

# Como Desenvolver um Chatbot Educacional com Watson Assistant

Este artigo fornece um guia detalhado para criar um chatbot educacional usando o IBM Watson Assistant.

O objetivo é permitir que o chatbot consulte informações sobre escolaridade em estados, municípios e escolas, utilizando integrações com APIs, documentos ou bases de dados funcionais.

Com um design centrado no desempenho, apresentamos exemplos práticos para implementar soluções escaláveis e eficientes.

Com este guia, você pode criar um chatbot funcional e escalável para atender às necessidades educacionais.

A integração com APIs, documentos ou OCR garante que o sistema forneça respostas precisas.

Os exemplos de código e as ferramentas apresentadas ajudam a criar soluções robustas e otimizadas para diferentes cenários.

## 1. Configurações Iniciais

### 1.1. Criar uma Conta no IBM Cloud

1. Acesse o [IBM Cloud](#) e registre-se gratuitamente.
2. Busque por **Watson Assistant** no catálogo de serviços e ative o serviço.

### 1.2. Configurar o Watson Assistant

1. Crie um novo **Assistente** no painel do Watson Assistant.
2. Adicione um **Workspace** para configurar intenções, entidades e fluxos de conversação.



## 2. Planejamento das Funcionalidades

Defina os recursos que o chatbot deve oferecer, como:

- Consultar escolas públicas ou privadas.
- Informar sobre estados e municípios.
- Fornecer detalhes como número de alunos, localização e tipos de ensino.

**Exemplo de Cenário:** O usuário pergunta: "Quais escolas públicas têm ensino médio em São Paulo?" O chatbot deve responder com uma lista filtrada de escolas relevantes.

## 3. Estrutura do Chatbot

### 3.1. Intenções (Intents)

Intenções representam o objetivo do usuário. Exemplos:

- **ConsultarEscolas:** "Quais escolas existem no Rio de Janeiro?"
- **DetalhesEscola:** "Qual o número de alunos da escola XYZ?"

### 3.2. Entidades (Entities)

Entidades são elementos específicos reconhecidos na entrada do usuário. Exemplos:

- @TipoEscola: pública, privada.
- @Estado: São Paulo, Rio de Janeiro.
- @Municipio: Nome do município.

### 3.3. Fluxo de Diálogo

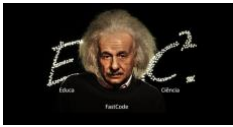
Crie um diálogo que responda com base nas intenções e entidades detectadas. Exemplo:

*Se intenção = ConsultarEscolas*

*E @Estado = São Paulo*

*E @TipoEscola = Pública*

*Responder: "Aqui estão as escolas públicas de ensino médio em São Paulo: ..."*



## 4. Integração com Dados Externos

### 4.1. Utilizando uma Base de Dados

1. **Estrutura da Base de Dados:**
  - **Tabela Estados:** ID, Nome.
  - **Tabela Municípios:** ID, Nome, EstadoID.
  - **Tabela Instituições:** ID, Nome, Tipo (Pública/Privada), MunicípioID, Nível de Ensino.
2. **Importação de Dados:** Obtenha informações do INEP e importe para o banco.
3. **API Backend:** Desenvolva uma API para consultas dinâmicas.
  - **Endpoint exemplo:** /consultar?estado=Sao%20Paulo&nivel=medio.

### 4.2. Integração com API

#### Exemplo de Integração com Python:

```
import requests

def consultar_escolas(estado, nivel):
    url = f"http://seu-backend/consultar?estado={estado}&nivel={nivel}"
    response = requests.get(url)
    return response.json()

estado = "São Paulo"
nivel = "médio"
escolas = consultar_escolas(estado, nivel)
print(escolas)
```

## 5. Pesquisa em Documentos e Imagens

### 5.1. Configuração para Leitura de PDFs

Use bibliotecas especializadas para extrair texto de arquivos PDF. Uma das opções mais robustas é o uso da biblioteca pdfplumber em Python.

#### Exemplo de Leitor de PDF:

```
import pdfplumber

def extrair_texto_pdf(caminho_pdf):
    texto_extraido = ""
    with pdfplumber.open(caminho_pdf) as pdf:
        for pagina in pdf.pages:
            texto_extraido += pagina.extract_text()
    return texto_extraido
```



```
# Exemplo de uso
caminho = "seu_arquivo.pdf"
texto = extrair_texto_pdf(caminho)
print(texto)
```

## 5.2. OCR para Imagens

Se os dados estão em imagens dentro de PDFs ou arquivos separados, use OCR (Reconhecimento Óptico de Caracteres). A biblioteca pytesseract é uma boa opção.

### Exemplo de Uso do Tesseract:

```
from PIL import Image
import pytesseract

# Configuração do caminho do executável Tesseract
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

def extrair_texto_imagem(caminho_imagem):
    texto = pytesseract.image_to_string(Image.open(caminho_imagem))
    return texto

# Exemplo de uso
texto_imagem = extrair_texto_imagem("imagem_exemplo.png")
print(texto_imagem)
```

## 5.3. Integração com o Watson Assistant

Depois de extrair o texto, envie-o como entrada para o Watson Assistant:

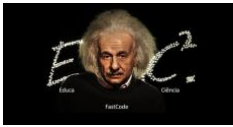
```
from ibm_watson import AssistantV2
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

authenticator = IAMAuthenticator('sua-chave-de-api')
assistant = AssistantV2(
    version='2021-11-27',
    authenticator=authenticator
)
assistant.set_service_url('sua-url-do-servico')

texto_pdf = extrair_texto_pdf("seu_arquivo.pdf")

response = assistant.message_stateless(
    assistant_id='seu-assistente-id',
    input={'text': texto_pdf}
).get_result()

print(response)
```



## 6. Testes e Validação

### 6.1. Testes Automatizados

Crie scripts para simular cenários reais. **Exemplo em Python:**

```
from ibm_watson import AssistantV2
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

authenticator = IAMAuthenticator('sua-chave-de-api')
assistant = AssistantV2(
    version='2021-11-27',
    authenticator=authenticator
)
assistant.set_service_url('sua-url-do-servico')

def teste_chatbot(mensagem):
    response = assistant.message_stateless(
        assistant_id='seu-assistente-id',
        input={'text': mensagem}
    ).get_result()
    return response

testes = [
    "Concluí o ensino médio em 2015 na Escola Estadual ABC.",
    "Terminei a faculdade em 2020 na Universidade XYZ.",
    "Não concluí o ensino médio."
]

for teste in testes:
    resposta = teste_chatbot(teste)
    print(f"Pergunta: {teste}\nResposta: {resposta['output'][0]['generic'][0]['text']}\n")
```

### 6.2. Monitoramento

1. Verifique os logs do Watson Assistant.
2. Analise falhas de reconhecimento de intenções e erros de diálogo.

Com este guia, você pode criar um chatbot funcional e escalável para atender às necessidades educacionais. A integração com APIs, documentos ou OCR garante que o sistema forneça respostas precisas. Os exemplos de código e as ferramentas apresentadas ajudam a criar soluções robustas e otimizadas para diferentes cenários.

**EducaCiência FastCode para a comunidade**