

# ARDUINO NA WEB

— CONTROLANDO SEU ARDUINO COM BOTS —

A photograph of two young women, likely students, working together on a project. They are leaning over a table, looking at an Arduino board connected to a laptop. The woman on the left is wearing a dark sweater and a grey scarf, and the woman on the right is wearing a blue shirt. They are both focused on the task. The background is slightly blurred, showing what appears to be a workshop or classroom setting.

Cientista da Computação

Engenheira de Software na HP Inc.

Oficinas de Arduino

/Maratona Maker Intel

/Arduino4Girls

/FISL

# CONTEXTO

INTERNET DAS  
COISAS?

---

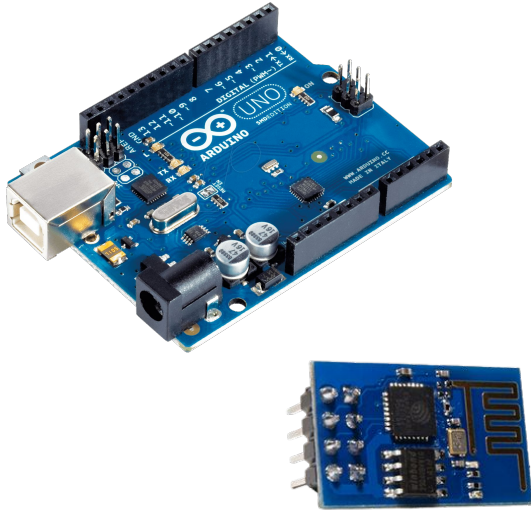
O QUE?  
PRA QUÊ?

---

# ESTRUTURA

---

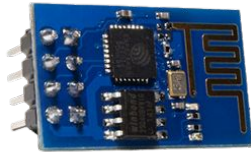
# ESTRUTURA



# ESTRUTURA

Conecta no WiFi e gera um IP local

---



# ESTRUTURA

Conecta no WiFi e gera um IP local



Túnel



# ESTRUTURA

Conecta no WiFi e gera um IP local

---



Expõe o IP local para acesso público

---

Túnel





# ESTRUTURA

Conecta no WiFi e gera um IP local

---



Expõe o IP local para acesso público

---

Túnel

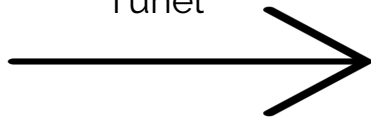


# ESTRUTURA

IP Local: 192.168.0.36



Túnel

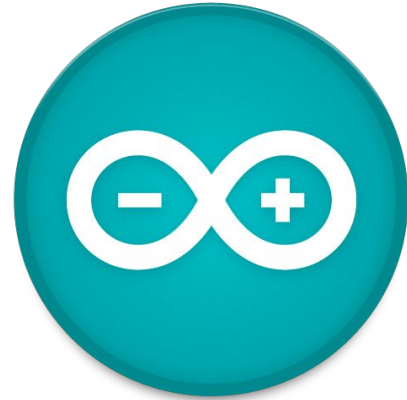
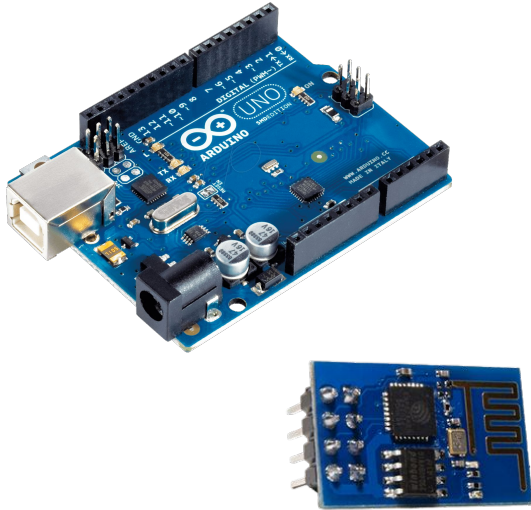


Recebe o GET  
e faz ação

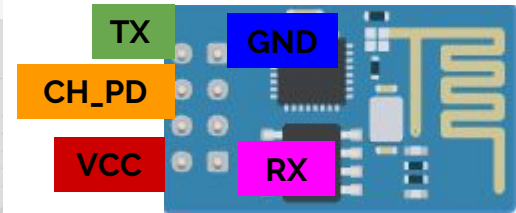
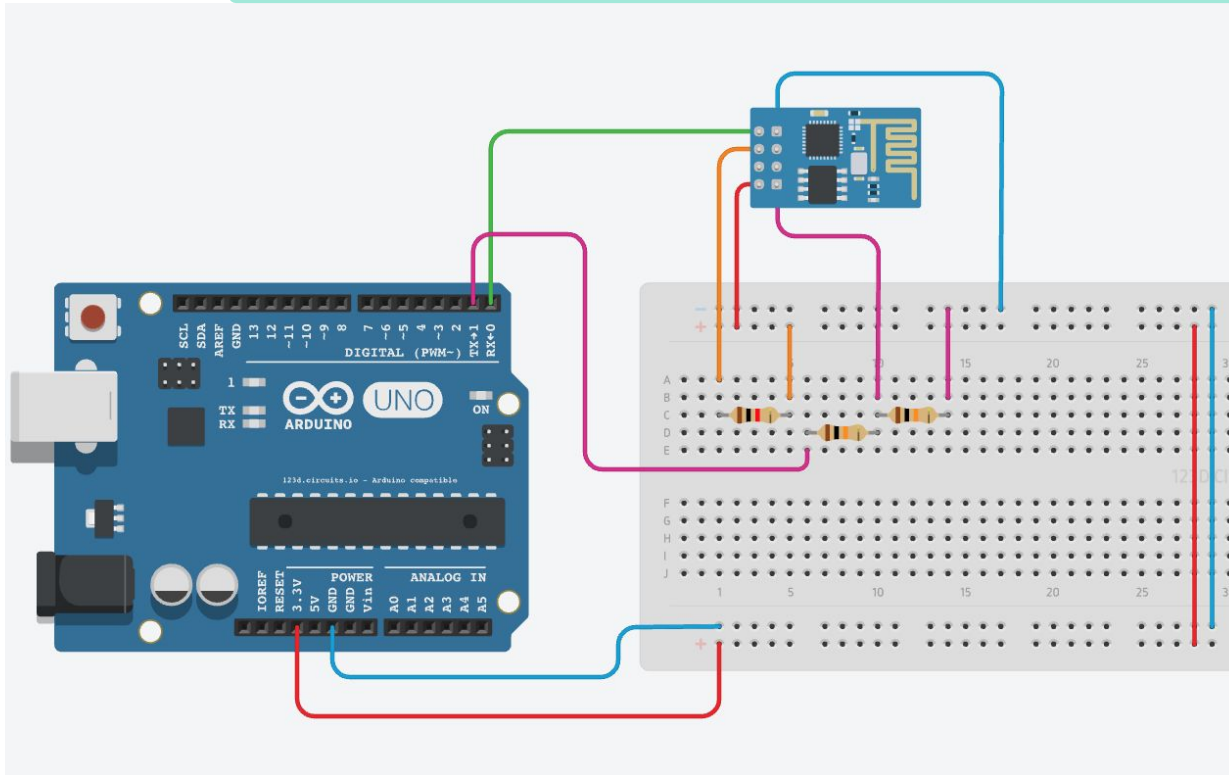


HTTP GET

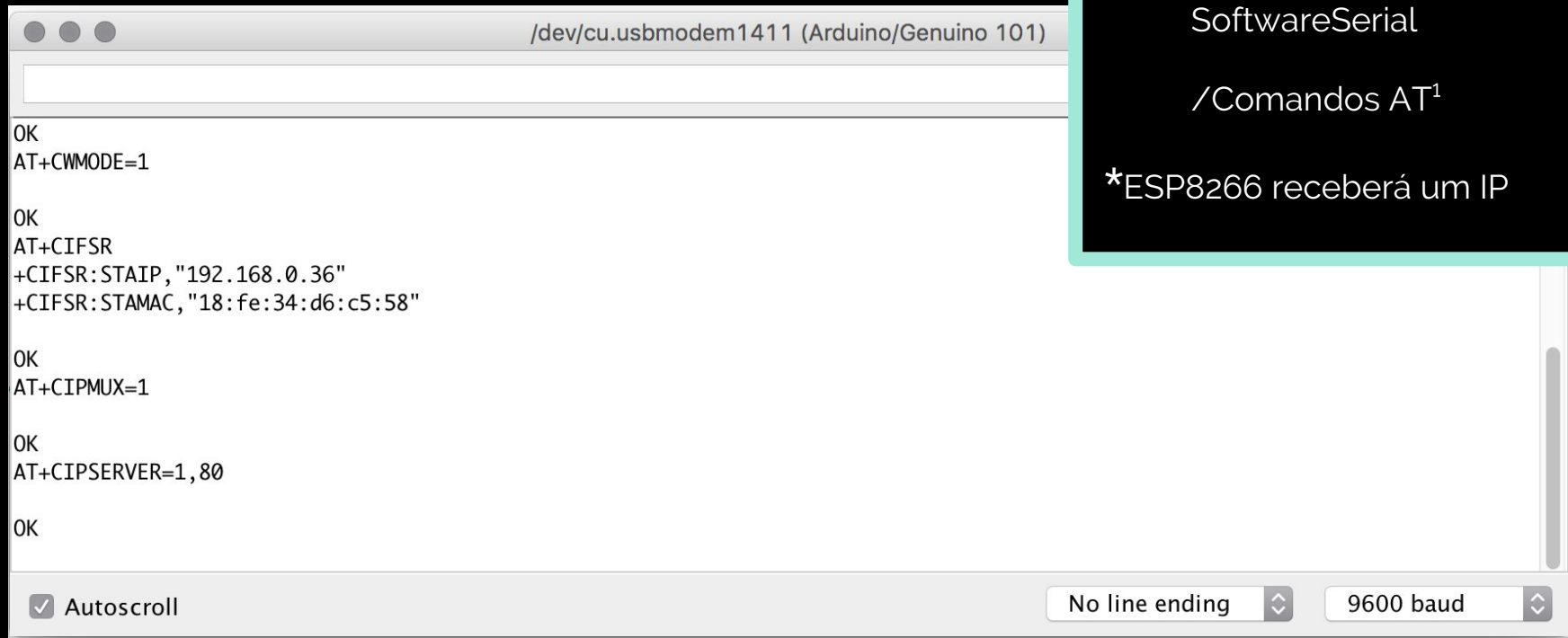
# ARDUINO + ESP8266



# ARDUINO + ESP8266



# INICIALIZANDO O SERVIDOR



The screenshot shows a serial monitor window titled "/dev/cu.usbmodem1411 (Arduino/Genuino 101)". The window displays the following AT commands and responses:

```
OK
AT+CWMODE=1

OK
AT+CIFSR
+CIFSR: STAIP, "192.168.0.36"
+CIFSR: STAMAC, "18:fe:34:d6:c5:58"

OK
AT+CIPMUX=1

OK
AT+CIPSERVER=1,80

OK
```

At the bottom of the window, there are settings: "Autoscroll" is checked, "No line ending" is selected, and "9600 baud" is selected.

\*Comunicação com a ESP8266

/pelos pinos RX, TX com  
a biblioteca  
SoftwareSerial

/Comandos AT<sup>1</sup>

\*ESP8266 receberá um IP

# ACESSANDO O SERVIDOR

\*Acessar pelo Browser

/HTTP GET

/Action como parâmetro



## ESP8266 - Web Server

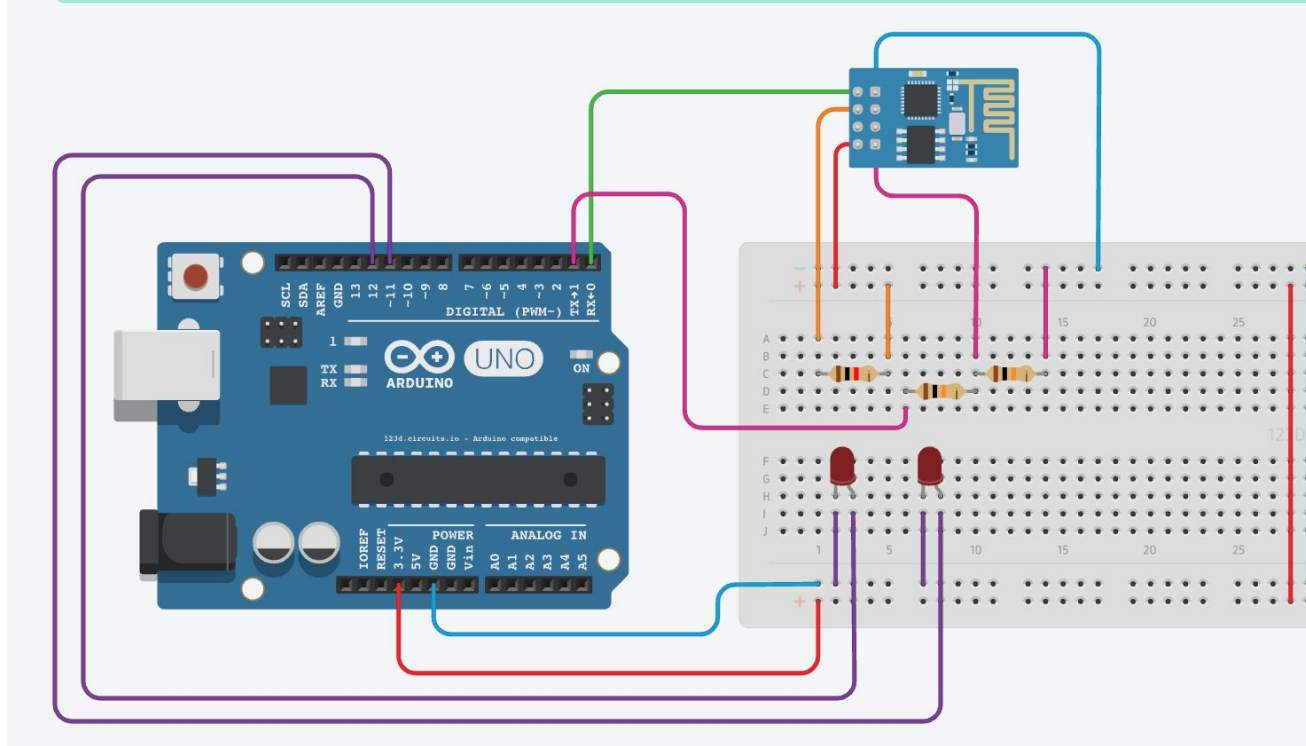
**Action: 0**

```
ESPWebServer

//set cursor to the word "action=" and gets its value
// 1 to turn on, 0 to turn off
esp8266.find("action=");
int action = esp8266.read()-48;

// displays in web page the status
String webpage = "<head><meta http-equiv=\"\"refresh\"\" content=\"\"3\"\">";
webpage += "</head><h1><u>ESP8266 - Web Server</u></h1>";
webpage += "<h2>Action: ";
webpage += action;
webpage += "</h2>";
```

# ADICIONANDO LEDs



# TRATANDO OS COMANDOS

\*Action: 1 ou 0

\*Pin: 11 ou 12

```
//set cursor to the word "action=" and gets its value
// 1 to turn on, 0 to turn off
esp8266.find("action=");
int action = esp8266.read()-48;

esp8266.find("pin=");
int pin = (esp8266.read()-48)*10;
pin += esp8266.read()-48;

//sends the action to defined pin
digitalWrite(pin, action);
```



TUNEL



# EXPONDO O LOCALHOST

\*NGROK

/ngrok http  
192.168.0.36:80

```
taisbellini001 — ngrok http 192.168.0.36:80 — 80x24
ngrok by @inconshreveable (Ctrl+C to quit)

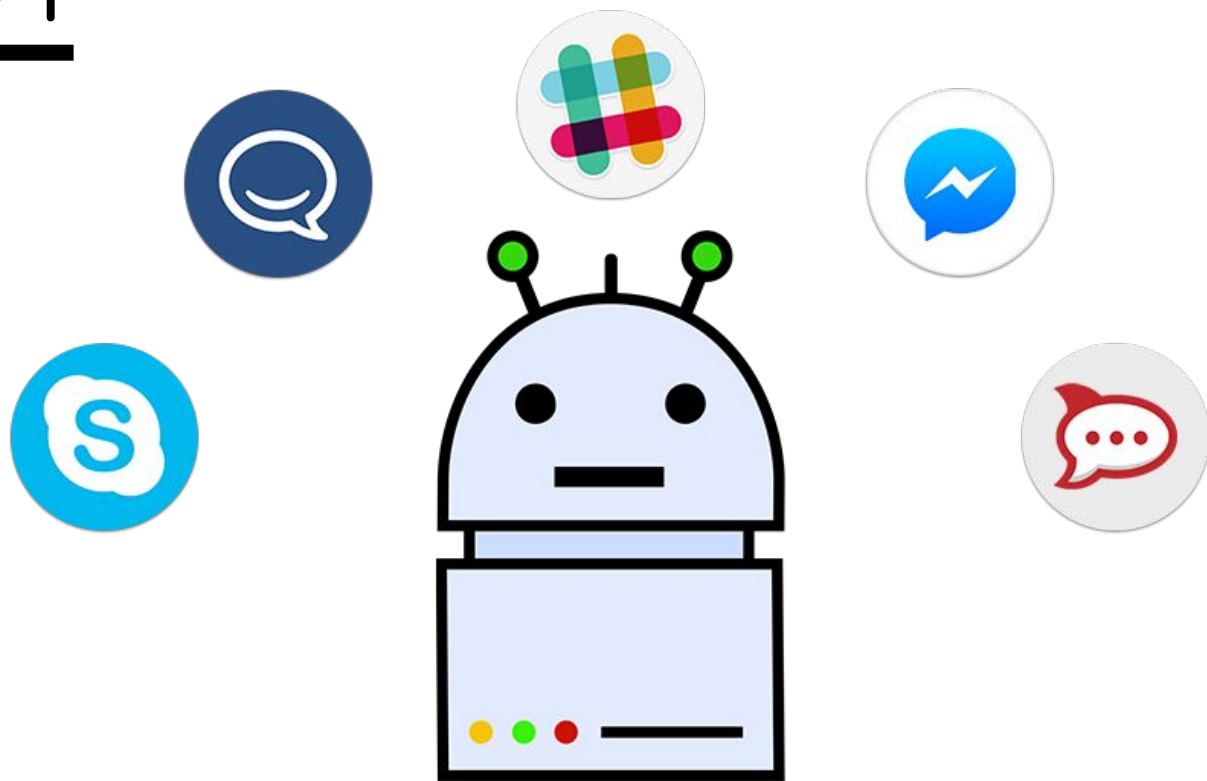
Session Status      online
Version             2.1.18
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding            http://0fa482ed.ngrok.io -> 192.168.0.36:80
Forwarding            https://0fa482ed.ngrok.io -> 192.168.0.36:80

Connections          ttl      opn      rt1      rt5      p50      p90
26                0        0.00     0.00     69.05    300.10

HTTP Requests
-----
GET /favicon.ico
GET /
GET /favicon.ico
```

URL: <https://0fa482ed.ngrok.io/?action=1&pin=12>

# BOT



# FACEBOOK | CRIAR PÁGINA E APP

Página

Mensagens <sup>1</sup>

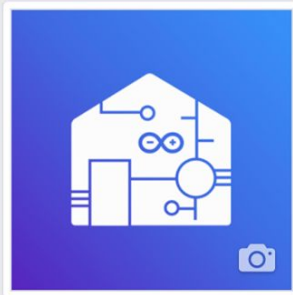
Notificações

Informações

Ferramentas de publicação

Configurações

Ajuda ▾



**MyHome**

Criar @nome de usuário da  
Página


Página inicial


Publicações

Avaliações

Fotos

Loja

 Adicionar uma capa

 Curtir

 Seguir

 Compartilhar

...

+ Adicionar um botão

Bem-vinda à sua nova Página

Antes de compartilhar sua Página com outras pessoas,  
verifique essas dicas que ajudam você a descrever seu

Conhece amigos que podem curtir sua Página?

Convidar amigos para curtir sua Página ajuda a  
atrair outras pessoas.

# WEBHOOK | CÓDIGO

```
7 // The rest of the code implements the routes for our Express server.
8 let app = express();
9
10 app.use(bodyParser.json());
11 app.use(bodyParser.urlencoded({
12   extended: true
13 }));
14
15 // Webhook validation
16 app.get('/webhook', function(req, res) {
17   if (req.query['hub.mode'] === 'subscribe' &&
18       req.query['hub.verify_token'] === 'myhometoken') {
19     console.log("Validating webhook");
20     res.status(200).send(req.query['hub.challenge']);
21   } else {
22     console.error("Failed validation. Make sure the validation tokens match.");
23     res.sendStatus(403);
24   }
25 });
26
```

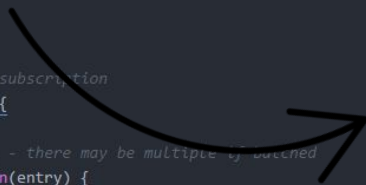
# WEBHOOK | CÓDIGO

```
27 // Message processing
28 app.post('/webhook', function (req, res) {
29   console.log(req.body);
30   var data = req.body;
31
32   // Make sure this is a page subscription
33   if (data.object === 'page') {
34
35     // Iterate over each entry - there may be multiple if batched
36     data.entry.forEach(function(entry) {
37       var pageID = entry.id;
38       var timeOfEvent = entry.time;
39
40       // Iterate over each messaging event
41       entry.messaging.forEach(function(event) {
42         if (event.message) {
43           receivedMessage(event);
44         } else {
45           console.log("Webhook received unknown event: ", event);
46         }
47       });
48     });
49
50     // Assume all went well.
51     //
52     // You must send back a 200, within 20 seconds, to let us know
53     // you've successfully received the callback. Otherwise, the request
54     // will time out and we will keep trying to resend.
55     res.sendStatus(200);
56   }
57 });
```

```
59 // Incoming events handling
60 function receivedMessage(event) {
61   var senderID = event.sender.id;
62   var recipientID = event.recipient.id;
63   var timeOfMessage = event.timestamp;
64   var message = event.message;
65
66   console.log("Received message for user %d and page %d at %d with message:",
67     senderID, recipientID, timeOfMessage);
68   console.log(JSON.stringify(message));
69
70   var messageId = message.mid;
71
72   var messageText = message.text;
73
74   let pin = 12;
75   if (messageText.includes('red')){
76     pin = 11;
77   }
78   let action = 0;
79   if (messageText.includes('on')) {
80     action = 1;
81   }
82   const params = {
83     url: 'https://c22a7e44.ngrok.io',
84     qs: {
85       action: action,
86       pin: pin
87     }
88   }
89   request.get(params, (err, httpResponse, body) => {
90     console.log('err', err);
91     console.log('response', httpResponse);
92     console.log('body', body);
93     sendTextMessage(senderID, 'Done!');
94   });
95 }
```

# WEBHOOK | CÓDIGO

```
27 // Message processing
28 app.post('/webhook') function (req, res) {
29   console.log(req.body);
30   var data = req.body;
31
32   // Make sure this is a page subscription
33   if (data.object === 'page') {
34
35     // Iterate over each entry - there may be multiple if batched
36     data.entry.forEach(function(entry) {
37       var pageID = entry.id;
38       var timeOfEvent = entry.time;
39
40       // Iterate over each messaging event
41       entry.messaging.forEach(function(event) {
42         if (event.message) {
43           receivedMessage(event);
44         } else {
45           console.log("Webhook received unknown event: ", event);
46         }
47       });
48     });
49
50     // Assume all went well.
51     //
52     // You must send back a 200, within 20 seconds, to let us know
53     // you've successfully received the callback. Otherwise, the request
54     // will time out and we will keep trying to resend.
55     res.sendStatus(200);
56   }
57 });
```




HTTP POST para  
<https://domain.com/webhook>

```
59 // Incoming events handling
60 function receivedMessage(event) {
61   var senderID = event.sender.id;
62   var recipientID = event.recipient.id;
63   var timeOfMessage = event.timestamp;
64   var message = event.message;
65
66   console.log("Received message for user %d and page %d at %d with message:",
67     senderID, recipientID, timeOfMessage);
68   console.log(JSON.stringify(message));
69
70   // Message type is 'page' and the page ID
71   var pageID = message.page.id;
72   var messageText = message.text;
73
74   let pin = 12;
75   if (messageText.includes('red')){
76     pin = 11;
77   }
78   let action = 0;
79   if (messageText.includes('on')) {
80     action = 1;
81   }
82   const params = {
83     url: 'https://c22a7e44.ngrok.io',
84     qs: {
85       action: action,
86       pin: pin
87     }
88   }
89   request.get(params, (err, httpResponse, body) => {
90     console.log('err', err);
91     console.log('response', httpResponse);
92     console.log('body', body);
93     sendTextMessage(senderID, 'Done!');
94   });
95 }
```

# WEBHOOK | CÓDIGO

```
27 // Message processing
28 app.post('/webhook', function (req, res) {
29   console.log(req.body);
30   var data = req.body;
31
32   // Make sure this is a page subscription
33   if (data.object === 'page') {
34
35     // Iterate over each entry - typically up to 10 per webhook received
36     data.entry.forEach(function(entry) {
37       var pageID = entry.id;
38       var timeOfEvent = entry.time;
39
40       // Iterate over each message in the event
41       entry.messaging.forEach(function(event) {
42         if (event.message) {
43           receivedMessage(event);
44         } else {
45           console.log("Webhook received unknown event: ", event);
46         }
47       });
48     });
49
50     // Assume all went well.
51     //
52     // You must send back a 200, within 20 seconds, to let us know
53     // you've successfully received the callback. Otherwise, the request
54     // will time out and we will keep trying to resend.
55     res.sendStatus(200);
56   }
57 });
```

Lógica para  
seleccionar o  
pino



```
59 // Incoming events handling
60 function receivedMessage(event) {
61   var senderID = event.sender.id;
62   var recipientID = event.recipient.id;
63   var timeOfMessage = event.timestamp;
64   var message = event.message;
65
66   console.log("Received message for user %d and page %d at %d with message:",
67     senderID, recipientID, timeOfMessage);
68   console.log(JSON.stringify(message));
69
70   var messageId = message.mid;
71
72   var messageText = message.text;
73
74   let pin = 12;
75   if (messageText.includes('red')){
76     pin = 11;
77   }
78   let action = 0;
79   if (messageText.includes('on')) {
80     action = 1;
81   }
82   const params = {
83     url: 'https://c22a7e44.ngrok.io',
84     qs: {
85       action: action,
86       pin: pin
87     }
88   }
89   request.get(params, (err, httpResponse, body) => {
90     console.log('err', err);
91     console.log('response', httpResponse);
92     console.log('body', body);
93     sendTextMessage(senderID, 'Done!');
94   });
95 }
```



# WEBHOOK | CÓDIGO

```
27 // Message processing
28 app.post('/webhook', function (req, res) {
29   console.log(req.body);
30   var data = req.body;
31
32   // Make sure this is a page subscription
33   if (data.object === 'page') {
34
35     // Iterate over each entry - there may be multiple if batched
36     data.entry.forEach(function(entry) {
37       var pageID = entry.id;
38       var timeOfEvent = entry.time;
39
40       // Iterate over each messaging event
41       entry.messaging.forEach(function(event) {
42         if (event.message) {
43           receivedMessage(event);
44         } else {
45           console.log('Unknown event type: ' + event.type);
46         }
47       });
48     });
49
50     // Assume all went well
51     // If you've successfully received the callback, otherwise, the request
52     // will be rescheduled and we will keep trying to resend.
53     // https://api.slack.com/docs/webhooks
54     res.sendStatus(200);
55   }
56 });
```

Request para o **ngrok**,  
no qual está rodando  
a ESP8266, com  
informações de pino e  
ação.

```
59 // Incoming events handling
60 function receivedMessage(event) {
61   var senderID = event.sender.id;
62   var recipientID = event.recipient.id;
63   var timeOfMessage = event.timestamp;
64   var message = event.message;
65
66   console.log("Received message for user %d and page %d at %d with message:",
67     senderID, recipientID, timeOfMessage);
68   console.log(JSON.stringify(message));
69
70   var messageId = message.mid;
71
72   var messageText = message.text;
73
74   let pin = 12;
75   if (messageText.includes('red')){
76     pin = 11;
77   }
78   let action = 0;
79   if (messageText.includes('on')) {
80     action = 1;
81   }
82   const params = {
83     url: 'https://c22a7e44.ngrok.io',
84     qs: {
85       action: action,
86       pin: pin
87     }
88   }
89   request.get(params, (err, httpResponse, body) => {
90     console.log('err', err);
91     console.log('response', httpResponse);
92     console.log('body', body);
93     sendTextMessage(senderID, 'Done!');
94   });
95 }
```

# WEBHOOK | CÓDIGO

Restart

 MyHome

Painel

Configurações

Funções

Alertas

Revisão do Aplicativo

PRODUTOS

Messenger

Webhooks

+ Adicionar produto

ID do APLICATIVO: 1434260933547713

Visualizar o Analytics

Ferramentas

Painel de controle



MyHome

Este aplicativo está em modo de desenvolvimento e só pode ser usado por administradores, desenvolvedores e testadores [?]

Versão da API [?]

ID do Aplicativo

v2.8

1434260933547713

Chave Secreta do Aplicativo

.....

Mostrar

 Comece com o SDK do Facebook

Use our quick start guides to set up the Facebook SDK for your iOS or Android app, Facebook Web Game or website.

Escolher plataforma

Facebook Analytics para Aplicativos

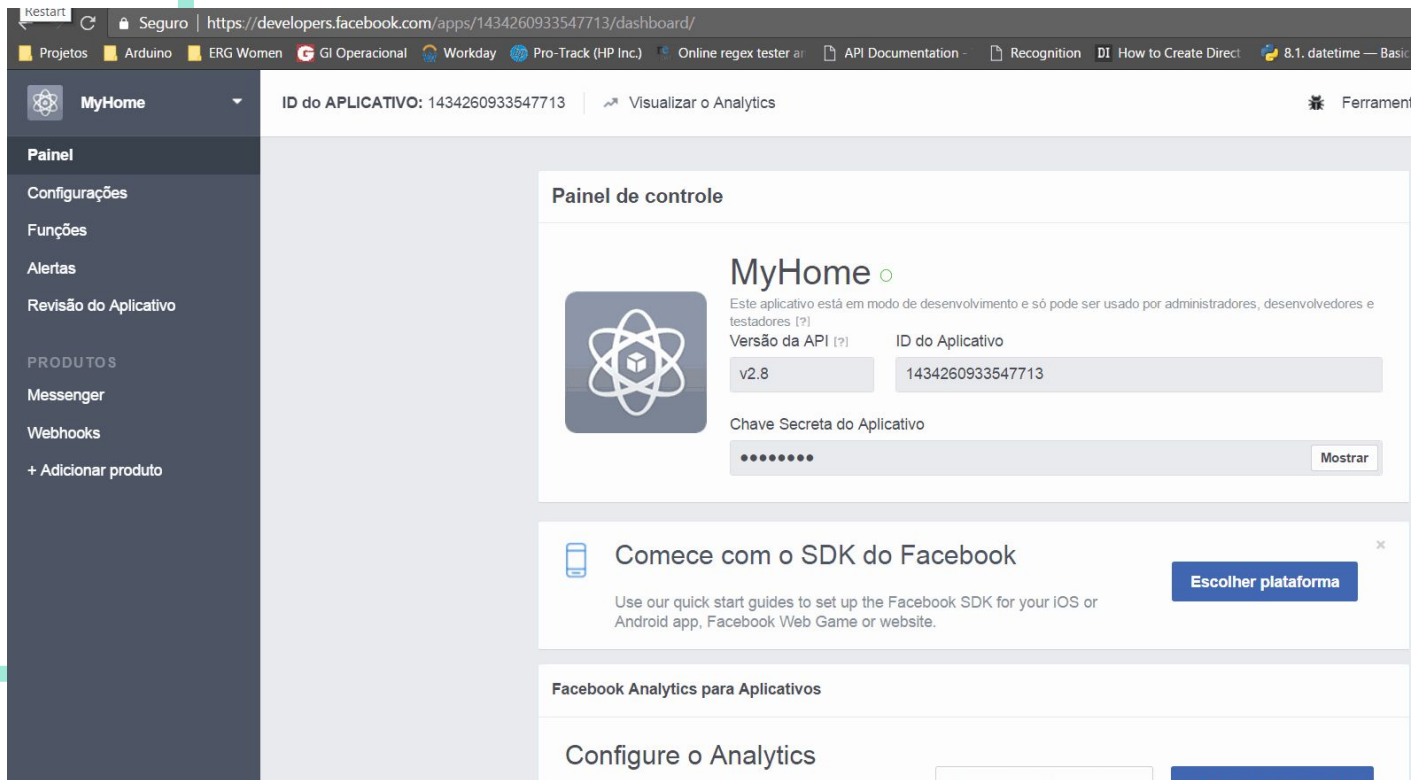
Configure o Analytics

O Analytics para Aplicativos ajuda você a desenvolver sua empresa e a saber mais sobre as ações que as pessoas realizam em seu aplicativo. A configuração leva apenas 5 minutos.

Experimente a demonstração

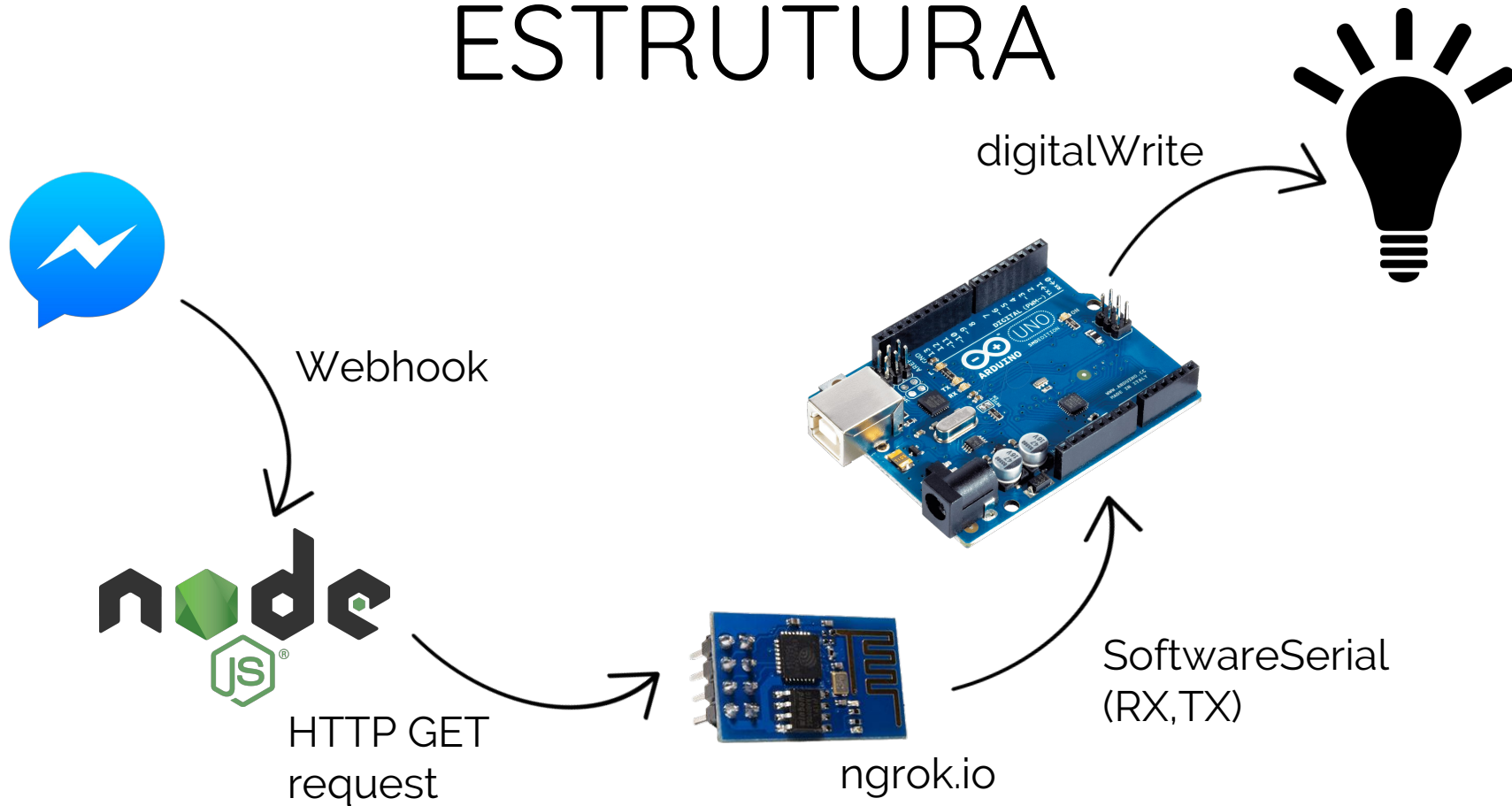
Exibir guia de início rápido

# FACEBOOK DEVELOPER | CRIAR UM WEBHOOK





# ESTRUTURA



# BOT





NÃO ERA BOTS  
NO PLURAL???















**\*Design de cenários de  
conversa**

**\*Integração com plataformas**


## Integrations

### One-click integrations

 Actions on Google <input type="checkbox"/>	 Facebook Messenger <input type="checkbox"/>	 Slack <input type="checkbox"/>	 Twitter <input type="checkbox"/>
 Twilio IP <input type="checkbox"/>	 Twilio (Text messaging) <input type="checkbox"/>	 Skype <input type="checkbox"/>	 Tropo (Text messaging) <input type="checkbox"/>
 Telegram <input type="checkbox"/>	 Kik <input type="checkbox"/>	 LINE <input type="checkbox"/>	 Cisco Spark <input type="checkbox"/>



# API.AI INTENTS

api.ai

MyHome ▾ ⚙

Intents +

Entities +

Domains

Training <sup>[beta]</sup>

Integrations

Fulfillment

> Docs

> Forum

T|

SAVE ⋮

Contexts ▾

User says 

Search in user says... 🔍

 ^

” Add user expression

Events ? ▾

Action ^

Enter action name...

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Response ? ^

Text response ? 🗑

1 Enter a text response...

# API.AI INTENTS

Restart

Intents ?

CREATE INTENT

Try it now...



Search intents...



Default Fallback Intent

Default Welcome Intent


TurnLightOff

TurnLightsOn

**i** Please use test console above to try a sentence.

# API.AI FULFILLMENT

Restart

 **api.ai** Record

MyHome

Intents

Entities

Domains

Training [beta]

Integrations

Fulfillment

Docs

Forum

Intents

CREATE INTENT

Search intents...


Default Fallback Intent



Default Welcome Intent









TurnLightOff

TurnLightsOn



# API.AI INTEGRATIONS






MyHome  


-  Intents 
-  Entities 
-  Domains
-  Training <sup>[beta]</sup>
-  Integrations
-  Fulfillment





---


-  Docs
-  Forum

 Intents 

[CREATE INTENT](#) 

Search intents... 

 Default Fallback Intent
 Default Welcome Intent
 TurnLightOff
 TurnLightsOn

 Please try a se

# API.AI INTEGRATIONS

# ESTRUTURA COM API.AI

---



# ESTRUTURA COM API.AI

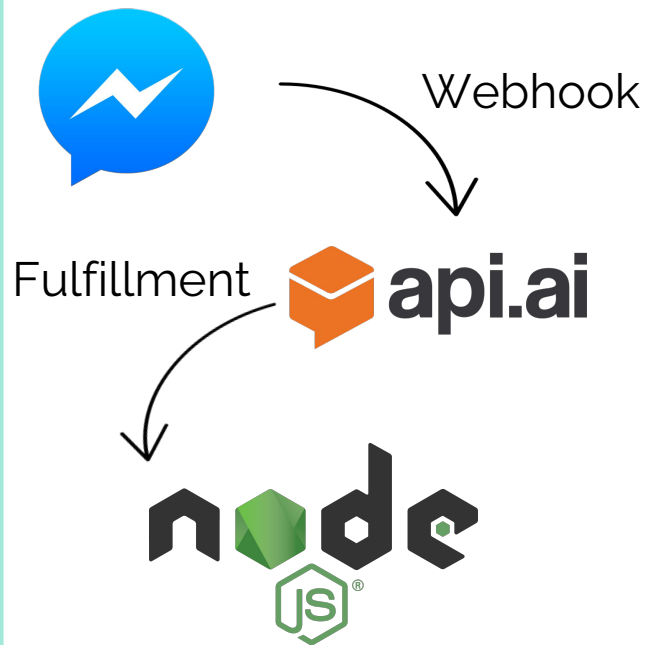


Webhook



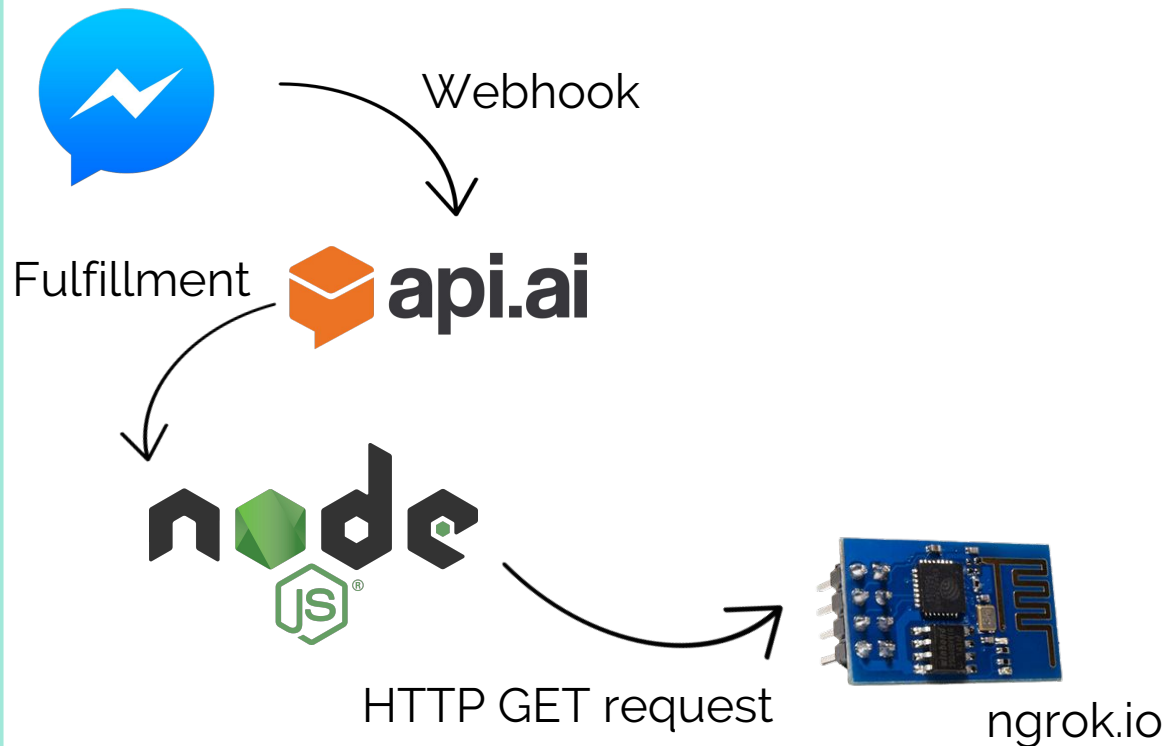
**api.ai**

# ESTRUTURA COM API.AI

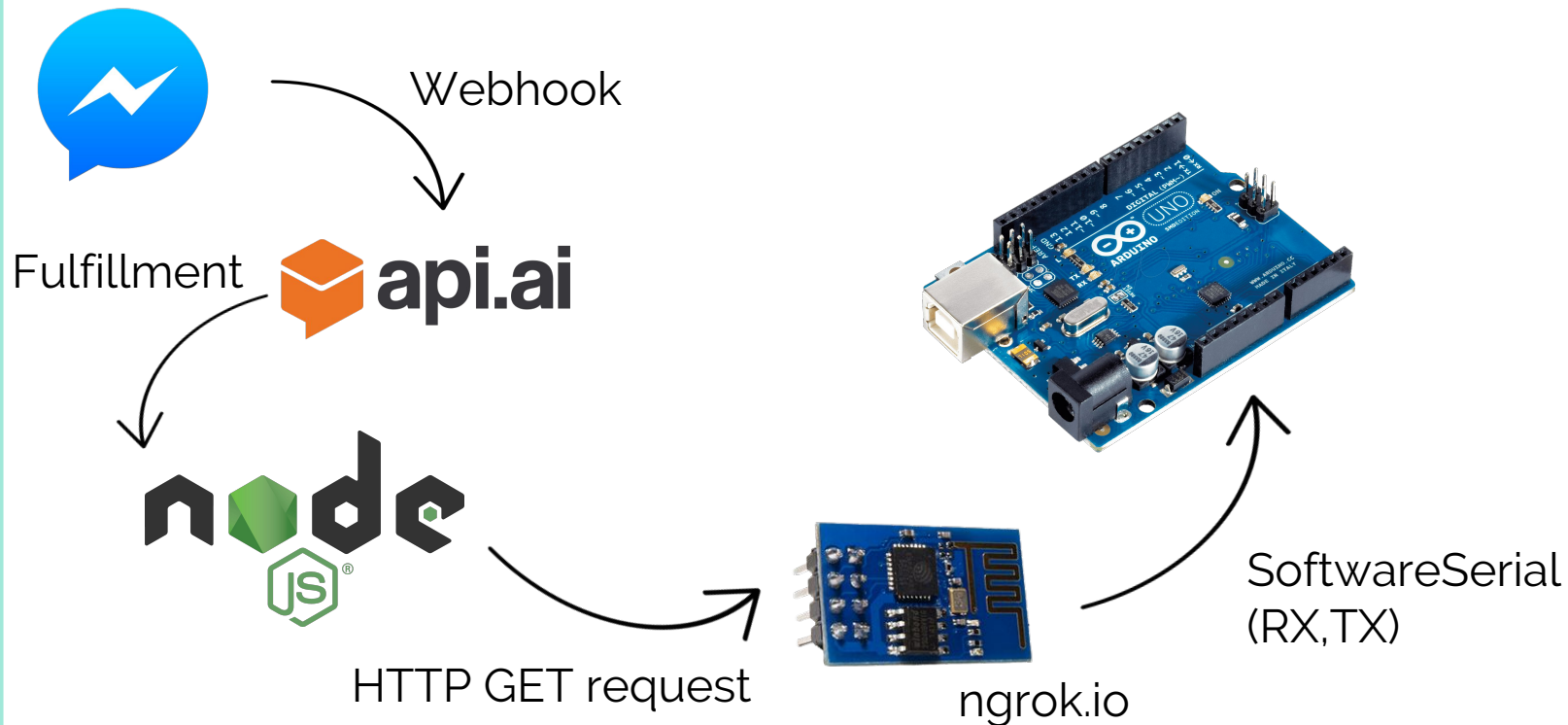




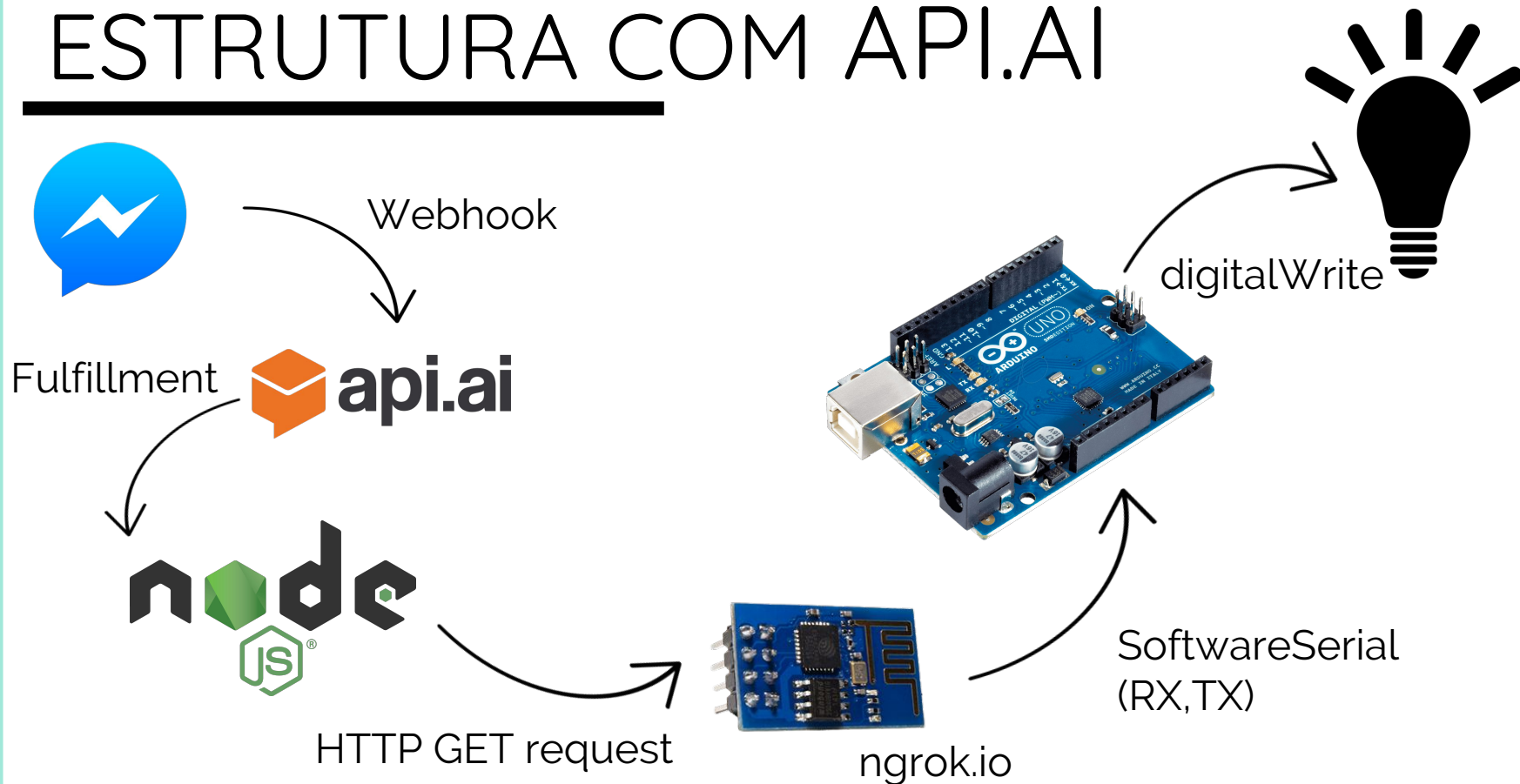
# ESTRUTURA COM API.AI



# ESTRUTURA COM API.AI



# ESTRUTURA COM API.AI



# API.AI | VARIÁVEIS

The screenshot displays the API.AI console interface for configuring an intent named 'TurnLightsOn'. The left sidebar contains navigation links: MyHome, Intents (selected), Entities, Domains, Training (beta), Integrations, Fulfillment, Docs, Forum, Account, and Logout. The main panel is titled 'TurnLightsOn' and includes a 'SAVE' button. It is divided into several sections: 'Contexts' (empty), 'User says' (containing 'Add user expression', 'turn the lights on', and 'turn on the lights'), 'Events' (empty), 'Action' (containing 'lights\_on'), a table for parameters, and 'Response' (containing a 'Text response' list).

**Contexts**

User says

“ Add user expression

“ turn the lights on

“ turn on the lights

**Events**

**Action**

lights\_on

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

[+ New parameter](#)

**Response**

**Text response**

- 1 Done!
- 2 Enter a text response variant...

# API.AI | CÓDIGO

```
16 // Message processing
17 app.post('/webhook', function (req, res) {
18   console.log(req.body);
19   var data = req.body.result;
20
21   let color = data.parameters.color;
22   let action = 1;
23   let pin = 11;
24
25   if (data.action == 'lights_off') {
26     action = 0;
27   }
28
29   if (color == 'green') {
30     pin = 12;
31   }
32 }
```

```
33 const params = {
34   url: 'https://c22a7e44.ngrok.io',
35   qs: {
36     action: action,
37     pin: pin
38   }
39 }
40 request.get(params, (err, httpResponse, body) => {
41   console.log('err', err);
42   console.log('response', httpResponse);
43   console.log('body', body);
44
45   res.sendStatus(200);
46 });
47
48
49 });
50
```

# API.AI | CÓDIGO

```
16 // Message processing
17 app.post('/webhook', function (req, res) {
18   console.log(req.body);
19   var data = req.body.result;
20
21   let color = data.parameters.color;
22   let action = 1;
23   let pin = 11;
24
25   if (data.action == 'lights_off') {
26     action = 0;
27   }
28
29   if (color == 'green') {
30     pin = 12;
31   }
32 }
```

```
33   const params = {
34     url: 'https://c22a7e44.ngrok.io',
35     qs: {
36       action: action,
37       pin: pin
38     }
39   }
40
41   request.get(params, (err, httpResponse, body) => {
42     console.log('err', err);
43     console.log('response', httpResponse);
44     console.log('body', body);
45
46     res.sendStatus(200);
47   });
48
49 });
50 }
```

# API.AI | CÓDIGO


```
16 // Message processing
17 app.post('/webhook', function (req, res) {
18   console.log(req.body);
19   var data = req.body.result;
20
21   let color = data.parameters.color;
22   let action = 1;
23   let pin = 11;
24
25   if (data.action == 'lights_off') {
26     action = 0;
27   }
28
29   if (color == 'green') {
30     pin = 12;
31   }
32 }
```

```
33 const params = {
34   url: 'https://c22a7e44.ngrok.io',
35   qs: {
36     action: action,
37     pin: pin
38   }
39 }
40 request.get(params, (err, httpResponse, body) => {
41   console.log('err', err);
42   console.log('response', httpResponse);
43   console.log('body', body);
44
45   res.sendStatus(200);
46 });
47
48
49 });
50 }
```

# API.AI | WELCOME INTENT

Seguro | https://console.api.ai/api-client/#/agent/dcdf0d02-b31e-4ef0-8923-739f774af3ec/intents

Projetos Arduino ERG Women GI Operacional Workday Pro-Track (HP Inc.) Online regex tester API Documentation Recognition DI How to Create Direct 8.1. datetime — Basic 26.4. unittest — Unit



MyHome

- Intents
- Entities
- Training (beta)
- Integrations
- Fulfillment
- Prebuilt Agents
- Small Talk
- > Docs
- > Forum

Intents

CREATE INTENT

Try it now...

Search intents...

Default Fallback Intent
Default Welcome Intent
TurnLightOff
TurnLightsOn


Please use test console above to try a sentence.



# API.AI | FALLBACK INTENT

← Restart | Seguro | https://console.api.ai/api-client/#/agent/dcdf0d02-b31e-4ef0-8923-739f774af3ec/intents

Projetos | Arduino | ERG Women | GI Operacional | Workday | Pro-Track (HP Inc.) | Online regex tester | API Documentation | Recognition | DI | How to Create Direct | 8.1. datetime — Basic | 26.4. unittest — Unit



## Intents

CREATE INTENT

Try it now...

MyHome

- Intents
- Entities
- Training <sup>(beta)</sup>
- Integrations
- Fulfillment
- Prebuilt Agents
- Small Talk
- > Docs
- > Forum

Search intents...

Default Fallback Intent
Default Welcome Intent
TurnLightOff
TurnLightsOn

Please use test console above to try a sentence.

# BOT

---

# REFERÊNCIAS

---

[1] AT Commands:

<http://www.pridopia.co.uk/pi-doc/ESP8266ATCommandsSet.pdf>

[2] <http://blog.filipeflop.com/wireless/esp8266-arduino-tutorial.html>

[3] <http://pastebin.com/AdGWPNoy>

[4] <http://pastebin.com/WmHHsnZ5>

[5] <http://www.instructables.com/id/Control-ESP8266-Over-the-Internet-from-Anywhere/?ALLSTEPS>

# CONTATO

Email: tais38@gmail.com

Github: taisbellini

Facebook: Taís Bellini

Repositório da palestra:

<https://github.com/taisbellini/arduinoaday>