UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

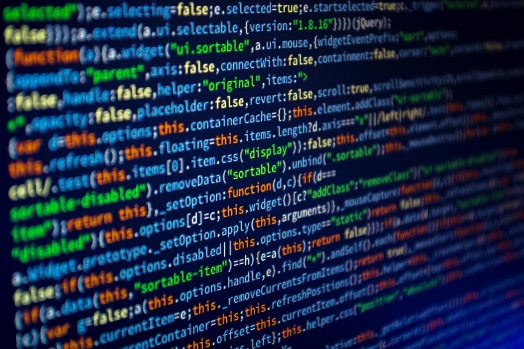
FACULTAD DE CIENCIAS QUÍMICA E INGENIERÍAS



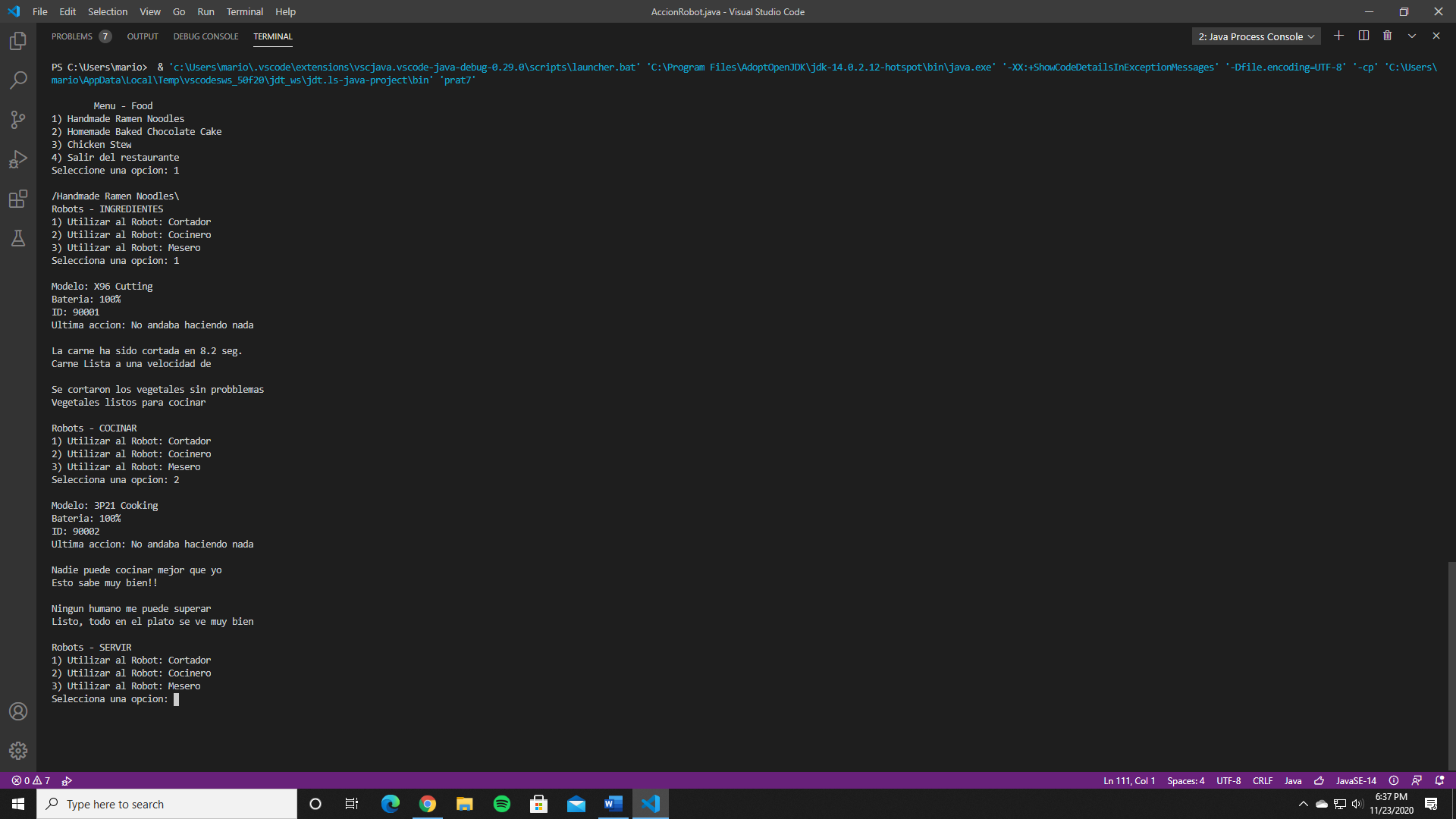
Prof. Mayra Duran Rodríguez  
Programación orientada a objetos

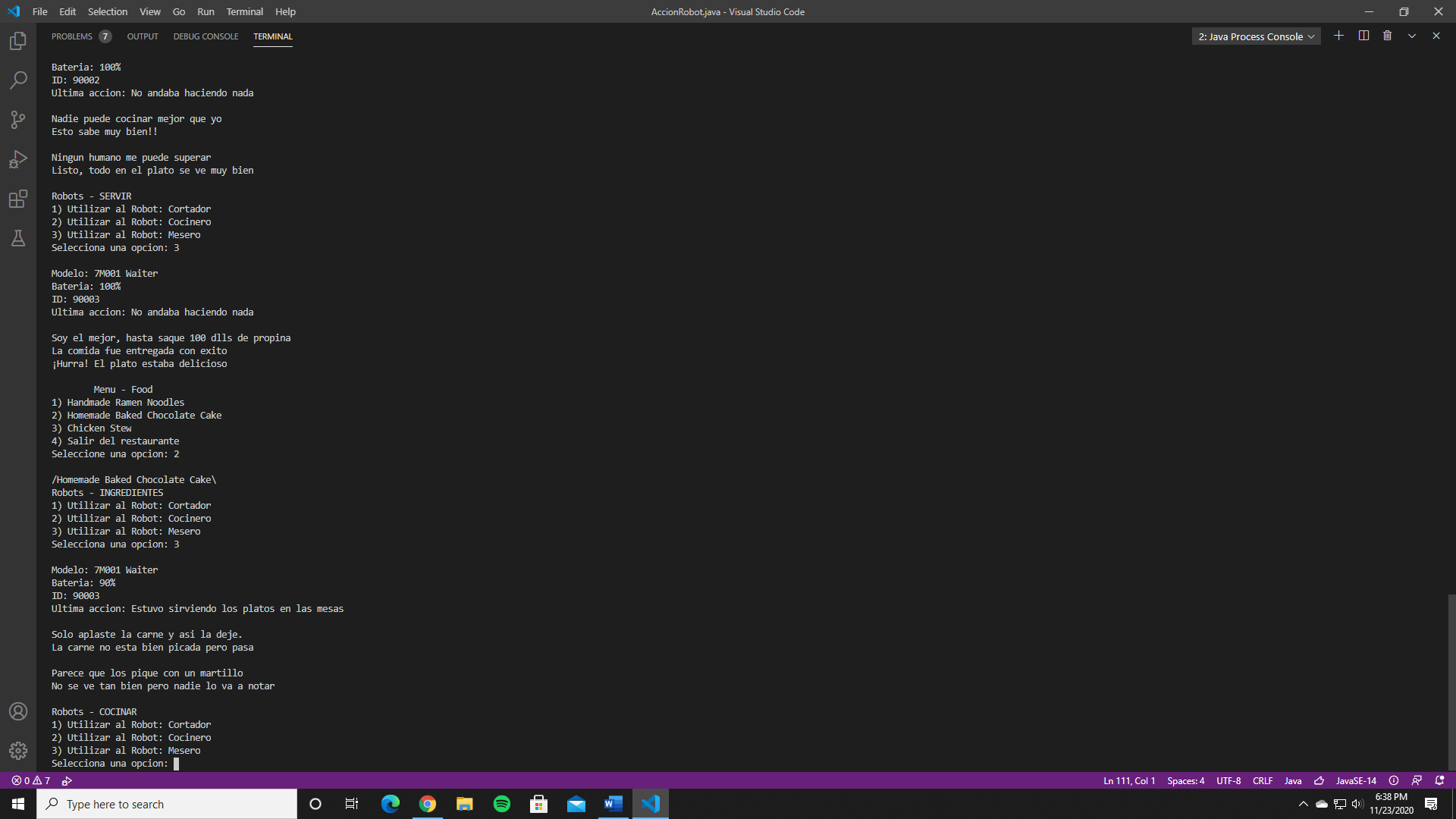
**Practica #7**

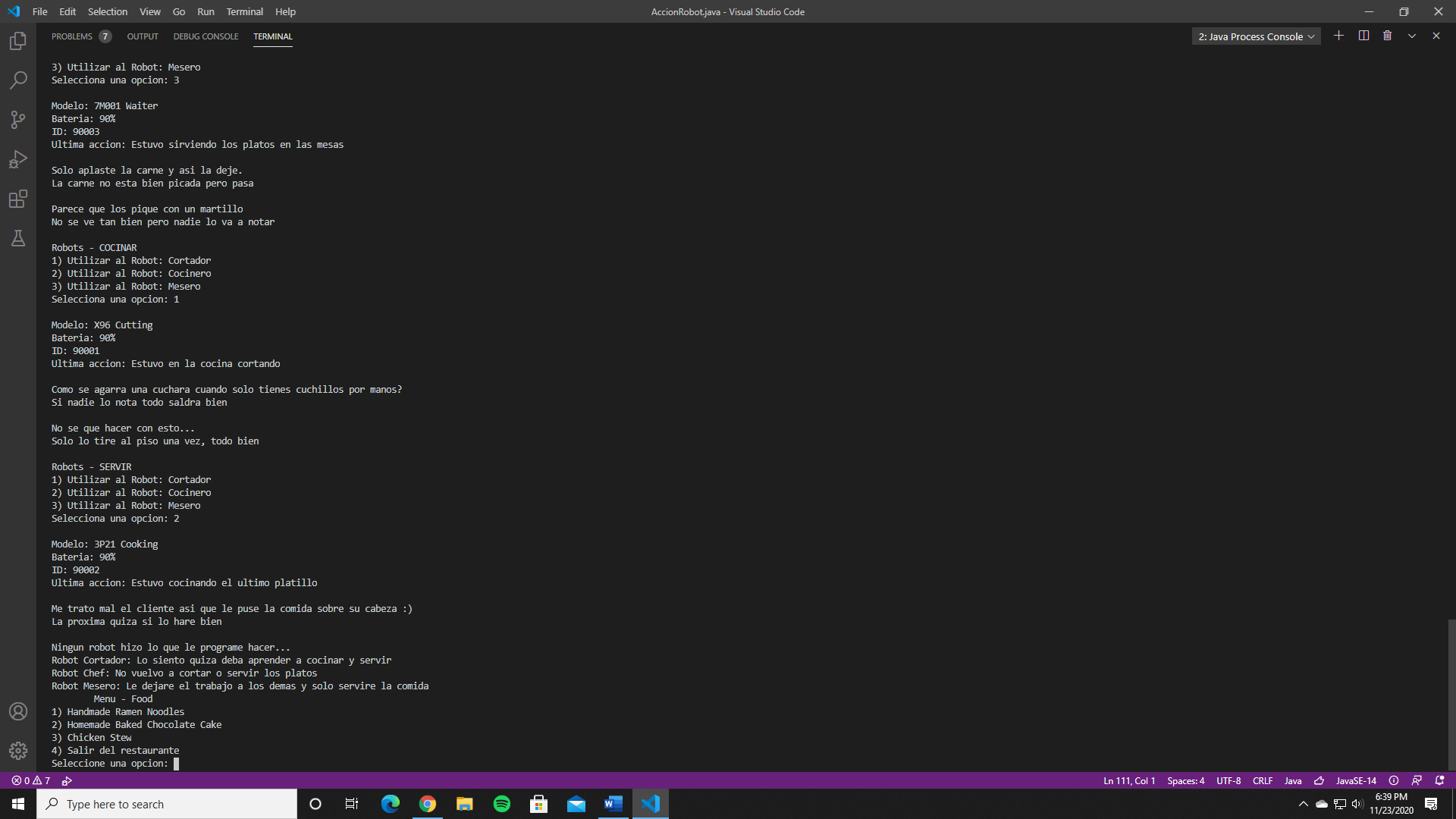
Mario Mendoza Virgen   
Matricula 1261594

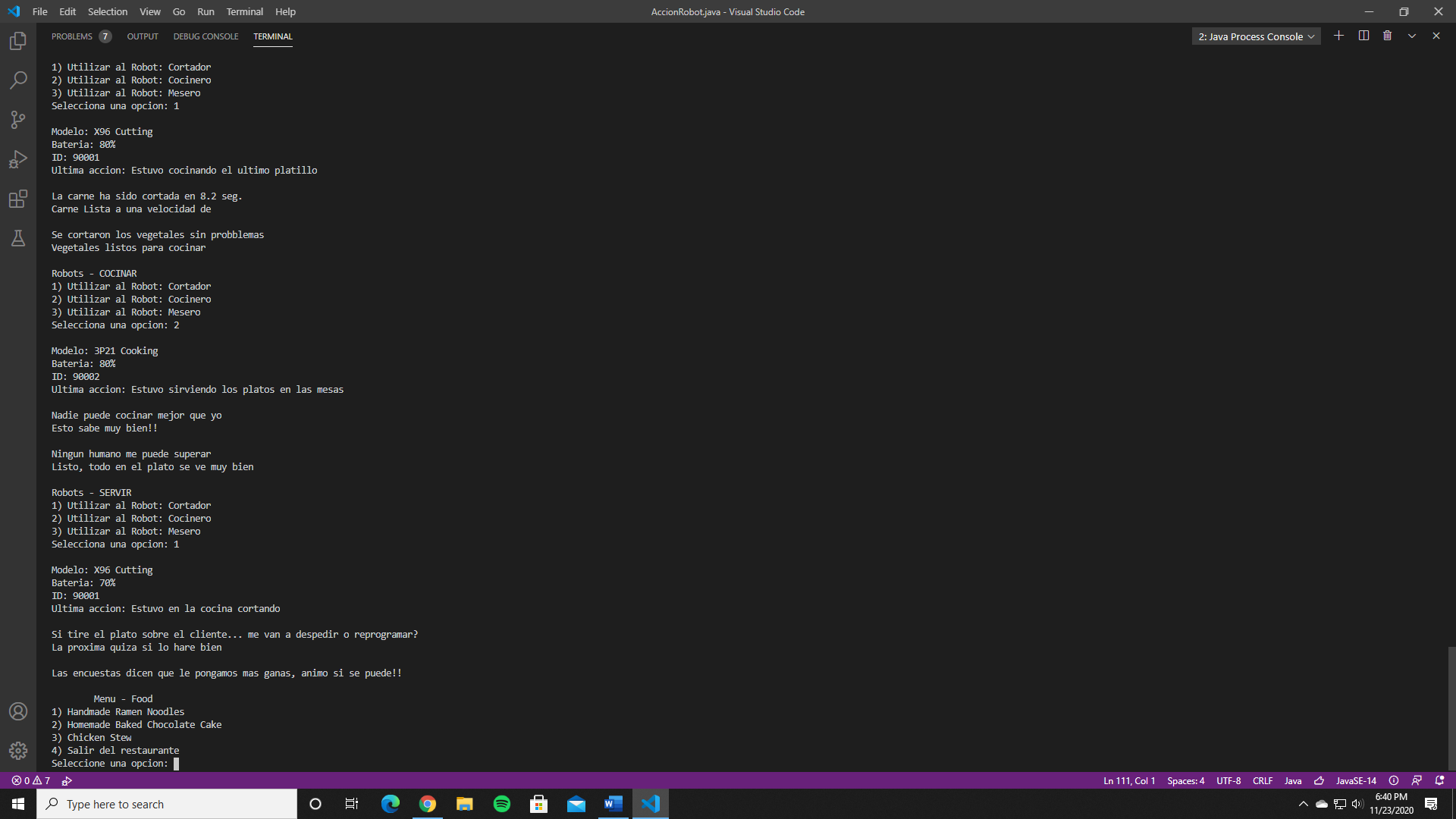


Tijuana, Baja California, México 24 de noviembre de 2020









Pract7.java

import java.util.Scanner;

public class prat7 {

    public static void main(String args[]){

        char opc;

        Scanner sc = new Scanner(System.in);

        Robot bot[] = new Robot[3];

        bot[0] = new RobotCutMeat("X96 Cutting", 100, 90001, 0);

        bot[1] = new RobotChef("3P21 Cooking", 100, 90002, 0);

        bot[2] = new RobotWaiter("7M001 Waiter", 100, 90003, 0);

        do{

            menuPrincipal();

            opc = sc.nextLine().charAt(0);

        switch(opc){

            case '1':

                System.out.println("\n/Handmade Ramen Noodles\\");

                AccionRobot.ingredientes(bot);

                AccionRobot.cocinar(bot);

                AccionRobot.servir(bot);

                break;

            case '2':

                System.out.println("\n/Homemade Baked Chocolate Cake\\");

                AccionRobot.ingredientes(bot);

                AccionRobot.cocinar(bot);

                AccionRobot.servir(bot);

                break;

            case '3':

                System.out.println("\n/Boil Stew Chicken\\");

                AccionRobot.ingredientes(bot);

                AccionRobot.cocinar(bot);

                AccionRobot.servir(bot);

                break;

            case '4':

                System.out.println("\nSaliendo del restaurante...");

                break;

            default:

                System.out.println("\nLo sentimos, esa opcion no esta disponible");

                break;

        }

        if(opc != 4)

            AccionRobot.rateFood(bot);

        chargeRobot(bot);

    }while(opc != '4');

        sc.close();

    }

    public static void chargeRobot(Robot bot[]){

        for(int i = 0; i<3; i++){

            if(bot[i].battery <= 0){

                bot[i].battery = 100;

                bot[i].action = 4; //Id de carga para la ultima accion

            }

        }

    }

    public static void menuPrincipal(){

        System.out.println("\n\tMenu - Food");

        System.out.println("1) Handmade Ramen Noodles");

        System.out.println("2) Homemade Baked Chocolate Cake");

        System.out.println("3) Chicken Stew");

        System.out.println("4) Salir del restaurante");

        System.out.print("Seleccione una opcion: ");

    }

}

Interfaz.java

public interface Interfaz {

    public void battery(int num);

    public void name(String name);

    public void id(int num);

    public void lastAction(int num);

}

Robot.java

public abstract class Robot implements Interfaz{

    String name;

    int battery;

    int id;

    int action;

    public Robot(String name, int battery, int id, int action){

        this.name = name;

        this.battery = battery;

        this.id = id;

        this.action = action;

    }

    public void battery(){}

    public void name(){}

    public void id(){}

    public void lastAction(){}

    protected abstract boolean cutMeat();

    protected abstract boolean cutVeggies();

    protected abstract boolean cookFood();

    protected abstract boolean prepare();

    protected abstract boolean serveFood();

    protected abstract void error();

}

RobotCutMeat.java

public class RobotCutMeat extends Robot{

    public RobotCutMeat(String name, int battery, int id, int action) {

        super(name, battery, id, action);

    }

    public void name(String name) {

        this.name = name;

    }

    public void battery(int battery) {

        this.battery = battery;

    }

    public void id(int id) {

        this.id = id;

    }

    public void lastAction(int num) {

        this.action = num;

    }

    @Override

    protected boolean cutMeat(){

        System.out.println("\nLa carne ha sido cortada en 8.2 seg.");

        return true;

    }

    @Override

    protected boolean cutVeggies(){

        System.out.println("\nSe cortaron los vegetales sin probblemas");

        return true;

    }

    @Override

    protected boolean cookFood() {

        System.out.println("\nComo se agarra una cuchara cuando solo tienes cuchillos por manos?");

        return false;

    }

    @Override

    protected boolean prepare() {

        System.out.println("\nNo se que hacer con esto...");

        return false;

    }

    @Override

    protected boolean serveFood() {

        System.out.println("\nSi tire el plato sobre el cliente... me van a despedir o reprogramar?");

        return false;

    }

    @Override

    protected void error() {

        System.out.print("Robot Cortador: Lo siento quiza deba aprender a cocinar y servir");

    }

}

RobotChef.java

public class RobotChef extends Robot{

    public RobotChef(String name, int battery, int id, int action) {

        super(name, battery, id, action);

    }

    public void name(String name) {

        this.name = name;

    }

    public void battery(int battery) {

        this.battery = battery;

    }

    public void id(int id) {

        this.id = id;

    }

    public void lastAction(int num) {

        this.action = num;

    }

    @Override

    protected boolean cutMeat(){

        System.out.println("\nLa carne fue cortada demasiado fina.");

        return false;

    }

    @Override

    protected boolean cutVeggies(){

        System.out.println("\nSe cortaron los vegetales sin probblemas pero se me olvido cortar tomate");

        return false;

    }

    @Override

    protected boolean cookFood() {

        System.out.println("\nNadie puede cocinar mejor que yo");

        return true;

    }

    @Override

    protected boolean prepare() {

        System.out.println("\nNingun humano me puede superar");

        return true;

    }

    @Override

    protected boolean serveFood() {

        System.out.println("\nMe trato mal el cliente asi que le puse la comida sobre su cabeza :)");

        return false;

    }

    @Override

    protected void error() {

        System.out.print("\nRobot Chef: No vuelvo a cortar o servir los platos");

    }

}

RobotWaiter.java

public class RobotWaiter extends Robot{

    public RobotWaiter(String name, int battery, int id, int action) {

        super(name, battery, id, action);

    }

    public void name(String name) {

        this.name = name;

    }

    public void battery(int battery) {

        this.battery = battery;

    }

    public void id(int id) {

        this.id = id;

    }

    public void lastAction(int num) {

        this.action = num;

    }

    @Override

    protected boolean cutMeat(){

        System.out.println("\nSolo aplaste la carne y asi la deje.");

        return false;

    }

    @Override

    protected boolean cutVeggies(){

        System.out.println("\nParece que los pique con un martillo");

        return false;

    }

    @Override

    protected boolean cookFood() {

        System.out.println("\nSe le pone sal o azucar?");

        return false;

    }

    @Override

    protected boolean prepare() {

        System.out.println("\nNo se donde poner este steak");

        return false;

    }

    @Override

    protected boolean serveFood() {

        System.out.println("\nSoy el mejor, hasta saque 100 dlls de propina");

        return true;

    }

    @Override

    protected void error() {

        System.out.print("\nRobot Mesero: Le dejare el trabajo a los demas y solo servire la comida");

    }

}

AccionRobot.java

import java.util.Scanner;

public class AccionRobot {

    static int ref=0;

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

    public static void robotStatistics(Robot bot[],int a){

        String [] acciones = new String [5];

        acciones[0] = "No andaba haciendo nada";

        acciones[1] = "Estuvo en la cocina cortando";

        acciones[2] = "Estuvo cocinando el ultimo platillo";

        acciones[3] = "Estuvo sirviendo los platos en las mesas";

        acciones[4] = "Estuvo recargando bateria en el centro de carga";

        System.out.println("\nModelo: " + bot[a].name);

        System.out.println("Bateria: " + bot[a].battery + "%");

        System.out.println("ID: " + bot[a].id);

        System.out.println("Ultima accion: " + acciones[bot[a].action]);

    }

    public static void ingredientes(Robot bot[]){

        int a = 0;

        do{

        menuRobots();

        a = sc.nextInt() - 1;

        sc.nextLine();

        if(a > 3 && a < 0){

            System.out.println("Error, vuelve a intentarlo\n");

        }

    }while(a > 3 && a < 0);

        robotStatistics(bot, a);

        if(bot[a].cutMeat()){

            System.out.println("Carne Lista a una velocidad de ");//Agregar la velocidad del robot

        }

        else{

            System.out.println("La carne no esta bien picada pero pasa");

        }

        if(bot[a].cutVeggies()){

            System.out.println("Vegetales listos para cocinar");

            ref++;

        }

        else{

            System.out.println("No se ve tan bien pero nadie lo va a notar");

        }

        bot[a].action = 1; // 1 es el id de Ingredientes para la ultima accion que realizo el robot

        bot[a].battery -= 10;

    }

    public static void cocinar(Robot bot[]){

        int a = 0;

        do{

        menuRobots2();

        a = sc.nextInt() - 1;

        sc.nextLine();

        if(a > 3 && a < 0){

            System.out.println("Error, vuelve a intentarlo\n");

        }

    }while(a > 3 && a < 0);

        robotStatistics(bot, a);

        if(bot[a].cookFood()){

            System.out.println("Esto sabe muy bien!!");

        }

        else{

            System.out.println("Si nadie lo nota todo saldra bien");

        }

        if(bot[a].prepare()){

            System.out.println("Listo, todo en el plato se ve muy bien");

            ref++;

        }

        else{

            System.out.println("Solo lo tire al piso una vez, todo bien");

        }

        bot[a].action = 2;

        bot[a].battery -= 10;

    }

    public static void servir(Robot bot[]){

        int a = 0;

        do{

        menuRobots3();

        a = sc.nextInt() - 1;

        sc.nextLine();

        if(a > 3 && a < 0){

            System.out.println("Error, vuelve a intentarlo\n");

        }

    }while(a > 3 && a < 0);

        robotStatistics(bot, a);

        if(bot[a].serveFood()){

            System.out.println("La comida fue entregada con exito");

            ref++;

        }

        else{

            System.out.println("La proxima quiza si lo hare bien");

        }

        bot[a].action = 3;

        bot[a].battery -= 10;

    }

    public static void rateFood(Robot bot[]){

        if(ref == 0){

            System.out.println("\nNingun robot hizo lo que le programe hacer...");

           bot[0].error();

           bot[1].error();

           bot[2].error();

        }

        if(ref == 1){

            System.out.println("\nAl menos el cliente nos puso 3 estrellas");

        }

        if(ref == 2){

            System.out.println("\nLas encuestas dicen que le pongamos mas ganas, animo si se puede!!");

        }

        if(ref == 3){

            System.out.println("¡Hurra! El plato estaba delicioso");

        }

        ref = 0;

    }

    public static void menuRobots(){

        System.out.println("Robots - INGREDIENTES");

        System.out.println("1) Utilizar al Robot: Cortador");

        System.out.println("2) Utilizar al Robot: Cocinero");

        System.out.println("3) Utilizar al Robot: Mesero");

        System.out.print("Selecciona una opcion: ");

    }

    public static void menuRobots2(){

        System.out.println("\nRobots - COCINAR");

        System.out.println("1) Utilizar al Robot: Cortador");

        System.out.println("2) Utilizar al Robot: Cocinero");

        System.out.println("3) Utilizar al Robot: Mesero");

        System.out.print("Selecciona una opcion: ");

    }

    public static void menuRobots3(){

        System.out.println("\nRobots - SERVIR");

        System.out.println("1) Utilizar al Robot: Cortador");

        System.out.println("2) Utilizar al Robot: Cocinero");

        System.out.println("3) Utilizar al Robot: Mesero");

        System.out.print("Selecciona una opcion: ");

    }

}