Team 6: GitSoft Team
Project Name: Italian Restaurant “Di Tutto`s”
Github: https://github.com/YormanOna/T06-GitSoftTeam.git
Leader: Yorman Oña
MongoDB URL: mongodb+srv://poo:poo1@cluster0.jfzncfw.mongodb.net/test
Inspector: Team 5 Erick Lasluisa

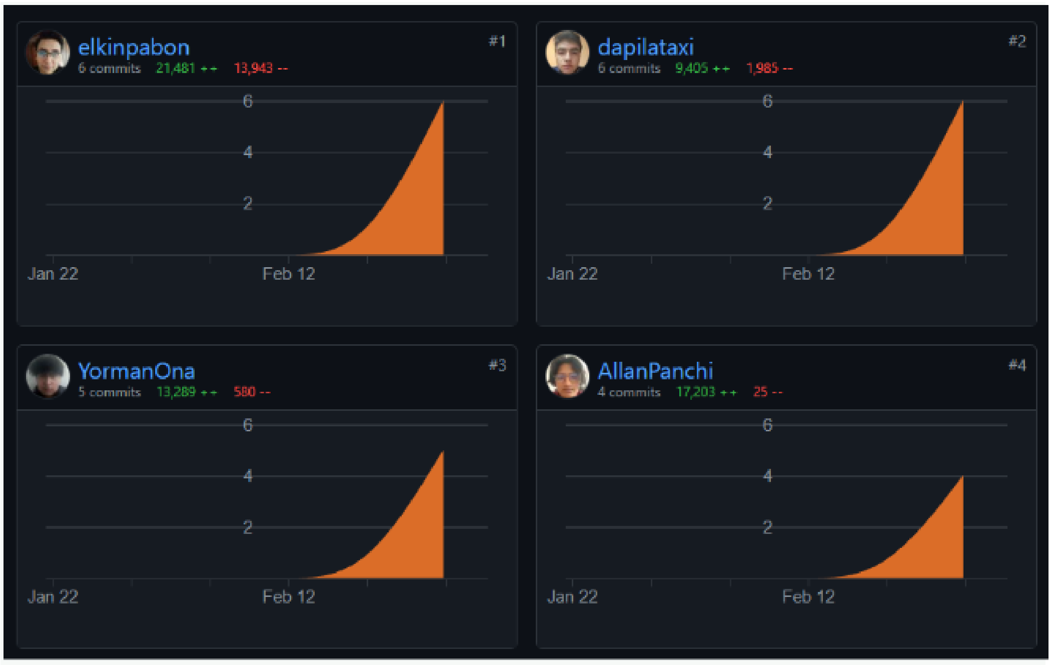
22	OÑA GAMARRA YORMAN JAVIER	10
23	PABON GONZALEZ ELKIN ANDRES	13
24	PANCHI PILLAJO ALLAN VINICIO	10
25	PILATAXI MIRANDA DIEGO ALEJANDRO	13

RUBRIC

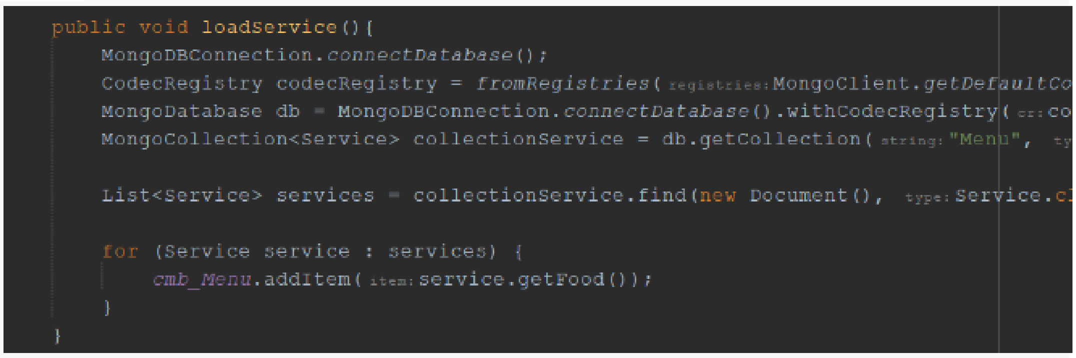
GitHub	/20	
SOLid	19/20	-> single responsibility -1
		DB conection in PanelRegisterOrder (-1)
GoF Patterns	10/20	-> every design pattern /10
		There’s just one design pattern (-10)
Functionality	10/20	every collection/1, every functionality/1
Unit Tests	10/20	failing to pass/5, pass to fail/5, 100 test cases/10
		100 test cases in an excel document (5/10)
		failing to pass 5/5
		pass to fail 0/5
Code Quality	-3/-20	every pitfall -1
		Uncommunicative names -1
		Types Embedded in Names -1
		Dead Code -1

____

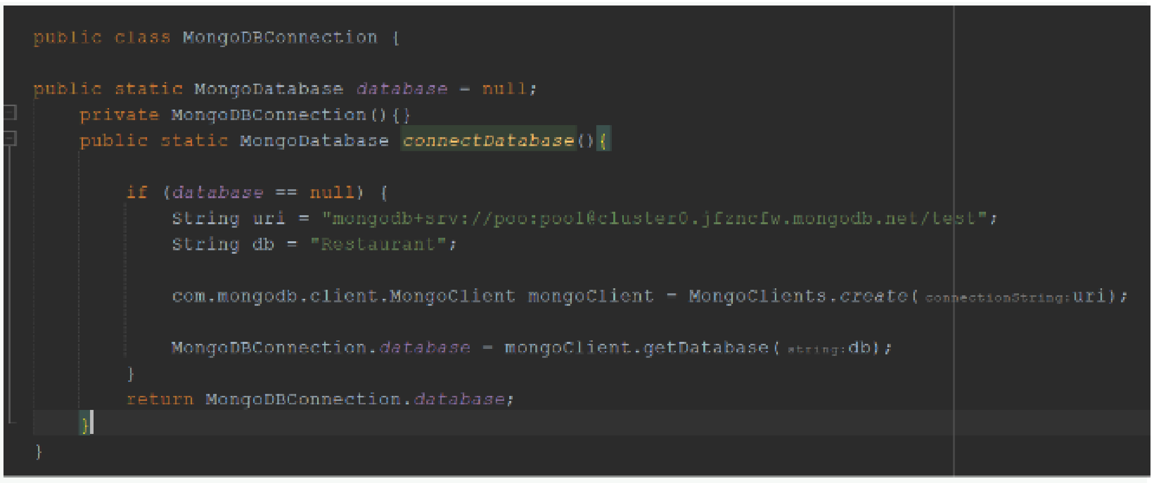
TOTAL 49/100


elkinpabon
6 commits 21,481 ++ 13,943 --
#1
dapilataxi
6 commits 9,405 ++ 1,985 --
#2
YormanOna
5 commits 13,289 ++ 580 --
#3
AllanPanchi
4 commits 17,203 ++ 25 --
#4
6
4
2
Jan 22
Feb 12

SOLid	Single Responsibility Code (classes and methods not related and methods that do too much) View does not have algorithms

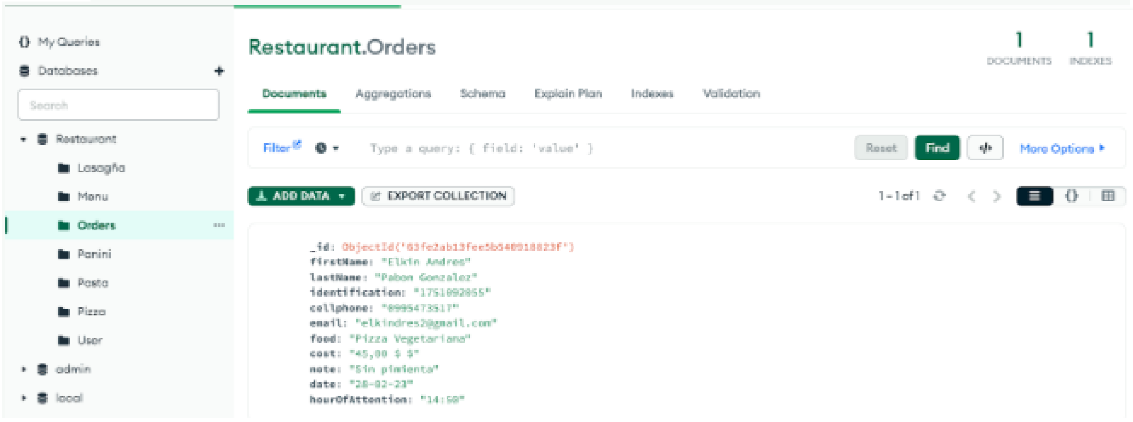


```java
public void loadService(){
    MongoDBConnection.connectDatabase();
    CodecRegistry codecRegistry = fromRegistries( registries:MongoClient.getDefaultCo
    MongoDatabase db = MongoDBConnection.connectDatabase().withCodecRegistry( cr:co
    MongoCollection<Service> collectionService = db.getCollection( string:"Menu",  ty

    List<Service> services = collectionService.find(new Document(),  type:Service.cl

    for (Service service : services) {
        cmb_Menu.addItem( item:service.getFood());
    }
}
```

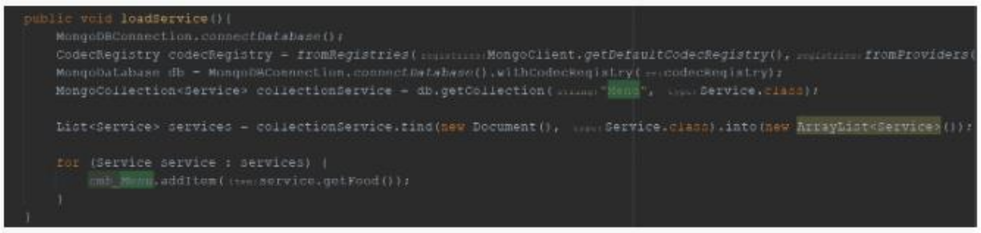


```java
public class MongoDBConnection {

public static MongoDatabase database = null;
    private MongoDBConnection(){}
    public static MongoDatabase connectDatabase(){

        if (database == null) {
            String uri = "mongodb+srv://poo:poo1@cluster0.jfzncfw.mongodb.net/test";
            String db = "Restaurant";

            com.mongodb.client.MongoClient mongoClient = MongoClients.create( connectionString:uri);

            MongoDBConnection.database = mongoClient.getDatabase( string:db);
        }
        return MongoDBConnection.database;
    }
}
```

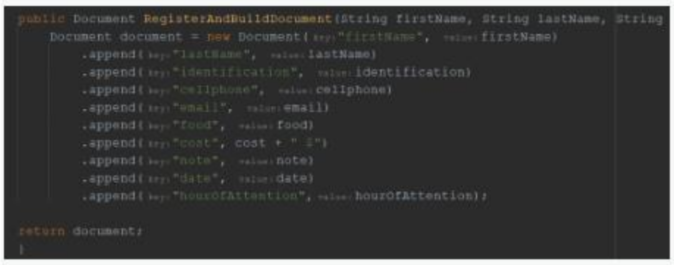
Functionality	at least 10 Collections, and 10 operations Screenshots



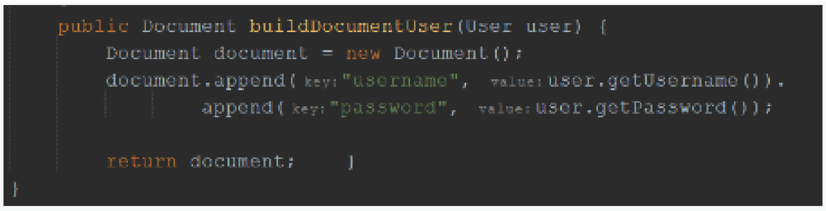
```
_id: ObjectId('63fe2ab13fee5b540918823f')
firstName: "Elkin Andres"
lastName: "Pabon Gonzalez"
identification: "1751092855"
cellphone: "0995473517"
email: "elkindres2@gmail.com"
food: "Pizza Vegetariana"
cost: "45,00 $ $"
note: "Sin pimienta"
date: "28-02-23"
hourOfAttention: "14:50"
```



```java
public void loadService(){
    MongoDBConnection.connectDatabase();
    CodecRegistry codecRegistry = fromRegistries( registries:MongoClient.getDefaultCodecRegistry(),  registries:fromProviders(
    MongoDatabase db = MongoDBConnection.connectDatabase().withCodecRegistry( cr:codecRegistry);
    MongoCollection<Service> collectionService = db.getCollection( string:"Menu",  type:Service.class);

    List<Service> services = collectionService.find(new Document(),  type:Service.class).into(new ArrayList<Service>());

    for (Service service : services) {
        cmb_Menu.addItem( item:service.getFood());
    }
}
```

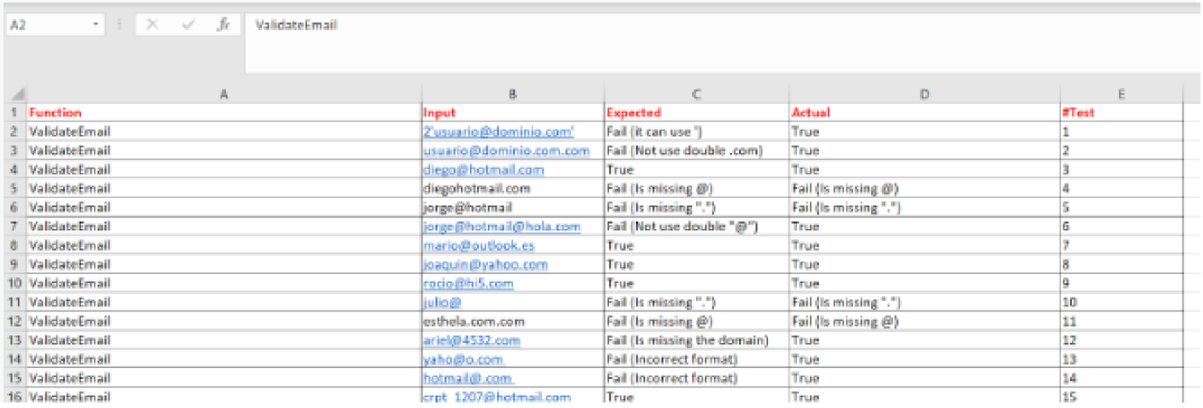


```java
public Document RegisterAndBuildDocument(String firstName, String lastName, String
    Document document = new Document( key:"firstName",  value:firstName)
        .append( key:"lastName",  value:lastName)
        .append( key:"identification",  value:identification)
        .append( key:"cellphone",  value:cellphone)
        .append( key:"email",  value:email)
        .append( key:"food",  value:food)
        .append( key:"cost", cost + " $")
        .append( key:"note",  value:note)
        .append( key:"date",  value:date)
        .append( key:"hourOfAttention",  value:hourOfAttention);

return document;
}
```



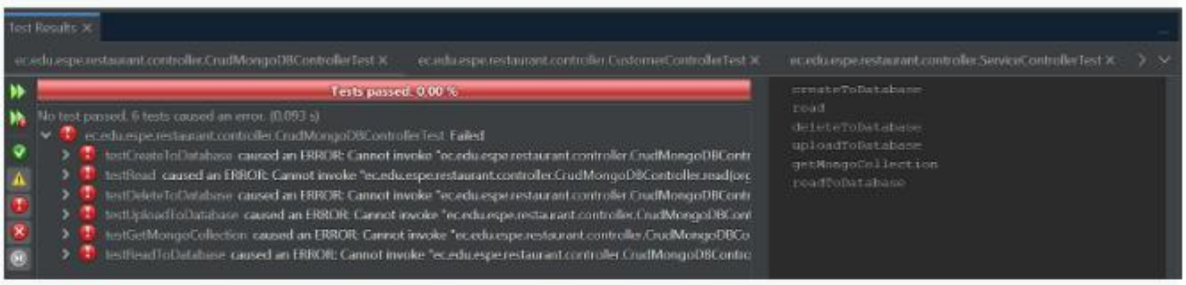
```java
public Document buildDocumentUser(User user) {
    Document document = new Document();
    document.append( key:"username",  value: user.getUsername()).
            append( key:"password",  value: user.getPassword());

    return document;     }
}
```

Unit Tests   100 unit test cases
              two counter cases



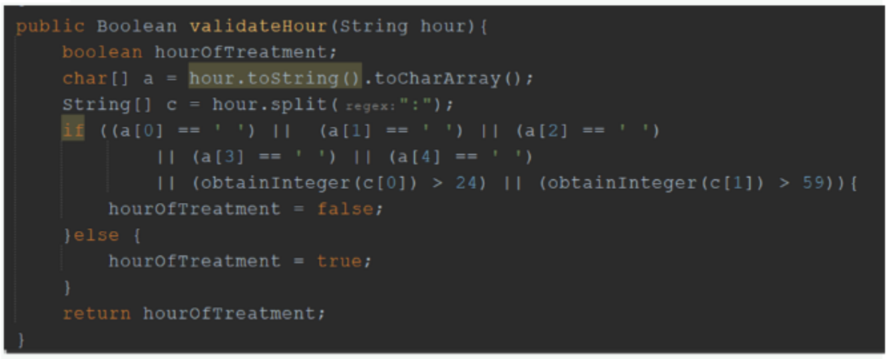
A2   ValidateEmail

	A	B	C	D	E
1	Function	Input	Expected	Actual	#Test
2	ValidateEmail	2'usuario@dominio.com'	Fail (it can use ')	True	1
3	ValidateEmail	usuario@dominio.com.com	Fail (Not use double .com)	True	2
4	ValidateEmail	diego@hotmail.com	True	True	3
5	ValidateEmail	diegohotmail.com	Fail (Is missing @)	Fail (Is missing @)	4
6	ValidateEmail	jorge@hotmail	Fail (Is missing ".")	Fail (Is missing ".")	5
7	ValidateEmail	jorge@hotmail@hola.com	Fail (Not use double "@")	True	6
8	ValidateEmail	mario@outlook.es	True	True	7
9	ValidateEmail	joaquin@yahoo.com	True	True	8
10	ValidateEmail	rocio@hi5.com	True	True	9
11	ValidateEmail	julio@	Fail (Is missing ".")	Fail (Is missing ".")	10
12	ValidateEmail	esthela.com.com	Fail (Is missing @)	Fail (Is missing @)	11
13	ValidateEmail	ariel@4532.com	Fail (Is missing the domain)	True	12
14	ValidateEmail	yaho@o.com	Fail (Incorrect format)	True	13
15	ValidateEmail	hotmail@.com	Fail (Incorrect format)	True	14
16	ValidateEmail	crpt_1207@hotmail.com	True	True	15

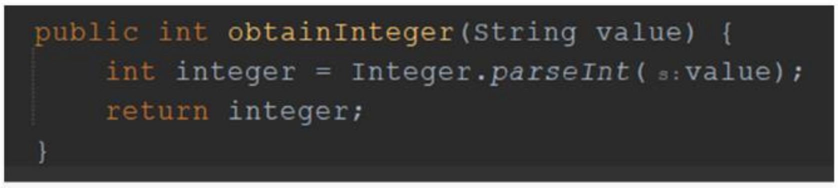


```
Test Results
ec.edu.espe.restaurant.controller.CrudMongoDBControllerTest   ec.edu.espe.restaurant.controller.CustomerControllerTest   ec.edu.espe.restaurant.controller.ServiceControllerTest

Tests passed: 0.00 %
No test passed, 6 tests caused an error. (0.093 s)
  ec.edu.espe.restaurant.controller.CrudMongoDBControllerTest Failed
    testCreateToDatabase caused an ERROR: Cannot invoke "ec.edu.espe.restaurant.controller.CrudMongoDBContr
    testRead caused an ERROR: Cannot invoke "ec.edu.espe.restaurant.controller.CrudMongoDBController.read(org
    testDeleteToDatabase caused an ERROR: Cannot invoke "ec.edu.espe.restaurant.controller.CrudMongoDBContr
    testUploadToDatabase caused an ERROR: Cannot invoke "ec.edu.espe.restaurant.controller.CrudMongoDBCont
    testGetMongoCollection caused an ERROR: Cannot invoke "ec.edu.espe.restaurant.controller.CrudMongoDBCo
    testReadToDatabase caused an ERROR: Cannot invoke "ec.edu.espe.restaurant.controller.CrudMongoDBContro

createToDatabase
read
deleteToDatabase
uploadToDatabase
getMongoCollection
readToDatabase
```

Code Quality   Follow Clean Code book and Pitfalls snippets
                Uncommunicative Names: Char[] a

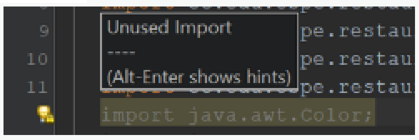


```java
public Boolean validateHour(String hour){
    boolean hourOfTreatment;
    char[] a = hour.toString().toCharArray();
    String[] c = hour.split( regex: ":");
    if ((a[0] == ' ') ||  (a[1] == ' ') || (a[2] == ' ')
            || (a[3] == ' ') || (a[4] == ' ')
            || (obtainInteger(c[0]) > 24) || (obtainInteger(c[1]) > 59)){
        hourOfTreatment = false;
    }else {
        hourOfTreatment = true;
    }
    return hourOfTreatment;
}
```

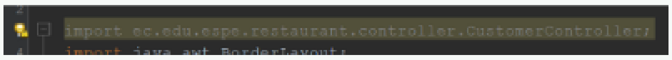


```java
public int obtainInteger(String value) {
    int integer = Integer.parseInt( s:value);
    return integer;
}
```

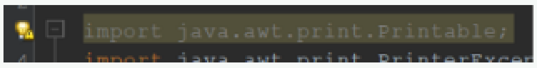
Dead code (Unused Imports):



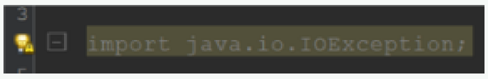
```
Unused Import
----
(Alt-Enter shows hints)
                  pe.restau
                  pe.restau
                  pe.restau
import java.awt.Color;
```



```java
import ec.edu.espe.restaurant.controller.CustomerController;
import java.awt.BorderLayout;
```



```java
import java.awt.print.Printable;
import java.awt.print.PrinterExcep
```



```java
import java.io.IOException;
```