

## Validação de CNPJ usando go lang

```
package cnpj

import (
    "errors"
    "strconv"
    "strings"
    "regexp"
)

func Valid(digits string) (bool, error) {
    return valid(digits)
}

func sanitize(data string) string {
    data = strings.Replace(data, ".", "", -1)
    data = strings.Replace(data, "-", "", -1)
    data = strings.Replace(data, "/", "", -1)
    return data
}

func valid(data string) (bool, error) {
    data = sanitize(data)

    if len(data) != 14 {
        return false, errors.New("Invalid length")
    }

    if strings.Contains(blacklist, data) || !check(data) {
        return false, errors.New("Invalid value")
    }

    return true, nil
}

const blacklist = `00000000000000
1111111111111111
2222222222222222
3333333333333333
4444444444444444
5555555555555555
6666666666666666
7777777777777777
8888888888888888
9999999999999999`

func stringToIntSlice(data string) (res []int) {
    for _, d := range data {
        x, err := strconv.Atoi(string(d))
        if err != nil {
            continue
        }
        res = append(res, x)
    }
    return
}

func check(data string) bool {
    return verify(stringToIntSlice(data), 5, 12) && verify(stringToIntSlice(data), 6, 13)
}

func verify(data []int, j int, n int) bool {
    soma := 0

    for i := 0; i < n; i++ {
        v := data[i]
        soma += v * j

        if j == 2 {
            j = 9
        } else {
            j -= 1
        }
    }

    resto := soma % 11
```

```
v := data[n]
x := 0

if resto >= 2 {
    x = 11 - resto
}

if v != x {
    return false
}

return true
}

func FilterNumber(text string) string{
    re := regexp.MustCompile("[0-9]+")
    result := re.FindAllString(text, -1)
    number := ""
    for _, s := range result {
        number += s
    }

    return number
}
```