**Universidad Tecnológica Metropolitana**

**4F**

**ACTIVIDAD # 5-7**

**Nombre alumno:**

**Fernando David Sanchez Sacnhez**

**Nombre del profesor(a): Ruth Dominguez**

**Fecha de entrega: Mérida, Yucatán a martes 12 de octubre de 2024**

Practica 5.

**Variable listasPorLetra:**

* listasPorLetra es un objeto que actúa como un "contenedor" para las listas enlazadas, cada una correspondiente a una letra del alfabeto. Este objeto almacena las listas enlazadas en las que cada palabra está clasificada según la primera letra.

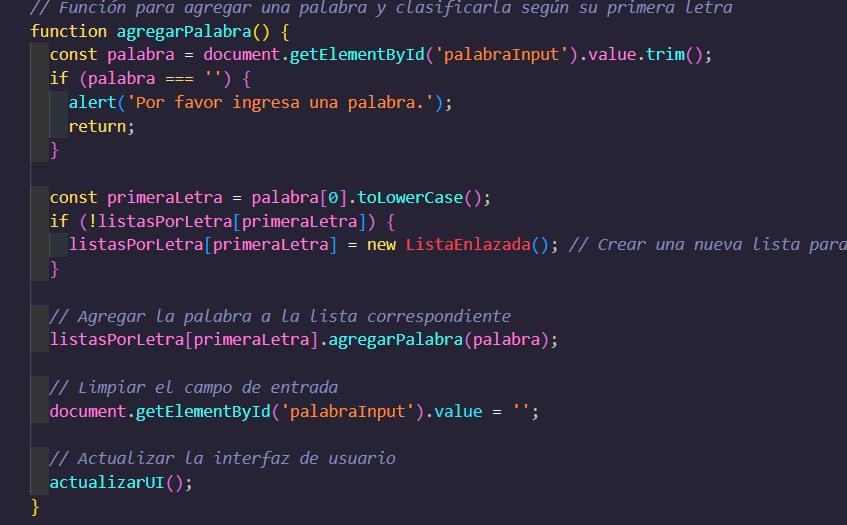


**Función agregarPalabra:**

* **Objetivo:** Esta función permite agregar una palabra introducida por el usuario, la clasifica por la primera letra y la almacena en la lista enlazada correspondiente.

**Pasos:**

* Obtener la palabra ingresada por el usuario del campo de texto.
* Validar que no esté vacía.
* Obtener la primera letra de la palabra, convertirla a minúscula y verificar si ya existe una lista enlazada para esa letra.
* Si no existe, crear una nueva lista enlazada para esa letra.
* Agregar la palabra a la lista correspondiente.
* Limpiar el campo de entrada y actualizar la interfaz de usuario.

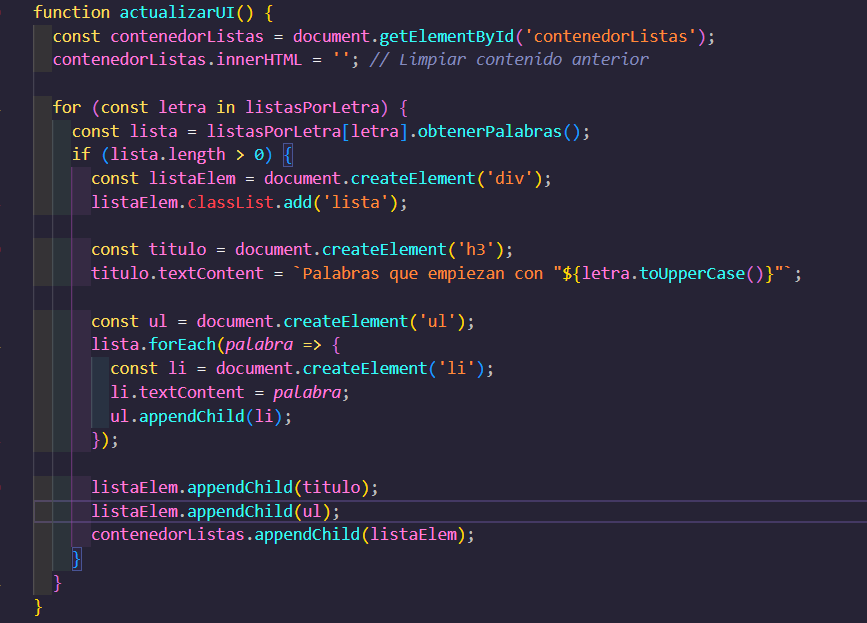


**Función actualizarUI:**

* **Objetivo:** Esta función se encarga de actualizar la interfaz de usuario para mostrar las palabras clasificadas según su primera letra. Cada vez que se agrega una palabra nueva, la interfaz se actualiza para reflejar los cambios.

**Pasos:**

* Limpia el contenido anterior de la interfaz.
* Itera sobre el objeto listasPorLetra, que contiene las listas clasificadas por letra.
* Por cada letra que tiene palabras, crea un nuevo elemento visual que incluye un título con la letra y una lista de palabras.
* Agrega esos elementos al contenedor principal.

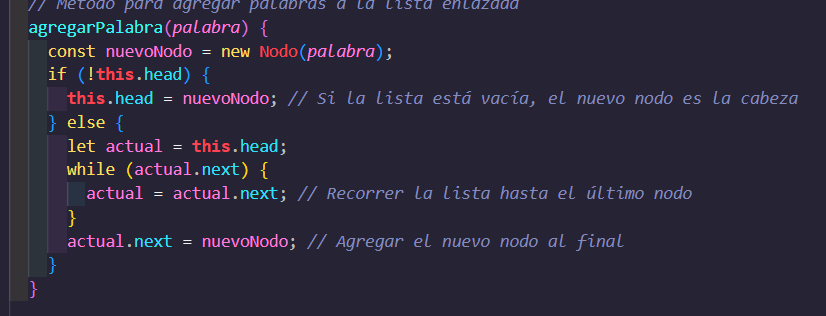


**Clase ListaEnlazada:**

* **Objetivo:** La clase ListaEnlazada es la que gestiona la lista de nodos. Es responsable de agregar palabras, almacenarlas en la lista, y permitir que se puedan recuperar en orden.

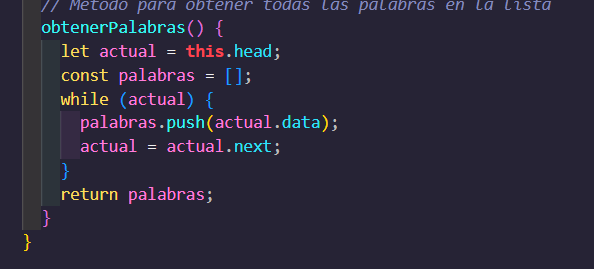
**Método agregarPalabra:**

* **Propósito:** Agregar una palabra a la lista enlazada. Si la lista está vacía, el nodo se convierte en la cabeza de la lista. Si no está vacía, el método recorre la lista hasta encontrar el último nodo y agrega el nuevo nodo al final.



**Método obtenerPalabras:**

* **Propósito:** Este método permite obtener todas las palabras almacenadas en la lista enlazada. Recorre la lista desde el principio hasta el final, agregando las palabras a un resultado.



Practica 6.

**Crear una instancia de la lista enlazada:**

* Aquí se crea una nueva instancia de ListaEnlazada, la cual se utilizará para almacenar las letras de la palabra ingresada en orden inverso.

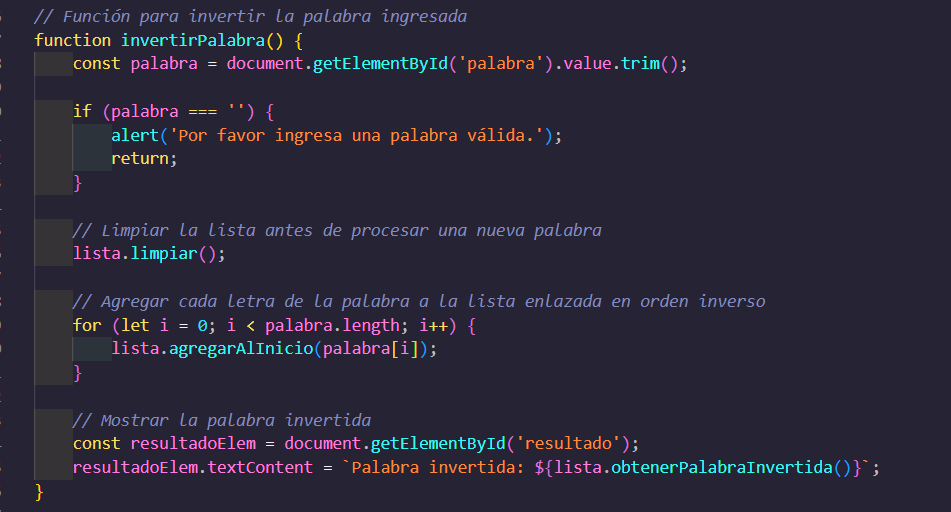


**Función invertirPalabra:**

* **Objetivo:** Esta función invierte el orden de las letras de la palabra ingresada por el usuario y las almacena en una lista enlazada.

**Pasos:**

* Obtener la palabra ingresada por el usuario.
* Validar que no esté vacía.
* Limpiar la lista enlazada antes de procesar la nueva palabra.
* Recorrer cada letra de la palabra y agregarla al inicio de la lista enlazada para invertir el orden.
* Mostrar la palabra invertida en el elemento correspondiente de la interfaz.

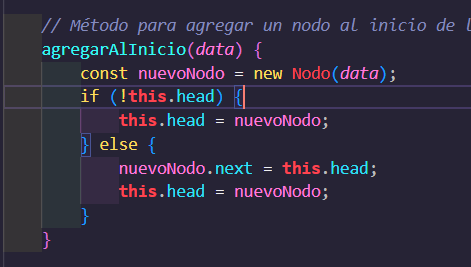


**Clase ListaEnlazada:**

* **Objetivo:** La clase ListaEnlazada es responsable de gestionar los nodos de la lista. Se utiliza para agregar letras en orden inverso, recorrer la lista y construir la palabra invertida.

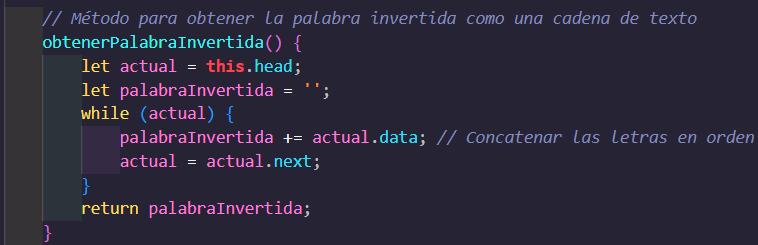
**Método agregarAlInicio:**

* **Propósito:** Este método agrega una letra al inicio de la lista enlazada. Esto permite que las letras se agreguen en orden inverso al ingresarlas una por una.



**Método obtenerPalabraInvertida:**

* **Propósito:** Este método recorre la lista enlazada, construyendo una cadena que contiene las letras en orden inverso.



Practica 7.

**Importar la clase ListaEnlazada:**

* Se importa la clase ListaEnlazada que será usada para almacenar las letras de la palabra y realizar las operaciones de verificación de palíndromo.

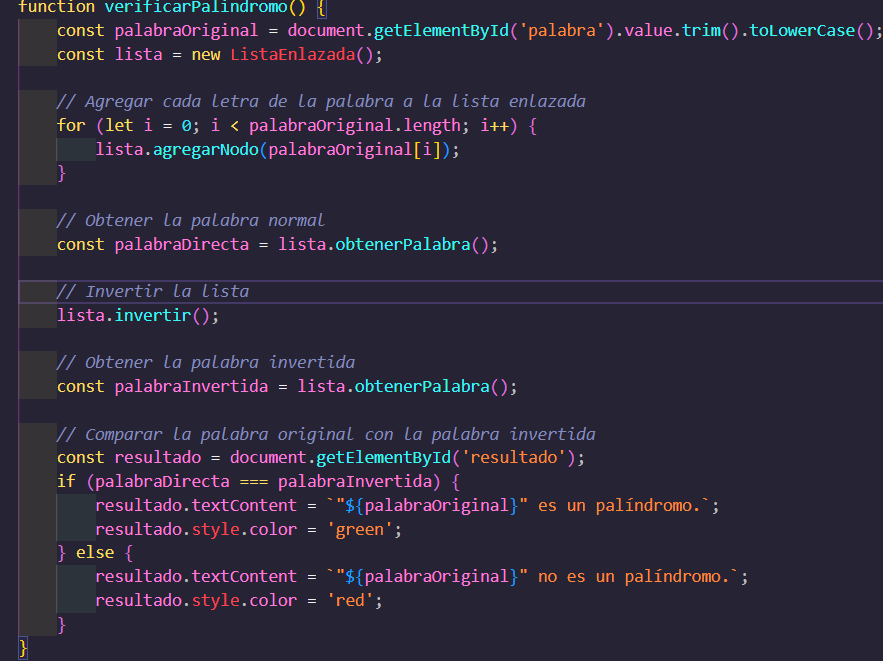


**Función verificarPalindromo:**

* **Objetivo:** Esta función verifica si la palabra ingresada por el usuario es un palíndromo (se lee igual hacia adelante que hacia atrás).

**Pasos:**

* Obtener la palabra ingresada y convertirla a minúsculas para asegurar que la comparación no sea sensible a mayúsculas/minúsculas.
* Crear una instancia de ListaEnlazada para almacenar las letras de la palabra.
* Recorrer la palabra original, agregando cada letra a la lista enlazada.
* Obtener la palabra original directamente desde la lista enlazada.
* Invertir el orden de la lista y obtener la palabra invertida.
* Comparar la palabra directa con la palabra invertida.
* Mostrar el resultado en el DOM, indicando si es o no un palíndromo.

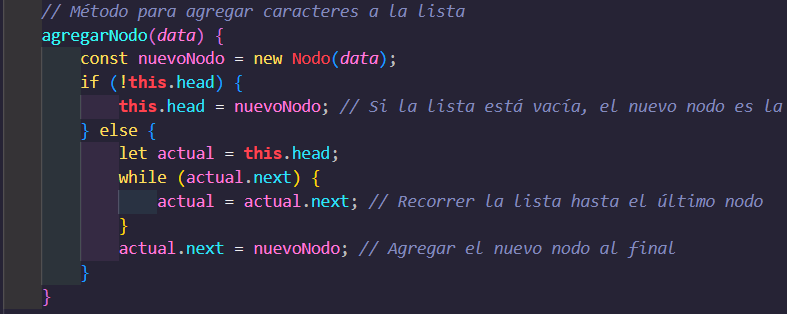


**Clase ListaEnlazada:**

* **Objetivo:** La clase ListaEnlazada es responsable de gestionar los nodos que contienen las letras de la palabra. Se encarga de agregar nodos, invertir la lista y devolver la palabra para verificar si es un palíndromo.

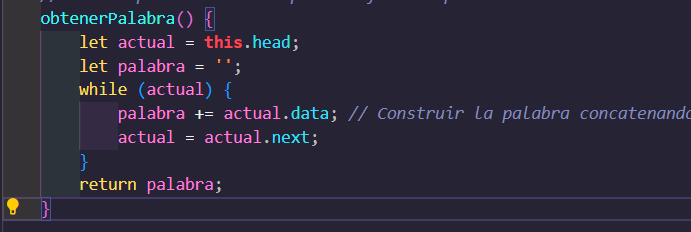
**Método agregarNodo:**

* **Propósito:** Este método agrega un nodo con la letra de la palabra al final de la lista enlazada.



**Método obtenerPalabra:**

* **Propósito:** Este método recorre la lista enlazada para construir una palabra a partir de los nodos que contienen las letras. Se utiliza para obtener tanto la palabra original como la invertida.



**Método invertir:**

* **Propósito:** Este método invierte el orden de los nodos en la lista enlazada, cambiando el puntero siguiente de cada nodo para invertir la secuencia de letras.

