

UNIVERSIDAD PRIVADA

“FRANZ TAMAYO”



ESTUDIANTES:

- SAUL ESCOBAR SERRANO
- WILMER DAVID CORI APAZA
- RODRIGO CRISTHIAN TORREZ DE LA CRUZ
- HEBER MOLLERICONA MIRANDA
- HANS CRISTHIAN QUISBERT VARGAS
- JOSE LUIS YANAHUAYA MAMANI

Asignatura: BASE DE DATOS II

Docente: LIC. WILLIAM BARRA PAREDES

CARRERA: INGENIERIA DE SISTEMAS.

LA PAZ -EL ALTO, 27/06/2023

1. INTRODUCCION

El objetivo del proyecto es implementar un sistema de chatbot fitness para un gimnasio.

Los chatbots, o asistentes virtuales de chat, son programas informáticos que simulan conversaciones humanas. Utilizando inteligencia artificial y algoritmos de aprendizaje automático, los chatbots pueden interactuar con los usuarios de manera conversacional, proporcionando respuestas rápidas y precisas a través de mensajes de texto o voz. Su popularidad ha crecido rápidamente debido a su capacidad para automatizar tareas y mejorar la experiencia del cliente en diversos sectores. El objetivo principal de los chatbots es proporcionar asistencia y soporte eficiente, optimizando la comunicación y facilitando la interacción entre usuarios y empresas.

2. Análisis y diseño del sistema

- a. **Nombre del proyecto:** Chatbot_Fitness
- b. **Definición de las plataformas:** Bueno en este caso la clase se llama rutina y los métodos de la parte BIBLIOTECA tenemos los siguientes para la elaboración del CHATBOT:

- qrCode-terminal
- WhatsApp-Web.js
- MySQL
- Node.js.
- Javascript

Usuario		
Rutina		

qrCode-terminal	<pre>2 const qrcode = require('qrcode-terminal');</pre>
WhatsApp-Web.js	<pre>"keywords": [], "author": "", "license": "ISC", "dependencies": { "mysql": "^2.18.1", "mysql.js": "^1.0.2", "qrcode-terminal": "^0.12.0", "whatsapp-button": "^1.0.14", "whatsapp-web.js": "^1.21.0" }, "description": "" }</pre>
MySQL	<pre>const connection = mysql.createConnection({ host: 'localhost', user: 'Heber', password: '123456', database: 'chatbotwhatsapp' }); connection.connect((err) => { if (err) throw err console.log('La conexión funciona') });</pre>
Node.js.	Node.js es un entorno de tiempo de ejecución de JavaScript basado en el motor de JavaScript V8 de Google. A diferencia de JavaScript, que generalmente se ejecuta en el navegador web, Node.js permite ejecutar código JavaScript en el servidor. Proporciona una plataforma para construir aplicaciones de red escalables y de alta velocidad.
MySQL	Base de datos utilizado para la implementación del chatbot fitness.
Javascript	Lenguaje de programación utilizado para implementar el chatbot fitness.

c. Implementación:

La implementación fue en iniciar primeramente nuestro editor de JavaScript la cual fue realizado este proyecto

Ahora para la elaboración de este CHATBOT fue la siguiente

Primeramente para la realización quisimos que primero interactuara con nosotros a base de una conversación normal como si fuéramos personas pero en este case seria entre el humano y maquina y fue la siguiente referencia

```
index.js > client.on('message') callback
563
564
565 let iniciarSwitch = false;
566 client.on('message', message => {
567   if (procesoTerminado === true) {
568
569     const insertar2 = "INSERT INTO mensajeria (numero_celular_CLIENTE, mensaje_enviado, mensaje_recibido) values(?,?,?)";
570
571     if (iniciarSwitch === true) {
572       switch (message.body) {
573         case '1':
574           client.sendMessage(message.from, ruti.getPecho());
575           connection.query(insertar2, [message.from, ruti.getPecho(), message.body], (err) =>{
576             if(err) throw err
577             console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
578           })
579           break;
580         case '2':
581           client.sendMessage(message.from, ruti.getEspalda());
582           connection.query(insertar2, [message.from, ruti.getEspalda(), message.body], (err) =>{
583             if(err) throw err
584             console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
585           })
586           break;
587         case '3':
588           client.sendMessage(message.from, ruti.getHombros());
589           connection.query(insertar2, [message.from, ruti.getHombros(), message.body], (err) =>{
590             if(err) throw err
591             console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
592           })
593           break;
594         case '4':
595           client.sendMessage(message.from, ruti.getPiernas());
596           connection.query(insertar2, [message.from, ruti.getPiernas(), message.body], (err) =>{
597             if(err) throw err
598             console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
599           })
600           break;
601         case '5':
602           client.sendMessage(message.from, ruti.getBiceps());
603           connection.query(insertar2, [message.from, ruti.getBiceps(), message.body], (err) =>{
604             if(err) throw err
605             console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
606           })
607           break;
608         case '6':
609           client.sendMessage(message.from, ruti.getTriceps());
610           connection.query(insertar2, [message.from, ruti.getTriceps(), message.body], (err) =>{
611
```

Este fue el método pusimos para que se nos genere las preguntas que en este caso el mensaje que estamos lanzando será la cual escriba el usuario esto ayuda que todo el código que pusimos será la cual demos una pregunta o consulta y el programa responderá automáticamente lo que pusimos que responda el CHATBOT

```

else if (edadCl.includes(Number(message.body))) {
  anio = message.body.trim();
  client.sendMessage(message.from, 'Entendido. ¿Cual es tu CI? ');
  client.sendMessage(message.from, 'Envia el CI con la terminacion de tu Departamento, ejemplo: 123456789 LP');

  connection.query(insertar, [message.from, 'Entendido. ¿Cual es tu CI?', message.body], (err) =>{
    if(err) throw err
    console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
  })
}
}

```

Ahora que esta haciendo, como muestra el código estamos diciendo de que el CHATBOT le va a responder al usuario

```

client.sendMessage(message.from, 'Entendido. ¿Cual es tu CI? ');
client.sendMessage(message.from, 'Envia el CI con la terminacion de tu Departamento, ejemplo: 123456789 LP');

```

Será el mensaje que mandara y claro este no solo este sino empezara desde lo mas primordial en un registro empezando desde el nombre

```

let ci="";

client.on('message', message => {
  if (procesoTerminado === false) {

    const insertar = "INSERT INTO mensajería (numero_celular_CLIENTE, mensaje_enviado, mensaje_recibido) values(?,?,?)";

    if (ClienSaludo.includes(message.body)) {
      client.sendMessage(message.from, 'Hola, bienvenido/a a nuestro chatbot de fitness. ');
      client.sendMessage(message.from, '¿Cuál es tu nombre?');
      connection.query(insertar, [message.from, 'Hola, bienvenido/a a nuestro chatbot de fitness. ¿Cuál es tu nombre?', message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
      })
    }

    else if (nombres.includes(message.body)) {
      nombreCliente = message.body.trim();
      client.sendMessage(message.from, 'Hola, ${nombreCliente}!');
      client.sendMessage(message.from, '¿Cual es tu apellido?');
      connection.query(insertar, [message.from, '¿Cual es tu apellido?', message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
      })
    }

    else if (apellido.includes(message.body)) {
      ap11 = message.body.trim();
      client.sendMessage(message.from, '¿Cual es tu genero?');
      connection.query(insertar, [message.from, '¿Cual es tu genero?', message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
      })
    }

    else if (gen.includes(message.body)) {
      gener = message.body.trim();
      client.sendMessage(message.from, 'Perfecto');
      client.sendMessage(message.from, '¿Cuántos años tienes? *(solo numero)*');
      connection.query(insertar, [message.from, 'Perfecto ¿Cuántos años tienes? *(solo numero)*', message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
      })
    }
  }
})

```

Empezara a dar y obtener información del usuario la cual le pedirá nombre, apellido, entre otros



Por que aparece un QR

El QR aparece cuando usemos el método ya mostrado para que salga el QR, cuál es su función su función será que como ya hemos emparejado todo solo será de escanearlo el QR con el dispositivo la cual tendremos acceso al CHATBOT con todos los mensajes que podamos responder el CHATBOT responderá con los parámetros que se lo pusimos


```

423 client.sendMessage(message.from, "Envía el CI con la terminación de tu Departamento, ejemplo: 123456789 LP");
424
425 connection.query(insertar, [message.from, "Entendido, ¿Cual es tu CI?", message.body], (err) => {
426   if(err) throw err;
427   console.log("Mensaje insertado correctamente en la tabla 'mensajería'");
428 })
429

```


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\VASUS\Desktop\CHATBOTWATSAPP> npm start

> chatbotwhatsapp@1.0.0 start

> node index.js

La conexión funciona



Client is ready!

Mensaje insertado correctamente en la tabla 'mensajería'

Mensaje insertado correctamente en la tabla 'mensajería'

Mensaje insertado correctamente en la tabla 'mensajería'

Mensaje insertado correctamente en la tabla 'mensajería'

Mensaje insertado correctamente en la tabla 'mensajería'

Mensaje insertado correctamente en la tabla 'mensajería'

Mensaje insertado correctamente en la tabla 'mensajería'

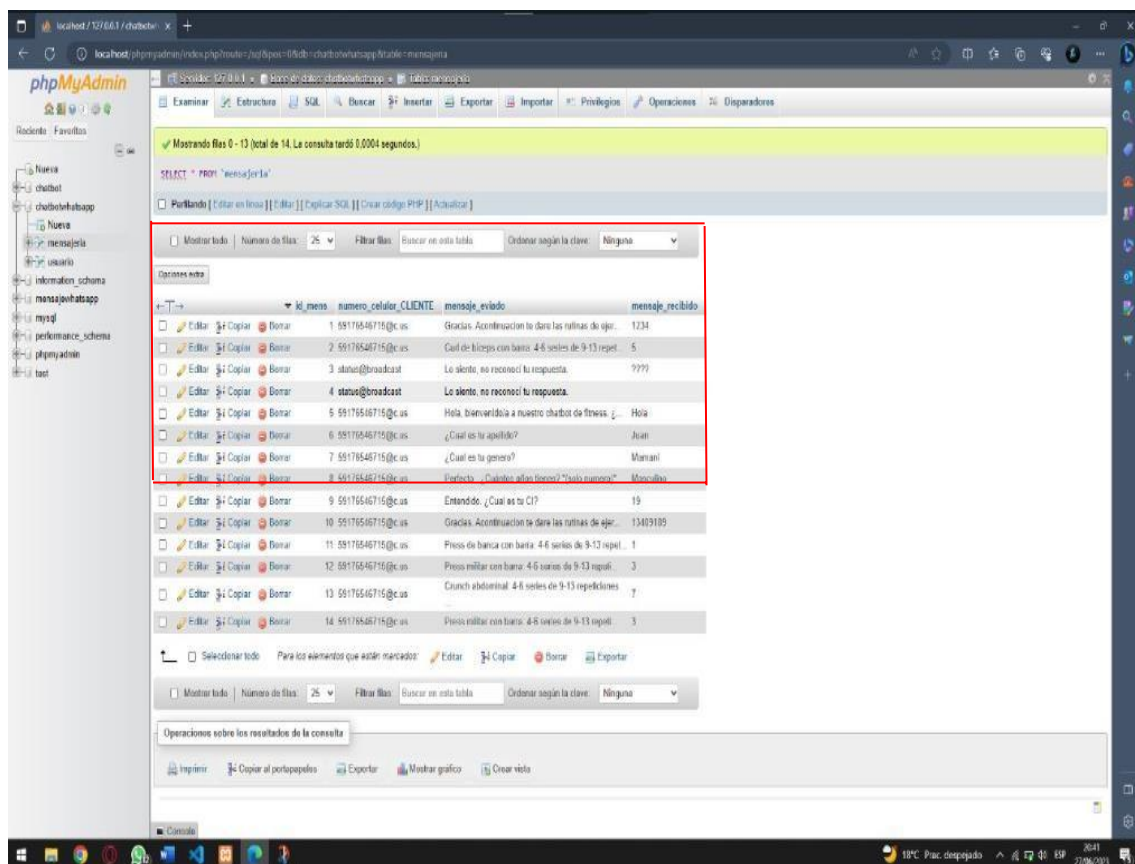
Mensaje insertado correctamente en la tabla 'mensajería'

La entrada

La salida



Como podemos observar nuestro CHATBOT nosotros primeramente iniciaremos una conversación normal nos empezará a preguntar algunos de nuestros datos, sabemos que cada registros debemos dar información personal la cual se automatiza y se guarda en una base de datos la cual será la siguiente



Como podemos observar en la siguiente imagen estaríamos hablando de la plataforma de MySQL la cual esta plataforma que tiene que ver con el CHATBOT.

Nos sirve demasiado esta plataforma por que de esta manera podemos ver que persona se registraron a si como los mensajes realizados en las últimas semanas mediante la inteligencia artificial.

3. Usabilidad

Las pruebas para este proyectos fueron muy concretas porque teníamos que controlar muchas cosas en una de ellas aparte del código de JavaScript teníamos que revisar sobre los mensajes que nuestro CHATBOT diría y que nosotros respondamos la cual eso era un problema más, ahora otra cosa más que teníamos que revisar era la base de datos donde las personas se registraban o enviaban mensajes con toda la información

VIDEO DE PRUEBA

<https://drive.google.com/file/d/17HFmHlm7AoOM4pdSyLyy0ls99-ItMOgh/view?usp=drivesdk>

IMÁGENES DE PROYECTO

```
let iniciarSwitch = false;
client.on('message', message => {
  if (procesoTerminado === true) {

    const insertar2 = "INSERT INTO mensajeria (numero_celular_CLIENTE, mensaje_enviado, mensaje_recibido) values(?,?,?)";

    if (iniciarSwitch === true) {
      switch (message.body) {
        case '1':
          client.sendMessage(message.from, ruti.getPecho());
          connection.query(insertar2, [message.from, ruti.getPecho(), message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
          })

          break;
        case '2':
          client.sendMessage(message.from, ruti.getEspalda());
          connection.query(insertar2, [message.from, ruti.getEspalda(), message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
          })

          break;
        case '3':
          client.sendMessage(message.from, ruti.getHombros());
          connection.query(insertar2, [message.from, ruti.getHombros(), message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
          })

          break;
        case '4':
          client.sendMessage(message.from, ruti.getPiernas());
          connection.query(insertar2, [message.from, ruti.getPiernas(), message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
          })

          break;
        case '5':
          client.sendMessage(message.from, ruti.getBiceps());
          connection.query(insertar2, [message.from, ruti.getBiceps(), message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajeria'")
          })

          break;
        case '6':
          client.sendMessage(message.from, ruti.getTriceps());
          connection.query(insertar2, [message.from, ruti.getTriceps(), message.body], (err) =>{
```



```

        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
    })
    break;
case '4':
    client.sendMessage(message.from, ruti.getPiernas());
    connection.query(insertar2, [message.from, ruti.getPiernas(), message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
    })
    break;
case '5':
    client.sendMessage(message.from, ruti.getBiceps());
    connection.query(insertar2, [message.from, ruti.getBiceps(), message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
    })
    break;
case '6':
    client.sendMessage(message.from, ruti.getTriceps());
    connection.query(insertar2, [message.from, ruti.getTriceps(), message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
    })
    break;
case '7':
    client.sendMessage(message.from, ruti.getAbdominales());
    connection.query(insertar2, [message.from, ruti.getAbdominales(), message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
    })
    break;
default:
    client.sendMessage(message.from, 'Por favor, elige del *1 - 7* ');
    connection.query(insertar2, [message.from, 'Por favor, elige del *1 - 7* ', message.body], (err) =>{
        if(err) throw err
        console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
    })
    break;
    })
}
client.sendMessage(message.from, '(Qué grupo muscular te gustaría trabajar?');
client.sendMessage(message.from, '1. Pecho \n2. Espalda \n3. Hombros\n 4. Piernas\n 5. Biceps\n 6. Triceps\n 7. Abdominales');
iniciarSwitch = true;
}
));

```

```

        connection.query(insertar, [message.from, 'Perfecto ¿Cuántos años tienes? *(solo numero)*', message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
        })
    }
    else if (edadCl.includes(Number(message.body))) {
        anio = message.body.trim();
        client.sendMessage(message.from, 'Entendido. ¿Cual es tu CI? ');
        client.sendMessage(message.from, 'Envía el CI con la terminación de tu Departamento, ejemplo: 123456789 LP');

        connection.query(insertar, [message.from, 'Entendido. ¿Cual es tu CI?', message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
        })
    }

    else if (Number(message.body) > 100) {
        ci = message.body.trim();
        const usua = new Usuario(nombreCliente, ap11, anio, gener, ci);/*
        registrarUsuario(ci, nombreCliente, ap11, anio, gener);*/

        client.sendMessage(message.from, 'Gracias. Acontinuacion te dare las rutinas de ejercicios');

        connection.query(insertar, [message.from, 'Gracias. Acontinuacion te dare las rutinas de ejercicios', message.body], (err) =>{
            if(err) throw err
            console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
        })
    }

    else {
        client.sendMessage(message.from, 'Lo siento, no reconocí tu respuesta.');
```

```

client.on('message', message => {
    if (procesoTerminado === false) {

        const insertar = "INSERT INTO mensajería (numero_celular_CLIENTE, mensaje_enviado, mensaje_recibido) values(?,?,?)";

        if (ClienSaludo.includes(message.body)) {
            client.sendMessage(message.from, 'Hola, bienvenido/a a nuestro chatbot de fitness. ');
            client.sendMessage(message.from, '¿Cuál es tu nombre?');
            connection.query(insertar, [message.from, 'Hola, bienvenido/a a nuestro chatbot de fitness. ¿Cuál es tu nombre? ', message.body], (err) =>{
                if(err) throw err
                console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
            })
        }

        else if (nombres.includes(message.body)) {
            nombreCliente = message.body.trim();
            client.sendMessage(message.from, 'Hola, ${nombreCliente}!');
            client.sendMessage(message.from, '¿Cual es tu apellido?');
            connection.query(insertar, [message.from, '¿Cual es tu apellido?', message.body], (err) =>{
                if(err) throw err
                console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
            })
        }

        else if (apellido.includes(message.body)) {
            ap11 = message.body.trim();
            client.sendMessage(message.from, '¿Cual es tu genero?');
            connection.query(insertar, [message.from, '¿Cual es tu genero?', message.body], (err) =>{
                if(err) throw err
                console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
            })
        }

        else if (gen.includes(message.body)) {
            gener = message.body.trim();
            client.sendMessage(message.from, 'Perfecto');
            client.sendMessage(message.from, '¿Cuántos años tienes? *(solo numero)*');

            connection.query(insertar, [message.from, 'Perfecto ¿Cuántos años tienes? *(solo numero)*', message.body], (err) =>{
                if(err) throw err
                console.log("Mensaje insertado correctamente en la tabla 'mensajería'")
            })
        }

        else if (edadCl.includes(Number(message.body))) {
            anio = message.body.trim();
            client.sendMessage(message.from, 'Entendido. ¿Cual es tu CI? ');
```

4. Conclusiones

Como primer punto podemos decir que esto de la inteligencia artificial es muy compleja mas que todo en la creación de CHATBOT pero también podemos dar algunas de las ventajas:

1. Ayudaría mucho en cualquier tipo de empresa así facilitaría mas a los clientes y al personal de la empresa
2. Mejora de la experiencia del cliente: Un chatbot bien diseñado y correctamente implementado puede mejorar significativamente la experiencia del cliente. Al proporcionar respuestas rápidas y precisas a las consultas de los usuarios, los chatbots pueden ofrecer un servicio al cliente más eficiente y disponible las 24 horas del día. Esto resulta en una mayor satisfacción del cliente y en la mejora de la imagen de la empresa.
3. Automatización de tareas rutinarias: Los chatbots son capaces de realizar tareas rutinarias y repetitivas de manera automatizada. Esto libera tiempo y recursos para el personal humano, que puede enfocarse en actividades más complejas y de alto valor. La automatización a través de un chatbot puede agilizar procesos como el registro de información, la programación de citas y la entrega de respuestas a preguntas frecuentes, lo que ahorra tiempo tanto a los usuarios como a la empresa.
4. Escalabilidad y disponibilidad constante: Los chatbots pueden manejar múltiples conversaciones simultáneamente sin comprometer la calidad de las respuestas. Esto los convierte en herramientas altamente escalables y capaces de brindar asistencia a un gran número de usuarios al mismo tiempo. Además, a diferencia de los agentes humanos, los chatbots están disponibles las 24 horas del día, los 7 días de la semana, lo que permite una atención continua y sin interrupciones.