

DOCUMENTO DO PROJETO

**CHATBOT secretária e vestibular da FATEC -
Mogi Mirim – SP**

**Mogi Mirim – SP
2021**

Programa de Bolsas
Compasso UOL em parceria com a FATEC - Mogi Mirim - SP



Desenvolvedores:

Caio Gabriel Ferreira
Gustavo Goetze Marcello
João Luiz Reis Bassi
João Victor P. Barbosa
Joice Gonçalves Furigo
Michael Henrique Jacinto
Rafael Inácio
Thiago Oliveira Claudino
Vanessa Amélia Rotoli
Victor Balbino Araujo

SUMÁRIO

1.	DIRETRIZES DE PROJETO	3
2.	CLIENTE E OUTROS INTERESSADOS	5
3.	USUÁRIOS	6
4.	RASA.....	7
5.	O ESCOPO DO PRODUTO.....	19

1. DIRETRIZES DE PROJETO

1.1 NEGÓCIO DO USUÁRIO

Pelo Decreto no. 51.878, de 06/06/2007, criou-se a *Faculdade de Tecnologia de Mogi Mirim*, como uma unidade de ensino Tecnológico do Centro Estadual de Educação Tecnológica Paula Souza, administração do governo José Serra.

A partir do Convênio de Cooperação Técnico-Educacional celebrado entre o município de Mogi Mirim e o Centro Estadual de Educação Tecnológica Paula Souza, nas pessoas do prefeito Senhor Carlos Nelson Bueno e da Diretora Superintendente, professora Laura M. J. Laganá, em 28/06/2007, foi firmado o contrato entre as partes, governo estadual e municipal, objetivando o desenvolvimento e a expansão da educação profissional gratuita, por meio de cursos de nível superior tecnológico, fomentando a pesquisa e a prestação de serviços à comunidade de Mogi Mirim e seu entorno.

Os primeiros cursos em nível de ensino superior oferecidos pela FATEC Mogi Mirim foram os de Graduação de Tecnologia em Projetos Mecânicos, e Mecânica de Precisão. O primeiro dia de aula ocorreu em 10 de setembro de 2007. Os cursos ficaram distribuídos nos períodos da manhã e noite, Projetos Mecânicos, e no período da tarde, Mecânica de Precisão. No primeiro semestre de 2010, iniciou-se mais um curso de Graduação, o de Tecnologia em **Análise e Desenvolvimento de Sistemas**, nos períodos da manhã e noite.

Os cursos ministrados por uma FATEC são concebidos, desenvolvidos e ministrados visando a atender segmentos atuais e emergentes do mercado de trabalho. Dada à vocação econômica de uma região, os cursos são propostos, alinhando-se à missão da instituição: "formar a competência profissional adequada às necessidades dos diferentes mercados de trabalho com o propósito

de contribuir para a melhoria do padrão de vida do trabalhador e para a elevação da qualidade e produtividade de processos, produtos e serviços".

1.2 O PROPÓSITO DO PRODUTO

O projeto será para a FATEC, queremos oferecer para os usuários do sistema e para as pessoas que trabalham na secretaria e vestibular para uma otimização dos processos e mais agilidade na comunicação e disponibilização das informações, através de um chatbot. Atualmente com a pandemia, a secretaria da FATEC está com uma demanda de atendimentos telefônicos e por e-mail consideravelmente alto, e com a implantação desse chatbot, trará grandes benefícios para os funcionários e para os alunos, ex-alunos, e futuros alunos, que querem saber de algum determinado assunto, com o chatbot o processo do retorno das informações será instantânea, sendo mais ágil e otimizado.

2. CLIENTE E OUTROS INTERESSADOS

2.1 – CLIENTE

Fatec-Mogi Mirim “Arthur de Azevedo”

2.2 – COMPRADOR

Para finalização do projeto de bolsas da Compasso UOL foi proposto que desenvolvêssemos algo para a comunidade ou para a FATEC, como forma de aprimorarmos nossos treinamentos durante esses três meses dentro da companhia, e a nossa proposta foi unir ambos as squads e desenvolver um chatbot para o site da FATEC de Mogi Mirim – SP.

2.3 – OUTROS INTERESSADOS

Compasso Uol

Caio Gabriel Ferreira

Gustavo Goetze Marcello

João Luiz Reis Bassi

João Victor Palhares Barbosa

Joice Gonçalves Furigo

Michael Henrique Jacinto

Rafael Inácio

Thiago Oliveira Claudino

Vanessa Amélia Rotoli

Victor Balbino Araujo

3. USUÁRIOS

3.1 – USUÁRIOS DO PRODUTO

Para este caso, foram identificados três tipos de usuários do produto:

- Usuário Funcionário:

Acesso limitado a consulta;

- Usuário Técnico de Informática:

Acesso total a inclusões, alterações, consultas e relatórios;

- Usuário Público:

Acesso limitado a consulta;

4. RASA

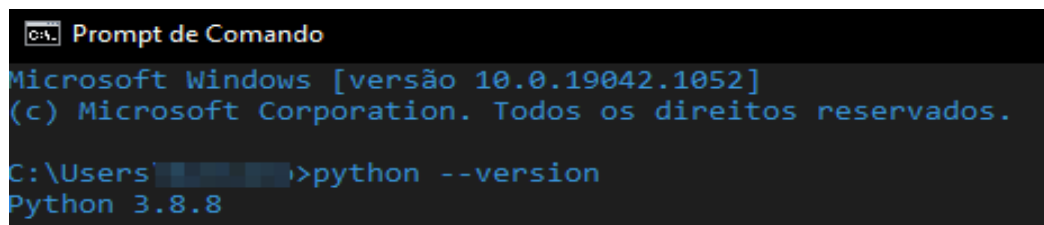
Nesta seção do Documento de Requisitos do Sistema, serão apresentadas as restrições identificadas até o presente momento, podendo sofrer alterações a qualquer hora, em função do andamento do projeto do sistema. Foi escolhida esse software por ele ser uma ferramenta que evolui consideravelmente rápido. Ele é dividido em dois componentes principais, que o permitem desenvolver todas as suas tarefas.

O primeiro componente é considerado o raciocínio e a inteligência da conversa de um usuário com um chatbot, o rasa é um software livre, é gratuito e você pode usar e alterá-lo à vontade. Esse componente também pode ser chamado do núcleo do RASA e para utilizá-lo, você só precisa adicionar a interface de usuário e as integrações que deseja fazer com outros sistemas.

O segundo componente oferece ferramentas mais elaboradas que permitem que a empresa administre a aprendizagem do chatbot. Ele possui código fechado, sendo assim, não é possível incrementar ou adaptá-lo. A empresa tem a opção de trabalhar com uma versão comunitária mais básica ou compre uma licença do Rasa para acessar todos os recursos.

4.1 – Instalação Rasa

Para iniciar a instalação é necessário que seja verificado a versão do Python instalado em sua máquina, para realizar isso abra o CMD e digite '**python --version**', essa etapa é importante devido o rasa, ser apenas compatível atualmente com as versões 3.6,3.7,3.8. Para esta instalação de demonstração será utilizado a versão descrita na imagem abaixo:



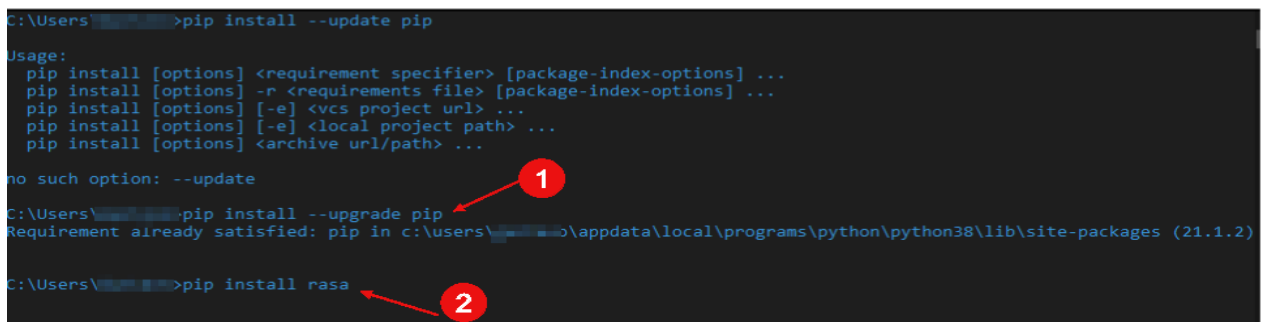
```

C:\> Prompt de Comando
Microsoft Windows [versão 10.0.19042.1052]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\>python --version
Python 3.8.8
```


Caso a versão não seja a recomendada, será necessário que haja a desinstalação e instalação da versão correta e adequada ao seu sistema operacional, essa informação está disponível neste link: <https://www.python.org/downloads/release/python-388/>

Com a versão correta já instalada, abra o CMD e verifique se o comando pip está com sua versão atualizada, caso não digite o comando '**pip install --upgrade pip**'. Após isso é só rodar o comando '**pip install -U rasa**' e aguardar a instalação como mostra imagem a seguir:



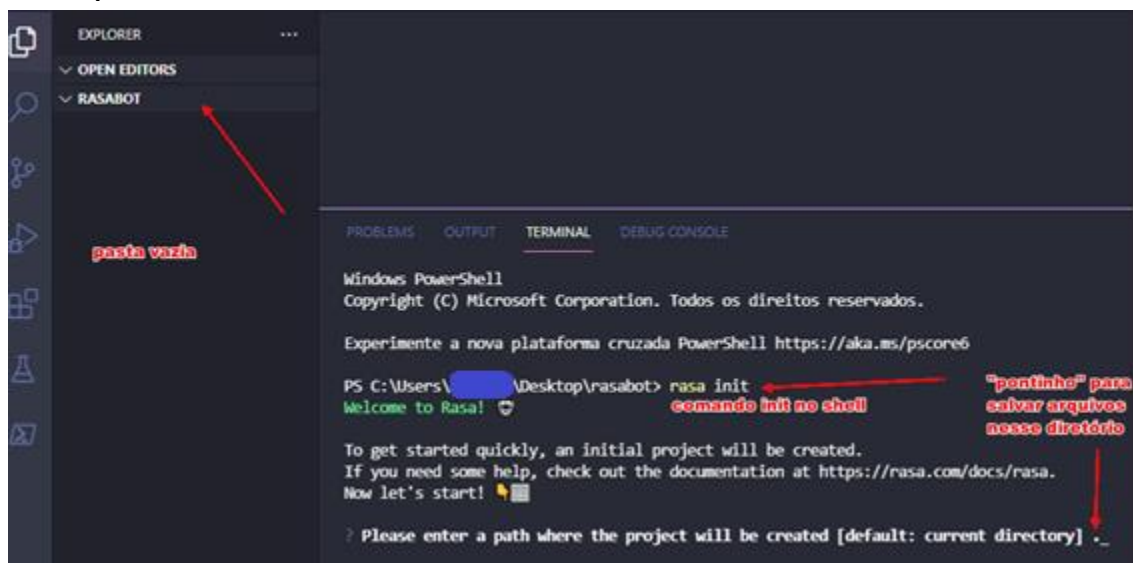
```
C:\Users\...>pip install --update pip
Usage:
  pip install [options] <requirement specifier> [package-index-options] ...
  pip install [options] -r <requirements file> [package-index-options] ...
  pip install [options] [-e] <vcs project url> ...
  pip install [options] [-e] <local project path> ...
  pip install [options] <archive url/path> ...

no such option: --update

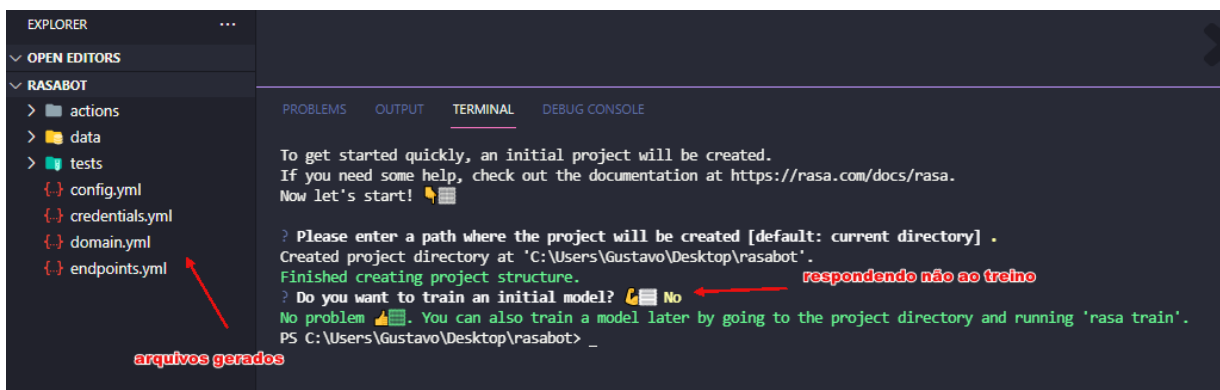
C:\Users\...>pip install --upgrade pip
Requirement already satisfied: pip in c:\users\...\appdata\local\programs\python\python38\lib\site-packages (21.1.2)

C:\Users\...>pip install rasa
```

Agora com rasa instalado crie uma pasta no CMD com o nome de sua preferência para criação do bot, abra com seu editor de texto python, nesse exemplo usaremos a ferramenta Vscode.



Será feita uma próxima pergunta de se deseja treinar imediatamente o bot, porém neste caso iremos realizar a tradução nosso bot para português, então resposta 'n' e siga para os próximos passos.



4.2 – Tradução do Bot

Começaremos pela pasta '**domain.yml**', sendo o arquivo responsável pelas respostas do bot e traduzir os '**text:**' Em nosso projeto utilizamos os dados fornecidos na planilha da secretaria da fatec no anexo 1.

```

1 version: "2.0"
2
3 intents:
4   - greet
5   - goodbye
6   - affirm
7   - deny
8   - mood_great
9   - mood_unhappy
10  - bot_challenge
11
12 actions:
13   - action_hello_world
14
15 #essas são as respostas enviadas pelo bot, ao usuário
16 responses:
17   utter_greet:
18     - text: "Oi, tudo certo?"
19
20   utter_cheer_up:
21     - text: "Olha que gatinho fofinho pra te animar :3:"
22     - image: "https://i.imgur.com/nGF1K8f.jpg"
23
24   utter_did_that_help:
25     - text: "Isso te ajudou?"
26
27   utter_happy:
28     - text: "nice, show de bola!"
29
30   utter_goodbye:
31     - text: "flw, até+"
32
33   utter_iamabot:
34     - text: "Eu queria ser o Optimusprime, mas sou um Bot rasa mesmo xD."
35
36 session_config:
37   session_expiration_time: 60
38   carry_over_slots_to_new_session: true
39

```

No Git bash, realize comando **'pip install spacy'**. Após a conclusão da instalação, realizar o download da base de dados em português do spaCy com o comando **'spacy download pt_core_news_lg'**. Em caso de dúvidas acesse esse link para obter mais informações: <https://spacy.io/usage>.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\████████\OneDrive - Fatec Centro Paula Souza\Python\Rasa\bot1> pip install spacy_

```

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\████████\OneDrive - Fatec Centro Paula Souza\Python\Rasa\bot tabuada> spacy download pt_core_news_lg_

```

O próximo passo é alterar aba **'config.yml'** e definir as configurações do bot. No exemplo que estamos deixando, usaremos a biblioteca do spaCy, então lembre-se de instalá-la também, junto com o **'model: pt_core_news_lg'**.

```
# Configuration for Rasa NLU.
# https://rasa.com/docs/rasa/nlu/components/
language: pt

pipeline:
# setup básico de configuração do rasa rodando com o Spacy.
- name: SpacyNLP
  model: pt_core_news_lg
- name: SpacyTokenizer
- name: SpacyFeaturizer
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100
  constrain_similarities: True
  model_confidence: linear_norm
- name: EntitySynonymMapper
- name: FallbackClassifier
  threshold: 0.7
  ambiguity_threshold: 0.1

# Configuration for Rasa Core.
```

Agora, acesse a pasta **'data'** e o arquivo **'nlu.yml'**. Nesse arquivo, colocaremos exemplos de possíveis inputs de usuários ao Bot. Essa etapa pode variar conforme o tema do bot e sua demanda. Utilizamos novamente as respostas concedida na planilha pela secretaria da fatec no anexo.

```

1 version: "2.0"
2 #essas são as possíveis perguntas ou interações do usuário ao bot
3 nlu:
4 - intent: greet
5   examples: |
6     - Oi
7     - Olá
8     - Eae
9     - tudo bem?
10    - bom dia
11    - boa tarde
12    - boa noite
13    - Fala mano
14
15 - intent: goodbye
16   examples: |
17     - té+
18     - valeu, até mais!
19     - tamo junto!
20     - flw ;)
21     - Obg, té+
22     - Tchau, obrigado!
23     - bye bye
24
25
26 - intent: affirm
27   examples: |
28     - sim
29     - isso
30     - sim
31     - ok
32     - beleza
33     - blz
34
35 - intent: deny
36   examples: |
37     - não
38     - n
39     - nem ferrando
40     - acho que não
41     - assim não

```

Caso seu bot tenha que executar algumas ações, devemos então acessar o arquivo **'actions.py'**, dentro da pasta **'actions'**. Basta descomentar o código exemplo, conforme o modelo abaixo. Feito isso, acesse o arquivo **'endpoints.yml'** e retire o comentário do **'action_endpoint:'**, também conforme a imagem abaixo.

```

1 from typing import Any, Text, Dict, List
2
3 from rasa_sdk import Action, Tracker
4 from rasa_sdk.executor import CollectingDispatcher
5
6
7 class ActionHelloWorld(Action):
8
9     def name(self) -> Text:
10         return "action_hello_world"
11
12     def run(self, dispatcher: CollectingDispatcher,
13            tracker: Tracker,
14            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
15
16         dispatcher.utter_message(text="Hello World!")
17         print(["funcionou!!!"])
18         return []
19

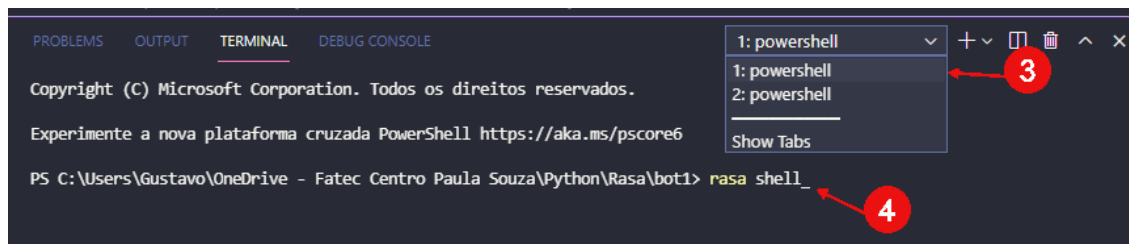
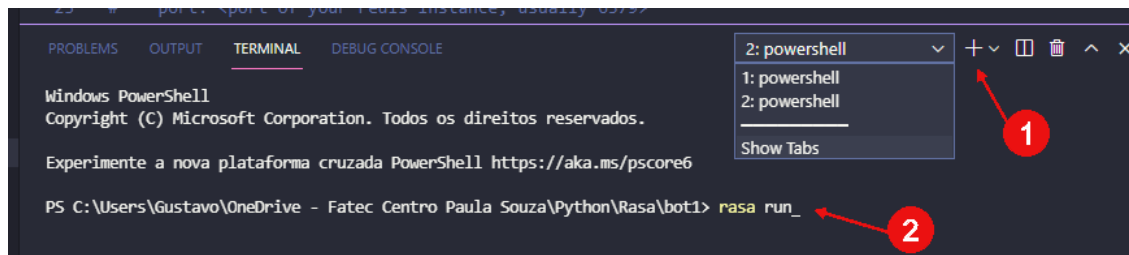
```

```

12
13 action_endpoint:
14     url: "http://localhost:5055/webhook"
15

```

Vamos agora começar a conversar com o bot, abra um novo batch, e inicie o servidor do rasa com o comando **'rasa run'**. Retorne agora ao bash 1, e rode o comando **'rasa shell'**.



Agora setup básico de um bot em português foi finalizado. Para mais informações é só acessar o site <https://rasa.com/docs/rasa/installation/>

4.3 – Rasa FAQs

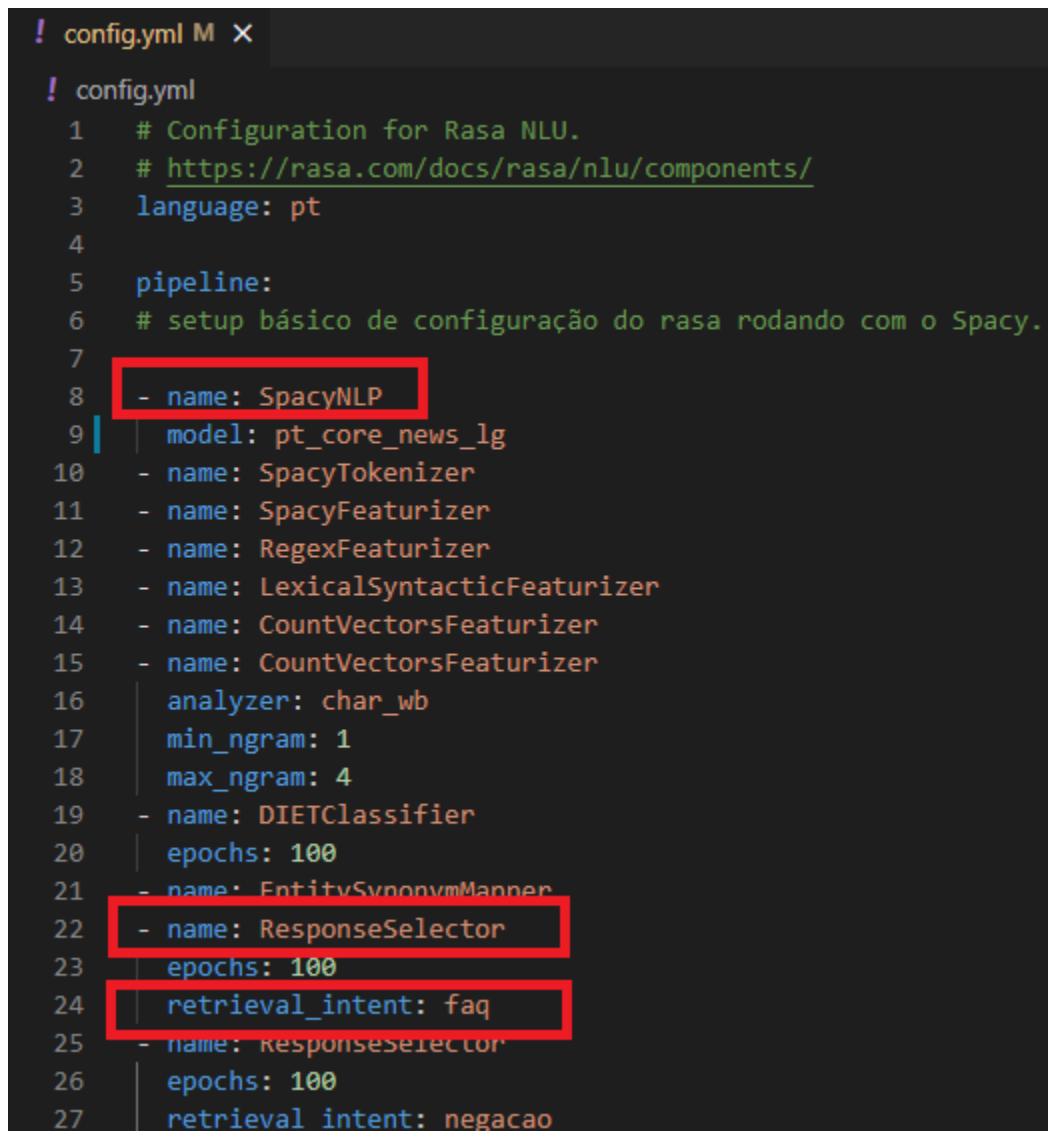
Assistentes de FAQ são os assistentes mais simples de construir e, normalmente, o primeiro tipo de assistente que alguém constrói.

FAQs e bate-papo são dois casos em que o assistente de conversação responde com um conjunto fixo de mensagens, e o assistente deve sempre responder da mesma forma, não importa o que tenha acontecido anteriormente na conversa. Por exemplo, na conversa a seguir, todas as perguntas podem ser

feitas em qualquer ponto da conversa, com a resposta sendo independente de qualquer coisa que o usuário tenha dito anteriormente. Para mais informações é só acessar o site <https://rasa.com/docs/rasa/chitchat-faqs>.

Para o nosso projeto utilizamos o FAQs, abaixo será mostrado o que foram alterados para que atendesse as necessidades da secretaria da FATEC.

Foi utilizado na aba config.yml as bibliotecas do Spacy, adicione RulePolicy às suas políticas e ResponseSelector, e retrieval_intent: faq ao seu pipeline em config.yml, conforme segue imagem a seguir:



```
! config.yml M X
! config.yml
1 # Configuration for Rasa NLU.
2 # https://rasa.com/docs/rasa/nlu/components/
3 language: pt
4
5 pipeline:
6 # setup básico de configuração do rasa rodando com o Spacy.
7
8 - name: SpacyNLP
9   model: pt_core_news_lg
10 - name: SpacyTokenizer
11 - name: SpacyFeaturizer
12 - name: RegexFeaturizer
13 - name: LexicalSyntacticFeaturizer
14 - name: CountVectorsFeaturizer
15 - name: CountVectorsFeaturizer
16   analyzer: char_wb
17   min_ngram: 1
18   max_ngram: 4
19 - name: DIETClassifier
20   epochs: 100
21 - name: EntitySynonymMapper
22 - name: ResponseSelector
23   epochs: 100
24   retrieval_intent: faq
25 - name: ResponseSelector
26   epochs: 100
27   retrieval_intent: negacao
```

```

33
34 # Configuration for Rasa Core.
35 # https://rasa.com/docs/rasa/core/policies/
36 policies:
37   - name: RulePolicy
38   - name: MemoizationPolicy
39   - name: TEDPolicy
40     max_history: 5
41     epochs: 100
42     constrain_similarities: True
43     model_confidence: linear_norm
44

```

Nas rules adicione pelo menos uma regra para responder a perguntas frequentes / bate-papo, foi criado uma intent: `faq` e a action: `utter_faq`, muito importante seguir esse padrão para que o bot não se perca na hora da execução.

```

19 - rule: respond to FAQs
20   steps:
21     - intent: faq
22     - action: utter_faq
23

```

Na aba NLU deve-se colocar a intent: `faq/nome_da_variável`, que deverá ser a mesma no `domain.yml`, para que o bot prossiga com a execução sem falhas, porque cada NLU ele busca a sua utter corresponde no `domain.yml`. Conforme é mostrado nas imagens a seguir.


```
! nlu.yml ×
data > ! nlu.yml
1 version: "2.0"
2 nlu:
3   - intent: saudacao
4     examples: |
5       - Oi
6       - Olá
7       - olá
8       - ola
9       - Eae
10      - tudo bem?
11      - bom dia
12      - boa tarde
13      - boa noite
14      - Fala mano
15      - fala ae
16      - oi
```

```
! domain.yml ×
! domain.yml
1 version: '2.0'
2 session_config:
3   session_expiration_time: 60
4   carry_over_slots_to_new_session: true
5 intents:
6   - negacao:
7     | is_retrieval_intent: true
8   - faq:
9     | is_retrieval_intent: true
10  - saudacao
11  - obrigado
12  - tchau
13  - bot
14 responses:
15   utter_saudacao:
16     - text: Bem vindo ao Sistema FATEC MM
```

4.4. – WebChat

Para que seja possível hospedar e conversar com o bot no website, precisa ser feito um webChat. O WeChat é um pequeno campo desenvolvido em HTML5, CSS3 e Java Script, que o usuário irá clicar e abrir o chat para iniciar uma conversa com o bot.

Para integrar o bot com o webchat, é necessário tirar o comentário do campo socketio no arquivo credential.yml, e chamar os eventos: user_uttered para o campo user_message_evt, e bot_uttered para o campo bot_message_evt. No campo session_persistence, deve colocar o valor true.



4.5. – Manutenção do Bot

Para a realização de manutenção e alterações futuras o processo transcorre de maneira simples. Caso seja alguma informação da FATEC que altere, como por exemplos alteração de horários ou inserção de novas respostas para o usuário, o desenvolvedor responsável deverá alterar os seguintes campos:

```

! domain.yml X
! domain.yml
14 responses:
15   utter_saudacao:
16     - text: Bem vindo ao Sistema FATEC MM
17   utter_pergunta:
18     - text: Em que posso te ajudar, digite o assunto desejado?
19   utter_desculpa:
20     - text: Desculpa eu não compreendo a sua pergunta, poderia informa novamente?
21   utter_tchau:
22     - text: Tchau até mas!
23     - text: Tchau até logo!
24     - text: Tchau fico feliz por ter usado nosso bot!
25   utter_bot:
26     - text: Olá eu sou um bot criado pela FATEC!
27   utter_faq/telefone:
28     - text: 'Telefone: (19) 3806-2181 e 3806-3139'
29   utter_faq/endereco:
30     - text: Rua Ariovaldo Silveira Franco, 567 - Jd. 31 de Março - CEP 13801-005 - Mogi Mirim(SP)
31   utter_faq/horario:
32     - text: 'Horário de funcionamento de seg. à sexta: das 8h às 11h / das 13h às 17h / e das 18h30 às 22h!'
33   utter_negacao/negandopergunta:
34     - text: Não entendi poderia me repetir novamente sua pergunta!
35   utter_negacao/teste:
36     - text: Não entendi poderia me repetir novamente sua pergunta!

```

Caso existam alguma manutenção a ser feita no aspecto de criar uma informação da FATEC, o desenvolvedor deverá alterar os seguintes campos:

```

55 - intent: faq/horario
56   examples: |
57     - Que horas a Fatec trabalha?
58     - Horas que abre?
59     - hora de funcionamento da secretaria?
60     - Qual é o horário de funcionamento?
61     - qual o horário de funcionamento da secretaria?
62     - Que horas a Fatec abre?
63     - A fatec está aberta amanhã?
64     - A fatec está aberta hoje?
65     - A fatec abre de sábado?

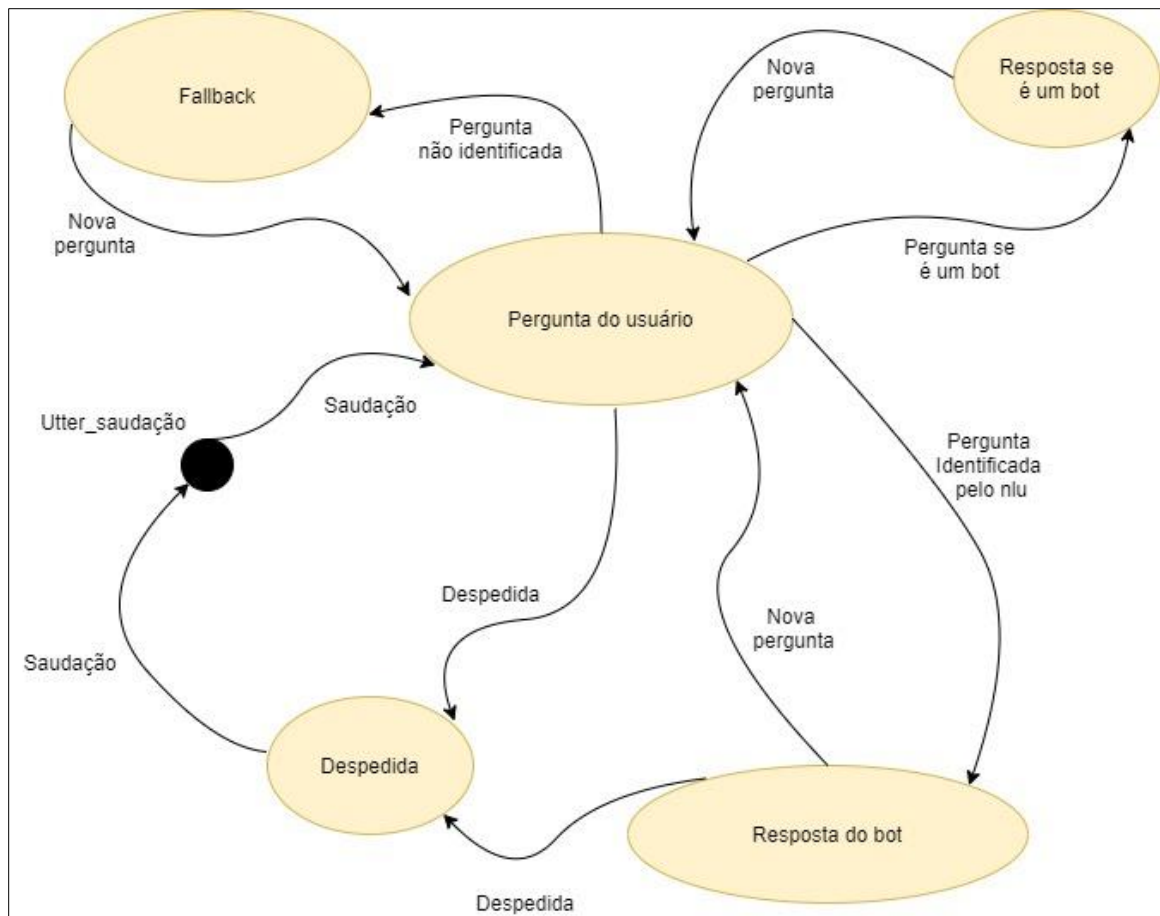
```

Inserir ou Alterar Frases no examples

5. O ESCOPO DO PRODUTO

Modelagem do projeto

5.1– DIAGRAMA



Essa imagem representa o fluxograma de interatividade entre o usuário e o bot.

5.2– GRAFICO TESTE em conclusão

6. CONCLUSÃO

Com a realização desse projeto, demos um início para que a FATEC de Mogi Mirim – SP tenha um chatbot funcional e que, as demandas de perguntas feitas pelo telefone e e-mail sejam otimizadas diminuindo assim o fluxo de retrabalho dos funcionários da secretaria. Com base em informações coletadas pelos funcionários da secretaria a uma expectativa de redução atendimentos de telefones e e-mail de 50% e de atendimento via WhatsApp 20%. Dessa forma concluímos que nosso projeto atendera o objetivo proposto e será de grande avalia sua utilização.