



PAFSE: Partnerships for Science Education

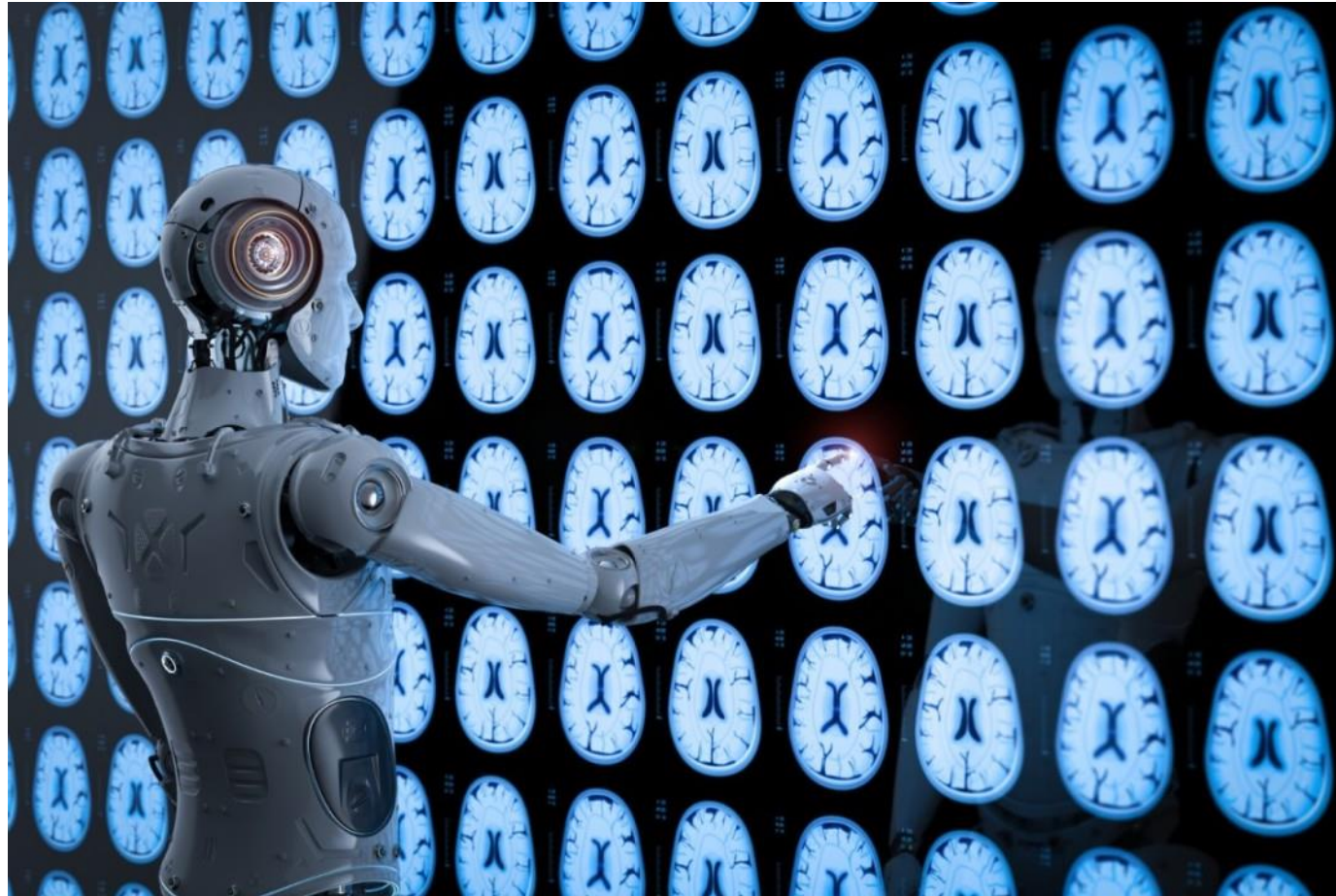
Project approved under Horizon 2020 Call: H2020-SwafS-2018-2020
(Science with and for Society)
Topic: Open schooling and collaboration on science education

Al nos cuidados de saúde

2022

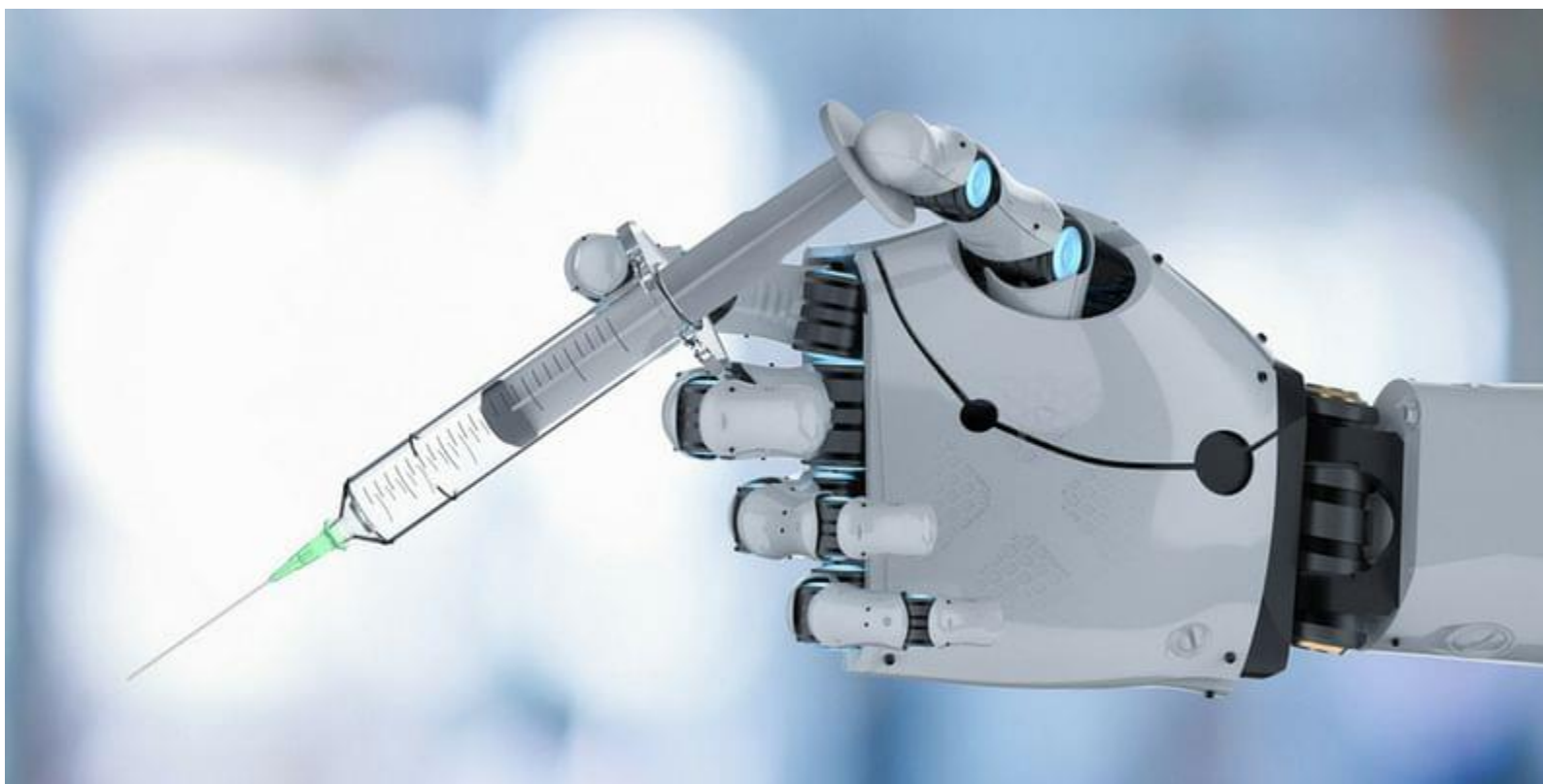


Diagnóstico



Fonte: <https://www.audiology-worldnews.com/research/3630-is-ai-as-good-as-the-professional-eye-in-diagnosing-from-medical-imaging>

Desenvolvimento de protocolo de tratamento



Fonte: <https://www.istockphoto.com/pt/foto/robotic-hand-holding-syringe-gm968084892-264002010>

Desenvolvimento de medicamentos e medicina personalizada



Fonte: <https://www.drugdiscoverytrends.com/understanding-gene-interactions-holds-key-to-personalized-medicine/>



Acompanhamento e cuidado do paciente



Fonte: <https://www.mozzaz.com/blog/ai-and-remote-patient-monitoring>

CONSORTIUM



Universidade NOVA de
Lisboa
(coordinator)



Universidade
do Minho



Instituto Superior de
Engenharia de Lisboa



Computer Technology
Institute and Press
"Diophantus"



University of
Ioannina



INESC TEC



Prevenção Rodoviária Portuguesa



University of Cyprus



Adam Mickiewicz
University



PAFSE: Partnerships for Science Education

Project approved under Horizon 2020 Call: H2020-SwafS-2018-2020
(Science with and for Society)
Topic: Open schooling and collaboration on science education

O que é a Inteligência Artificial?

2022

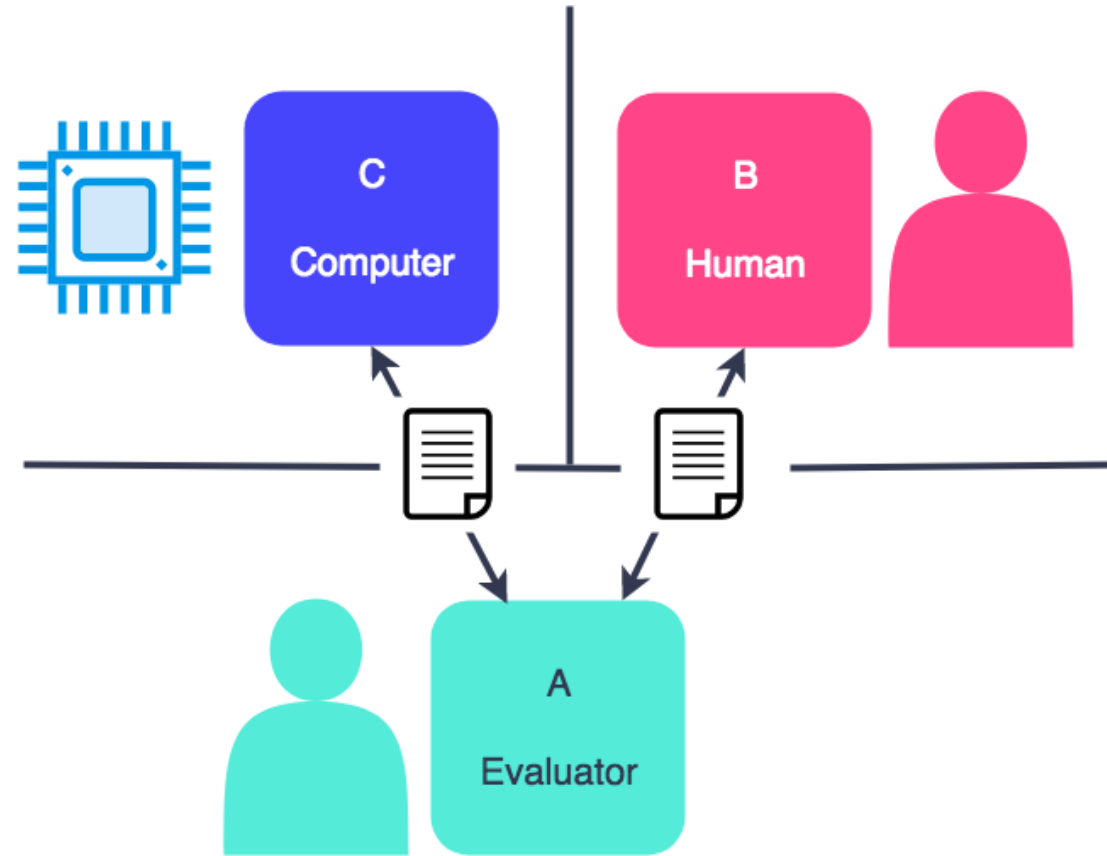
NOVA

IMS

Information
Management
School

NOVA

Teste de Turing



As máquinas podem pensar?



Fonte: Sagal, Paul T., *Mente Homem e Máquina*, Gradiva, 1996.

<https://www.researchgate.net/publication/322071516> Robotics and Artificial Intelligence

Alan Turing

- Alan Mathison Turing foi um matemático inglês, cientista da computação, lógico, criptoanalista, filósofo e biólogo teórico. Turing teve grande influência no desenvolvimento da ciência da computação teórica, propondo uma formalização dos conceitos de algoritmo e computação com a sua “máquina de Turing”, que é considerada um modelo do computador geral. Turing é apontado, quase unanimamente, como sendo o pai da ciência da computação teórica e da inteligência artificial.



Eliza

- É um famoso programa de computador de processamento de linguagem natural inicial criado de 1964 a 1966 no Laboratório de Inteligência Artificial do MIT por Joseph Weizenbaum.
- Weizenbaum, considerou o programa como sendo um método para mostrar a superficialidade da comunicação entre homem e máquina, mas ficou muito surpreso com o número de indivíduos que atribuíram sentimentos humanos ao programa de computador, incluindo a própria secretária de Weizenbaum.

```

Welcome to
          EEEEE LL   IIII  ZZZZZZ  AAAAA
          EE   LL   II    ZZ   AA  AA
          EEEEE LL   II    ZZ   AAAAAA
          EE   LL   II    ZZ   AA  AA
          EEEEE LLLLLL IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:  Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:  They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:  Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:  He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:  It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
  
```

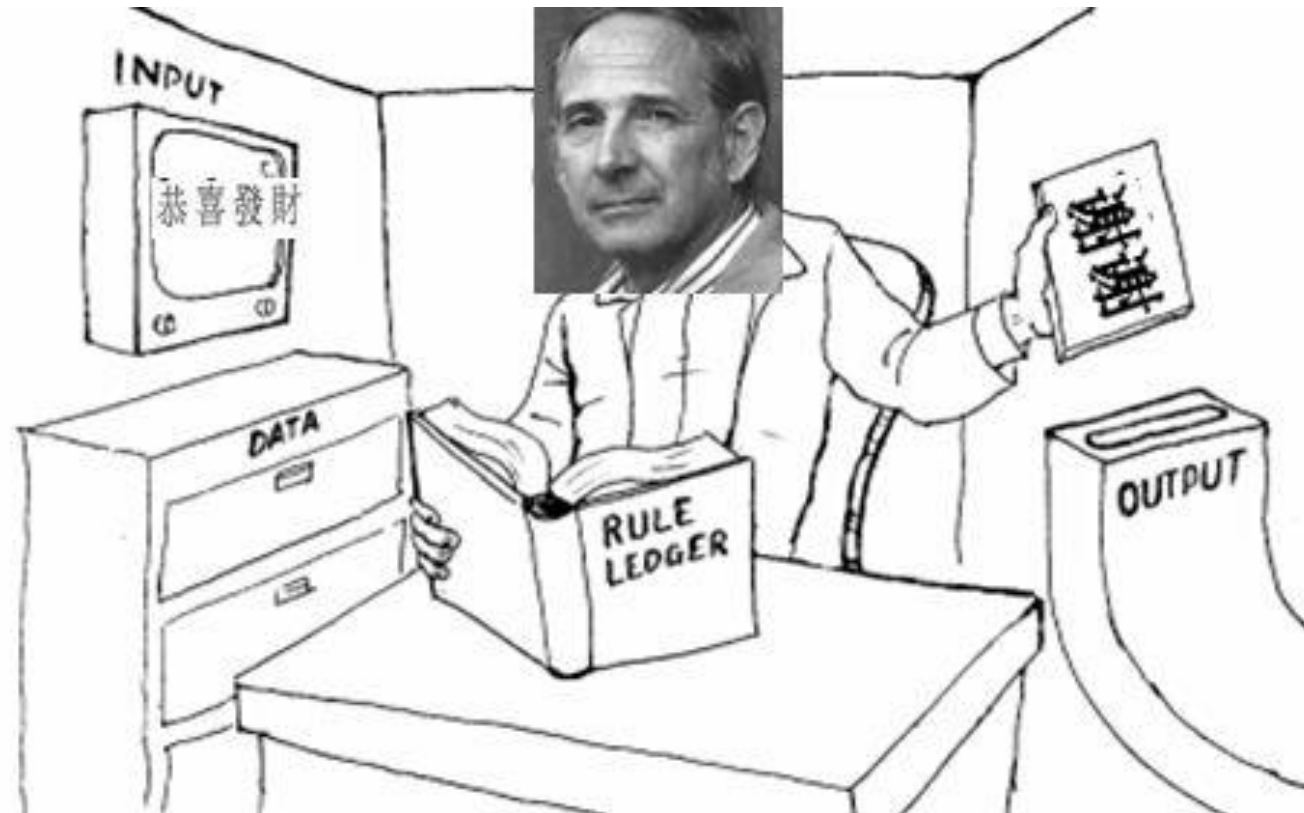


[Eliza, Computer Therapist \(fullerton.edu\)](http://fullerton.edu)

Sofia



Quarto Chinês- John Searle



Fonte: alexandria.nu

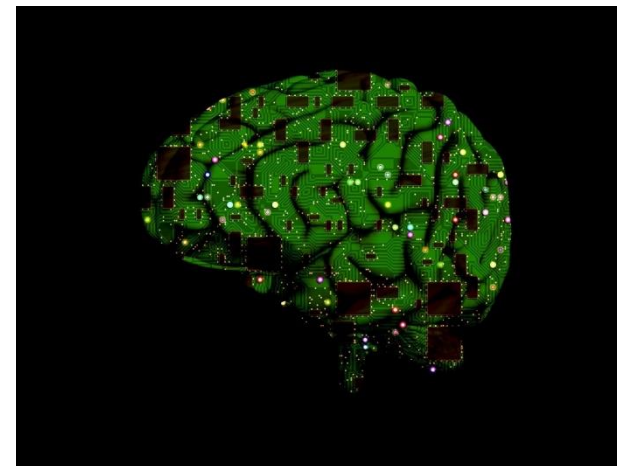
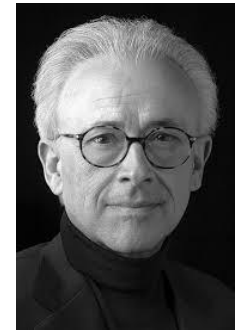
[Minds, Brains and Programs, published in Behavioural and Brain Sciences in 1980:
The paper which presents the Chinese Room Experiment](#)

[An Article on the book Singularity is Near by Ray Kurzweil: The book that brought the notion of Strong AI](#)

Antonio Damasio

- O neurocientista da USC Antonio Damásio argumenta que sentimentos e emoções são o que compõem a inteligência humana, a consciência e a capacidade de criação cultural. Um mapa da mente computacional, diz ele, não é o mesmo que ser humano.
- Assim, para Damásio, inteligência humana não pode ser transferida para máquinas

<https://ngp.usc.edu/usc-neuroscientist-antonio-damasio-argues-that-feelings-and-emotions-are-what-make-up-human-intelligence-consciousness-and-the-capacity-for-cultural-creation-a-map-of-the-computational-mind-he-says/>



CONSORTIUM



Universidade NOVA de Lisboa
(coordinator)



Universidade do Minho



Instituto Superior de Engenharia de Lisboa



Computer Technology Institute and Press "Diophantus"



University of Ioannina



INESC TEC



Prevenção Rodoviária Portuguesa



University of Cyprus



Adam Mickiewicz University



PAFSE: Partnerships for Science Education

Project approved under Horizon 2020 Call: H2020-SwafS-2018-2020
(Science with and for Society)
Topic: Open schooling and collaboration on science education

2022

Conceitos de interfaces de utilizador inteligentes

NOVA

IMS

Information
Management
School

NOVA

Interfaces de utilizador inteligentes

Conceito

Uma interface de utilizador inteligente é uma interface que envolve algum aspecto de inteligência artificial

- Concebidos como assistentes inteligentes para tarefas complexas
- Promover a adaptação do utilizador a sistemas interativos
- Incorporar paradigmas proativos e baseados em agentes para interação do utilizador
- Projetado como interfaces instruíveis (que aprendem)
- Apresentados como *front-ends* inteligentes para sistemas de software



Interfaces de utilizador inteligentes

Tradicionalmente, são agentes de software capazes de realizar tarefas ou serviços baseados em comandos ou perguntas, que podem ser fornecidos por texto ou voz
Exemplos: Siri, Alexa e Cortana.



<https://www.apple.com/br/siri/>

<https://developer.amazon.com/pt-BR/alexa>

<https://www.microsoft.com/en-us/cortana>

Interfaces de utilizador inteligentes

Características

- Toma decisões ideais com base nos *inputs* do utilizador
- Adapta-se às necessidades dos utilizadores
- Prevê a próxima solicitação ou ação do utilizador com base no estado percebido do utilizador e no estado atual do sistema
- Apresenta de forma proeminente as opções mais prováveis para o utilizador
- O sistema “sabe” o que fazer.
- Aprende com as sequências de ações do utilizador
- Aumenta a produtividade em relação às versões não inteligentes da interface.
- Aprende com o ambiente

Interfaces de utilizador inteligentes na saúde



Vantagens

- linhas de apoio humanas gratuitas para casos de diagnóstico médico
- atendimento rápido
- triagem e atuar como a primeira linha de esclarecimento
- poderosa ferramenta para combater a falta de informação e ajuda na gestão do pânico gerado pelas cadeias de partilha nas redes sociais
-

CONSORTIUM



Universidade NOVA de
Lisboa
(coordinator)



Universidade
do Minho



Instituto Superior de
Engenharia de Lisboa



Computer Technology
Institute and Press
"Diophantus"



University of
Ioannina



INESC TEC



Prevenção Rodoviária Portuguesa



University of Cyprus



Adam Mickiewicz
University

Implementação de um chatbot no Dialogflow a partir do zero

Manual ,passo a passo, para ser usado nas aulas

Índice:

1. O guia passo a passo para construir o chatbot
 - a. Configurar o Chatbot e criar o Agente (1ª aula)
 - b. Criar *intents* fixos (1ª aula)
 - c. Configurar a base de dados Airtable e gerar a chave API (2ª aula)
 - d. Criar *intents* dinâmicos (2ª aula)
 - e. Configurar o código do *Runkit* (2ª aulas)
 - f. Testar o Chatbot (3ª aula)
 - g. Inserir o Chatbot num site (3ª aula)
2. O artefato já construído “VirtualDoctor” que pode ser seguido como exemplo

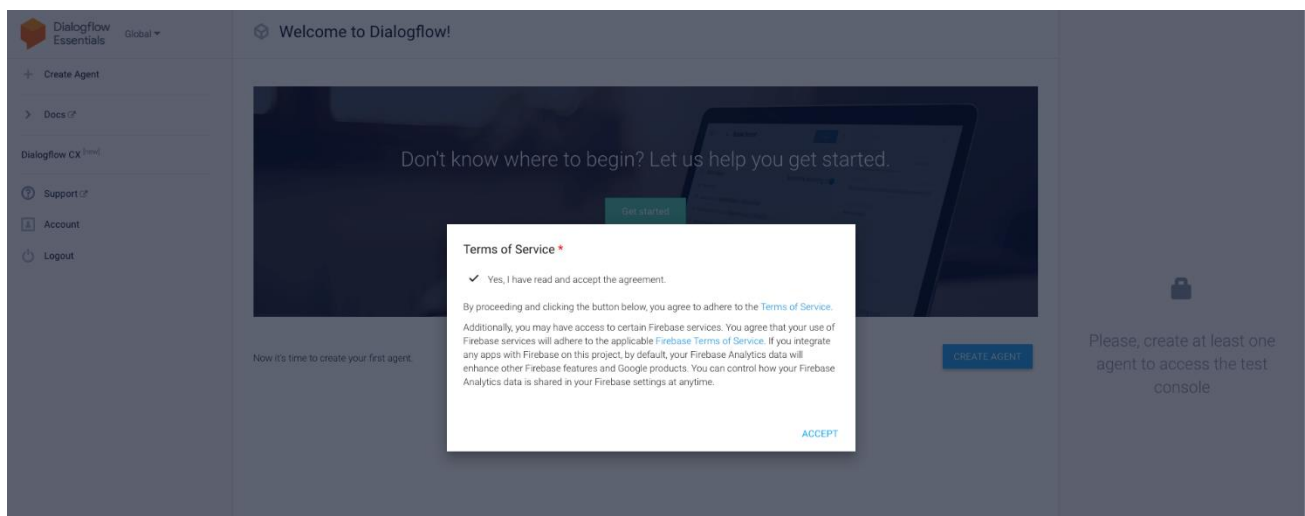
Pré-requisitos de desenvolvimento:

- Certifique-se que todos os computadores dos alunos têm **acesso à internet** e o navegador **Google Chrome** instalado;
- Certifique-se que os alunos têm um aplicativo de notebook (**notepad++** recomendado, mas outros funcionam)
- Certifique-se de que todos os alunos têm uma conta do **Gmail**, que será usada como conta para configurar o dialogflow, o airtable e as contas do runkit

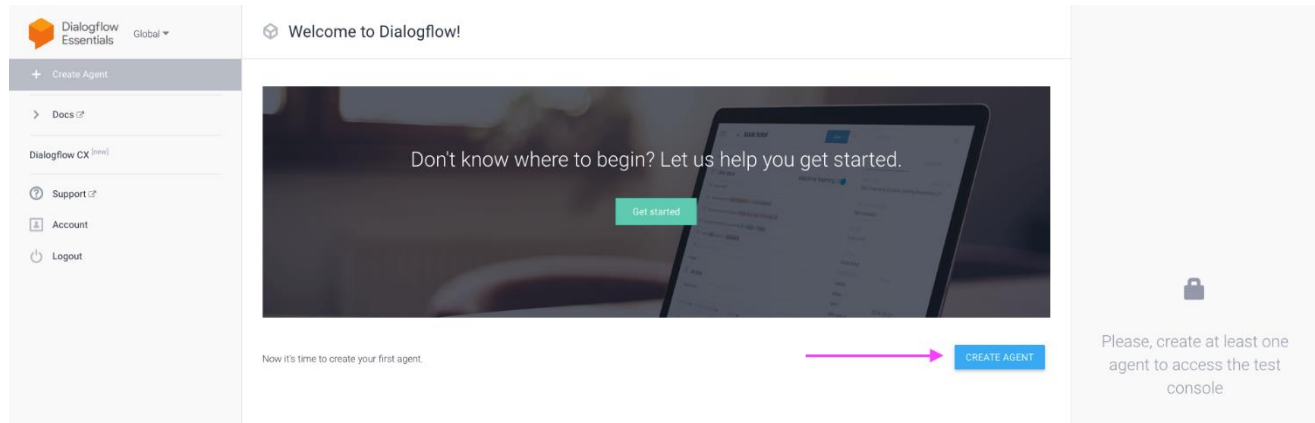
Guia passo a passo para construir um chatbot

a. Configurar o Chatbot e criar o Agente

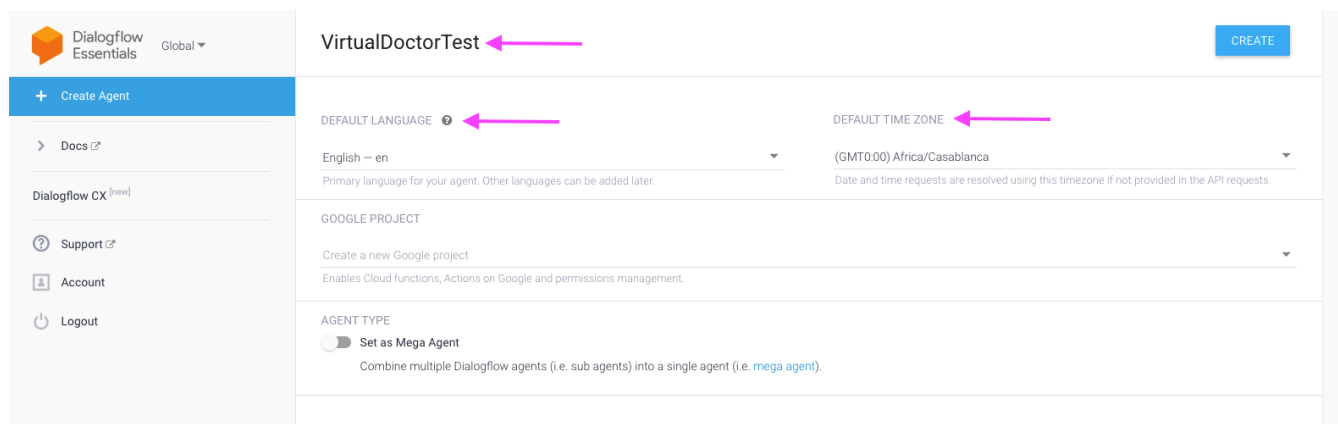
1. Faça login na sua conta **Gmail** (usando o navegador Google Chrome) e certifique-se que irá usar o mesmo navegador para configurar os aplicativos sem precisar digitar muitas vezes utilizador e senha;
2. Faça login no consola do Dialogflow e aceite os Termos de Serviço no seguinte link: <https://dialogflow.cloud.google.com/>



3. Clique no botão “+ **Create Agent**”. O agente é o primeiro componente a ser considerado ao se criar um chatbot no Dialogflow. É a essência do aplicativo chatbot, e é o objeto que irá interagir com os utilizadores via conversação.



4. Adicione o nome do agente (sem espaços), configure o idioma padrão para “**english-en**”, e o padrão Time Zone “**(GMT0:00 Africa/Casablanca)**” - posteriormente importante para garantir boas conexões da API e clique em “**Create**”



5. Agora que a estrutura básica do agente foi criada, precisamos adicionar algumas configurações adicionais. Clique no botão **Settings**.

Dialogflow Essentials Global

VirtualDoctorTest en

- Intents
- Entities
- Knowledge [beta]
- Fulfillment
- Integrations
- Training
- Validation

Intents

CREATE INTENT

Search intents

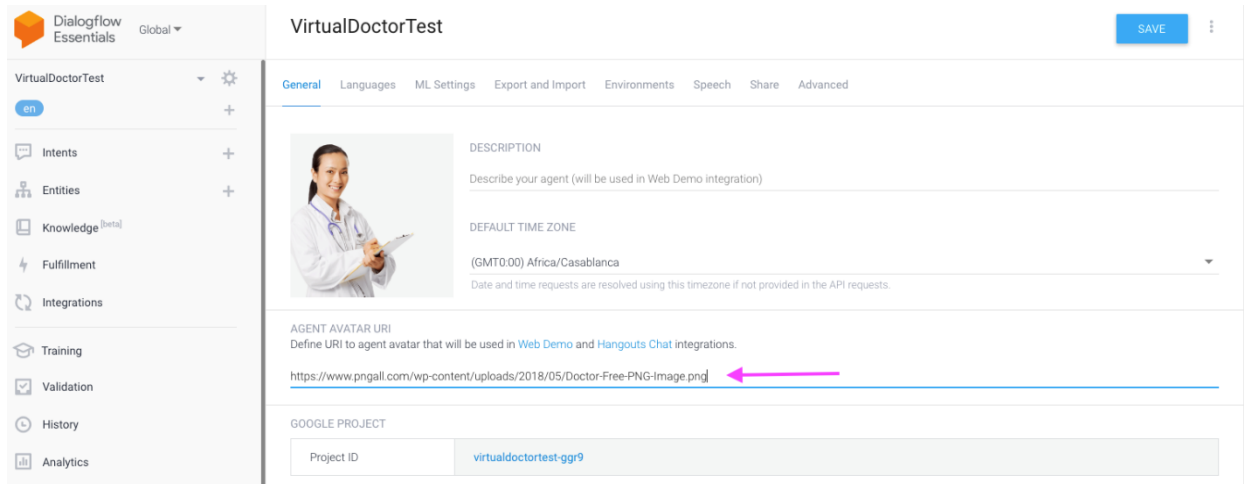
- Default Fallback Intent
- Default Welcome Intent

No regular intents yet. [Create the first one.](#)

Intents are mappings between a user's queries and actions fulfilled by your software. [Read more here.](#)

Before you start, check out [Prebuilt Agents](#), a collection of agents developed by the Dialogflow team.

6. Na página de configurações vamos agora adicionar uma imagem para que o chatbot tenha uma “cara” quando as pessoas o usarem. No campo “Agent Avatar URL” vamos adicionar uma imagem url (e.g. <https://www.pngall.com/wp-content/uploads/2018/05/Doctor-Free-PNG-Image.png>) e clicar **Save**:



Neste momento, temos já a estrutura básica do chatbot criada e já foram criados dois *intents standard* (* ver Nota). Um intent de resposta standard (**default fallback intent**) e um intent de boas-vindas standard (**default welcome intent**), que são os intents standard de uma conversa.

Para iniciar o processo de tornar o chatbot mais robusto, Podemos, agora, adicionar um conjunto de intents fixos (**fixed intents**), cuja execução funcionará como uma conversa básica do chatbot.

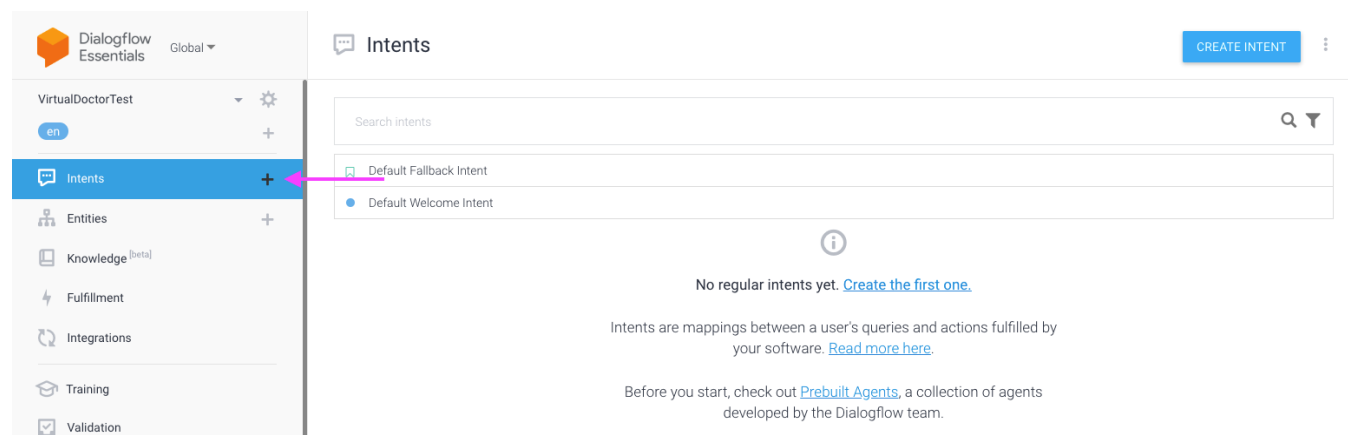
*Nota: Dentro de um chatbot, a intenção (intent) refere-se ao objetivo que o cliente tem em mente ao digitar uma pergunta ou comentário.

b. Criar Fixed Intents

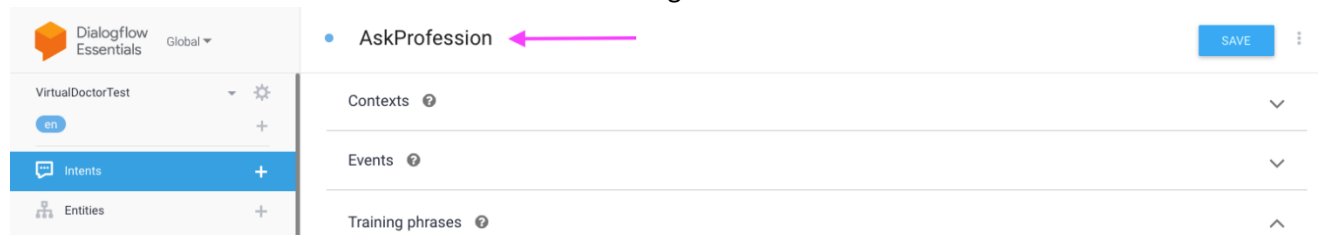
As intenções fixas (**Fixed Intents**) funcionarão como as perguntas e respostas fixas que o bot deve ser capaz de responder. Normalmente, coisas como “o que você faz”, “de onde você é”, são intenções fixas típicas que podem ser usadas sem nenhuma inteligência, apenas perguntas e respostas standard que são importantes para tornar o bot mais robusto. Para isso.

Criando uma nova intenção para "Ask Profession"

1. Clicar no botão “Intent +” no menu superior esquerdo do Dialogflow:



2. Adicione um nome à “AskProfession” ao novo Intent e grave-o:



- Agora clique na opção **“Add Training Phrases”**. Vamos adicionar um conjunto de perguntas de treino , que devem ser próximas do que o utilizador pergunta e que queremos que correspondam à intenção (não é necessário adicionar muito, pois o bot irá treinar-se e entender os sinónimos. Adicione alguns, tais, como apresentado na imagem abaixo e clique em **Save**:

VirtualDoctorTest Global

en

Intents

Entities

Knowledge ^{beta}

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Small Talk

AskProfession

SAVE

Contexts

Events

Training phrases

Search training phrases

⚠️ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression

” What is your current role

” What do you do for living?

” What do you do?

” Who are you?

- Depois das frases de treino serem adicionadas, adicione agora uma nova Resposta (**Response**). A resposta é o que o bot responderá quando a intenção for correspondida pela pergunta do utilizador. Adicione uma resposta como a abaixo e clique em **Save**:

VirtualDoctorTest Global

en

Intents

Entities

Knowledge ^{beta}

Fulfillment

Integrations

Training

Validation

History

Analytics

Prebuilt Agents

Small Talk

Docs

AskProfession

SAVE

” Add user expression

” What is your current role

” What do you do for living?

” What do you do?

” Who are you?

Action and parameters

Responses

DEFAULT

Text Response

1 I am Paula, a virtual doctor, that is here ready to help you when you have symptoms, any question related to Covid-19, or if you want to know anything about traveling to countries like Spain, Italy, Brazil, and others.

2 Enter a text response variant

5. A primeira intenção fixa agora é criada. Podemos usar o recurso de teste no canto superior direito para testar a intenção que acabamos de criar. Use a caixa **“Try it Now”** para inserir a primeira conversa. Tente adicionar algo semelhante à frase de treino, como **“Hi”**, e depois **“What do you do?”**. O resultado deve ser algo como:

The screenshot shows a 'Try it now' button at the top right. Below it, the 'Agent' section is displayed. The 'USER SAYS' field contains the text 'What do you do?' and is highlighted with a pink box and labeled '1'. To its right is a 'COPY CURL' link. Below the user input, the 'DEFAULT RESPONSE' field contains the text: 'I am Paula, a virtual doctor, that is here ready to help you when you have symptoms, any question related to Covid-19, or if you want to know anything about traveling to countries like Spain, Italy, Brazil, and others.' This field is also highlighted with a pink box and labeled '2'. Below the response, the 'INTENT' field shows 'AskProfession' and is highlighted with a pink box and labeled '3'. Below the intent, the 'ACTION' field shows 'Not available'. Below the action, the 'SENTIMENT' field shows 'Query Score: -0.1'. At the bottom, there is a 'DIAGNOSTIC INFO' button.

Podemos validar as seguintes configurações, conforme mostrado na imagem acima:

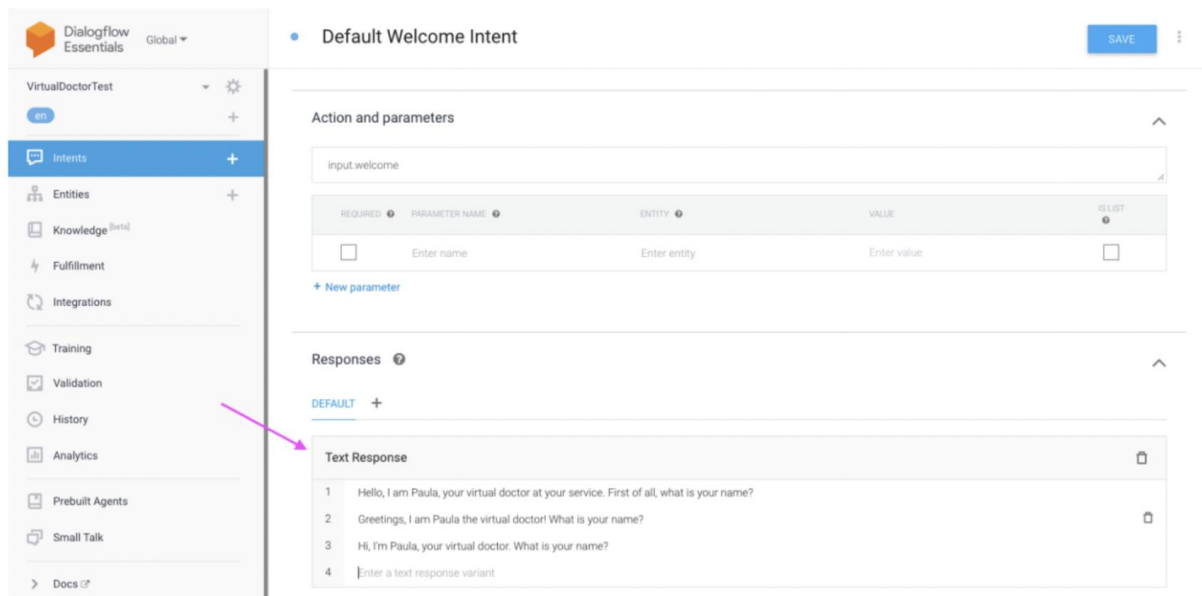
1. O último input do utilizador, questionando **“What do you do?”**
2. A resposta padrão que configuramos na etapa anterior;
3. A *intent* que correspondeu de acordo com a entrada do utilizador configurado

Passando para uma intenção ainda fixa, mas mais dinâmica. O objetivo agora será fazer com que o bot peça o nome do utilizador e armazene as informações para serem utilizadas na próxima resposta. Essa é uma estratégia comum para tornar o chatbot mais pessoal e, portanto, mais agradável de utilizar

- Volte para o ecrã inicial e clique em **“Default Welcome Intent”** para editá-lo. isto é o que usaremos para obter o nome, pois é a primeira interação com o utilizador

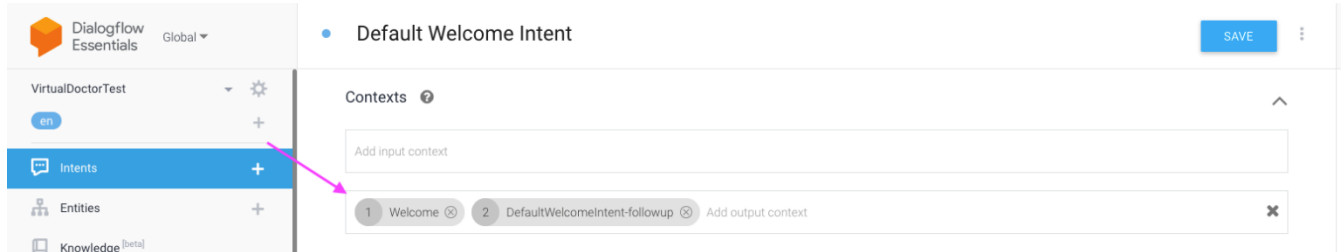


- Atualize as respostas predefinidas, para algo como a imagem abaixo. Lembre-se de adicionar algo para perguntar o nome, então a última frase deve ser algo como **“Qual é o seu nome?”** (**“What is your name?”**) Após editá-lo, clique em **Save**:

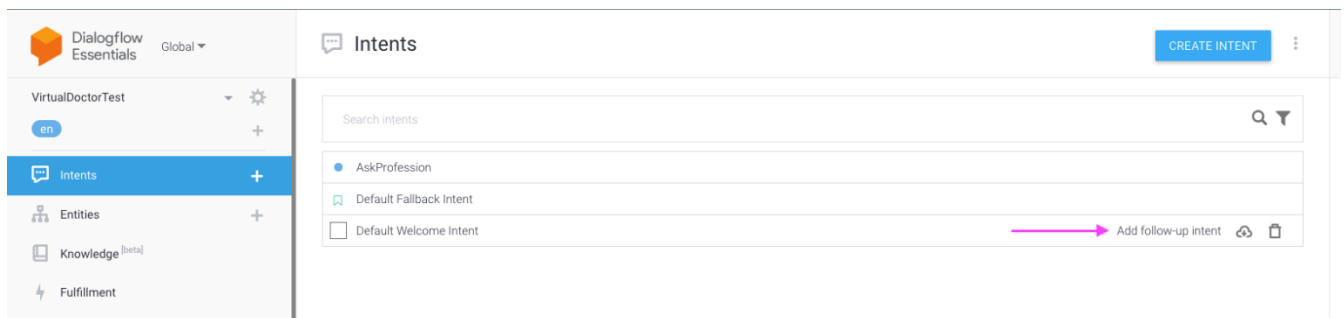


Agora scroll-up (role) até o topo da página. A primeira seção refere-se a **Contexts** (Contextos). Esse é um recurso muito importante para garantir que possamos passar informações de uma intenção para a próxima e, portanto, ter conversas contextualizadas. Usaremos esta funcionalidade neste intent para armazenar o nome do utilizador e passá-lo para o próximo intent.

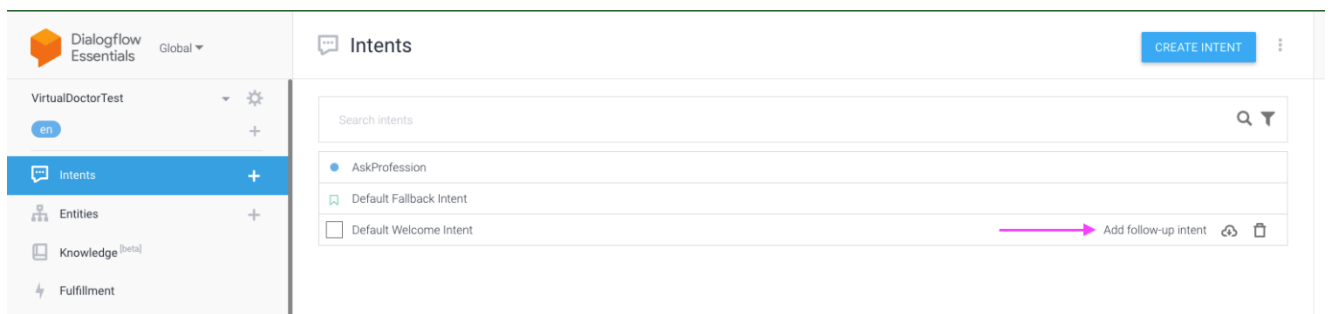
8. Adicione **Output Contexts** na caixa “Add output context”:
 - a. Welcome (Bem-vindo)
 - b. DefaultWelcomeIntent-followup
9. Atualize a lifespan (número antes do contexto) de “5” para “1” e “2”, respectivamente, e clique em **Save**:






10. Volte para o ecrã tela inicial e **pass**e o rato sobre “Default Welcome Intent”. Verá a opção “**Add follow-up intent**” (“Adicionar intenção de acompanhamento”). Clique nela e selecione a opção “**custom**” (“personalizado”):




11. Abra o recém-criado “**Default Welcome Intent - custom**”. Altere o nome da intenção para “**GetName**” e confirme se os contextos de saída criados anteriormente agora estão definidos como os contextos de entrada da nova intenção (intent):



12. Adicione algumas Frases de Treino que são esperadas como respostas, nomes, como “Paulo”, “Mike”, “Eva”, e clique **Save**:

Training phrases  Search training phrases  

 Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression


” Paulo

” Miguel





” Eva

13. Agora, precisamos criar um parâmetro (**parameter**) que será responsável por armazenar o nome do utilizador. Na seção “**Action and parameters**” (“Ação e parâmetros”), crie uma nova instância com as seguintes informações, e grave (**Save**)

- **Parameter Name:** name
- **Entity:** @sys.any (*esta é uma entidade standard do sistema que pode ser usada para diferentes objetivos*)
- **Value:** \$name

Action and parameters 

DefaultWelcomeIntent.DefaultWelcomeIntent-custom

REQUIRED 	PARAMETER NAME 	ENTITY 	VALUE	IS LIST 
<input type="checkbox"/>	name	@sys.any	\$name	<input type="checkbox"/>

[+ New parameter](#)

14. Agora, depois de gravar, volte para as frases de treino e clique duas vezes nos nomes adicionados na etapa 12. Selecione a opção que acabamos de criar como parâmetro (**@sys.any:name**). Faça isso para todos os nomes e grave:

Training phrases ⓘ Search training phrases 🔍 ^

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression

” Pai

” Miguel

” Eva

Filter

@sys.any:name

@sys.place-attraction-us

@sys.ordinal

@sys.currency-name

@sys.person

@sys.unit-volume-name

+ Create new

+ New parameter

ENTITY ⓘ	VALUE	IS LIST ⓘ
@sys.any	\$name	<input type="checkbox"/>

Deve atribuir e configurar automaticamente a relação entre o nome e o parâmetro, destacando-o com uma cor diferente, assim:

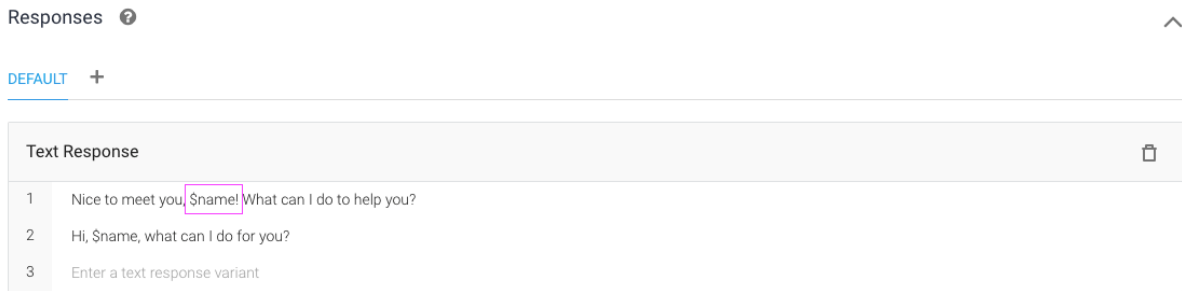
” Paulo

” Miguel

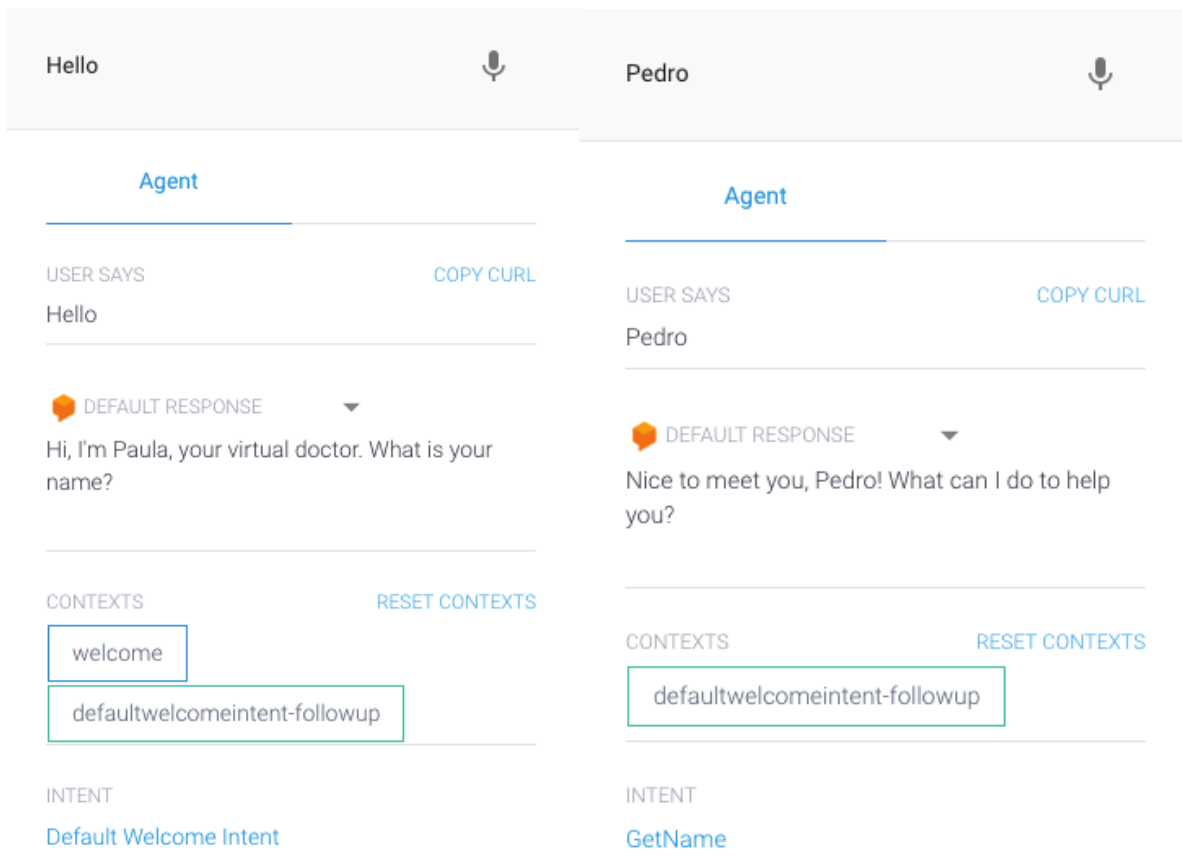
” Eva

PARAMETER NAME	ENTITY	RESOLVED VALUE
name	@sys.any	Eva

15. Para finalizar esta etapa, devemos adicionar algumas “**Responses**” a este intent, usando o valor do parâmetro que definimos antes (**\$name**), como na imagem abaixo. Adicione as respostas e grave (**Save**):



Agora devemos ser capazes de testar o novo intent, trabalhando com o parâmetro e respondendo ao utilizador com seu próprio nome. E o primeiro passo está feito 😊

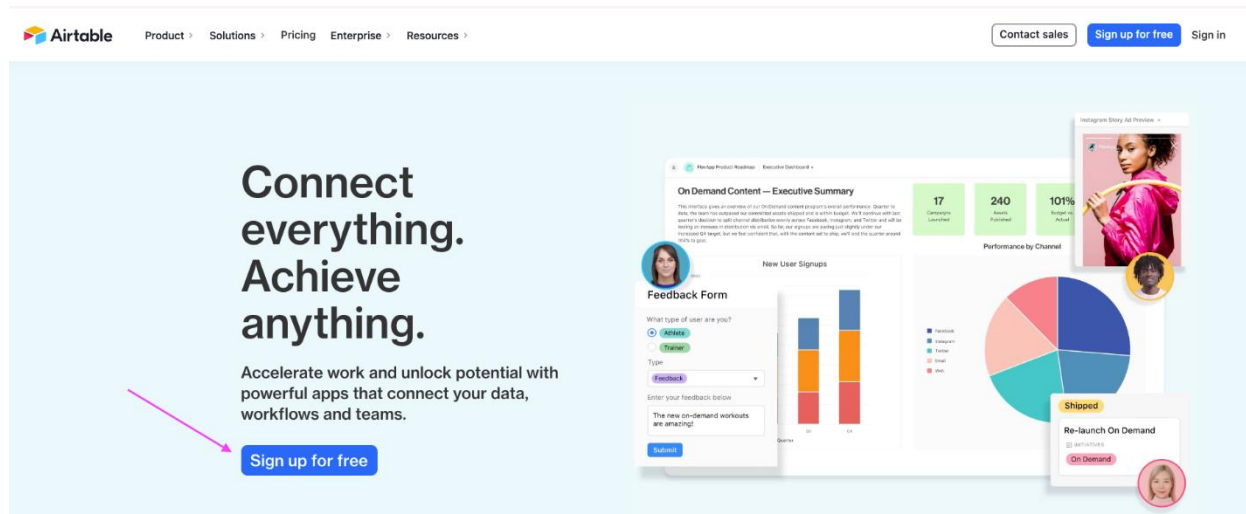


Podemos ver na primeira captura de ecrã que os contextos (**contexts**) estão a ser automaticamente passados da intenção 1 (**Default Welcome Intent**) para a seguinte. Na segunda captura de ecrã, podemos ver que, como o parâmetro **name** foi passado da intent 1, o bot já está a utilizar o valor armazenado (neste exemplo, “Pedro”).

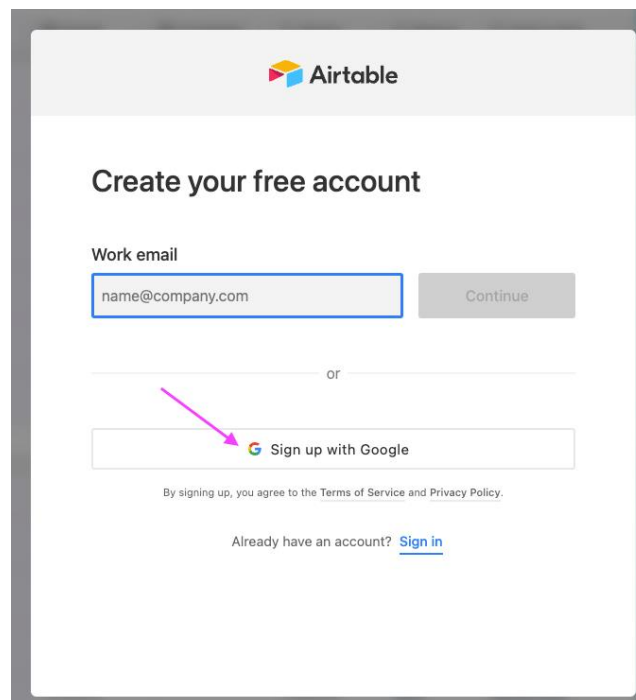
c. Configuração da base de dados Airtable & configurar a conexão

Para garantir que o chatbot seja mais dinâmico e que possa, de facto, ir buscar informações a uma base de dados externa (sem ter todas as respostas como *intents* fixas), temos que criar uma base de dados, que terá algumas informações necessárias para **endereçar** as dúvidas dos utilizadores. Usaremos a plataforma **Airtable**, que é uma forma de folha de cálculo (spreadsheet) muito simples e fácil de usar para criar base de dados simples com tabelas (linhas e colunas). Vamos lá fazer isto!!

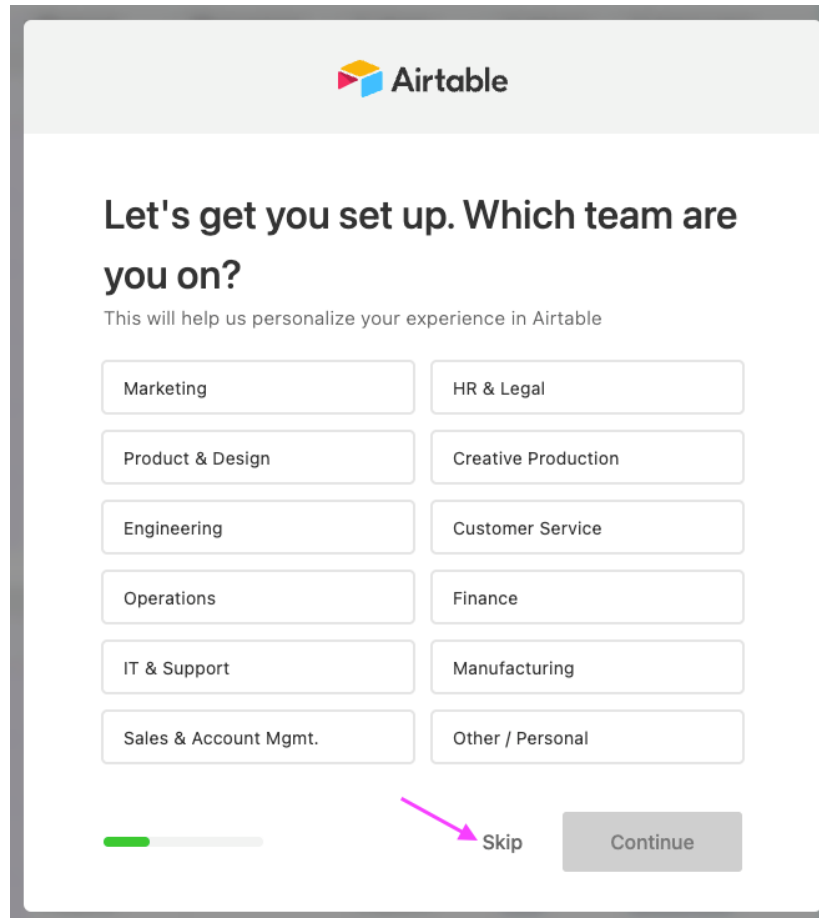
1. Abrir o site Airtable (<https://www.airtable.com/>) e clicar no botão “Sign up for free”:



2. Para garantir que usamos a mesma conta para tudo, devemos usar a opção “Sign up with Google” e usar a mesma conta de antes. Permitir permissões para Airtable:

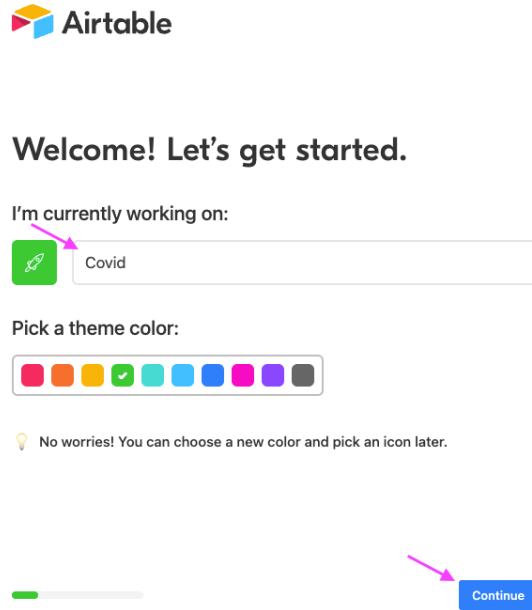


- Quando já estivermos "logados", devemos "saltar" todas as etapas do assistente do Airtable, para que possamos criar uma nova base de Dados de raiz e "Start Building!" ("Começar a Construir!"):



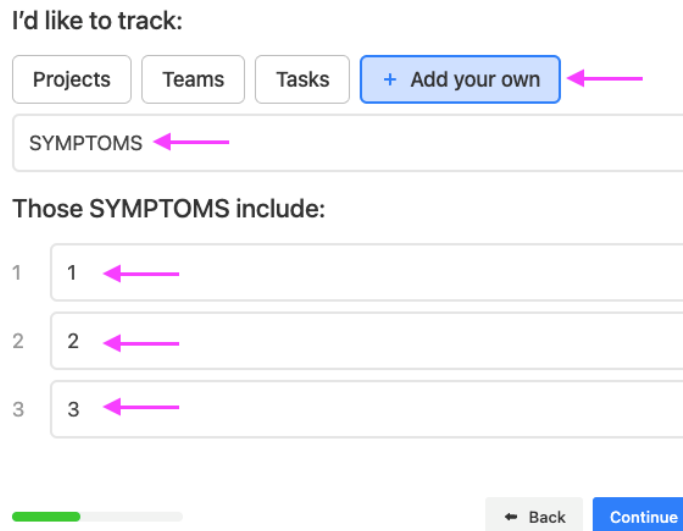
The image shows the Airtable onboarding interface. At the top, the Airtable logo is displayed. Below it, the heading reads "Let's get you set up. Which team are you on?". A subtext says "This will help us personalize your experience in Airtable". There are two columns of buttons representing different teams: Marketing, HR & Legal, Product & Design, Creative Production, Engineering, Customer Service, Operations, Finance, IT & Support, Manufacturing, Sales & Account Mgmt., and Other / Personal. At the bottom, there is a progress bar on the left, a "Skip" button with a pink arrow pointing to it, and a "Continue" button.

4. Devemos dar um nome ao Projeto (por exemplo, “Covid”) e selecionar uma cor para a base de dados:



5. Na próxima etapa, devemos criar a primeira **tabela** (pertencente à base de dados que estamos a utilizar). Devemos clicar no botão “+ Add your own” (“+ Adicionar o seu”) e nomeá-la “**SYMPTOMS**” (“SINTOMAS”). Devemos também adicionar alguns “default row numbers” com 1, 2 e 3 (mais tarde vamos mudar isto...). Clique em **Continue** (Continuar):

Now, we'll add some info.



6. Na próxima etapa, precisaremos adicionar pelo menos 1 coluna na tabela. Vamos adicionar o “FEVER”, com o tipo de dados “Single Line Text”, e clicar em continuar:

Great job! Let’s keep going.

For each row in SYMPTOMS, I’d also like to track:

Status Priority Start date Deadline

Assets Tags Flagged Assignee

A Single line text

Pick up to four now — you’ll be able to add more later.

7. A próxima etapa é a configuração da automação. Como está fora do âmbito do projeto, devemos clicar na opção “Create your own workflow later” (Criar seu próprio fluxo de trabalho mais tarde”) e clicar em **Continue** (Continuar):

We’ll handle it, auto-magically.

Automations make it easy to put workflows on autopilot.

Send a Slack message
When a record is updated in SYMPTOMS, send a Slack message

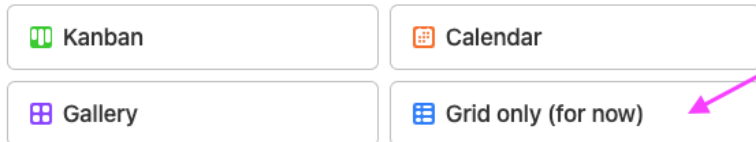
Send a weekly digest
At 12:00 am Monday, email records where Name is not empty

Create your own workflow later
Set up your own automation in the future

- O último passo na etapa de configuração é configurar o visual da base de dados. Assim, vamos seleccionar a opção mais simples que é **“Grid only (for now)”** e clicar em **Continue**:

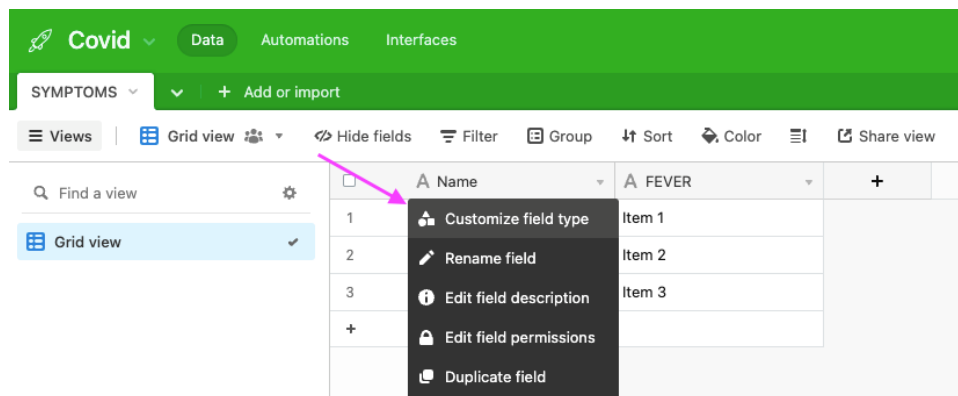
See your info in a new light.

In addition to a grid, I'd also like to see my data as a:

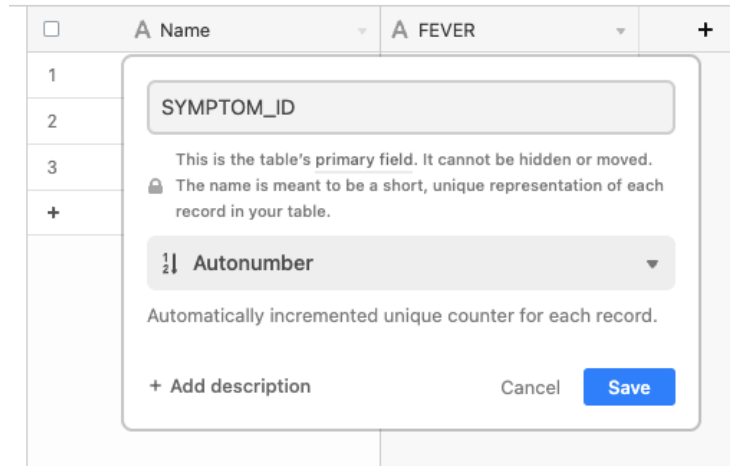


A Base de Dados denominada **“Covid”** deverá agora ter a sua estrutura principal configurada, com a tabela **“SYMPTOMS”** (“SINTOMAS”), as colunas **“Name”** (default) e **“FEVER”** e os 3 registos por defeito que foram criados como parte do wizard também devem existir. Em seguida, devemos saltar para a tarefa de configurar a estrutura da tabela e adicionar um conjunto de registos de sintomas.

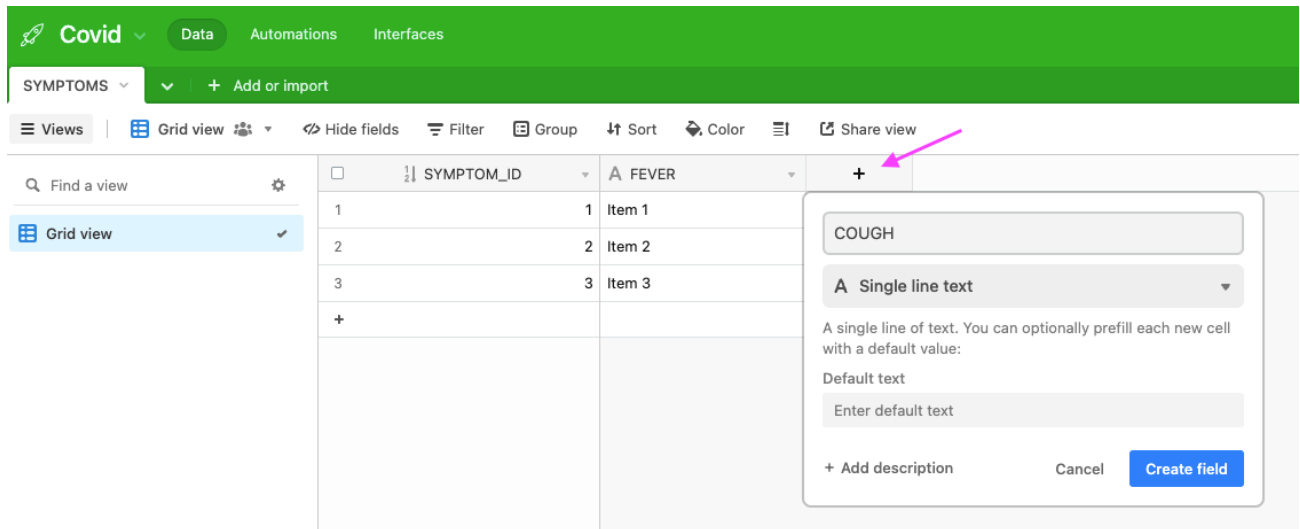
- Atualize a coluna **“Name”** (“Nome”). Primeiro, clique na seta para baixo e selecione a opção **“Customize field type”** (Personalizar tipo de campo):



10. Defina o nome como **“SYMPTOM_ID”**, selecione o tipo de dados **“Autonumber”** e clique em **Save**:



11. Clique no botão **“+”** (ao lado da última coluna **“FEVER”**) e adicione uma nova coluna. Devemos adicionar 3 novos, chamados **“COUGH”**, **“DIARRHEA”** e **“DIAGNOSTIC”**, todos com o mesmo tipo de dados **“Single line text”**:



Depois de concluir a etapa 11, a estrutura da tabela deve ficar assim:

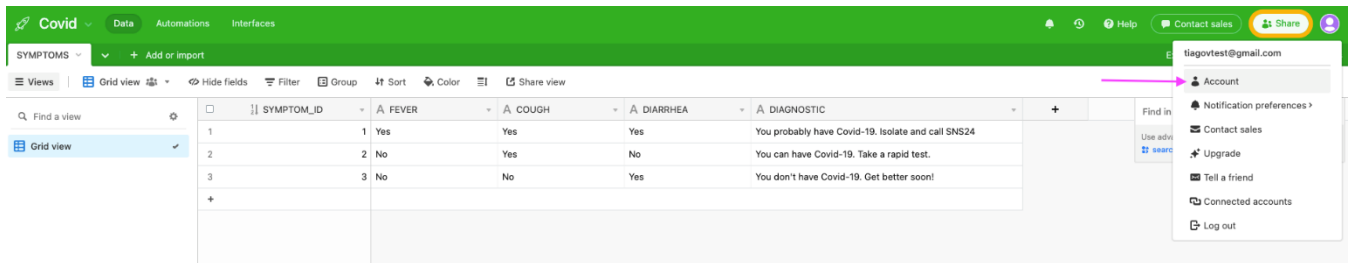
	SYMPTOM_ID	FEVER	COUGH	DIARRHEA	DIAGNOSTIC
1	1	Item 1			
2	2	Item 2			
3	3	Item 3			
+					

12. Preencha os registos com informações, que serão utilizadas pelo bot como base para **identificar os sintomas** e fornecer um **diagnóstico**. Preencha os registos conforme a imagem abaixo:

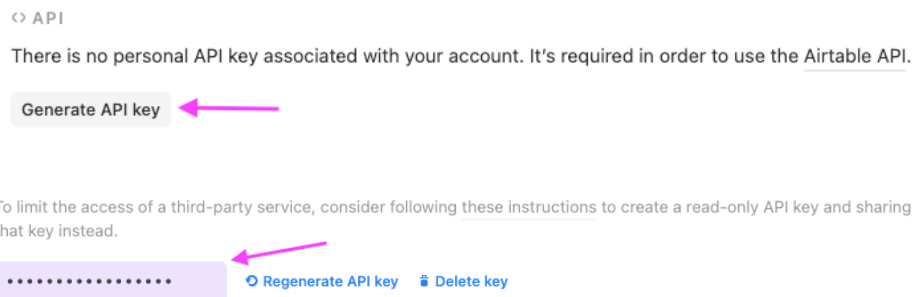
SYMPTOM_ID	FEVER	COUGH	DIARRHEA	DIAGNOSTIC
1	Yes	Yes	Yes	You probably have Covid-19. Isolate and call SNS24
2	No	Yes	No	You can have Covid-19. Take a rapid test.
3	No	No	Yes	You don't have Covid-19. Get better soon!

13. O último passo que devemos realizar no Airtable, que posteriormente será utilizado na configuração do Runkit – é **gerar e gravar** (num arquivo .txt por exemplo) a **Airtable API Key** (Chave da API do Airtable) – será muito importante para ser posteriormente incluída no código que irá conectar o Dialogflow ao Airtable:

- Clique no **Profile Icon** (ícone do perfil) e na opção **“Account”** (“Conta”)

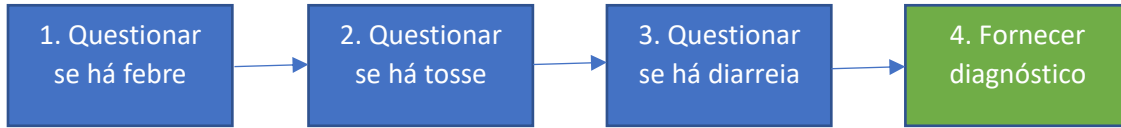


- Na seção **“ <> API ”** clique no botão **“Generate API Key”**. (“Gerar chave de API”). Depois de gerado, podemos clicar na **caixa roxa**, que mostrará a chave da API para esta base de dados Airtable específica:



d. Criar Dynamic Intents

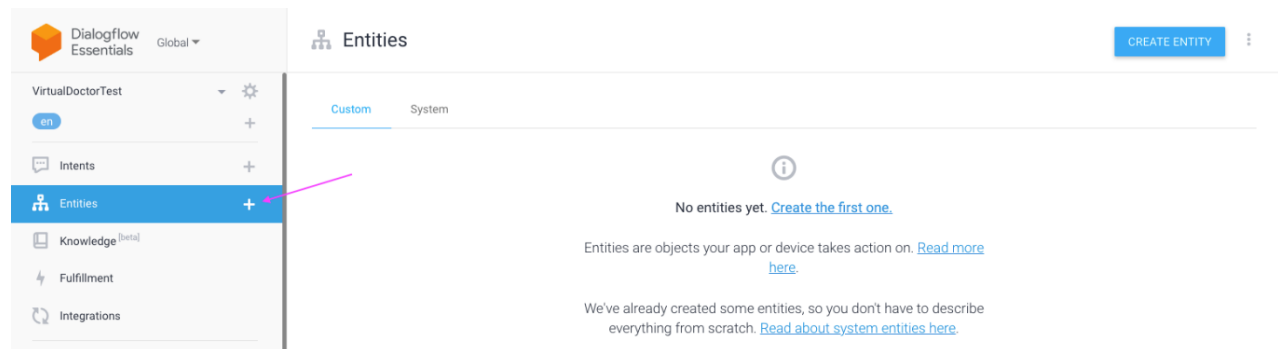
Nesta etapa, vamos configurar as intenções dinâmicas e o fluxo sequencial de conversa entre o utilizador e o chatbot. O objetivo é que o bot identifique o que o utilizador está a perguntar (relacionado com o Covid) e faça um conjunto de perguntas, a fim de conseguir dar um bom diagnóstico. Vamos configurar um total de 4 *intents* e 1 *entidade*, pela seguinte ordem



A última *intent* (intenção) (4. Fornecer diagnóstico) é a intenção que deve capturar todas as entradas do utilizador, conectar-se à base de dados Airtable que foi configurada na etapa anterior e fornecer o diagnóstico correto ao utilizador.

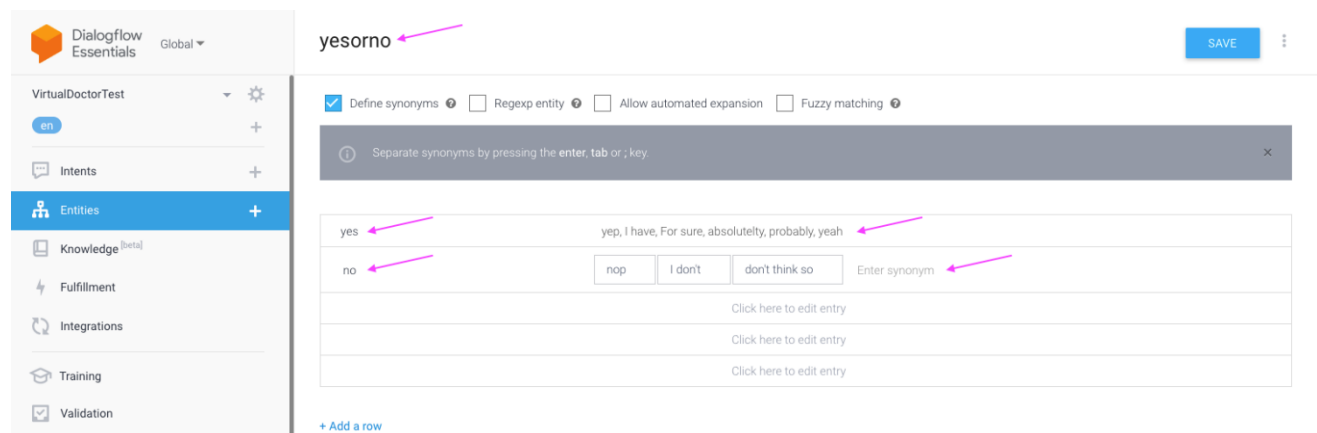
Vamos começar com a criação da Entidade.

1. Clique no botão "+" no menu **Entities** (Entidades), no canto superior esquerdo do menu do dialogflow



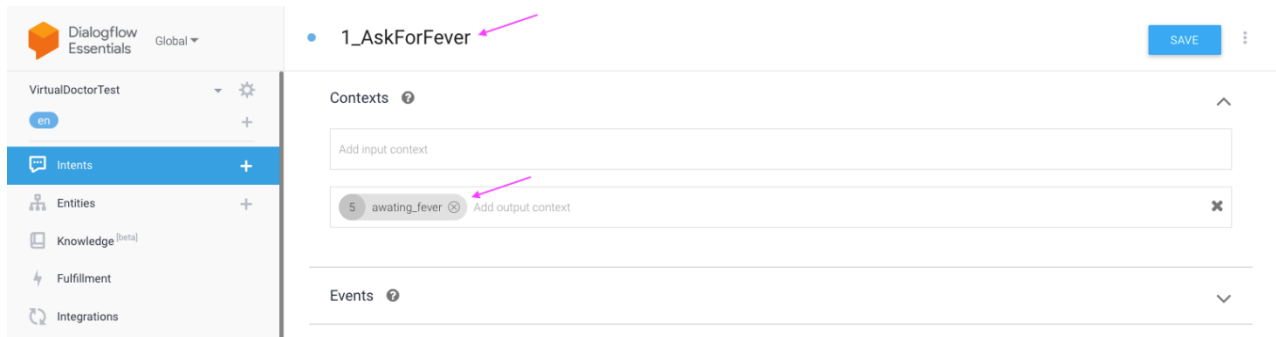
2. Agora crie a entidade com as seguintes configurações (como mostra a imagem abaixo) e clique em **Save**:

- **Name:** yesorno
- **Entries:** yes and no
- **Synonyms:** um conjunto de sinónimos para "yes" e para "no" words (yep, I have, I don't, nop, etc)

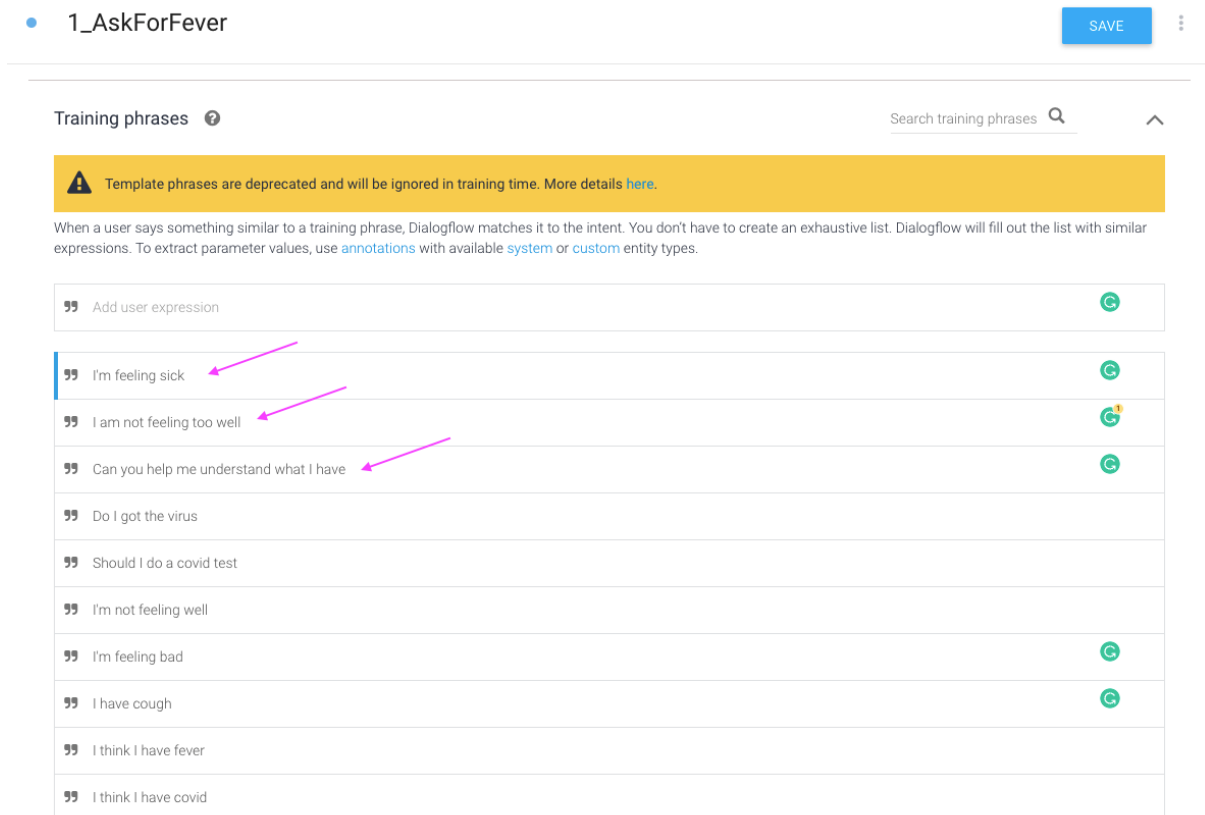


Essa entidade corresponderá aos parâmetros que adicionaremos nas próximas intenções e funcionará como as respostas que o utilizador fornecerá quando for questionado sobre perguntas sobre doenças (por exemplo, "Você está com febre?"). Devemos agora passar para a criação dos **4 intents** que irão ditar o fluxo da conversa.

3. Clique no botão "+" para criar um novo **Intent**. Vamos adicionar o nome "**1_AskForFever**" e o contexto de saída "**awaiting_fever**". Esse contexto será fundamental para passar as informações recolhidas para o próximo intent:




4. Devemos agora adicionar um conjunto de **Frases de Treino (Training Phrases)** a esta intenção. Lembre-se que esta é a primeira interação com o utilizador, portanto, devemos incluir coisas como "I feel sick" ou "Should I do a covid test". Adicione-os (tal como na imagem abaixo) e clique em **Save** (Gravar):



5. Certifique-se de que, ao adicionar as frases de treino, todas as associações com a entidade criada na etapa 2 são removidas. Para remover essa associação, clique na caixa “x” no lado direito de cada frase de treino:

Training phrases ? Search training phrases

 Template phrases are deprecated and will be ignored in training time. [More details here.](#)

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression G								
” I have cough G								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">PARAMETER NAME</th> <th style="width: 30%;">ENTITY</th> <th style="width: 30%;">RESOLVED VALUE</th> <th style="width: 10%;"></th> </tr> </thead> <tbody> <tr> <td>yesorno</td> <td>@yesorno</td> <td>I have</td> <td style="text-align: right;">x</td> </tr> </tbody> </table>	PARAMETER NAME	ENTITY	RESOLVED VALUE		yesorno	@yesorno	I have	x
PARAMETER NAME	ENTITY	RESOLVED VALUE						
yesorno	@yesorno	I have	x					
” I think I have fever G								
” I think I have covid								

6. Adicione uma resposta (**Response**) a esta intenção “Entendi, antes de tudo, fique calmo 😊 Você está com febre, sim ou não?” (“**Got it, first of all, stay calm 😊 Do you have fever, yes or no?**”) e clique em **Save** (Gravar):

Responses ? ^

DEFAULT +

Text Response 🗑

1	Got it, first of all, stay calm :) Do you have fever, yes or no? ←
2	Enter a text response variant

ADD RESPONSES

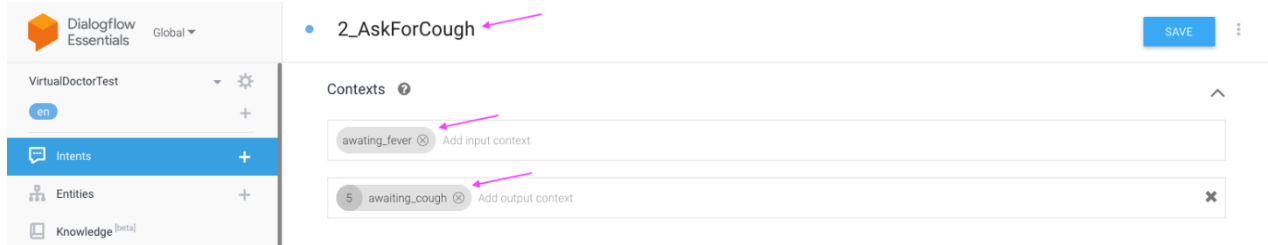
7. Certifique-se de que as 2 **Fulfillment toggles** estejam ativadas (isso será importante para se poder utilizar posteriormente o bloco de código). Depois de ativado, clique em **Save** (Gravar). A primeira intenção dinâmica é agora criada:

Fulfillment ? ^

Enable webhook call for this intent

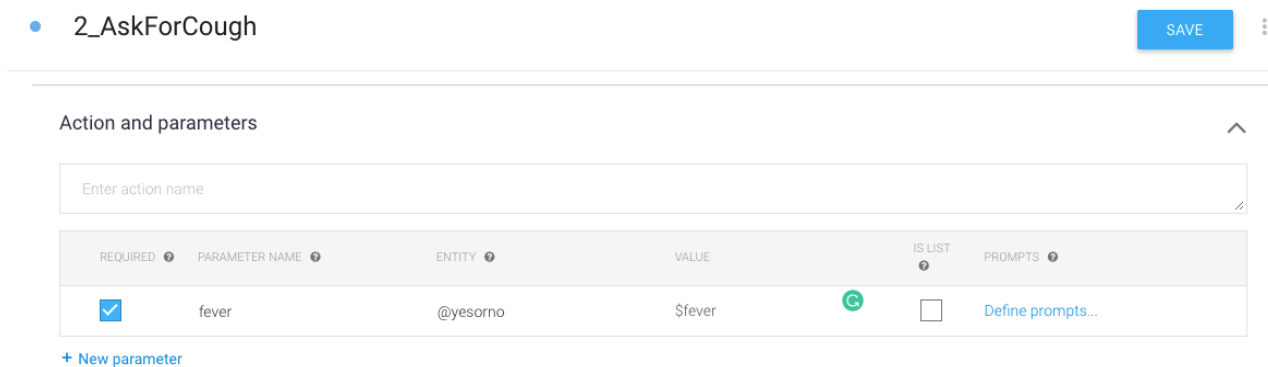
Enable webhook call for slot filling

8. Crie o 2º intent, clicando novamente no botão “+”, adicione o nome “**2_AskForCough**”, configure o contexto de entrada como “**awaiting_fever**” (será integrada a saída do intent anterior) e o esatbelcido o contexto de saída como “**awaiting_cough**”. Faça isso e clique em **Save** (Gravar).



9. Antes de adicionar as Frases de Treino, adicionar um **novo parâmetro**, com as seguintes configurações. Em seguida clique em **Save** (Gravar).

- **Required:**
- **Parameter Name:** fever
- **Entity:** @yesorno
- **Value:** \$fever



10. Agora vamos adicionar as Frases de Treino (**Training Phrases**) que esperamos que o utilizador responda (respostas relacionadas sim ou não). Adicione um conjunto deles e clique em **Save** (Gravar).

Training phrases ⓘ

Search training phrases 🔍

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression	⊕
” Nop	⊕
” No	⊕
” Yeah	
” Yes	
” I think so	
” Likely	
” For sure	

- Depois de todas as frases de treino terem sido adicionadas, devemos agora **associá-las** ao parâmetro que acabamos de criar. Para fazer isso, devemos **passar o mouse** sobre o texto da resposta, a resposta inteira (por exemplo, resposta “Acho que sim”, selecione tudo), clique nela e selecione o parâmetro “**@yesorno:fever**”. Devemos fazer isso para todas as frases de treino e clicar em **Save** (Gravar).

• 2_AskForCough

SAVE

Training phrases

Search training phrases

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression

” Nop

Filter

yesc @yesorno:fever

” N @yesorno:yesorno

” Y @sys.date-time

” Y @sys.geo-capital

” It @sys.geo-state

” Li @sys.zip-code

” Fr

+ Create new

ENTITY	RESOLVED VALUE
@yesorno	Nop

12. Devemos adicionar a resposta (**Response**) que deve direcionar o utilizador para a próxima intenção. Digite a resposta de texto **“Ok, and do you have cough, yes or no?”** (“Ok, e você tem tosse, sim ou não?”) e clicar em **Save** (Gravar).

Responses

DEFAULT +

Text Response

- 1 Ok, and do you have cough, yes or no?
- 2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

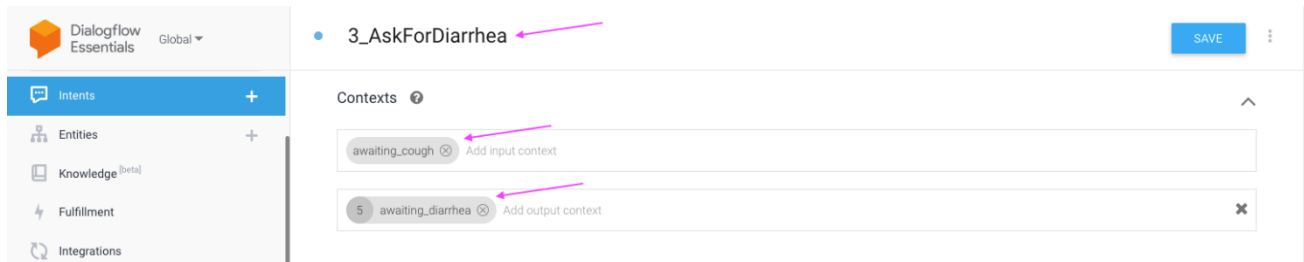
13. Por fim, de modo semelhante ao intent anterior, vamos habilitar as 2 opções de **fulfillment** e clicar em **Save** (Gravar):

Fulfillment ?

Enable webhook call for this intent

Enable webhook call for slot filling

14. Crie a 3ª intenção, clicando novamente no botão “+”, adicione o nome “3_AskForDiarrhea”, configure o contexto de entrada como “awaiting_cough” e o contexto de saída como “awaiting_diarrhea”. Faça isso e clique em **Save** (Gravar):



15. Tal como anteriormente, vamos adicionar um novo **parâmetro**, com as seguintes configurações e clicar **Save** (Gravar):

- **Required:**
- **Parameter Name:** cough
- **Entity:** @yesorno
- **Value:** \$cough

Action and parameters ?

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	cough	@yesorno	\$cough	<input type="checkbox"/>	Define prompts...

[+ New parameter](#)

16. Vamos agora adicionar as **Training Phrases** (Frases de Treino) (iguais, para respostas do tipo sim e não) e clicar em cada resposta para associar a resposta ao **parâmetro tosse (cough parameter)** que acabamos de criar (**@yesorno:cough**). Depois de tudo associado, clique **Save** (Gravar):

3_AskForDiarrhea

SAVE

Training phrases

Search training phrases

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

PARAMETER NAME	ENTITY	RESOLVED VALUE
cough	@yesorno	No

Other training phrases: No, Yes, Yep, Nop, For sure, Likely, I do

17. Agora vamos adicionar a Resposta **"Got it, and do you have diarrhea, yes or no?"** ("Entendi, e você está com diarreia, sim ou não?") que direcionará o utilizador para a próxima intenção e **active as 2 fulfilment toggles** (duas opções de atendimento). Após isso, clique **Save** (Gravar):

Responses

DEFAULT +

Text Response

- 1 Got it, and do you have diarrhea, yes or no?
- 2 Enter a text response variant

ADD RESPONSES

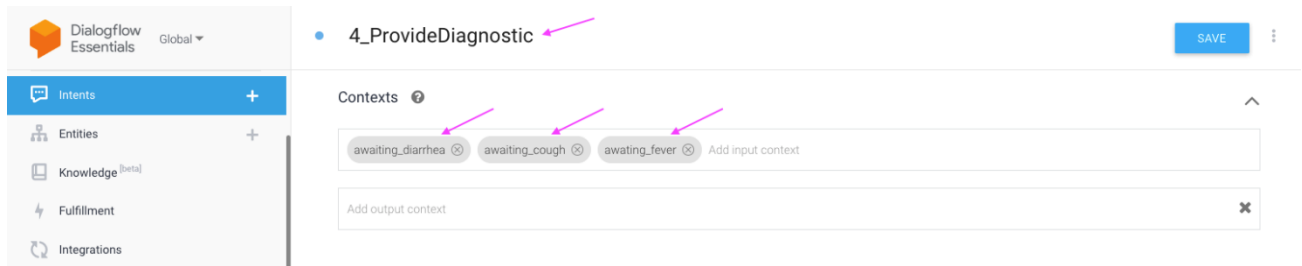
Set this intent as end of conversation

Fulfillment

Enable webhook call for this intent

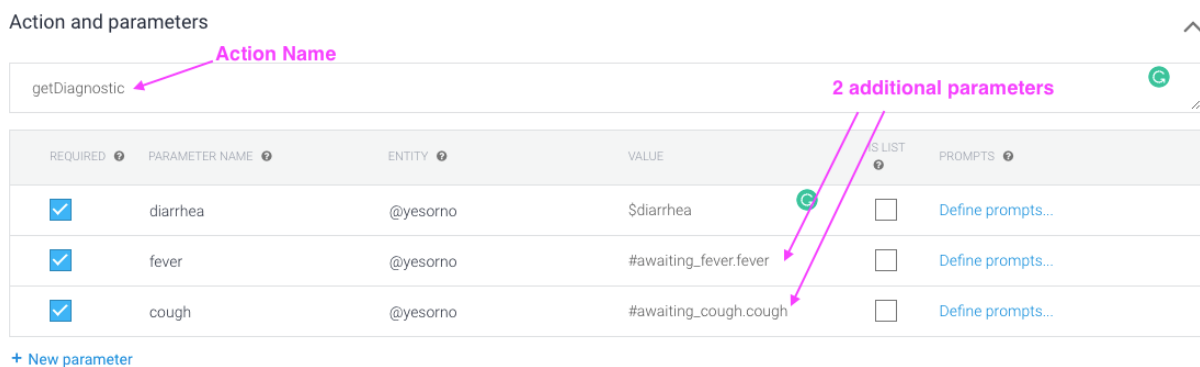
Enable webhook call for slot filling

18. Passando para a 4ª e **última intent**, que é a que servirá para acionar o webhook para ir buscar os dados na base de dados do Airtable, na posse de todas as informações já fornecidas pelo utilizador. Em seguida, clicar no botão “+” para criar uma nova intenção, adicionar o nome “**4_ProvideDiagnostic**” e também os **3 contextos de entrada** (awaiting_diarrhea, awaiting_cough e awaiting_fever). Certifique-se de remover também os contextos de saída gerados, pois sendo a última intenção não é necessário passar adicionais. Após, clique **Save** (Gravar):



19. Passando para a seção **Action and Parameters** (Ação e Parâmetros). Sendo a última intenção, temos que configurá-la de forma um pouco diferente dos anteriores. Vamos configurar::

- Adicione o **Action Name** com “**getDiagnostic**” (mais tarde chamaremos essa ação no código);
- Adicione o parâmetro “**diarrhea**” semelhante aos dos intents anteriores (obrigatório , nome “**diarrhea**”, entidade “**@yesorno**”, valor “**\$diarrhea**”;
- Adicione 2 **parâmetros** adicionais, para febre e tosse. Defina os seus valores como “**#awaiting_fever.fever**” e “**#awaiting_cough.cough**”



20. Agora vamos adicionar as **Training Phrases** (Frases de Treino) deste intent e associá-lo ao parâmetro criado para diarreia (**@yesorno:diarrhea**). Depois disso, clique **Save** (Gravar):



⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

” Add user expression G

” **No** G

PARAMETER NAME	ENTITY	RESOLVED VALUE
diarrhea	@yesorno	No

” **Yes** G

” **I think so** G

” **I do** G

21. Por último, vamos adicionar uma resposta de texto **Text Response** e ativar as **22 Fulfillment toggles**. Para a resposta de texto, podemos adicionar apenas uma resposta de teste, para testar se a conversa sequencial está funcionando (por exemplo, “((aqui quero obter o diagnóstico do airtable)))”, pois posteriormente, quando implementarmos o código, ele deverá utilizar a resposta no Airtable. Feito isso, clique **Save** (Gravar):



DEFAULT +

Text Response 🗑

- 1 ((here I want to get the diagnostic from airtable))
- 2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation ?



- Enable webhook call for this intent
- Enable webhook call for slot filling

As intenções dinâmicas (**Dynamic Intents**) agora estão totalmente configuradas. Os próximos passos são implementar a conexão que vai usar o diagnóstico no Airtable e que usará a ferramenta de código **runkit** (**runkit code**). Antes de passar para a próxima etapa, certifique-se de testar a conversa sequencial entre o utilizador e o bot, conversando e fornecendo uma entrada para a primeira intenção, como "Estou a sentir-me doente" ("**I'm feeling sick**"). O resultado deve retornar a última resposta de espaço reservado que configuramos:

The screenshot shows a chatbot interface with the following components:

- A header bar with "Try it now" and a microphone icon.
- An "Agent" section header.
- A "USER SAYS" section with the text "yes" and a "COPY CURL" link.
- A "DEFAULT RESPONSE" section containing the text "((here I want to get the diagnostic from airtable))".
- An "Expected Answer" section header.
- A "CONTEXTS" section with "RESET CONTEXTS" and three buttons: "awaiting_fever", "awaiting_diarrhea", and "awaiting_cough".
- An "INTENT" section with "4_ProvideDiagnostic" and the label "Last intent matched".
- An "ACTION" section with a table of parameters collected.

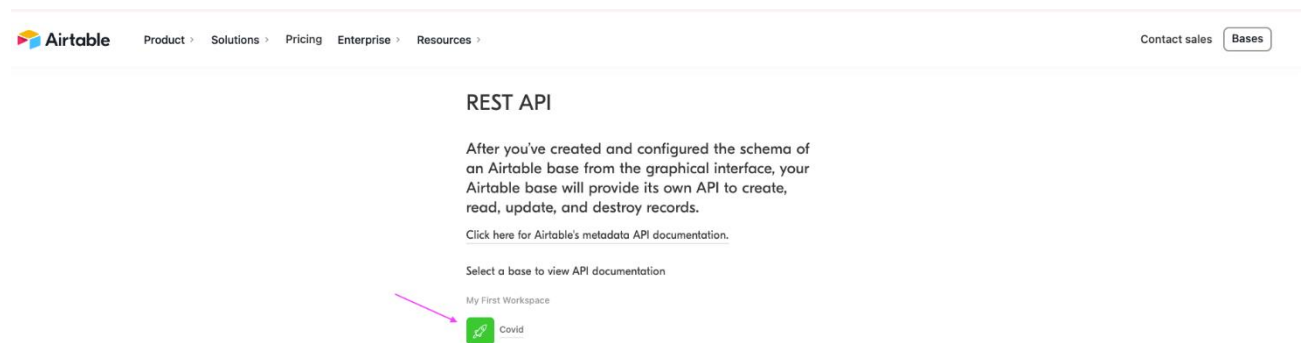
PARAMETER	VALUE
fever	yes
cough	yes
diarrhea	yes

e. Configurar o código de conexão (Runkit Code)

Nesta fase iremos configurar o código de conexão (entre o **Dialogflow** e o **Airtable**) que permitirá acionar o webhook para recolher as informações de diagnóstico, com base nas respostas do utilizador às perguntas do chatbot sobre sintomas.

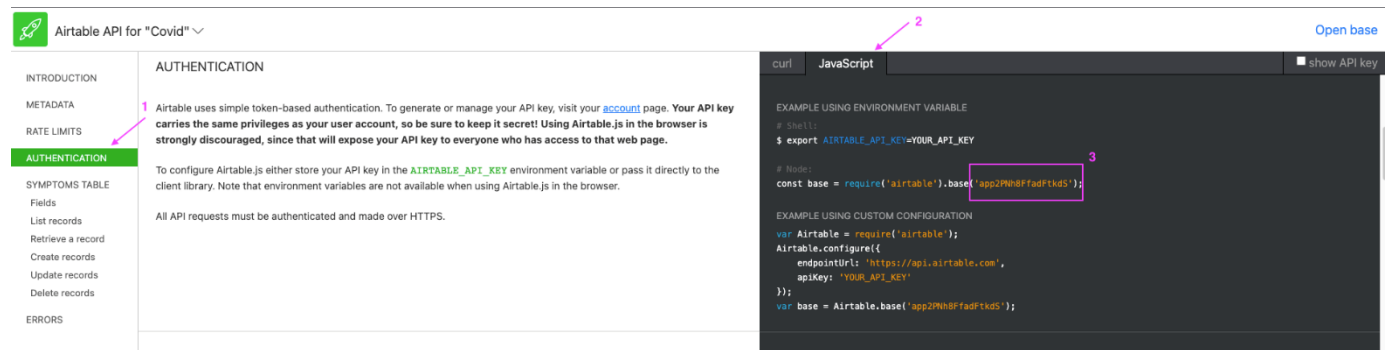
Teremos que criar e configurar uma conta runkit que armazenará esse código, mas primeiro, vamos utilizar as chaves de API necessárias para a conexão

1. Abrir o url <https://airtable.com/api> (já logado no airtable) e selecione a base de dados do Airtable criado “Covid”:



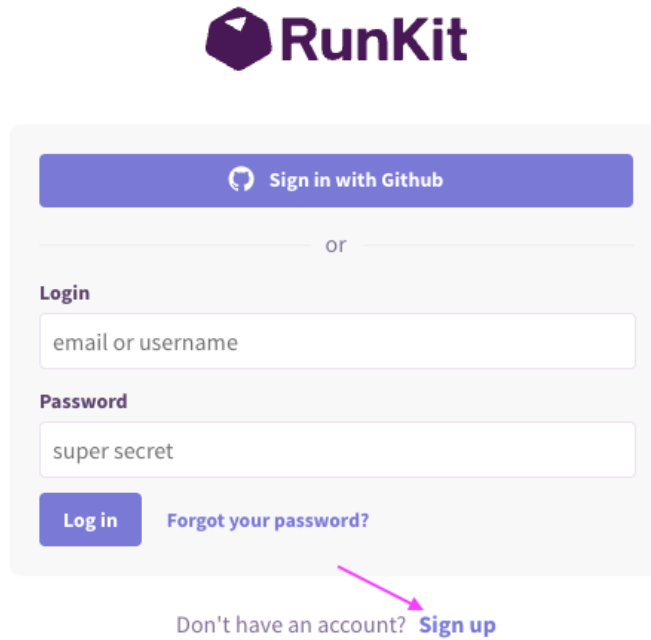
2. Após selecionar a Base de Dados, devemos:

- Clicar no menu de **Authentication** (em cima e à esquerda)
- Selecionar o tabulador **JavaScript** (à direita);
- Copiar a chave de autenticação da base de dados do código e armazenar num bloco de notas (na imagem abaixo, a chave é `app2PNh8FfadFtkdS`)

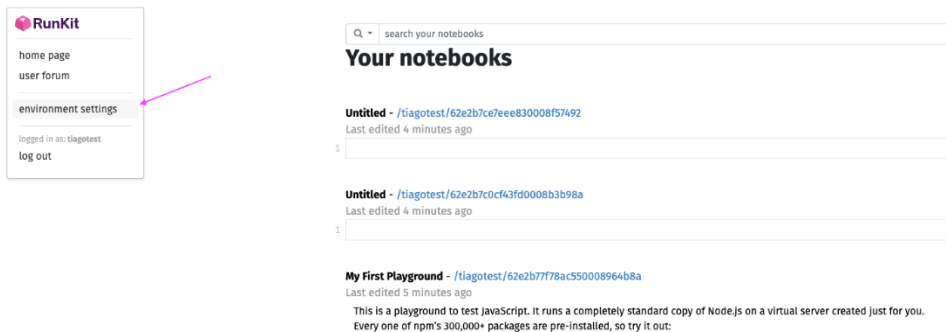


3. A chave mencionada acima é a **database api key** (chave API da base de dados). Também precisamos da instância para a chave da API Airtable, o já que fizemos numa etapa anterior. Caso não o tenhamos feito, basta clicar **profile icon** (ícone do perfil), selecionar a opção **account** (conta) e obter a API na **<> API section** (seção <> API).
4. Em seguida vamos criar a conta do runkit. Devemos abrir a url <https://runkit.com/home> e clicar na opção **“Sign In”** (canto superior direito):

5. Clique na opção **“Sign Up”** (Registo) no ecrã de login e adicione a **mesma conta do Gmail** que usou no Dialogflow e no Airtable:



6. Após o login, passar o mouse sobre o ícone do Runkit (menu superior esquerdo) e clicar na opção **environment settings”** (configurações do ambiente):



7. Seguidamente, clicar no botão **“+ Add New”** na seção **Environment Variables**:

Settings

Email

Environment

Integrations

Change password

Email

The following emails are associated with your account. Your username or primary email address can be used for signing in to Runkit, and any communication from us will be sent to this address. Your email address must be verified before it can become primary.

tiagovtest@gmail.com PRIMARY Resend verification Delete

+ Add New

Environment Variables

Set environment variables that will be available in every notebook, but only visible to you. These are great for things like API keys. When someone else runs your document, their own environment variables will be used. Here is an example:

```
mapboxgl.accessToken = process.env.MAPBOX
```

+ Add New

8. Vamos criar **2 novas variáveis**, uma para a chave de API e outra para a chave de base de dados, e devemos copiar as 2 chaves de API que obtivemos nas etapas 2 e 3:
- **AIRTABLE_API_KEY** -> app2PNh8FfadFtkdS
 - **COVID_BASE** -> keyQBcmhehpySEstz

Environment Variables

Set environment variables that will be available in every notebook, but only visible to you. These are great for things like API keys. When someone else runs your document, their own environment variables will be used. Here is an example:

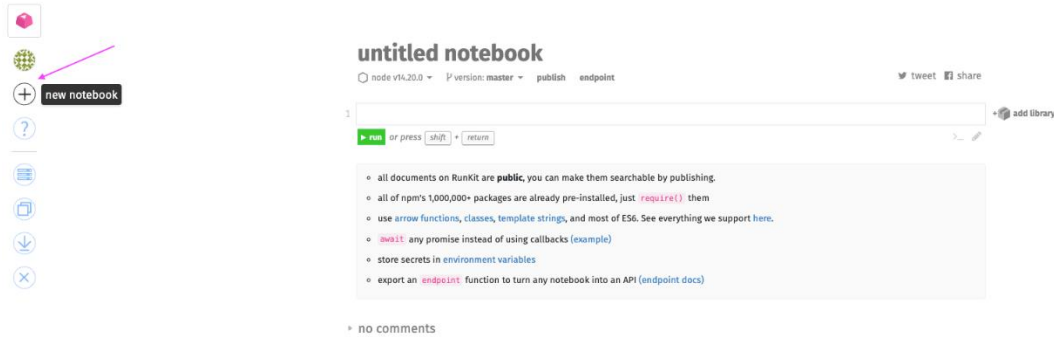
```
mapboxgl.accessToken = process.env.MAPBOX
```

Name	Value	
AIRTABLE_API_KEY	app2PNh8FfadFtkdS	🗑️
COVID_BASE	keyQBcmhehpySEstz	🗑️

+ Add New

Nota importante: cada chave é diferente! Não copie e cole os que estão apresentados na página anterior, pois eles são dados apenas para o exemplo.

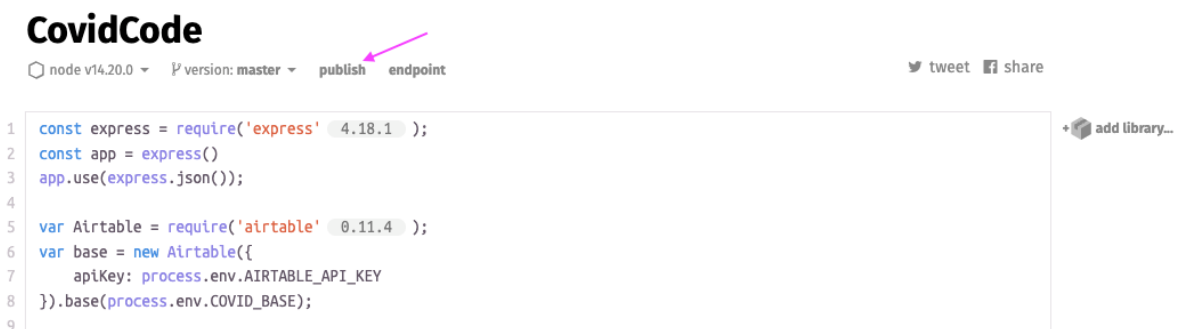
9. Temos agora que retornar à página inicial e clicar no botão “+” para criar um “**New Notebook**” (Novo Notebook) (canto superior esquerdo):



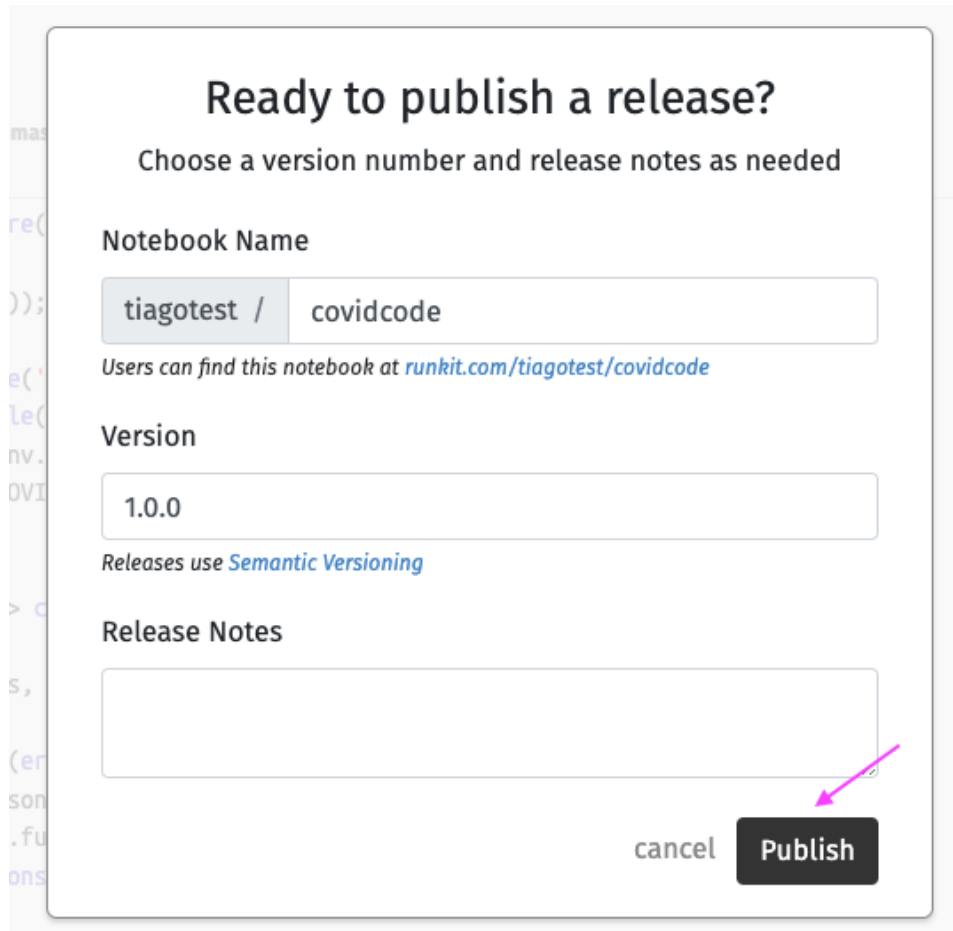
10. Em seguida dar um nome ao notebook “**CovidCode**” e copiar e colar o **bloco de código** apresentado (encontra-se na pasta do projeto, chamado “**chatbot.txt**” (só tem 1 linha em branco antes de copiar e colar). Nota: Se por acaso configurarmos as variáveis com nomes diferentes, o quadrado rosa é onde devemos substituí-las:



11. O passo seguinte é publicar o código do webhook, clicando na opção “**publish**” (publicar) no topo do bloco de código:



12. No assistente de Publicação de Lançamento (Release Publishing), clicar no botão “**Publish**” (Publicar):



Ready to publish a release?

Choose a version number and release notes as needed

Notebook Name

tiagotest / covidcode

Users can find this notebook at runkit.com/tiagotest/covidcode

Version

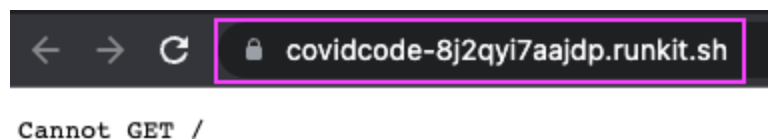
1.0.0

Releases use [Semantic Versioning](#)

Release Notes

cancel **Publish**

13. Por último, depois de publicado, devemos clicar na opção “**endpoint**” (ao lado do botão publicar). Em seguida abrir-se-á uma página em branco com a mensagem “Cannot GET /”. Devemos copiar a URL e grava-la para podermos usar na próxima etapa:



14. No último passo, desta vez, é necessário abrir a consola do **Dialogflow**, clicar no menu **“Fulfillment”** (lado esquerdo da página), ativar o botão **“Webhook”**, e no campo **“URL*”** passar o endpoint URL obtido no passo 13. Depois disso, clicar em Gravar **“Save”**:

The screenshot displays the Dialogflow Fulfillment configuration interface. On the left sidebar, the 'Fulfillment' menu item is highlighted. The main content area is titled 'Fulfillment' and features a 'Webhook' section. The 'ENABLED' toggle is turned on. The 'URL*' field is populated with the URL 'https://covidcode-8j2qyi7aajdp.runkit.sh/'. Below this, there are sections for 'BASIC AUTH' (with 'Enter username' and 'Enter password' fields), 'HEADERS' (with 'Enter key' and 'Enter value' fields, and an 'Add header' button), and 'SMALL TALK' (with a 'Disable webhook for Smalltalk' option). At the bottom, the 'Inline Editor' section is shown as 'DISABLED', with a note: 'Newly created cloud functions now use Node.js 10 as runtime engine. Check [migration guide](#) for more details.'

f. Testar o Chatbot

1. Após habilitar e **gravar** a tela Fulfillment, o chatbot já deve estar pronto e é conveniente testá-lo, por exemplo, seguindo o exemplo da conversa sequencial abaixo
 - a. **Utilizador:** Hi (Olá)
 - b. **Bot:** Hello, I am Paula, your virtual doctor at your service. First of all, what is your name? (Olá, sou a Paula, a sua médica virtual ao seu serviço. Em primeiro lugar, qual é o seu nome?)
 - c. **Utilizador:** Mike
 - d. **Bot:** Nice to meet you, Mike! What can I do to help you? (Prazer em conhecê-lo, Mike! O que posso fazer para te ajudar?)
 - e. **Utilizador:** I'm feeling sick (Estou a sentir-me doente)
 - f. **Bot:** Got it, first of all, stay calm :) Do you have fever, yes or no? (Entendido, antes de mais nada, fique calmo :) Você tem febre, sim ou não?)
 - g. **Utilizador:** yes (sim)
 - h. **Bot:** Ok, and do you have cough, yes or no? (Ok, e você tem tosse, sim ou não?)
 - i. **Utilizador:** yes (sim)
 - j. **Bot:** Got it, and do you have diarrhea, yes or no? (Entendi, e você tem diarreia, sim ou não?)
 - k. **Utilizador:** yes (sim)
 - l. **Bot:** You probably have Covid-19. Isolate and call SNS24 (Provavelmente você está com Covid-19. Isole-se e ligue para SNS24)

Na última frase (letra l.), o bot combina todas as entradas de parâmetro fornecidas e responde com o registo correspondente na base de dados Airtable.

2. Podemos utilizar o assistente de teste (canto superior direito), introduzindo as frases acima e esperando um obter um resultado tal como o apresentado abaixo:

Try it now 

Agent

USER SAYS COPY CURL
yes

 DEFAULT RESPONSE ▼
You probably have Covid-19. Isolate and call SNS24

CONTEXTS RESET CONTEXTS

awaiting_fever	awaiting_diarrhea
awaiting_cough	

INTENT
4_ProvideDiagnostic

ACTION
getDiagnostic

PARAMETER	VALUE
diarrhea	yes
cough	yes
fever	yes

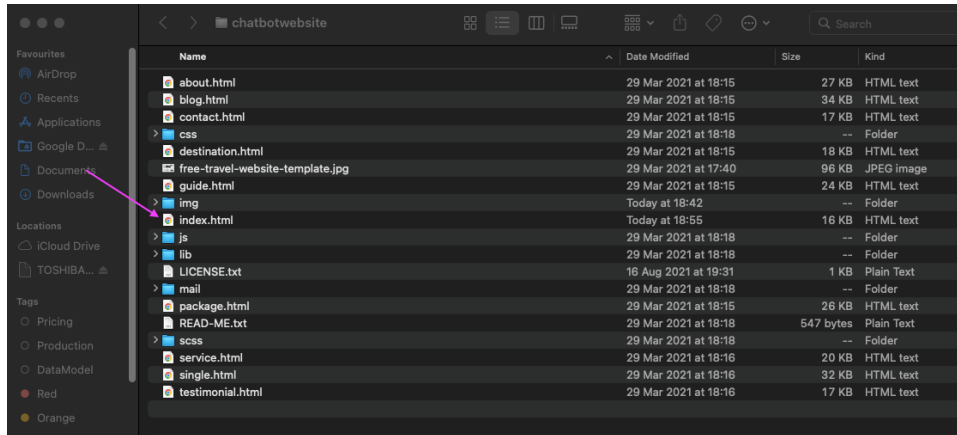
SENTIMENT
Query Score: 0.6

DIAGNOSTIC INFO

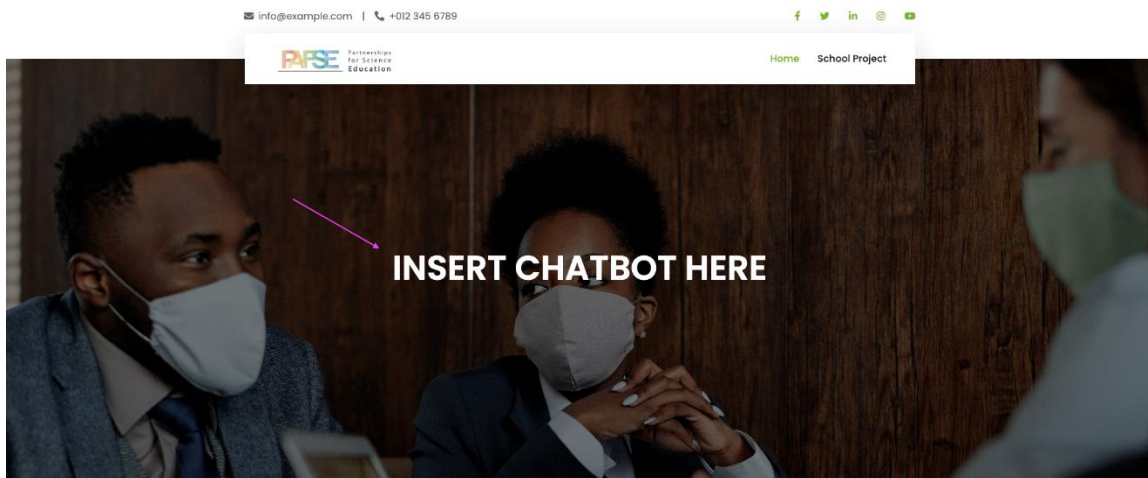
g. Inclusão do bot num Website

Depois do chatbot estar pronto e testado, agora é altura de o ligar a um site. Para isso, consulte a pasta fornecida (**chatbotwebsite**) que contém todos os ficheiros necessários para ligar o bot a um site.

1. Abra a pasta “**chatbotwebsite**” e faça um duplo clique no arquivo “**index.html**”, para que se possa abrir o ficheiro num navegador, e por forma a testarmos as alterações realizadas



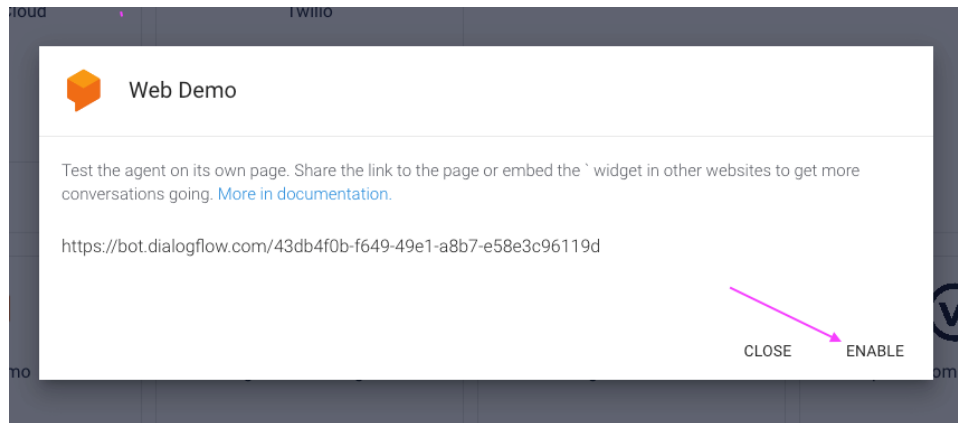
2. O ficheiro será aberto no navegador Google Chrome, e a área que diz **INSERT CHATBOT HERE** (“INSERIR CHATBOT AQUI”) que é a área onde colocaremos o chatbot



- Em seguida, devemos obter o código HTML do chatbot por forma a que possamos colocá-lo no site. Abrir o **Dialogflow** e clicar no menu “**Integrations**”:



- Clicar na opção “**Web Demo**” e depois clicar em **Enable**:



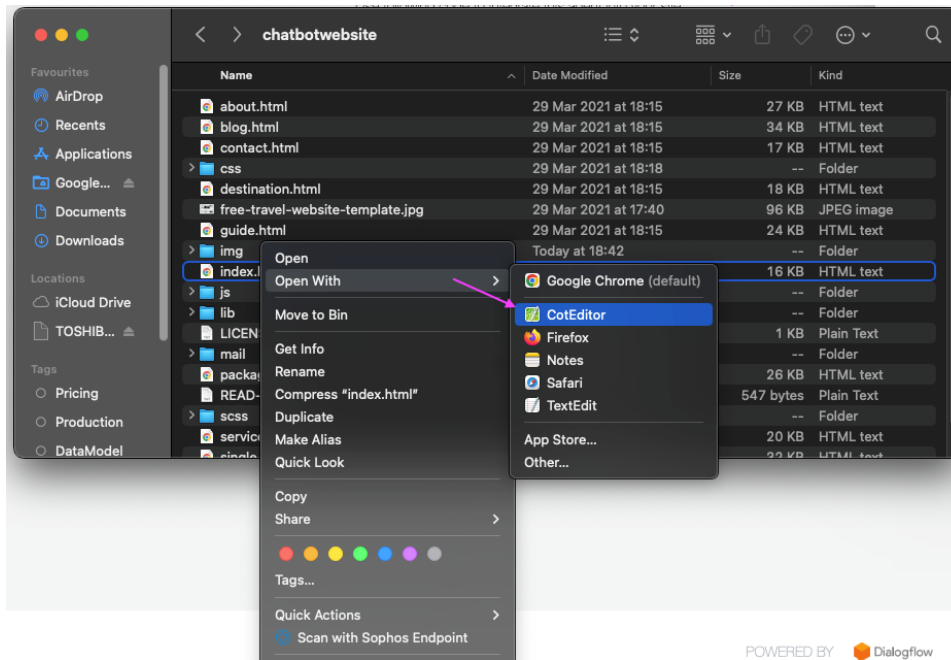
- Depois, devemos clicar na URL gerada que abrirá o **Chatbot** e seu código **HTML**, pois é a peça que devemos copiar. Devemos copiar o código HTML e gravá-lo em algum lugar:

The screenshot shows the Dialogflow console interface. At the top, there is a dark blue header with the Dialogflow logo, navigation links for 'API & DOCS' and 'PRICING', and a 'GO TO CONSOLE' button. The main content area is titled 'VirtualDoctorTest'. Below the title, there is a text prompt: 'Use following code to integrate this agent into your site:'. A pink arrow points from this text to a code block containing an iframe embed code. To the right of the code block is a preview of the chatbot interface, which includes a header with the agent name and a 'POWERED BY Dialogflow' logo, a large white chat area, and a bottom input field with the placeholder text 'Ask something...' and a microphone icon.

Neste exemplo o código HTML obtido foi:

```
<iframe width="350" height="430" allow="microphone;"  
src="https://console.dialogflow.com/api-client/demo/embedded/43db4f0b-f649-49e1-a8b7-e58e3c96119d"></iframe>
```

6. Agora tem que se abrir a pasta **chatbotwebsite** novamente, e no ficheiro "**index.html**", por forma a se poder editar, ao invés de se abrir com o navegador, deve ser aberto com uma ferramenta diferente (notepad, notepad++, etc). A imagem exemplo abaixo refere-se a um macbook:



7. Agora procure no código (fazendo **CTRL + F**) a palavra **INSERT CHATBOT HERE** (“INSERIR CHATBOT AQUI”) e encontrará o pedaço de código que queremos usar:

- Devemos comentar o bloco `<h3>`, para que possamos adicionar a nossa imagem aqui. Para isso, devemos adicionar as tags HTML “`<!--`” antes do `<h3>` e a tag “`-->`” após o `</3>`. Assim:

```
<!-- <h3 class="display-4 text-white text-uppercase">INSERT CHATBOT HERE</h3> -->
```

- A adicionar o código do chatbot aqui, após a linha que acabamos de comentar. Deve se parecer com algo assim:

```
<!-- Header Start -->
<div class="container-fluid page-header">
  <div class="container">
    <div class="d-flex flex-column align-items-center justify-content-center" style="min-height: 700px">
      <!-- <h3 class="display-4 text-white text-uppercase">INSERT CHATBOT HERE</h3> -->
      <iframe width="350" height="430" allow="microphone;" src="https://console.dialogflow.com/api-client/demo/embedded/43db4f0b-f649-49e1-a8b7-e58e3c96119d"></iframe>
    </div>
  </div>
</div>
<!-- Header End -->
```

8. Depois de gravar as alterações acima, agora temos o chatbot inserido site e podemos interagir novamente com ele. 😊 !!

