

Rutas favoritas

Para las rutas favoritas se utilizo la estructura de tipo lista enlazada simple, se ve de la siguiente manera.

```
private rutaFavorita siguiente = null;
private rutaFavorita raiz = null;
private String ruta;
private int numeroVisitas = 0;
```

Al momento de insertar se verificara si existe en la lista simple, si ya existe entonces esta sera sumada en numero de visitas, de lo contrario se agregara al listado.

```
public static rutaFavorita verificarRuta (String ruta,rutaFavorita entrada){
    rutaFavorita aux = entrada;
    rutaFavorita aux1 = null;
    if(entrada == null){
        return insertar(entrada, ruta);
    }else
    while (aux!=null) {
        if(aux.ruta.equals(ruta)){
            aux.numeroVisitas++;
            return aux.raiz;
        }else
        aux1 = aux;
        aux = aux.siguiente;
    }
    return insertar(entrada, ruta);
}
```

Para poder visualizar el listado este es almacenado en un Jlist desplegable, se verifica que este tenga visitas mayores a 3 visitas, y se agrega al listado del Jlist, reocorre todo el listado simple y solo los que cuenten con la condicion entonces sera vista en la lista.

```
public static ArrayList<String> rutasFav(rutaFavorita entrada,ArrayList<String> listado){
    rutaFavorita aux1 = entrada;
    listado.clear();
    while (aux1!=null) {
        if(aux1.numeroVisitas>=3){
            listado.add(aux1.ruta+" # visitas: "+aux1.numeroVisitas);
        }
        aux1 = aux1.siguiente;
    }
    return listado;
}
```

Arbol-B

El arbol sera manejado como el siguiente ejmplo

```
digraph g {
node [shape = record,height=.1];
node0[label = "<f0> a|<f2> b|<f1> G|<f2> "];
node1[label = "<f0> |<f1> E|<f2> "];
node2[label = "<f0> |<f1> B|<f2> "];
node3[label = "<f0> |<f1> F|<f2> "];
node4[label = "<f0> |<f1> R|<f2> "];
node5[label = "<f0> |<f1> H|<f2> "];
node6[label = "<f0> |<f1> Y|<f2> "];
node7[label = "<f0> |<f1> A|<f2> "];
node8[label = "<f0> |<f1> C|<f2> "];
node9[label = "<f0> F|<f1> D|<f2> G"];
"node0":f2 -> "node4":f1;
"node0":f0 -> "node1":f1;
"node1":f0 -> "node2":f1;
"node1":f2 -> "node3":f1;
"node2":f2 -> "node8":f1;
"node2":f0 -> "node7":f1;
"node4":f2 -> "node6":f1;
"node4":f0 -> "node5":f1;
}
```

Este sera llenado desde una funcion estatica, que crea el archivo .dot, y luego con una funcion estatica se manda el comando para que sea analize el .dot y crear el .png