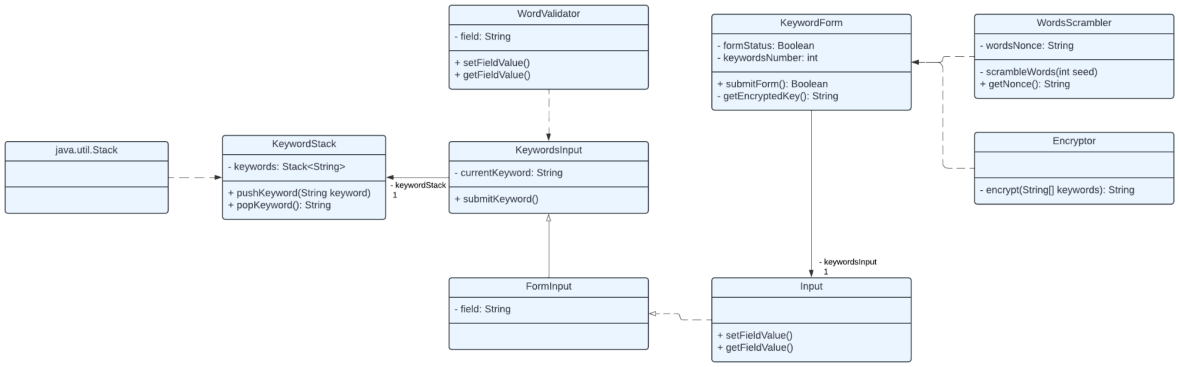
Pedro Henrique de Almeida
148526

Guilherme Cesar dos Santos
133647

A proposta é fazer um serviço de encriptação de dados, gerando um hash qualquer.
Isso deve ser feito de forma semelhante a como são geradas carteiras de criptomoedas.
Isto é, o usuário entraria com N palavras, e a partir delas geraríamos uma resposta encriptografada.



Nas implementacoes podemos ver os principios SOLID.
Em Single Responsability temos a classe KeywordStack que é responsável somente por gerir a stack de palavras-chave.

```java
import java.util.Stack;

public class KeywordStack {
    private Stack<String> keywords;

    public KeywordManager() {
        this.keywords = new Stack<>();
    }

    public void pushKeyword(String keyword) {
        keywords.push(keyword);
    }

    public String popKeyword() {
        return keywords.isEmpty() ? null : keywords.pop();
    }

    public Stack<String> getKeywords() {
        return keywords;
    }
}
```

Quanto ao Open/Closed Principle, temos a classe Input, que é boa para extensão para diferentes tipos de Inputs, enviando que ela seja alterada.

```java
import java.util.Stack;

public class FormInput{
    private String field;

    public void getField() {
      return field;
    }

    public setField(String field){
      this.field = field;
    }

}
```

Por outro lado, podemos observar o principio de Liskov substitution na classe KeywordInput que pode ser substituido por FormInput sem problemas na classe KeywordForm.

```java
import java.util.Stack;

public class KeywordInput extends FormInput implements Input{
    private String currentKeyword;
    private Stack<String> keywordStack;

    public KeywordInput() {
        this.keywordStack = new Stack<>();
        this.currentKeyword = null;
    }

    public void submitKeyword() {
        if(!validateWord(keyword)) return;

        this.currentKeyword = this.field;
        this.keywordStack.push(keyword);
    }

}
```

Por outro lado podemos ver o princípio de Dependency Inversion em KeywordForm que depende de Input, não necessariamente de KeywordInput.

```java
public class KeywordForm {

    private Input keywordInput;
```

Por outro lado, podemos ver também o princípio de interface segregation em Input, uma vez que ela é uma interface bem concisa e com funções bem definidas

```java
public interface Input {
    public String getField();
    public void setField(String field);
}
```