

Especialización en Back End III

Taller de código: Desafío Go

Objetivo

A continuación se plantea un desafío integrador que nos permitirá evaluar los temas que hemos visto hasta el momento.

Planteo

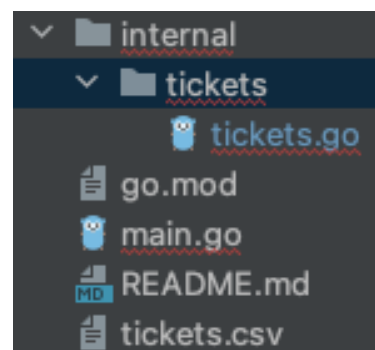
Una aerolínea pequeña tiene un sistema de reservas de pasajes a diferentes países. Este retorna un archivo con la información de los pasajes sacados en las últimas 24 horas. La aerolínea necesita un programa para extraer información de las ventas del día y, así, analizar las tendencias de compra.

El archivo en cuestión es del tipo valores separados por coma (**CSV**), donde los campos están compuestos por: **id, nombre, email, país de destino, hora del vuelo y precio**.

¡Mucha suerte!

Desafío

Realizar un programa que sirva como herramienta para calcular diferentes datos estadísticos. Para lograrlo, deberás descargar este [repositorio](#) que contiene un archivo CSV con datos generados y un esqueleto del proyecto.



¡Atención! Los ejemplos a continuación son solo a modo de guía. El desafío se puede resolver de múltiples maneras.

Requerimiento 1

Una función que calcule cuántas personas viajan a un país determinado.

```
func GetTotalTickets(destination string) (int, error) {}
```

(ejemplo 1)



Tip: VS Code nos permite buscar una palabra en un archivo con **Ctrl + F** o **⌘ + F**.

Requerimiento 2:

Una o varias funciones que calculen cuántas personas viajan en **madrugada (0 → 6)**, **mañana (7 → 12)**, **tarde (13 → 19)**, y **noche (20 → 23)**.

```
func GetCountByPeriod(time string) (int, error) {}
```

(ejemplo 2)



Tip: En Go, para manipular caracteres, tenemos el paquete [strings](#).

Requerimiento 3:

Calcular el **porcentaje** de personas que viajan a un país determinado en un día.

```
func PercentageDestination(destination string, total int) (float64, error) {}
```

(ejemplo 3)



Tip: El **porcentaje de X** se calcula como: $\bar{x} = \frac{\sum x}{n} \cdot 100$

Requerimiento 4:

Ejecutar al menos una vez cada requerimiento en la función main. Las ejecuciones deben realizarse de manera **concurrente** (utilizando diferentes **goroutines**).

```
go func(){  
    totalTickets, err := GetTotalTickets("Brazil")  
    // process all the data  
}()
```

(ejemplo 4)



Tip: Se pueden enviar parámetros a la función anónima de la siguiente manera: `go func(param string)`
`{ // function detail } ("param string")`