PRÁCTICA 5 - PTI

**Kais Harim, Omair Iqbal**

8.03.2017

# 

# 

# 

[**INTRODUCCIÓN**](#_afbvsin9fb62) **3**

[GO](#_hi48kljilqpu) 3

[**RECURSOS**](#_lquiyrwpy6ke) **3**

[**PROCEDIMIENTO**](#_kn5uvgo00ajj) **3**

[**CONCLUSIONES**](#_chou9188p6co) **5**

# 

# 

# INTRODUCCIÓN

El objetivo de esta sesión es aprender este nuevo lenguaje, GO, y resolver el mismo problema que en las sesiones anteriores pero con esta nueva tecnología.

## GO

Go es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [concurrente](https://es.wikipedia.org/wiki/Computaci%C3%B3n_concurrente) y [compilado](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_compilado) inspirado en la sintaxis de [C](https://es.wikipedia.org/wiki/C). Ha sido desarrollado por [Google](https://es.wikipedia.org/wiki/Google) y sus diseñadores iniciales son [Robert Griesemer](https://es.wikipedia.org/w/index.php?title=Robert_Griesemer&action=edit&redlink=1), [Rob Pike](https://es.wikipedia.org/wiki/Rob_Pike) y [Ken Thompson](https://es.wikipedia.org/wiki/Ken_Thompson). Actualmente está disponible en formato binario para los [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo) [Windows](https://es.wikipedia.org/wiki/Microsoft_Windows), [GNU/Linux](https://es.wikipedia.org/wiki/GNU/Linux), [FreeBSD](https://es.wikipedia.org/wiki/FreeBSD) y [Mac OS X](https://es.wikipedia.org/wiki/OS_X), pudiendo también ser instalado en estos y en otros sistemas con el código fuente. Go es un lenguaje de programación compilado, concurrente, imperativo, estructurado, orientado a objetos —de una manera bastante especial— y con recolector de basura que de momento está soportado en diferentes tipos de sistemas UNIX, incluidos [Linux](https://es.wikipedia.org/wiki/Linux), [FreeBSD](https://es.wikipedia.org/wiki/FreeBSD) y [Mac OS X](https://es.wikipedia.org/wiki/Mac_OS_X). También está disponible en [Plan 9](https://es.wikipedia.org/wiki/Plan_9_from_Bell_Labs) puesto que parte del compilador está basado en un trabajo previo sobre el sistema operativo [Inferno](https://es.wikipedia.org/wiki/Inferno). Las arquitecturas soportadas son i386, amd64 y ARM.

# RECURSOS

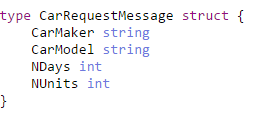
1. Git de la asignatura
2. GO
3. Herramientas para editar documentos GO

# PROCEDIMIENTO

En primer lugar hemos tenido que instalar los recursos necesarios para proceder con la práctica. Hemos instalado Curl, para las llamadas al servidor; GO, para poder compilar el programa; y GIT para poder obtener los archivos a implementar del GIT de la asignatura.

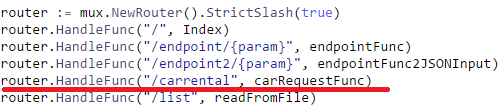
Posteriormente, hemos implementado una serie de programas simples que devuelven un valor, para adaptarnos a este lenguaje.

A continuación hemos procedido con el ejercicio que pide la práctica. Como tenemos que tratar con diversos valores relacionados con la reserva de un vehículo, utilizamos JSON para pasar los valores.



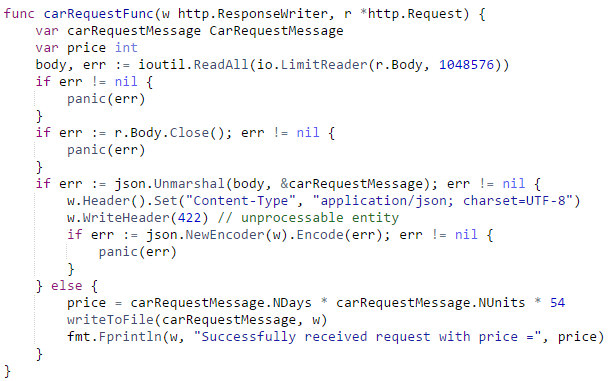
*Figura 1: Estructura de valores del JSON*

Añadimos un endpoint para realizar la petición con el navegador o mediante la llamada con CURL, añadiendo la siguiente línea de código:



*Figura 2: Creación del endpoint*

Una vez, creado el endpoint, implementamos la función correspondiente para realizar la reserva:

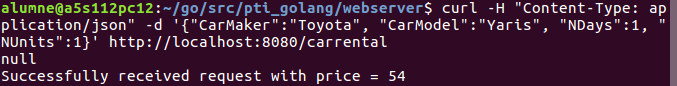


*Figura 4: Función asociada al carRequest*

En esta función, recibimos los valores escritos por terminal mediante la llamada curl y se procesan para calcular el precio final de la reserva, finalmente se muestra por el terminal al usuario.

2.png

*Figura 5: llamada CURL al carRental*



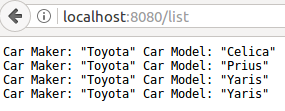
*Figura 6: La función devuelve el precio del alquiler*

Una vez hecha la petición, ésta se añade a un documento .csv para almacenarla persistentemente y poder consultar las reservas cuando se desee. Al hacer la primera reserva este documento se crea automáticamente bajo el nombre de rentals.csv y tiene la siguiente forma:



*Figura 7: Contenido del rentals.csv*

También se puede visualizar desde un browser accediendo al endpoint. Veríamos la siguiente información:



*Figura 8: Muestreo de alquileres desde el browser*

# 

# CONCLUSIONES

Esta práctica nos ha resultado muy útil, ya que nos ha permitido aprender el funcionamiento de un recurso que no conocíamos como es GO y ver el gran potencial que tiene. En definitiva una práctica muy interesante acorde con la materia de la especialidad y muy útil para mejorar nuestras capacidades. Sobre todo el hecho de realizar el mismo ejercicio de distintas maneras, nos está haciendo madurar nuestro criterio para resolver los problemas que se nos presenten en el futuro.