



xideral



Juan Jesús Ávila Mota

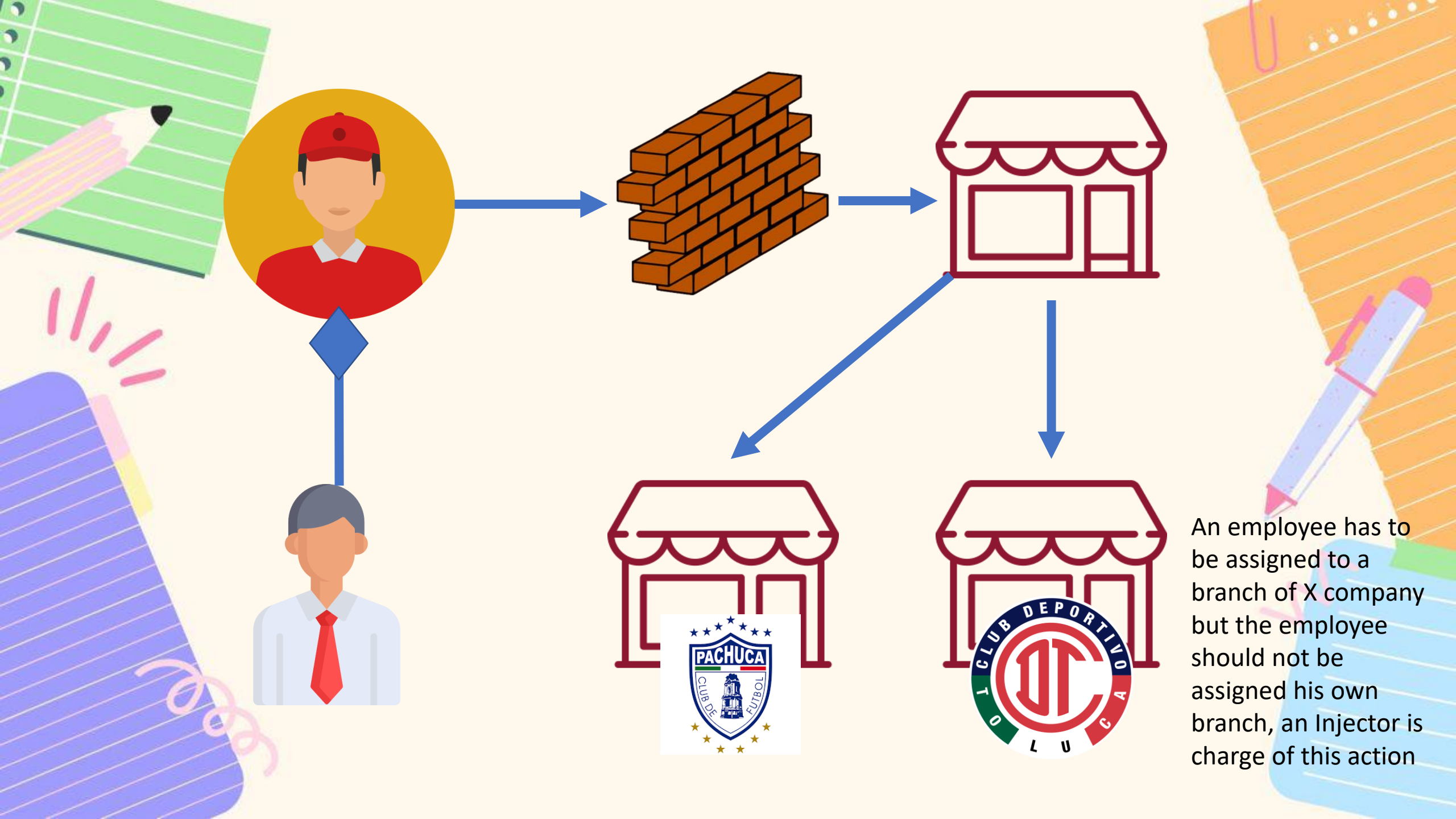
Examen Semana02
Academia Java Xideral

Noviembre 2022

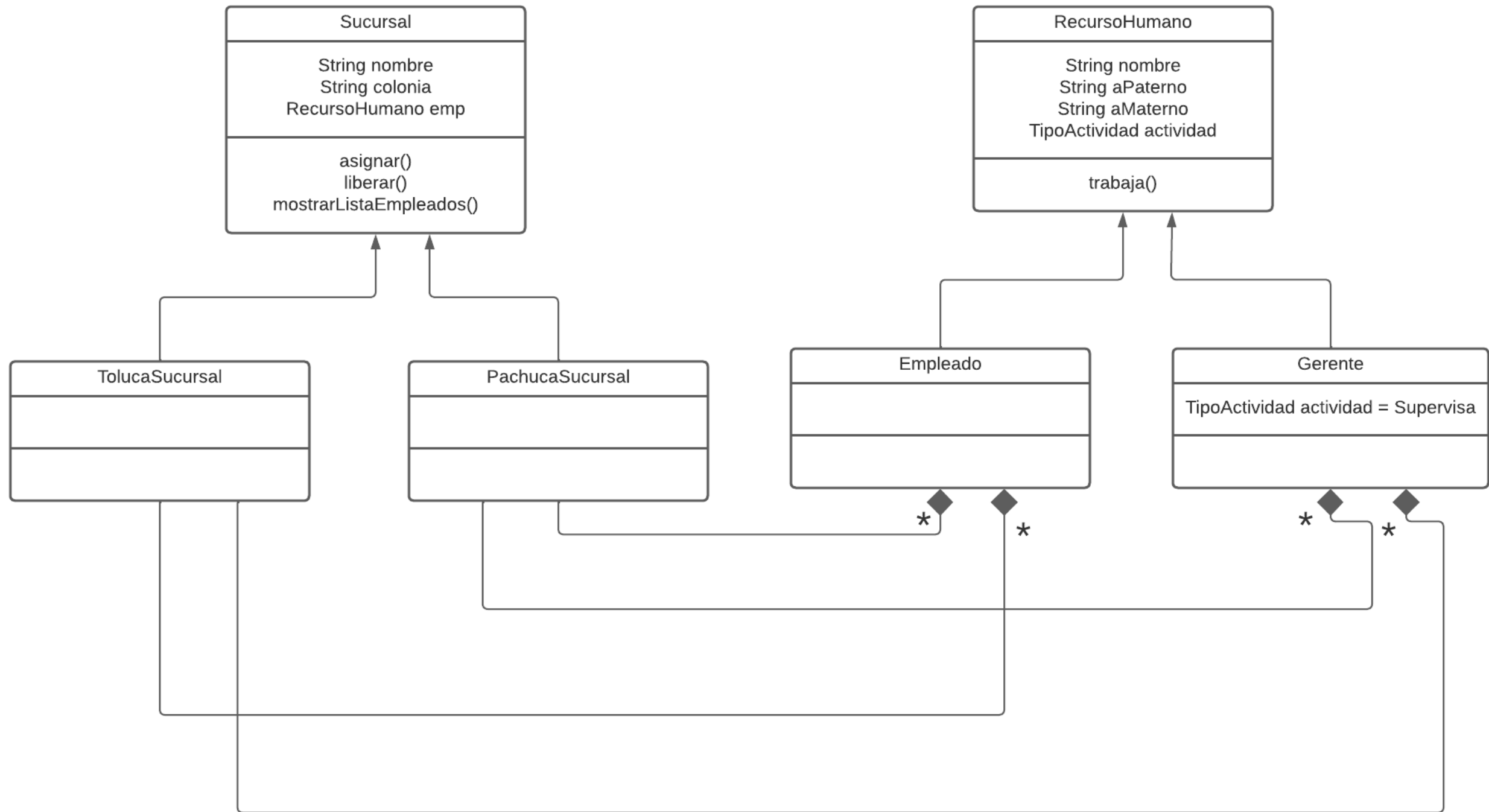
xideral

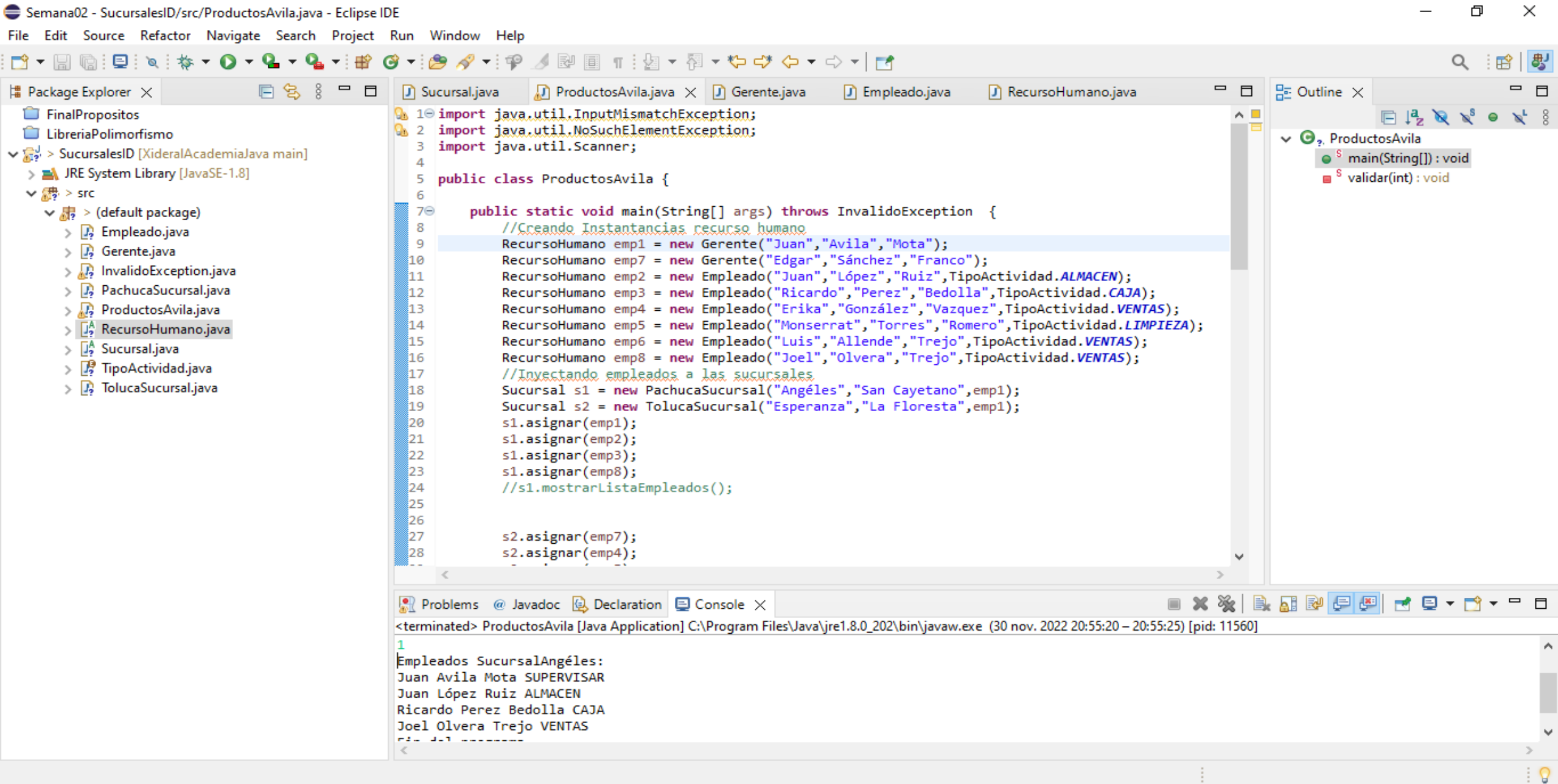


Dependency Inversion (DI)



An employee has to be assigned to a branch of X company but the employee should not be assigned his own branch, an Injector is charge of this action





<https://github.com/mota1299/XideralAcademiaJava/tree/main/Ex%C3%A1menes/Semana02/SucursalesID/src>



xideral



MVC



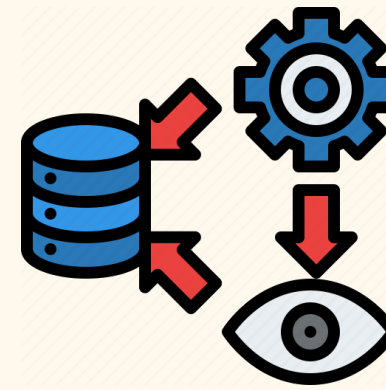


Request
Response

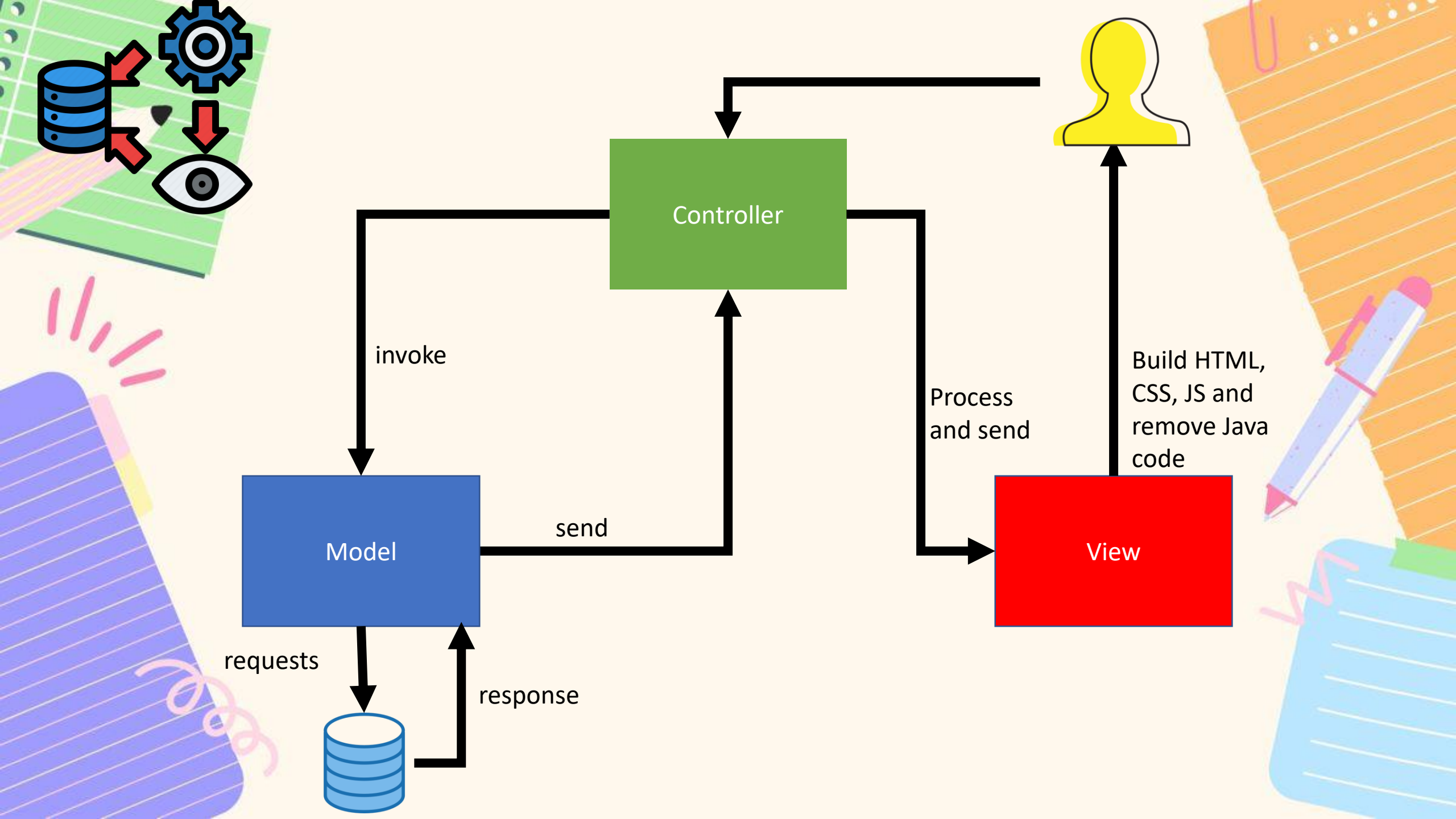


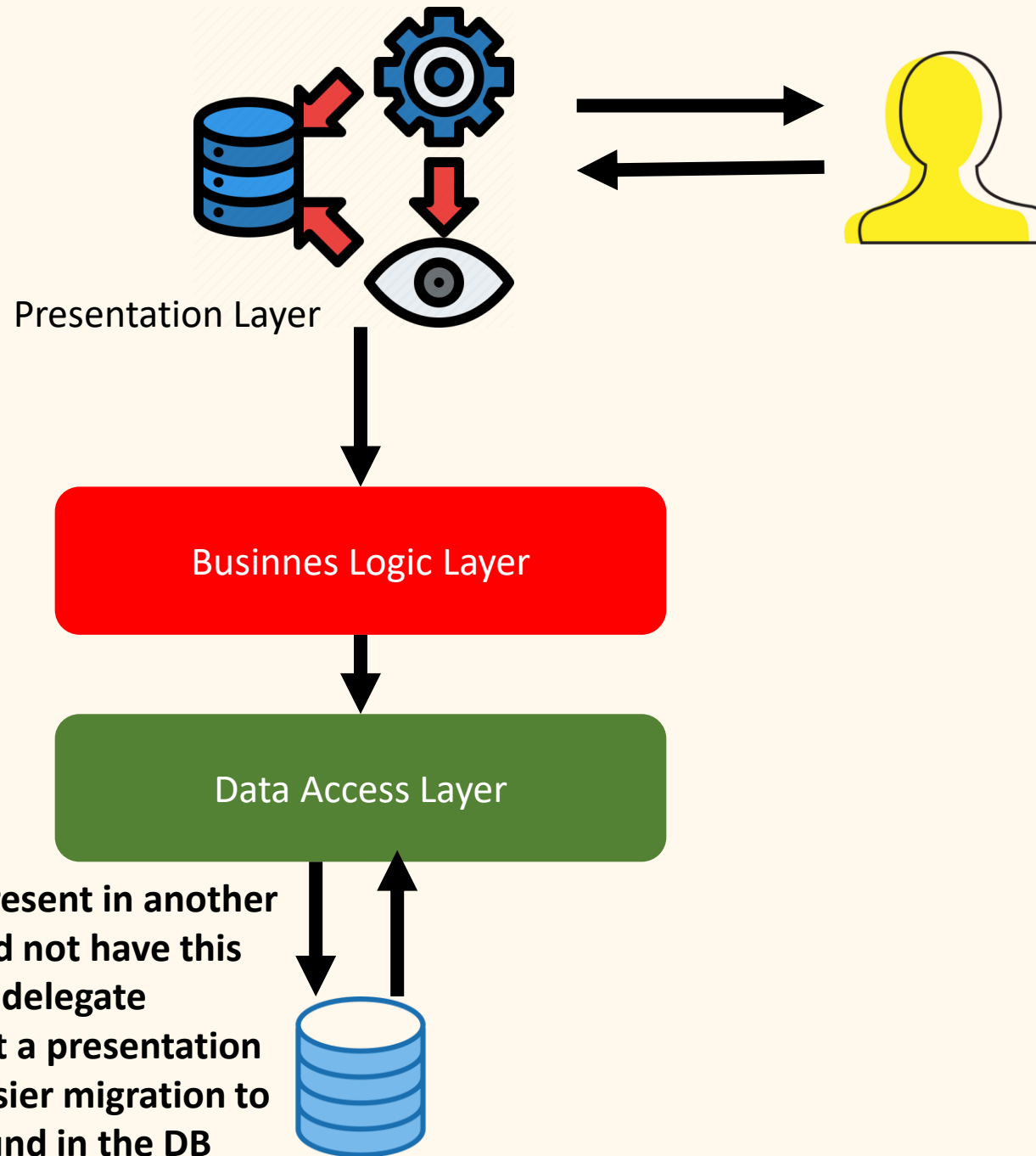
HTTP
• get
• put
• post
• delete
• Etc.

Response: • Pdf
• Json • Excel
• Xml • Html
• Txt • etc



The architecture of this model requires a request by the user through the browser to HTTP, using one of its protocols, this will be sent to the server that contains Java which communicates directly with the MVC and the database.



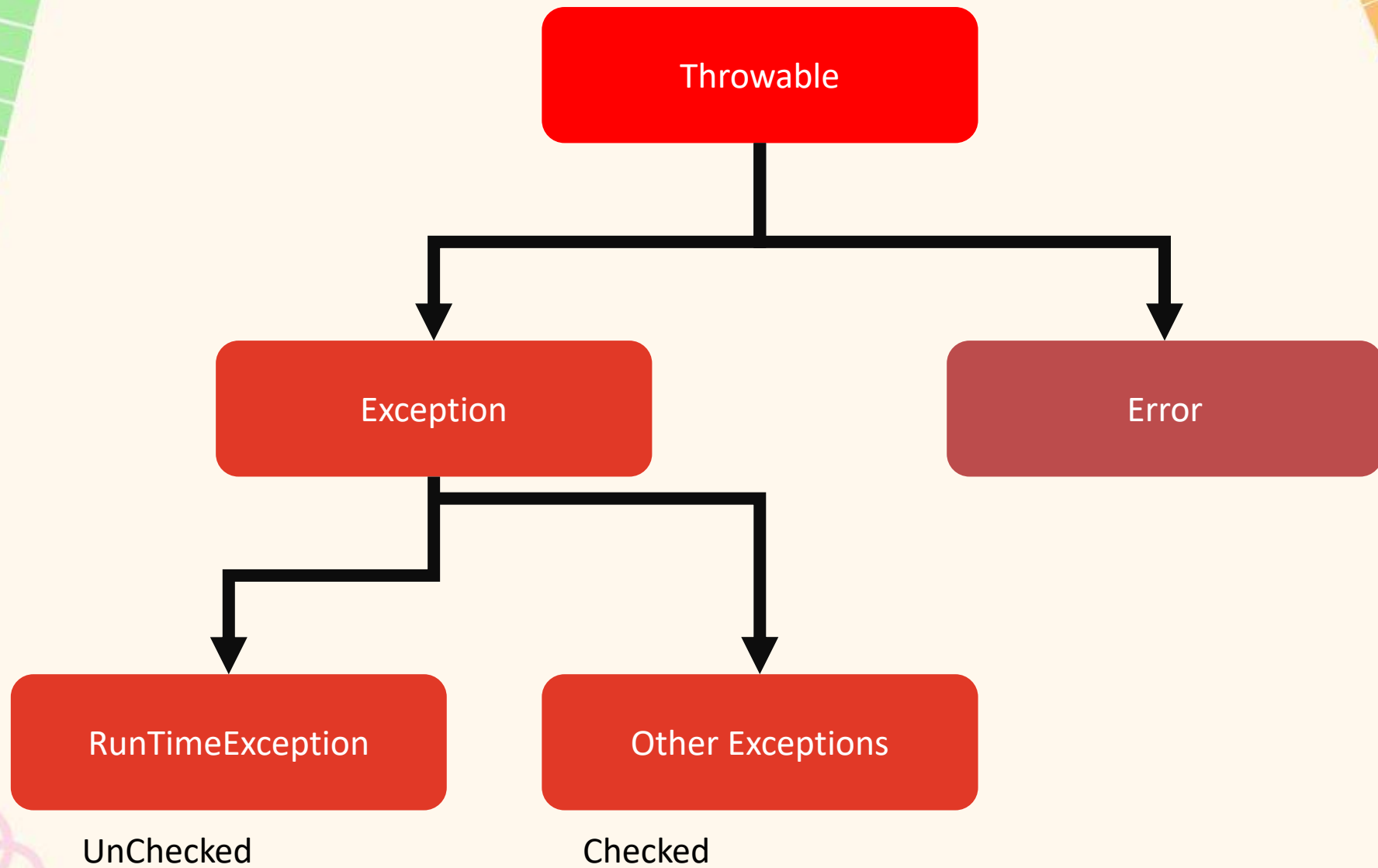


In this model, the business logic must be present in another layer, due to the fact that the MVC should not have this logic in its controller, it should only delegate responsibilities, in addition to the fact that a presentation like this allows better maintenance and easier migration to another SMDB since only tables are found in the DB

xideral



Types of Exceptions



xideral

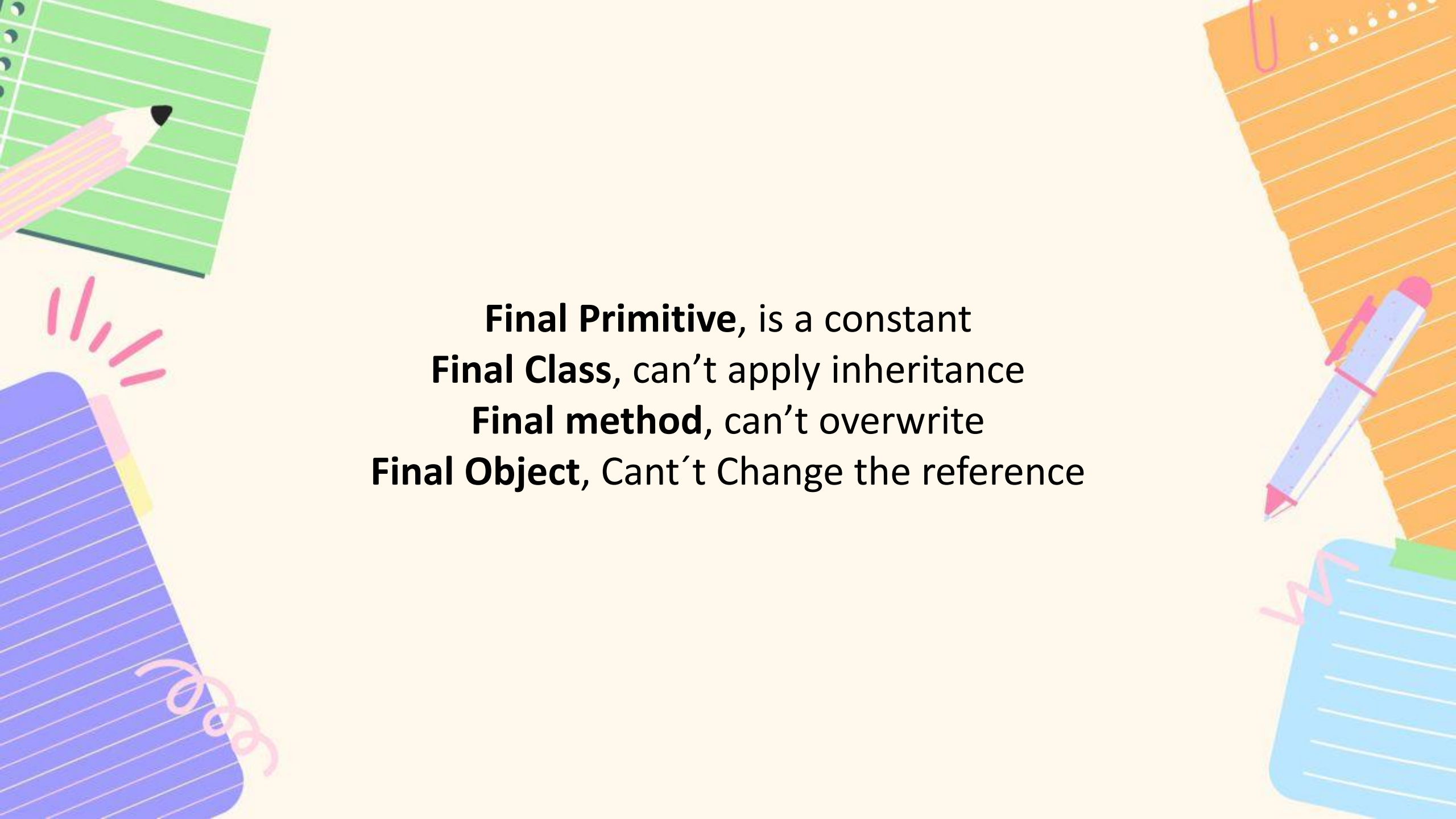


The exceptions have Throwable as a superclass, There are three types of exceptions, The errors are exceptions for which it is not possible to do something, for example, a damage to the hard disk, The RuntimeException, UnChecked, the compiler does not force to do something with them, no it is necessary to catch them, contrary to Checked Exceptions, which compulsorily forces us to treat them either with a catch or throw them to the superior method, having as limit the JVM

xideral



Purposes of final

The background is a light cream color. In the top-left corner, there is a green notepad with a yellow pencil resting on it. In the top-right corner, there is an orange notepad with a pink paperclip and a blue pen. In the bottom-left corner, there is a purple notepad with pink squiggly lines. In the bottom-right corner, there is a blue notepad with pink squiggly lines.

Final Primitive, is a constant
Final Class, can't apply inheritance
Final method, can't overwrite
Final Object, Can't Change the reference

Semana02 - FinalPropositos/src/Principal.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- FinalPropositos [XideralAcademiaJava main]
 - JRE System Library [JavaSE-1.8]
 - src
 - (default package)
 - Antirrobo.java
 - Cangurera.java
 - Mochila.java
 - Principal.java
 - Producto.java
- LibreriaPolimorfismo
 - SucursalesID [XideralAcademiaJava main]
 - JRE System Library [JavaSE-1.8]
 - src
 - (default package)
 - Empleado.java
 - Gerente.java
 - InvalidoException.java
 - PachucaSucursal.java
 - ProductosAvila.java
 - RecursoHumano.java
 - Sucursal.java
 - TipoActividad.java
 - TolucaSucursal.java

Principal.java

```
1 |
2 public class Principal {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Mochila m1 = new Cangurera(150.0,0.30);
7         final Mochila m2 = new Antirrobo(350.0,0.30);
8
9         double m1Precio = m1.getPrecioVenta();
10        System.out.println("Precio de m1 " +m1Precio);
11
12        double m2Precio = m2.getPrecioVenta();
13        System.out.println("Precio de m2 " +m2Precio);
14        //m2 = new Cangurera(130,0.10);// no se puede cambiar de referencia
15
16
17        m1 = new Antirrobo(500,20);
18        m1Precio = m1.getPrecioVenta();
19        System.out.println("Precio de m1 despues de cambiar de referencia: " +m1Precio);
20
21        System.out.println("////////////////////////");
22        final StringBuilder sb = new StringBuilder("Hola");
23        sb.append(" Mundo");
24        //sb = sb.append(" Mundo");
25        System.out.println(sb);
26    }
27
28 }
```

Outline

- Principal
 - main(String[]) : void

Problems Javadoc Declaration Console

<terminated> Principal [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe (30 nov. 2022 20:57:58 – 20:57:58) [pid: 6744]

```
Precio de m1 226.2
Precio de m2 527.8
Precio de m1 despues de cambiar de referencia: 12180.0
////////////////////////
Hola Mundo
```

Writable Smart Insert 1:1:0

<https://github.com/mota1299/XideralAcademiaJava/tree/main/Ex%C3%A1menes/Semana02/FinalPropositos/src>

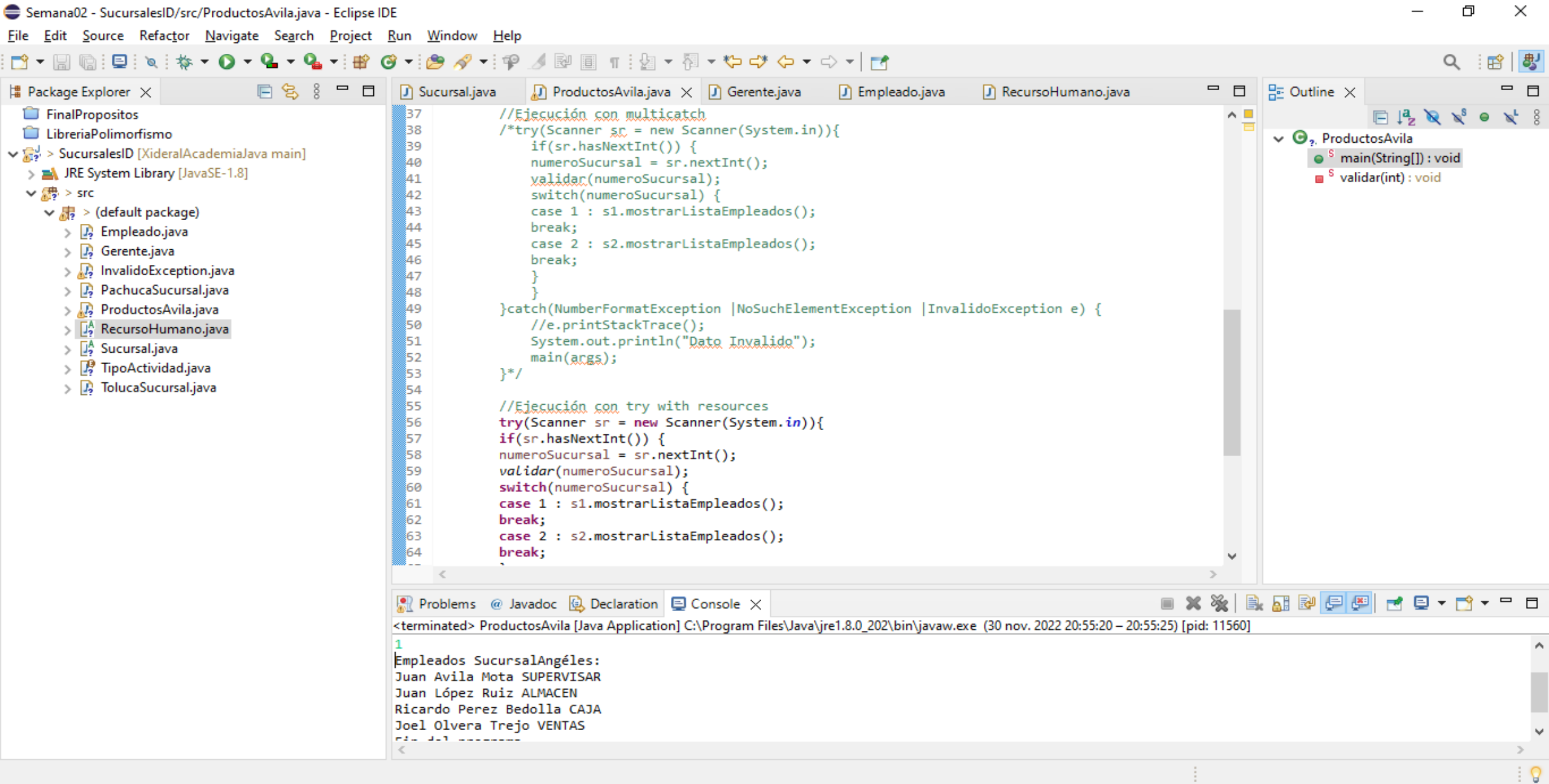


xideral



Multicatch and Try with resources





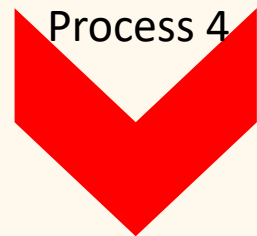
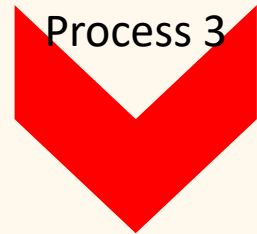
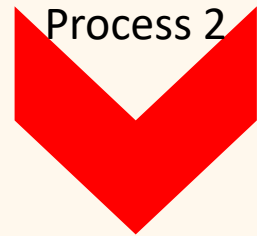
<https://github.com/mota1299/XideralAcademiaJava/tree/main/Ex%C3%A1menes/Semana02/SucursalesID/src>

xideral



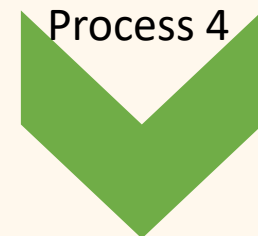
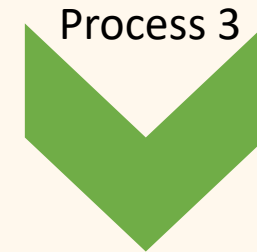
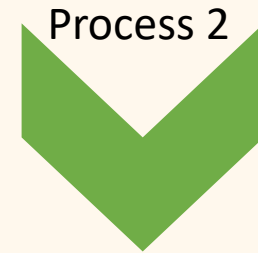
Asynchronous vs Synchronous

SYNCHRONOUS



Synchronous
The execution is sequential, it returns when the operation has been completed in its entirety, the processes go one by one

ASYNCHRONOUS

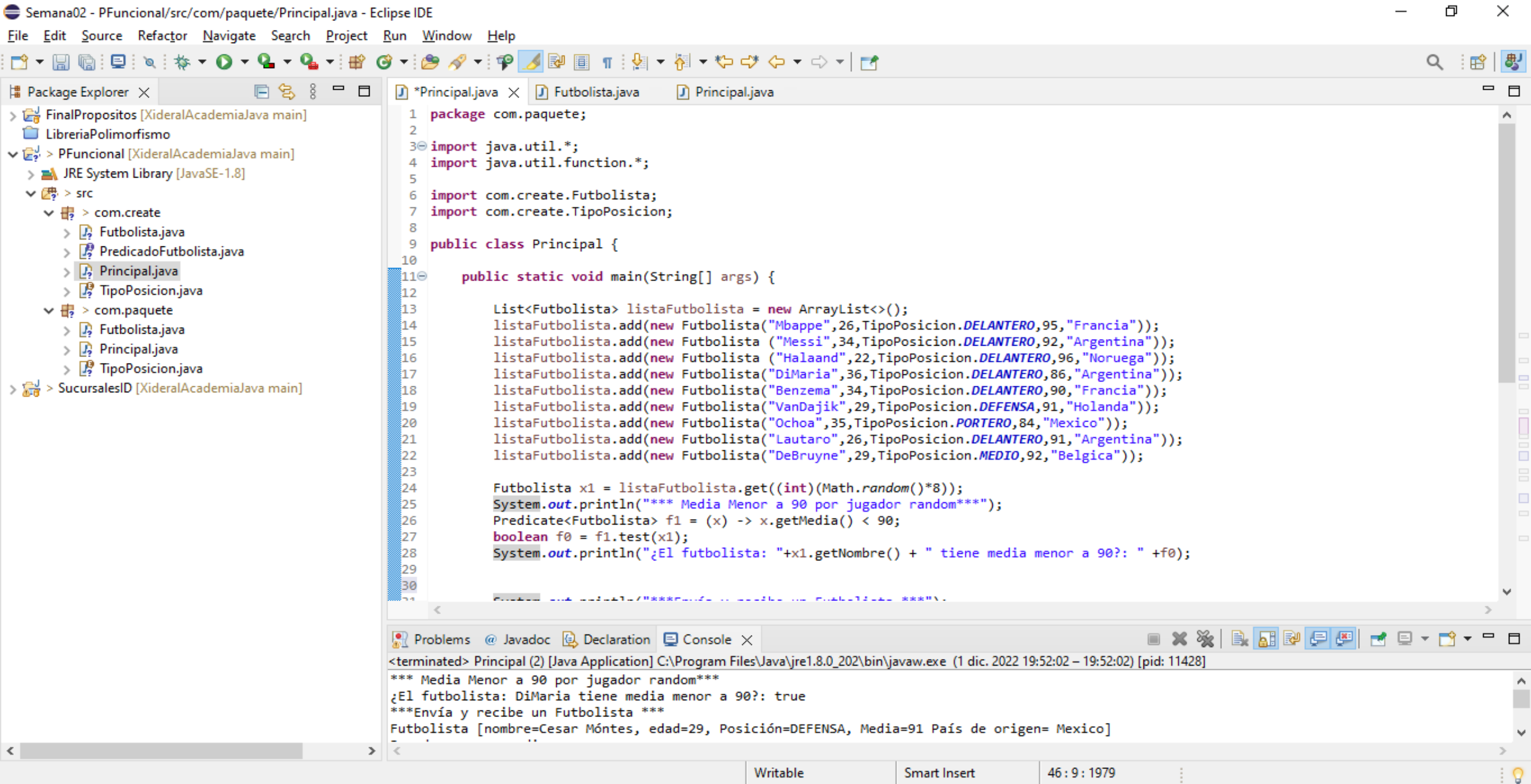


Processes do not depend on other processes to start or finish, they can be working simultaneously and interact with each other.

xideral



Functional Lambdas



<https://github.com/mota1299/XideralAcademiaJava/tree/main/Ex%C3%A1menes/Semana02/PFuncional/src/com>