



La empresa YoTeLoLLevo quiere implementar una aplicación para almacenar sus envíos de paquetería. De un **envío** debe almacenar la siguiente información: **número de envío, nombre del remitente, dirección completa del remitente, código postal del remitente (formado por 5 dígitos y puede empezar por 0), nombre del destinatario, dirección completa del destinatario y código postal del destinatario (formado por 5 dígitos y puede empezar por 0)**. El número de envío será asignado por el programa de forma que al primer pedido se le asigna el 1, al segundo el 2 y así sucesivamente.

La empresa desea almacenar dos tipos de envíos distintos: **envíos ordinarios** y **envíos con seguimiento**. Los envíos con seguimiento almacenarán también un **código de seguimiento** que estará formado por las letras **PK** seguidas de 10 dígitos, los 5 primeros serán el número de envío (se completará con ceros a la izquierda hasta completar los 5 dígitos) y los 5 siguientes serán el código postal del destinatario. Así, por ejemplo, un código de seguimiento podría ser **PK0000128040**. Para generar el código los envíos con seguimiento implementarán una interfaz llamada **ISeguimiento** con dos métodos: **generarCodigo** que devolverá el código de seguimiento y **mostrarSeguimiento** que mostrará un mensaje por pantalla con el texto *“Mostrando seguimiento del envío PK0000128040”*, donde se mostrará el código de seguimiento que corresponda. También generará un fichero de texto con el mismo texto *“Mostrando seguimiento del envío PK0000128040”* y cuyo nombre será el código de seguimiento con la extensión **txt**.

Se implementarán únicamente los constructores con parámetros y los gets y sets que sean necesarios. Se implementarán métodos **toString** que devolverá un **String** con la información de un envío donde aparecerá toda la información incluido el número de seguimiento en el caso de los envíos con seguimiento.

Se pide:

- Implementar las clases descritas ajustándose al esquema diseñado en el apartado anterior. Se implementarán los métodos que se consideren necesarios y se valorará el correcto uso de los modificadores.
- Implementar un programa **Mensajería** que contenga un método *main*. Dicho programa creará un array de envíos cuyo tamaño se desconoce (un único array para todos ellos) y mostrará un menú con las siguientes opciones, cuya función se describe a continuación de cada una de ellas:
 1. **Nuevo envío**: si se elige esta opción se comprobará si hay espacio para nuevos envíos en el array y, si lo hay, pedirá al usuario todos los datos del envío y posteriormente a eso se preguntará si se desea que el envío se haga con seguimiento o no. Se añadirá el envío al array si se trata de un envío con seguimiento se mostrará el texto *“Se ha generado un envío con código de seguimiento PX0000128040”* sustituyendo el código por el que corresponda. Si no hay espacio en el array se mostrará un mensaje indicándolo.
 2. **Envío entregado**: se pedirá un número de envío y, si éste se encuentra en el array, lo elimina. Si no se encuentra muestra un mensaje indicándolo.
 3. **Listado de envíos**: mostrará un listado con la información de todos los envíos.
 4. **Mostrar seguimiento**: pedirá un código de seguimiento de un envío y si el envío se encuentra en el array, llamará a su método *mostrarSeguimiento*.
 5. **Salir**: el programa finalizará al seleccionar esta opción.

A no ser que se elija la opción salir después de cada acción se volverá a mostrar el menú. Cualquier decisión que se tome por no estar especificada en el enunciado y que no modifique ninguna de las especificaciones deberá ser explicada razonadamente.

Para evitar que al salir del programa se pierdan todos los datos se deberá guardar todos los envíos del array en un fichero binario antes de terminar la ejecución del programa.

Al arrancar el programa se comprobará si el fichero binario de los envíos existe, si es así, rellena el array con los datos del fichero

Se deberá controlar cualquier posible excepción que pudiera darse.

Para aprobar el examen será imprescindible que éste no tenga errores de compilación.

Criterios de calificación	
Organización de clases, métodos y uso de modificadores	2 puntos
Uso de ficheros binarios	1,5 puntos
Uso de ficheros de texto	1,5 puntos
Funcionamiento del menú	1 punto
Funcionamiento general del programa siguiendo las especificaciones dadas	3 puntos
Control de excepciones	1 punto