Crab IOT

...

Objetivo

- Backend de un HUB para dispositivos IOT.
- Orientado a la privacidad.
- Fácil de desplegar privadamente.
- Sistema de plugins.

Historias de usuario

1. Consultar información recopilada por un dispositivo.
2. Modificar estado de un actuador.
3. Automatización de acciones.
4. Añadir soporte para nuevos dispositivos fácilmente.

Herramientas

- **cargo:** Herramienta todoterreno de Rust.
  - Gestor de paquetes.
  - Gestor de dependencias.
  - Automatiza la compilación.
  - Marco de tests y benchmarks.
  - Publicación de paquetes.
  - Generación de documentación.
- **make.**

Tests

- Marco de tests de Rust.
- Biblioteca de aserciones: pretty_assertions.
- Alternativas consideradas:
  - totems.
  - galvanic-assert y galvanic-test.

```
   Compiling assertion v0.1.0 (/home/arturo/assertion)
   Finished dev [unoptimized + debuginfo] target(s) in 0.21s
    Running `target/debug/assertion`
thread 'main' panicked at 'assertion failed: `(left == right)`
 left: `Prueba { campo1: "valor del campo 1", campo2: "valor del campo 2", campo3: "valor del
campo 3", campo4: "valor del campo 4" }`,
right: `Prueba { campo1: "valor  erróneo 1", campo2: "valor del campo 2", campo3: "valor del
campo 3", campo4: "valor del campo 4" }`',
```

```
  Compiling assertion v0.1.0 (/home/arturo/assertion)
   Finished dev [unoptimized + debuginfo] target(s) in 0.26s
    Running `target/debug/assertion`
thread 'main' panicked at 'assertion failed: `(left == right)`

Diff < left / right > :
Prueba {
<    campo1: "valor  erróneo 1",
>    campo1: "valor del campo 1",
    campo2: "valor del campo 2",
    campo3: "valor del campo 3",
    campo4: "valor del campo 4"
}
```

Docker

- Imagen base Ubuntu 18.04 LTS.
- La imagen tiene 5 capas, dos más. que la imagen base.
- Peso medio-bajo.
- Imagen subida a docker hub y a github container registry

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
rust	alpine	aeced591196c	3 days ago	642MB
rust	1-slim-buster	e2cf0322c151	3 days ago	681MB
rust	latest	a4b51fc0e875	3 days ago	1.29GB

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
arturocs/crab-iot	latest	23955a504a62	5 seconds ago	678MB

Integración Contínua

- Travis CI
- CircleCI: por su fácil integración con Docker.
- Github Actions: por su flexibilidad
  - Tests en las versiones stable, beta y nightly de Rust, así como la versión mínima.
  - Formateo automático del código.
  - Detección de malas prácticas de Rust.

API REST

Opciones más populares:

- Rocket
- **Actix-web**
- Warp

Logger y configuración distribuida

- log
- simple_logger
- etcd-client

Estructura del proyecto

Device

- Representación de los dispositivos:
  - rdvice.rs Sensores
  - rwdevice.rs Actuadores
- Carga de plugins y su utilización.
- Manejo de listas de dispositivos

```
device
├── Cargo.toml
└── src
    ├── device_list.rs
    ├── lib.rs
    ├── plugin.rs
    ├── rdevice.rs
    └── rwdevice.rs
```

Plugins

- fake_plugin: Plugin falso que simula ser un interruptor.
- weather_fake_plugin: Plugin falso que devuelve el clima contenido en el json.
- weather_plugin: Plugin que se conecta mediante CoAP al dispositivo y recibe información climática.

```
fake_plugin
├── Cargo.toml
├── README.md
└── src
    └── lib.rs

weather_fake_plugin
├── 1769_es.json
├── Cargo.toml
├── README.md
└── src
    └── lib.rs

weather_plugin
├── Cargo.toml
├── README.md
└── src
    └── lib.rs
```

Main

- Programa principal: Tiene carga los plugins y permite interactuar con ellos mediante peticiones rest.

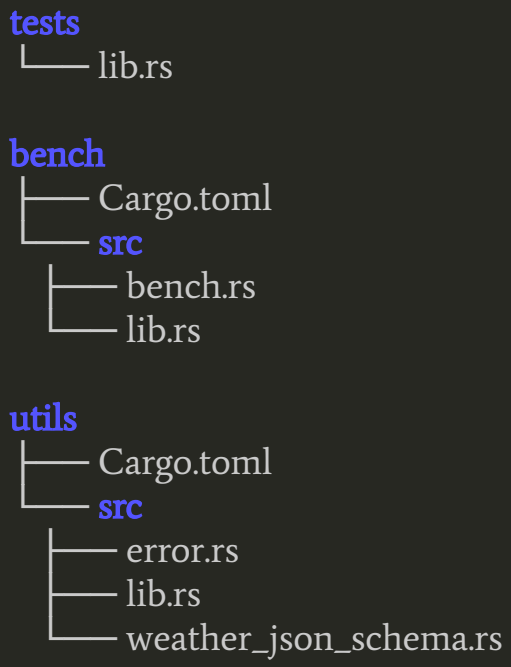
```
src
├── handler.rs
├── main.rs
└── test_routes.rs
```

- mockup_device: Simula ser un termómetro, descarga información climática cada hora y la hace disponible mediante el protocolo CoAP.

```
mockup_device
└── src
    └── main.rs
```

Otros

- Test unitarios
- Benchmarks, principalmente del sistema de plugins.
- Utilidades extra: Tipo de dato error y struct de la información climática.



FIN