

# TRABALHO CHATBOT

## 1. Objetivo

Criar um chatbot, com interação com o usuário, através das ferramentas ensinadas em sala de aula. Como desafio, capturar um valor numérico no contexto, e fazer uma consulta externa.

## 2. Descritivo do caso de uso deste chatbot

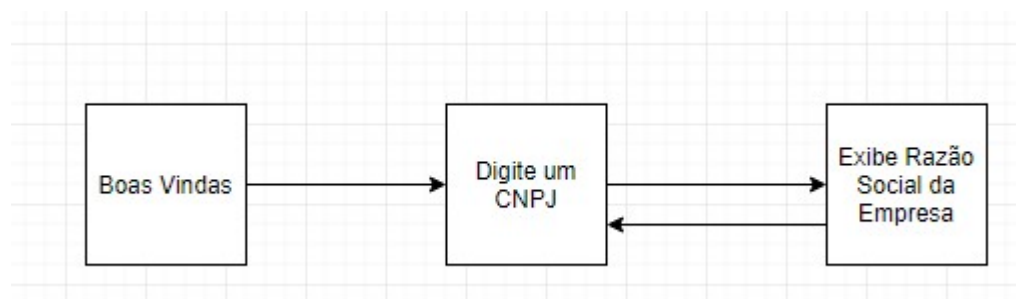
O objetivo dessa ferramenta é realizar pesquisas com base na receita federal. A princípio foi feito uma **PoC** de uma ferramenta de pesquisa por CNPJ, e retornado a razão social dessa empresa.

Para isso foi utilizado uma consulta à base da receita federal. Essa ferramenta pode evoluir e buscar outras informações como CNAE, quadro societário, entre outras.

Isso seria um ganho para consultas, em um backoffice para checagem de informações passadas pelos clientes por telefone. Através de uma ferramenta única, o atendente confronta as informações com a base da receita (sem **CAPTCHA**), e abre chamado etc, tudo dentro da mesma ferramenta.

## 3. Intenções

As intenções foram criadas para atender o seguinte fluxo:



Search intents	🔍
• br.com.garanito.dialogflow.agent.actions.gettimeclock	
• br.com.garanito.infoempresas.intents.assistant.welcome	
🔖 br.com.garanito.infoempresas.intents.Fallback	
• br.com.garanito.infoempresas.intents.getcnpj	
• br.com.garanito.infoempresas.intents.goodbye	
• br.com.garanito.infoempresas.intents.welcome	

Utilizamos contexto, para ficar “preso” dentro desse fluxo, então o operador pode ficar consultando CNPJ até cancelar.

• br.com.garanito.infoempresas.intents.getcnpj

SAVE

## CONTEXTOS

**Contexts** ⓘ

await\_cnpj ⓘ Add input context

5 await\_cnpj ⓘ Add output context

## Events ⓘ

## INTENÇÕES

## Training phrases ⓘ

Search training phrases 🔍

” Add user expression

” O cnpj é 3470727

” O cnpj é 03 470 727 0004 73

” cnpj: 03 470 727 0004 73

” 1109184000493

” empresa 1109184000193

” Da minha empresa é 1109184000193

” 03470727000473

” 1109184000193

” certeza 1109184000193

” O cnpj é 01109184000195

1 OF 2

O intuito dessas intenções é fazer o chat aprender a reconhecer um CNPJ, dentro da frase digitada pelo usuário.

Outras intenções obrigatórias foram criadas como, a de boas, e de saída do chat, por exemplo.

## 4. Parâmetros

Aqui criamos uma entidade, para receber o CNPJ digitado pelo usuário, e trata-lo via **fulfillment**. Esse CNPJ é capturado, e processado pelo backend. Para captura-lo deve-se usar Regex.

digitado\_cnpj

SAVE

☒ Define synonyms

☒ Regexp entity

☐ Allow automated expansion

☐ Fuzzy matching

\d{1,2}\d{1,3}\d{1,3}\d{1,4}\d{1,2}

\d{1,14}

\d{1,14}

\d{1,14}

\d{1,14}

Enter value

+ Add a row

Fulfillment

Webhook

DISABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

Inline Editor (Powered by Cloud Functions for Firebase)

ENABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

index.jspackage.json

35var url = encodeURI('http://garanitobot.eastus.cloudapp.azure.com/receita/receita/details/'+vtext);

36return https.get(url)

37.catch( err =>

38{

39console.error(err);

40agent.add( 'Lamento, mas o sistema de consulta está fora do ar. Tente novamente mais tarde.' );

41}

42.then( body => {

43console.log(body);

44const a = JSON.parse(body);

45//var val = body.someParameter;

46//var msg = `The value is \${val}`;

47agent.add( `A resposta no meu catálogo online foi essa, ""+a.RAZAOSOCIAL+"". Obrigado por usar nossos serviços.`);

48});

49

50}

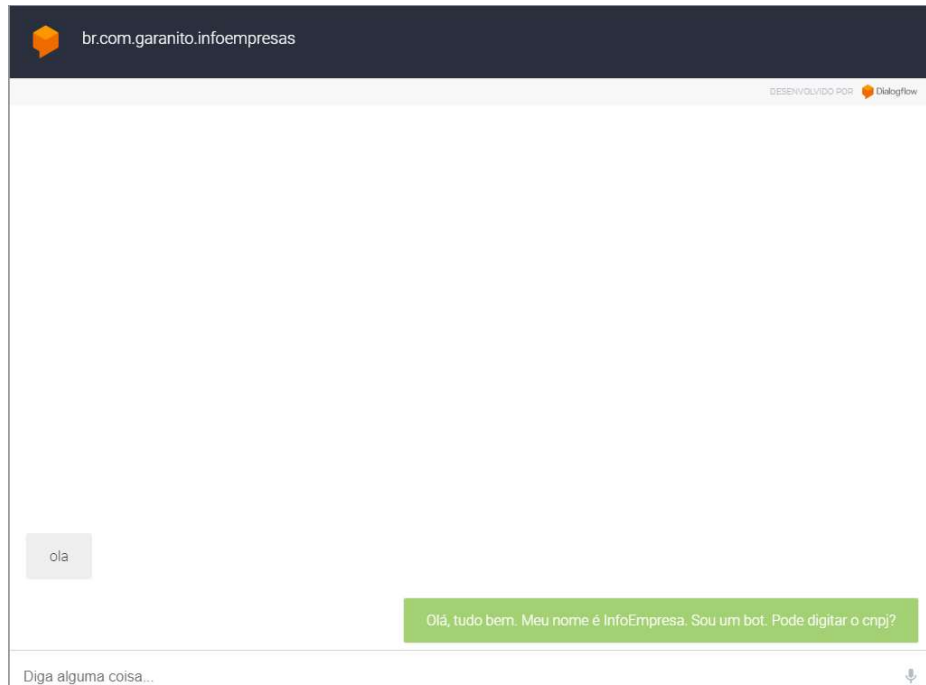
View execution logs in the Firebase console

Last deployed on 09/15/2019 22:38

DEPLOY

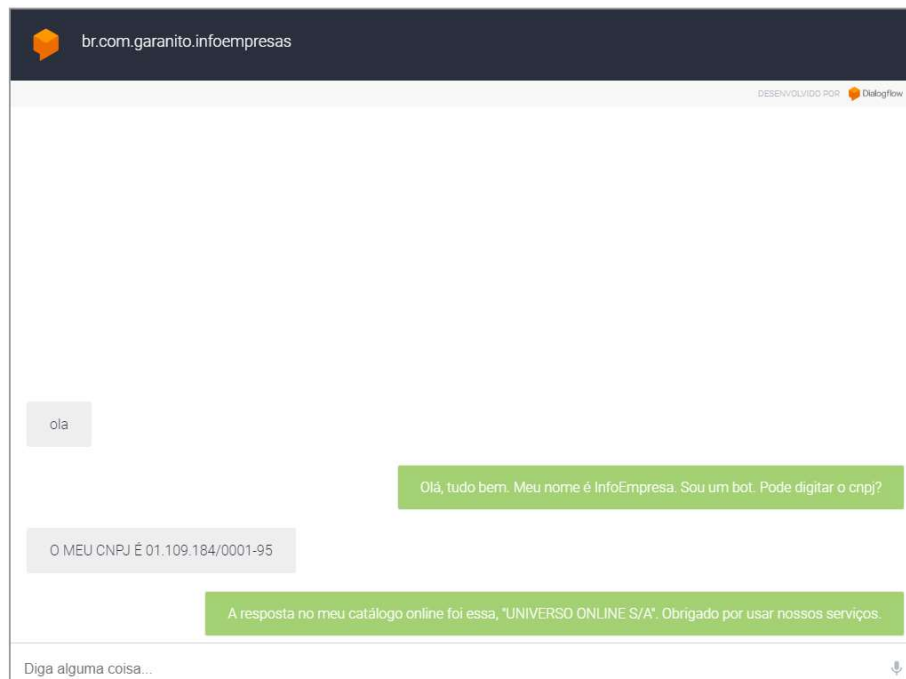
## 5. Testes

Para começar, é necessário dizer uma mensagem de boas-vindas, como “OLA”, por exemplo.

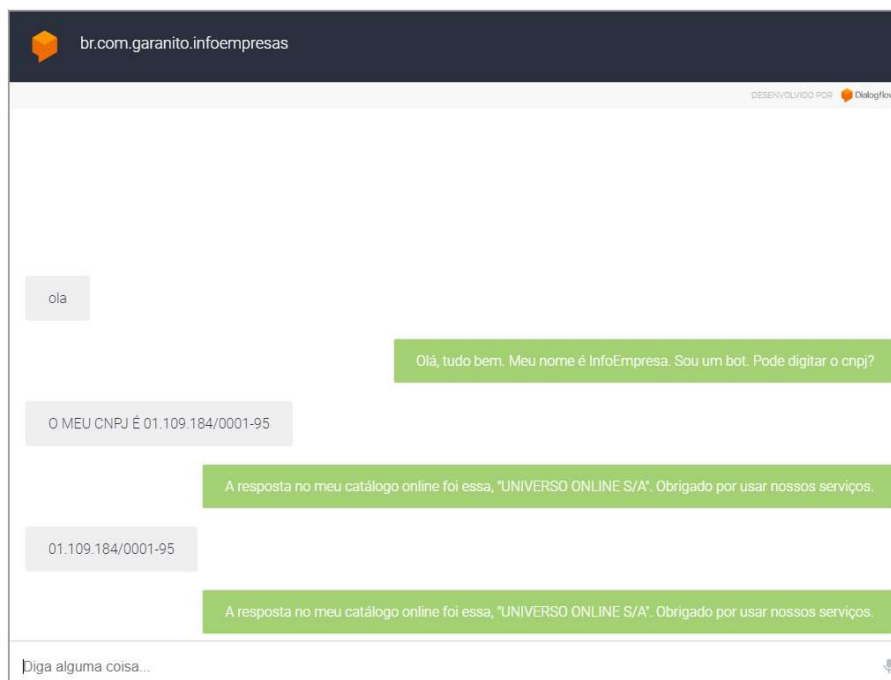


Na sequência deve-se digitar o CNPJ a ser procurado, no backend. Isso pode ser feito de algumas maneiras.

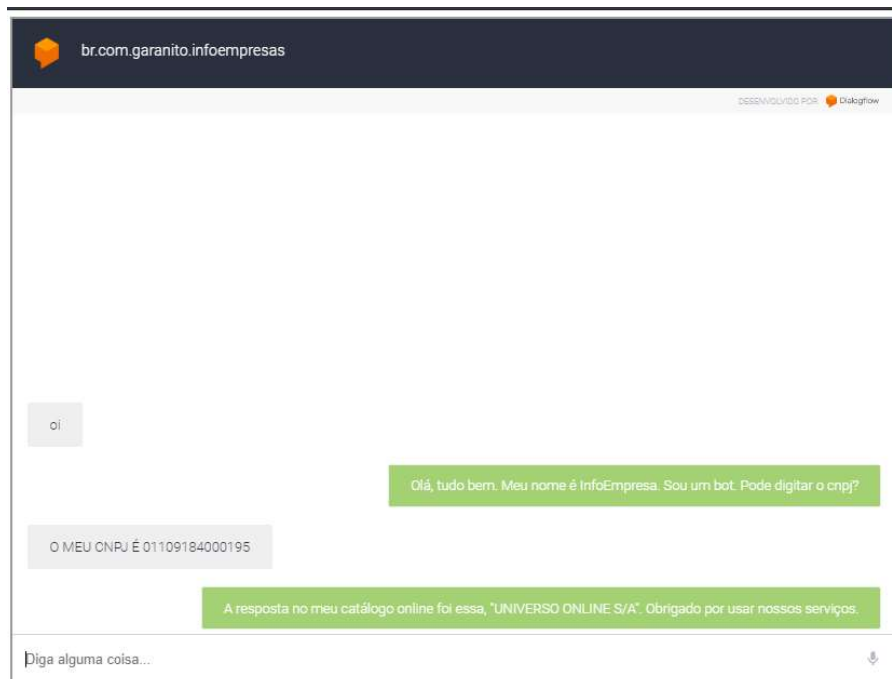
- **Com máscara de CNPJ: “O meu CNPJ é xx.xxx.xxx/xxxx-xx”**



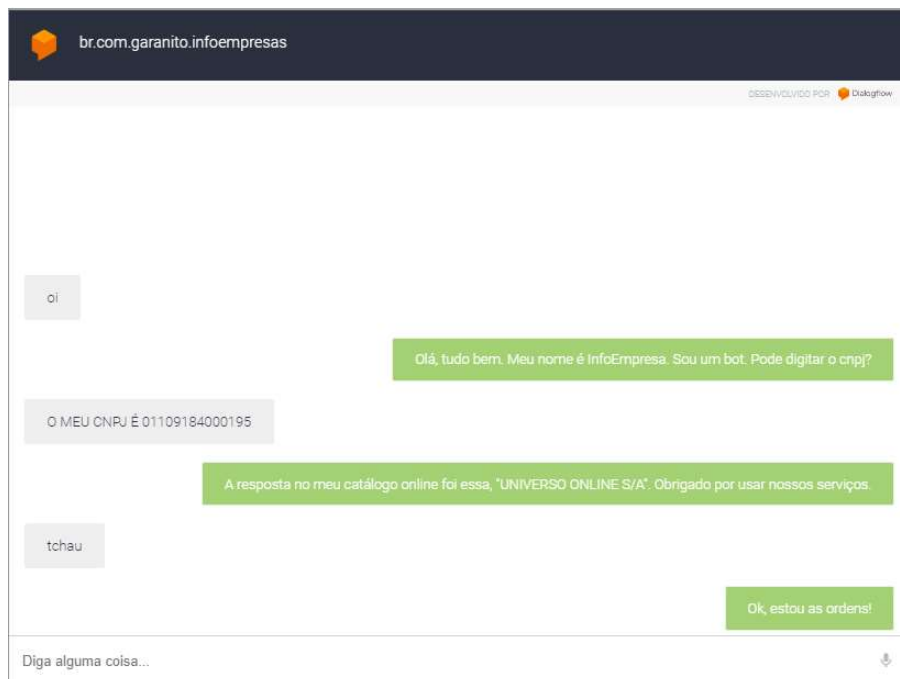
- **Só o CNPJ com máscara: “xx.xxx.xxx/xxxx-xx”**



- Um texto e um número sem máscara: “O MEU CNPJ É xxxxxxxxxxxxxxxx”



- Uma frase de despedida, cancela o bot, e perde o “contexto”: “Tchau”



## 6. Dados de acesso

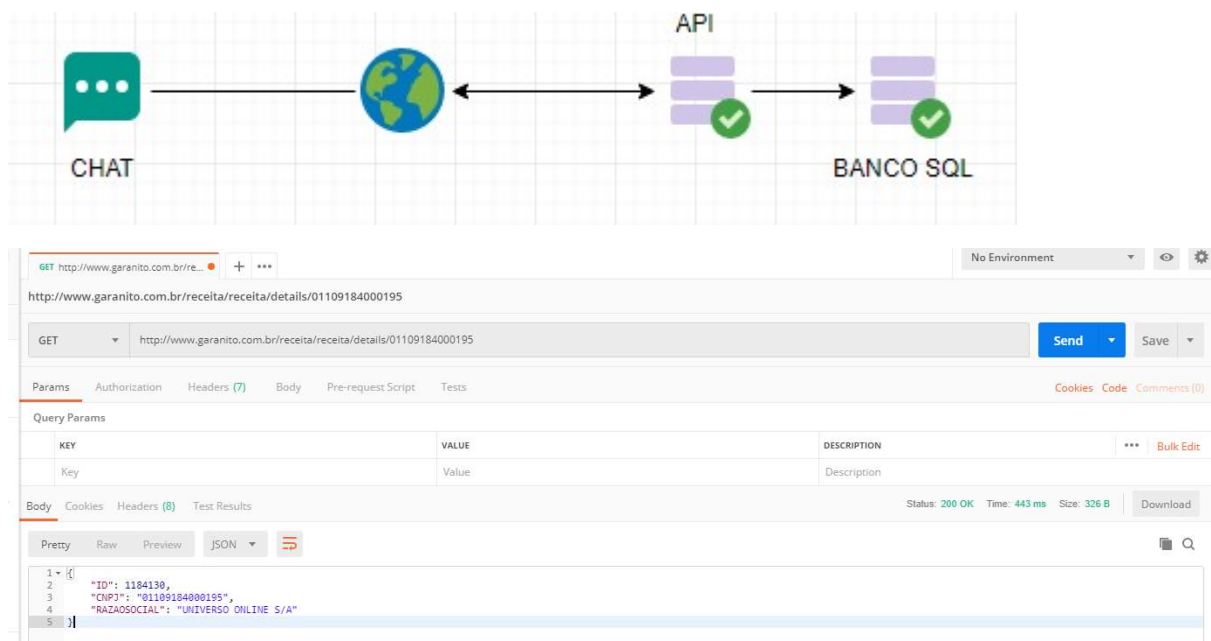
Para acesso no sistema:

• Primário: <a href="http://www.garanito.com.br">http://www.garanito.com.br</a> (DNS de apontamento para o servidor Azure)
• Secundário: <a href="http://garanitobot.eastus.cloudapp.azure.com/">http://garanitobot.eastus.cloudapp.azure.com/</a> (Acesso direto Azure)
• Terceiro: <a href="https://bot.dialogflow.com/1deef039-6611-4e89-84fd-991b948c02ec">https://bot.dialogflow.com/1deef039-6611-4e89-84fd-991b948c02ec</a> (Direto pela ferramenta DIGALOGFLOW)

Não possui autenticação do usuário.

## 7. Acesso a API Externa

A base da receita federal é disponibilizada, em arquivo. Esse arquivo foi carregado dentro de uma base de dados SQL Server, e uma API em C# foi construída para acesso à base.



O CHATBOT faz consultas via Fulfillment, como descreve o código abaixo.

```
// See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
// for Dialogflow fulfillment library docs, samples, and to report issues
'use strict';
```

```
const https = require('request-promise-native');
const functions = require('firebase-functions');
const {WebhookClient} = require('dialogflow-fulfillment');
const {Card, Suggestion} = require('dialogflow-fulfillment');
```

```

process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements

exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
  const agent = new WebhookClient({ request, response });
  console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
  console.log('Dialogflow Request body: ' + JSON.stringify(request.body));

  function welcome(agent) {
    agent.add('Welcome to my agent!');
  }

  async function httpGet(agent, text) {
    let vtext = '';
    vtext = text;
    console.log(vtext);
    vtext = vtext.replace(' ', '');
    console.log(vtext);
    vtext = vtext.replace(/\.\/g, '');
    console.log(vtext);
    vtext = vtext.replace(',', '');
    console.log(vtext);
    vtext = vtext.replace('-', '');
    console.log(vtext);
    vtext = vtext.replace('/', '');
    console.log(vtext);
    var url = encodeURI('http://garanitobot.eastus.cloudapp.azure.com/receita/receita/details/' +
vtext);
    return https.get(url)
      .catch(err => {
        console.error(err);

        agent.add('Lamento, mas o sistema de consulta está fora do ar. Tente novamente mais tarde.')
      });

    .then(body => {
      console.log(body);
      const a = JSON.parse(body);
      agent.add(`A resposta no meu catálogo online foi essa, "${a.RAZAOSOCIAL}". Obrigado por usar nos
sos serviços.`);
    });
  }

  async function getcnpj(agent) {
    var cnpj = agent.parameters.digitado_cnpj;
    if(cnpj){
    }else {
      const regex = /(?:\d{1,14})/g;
      const str = request.body.queryResult.queryText;
      let m;

      while ((m = regex.exec(str)) !== null) {
        if (m.index === regex.lastIndex) {
          regex.lastIndex++;
          m.forEach((match, groupIndex) => {
            console.log(`Found match, group ${groupIndex}: ${match}`);
            cnpj = match;
          });
        }
      }
    }
  }

```



```
    }  
  }  
  const response = await httpGet(agent,cnpj);  
}  
  
function fallback(agent) {  
  agent.add(`I didn't understand`);  
  agent.add(`I'm sorry, can you try again?`);  
}  
  
let intentMap = new Map();  
  
intentMap.set('br.com.garanito.infoempresas.intents.getcnpj', getcnpj);  
agent.handleRequest(intentMap);  
});
```