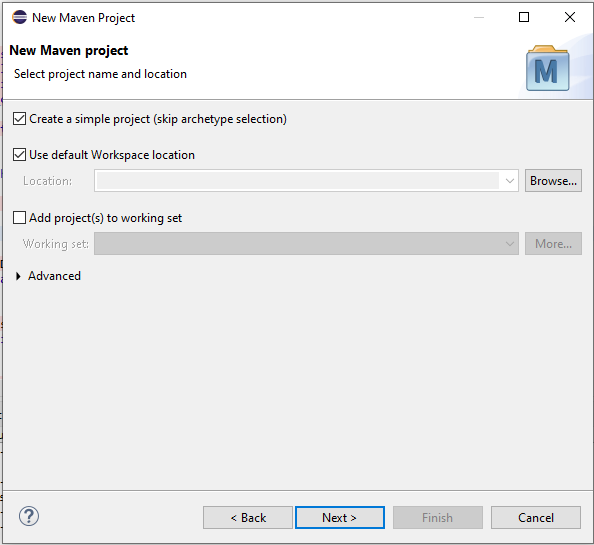
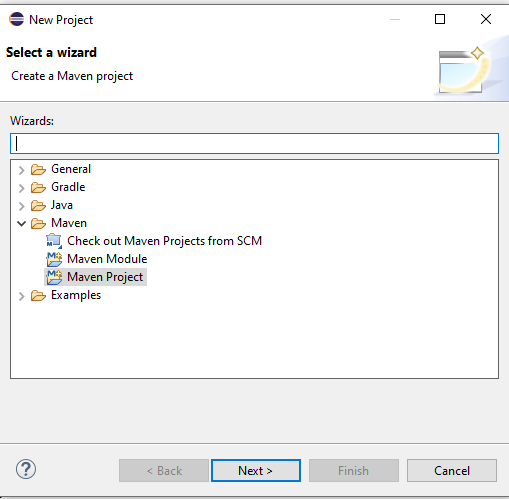
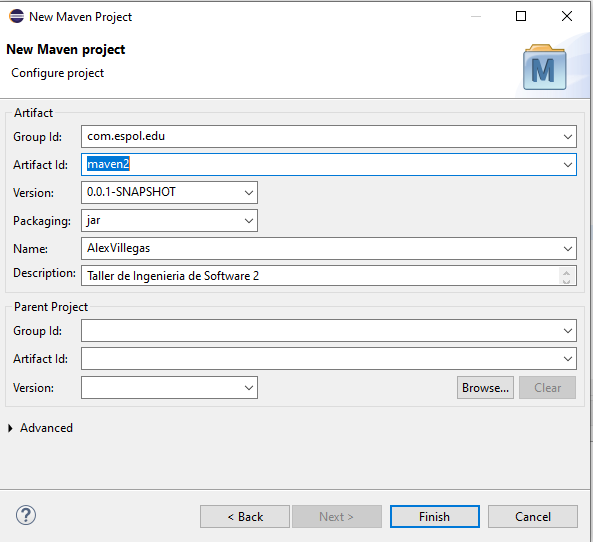
**Name:** Alex Joshua Villegas Paucar

**Date:** 01/06/2023

Lab report for Coding Standards

First, we start creating a new maven single project in eclipse.





Then, we generate the code according to the user requirements of the workshop.

**package** maven;

**public** **class** Vacation {

**private** String destination;

**private** **int** travelers;

**private** **int** duration;

**private** **int** base\_cost = 1000;

**private** **double** total\_cost = 0;

**public** Vacation() {

}

**public** String getDestination() {

**return** destination;

}

**public** **void** setDestination(String destination) {

**this**.destination = destination;

}

**public** **int** getTravelers() {

**return** travelers;

}

**public** **void** setTravelers(**int** travelers) {

**this**.travelers = travelers;

}

**public** **int** getDuration() {

**return** duration;

}

**public** **void** setDuration(**int** duration) {

**this**.duration = duration;

}

**public** **int** getBase\_cost() {

**return** base\_cost;

}

**public** **void** setBase\_cost(**int** base\_cost) {

**this**.base\_cost = base\_cost;

}

**public** **double** getTotal\_cost() {

**if** (total\_cost == 0){

**return** -1;

}

**return** total\_cost;

}

**public** **void** setTotal\_cost(**double** total\_cost) {

**this**.total\_cost = total\_cost;

}

**public** **double** verifyDestiny(String destination) {

**int** additional = 0;

**if**(destination.toUpperCase().equals("Paris")) {

additional = 500 + **this**.base\_cost;

}

**else** **if**(destination.toUpperCase().equals("New York City")) {

additional= 600 + **this**.base\_cost;

}

**return** additional + **this**.base\_cost;

}

**public** **void** getCost(**double** totalcost, **int** travelers, **int** duration) {

**double** discount = 0;

**double** fee = 200;

**if**(travelers>4 && travelers <10) {

discount = total\_cost\*0.10;

**this**.total\_cost = totalcost - discount;

}

**if**(travelers>10) {

discount = total\_cost\*0.20;

**this**.total\_cost = totalcost - discount;

}

**if**(duration<7) {

**this**.total\_cost = totalcost + fee;

}

**if**(duration>30 || travelers == 2) {

**this**.total\_cost = totalcost - fee;

}

**else** {

**this**.total\_cost = **this**.base\_cost;

}

}

}

**package** maven;

**import** java.util.Scanner;

**public** **class** VacationEstimator {

//CHECKSTYLE:OFF

**public** **static** **void** main(String[] args) {

//CHECKSTYLE:ON

Vacation vac1 = **new** Vacation();

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("\*\*\*\*\*Vacation Package Estimator\*\*\*\*\*");

System.***out***.println("Enter your destination:");

vac1.setDestination(sc.nextLine());

System.***out***.println("Enter the number of travelers:");

vac1.setTravelers(sc.nextInt());

System.***out***.println("Enter the duration in days:");

vac1.setDuration(sc.nextInt());

**if**(vac1.getTravelers()>80) {

System.***out***.println("The vacation package is not available for groups of more than 80 persons");

}

**else** {

**double** total\_cost = vac1.verifyDestiny(vac1.getDestination());

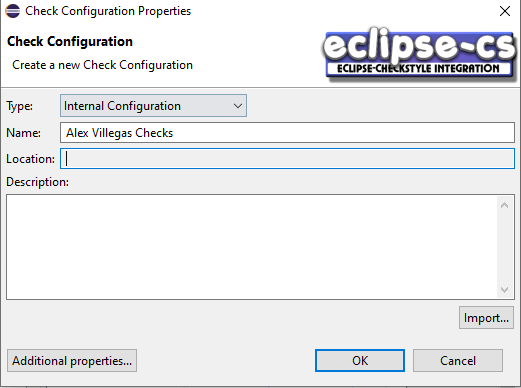
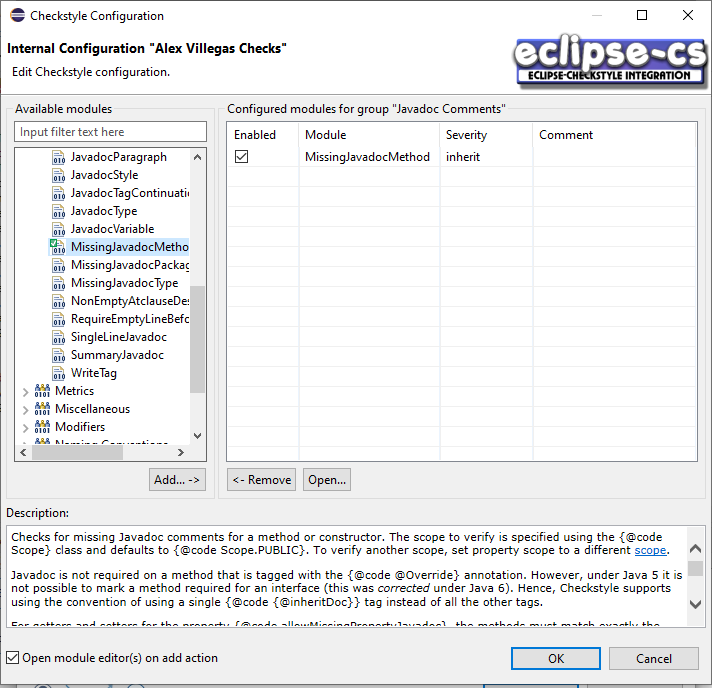
vac1.getCost(total\_cost, vac1.getTravelers(), vac1.getDuration());

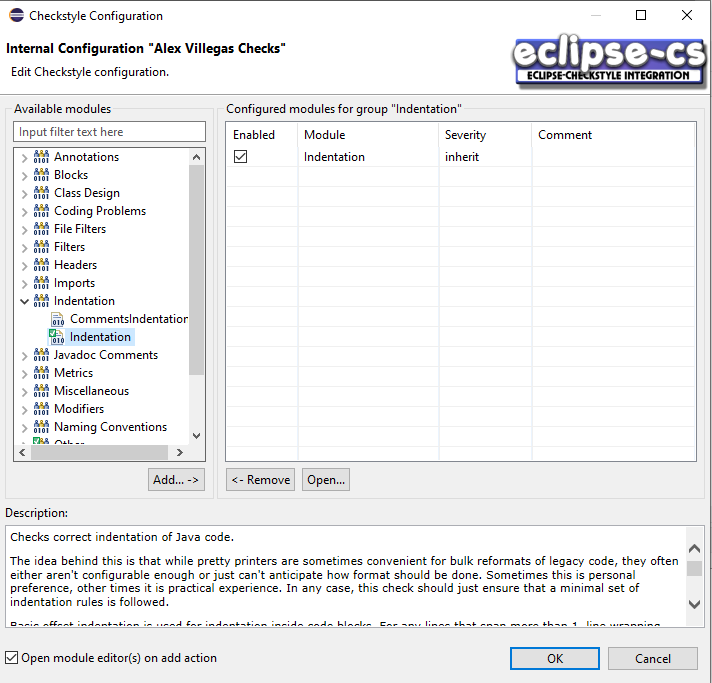
System.***out***.println("The total cost of the vacation package is " + vac1.getTotal\_cost());

}

}

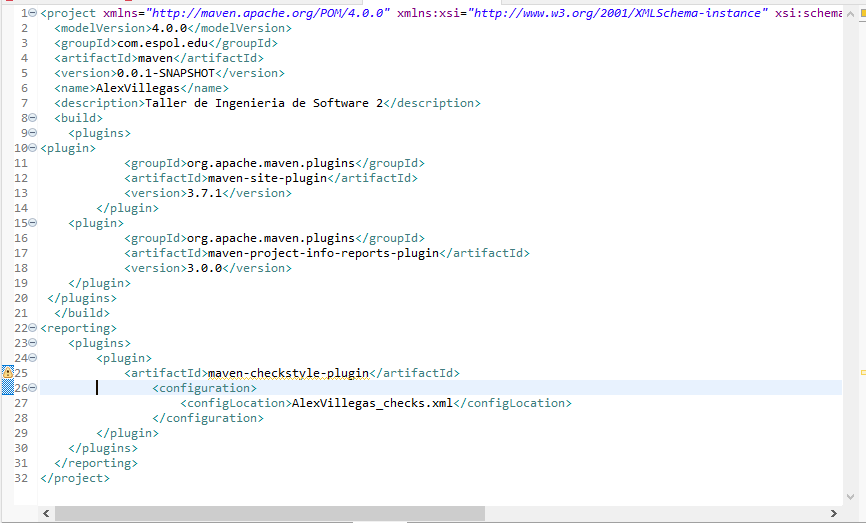
}

We proceed to configure the checkstyle for our project.

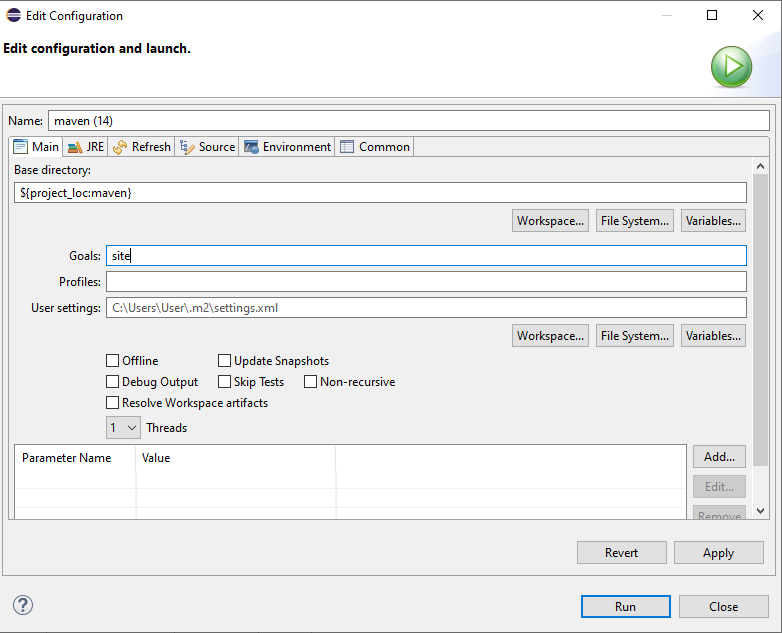


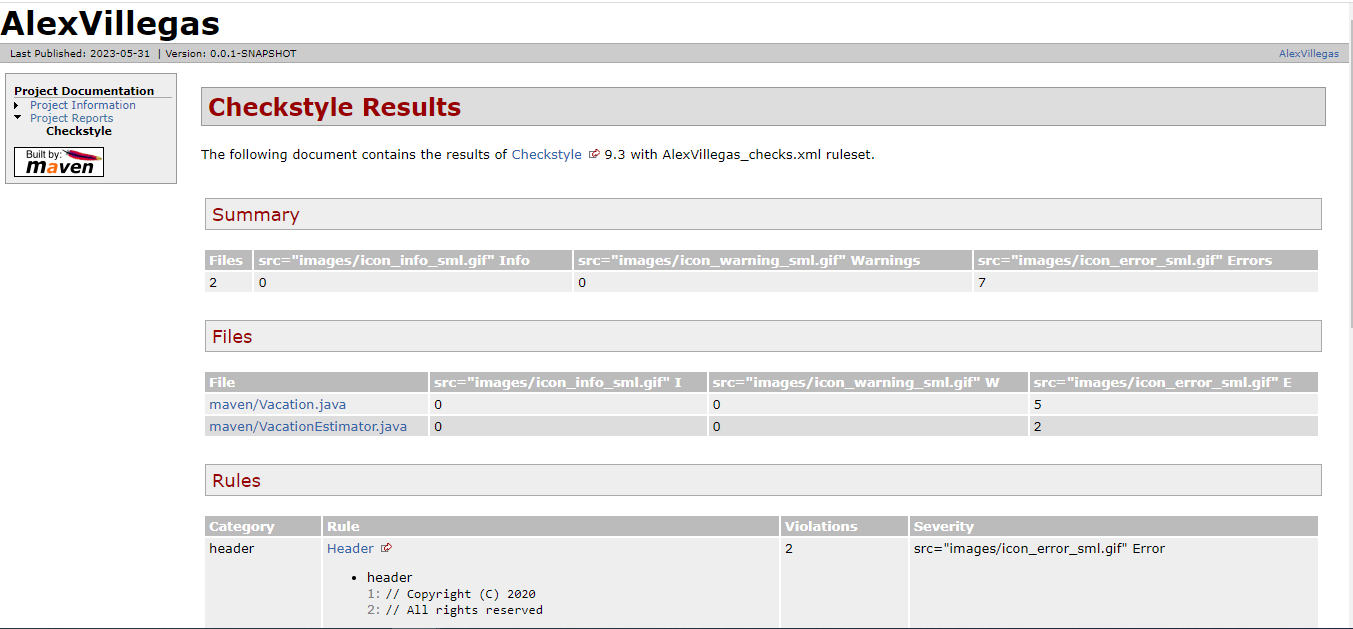
Then, we configure the checkstyle with XML adding the rules presented in the guide and 6 new rules.



We set up the basic configuration of your pom.xml, adding some configurations given in the guide.

Then, we run the pom.xml archive as “maven build…”, set the goal as “site”, and wait for the corresponding report of the results.

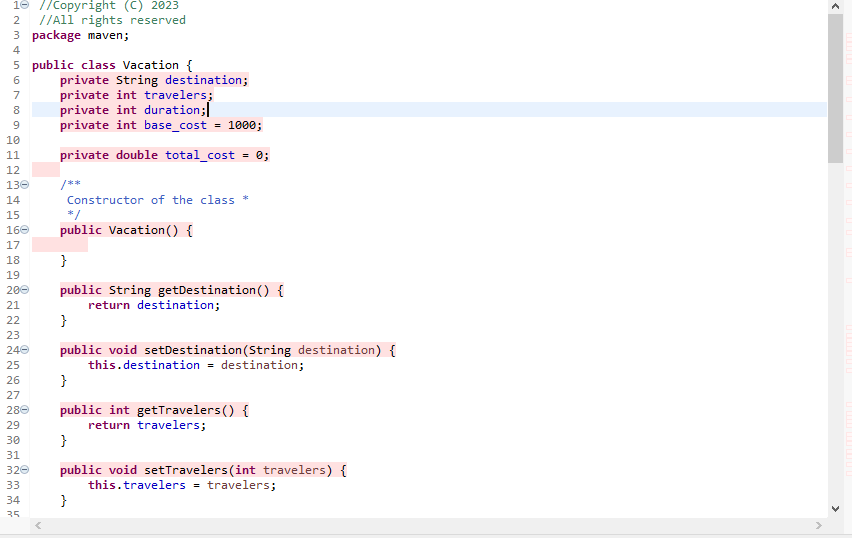




Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

After the report is done, we can see that wee have some errors that we must correct and at what type of error we are fighting. So, we proceed to resolve it.



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteHaving fixed totally or partial the errors, we run again the pom.xml file to see if in the report the errors were fixed or not.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

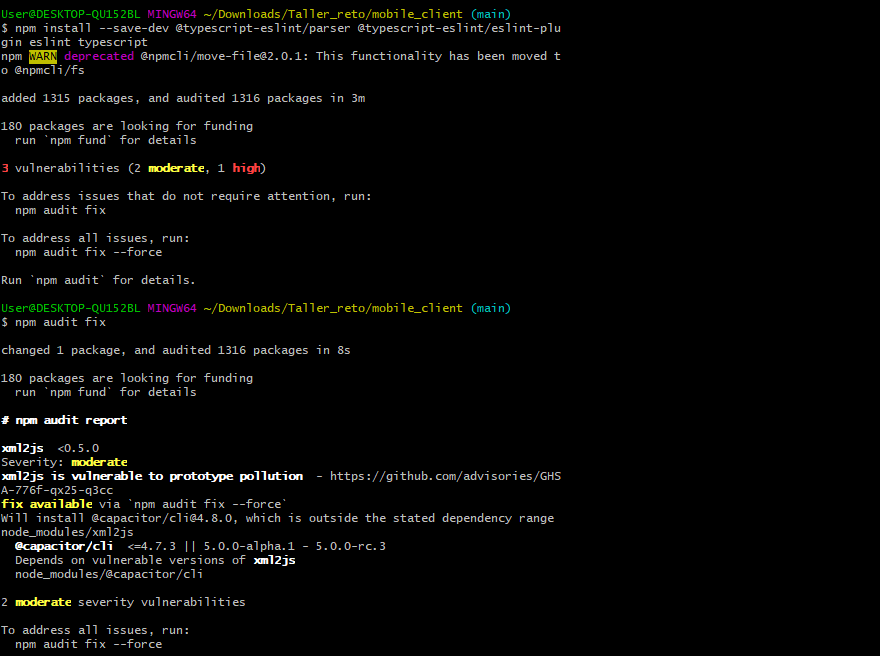
Descripción generada automáticamente

Unfortunately, we have corrected the majority of the errors, the principals, that means that our code style is better.

# Extra:

For our project of the subject, we simulate a similar process using a different codding standard tool. The tool that we use in this case is “typescript-eslint”, it is used for typescript files.

First, install the required packages for ESLint, TypeScript, and this plugin:



Then, we create a “. eslintrc.cjs” config file in the root of the project, and write inside it the following:

/\* eslint-env node \*/

module.exports = {

extends: ['eslint:recommended', 'plugin:@typescript-eslint/recommended'],

parser: '@typescript-eslint/parser',

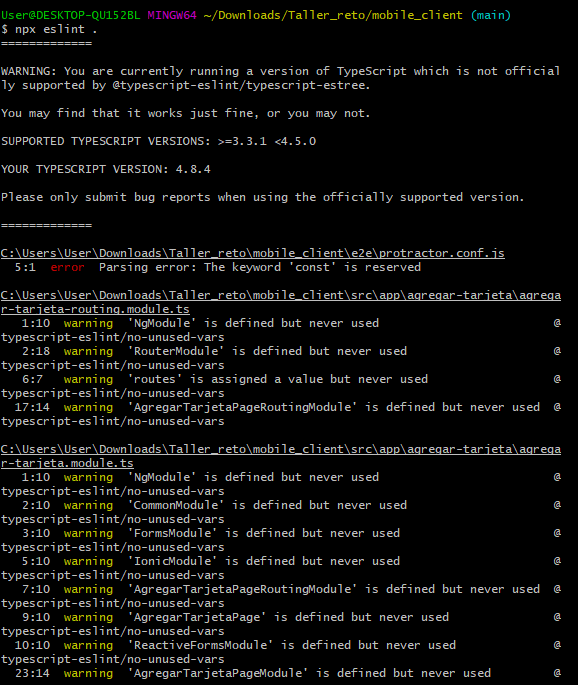
plugins: ['@typescript-eslint'],

root: true,

};

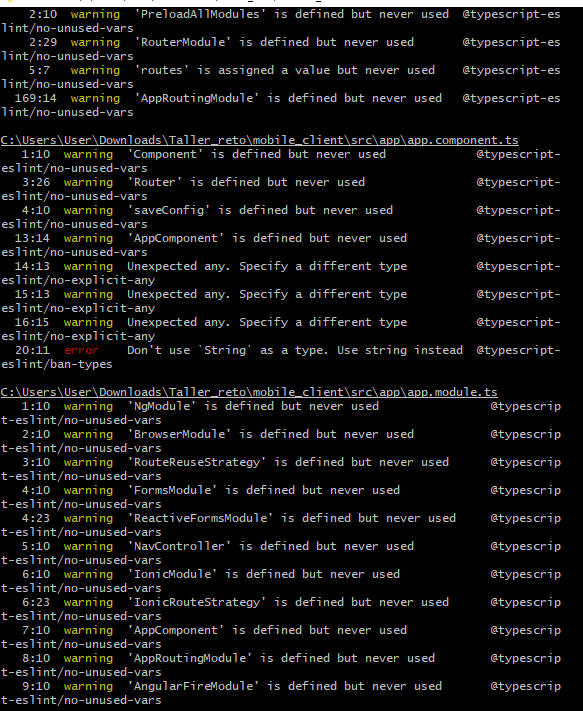
Finally, we run the commad “ npx eslint .” in a new terminal open in the root of the project.

ESLint will lint all TypeScript compatible files within the current folder, and will output the results to your terminal.



Texto

Descripción generada automáticamente



Texto

Descripción generada automáticamente

Having that report in our terminal, we know what the errors are we have in our project, and we can fix it property.

# Resources:

***Link of the public repository:*** <https://github.com/ajvilleg10/CodingStandardLabING2>