

Resumen:

En este módulo, se han explorado las diferentes **expresiones de bucle** disponibles en Rust y se ha descubierto cómo trabajar con **mapas hash**. Los datos se almacenan en un mapa hash como pares clave-valor. Los mapas hash se pueden aumentar.

La expresión **loop** repite las acciones hasta que se detiene manualmente el proceso. Puede crear un bucle con la expresión **while** para repetir acciones siempre y cuando una condición siga siendo true. La expresión **for** se usa para iterar una colección de datos.

En el ejercicio, ha ampliado un programa de automóviles para recorrer en bucle las acciones repetidas a fin de cumplimentar todos los pedidos. Ha implementado un mapa hash para realizar un seguimiento de los pedidos.

En el siguiente módulo de esta ruta de aprendizaje, profundizará en la forma en que se tratan los errores en el código de Rust.

Más información:

Visite los vínculos siguientes para obtener más información sobre algunos de los puntos analizados en este módulo:

- [Introducción a Rust](#)
- [Más información sobre el tipo de estructura de programación clásica de C](#)
- [Revisión de los tipos de datos algebraicos](#)

Documentación de referencia de Rust:

- [El lenguaje de programación Rust](#)
- [Revisión de las palabras clave de Rust](#)

Rust: tipos de datos:

- [Trabajo con matrices](#)
- [Trabajo con valores booleanos](#)
- [Trabajo con caracteres](#)
- [Trabajo con números decimales y tipos de número de punto flotante](#)
- [Trabajo con enumeraciones](#)
- [Trabajo con mapas hash](#)
- [Trabajo con tipos enteros](#)
- [Operaciones con cadenas](#)
- [Trabajo con estructuras](#)
- [Trabajo con tuplas](#)
- [Trabajo con vectores](#)

Rust:

Conceptos:

- [Comprensión de las variables, la mutabilidad y la propiedad reemplazada](#)
- [Más información sobre las funciones](#)
- [Usar condiciones if/else](#)
- [Incorporación de flujo de control con bucles](#)
- [Muestra de la salida con la macro println!](#)