

TEAM 05 DEES Dev's

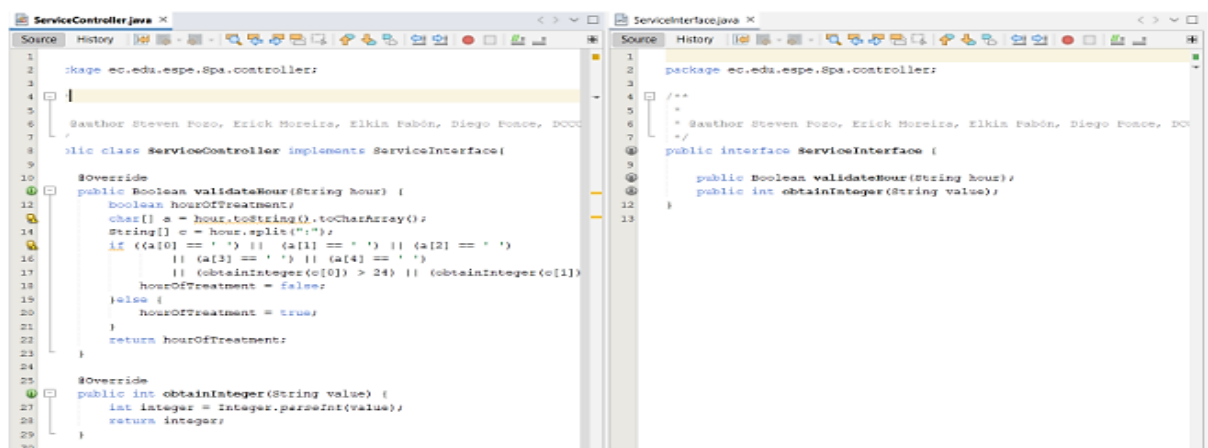
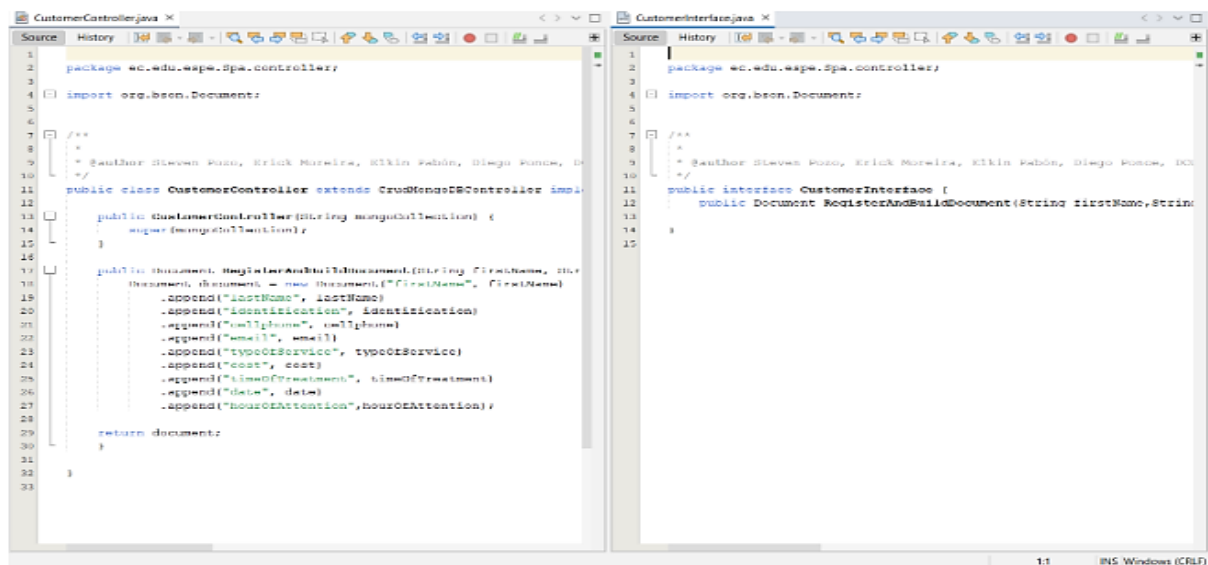
- 16. Moreira Vinueza Erick Patricio
- 17. Pabon Gonzalez Elkin Andres
- 18. Ponce Arguello Diego Armando
- 19. Pozo Analuisa Steven Jefferson

SOLID PRINCIPLE IN THE PROJECT:

- **Single principle:**
(they only do one thing)

```
11 public class MongoDBConnection {
12
13     public static MongoDBDatabase database = null;
14
15     public MongoDBDatabase connectDatabase() {
16
17         if (database == null) {
18             String uri = "mongodb://ec.edu.espe:27020/mongo03?authSource=admin&ssl=false&replicaSet=rs0";
19             String db = "icostatisticsvotest";
20
21             com.mongodb.client.MongoClient mongoClient = MongoClient.createFromUri(uri);
22
23             MongoDBConnection.database = mongoClient.getDatabase(db);
24
25         }
26         return MongoDBConnection.database;
27     }
28 }
```

- **Open/Closed principle:**
(There is no class change when it is extended)



- **Liskov Substitution principle:**
(meaning that we must make sure that new derived classes are extending the base classes without changing their behavior)

```

// CrudMongoDBController.java
package ec.edu.espe.spa.controller;

import com.mongodb.client.MongoCollection;
import static com.mongodb.client.model.Filters.eq;
import com.mongodb.client.model.ReplaceOptions;
import org.bson.Document;

/**
 *
 * @author Steven Pozo, Erick Moreira, Ekin Pabón, Diego Ponce, D
 */
public class CrudMongoDBController implements CrudMongoDBInterface {

    private MongoCollection<Document> mongoCollection;

    public CrudMongoDBController(String mongoCollection) {

        this.mongoCollection = MongoDbConnection.database.getCollection(mongoCollection);

    }

    @Override
    public void createToDatabase(Document document) {
        mongoCollection.insertOne(document);
    }

    @Override
    public Document readToDatabase(String key, String value) {
        return mongoCollection.find(eq(key, value)).first();
    }

    @Override
    public void uploadToDatabase(Document query, Document upload) {
        ReplaceOptions options = new ReplaceOptions().upsert(true);
        mongoCollection.replaceOne(query, upload, options);
    }
}

// CrudMongoDBInterface.java
package ec.edu.espe.spa.controller;

import com.mongodb.client.MongoCollection;
import org.bson.Document;

/**
 *
 * @author Steven Pozo, Erick Moreira, Ekin Pabón, Diego Ponce, D
 */
public interface CrudMongoDBInterface {

    public void createToDatabase(Document document);
    public Document readToDatabase(String key, String value);
    public Document read(Document document);
    public void uploadToDatabase(Document query, Document upload);
    public void deleteToDatabase(Document document);
    public MongoCollection<Document> getMongoCollection();
}

```

- **Interface segregation principle:**
(Interfaces should only have functions that are being used)

```

// CustomerInterface.java
package ec.edu.espe.spa.controller;

import org.bson.Document;

/**
 *
 * @author Steven Pozo, Erick Moreira, Ekin Pabón, Diego Ponce, D
 */
public interface CustomerInterface {

    public Document registerAndBuildDocument(String firstName, String lastName);
}

// UserInterface.java
package ec.edu.espe.spa.controller;

import ec.edu.espe.spa.model.User;
import org.bson.Document;

/**
 *
 * @author Steven Pozo, DCCD-ESPE, DEES Developers
 */
public interface UserInterface {

    public Document buildDocumentUser(User user);
}

// ServiceInterface.java
package ec.edu.espe.spa.controller;

/**
 *
 * @author Steven Pozo, Erick Moreira, Ekin Pabón, Diego Ponce, D
 */
public interface ServiceInterface {

    public Boolean validateHour(String hour);
    public int obtainInteger(String value);
}

// CrudMongoDBInterface.java
package ec.edu.espe.spa.controller;

import com.mongodb.client.MongoCollection;
import org.bson.Document;

/**
 *
 * @author Steven Pozo, Erick Moreira, Ekin Pabón, Diego Ponce, D
 */
public interface CrudMongoDBInterface {

    public void createToDatabase(Document document);
    public Document readToDatabase(String key, String value);
    public Document read(Document document);
    public void uploadToDatabase(Document query, Document upload);
    public void deleteToDatabase(Document document);
    public MongoCollection<Document> getMongoCollection();
}

```