

Informe PTI

Lab 5

Autores
Marc Cabezas
Antonio Arellano

1. Familiarización con el entorno	3
2. Caso práctico	4
2.1.Webserver.go	4
3. Conclusiones	6
3.1 Conclusiones de la familiarización con el entorno	6
3.2 Conclusiones del caso práctico	6
3.3 Conclusiones finales	6

1. Familiarización con el entorno

Se ha configurado la máquina, instalando Go, CURL y GIT para poder descargar correctamente los archivos y poder ejecutar los ficheros .go.

Se han ejecutado estos comandos:

```
sudo apt-get install curl
```

```
sudo apt-get install git
```

```
wget https://storage.googleapis.com/golang/go1.7.1.linux-amd64.tar.gz
```

```
sudo tar -C /usr/local -xzf go1.7.1.linux-amd64.tar.gz
```

```
sudo chmod -R 777 /usr/local/go
```

```
export PATH=$PATH:/usr/local/go/bin
```

```
mkdir $HOME/go
```

```
mkdir $HOME/go/src
```

```
export GOPATH=$HOME/go
```

```
git clone https://github.com/YOUR_GITHUB_USER/pti_golang.git
```

```
cd $HOME/go/src/pti_golang
```

```
mkdir hello
```

```
cd hello
```

```
wget https://gitlab.fib.upc.edu/pti/pti/raw/master/p5_goREST/src/hello/hello.go
```

```
go install pti_golang/hello
```

```
$HOME/go/bin/hello
```

Una vez estaba todo instalado se han realizado pruebas para comprobar que todo se había instalado correctamente.

Se han realizado más pruebas con varios ejemplos:

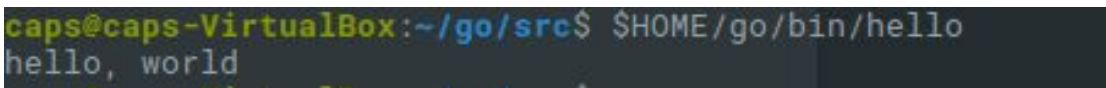
A terminal window with a dark background. The prompt is 'caps@caps-VirtualBox:~/go/src\$'. The user enters '\$HOME/go/bin/hello'. The output is 'hello, world'.

Figura 1. Salida realizada por el ejemplo hello.go

Después se han realizado ejemplos con un Endpoint que nosotros mismos hemos creado y que nos devolvía un valor estático.

2. Caso práctico

En esta sección se explicarán las soluciones adoptadas para los problemas propuestos, luego éstas soluciones nos ayudarán a elaborar un conclusión.

2.1.Webserver.go

Se nos pedía crear un programa con el cual se pudiera hacer diferentes acciones respecto a unos endpoints.

Las acciones son las siguientes:

-Guardar datos mediante una llamada al Endpoint

Con un endpoint poder guardar los nuevos alquileres que se pasen por parámetros:

```
caps@caps-VirtualBox:~$ curl -H "Content-Type: application/json" -d '{"carMaker":"BMW", "carModel":"Z4", "numDays":"2", "numUnits":"1"}' http://localhost:8081/registerCar/1234?queryparam1=Value3
Successfully received car registration. Here is your rental:
{BMW Z4 2 1}
precio 200
```

Figura 2. Salida por terminal al realizar correctamente el alta de un alquiler.

```
1
2 a,a,2,1,200
3 a,a,2,1,200
4 a,a,2,1,200
5
```

Figura 3. Fichero donde se almacena toda la información de los alquileres.

-Obtener una lista con todos los alquileres

Con este Endpoint podemos obtener todos los alquileres que hemos realizado.

```
{
  {
    CarMaker: a
    CarModel: a
    NumDays: 2
    NumUnits: 1
    Precio: 200
  }
  {
    CarMaker: a
    CarModel: a
    NumDays: 2
    NumUnits: 1
    Precio: 200
  }
  {
    CarMaker: a
    CarModel: a
    NumDays: 2
    NumUnits: 1
    Precio: 200
  }
  {
    CarMaker: BMW
    CarModel: Z4
    NumDays: 2
    NumUnits: 1
    Precio: 200
  }
}
```

Figura 4. Lista de todos los alquileres realizados.

3. Conclusiones

Las conclusiones se vuelven a estructurar de la siguiente manera, primero comentaremos las conclusiones que hemos sacado del punto 1 (Familiarización con el entorno), luego las del punto 2 (Caso práctico) y finalmente a partir de las conclusiones anteriores, intentaremos sacar una conclusión definitiva de las tecnologías con las que hemos trabajado, así como puntos a favor y en contra o posibles escenarios en donde éstas tecnologías serían de gran utilidad.

3.1 Conclusiones de la familiarización con el entorno

La instalación de los paquetes necesarios no ha dado problemas. En la prueba de los ejemplos no ha habido ningún problema. Un punto positivo es el entorno de desarrollo, Go es un nuevo idioma que puede resultar muy útil tener unas nociones básicas. A parte de tener una comunidad bastante activa y en auge, cosa que es bastante positiva ya que facilita la resolución rápida de problemas por los recursos que podemos encontrar en la red. Otro punto a favor de este lenguaje es su simplicidad, tanto en la sintaxis como en las herramientas para su funcionamiento, con un simple editor de texto plano podemos, sin muchos problemas, codificar un programa complejo.

3.2 Conclusiones del caso práctico

No ha sido difícil entender cómo poder generar un Endpoint en el que poder cargar los datos que son necesarios para ir creando instancias e ir las guardando en un fichero .csv.

Tampoco ha sido muy costoso realizar otra llamada que nos devolviese todos los alquileres guardados y que nos lo mostrase en un formato JSON.

Podemos concluir que Go es una herramienta especialmente útil y rápida para desplegar entornos en los que son necesarias llamadas a una API. Se ha destacado anteriormente la simplicidad del lenguaje, pero la volvemos a remarcar, al ser un lenguaje simple y bastante potente no ha sido laborioso realizar lo que se nos pedía para el caso práctico, este mismo ejercicio con otro lenguaje hubiese sido un tedioso trabajo en el que se tendría que haber estado pendiente de ciertas dependencias y caprichos del lenguaje, con Go en cambio, ha sido bastante ameno.

3.3 Conclusiones finales

Basándonos en conclusiones anteriores, encontramos que el lenguaje Go es un gran herramienta para el desarrollo de entornos BackEnd y nos brinda muchas facilidades

con respecto otros lenguajes. Nos ha sorprendido su simplicidad y lo limpio que queda el código cuando se pretende llevar a cabo tareas difíciles. Pensamos que es una tecnología bastante interesante de investigar, no es un lenguaje muy extendido, comparado con otros como Java o Python, pero con una comunidad detrás bastante activa, por lo que lo convierte en unos de esos lenguajes que van escalando muy rápidamente posiciones en los rankings de popularidad. Dicho lo cual, recomendaríamos su uso para llevar a cabo lo relacionado con el diseño BackEnd de algún proyecto, y hasta apostaríamos por él para otras partes del proyecto diferentes al BackEnd. Con respecto otras tecnologías trabajadas en anteriores prácticas, podríamos decir que Go está al nivel de simplicidad de Python y rozando la potencia de Java (en algunos casos). Por lo que si lo que necesitamos es una tecnología que su deployment sea rápido y sencillo, y a su vez nos proporcione una potencia media (más que Python pero rozando Java), Go suple ese lugar con creces.