

PY01 - Estructura de Datos

Objetivo

Desarrollar una clase en el lenguaje C++ que permita el manejo de cadenas de caracteres.

Implementación

Esta librería debe desarrollarse utilizando:

- **Un archivo de encabezado (String.hpp) donde se definan las funciones y métodos requeridos.**
- **Un archivo de implementación (String.cpp) donde se desarrollen dichas funciones.**

La clase String deberá implementar las siguientes funcionalidades:

Funcionalidades requeridas

Funcionalidad	Descripción
Constructor	Inicializa una instancia con una cadena (char*).
Destructor	Libera la memoria utilizada por la instancia.
caracterEn()	Retorna el carácter en un índice dado; si es inválido, retorna el carácter nulo.
contarCaracter()	Retorna la cantidad de ocurrencias de un carácter específico.
ultimoIndice()	Devuelve el índice de la última ocurrencia de un carácter.
cambiarCadena()	Cambia la cadena actual por una nueva (char*).
len()	Retorna la longitud de la cadena.
Operador equals (==)	Permite comparar objetos String.
split()	Retorna un string** separando la cadena por un carácter delimitador.
concatenarEn()	Inserta una cadena en una posición dada. Si el índice

Funcionalidad	Descripción
	es inválido, se normaliza.
<code>concatenar()</code>	Agrega una cadena al final del objeto actual.
<code>concatenarCadenas()</code>	Concatena múltiples cadenas (<code>char**</code>).
<code>reemplazarEn()</code>	Inserta una cadena en un índice dado. Si sobrepasa el tamaño, se agregan los caracteres extra.
<code>reemplazarOcurrencias()</code>	Reemplaza todas las ocurrencias de una subcadena por otra.
<code>guardarEnArchivo()</code>	Guarda la cadena en un archivo (modo de escritura configurable).
<code>leerArchivo()</code>	Lee una cadena desde un archivo y actualiza el contenido de la clase.

Nota: Se deben seguir los estándares de codificación provistos al final de esta especificación. Respeta los tipos de parámetros y valores de retorno indicados.

Tecnología

- El código debe compilarse con Borland C++ 10.2 bajo Windows 10/11.
 - Solo se permiten estructuras dinámicas lineales creadas por el equipo.
 - El uso de arreglos estáticos requiere autorización del profesor.
 - Queda prohibido el uso de la librería `string.h` y cualquier otra no personal.
 - Solo se permiten librerías vistas en clase.
-

Grupos de trabajo

- Los equipos deben ser de 2 o 3 estudiantes.
- No se aceptarán otros tamaños de grupo sin aprobación previa.
- Los grupos deben registrarse en el Foro del Proyecto 01.
- Si no formas parte de un grupo, serás asignado automáticamente. Si no recibes notificación, contacta al profesor.

Datos administrativos

- **Valor:** 15% de la nota del semestre.
- **Fecha de entrega:** Miércoles 23 de agosto, a más tardar 1:00 PM (hora del servidor).
- **Penalización:** Por cada día de retraso, se descontarán 2 puntos por día.
- **Entrega:** Subir el enlace al repositorio del proyecto grupal en la plataforma del curso.
- **Solo se considerarán válidos los *commits* realizados antes de la fecha límite, salvo indicación de entrega tardía.**

Evaluación

- **Se aplicará la rúbrica de evaluación publicada.**
- **El código debe ser autoexplicativo y funcional sin intervención del autor.**
- **El estándar de codificación será evaluado como parte importante de la nota.**

Recomendaciones

- **Comunicación:** Pregunta, comparte ideas y consulta dudas en el foro o al profesor.
- **Tiempo:** Comienza con anticipación para evitar problemas de última hora.
- **Código limpio:** Mantén un código organizado y comprensible.
- **Trabajo colaborativo:** Distribuyan tareas, discutan soluciones y usen el repositorio de forma activa.
- **Práctica:** Mejora tus habilidades programando de forma continua.

Tip: Usa el repositorio como respaldo, espacio de colaboración y para registrar tu progreso.