# ChatBots

## *DESENVOLVIMENTO ORIENTADO A CONVERSAS*

*Versão Extendida - Com LiveCode - SEC 2019*

# Otacilio Maia


https://otaciliomaia.com

https://www.linkedin.com/in/otacilio/

$ npm i -g otaciliomaia

# Tipos de ChatBots

## Regras

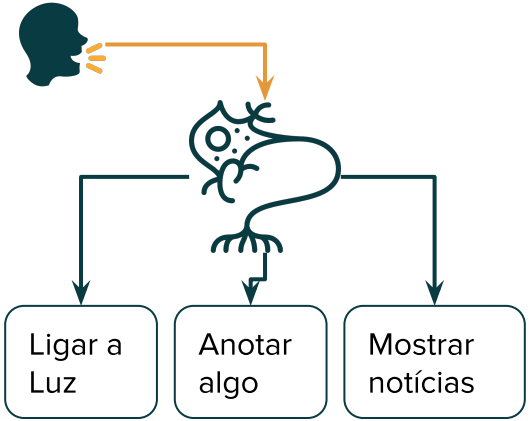
```
if(user.say("Ligue a luz")){
        light.turnOn();
}
```

Acenda a lâmpada

```
if(user.say.contains("Ligue" or
"Acenda") and
user.say.contains("Luz" or
"Lâmpada") ){
        light.turnOn();
}
```

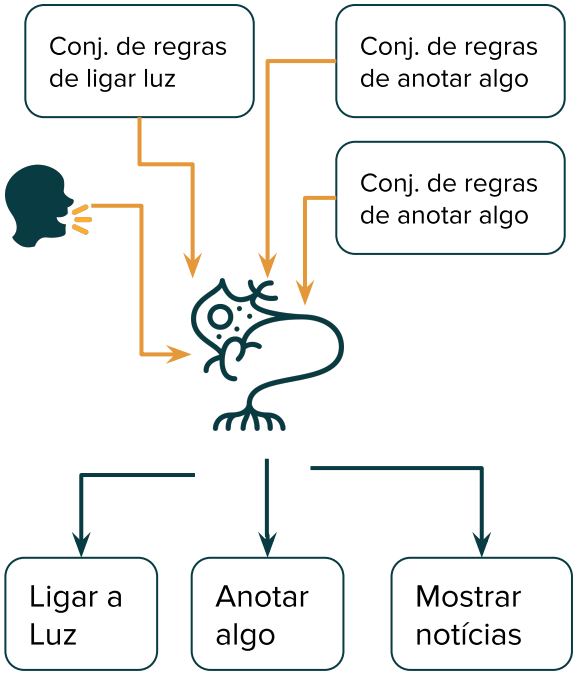
Ligue minha tv no filme Luz, Câmera e Ação

## Inteligência Artificial




Tive uma ideia!

## Híbrido



**rules.py exemplo de ChatBot baseado em regras**

```
def bot_answer(user_text):
    if(user_text == "Ligue a luz"):
        print("Bot: Luz ligada com sucesso")
    elif(("Ligue" in user_text or "Acenda" in user_text) and ("luz" in user_text or "lampada" in user_text)):
        print("Bot: Luz ligada com sucesso")
    elif (user_text == "sair"):
        return
    else:
        print("Bot: Nao entendi o que voce falou")


def main():
    user_text = ""
    while(user_text != "sair"):
        user_text = input("Voce: ")
        bot_answer(user_text)


if __name__ == "__main__":
    main()
```

**chatterExample.py exemplo de ChatBot baseado em IA**

```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer

conversation = [
    "Oi",
    "Oi, tudo bom?",
    "Qual a primeira palestra do dia 01?",
    "A primeira palestra será, ChatBots:
Desenvolvimento Orientado a Conversas",
    "E onde vai ser o evento?",
    "Vai ser na Poli-UPE, que fica na benfica
em frente ao internacional",
    "A que horas começa?",
    "Às 8:00 da manhã",
    "Obrigado!",
    "De nada."
]
```

```
def main():
    chatbot = ChatBot('ChatBot SEC')
    trainer = ListTrainer(chatbot)
    trainer.train(conversation)

    user_text = ""
    while(user_text != "sair"):
        user_text = input("Voce: ")
        response =
chatbot.get_response(user_text).text.encode('u
tf-8')
        print(response.decode())

if __name__ == "__main__":
    main()
```

chatterExample.py exemplo de ChatBot baseado em IA

```python
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer


def main():
    chatbot = ChatBot('Chat AI')
    trainer = ChatterBotCorpusTrainer(chatbot)
    trainer.train("chatterbot.corpus.portuguese")

    user_text = ""
    while(user_text != "sair"):
        user_text = input("Voce: ")
        response = chatbot.get_response(user_text).text.encode('utf-8')
        print(response.decode())


if __name__ == "__main__":
    main()
```

# NLP

*Natural Language Processing*

*O SEC é um evento arretado.*

*O SEC é um evento arretado.*

O - Artigo definido.

SEC - substantivo próprio simples.

é - verbo na terceira pessoa do singular, no presente do indicativo.

um - artigo indefinido.

evento - substantivo comum primitivo simples.

arretado - adjetivo

*Encontre os sabores de tapioca na frases:*

**Quero uma tapioca de coco e queijo.**

**Me vê uma tapioca de frango com queijo, por favor!**

**Têm tapioca de cartola? Quero uma!**

**De charque tem? Se tiver me vê duas tapiocas.**

*Encontre os sabores de tapioca na frases:*

**Quero uma tapioca de coco e queijo.**

**Me vê uma tapioca de frango com queijo, por favor!**

**Têm tapioca de cartola? Quero uma!**

**De charque tem? Se tiver me vê duas tapiocas.**

*Encontre os sabores de tapioca na frases:*

**Quero uma tapioca de coco e queijo.**

**Me vê uma tapioca de frango com queijo, por favor!**

**Têm tapioca de cartola? Quero uma!**

**De charque tem? Se tiver me vê duas tapiocas.**

*Encontre os sabores de tapioca na frases:*

**Quero uma tapioca de coco e queijo.**

**Me vê uma tapioca de frango com queijo, por favor!**

**Têm tapioca de cartola? Quero uma!**

**De charque tem? Se tiver me vê duas tapiocas.**

*Encontre os sabores de tapioca na frases:*

Quero uma tapioca de coco e queijo.

Me vê uma tapioca de frango com queijo, por favor!

Têm tapioca de cartola? Quero uma!

De charque tem? Se tiver me vê duas tapiocas.

Meu irmão, derrubaram minha tapioca, tô arretado!

Rapaz, tô na mágua, não vou pro carnaval esse ano.

Que arretado, vai ter show de Nação Zumbi no zero hoje!

# NLTK

*Python Natural Language Toolkit*

**O S.E.C é um evento arretado. Valeu pessoal!**

```
'O',
'S.E.C.',
 'é',
 'um',
 'evento',
 'arretado',
 '',
 'Valeu',
'pessoal',
 '!'
```

# E qual a diferença disso pra um simples split?

```
import nltk

sentence = "O S.E.C é um evento arretado. Valeu pessoal!"
splited = sentence.split()
tokens = nltk.word_tokenize(sentence)

print("\n\nA versão com split")
print(splited)

print("\n\nA versão com tokens")
print(tokens)
print("\n")
```

## NLTK - Tokenização - Experimento

```
otacilio@Wucheria ~/D/Chatbots-Desenvolvimento-Orientado-a-Conversas> python3 4_nltk_token.py


A versão com split
['O', 'S.E.C', 'é', 'um', 'evento', 'arretado.', 'Valeu', 'pessoal!']


A versão com tokens
['O', 'S.E.C', 'é', 'um', 'evento', 'arretado', '.', 'Valeu', 'pessoal', '!']


otacilio@Wucheria ~/D/Chatbots-Desenvolvimento-Orientado-a-Conversas>
```

# Spacy - Pos Tag

*Análise de sintaxe*

## Spacy - Post Tag

```
import spacy

nlp = spacy.load('pt')
my_doc = nlp('O S.E.C. é um
evento incrível!')

for token in my_doc:
    print(token.orth_, token.pos_)
```

```
# ADJ: adjective
# ADP: adposition
# ADV: adverb
# AUX: auxiliary
# CCONJ: coordinating conjunction
# DET: determiner
# INTJ: interjection
# NOUN: noun
# NUM: numeral
# PART: particle
# PRON: pronoun
# PROPN: proper noun
# PUNCT: punctuation
# SCONJ: subordinating conjunction
# SYM: symbol
# VERB: verb
# X: other
```

# Spacy - Ententies

*Localizando entidades*

```
import spacy

nlp = spacy.load('pt')
doc = nlp('Gildo lanches é o melhor lugar de Recife para
lanchar, fica perto da Poli')

for ent in doc.ents:
   print(ent.text, ent.label_)
```

# Intents

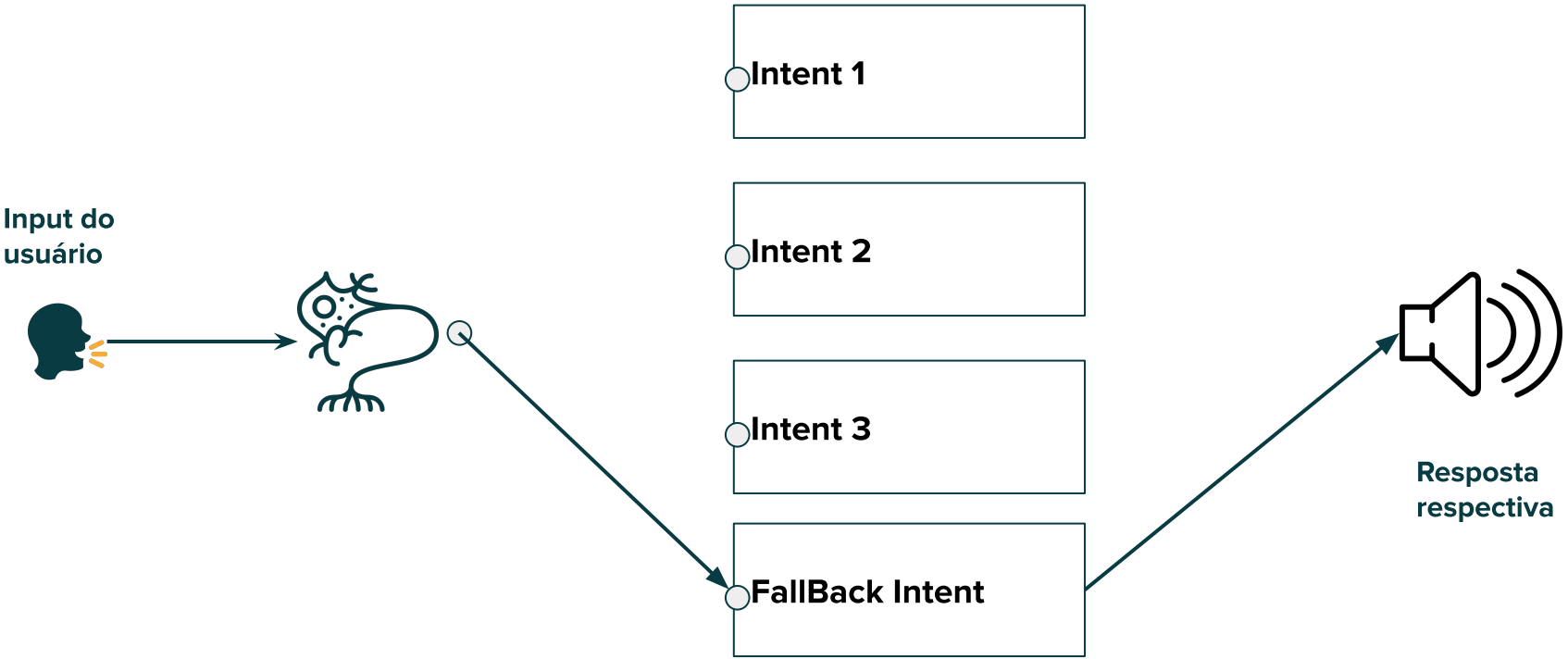
*De quantas formas um usuário pode pedir um lanche?*

**Quero um caldo de cana e duas coxinhas!**

**Me vê duas pipocas, uma doce e uma salgada.**

**Tem bolo de rolo? Quero duas fatias.**

**Por favor, eu gostaria de um pastel.**


Intent 1
Intent 2
Intent 3
FallBack Intent
Input do usuário
Resposta respectiva

# Perguntas?

- https://otaciliomaia.com/
- https://github.com/otacilion
- http://bit.ly/chatbots-sec
- https://github.com/OtacilioMaia/Chatbots-Desenvolvimento-Orientado-a-Conversas