

# Ingeniería de Software II

Nombres: Andrés Felipe Herrera Artunduaga  
Andres Felipe Ocampo Chaguendo  
Daniel Santiago Muñoz Rodríguez  
Isabella Solarte Sandoval  
Carolina Solarte Lopez  
Daniela Riascos Urrego

## 1 Pregunta 1 Valor 2.0

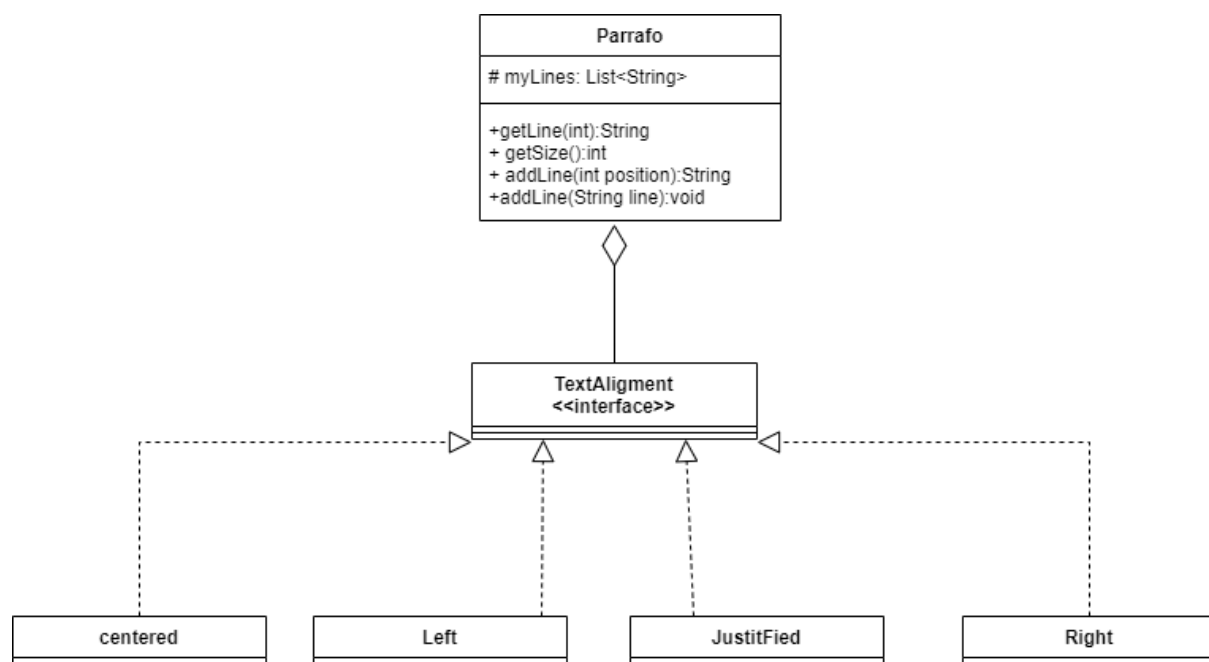
Deseamos implementar un párrafo de texto (Paragraph). Un párrafo es una secuencia de líneas. Cada línea es representada por un String. A continuación una implementación previa:

Fragmento de Código Java de la Clase Paragraph

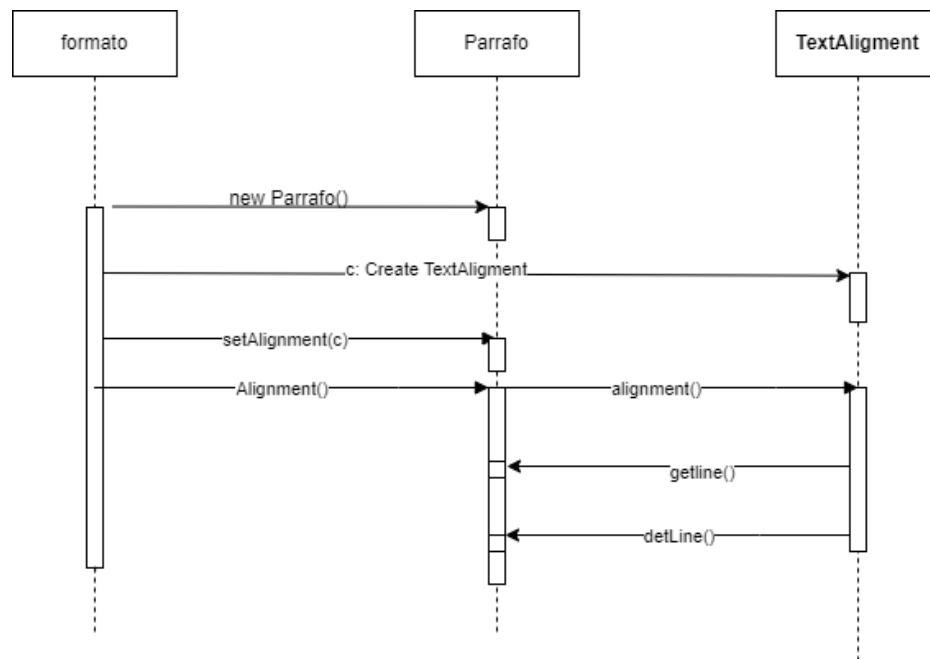
```
1 public class Paragraph{
2     protected List<String> myLines;
3     public Paragraph(){ myLines = new ArrayList<>();}
4     public List<String> textAlignment(){
5         //... algorithm to alignment
6         return myLines;
7     }
8     public String getLine(int position) throws Exception{
9         if (position >= 0 && position < myLines.size())
10             return myLines.get(position);
11         else throw new Exception ("Out of range");
12     }
13     public int getSize(){ return myLines.size();}
14     public void addLine(String line){ myLines.add(line);}
15 }
```

### 1. Pregunta 1

1. El diseño estructural de la solución(diagrama de clases)



- El diseño del comportamiento para la acción de alineación del texto(diagrama de secuencia)



- La implementación de cada una de las interfaces o clases abstractas

```

public interface TextAligment{
    public String alingText(){
    }
}
  
```

## 2. Pregunta 2

```

public String getLine(int position) throws Exception{
    if (position >= 0 && position < myLines.size())
        return myLines.get(position);
    else throw new Exception ("Out of range");
}
  
```

```

@Override
public String getLine(int position) throws Exception{
    if (position > 0 && position < myLines.size())
        return myLines.get(position);
    else throw new Exception ("Out of range");
}
  
```

Handwritten note:  $0 \dots n+1$

- Es claro que está violando el principio de substitución de Liskov. Indique cuales de las leyes de varianza no se cumplen, lo cual causa dicha violación.  
En este ejemplo podemos observar que en Tipo hay Bi-varianza. Sin embargo, podemos observar que en cuanto a las restricciones de salida, la clase sobre escrita es un subconjunto de la clase padre, en ese caso es covariante, pero vemos una contradicción, y es que el padre puede aceptar como resultado 0, pero la clase sobreescrita no, por ende la clase hija no es capaz de sustituir la clase padre.

2. Plantee una solución(diagrama UML) que resuelva el problema, recuerde el principio de favorecer la composición sobre la herencia.

