

Diagrama 1

```
public interface Collection {  
    public Boolean equal (Object o);  
    public static void Coll () {  
        Set <String> s = new Set <String> ();  
        s.add ("i");  
        System.out.println ("set: " + s);  
    }  
}
```

```
public interface List implements Collection {  
    public static void Lis {  
        Lista ll = new Lista ();  
        ll.setl (0);  
        System.out.println (ll.getl (0));  
    }  
}
```

```
public class Pedida {  
    public String getl (0) {  
        return itens - Linha;  
    }  
}
```

```
public abstract class Abstract_List {  
    public abstract Boolean (Object o);  
    public String get itens - Linha (0) {  
        return itens - linha;  
    }  
}
```

```
public class ArrayList extends AbstractList {
```

```
    public static void Array () {
```

```
        Set<String> arl = new Set<String>();
```

```
        s.add(" ");
```

```
        System.out.println (" set: " + s);
```

```
    }
```

```
    public String getarlist () {
```

```
        return arraylist;
```

```
}
```

```
}
```


Diagrama 2

```
public class project {  
    public String name;  
    public String description;  
    source [] sources;  
    reference-sequence [] ref-sequence;  
    feature [] features;  
    alignment [] Alignments;  
}
```

```
public class Alignment {  
    public String name;  
    public String display-name;  
    public String description;  
    AlignmentMember [] alignmentMembers;  
}
```

```
public class AlignmentMember {  
    public String ref-number;  
    AlignedSegment [] aligned-seg;  
}
```

```
public class AlignedSegment {  
    public String refStart;  
    public String refEnd;  
    public String memberStart;  
    public String memberEnd;  
}
```

```
public class Feature {  
    public String name;  
    public String display-name;  
    public String description;  
}
```

```
public class FeatureLocation {  
    FeatureSegment [] feature-seg;  
    Variation [] variations;  
}
```

```
public class PatternLocation {  
    public String refStart;  
    public String refEnd;  
    public String pattern;  
}
```

```
public class ReferenceSequence {  
    public String name;  
    public String display-name;  
    FeatureLocation [] feature-locations;  
}
```

```
public class Sequence {  
    public int sequenceID;  
    public String format;  
}
```

```
public class Source {  
    public String name;  
    Sequence [] sequences;  
}
```

```
public class Variation {  
    public String name;  
    public String display-name;  
    public String description;  
    public String scannerModule name;  
    public String translation type;  
    PatternLocation [] patternLocations;  
}
```

Diagrama 3

```
public class Population {  
    public String race;  
    public String ethnicity;  
    public String primary-language;  
    public String language-family;  
    public boolean pooled;  
    public String type;  
    Individual[] individual;  
}
```

```
public class taxon {  
    public String rank;  
    public String scientific-name;  
}
```

```
public class Geographic-location {  
    public double max-longitude;  
    public double min-longitude;  
    public double max-latitude;  
    public double min-latitude;  
}
```

```
import java.util.ArrayList;  
public class Molecular-sample {  
    public String molecule;  
    ArrayList<Molecular-sample> molecular-samples;  
    ArrayList<Anatomic-location> anatomic-location;  
}
```

```
import java.util.ArrayList;  
public class Anatomic-location {  
    ArrayList<Molecular-sample> molecular-samples;  
}
```

```
public class Individual extends Population {  
    public String father-id;  
    public String mother-id;  
    public String sex;  
    public String birth-date;  
    public int death-date;  
}
```

```
public class Panel extends Population {  
    public long size;  
    public String count-unit;  
}
```