

# AI & CHATBOT

Aula 05 – Watson Assistant e  
Slots e melhorias

Prof. Henrique Ferreira  
Prof. Miguel Bozer  
Prof. Guilherme Aldeia  
Prof. Michel Fornaciali  
Prof. Daniel Petrini

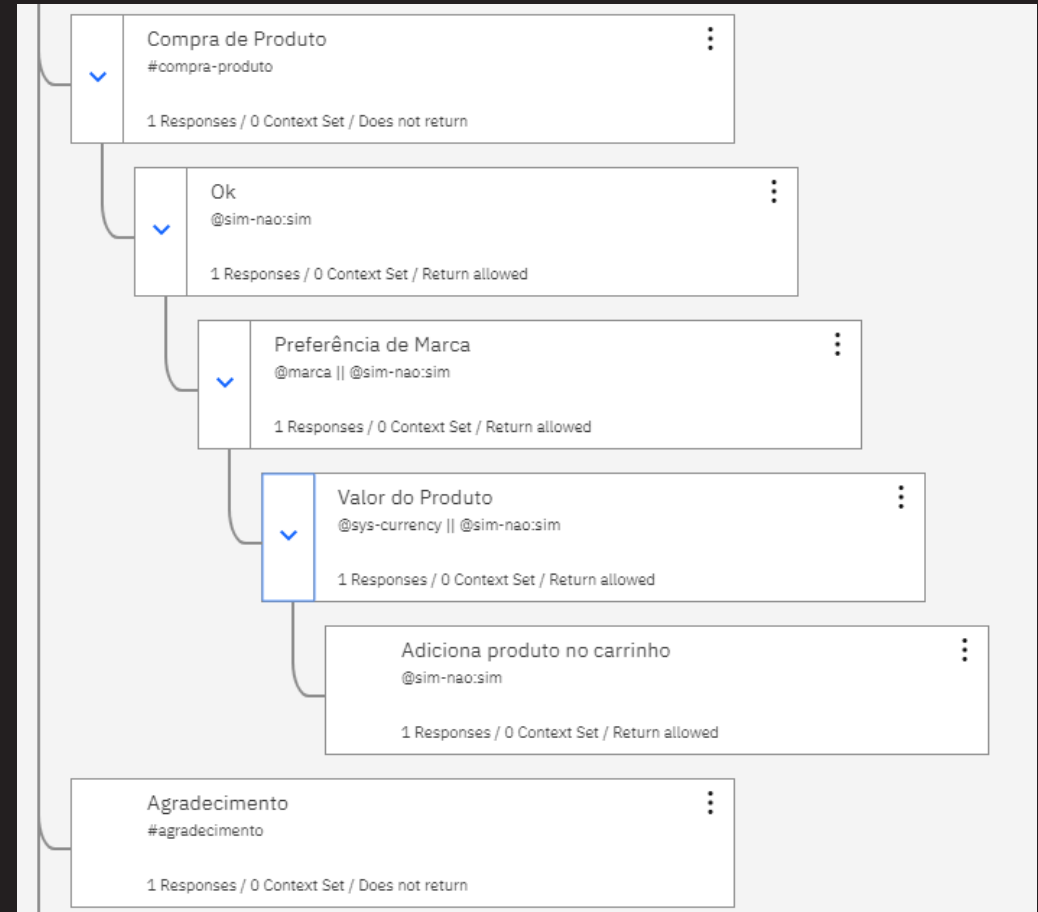
FIAP  
GRADUAÇÃO

# Melhorando o diálogo

Transformando em um único nó.

# Transformando em um único nó.

- Veja como ficou nossa árvore.
- Ela está grande demais, e isso pode causar problemas de desempenho, não só computacional, mas também na manutenção do fluxo da conversa.
- Vamos melhorar esse fluxo.

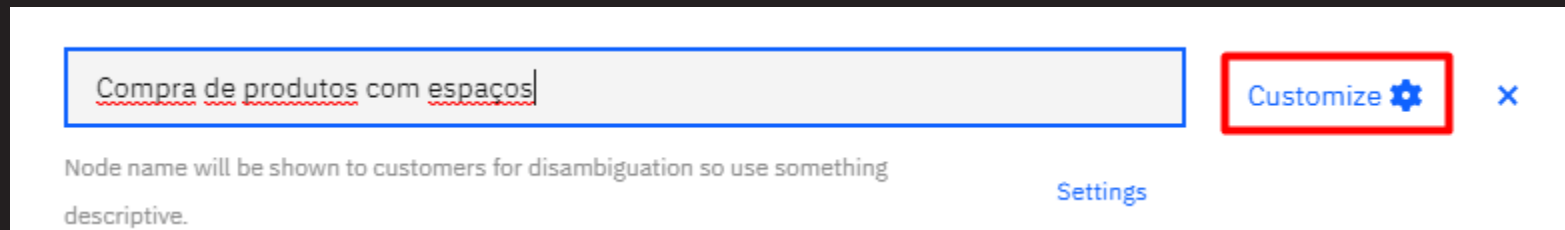


# Transformando em um único nó.

- Um dos pontos que mais geram problemas em árvores grandes são as manutenções, assim como a reutilização tanto das informações, quanto dos nós.
- Dessa maneira, o Watson disponibiliza para nós alguns recursos do Assistant para nos auxiliar nessas tarefas.
- Vamos criar um novo nó chamado Compra de produto com espaços, para deixarmos os dois fluxos como comparação.

# Transformando em um único nó.

- Antes de continuarmos a configuração que conhecemos, vamos selecionar a opção Customize, disponível logo ao lado do nome.



The screenshot shows a configuration interface with a text input field containing the text "Compra de produtos com espaços". To the right of the input field is a button labeled "Customize" with a gear icon, which is highlighted with a red rectangular box. To the right of the "Customize" button is a small blue "x" icon. Below the input field, there is a small text label that reads "Node name will be shown to customers for disambiguation so use something descriptive." and a link labeled "Settings".


# Transformando em um único nó.

- Habilite a opção de slots e clique em Apply

Customize "Compra de produtos com espaços"

Customize node      Digressions

---

Slots ⓘ  On

Enable this to gather the information your bot needs to respond to a user within a single node.

☐ Prompt for everything

Enable this to ask for multiple pieces of information in a single prompt, so your user can provide them all at once and not be prompted for them one at a time.

# Transformando em um único nó.


- Vamos preencher a intenção para reconhecimento e também a primeira condição de sim e não.

If assistant recognizes

#compra-produto - +





---

Then check for 0 Manage handlers

	Check for	Save it as	If not present, ask	Type	
1	@sim-nao	\$sn	Legal, vou te	Required	 -

# Transformando em um único nó

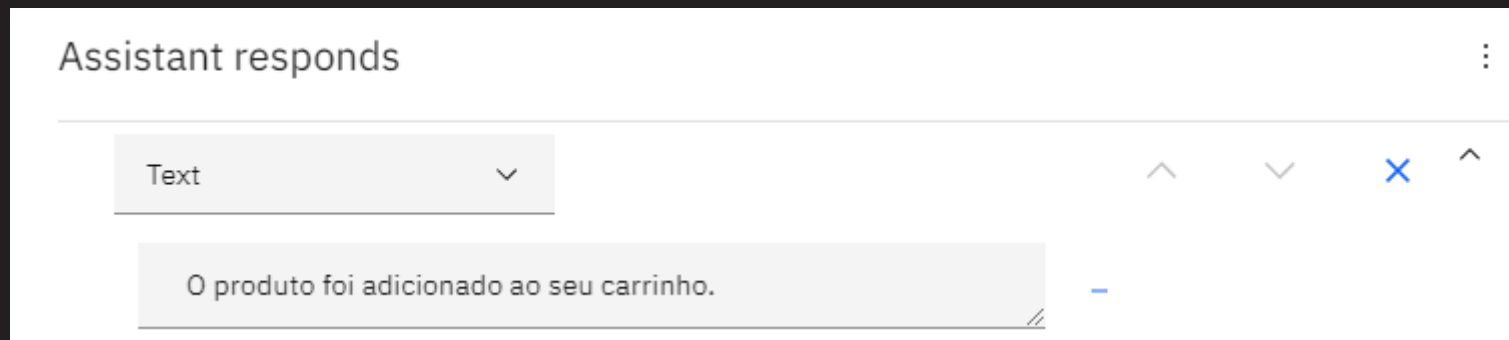
- Vamos continuar preenchendo os slots de acordo com o diálogo. Esse é o resultado:

	Check for	Save it as	If not present, ask	Type		
1	@sim-nao	\$sn	Legal, vou te	Required		—
2	@marca	\$marca	Você tem pr	Required		—
3	@sys-curren	\$currency	E você tem a	Required		—
^						
4	@sim-nao	\$sim_nao	Eu tenho um	Required		—



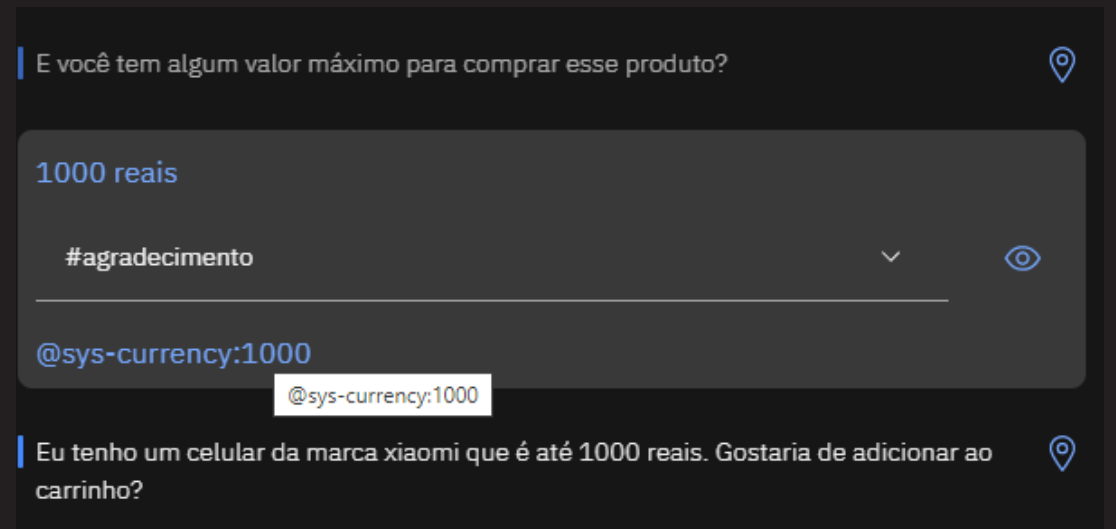
# Transformando em um único nó

- No slot 4, aproveitamos e utilizamos as variáveis salvas \$marca e \$currency para utilizá-las na pergunta:
  - Eu tenho um celular da marca \$marca que é até \$currency reais. Gostaria de adicionar ao carrinho?
- Agora colocamos a resposta final:



# Transformando em um único nó

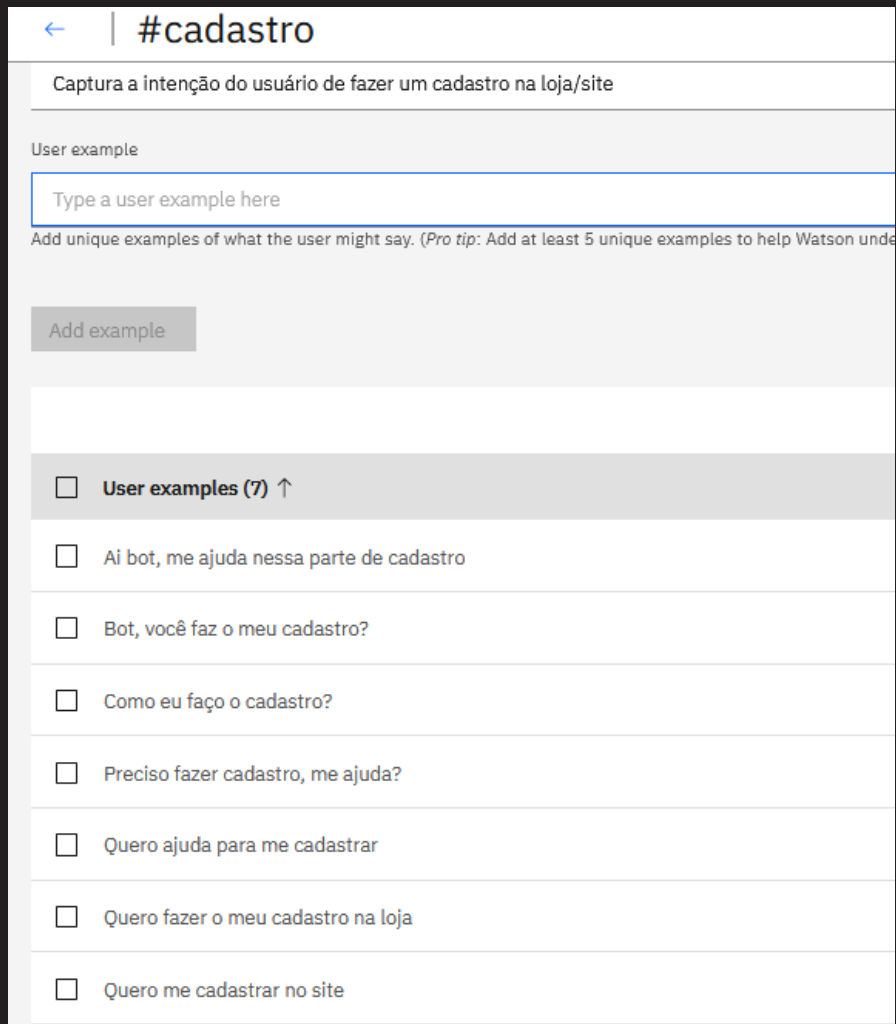
- Hora de testar!
- Perceba que funcionou a marca e a moeda dentro da sequência de espaço.
- Pronto! Agora nosso chatbot ficou mais leve e de fácil manutenção =D



# Capturando varias informações

Expressões Regulares, Slots e Variáveis de Contexto.

# Vamos criar um nó de cadastro para pegar essas informações: precisamos de intenções e entidades



The screenshot shows the IBM Watson Assistant console interface for configuring an intent named '#cadastro'. At the top, there is a back arrow and the intent name '#cadastro'. Below this, a description box contains the text 'Captura a intenção do usuário de fazer um cadastro na loja/site'. Underneath the description is a section labeled 'User example' with a text input field containing 'Type a user example here'. Below the input field is a note: 'Add unique examples of what the user might say. (Pro tip: Add at least 5 unique examples to help Watson understand)'. There is an 'Add example' button. Below this is a list of user examples, each with a checkbox and a text input field. The list is titled 'User examples (7) ↑'.

<input type="checkbox"/>	User examples (7) ↑
<input type="checkbox"/>	Ai bot, me ajuda nessa parte de cadastro
<input type="checkbox"/>	Bot, você faz o meu cadastro?
<input type="checkbox"/>	Como eu faço o cadastro?
<input type="checkbox"/>	Preciso fazer cadastro, me ajuda?
<input type="checkbox"/>	Quero ajuda para me cadastrar
<input type="checkbox"/>	Quero fazer o meu cadastro na loja
<input type="checkbox"/>	Quero me cadastrar no site

- Primeiramente temos que criar uma intenção para iniciar o cadastro. Adicionem mais exemplos.

# Criando as variáveis que serão capturadas

- Agora, vamos criar as entidades cep, email e telefone para incluir os padrões a serem reconhecidos. Vamos criar cada um delas separadamente.
- Vamos começar com o cep, basta digitar cep, desabilitar o “Fuzzy matching” e clicar em criar.

IBM Watson Assistant Lite [Upgrade](#)

← | Create new entity

Entity name  
Name your entity to match the category of values that it will detect.

@cep

[Create entity](#)

Fuzzy matching ⓘ  
☐ Off

# Usando Expressões Regulares para capturar padrões

- Temos que selecionar “Patterns” e incluir o padrão para extrair o cep. Vamos supor que o cep tem dois padrões: 00000-000 ou 00000 000.
- Para isso vamos inserir o comando `\d` que procura por qualquer número e dentro dos parênteses a quantia de números, ou seja, 5 e depois 3 números.
- Isso implica que temos `\d{5}` que procurará por cinco números em sequência e `\d{3}` que procurará por três números em sequência.

The screenshot shows a web interface for defining a pattern for the entity '@cep'. The interface has a header with a back arrow and the entity name '@cep'. Below the header, there is a section for 'Entity name' with a text input field containing '@cep'. To the right of this section is a 'Fuzzy matching' toggle switch, which is currently turned off. Below the 'Entity name' section, there is a 'Value' section with a text input field containing 'cep'. To the right of the 'Value' section, there is a 'Patterns' section. This section contains a dropdown menu with 'Patterns' selected, followed by a text input field containing the regular expression `(\d{5}-\d{3})`, and then a text input field containing the regular expression `(\d{5} \d{3})`. The first two input fields are highlighted with red boxes. At the bottom left of the interface, there is a blue button labeled 'Add value'.

# Usando Expressões Regulares para capturar padrões

- O resultado final do cep deveria ser como a imagem abaixo:

The screenshot shows a web interface for defining an entity. At the top, there's a header with a back arrow and the text "@cep". Below this, the "Entity name" section has a text input field containing "@cep" and a "Fuzzy matching" toggle switch set to "off". The "Value" section has a text input field with the placeholder "Type a value, e.g. Checking". The "Synonyms" section has a dropdown menu labeled "Synonyms" and a text input field with the placeholder "Type a synonym, e.g. Deposit", followed by a "+" button. Below these sections is an "Add value" button. At the bottom, there's a table with the following content:

<input type="checkbox"/>	Values (1) ↑	Type
<input type="checkbox"/>	cep	Patterns
		(\d{5}-\d{3}), (\d{5} \d{3})

# Usando Expressões Regulares para capturar padrões

- Criamos a entidade telefone, com os valores fixo e celular e com os padrões demonstrados abaixo:

← | @telefone

Last updated: há poucos segundos

Try

Entity name

@telefone

Fuzzy matching ⓘ

Off

Name your entity to match the category of values that it will detect.

Value

Synonyms

Type a value

Synonyms ▾

Type a synonym

+

Add value

☐ Values (2) ↓

Type

☐ fixo

Patterns

(\d{2}) (\d{4}-\d{4}), (\d{2}) (\d{8})

☐ celular

Synonyms ▾

(\d2) (\d{5}-\d{4})

-

(\d2) (\d{9})

-



# Usando Expressões Regulares para capturar padrões

- Criamos a entidade email, com o valor email e com o padrão demonstrado abaixo:

← | @email

Entity name

Name your entity to match the category of values that it will detect.

@email

Fuzzy matching ⓘ

☐ Off

Value

Type a value, e.g. Checking

Synonyms

Synonyms ▼

Type a synonym, e.g. Deposit

+

Add value

([a-zA-Z0-9+.\_-]+@[a-zA-Z0-9.\_-]+\.[a-zA-Z0-9\_-]+)

\b[A-Za-z0-9.\_%+-]+@([A-Za-z0-9-]+\.)+[A-Za-z]{2,}\b

☐ Values (1) ↑

email

Patterns

\b[A-Za-z0-9.\_%+-]+@([A-Za-z0-9-]+\.)+[A-Za-z]{2,}\b

# Vamos usar Slot para criar o nó de Cadastro

- Basta criar agora o nó de diálogo, inserir a intenção #cadastro e ativar o Slot:

Enter node name (optional) Customize

Node name will be shown to customers for disambiguation so use something descriptive. [Settings](#)

If assistant recognizes

#cadastro

Then check for [Manage handlers](#)

	Check for	Save it as	If not present, ask	Type
1	Enter condition	Enter variable	Enter prompt	Optional

[Add slot](#)

Assistant responds

Text

Enter response text

Response variations are set to **sequential**. Set to [random](#) | [multiline](#)



Customize "Untitled"

[Customize node](#) [Digressions](#)

Slots [?](#)

☒ On

Enable this to gather the information your bot needs to respond to a user within a single node.

☐ Prompt for everything

Enable this to ask for multiple pieces of information in a single prompt, so your user can provide them all at once and not be prompted for them one at a time.

Webhooks ☐ Off

Enable this setting to send a POST request from this dialog node to the webhook URL. The URL and headers are defined in the Webhooks settings of the Options tab. After you enable this setting, the Multiple conditional responses setting is enabled automatically to support adding a response to show when the request is successful and another response to show if the request fails. [Learn more](#)

Multiple conditioned responses [?](#) ☐ Off

[Cancel](#) [Apply](#)

18/51

# Criando o Slot para o nó de Cadastro




- Basta criar adicionar as entidades para identificar cada uma delas e inserir um nome para variável de contexto, como segue :

If assistant recognizes

#cadastro - +

---

Then check for 0 Manage handlers

	Check for	Save it as	If not present, ask	Type	
1	@cep	\$cep	Qual o cep?	Required	<div> -</div>
2	@email	\$email	Qual o e-mail?	Required	<div> -</div>
3	@telefone	\$telefone	Qual o telefone?	Required	<div> -</div>

# Criando o Slot para o nó de Cadastro

- Basta configurar cada slot individualmente:

Configure slot 1

Check for: @cep

Save it as: \$cep

If slot context variable is not present ask: Slot is required ☒

Text

Qual o cep?

Enter response variation

Response variations are set to **sequential**. Set to [random](#)

[Learn more](#)

Cancel Save

# Slots, Variáveis de Contexto e Expressões Regulares

- Basta inserir `.literal` depois da referência da entidade para garantir que o valor extraído seja reconhecido e não a referência.

Configure slot 1

```
1 {
2   "context": {
3     "cep": "@cep"
4   }
5 }
```

If slot context variable is not present ask: Slot is required ⓘ

Text

Cancel Save



Configure slot 1

```
1 {
2   "context": {
3     "cep": "@cep.literal"
4   }
5 }
```

If slot context variable is not present ask: Slot is required ⓘ

Text

Cancel Save

21/51

# Slots, Variáveis de Contexto e Expressões Regulares

Basta repetir o processo para as outras entidades: email e telefone

Configure slot 2

```
1 {
2   "context": {
3     "email": "@email.literal"
4   }
5 }
```

If slot context variable is not present ask: Slot is required ⓘ

Text

Cancel Save

Configure slot 3

```
1 {
2   "context": {
3     "telefone": "@telefone.literal"
4   }
5 }
```

If slot context variable is not present ask: Slot is required ⓘ



Text

Cancel Save

# Slots, Variáveis de Contexto e Expressões Regulares


- Vamos adicionar um pergunta de cabeçalho para saber que estamos iniciando o cadastro:

Cadastro

Customize  







Node name will be shown to customers for disambiguation so use something descriptive. [Settings](#)


If assistant recognizes

#cadastro  +

Then check for

0 Manage handlers

	Check for	Save it as	If not present, ask	Type		
1	@cep	\$cep	Qual o cep?	Required		
2	@email	\$email	Qual o e-mail?	Required		
3	@telefone	\$telefone	Qual o telefon	Required		

Manage handlers for "Cadastro" 

Handlers are how your bot will respond when the users answer to a prompt is not found. These handlers will be checked before trying the "Not found" responses in a slot.

If answer to any prompt is not found and

Add handler +

Cancel Save

# Slots, Variáveis de Contexto e Expressões Regulares

- Vamos adicionar um pergunta de cabeçalho para saber que estamos iniciando o cadastro:

Manage handlers for "Cadastro" ×

Handlers are how your bot will respond when the users answer to a prompt is not found. These handlers will be checked before trying the "Not found" responses in a slot.

If answer to any prompt is not found and

	If assistant recognizes	Respond with	
1	true	Ok, vou realizar seu cadastro.	<span>⚙️</span> <span>🗑️</span>

[Add handler +](#)

Cancel Save

Entrega de Produto  
#entrega-produto



# Slots, Variáveis de Contexto e Expressões Regulares

Testando:

Oi, tudo bem? Posso te ajudar?

quero realizar o cadastro

#cadastro

Ok, vou realizar seu cadastro

Você poderia me informar o seu CEP?

Se, contudo, errarmos o cep, o que acontecerá?



Oi, tudo bem? Posso te ajudar?

quero realizar o cadastro

#cadastro

Ok, vou realizar seu cadastro

Você poderia me informar o seu CEP?

04041004

Irrelevant

Ok, vou realizar seu cadastro

Você poderia me informar o seu CEP?



# Slots, Variáveis de Contexto e Expressões Regulares

- Uma forma de evitar isso é fazer o handler reconhecer a intenção de cadastro:

Manage handlers for "Cadastro" ×

Handlers are how your bot will respond when the users answer to a prompt is not found. These handlers will be checked before trying the "Not found" responses in a slot.


If answer to any prompt is not found and

	If assistant recognizes	Respond with	
1	#cadastro	Ok, vou realizar seu cadastro	 


[Add handler +](#)


Cancel Save

## Testando:

Olá. Como posso te ajudar? 


quero realizar o cadastro


#cadastro × ∨ 

Ok, vou realizar seu cadastro 

Você poderia me informar o seu CEP?

04041004

Irrelevant × ∨ 

Você poderia me informar o seu CEP? 

# Slots, Variáveis de Contexto e Expressões Regulares

- Por fim, vamos terminar de configurar o nó de cadastro perguntando se o usuário confirma os dados:

The screenshot shows a configuration interface for a chatbot node named 'Cadastro'. At the top, there's a header bar with the node name 'Cadastro' and a 'Customize' button with a gear icon. Below the header, a note states: 'Node name will be shown to customers for disambiguation so use something descriptive.' with a 'Settings' link. The main area is titled 'Assistant responds' and contains a list of response slots. The first slot is a 'Text' type, highlighted with a red border, containing the message: 'Poderia confirmar que o seu cep é \$cep, o e-mail é \$email e o seu telefone é \$telefone'. Below this is a text input field labeled 'Enter response variation'. At the bottom, a note indicates that response variations are set to 'sequential' and provides links for 'random' and 'multiline' settings, along with a 'Learn more' link. The page number '27/51' is visible in the bottom right corner.

Cadastro [Customize](#)

Node name will be shown to customers for disambiguation so use something descriptive. [Settings](#)

[Add slot](#) +

Assistant responds

Text

Poderia confirmar que o seu cep é \$cep, o e-mail é \$email e o seu telefone é \$telefone

Enter response variation

Response variations are set to **sequential**. Set to [random](#) | [multiline](#)

[Learn more](#)

27/51

# Testando

Try it out

Clear Manage Context 3

✕

Oi, tudo bem? Posso te ajudar?

quero realizar o cadastro na loja

#cadastro

Ok, vou realizar seu cadastro.

Qual o cep?

04023-018

Irrelevant

@cep:cep

Qual o e-mail?

Try it out

Clear Manage Context 5

⋮

Qual o e-mail?

professor@teste.com

Irrelevant

@email:email

Qual o telefone?

11 55382650

Irrelevant

@telefone:fixo

@cep:cep

Poderia confirmar que o seu cep é 04023-018, o e-mail é professor@teste.com e o seu telefone é 11 55382650

# Descanso

Do Professor =D

# Exercícios

1. Faça o planejamento dos slots com a entidade de produto.
2. Termine o fluxo de cadastro pensando que o usuário pode responder sim ou não a confirmação dos dados. Em caso negativo, como você poderia lidar com a situação?
3. Crie um fluxo para capturar o nome do usuário e fazer o bot responder usando este nome. Dicas: você precisará criar uma variável de contexto e escolher entre criar uma entidade ou usar toda a entrada. Tente passar o valor `<? input.text ?>` como parâmetro para o contexto.

# Estudo Complementar

Dicas, links e livros

# Estudo Complementar

1. Documentação do Watson Assistant:  
<https://cloud.ibm.com/docs/assistant?topic=assistant-dialog-runtime-context>  
<https://cloud.ibm.com/docs/assistant?topic=assistant-dialog-methods>  
<https://cloud.ibm.com/docs/assistant?topic=assistant-expression-language#expression-language-shorthand-context>
1. Exemplos de manipulação de variável de contexto:  
<https://www.ibm.com/cloud/blog/enhance-chatbot-conversation-context-variables-system-entities>
2. Mais sobre JSON: <https://www.json.org/json-en.html>
3. Mais sobre Expressões Regulares:  
[https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression)



# Próximos Passos

O que veremos na próxima aula

# Na próxima aula...

- Programação com Node RED
- Integrando o Watson Assistant para utilizar outros serviços em nuvem

**Copyright © 2022**

**Slides criados por Henrique Ferreira e Miguel Bozer, adaptados do material do Prof. Andrey Masiero e Marcelo Grave - FIAP**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).