

# PY03: Proyecto - Librería para Construcción de Archivos XML

---

## Código: PY03

### Objetivo

Construir una librería de funciones que se pueda agregar a otros programas para confeccionar estructuras y archivos basados en XML bien formados.

### Implementación

La librería debe construirse con los siguientes lineamientos:

Se creará una clase llamada `XMLDocument`, la cual será un controlador de la creación de las estructuras y los archivos XML, que podrá ser incluida en cualquier otro programa por medio de un archivo de encabezado. Esta debe colocarse en un archivo llamado `XMLDocument.hpp`

La librería no debe hacer ningún tipo de impresión en pantalla, ni debe solicitar información de teclado. Únicamente debe trabajar con los parámetros de las funciones públicas de la clase.

Esta librería debe permitir realizar las siguientes acciones:

- Insertar un nuevo nodo raíz
- Seleccionar un nodo del documento
- Agregar un nodo hijo al nodo seleccionado
- Editar atributos o contenido de un nodo actual
- Modificar el modo de visualización del contenido del nodo seleccionado
- Configurar el nodo seleccionado como raíz de un subárbol balanceado biselado (display)
- Eliminar el nodo seleccionado
- Visualizar subárbol
- Visualizar nodo seleccionado
- Previsualizar XML
- Guardar el árbol en un archivo

En todos los casos que no se especifica un valor de retorno, se utilizan valores booleanos para indicar el éxito o el fallo en el proceso solicitado. Si los retornos están especificados, deben retornarse valores que se puedan reconocer como error en caso que sucedan circunstancias que eviten el proceso solicitado.

## Modos de edición de atributos o contenido

Las operaciones de edición se pueden ejecutar en uno de cuatro modos: agregar, adjuntar, sobrescribir o borrar. Agregar y adjuntar mantienen el texto previo pero agregan el texto recibido por parámetro al inicio o final, respectivamente. Sobrescribir reemplaza el texto anterior con el nuevo. Borrar elimina todas las instancias del texto que se recibe por parámetro.

## Pruebas

Puede incluir un archivo principal cpp con algunas llamadas de funciones dentro de la librería o que pueda manipular llamadas a esta por medio de un menú, teniendo en cuenta que esto no es parte del programa por construir y que no será parte de la evaluación del proyecto.

## Plantilla de XMLDocument.hpp

Se debe respetar la siguiente plantilla general de los contenidos públicos que deben encontrarse en el archivo XMLDocument.hpp:

```
class XMLDocument {
    const int ADD = 100;
    const int OVR = 200;
    const int APD = 300;
    const int DEL = 400;
    const int BEG = 10;
    const int END = 20;

    int Root (char *name);
    bool Select (int id);
    int Child (char *name);
    int Child (char *name, char *content);
    bool EditAttribute (char *key, char *value, int mode);
    bool EditContent (char *content, int mode);
    bool SwitchContentPosition ();
    bool SetBalanced ();
    bool Delete ();
    char* ViewTree ();
    char* ViewNode ();
    char* ViewXML ();
    bool Save (char *filename);
};
```

---

## Tecnología

El código fuente debe poder utilizarse con g++ en Debian 10 o similares. Deben implementarse estructuras de datos lineales dinámicas. No se permiten arreglos estáticos

sin aprobación del profesor. No está permitido usar librerías externas no vistas en clase, incluyendo string.h.

### **Grupos de trabajo**

Debe desarrollarse en grupos de 2 o 3 estudiantes. No se evaluarán entregas de otros tamaños sin justificación. Los grupos deben registrarse a más tardar el 29 de abril en el foro Proyecto 03 - Foro.

### **Datos administrativos**

Este proyecto tiene un valor de 15%. La fecha límite es el jueves 18 de mayo a las 5:00 PM. Entregas tardías pierden 2 puntos porcentuales por cada día de atraso. Debe usarse el repositorio <https://git.mora.tk/>. Solo se considerarán los commits anteriores a la fecha límite, salvo indicación expresa del grupo para revisión tardía.

### **Calificación**

Se usará la rúbrica oficial de evaluación. No habrá citas de revisión. El código y la documentación deben poder entenderse y funcionar sin ayuda externa.

### **Recomendaciones**

Se recomienda crear una librería propia para manejar cadenas de texto, ya que no se pueden usar librerías estándar. En ese caso se permite usar arreglos estáticos sin aprobación previa.

¡Muchos éxitos con el trabajo!