

Final 2024

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.**

Se proporciona una página para resolver la prueba. El código HTML de la página no podrá ser modificado salvo para cargar un nuevo JS o por resolver algún posible error.

La página está construida con Bootstrap. Se valorará su uso, aunque no es necesario para superar el 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.

Los ejercicios se podrán resolver tanto utilizando el API estándar que ofrece el intérprete de JavaScript, o también utilizando la librería de jQuery. Despliega la web en el servidor web para poder realizarla.

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. Creación de entidades (1 punto)

Crea una de objetos **Person** según la tabla adjunta.

Descripción del objeto Person	
Propiedad o método	Descripción
Member(firstname, lastname, age)	Constructor
firstname	Nombre
lastname	Apellido
Age	Edad
toString()	Método que transforma el objeto en un String.

Tabla 1 Objeto Person

Crea una clase **Employee** que herede de la clase **Person**.

Descripción del objeto Employee	
Propiedad o método	Descripción
Member(firstname, lastname, age, jobtitle, location)	Constructor
jobtitle	Puesto de trabajo.
location	Localización.
toString()	Método que transforma el objeto en un String invocando al su homólogo de la clase padre.

Tabla 2 Objeto Employee

2. Creación de un modelo (3 puntos)

Construye un objeto **Company** que permita gestionar objetos Employee. **El objeto debe ser único** a partir de un patrón **Singleton**. Este objeto debe contener la estructura para almacenar los miembros del club de forma privada y garantizando el acceso seguro. Los métodos que debe implementar y su funcionalidad quedan recogidos en la siguiente tabla.

Método	Descripción
Company()	Constructor del objeto.
insert(Employee)	Permite añadir un empleado. Utiliza como clave la conjunción entre firstname y lastname, no pudiendo añadir un objeto que ya exista. El método tiene la peculiaridad de que debe permitir añadir más de un electrodoméstico de forma simultánea. El método debe poder encadenarse.
delete(firstname, lastname)	Nos permite eliminar un empleado de la aplicación a partir del no nombre. El método debe poder encadenarse.
find(function)	Devolvemos un iterador con los objetos 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.

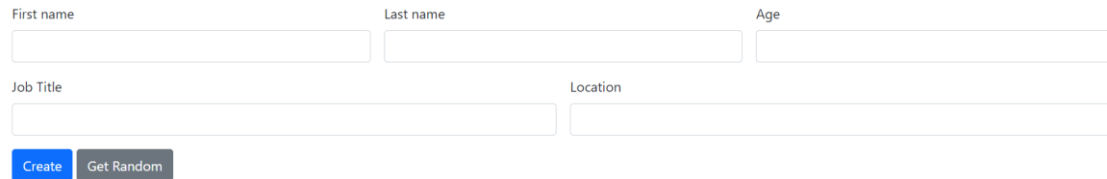
Tabla 3 Descripción del objeto Company

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

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.

3. Validación del formulario (1,5 puntos)

En el index.html de la web proporcionada está dividido en dos partes. Por un lado, tenemos un formulario sobre el cual recogemos una serie de datos de un empleado, y por otro disponemos de una tabla donde mostrar los datos recogidos, la cual dispone de algunos ejemplos. La siguiente imagen muestra el formulario de recogida de datos.



Formulario de recogida de datos con los siguientes campos:

- First name
- Last name
- Age
- Job Title
- Location

Botones: Create, Get Random

Imagen 1 Formulario de recogida de datos

Los datos que se solicitan son los de un empleado, siendo **todos obligatorios**.

Los datos del formulario deben ser **validados** e indicar al usuario **cuál es el error exacto en el que se ha producido** o mostrar la **retroalimentación positiva**.

La validación debe ser final y en línea, es decir, mostrar la retroalimentación con el error según se cumplimenten los datos. Dispones de un modal si necesitas utilizarlo.

Una vez validados los datos del empleado, debes crear un objeto Employee y añadirlo al modelo. La retroalimentación debe estar en base a las posibles excepciones generadas en el caso de que se produzca.

El botón **create** debe capturar el evento *submit* y cancelar su envío al servidor. Una vez recogido los datos el formulario debe resetearse para comenzar desde cero.

4. Listado de empleados (1,5 punto)

Cada vez que se cree un nuevo empleado se debe generar la tabla con los empleados almacenados en el objeto Company a partir de su iterador. La tabla de datos puede crearse utilizando el API de DOM o jQuery, pero **sin utilizar la propiedad innerHTML**.

La siguiente imagen muestra la tabla con los datos que se deberán recoger.

Full Name	Age	Job Title	Location
Vincent Williamson	31	iOS Developer	Washington
Joseph Smith	27	Project Manager	Somerville, MA

Botones: Send, Delete

Imagen 2 Tabla de datos

5. Recogida de datos (1,5 puntos)

En el directorio php dispones de un script denominado **getEmployee.php** al ser invocado devolverá un objeto JSON con los datos de un empleado. El siguiente código muestra los datos obtenidos.

```
{ "firstname": "Joseph", "lastname": "Smith", "age": 27, "jobtitle": "Project Manager", "location": "Somerville, MA" }
```

Al clicar en el botón **Get Random** del formulario deberás recoger este objeto, y a partir de él restaurar con sus datos los campos del formulario utilizando el API de fetch.

6. DOM eliminación de elementos (1,5 puntos)

El botón de **delete** debe borrar el objeto Employee en base a la información de la tabla del modelo. La fila debe borrarse de la tabla, únicamente esa fila.