

# Primer Parcial Mayo 2021

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 inmobiliaria Acme S.A. ofrece servicios de **venta y alquiler** de inmuebles, y necesita una aplicación JavaScript para la gestión de toda su oferta.

Para cada **inmueble** necesitamos saber el **precio**, la **superficie** en metros cuadrados, y la **dirección**.

Tenemos dos tipos de inmuebles, **viviendas** y **locales comerciales**. Los locales debemos saber el **número de plantas**, y si el número de metros de **fachada**.

Las viviendas tienen un **número de habitaciones** y el **número de baños**. Cada vivienda podrá ser:

- **Casa**: Incluirá el **número de plantas** y si tiene **jardín**.
- **Apartamento**: incluirá la **planta** donde se encuentra ubicado, y si tiene **plaza de garaje** y **trastero**.

Cada inmueble debemos integrar la ubicación en un objeto que incluya la **latitud** y la **longitud**.

Debemos crear un objeto llamado inmobiliaria **inmobiliaria** donde podamos agregar los diferentes inmuebles que disponga la empresa.

### Puntuación:

1. Implementación del objeto **Inmobiliaria** haciendo que solamente podamos instanciar un único objeto (1 punto).
2. Implementación de la jerarquía de clases de **Inmueble** con los getter setter asociados a las propiedades, incluido la ubicación. Los setter deben validar los datos recibidos, en caso de no válido debemos lanzar una excepción. (1 punto)

3. El objeto ubicación debe estar compuesto por dos propiedades, latitud y longitud cuyo formato debe ser:
  - a. Latitud: Ndd°dd'dd'' o Sdd°dd'dd''
  - b. Longitud: Edd°dd'dd'' o Odd°dd'dd''

Donde el carácter 'd' representa un dígito. Ambas propiedades deben estar validadas por una expresión regular. (1 punto)

4. Implementación del método para añadir un nuevo inmueble a la inmobiliaria (0.5 puntos)
5. Haz que el método anterior pueda crearse sobre un número de argumentos indefinidos. Además, el método debe poder encadenarse. (1 punto)
6. Implementación de métodos toString() para cada inmueble. (0.5 puntos)
7. Implementación de **jerarquía abstracta de clases**. (2 puntos)
8. Crear un iterador que devuelva los inmuebles de la inmobiliaria ordenados por número de habitaciones. Deberemos demostrar el funcionamiento del iterador. (1 punto)
9. Crear un método que devuelva los inmuebles en un iterador por tipo de producto. Deberemos demostrar el funcionamiento del iterador. (1 punto)
10. Crea un método que devuelva un iterador con los inmuebles que cumplan un criterio concreto. Ejemplos podría ser los inmuebles de más de 3 habitaciones, los que tengan un precio entre 1000000 y 1200000 euros, los que solamente tenga un baño, etc. (1 punto).