Challenge

Frontend

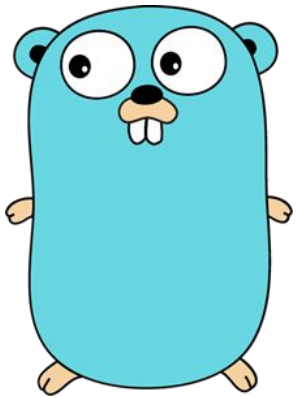
- Vue.js - framework
- Tailwind CSS - CSS framework
- Sweet Alert
- Axios - cliente HTTP basado en promesas

Frontend

- SPA, con 3 vistas: Home, Endpoint 1, Endpoint 2
- 2 componentes: Header y Button
- Seguridad: posee regex para verificar forma de dominio web
- UX: mensajes de confirmación y error
- Consume API de webserver en Go

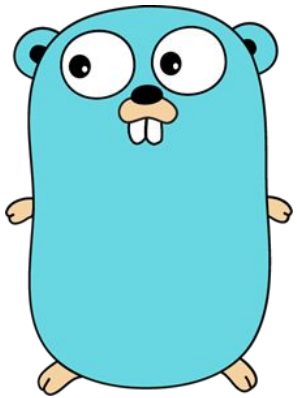
Backend

- Golang
- Dos endpoints:
  - Endpoint #1: POST, para consultar los dominios y escribirlos en la DB /info/servers/:domain
  - Endpoint #2: GET, para revisar en la base de datos los dominios consultados /queried/domains



Backend

- En cada Endpoint se proyecta JSON con la información correspondiente
- Seguridad: posee consultas SQL parametrizadas para evitar SQL injection
- Web Scrapping:
  - uso de librería /x/net/html para extraer title de página web
  - uso de librería PuerkitoBio/goquery para extraer URL de logo página



Database

- CockroachDB
- Script construye el esquema de la DB automáticamente
- Manejo de JSONB para almacenar información correspondiente a dominio web

Containers

- Docker
- 4 contenedores, 3 reales
- Uno para cada capa de la aplicación (front, back, db)
- Uno de los contenedores sólo sirve para configurar la base de datos y muere al terminar su tarea
- Despliegue con un solo comando