

# Examen-lab-Mando-programable.pdf



**sofiisuares**



**Diseño del Software**



**3º Grado en Ingeniería Informática del Software**



**Escuela de Ingeniería Informática  
Universidad de Oviedo**

## Máster Online en Ciberseguridad

Nº1 en España según El Mundo



**Hasta el 46%  
de beca**



Mejor Máster  
según el  
Ranking de  
ELMUNDO

Para ser el mejor hay que aprender  
de los mejores.

**IMEF**

Smart Education

**Deloitte**


**Infórmate**

# Consigue Empleo o Prácticas

Matricúlate en IMF y accede sin coste a nuestro servicio de Desarrollo Profesional con más de 7.000 ofertas de empleo y prácticas al mes.



IMF  
Smart Education



Escuela de Ingeniería Informática  
Diseño del Software  
Lunes, 20 de enero de 2020

Universidad de Oviedo

## Examen práctico: mando programable

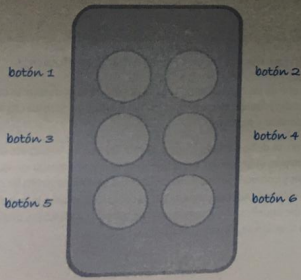
Alumno: \_\_\_\_\_

### Instrucciones

IMPORTANTE: una vez hayáis creado un nuevo espacio de trabajo e importado el proyecto inicial descargado del campus, no olvidéis **renombrarlo dentro de Eclipse** con el siguiente formato: **apellido1\_apellido2\_nombre**, en minúsculas y sin acentos ni ningún otro carácter especial. Por ejemplo, una alumna llamada **María Peláez Nuño** tendría que dar a su proyecto en Eclipse el siguiente nombre: **pelaez\_nuno\_maria**.

### Introducción

Se pretende realizar el software de un mando a distancia capaz de controlar diversos dispositivos. El mando cuenta con seis botones programables, cada uno de los cuales es a su vez una pequeña pantalla capaz de mostrar una imagen, de tal forma que, al programar el mando, además de poder realizar una u otra operación, cada botón mostrará también un icono representativo de dicha operación (en nuestro caso, al simular la interacción con el mando en modo texto, en vez de un icono se mostrará un texto descriptivo de qué hace cada uno).



botón 1      botón 2  
botón 3      botón 4  
botón 5      botón 6

### Código entregado

En el proyecto inicial que se os entrega podéis encontrar una implementación de prueba de un mando muy rudimentario, con cada botón codificado con la operación a realizar (es decir, no se puede programar). Igualmente, y en distintas carpetas de código fuente, los controladores de distintos dispositivos: un **televisor**, una **bombilla** y un aparato de **aire acondicionado**. Dichas clases son totalmente funcionales y, de hecho, son proporcionadas por los fabricantes de dichos dispositivos y, por tanto, no se pueden modificar.

### Qué se pide

Mejorar el diseño para hacer el **mando programable**

La idea es que o bien los fabricantes de cada dispositivo, o nosotros mismos, u otros desarrolladores, registren a través de una aplicación web las distintas operaciones que se pueden realizar sobre cada aparato a manejar. Posteriormente, una aplicación móvil desarrollada por nosotros se emparejará con el mando y se conectará con dicho servidor para descargar todas las operaciones disponibles. El usuario podrá de esa forma configurar cada botón con cualquiera de las operaciones disponibles. Dicha aplicación web cae fuera del ámbito de este proyecto, y será responsabilidad de otro equipo de desarrollo: nosotros, para probarlo, nos limitaremos a indicar la configuración del mando desde el programa principal.

En concreto, una vez rediseñado, se pide lo siguiente:

- **Hacer una prueba con la configuración inicial**  
Es decir, tras el rediseño, configurar el mando desde el main para que se comporte igual que el prototipo inicial (conf1g1).
- **Hacer una reasignación de botones**  
A continuación, comentar la línea anterior y ejecutar una nueva configuración (conf1g2) de modo que el botón uno encienda la luz, el tres cambie al canal 1 del televisor y los botones dos, cuatro y seis pongan el aire acondicionado en los modos OFF (apagado), LOW y HIGH, respectivamente; el botón cinco en esta configuración quedará sin asignar.

Poder asignar varias operaciones a un mismo botón

Como contamos con un número de botones muy reducido, ahora se decide dar al usuario la posibilidad de asignar a un botón distintas operaciones. En concreto, para probar dicha nueva funcionalidad, una vez hechos los cambios necesarios en el diseño, se pide que crees una configuración del mando (conf1g3) que sea igual que la segunda.

¿Quieres conocer todos los servicios?



WUOLAH

pero haciendo ahora que el botón cinco, al que llamaremos *Noche*, baje el volumen del televisor dos veces y apague el aire acondicionado y la luz.

### Ejemplo de interacción

A continuación se muestra cómo debería comportarse el mando una vez realizadas todas las tareas y con esta tercera configuración:

```
Botones:
1. Encender la luz
2. Apagar el aire acondicionado
3. Poner el canal 1 del televisor
4. Poner el aire acondicionado a baja potencia
5. Noche
6. Poner el aire acondicionado a alta potencia
```

Pulse uno de los botones del mando ([S] para salir):

```
> 1
>>> Se encendió la luz
```

```
Botones:
1. Encender la luz
2. Apagar el aire acondicionado
3. Poner el canal 1 del televisor
4. Poner el aire acondicionado a baja potencia
5. Noche
6. Poner el aire acondicionado a alta potencia
```

Pulse uno de los botones del mando ([S] para salir):

```
> 3
>>> Se cambió el televisor al canal 1
```

```
Botones:
1. Encender la luz
2. Apagar el aire acondicionado
3. Poner el canal 1 del televisor
4. Poner el aire acondicionado a baja potencia
5. Noche
6. Poner el aire acondicionado a alta potencia
```

Pulse uno de los botones del mando ([S] para salir):

```
> 6
>>> Aire acondicionado en modo HIGH
```

Botones:

1. Encender la luz
2. Apagar el aire acondicionado
3. Poner el canal 1 del televisor
4. Poner el aire acondicionado a baja potencia
5. Noche
6. Poner el aire acondicionado a alta potencia

Pulse uno de los botones del mando ([S] para salir):

```
> 5
>>> Se bajó el volumen del televisor: 2
>>> Se bajó el volumen del televisor: 1
>>> Aire acondicionado en modo OFF
>>> Se apagó la luz
```

Botones:

1. Encender la luz
- [...] (se omiten el resto por brevedad)

Pulse uno de los botones del mando ([S] para salir):

```
> 5
¡Adiós!
```

```
-----
Televisor apagado (Canal = 1, Volumen = 1)
Luz apagada
Aire acondicionado (OFF)
-----
```

### Identificar los patrones y cambios realizados

En cada una de las clases entregadas se deberá indicar, mediante comentarios en el propio código fuente, qué cambios se han realizado con respecto al código inicial proporcionado y para qué.

Así mismo, se deberá añadir al proyecto de Eclipse, en su carpeta raíz, un fichero de texto *Patrones utilizados.txt*, en el que se identifique correctamente el patrón o patrones utilizados y con qué participantes del mismo se corresponden las clases e interfaces de vuestra solución. Además, tenéis que hacer lo mismo para los métodos relevantes del patrón (de aquéllos que aparecen en la estructura del patrón, decir con qué métodos de vuestra solución se corresponden).