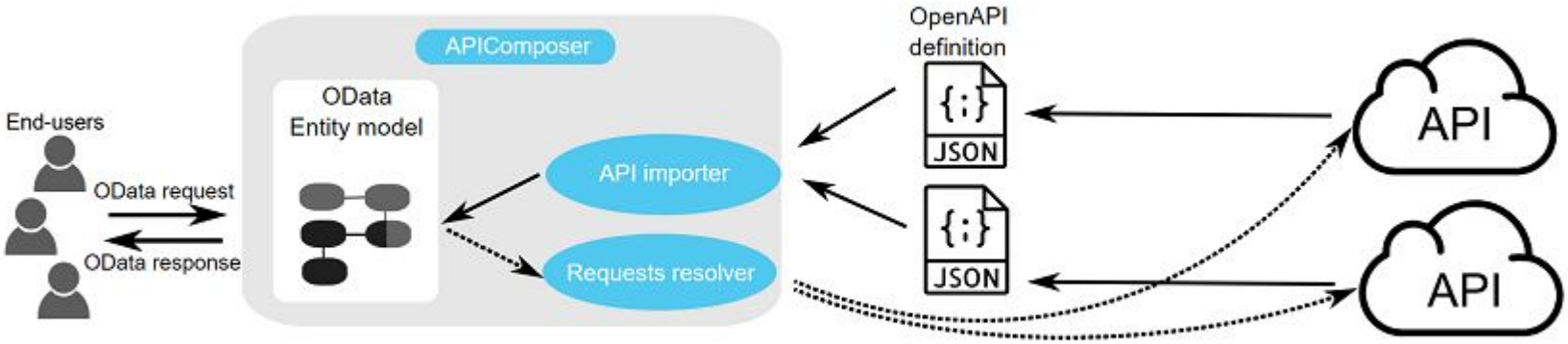
albor

tu software de campo

Api Clima

alborForecast



Una API o interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

¿Qué es Node.js?

Definición de libro: Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

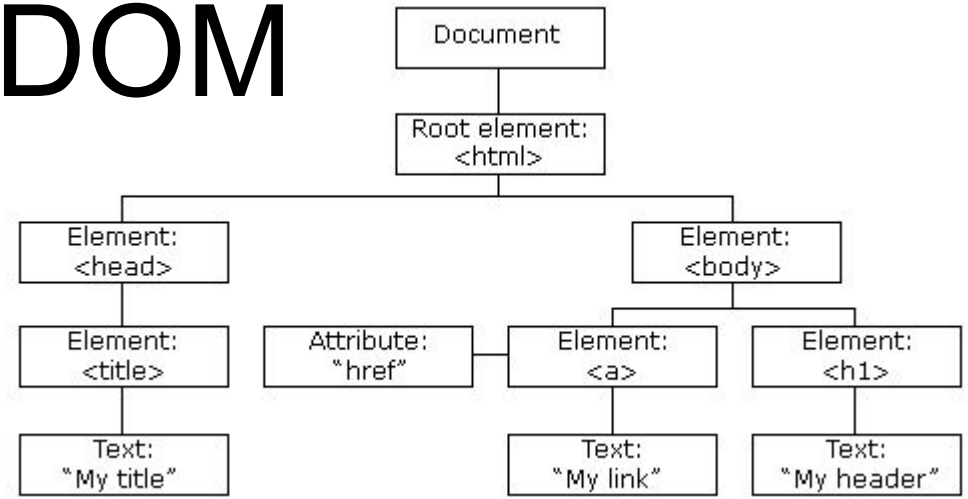
Pero… ¿Qué es?

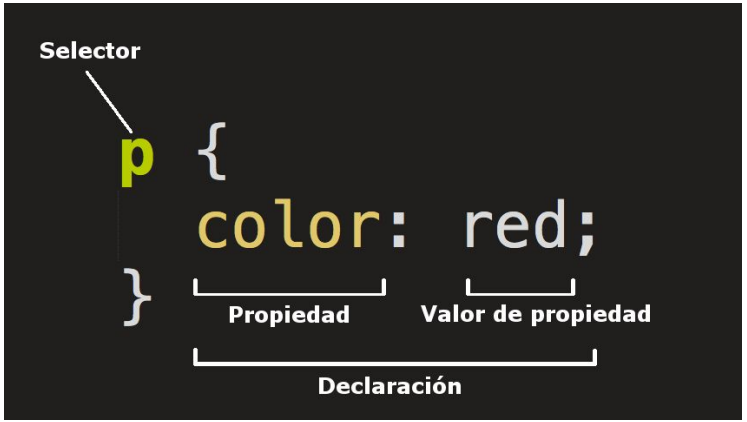

JS


HTML

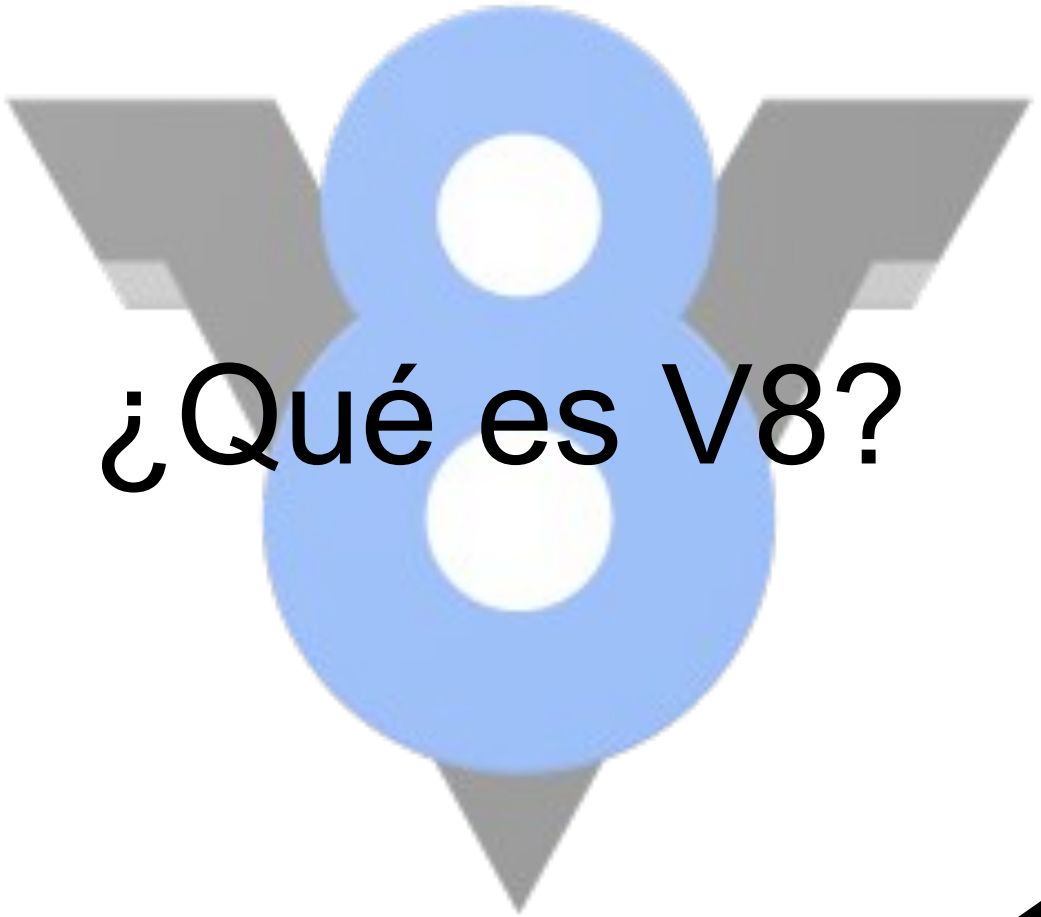

CSS

DOM





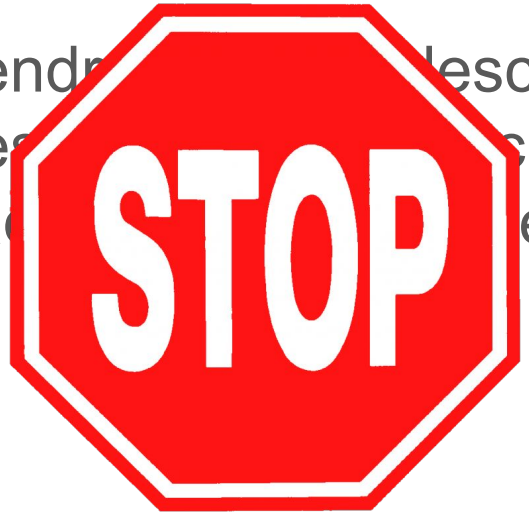

```
let albor = "Hi World";
console.log(albor);
```



¿Qué es V8?

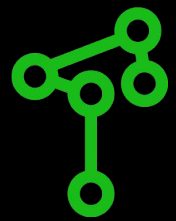
Manos a la obra...

Primero lo primero tendr descargar Node.js, lo podemos hacer des cial. Una vez descargado e instala ezar a jugar un poco…

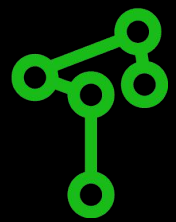


```
C:/>node
> 2 * 3
6
> console.log(“Hello”)
Hello
undefined
> Math.pow(2, 20)
1048576
```

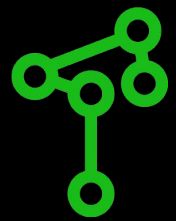
/*Podemos escribir el comando node en una consola y nos abrirá un intérprete de js.*/



```
function derivative_(f, x) {
   const h = 0.001
   return (f(x+h)-f(x))/h
}
function derivative(f, x) {
   return x => derivative_(f, x)
}
const f = x => Math.pow(x, 2)
const df = derivative(f)

//...
```

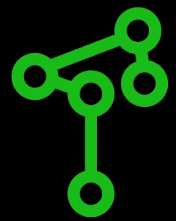


```
//...

for(let x = -5.0; x <= 5.0; x += 0.01) {
   console.log(`f(${x}) = ${f(x)},\
               f'(${x}) = ${df(x)}`)
}

//Ejecutamos node main.js y entonces...
```



```
f(4.769999999999881) = 22.8        f'(4.769999999999881) = 9.54
f(4.77999999999988) = 22.8        f'(4.77999999999988) = 9.56
f(4.78999999999988) = 22.9        f'(4.78999999999988) = 9.58
f(4.79999999999988) = 23.0        f'(4.79999999999988) = 9.60
f(4.80999999999988) = 23.1        f'(4.80999999999988) = 9.62
f(4.8199999999998795) = 23.2        f'(4.8199999999998795) = 9.64
f(4.829999999999879) = 23.3        f'(4.829999999999879) = 9.66
f(4.839999999999879) = 23.4        f'(4.839999999999879) = 9.68
f(4.849999999999879) = 23.5        f'(4.849999999999879) = 9.70
f(4.859999999999879) = 23.6        f'(4.859999999999879) = 9.72
f(4.869999999999878) = 23.7        f'(4.869999999999878) = 9.74
f(4.879999999999878) = 23.8        f'(4.879999999999878) = 9.76
f(4.889999999999878) = 23.9        f'(4.889999999999878) = 9.78
f(4.899999999999878) = 24.0        f'(4.899999999999878) = 9.80
f(4.909999999999878) = 24.1        f'(4.909999999999878) = 9.82
f(4.919999999999877) = 24.2        f'(4.919999999999877) = 9.84
f(4.929999999999877) = 24.3        f'(4.929999999999877) = 9.86
f(4.939999999999877) = 24.4        f'(4.939999999999877) = 9.88
f(4.949999999999877) = 24.5        f'(4.949999999999877) = 9.90
f(4.9599999999998765) = 24.6        f'(4.9599999999998765) = 9.92
f(4.969999999999876) = 24.7        f'(4.969999999999876) = 9.94
f(4.979999999999876) = 24.8        f'(4.979999999999876) = 9.96
f(4.989999999999876) = 24.9        f'(4.989999999999876) = 9.98
f(4.999999999999876) = 25.0        f'(4.999999999999876) = 10.0
```

/*Boom!
Nuestro código
corre en
consola.*/



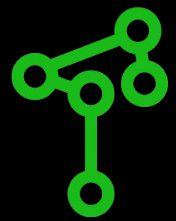
¿Node nos sirve a la hora de escribir una API?

Sí...

Pero vamos a empezar por el principio ¿cómo escribimos nuestro primer servidor?

```
const http = require('http')

const server = http.createserver(
   (req, res) => {
      res.write('<h1>Hello Node.js with http</h1>')
      res.end()
   }
).listen(80);
```

```
/*¿Es eso sólo? Si,
con estas pocas líneas
basta para poder tener
un servidor*/
```



¿Cómo instalamos una librería en Node?

+

Por suerte contamos con una herramienta que viene junto a Node llamada npm, esta vendría a ser el equivalente de lo que en C# es NuGet, es un gestor de paquetes.

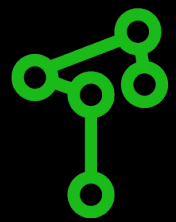
¿Cómo lo usamos? Fácil solo tenemos que ir a una consola y escribir npm install <nombre_del_paquete>, para instalar express sólo tendremos que escribir dentro del directorio de nuestro proyecto los siguientes comandos:

npm init --yes

npm i express

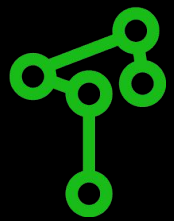
```
{
   “name”: “dev”,
   “version”: “1.0.0”,
   “main”: “server.js”,
   “scripts”: {
   “start”: “node server.js”
   },
   “keywords”: [],
   “author”: “”,
   “license”: “ISC”,
   “dependencies”: {
      “express”: “^4.17.1”
   }
}
```

```
/* Nos genera las
dependencias */
```



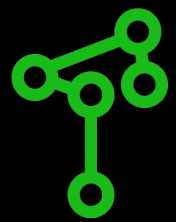
```
const app = require('express')()

app.get('/', (req, res) =>
   res.send('<h1>Hello Node.js with Express</h1>')
)

app.listen(80);
```

```
/*Con estas tres
líneas ya tenemos
nuestro servidor con
Express*/
```

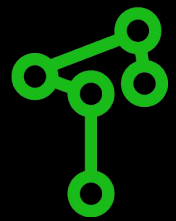


¿Cómo podemos construir nuestra primera API?

```
const express = require('express')()
const app = express()
app.listen(80)
const db = [{}]
app.get('/api', (req, res) =>
   res.json(db)
)
app.post('/api', (req, res) =>
   const { name, dni, vote } = req.body
   if(!name || !dni || !vote)
      res.status(400).send('Invalid
   request\'s body')
   //...
```

```
//...
   db.push({ name, dni, vote })
   res.send('All ok')
})

 /* Aunque simple, esta es nuestra
 primer API, pero ¿Es funcional?...
 Vamos a probarla!*/
```



albor

tu software de campo

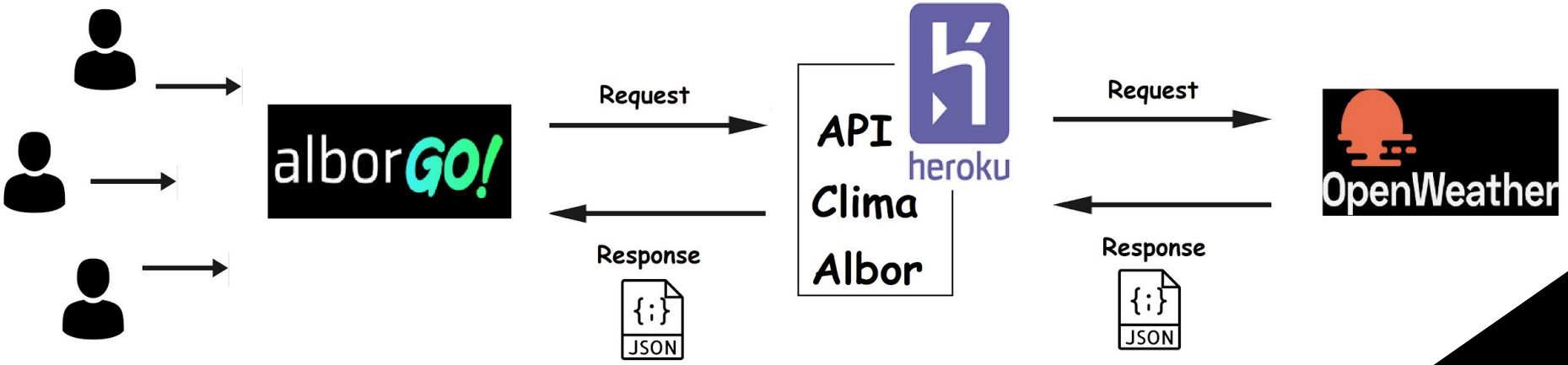
¿Cómo funciona alborForecast?

Situación problemática

El uso de APIs de terceros es algo habitual en la mayoría de apps porque nos facilita no tener grandes gastos iniciales, suelen estar bien soportadas, cualquier fallo suele repararse inmediatamente… pero no todo lo que brilla es oro, no son pocas la veces en las que una API gratuita se vuelve paga.

Nuestra solución

La primera pregunta que nos hicimos fué ¿Qué es común a todas las API climáticas? La respuesta a esto sería la respuesta a conseguir una API que interactúe con esas otras y con la mínima cantidad de cambios, conseguir que la app del cliente no cambie, no requiera actualizar, ni descargar nuevas versiones.


albor GO!
Request
Response
API
Clima
Albor
heroku
Request
Response
OpenWeather
JSON
JSON

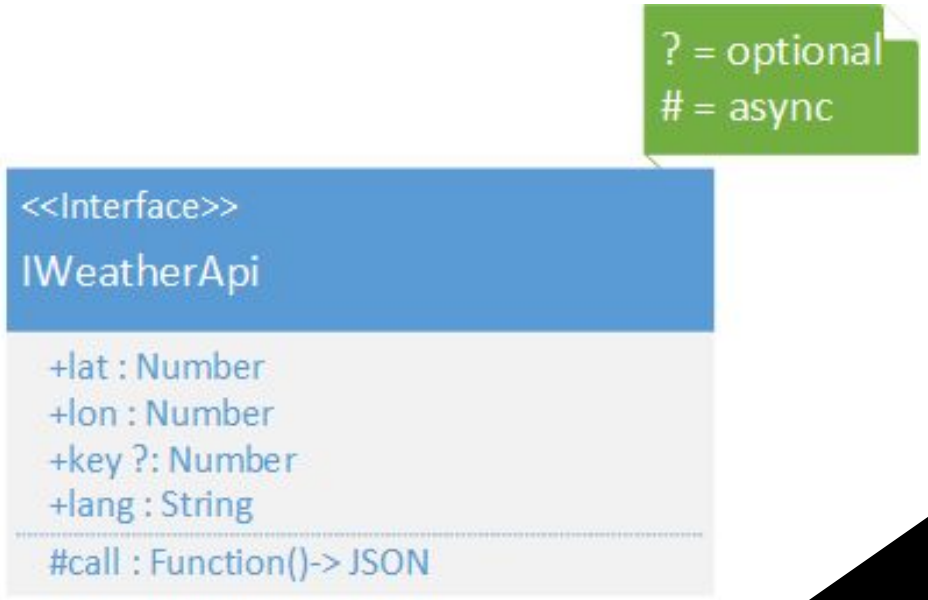
Para uso público

Se cuenta con dos rutas /api que redirecciona a una simple respuesta html que nos indica que el servidor está corriendo. Luego para su uso específico se debe hacer un get a /api.

http://api-clima-albor.herokuapp.com/api/lat&lon&lang&key

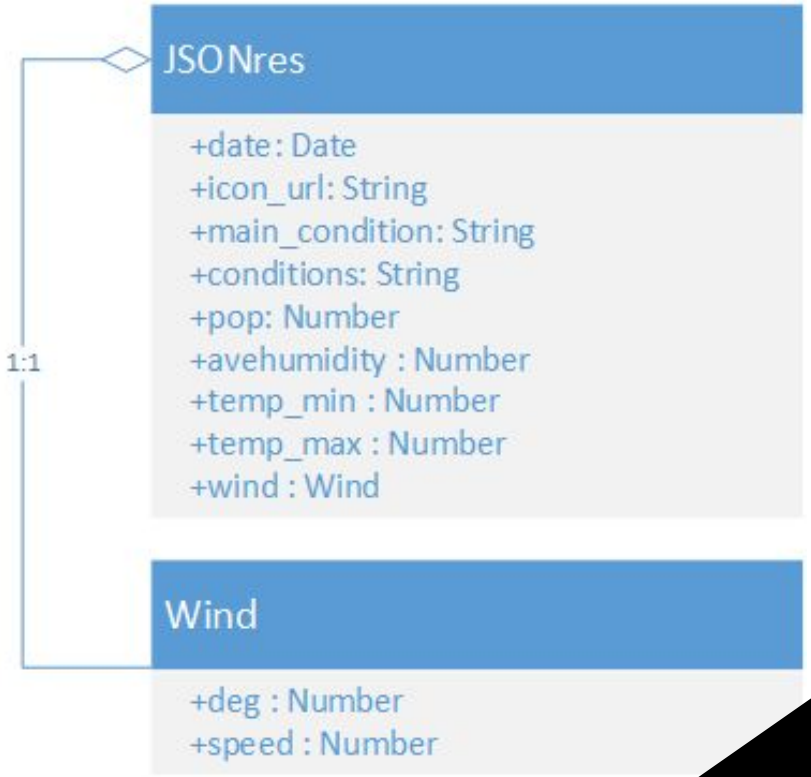
Para uso interno

¿Qué es lo interesante? que con **alborForecast** el único cambio que hay que hacer es agregar la siguiente clase.



Para uso interno

La respuesta JSON que espera la app final debe ser la siguiente.



Recomendaciones

- Una de la principales, que seguimos nosotros en la API del clima fué usar **Typescript**, ayuda un montón a encontrar errores, solucionar fallos, etc.
- Pensar antes de empezar a escribir el código hacer los diagramas de **secuencia**, **clase**, los que necesites para entender cómo, qué, cuándo y dónde tu API tiene que funcionar.

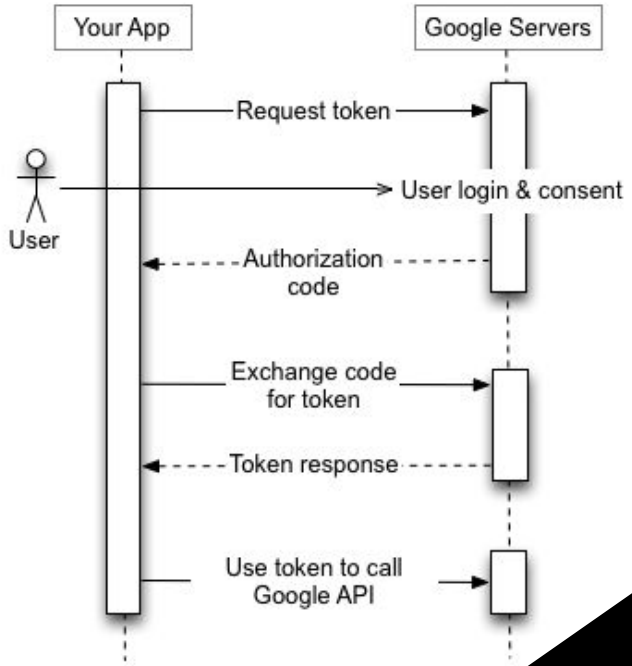
Heroku

Es una plataforma en la nube como servicio (PaaS) que admite varios lenguajes de programación.

¿Qué sigue?

¿Qué es OAuth 2.0?

OAuth 2.0 es el protocolo de autorización estándar de la industria.



Y gracias por el espacio

Diapos y ejemplos:

- https://github.com/Paracefas/aticma-2019

Roman:

- stelleroman@gmail.com
- https://github.com/R0m4n007

Emanuel:

- https://github.com/Paracefas
- emanuelclur5@gmail.com