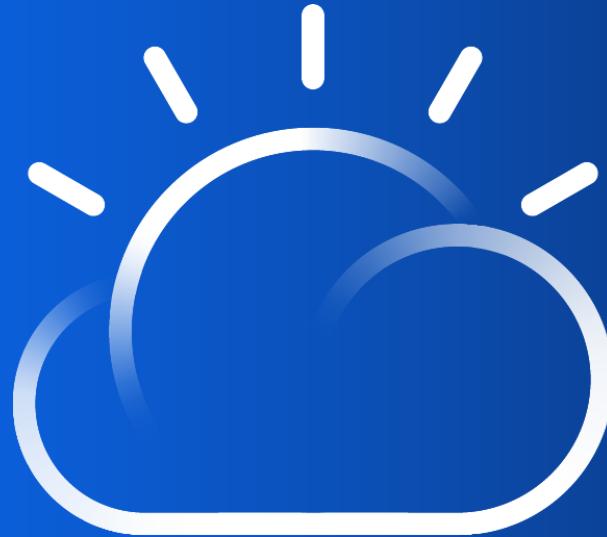


Controlando dispositivos IoT remotamente através de linguagem natural



—
Eduardo Petecof Mattoso
João Pedro Poloni Ponce

CodeBakery Developers



Conteúdo

Introdução	01	Importe o Flow	49
Internet das coisas	03	Configurando a conexão MQTT	51
Interagindo em linguagem natural	07	Verificando a conexão	57
Computação em nuvem	09		
Arquitetura proposta	14	Enviando comandos com Cloud Functions	63
		Criando as funções	63
	16	Testando a primeira função	72
Watson IoT Platform	17	Criando a segunda função	77
Criando o serviço	20	Obtendo as credenciais de uso	84
Criando um Device Type	23		
Criando um Device	29	Criando o Assistente Virtual	88
Configurando o tipo de conexão	32	Criando uma instância do Watson Assistant	88
Ativando o Cache de eventos		Fazendo o upload do Workspace	92
	33	Credenciais da sua Cloud Function	94
Simulando um Device com o Node-RED		Testando o Assistente Virtual	103
Criando uma instância do Node-RED	36		
Configurando o seu Node-RED	44		
Instale o Dashboard	49	Conclusão	104

Internet das coisas:

Digitalização do mundo físico.



Crie a representação virtual de um ativo real.

Monitoramento contínuo do estado desse ativo através de sensores.

Controle remoto do ativo submetendo comandos para ele em qualquer lugar.

Armazenamento de dados de utilização para aprendizado contínuo.

Integração com ferramentas cognitivas: Interaja com o usuário em linguagem natural.

“The Digital Twin is the virtual, statefull representation of a physical object or system across its life-cycle (design, build, operate) using operational real-time data and other sources to enable understanding, learning, reasoning, and dynamic recalibration for improved decision making.”

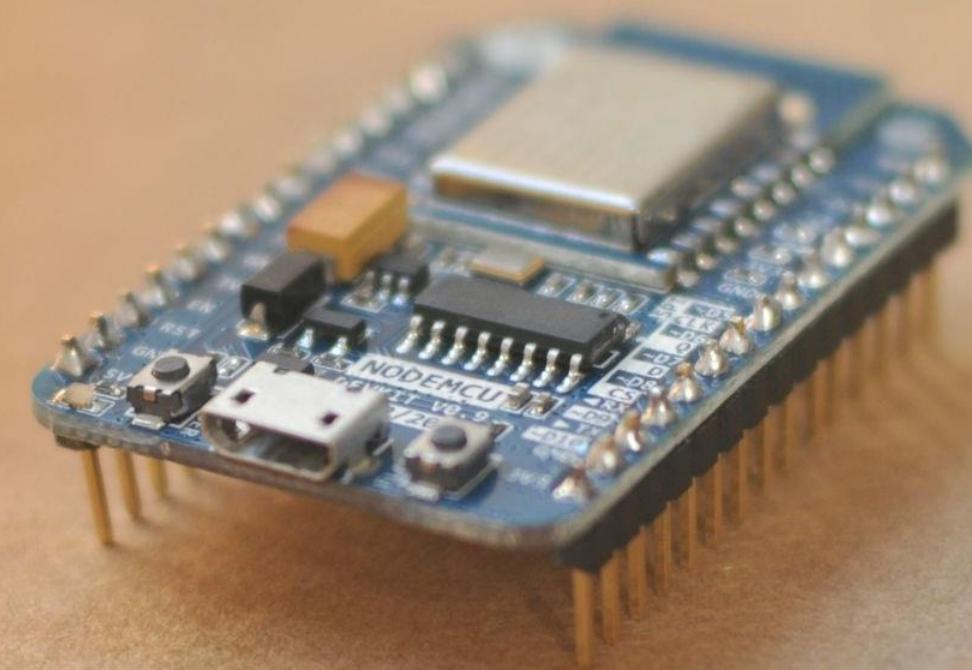
Andrew Foster

Senior Offering Manager - Watson IoT Platform

NodeMCU

“The Development Kit based on ESP8266, integrates GPIO, PWM, IIC, 1-Wire and ADC all in one board. Power your development in the fastest way combining with NodeMcu Firmware!”

<https://nodemcu.com>



Omega2

“WiFi-enabled devices ideal for rapid IoT development. Learn, prototype to validate your idea, and go to production, all with one platform.”

<https://onion.io/>



Assistentes Virtuais:

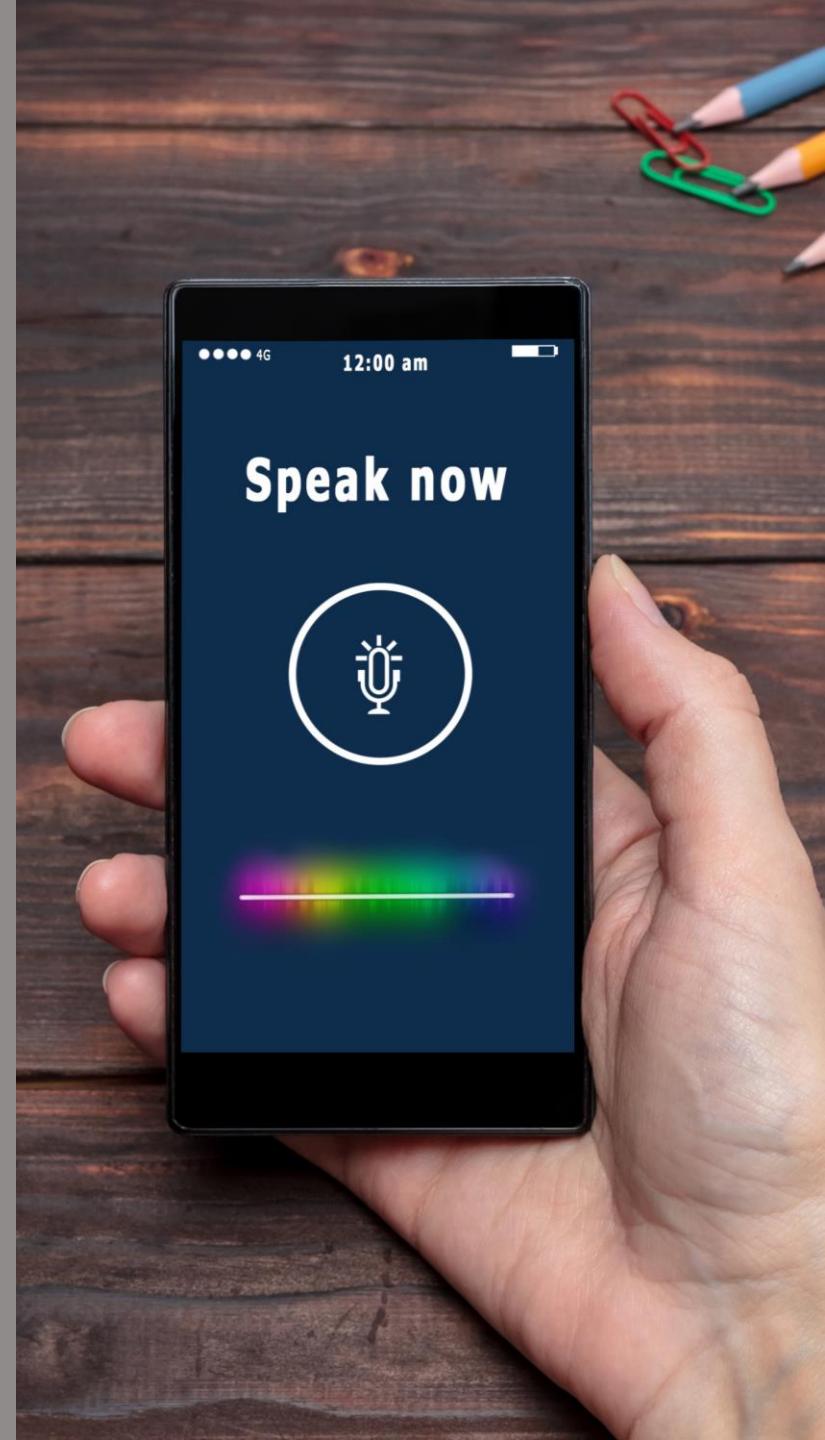
Interagindo em linguagem natural.



A linguagem natural permite que o usuário vá **direto ao ponto** que o interessa, sem precisar de navegações.

Um diálogo permite a extração de várias informações sobre o usuário. **Muito além de um simples apertar de botão.**

Também é possível o uso da voz e do som, os quais permitem que o usuário **mantenha livre a sua visão e as suas mãos.**



Computação em nuvem:

**Escalável e
sob demanda.
24h por dia.**

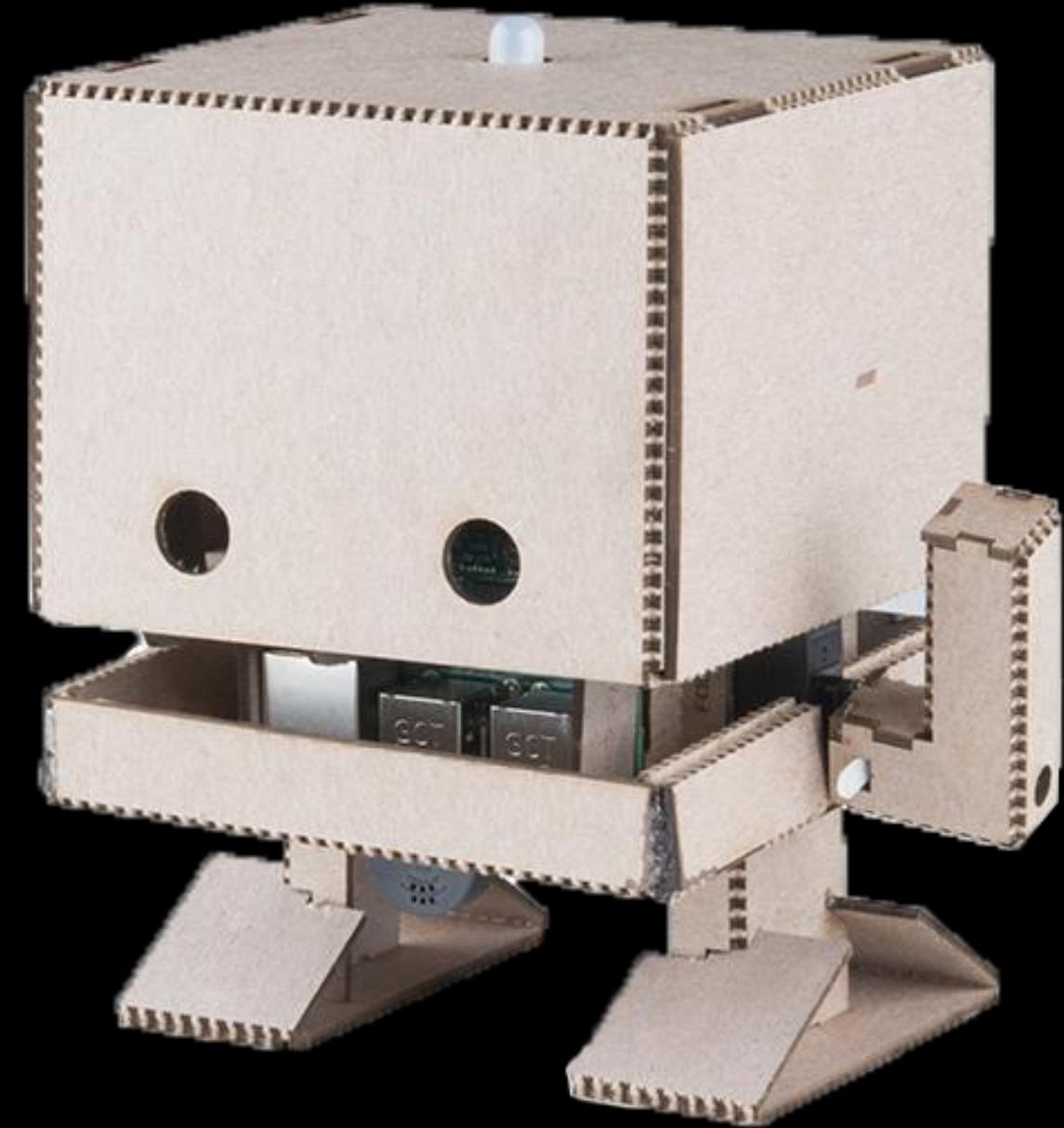


Como você
faria?

Opção válida:

Dispositivo no centro:
Um dispositivo poderoso gerencia as interações.

Ex: TJ BOT



Gerenciamento na nuvem:

Permite o uso de diversos dispositivos, até os mais simples.

Monitore todos os passos da sua aplicação.

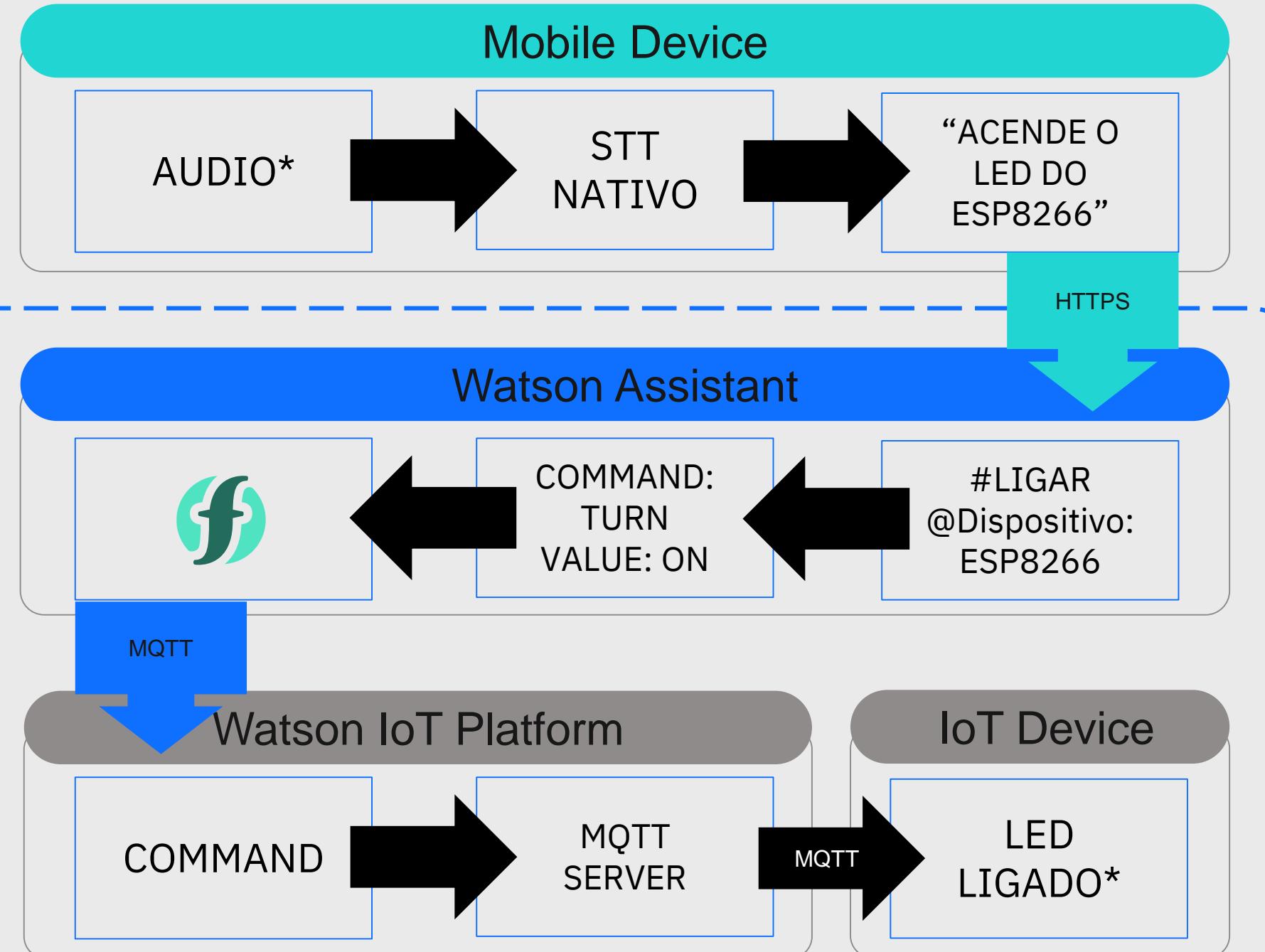
Consuma recursos sob demanda e aumente a escala de forma gradual.

Inclua novas ferramentas e serviços facilmente. Sem mudar o código do seu dispositivo.

Arquitetura completa



IBM Cloud



Serviços utilizados:



Watson Assistant

Ferramenta para construção de assistentes virtuais em linguagem natural



IBM Cloud

Nuvem que suporta a aplicação, fornecendo os recursos apresentados



Watson IoT Platform

Ambiente comum para gerenciar e se comunicar com os dispositivos IoT



IBM Cloud Functions

Código executado sob demanda na nuvem

Por que comprar um
assistente virtual se você
pode fazer o seu? É simples,
quer ver?

1. Watson IoT Platform

1.1. Criando um novo recurso:

- No seu Dashboard clique no botão “Create Resource”

The screenshot shows the IBM Cloud Dashboard interface. At the top, there are tabs for 'Dashboard - IBM Cloud' and 'Node-RED : lab-iot-jppp.mybluemix.net'. The main content area is titled 'Dashboard' and contains sections for 'Cloud Foundry Applications' and 'Cloud Foundry Services'. In the top right corner, there is a blue button labeled 'Create resource' which is highlighted with a red rectangular box. The 'Cloud Foundry Applications' section lists one application: 'Lab-IoT-JPPP' with details: Region US South, CF Org JoaoPedroPP52..., CF Space dev, Memory (MB) 256, Status Running (1/1). The 'Cloud Foundry Services' section lists three services: 'Lab-IoT-JPPP-cloudantNoSQLDB' (with a refresh icon), 'Watson Assistant-2s', and 'omega', all with similar details. The bottom right corner of the dashboard has a 'FEEDBACK' button.

Name	Region	CF Org	CF Space	Memory (MB)	Status
Lab-IoT-JPPP	US South	JoaoPedroPP52...	dev	256	Running (1/1)

Name	Region	CF Org	CF Space	Plan	Service Offering
Lab-IoT-JPPP-cloudantNoSQLDB	US South	JoaoPedroPP52...	dev	Lite	Cloudant NoSQL ...
Watson Assistant-2s	US South	JoaoPedroPP52...	dev	Lite	Watson Assistan...
omega	US South	jp052@hotmail....	dev	Lite	Internet of Thing...

1. Watson IoT Platform

1.1. Criando um novo recurso:

- No seu Dashboard clique no botão “Create Resource”

- Na aba se busca digite “iot platform” e ache a “Internet of Things Platform”

The screenshot shows the IBM Cloud Catalog interface. At the top, there are two tabs: "Catalog - IBM Cloud" and "Node-RED : lab-iot-jppp.mybluemix.net". The URL in the address bar is <https://console.bluemix.net/catalog/?search=iot%20platform>. The main content area is titled "Catalog" and has a search bar containing "iot platform". To the left of the search bar is a red arrow pointing to it. Below the search bar is a sidebar with "All Categories (2)" and a list of categories: Compute, Containers, Networking, Storage, AI, Analytics, Databases, Developer Tools, Integration, Internet of Things (1), Security and Identity, Starter Kits (1), Web and Mobile, and Web and Application. On the right side, under the "Internet of Things" heading, there is a service card for "Internet of Things Platform" (Lite • IBM). This card is highlighted with a red box. The card description states: "This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and". Below this is another section titled "Starter Kits" with a card for "Internet of Things Platform Starter" (Lite • IBM). The card description starts with: "Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the Starter, you can quickly simulate an Internet of".

1. Watson IoT Platform

1.1. Criando um novo recurso:

- No seu Dashboard clique no botão “Create Resource”
- Na aba se busca digite “iot platform” e ache a “Internet of Things Platform”
- Dê um nome para o serviço e clique no botão “Create”

The screenshot shows the IBM Cloud Catalog interface. In the top navigation bar, there are tabs for Catalog, Docs, Support, and Manage, along with a search bar and user information. The main content area displays the "Internet of Things Platform" service card. The service name is "IoT-Lab", selected from a dropdown menu. The deployment region is set to "US South", the organization to "JoaoPedroPP520ORG", and the space to "dev". The "Features" section lists "Connect" and "Information Management". At the bottom right of the card, a large blue "Create" button is highlighted with a red border.

1. Watson IoT Platform

1.1. Criando um novo recurso:

- Clique no botão Launch Tool para começar a usar o serviço.

The screenshot shows the IBM Cloud Service Details page for the IoT-Lab service. The URL in the browser is <https://console.bluemix.net/services/iotf-service/c86bba6b-3a47-43f>. The page title is "Service Details - IBM Cloud" and the sub-title is "Node-RED : lab-iot-jppp.mybluemix.net". The main content area displays the IoT-Lab service details: "1.52% Used | 196.96 Megabyte exchanged available", "Location: US South", "Org: JoaoPedroPP520RG", and "Space: dev". A large graphic of a central processing unit (CPU) with arrows indicating data flow is centered on the page. Below it, the text "Let's get started with Watson IoT Platform" is displayed in a large, bold, teal font. A red box highlights the "Launch" button, which is located at the bottom right of the main content area. The top navigation bar includes links for Catalog, Docs, Support, Manage, and a search bar. The right side of the screen shows a user profile with the ID 1507559 - IBM.

1. Watson IoT Platform

1.2. Criando um Device Type:

- Vá para a aba Device Types;
- Clique no botão “Add Device Type”

The screenshot shows a browser window with four tabs: 'Dashboard - IBM Cloud', 'IBM Watson IoT Platform' (which is active), 'Node-RED : device-simulator-d' (disabled), and 'New Tab'. The URL in the address bar is <https://7p8n56.internetofthings.ibmcloud.com/dashboard/#/devices/deviceTypes-v2>. On the right side of the header, there is a user profile for 'epetecof@br.ibm.com' with ID '7p8n56'. The main content area has a dark header with the title 'Device Types'. Below it is a search bar with placeholder text 'Type the name to search for' and a magnifying glass icon. A red box highlights the 'Device Types' tab in the navigation bar. Another red box highlights the 'Add Device Type' button in the top right corner of the content area. To the left of the content area is a vertical sidebar with various icons: a grid, a gear, a bar chart, a fingerprint, a line graph, a document, a lock, a gear, and a compass.

Device Types

This table lists all device types that are defined. You can filter the list and search for the name and description. You can modify and configure existing device types and add new device types.

<input type="checkbox"/>	Name	Description	Number of Devices	Class ID	
--------------------------	------	-------------	-------------------	----------	--

You don't have any device types created.

Add Device Type

1. Watson IoT Platform

1.2. Criando um Device Type:

- Vá para a aba Device Types;
- Clique no botão “Add Device Type”
- Dê um nome e uma descrição para o seu Device Type, em seguida clique em “Next”

The screenshot shows the 'Add Type' page in the IBM Watson IoT Platform. The 'Type' dropdown is set to 'Device'. The 'Name' field contains 'Simulated-Device' and the 'Description' field contains 'Um device simulado pelo Node-RED'. Both fields are highlighted with red boxes. The 'Next' button at the bottom right is also highlighted with a red box.

1. Watson IoT Platform

1.2. Criando um Device Type:

- Vá para a aba Device Types;
- Clique no botão “Add Device Type”
- Dê um nome e uma descrição para o seu Device Type, em seguida clique em “Next”.
- Deixe vazios os campos de informação e clique em “Done”

The screenshot shows the 'Device Types' section of the IBM Watson IoT Platform. On the left, there's a vertical sidebar with various icons. The main area has tabs for 'Add Type', 'Identity', and 'Device Information'. The 'Device Information' tab is active. It contains fields for Serial Number, Model, Description, Hardware Version, Manufacturer, Device Class, Firmware Version, and Descriptive Location. A blue button labeled '+ Add Metadata' is visible. In the bottom right corner, there's a blue 'Done' button with a red border, which is highlighted with a red rectangle. The browser address bar shows the URL: https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/devices/deviceTypes-v2.

1. Watson IoT Platform

1.3. Criando um Device:

- Depois de criado seu Device Type, clique no botão “Register Devices”

The screenshot shows a browser window with four tabs: 'Dashboard - IBM Cloud', 'IBM Watson IoT Platform' (active), 'Node-RED : device-simulator-d' (disabled), and 'New Tab'. The main content area is titled 'IBM Watson IoT Platform' and has a sidebar with various icons. The 'Device Types' tab is selected. A message says 'You added the new device type: Simulated-Device'. Below it are two buttons: 'Register Device' (highlighted with a red box) and 'Advanced Flow'. A large circular icon with a central gear symbol is on the right. The URL in the address bar is <https://7p8n56.internetofthings.ibmcloud.com/dashboard/#/devices/deviceTypes-v2>.

1. Watson IoT Platform

1.3. Criando um Device:

- Depois de criado seu Device Type, clique no botão “Register Devices”
- Dê um nome para o seu device e clique em “Next”

The screenshot shows the 'Add Device' interface in the IBM Watson IoT Platform. On the left, there's a vertical sidebar with various icons. The main area has tabs for 'Browse', 'Action', 'Device Types', 'Add Device', 'Identity', 'Device Information', 'Security', and 'Summary'. The 'Identity' tab is active. It asks to select a device type and give it a unique ID. The 'Device Type' dropdown is set to 'Simulated-Device'. The 'Device ID' input field contains 'my-first-device', which is highlighted with a red box. At the bottom right, there are 'Cancel' and 'Next' buttons, with 'Next' also highlighted with a red box.

1. Watson IoT Platform

1.3. Criando um Device:

- Depois de criado seu Device Type, clique no botão “Register Devices”
- Dê um nome para o seu device e clique em “Next”
- Deixe vazios os campos de informação e clique em “Next”

The screenshot shows the 'Device Information' page of the IBM Watson IoT Platform. The left sidebar has icons for various device types: Grid, Sensor, Actuator, Fingerprint, Network, Storage, Security, Configuration, and Location. The main navigation bar includes 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Node-RED : lab-iot-jppp.mybluemix.net'. The URL in the address bar is <https://1603q6.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2>. The top right corner shows the user's email (jp052@hotmail.com) and ID (1603q6). The page title is 'IBM Watson IoT Platform' with tabs for 'Browse', 'Action', and 'Device Types'. The 'Device Information' section contains fields for 'Serial Number', 'Model', 'Description', 'Hardware Version', 'Manufacturer', 'Device Class', 'Firmware Version', and 'Descriptive Location'. A blue button labeled '+ Add Metadata' is at the bottom left of the form area. At the bottom right, there are 'Back' and 'Next' buttons, with the 'Next' button being red and highlighted with a red box.

1. Watson IoT Platform

1.3. Criando um Device:

- Deixe vazio o campo do token e clique em “Next” para gerar um token automaticamente.

The screenshot shows the IBM Watson IoT Platform interface for creating a new device. The top navigation bar includes tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Node-RED : lab-iot-jppp.myblu'. The main title is 'IBM Watson IoT Platform' with sub-tabs 'Browse', 'Action', and 'Device Types'. A sidebar on the left lists various device types with icons: Grid, Sensors, People, Location, Actions, Fingerprint, Accelerometer, Temperature, Humidity, Light, and GPS. The current step is 'TOKEN (DEFAULT)'. The description states: 'Allow the service to generate an authentication token for you. Tokens are 18 characters and contain a mix of alphanumeric characters and symbols. The token is returned to you at the end of the device registration process.' Below this is a field labeled 'Authentication Token' with a placeholder 'Enter an optional token' and an information icon. A note below the field says: 'Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.' Another note states: 'Authentication token are encrypted before we store them.' At the bottom right, there are 'Back' and 'Next' buttons, with the 'Next' button highlighted by a red rectangle.

1. Watson IoT Platform

1.3. Criando um Device:

- Deixe vazio o campo do token e clique em “Next” para gerar um token automaticamente.

- Clique em “Done” para terminar a criação e gerar o token.

The screenshot shows a browser window with four tabs: 'Dashboard - IBM Cloud', 'IBM Watson IoT Platform', 'Node-RED : device-simulator-d', and 'New Tab'. The active tab is 'IBM Watson IoT Platform' at the URL <https://7p8n56.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2?add=Simulated-Device>. The page title is 'IBM Watson IoT Platform' and the sub-page title is 'Add Device'. The main content area is titled 'Summary' and contains the following information:

- Device Type: Simulated-Device
- Device ID: my-first-device
- Security Token: To be generated

A blue button labeled 'View Metadata' is visible. At the bottom right, there are two buttons: a blue square with a white left arrow and a red-bordered blue rectangle labeled 'Done'.

1. Watson IoT Platform

1.3. Criando um Device:

- Deixe vazio o campo do token e clique em “Next” para gerar um token automaticamente.

- Clique em “Done” para terminar a criação e gerar o token.

- Copie seu token e guarde no bloco de notas para usar no futuro.

The screenshot shows a web browser window with four tabs: 'Dashboard - IBM Cloud', 'IBM Watson IoT Platform', 'Node-RED : device-simulator-d', and 'New Tab'. The active tab is 'IBM Watson IoT Platform' at the URL <https://7p8n56.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2?add=Simulated-Device>. The user is logged in as 'epetecof@br.ibm.com' (ID: 7p8n56). The main content area displays a device named 'my-first-device'. On the left, there's a sidebar with icons for various device management tasks. The 'DEVICE DRILLDOWN' section lists: Device Credentials, Connection Information, Recent Events, State, Device Information, Metadata, Extension Configuration, Diagnostics, Connection Logs, and Device Actions. The 'Device Credentials' section contains the following information:

Organization ID	7p8n56
Device Type	Simulated-Device
Device ID	my-first-device
Authentication Method	use-token-auth
Authentication Token	[Redacted]

A warning message at the bottom states: 'Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token.'

Checkpoint 1

Depois desses passos devem estar preenchidas as seguintes propriedades do seu modelo.

1.1. Device Credentials

- Organization ID (ORG):
- Device Type:
- Device ID:
- Authentication Method: use-token-auth
- Authentication Token (Device):

Essas credenciais são necessárias para identificar e autorizar um dispositivo que tenta se conectar na sua plataforma.

1. Watson IoT Platform

1.4. Configurando o tipo de conexão utilizada.

- Usando a navegação lateral, entre no menu de segurança (Cadeado). Em seguida clique no botão relacionado a “Connection Security”.

The screenshot shows the IBM Watson IoT Platform Policies dashboard. The left sidebar has a 'Policies' tab selected, along with other icons for Service Details, Watson Assistant, Node-RED, and NLU. The main content area is titled 'Policies' and contains three sections: 'Connection Security' (with a red box around the edit icon), 'Blacklist' (disabled), and 'Whitelist' (disabled). A 'Cookie Preferences' button is at the bottom right.

IBM Watson IoT Platform

Policies

Policies

You can configure policies to enhance connection security and control access to the server from devices.

Connection Security
Configure the security level for device connection.

Blacklist
Block access from specific IP addresses and countries. Activating a blacklist disables an active whitelist.

Disabled

Whitelist
Allow access from specific IP addresses and countries. Activating a whitelist disables an active blacklist.

Disabled

Cookie Preferences

1. Watson IoT Platform

1.4. Configurando o tipo de conexão utilizada.

- Usando a navegação lateral, entre no menu de segurança (Cadeado). Em seguida clique no botão relacionado a “Connection Security”.

- Na opção Security Level selecione “TLS Optional”

The screenshot shows the IBM Watson IoT Platform security configuration interface. The left sidebar has a 'Connection Security' icon highlighted. The main page title is 'Default Rule'. It defines the default connection security level for all device types. A note states that device numbers and predicted compliance values are estimates based on a report running at varying intervals. The table shows one device with TLS with Token Authentication selected. A dropdown menu for 'Security Level' is open, showing 'TLS Optional' selected and highlighted with a red box. Other options in the dropdown include 'TLS with Client Certificate Authentication', 'TLS with Client Certificate AND T...', and 'TLS with either Client Certificate ...'. A note below the table says 'Custom Rules' allow defining connection rules for specified device types. The top navigation bar shows tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, Node-RED : lab-iot-jp..., NLU-IoT/esp-sensor.js, and a '+' button. The URL is https://16o3q6.internetofthings.ibmcloud.com/dashboard/#/security/. The top right shows the user's email (jp052@hotmail.com) and ID (16o3q6).

Scope	Security Level	Predicted Compliance	# of Devices
Default	TLS with Token Authentication	<div style="width: 20%; background-color: red; height: 10px;"></div> 0 Pass 1 Fail 0 Unknown	1 device
	TLS Optional		
	TLS with Client Certificate Authentication		
	TLS with Client Certificate AND T...		
	TLS with either Client Certificate ...		

1. Watson IoT Platform

1.4. Configurando o tipo de conexão utilizada.

- Usando a navegação lateral, entre no menu de segurança (Cadeado). Em seguida clique no botão relacionado a “Connection Security”.

- Na opção Security Level selecione “TLS Optional”. Salve.

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, Node-RED : lab-iot-jp, and NLU-IoT/esp-sensor.js. The main title is "IBM Watson IoT Platform" under "Default Rule". The left sidebar has icons for various services: Cloud Foundry, Watson Assistant, Watson Assistant, Node-RED, and NLU-IoT/esp-sensor.js. The main content area displays a table with one row:

Scope	Security Level	Predicted Compliance	# of Devices
Default	TLS Optional	Refresh compliance	1 device

A red box highlights the "TLS Optional" dropdown in the "Security Level" column. Below the table, there's a section titled "Custom Rules" with the following text: "You can define custom connection rules for specific device types. Custom rules overwrite the default rule for the specified device types. The predicted compliance value is updated to reflect the default settings and the custom settings." At the bottom right of the main area are "Cancel" and "Save" buttons, with "Save" being highlighted by a red box.

1. Watson IoT Platform

1.5. Ativando o cache de eventos.

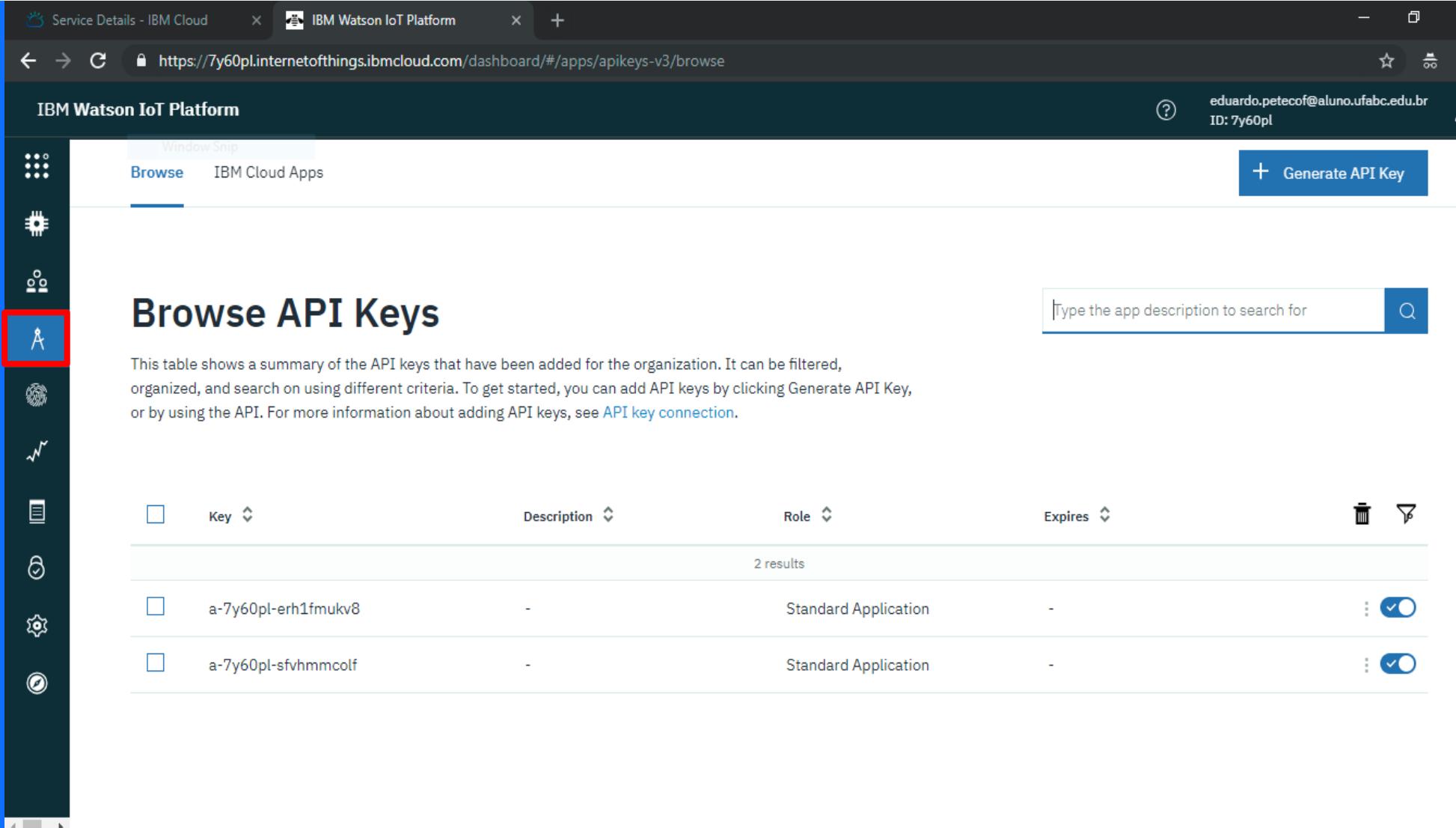
- Na aba se configurações gerais (Engrenagem). Procure pela seção “Last Event Cache” e ative a opção.

The screenshot shows the IBM Watson IoT Platform settings interface. On the left, there's a sidebar with various icons and sections: PLATFORM (About, Identity, Experimental Features - Last Event Cache selected), DATA AND DEVICES (Custom Device Management, Packages, Device Simulator), and SECURITY (Connection Security, CA Certificates, Messaging Server Certificates, Group Access beta). The main content area is titled "Last Event Cache". It explains what LEC is and provides documentation link. It has a toggle switch labeled "Activate Last Event Cache" which is turned on (indicated by a checked checkbox). Below it is a button for "Events stored in cache" with two options: "1 day" and "7 days", where "7 days" is highlighted with a red box. The section title "Custom Device Management Packages" is also present with its own documentation link and an "Add Package" button.

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Na aba de aplicações (Compasso).



The screenshot shows the 'IBM Watson IoT Platform' dashboard with the 'Browse' tab selected. On the left, there is a vertical sidebar with various icons. The icon for 'Applications' (a compass-like icon) is highlighted with a red box. The main content area is titled 'Browse API Keys'. It contains a table with two rows of API keys:

Key	Description	Role	Expires	Actions
a-7y60pl-erh1fmukv8	-	Standard Application	-	⋮ trash
a-7y60pl-sfvhmmcolf	-	Standard Application	-	⋮ trash

A search bar at the top right says 'Type the app description to search for' with a magnifying glass icon. The top navigation bar shows the URL 'https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/apps/apikeys-v3/browse' and the user information 'eduardo.petecof@aluno.ufabc.edu.br ID: 7y60pl'.

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Na aba de aplicações (Compasso).
- Clique no botão “Generate API Key”.

IBM Watson IoT Platform

IBM Watson IoT Platform

Window Snip

Browse IBM Cloud Apps

+ Generate API Key

Type the app description to search for

Browse API Keys

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).

Key	Description	Role	Expires
a-7y60pl-erh1fmukv8	-	Standard Application	-
a-7y60pl-sfvhmmcolf	-	Standard Application	-

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Na aba de aplicações (Compasso).
- Clique no botão “Generate API Key”.
- Nenhuma informação de descrição é necessária, clique no botão “Next”.

The screenshot shows a web browser window titled 'IBM Watson IoT Platform'. The URL in the address bar is <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/apps/apikeys-v3/browse>. The user is logged in with the email 'eduardo.petecof@aluno.ufabc.edu.br' and ID '7y60pl'. On the left, there's a sidebar with various icons. The main area has tabs for 'Browse' and 'IBM Cloud Apps', with 'Browse' selected. Below the tabs, there are three buttons: 'Generate API Key', 'Information', and 'Permissions'. The 'Generate API Key' button is highlighted. A 'Description' input field is empty. An 'API Key Expires' section shows a toggle switch set to 'Off' with an 'On' option and a 'Choose date' button. At the bottom right are 'Cancel' and 'Next' buttons, with 'Next' being highlighted with a red box.

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Na aba de aplicações (Compasso).
- Clique no botão “Generate API Key”.
- Nenhuma informação de descrição é necessária, clique no botão “Next”.
- Clique no Dropdown.

The screenshot shows the 'IBM Watson IoT Platform' dashboard with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/apps/apikeys-v3/browse>. On the left, there's a sidebar with various icons. The main area has tabs for 'Browse' and 'IBM Cloud Apps', with 'Permissions' selected. A sub-section titled 'Generate API Key' shows a dropdown menu with 'Visualization Application' selected, and a red box highlights the dropdown arrow. Below it, a note says 'The application will have access for the following role:' followed by a link to 'User, application, and gateway roles'. At the bottom right are buttons for 'Generate Key' and a search bar with placeholder text 'Type the app description to search for'.

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Selecione a opção “Standard Application”.

The screenshot shows the 'IBM Watson IoT Platform' dashboard with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/apps/apikeys-v3/browse>. The user is logged in as eduardo.petecof@aluno.ufabc.edu.br (ID: 7y60pl). On the left, there is a sidebar with various icons. The main area is titled 'Generate API Key' and has tabs for 'Information' and 'Permissions'. A dropdown menu under 'Role' shows several options: 'Visualization Application' (selected), 'Data Processor Application', 'Device Application', 'Operations Application', 'Standard Application' (highlighted with a red box), and 'Visualization Application' again at the bottom. At the bottom right of the main panel are 'Generate Key' and 'Cancel' buttons. Below the main panel, the title 'Browse API Keys' is visible along with a search bar.

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Selecione a opção “Standard Application”.
- Clique no botão “Generate Key”.

The screenshot shows the 'IBM Watson IoT Platform' dashboard with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/apps/apikeys-v3/browse>. On the left, there is a sidebar with various icons. The main area is titled 'Browse API Keys' and shows a section for generating a new API key. It asks 'The application will have access for the following role:' and provides a dropdown menu set to 'Standard Application'. A red box highlights the 'Generate Key' button at the bottom right of the form. The top right corner shows the user's email 'eduardo.petecof@aluno.ufabc.edu.br' and ID '7y60pl'.

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Selecione a opção “Standard Application”.
- Clique no botão “Generate Key”.
- Copie os valores de sua API Key e Authentication Token para o NotePad.

The screenshot shows a browser window titled "IBM Watson IoT Platform" with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/apps/apikeys-v3/browse>. The user is logged in as "eduardo.petecof@aluno.ufabc.edu.br" (ID: 7y60pl). The left sidebar has a "Browse" tab selected. The main content area displays a success message: "The API key has been added." Below it, a note states: "Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the API key to generate a new authentication token." The "Generated Details" section shows the "API Key" and "Authentication Token" fields, both of which are redacted with a large red rectangle. The "API Key Information" section shows the "Role" as "Standard Application" and "Expires" as "Never". A warning icon with the text "Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token." is present. At the bottom are buttons for "View API Key", "Add Another", and "Close".

1. Watson IoT Platform

1.6. Criando uma Application Key.

- Selecione a opção “Standard Application”.
- Clique no botão “Generate Key”.
- Copie os valores de sua API Key e Authentication Token para o NotePad.
- Clique no botão “Close”.

The screenshot shows a browser window titled "IBM Watson IoT Platform" with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/apps/apikeys-v3/browse>. The user is logged in as "eduardo.petecof@aluno.ufabc.edu.br" (ID: 7y60pl). On the left, there's a sidebar with various icons. The main content area displays a success message: "The API key has been added." Below it, under "Generated Details", the API Key and Authentication Token fields are shown, both redacted with a large red box. To the right, under "API Key Information", the role is listed as "Standard Application" and the expiration is "Never". A warning message below the generated details says: "Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token." At the bottom, there are three buttons: "View API Key", "Add Another", and "Close", with "Close" being highlighted by a red box.

Checkpoint 2

Depois desses passos devem estar preenchidas as seguintes propriedades do seu modelo.

1.2 Application Credentials

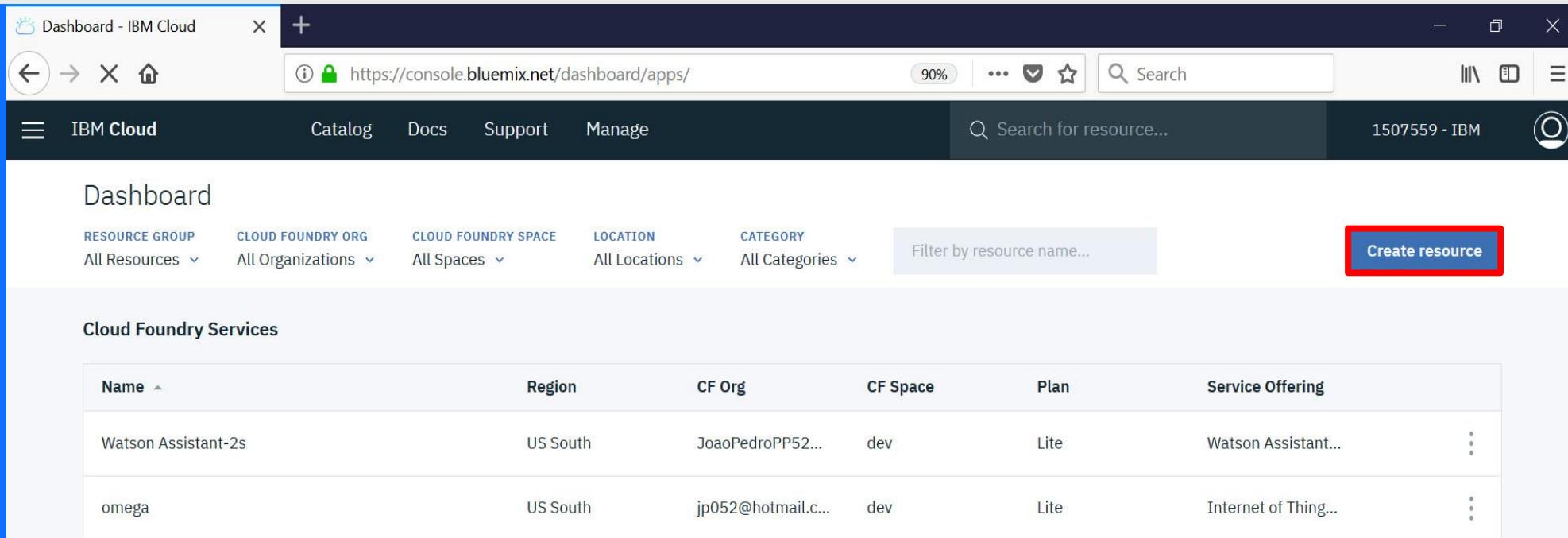
- API Key:
- Authentication Token (App):

Essas credenciais são necessárias para identificar e autorizar uma aplicação (ou função) a ler os dados dos seus dispositivos e mandar comandos.

3. Criando um dispositivo simulado com o Node-RED

3.1. Criando uma instância do Node-RED.

- Voltando ao Dashboard. Crie um novo recurso.



The screenshot shows the IBM Cloud Dashboard interface. At the top, there is a header with the title 'Dashboard - IBM Cloud', a search bar containing 'https://console.bluemix.net/dashboard/apps/', and a red box highlighting the 'Create resource' button on the right side of the header. Below the header, there is a navigation bar with links for 'Catalog', 'Docs', 'Support', and 'Manage'. A search bar labeled 'Search for resource...' is also present. The main area is titled 'Cloud Foundry Services' and contains a table with two rows of data. The columns in the table are 'Name', 'Region', 'CF Org', 'CF Space', 'Plan', and 'Service Offering'. The first row has a name of 'Watson Assistant-2s', a region of 'US South', a CF Org of 'JoaoPedroPP52...', a CF Space of 'dev', a Plan of 'Lite', and a Service Offering of 'Watson Assistant...'. The second row has a name of 'omega', a region of 'US South', a CF Org of 'jp052@hotmail.c...', a CF Space of 'dev', a Plan of 'Lite', and a Service Offering of 'Internet of Thing...'. Each row has a three-dot menu icon on the far right.

Name	Region	CF Org	CF Space	Plan	Service Offering
Watson Assistant-2s	US South	JoaoPedroPP52...	dev	Lite	Watson Assistant...
omega	US South	jp052@hotmail.c...	dev	Lite	Internet of Thing...

3. Criando um dispositivo simulado com o Node-RED

3.1. Criando uma instância do Node-RED.

- Voltando ao Dashboard. Crie um novo recurso.
- Na aba de busca digite “node-red” e ache o “Node-RED Starter”.

The screenshot shows the IBM Cloud Catalog interface. The search bar at the top contains the text "node-red". Below the search bar, there is a sidebar with "All Categories (2)" and a list of categories: Compute, Containers, Networking, Storage, AI, Analytics, Databases, Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits (2), Web and Mobile, and Web and Application. The main content area is titled "Starter Kits" and lists two items: "Internet of Things Platform Starter" and "Node-RED Starter". The "Node-RED Starter" item is highlighted with a red rectangular box. It includes a small icon of a wrench and screwdriver, the name "Node-RED Starter", the status "Lite • Community", and a brief description: "This application demonstrates how to run the Node-RED open-source project within IBM Cloud." At the bottom right, there is a section titled "Looking for more?" with a flask icon and a link to "IBM Cloud Experimental Services".

3. Criando um dispositivo simulado com o Node-RED

3.1. Criando uma instância do Node-RED.

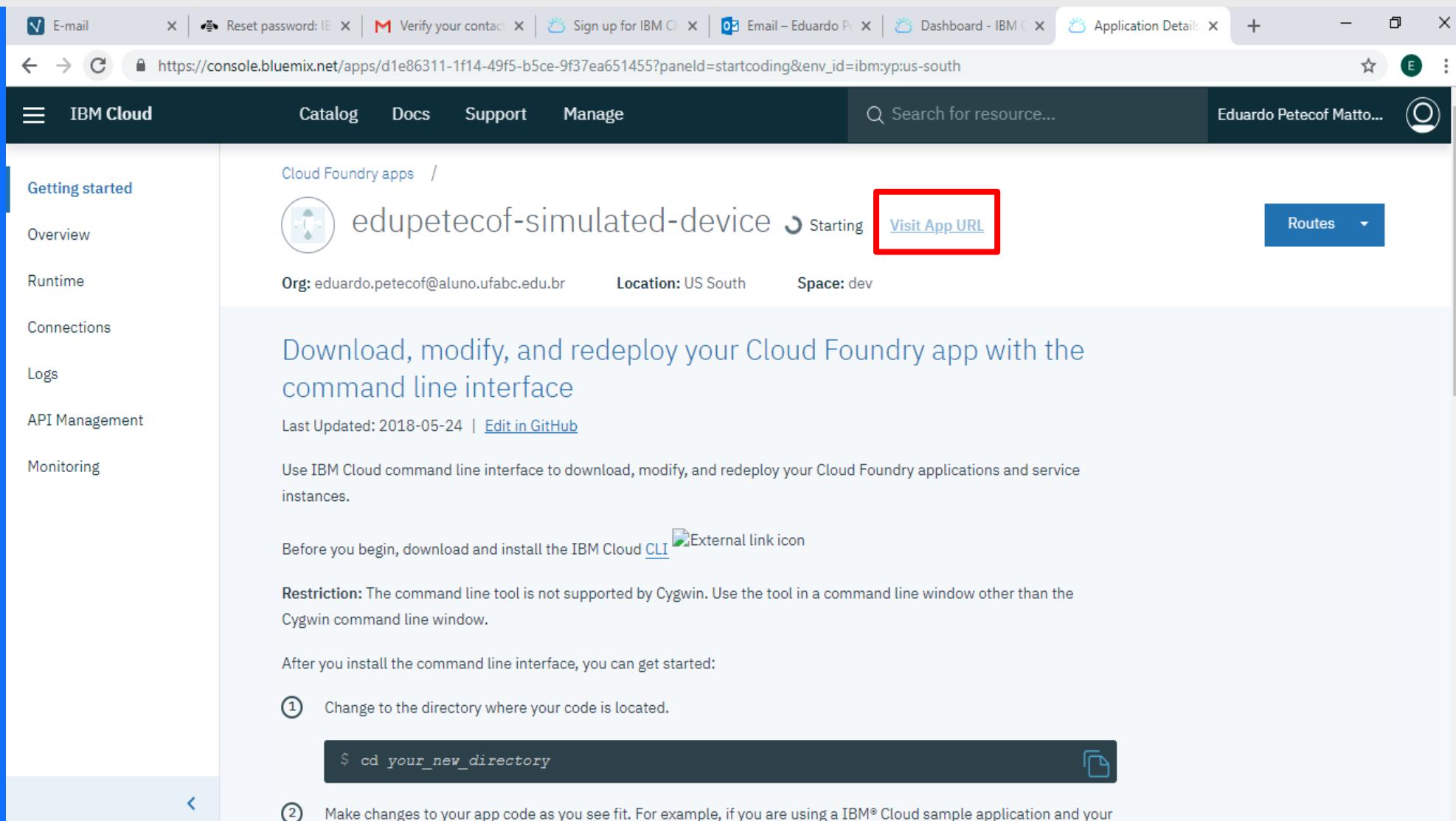
- Voltando ao Dashboard. Crie um novo recurso.
- Na aba de busca digite “node-red” e ache o “Node-RED Starter”.
- Dê um nome para a sua aplicação (recomenda-se que ele tenha uma referência ao seu nome) e crie.

The screenshot shows a browser window with four tabs: "Node-RED Starter - IBM Cloud", "IBM Watson IoT Platform", "Node-RED : device-simulator-d", and "New Tab". The main content is the IBM Cloud Catalog page for the "Node-RED Starter" application. The "App name:" field contains "epetecof-simulated-device" and the "Create" button at the bottom right is highlighted with a red box. Other fields include "Host name:", "Domain:", "Choose a region/location to deploy in:", "Choose an organization:", and "Choose a space:". The "Selected Plan:" dropdown shows "SDK for Node.js™" and "Cloudant". Navigation links at the bottom include "Need Help?", "Contact IBM Cloud Support", "Estimate Monthly Cost", and "Cost Calculator".

3. Criando um dispositivo simulado com o Node-RED

3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”



The screenshot shows the IBM Cloud Application Details page for an application named "edupetecof-simulated-device". The application is currently "Starting". A red box highlights the "Visit App URL" button. The page includes sections for Overview, Runtime, Connections, Logs, API Management, and Monitoring. It also provides instructions for using the IBM Cloud command line interface (CLI) and lists restrictions regarding Cygwin support. Step-by-step instructions for getting started are provided, including changing to the directory where the code is located and making changes to the app code.

Cloud Foundry apps /

edupetecof-simulated-device  Starting [Visit App URL](#) 

Org: eduardo.petecof@aluno.ufabc.edu.br Location: US South Space: dev

Download, modify, and redeploy your Cloud Foundry app with the command line interface

Last Updated: 2018-05-24 | [Edit in GitHub](#)

Use IBM Cloud command line interface to download, modify, and redeploy your Cloud Foundry applications and service instances.

Before you begin, download and install the IBM Cloud [CLI](#) 

Restriction: The command line tool is not supported by Cygwin. Use the tool in a command line window other than the Cygwin command line window.

After you install the command line interface, you can get started:

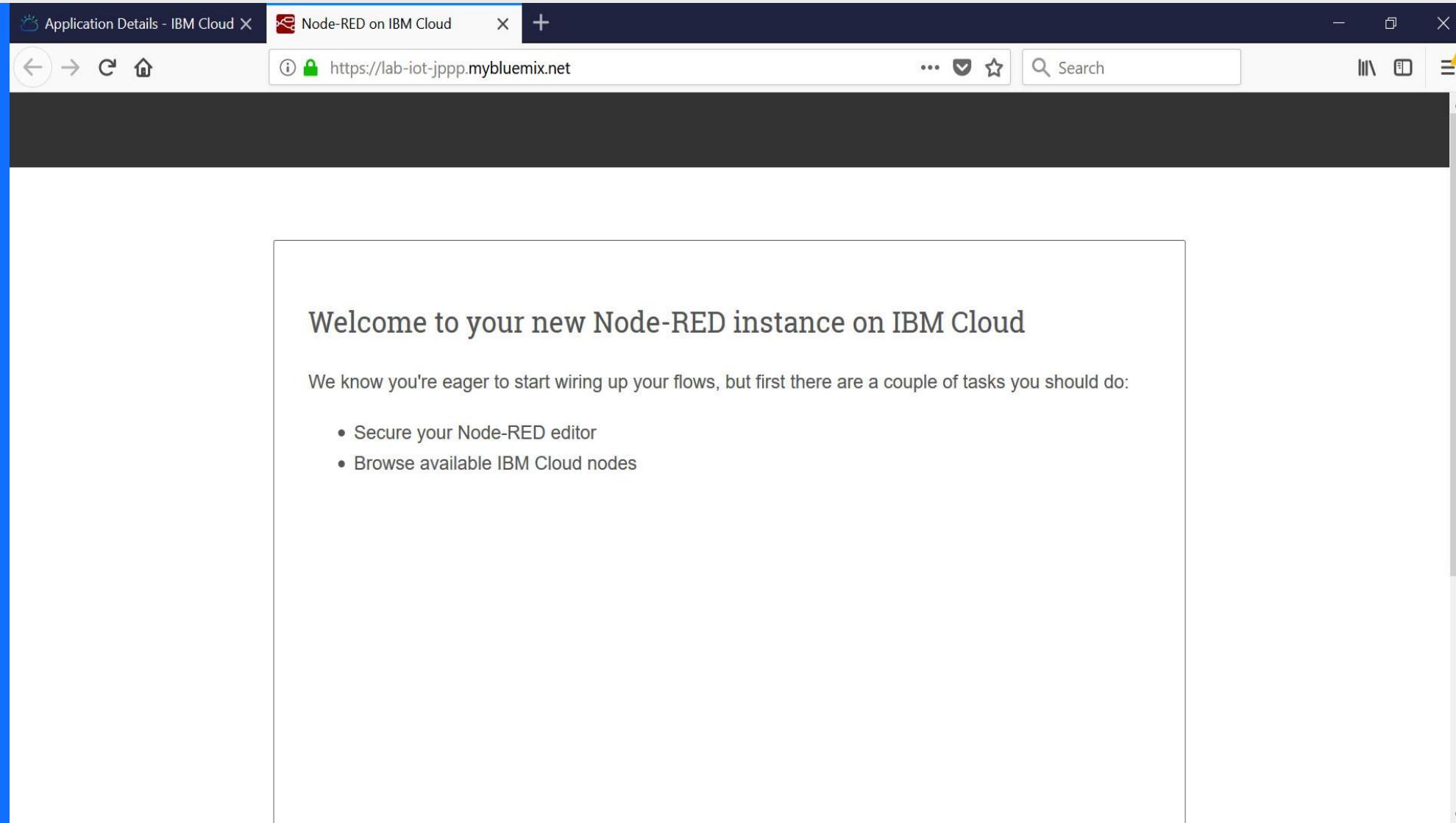
- ① Change to the directory where your code is located.

```
$ cd your_new_directory
```
- ② Make changes to your app code as you see fit. For example, if you are using a IBM® Cloud sample application and your

3. Criando um dispositivo simulado com o Node-RED

3.2. Configurando o seu Node-RED.

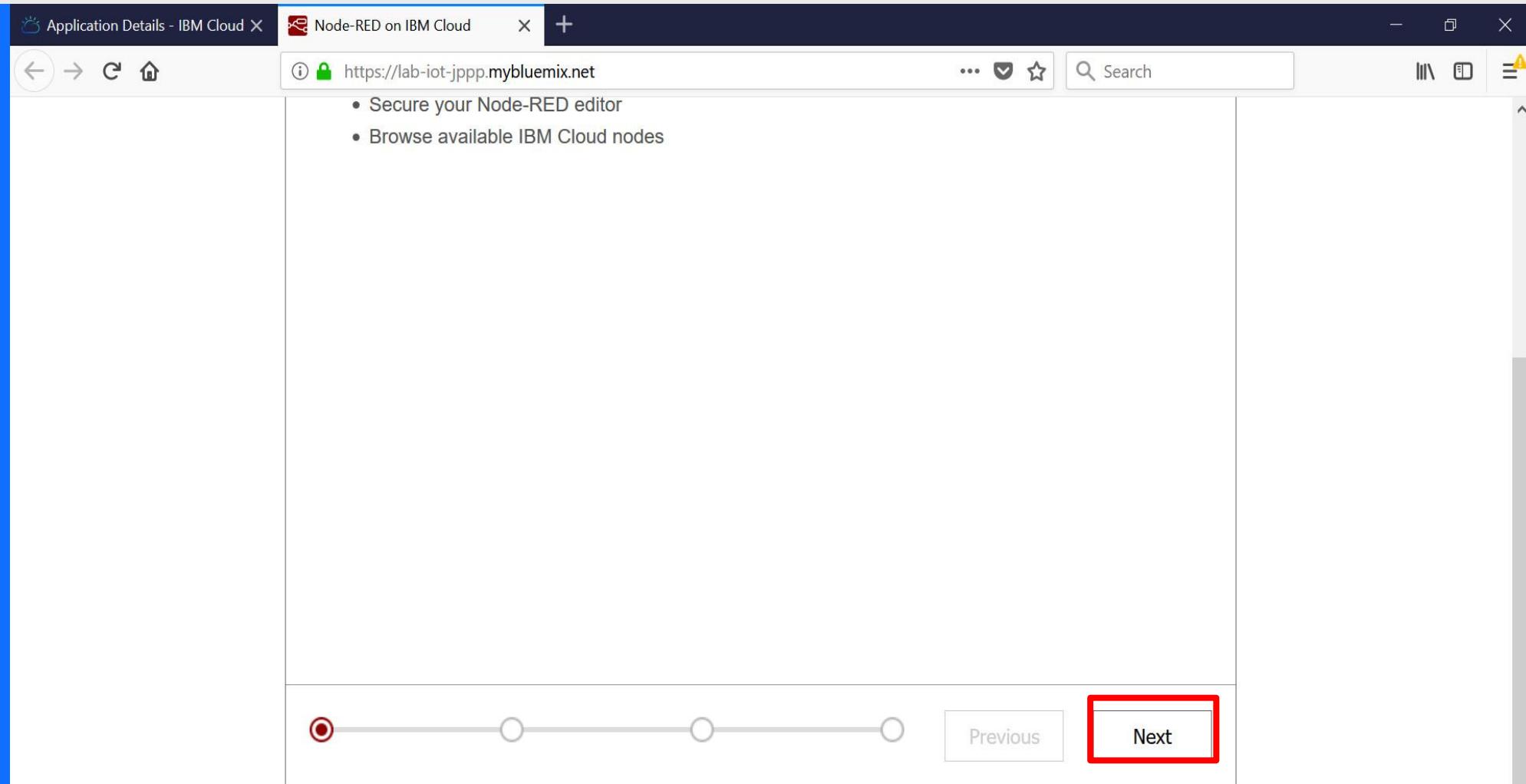
- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.



3. Criando um dispositivo simulado com o Node-RED

3.2. Configurando o seu Node-RED.

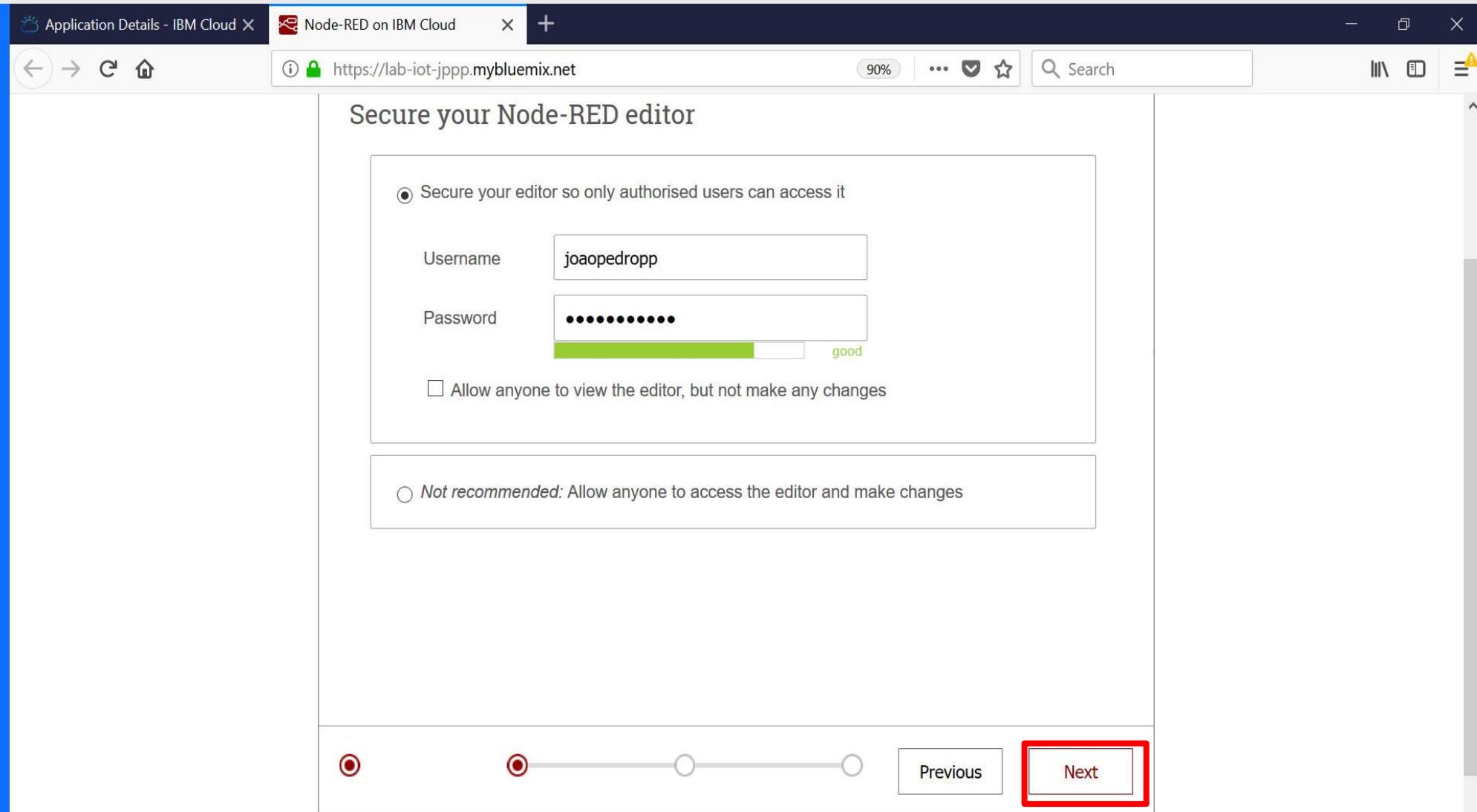
- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.



3. Criando um dispositivo simulado com o Node-RED

3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.
- Defina um username e uma senha para editors.



3. Criando um dispositivo simulado com o Node-RED

3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.
- Defina um username e uma senha para editors.
- Pule as próximas telas.

Application Details - IBM Cloud X Node-RED on IBM Cloud X +

https://lab-iot-jppp.mybluemix.net 90% Search

Browse available IBM Cloud nodes

There are lots of nodes available from the community that can be used to add more capabilities to your application. The list below is just a small selection.

You can find many more nodes on the [Flow Library](#).

You can use the Palette Manager built into editor to search for and install nodes. Alternatively, you can also edit your application's package.json file and adding them to the dependencies section.

node-red-dashboard
Quickly create dashboards driven by Node-RED

node-red-contrib-ibm-wiotp-device-ops
Perform device and gateway operations using the Watson IoT Platform

node-red-contrib-iot-virtual-device
Simulate device behavior and use it to run many device instances

node-red-contrib-objectstore
Store, delete and restore objects in the ObjectStore service

node-red-contrib-bluemix-hdfs

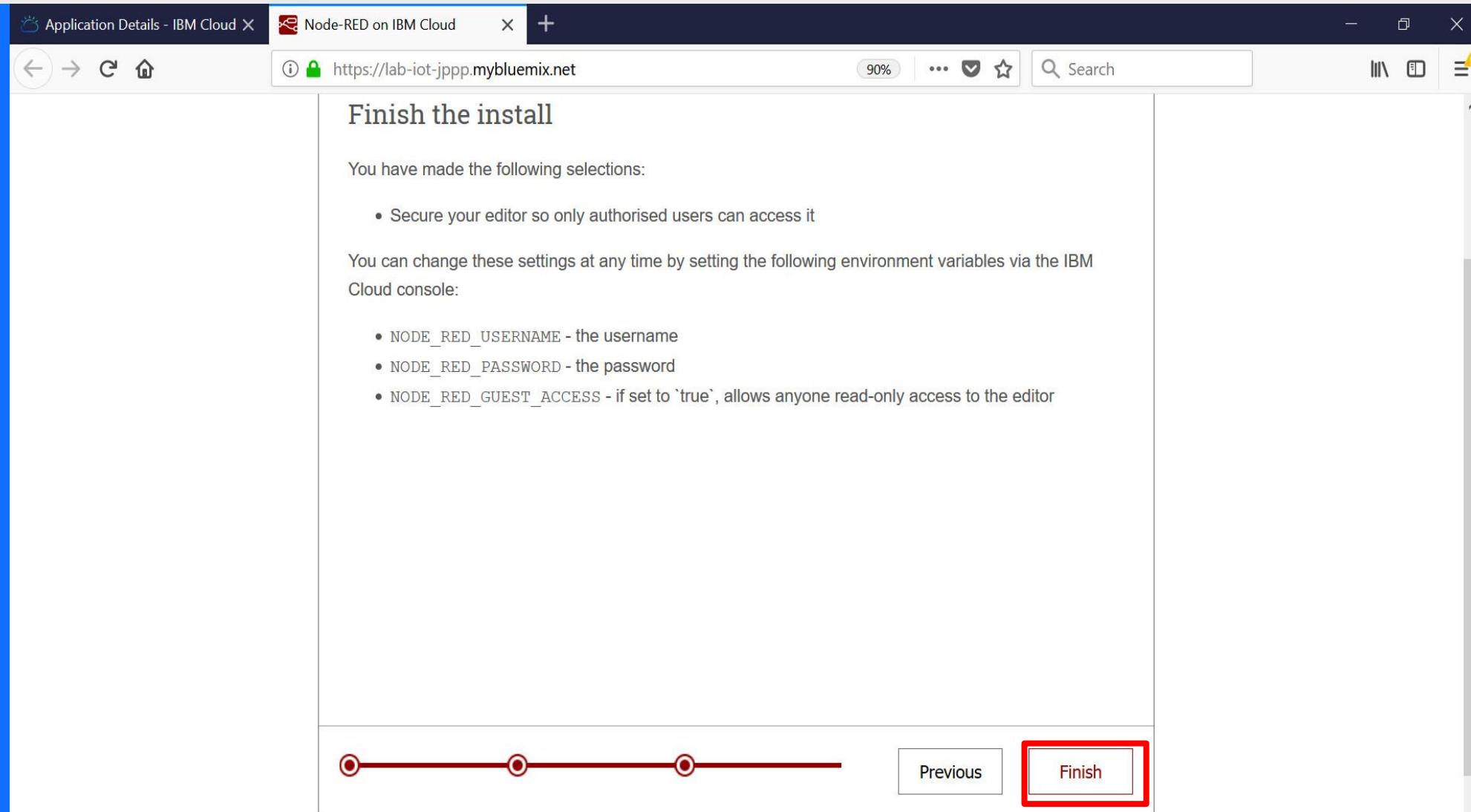
node-red-contrib-ibmpush

Previous Next

3. Criando um dispositivo simulado com o Node-RED

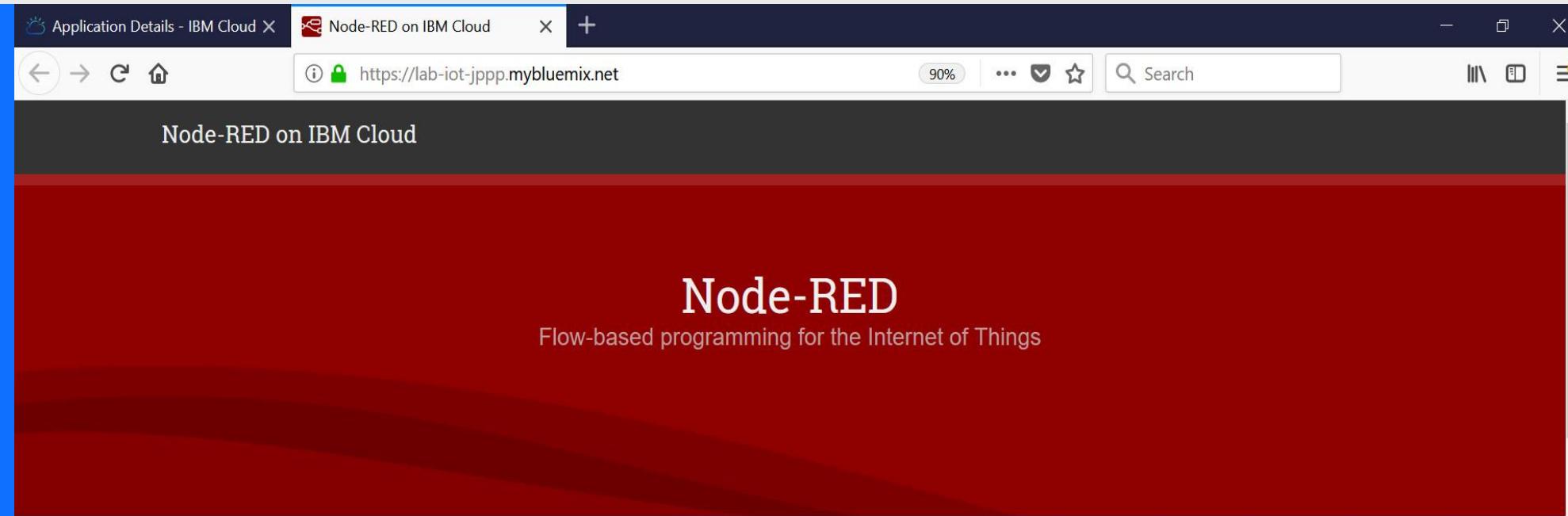
3.2. Configurando o seu Node-RED.

- Assim que a aplicação iniciar (aguarde alguns segundos), clique em “Visit App URL”
- Siga as etapas de segurança.
- Defina um username e uma senha para editors.
- Pule as próximas telas.



3. Criando um dispositivo simulado com o Node-RED

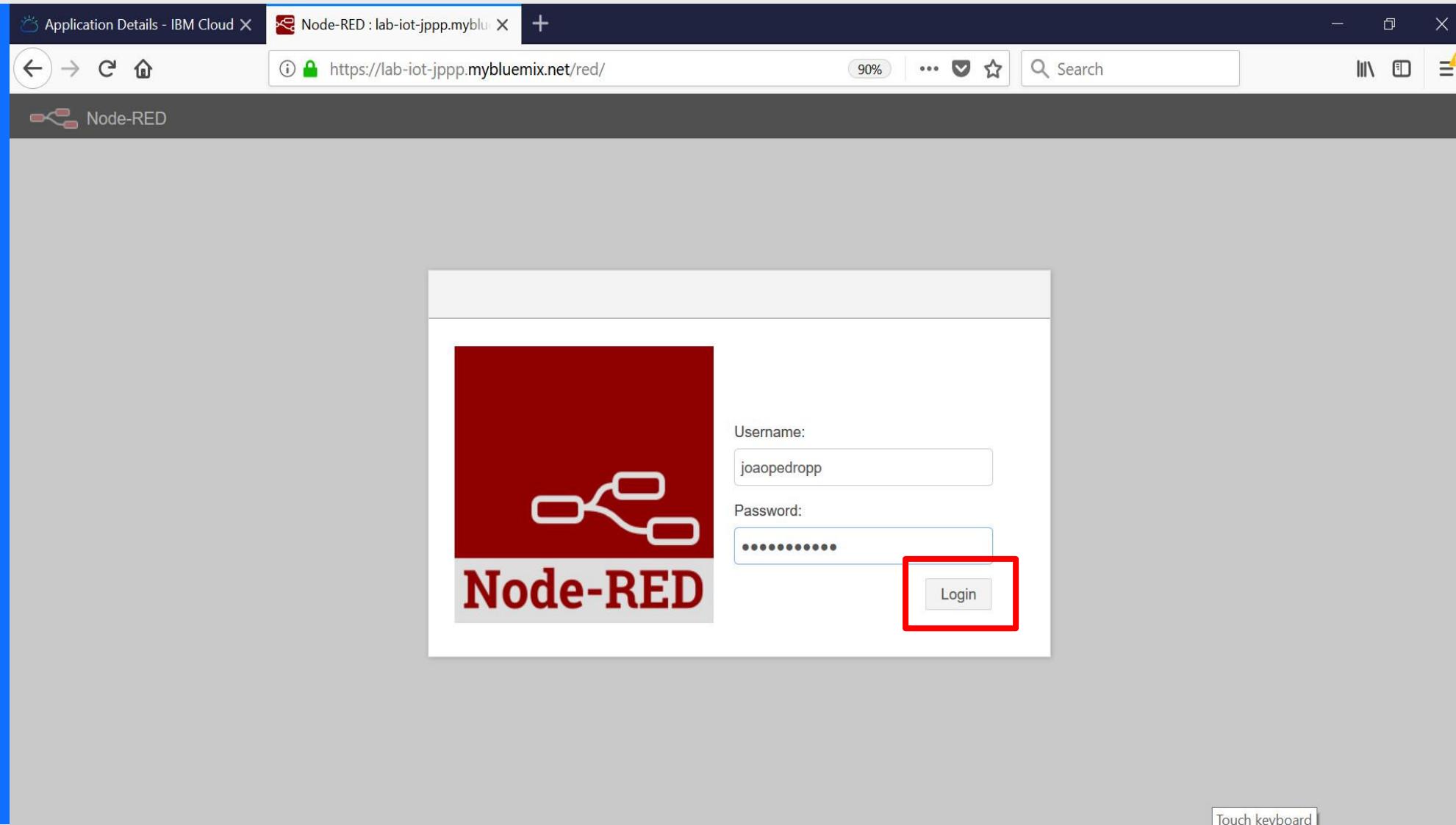
3.2. Configurando o seu Node-RED.



3. Criando um dispositivo simulado com o Node-RED

3.2. Configurando o seu Node-RED.

- Faça login com o username e senha escolhidos anteriormente.



3. Criando um dispositivo simulado com o Node-RED

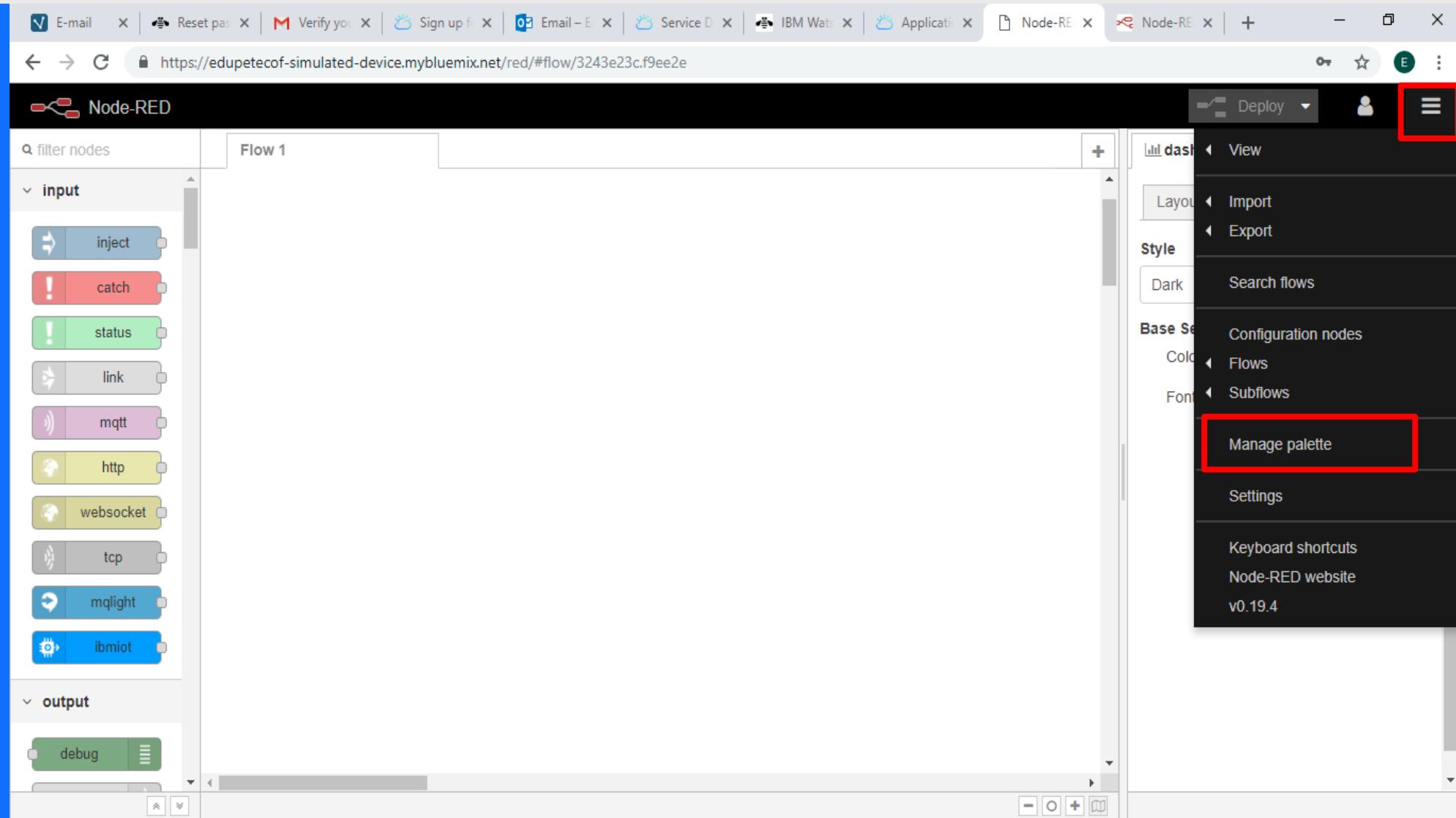
3.3. Instale o dashboard.

The screenshot shows the Node-RED interface running on IBM Cloud. The browser title is "Node-RED : lab-iot-jppp.mybluemix.net". The main area displays "Flow 1" which contains a single "inject" node connected to a "debug" output node. The left sidebar lists various input nodes: inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, and ibmiot. The right sidebar shows the "info" panel with the flow details: Flow ID "1ebc4a3c.e6699e", Name "Flow 1", and Status "Enabled". A note at the bottom right says: "You can confirm your changes in the node edit tray with **ctrl-enter** or cancel them with **ctrl-escape**".

3. Criando um dispositivo simulado com o Node-RED

3.3. Instale o dashboard.

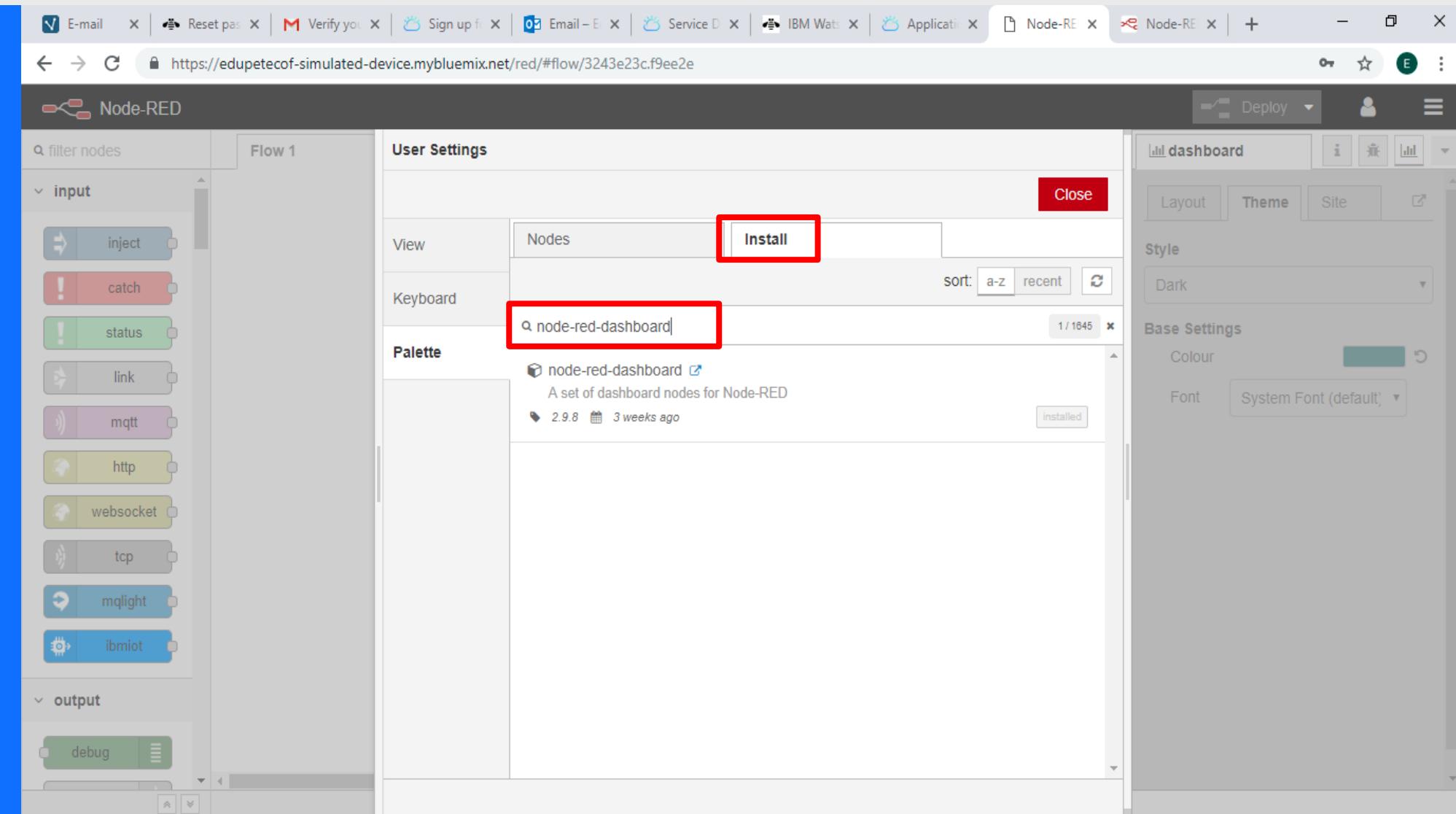
- No menu à direta, navegue para “Manage Palette”.



3. Criando um dispositivo simulado com o Node-RED

3.3. Instale o dashboard.

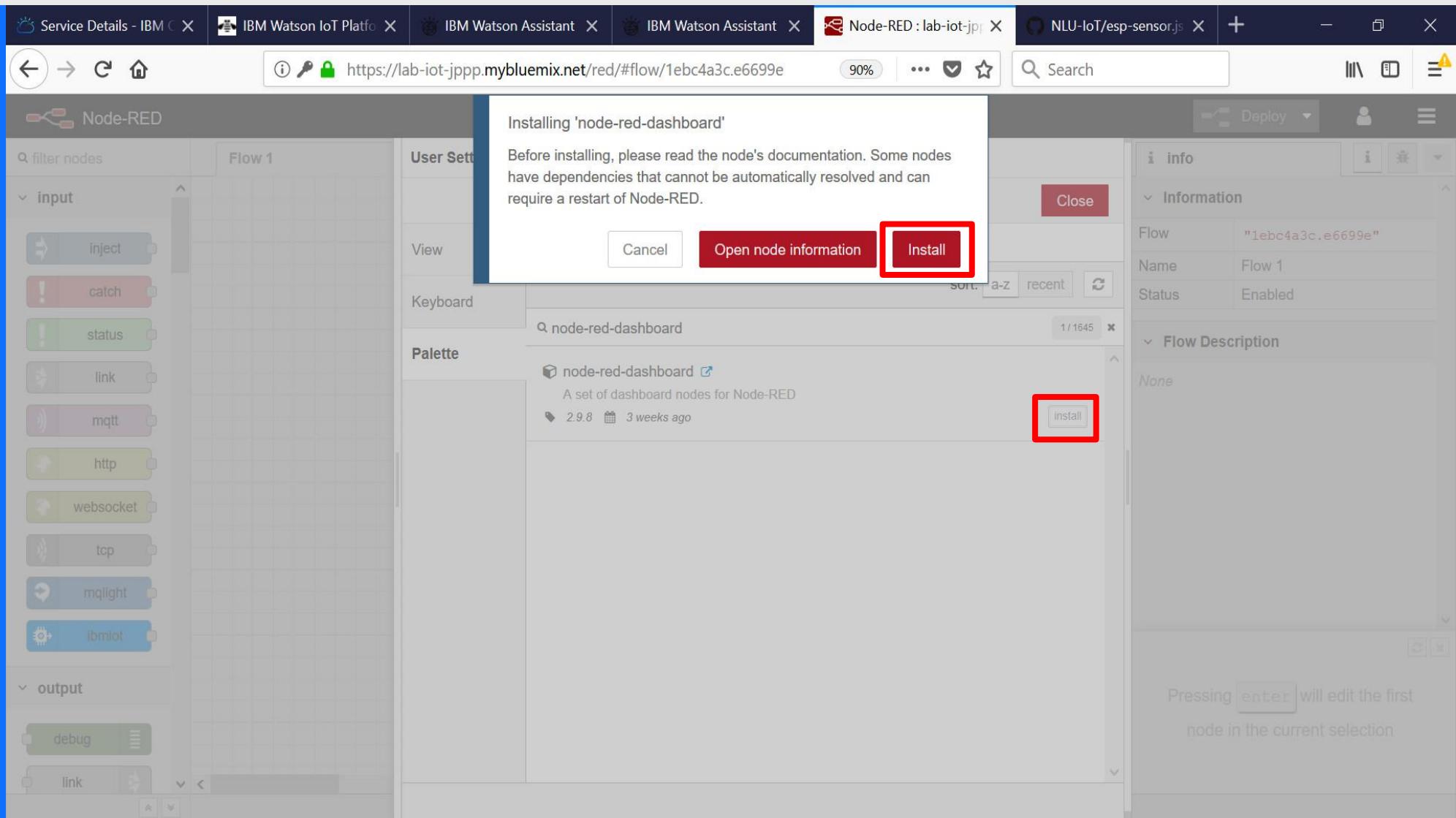
- No menu à direta, navegue para “Manage Palette”.
- Na aba de busca procure por “node-red-dashboard”.



3. Criando um dispositivo simulado com o Node-RED

3.3. Instale o dashboard.

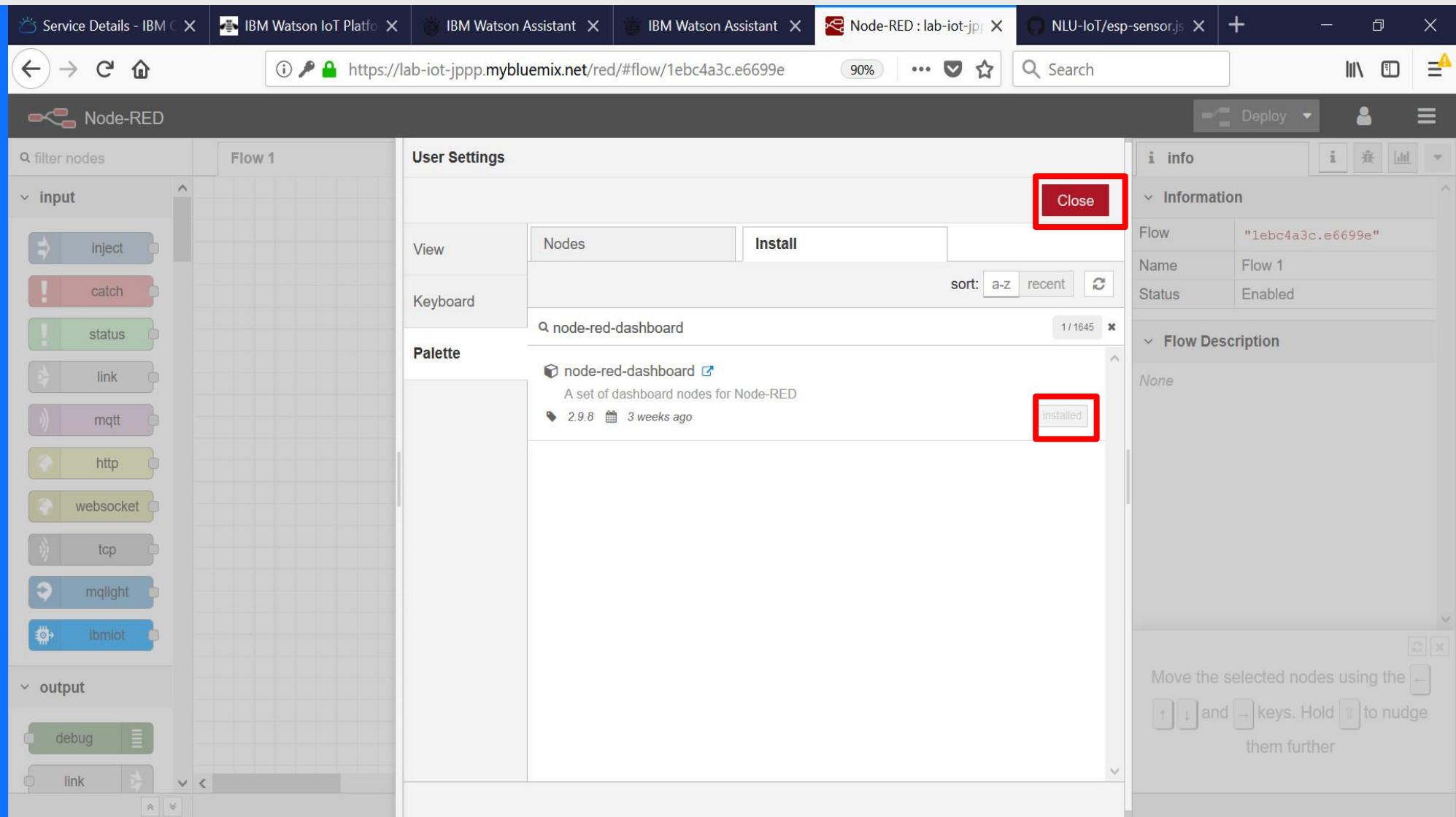
- No menu à direta, navegue para “Manage Palette”.
- Na aba de busca procure por “node-red-dashboard”.
- Instale o pacote.



3. Criando um dispositivo simulado com o Node-RED

3.3. Instale o dashboard.

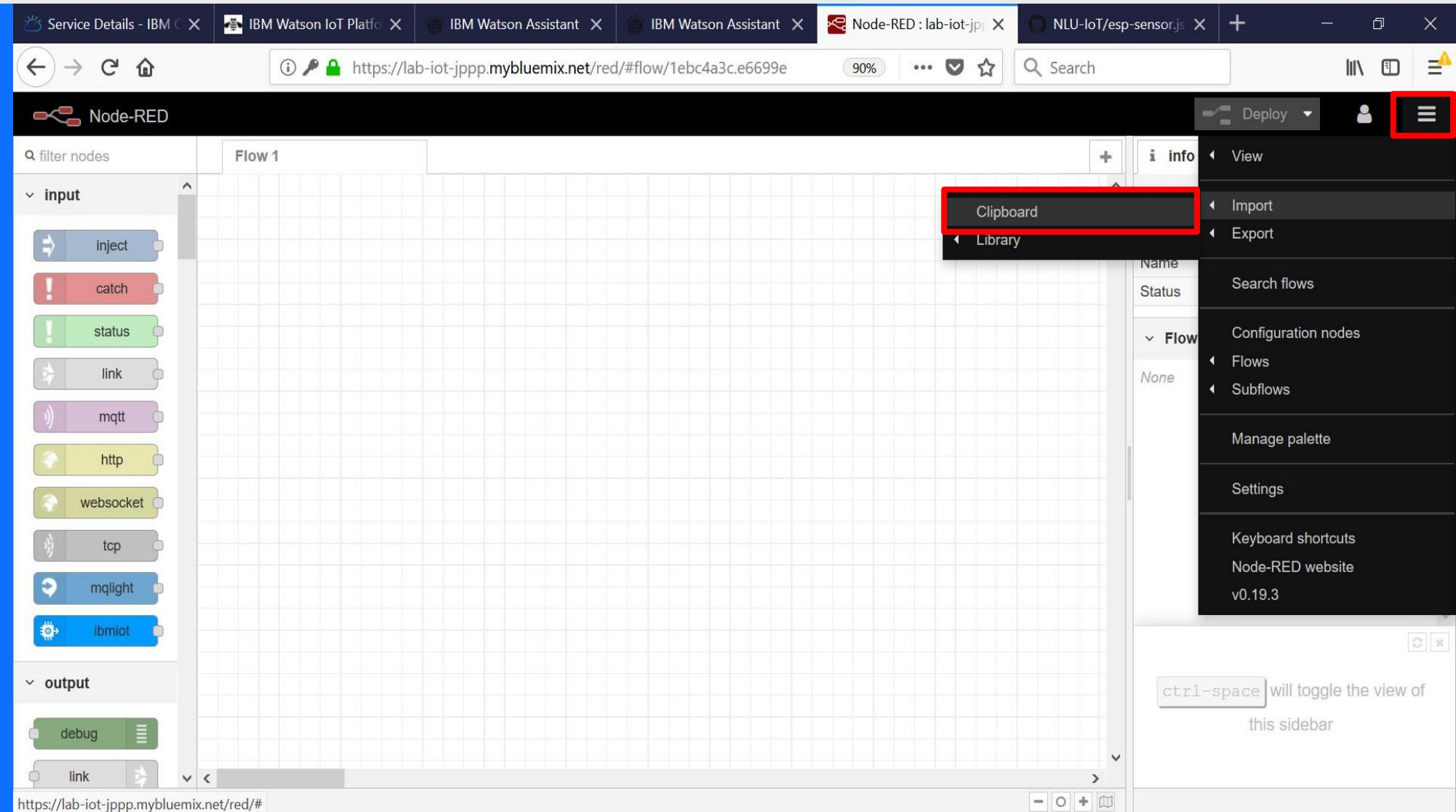
- No menu à direta, navegue para “Manage Palette”.
- Na aba de busca procure por “node-red-dashboard”.
- Instale o pacote. Volte para a tela principal.



3. Criando um dispositivo simulado com o Node-RED

3.4. Import o flow.

- No menu à direta,
navegue para
“Import >
Clipboard”.



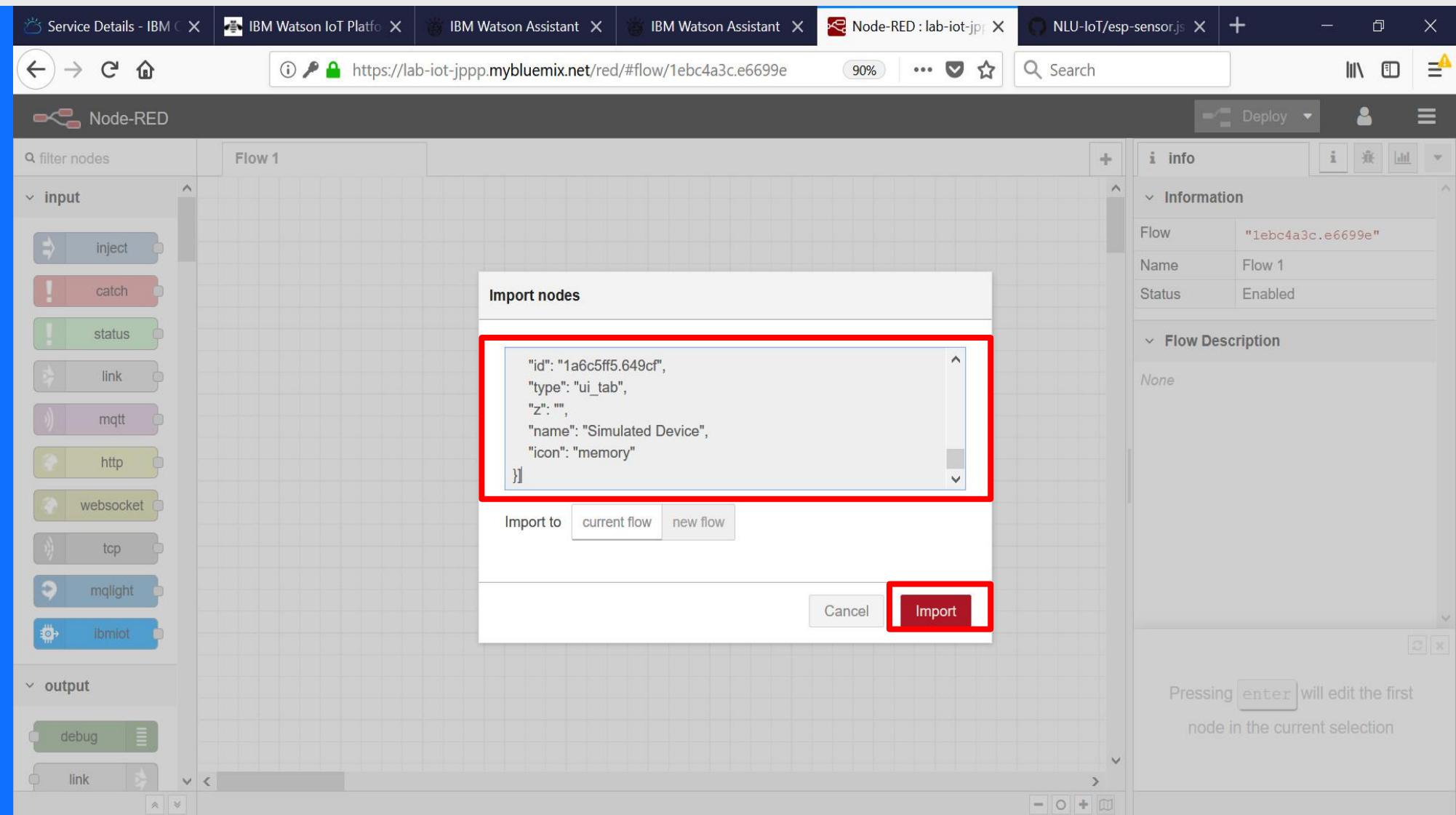
3. Criando um dispositivo simulado com o Node-RED

3.4. Import o flow.

- No menu à direta, navegue para “Import > Clipboard”.

- Copie o código no git:
<https://github.com/Bluedev-BR/NLU-IoT/blob/master/Node-RED/hands-on-flow.json>

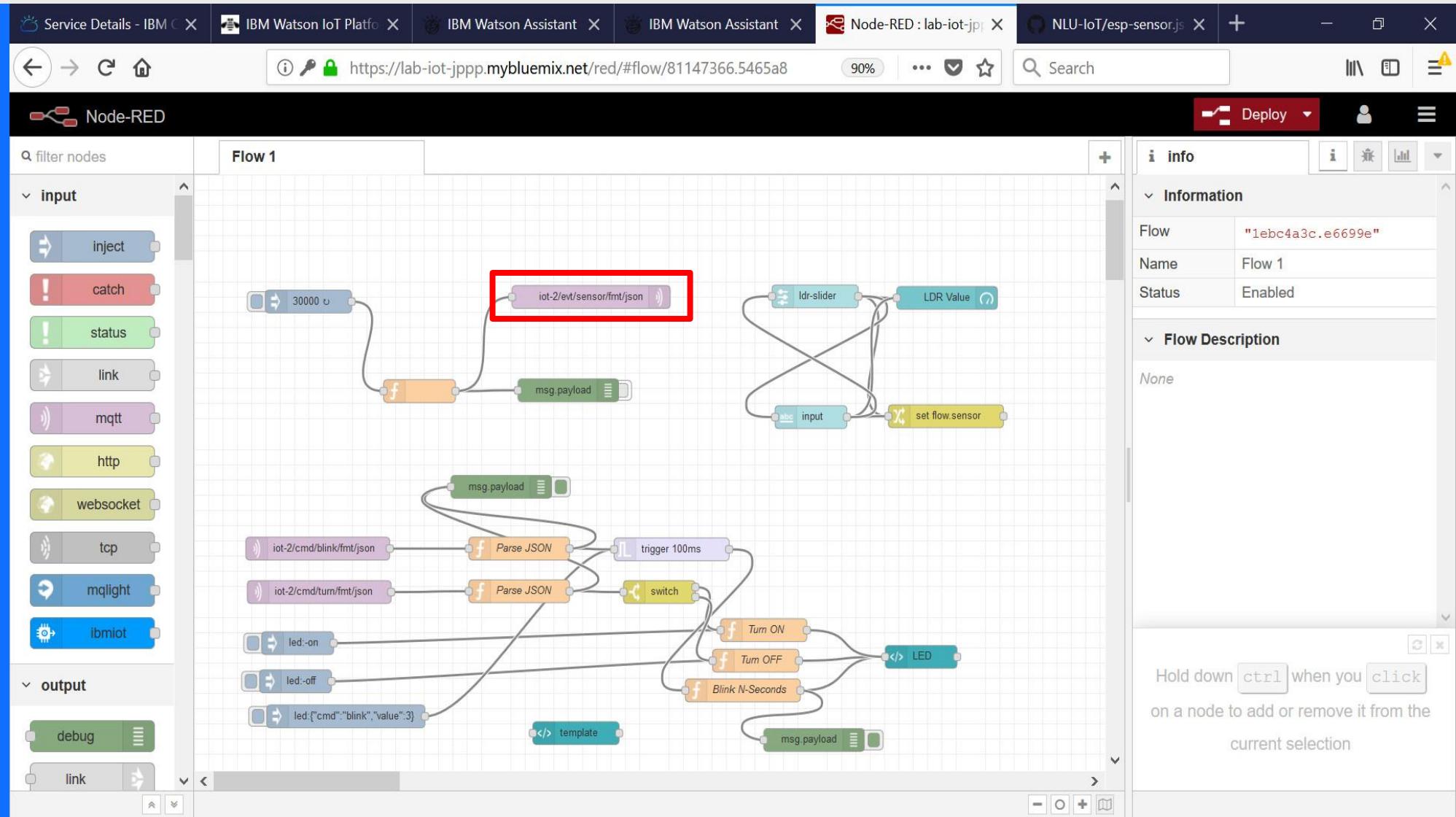
- Cole na caixa e importe.



3. Criando um dispositivo simulado com o Node-RED

3.5. Configurando a comunicação MQTT.

- Clique em um nó de comunicação MQTT.



3. Criando um dispositivo simulado com o Node-RED

3.5. Configurando a comunicação MQTT.

- Clique em um nó de comunicação MQTT.
- Edite as configurações do broker.

3.5. Configurando a comunicação MQTT.

- Clique em um nó de comunicação MQTT.

- Edite as configurações do broker.

Node-RED Flow Details:

- Input Nodes:** inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, ibmiot.
- Output Node:** debug, link.
- MQTT Configuration (highlighted):**
 - Server:** d:{your_org}:{your_device_type}:{your_d} (with edit icon)
 - Topic:** iot-2/evt/sensor/fmt/json
 - QoS:** 0
 - Retain:** true
 - Name:** Name
- Tip:** Leave topic, qos or retain blank if you want to set them via msg properties.

3. Criando um dispositivo simulado com o Node-RED

3.5. Configurando a comunicação MQTT.

- Clique em um nó de comunicação MQTT.
- Edite as configurações do broker.
- Coloque as informações da plataforma de IoT. O nome da sua org, o seu Device type e o Device ID.

The screenshot shows the Node-RED interface with a flow editor and a configuration panel for an mqtt-broker node.

Flow Editor: On the left, there's a sidebar with categories like 'input' (inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, ibmiot) and 'output' (debug, link). The main area shows a flow starting with an 'inject' node, followed by a 'status' node, a '30000' timer node, and an 'f' function node. This leads to an 'iot-2/cmd/blink/fmt/json' message node, which then branches to two 'led-on' and 'led-off' nodes, and finally to a 'led:[{"cmd": "blink", "value": 3}]' message node.

Configuration Panel: The right side shows the 'Edit mqtt out node > Edit mqtt-broker node' dialog. It has tabs for 'Connection', 'Security', and 'Messages'. Under 'Connection', the 'Server' field is set to '{your_org}.messaging.internetofthings' (highlighted with a red box), and the 'Port' is 1883. The 'Client ID' field is set to 'd:{your_org}:{your_device_type}:{your_device_id}' (also highlighted with a red box). Other settings include 'Keep alive time (s)' at 60, 'Use clean session' checked, and 'Use legacy MQTT 3.1 support' checked. Buttons for 'Delete', 'Cancel', and 'Update' are at the top right.

Info Panel: On the far right, there's an 'info' panel with 'Information' and 'Node Help' sections. The 'Information' section shows the node ID ('#12dc1008.03a508'), type ('mqtt-broker'), and a 'show more' button. The 'Node Help' section provides instructions for connecting to an MQTT broker, mentioning single connections and reuse by MQTT In and MQTT Out nodes. It also notes the generation of random Client IDs if none are set and configured for Clean Session connections.

3. Criando um dispositivo simulado com o Node-RED

3.5. Configurando a comunicação MQTT.

- Adicione a autenticação: No campo “Username”, escreva “use-auth-token”. No campo “Password”, use o token gerado automaticamente no momento da criação do device.

The screenshot shows the Node-RED interface within an IBM Cloud application. The left sidebar lists various node categories: input (inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, ibmiot) and output. The main canvas displays a flow labeled "Flow 1" with several nodes connected. A modal window titled "Edit mqtt out node > Edit mqtt-broker node" is open, showing configuration fields for a connection. The "Security" tab is selected, displaying "Username: use-token-auth" and "Password: [REDACTED]". The "Connection" tab is also visible. On the right side, there is a panel with "Information" and "Node Help" sections, and a bottom section with instructions about quick-wiring.

Application Details - IBM Cloud X Node-RED : lab-iot-jppp.mybluemix.net X + https://lab-iot-jppp.mybluemix.net/red/#flow/fc46bad1.3f65d8 ... Search Deploy

3.5. Configurando a comunicação MQTT.

Edit mqtt out node > Edit mqtt-broker node

Name Name

Connection Security Messages

Username use-token-auth

Password [REDACTED]

3 nodes use this config On all flows

info

Information

Node "53504ed.2d064b"
Type mqtt-broker

Node Help

Configuration for a connection to an MQTT broker.

This configuration will create a single connection to the broker which can then be reused by **MQTT In** and **MQTT Out** nodes.

The node will generate a random Client ID if one is not set and the node is configured to

Hold down **ctrl** when you **click** on a node port to enable quick-wiring

3. Criando um dispositivo simulado com o Node-RED

3.5. Configurando a comunicação MQTT.

- Adicione a autenticação: No campo “Username”, escreva “use-auth-token”. No campo “Password”, use o token gerado automaticamente no momento da criação do device.

- Clique no botão “Update”.

The screenshot shows the IBM Cloud Node-RED interface. On the left, there's a sidebar with categories like 'input' (inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, ibmiot) and 'output'. In the center, a flow diagram titled 'Flow 1' is visible. On the right, a modal window titled 'Edit mqtt out node > Edit mqtt-broker node' is open. It has tabs for 'Connection' (selected), 'Security', and 'Messages'. Under 'Security', there are fields for 'Username' (set to 'use-token-auth') and 'Password' (represented by a series of dots). A red box highlights the 'Update' button at the top right of the modal. To the right of the modal, there's an 'info' panel showing 'Node' (53504ed.2d064b) and 'Type' (mqtt-broker). Below it is a 'Node Help' section with instructions about the configuration.

3. Criando um dispositivo simulado com o Node-RED

3.5. Configurando a comunicação MQTT.

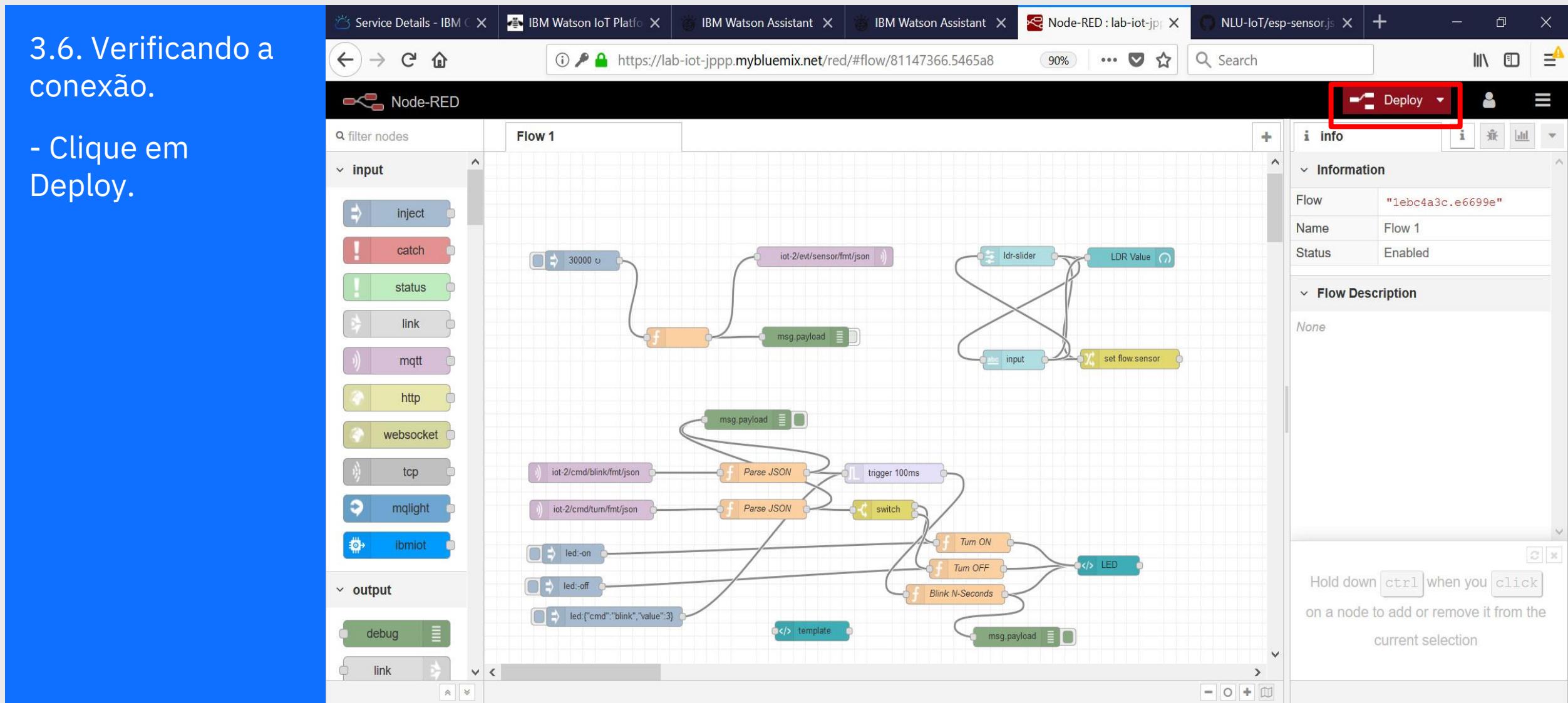
- Verifique o tópico: Deve ser algo parecido com “iot-2/evt/sensor/fmt/json”.
- Clique em “Done”

The screenshot shows the Node-RED interface with a flow titled "Flow 1". The flow consists of various nodes connected by wires. On the left, there's a sidebar with categories like "input" and "output". In the main area, an "inject" node is connected to a "catch" node. This is followed by a "status" node, a "link" node, and an "mqtt" node. Below these are "http", "websocket", "tcp", "mqlight", and "ibmiot" nodes. The "output" category includes a "debug" node and a "link" node. A specific "mqtt out" node is highlighted with a red box around its "Topic" field, which contains "iot-2/evt/sensor/fmt/json". The "Done" button for this node is also highlighted with a red box. The "Edit mqtt out node" dialog is open, showing fields for "Server", "Topic", "QoS", and "Name". A tip at the bottom of the dialog says: "Tip: Leave topic, qos or retain blank if you want to set them via msg properties." To the right of the editor, there's a panel with "Information" and "Node Help" sections, and a "node settings" section at the bottom.

3. Criando um dispositivo simulado com o Node-RED

3.6. Verificando a conexão.

- Clique em Deploy.

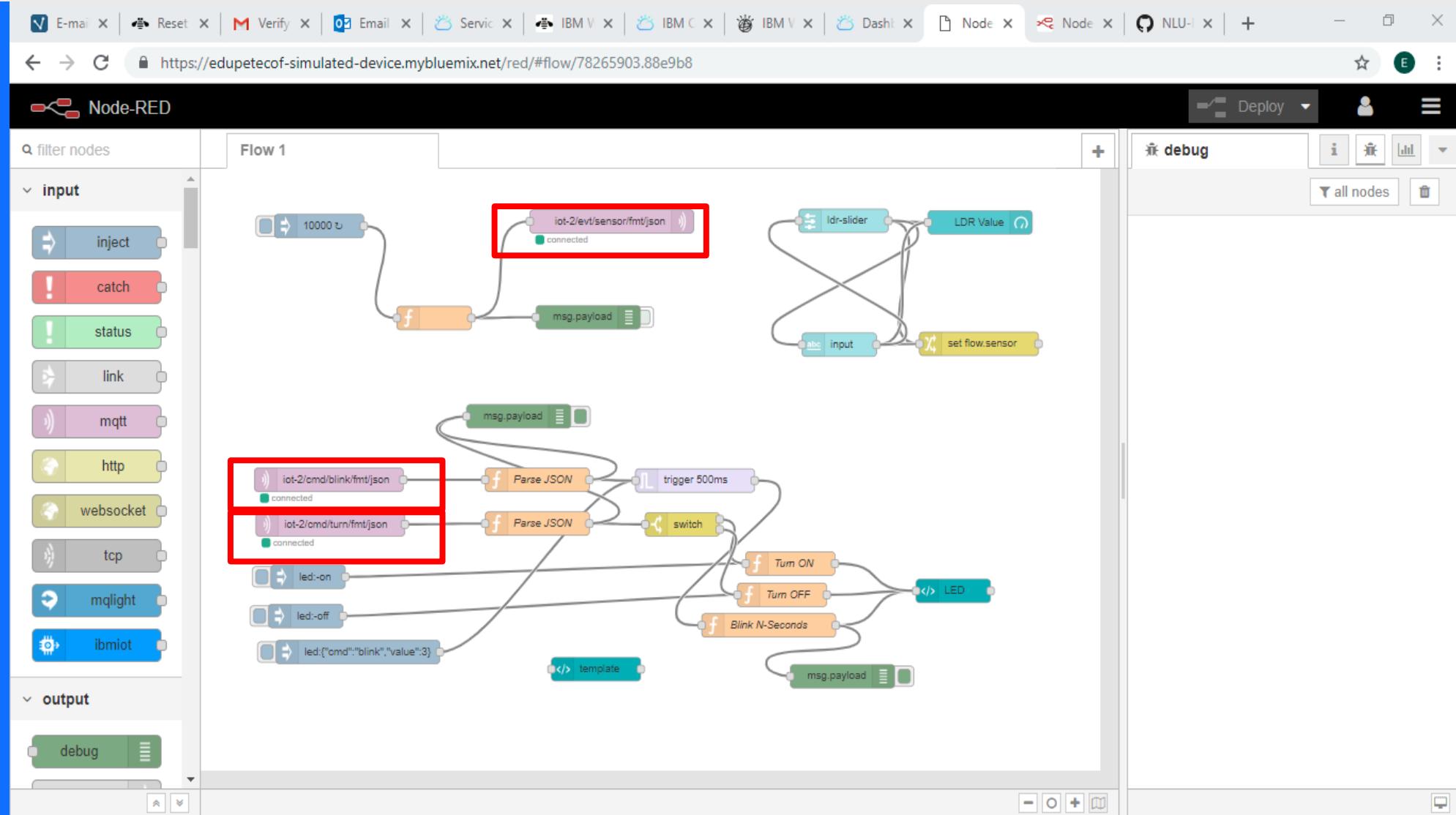


3. Criando um dispositivo simulado com o Node-RED

3.6. Verificando a conexão.

- Clique em Deploy.

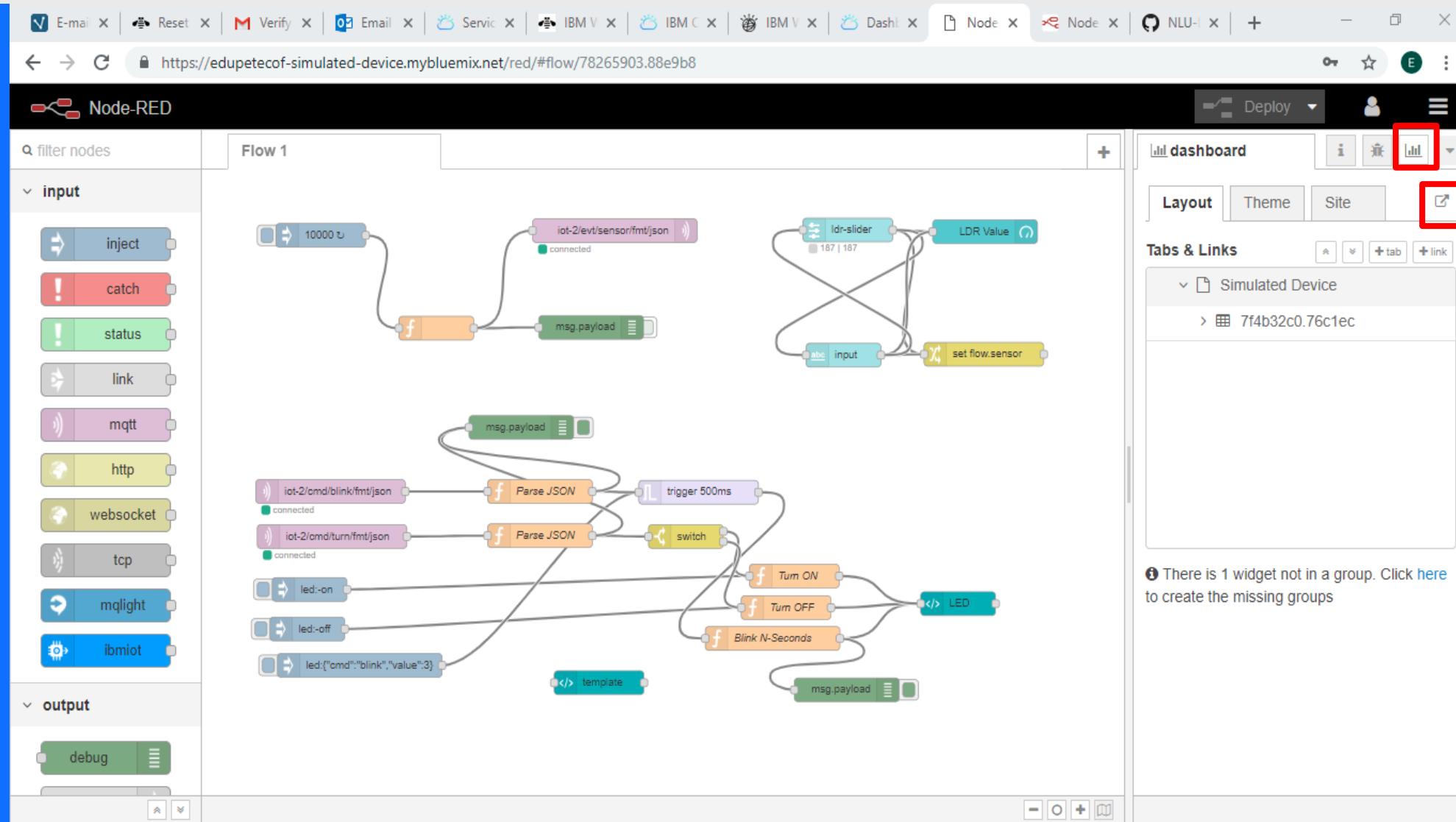
- Observe a conexão dos nós MQTT.



3. Criando um dispositivo simulado com o Node-RED

3.6. Verificando a conexão.

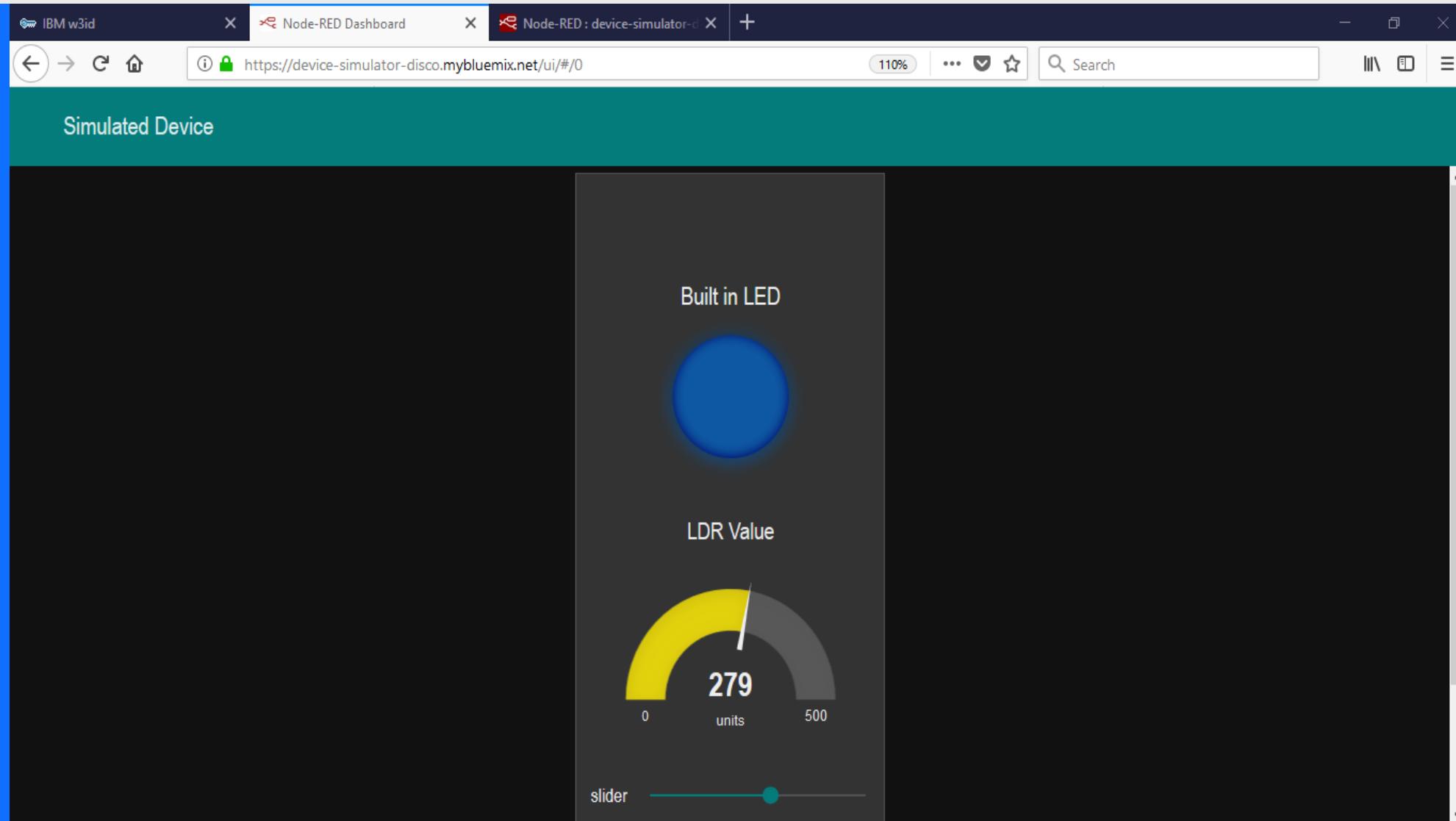
- Clique em Deploy.
- Observe a conexão dos nós MQTT.
- Abra o Dashboard acessando as abas do lado direito e clicando no botão de URL logo abaixo.



3. Criando um dispositivo simulado com o Node-RED

3.6. Verificando a conexão.

- Clique em Deploy.
- Observe a conexão dos nós MQTT.
- Abra o Dashboard acessando as abas do lado direito e clicando no botão de URL abaixo.
- Mova o slider para mudar o valor do sensor.



3. Criando um dispositivo simulado com o Node-RED

3.6. Verificando a conexão.

- Verifique a conexão na plataforma de IoT.

The screenshot shows the 'IBM Watson IoT Platform' dashboard with the URL <https://7y60pl.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2>. The top navigation bar includes links for E-mail, Reset, Verify, Email, Service, IBM V, IBM C, IBM V, IBM V, Dash, Node, Node, NLU-, and a plus sign. The user is logged in as eduardo.petecof@aluno.ufabc.edu.br with ID: 7y60pl. The main title is 'IBM Watson IoT Platform' and the sub-section is 'Browse Devices'. On the left, there is a vertical sidebar with icons for Home, Action, Device Types, All Devices, Diagnose, and a search bar. The main content area is titled 'Browse Devices' with a sub-instruction: 'This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.' A table lists one device: 'my-simulated-device' (Device Type: Simulated-Device, Class ID: Device, Date Added: Sep 30, 2018 4:27 PM). The first column of the table has a red box around its header and the first row's first cell.

Device ID	Device Type	Class ID	Date Added
my-simulated-device	Simulated-Device	Device	Sep 30, 2018 4:27 PM

3. Criando um dispositivo simulado com o Node-RED

3.6. Verificando a conexão.

- Verifique a conexão na plataforma de IoT.
- Selecione o Device e veja os eventos chegando na aba “Recent Events”.

The screenshot shows the IBM Watson IoT Platform dashboard. At the top, there are four tabs: "Service Details - IBM Cloud", "IBM Watson IoT Platform", "Node-RED Dashboard", and "Node-RED : device-simulator-". The "IBM Watson IoT Platform" tab is active, displaying the "Browse" section. A sidebar on the left contains icons for various device types. The main area shows a table with columns: Device ID, Device Type, Class ID, and Date Added. One row is selected, showing "teste_sim" as the Device ID, "Simulated" as the Device Type, "Device" as the Class ID, and "Aug 28, 2018 6:19 PM" as the Date Added. Below the table, there are tabs for Identity, Device Information, Recent Events, State, and Logs. The "Recent Events" tab is selected. It displays a table with columns: Event, Value, Format, and Last Received. Two rows are listed, both labeled "sensor". The first row has a value of "[{"d": [{"counter": 360000, "sensor": 279}]}]" and was received "a few seconds ago". The second row has a value of "[{"d": [{"counter": 330000, "sensor": 279}]}]" and was received "a few seconds ago". The "Value" column for the second row is highlighted with a red box. On the right side of the dashboard, there is a small icon of a gear with three dots.

Event	Value	Format	Last Received
sensor	[{"d": [{"counter": 360000, "sensor": 279}]}]	json	a few seconds ago
sensor	[{"d": [{"counter": 330000, "sensor": 279}]}]	json	a few seconds ago

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.

The screenshot shows the IBM Cloud dashboard interface. At the top, there are three tabs: 'Dashboard - IBM Cloud', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The main content area is titled 'Dashboard' and includes filters for 'RESOURCE GROUP', 'CLOUD FOUNDRY ORG', 'CLOUD FOUNDRY SPACE', 'LOCATION', and 'CATEGORY'. A prominent red box highlights the 'Create resource' button in the top right corner. Below this, a message box says 'Learn more about migrating your eligible service instances to resource groups.' Under the 'Cloud Foundry Applications' section, a table shows one application: 'Lab-IoT-JPPP' in 'US South' region, 'JoaoPedroPP52...' org, 'dev' space, 256 MB memory, and 'Running (1/1)' status. The 'Clusters' section shows one cluster: 'mycluster' in 'US South' region, 1 node, '1.10.7_1520' Kube version, and 'Normal' status. The bottom section, 'Cloud Foundry Services', is partially visible. On the far right, a vertical bar has 'FEEDBACK' at the top.

Name	Region	CF Org	CF Space	Memory (MB)	Status
Lab-IoT-JPPP	US South	JoaoPedroPP52...	dev	256	Running (1/1)

Name	Location	Nodes	Kube version	Status
mycluster	US South	1	1.10.7_1520	Normal

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.

The screenshot shows the IBM Cloud Catalog interface. At the top, there are three tabs: "Catalog - IBM Cloud", "IBM Watson Assistant", and "Node-RED : lab-iot-jppp.myblu". The URL in the address bar is <https://console.bluemix.net/catalog/?search=functions>. The main content area is titled "Catalog" and features a search bar with the text "functions". On the left, a sidebar lists categories: All Categories (4) (selected), Compute, Analytics, Integration, Internet of Things, Security and Identity, Starter Kits, Web and Mobile, and Web and Application (1). The "Compute" section is expanded, showing sub-categories: Compute (1), Containers, Networking, Storage, AI, Analytics (1), Databases (1), Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits, Web and Mobile, and Web and Application (1). The "Compute" section also contains a "Serverless Compute" heading. The "Functions" service card is highlighted with a red box. It has a green icon with a white 'f', the text "Functions", and "IBM". Below it, a description reads: "IBM Cloud Functions is a Function-as-a-Service (FaaS) platform which executes functions in response to incoming events." The "Analytics" section is partially visible below it, featuring the "Db2 Warehouse" service with its own icon and description.

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.
- Clique no botão “Start Creating”.

The screenshot shows the IBM Cloud Functions "Getting Started" page. The URL in the browser is https://console.bluemix.net/openwhisk/?bss_account=1dfcc0aea. The left sidebar has a "Functions" section with "Overview" selected, and other options like "Getting Started", "Pricing", "Concepts", etc. The main content area features a large "Getting Started with IBM Cloud Functions" heading, a brief description of the service, and two buttons: "Start Creating" (which is highlighted with a red box) and "Download CLI". Below these are sections for "What's New" and "Recent Changes". A large graphic on the right shows a green "f" symbol with a blue speech bubble and a grey speech bubble, with three yellow stars at the bottom.

Getting Started with
IBM Cloud Functions

IBM Cloud Functions (based on Apache OpenWhisk) is a Function-as-a-Service (FaaS) platform which executes functions in response to incoming events and costs nothing when not in use. [Learn More](#)

[Start Creating](#) [Download CLI](#)

★ What's New:

- Updated Action runtimes: NodeJS 8, Swift 4
- Get started quickly with Templates: [Try it now](#)
- New logging service integration (see the [Logs](#) link in the left navigation)

★ Recent Changes:

- Compliance: New ISO certifications ([ISO 27001](#), [ISO 27017](#), [ISO 27018](#))
- Available in new datacenter: Frankfurt, Germany (see the [Region selector](#))

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.
- Clique no botão “Start Creating”.
- Caso seu Space não seja encontrado. Mude a região.

The screenshot shows a web browser window with three tabs at the top: "IBM Cloud Functions - Create", "IBM Watson Assistant", and "Node-RED : lab-iot-jppp.myblu". The main content area is titled "IBM Cloud" and has a "Functions" section on the left with options like "Getting Started", "Actions", "Triggers", "Monitor", "Logs", and "APIs". A modal dialog box is centered over the page, displaying the message "No Cloud Foundry Space" and "Current Namespace:". It includes a note: "Before you can proceed with Cloud Functions, you must create a Cloud Foundry space. Do this now".

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “functions” e ache as “IBM Cloud Functions”.
- Clique no botão “Start Creating”.
- Caso seu Space não seja encontrado. Mude a região.

The screenshot shows a browser window with several tabs open: 'IBM Cloud Functions - Create', 'IBM Cloud Account', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The main content area is titled 'IBM Cloud' with tabs for 'Catalog', 'Docs', 'Support', and 'Manage'. A search bar says 'Search for resource...'. On the left, there's a sidebar with 'Functions' selected, and options like 'Getting Started', 'Actions', 'Triggers', 'Monitor', 'Logs', and 'APIs'. The main panel has sections for 'REGION', 'CLOUD FOUNDRY ORG', and 'CLOUD FOUNDRY SPACE'. Under 'REGION', 'Germany' is selected. A dropdown menu lists 'Germany', 'United Kingdom', 'US East', and 'US South'. The 'US South' option is highlighted with a red rectangle. At the bottom of the page, there's a footer with 'javascript:void(0)' and a back arrow.

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Clique em “Create Action”.

The screenshot shows the 'Create' page for IBM Cloud Functions. On the left, there's a sidebar with 'Functions' selected, followed by 'Getting Started', 'Actions', 'Triggers', 'Monitor', 'Logs', and 'APIs'. The main area has tabs for 'REGION' (US South), 'CLOUD FOUNDRY ORG' (JoaoPedroPP52ORG), and 'CLOUD FOUNDRY SPACE' (dev). The central part of the screen displays several options: 'Quickstart Templates' (with a brief description), 'Create Action' (which is highlighted with a red box), 'Create Sequence' (with a brief description), 'Create Trigger' (with a brief description), and 'Install Packages' (with a brief description). At the bottom, there's a 'Cancel' button.

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Clique em “Create Action”.

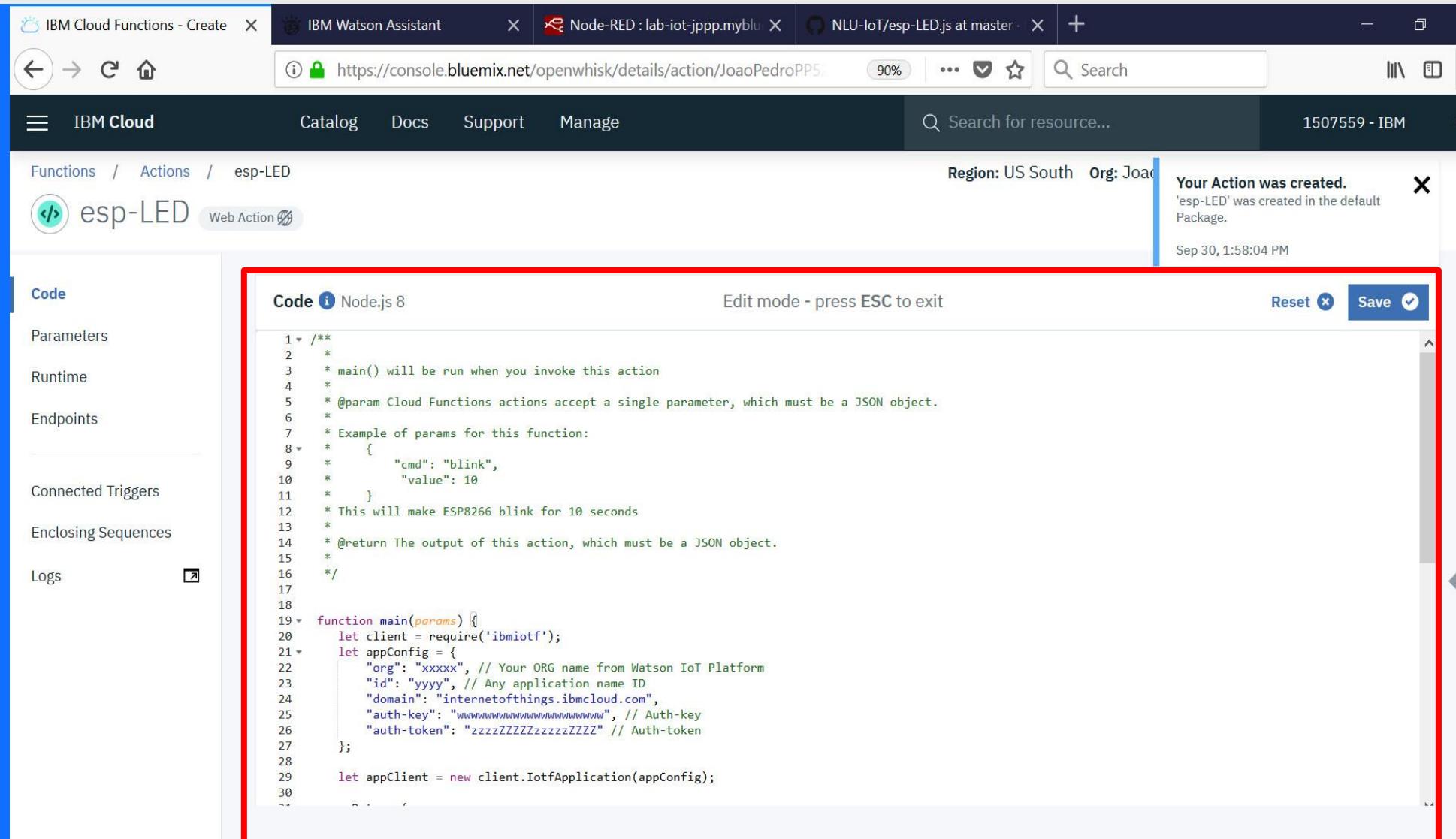
- Dê um nome para a sua primeira Function. Ela será usada para controlar o LED do seu dispositivo.

The screenshot shows the 'Create Action' page in the IBM Cloud Functions interface. The 'Action Name' field is highlighted with a red box and contains the value 'esp-LED'. The 'Enclosing Package' dropdown is set to '(Default Package)'. The 'Runtime' dropdown is set to 'Node.js 8'. A note at the bottom states: 'Looking for Java or Docker? [Java](#) and [Docker](#) Actions can be created with the [CLI](#)'. Navigation buttons at the bottom include 'Cancel', 'Previous', and 'Create'.

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Copie o código do git:
<https://github.com/Bluedev-BR/NLU-IoT/blob/master/Functions/ESP8266/esp-LED.js> e cole na caixa de código.



The screenshot shows the IBM Cloud Functions interface with a new action named 'esp-LED' being created. The code editor displays the following Node.js 8 script:

```
1  /**
2  * main() will be run when you invoke this action
3  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
4  *
5  * Example of params for this function:
6  * {
7  *   "cmd": "blink",
8  *   "value": 10
9  * }
10 * This will make ESP8266 blink for 10 seconds
11 *
12 * @return The output of this action, which must be a JSON object.
13 *
14 */
15
16
17
18
19 function main(params) {
20   let client = require('ibmiotf');
21   let appConfig = {
22     "org": "xxxxx", // Your ORG name from Watson IoT Platform
23     "id": "yyyy", // Any application name ID
24     "domain": "internetofthings.ibmcloud.com",
25     "auth-key": "xxxxxxxxxxxxxxxxxxxx", // Auth-key
26     "auth-token": "zzzzZZZZzzzzZZZ" // Auth-token
27   };
28
29   let appClient = new client.IotfApplication(appConfig);
30 }
```

A red box highlights the code editor area. In the top right corner of the interface, a message says 'Your Action was created. 'esp-LED' was created in the default Package.' with a timestamp of 'Sep 30, 1:58:04 PM'.

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Adicione seu do
seu txt as
propriedades
Organization ID
(ORG) e as
Application
Credentials: API
Key (auth-key),
Authentication
Token (auth-token)

The screenshot shows the IBM Cloud Functions interface with a browser tab for 'IBM Cloud Functions - Create' and a sidebar with tabs like 'Catalog', 'Docs', 'Support', and 'Manage'. The main area shows a 'Functions / Actions / esp-LED' view for a 'Web Action'. The 'Code' tab is selected, displaying Node.js code:

```
1  /**
2  * main() will be run when you invoke this action
3  *
4  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
5  *
6  * Example of params for this function:
7  * {
8  *   "cmd": "blink",
9  *   "value": 10
10 * }
11 *
12 * This will make ESP8266 blink for 10 seconds
13 *
14 * @return The output of this action, which must be a JSON object.
15 *
16 */
17
18
19 Function main(params) {
20   let client = require('ibmiotf');
21   let appConfig = {
22     "org": "xxxxx", // Your ORG name from Watson IoT Platform
23     "id": "yyyy", // Any application name ID
24     "domain": "internetofthings.ibmcloud.com",
25     "auth-key": "xxxxxxxxxxxxxxxxxxxx", // Auth-key
26     "auth-token": "zzzzZZZZzzzzzzzz" // Auth-token
27   };
28
29   let appClient = new client.IotfApplication(appConfig);
30 }
```

A red box highlights the configuration object in the code:

```
let appConfig = {  
  "org": "xxxxx", // Your ORG name from Watson IoT Platform  
  "id": "yyyy", // Any application name ID  
  "domain": "internetofthings.ibmcloud.com",  
  "auth-key": "xxxxxxxxxxxxxxxxxxxx", // Auth-key  
  "auth-token": "zzzzZZZZzzzzzzzz" // Auth-token  
};
```

The status bar on the right indicates 'Your Action was created.' and 'esp-LED' was created in the default Package. The timestamp is Sep 30, 1:58:04 PM. There are 'Reset' and 'Save' buttons at the bottom right.

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Mude essa linha trocando “Your Device Type” e “Your Device ID” para as propriedades: Device Type e Device ID do seu txt.

The screenshot shows the IBM Cloud Functions Actions interface. The title bar includes tabs for "IBM Cloud Functions - Actions" and "Node-RED : lab-iot-jppp.mybluemix.net". The URL in the address bar is <https://console.bluemix.net/openwhisk/details/action/JoaPedroPP520RG>. The main content area shows a "Functions / Actions / esp-LED" page for a "Web Action" named "esp-LED". On the left, there's a sidebar with sections for "Code", "Parameters", "Runtime", "Endpoints", "Connected Triggers", "Enclosing Sequences", and "Logs". The "Code" section contains Node.js 8 code. A red box highlights the line of code: `appClient.publishDeviceCommand("Your device type", "Your device name", params.cmd, "json", myData);`. The code block also includes other parts of the function like connecting to IoT and sending JSON data. The top right has "Reset" and "Save" buttons. The bottom right has a vertical scrollbar.

```
Code i Node.js 8
Edit mode - press ESC to exit
Reset Save

29 let appClient = new client.IotfApplication(appConfig);
30
31 myData = {
32   "d": {
33     "fields": [
34       {
35         "field": params.cmd,
36         "value": params.value
37       }
38     ]
39   }
40 }
41
42 function (params) {
43   appClient.publishDeviceCommand("Your device type", "Your device name", params.cmd, "json", myData);
44   callback();
45 }
46
47 let promise = new Promise((resolve, reject) => {
48   appClient.connect();
49   appClient.on("connect", function () {
50
51     myData = JSON.stringify(myData);
52
53     send(function () {
54       appClient.disconnect();
55       resolve({"response": 200});
56     });
57   });
58 })
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.1. Criando as funções.

- Mude essa linha trocando “Your Device Type” e “Your Device ID” para as propriedades: Device Type e Device ID do seu txt.

- Salve o código.

The screenshot shows the IBM Cloud Functions Actions interface. The title bar includes tabs for "IBM Cloud Functions - Actions" and "Node-RED : lab-iot-jppp.mybluemix.net". The URL in the address bar is <https://console.bluemix.net/openwhisk/details/action/JoaPedroPP520RG>. The main navigation bar has links for "Catalog", "Docs", "Support", and "Manage", along with a search bar and user information "1507559 - IBM". Below the navigation, it says "Region: US South Org: JoaoPedroPP520RG Space: dev". The left sidebar lists "Functions / Actions / esp-LED" and "esp-LED Web Action". The right panel shows the "Code" tab for "Node.js 8" with the following code:

```
29 let appClient = new client.IotfApplication(appConfig);
30
31 myData = {
32   "d": {
33     "fields": [
34       {
35         "field": params.cmd,
36         "value": params.value
37       }
38     ]
39   }
40 }
41
42 function send(callback) {
43   appClient.publishDeviceCommand("Your device type", "Your device name", params.cmd, "json", myData);
44   callback();
45 }
46
47 let promise = new Promise((resolve, reject) => {
48   appClient.connect();
49   appClient.on("connect", function () {
50
51     myData = JSON.stringify(myData);
52
53     send(function () {
54       appClient.disconnect();
55       resolve({"response": 200});
56     });
57   });
58});
```

The "Save" button in the top right corner is highlighted with a red box.

4. Enviando comandos ao seu Device usando Cloud Functions

4.2. Testando a primeira função.

- Clique no botão “Change Input”.

The screenshot shows a code editor with a file named 'node.js 8'. The code is a Node.js script for an IBM IoT Function. It includes comments explaining the function's behavior and its parameters. A red box highlights the 'Change Input' button in the toolbar, which is the target of the user's click action. The code itself contains several sensitive configuration details, such as organization names ('vrreja'), application IDs ('app'), and cloud domains ('internetofthings.ibmcloud.com'), which are also highlighted with a red box.

```
ode.js 8

in() will be run when you invoke this action
aram Cloud Functions actions accept a single parameter, which must be a JSON object.
return The output of this action, which must be a JSON object.

on main(params) {
  t client = require('ibmiotf');
  t appConfig = {
    "org": "vrreja",
    "id": "app",
    "domain": "internetofthings.ibmcloud.com",
    "auth-key": "REDACTED"
    "auth-token": "REDACTED"

  t appClient = new client.IotfApplication(appConfig);

  Data = {
    "d": {
      "fields": [
        {
          "field": params.cmd,
          "value": params.value
        ]
      }
    }
  }
}

module.exports = main;
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.2. Testando a primeira função.

- Clique no botão “Change Input”.
- Escreva na caixa um JSON como o da imagem ao lado. “cmd” indica o comando desejado, nessa caso vamos fazer o LED piscar. “value” indica por quantos segundos ele permanecerá piscando.

Code Node.js 8

```
1 /**
2  * 
3  * main() will be r
4  * 
5  * @param Cloud Fun
6  * 
7  * @return The outp
8  * 
9 */
10
11
12 function main(params
13   let client = req
14   let appConfig =
15     "org": "vrre
16     "id": "app",
17     "domain": "i
18     "auth-key":
19     "auth-token"
20   };
21
22   let appClient =
23
24   myData = {
25     "d": {
26       "fields": [
27         {
28           "field": params.cmd,
29           "value": params.value
30         }
31       ]
32     }
33   };
34
35   client.send(myData);
36
37   return Promise.all([
38     sleep(5000),
39     client.send({ "d": { "fields": [ { "field": "led", "value": "off" } ] } })
40   ])
41 }
```

Change Action Input

Edit mode - press ESC to exit

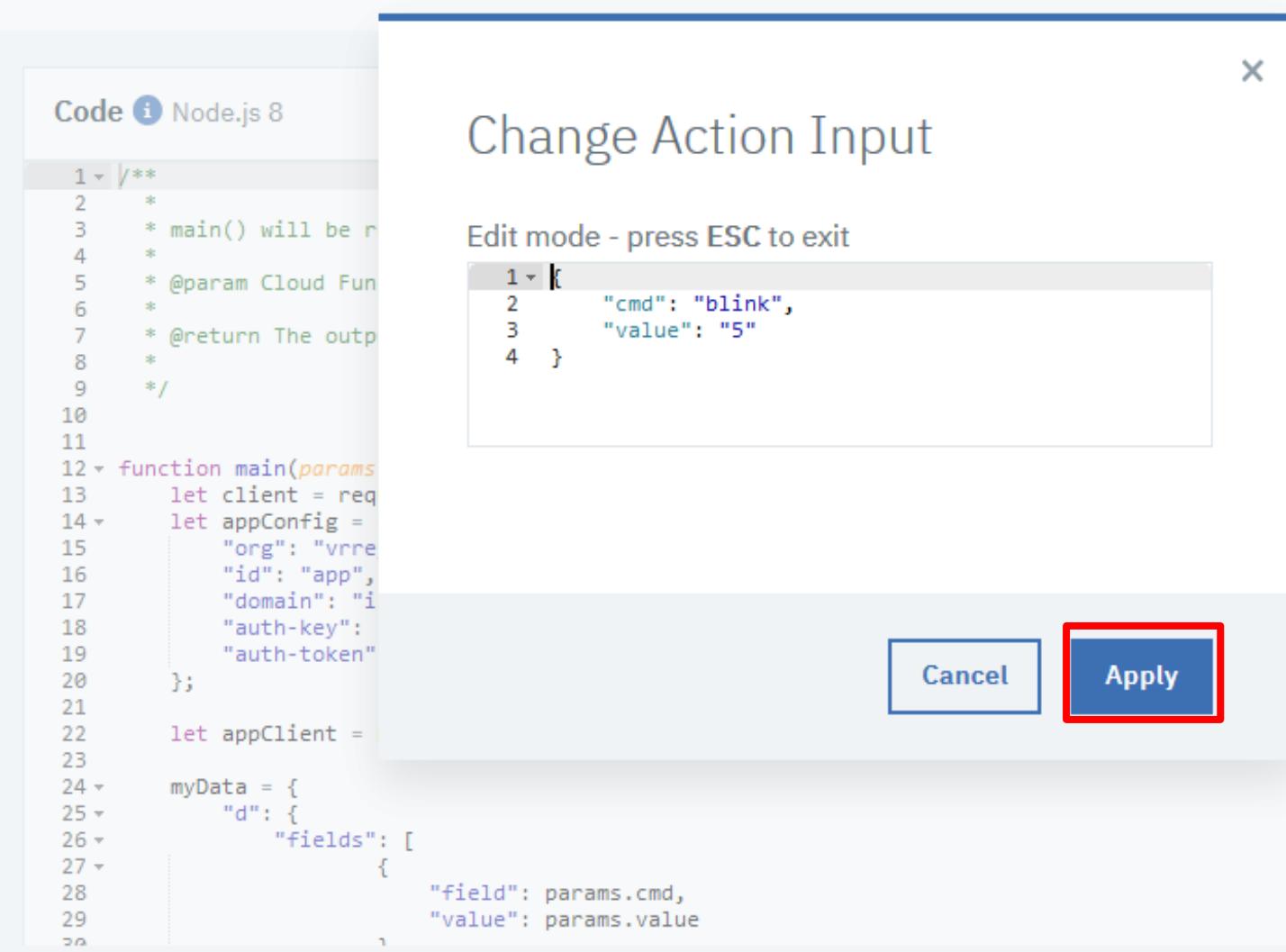
```
{ "cmd": "blink", "value": "5"}
```

Cancel Apply

4. Enviando comandos ao seu Device usando Cloud Functions

4.2. Testando a primeira função.

- Clique no botão “Apply”.



4. Enviando comandos ao seu Device usando Cloud Functions

4.2. Testando a primeira função.

- Clique no botão “Apply”.
- Clique no botão “Invoke para executar a sua Action com a entrada definida.



The screenshot shows the IBM Cloud Functions interface. On the right side, there are two buttons: "Change Input" with a pencil icon and "Invoke" with a circular arrow icon. The "Invoke" button is highlighted with a red rectangular box. The main area contains a code editor with the following JavaScript code:

```
const client = require('ibm-watson-iot-client');
const appConfig = require('./appConfig');

let myData = {
    "cmd": "test",
    "value": "on"
};

let params = {
    "cmd": "test",
    "value": "on"
};

let ent = new client.IotfApplication(appConfig);

let fields = [
    {
        "field": params.cmd,
        "value": params.value
    }
];

end(callback) {
    pClient.publishDeviceCommand("ESP8266", "teste", params.cmd, "json", myData);
    pClient.publishDeviceCommand("Simulated", "teste_sim", params.cmd, "json", myData);
    callback();
}

let e = new Promise((resolve, reject) => {
    ent.connect();
    ent.on("connect", function () {
        Data = JSON.stringify(myData);
        end(function () {
            appClient.disconnect();
        });
    });
});
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.2. Testando a primeira função.

- Clique no botão “Apply”.
- Clique no botão “Invoke para executar a sua Action com a entrada definida.
- Se a resposta for 200 (HTTP – OK), o LED do seu Device simulado deve piscar.

The screenshot shows the IBM Cloud Functions interface. On the left, there is a code editor with the following JavaScript code:

```
client = new client.IotfApplication(appConfig);
// ...
fields": [
  {
    "field": params.cmd,
    "value": params.value
  }
]

end(callback) {
  appClient.publishDeviceCommand("ESP8266", "teste", params.cmd, "json");
  appClient.publishDeviceCommand("Simulated", "teste_sim", params.cmd, "json");
  callback();
}

myData = new Promise((resolve, reject) => {
  client.connect();
  client.on("connect", function () {
    myData = JSON.stringify(myData);
  });
  end(function () {
    appClient.disconnect();
  });
}
```

At the top right of the code editor are two buttons: "Change Input" and "Invoke".

To the right of the code editor is a panel titled "Activations" which lists two entries:

- triggerBlink** (green checkmark) - Activation ID: 6eb979e3bf384cb9b979e3bf382cb92e - Results: { "response": 200 } (This result is highlighted with a red box.)
- triggerBlink** (red X) - Activation ID: 5d8c6309f3004ea18c6309f300aea126 - Results: (empty)

Below the activation log is a "Logs" section which is currently empty: []

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

The screenshot shows the IBM Cloud Functions interface. The top navigation bar includes tabs for 'Create', 'Watson Assistant', 'Node-RED', and 'Cloud Functions'. The main title is 'IBM Cloud Functions - Create'. Below the title, there's a breadcrumb navigation: 'Functions / Actions / esp-LED'. A red box highlights the 'Actions' tab. On the left, a sidebar lists 'Code', 'Parameters', 'Runtime', 'Endpoints', 'Connected Triggers', 'Enclosing Sequences', and 'Logs'. The 'Code' section displays a Node.js script for the 'esp-LED' action. The script starts with a multi-line comment explaining the function's purpose and parameters. It defines a 'main' function that requires the 'ibmiotf' module, sets up an 'appConfig' object with Watson IoT Platform details, and creates an 'appClient' instance. A success message box is visible on the right, stating 'Your Action was created. 'esp-LED' was created in the default Package.' with a timestamp of 'Sep 30, 1:58:04 PM'. There are 'Change Input' and 'Invoke' buttons at the bottom right of the code editor.

```
Code i Node.js 8
1  /**
2   * main() will be run when you invoke this action
3   *
4   * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
5   *
6   * Example of params for this function:
7   * {
8   *     "cmd": "blink",
9   *     "value": 10
10  * }
11  *
12  * This will make ESP8266 blink for 10 seconds
13  *
14  * @return The output of this action, which must be a JSON object.
15  *
16  */
17
18
19  function main(params) {
20      let client = require('ibmiotf');
21      let appConfig = {
22          "org": "xxxxx", // Your ORG name from Watson IoT Platform
23          "id": "yyyy", // Any application name ID
24          "domain": "internetofthings.ibmcloud.com",
25          "auth-key": "xxxxxxxxxxxxxxxxxxxx", // Auth-key
26          "auth-token": "zzzzZZZZzzzzzzzz" // Auth-token
27      };
28
29      let appClient = new client.IotfApplication(appConfig);
30  }
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

- Clique no botão “Create” para criar outra Cloud Function.

The screenshot shows the IBM Cloud Functions Actions page. The URL in the browser is https://console.bluemix.net/openwhisk/actions. The page displays two Cloud Functions:

NAME	RUNTIME	WEB ACTION	MEMORY	TIMEOUT
esp-LED	Node.js 8	Not Enabled	256 MB	60 s
omega	Node.js 8	Not Enabled	256 MB	60 s

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

- Clique no botão “Create” para criar outra Cloud Function.

- Clique em “Create Action”

The screenshot shows the 'Create' page for IBM Cloud Functions. The URL in the browser is <https://console.bluemix.net/openwhisk/create>. The page has a sidebar on the left with 'Functions' selected, and a main area with the following options:

- Quickstart Templates**: Get started quickly using one of the Templates. A number of use cases are available, from a hello world action to invoking functions from Cloudant or Message Hub events.
- Create Action**: Actions contain your function code and are invoked by events or REST API calls. This option is highlighted with a red box.
- Create Sequence**: Sequences invoke Actions in a linear order, passing parameters from one to the next.
- Create Trigger**: Triggers receive events from outside IBM Cloud Functions and invoke all connected Actions.
- Install Packages**: Installing Packages installs reusable Actions into your namespace.

At the bottom of the page is a 'Cancel' button.

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Clique em “Actions” para retornar ao menu de Cloud Functions.

- Clique no botão “Create” para criar outra Cloud Function.

- Clique em “Create Action”

- Dê um nome para sua outra Function.

The screenshot shows the 'Create Action' page in the IBM Cloud Functions interface. The URL in the browser is https://console.bluemix.net/openwhisk/create/action. The left sidebar is titled 'Functions' and includes links for Getting Started, Actions (which is selected), Triggers, Monitor, Logs, and APIs. The main form has the following fields:

- Action Name:** A text input field containing "esp-sensor" is highlighted with a red border.
- Enclosing Package:** A dropdown menu set to "(Default Package)" with a "Create Package" button next to it.
- Runtime:** A dropdown menu set to "Node.js 8".

Below the form, a note states: "Looking for Java or Docker? [Java](#) and [Docker](#) Actions can be created with the [CLI](#)". At the bottom of the page are "Cancel", "Previous", and "Create" buttons.

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Clique no botão “Create”.

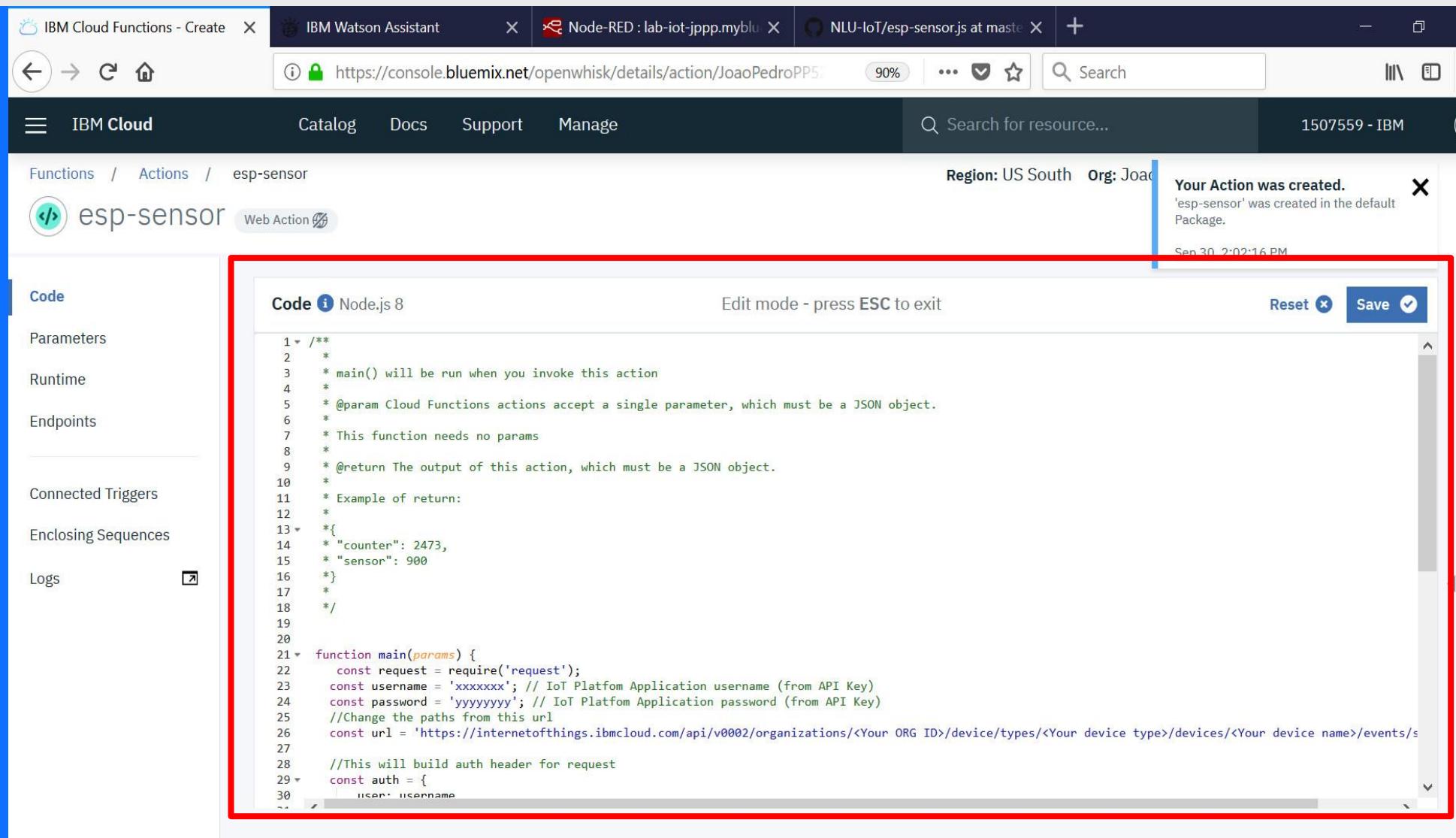
The screenshot shows the 'Create Action' interface in the IBM Cloud Functions service. The left sidebar lists 'Getting Started', 'Actions', 'Triggers', 'Monitor', 'Logs', and 'APIs'. The main area has tabs for 'REGION' (US South), 'CLOUD FOUNDRY ORG' (JoaoPedroPP52ORG), and 'CLOUD FOUNDRY SPACE' (dev). The title 'Create Action' is centered above the form fields. The 'Action Name' field contains 'esp-sensor'. The 'Enclosing Package' dropdown is set to '(Default Package)' with a 'Create Package' button next to it. The 'Runtime' dropdown is set to 'Node.js 8'. A note at the bottom says 'Looking for Java or Docker? [Java](#) and [Docker](#) Actions can be created with the [CLI](#)'. At the bottom are 'Cancel', 'Previous', and 'Create' buttons, with 'Create' being highlighted with a red box.

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Clique no botão “Create”.

- Copie o código do git:
<https://github.com/Bluedev-BR/NLU-IoT/blob/master/Functions/ESP8266/esp-sensor.js> e cole na caixa de código.



The screenshot shows the IBM Cloud Functions - Create interface. The title bar has tabs for 'IBM Cloud Functions - Create', 'IBM Watson Assistant', 'Node-RED : lab-iot-jppp.myblu...', and 'NLU-IoT/esp-sensor.js at master'. The main area shows the 'Actions' section with 'esp-sensor' selected. On the left, there's a sidebar with tabs for 'Code', 'Parameters', 'Runtime', 'Endpoints', 'Connected Triggers', 'Enclosing Sequences', and 'Logs'. The 'Code' tab is active, displaying a Node.js script. The code starts with a multi-line comment explaining the function's purpose and parameters, followed by a main function 'main' that constructs a POST request to an IoT Platform endpoint. A red box highlights the code editor area. In the top right, a success message says 'Your Action was created. 'esp-sensor' was created in the default Package.' Below it, a timestamp shows 'Sep 30, 2:02:16 PM'. There are 'Reset' and 'Save' buttons at the bottom right of the code editor.

```
1 *//**
2 *
3 * main() will be run when you invoke this action
4 *
5 * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
6 *
7 * This function needs no params
8 *
9 * @return The output of this action, which must be a JSON object.
10 *
11 * Example of return:
12 *
13 *{
14 *  "counter": 2473,
15 *  "sensor": 900
16 *}
17 *
18 */
19
20
21 function main(params) {
22   const request = require('request');
23   const username = 'xxxxxxxx'; // IoT Platform Application username (from API Key)
24   const password = 'yyyyyyyy'; // IoT Platform Application password (from API Key)
25   //Change the paths from this url
26   const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/sensor';
27
28   //This will build auth header for request
29   const auth = {
30     user: username
31   }
32
33   const options = {
34     url: url,
35     method: 'POST',
36     auth: auth
37   }
38
39   request(options, function(error, response, body) {
40     if (!error && response.statusCode === 200) {
41       console.log('Success');
42     } else {
43       console.error('Error:', error);
44     }
45   });
46 }
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Adicione seu do
seu txt as
Application
Credentials: API
Key (username),
Authentication
Token (password).

The screenshot shows the IBM Cloud Functions Actions interface. The URL in the browser is https://console.bluemix.net/openwhisk/details/action/JoaopPedroPP520RG. The page title is "esp-sensor". The left sidebar has tabs for Code, Parameters, Runtime, Endpoints, Connected Triggers, Enclosing Sequences, and Logs. The main area is titled "Code Node.js 8" and contains the following Node.js code:

```
    * This function needs no permissions.
8    *
9    * @return The output of this action, which must be a JSON object.
10   *
11  * Example of return:
12  *
13  * {
14  *   "counter": 2473,
15  *   "sensor": 900
16  * }
17  *
18 */
19
20
21  function main(params) {
22    const request = require('request');
23    const username = "REDACTED" // IoT Platform Application username (from API Key)
24    const password = "REDACTED" IoT Platform Application password (from API Key)
25    //Change the paths from this url
26    const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/<Your event type>?lastX=10&sortOrder=desc'
27
28    //This will build auth header for request
29    const auth = {
30      user: username,
31      pass: password
32    };
33    let qs = null
34    let options = {method: 'GET', url, qs, auth};
35    let promise = new Promise((resolve, reject) => {
36      request(options, (error, response, body) => {
37        if (error) {
38          reject(error);
39        } else {
40          resolve(body);
41        }
42      });
43    }
44    return promise;
45  }
46
47  module.exports = main;
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Adicione seu do seu txt as Application Credentials: API Key (username), Authentication Token (password).
- Na variável “url”, troque Organization ID (ORG), Device Type e Device ID, com base no seu txt.

The screenshot shows the IBM Cloud Functions Actions interface. The URL in the browser is https://console.bluemix.net/openwhisk/details/action/JoaopPedroPP520RG. The page title is "esp-sensor". The left sidebar has tabs for Code, Parameters, Runtime, Endpoints, Connected Triggers, Enclosing Sequences, and Logs. The main area is titled "Code Node.js 8" and contains the following Node.js code:

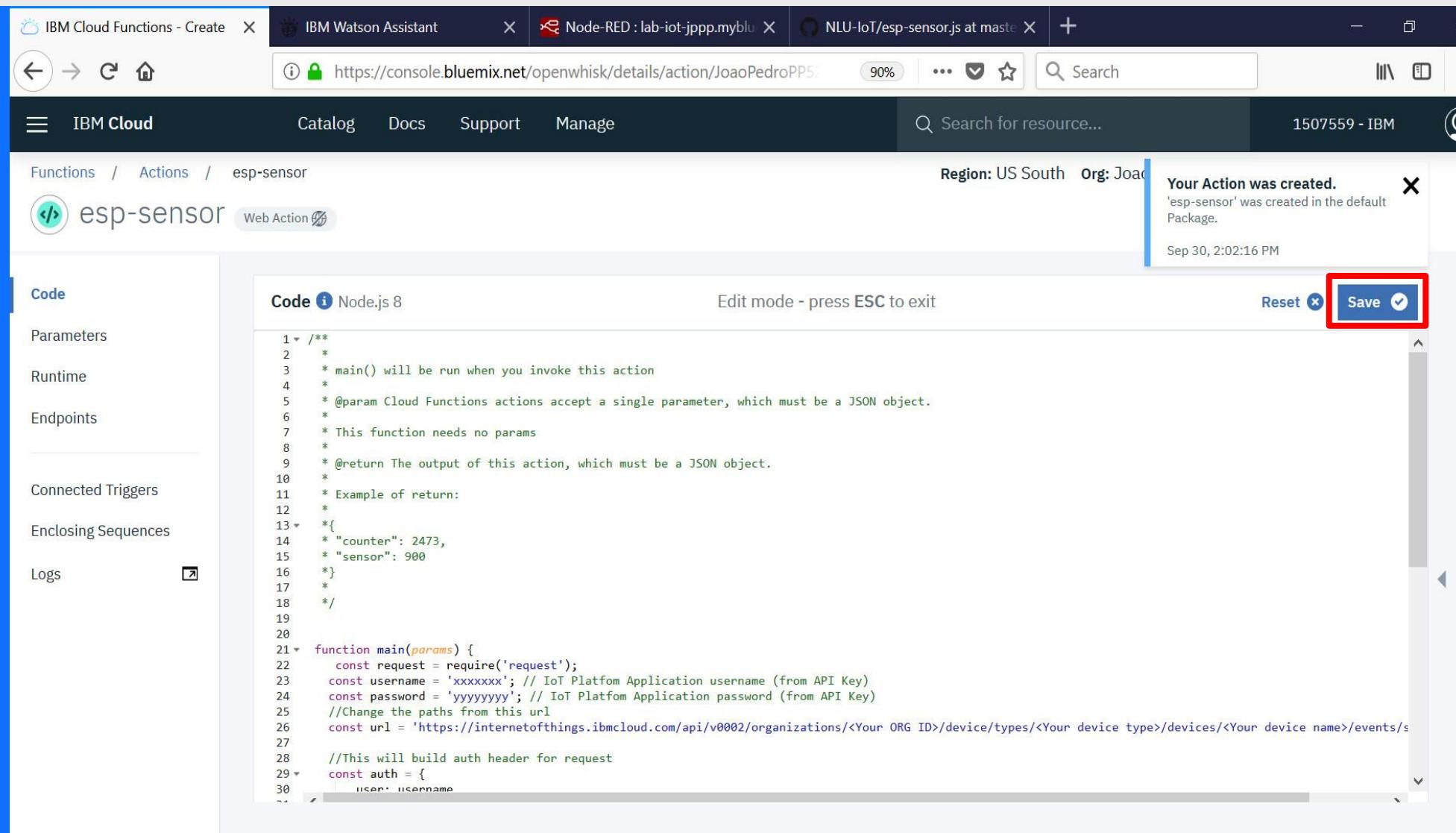
```
8 * This function needs no permissions.
9 *
10 * @return The output of this action, which must be a JSON object.
11 * Example of return:
12 *
13 * {
14 *   "counter": 2473,
15 *   "sensor": 900
16 * }
17 *
18 */
19
20
21 function main(params) {
22   const request = require('request');
23   const username = '1-enf&ys-fjoiehgfhvwe'; // IoT Platform Application username (from API Key)
24   const password = 'wefoil#8h3bJOU23mn'; // IoT Platform Application password (from API Key)
25   //Change the paths from this url
26   const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/<Your event type>';
```

The URL line in the code is highlighted with a red box.

4. Enviando comandos ao seu Device usando Cloud Functions

4.3. Criando a segunda função.

- Salve o código.



The screenshot shows the IBM Cloud Functions interface with a browser tab for 'IBM Cloud Functions - Create'. The main area displays the 'esp-sensor' action details. On the left, a sidebar lists 'Parameters', 'Runtime', 'Endpoints', 'Connected Triggers', 'Enclosing Sequences', and 'Logs'. The 'Code' section is selected, showing a Node.js 8 script. The script starts with a multi-line comment explaining the function's purpose and parameters. It then defines a 'main' function that requires the 'request' module. The function variables are set to IoT Platform Application credentials ('username' and 'password'). The URL for sending data is constructed using these variables. The code ends with a multi-line comment about building an auth header. A success message box is visible on the right, stating 'Your Action was created.' and providing a timestamp. The 'Save' button at the bottom right of the code editor is highlighted with a red box.

```
1 *//**
2 *
3 * main() will be run when you invoke this action
4 *
5 * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
6 *
7 * This function needs no params
8 *
9 * @return The output of this action, which must be a JSON object.
10 *
11 * Example of return:
12 *
13 *{
14 *  "counter": 2473,
15 *  "sensor": 900
16 *}
17 *
18 */
19
20
21 function main(params) {
22   const request = require('request');
23   const username = 'xxxxxxxx'; // IoT Platform Application username (from API Key)
24   const password = 'yyyyyyyy'; // IoT Platform Application password (from API Key)
25   //Change the paths from this url
26   const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/sensor';
27
28   //This will build auth header for request
29   const auth = {
30     user: username
31   }
32
33   //Send data to device
34   request({
35     url: url,
36     method: 'POST',
37     auth: auth,
38     json: true
39   }, function(error, response, body) {
40     if (!error && response.statusCode === 200) {
41       console.log('Data sent successfully');
42     } else {
43       console.error('Error sending data: ', error);
44     }
45   });
46 }
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.4. Obtendo as credenciais para uso das Functions.

- Clique em “Endpoints”.

The screenshot shows the IBM Cloud Functions interface. The browser tabs at the top include "IBM Cloud Functions - Create", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.myblu", and "NLU-IoT/esp-sensor.js at master". The main page title is "IBM Cloud" with sub-navigations for "Catalog", "Docs", "Support", and "Manage". A search bar says "Search for resource...". On the right, it shows "Region: US South" and "Org: JoaoPedroPPS". A message box says "Your Action was created. 'esp-sensor' was created in the default Package." with a timestamp "Sep 30, 2:02:16 PM". The main content area shows an "esp-sensor" action as a "Web Action". On the left, a sidebar has tabs: "Code" (selected), "Parameters", "Runtime", "Endpoints" (which is highlighted with a red box), "Connected Triggers", "Enclosing Sequences", and "Logs". The "Code" tab displays Node.js code for the "main" function:

```
1  /**
2  *
3  * main() will be run when you invoke this action
4  *
5  * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
6  *
7  * This function needs no params
8  *
9  * @return The output of this action, which must be a JSON object.
10 *
11 * Example of return:
12 *
13 * {
14 *   "counter": 2473,
15 *   "sensor": 900
16 * }
17 *
18 */
19
20
21 function main(params) {
22   const request = require('request');
23   const username = 'xxxxxxxx'; // IoT Platform Application username (from API Key)
24   const password = 'yyyyyyyy'; // IoT Platform Application password (from API Key)
25   //Change the paths from this url
26   const url = 'https://internetofthings.ibmcloud.com/api/v0002/organizations/<Your ORG ID>/device/types/<Your device type>/devices/<Your device name>/events/sensor';
27
28   //This will build auth header for request
29   const auth = {
30     user: username
31   }
32
33   const options = {
34     url: url,
35     method: 'POST',
36     headers: {
37       'Content-Type': 'application/json'
38     },
39     auth: auth
40   }
41
42   request(options, (error, response, body) => {
43     if (error) {
44       console.error(error);
45       return res.status(500).send(`Error: ${error}`);
46     }
47     if (response.statusCode >= 400) {
48       return res.status(response.statusCode).send(`Error: ${body}`);
49     }
50     res.status(200).send(`Success: ${body}`);
51   });
52 }
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.4. Obtendo as credenciais para uso das Functions.

- Clique em “Endpoints”.

- Na seção CURL, revele a autorização utilizada para uso da sua Function clicando no ícone semelhante a um olho.

IBM Cloud Functions - Create X IBM Watson Assistant X Node-RED : lab-iot-jppp.myblu X NLU-IoT/esp-sensor.js at master X +

IBM Cloud Catalog Docs Support Manage Search for resource... 1507559 - IBM

Functions / Actions / esp-sensor Region: US South Org: JoaoPedroPP52ORG Space: dev

esp-sensor Web Action

Code Parameters Runtime Endpoints Connected Triggers Enclosing Sequences Logs

Web Action

Enable as Web Action Allow your Cloud Functions actions to handle HTTP events. Learn more about [Web Actions](#).

Raw HTTP handling When enabled your Action receives requests in plain text instead of a JSON body

REST API

HTTP METHOD	AUTH	URL
POST	API-KEY	https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaoPedroPP52ORG_dev/actions/esp-sensor

CURL

```
curl -u API-KEY -X POST https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaoPedroPP52ORG_dev/actions/esp-sensor?blocking=true
```

4. Enviando comandos ao seu Device usando Cloud Functions

4.4. Obtendo as credenciais para uso das Functions.

- Copie para o Note Pad toda a chamada CURL.
- As credenciais estão divididas em duas partes. A primeira (vermelho) é o “username” e depois dos dois pontos temos o “password”.

The screenshot shows the IBM Cloud Functions - Action interface. The URL in the browser is https://console.bluemix.net/openwhisk/details/action/JoaPedroPP52ORG_dev/esp-sensor. The page displays the configuration for the 'esp-sensor' action, specifically the 'Web Action' and 'REST API' sections. The 'Endpoints' tab is selected in the sidebar. In the 'Web Action' section, there are two checkboxes: 'Enable as Web Action' (unchecked) and 'Raw HTTP handling' (unchecked). In the 'REST API' section, the 'HTTP METHOD' is set to 'POST', 'AUTH' is set to 'API-KEY', and the 'URL' is https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaPedroPP52ORG_dev/actions/esp-sensor. Below this, a 'CURL' section shows the command: curl -u [REDACTED]:[REDACTED] POST https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaPedroPP52ORG_dev/actions/esp-sensor?blocking=true. The entire URL part after the colon in the password is highlighted in yellow.

4. Enviando comandos ao seu Device usando Cloud Functions

4.4. Obtendo as credenciais para uso das Functions.

- Veja que o seu Namespace também aparece na chamada. Vamos usar as credenciais e esse Namespace no nosso próximo serviço. Antes do “underline” está seu namespace_org, depois está o namespace_space

The screenshot shows the IBM Cloud Functions - Action details page for an action named "esp-sensor". The left sidebar lists "Endpoints" as the selected tab, along with other options like "Code", "Parameters", "Runtime", "Connected Triggers", "Enclosing Sequences", and "Logs". The main content area is divided into sections: "Web Action" and "REST API". Under "Web Action", there are two checkboxes: "Enable as Web Action" (unchecked) and "Raw HTTP handling" (unchecked). Under "REST API", the "HTTP METHOD" is set to "POST", "AUTH" is set to "API-KEY", and the "URL" is "https://openwhisk.ng.bluemix.net/api/v1/namespaces/JoaoPedroPP52ORG_dev/actions/esp-sensor". Below this, a "CURL" section provides a command-line example:

```
curl -u 4582fe5a-b444-4527-8e35-4247ee96a477:wRnjca5HAvq1XBPCcv40PwvaiF1ch7ttdYm9t7B3nFYM71APBHQX5G6zVxm2pq4K -X POST https://openwhisk.ng.bluemix.net/api/v1/namespaces JoaoPedroPP52ORG dev/actions/esp-sensor blocking=true
```

A red box highlights the "JoaoPedroPP52ORG dev/actions/esp-sensor" part of the curl command, specifically the namespace and action name.

Checkpoint 3

Depois desses passos devem estar preenchidas as seguintes propriedades do seu modelo.

3. IBM Cloud Functions

- Auth Username:
- Auth Password:
- Namespace_org:
- Namespace_space
- Action 1:
- Action 2:

Essas credenciais são necessárias para chamar a sua Cloud Function por uma chamada HTTP ou por um serviço externo.

5. Criando o Assistente Virtual com o Watson Assistant

5.1. Criando uma instância do Watson Assistant.

- Volte ao Dashboard e crie um novo recurso.

The screenshot shows the IBM Cloud dashboard interface. At the top, there are tabs for 'Dashboard - IBM Cloud' and 'Node-RED : lab-iot-jppp.mybluemix.net'. The main content area is titled 'Dashboard' and contains three sections: 'Cloud Foundry Applications', 'Clusters', and 'Cloud Foundry Services'. In the top right corner of the dashboard area, there is a blue button labeled 'Create resource' which is highlighted with a red rectangle. The 'Cloud Foundry Applications' section shows one application named 'Lab-IoT-JPPP' with details: Region US South, CF Org JoaoPedroPP52..., CF Space dev, Memory (MB) 256, and Status Running (1/1). The 'Clusters' section shows one cluster named 'mycluster' with details: Location US South, Nodes 1, Kube version 1.10.7_1520, and Status Normal. The 'Cloud Foundry Services' section shows one service named 'IoT-Lab' with details: Region US South, CF Org JoaoPedroPP52..., Plan iotf-service-free, Service Offering iotf-service. The bottom right corner of the dashboard has a small 'FEEDBACK' button.

5. Criando o Assistente Virtual com o Watson Assistant

5.1. Criando uma instância do Watson Assistant.

- Volte ao Dashboard e crie um novo recurso.

- Na aba de busca digite “watson assistant” e ache o serviço “Watson Assistant”.

The screenshot shows the IBM Cloud Catalog interface. At the top, there are two tabs: 'Catalog - IBM Cloud' and 'Node-RED : lab-iot-jppp.mybluemix.net'. The URL in the address bar is <https://console.bluemix.net/catalog/?search=watson%20assistant>. The page title is 'Catalog'. Below the title, there are navigation links for 'IBM Cloud', 'Catalog', 'Docs', 'Support', and 'Manage', along with a search bar that says 'Search for resource...'. On the right side, there is a user profile icon and the text '1507559 - IBM'. A sidebar on the left lists categories: All Categories (2), Compute, Containers, Networking, Storage, AI (1), Analytics, Databases, Developer Tools, Integration, Internet of Things, Security and Identity, Starter Kits (1), Web and Mobile, and Web and Application. The 'AI' category is selected. In the main content area, under 'AI', the 'Watson Assistant (formerly Conversation)' service is listed. It has a purple icon of two speech bubbles. The text reads: 'Watson Assistant (formerly Conversation) Lite • IBM Add a natural language interface to your application to automate interactions with your end users. Common applications include virtual agents'. This entire service listing is enclosed in a red rectangular box. Below this, there is a section titled 'Starter Kits' with a card for 'Watson Assistant Basic'.

5. Criando o Assistente Virtual com o Watson Assistant

5.1. Criando uma instância do Watson Assistant.

- Volte ao Dashboard e crie um novo recurso.
- Na aba de busca digite “watson assistant” e ache o serviço “Watson Assistant”.
- Clique no botão Launch Tool.

The screenshot shows the IBM Cloud interface for managing Watson Assistant services. The left sidebar is titled 'Manage' and lists 'Service credentials', 'Plan', and 'Connections'. The main content area is titled 'Watson Assistant-Lab' and shows the location as 'US South', organization as 'JoaoPedroPP520RG', and space as 'dev'. A central callout box contains the text 'Get started by launching the tool.' and features a blue 'Launch tool' button, which is highlighted with a red rectangular border. Below this, there are links for 'Getting started tutorial' and 'API reference'. The 'Plan' is listed as 'Lite' with an 'Upgrade' link. At the bottom, there's a 'Credentials' section with fields for 'Url' (set to <https://gateway.watsonplatform.net/assistant/api>), 'Username' (redacted), and 'Password' (redacted). A yellow 'Let's talk' button is located at the bottom right.

5. Criando o Assistente Virtual com o Watson Assistant

5.2. Fazendo o upload do Workspace da demonstração.

- Entre na aba “Workspaces”.

The screenshot shows a web browser window with three tabs open: "IBM Watson Service Page", "IBM Watson Assistant", and "Node-RED : lab-iot-jppp.mybluemix.net". The "IBM Watson Assistant" tab is active, displaying the "IBM Watson Assistant" landing page. The navigation bar at the top has two items: "Home" and "Workspaces", with "Workspaces" highlighted by a red box. Below the navigation, the page title is "Introducing IBM Watson Assistant". A sub-headline states: "Watson Conversation is evolving to simplify how you build and scale virtual assistants. See what's new". The main content is titled "Three easy steps" and lists three steps: 1. Create intents and entities, 2. Build your dialog, and 3. Test your dialog. Each step has a large blue number, a brief description, and a "Learn more" link. The background features a light blue gradient with white speech bubble icons.

IBM Watson Assistant

Introducing

IBM Watson Assistant

Watson Conversation is evolving to simplify how you build and scale virtual assistants. [See what's new](#)

Three easy steps

Follow these steps to create a virtual assistant.

1 Create intents and entities
Determine what your virtual assistant will understand by providing training examples so Watson can learn.
[Learn more](#)

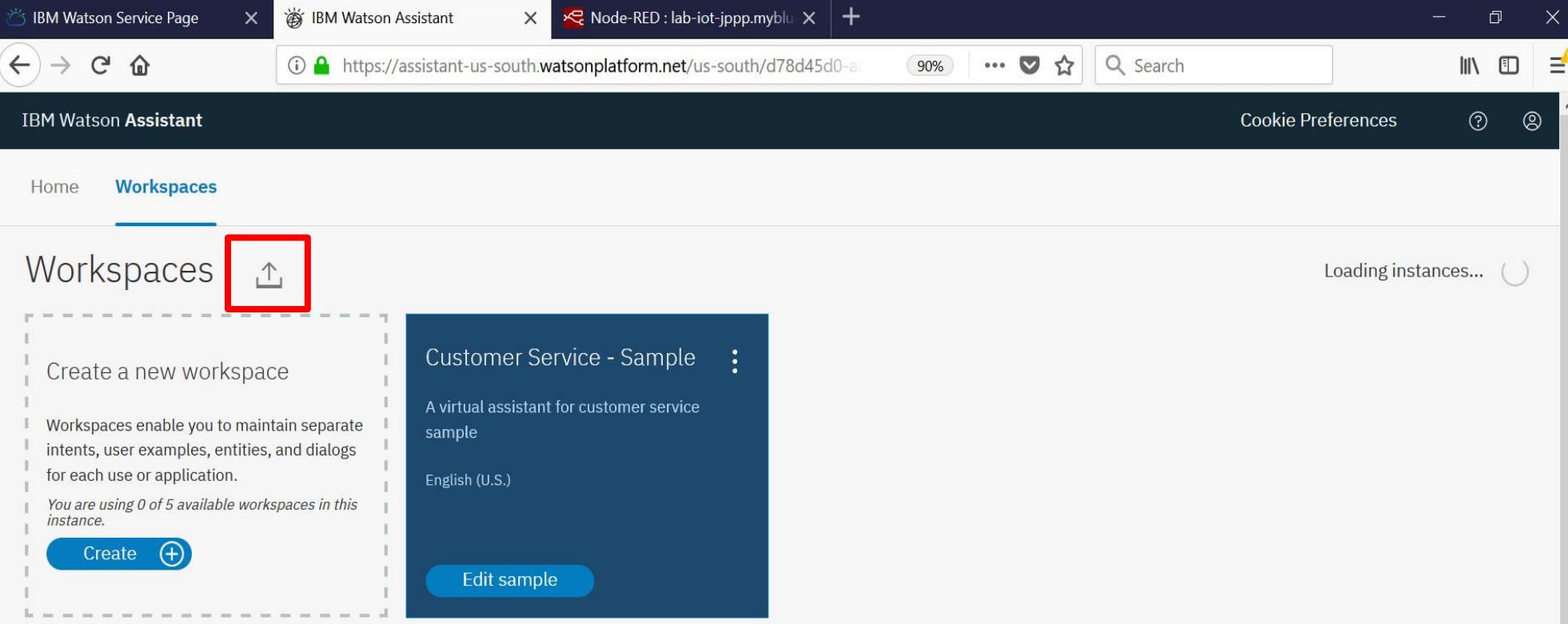
2 Build your dialog
Utilize the intents and entities you created, plus context from the application, so your virtual assistant responds appropriately.
[Learn more](#)

3 Test your dialog
Try out the virtual assistant in the tool to see how it recognizes the intents and entities and how it responds firsthand.
[Learn more](#)

5. Criando o Assistente Virtual com o Watson Assistant

5.2. Fazendo o upload do Workspace da demonstração.

- Salve no seu computador o arquivo JSON desse link:
<https://github.com/Bluedev-BR/NLU-IoT/blob/master/Assistant/workspace-f082e4e6-ecb0-4439-9b69-a9ccfa7a4a9c.json>



The screenshot shows the IBM Watson Assistant interface in a web browser. The title bar includes tabs for 'IBM Watson Service Page', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.mybluemix.net'. The main content area has a dark header with 'IBM Watson Assistant' and a navigation bar with 'Home' and 'Workspaces' (which is underlined). Below this, there's a section for 'Workspaces' with a large 'Upload' icon (an upward arrow) highlighted by a red box. To the right, a sample workspace titled 'Customer Service - Sample' is displayed, showing it's an English (U.S.) virtual assistant for customer service. At the bottom, there's a promotional section for the latest features with a 'Try out the latest features' button, 'Learn More', and 'Request Beta' links.

IBM Watson Assistant

Home Workspaces

Workspaces Upload

Create a new workspace

Workspaces enable you to maintain separate intents, user examples, entities, and dialogs for each use or application.

You are using 0 of 5 available workspaces in this instance.

[Create](#) +

Customer Service - Sample ...

A virtual assistant for customer service sample

English (U.S.)

[Edit sample](#)

Try out the latest features

A new way to build virtual assistants is coming. Be one of the first to test drive this new approach in your non-production service instance.

[Learn More](#) [Request Beta](#)

5. Criando o Assistente Virtual com o Watson Assistant

5.2. Fazendo o upload do Workspace da demonstração.

- Clique no símbolo indicado para fazer um upload.
- Escolha o arquivo JSON baixado do git.

The screenshot shows the IBM Watson Assistant interface in a web browser. The title bar includes tabs for 'IBM Watson Service Page', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The main content area has a dark header with 'IBM Watson Assistant' and a navigation bar with 'Home' and 'Workspaces' (which is underlined). Below this, there's a section for 'Workspaces' with a red box around the upward arrow icon. To the right, a sample workspace titled 'Customer Service - Sample' is displayed with a description and an 'Edit sample' button. At the bottom, there's a promotional message about trying out beta features.



Try out the latest features

A new way to build virtual assistants is coming. Be one of the first to test drive this new approach in your non-production service instance.

[Learn More](#)

[Request Beta](#)

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Ache o nó com a intenção “#Medicao”.

The screenshot shows the IBM Watson Assistant interface within a browser window. The title bar includes tabs for 'IBM Watson Service Page', 'IBM Watson Assistant', and 'Node-RED : lab-iot-jppp.myblu'. The main content area is titled 'IBM Watson Assistant' and shows the 'Dialog' tab selected. Below it, there are buttons for 'Add node', 'Add child node', 'Add folder', and 'Settings'. The main workspace is labeled 'Lab' and contains four intent cards:

- Bem-vindo (welcome) - 1 Response / 0 Context set / Does not return
- #Saudacao - 1 Response / 0 Context set / Does not return
- #Pisca - 1 Response / 0 Context set / Does not return
- #Medicao - This intent is highlighted with a red rectangular box.

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Ache o nó com a intenção “#Medicao”. Clique para editá-lo.

- Clique no ícone com 3 pontos na vertical e selecione a opção “Open JSON editor”

IBM Watson Assistant

Workspaces / Lab / Build

Intents Entities Dialog Content Catalog

Add node Add child node Add folder

Lab

Bem-vindo
welcome
1 Response / 0 Context set / Does not return

#Saudacao
1 Response / 0 Context set / Does not return

#Pisca
1 Response / 0 Context set / Does not return

#Medicao

Name this node...

If bot recognizes:
#Medicao

Then respond with:

Text

Enter response text

Response variations are set to sequential. Set to random | multiline

+ Add response type

Open JSON editor

Open context editor

https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-ad15-407e-9e61-036749f38f39/workspaces/9c5cd84-0ce7-4676-aac2-a5a9150eec19/build/dialog#

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Preencha as propriedades “user” e “password” com as credenciais usadas pela requisição CURL exemplo de sua Cloud Function:
Auth Username e Auth Password do seu txt.

Name this node...

Then respond with:

```
1 | {
2 |   "context": {
3 |     "private": {
4 |       "my_credential": {
5 |         "user": "redacted",
6 |         "password": "redacted"
7 |       }
8 |     }
9 |   },
10 |   "output": {
11 |     "generic": [
12 |       {
13 |         "values": [],
14 |         "response_type": "text",
15 |         "selection_policy": "sequential"
16 |       }
17 |     ],
18 |     "actions": []
19 |   }
}
```

Tips:

- Array [value1, value2]
- Complex object {"variable1": value1, "variable2": value2}
- String "value1"

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Preencha as propriedades “Sua_ORG”, “Seu_SPACE” com os parâmetros usadas pela requisição CURL exemplo de sua Cloud Function. Adicione, ao final, o nome de sua action que pega a medição do sensor (esp-sensor).

The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar includes tabs for "IBM Cloud Functions - Action", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.mybluemix.net", "Bluemix-Estag/NLU-IoT", and a "+" button. The main area has a dark header with "IBM Watson Assistant" and a "Cookie Preferences" button. Below the header, there's a breadcrumb navigation: "Workspaces / Lab / Build". The "Dialog" tab is selected, showing a list of nodes under the "Lab" workspace. The nodes include "Bem-vindo" (welcome), "#Saudacao" (Saudacao), "#Pisca" (Pisca), "#Medicao" (Medicao), and a partially visible node starting with "#". Each node has a "1 Response / 0 Context set / Does not return" status. To the right of the nodes is a large text area titled "Name this node..." with a "Customize" button. Below it, the text "Then respond with:" is followed by a JSON code editor. The JSON code is as follows:

```
11  "generic": [
12    {
13      "values": [],
14      "response_type": "text",
15      "selection_policy": "sequential"
16    }
17  ],
18  "actions": [
19    {
20      "name": "/<Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
21      "type": "server",
22      "parameters": {
23        "cmd": ""
24      },
25      "credentials": "$private.my_credential",
26      "result_variable": "$result"
27    }
28  ]
29 }
30 }
```

A red rectangular box highlights the "name" field in the JSON code, which contains the placeholder "/<Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>".

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- “Sua_ORG”, “Seu_SPACE” são análogos a Namespace_org e Namespace_space . O nome das suas actions foi salvo em Action1 ou Action 2.

The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar includes tabs for "IBM Cloud Functions - Action", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.mybluemix.net", and "Bluemix-Estag/NLU-IoT". The main area is titled "IBM Watson Assistant" and shows the "Dialog" tab selected. On the left, there's a sidebar with icons for Workspaces, Lab, Intents, Entities, and Dialog. The "Lab" section contains four dialog nodes: "Bem-vindo" (welcome), "#Saudacao" (Saudacao), "#Pisca" (Pisca), and "#Medicao" (Medicao). Each node has a "Name this node..." input field above it and a "Then respond with:" section below. The "Then respond with:" section for "#Medicao" contains a JSON code editor. A red box highlights the "name" field in the JSON code:

```
11  "generic": [
12    {
13      "values": [],
14      "response_type": "text",
15      "selection_policy": "sequential"
16    }
17  ],
18  "actions": [
19    {
20      "name": "/<Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
21      "type": "server",
22      "parameters": {
23        "cmd": ""
24      },
25      "credentials": "$private.my_credential",
26      "result_variable": "$result"
27    }
28  ]
29 ]
30 }
```

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Siga os mesmos passos para os nós “#apaga”, “#acende” e “#pisca” .

The screenshot shows the IBM Watson Assistant interface in a browser window. The URL is <https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-ad15-407e-9e61-036749f38f39/workspaces/9c5cd84-0ce7-4676-aac2-a5a9150eec19/build/dialog#>. The left sidebar shows 'Workspaces / Lab / Build' with tabs for 'Intents', 'Entities', 'Dialog' (selected), and 'Content Catalog'. The main area displays a list of intents: '#Saudacao', '#Pisca', '#Medicao', '#Ascende', and '#Apaga'. A modal dialog is open for the '#Apaga' intent, titled 'Name this node...'. It contains fields for 'If bot recognizes:' with the value '#Apaga' and 'Then respond with:' which is currently set to 'Text'. A dropdown menu in the top right corner of this modal offers 'Open JSON editor' and 'Open context editor' options. The status bar at the bottom of the browser window shows the same URL.

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Siga os mesmos passos para os nós “#apaga”, “#acende” e “#pisca”.

- Preencha as propriedades “user” e “password”.

The screenshot shows the IBM Watson Assistant interface in a browser window. The URL is https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-a... The main area displays a list of intents: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. The #Apaga intent is currently selected, indicated by a blue border around its card. To the right, a modal dialog is open for creating a new node. The title of the dialog is "Name this node..." and it contains the text "#Apaga". Below this, the instruction "Then respond with:" is followed by a code editor containing a JSON-like response template. The template includes fields for context, private data (with user and password), output (generic responses), and actions. The "user" and "password" fields in the private data section are highlighted with red boxes, indicating they are the values to be filled in from the Cloud Function credentials.

```
1 {
2   "context": {
3     "private": {
4       "my_credential": {
5         "user": "REDACTED",
6         "password": "REDACTED"
7       }
8     }
9   },
10  "output": {
11    "generic": [
12      {
13        "values": [],
14        "response_type": "text",
15        "selection_policy": "sequential"
16      }
17    ],
18  },
19  "actions": [
```

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Siga os mesmos passos para os nós “#apaga”, “#acende” e “#pisca”.

- Preencha as propriedades “user” e “password”.

- Preencha as propriedades “Sua_ORG”, “Seu_SPACE”

Name this node...

men responda com:

```
12      "values": [],
13      "response_type": "text",
14      "selection_policy": "sequential"
15    }
16  ],
17  "actions": [
18    {
19      "name": "/<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
20      "type": "server",
21      "parameters": {
22        "cmd": "turn",
23        "value": "off"
24      },
25      "credentials": "$private.my_credential",
26      "result_variable": "$result"
27    }
28  ]
29 }
30 }
31 }
```

Tips:

- Array [value1, value2]
- Complex object {"variable1": value1, "variable2": value2}
- String "value1"

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Coloque o nome da Action que controla o LED (no exemplo usamos esp-LED).

The screenshot shows the IBM Watson Assistant interface in a browser window. The URL is https://assistant-us-south.watsonplatform.net/us-south/d78d45d0-a... The main area displays a list of intents: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. The #Apaga intent is currently selected. A modal window titled 'Name this node...' is open, with the placeholder 'men responda com:'. Below the placeholder is a code editor containing the following JSON-like code:

```
12   "values": [],
13   "response_type": "text",
14   "selection_policy": "sequential"
15 }
16 ],
17 "actions": [
18   {
19     "name": "/<Seu_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
20     "type": "server",
21     "parameters": {
22       "cmd": "turn",
23       "value": "off"
24     },
25     "credentials": "$private.my_credential",
26     "result_variable": "$result"
27   }
28 ]
29 ]
30 }
31 }
```

A red box highlights the action name in the code: "/<Seu_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>".

5. Criando o Assistente Virtual com o Watson Assistant

5.3. Colocando as credenciais da sua Cloud Function.

- Coloque o nome da Action que controla o LED (no exemplo usamos esp-LED).

- Também precisamos mandar os parâmetros da função. Como queremos desligar: comando = “turn”, valor = “OFF”.

The screenshot shows the IBM Watson Assistant interface in a browser window. The title bar has tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, IBM Watson Assistant, Node-RED : lab-iot-jp, NLU-IoT/esp-sensor.js, and a plus sign. The main area is titled 'IBM Watson Assistant' with sub-sections 'Workspaces / Lab / Build'. The 'Dialog' tab is selected. On the left, there's a sidebar with icons for Workspaces, Entities, Dialog, Content Catalog, and a refresh button. The main content area lists five intents: '#Saudacao', '#Pisca', '#Medicao', '#Ascende', and '#Apaga'. Each intent has a description below it: '1 Response / 0 Context set / Does not return'. To the right of the intents, a modal window is open with the heading 'Name this node...'. Below it, it says 'then respond with:' followed by a code editor. The code is a JSON snippet:

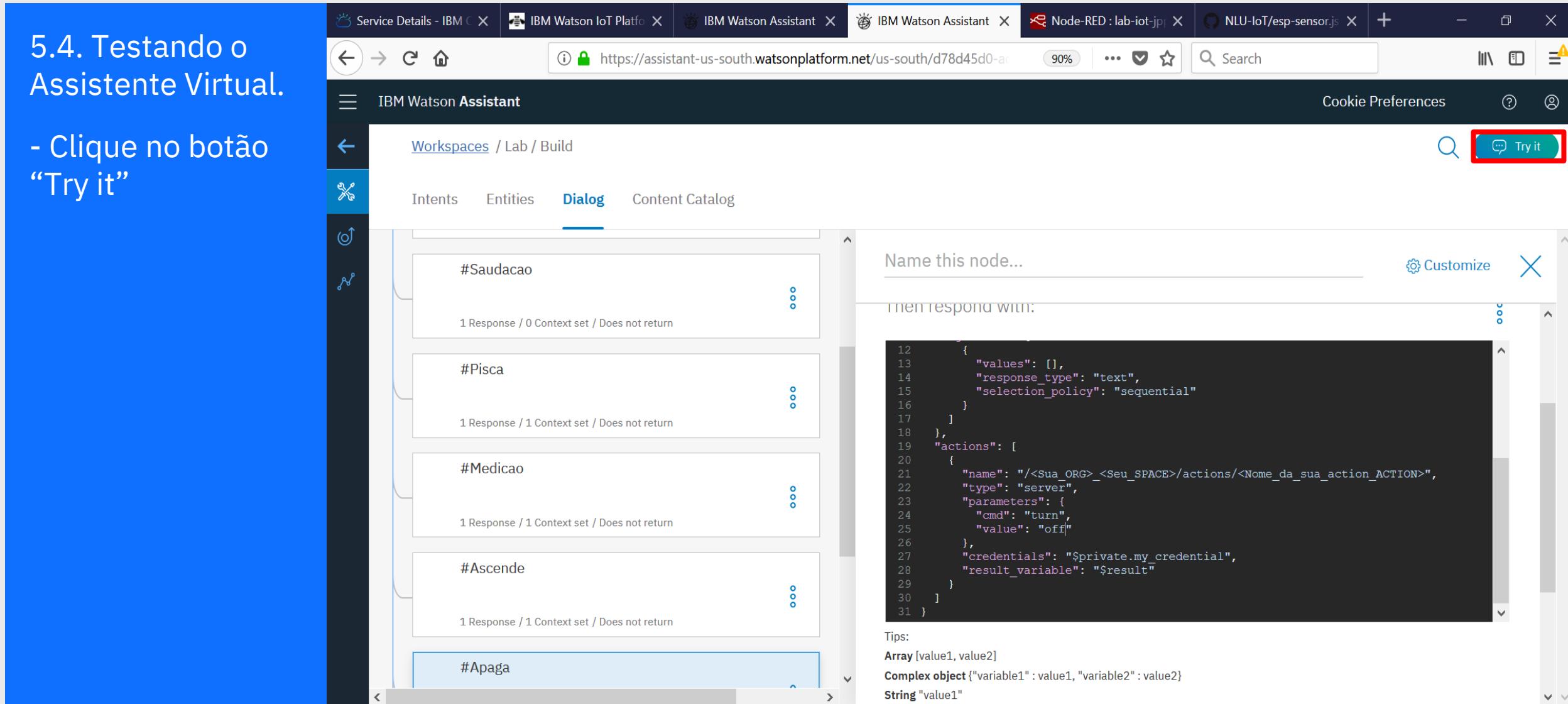
```
12     "values": [],
13     "response_type": "text",
14     "selection_policy": "sequential"
15   }
16 ],
17 "actions": [
18   {
19     "name": "</Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
20     "type": "server",
21     "parameters": {
22       "cmd": "turn",
23       "value": "off"
24     },
25     "credentials": "$private.my_credential",
26     "result_variable": "$result"
27   }
28 ]
29 ]
30 }
31 }
```

A red box highlights the 'parameters' section of the JSON code. Below the code editor, there are 'Tips:' and three examples: 'Array [value1, value2]', 'Complex object {"variable1": value1, "variable2": value2}', and 'String "value1"'.

5. Criando o Assistente Virtual com o Watson Assistant

5.4. Testando o Assistente Virtual.

- Clique no botão “Try it”



The screenshot shows the IBM Watson Assistant interface within a browser window. The title bar includes tabs for Service Details - IBM, IBM Watson IoT Platfo, IBM Watson Assistant, IBM Watson Assistant, Node-RED : lab-iot-jp, NLU-IoT/esp-sensor.js, and a plus sign. The main content area has a dark header with 'IBM Watson Assistant' and a navigation menu. Below is a 'Workspaces / Lab / Build' section with tabs for Intents, Entities, Dialog (which is selected), and Content Catalog. On the left, there's a sidebar with icons for workspace management. The main workspace displays five intents: #Saudacao, #Pisca, #Medicao, #Ascende, and #Apaga. Each intent card shows '1 Response / 0 Context set / Does not return'. To the right, a modal dialog is open with the heading 'Name this node...'. Below it, the text 'then respond with:' is followed by a code editor containing a JSON-like configuration. The code includes variables like \$private.my_credential and \$result. A red box highlights the 'Try it' button in the top right corner of the modal. At the bottom of the modal, there are tips for Array, Complex object, and String data types.

```
12   "values": [],
13   "response_type": "text",
14   "selection_policy": "sequential"
15 }
16 ],
17 "actions": [
18   {
19     "name": "/<Sua_ORG>_<Seu_SPACE>/actions/<Nome_da_sua_action_ACTION>",
20     "type": "server",
21     "parameters": {
22       "cmd": "turn",
23       "value": "off"
24     },
25     "credentials": "$private.my_credential",
26     "result_variable": "$result"
27   }
28 ]
29 ]
30 }
31 }
```

Tips:

- Array [value1, value2]
- Complex object {"variable1" : value1, "variable2" : value2}
- String "value1"

5. Criando o Assistente Virtual com o Watson Assistant

5.4. Testando o Assistente Virtual.

- Clique no botão “Try it”

- Faça uma interação com o assistente, exemplo: “Faça o LED do ESP ligar”.

The screenshot shows the IBM Watson Assistant interface in a web browser. The title bar includes tabs for "IBM Cloud Functions - Action", "IBM Watson Assistant", "IBM Watson Assistant", "Node-RED : lab-iot-jppp.mybluemix.net", and "NLU-IoT/esp-LED.js at master". The main window displays a "Dialog" view with a tree structure under "Lab". The nodes are:

- Bem-vindo (welcome)
- #Saudacao
- #Pisca
- #Medicao

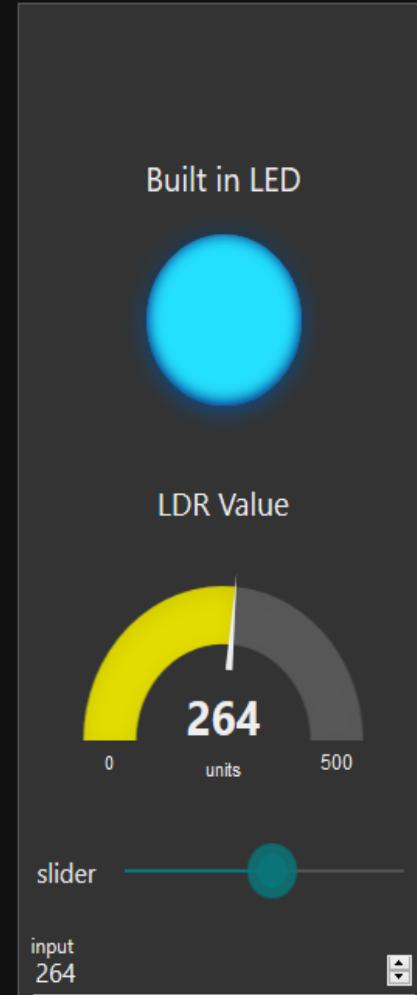
Each node has a status line below it: "1 Response / 0 Context set / Does not return". To the right of the tree, there is a "Try it out" section with a text input field containing "Olá. Como posso te ajudar?". A red box highlights this input field. Below the input field, instructions say "Enter something to test your virtual assistant" and "Use the up key for most recent".

5. Criando o Assistente Virtual com o Watson Assistant

5.4. Testando o Assistente Virtual.

- Clique no botão “Try it”
- Faça uma interação com o assistente, exemplo: “Faça o LED do ESP ligar”.
- O LED do dispositivo simulado deve acender.
- Faça os testes que quiser!

Simulated Device

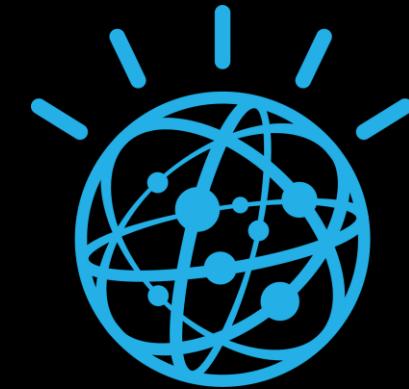


Concluindo:

Digitalizamos o mundo físico.

Interagimos em
linguagem natural.

**De forma escalável
e eficiente.**



Obrigado

Eduardo Petecof Mattoso

Watson & Cloud TechSales Intern

epetecof@br.ibm.com

João Pedro Poloni Ponce

Value Creation Intern - BlueDev

joao.poloni@ibm.com

