

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 1

**INSTRUCCIONES PARA LA CREACIÓN Y USO DE BIBLIOTECAS ESTÁTICAS**

**Indicaciones Generales:**

El presente documento tiene por objetivo presentar a los alumnos del curso, las pautas necesarias para crear una librería estática y para que la utilice en un proyecto.

---

**Conceptos generales**

Una biblioteca es una agrupación de datos, estructuras de datos y funcionalidades organizados de forma que sean reutilizables en más de un proyecto.

Adicionalmente a las bibliotecas estándar de C/C++, y a las bibliotecas provistas con la distribución de C/C++ que tengamos, usted puede tener la necesidad de crear sus propias bibliotecas para usarlas en los proyectos que desarrolle. También puede desear que otros desarrolladores utilicen sus bibliotecas, pero protegiendo su propiedad intelectual. En ese sentido una biblioteca estática, a diferencia de una común, estará compilada, es decir que usted no entregará el código fuente sino que entregará el código en lenguaje máquina.

Existen dos tipos de bibliotecas, las bibliotecas estáticas (en C/C++ comúnmente archivos con extensiones ".lib" y ".a" y las bibliotecas dinámicas (comúnmente con extensión ".dll", acrónimo de dynamic linked libraries). Las primeras son ensambladas o compiladas y se enlazan junto con el programa que las usa, y por tanto forman parte del archivo ejecutable que se crea. Las segundas no, y el enlace ocurre durante la ejecución del programa que utiliza la biblioteca. No todos los lenguajes de programación permiten la creación y el uso de ambos tipos de bibliotecas. C y C++ permiten crear y usar ambos tipos mientras que Java y C# solo trabajan con bibliotecas dinámicas.

Los archivos ".a" (o ".lib") son parecidos a los archivos ".o" (o ".obj") con una diferencia sustancial, cuando se enlazan un archivo ".o" en un proyecto, se añaden a este último absolutamente todas las funciones que contiene, esto se debe en parte a que en el archivo ".o" sólo se guarda el código compilado de todas las funciones. Los archivos ".a" por el contrario guardan, además del código, el nombre de cada función, esto hace que en el momento de enlazar el programa, cuando un módulo hace referencia a una función, el enlazador añade el código objeto de la función al programa ejecutable. De esta manera sólo se añaden al programa ejecutable aquellas funciones que realmente se utilizan en el programa.

**Caso práctico: Biblioteca para funciones matemáticas**

Si bien la biblioteca (library) estándar de C/C++ incluye funciones matemáticas, en este caso crearemos una biblioteca que incluya las siguientes funciones programadas: factorial, potencia, valor absoluto y raíz enésima.

La nueva biblioteca tendrá las siguientes funciones matemáticas:

// Función factorial int factorial (int n);	// Función valor absoluto float abs (float numero);	// Función raíz float raiz_n (int n, float q);
// Función potencia float potencia (float base, int exp);		

## Primera Parte: Creación del módulo con la biblioteca de funciones

### DESDE UNA VENTANA DE COMANDOS:

Cree los archivos "funcimat.cpp" y "funcimat.h"; El contenido de los archivos se encuentra en la misma carpeta en donde se encuentra este documento.

1. Copie los archivos mencionados a una carpeta local con nombre **prog1**. Abra una ventana de comandos y cambie el directorio de trabajo (usando el comando "cd") a la carpeta **prog1**. Configure la variable de entorno "path" para que encuentre el compilador de C/C++, ejecutando el comando:

minGW	set path=d:\minGW\bin;%path%
-------	------------------------------

2. Compile el archivo "funcimat.cpp" ejecutando el comando:

minGW	gcc -c -o funcimat.o funcimat.cpp
-------	-----------------------------------

Si el comando ejecutó correctamente debería usted encontrar en la carpeta de trabajo el archivo objeto "funcimat.obj" o "funcimat.o".

3. Ahora cree la biblioteca estática a partir del archivo ejecutando el comando:

minGW	ar r libfuncimat.a funcimat.o
-------	-------------------------------

Si el comando se ejecutó correctamente debería usted encontrar en la carpeta de trabajo el archivo objeto "libfuncimat.a".

### DESDE UN IDE:

En NetBeans también puede crear la biblioteca desde el entorno de desarrollo, elija la opción File/New Project..., en la ventana que aparece, en "Categories:" elija C/C++, en "Projects:" elija C/C++ Static Library. Elimine el módulo "main" (las nuevas versiones ya no lo crea por lo que no es necesario borrarlo) y agregue un nuevo archivo fuente C++ al proyecto llamado "funcimat" y copie en él el código de las funciones, finalmente construya el proyecto (Build), si no hay errores encontrará dentro de la carpeta dist\Debug\MinGW-Windows el archivo libfuncimat.a.

## Segunda Parte: Creación de un programa que utilice la biblioteca

Escriba un programa que pruebe las funciones matemáticas. El programa debe leer los datos para cada función, calcular su resultado y mostrarlo por la salida estándar. El programa debe verificar que las funciones sean válidas (factorial, potencia, valor absoluto y raíz enésima).

El programa debe ser escrito en un archivo "pruebafun.cpp". Dicho archivo, así como la librería "funcimat.lib" o "funcimat.a", deberá incluirlos en un proyecto "Lab2", y en una carpeta **prog2** de la siguiente forma:

NetBeans	En el menú File (Archivo) elija Project Properties (pruebafun), en Categories elegir Linker, a la derecha en la opción "Additional Library Directories" presione el botón con [...].En la ventana que aparece elegir "Add", luego seleccione la carpeta donde se encuentra el archivo "funcimat.a" o libfuncimat.a Luego en la opción Libraries presione el botón con [...].En la ventana que aparece elegir "Add Library..." elegir el archivo "funcimat.a" o libfuncimat.a
----------	---

Compile su proyecto, creando un ejecutable, y pruébelo.

**Profesores del curso: Miguel Guanira**

San Miguel, 9 de septiembre del 2019.