

**NOMBRE:** Rolando reyes martinez

**UNIDAD DE ESTUDIO:** construcción del software

**GUIA3**

**Educación virtual**

1. Empezamos como es costumbre creando la clase Ascensor.

```
package paquete;  
  
public class Ascensor {  
    private int noPisos;  
    private int noSotanos;  
    private int pisoActual;
```

2. Guardaremos el numero de pisos y de sótanos,junto con el piso actual en el que se se encuentra el mismo.
3. Procedemos a escribir los constructores

```
public Ascensor() {}  
  
public Ascensor(int noPisos, int noSotanos, int pisoActual) {  
    this.noPisos = noPisos;  
    this.noSotanos = noSotanos;  
    this.pisoActual = pisoActual;  
}
```

4. Procedemos a construir los métodos encargados del intercambio de datos.

```

public int getNoPisos() {
    return noPisos;
}

public void setNoPisos(int noPisos) {
    this.noPisos = noPisos;
}

public int getNoSotanos() {
    return noSotanos;
}

public void setNoSotanos(int noSotanos) {
    this.noSotanos = noSotanos;
}

public int getPisoActual() {
    return pisoActual;
}

public void setPisoActual(int pisoActual) {
    this.pisoActual = pisoActual;
}

```

5. La función “noPisosAtendidos” nos retornara la suma de los pisos y de los sótanos

```

public int noPisosAtendidos()
{
    return this.noPisos + this.noSotanos;
}

```

6. Como podremos ver en la siguiente gráfica creamos dos funciones miembro...
7. En una nos encargamos de subir el ascensor, cambiarlo así de posición actual y hacer las respectivas validaciones.
8. En la otra nos concentramos en hacer que el ascensor baje.

```

public void sube()
{
    if(this.pisoActual == this.noPisos){
        System.out.println("Ya estás en el piso mas alto");
    }else{
        if(this.pisoActual == -1){
            this.pisoActual = this.pisoActual+2;
            System.out.println("Subiendo");
        }else{
            this.pisoActual++;
            System.out.println("Subiendo");
        }
    }
}

public void baja()
{
    if(this.pisoActual == this.noSotanos*-1){
        System.out.println("Ya estás en el piso mas bajo");
    }else{
        if(this.pisoActual == 1){
            this.pisoActual = this.pisoActual-2;
        }else{
            this.pisoActual--;
        }
    }
}
}

```

9. Gracias a la función toString podemos ver todo el contenido del objeto.

```

public String toString() {
    return "Numero de pisos: " + noPisos + "\n"+
        "Numero de sotanos: " + noSotanos + "\n"+
        "Piso actual: " + pisoActual + "\n"+
        "Numero de pisos atendidos: "+this.noPisosAtendidos()+"\n";
}

```

10. Ya creada y bien estructurada nuestra clase procedemos a trabajar con ella desde Main.
11. Primero llamamos o importamos la clase ascensor.

```

package paquete;
import paquete.Ascensor;

```

Inicializamos una variable de tipo Ascensor en null, ya que luego daremos opción de crear el objeto... Pero aun no.

```
public class Main {

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

    public static void main(String[] args) {

        Ascensor ObjA = null;
        int opcion = 0;
```

12. Como podemos ver en la siguiente gráfica prácticamente obligamos al usuario a seleccionar la opción 1 en su menú para así ejecutar la función encargada de crear nuestro objeto ascensor.
13. Si el usuario digita 6 puede salir del programa.

```
while(ObjA == null && opcion !=1){
    System.out.println("Antes crea el objeto Ascensor");
    opcion = mostrarMenu();
    if(opcion == 6)
        System.exit(0);
}
```

14. En la siguiente imagen podemos ver la función menú, que muestra las opciones posibles de nuestro ascensor

```
public static int mostrarMenu()
{
    int opcion;

    System.out.println("1) Crea Ascensor");
    System.out.println("2) Numero de pisos atendidos");
    System.out.println("3) Piso actual");
    System.out.println("4) Subir");
    System.out.println("5) Bajar");
    System.out.println("6) Mostrar Ascensor");
    System.out.println("7) Salir");
    opcion = in.nextInt();

    return opcion;
}

}
```

15. En Main Class luego de que la función del menú nos retorna los datos se ubica el "switch" para hacer funcionar cada opción de nuestro menú.
16. Podemos ver que se ha implementado un doWhile para evitar que se salga sin dar en la opción salir.
17. Vemos al principio que si el objeto es null no me muestra el menú, ya que en la instrucción pasada mostramos e menú insistentemente hasta que el usuario escoja la opción 1 para crear el objeto. Al entrar al doWhile que vemos en la siguiente imagen no hay por que mostrarle de nuevo el menú.

```
do{
    if(ObjA != null)
        opcion = mostrarMenu();
    switch (opcion) {
        case 1:
            ObjA = crearAscensor();
            break;
        case 2:
            System.out.println("EL numero es: " + ObjA.noPisosAtendidos());
            break;
        case 3:
            System.out.println("EL piso actual es: " + ObjA.getPisoActual());
            break;
        case 4:
            ObjA.sube();
            break;
        case 5:
            ObjA.baja();
            break;
        case 6:
            ObjA.toString();
            break;
        case 7:
            System.out.println("Saliendo...");
            break;
        default:
            System.out.println("Digita un numero valido");
            break;
    }
}
```

18. Vemos que el objeto ya creado ejecuta los métodos que tiene dentro de su estructura. Métodos que ya repasamos.

```
        default:
            System.out.println("Dig
            break;
        }
    }while(opcion != 7);
```

19. Se mostrara el menú y se repetirá el proceso hasta que el usuario desee salir.

20. Por ultimo mostramos como se crea nuestro objeto.
21. La función "crearAscensor" es de tipo Ascensor para así retornar sin problema nuestro objeto.

```
public static Ascensor crearAscensor()
{
    int pisos, sotanos, pisoActual;

    System.out.println("Digita numero de pisos");
    pisos = in.nextInt();
    System.out.println("Digita numero de sotanos");
    sotanos = in.nextInt();
    do{
        System.out.println("Digita piso inicial");
        pisoActual = in.nextInt();
    }while(pisoActual > pisos || pisos<(sotanos*-1) || pisoActual == 0);
    Ascensor ObjA = new Ascensor(pisos, sotanos, pisoActual);
    System.out.println("¡Ascensor CREADO!");

    return ObjA;
}
```