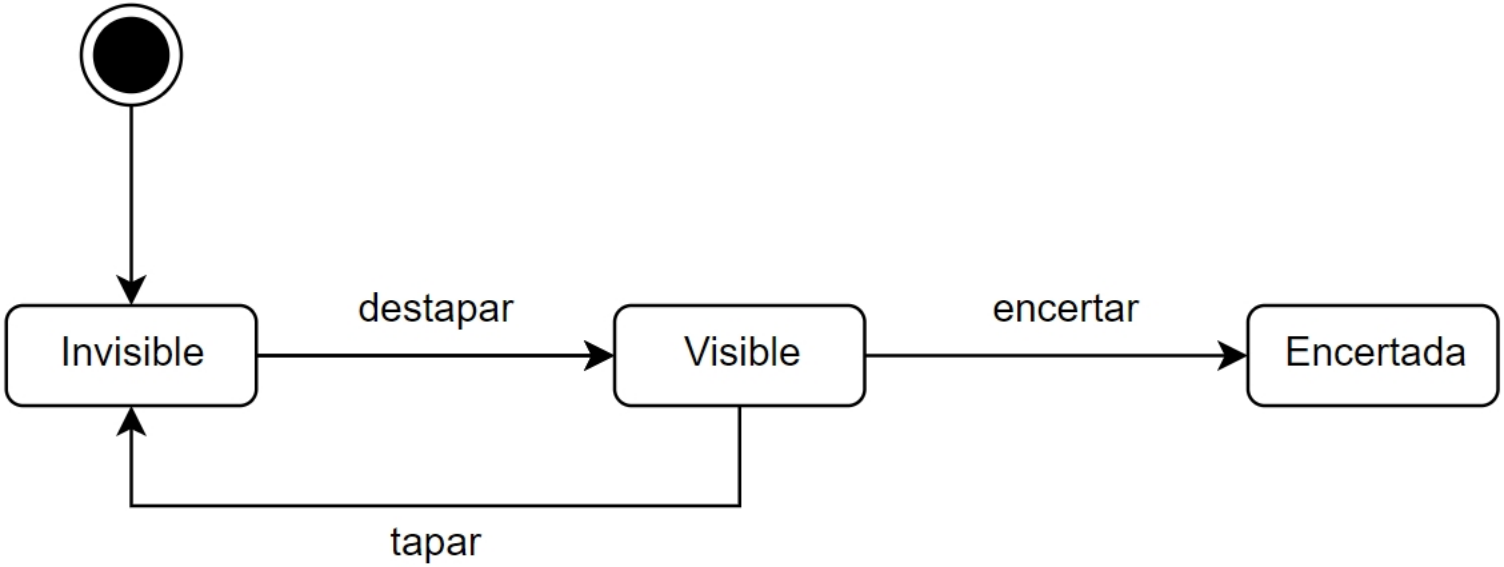
Invisible
destapar
Visible
encertar
Encertada
tapar

```java
package domini.estat;

import domini.Casella;

public interface IEstatCasella {
    // crearem 3 classes que implementin aquesta interfície
    // 3 estats (nodes) = 3 classes

    // mètode estàtic que determina l'estat inicial
    public static IEstatCasella getEstatInicial() {
        return new CasellaInvisible();
    }

    // mètode per sapiguer si podem mostrar el contingut
    public abstract boolean isVisible();

    // mètodes que corresponen a les transicions
    // 3 transicions (fletxes) = 3 mètodes
    public abstract IEstatCasella destapar();
    public abstract IEstatCasella tapar();
    public abstract IEstatCasella encertar();
}

```

```java
package domini.estat;

public class CasellaVisible implements IEstatCasella{

    @Override
    public boolean isVisible() {
        return true;
    }

    @Override
    public IEstatCasella destapar() {
        throw new RuntimeException("Error la casella ja està destapada.");
    }

    @Override
    public IEstatCasella tapar() {
        return new CasellaInvisible();
    }

    @Override
    public IEstatCasella encertar() {
        return new CasellaEncertada();
    }

}

```

```java
package domini.estat;

public class CasellaInvisible implements IEstatCasella {

    @Override
    public boolean isVisible() {
        return false;
    }

    @Override
    public IEstatCasella destapar() {
        return new CasellaVisible();
    }

    @Override
    public IEstatCasella tapar(){
        throw new RuntimeException("Error casella ja tapada.");
    }

    @Override
    public IEstatCasella encertar() {
        throw new RuntimeException("Error casella tapada no es pot encertar.");
    }

}

```

```java
package domini.estat;

public class CasellaEncertada implements IEstatCasella{

    @Override
    public boolean isVisible() {
        return true;
    }

    @Override
    public IEstatCasella destapar() {
        throw new RuntimeException("Error l'estat no es pot modificar.");
    }

    @Override
    public IEstatCasella tapar() {
        throw new RuntimeException("Error l'estat no es pot modificar.");
    }

    @Override
    public IEstatCasella encertar() {
        throw new RuntimeException("Error l'estat no es pot modificar.");
    }

}

```

```java
package domini;

import domini.estat.IEstatCasella;

public class Casella {
    private final char contingut;
    private IEstatCasella estat;

    public Casella( char contingut)
    {
        this.contingut = contingut;
        this.estat = IEstatCasella.getEstatInicial();
    }

    public char getContingut() {
        if(!estat.isVisible())
            return TaulerMemory.celdaBuida;
        return contingut;
    }

    public void setInvisible() {
        this.estat = this.estat.tapar();
    }

    public void setVisible() {
        this.estat = this.estat.destapar();
    }

    public void setEncertada() {
        this.estat = this.estat.encertar();
    }

    //afegit per fer comprovacions al testing
    public IEstatCasella getEstat() {
        return this.estat;
    }
}

```

```java
package test;

import static org.junit.jupiter.api.Assertions.*;

class CasellaTest {

    Casella casella;

    @BeforeEach
    void setUp() throws Exception {
        casella = new Casella('A');
    }

    @Test
    void testEstatInicial() {
        // Comprovem que l'estat inicial d'una casella sigui adequat.

        // l'estat inicial hauria de ser Invisible
        assertTrue(casella.getEstat() instanceof CasellaInvisible);
        assertFalse(casella.getEstat().isVisible());

        // no es pot posar com a ENCERTADA ni INVISIBLE
        assertThrows( RuntimeException.class, () ->{casella.setEncertada();});
        assertThrows( RuntimeException.class, () ->{casella.setInvisible();});
    }

    @Test
    void testCasellaVisible() {
        // Comprovem que una casella encertada no és pugui modificar el seu estat a cap altre estat.

        // primer posem la casella a estat visible
        casella.setVisible();
        assertTrue(casella.getEstat() instanceof CasellaVisible);
        assertTrue(casella.getEstat().isVisible());


        // ara comprovem que no es pot VISIBLE
        assertThrows( RuntimeException.class, () ->{casella.setVisible();});

    }

    @Test
    void testCasellaEncertada() {
        // Comprovem que una casella encertada no és pugui modificar el seu estat a cap altre estat.

        // primer posem la casella a estat visible
        casella.setVisible();
        assertTrue(casella.getEstat() instanceof CasellaVisible);
        assertTrue(casella.getEstat().isVisible());

        // després posem la casella a estat encertada
        casella.setEncertada();
        assertTrue(casella.getEstat() instanceof CasellaEncertada);
        assertTrue(casella.getEstat().isVisible());

        // ara comprovem que no es pot canviar l'estat
        assertThrows( RuntimeException.class, () ->{casella.setVisible();});
        assertThrows( RuntimeException.class, () ->{casella.setInvisible();});
        assertThrows( RuntimeException.class, () ->{casella.setEncertada();});

```

```
68      }
69
70      @Test
71      void testRecorregutCorrecte() {
72          // Comprovem que els altres estats es poden modificar tranquil·lament.
73
74          // primer posem la casella a estat visible
75          casella.setVisible();
76          assertTrue(casella.getEstat() instanceof CasellaVisible);
77          assertTrue(casella.getEstat().isVisible());
78
79          // ara posem la casella a estat invisible
80          casella.setInvisible();
81          assertTrue(casella.getEstat() instanceof CasellaInvisible);
82          assertFalse(casella.getEstat().isVisible());
83
84          // ara tornem a posar la casella a estat visible
85          casella.setVisible();
86          assertTrue(casella.getEstat() instanceof CasellaVisible);
87          assertTrue(casella.getEstat().isVisible());
88
89          // després posem la casella a estat encertada
90          casella.setEncertada();
91          assertTrue(casella.getEstat() instanceof CasellaEncertada);
92          assertTrue(casella.getEstat().isVisible());
93      }
94  }
95
```