



**IOT & COGNITIVE COMPUTING** BY JOSE WILKER

---

**A.T.H.E.N.A**

IOT NA PALMA DA MÃO

## BIOGRAFIA



**Desenvolvedor, Empreendedor, Maker e Gamer.** Se envolve com tecnologia há 20 anos, atuando em setores como Educação, Saúde, Telecomunicações, Recursos Humanos entre outros, com ênfase em melhoria de processos e desenvolvimento de soluções para a web.

**ATUAL :** CTO / FOUNDER / SMARTAPPS



---

# A.T.H.E.N.A

A FSM FOR IOT-

UMA MÁQUINA DE ESTADO FINITO PARA  
INTERNET DAS COISAS

# INTRODUÇÃO

**"UMA MÁQUINA DE ESTADO FINITO OU AUTÔMATO FINITO É UM MODELO MATEMÁTICO USADO PARA REPRESENTAR PROGRAMAS DE COMPUTADORES OU CIRCUITOS LÓGICOS."**

**EM INGLÊS, FSM (FINITE STATE MACHINE).**

# O QUE É A.T.H.E.N.A ?

ATHENA É UM ACRÔNIMO PARA...

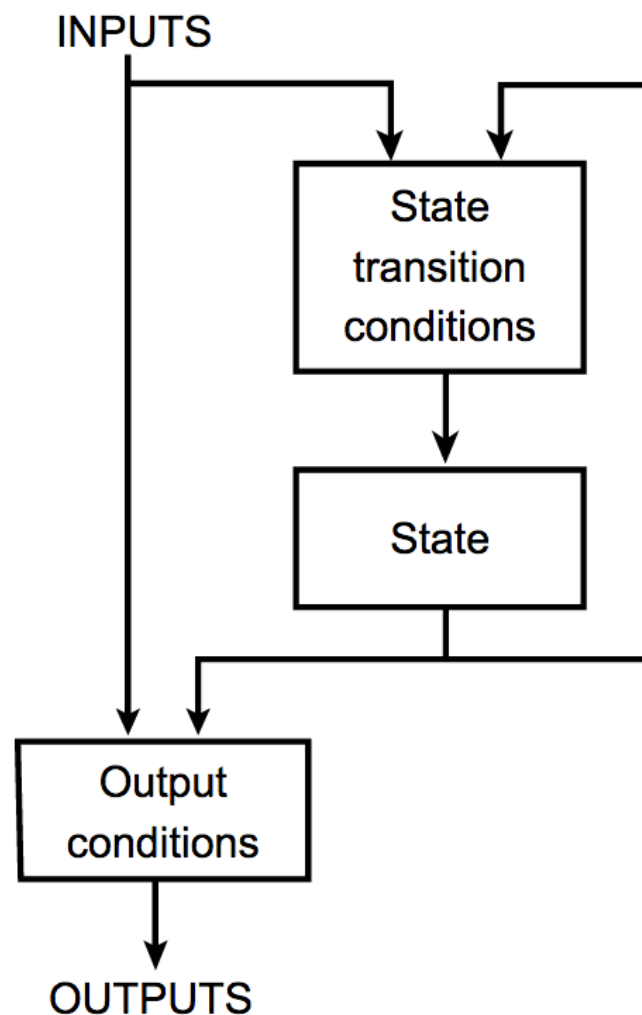
**A**DVANCED **T**INY **H**YPER AND **E**NDLESS **N**ODE OF **A**CTIONS

# O QUE É A.T.H.E.N.A ?

**É UMA REPRESENTAÇÃO EM NODEJS DO MODELO LÓGICO PARA CRIAÇÃO DE UMA MÁQUINA DE ESTADO FINITO COM AÇÕES CUSTOMIZADAS PARA CADA ESTADO.**

# COMO FUNCIONA ?

## DIAGRAMA



## DESCRITIVO

**INPUT (Entrada)** : o que é realizado ao entrar no estado

**OUTPUT (Saída)** : o que é executado ao sair do estado.

**State Transitions / Conditions** : A transição é um conjunto de ações a serem executadas quando uma condição for cumprida ou quando um evento é recebido.

**Output Conditions** : Condições para executar uma determinada saída.

## CASO REAL

**PLAYSTATION! PLAYSTATION! PLAYSTATION! – YUDI**

**1. INPUT (POWER / PLAY / STORE)**

**2. CONDITIONS (SCREEN / LOAD)**

**3. OUTPUT CONDITIONS (LOAD / PLAY / STORE / POWER)**

**4. OUTPUT (SCREEN / –INPUT / –OUTPUT)**





# A.T.H.E.N.A

---

# INSTALAÇÃO

# INSTALAÇÃO (PRIMEIRO PASSO)

- ▶ Intel Edison (Yocto Linux)
- ▶ Install NodeJS
- ▶ Install MRAA
- ▶ Download A.T.H.E.N.A from [git](https://github.com/josewilker/ATHENA).  
(git clone <https://github.com/josewilker/ATHENA>)
- ▶ Install [jhonny-five](#)  
(npm install johnny-five)
- ▶ Install [ip](#) package  
(npm install ip)

## INSTALAÇÃO (SEGUNDO PASSO)

▶ [galileo-io]

git clone <https://github.com/rwaldron/galileo-io>.git **galileo-io**

▶ [es6-shim]

git clone <https://github.com/paulmillr/es6-shim>.git **es6-shim**

▶ [remapped]

git clone <https://github.com/tkellen/js-remapped>.git **remapped**

▶ [traverse]

git clone <https://github.com/substack/js-traverse>.git **traverse**

▶ [getobject]

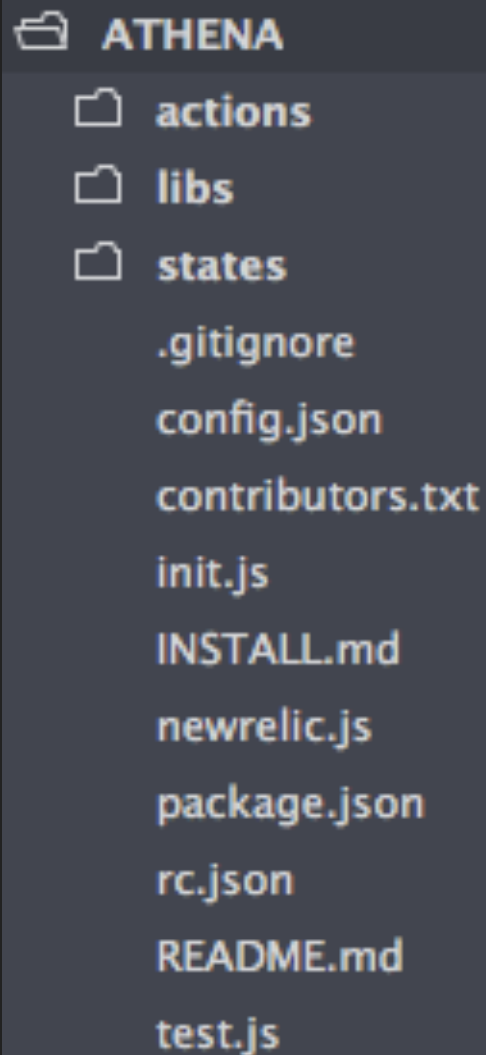
git clone <https://github.com/cowboy/node-getobject>.git **getobject**



---

# CONFIGURAÇÃO & CUSTOMIZAÇÃO

## /CONFIG.JSON



ATHENA

- actions
- libs
- states
- .gitignore
- config.json
- contributors.txt
- init.js
- INSTALL.md
- newrelic.js
- package.json
- rc.json
- README.md
- test.js

```
1
2 {
3   "name" : "A.T.H.E.N.A",
4   "version" : "1.0",
5   "socket" : {
6     "port" : 1884 // layer, it's the first action.
7   },
8   "hw" : {
9     // array of hardwares
10  },
11  "actionDefault" : "", // when machine starts, it's the first action.
12  "actions" : {
13    // array of actions
14  }
15 }
```

## /CONFIG.JSON : HARDWARES

**TINY**  
LCD, RELAY, LED

```
"hw" : {  
  "lcd" : {  
    "id" : "JHD1313M1"  
  },  
  "led" : {  
    "0" : {  
      "obj" : "activity",  
      "pname" : "LED Blue",  
      "pin" : 4  
    },  
    "1" : {  
      "obj" : "error",  
      "pname" : "LED Red",  
      "pin" : 2  
    }  
  }  
},
```

```
"hw" : { // hardware nodes, all objects can be loaded with htriggers object.  
  "button" : { // hardware type  
    "0" : {  
      "obj" : "one", // if need access this button in a action  
      "pname" : "B.Select 1", // a text name for the button  
      "pin" : 8, // pin of GPIO  
      "file" : "btn0.js" // file of state flow for this button  
    },  
    "1" : {  
      "obj" : "two",  
      "pname" : "B.Select 2",  
      "pin" : 7,  
      "file" : "btn1.js",  
      "action" : [{ // work with states and call action in a specific time.  
        "times" : 3,  
        "name" : "athenaIp.ip"  
      }]  
    }  
  }  
},
```

**ADVANCED**  
BUTTONS

## /CONFIG.JSON : ACTIONS

```
ATHENA
├── actions
│   ├── deploy
│   ├── didact
│   ├── flow
│   ├── greetings
│   ├── ip
│   ├── lights
│   ├── mtemp
│   └── songs
├── libs
├── states
├── .gitignore
├── config.json
├── contributors.txt
├── init.js
├── INSTALL.md
├── newrelic.js
├── package.json
├── rc.json
├── README.md
└── test.js
```

### CARREGANDO AÇÕES

```
"actions" : { // actions that you want load
  "0" : {
    "text" : "Temperatura",
    "action" : "mtemp.js",
    "lib" : "mtemp"
  },
  "1" : {
    "text" : "Endereço IP",
    "lib" : "ip",
    "action" : "ip.js"
  },
  "2" : {
    "text" : "Sons",
    "lib" : "songs",
    "action" : "songs.js"
  }
}
```





# CUSTOMIZAÇÃO

---

## AÇÕES & ESTADOS



## /ACTIONS/IP.JS

### CRIANDO AÇÃO

```
1  athenaIp = {}
2
3  athenaIp.ip = false;
4
5  athenaIp.config = function(build) {
6    athenaIp.ip = require('ip');
7  }
8
9  athenaIp.run = function(context, events) {
10
11    events.speakClear();
12
13    if (athena.ip !== false) {
14      events.speak(0, "MY IP: :heart: :duck: :check:");
15      events.speak(1, athenaIp.ip.address());
16    }
17  }
18
19  module.exports = athenaIp;
```

## /STATES/DEFAULT.JS

### INPUT

```
htriggers.queue.states['temp'] = [];  
htriggers.queue.states['temp']['input'] = function() {  
  
    htriggers.led.on(htriggers.oled.activity);  
  
    events.speak(0, " > CHECAR          ");  
    events.speak(1, " > TEMPERATURA      ");  
  
    htriggers.led.off(htriggers.oled.activity);  
  
    wdactions.io.didact.obj.talk(this, events, "Deseja saber a temperatura do ambiente ?", 1);  
  
    athena.wait(5);  
  
};
```

## /STATES/DEFAULT.JS

### OUTPUT

```
htriggers.queue.states['temp']['output'] = function() {  
    atualTemp = wdactions.io.mtemp.obj.getValue(this, events);  
  
    events.speak(0,"TEMPERATURA :heart:");  
    events.speak(1,"AMBIENTE: " + atualTemp);  
  
    wdactions.io.didact.obj.talk(this, events, "A temperatura do ambiente é " + atualTemp + " graus celsius", 1);  
  
    athena.wait(5);  
  
    events.speakClear();  
  
    htriggers.queue.states['index']['input']();  
};
```

## /STATES/DEFAULT.JS

### OUTPUT

```
htriggers.queue.states['temp']['output'] = function() {  
    atualTemp = wdactions.io.mtemp.obj.getValue(this, events);  
  
    events.speak(0,"TEMPERATURA :heart:");  
    events.speak(1,"AMBIENTE: " + atualTemp);  
  
    wdactions.io.didact.obj.talk(this, events, "A temperatura do ambiente é " + atualTemp + " graus celsius", 1);  
  
    athena.wait(5);  
  
    events.speakClear();  
  
    htriggers.queue.states['index']['input']();  
};
```



---

# EVENTOS GLOBAIS & FLUXOS DE TRABALHO

# EVENTOS GLOBAIS

/LIBS/**EVENTS**.JS

```
events = {}

events.speak = function(l,m) {
  athenaHWT.lcd.write(l,0,m);
}

events.speakClear = function() {
  events.speak(0,"");
  events.speak(1,"");
}

events.buzz = function() {
  var beethoven = songs.load("beethovens-fifth");
  buzzer.play(beethoven);
}

module.exports = events;
```

# HARDWARE : FLUXO DE TRABALHO

/LIBS/{HARDWARE-FILE}.JS

```
module.exports = function(objThis, objButton, objLed){  
    athenaHWT.button.flow(objButton.one, 1000, function(c, htriggers){  
        // input state  
    }, function(c){  
        // output state  
    });  
}
```



---

**NOW!MAGIC**



# ENVIE COMANDOS PARA SUA MÁQUINA REMOTAMENTE PELA REDE!

MQTT:PORT

TOPIC

RESPONSE

/RC.JSON

```
[
  {
    "topic" : "show.ip",
    "cmd" : "_ip.run(this, events)",
    "wait" : "10"
  },
  {
    "topic" : "songs.mario",
    "cmd" : "_songs.mario(this, events)",
    "wait" : "2"
  },
  {
    "topic" : "songs.starwars",
    "cmd" : "_songs.sw(this, events)",
    "wait" : "2"
  },
  {
    "topic" : "greetings.begin",
    "cmd" : "_greetings.begin(this, events)",
    "wait" : "2"
  }
]
```



# ENTÃO...



# VOCÊ PODE UTILIZAR PARA...

- ▶ Controlar dispositivo eletroeletrônico remotamente.  
(Exemplo: Máquina de café, TV e etc...)
- ▶ Verifique estados de sensores remotamente.  
(Sensor de temperatura, Umidade, Luminosidade e etc...)
- ▶ Ligar ou desligar lâmpadas.  
(Relés)
- ▶ Enviar emails.
- ▶ Atualizar aplicações.  
(deploy automatizado)
- ▶ Monitorar redes sociais.
- ▶ Rede privada de mensagens.
- ▶ Verificar o tempo atual e muito mais...



# OBRIGADO!

---

**JOSÉ WILKER**

[SMARTAPPS.COM.BR](http://SMARTAPPS.COM.BR) / [WILKER.COM.BR](http://WILKER.COM.BR)

TWITTER	<a href="#">/JOSEWILKER</a>
LINKEDIN	<a href="#">/JOSEWILKERDEV</a>
GITHUB	<a href="#">/JOSEWILKER</a>

**ATHENA**  
[GITHUB.COM/JOSEWILKER/ATHENA](https://github.com/josewilker/athena)

