

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Laboratorio de Estructura de Datos

TAREA 5
ESTRUCTURAS FASE2

Carlos Roberto Rangel Castillo

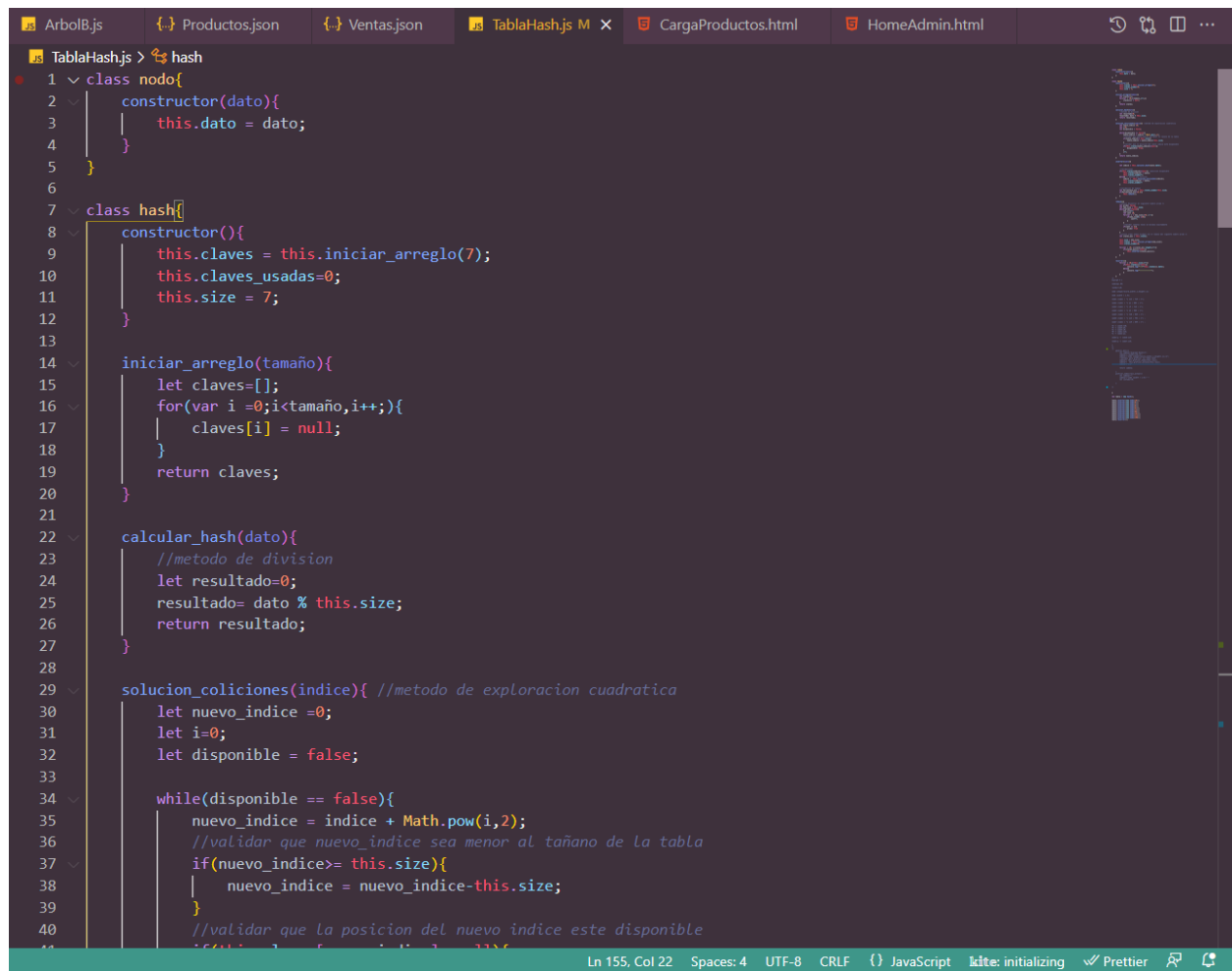
201907636

```
1 class nodoB{
2   constructor(dato, nombre, precio, cantidad){
3     this.dato = dato;
4     this.nombre = nombre;
5     this.precio = precio;
6     this.cantidad = cantidad;
7     //apuntadores de lista - tipo nodoB
8     this.siguiente = null;
9     this.anterior = null;
10    //apuntadores de arbol - tipo pagina
11    this.izq = null;
12    this.der = null;
13  }
14 }
```

Constructor del arbol B en donde se definen los datos a entrar al arbol B.

```
177 class Arbol_B{
178   constructor(){
179     this.raiz = null;
180     this.orden =5;
181     this.altura =0;
182   }
183
184   insertar_nodo(dato, nombre, precio, cantidad){
185     let nuevo = new nodoB(dato, nombre, precio, cantidad);
186
187     if(this.raiz == null){
188       this.raiz = new pagina();
189       this.raiz.raiz = true;
190       this.raiz = this.raiz.insertar_EnPagina(nuevo);
191       //console.log("se inserto el valor "+this.raiz.claves.primerio.dato);
192     }else{
193       if(this.altura==0){
194         let respuesta = this.raiz.insertar_EnPagina(nuevo);
195         if(respuesta instanceof pagina){ // la raiz no se dividio
196           this.raiz = respuesta;
197         }else{
198           this.altura++;
199           this.raiz = new pagina();
200           this.raiz = this.raiz.insertar_EnPagina(respuesta);
201         }
202       }else{ // ya existe mas de una pagina, hay que recorrer el arbol para insertar el nuevo
203         if(this.raiz == null){
204           console.log("la raiz es null ")
205           return;
206         }
207         let respuesta = this.insertar_recorrer(nuevo,this.raiz);
208         if(respuesta instanceof nodoB){ // la raiz se dividio
209           this.altura++;
210           this.raiz = new pagina();
211           this.raiz = this.raiz.insertar_EnPagina(respuesta);
212         }else if(respuesta instanceof pagina){
213           this.raiz = respuesta;
214         }
215       }
216     }
217   }
218 }
```

En el constructor de la clase arbol en donde se define la raíz, el orden y la altura. Para este proyecto se usara un arbol de orden 5. En donde la función insertar_nodo creara el nodo de cada arbol, la variable llamada dato es el id del producto el cual servirá para poder ordenar los elementos dentro del arbol.



```
1 class nodo{
2   constructor(dato){
3     this.dato = dato;
4   }
5 }
6
7 class hash{
8   constructor(){
9     this.claves = this.iniciar_arreglo(7);
10    this.claves_usadas=0;
11    this.size = 7;
12  }
13
14  iniciar_arreglo(tamaño){
15    let claves=[];
16    for(var i =0;i<tamaño,i++){
17      claves[i] = null;
18    }
19    return claves;
20  }
21
22  calcular_hash(dato){
23    //metodo de division
24    let resultado=0;
25    resultado= dato % this.size;
26    return resultado;
27  }
28
29  solucion_colisiones(indice){ //metodo de exploracion cuadratica
30    let nuevo_indice =0;
31    let i=0;
32    let disponible = false;
33
34    while(disponible == false){
35      nuevo_indice = indice + Math.pow(i,2);
36      //validar que nuevo_indice sea menor al tamaño de la tabla
37      if(nuevo_indice>= this.size){
38        nuevo_indice = nuevo_indice-this.size;
39      }
40      //validar que la posición del nuevo índice este disponible
41      if(this.claves[nuevo_indice] == null){
42        disponible = true;
43      }
44      i++;
45    }
46    return nuevo_indice;
47  }
48}
```

La tabla hash es una estructura que nos servirá para almacenar en la lista en vertical los vendedores y estos mismos apuntarán a una lista en horizontal con los productos que están referenciados a estos vendedores.

Se utiliza la exploración cuadrática para el nuevo índice el cual es tener i^2 .

```

11 class lista_adyacentes{
12     constructor(){
13         this.primeros = null;
14         this.ultimo = null;
15     }
16
17     insertar(id,p){
18         let nuevo = new nodo(id);
19         nuevo.ponderacion = p;
20         if(this.primeros == null){
21             this.primeros = nuevo;
22             this.ultimo = nuevo;
23         }else{
24             if(this.primeros == this.ultimo){
25                 this.primeros.siguiente = nuevo;
26                 nuevo.anterior = this.primeros;
27                 this.ultimo = nuevo;
28             }else{
29                 nuevo.anterior = this.ultimo;
30                 this.ultimo.siguiente = nuevo;
31                 this.ultimo = nuevo;
32             }
33         }
34     }
35 }

```

Lista de adyacencia, esta lista nos servirá para las rutas de las bodegas de los proveedores y para poder hacer un calculo de las rutas mas óptimas para los recorridos.

Blockchain

Esta estructura se utilizará para almacenar todas las transacciones de ventas realizadas. EL registro se hace cada vez que se presiona el botón para realizarlo y así se pueda realizar una creación de bloques con la información que se quiera almacenar y hacer inmutable.