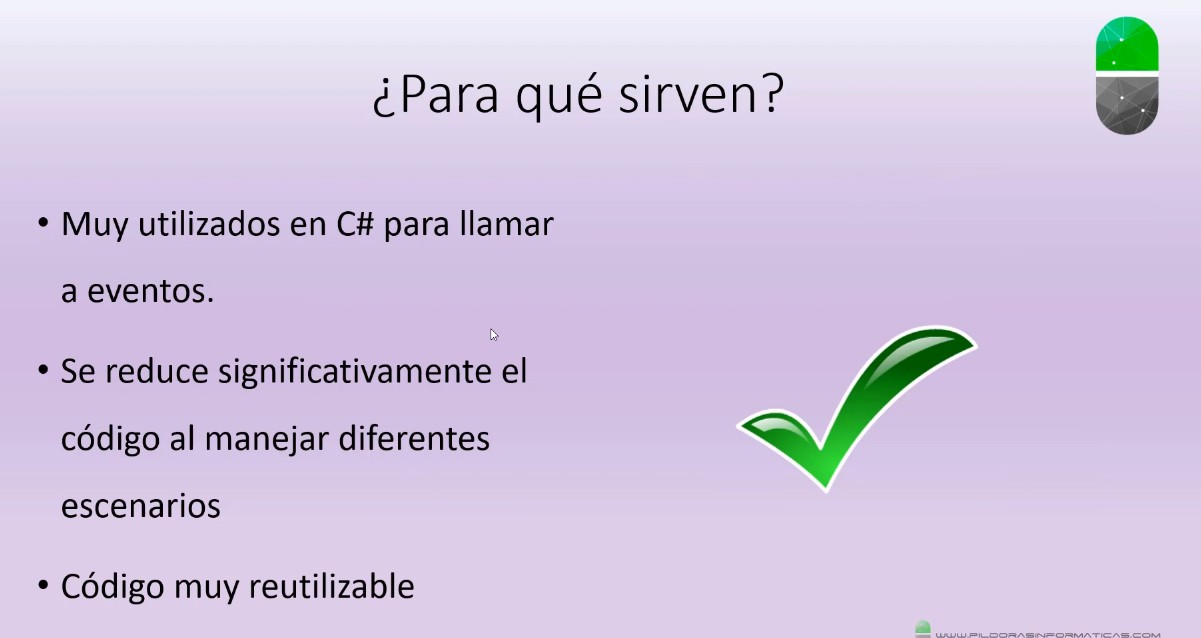
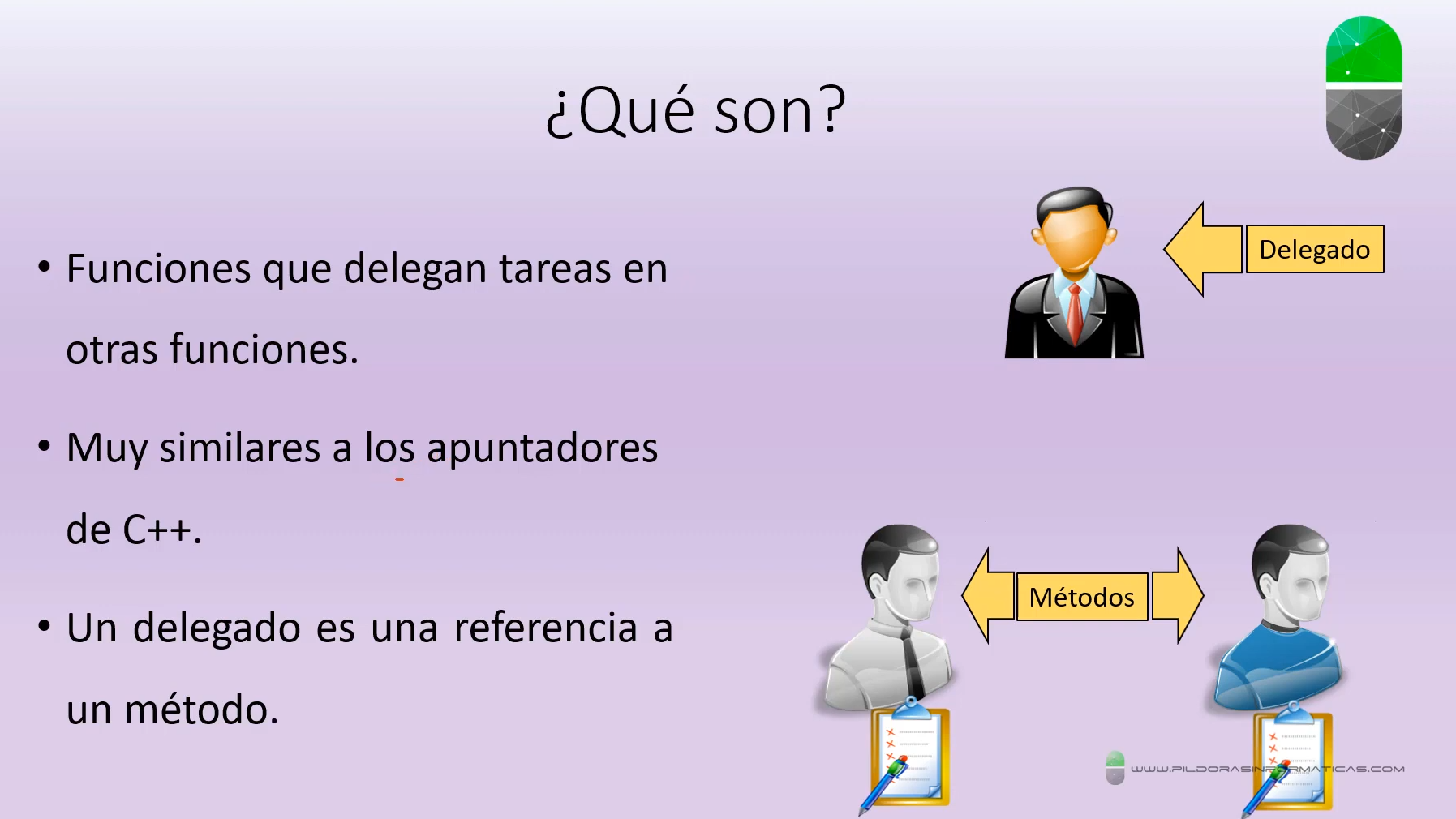
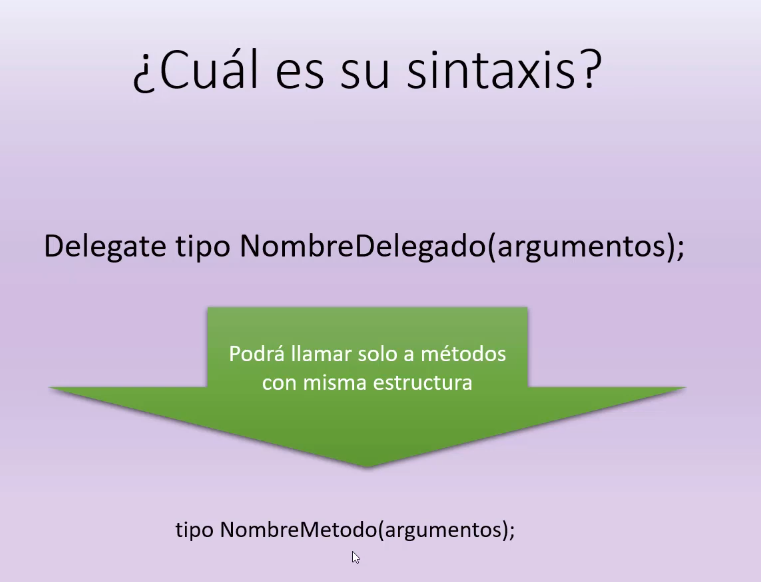
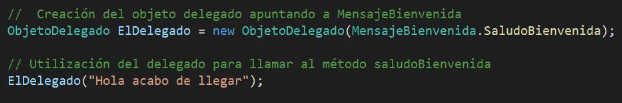
Delegados

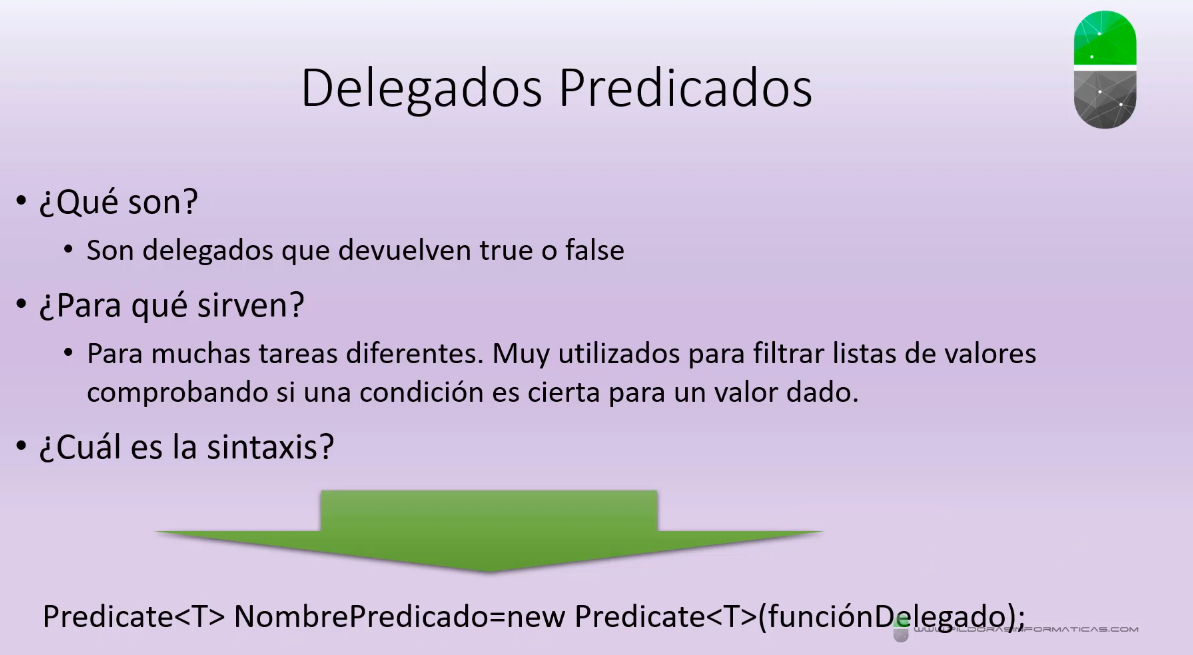


- El delegado y el método deben tener el mismo tipo y argumentos



Teoría:

Los **delegados** en C# son tipos que representan referencias a métodos. Son fundamentales porque permiten trabajar con métodos de manera flexible, como si fueran datos. La idea es que puedes usar delegados para encapsular métodos y luego invocarlos, independientemente de cuál sea la implementación concreta del método.



Un **predicado** en C# es un delegado predefinido que representa un método que toma un único parámetro de tipo genérico y devuelve un valor booleano (true o false). Es utilizado principalmente para evaluar condiciones o realizar comparaciones en colecciones como listas. Su definición genérica es Predicate<T>, donde T es el tipo del parámetro de entrada. Los predicados son comúnmente utilizados junto con métodos de colecciones como FindAll, que devuelve todos los elementos que cumplen una condición, Find, que busca el primer elemento que la cumple, Exists, que verifica si al menos un elemento cumple la condición, y RemoveAll, que elimina los elementos que la satisfacen.

Un ejemplo práctico sería filtrar números pares de una lista. Se puede definir un predicado como Predicate<int> esPar = num => num % 2 == 0; y luego usarlo con el método FindAll para obtener todos los números pares de la lista. Este enfoque simplifica la lógica de búsqueda y la hace reutilizable. Los predicados pueden definirse como métodos independientes, expresiones lambda o funciones anónimas, lo que permite un código más limpio y legible.

En resumen, los predicados son una herramienta versátil que facilita la manipulación de colecciones al permitir evaluar condiciones de manera eficiente y expresiva, promoviendo la reutilización del código y haciendo que las operaciones como búsquedas y filtrados sean más intuitivas.

