**Etapa 1:**

En esta primera etapa se implementan las funciones que permiten el encendido y apagado de los botones al pedir el ascensor.

Entrada:

1 1 U

3 0 3

4 2 D

6 0 1

8 4 D

10 0 2

Salida:

1 0000 1000 0000

3 0010 0000 0000

4 0000 0000 0100

6 1000 0000 0000

8 0000 0000 0001

10 0100 0000 0000

Dificultades:

Solo el adaptarse al modelo empleado, comprendiendo el flujo de información y cómo mejorarlo.

**Etapa 2:**

En esta primera segunda etapa se implementan las funciones que permiten el funcionamiento de los sensores de piso, logrando detectar la cabina en su recorrido.

Entrada:

**while**(cabina.readFloorIndicator()< numPisos)

cabina.move(deltaHight);

**while**(cabina.readFloorIndicator()> 1)

cabina.move(-deltaHight);

**while**(cabina.readFloorIndicator()< numPisos)

cabina.move(deltaHight);

**while**(cabina.readFloorIndicator()> 1)

cabina.move(-deltaHight);

* Lo anterior implica subir al último piso y volver al primero, dos veces en total.

Salida:

2 01000000

2 01000000

2 01000000

2 01000000

2 01000000

3 00100000

3 00100000

3 00100000

3 00100000

3 00100000

4 00010000

4 00010000

4 00010000

4 00010000

4 00010000

5 00001000

5 00001000

5 00001000

5 00001000

5 00001000

6 00000100

6 00000100

6 00000100

6 00000100

6 00000100

7 00000010

7 00000010

7 00000010

7 00000010

7 00000010

8 00000001

7 00000010

7 00000010

7 00000010

7 00000010

7 00000010

6 00000100

6 00000100

6 00000100

6 00000100

6 00000100

5 00001000

5 00001000

5 00001000

5 00001000

5 00001000

4 00010000

4 00010000

4 00010000

4 00010000

4 00010000

3 00100000

3 00100000

3 00100000

3 00100000

3 00100000

2 01000000

2 01000000

2 01000000

2 01000000

2 01000000

1 10000000

2 01000000

2 01000000

2 01000000

2 01000000

2 01000000

3 00100000

3 00100000

3 00100000

3 00100000

3 00100000

4 00010000

4 00010000

4 00010000

4 00010000

4 00010000

5 00001000

5 00001000

5 00001000

5 00001000

5 00001000

6 00000100

6 00000100

6 00000100

6 00000100

6 00000100

7 00000010

7 00000010

7 00000010

7 00000010

7 00000010

8 00000001

7 00000010

7 00000010

7 00000010

7 00000010

7 00000010

6 00000100

6 00000100

6 00000100

6 00000100

6 00000100

5 00001000

5 00001000

5 00001000

5 00001000

5 00001000

4 00010000

4 00010000

4 00010000

4 00010000

4 00010000

3 00100000

3 00100000

3 00100000

3 00100000

3 00100000

2 01000000

2 01000000

2 01000000

2 01000000

2 01000000

1 10000000

Gráfico:

Dificultades:

-El uso del método isInRange en la clase Sensor permite un rango de acción que no necesariamente está proporcional al tamaño de la cabina, por tanto, era un esfuerzo extra el imaginar posibles errores de medición pese a que el comportamiento fuera el esperado.

**Etapa 3:**

En esta primera etapa se implementan la unidad de control, la cual permite la comunicación entre botoneras y sensores. Logrando así el comportamiento deseado, moviendo el ascensor en función de los eventos de entrada y los sensores de piso.

Entrada:

1 2 U

4 3 D

8 4 D

10 6 D

11 3 U

Salida:

piso EstadoM sensor UpReq DownReq

2 0 010000 00000- -00000

Pausing cabina .....

6 0 000001 00100- -01100

Pausing cabina .....

4 1 000100 00100- -01000

Pausing cabina .....

3 1 001000 00100- -00000

Pausing cabina .....

3 1 001000 00000- -00000

Pausing cabina .....

Gráfico:

Dificultades:

El comportamiento automatizado sin uso de colas, por medio de los sensores dificultó su implementación. Pero principalmente por una preconcepción al enfrentar el problema, al comprender el uso de las funciones entregadas se logró dar con la solución propuesta por el modelo.

**Etapa 4:**

En esta primera etapa se conecta la botonera de la cabina, permitiendo el comportamiento completo del ascensor. Este comportamiento reside principalmente en cambios en la unidad de control y el flujo de información entre esta última, la cabina y su botonera.

Entrada:

1 2 U

4 0 5

8 4 D

10 0 1

11 3 U

Salida:

Gráfico:

Dificultades:

El comportamiento asincrónico del motor junto con eventos en distinto tiempo hace difícil predecir el resultado, por tanto, había que probar distintos casos para ir probando la solución.