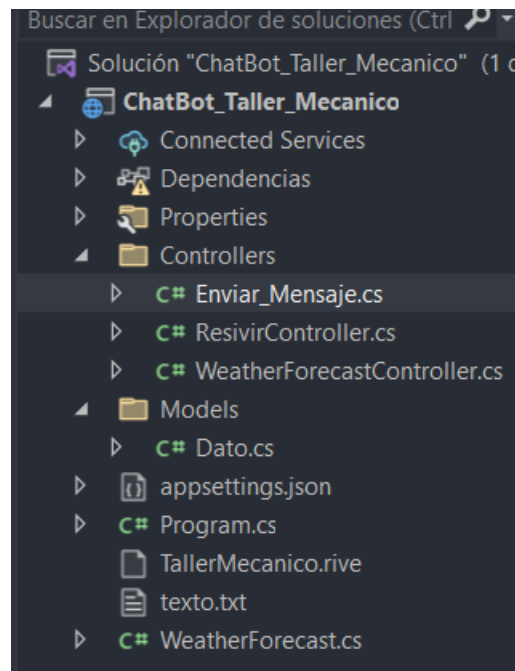


Explicación de la creación de Chatbot para un taller de mecánica automotriz

Nombre: Jose Luis Yanahuaya Mamani

Base de datos II



El código muestra un controlador de una aplicación web utilizando el framework ASP.NET Core.

El controlador se llama "Enviar_Mensaje" y contiene un método llamado "enviaAsync". A continuación, se proporciona una explicación detallada del código:

Inclusion de Bibliotecas:

```
1 using System.Net.Http;  
2 using Microsoft.AspNetCore.Mvc;  
3 using System.Net.Http.Headers;  
4  
5
```

En esta sección, se importan las bibliotecas necesarias para trabajar con HTTP, como HttpClient para realizar solicitudes y respuestas HTTP, y HttpHeaders para establecer encabezados de solicitud.

Declaración del espacio de nombres y clase:

```

namespace ChatBot_Taller_Mecanico.Controllers
{
    0 referencias
    public class Enviar_Mensaje : ControllerBase
    {
        //RECIBIMOS LOS DATOS DE VALIDACION VIA GET
    }
}

```

El código se coloca dentro de un espacio de nombres llamado "ChatBot_Taller_Mecanico.Controllers" y la clase se llama "Enviar_Mensaje". La clase hereda de ControllerBase, que es una clase base proporcionada por ASP.NET Core para los controladores.

Método enviaAsync:

```

//RECIBIMOS LOS DATOS DE VALIDACION VIA GET
[HttpGet]
//DENTRO DE LA RUTA envia
[Route("envia")]
//RECIBIMOS LOS PARAMETROS QUE NOS ENVIA WHATSAPP PARA VALIDAR NUESTRA URL
0 referencias

```

Este es el método principal del controlador y se configura para responder a las solicitudes HTTP GET en la ruta "/envia". El modificador "async" indica que el método se puede ejecutar de forma asincrónica y el tipo de retorno "Task" indica que el método puede devolver un resultado de manera asincrónica.

Variables:

```

//identificador de acceso temporal
string token = "EAADNbOrZALSMBAOZCkIWA0Hq0U910IKZCsX0Y0UvsvTjU8bCMGRmxaflTzKHqUDFDpG0U5Kxltzq4Na
//Identificador de número de teléfono
string idTelefono = "110623718730841";
//Nuestro telefono
string telefono = "59163147424";
HttpClient client = new HttpClient();

```

Estas variables almacenan información utilizada para realizar la solicitud HTTP. "token" representa el token de acceso necesario para autenticar la solicitud, "idTelefono" es el identificador

del número de teléfono y "telefono" es el número de teléfono de destino al que se enviará el mensaje.

Configuración de HttpClient y HttpRequestMessage:

```
string telefono = "59163147424";  
HttpClient client = new HttpClient();  
HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Post, "https://graph.facebook.com/v15.0/" + idTelefono  
request.Headers.Add("Authorization", "Bearer " + token);
```

Se crea una instancia de HttpClient para enviar solicitudes HTTP y se crea un objeto HttpRequestMessage para configurar la solicitud. Se especifica el verbo HTTP como POST y se proporciona la URL de destino, que incluye el identificador del número de teléfono.

Configuración del contenido de la solicitud:

```
request.Content = new StringContent("{\"messaging_product\": \"whatsapp\", \"recipient_type\": \"individual\", \"to\": \"  
request.Content.Headers.ContentType = new MediaTypeHeaderValue(\"application/json\");
```

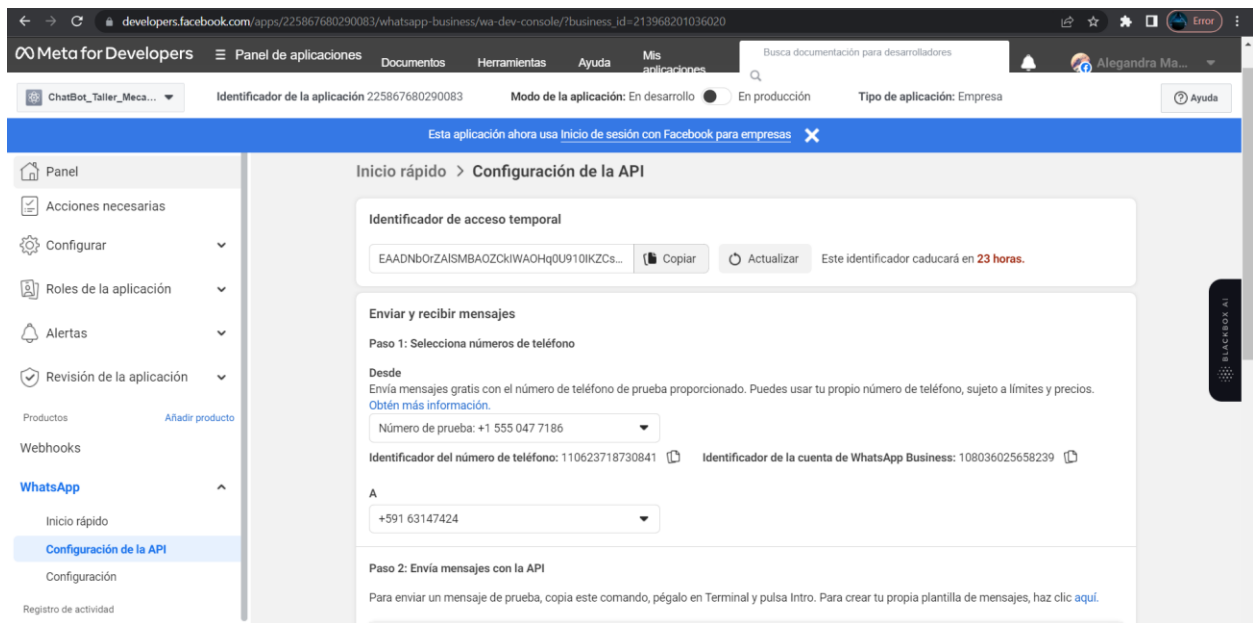
Se configura el contenido de la solicitud en formato JSON. El contenido contiene información sobre el producto de mensajería (en este caso, WhatsApp), el tipo de destinatario (individual), el número de teléfono de destino y el texto del mensaje.

Envío de la solicitud y obtención de la respuesta:

```
HttpResponseMessage response = await client.SendAsync(request);  
//response.EnsureSuccessStatusCode();  
string responseBody = await response.Content.ReadAsStringAsync();
```

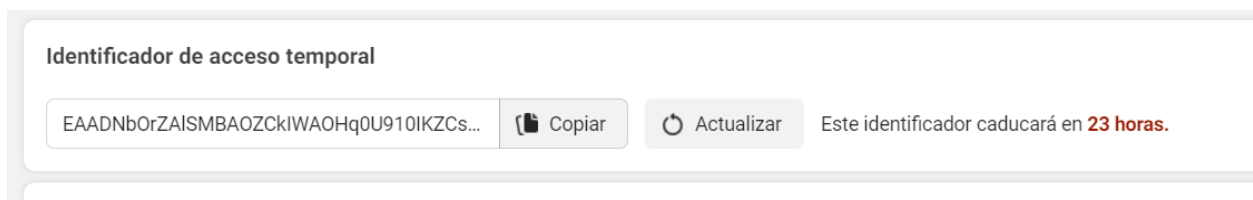
Se envía la solicitud asincrónicamente utilizando el método SendAsync de HttpClient y se obtiene la respuesta como un objeto HttpResponseMessage. Luego, se lee el contenido de la respuesta como una cadena utilizando ReadAsStringAsync y se almacena en la variable "responseBody".

Este código muestra un controlador en ASP.NET Core que utiliza HttpClient para enviar una solicitud HTTP POST a la API de Graph de Facebook. La solicitud está destinada a enviar un mensaje de texto a un número de teléfono específico a través de WhatsApp. El controlador recibe los parámetros necesarios, configura la solicitud HTTP y envía la solicitud para su procesamiento.

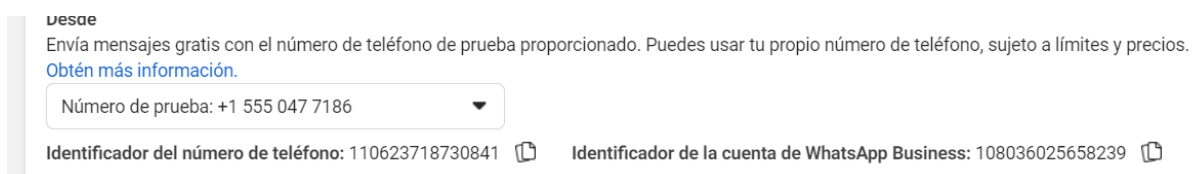


Sacamos los datos de la página <https://developers.facebook.com>

- Token



- idTelefono



- teléfono

A

+591 63147424

▼

Ejecutamos el código y nos manda lo siguiente:

ChatBot_Taller_Mecanico

1.0 OAS3

<http://localhost:5093/swagger/v1/swagger.json>

Enviar_Mensaje

GET /envia

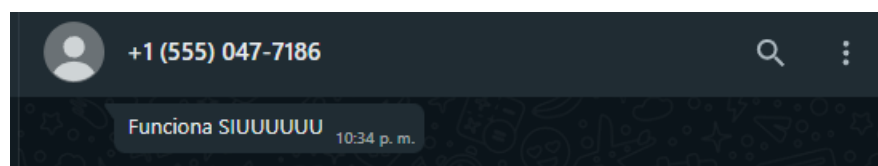
Parameters

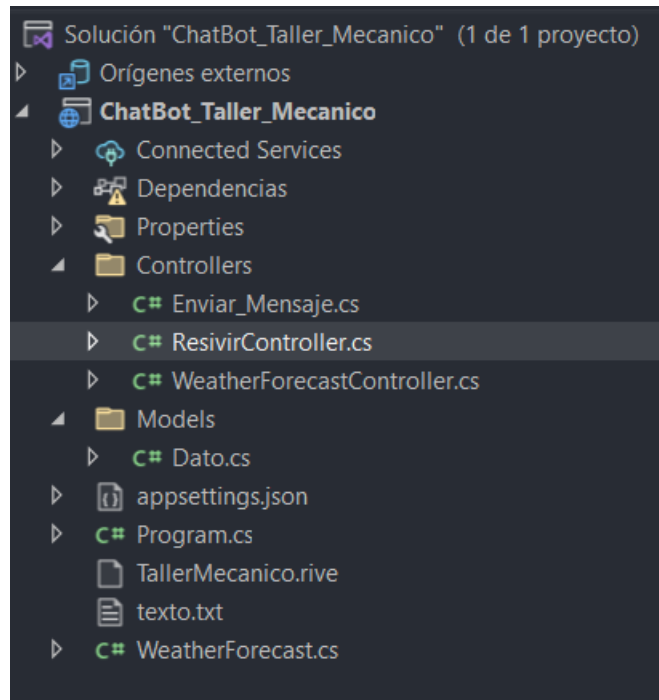
No parameters

Execute

Cancel

En donde si damos a ejecutar nos enviara un mensaje al número que le dimos al código





El código muestra un controlador de una aplicación web utilizando el framework ASP.NET Core. Este controlador se llama "ResivirController" y contiene dos métodos: "Webhook" y "datos". A continuación, se proporciona una explicación detallada del código:

Inclusión de bibliotecas:

```
Bot_Taller_Mecanico ChatBot_T
1  using System.Threading.Tasks;
2  using ChatBot_Taller_Mecanico.Models;
3  using Microsoft.AspNetCore.Mvc;
4  using System.Net;
5  using System.Net.Http.Headers;
6  using System.IO;
7  using RiveScript;
8
```

En esta sección, se importan las bibliotecas necesarias para trabajar con tareas asincrónicas, modelos de datos, controladores de MVC, HTTP, manipulación de archivos y RiveScript (un lenguaje de scripting para chatbots).

Declaración del espacio de nombres y clase:

```
namespace ChatBot_Taller_Mecanico.Controllers
{
    0 referencias
    public class ResivirController
    {
        //RECIBIMOS LOS DATOS DE VALIDACION VIA GET
    }
}
```

El código se coloca dentro de un espacio de nombres llamado "ChatBot_Taller_Mecanico.Controllers" y la clase se llama "ResivirController".

Método Webhook:

```
//RECIBIMOS LOS DATOS DE VALIDACION VIA GET
[HttpGet]
//DENTRO DE LA RUTA webhook
[Route("webhook")]
//RECIBIMOS LOS PARAMETROS QUE NOS ENVIA WHATSAPP PARA VALIDAR NUESTRA URL
0 referencias
public string Webhook(
    [FromQuery(Name = "hub.mode")] string mode,
    [FromQuery(Name = "hub.challenge")] string challenge,
    [FromQuery(Name = "hub.verify_token")] string verify_token
)
```

Este método es un controlador que responde a las solicitudes HTTP GET en la ruta "/webhook". Los parámetros de la consulta HTTP ("hub.mode", "hub.challenge" y "hub.verify_token") se reciben como argumentos de método. Si el valor de "verify_token" coincide con "hola", se devuelve el valor de "challenge". De lo contrario, se devuelve una cadena vacía.

Método datos:

```
//RECIBIMOS LOS DATOS DE VIA POST
[HttpPost]
//DENTRO DE LA RUTA webhook
[Route("webhook")]
//RECIBIMOS LOS DATOS Y LOS GUARDAMOS EN EL MODELO WebHookResponseModel
0 referencias
public dynamic datos([FromBody] WebHookResponseModel entry)
{
    //...
}
```

Este método es otro controlador que responde a las solicitudes HTTP POST en la ruta "/webhook". Los datos se reciben en el cuerpo de la solicitud y se deserializan en un objeto de tipo WebHookResponseModel utilizando el atributo [FromBody]. El método procesa los datos recibidos, como extraer el mensaje, el ID único y el número de teléfono del mensaje. Luego, crea una instancia de la clase Dato y realiza algunas operaciones, como cargar un archivo de configuración y obtener una respuesta utilizando RiveScript. También inserta registros en una base de datos y llama al método "enviaAsync" para enviar una respuesta al número de teléfono.

Método enviaAsync:

```
0 referencias
public async Task enviaAsync(string telefono, string mensaje)
{
    string token = "EAADNbOrZALSMBA0ZCkIWA0Hq0U910IKZCsX0Y0UvsvTjU8bCMGRmxafLTzKHqUDFDpG0U5Kxltzq4I...";
    //Identificador de número de teléfono
    string idTelefono = "110623718730841";
    HttpClient client = new HttpClient();
    HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Post, "https://graph.facebook.com/110623718730841");
    request.Headers.Add("Authorization", "Bearer " + token);
    string json = "{\"messaging_product\": \"whatsapp\", \"recipient_type\": \"individual\", \"to\": \"110623718730841\", \"message\": \"\"}";
    request.Content = new StringContent(json);
    request.Content.Headers.ContentType = new MediaTypeHeaderValue("application/json");
    HttpResponseMessage response = await client.SendAsync(request);
    //response.EnsureSuccessStatusCode();
    string responseBody = await response.Content.ReadAsStringAsync();
}
```

Este método realiza una solicitud HTTP POST asincrónica utilizando HttpClient para enviar un mensaje de respuesta a un número de teléfono específico. Utiliza el token de acceso, el

identificador del número de teléfono, el número de teléfono y el mensaje proporcionados como parámetros para configurar la solicitud. Luego, envía la solicitud y obtiene la respuesta.

Este código muestra un controlador en ASP.NET Core que maneja las solicitudes de un webhook de WhatsApp. El método "Webhook" se utiliza para validar la URL y el método "datos" procesa los datos recibidos del webhook, interactúa con un bot de RiveScript, guarda registros en una base de datos y envía respuestas a través de la API de Graph de Facebook.

Para su ejecución necesitamos una web para enviar y recibirlos mensajes ahí y nos redirija a nuestro WhatsApp:

```
ngrok http foo.dev:80           # tunnel to host:port instead of localhost
ngrok http https://localhost    # expose a local https server
ngrok tcp 22                    # tunnel arbitrary TCP traffic to port 22
ngrok tls --domain=foo.com 443  # TLS traffic for foo.com to port 443
ngrok start foo bar baz        # start tunnels from the configuration file

COMMANDS:
api                use ngrok agent as an api client
completion         generates shell completion code for bash or zsh
config             update or migrate ngrok's configuration file
credits            prints author and licensing information
diagnose           diagnose connection issues
help              Help about any command
http              start an HTTP tunnel
service           run and control an ngrok service on a target operating system
start             start tunnels by name from the configuration file
tcp              start a TCP tunnel
tls              start a TLS tunnel
tunnel            start a tunnel for use with a tunnel-group backend
update           update ngrok to the latest version
version          print the version string

OPTIONS:
--config strings  path to config files; they are merged if multiple
-h, --help       help for ngrok
-v, --version    version for ngrok

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\FROILAN MAMANI\Desktop>ngrok http 5093
```

Ejecutamos el comando NGROOK HTTP 5093

“5093” Es el puerto que nos da del local host al ejecutar el código

Nos brindará un link con el cual podemos trabajar y otro que nos servirá de controlador ya que a través de este podremos ver los mensajes que se reciben y se envían.

```
ngrok (Ctrl+C to quit)
Announcing ngrok's Kubernetes Ingress Controller: https://ngrok.com/s/k8s-ingress

Session Status      online
Account             Froilan (Plan: Free)
Version             3.3.1
Region              South America (sa)
Latency              -
Web Interface        http://127.0.0.1:4040
Forwarding            https://6bcf-2800-cd0-2b04-4000-7c74-20e5-12a9-7dee.ngrok-free.app -> http://localhost:5093

Connections
  ttl    opn    rt1    rt5    p50    p90
    0     0     0.00  0.00  0.00  0.00
```

Mandamos estos datos a Facebook para que nos pueda validar la web que creamos:

Inicio rápido > Configuración

Webhook

Para que se te envíe una alerta cuando recibas un mensaje o cuando cambie el estado de un mensaje, debes configurar un extremo de webhook para tu aplicación. Obtén más información [sobre cómo configurar un webhook](#).

URL de devolución de llamada ⓘ
https://7f8a-2800-cd0-2b04-4000-7c74-20e5-12a9-7dee.ngrok-fr...

Identificador de verificación ⓘ
.....

Editar

Campos de webhook ⓘ
messages

Administrar

Ejecución del código:

ChatBot_Taller_Mecanico 1.0 OAS3

http://localhost:5093/swagger/v1/swagger.json

Enviar_Mensaje

GET /envia

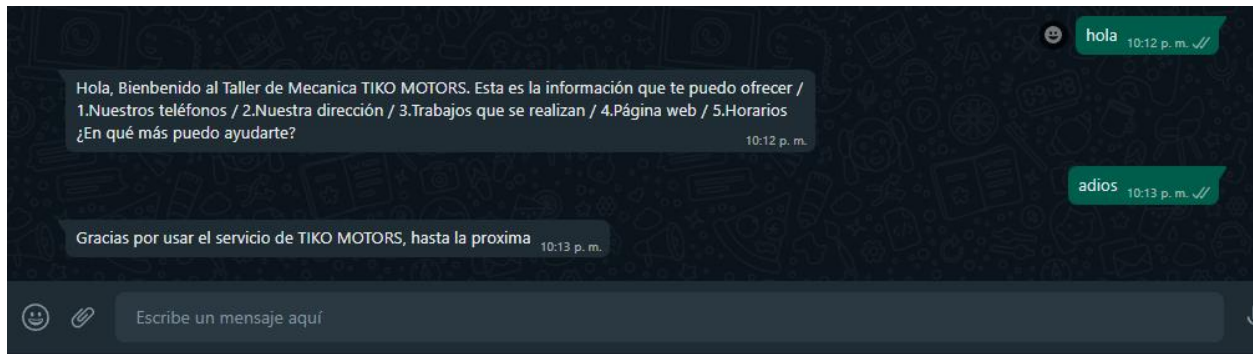
Resivir

GET /webhook

POST /webhook

WeatherForecast

GET /WeatherForecast



Comprobamos que el código está funcionando de manera correcta.

Este código muestra un namespace llamado "ChatBot_Taller_Mecanico.Models" que contiene varias clases relacionadas con un bot de chat para un taller mecánico. A continuación, se explica cada una de las clases y sus propiedades y métodos:

Clase "Dato":

```
using MySqlConnection,
2 referencias
public class Dato
{
    MySqlConnection connection = null;
    1 referencia
    public void InsertarRegistro(string mensajeR
```

- Contiene una instancia de MySqlConnection llamada "connection" para establecer la conexión con una base de datos MySQL.
- El método "InsertarRegistro" recibe varios parámetros (mensajeRecibido, mensajeEnviado, idWa, telefonoWa) y realiza una inserción de registros en una tabla llamada "registro" de la base de datos "chat". Utiliza parámetros para evitar posibles ataques de inyección SQL.
- El método "idWaExiste" recibe un parámetro "id_wa" y verifica si existe un registro en la tabla "registro" con ese valor en la columna "id_wa". Devuelve un valor booleano indicando si existe o no.

```

string connectionString = "Server=localhost;User ID=root;Password=;Database=chat";
 MySqlConnection connection = new MySqlConnection(connectionString);

try
{
    MySqlCommand command = connection.CreateCommand();
    command.CommandText = "INSERT INTO `registro` " +
        "(`mensaje_recibido`, `mensaje_enviado`, `id_wa`, `telefono_wa`) VALUES " +
        "(@mensajeRecibido, @mensajeEnviado, @idWa, @telefonoWa);";

    command.Parameters.AddWithValue("@mensajeRecibido", mensajeRecibido);
    command.Parameters.AddWithValue("@mensajeEnviado", mensajeEnviado);
    command.Parameters.AddWithValue("@idWa", idWa);
    command.Parameters.AddWithValue("@telefonoWa", telefonoWa);

    connection.Open();
}

```

Clase "WebHookResponseModel":

- Contiene una propiedad llamada "entry" que es un arreglo de objetos de la clase "Entry".

```

public class WebHookResponseModel
{
    3 referencias
    public Entry[] entry { get; set; }
}

```

Clase "Entry":

```

public class Entry
{
    3 referencias
    public Change[] changes { get; set; }
}
1 referencia

```

- Contiene una propiedad llamada "changes" que es un arreglo de objetos de la clase "Change".

Clase "Change":

- Contiene una propiedad llamada "value" que es un objeto de la clase "Value".

```

public class Change
{
    3 referencias
    public Value value { get; set; }
}

1 referencia

```

Clase "Value":

```

1 referencia
public class Value
{
    0 referencias
    public int ad_id { get; set; }
    0 referencias
    public long form_id { get; set; }
    0 referencias
    public long leadgen_id { get; set; }
    0 referencias
    public int created_time { get; set; }
    0 referencias
    public long page_id { get; set; }
    0 referencias
    public int adgroup_id { get; set; }
    3 referencias
    public Messages[] messages { get; set; }
}

```

- Contiene varias propiedades que representan diferentes atributos de un mensaje recibido en el webhook, como "ad_id", "form_id", "leadgen_id", "created_time", "page_id", "adgroup_id" y "messages".

- La propiedad "messages" es un arreglo de objetos de la clase "Messages".

Clase "Messages":

- Contiene propiedades como "id", "from" y "text", que representan diferentes atributos de un mensaje enviado o recibido en el chat.

```

1 referencia
public class Messages
{
    1 referencia
    public string id { get; set; }
    1 referencia
    public string from { get; set; }
    1 referencia
    public Text text { get; set; }
}
1 referencia

```

Clase "Text":

```

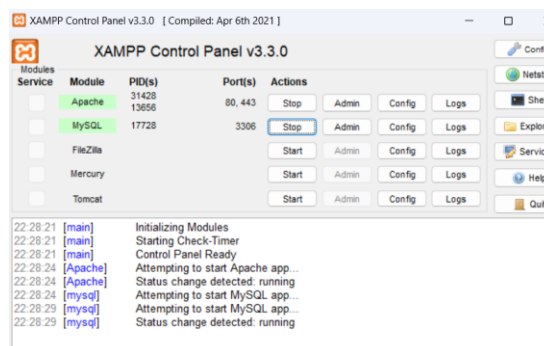
1 referencia
public class Text
{
    1 referencia
    public string body { get; set; }
}

```

- Contiene una propiedad llamada "body" que representa el contenido del texto en un mensaje.

Este código muestra una implementación básica para el manejo de registros en una base de datos y la estructura de los objetos utilizados para representar datos en el chat del bot. Es importante destacar que este código no incluye la implementación completa del bot de chat, sino solo las clases relacionadas con los datos y la estructura del mensaje.

El chatbot está conectado a una base de datos para poder controlar las conversaciones de este mismo o generar reportes más fácilmente.



En este caso estamos usando XAMPP Y PHP MYADMIN en donde está la base de Datos:

Tenemos creado una tabla dentro de la base de Datos:

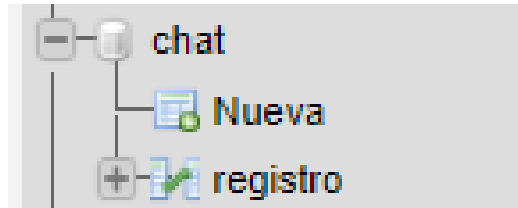


Tabla Registro:

				id	fecha_hora	mensaje_recibido	mensaje_enviado	id_wa	telefono_wa
<input type="checkbox"/>				1	2023-06-14 05:23:33	hola	Hola! Bienvenido al Taller de Mecánica Automotriz....	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAzNzICMzEyNEU0RUQ2...	59163147424
<input type="checkbox"/>				2	2023-06-14 06:23:26	hola	Hola! Bienvenido al Taller de Mecánica Automotriz....	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAwMkxQzhdM0UwMDQ5...	59163147424
<input type="checkbox"/>				3	2023-06-14 06:26:45	.	Intenta hacer tu pregunta de otra manera.	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCBOTBFNzJBREE0OEU4...	59163147424
<input type="checkbox"/>				4	2023-06-14 06:31:30	.	Recuerda que soy un robot, me podrías preguntar de...	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAxQzFFODJERjhBNUY5...	59163147424
<input type="checkbox"/>				5	2023-06-14 06:31:39	hola	Hola! Bienvenido al Taller de Mecánica Automotriz....	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAyRkM1MjkwMQ1NkFD...	59163147424
<input type="checkbox"/>				6	2023-06-14 06:32:10	Buenos días	Intenta hacer tu pregunta de otra manera.	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCA0MjkyOTdENzE5MjUy...	59163147424
<input type="checkbox"/>				7	2023-06-14 06:32:35	horario	Intenta hacer tu pregunta de otra manera.	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAyNzQNDJBN0M4QUZB...	59163147424
<input type="checkbox"/>				8	2023-06-14 06:33:06	telefono	Recuerda que soy un robot, me podrías preguntar de...	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCBGQkMzRDY1QzUzRjY1...	59163147424
<input type="checkbox"/>				9	2023-06-14 06:33:31	lugar	Intenta hacer tu pregunta de otra manera.	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCBFNEZDRjIDMUQ0RTE...	59163147424
<input type="checkbox"/>				10	2023-06-14 06:33:45	direccion	Recuerda que soy un robot, me podrías preguntar de...	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCA0OUJDRDM3NDJBNEM3...	59163147424
<input type="checkbox"/>				11	2023-06-14 06:34:06	ubicacion	Recuerda que soy un robot, me podrías preguntar de...	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCBBNjMyMDBDMyY1ODE0...	59163147424
<input type="checkbox"/>				12	2023-06-14 06:34:14	hola	Hola! Bienvenido al Taller de Mecánica Automotriz....	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCA3MDA4ODgyNDJFNDRy...	59163147424
<input type="checkbox"/>				13	2023-06-14 06:34:36	*	Intenta hacer tu pregunta de otra manera.	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAwNDc0N0U3MTAzMkU2...	59163147424
<input type="checkbox"/>				14	2023-06-14 06:37:46	hola	Hola! Bienvenido al Taller de Mecánica Automotriz....	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCBNCndBME2MjgwNDID...	59163147424
<input type="checkbox"/>				15	2023-06-14 06:38:04	.	Recuerda que soy un robot, me podrías preguntar de...	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCA1RTI4ODcwQTJGMjhc...	59163147424
<input type="checkbox"/>				16	2023-06-14 06:39:13	Hola	Hola! Bienvenido al Taller de Mecánica Automotriz....	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCBDM0M4NUNBRjRFRDVG...	59163147424
<input type="checkbox"/>				17	2023-06-14 06:39:39	Domingo	Intenta hacer tu pregunta de otra manera.	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAwRkQ1QkVGNEY0MjE...	59163147424
<input type="checkbox"/>				18	2023-06-14 06:39:46	domingo	Recuerda que soy un robot, me podrías preguntar de...	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCBBO0E0MjBDNDZCOEJF...	59163147424
<input type="checkbox"/>				19	2023-06-14 06:55:00	.	Intenta hacer tu pregunta de otra manera.	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAyQ0JBNUI1NzQzMTU4...	59163147424
<input type="checkbox"/>				20	2023-06-14 06:55:06	Hola	Hola! Bienvenido al Taller de Mecánica Automotriz....	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAxOTFCMTFERjQ4NQ3...	59163147424

Los campos de la tabla:

id	fecha_hora	mensaje_recibido	mensaje_enviado
id_wa	telefono_wa	wamid.HBgLNTkxNjMxNDc0MjQVAgASGCAzNzICMzEyNEU0RUQ2... 59163147424	

Prueba al chatBot Taller de Mecanica Automotriz:

