

1.

```
package main

import (
    "errors"
    "fmt"
    "os"
)

type produto struct {
    id          int
    quantidade  int
    preco       float64
}

func (p produto) gerarLinhaCSV() string {
    return fmt.Sprintf("%d,%d,%.2f\n", p.id, p.quantidade, p.preco)
}

func (p produto) gerarCabecalhoCSV() string {
    return "id,quantidade,preco\n"
}

func gerarCsv(caminho string, produtos []produto) error {

    if len(produtos) == 0 {
        return errors.New("quantidade de produto inválida")
    }

    file, err := os.OpenFile(caminho, os.O_WRONLY|os.O_CREATE, 0600)
    if err != nil {
        return fmt.Errorf("erro ao abrir arquivo: %w", err)
    }
    defer file.Close()

    p := produtos[0]

    if _, err = file.WriteString(p.gerarCabecalhoCSV()); err != nil {
        return fmt.Errorf("erro ao gerar cabeçalho: %w", err)
    }

    for _, p = range produtos {
```

```
        if _, err = file.WriteString(p.gerarLinhaCSV()); err != nil {
            return fmt.Errorf("erro ao salvar linha: %w", err)
        }
    }

    return nil
}

func main() {

    produtos := []produto{
        {
            id:          3,
            quantidade: 8,
            preco:       6.99,
        },
        {
            id:          2,
            quantidade: 20,
            preco:       12.99,
        },
        {
            id:          1,
            quantidade: 10,
            preco:       9.99,
        },
    }

    gerarCsv("productos.csv", produtos)
}
```

2.

```
package main

import (
    "bufio"
    "fmt"
    "log"
    "os"
    "strings"
    "text/tabwriter"
)

func main() {
    file, err := os.Open("productos.csv")
    if err != nil {
        log.Fatal(err)
    }
    defer file.Close()

    w := tabwriter.NewWriter(os.Stdout, 20, 30, 1, '\t',
tabwriter.AlignRight)

    scanner := bufio.NewScanner(file)

    scanner.Scan()

    cabecalho := strings.Split(scanner.Text(), ",")

    for _, c := range cabecalho {
        fmt.Fprintf(w, "%s\t", c)
    }

    fmt.Fprintln(w)

    for scanner.Scan() {
        values := strings.Split(scanner.Text(), ",")
        for _, v := range values {
            fmt.Fprintf(w, "%s\t", v)
        }
        fmt.Fprintln(w)
    }

    w.Flush()
}
```

```
}
```