

YoFio - Backend Golang - Prueba técnica

Problema

Cuando disponemos de recursos económicos de nuevos inversionistas nos gusta plantearnos cuál es la mejor opción para invertirlos. Tenemos 3 montos de créditos que damos a nuestros clientes (\$300, \$500 y \$700). Cuando llega el dinero de inversión, queremos determinar cuántos créditos de cada monto podríamos asignar con ese dinero, sin que nos sobre 1 peso. Tu trabajo para este ejercicio es ayudarnos a calcular las posibles cantidades de créditos de \$300, \$500 y \$700 que podemos otorgar con el total de la inversión. Si existe más de una opción podrías seleccionar cualquiera de ellas. Aquí algunos ejemplos:

- Si nos llegan \$3,000 de inversión, podríamos asignar 2 créditos de \$300, 2 de \$500 y 2 de \$700, así: $2 \times \$300 + 2 \times \$500 + 2 \times \$700 = \$3,000$.
- Si nos llegan \$6,700 de inversión, podríamos asignar 7 créditos de \$300, 5 de \$500 y 3 de \$700, así: $7 \times \$300 + 5 \times \$500 + 3 \times \$700 = \$6,700$.
- Si nos llegan \$400 de inversión, no podríamos asignar ninguna combinación válida ya que sobraría dinero en cualquier caso, e.g. asignando un crédito de \$300 sobrarían \$100 pesos que no podrían ser asignados.

Para esto, es necesario que construyas un programa en Golang con un método que satisfaga la siguiente interfaz:

```
type CreditAssigner interface {  
    Assign (investment int32) ( int32 , int32 , int32 , error )  
}
```

Donde recibas como parámetro la inversión y retornes 3 valores, que corresponden a cuántos créditos de cada tipo podemos asignar. Si no es posible asignar un valor, todos los valores deben ir en cero y retornar un error en el último parámetro. Está garantizado que el monto de inversión es un múltiplo de 100.

Retos

Nivel básico

Crea una implementación en Golang que satisfaga la anterior interfaz.

Nivel Intermedio

Crea una API REST, y alójala en cualquier nube que desees (GCP, AWS, Azure, etc). Crea un servicio Rest /credit-assignment en donde se retorna el número de créditos a asignar por cada tipo:

POST → /credit-assignment

{“investment”: 3000}

En caso de que se pueda asignar el crédito debería retornar un código HTTP 200 con esta respuesta:

{“credit_type_300”: 2, “credit_type_500”: 2, “credit_type_700”: 2}

En caso de que no pueda realizarse la asignación deberá retornar un código HTTP 400.

Nivel Avanzado

Crear una base de datos para almacenar las asignaciones de créditos realizadas con la API. Crear un nuevo servicio POST → /statistics para retornar los estadísticos de las asignaciones con los siguientes datos:

- Total de asignaciones realizadas (e.g. 100)
- Total de asignaciones exitosas (e.g. 40)

- Total de asignaciones no exitosas (e.g. 60)
- Promedio de inversión exitosa (e.g. 3545.6)
- Promedio de inversión no exitosa (e.g. 350.3)

Nota importante: La API puede recibir altos niveles de tráfico en las pruebas de estrés que le realicemos (más de 1,000 transacciones por segundo).

Crear las pruebas unitarias y de integración con cobertura superior al 90%.

Entrega

Se debe enviar un correo electrónico a gcastillo@yofio.co con lo siguiente:

- Código Fuente (Para niveles intermedios y avanzados: En repositorio GitHub o GitLab).
- Instrucciones requeridas para la ejecución del programa o de la API. Para los niveles intermedios y avanzados un archivo README.md en el repositorio.
- Para los niveles intermedios y avanzados la URL de la API para consumirla.