

Tecnólogo en Análisis y Desarrollo de Sistemas de Información

ID 2205627

Competencia: 35320 - Construir el sistema que cumpla con los requisitos de la solución informática.

Resultado de Aprendizaje: 436472 - Ejecutar y documentar las pruebas del software, aplicando técnicas de ensayo-error, de acuerdo con el plan diseñado y los procedimientos establecidos por la empresa.

Actividad de aprendizaje: Crear una tienda de productos utilizando Programación Orientada a Objetos (Mercadolibre).

La Formación Profesional Integral es el proceso mediante el cual la persona adquiere y desarrolla de manera permanente conocimientos, destrezas y aptitudes e identifica, genera y asume valores y actitudes para su realización humana y su participación en el trabajo productivo y en la toma de decisiones sociales. Acuerdo 12 de 1985 – Unidad Técnicopedagógica.

Lo invito a que de manera dinámica, atenta y propositiva realice el presente instructivo con el ánimo de generar un proceso de aprendizaje (apropiación de conocimientos).

Actividad de aprendizaje: Crear una tienda de productos utilizando Programación Orientada a Objetos (Mercadolibre).

Descripción de la Actividad

Esta actividad pertenece al segundo momento del aprendizaje “mientras aprendo”, busca generar proceso de apropiación de conocimiento en torno a una temática definida.

El instructor procederá a explicar de qué forma a través del DOM podemos manipular el documento html, como podemos crear e insertar elementos y como a través de estos métodos podemos generar una tabla en html de forma dinámica con la información suministrada en el formulario.

Una vez el instructor termine su demostración usted deberá seguir las instrucciones de este instructivo para entregar la evidencia solicitada al final.

Soporte técnico:

Revisar el material de apoyo suministrado por el instructor.

Seguir los procedimientos y explicaciones proporcionadas por el instructor



Herramientas Software:

Para el desarrollo del Laboratorio se requiere tener acceso a internet para descargar la aplicación Visual Studio Code y git.

Procedimiento

Tenga en cuenta que una parte de los ejercicios están basados en actividades y procedimientos que hemos desarrollado con anterioridad y por lo tanto no son tema de este instructivo ya que su desarrollo fue compartido en sesiones anteriores.

Temas vistos con anterioridad que serán necesarios para el desarrollo de la presente actividad:

HTML, CSS, CSS Grid, Flexbox.

Para completar los ejercicios debe tener en cuenta cómo los siguientes ítems:

P.O.O. (Programación Orientada a Objetos)

Cuando hablamos de Programación Orientada a Objetos, estamos hablando del paradigma más popular, no solamente entre la comunidad de desarrolladores sino también a nivel de empresas.

La P.O.O. toma como base los objetos y las clases y estos pueden ser reutilizados las veces que sea necesario.

Un objeto es un ente abstracto que representa alguna entidad y de hecho, en la mayoría de los casos se pueden comparar con objetos de la vida real. Para poner un ejemplo, podríamos hablar de un objeto Motocicleta. Este objeto tiene unas características (atributos) que podrían ser: Marca, Modelo, año, kilometraje, etc. y tiene unos comportamientos (eventos) que podrían ser: encender, apagar, rodar, etc. Existen mucho otros conceptos asociados a la P.O.O. que iremos estudiando durante las próximas semanas.

Construir Objetos

Para la actividad que vamos a desarrollar y poder poner en práctica el concepto de objetos, vamos a tomar como base la plataforma MercadoLibre.

El objeto en este caso será **Automóvil** y a este objeto le añadiremos los siguientes atributos:

- Marca
- Modelo
- Precio
- Año
- Kilometraje recorrido
- Ciudad donde se encuentra ubicado.

Para crear objetos contamos con tres métodos. El primero es con un constructor de clase, el segundo es con una función constructora y el tercero con un iniciador de objetos.

Para el presente ejercicio vamos a crear nuestro objeto con una clase.

Para esto, creamos la clase en el mismo archivo .js y le pasamos los argumentos (atributos) que tiene dicho objeto:



```
class Automovil {  
    constructor(marca, modelo, anyo) {  
        this.marca = marca;  
        this.modelo = modelo;  
        this.anyo = anyo;  
    }  
}
```

Podríamos hacer la comparación del concepto de constructor de clase con unos planos para una casa. Estos planos podrían funcionar no solamente para una casa sino para muchas casas, siempre y cuando estas tengan las mismas dimensiones.

Para el caso de la palabra clave **this**, su comportamiento depende de cómo sea llamado. En este caso, **this** hace referencia al objeto que se está instanciando (auto).

Instanciar un objeto

El siguiente paso consiste en instanciar un objeto. Para esto, utilizamos la palabra clave **new**:

```
let auto = new Automovil("Mazda", "3", 2019);
```

En el caso anterior y teniendo en cuenta el orden en el que está estructurada la clase Automóvil, el atributo "Mazda" hace referencia a la marca, "3" hace referencia al modelo y 2019 hace referencia al año.

Teniendo en cuenta lo anterior, ¿cómo podríamos acceder a los atributos de este objeto? Para esto, llamamos al objeto junto con el atributo creado en el constructor de la clase:

```
alert(auto.marca + " " + auto.modelo);
```

El resultado sería el siguiente:





Funciones Anónimas

Como bien sabemos, las funciones son muy útiles ya que nos permite ejecutar acciones determinadas a través de la llamada a dicha función. ¿Pero que sucede cuando no necesito un nombre para una función y solamente deseo que se ejecuten una serie de acciones en un momento dado?. En este caso, una muy buena opción serían las funciones anónimas. Como su nombre lo indica, son muy similares a las funciones que hemos venido trabajando con anterioridad, con la diferencia que no tienen nombre. En el siguiente ítem, veremos un ejemplo de cómo puedo utilizar funciones anónimas y en que contexto se puedo hacer uso de ellas.

Método AddEventListener

El método `addEventListener` es uno de los mas utilizados en el lenguaje javascript y su función principal es mantener una escucha activa del formulario html.

Para entender esto un poco mejor, hagamos la comparación con una parada de autobús. Al llegar al paradero, pasamos a un estado en donde quedamos atentos a que llegue nuestro bus. Una vez nuestro autobús se acerca a la estación procedemos a subir a este. Lo mismo sucede con el evento `addEventListener`. Una vez cargado el archivo javascript, el método entra en un estado de escucha permanente.

A continuación analicemos como nuestro evento queda en escucha para cuando se produzca una pulsación en el teclado:

```
82 window.addEventListener("keydown", function(event) {  
83     let busqueda = document.getElementById("textoBusqueda").value;  
84     if(event.key == "Enter") {  
85         limpiarVentana();  
86         if (busqueda == "Mazda 3") {  
87             mostrarVehiculo(auto2);  
88             mostrarVehiculo(auto1);  
89             mostrarVehiculo(auto3);  
90             mostrarVehiculo(auto4);  
91         }  
92         else if (busqueda == "Toyota Corolla") {  
93             mostrarVehiculo(auto5);  
94         }  
95         else {  
96             this.alert("No se han encontrado coincidencias.");  
97         }  
98     }  
99 });
```

En la línea 82 encontramos una palabra que no habíamos visto en los anteriores instructivos (**window**). `window` hace referencia a la ventana que contiene el documento DOM (la representación de la página



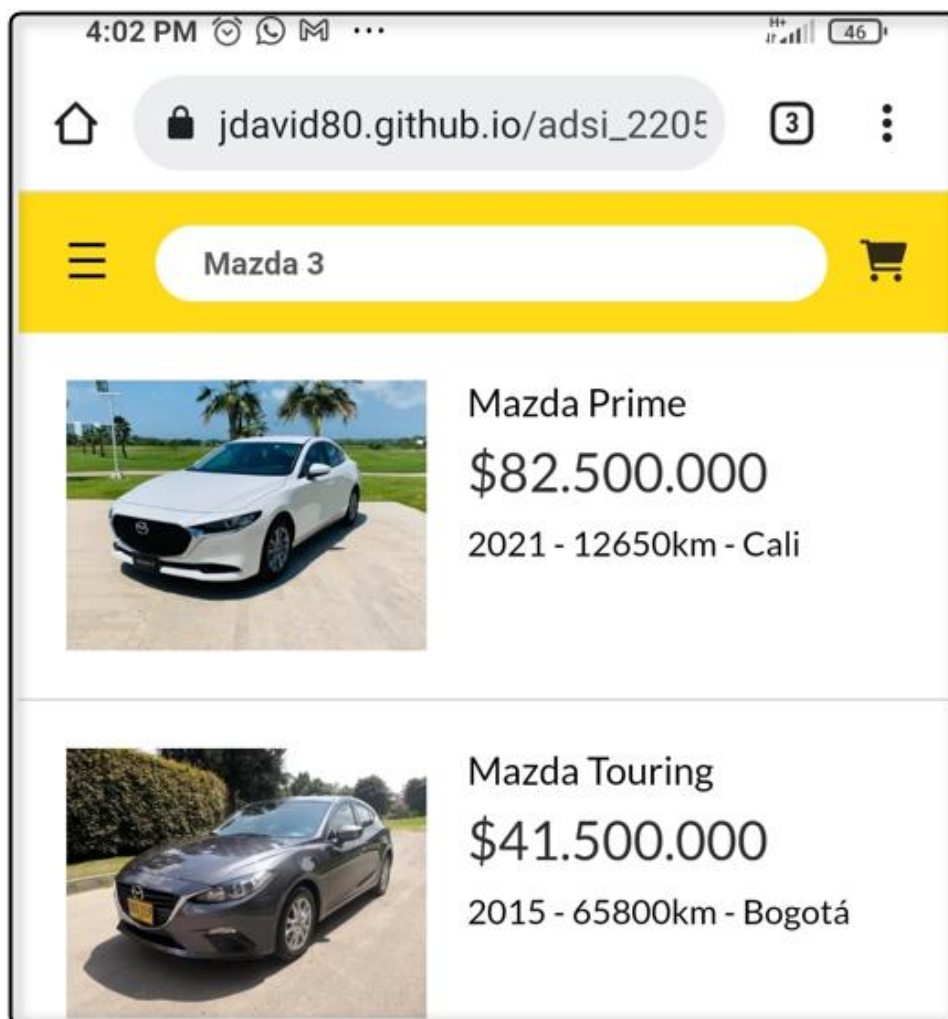
web) y este queda escuchando lo que suceda en la interfaz. Una vez sucede el evento que está esperando, se ejecuta la función anónima.

Observemos que el método `addEventListener` tiene dos parámetros. El primer parámetro contiene el tipo de evento que está esperando (**keydown** o pulsación de tecla) y el segundo parámetro ejecuta la función anónima siempre y cuando el primer parámetro se cumpla.

En la línea 84 pasamos un condicional para identificar que tecla pulso el usuario. Si la tecla pulsada es enter, carga un nuevo condicional para la búsqueda del vehículo (Mazda 3, Toyota Corolla, etc.)

Actividad a Realizar

Teniendo en cuenta el presente instructivo, el aprendiz creara una interfaz con un conjunto de objetos y sus respectivos atributos. Para el caso del siguiente ejemplo, hemos tomado como ejemplo, la plataforma de Mercadolibre, en donde los objetos son los vehículos y los atributos son las características individuales de estos vehículos:





**CENTRO DE GESTIÓN Y DESARROLLO SOSTENIBLE
SURCOLOMBIANO**

INSTRUCTIVO PARA LA EJECUCIÓN DE LA FPI



EVIDENCIA

Teniendo en cuenta que esta actividad de apropiación es evaluativa y genera una evidencia de producto, usted, estimado aprendiz, deberá adjuntar los archivos y la url del sitio web en el cual implemento la actividad solicitada.

Se recomienda seguir activamente las siguientes instrucciones:

- Leer detenidamente y por completo el procedimiento, hasta que pueda identificar claramente qué es lo que se está solicitando.

Entregue el documento por alguno de los canales suministrados por el instructor

Listado de recursos, materiales, equipos, herramientas	<ul style="list-style-type: none">- Computador.- Conexión a internet o datos de celular.- Lápiz y papel							
Tipo de Evidencia	Desempeño		Conocimiento		Producto	X	No produce entrega de evidencia	
Producto y Forma de Entrega	<p>Entregable: Comprimido con los archivos del sitio web y la url para acceder remotamente al sitio.</p> <p>Forma de entrega: Preferiblemente enlace correspondiente en TERRITORIUM, pero en caso de no ser posible, también puede usar email misena, Gmail, u Outlook. El archivo debe tener la siguiente estructura en su nombre: PrimerNombre_PrimerApellido_ActProductoI_RAP436472.</p>							
Fecha de Inicio:	15/03/2022							
Fecha de Fin:	21/03/2022							
Intensidad /duración	8 horas.							