

chatbot con voz en la API de Gemini.

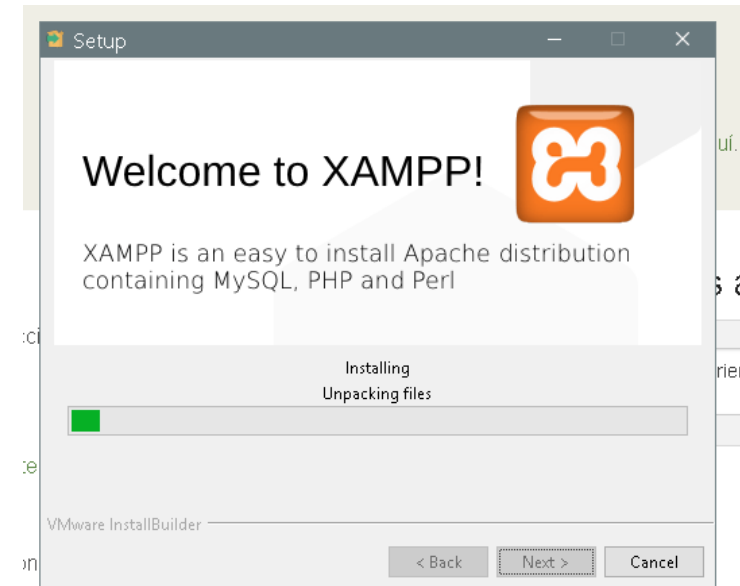
2024

https://github.com/profejavierc/chat_gemini_con_voz_xampp/tree/main

<https://www.apachefriends.org/es/download.html>

XAMPP para **Windows** 8.0.30, 8.1.25 & 8.2.12

Versión		Suma de comprobación			Tamaño
		md5	sha1		
8.0.30 / PHP 8.0.30	¿Qué está incluido?.	md5	sha1	Descargar (64 bit)	144 Mb
8.1.25 / PHP 8.1.25	¿Qué está incluido?.	md5	sha1	Descargar (64 bit)	148 Mb
8.2.12 / PHP 8.2.12	¿Qué está incluido?.	md5	sha1	Descargar (64 bit)	149 Mb



shutdown /r

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

9:49:57 a.m. [main] Initializing Modules
9:49:57 a.m. [mysql] Problem detected!
9:49:57 a.m. [mysql] Port 3306 in use by "Unable to open process"
9:49:57 a.m. [mysql] MySQL WILL NOT start without the configu
9:49:57 a.m. [mysql] You need to uninstall/disable/reconfigure the
9:49:57 a.m. [mysql] or reconfigure MySQL and the Control Panel
9:49:57 a.m. [main] Starting Check-Timer
9:49:57 a.m. [main] Control Panel Ready

Crear la carpeta en hdocs

Gemini_Chat_bot

xampp

Inicio Compartir Vista

> Este equipo > Disco local (C:) > xampp

Nombre
anonymous
apache
cgi-bin
contrib
FileZillaFTP
htdocs
img
install
licenses

> Este equipo > Disco local (C:) > xampp > htdocs

Nombre
dashboard
Gemini_Chat_bot
img
webalizer
xampp
applications
bitnami
favicon
index

<https://ai.google.dev/gemini-api?hl=es-419>




Gemini

Integra fácilmente en tus apps el modelo de IA más grande y potente de Google

[Obtén una clave de API en Google AI Studio](#)

[Leer la documentación de la API](#)

 [Prueba la IA lista para empresas de Google Cloud >](#)

Gemini 1.5 Pro 2M context window, code execution capabilities, and Gemma 2 are now available. [Learn more.](#)

inglés

español



Got it

Google Translate

Google AI Studio

Get API key

Create new prompt

New tuned model

My library

Allow Drive access

Getting started

Documentation

Prompt gallery

Gemini cookbook

Discourse forum

Build with Vertex AI on
Google Cloud

Settings



Obtén una clave de API

Claves de APIs

Puedes crear un proyecto nuevo si aún no tienes uno o agregar claves de APIs a un proyecto existente. Todos los proyectos están sujetos a las [Condiciones del Servicio de Google Cloud Platform](#), que aceptas cuando creas un proyecto nuevo. El uso de la API de Gemini y de Google AI Studio está sujeto a las [Condiciones del Servicio de la API de Gemini](#).

Usa tus claves de APIs de forma segura. No las compartas ni las incorpores en código que el público pueda ver.

Si usas la API de Gemini desde un proyecto que tiene la facturación habilitada, tu uso estará sujeto a [precios de pago por uso](#).

Crear clave de API

Tus claves de APIs se enumeran a continuación. También puedes ver y administrar tu proyecto y tus claves de APIs en Google Cloud.

Número del proyecto	ID del proyecto	Clave de API	Fecha de creación	Plan
---------------------	-----------------	--------------	-------------------	------

Crea una clave de API para ver tus proyectos

S


nuevo si aún no tienes uno o agregar claves de APIs a un proyecto existente.

gura. No las compartas ni las incorpores en código que el público

U




Copiar



```
C:\xampp\htdocs\Gemini_Chat_bot\  
| index.php  
| script.js  
| response.php  
| composer.json
```

index.php:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Gemini AI Chatprofejavierc</title>  
</head>  
<body>  
  <h1>Gemini AI Chatprofejavierc</h1>  
  <input type="text" id="userInput" placeholder="Escriba su pregunta">  
  <button onclick="startSpeechRecognition()">Hablar</button>  
  <button onclick="generateResponse()">Enviar</button>  
  <div id="responseDiv"></div>  
  
  <script src="script.js"></script>  
</body>  
</html>
```



```
C:\xampp\htdocs\Gemini_Chat_bot\  
| index.php  
| script.js  
| response.php  
| composer.json
```

```
<?php  
header('Content-Type: application/json');
```

```
// Configura tu clave API  
$api_key = 'your APY KEY';
```

```
// Obtén la entrada del frontend  
$data = json_decode(file_get_contents('php://input'), true);  
$text = $data['text'] ?? '';
```

```
if (empty($text)) {  
    echo json_encode(['error' => 'No se proporcionó texto de entrada']);  
    exit;  
}
```

```
// Configura la solicitud a la API de Gemini  
$url = 'https://generativelanguage.googleapis.com/v1beta/models/gemini-pro:generateContent?key=' .  
$api_key;  
$data = [  
    'contents' => [  
        [  
            'parts' => [  
                ['text' => $text]  
            ]  
        ]  
    ]  
];
```



API
key

response.php:

```
// Inicializa cURL  
$curl = curl_init();
```

```
// Configura las opciones de cURL  
curl_setopt_array($curl, [  
    CURLOPT_URL => $url,  
    CURLOPT_RETURNTRANSFER => true,  
    CURLOPT_POST => true,  
    CURLOPT_POSTFIELDS => json_encode($data),  
    CURLOPT_HTTPHEADER => [  
        'Content-Type: application/json'  
    ]  
]);
```

```
// Envía la solicitud y obtén la respuesta  
$response = curl_exec($curl);
```

```
// Verifica si hubo errores  
if (curl_errno($curl)) {  
    echo json_encode(['error' => 'Error en cURL: ' . curl_error($curl)]);  
} else {  
    // Decodifica la respuesta JSON  
    $result = json_decode($response, true);
```

```
    // Extrae el texto de la respuesta  
    if (isset($result['candidates'][0]['content']['parts'][0]['text'])) {  
        echo json_encode(['text' =>  
$result['candidates'][0]['content']['parts'][0]['text']]);  
    } else {  
        echo json_encode(['error' => 'No se pudo obtener una respuesta válida  
de la API', 'raw_response' => $response]);  
    }  
}
```

```
// Cierra la conexión cURL  
curl_close($curl);
```

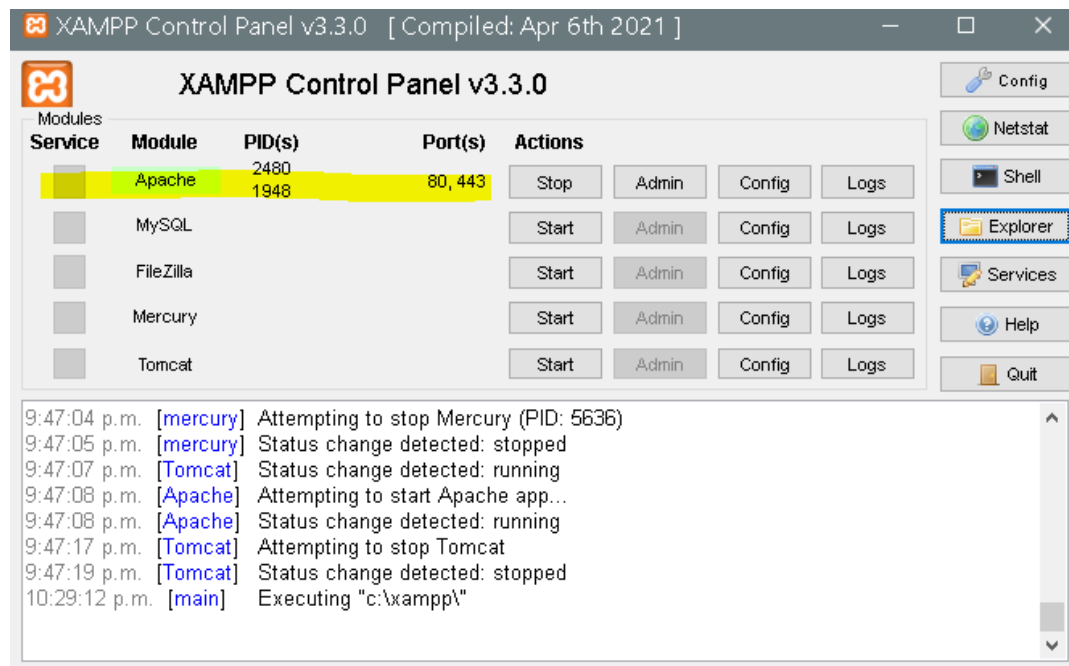



```
C:\xampp\htdocs\Gemini_Chat_bot\  
| index.php  
| script.js  
| response.php  
| composer.json
```

script.js:

```
let recognition;  
  
function startSpeechRecognition() {  
  if ('webkitSpeechRecognition' in window) {  
    recognition = new webkitSpeechRecognition();  
    recognition.lang = 'es-ES';  
    recognition.onresult = function(event) {  
      const transcript = event.results[0][0].transcript;  
      document.getElementById('userInput').value = transcript;  
      generateResponse();  
    };  
    recognition.start();  
  } else {  
    alert('Tu navegador no soporta reconocimiento de voz.');  }  
}  
  
function speak(text) {  
  if ('speechSynthesis' in window) {  
    const utterance = new SpeechSynthesisUtterance(text);  
    utterance.lang = 'es-ES';  
    speechSynthesis.speak(utterance);  
  } else {  
    console.log("La síntesis de voz no está soportada en este navegador.");  
  }  
}  
  
function generateResponse() {  
  const userInput = document.getElementById('userInput').value;  
  const responseDiv = document.getElementById('responseDiv');  
  fetch('response.php', {  
    method: 'POST',  
    headers: {  
      'Content-Type': 'application/json',  
    },  
    body: JSON.stringify({ text: userInput })  
  })
```

```
.then(response => {  
  if (!response.ok) {  
    throw new Error(`HTTP error! status: ${response.status}`);  
  }  
  return response.text();  
})  
.then(text => {  
  try {  
    const data = JSON.parse(text);  
    if (data.error) {  
      responseDiv.innerHTML = `Error: ${data.error}`;  
      if (data.raw_response) {  
        console.log('Respuesta cruda de la API:', data.raw_response);  
      }  
    } else if (data.text) {  
      responseDiv.innerHTML = data.text;  
      speak(data.text);  
    } else {  
      responseDiv.innerHTML = 'Respuesta inesperada del servidor';  
    }  
  } catch (e) {  
    console.error('Error al parsear JSON:', e);  
    responseDiv.innerHTML = `Error al parsear la respuesta del  
servidor: ${text}`;  
  }  
})  
.catch(error => {  
  console.error('Error:', error);  
  responseDiv.innerHTML = `Error: ${error.message}`;  
});  
}
```



Create a folder inside xampp on the local drive and mount the three files. Initialize by setting xampp to start in Apache and then open the browser with the URL localhost/ and the URL of where index.php is located.

