Código Fuente

```java
package States;
/**
 * @author Victor Lavalle
 */
public class Turnstile {

 public void lock(){
 System.out.println("Locked\n\n");
 }

 public void unlock(){
 System.out.println("Unlocked\n\n");
 }

 public void thanks(){
 System.out.println("Thanks!\n\n");
 }

 public void alarm(){
 System.out.println("Sounding alarm...!\n\n");
 }

 public void resetAlarm(){
 System.out.println("Alarm stopped\n\n");
 }

}
```

```java
package States;
/**
 * @author Victor Lavalle
 */
public class TurnstileFSM extends Turnstile {
 private States state;

 public TurnstileFSM(){
 state = new Locked();
 }

 public void setState(States s){ state = s;}

 public void coin(){ state.coin(this); }

 public void pass(){ state.pass(this); }

 public void reset(){ state.reset(this); }

 public void ready(){ state.ready(this); }

}
```

```java
package States;
/**
 * @author Victor Lavalle
 */
public interface States {
 public void coin(TurnstileFSM f);

 public void pass(TurnstileFSM f);

 public void reset(TurnstileFSM f);

 public void ready(TurnstileFSM f);
}
```

```java
package States;
/**
 * @author Victor Lavalle
 */
public class Locked implements States{

 @Override
 public void coin(TurnstileFSM f) {
 f.setState(new Unlocked());
 f.unlock();     }

 @Override
 public void pass(TurnstileFSM f) {
 f.alarm();
 f.setState(new Violation());    }

 @Override
 public void reset(TurnstileFSM f) { }

 @Override
 public void ready(TurnstileFSM f) { }
}
```

```java
package States;
/**
 * @author Victor Lavalle
 */
public class Unlocked implements States{
@Override
 public void coin(TurnstileFSM f) {
 f.thanks();  }

 @Override
 public void pass(TurnstileFSM f) {
 System.out.println("Passing[...]");
 f.setState(new Locked());
 f.lock();   }

 @Override
 public void reset(TurnstileFSM f) { }

 @Override
 public void ready(TurnstileFSM f) { }
}
```

```java
package States;
/**
 * @author Victor Lavalle
 */
public class Violation implements States {

@Override
 public void coin(TurnstileFSM f) { }

 @Override
 public void pass(TurnstileFSM f) { }

 @Override
 public void reset(TurnstileFSM f){
 f.resetAlarm();
 }

 @Override
 public void ready(TurnstileFSM f){
 f.resetAlarm();
 f.setState(new Locked());
 f.lock();
 }

}
```

```java
package Launcher;
/**
 * @author Victor Lavalle
*/
import States.*;
import java.util.Scanner;

public class Main {

    public static void Menu(){
     System.out.println("Select an Option:"
             + "\n _____________________"
             + "\n|1. Insert Coin    |"
             + "\n|2. Pass           |"
             + "\n|3. Reset          |"
             + "\n|4. Ready          |"
             + "\n|0. Exit           |"
             + "\n --------------------"
             + "\n>>");
    }

    public static void main(String[] args) {

        Scanner scanAction = new Scanner(System.in);
        TurnstileFSM turnstile = new TurnstileFSM();

        turnstile.setState(new Locked());//It starts being locked

        System.out.println("   *Turnstile*");

        while(true){
           Menu();
           String  action= scanAction.nextLine();
            switch(action){
```

```
35
36                  case "0":
37                      System.exit(0);
38                  break;
39                  case "1":
40                     turnstile.coin();
41                  break;
42
43                  case "2":
44                     turnstile.pass();
45                  break;
46
47                  case "3":
48                      turnstile.reset();
49                   break;
50
51                   case "4":
52                      turnstile.ready();
53                   break;
54
55                  default:  System.out.println("Invalid Option!");
56              }
57          }
58      }
59  }
```

Diagrama de Clases

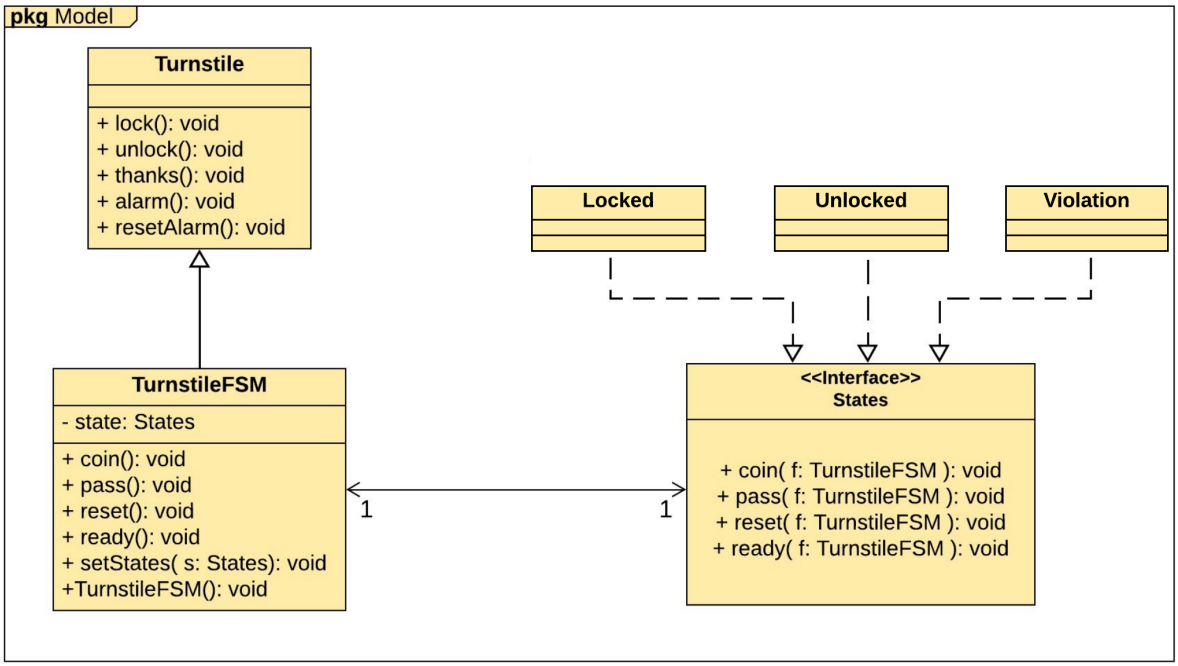
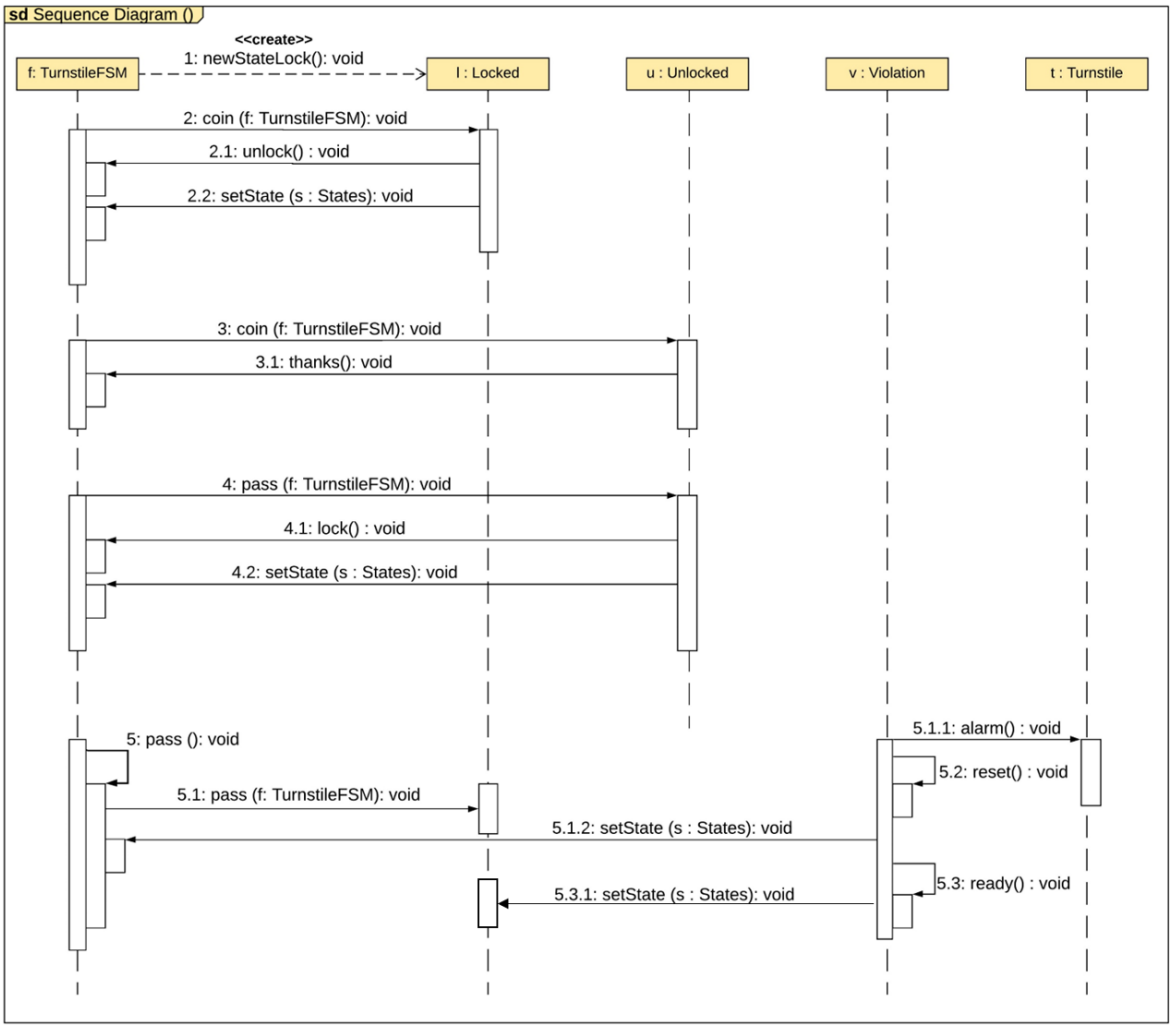
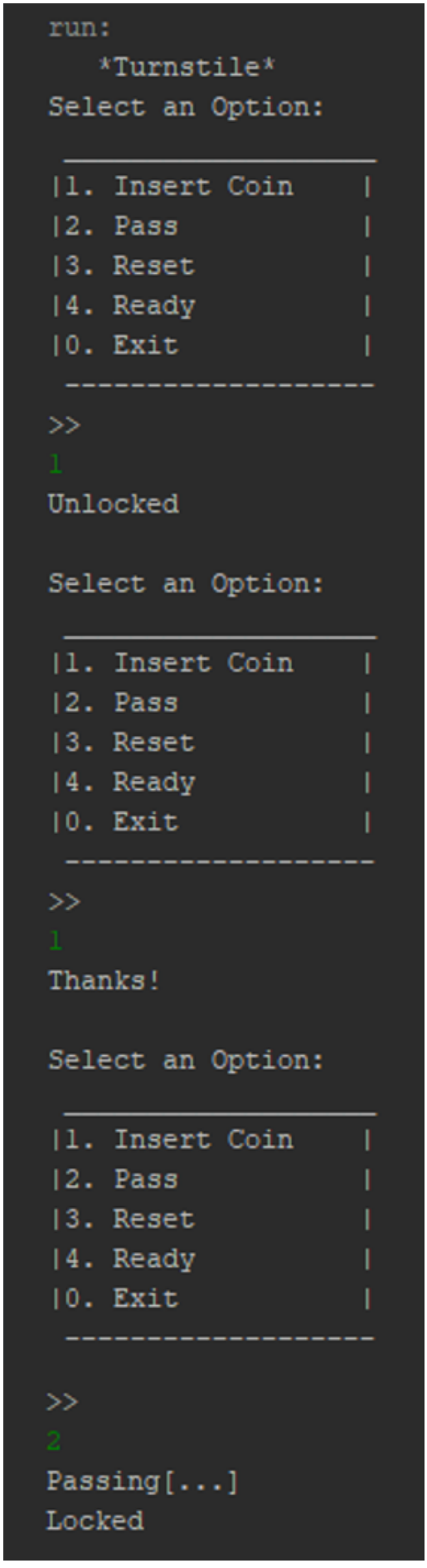


Diagrama de Secuencia

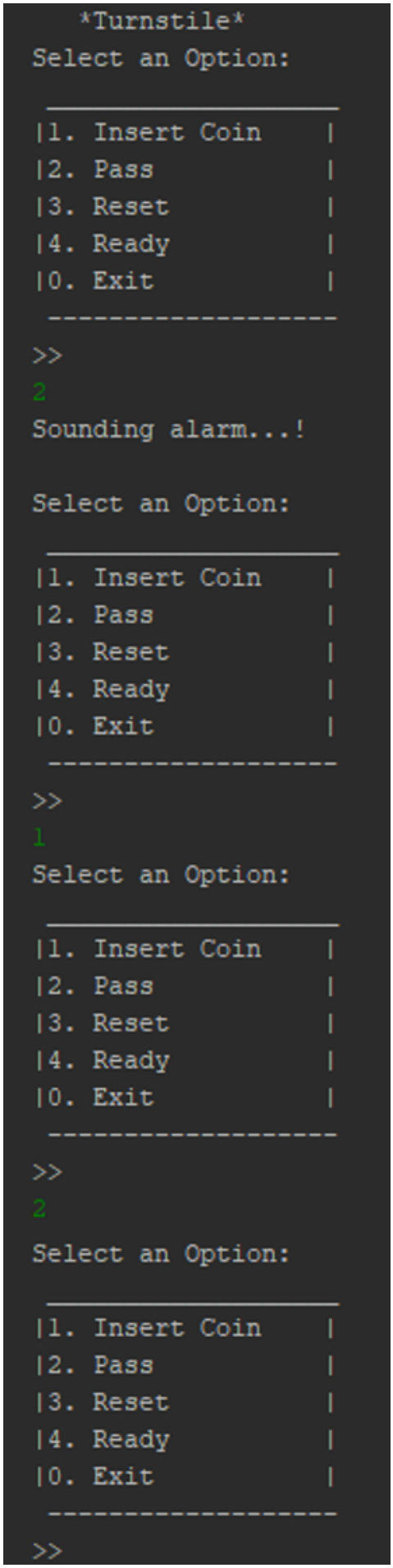


Ejecución del Programa

Ejecución Exitosa



Ejecución cuando se quiere pasar sin haber ingresado una moneda y entra al estado de violación.



*Continuación estando en el estado de violación y como se sale de este.