

## TAREA 4

### ESTRUCTURAS FASE 1

#### LISTA DOBLEMENTE ENLAZADA:

La lista doblemente enlazada utilizada en el proyecto es específicamente para el almacenamiento de los clientes, creando inicialmente un nodo con los parámetros de id, nombre, correo, y los apuntadores de siguiente y anterior para referenciar los nodos de la lista.

```
class nodo{ //Lista enlazada para lo
    constructor(id, nombre, correo){
        this.id = id;
        this.nombre = nombre;
        this.correo = correo;
        this.siguiente = null;
        this.anterior = null;
    }
}
```

Después se crea la clase de lista doble, añadiendo los métodos de insertar, con las validaciones correspondientes, el método mostrar, para desplegar la lista posteriormente.

```
class listaDoble{
    constructor(){
        this.primeros = null;
    }

    insertar(id,nombre, correo){
        let nuevo = new nodo(id, nombre, correo);

        if(this.primeros == null){ //la lista esta vacia
            this.primeros = nuevo;
        }else{
            let aux = this.primeros;
            while(aux.siguiente != null){
                aux = aux.siguiente;
            };
            aux.siguiente = nuevo;
            nuevo.anterior = aux;
        }
    }
}
```

## ARBOL BINARIO DE BÚSQUEDA (ABB)

En este árbol binario de búsqueda se crea para almacenar y manejar los datos de los proveedores. Respectivamente se insertan los parámetros de id, nombre, dirección, teléfono, correo. Los apuntadores de izquierda y derecha nulos para referenciar correctamente a los nodos.

```
class nodo{ //Este arbol binario de búsqueda será para los
    //Entonces declaro los datos de los proveedores
    constructor(id, nombre, direccion, telefono, correo){
        this.id = id;
        this.nombre = nombre;
        this.direccion = direccion;
        this.telefono = telefono;
        this.correo = correo;
        this.izq = null;
        this.der = null;
    }
}
```

Luego se creó la clase de abb, donde se inicializa el constructor con la raíz del árbol. Dentro de la misma clase se crean los métodos insertar, insertar\_nodo, el método Preorden, Inorden, Postorden, Generar Dot, Generar nodos del dot, Enlazar nodos, con sus respectivas validaciones para anidar las misas funciones.

```

class abb{
    constructor(){
        this.raiz = null;
    }

    insertar(id, nombre, direccion, telefono, correo){
        let nuevo = new nodo(id, nombre, direccion, telefono,
        if(this.raiz == null){
            this.raiz= nuevo;
        }else{
            this.raiz = this.insertar_nodo(this.raiz,nuevo);
        }
    }
}

```

## ARBOL AVL

El árbol AVL se crea para almacenar los usuarios tipo vendedor, la misma que tiene que apuntar a la lista doblemente enlazada de clientes y la lista doblemente enlazada de meses para anidarlas respectivamente y que funcionen juntas.

```

class nodo{
    constructor(vendor,nombre,edad,correo,password){
        this.vendor = vendor;
        this.nombre = nombre;
        this.edad = edad;
        this.correo = correo;
        this.password = password;
        this.izq = null;
        this.der = null;
        this.altura = 0;
        this.lista_clientes = new listaDoble()
        this.lista_meses = new listaDoble_meses()
    }
}

```

En el AVL, se crea una clase para desarrollar los métodos insertar, insertar nodo, altura, altura máxima, raíz izquierda, raíz derecha, raíz izquierda derecha, pero

orden, post orden, in orden, generar dot, nodos, enlazarlos, los métodos de buscar, y recursivamente, insertar cliente, mes y evento.

```
preorden(raiz_actual){...
}

inOrden(raiz_actual){...
}

postOrden(raiz_actual){...
}

generarDot(){...
}

generar_nodos(raiz_actual){ //metodo preorden...
}

enlazar(raiz_actual){...
}

metodo_buscar(vendor){...
}

buscar_recursivo(vendor, nodo_auxiliar){...
}

inner_costumer(vendor,id, nombre, correo){
  //Este método lo hago para insertar un cliente en efec
  let verificar = this.metodo_buscar(vendor)
```

## MATRIZ DINÁMICA

La matriz dinámica creada para el uso del calendario y almacenar respectivamente los meses, 12 meses como máximo, eventos, día y hora.

En el nodo principal se inicializan los parámetros de, evento, día, hora, siguiente, anterior, arriba y abajo, que son los apuntadores de la matriz.

Se crearon las clases con el contenido de lista interna, nodo cabecera, lista cabecera, matriz para anidarlas entre sí.

```

> class lista_interna{ ...
}

//***** CABECERAS *****/
> class nodo_cabecera{ ...
}

> class lista_cabecera{ ...
}

//***** Matriz *****/
> class matriz{ ...
}

// LISTA DE MESES
> class nodo_meses{ ...
}

> class listaDoble_meses{ ...
}

```