

1. (Listas Enlazadas)

```
class Polinomio{
protected Nodo primero;
public Polinomio(){primero=null;}
public void agregarCoeficiente(int potencia, double coef){
    primero=new Nodo(potencia, coef,primero);
}
public double valor(double x){
    double s=0;
    for(Nodo r=primero; r!=null; r=r.sigte)
        s += r.coeficiente * Math.pow(x,r.potenica);
    return s;
}
}
```

2. Árboles binarios de búsqueda

```
a. int hojas(Nodo x){
    if(x==null) return 0; //caso árbol vacío
    if(x.izq==null && x.der==null) return 1; //caso hoja
    return hojas(x.izq) + hojas(x.der); //caso general
}

b. int altura(Nodo x){
    if(x==null) return 0; //caso árbol vacío
    return 1+ Math.max(altura(x.izq),altura(x.der));
}

c. String sucesor(Nodo x){
    if(x==null || x.der==null) return null;
    Nodo r=x.der;
    while(r.izq != null)
        r=r.izq;
    return r.valor;
}
```

```
3. static public double raiz(double a, double b, double
    epsilon){
    double fa=f(a), fb=f(b), x;
    while(true){
        //determinar recta y=mx+c
        double m=(fa-fb)/(a-b), c=fa-m*a;
        //determinar raiz de f
        x=-c/m;
        if(b-a<epsilon) break;
        //descartar segmento
        double y=f(x);
        if(signo(y)==signo(fa)){a=x; fa=y;} else{ b=x; fb=y;}
    }
    return x;
}

//recursivo
static public double raiz(double a, double b, double epsilon){
    double fa=f(a), fb=f(b), x;
    return raiz(a,fa,b,fb,epsilon);
}

static public double raiz
(double a, double fa, double b, double fb, double epsilon){
    //determinar recta y=mx+c
    double m=(fa-fb)/(a-b), c=fa-m*a;
    //determinar raiz de f
    x=-c/m;
    if(b-a<epsilon) return x;
    //descartar segmento
    double y=f(x);
    if(signo(y)==signo(fa))
        return raiz(x,y,b,fb,epsilon);
    else
        return(a,fa,x,y,epsilon);
}

4. double raiz(double x, double epsilon, Funcion f, Funcion fl){
    while(true){
        double xnuevo=x-f.valor(x)/fl.valor(x);
        if(Math.abs(xnuevo-x)<epsilon) return xnuevo;
        x=xnuevo;
    }
    return 0;
}

//recursiva
double raiz(double x, double epsilon, Funcion f, Funcion fl){
    double xnuevo=x-f.valor(x)/fl.valor(x);
    if(Math.abs(xnuevo-x)<epsilon) return xnuevo;
    return raiz(xnuevo,epsilon,f,fl);
}
```