

```

a) public class Artigoc {
    private String tit;
    private ArrayList<String> autores;
    private LocalDate data;
    private int pag;

b) public Artigoc() {
    tit = "";
    autores = new ArrayList<>();
    data = LocalDate.now();
    pag = 0;
}

c) public String getTit() {
    return tit;
}
    public void setTit(String tit) {
        this.tit = tit;
    }
    public ArrayList<String> getAutores() {
        return autores;
    }
    public void setAutores(ArrayList<String> autores) {
        this.autores = (ArrayList<String>)autores.clone();
    }
    public LocalDate getData() {
        return data;
    }
    public void setData(LocalDate data) {
        this.data = data;
    }
    public int getPag() {
        return pag;
    }

    public void setPag(int pag) {
        this.pag = pag;
    }

d) @Override
    public String toString() {
        return "Artigoc [tit=" + tit + ", autores=" + autores + ", data=" + data + ",
pag=" + pag + "]\n";
    }

e) @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Artigoc other = (Artigoc) obj;
        return Objects.equals(autores, other.autores) && Objects.equals(data, other.data)
&& pag == other.pag && Objects.equals(tit, other.tit);
    }

```

f)

```
@Override
public Object clone() {
    ArtigoC copia = new ArtigoC();
    copia.tit = this.tit;
    copia.autores = (ArrayList<String>)this.autores.clone();
    copia.data = this.data;
    copia.pag = this.pag;
    return copia;
}
```

g)

```
public boolean verificaAutor (String s) {
    return autores.contains(s);
}
```

h)

```
public void addAutor (String a) throws AutorDuplicadoException {
    if (!autores.contains(a))
        autores.add(a);
    else
        throw new AutorDuplicadoException( "O autor " + a + "já existe");
}
```

8

```
public class AutorDuplicadoException extends Exception {
    public AutorDuplicadoException() {
        super();
    }
    public AutorDuplicadoException(String s) {
        super(s);
    }
}
```

9

```
public static ArrayList<ArtigoC> postDataArts ( ArrayList<ArtigoC>lista , LocalDate d) {
    ArrayList<ArtigoC> listaAux = new ArrayList<ArtigoC>();
    for (int i = 0; i < lista.size(); i++) {
        if ( lista.get(i).getData().compareTo(d) >= 0 )
            listaAux.add(lista.get(i));
    }
    return listaAux;
}
```

10

```
public static void main (String args[] ) {
    //a)
    ArtigoC c1 = new ArtigoC ();
    ArrayList<String> laut = new ArrayList<>();
    laut.add("P.P");
    laut.add("P.F.");
    c1.setTit("P00");
    c1.setAutores(laut);
    LocalDate d = LocalDate.of(2022, 2, 1);
    c1.setData(d);
    c1.setPag(10);

    //b)

    ArtigoC c1c = (ArtigoC) c1.clone();
    System.out.println (c1.equals(c1c));

    //c)
    c1c.setTit("P002");
    c1c.setData(LocalDate.of(2024,5,10));
}
```

```

//d)
    try {
        c1c.addAutor("MEF");
    } catch (AutorDuplicadoException e) {
        System.out.println( e.getMessage());
    }

//e)
    ArrayList<ArtigoC> la = new ArrayList<>();
    la.add(c1);
    la.add(c1c);
    System.out.println( postDataArts(la , LocalDate.of(2022,12,1) ) );
}

```

```

11
a)
public class ArtigoPublicado extends ArtigoC{
    private String doi;

    public ArtigoPublicado(ArtigoC a) {
        super();
        super.setTit(a.getTit());
        super.setAutores(a.getAutores());
        super.setData(a.getData());
        super.setPag(a.getPag());
        doi = "";
    }

b)
    @Override
    public String toString() {
        return "ArtigoPublicado [doi=" + doi + ", getDoi()=" + getDoi() + "]" +
            super.toString();
    }

c)
    @Override
    public boolean equals(Object obj) {
        if (obj!= null && obj.getClass() == this.getClass()) {
            ArtigoPublicado ap = (ArtigoPublicado) obj;
            return super.equals(obj) && this.doi.equals(ap.doi);
        }
        return false;
    }

12
    public static int artPub (ArrayList<ArtigoC> lista) {
        int c=0;
        for (ArtigoC a : lista )
            if (a instanceof ArtigoPublicado)
                c++;
        return c;
    }

13
    public static ArrayList<String> artDoi (ArrayList<ArtigoC> lista, String doi) {
        for (ArtigoC a : lista )
            if (a instanceof ArtigoPublicado)
                if ( ( (ArtigoPublicado)a).getDoi().equals (doi))
                    return a.getAutores();
        return null;
    }

```
