

# DESENVOLVENDO UMA CHAT COM INTELIGÊNCIA ARTIFICIAL EM PHP ACESSANDO A API DA OPENAI E TREINANDO O CHATGPT SOBRE MINHA EMPRESA

Autor: Ernesto Villafuerte Oyola

Mestre em Ciências de Computação – Inteligência Artificial  
COPPE/UFRJ

Diretor da Aomega Treinamento e Consultoria

[www.aomega.com.br](http://www.aomega.com.br)

12/05/2023

Projeto no GITHUB:

<https://github.com/evillafuerte/aomegachatgpt>



Neste artigo ensinarei como acessar a API da OpenAI dentro de nossa aplicação. Mas eu quero que ele responda mensagens específicas

sobre a minha empresa. Para isso eu preciso “ensinar” ele, passando informações sobre minha empresa que eu quero que ele esteja apto para responder a um visitante do meu site.

Eu desenvolvo há vários anos em PHP e gosto muito. O PHP já nasceu sendo uma linguagem de desenvolvimento Web , não preciso aprender muitas tecnologias nem frameworks para criar uma aplicação Web além de HTML ,CSS e o próprio PHP. Assim que foi disponibilizado a API criada pela OpenAI para fazer uso do ChatGPT dentro de várias linguagens meu interesse foi saber se já existia alguma coisa para o PHP. O próprio site da OpenAPI exibe uma forma de se conectar com a API via Curl no PHP. Mas não consegui fazê-la funcionar e vi que a solução adotada por muitos passava longe do uso do Curl e sim pelo uso de uma biblioteca desenvolvida por uma comunidade de software livre bastante divulgada, no projeto “openai\_php/client” que permite criar no PHP um cliente para a OpenAI API.

Vamos precisar trabalhar com o PHP versão 8.1 ou superior e ter o Composer instalado em nossa máquina.



Para baixar e instalar o Composer na sua máquina você pode acessar o link <https://getcomposer.org/download/> .

Instalar o Composer é muito simples (em ambos os casos você deve ter o comando PHP disponível no terminal):

- Se você estiver no Windows, só precisa baixar o arquivo composer-setup.exe que adicionará automaticamente o comando Composer ao PATH do seu sistema.
- No UNIX/Linux você deve baixar composer.phar digitando a linha `curl -sS https://getcomposer.org/installer | php` no seu terminal. Você pode então executar o compositor utilizando `php composer.phar`

Também precisaremos utilizar nossa chave (Key) para autorizar o nosso acesso à OpenAI API. Se você não possui uma Key precisa obter uma se registrando em <https://chat.openai.com>. O Composer é amplamente utilizado pela comunidade de desenvolvedores PHP para instalar pacotes adicionais no PHP. Em nosso caso, precisaremos do Composer para instalar o pacote do cliente da OpenAI API, denominado de OPENAI\_PHP/CLIENT. A seguir o comando para instalar o pacote na pasta raiz de nosso site.

```
composer require openai-php/client
```

Agora vamos criar um arquivo PHP, por exemplo `aomegachatgpt_000.php`, para criar o cliente do OPENAI API em nosso código:

```
require "vendor/autoload.php";
$yourApiKey = "xp-Tuskdk2993wo\eldldlXQ"; // chave fake
$client = OpenAI::client($yourApiKey);

$response = $client->chat()->create([
    'model' => 'gpt-3.5-turbo',
    'messages' => [
        ['role' => 'user',
         'content' => 'Qual é a atividade principal da
"Aomega Treinamento e Consultoria" cujo site está em
"http://www.aomega.com.br" ?'],
    ],
]);
```

```

foreach ($response->choices as $result) {
    $result->index; // 0
    $result->message->role; // 'assistant'
    $result->message->content; // '\n\nHello there! How
can I assist you today?'
    $result->finishReason; // 'stop'
}

echo $result->message->content;

```

**\$yourApiKey** - variável para armazenar nossa KEY pessoal obtida após nosso registro no site da OpenAI. A nossa KEY não pode ficar visível para o mundo, lembre que o uso de nossa KEY da OpenAI API é pago se exceder o limite gratuito para testes, e o valor pago varia de acordo com o tráfego acumulado de nossa aplicação. Algumas pessoas armazenam a KEY dentro de um arquivo de ambiente do tipo .ENV mas esse arquivo poderia ser acessado por alguém externamente e “roubar” nossa KEY. Então é melhor nossa KEY estar apenas no servidor, no caso dentro de nosso script PHP, assim ela fica mais protegida.

**\$client** – cliente da API obtido instanciando o método client da classe OpenAI do pacote instalado por nós utilizando o Composer.

Existem vários modelos disponíveis na API da OpenAI que podemos utilizar segundo a nossa conveniência. Não vou entrar aqui em detalhes sobre cada um desses modelos, alguns podem ser mais rápidos e outros mais poderosos, dependendo a finalidade que desejamos obter. Entre os modelos temos : gpt-3.5-turbo, da-vinci, curie, babbage, ada. O modelo da-vinci costuma ser mais rápido e com menor custo e atender em muitos casos as nossas necessidades. Em nosso exemplo estamos usando o modelo gpt-3.5-turbo mais versátil e poderoso, mais apropriado para um chat, embora acarrete num custo levemente maior.

**\$response** - Objeto gerado após a criação com nosso cliente de um chat com o modelo gpt.3-5.turbo.

**messages** – array atributo de nosso cliente contendo a chave “role” ou papel, que no caso é o usuário interagindo (“user”), a chave “content” com o texto de nosso prompt com nossa solicitação ao chatgpt.

E logo depois inseri o texto de minha solicitação:

Qual é a atividade principal da "Aomega Treinamento e Consultoria" cujo site está em <http://www.aomega.com.br> ?

Para capturar e visualizar a resposta do ChatGPT a minha solicitação posso acrescentar o seguinte código no final:

```
foreach ($response->choices as $result) {
    $result->index; // 0
    $result->message->role; // 'assistant'
    $result->message->content; // '
    $result->finishReason; // 'stop'
}
// echo echo $result->message->content; resposta do
chatgpt a ser exibida na tela de resposta ao usuário.
```

Repare que depois de instanciado nosso cliente com nosso prompt, o valor do "role" muda de "user" para "assistant". O assistente é que vai fornecer a resposta no atributo "content" atualizando o valor desta com a resposta do chatgpt à solicitação do usuário.

Bom, com esse código já dá para testar, mas reparem que eu forneci no código o texto do prompt, ele está fixo. O ideal seria ter uma tela com um formulário para que o usuário possa digitar o texto de sua solicitação em um formulário de um script PHP contendo o código mostrado anteriormente e exibisse a resposta na tela.

Vamos incrementar nosso script que vou denominar de aomegachatgpt2.php acrescentando o código HTML necessário no final.

```

<?php // código PHP mostrado anteriormente ?>
<!DOCTYPE html>
<html>
<head>
    <title>Converse com o Aomega-GPT e esclareça suas
dúvidas sobre nossos treinamentos</title>
</head>
<body>
    <h1>Converse com o Aomega-GPT e esclareça suas
dúvidas sobre nossos treinamentos</h1>
    <form method="POST" action="aomegachatgpt2.php">
        <?php if (isset($result)) {
            echo $result->message->content."<br/><br/>";
        } ?>
        <h3>O que você gostaria de saber?</h3><br>
        <textarea id="duvida" name="duvida" rows="4"
cols="50">
        </textarea><br>
        <input type="submit" value="Enviar">
    </form>
</body>
</html>

```

Agora nosso código PHP está perfeito, temos o ChatGPT embutido em nossa aplicação funcionando. Mas não é isso que quero apenas. Quero que o ChatGPT possa responder perguntas específicas sobre minha empresa...

O problema é que o ChatGPT não acessa a web e sua base de dados não engloba conhecimento até o dia atual. Além disso provavelmente ele não poderá responder perguntas muito específicas sobre minha empresa, apenas de cunho geral com sorte. Entretanto o modelo de Chat da OpenAI API permite que possamos “ensinar” o ChatGPT sobre minha empresa, basta para isso passar para ele as informações que gostaríamos que fossem disponibilizadas. Lembre que ele tem

inteligência artificial e ele saberá fornecer as informações corretas mesmo que perguntado de maneiras diferentes, ele vai “entender” o que você quer e procurar a resposta correta dentro das informações que foram “ensinadas” para ele. Isso será um processo interativo, de “aprendizado”, você irá aperfeiçoar o texto com as informações fornecidas a ele várias vezes até atingir o ideal, quando você ficar satisfeito com o nível de qualidade das respostas fornecidas pelo ChatGPT.

Vamos acrescentar então nossa “base de conhecimento” sobre nossa empresa:

```
$empresa = "Aomega Treinamento e Consultoria";  
$url = "http://www.aomega.com.br";  
$empresa_nick = "Aomega";  
  
$conhecimento = "Nós da $empresa, também conhecida como  
$empresa_nick,  
oferecemos serviços de treinamento em Excel níveis  
básico, intermediário e avançado,  
Excel VBA, Análise de Dados com Power BI, SQL e  
Inteligência  
artificial, engenharia de PROMPTS no CHATGPT e uso da  
API.  
Ministramos treinamentos online individuais e para grupos  
via  
videoconferência ou presencias para empresas (in company)  
.  
A URL de nosso site é $url .  
Para solicitar informações adicionais entre  
em contato conosco através de nosso formulário na página  
FALE CONOSCO .  
A duração dos treinamentos pode ser consultada nos links  
de aulas  
de nosso site. A duração pode variar para os treinamentos  
personalizados. Entre  
em contato conosco através de nosso formulário na página  
FALE CONOSCO ";
```

```
$prompt = "Responder a dúvida $duvida com base nas  
informações  
a seguir: $conhecimento ";
```

Criamos algumas variáveis como \$empresa e \$empresa\_nick e \$url para armazenar o nome de nossa empresa, o nome abreviado e a url de nosso website.

Nossa base de conhecimento ficou armazenada na variável \$conhecimento.

A variável \$duvida irá armazenar o texto com a dúvida/solicitação feita pelo usuário. Ele irá armazenar o valor do campo "duvida" do meu formulário HTML que criei após ser submetido o formulário.

A variável \$prompt será utilizada durante a criação dos atributos do objeto cliente do tipo Chat da OpenAI API e recebe como valor um texto instruindo ao Chat que deve responder as dúvidas com base nas informações da base de conhecimento que armazenei na variável \$conhecimento. Então o Chat irá procurar preferencialmente as respostas dentro de minha base de conhecimento, será o primeiro lugar onde ele vai procurar as respostas.

A seguir vou mostrar o trecho do código PHP criado que será modificado para incorporar o uso da variável \$prompt.

```
$response = $client->chat()->create([  
    'model' => 'gpt-3.5-turbo',  
    'max_tokens' => 300,  
    'temperature' => 0.3,  
    'messages' => [  
        ['role' => 'user',  
         'content' => $prompt],  
    ],  
]);
```



```
// $resposta = $result->message->content;
```

Ou seja o valor de \$prompt é atribuído ao atributo “content” do meu cliente \$response criado.

Finalmente após melhorar o HTML do meu código e acrescentar CSS e imagens para fazer uma personalização para ficar com a identidade visual do website de minha empresa ficou visualmente assim:

**AOMEGA**  
CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL

[Inicio](#) [Excel](#) [Power BI](#) [SQL](#) [ChatGPT](#) [Outros Treinamentos](#) [Fale Conosco](#)

## AOMEGA-GPT - A Inteligência Artificial da Aomega

### Converse com o Aomega-GPT e esclareça suas dúvidas sobre nossos treinamentos

O que você gostaria de saber?

Vocês ministram treinamento online? Seria por videoconferência?

[\[VOLTAR\]](#)

[Inicio](#) [Clientes](#) [Canal YouTube](#) [Blog](#) [Depoimentos](#) [Galeria de Fotos](#)

[Aviso de Privacidade](#)

E aqui a tela com a resposta do nosso Chat com inteligência artificial:



## AOMEGA-GPT - A Inteligência Artificial da Aomega

### Converse com o Aomega-GPT e esclareça suas dúvidas sobre nossos treinamentos

**P: Vocês ministram treinamento online? Seria por videoconferência?**

Sim, a Aomega Treinamento e Consultoria ministra treinamentos online tanto para indivíduos quanto para grupos, por meio de videoconferência. Além disso, a empresa também oferece treinamentos presenciais para empresas (in company). Para obter mais informações sobre os treinamentos oferecidos e a duração de cada um, é possível acessar os links de aulas no site da empresa ou entrar em contato por meio do formulário na página FALE CONOSCO.

O que você gostaria de saber?

Enviar

[\[VOLTAR\]](#)

Funcionando perfeitamente e respondendo de acordo com a base de conhecimento com a qual treinei o Chat. Arquivos do projeto disponíveis no Github em <https://github.com/evillafuerte/aomegachatgpt>.