

Árboles De Búsqueda Binaria
Aux: Carlos Reveco Díaz
16/8/2005

//pregunta 1

```
public String toMorse(String x){
    String palabra = "";
    char c ;
    for (int i =0 ; i < x.length();i++){
c = x.charAt(i);
palabra = palabra + codigo (c) + " ";
    }
return palabra;
}

public String toPalabra(String x){
    String codigo = "";
    char c ;
    String cod ="" ;
    for ( int k = 0; k < x.length(); k++){
        for (int i = k; i < x.length() ;i++){
            cod += c;
            k=i;}
        codigo = codigo + letra(cod);
        cod ="";}

return codigo;
}
```

//pregunta 2

```
public char letra (String c){

    Nodo aux = raiz;
    for (int i =0 ; i < x.length();i++){
        if(c.charAt(i) == '.'){
            if (aux.izq!= null)
                aux = aux.izq;
        }
        if(c.charAt(i) == '-'){
            if (aux.der!= null)
                aux = aux.der;
        }
    }

return aux.dato;
}
```

```
}
```

```
public StringCodigo( char l){  
    Return Codigo (l , raiz)  
}
```

```
public String Codigo( char l, Nodo x){  
    String Cod = "";  
  
    if(l == (x.izq).dato) return ".";  
    if(l == (x.der).dato) return "-";  
    if(l != (x.izq).dato) return "." + Codigo(l, x.izq) ;  
    if(l != (x.der).dato) return "-" + Codigo(l, x.der) ;  
  
    return "";  
}
```

```
//pregunta 3
```

```
class{
```

```
    Protected String [ ]cod = new String [128];
```

```
    public Morse{  
        for (int i =0 ; i < 128 ;i++){  
            Cod[ i ] = null; }  
    }
```

```
    public void Agregar(char l, String c){  
        cod[ l ] = c;  
    }
```

```
    public Codigo (char l){  
        return cod[ l ];  
    }
```

```
    public char Letra(String c){  
        for (char a= 'a'; a< 'z' ; a++){  
            if cod [a].compareTo(c))  
                return a;  
        }  
    }
```