeto Appliance	
Propiedad o método	Descripción
Appliance(brand, model, price)	
id	<p>Se trata de un String compuesto por:</p> <ul style="list-style-type: none"> - Las dos primeras letras de la marca. - Las dos primeras letras del modelo. - El año de la fecha de creación. - Un número de secuencia obtenido a partir del número de objetos creados para un año en concreto. <p>Todos tienen que estar separados por guiones, todo en mayúsculas.</p> <p>El siguiente ejemplo representaría el identificador para el objeto creado en el año que será un frigorífico de marca "ACME", modelo "A12345". El identificador será: AC-A1-2022-5</p>
brand	Marca del electrodoméstico.
model	Modelo del electrodoméstico.
price	Precio.
creationDate	Fecha de creación del objeto.
toString()	Método que transforma el objeto en un String.

Tabla 1 Objeto Appliance

Las clases hijas de Appliance especializan los electrodomésticos en dos tipos:

- **Television.**

Descripción del objeto: Television	
Propiedad o método	Descripción
Television(brand, model, price, inches)	Constructor del objeto.
inches	Pulgadas del televisor. Debe ser un entero positivo mayor de 0.

Tabla 2 Objeto Television

- **Fridge.**

Descripción del objeto: Fridge	
Propiedad o método	Descripción
Fridge(brand, model, price, color)	Constructor del objeto.
color	Color del frigorífico.

Tabla 3 Objeto Fridge

El segundo tipo objeto a implementar será **MaestreComponentes** que nos permite gestionar la aplicación. Este objeto debe contener la estructura para almacenar los electrodomésticos de la empresa de forma privada y garantizando el acceso seguro. Los métodos que debe implementar y su funcionalidad quedan recogidos en la siguiente tabla. **El objeto debe ser único.**

Método	Descripción
MaestreComponentes()	Constructor del objeto.
insert(Appliance)	<p>Permite añadir un nuevo electrodoméstico en la aplicación. Debemos chequear que no exista un electrodoméstico creado previamente.</p> <p>El método tiene la peculiaridad de que debe permitir añadir más de un electrodoméstico de forma simultánea.</p> <p>El método debe poder encadenarse.</p>
delete(id)	<p>Nos permite eliminar un electrodoméstico de la aplicación a partir de su identificador. Si el id no está registrada debemos comunicarlo.</p> <p>El método debe poder encadenarse.</p>
toString(function)	<p>Por defecto, el método muestra todos los electrodomésticos registrados en forma de String, y ordenados por el identificador.</p> <p>Si el método recibe una función, ésta debe ser utilizada para ordenar el resultado de la salida.</p>
filter(function)	<p>Devolvemos un iterador con los electrodomésticos que cumplan la condición configurada en la función. Los objetos deben ser devueltos según los vayamos encontrando. No será válido obtener crear un array para luego obtener el iterador en base al array.</p>

Tabla 4 Descripción del objeto MaestreComponentes

Además, el objeto MaestreComponentes debe ser un **objeto iterable e iterador**, en el que devuelva los objetos Appliance.

Nota: deberemos gestionar con excepciones todos los puntos del código donde detectemos algún dato no válido o circunstancia que no nos permita finalizar la ejecución del método. Las excepciones serán valoradas como parte de la implementación de los métodos.

Puntuación:

Las funcionalidades deben estar testeadas para poder ser puntuadas. En el caso de no ser testeadas en una función, la funcionalidad será valorada con un 0.

Funcionalidad	Puntuación
<p>Implementación de la estructura de objetos.</p> <ul style="list-style-type: none"> - Las propiedades deben ser implementados mediante campos privados. - En Appliance las propiedades id y creationDate debe ser de solo lectura. - Los métodos que puedan ser heredados deben ser implementados en el prototipo de los objetos. 	1

Función generador de identificadores	0,5
Validación del identificador.	0,5
Insertar objetos Appliance en la app.	1,5
Borrar objetos Appliance en la app.	1,5
toString de la app	1
Permitir ordenar los objetos con diferentes funciones.	1
Filtrado.	1,5
MaestreComponentes iterable e iterador.	1
MaestreComponentes como Singleton	0,5
Total	10 puntos