

**Objetivo:**

- Diseñar una implementación lineal y en memoria dinámica en C++ del TAD genérico “colecciónITF” y usarlo para implementar un programa que gestione una colección de programas informáticos.

**Fecha límite de entrega:** 10-11-2021 (incluido)

**Descripción detallada:**

Se trata de replicar lo realizado en el Ejercicio 2 de la Práctica 0, pero desarrollando una **implementación dinámica** (utilizando una lista enlazada) del TAD genérico “colecciónITF” (definido en la Tarea 1 del Ejercicio 2 de la Práctica 0), y utilizarlo para gestionar una colección de programas informáticos (exactamente igual que se hizo en las Tareas 2 y 3 del Ejercicio 2 de la Práctica 0). Al igual que en la Práctica 0, el iterador deberá permitir recorrer todos los ítems de la colección **por orden lexicográfico ascendente, de su identificador.**

El código fuente del programa de prueba (*main*) deberá encontrarse en un fichero llamado “*practical.cpp*” y cumplir escrupulosamente con el funcionamiento y formatos que se describieron en la tarea 3 del Ejercicio 2 de la Práctica 0, o la práctica no será evaluada.

**Observaciones.**

- El código fuente entregado será compilado y probado en *hendrix*, que es donde deberá funcionar correctamente.
- El código fuente entregado deberá compilar correctamente con la opción `-std=c++11` activada.
  - Esto significa que, si se trabaja con la línea de comandos, deberá compilarse con:  
`g++ -std=c++11 ficheros_compilar...`
  - Si se trabaja con algún entorno de programación (*Visual Studio Code*, *Code::Blocks*, *CodeLite*, etc) y no se utiliza la línea de comandos para compilar, el proyecto de la práctica deberá estar configurado para compilar con la opción `-std=c++11`.
- Todos los ficheros con código fuente que se presenten como solución de esta práctica deberán estar **correctamente documentados**.
- En el **comentario inicial de cada fichero** de código fuente se añadirán los **nombres completos y NIA de los autores** de la práctica.
- Los TADs deberán implementarse siguiendo las instrucciones dadas en las clases y prácticas de la asignatura, y no se permite utilizar Programación Orientada a Objetos.
- No se permite usar las clases o componentes de la *Standard Template Library (STL)*, ni similares.
- **Todas las indicaciones que se dan en los enunciados de las prácticas respecto a nombres de ficheros, programas, opciones del programa, formatos de los ficheros de entrada o de los ficheros de salida que deban generarse, etc., deben cumplirse escrupulosamente para que la práctica sea evaluada.**
- El código fuente del programa de prueba (*main*) deberá encontrarse en un fichero llamado “*practical.cpp*” y cumplir escrupulosamente con el funcionamiento y formatos que se han descrito en el enunciado.
- La ruta del fichero de entrada deberá ser “*entrada.txt*” (no “*entrada1.txt*”, “*datos/entrada.txt*”, ni similares). Ídem para el fichero “*salida.txt*”.
- La salida debe seguir las especificaciones del enunciado. Por ejemplo, cuando escribimos “NO INTRODUCIDO: ” en el fichero de salida, está escrito en mayúsculas y con un espacio en blanco tras ‘:’; lo mismo para el resto de instrucciones.

**Material a entregar. Instrucciones.**

- La práctica solo deberá someterla uno de los miembros del equipo de prácticas desde su cuenta de *hendrix*, y preferiblemente siempre el mismo para todas las prácticas.
- Conectarse a `hendrix-ssh.cps.unizar.es` según se explica en el documento “Realización y entrega de prácticas en los laboratorios del DIIS” disponible en moodle.
- Crear un directorio llamado *C\_p1* si tu profesor tutor de prácticas es Javier Campos, o *V\_p1* si tu profesora tutora de prácticas es Yolanda Villate, donde se guardará un directorio *practical* que contenga todos los ficheros desarrollados para resolver la práctica (este directorio, *practical*, deberá contener todos los ficheros con código fuente C++ necesarios para resolver la práctica y dos ficheros de texto *entrada.txt* y *salida.txt*, con los formatos explicados, pero que sean significativamente diferentes a los proporcionados como ejemplo, y con los que habréis probado la implementación realizada en vuestra práctica). A la hora de evaluar la práctica se utilizará tanto el fichero de prueba que se entregue, como ficheros de prueba entregados por otros compañeros, o ficheros propios de los profesores.
- Crear el fichero *X\_p1.tar*, con X igual a C o V, dependiendo de quién sea tu profesor tutor de prácticas, con el contenido del directorio *X\_p1* ejecutando la orden:

```
tar -cvf X_p1.tar X_p1
```

- Enviar el fichero X\_pl.tar, con X igual a C o V, dependiendo de quién sea tu profesor tutor de prácticas, mediante la orden:

```
someter -v eda_21 X_pl.tar
```

**ADVERTENCIA:** la orden `someter` no permite someter un fichero si el mismo usuario ha sometido antes otro fichero con el mismo nombre y para la misma asignatura, por lo tanto, **antes de someter vuestra práctica, aseguraos de que se trata de la versión definitiva que queréis presentar para su evaluación.**