Desenvolvimento Desktop

Rafael Vieira Coelho

Roteiro

- Importação da Biblioteca iText
- Principais Classes
- Escrevendo Texto Simples no PDF
- Atributos do PDF
- Proteger com Senha o Arquivo
- Lista de Itens
- Usando Fontes de Texto
- Adicionando Imagens
- Lendo o PDF

IMPORTAÇÃO DA BIBLIOTECA ITEXT

- Precisamos adicionar a dependência ao arquivo pom.xml do projeto.

- Lembre-se: crie o projeto do tipo Maven!

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    <modelVersion>4.0.0</modelVersion>
    <dependencies>
        <dependency>
            <groupId>com.itextpdf</groupId>
            <artifactId>itextpdf</artifactId>
            <version>5.0.6</version>
        </dependency>
    </dependencies>
    <groupId>org.example</groupId>
    <artifactId>ExemplosPDF</artifactId>
    <version>1.0-SNAPSHOT</version>


</project>
```

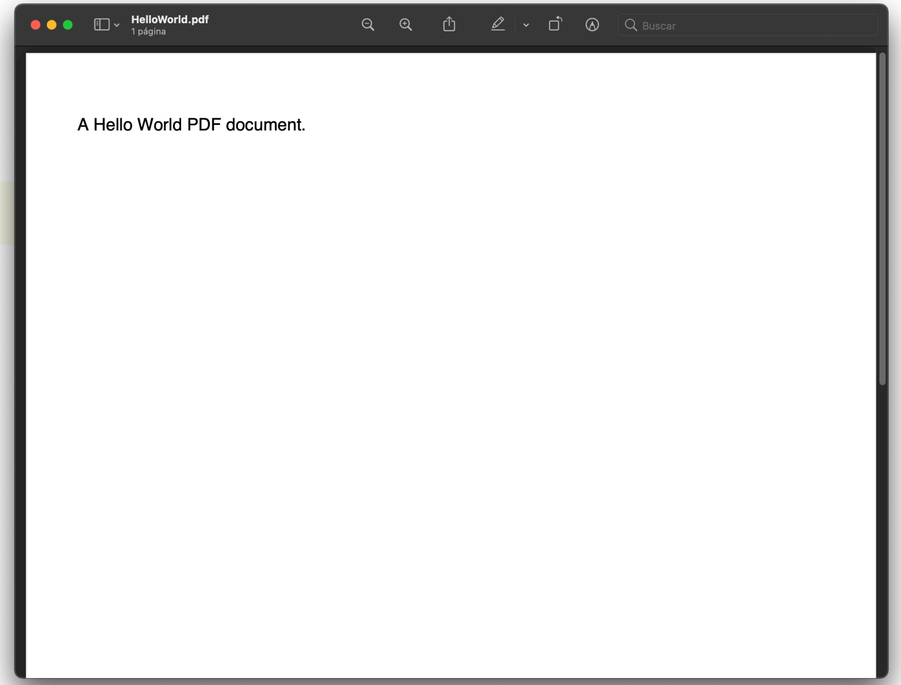
PRINCIPAIS CLASSES DO ITEXT

- com.itextpdf.text.Document: É a classe mais importante da biblioteca iText pois ela representa o documento PDF em si. É o primeiro objeto que precisamos criar. Podemos adicionar conteúdo ao documento e precisamos fechá-lo ao final.
- com.itextpdf.text.Paragraph: Esta classe representa um parágrafo de texto. Nele, podemos configurar o seu alinhamento, espaçamento, etc.
- com.itextpdf.text.Chapter: Representa um capítulo do documento. Deve-se passar um objeto Paragraph como título e uma inteiro como número do capítulo para o seu construtor.
- com.itextpdf.text.Font : Esta classe representa uma fonte e nela podemos definir a família da fonte, seu tamanho, estilo e cor. As diferentes fontes são declaradas como constantes na classe.
- com.itextpdf.text.List : Representa uma lista de itens (objetos ListItems).
- com.itextpdf.text.pdf.PDFPTable : Representa uma tabela que pode ser colocada em uma posição absoluta no documento ou adicionada na ordem dos elementos do documento.
- com.itextpdf.text.Anchor : Trata-se de uma referência ou destino de referência (similar a um link de uma página HTML).
- com.itextpdf.text.pdf.PdfWriter : Usado para escrever e gerar o arquivo PDF.
- com.itextpdf.text.pdf.PdfReader : Usado para ler um arquivo PDF.

```java
import ...

public class JavaPdfHelloWorld {

    public static void main(String[] args) {
        Document document = new Document();
        try {
            PdfWriter writer = PdfWriter.getInstance(
                    document,
                    new FileOutputStream( name: "HelloWorld.pdf")
            );
            document.open();
            document.add(new Paragraph( string: "A Hello World PDF document."));
            document.close();
            writer.close();
        } catch (DocumentException e) {
            e.printStackTrace();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```



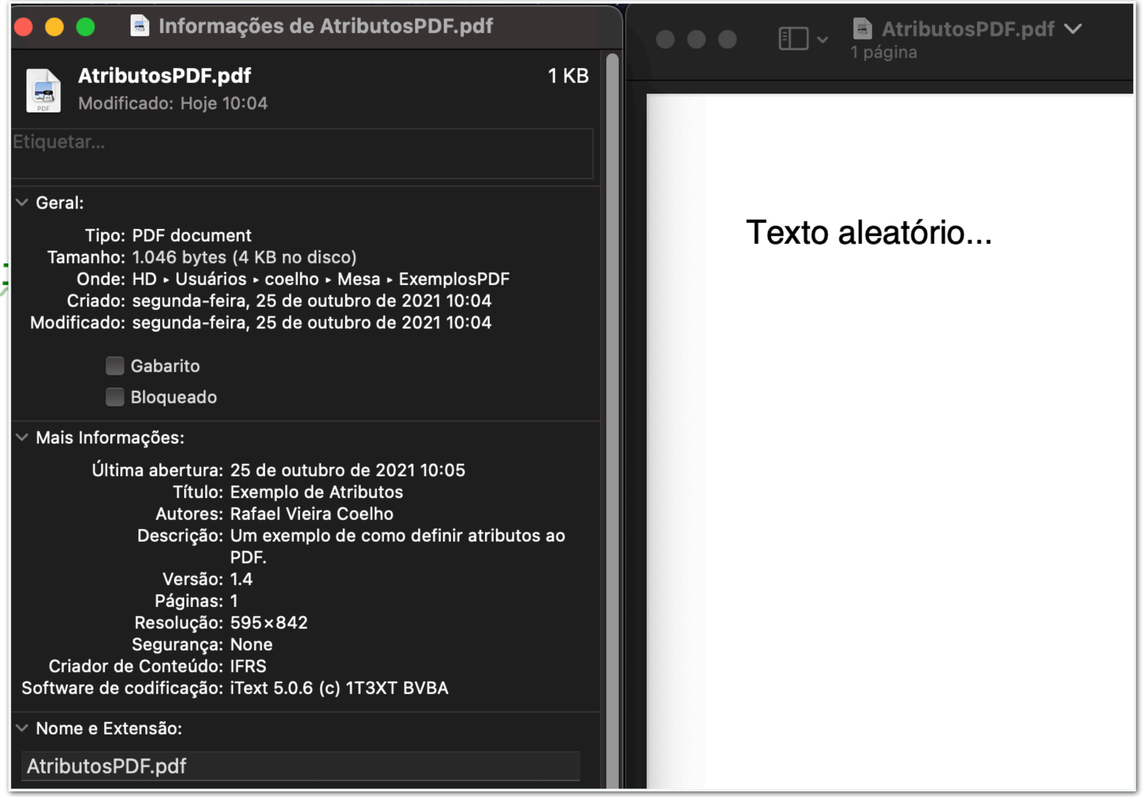
EXEMPLO DE TEXTO SIMPLES

ATRIBUTOS DO PDF

```java
public class SetPDFAttributes {
    public static void main(String[] args) {
        Document document = new Document();
        try {
            PdfWriter writer = PdfWriter.getInstance(document, new FileOutputStream(
                    name: "AtributosPDF.pdf"));
            document.open();
            document.add(new Paragraph( string: "Texto aleatório..."));

            //Set attributes here
            document.addAuthor("Rafael Vieira Coelho");
            document.addCreationDate();
            document.addCreator("IFRS");
            document.addTitle("Exemplo de Atributos");
            document.addSubject("Um exemplo de como def

            document.close();
            writer.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

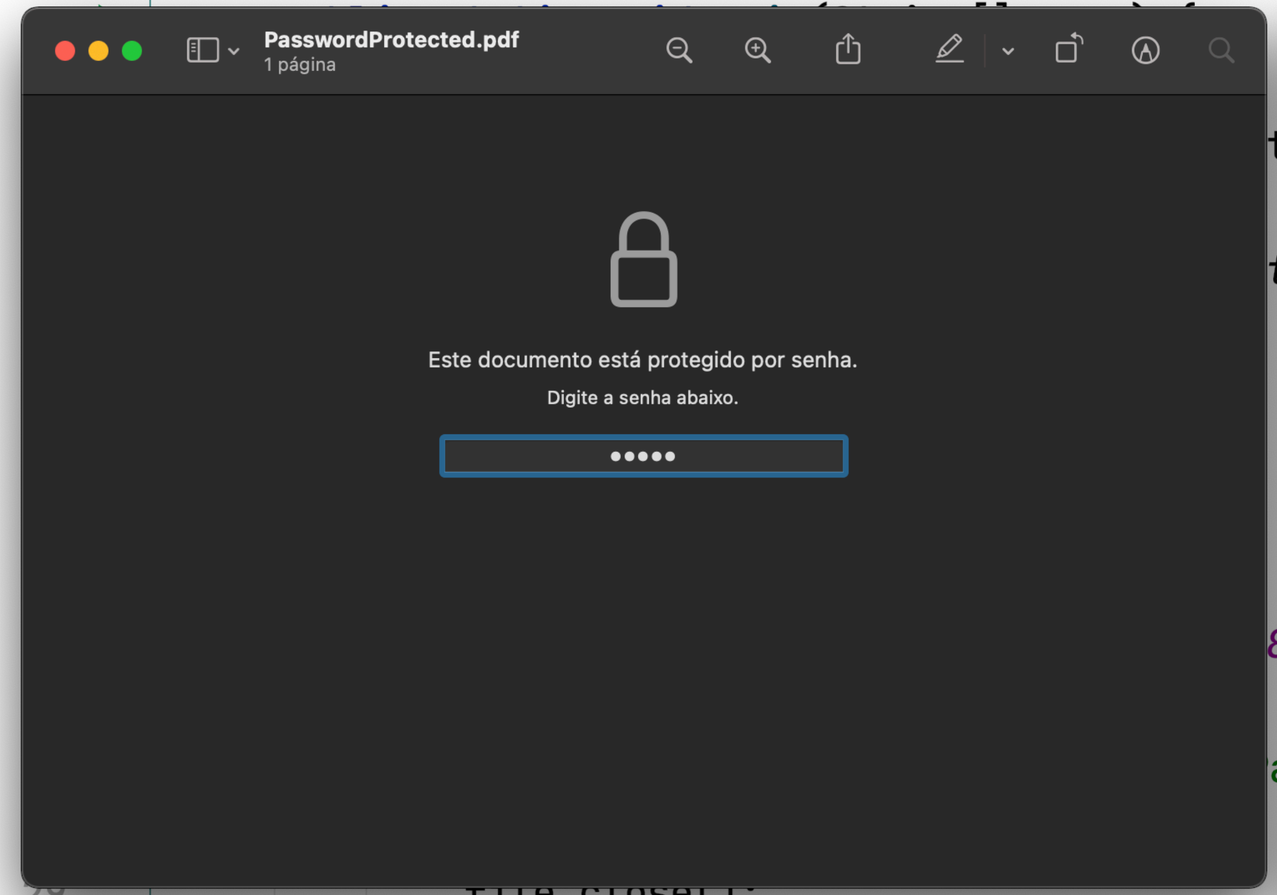


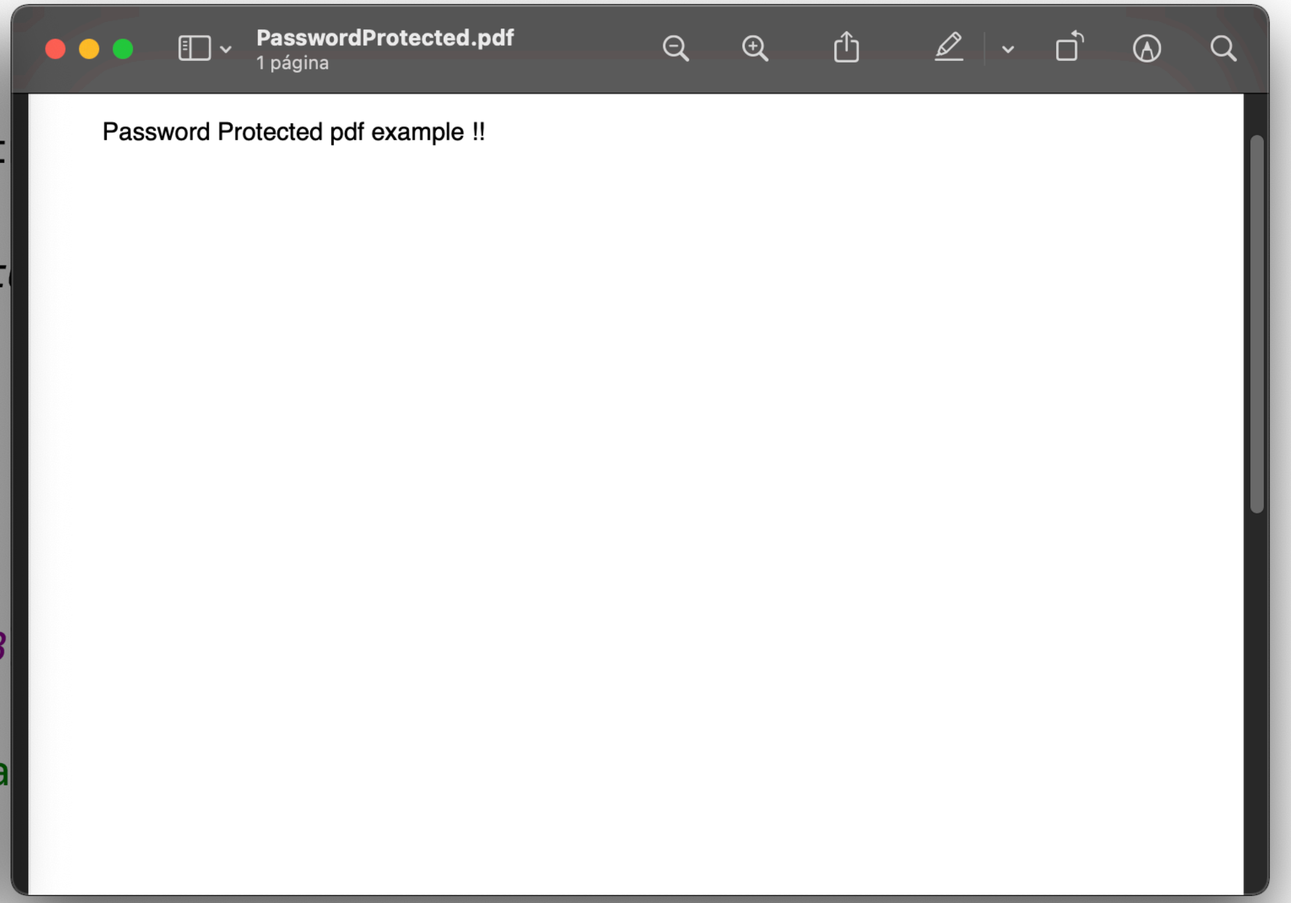
SENHA NO ARQUIVO

```java
11  public class PasswordProtectedPdfExample {
12      private static String USER_PASSWORD = "senha";
13      private static String OWNER_PASSWORD = "coelho";
14
15      public static void main(String[] args) {
16          try {
17              OutputStream file = new FileOutputStream(new File( pathname: "PasswordProtected.pdf
18              Document document = new Document();
19              PdfWriter writer = PdfWriter.getInstance(document, file);
20
21              writer.setEncryption(
22                      USER_PASSWORD.getBytes(),
23                      OWNER_PASSWORD.getBytes(),
24                      PdfWriter.ALLOW_PRINTING,
25                      PdfWriter.ENCRYPTION_AES_128);
26              document.open();
27              document.add(new Paragraph( string: "Password Protected pdf example !!"));
28              document.close();
29              file.close();
30          } catch (Exception e) {
31              e.printStackTrace();
32          }
33      }
```

SENHA NO ARQUIVO

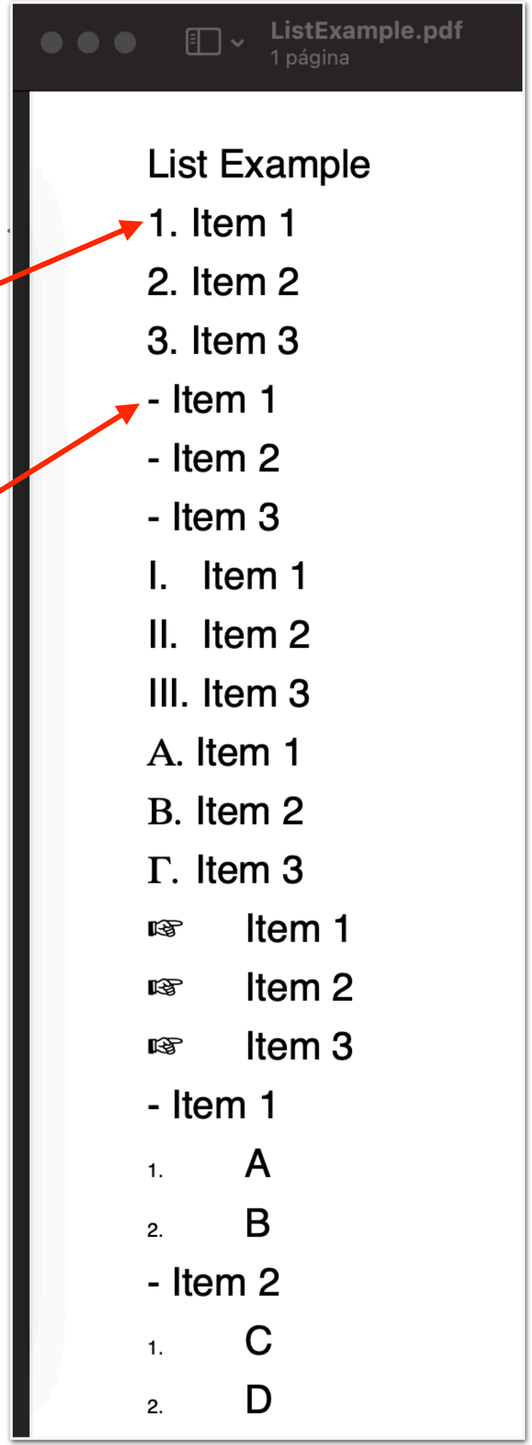
```java
11  public class PasswordProtectedPdfExample {
12      private static String USER_PASSWORD = "senha";
13      private static String OWNER_PASSWORD = "coelho";
14
29          file.close();
30      } catch (Exception e) {
31          e.printStackTrace();
32      }
33  }
```





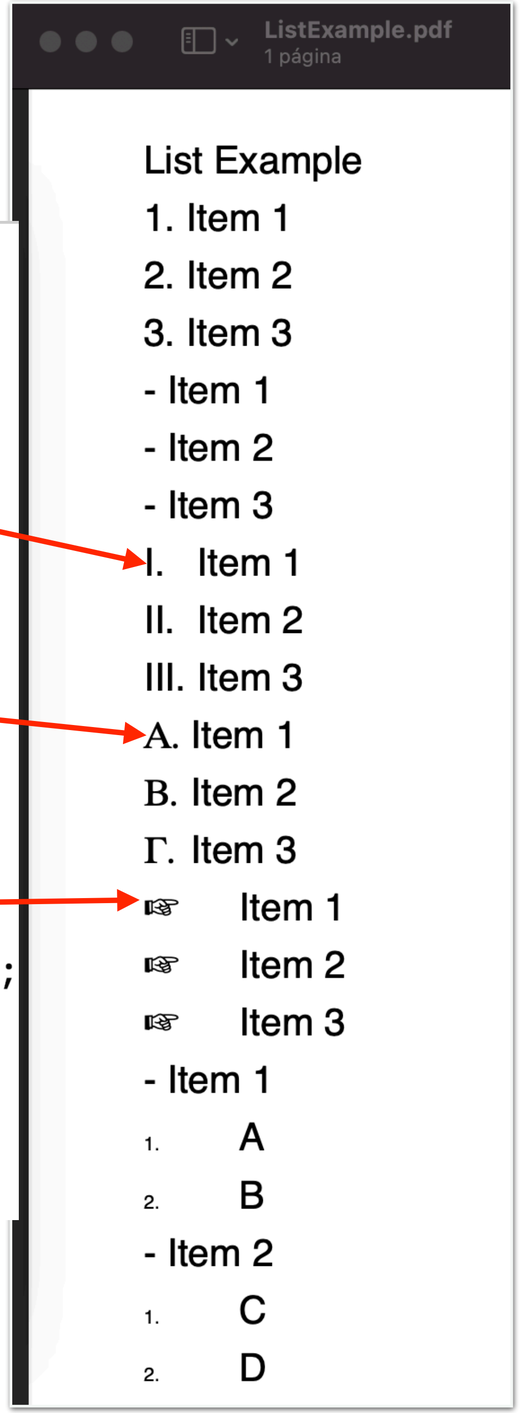
LISTA DE ITENS NO PDF

```java
15  public class CreateListExample {
16      public static void main(String[] args) {
17          Document document = new Document();
18          try {
19              PdfWriter writer = PdfWriter.getInstance(document,
20                      new FileOutputStream( name: "ListExample.pdf"));
21              document.open();
22              document.add(new Paragraph( string: "List Example"));
23
24              //Add ordered list
25              List orderedList = new List(List.ORDERED);
26              orderedList.add(new ListItem( string: "Item 1"));
27              orderedList.add(new ListItem( string: "Item 2"));
28              orderedList.add(new ListItem( string: "Item 3"));
29              document.add(orderedList);
30              //Add un-ordered list
31              List unorderedList = new List(List.UNORDERED);
32              unorderedList.add(new ListItem( string: "Item 1"));
33              unorderedList.add(new ListItem( string: "Item 2"));
34              unorderedList.add(new ListItem( string: "Item 3"));
35              document.add(unorderedList);
```



LISTA DE ITENS NO PDF

```
36        //Add roman list
37        RomanList romanList = new RomanList();
38        romanList.add(new ListItem( string: "Item 1"));
39        romanList.add(new ListItem( string: "Item 2"));
40        romanList.add(new ListItem( string: "Item 3"));
41        document.add(romanList);
42        //Add greek list
43        GreekList greekList = new GreekList();
44        greekList.add(new ListItem( string: "Item 1"));
45        greekList.add(new ListItem( string: "Item 2"));
46        greekList.add(new ListItem( string: "Item 3"));
47        document.add(greekList);
48        //ZapfDingbatsList List Example
49        ZapfDingbatsList zapfDingbatsList = new ZapfDingbatsList( zn: 43,  symbolIndent: 30);
50        zapfDingbatsList.add(new ListItem( string: "Item 1"));
51        zapfDingbatsList.add(new ListItem( string: "Item 2"));
52        zapfDingbatsList.add(new ListItem( string: "Item 3"));
53        document.add(zapfDingbatsList);
```



```java
54        //List and Sublist Examples
55        List nestedList = new List(List.UNORDERED);
56        nestedList.add(new ListItem( string: "Item 1"));
57        List sublist = new List( numbered: true,  lettered: false,  symbolIndent: 30);
58        sublist.setListSymbol(new Chunk( content: "",
59                FontFactory.getFont(FontFactory.HELVETICA,  size: 6)));
60        sublist.add("A");
61        sublist.add("B");
62        nestedList.add(sublist);
63        nestedList.add(new ListItem( string: "Item 2"));
64        sublist = new List( numbered: true,  lettered: false,  symbolIndent: 30);
65        sublist.setListSymbol(new Chunk( content: "",
66                FontFactory.getFont(FontFactory.HELVETICA,  size: 6)));
67        sublist.add("C");
68        sublist.add("D");
69        nestedList.add(sublist);
70        document.add(nestedList);
71
72        document.close();
73        writer.close();
74    } catch (Exception e) {
75        e.printStackTrace();
76    }
```

ListExample.pdf
1 página

List Example
1. Item 1
2. Item 2
3. Item 3
- Item 1
- Item 2
- Item 3
I. Item 1
II. Item 2
III. Item 3
A. Item 1
B. Item 2
Γ. Item 3
☞ Item 1
☞ Item 2
☞ Item 3
- Item 1
1. A
2. B
- Item 2
1. C
2. D

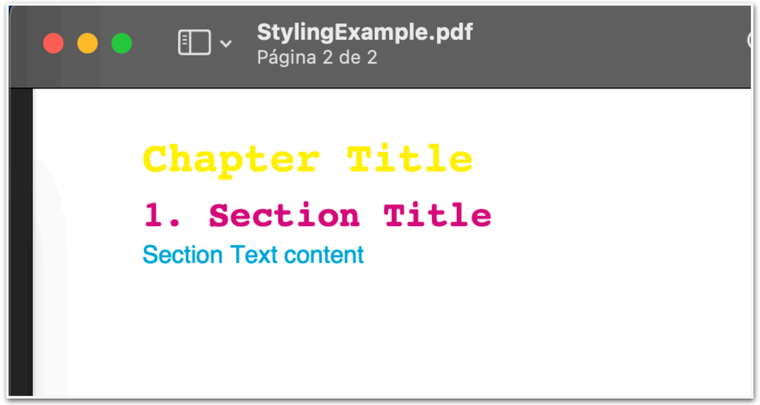
LISTA DE ITENS NO PDF

USANDO FONTES

```
13  public class UsandoFontes {
14      public static void main(String[] args) {
15          Font blueFont = FontFactory.getFont(FontFactory.HELVETICA, size: 8, Font.NORMAL,
16                  new CMYKColor( intCyan: 255, intMagenta: 0, intYellow: 0, intBlack: 0));
17          Font redFont = FontFactory.getFont(FontFactory.COURIER, size: 12, Font.BOLD,
18                  new CMYKColor( intCyan: 0, intMagenta: 255, intYellow: 0, intBlack: 0));
19          Font yellowFont = FontFactory.getFont(FontFactory.COURIER, size: 14, Font.BOLD,
20                  new CMYKColor( intCyan: 0, intMagenta: 0, intYellow: 255, intBlack: 0));
21          Document document = new Document();
22          try {
23              PdfWriter writer = PdfWriter.getInstance(document, new FileOutputStream(
24                      name: "StylingExample.pdf"));
25              document.open();
26              document.add(new Paragraph( string: "Styling Example"));
27
28              //Paragraph with color and font styles
29              Paragraph paragraphOne = new Paragraph( string: "Some colored paragraph text",
30                      redFont);
31              document.add(paragraphOne);
```

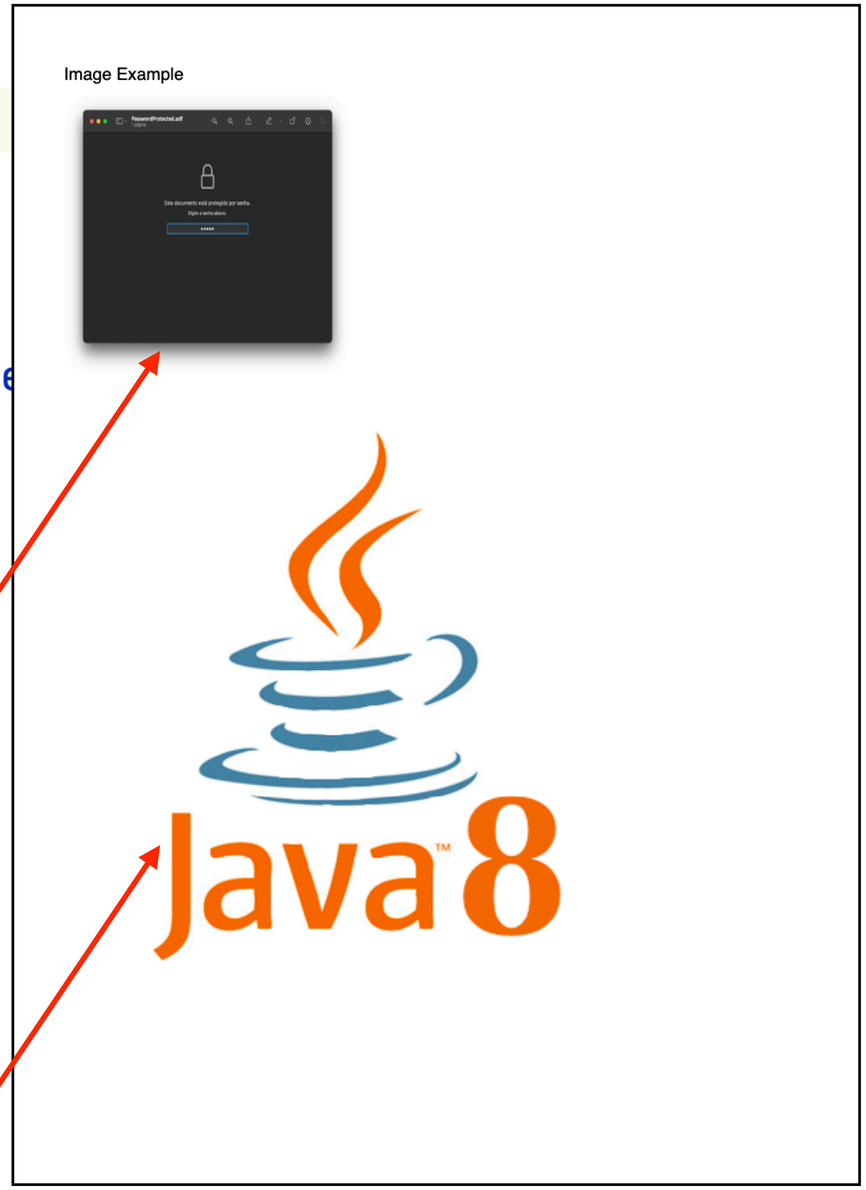
USANDO FONTES

```
33          //Create chapter and sections
34          Paragraph chapterTitle = new Paragraph( string: "Chapter Title", yellowFont);
35          Chapter chapter1 = new Chapter(chapterTitle,  number: 1);
36          chapter1.setNumberDepth(0);
37
38          Paragraph sectionTitle = new Paragraph( string: "Section Title", redFont);
39          Section section1 = chapter1.addSection(sectionTitle);
40
41          Paragraph sectionContent = new Paragraph( string: "Section Text content", blueFont);
42          section1.add(sectionContent);
43
44          document.add(chapter1);
45
46          document.close();
47          writer.close();
48      } catch (Exception e) {
49          e.printStackTrace();
50      }
51  }
```



ADICIONANDO IMAGENS NO PDF

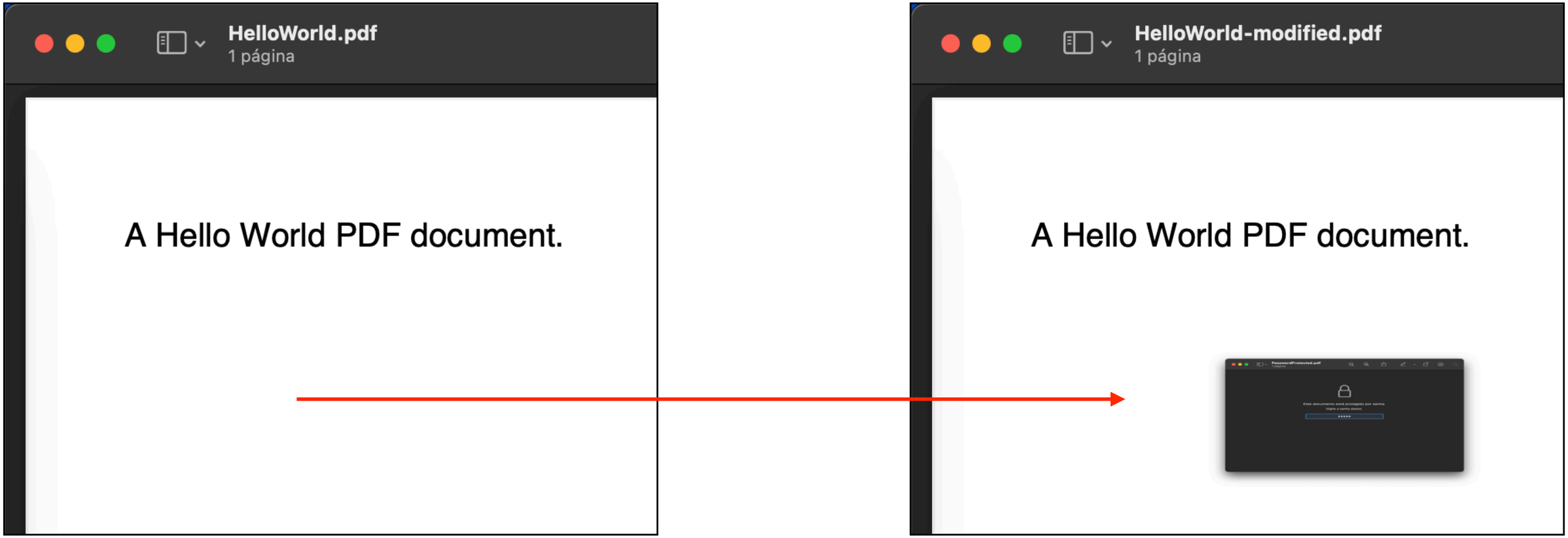
```java
10  public class AdicionandoImagem {
11      public static void main(String[] args) {
12          Document document = new Document();
13          try {
14              PdfWriter writer = PdfWriter.getInstance(document, ne
15                      name: "AddImageExample.pdf"));
16              document.open();
17              document.add(new Paragraph( string: "Image Example"));
18
19              //Add Image
20              Image image1 = Image.getInstance("imagem.png");
21              //For Fixed Positioning
22              //image1.setAbsolutePosition(100f, 550f);
23              //Scale to new height and new width of image
24              image1.scaleAbsolute( newWidth: 200,  newHeight: 200);
25              //Add to document
26              document.add(image1);
27              //Image from the Internet
28              String imageUrl = "http://www.eclipse.org/xtend/images/java8_logo.png";
29              Image image2 = Image.getInstance(new URL(imageUrl));
30              document.add(image2);
```



```java
11  public class LendoPDF {
12      public static void main(String[] args) {
13          try {
14              //Read file using PdfReader
15              PdfReader pdfReader = new PdfReader( filename: "HelloWorld.pdf");
16
17              //Modify file using PdfReader
18              PdfStamper pdfStamper = new PdfStamper(pdfReader, new FileOutputStream(
19                      name: "HelloWorld-modified.pdf"));
20
21              Image image = Image.getInstance("imagem.png");
22              image.scaleAbsolute( newWidth: 100,  newHeight: 50);
23              image.setAbsolutePosition( absoluteX: 100f,  absoluteY: 700f);
24
25              for (int i = 1; i <= pdfReader.getNumberOfPages(); i++) {
26                  PdfContentByte content = pdfStamper.getUnderContent(i);
27                  content.addImage(image);
28              }
29              pdfStamper.close();
30          } catch (IOException e) {
31              e.printStackTrace();
32          } catch (DocumentException e) {
33              e.printStackTrace();
34          }
```

LENDO E MODIFICANDO O PDF

ADICIONAMOS UMA IMAGEM AO PDF ORIGINAL



```java
14  public class LendoPDF {
15      public static void main(String[] args) {
16          try {
17              //Read file using PdfReader
18              PdfReader pdfReader = new PdfReader( filename: "HelloWorld.pdf");
19
20              //Modify file using PdfReader
21              PdfStamper pdfStamper = new PdfStamper(pdfReader, new FileOutputStream(
22                      name: "HelloWorld-modified.pdf"));
23
24              Image image = Image.getInstance("imagem.png");
25              image.scaleAbsolute( newWidth: 100,  newHeight: 50);
26              image.setAbsolutePosition( absoluteX: 100f,  absoluteY: 700f);
27
28              PdfReaderContentParser parser = new PdfReaderContentParser(pdfReader);
29              TextExtractionStrategy strategy;
30
31              for (int i = 1; i <= pdfReader.getNumberOfPages(); i++) {
32                  strategy = parser.processContent(i, new SimpleTextExtractionStrategy());
33                  System.out.println(strategy.getResultantText());
34
35                  PdfContentByte content = pdfStamper.getUnderContent(i);
36                  content.addImage(image);
37              }
```

```
Run: LendoPDF
/Library/Java/JavaVirtualMachines/jdk1.8
A Hello World PDF document.

Process finished with exit code 0
```

EXERCÍCIOS

- Implemente um cadastro completo de uma série de itens (você deve definir qual classe).
- Ao final da execução do programa os dados cadastrados devem ser salvos em um arquivo PDF.
- O nome do arquivo PDF deve conter a data e hora de quando foram salvos (dica: pesquise sobre as classes TimeStamp e SimpleDateFormat).
- O seu arquivo PDF deve conter uma imagem no canto superior direito que tenha relação com o seu cadastro.
- O seu programa principal deve permitir que o usuário solicite a leitura de um dos pdfs criados previamente (o usuário pode informar o nome do arquivo pdf). Use como base o exemplo modificado a seguir.