

# Examen primera evaluación

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.

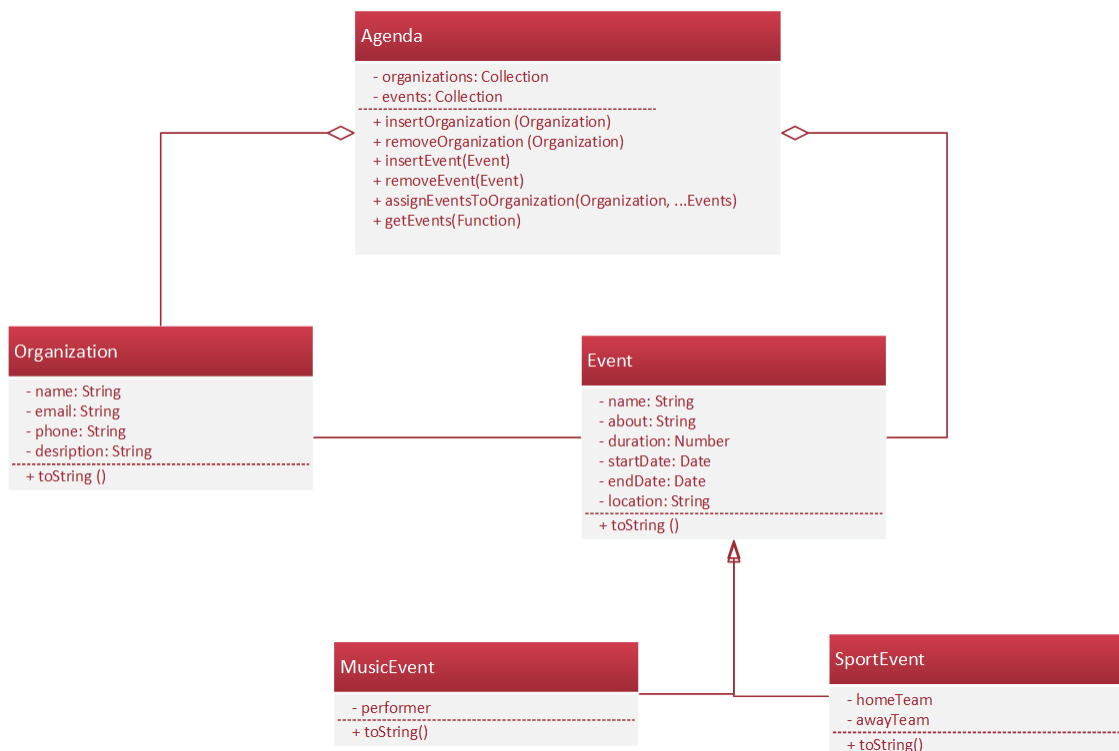
## Instrucciones de entrega

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

Necesitamos implementar la gestión de una **agenda cultural** que recoja los *eventos* organizados por diferentes *organizaciones*. Para ello disponemos del siguiente diagrama de clases.



## Funcionalidad

La funcionalidad a implementar es la siguiente.

## Entidades

En la siguiente tabla describimos la estructura del objeto **Event** que representará los eventos.

Descripción del objeto Event	
Propiedad o método	Descripción
<b>name</b>	Nombre del evento.
<b>about</b>	Descripción del evento.
<b>duration</b>	Duración
<b>startDate</b>	Fecha de inicio
<b>endDate</b>	Fecha de fin
<b>location</b>	Ubicación donde se producirá el evento. Será un string con el siguiente formato grados, minutos y segundos: 51° 30' 30'' N; 0° 7' 32'' O
<b>toString()</b>	Método que transforma el objeto en un String.

Tabla 1 Objeto Event

Las clases hijas de Event especializan los eventos en dos tipos:

- **SportEvent.**

Descripción del objeto: SportEvent	
Propiedad o método	Descripción
<b>homeTeam</b>	Equipo que juega en casa
<b>awayTeam</b>	Equipo visitante
<b>toString()</b>	Método que transforma el objeto en un String.

Tabla 2 Objeto SportEvent

- **MusicEvent.**

Descripción del objeto: MusicEvent	
Propiedad o método	Descripción
<b>performer</b>	Artista o grupo
<b>toString()</b>	Método que transforma el objeto en un String.

Tabla 3 Objeto MusicEvent

El objeto **Organization** recoge la siguiente información

Descripción del objeto Organization	
Propiedad o método	Descripción
<b>name</b>	Nombre de la organización.
<b>email</b>	Correo electrónico
<b>phone</b>	Teléfono
<b>description</b>	Descripción
<b>toString()</b>	Método que transforma el objeto en un String.

Tabla 4 Objeto Organization

## Objeto de aplicación

El objeto donde se implementa la aplicación será **Agenda**. Este objeto debe contener la estructura para almacenar los eventos y las organizaciones, así como sus relaciones 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
<b>insertOrganization (Organization)</b>	Permite añadir una nueva organización en la aplicación. Debemos chequear que no exista una organización con el mismo nombre previamente.
<b>insertEvent (Event)</b>	Permite añadir un nuevo evento en la aplicación. Debemos chequear que no exista una organización con el mismo nombre previamente. Si el evento está asignado a una organización debe ser borrado.
<b>removeEvent(Event)</b>	Borra un evento de la agenda.
<b>removeOrganization (Organization)</b>	Borra una organización de la agenda. También se deben borrar los eventos asociados a la misma.
<b>assignEventsToOrganization(Organization, ...Events)</b>	Asigna uno o más eventos a una determinada organización.
<b>getOrganizationEvents(Organization)</b>	Dada una organización, devuelven todos los eventos asignados a la misma.
<b>getEvents(Function)</b>	Devuelve un iterador con los eventos que cumplan el criterio de selección implementado en la función pasada como argumento.

Tabla 5 Descripción del objeto de aplicación.

Los métodos del objeto deben poder ser multiargumento y deben poder encadenarse.

Además, el objeto **Agenda** debe ser un **objeto iterable**, en el que devuelva los objetos literales con la propiedad event con el evento y organization, la organización que lo prepara.

```
{
    evento: Event,
    organization: Organization
}
```

**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.**

## Rúbrica

Todo el código debe estar debidamente testeado.

Funcionalidad	Puntuación
Implementación de objetos de entidad y excepciones.	0,5
Formato de la ubicación en base a latitud y longitud.	1
Métodos de inserción.	1
Métodos de borrado.	1,5
Método de asignación de un evento a una organización.	1
Método getEvents()	1,5
Iterador con todos los eventos del tipo SportEvent	0,75
Iterador con los eventos entre el 2023 y 2024.	0,75
Método getOrganizationEvents()	1
Agenda como objeto iterable e iterador.	0,5
Agenda como Singleton	0,5
<b>Total</b>	<b>10 puntos</b>

## Anexo: Definición de latitud y longitud

La latitud y la longitud son coordenadas geográficas que se utilizan para especificar la ubicación de un punto en la superficie de la Tierra.

### Latitud:

- La latitud se mide en grados y varía de  $0^\circ$  a  $90^\circ$  hacia el norte y de  $0^\circ$  a  $-90^\circ$  hacia el sur.
- Los puntos al norte del ecuador tienen latitudes positivas, mientras que los puntos al sur tienen latitudes negativas.

### Longitud:

- La longitud se mide en grados y varía de  $0^\circ$  a  $180^\circ$  hacia el este y de  $0^\circ$  a  $-180^\circ$  hacia el oeste.
- Los puntos al este del meridiano de Greenwich tienen longitudes positivas, mientras que los puntos al oeste tienen longitudes negativas.

### Ejemplos:

- Londres, Inglaterra ( $51^\circ 30' 30''$  N;  $0^\circ 7' 32''$  O)
- Berlín, Alemania ( $52^\circ 31' 28''$  N;  $13^\circ 24' 38''$  E)
- París, Francia ( $48^\circ 51' 12''$  N;  $02^\circ 20' 56''$  E)
- Madrid, España ( $40^\circ 24' 59''$  N;  $03^\circ 42' 09''$  O)
- Roma, Italia ( $41^\circ 53' 30''$  N;  $12^\circ 29' 39''$  E)
- Ámsterdam, Países Bajos ( $52^\circ 22' 26''$  N;  $4^\circ 53' 27''$  E)
- Washington DC, Estados Unidos ( $38^\circ 53' 42''$  N;  $77^\circ 2' 11''$  O)
- Nueva York, Estados Unidos ( $40^\circ 52' 51''$  N;  $73^\circ 56' 19''$  O)
- Buenos Aires, Argentina ( $34^\circ 36' 47''$  S;  $58^\circ 22' 38''$  O)
- Lima, Perú ( $12^\circ 02' 35''$  S;  $77^\circ 01' 42''$  O)
- Ciudad de México, México ( $19^\circ 25' 42''$  N;  $99^\circ 07' 40''$  O)
- El Cairo, Egipto ( $30^\circ 03' 45''$  N;  $31^\circ 14' 59''$  E)
- Ciudad del Cabo, Sudáfrica ( $33^\circ 55' 33''$  S;  $18^\circ 25' 30''$  E)
- Estambul, Turquía ( $41^\circ 0' 44''$  N;  $28^\circ 58' 34''$  E)
- Nueva Delhi, India ( $28^\circ 38' 8''$  N;  $77^\circ 13' 48''$  E)
- Wellington, Nueva Zelanda ( $41^\circ 17' 20''$  S;  $174^\circ 46' 37''$  E)
- Sídney, Australia ( $33^\circ 52' 0''$  S;  $151^\circ 12' 0''$  E)
- Moscú, Rusia ( $55^\circ 45' 8''$  N;  $37^\circ 36' 56''$  E)
- Tokio, Japón ( $35^\circ 36' 54''$  N;  $139^\circ 41' 30''$  E)
- Hong Kong, China ( $22^\circ 15' 00''$  N;  $114^\circ 10' 00''$  E)