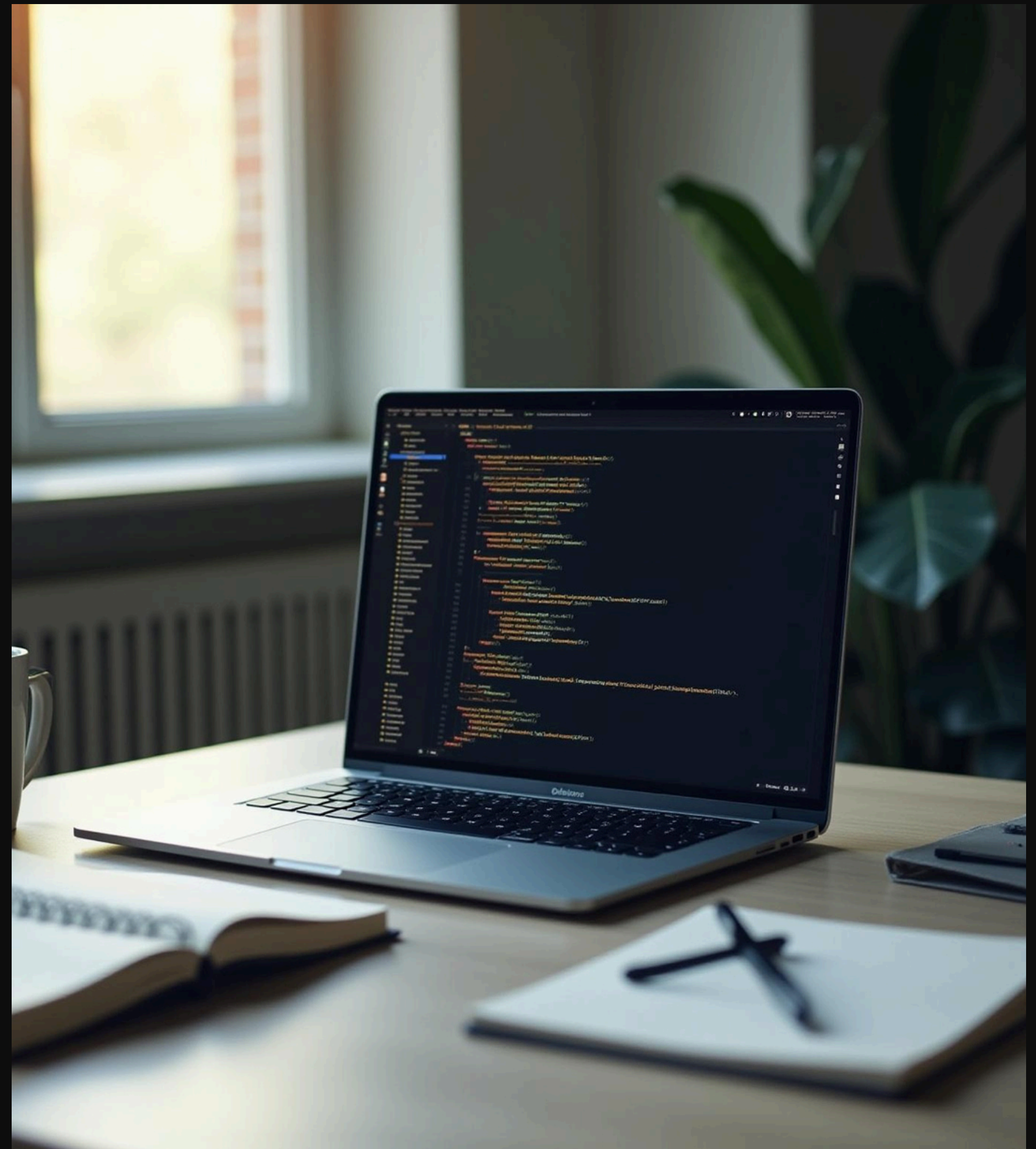


Golang Taller

Introducción y conceptos básicos

Introducción

Este taller se centra en los conceptos fundamentales de Go, un lenguaje de programación eficiente y concurrente. Exploraremos su historia, instalación y cómo escribir un programa básico para empezar a familiarizarse con su sintaxis y características principales.



Introducción a Golang

Historia y filosofía de Go

Go fue creado por Google para mejorar la productividad en programación. Se destaca por su simplicidad, rendimiento y soporte para programación concurrente, buscando combinar eficiencia con facilidad de uso en el desarrollo de software moderno.



Instalación y entorno de desarrollo

Para usar Go, primero se debe instalar su compilador y configurar el entorno. Los principales IDE soportan Go, facilitando la escritura, ejecución y depuración de códigos desde un entorno amigable y profesional.

Primer programa básico

El programa “Hola, mundo” es el punto de partida. Incluye la función principal y la impresión en consola, mostrando la estructura simple y clara que caracteriza a Go para iniciar el aprendizaje y desarrollo rápido.

Variables en Go

Declaración y tipos de datos

En Go, las variables se declaran con palabra clave `var` y tipos específicos. Soporta tipos básicos como enteros, cadenas y booleanos, esenciales para almacenar y manejar diferentes valores en aplicaciones.

Variables constantes

Las constantes se definen usando `const`, manteniendo su valor fijo durante la ejecución. Son útiles para valores que no deben cambiar como configuraciones o parámetros invariables en el código.



Ámbito y vida útil de variables

El ámbito define dónde una variable es accesible dentro del código, ya sea global o local a una función. La vida útil indica el tiempo que una variable existe en memoria durante la ejecución del programa. Comprender estos conceptos es esencial para evitar errores y gestionar recursos de manera eficiente.

Funciones y Métodos

Definición y sintaxis de funciones

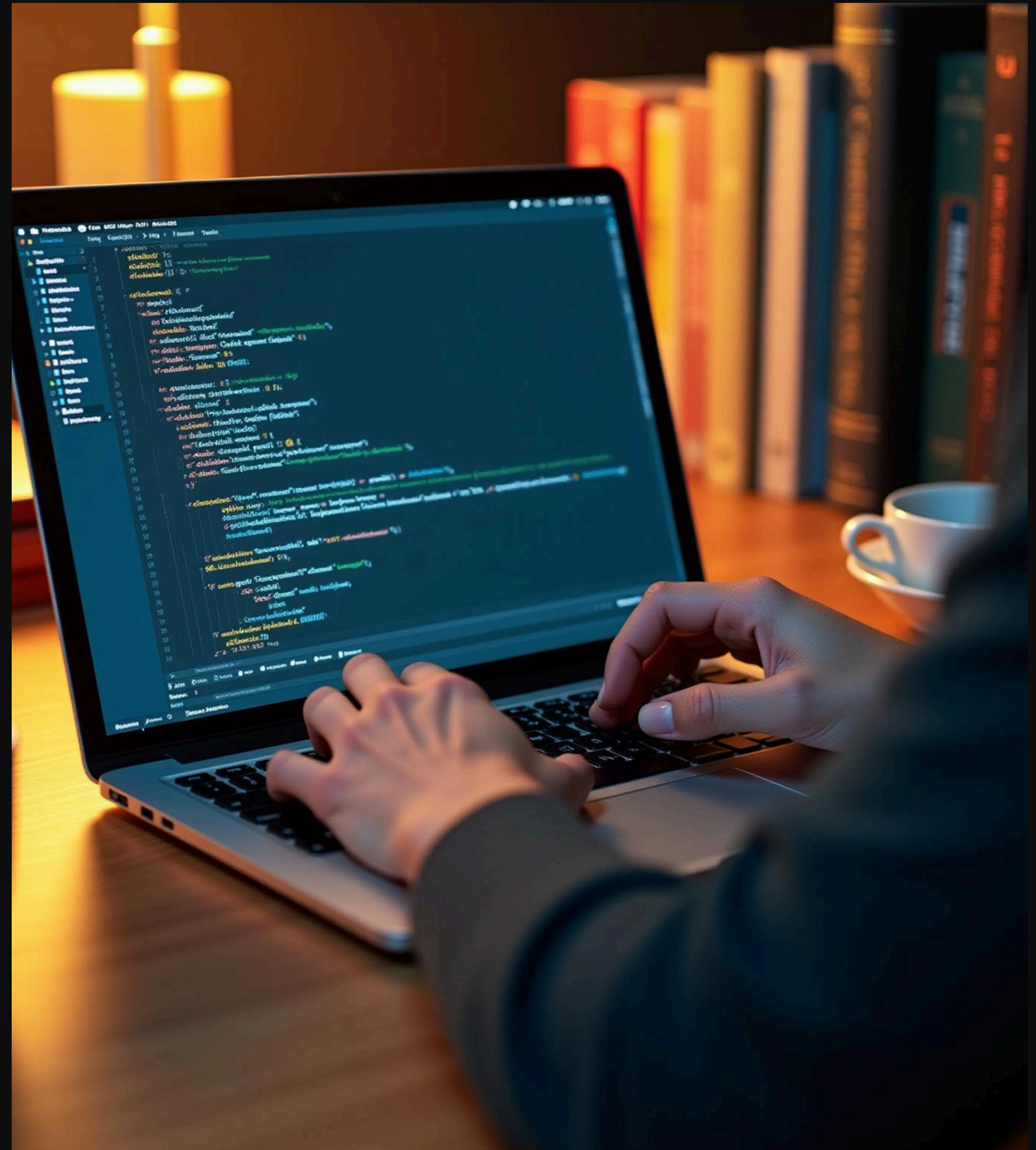
Una función en Go se define con la palabra clave `func`, seguida del nombre, parámetros y tipo de retorno. Las funciones permiten modularizar el código y reutilizar lógica, facilitando un desarrollo estructurado y mantenible.

Parámetros y valores de retorno

Las funciones en Go aceptan múltiples parámetros y pueden devolver varios valores. Esto ofrece flexibilidad para manejar distintas operaciones y resultados, optimizando la eficiencia y claridad del código.

Métodos asociados a structs

Los métodos son funciones que se asocian a un struct. Permiten operar sobre los datos encapsulados, promoviendo la programación orientada a objetos y mejorando la organización del código.



Structs y Punteros

Definición y uso de structs

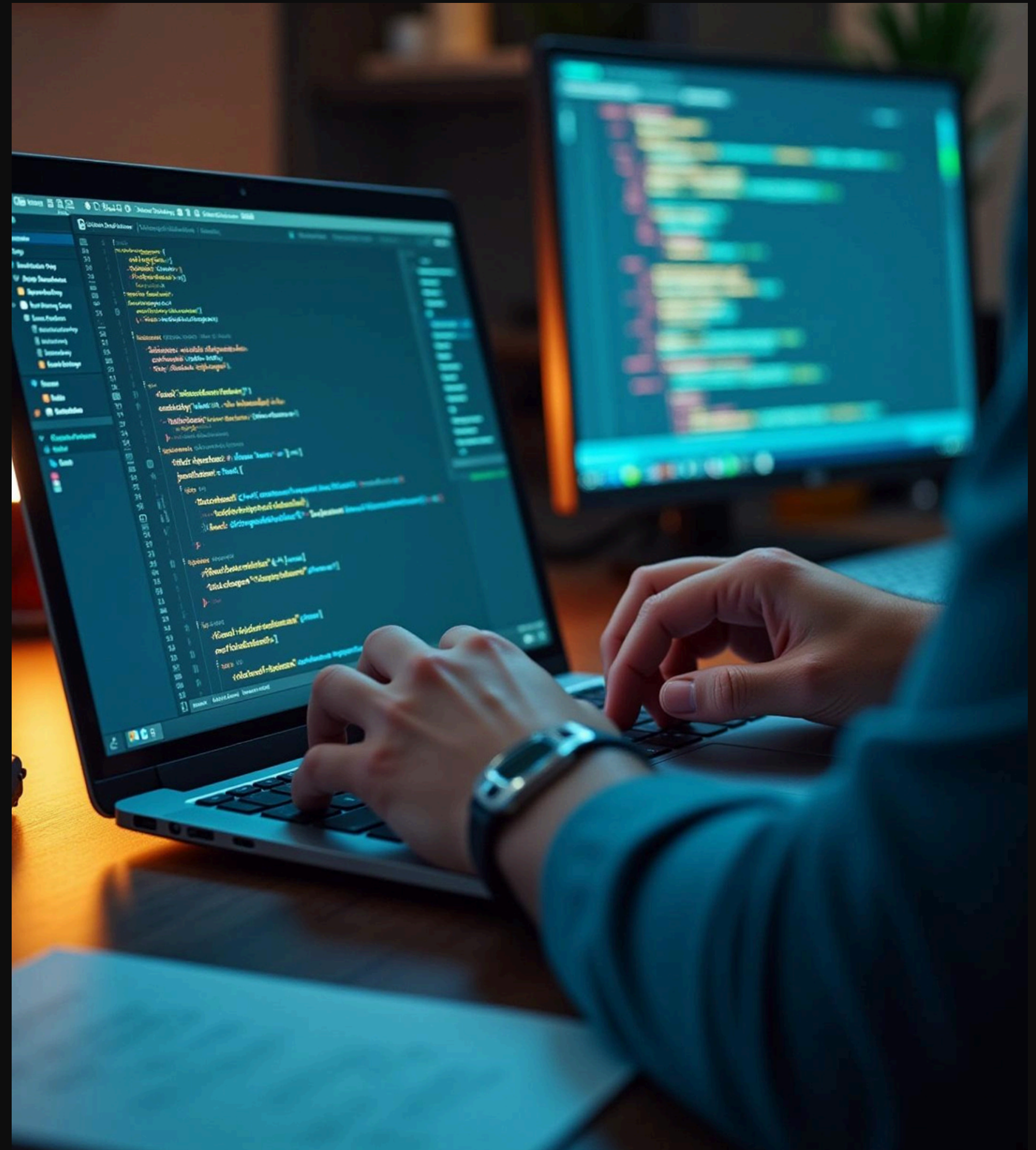
Un struct es una colección de campos que agrupa datos relacionados bajo un único tipo. Son fundamentales para modelar entidades complejas y organizar la información de manera estructurada y eficiente.

Concepto y uso de punteros

Un puntero almacena la dirección de memoria de una variable. Usar punteros permite modificar datos directamente y manejar estructuras dinámicas, crucial para optimizar recursos y rendimiento.

Manipulación de datos con punteros y structs

Combinando punteros y structs se logra acceso eficiente a datos complejos. Esto permite modificar campos sin copiar toda la estructura, mejorando la velocidad y uso de memoria en aplicaciones críticas.



Conclusiones

Este taller ha cubierto fundamentos de Go esenciales: variables, funciones, métodos, structs y punteros. Dominar estos elementos facilita el desarrollo de software robusto y eficiente aprovechando las ventajas de Go en entornos profesionales y corporativos.

Thank you

Do you have any questions?

3855921830108@ingenieria.usac.edu.gt

<https://github.com/yaniadavid/Taller-Golang>



CREDITS: This presentation template was created by **Slidesgo**, and includes icons, infographics & images by **Freepik**

Please keep this slide for attribution