

## Especialización en Back End III

# Ejercitación

- Ejercitación individual
- Nivel de complejidad: medio 🔥🔥

## Objetivo

El objetivo de esta guía práctica es que podamos afianzar los conceptos sobre **arquitectura web, JSON y Gin framework**, vistos en este módulo.

## Problema

Una empresa necesita un sistema para gestionar sus empleados. Para poder hacer esto, necesitan un **servidor** que ejecute una API que les permita manipular a los empleados.

## Ejercicio 1: Iniciando el proyecto

Debemos crear un repositorio en GitHub para poder subir nuestros avances. Este repositorio es el que vamos a utilizar para contener todo lo que realicemos durante las distintas prácticas de Go Web.

1. Primero debemos clonar el repositorio creado. Luego, iniciar nuestro proyecto de Go con el comando **go mod init**.
2. El siguiente paso será crear un archivo **main.go** donde deberán cargar un **slice**, desde una función que devuelva una lista de empleados. Este slice se debe cargar cada vez que se inicie la API para realizar las distintas consultas.

La estructura de los empleados es la siguiente:

<b>Id</b>	int
<b>Nombre</b>	string
<b>Activo</b>	bool

## Ejercicio 2: Empleados

Vamos a levantar un servidor utilizando el paquete [Gin](#) en el puerto 8080. Para probar nuestros endpoints, haremos uso de Postman.

1. Crear una ruta `/` que nos devuelva una bienvenida al sistema. Ejemplo: “¡Bienvenido a la empresa Gophers!”.
2. Crear una ruta `/employees` que nos devuelva la lista de todos los empleados en formato JSON.
3. Crear una ruta `/employees/:id` que nos devuelva un empleado por su ID. Manejar el caso de que no encuentre el empleado con ese ID.
4. Crear una ruta `/employeesparams` que nos permita crear un empleado a través de los params y lo devuelva en formato JSON.
5. Crear una ruta `/employeesactive` que nos devuelva la lista de empleados activos. También debería poder devolver la lista de los empleados no activos.