PISTAS.

1. Primera pista: Para crear un **std::map**, debes especificar dos tipos de datos: el tipo de dato de la clave y el tipo de dato del valor. Para añadir un elemento al mapa, utiliza la función **insert()** y crea un objeto **std::pair** con la clave y el valor. Por ejemplo: **mi\_mapa.insert(std::make\_pair(clave, valor))**.
2. Segunda pista: Para obtener el tamaño de un **std::map**, utiliza la función miembro **size()**. Por ejemplo: **mi\_mapa.size()**.
3. Tercera pista: Para eliminar un elemento de un **std::map**, utiliza la función miembro **erase()** y especifica la clave del elemento que deseas eliminar. Por ejemplo: **mi\_mapa.erase(clave)**.
4. Cuarta pista: Para recorrer un **std::map**, utiliza un bucle **for each** y una variable de tipo **std::pair**. La clave del elemento se encuentra en la propiedad **first** del objeto **std::pair**, y el valor se encuentra en la propiedad **second**. Por ejemplo:

for (const auto& par : mi\_mapa) { std::cout << "Clave: " << par.first << ", Valor: " << par.second << std::endl; }

1. Quinta pista: Para utilizar un tipo de dato personalizado en un **std::map**, define una estructura con los miembros que necesitas y utilízala como el tipo de dato del valor. Para insertar elementos en el mapa, crea un objeto de esta estructura y pásalo a la función **insert()**. Para acceder a los miembros de la estructura en un bucle **for each**, utiliza las propiedades **first** y **second** de **std::pair**, y accede a los miembros de la estructura a través de la propiedad **second**. Por ejemplo:

struct Persona { int edad; std::string genero; }; std::map<std::string, Persona> mi\_mapa; Persona p = { 20, "Femenino" }; mi\_mapa.insert(std::make\_pair("Ana", p)); for (const auto& par : mi\_mapa) { std::cout << "Clave: " << par.first << ", Edad: " << par.second.edad << ", Género: " << par.second.genero << std::endl; }