Ball Sort Puzzle

A continuación, se detallan los requisitos para la elaboración del proyecto de programación obligatorio de la asignatura Introducción a la programación para el curso 2020/2021.

El objetivo de este proyecto es diseñar, implementar en C++ y documentar una versión del juego **Ball Sort Puzzle** con ayuda de una serie de módulos ya escritos para manejar el entorno gráfico y la interacción.

¡IMPORTANTE!

Se deben cumplir todos los requisitos que se indican en este documento.

Este proyecto puede realizarse **en grupos de dos o individualmente**. La cantidad de trabajo que hay que realizar está pensada para dos estudiantes, pero, opcionalmente, se puede hacer de manera individual.

En el apartado correspondiente del aula virtual de la asignatura se puede encontrar toda la información sobre el proyecto, un ejecutable de prueba como modelo de lo que debe entregarse, un proyecto base con los archivos del entorno, una plantilla para la documentación y la rúbrica de evaluación. Además, se puede usar el foro de la asignatura para preguntar dudas y aclaraciones.

Es responsabilidad del estudiante la custodia y protección de su proyecto. **Solo las personas del grupo y los profesores de la asignatura pueden ver el código desarrollado para un proyecto**. ***Si ves el programa de otra persona, estás copiando. Si dejas tu proyecto a otra persona, estás copiando.***

Se utilizará un software de detección de copias en los programas entregados. En caso de encontrar similitudes en **partes significativas** de los programas, **todos los implicados** tendrán una nota de SUSPENSO (0) en la asignatura.

ENTREGA Y PRUEBA DE EVALUACIÓN

La entrega final del proyecto de programación se realizará mediante la actividad correspondiente del aula virtual.

El plazo de presentación se avisará con suficiente antelación. En cada convocatoria será siempre uno o dos días antes del día del examen oficial de la asignatura.

Cada miembro del grupo deberá entregar exactamente la misma información. Tanto en la documentación interna como en la externa deberán aparecer los nombres de las personas que han realizado el trabajo.

El día del examen final de la asignatura se realizarán las correspondientes pruebas de evaluación del proyecto, en el horario que se publicará en la convocatoria oficial. Esta prueba solo hay que hacerla si se ha entregado el proyecto en el plazo indicado.

La prueba se realizará de manera individual, independientemente de que el proyecto se haya realizado en pareja o individualmente.

Es necesario comprobar que se ha enviado la información correctamente y que está accesible en el aula virtual.

MATERIAL QUE SE DEBE ENTREGAR

Al entregar el proyecto se debe presentar un único fichero comprimido (.**zip** o .**tar.gz**) con el nombre de las personas que han hecho el programa, **nombre1_nombre2.zip** (o **.tar.gz**) que contenga los siguientes archivos:

- La documentación externa del proyecto en formato ODT, DOC o PDF.

  El directorio del proyecto comprimido con todos los archivos necesarios para poder compilarlo y ejecutarlo.

  Otros proyectos comprimidos con las ampliaciones (si se hacen).

  Nombre1 representa el nombre completo con los dos apellidos del primer componente del grupo (por orden alfabético) y nombre2 al del segundo. Por ejemplo, si el proyecto lo realizaran dos profesoras de la asignatura el fichero debería llamarse MariscalAraujoMAngeles_VicenteChicoteCristina.zip (o **.tar.gz**).

CALIFICACIÓN DEL PROYECTO

Un requisito previo a la evaluación del proyecto es la superación de la **prueba de evaluación**.

La prueba de evaluación del proyecto consistirá en unas preguntas sobre el proyecto y una modificación del funcionamiento básico. Las respuestas y la modificación se harán en papel, aunque se podrá consultar el código del proyecto en el ordenador.

Si no se supera la prueba de evaluación, la nota será 2.

Si es copia de otro proyecto, la nota será 0 en el bloque y en toda la asignatura para todos los implicados.

La calificación se hará en función del programa, de la documentación y de la prueba de evaluación del proyecto.

En los proyectos realizados de forma individual, la nota del bloque de proyecto se corresponderá con la nota de la evaluación del proyecto.

En los proyectos realizados en pareja, si ambos miembros del grupo superan la prueba de evaluación en la misma convocatoria y la calificación del proyecto es superior a 5, la **nota final del bloque del proyecto** se aumentará en 1 punto. En cualquier otro caso, se restará 1 punto a la calificación obtenida en el bloque del proyecto en la convocatoria en la que superen la prueba de evaluación

DESCRIPCIÓN DEL JUEGO *BALL SORT PUZZLE*

Ball Sort Puzzle es un juego para un solo jugador. Se pueden encontrar muchas versiones distintas para jugar en un navegador o en el móvil.

Por ejemplo, https://www.1001juegos.com/juego/bubble-sorting.

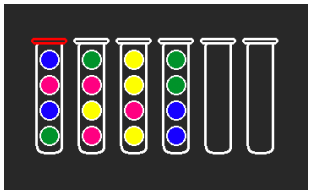
El juego se inicia con una serie de bolas de colores que llenan varias pilas, y algunas pilas vacías. El objetivo es agrupar las bolas del mismo color en cada pila.

El único movimiento permitido consiste en sacar la bola superior de una pila y ponerla en otra pila (que no esté llena) sobre una bola del mismo color (o en una pila vacía).

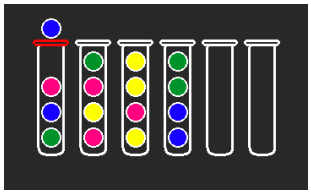
El número de colores distintos coincide con el número de pilas que están inicialmente llenas.

Cuantas más pilas vacías haya inicialmente, más fácil será conseguir el objetivo.

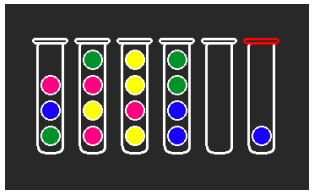
En el juego se van proponiendo configuraciones cada vez más complicadas de resolver.



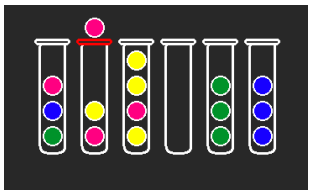
(1)



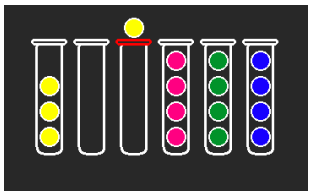
(2)



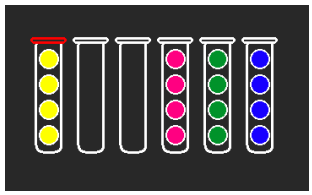
(3)



(4)



(5)



(6)

La figura 1 representa el estado inicial de una partida posible. Hay 4 pilas con bolas y 2 pilas vacías. Cada pila tiene espacio para 4 bolas. Inicialmente, las bolas están colocadas aleatoriamente.

Cada movimiento tiene dos fases: primero, se elige la pila de la que se quiere sacar la bola que está más arriba (figura 2) y, después, se suelta la bola en la pila de destino. Para que el movimiento sea correcto, la pila de destino debe estar vacía (figura 3) o, si tiene algún hueco libre, la bola que está en la cima debe ser del mismo color que la que se va a introducir. Si no se cumplen estas condiciones, la bola se queda en la pila donde estaba originalmente.

En la figura 4 se pueden ver un instante intermedio de la partida.

En la figura 5 y 6 se muestran las dos fases del último movimiento: se elige una bola de la tercera pila (figura 5) y se deja en la primera pila (figura 6). De esta forma, cada pila está llena de bolas del mismo color o está vacía, terminando el juego.

El juego también puede terminar si se abandona la partida por no conseguir el objetivo.

NUESTRA VERSIÓN DE *BALL SORT PUZZLE*

Versión básica

El programa servirá para una única partida.

Hay un fichero de configuración donde se determinarán las características iniciales de la partida. Su utilización se explica en el apartado correspondiente a la librería ***entorno***.

Se podrá configurar el número de pilas iniciales llenas y vacías. En total podrá haber entre 4 y 10 pilas. Siempre deberá haber un mínimo de una pila llena y una pila vacía para poder jugar.

El número de colores de las bolas es el número de pilas iniciales llenas.

También se puede configurar el número de bolas que caben en cada pila (un valor entre 3 y 6).

La configuración inicial se cargará con la información que haya en el fichero de configuración.

La persona que juegue utilizará las teclas de los cursores del teclado para mover la marca de pila seleccionada hacia la izquierda o hacia la derecha.

En la primera fase del movimiento, al pulsar la tecla **Enter** se seleccionará la bola de la cima de la pila donde esté la marca de selección. En la segunda fase del movimiento, cuando ya hay una bola seleccionada, si se pulsa Enter, se pondrá esa bola en la pila que estará marcada. Si no fuera un movimiento posible, la bola seleccionada se queda en su pila original, y se vuelve a la primera fase.

Al iniciarse el juego habrá una puntuación inicial. Con cada movimiento completo se restarán 10 puntos.

El juego terminará cuando las bolas del mismo color estén en la misma pila. También se acabará cuando la puntuación llegue a 0 o cuando se pulse la tecla **Escape**.

La nota máxima que se puede obtener en el proyecto con esta versión básica es de 8.

Ampliaciones

Existe la posibilidad de presentar, junto con el proyecto básico explicado anteriormente, versiones ampliadas.

Con estas ampliaciones se podrá llegar a obtener la calificación de 10 (personas que presenten individualmente el proyecto) o 12 (parejas), **siempre que el proyecto básico esté correctamente implementado y documentado.**

Si existen errores graves en la versión básica o en la documentación no se tendrán en cuenta las ampliaciones.

Ampliaciones propuestas para todos:

Se pueden incluir una, dos o tres propuestas de este bloque. Por tanto, se pueden conseguir 2 puntos como máximo.

- **Ayuda:** al pulsar la tecla F1 se añade una pila más vacía al final como ayuda. Solo se puede usar una vez. (Hasta 0.5 puntos.)
- **Animación**: se anima el movimiento de las bolas al pasar de una pila a otra y al caer. (Hasta 0.5 puntos.)
- **Deshacer:** al pulsar la tecla F2 se deshace el último movimiento completo. Se pueden deshacer los 5 últimos movimientos como máximo. Cada movimiento deshecho resta 10 puntos adicionales. (Hasta 1 punto.)

Ampliaciones propuestas solo para parejas:

Se pueden incluir una o dos propuestas de este bloque si se realizan también las ampliaciones propuestas para todos. Por tanto, se pueden conseguir 2 puntos como máximo.

- **Configuración aleatoria:** se añade la opción de generar aleatoriamente la situación inicial de las bolas de colores, según una opción del fichero de configuración. (Hasta 1 punto.)
- **Resolución automática:** el juego se ira resolviendo de forma automática sin intervención de ninguna persona. (Hasta 1 punto.)

No se admitirá ninguna ampliación distinta a las propuestas.

Las ampliaciones tienen que estar en un proyecto diferente al de la versión básica del juego. También debe incluirse en la documentación externa un pequeño resumen de lo que se ha cambiado con respecto a la original.

¡Atención! Algunas de estas ampliaciones pueden requerir una inversión considerable de tiempo por lo que cada estudiante debe sopesar si abordarlas o no, considerando su carga de trabajo en ésta y otras asignaturas.

NORMAS DE IMPLEMENTACIÓN

El proyecto debe implementarse usando exclusivamente programación imperativa con C++, usando las estructuras de datos y librerías que se han visto en clase.

El proyecto debe ejecutarse correctamente en la máquina virtual utilizada en la asignatura y en los ordenadores instalados en los laboratorios.

Se debe usar la librería *entorno* entregada por los profesores para gestionar el entorno del juego **sin ninguna modificación**. Para que esta librería funcione, es necesario tener instalada la librería ***allegro5***. (En la máquina virtual y en los laboratorios ya está instalada.)

Se deben definir, como mínimo, un tipo abstracto de datos para representar una pila, otro para el tablero completo (conjunto de pilas) y otro para representar el juego.

Las bolas del juego se pueden definir también con un TAD Bola, aunque no es necesario, ya que basta con un entero para representarlas y no hay operaciones especiales que hacer con ellas.

Cada TAD o librería debe estar definido usando un fichero .h y uno .cpp.

Cada librería o TAD utilizado debe venir acompañado de la implementación de la correspondiente librería con un **juego de pruebas completo.** No es necesario hacer juegos de prueba de la librería *entorno* ni del *TAD juego*.

En el aula virtual hay un proyecto de ***eclipse*** comprimido, ***BallSortPuzzleBase.zip***, con los archivos del entorno (entorno.h y entorno.cpp) y un programa simple de ejemplo que puede usarse como punto de partida para desarrollar el proyecto. Este proyecto está configurado ya con la información sobre las librerías necesarias.

En el aula virtual también hay un programa ejecutable del juego para tener una idea del aspecto final del proyecto.

TIPOS ABSTRACTOS DE DATOS

Tras un análisis del problema, surge la necesidad de modelar los siguientes elementos que se manejan con las librerías o tipos abstractos correspondientes:

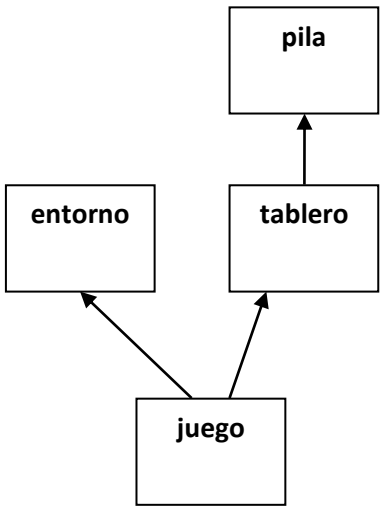
El entorno gráfico de la aplicación, responsable de la interacción con el usuario. La librería **entorno** es la que se encargará de gestionar la interfaz de usuario.

El TAD Pila gestionará una **pila** con las operaciones correspondientes.

El **tablero** está compuesto por un conjunto más o menos grande de pilas, según lo que se indique en el archivo de configuración. El TAD Tablero gestionará esa información con las correspondientes operaciones para iniciarlo a partir del fichero de configuración, determinar si una pila concreta está vacía o llena, poner una bola de un color en una pila concreta, sacar una bola de la cima de una pila concreta, etc.

El TAD **Juego** realiza el control de todo el proceso, encargándose de gestionar las teclas que se pulsan, los movimientos del cursor y el fin del juego, así como de mantener la coherencia entre la información que se almacena en el tablero y lo que se representa en pantalla en cada instante del juego.

Este es el esquema de relación entre los tipos abstractos de datos que se pueden definir y utilizar, y la librería principal del entorno que debe usarse para programar la interfaz del juego. Si se considera necesario, se pueden implementar más tipos. Este esquema es una simple recomendación.



LIBRERÍA ENTORNO

Con el fin de facilitar la labor de desarrollo se entrega, en un proyecto de ***eclipse***, una librería con los módulos necesarios para gestionar el entorno gráfico. Desde los módulos desarrollados por el programador solo se deben usar las operaciones definidas en ***entorno.h***. Estos módulos no pueden modificarse.

(Si alguna persona tiene dificultad para distinguir los colores propuestos, que contacte con el profesorado de la asignatura para que le pasemos una versión modificada del entorno.)

En el aula virtual hay un fichero ***BallSortPuzzleBase.zip*** con un proyecto base con los ficheros del entorno y un pequeño ejemplo de funcionamiento (no se usan todas las operaciones del entorno), que debe usarse como proyecto base para añadir el resto de módulos.

En ***entorno.h*** aparecen las constantes definidas y las funciones de manejo del entorno (con su especificación con pre/postcondiciones). Podemos suponer que todas las funciones del entorno tienen un coste constante, O(1).

También se define el tipo enumerado con los valores que devuelve la función **entornoLeerTecla**, con los valores de las únicas teclas aceptadas (cursor izquierda y derecha, Enter, F1, F2 y Escape). Si se pulsa cualquier otra tecla, la función devuelve TNada.

El juego se configurará a partir de la información del fichero ***ballSort.cnf*** que debe estar situado en el directorio del proyecto.

La estructura de este fichero de texto, que nos permite cargar la configuración inicial del juego, es la siguiente:

En la primera línea, el número de pilas llenas de bolas, **n,** que será un entero positivo.

En la segunda línea, el número de pilas vacías, **m**, que será un entero positivo.

La suma de **n** y **m** deberá estar siempre entre 4 y 10.

En la tercera línea, el número de bolas que caben en cada pila, **c**, que será un valor entre **3** y **6**.

En la cuarta línea, un valor 0 o 1. Si es 0, la configuración inicial de las bolas en las pilas se pondrá según la información que hay en este fichero de configuración. Si es 1, la configuración inicial de las bolas se hará aleatoriamente. *Este valor solo se debe tener en cuenta si se hace la ampliación* ***Configuración aleatoria****.*

En la quinta línea, un valor entero positivo, **p**, que indica la puntuación inicial de la partida.

En las siguientes líneas estará la información de la configuración inicial de las bolas en las pilas. (Si en la cuarta línea hay un 1 y se ha implementado la *ampliación* ***Configuración aleatoria***, no hacen falta más líneas en el fichero de configuración.)

Tiene que haber **c** filas con **n** valores separados por un espacio en blanco. Cada valor será un entero positivo entre 1 y **n**.

La primera fila tendrá la información del color de la fila superior (cima) de las pilas llenas; la segunda fila tendrá la información de la segunda fila de bolas, y así sucesivamente. La fila **n** tendrá los colores de la última fila (fondo) de las pilas inicialmente llenas de información.

Visto de otra forma, cada pila está representada en una columna de esa matriz de información.

Se puede suponer que el fichero de configuración es correcto. Si es incorrecto, el comportamiento de la aplicación no tiene que ser correcto.

Hay una operación en la librería entorno, ***entornoCargarConfiguracion***, que devuelve toda la información que hay en el fichero de configuración.

La matriz de enteros que se devuelve, con la información de la configuración inicial de las bolas, es la **traspuesta** de la información que hay en el fichero. Por ejemplo: la primera columna del fichero de configuración, que representa la información que hay en la primera pila, es la primera fila de la matriz **m** que devuelve ***entornoCargarConfiguración***.

Por tanto, en la matriz **m**, cada fila representa una de las pilas que se van a utilizar. La cima estará en la columna 0.

TAD PILA

El TAD Pila gestionará la información de una pila del tablero.

La estructura de datos para almacenar una pila debe guardar:

- El número de bolas que caben en la pila (variable entre 3 y 6).
- Espacio suficiente para almacenar el número máximo de bolas en la pila.
- El número de bolas en cada momento.

Además de las operaciones habituales del TAD Pila (iniciar, apilar, desapilar, cima, estaLlena, estaVacia), se podrían necesitar otras operaciones para poder facilitar el uso en el juego y su representación gráfica (por ejemplo, obtener el número de bolas que hay en la pila o saber si todas las bolas de la pila son del mismo color).

Hay que tener en cuenta que, en cada partida, según la información que se lee del fichero de configuración, la capacidad máxima de cada pila se fijará entre 3 y 6 (ambos incluidos). Por tanto, al iniciar la pila habrá que proporcionar esa información.

TAD TABLERO

El TAD Tablero es el que va a gestionar la información de todas las pilas que se están usando. En el tablero no se gestionan las teclas que se pulsan durante el juego, ni se actualiza la pantalla, sino que se definen las operaciones necesarias para modificar y obtener la información de las pilas.

La estructura de datos correspondiente al tablero debe guardar:

- Espacio suficiente para almacenar el número máximo de pilas.
- El número de pilas con el que se va a jugar.
- El número de pilas inicialmente llenas y vacías. (El número inicial de pilas llenas determina el número de colores distintos.)

Algunas de las operaciones que pueden ser útiles son las siguientes (pueden ser necesarias algunas más, o se pueden agrupar o dividir en otras similares; no siempre se indican todos los parámetros que deben pasarse):

- Iniciar el tablero con los datos necesarios, provenientes del fichero de configuración.
- Devolver el número de pilas del tablero.
- Todas las operaciones que se han implementado en el TAD Pila, pero referidas a una Pila concreta del tablero.
- Determinar si se puede colocar una bola de un color en una pila determinada o no.
- Determinar si todas las pilas están llenas con bolas del mismo color o vacías (es decir, si se ha resuelto el puzzle).

En el TAD Tablero se gestionará toda la información de los valores almacenados en la memoria. Los cambios que se produzcan en la pantalla se realizarán desde el gestor del juego, mediante las operaciones de la librería Entorno.

TAD JUEGO

El TAD Juego es el que realiza la gestión del juego completo. Será el que gestione el tablero, la interacción con el usuario a través del teclado y la actualización del entorno gráfico del juego (la pantalla).

La estructura de datos correspondiente al juego debe guardar, como mínimo:

- El tablero.
- Los datos de configuración del juego que no se almacenen en el tablero o en las pilas.
- Otros datos necesarios para las ampliaciones.

Como mínimo, deberá tener tres módulos:

- un módulo **iniciar** que inicie la estructura de datos del juego, según la configuración del fichero ***ballSort.cnf***
- un módulo **jugar** que realice la gestión general del juego (gestionar las teclas que se pulsen, actualizar el tablero y la pantalla, dar el juego por finalizado, etc.)
- un módulo **terminar** que termine el juego, mostrando un mensaje de despedida y cerrando el entorno gráfico.

PLANIFICACIÓN

El trabajo de implementación y documentación de este proyecto está planificado en menos de 50 horas de dedicación entre dos estudiantes que hayan realizado un seguimiento correcto de las clases, sesiones de laboratorio y actividades propuestas hasta el momento.

A continuación, se presenta una posible planificación, con las horas de dedicación y las principales tareas que hay que desarrollar. (Están detalladas por horas de trabajo individual; algunas serán conjuntas y otras por separado. Por ejemplo, la definición del tablero podría realizarse en una reunión de una hora de duración.)

Hay que ir anotando los tiempos dedicados a cada una de las tareas para incluirlos posteriormente en la documentación externa del proyecto.

Horas	Tarea
4	Lectura de la documentación inicial, planificación del trabajo en grupo (2 horas cada persona)
2	Prueba del proyecto base
2	Diseño general de la aplicación y de los TAD necesarios
4	Diseño e implementación del TAD Pila (estructura de datos, implementación de operaciones y pruebas)
2	Definición del TAD tablero (estructura de datos y operaciones necesarias)
10	Implementación del TAD tablero (juegos de pruebas y operaciones)
12	Definición e implementación del TAD juego (estructura de datos y operaciones)
4	Pruebas de integración del proyecto
6	Ampliaciones del proyecto
4	Escritura final de la documentación (una parte de esta tarea se debe realizar a lo largo de todo el proceso, mientras se implementan los TAD)

Total: 50 horas

DOCUMENTACIÓN

El proyecto debe ir acompañado de su correspondiente documentación interna y externa.

DOCUMENTACIÓN INTERNA

En el fichero .h de cada TAD se debe incluir la especificación con pre/postcondiciones y la complejidad de cada operación.
En el fichero .h de las librerías de prueba de cada TAD deben incluir, con comentarios, el diseño de las pruebas que se hacen en cada módulo.

DOCUMENTACIÓN EXTERNA

En el aula virtual se puede encontrar una plantilla que sirve de guía para la redacción de la documentación externa, con la información que debe contener.
El formato de la página de la cubierta debe ser el indicado en la plantilla de la documentación, que incluye la identificación de los estudiantes, el grupo al que pertenecen y su profesor de laboratorio.
A continuación, se explican brevemente los distintos apartados que, como mínimo, debe contener la documentación externa entregada.

Introducción

Toda documentación de programas debe incluir una breve introducción sobre el software desarrollado, explicando el objetivo principal y los requisitos satisfechos.

Análisis y diseño

Una vez que conocemos cuál es el problema en cuestión, descrito en la introducción, debemos comprender el problema y analizar qué entidades intervienen en el mismo. Además, será necesario asociar a cada una de estas entidades las principales acciones que pueden realizar.
La documentación relativa al análisis facilita la comprensión del problema, ya que, sin entrar en demasiados detalles, se describen los tipos abstractos de datos que posteriormente serán implementados y las principales decisiones tomadas.
En el diseño de la aplicación se detalla cada uno de los tipos abstractos de datos utilizados para desarrollar la aplicación. Para cada uno de los tipos abstractos se debe explicar su composición y las operaciones relacionadas.
Esta sección se debe iniciar con el esquema de los tipos abstractos usados (similar al que aparece en la página 4), pasando después a describir cada uno de los tipos de abstractos de datos y el programa principal.

Planificación y tareas

En este apartado de la documentación se debe incluir la planificación inicial del proyecto, las principales tareas realizadas en el desarrollo del programa, quién las ha realizado y el tiempo empleado en cada una de ellas.
Es fundamental incluir entre las tareas, las reuniones mantenidas, el tiempo empleado en cada una de ellas y los principales acuerdos alcanzados.

Conclusiones y principales problemas

En este apartado de la documentación se deben incluir las principales conclusiones extraídas por los autores del trabajo. Además, se debe reflexionar sobre los problemas encontrados a la hora de desarrollar la aplicación y lo que se ha aprendido.
Los listados con el código no deben incluirse en la documentación externa.