diseño de la base de datos:

1. **tabla usuarios**
   * **usuario\_id** (pk): un identificador único para cada usuario.
   * **nombre**: el nombre del usuario.
   * **correo**: el correo electrónico del usuario.
   * **contrasena**: la contraseña del usuario.
   * **fecha\_creacion**: fecha y hora de cuando se creó el usuario.
2. **tabla bots**
   * **bot\_id** (pk): un identificador único para cada bot.
   * **usuario\_id** (fk): el identificador del usuario que creó el bot. esto conecta la tabla **bots** con la tabla **usuarios**.
   * **nombre**: el nombre del bot.
   * **activo**: un booleano que indica si el bot está activo o no.
   * **fecha\_creacion**: fecha y hora de cuando se creó el bot.
   * **horario\_activo:** columna donde se almacena el horario durante el cual el bot está activo.
3. **tabla mensajes**
   * **mensaje\_id** (pk): un identificador único para cada mensaje.
   * **bot\_id** (fk): el identificador del bot que envió o recibió el mensaje. esto conecta la tabla **mensajes** con la tabla **bots**.
   * **mensaje**: el contenido del mensaje.
   * **enviado\_por\_bot**: un booleano que indica si el mensaje fue enviado por el bot o recibido por el bot.
   * **fecha\_creacion**: fecha y hora de cuando se envió o recibió el mensaje.
4. **tabla plantillas**
   * **plantilla\_id** (pk): un identificador único para cada plantilla de mensaje.
   * **contenido**: el contenido de la plantilla del mensaje.
   * **nombre**: el nombre de la plantilla.
5. **tabla programacion**: para la programación de mensajes automatizados.
   * **programacion\_id** (pk): un identificador único para cada programación de mensajes.
   * **bot\_id** (fk): el identificador del bot que programó el mensaje. esto conecta la tabla **programacion** con la tabla **bots**.
   * **hora\_envio**: la hora programada para enviar el mensaje.
   * **mensaje\_id** (fk): el mensaje programado para ser enviado. esto conecta la tabla **programacion** con la tabla **mensajes**.
   * **destinatario\_id** (fk): el destinatario del mensaje programado. esto conecta la tabla **programacion** con la tabla **clientes**.
6. **tabla clientes**: para almacenar información sobre los clientes que interactúan con los bots.
   * **cliente\_id** (pk): un identificador único para cada cliente.
   * **nombre**: el nombre del cliente.
   * **contacto**: información de contacto del cliente (p.ej. número de whatsapp).
   * **fecha\_creacion**: fecha y hora de cuando se añadió el cliente.
7. **tabla palabras\_clave**: para almacenar las palabras clave que deben activar el bot.
   * **palabra\_clave\_id** (pk): un identificador único para cada palabra clave.
   * **bot\_id** (fk): el identificador del bot que responde a la palabra clave. esto conecta la tabla **palabras\_clave** con la tabla **bots**.
   * **palabra\_clave**: la palabra o frase clave.
8. **tabla respuestas\_automaticas**: para almacenar las respuestas automáticas que el bot puede enviar.
   * **respuesta\_id** (pk): un identificador único para cada respuesta automática.
   * **bot\_id** (fk): el identificador del bot que usa la respuesta automática. esto conecta la tabla **respuestas\_automaticas** con la tabla **bots**.
   * **plantilla\_id** (fk): la plantilla de mensaje utilizada para la respuesta automática. esto conecta la tabla **respuestas\_automaticas** con la tabla **plantillas**.
   * **palabra\_clave\_id** (fk): la palabra clave que desencadena la respuesta automática. esto conecta la tabla **respuestas\_automaticas** con la tabla **palabras\_clave**.

sentencias 1 mysql:

use tfg\_dam\_app6;

-- tabla usuarios

create table usuarios (

usuario\_id int primary key,

nombre varchar(255),

correo varchar(255),

contrasena varchar(255),

fecha\_creacion datetime

);

-- tabla bots

create table bots (

bot\_id int primary key,

usuario\_id int,

nombre varchar(255),

activo boolean,

fecha\_creacion datetime,

horario\_activo varchar(255),

foreign key (usuario\_id) references usuarios(usuario\_id)

);

-- tabla mensajes

create table mensajes (

mensaje\_id int primary key,

bot\_id int,

mensaje text,

enviado\_por\_bot boolean,

fecha\_creacion datetime,

foreign key (bot\_id) references bots(bot\_id)

);

-- tabla plantillas

create table plantillas (

plantilla\_id int primary key,

bot\_id int,

contenido text,

nombre varchar(255),

foreign key (bot\_id) references bots(bot\_id)

);

-- tabla programacion

create table programacion (

programacion\_id int primary key,

bot\_id int,

hora\_envio datetime,

mensaje\_id int,

destinatario\_id int,

foreign key (bot\_id) references bots(bot\_id),

foreign key (mensaje\_id) references mensajes(mensaje\_id),

foreign key (destinatario\_id) references clientes(cliente\_id)

);

-- tabla clientes

create table clientes (

cliente\_id int primary key,

nombre varchar(255),

contacto varchar(255),

fecha\_creacion datetime

);

-- tabla palabras\_clave

create table palabras\_clave (

palabra\_clave\_id int primary key,

bot\_id int,

palabra\_clave varchar(255),

foreign key (bot\_id) references bots(bot\_id)

);

-- tabla respuestas\_automaticas

create table respuestas\_automaticas (

respuesta\_id int primary key,

bot\_id int,

plantilla\_id int,

palabra\_clave\_id int,

foreign key (bot\_id) references bots(bot\_id),

foreign key (plantilla\_id) references plantillas(plantilla\_id),

foreign key (palabra\_clave\_id) references palabras\_clave(palabra\_clave\_id)

);

sentencias 2 mysql:

use tfg\_dam\_app6;

-- tabla usuarios

create table usuarios (

usuario\_id int primary key,

nombre varchar(255),

correo varchar(255) unique,

contrasena varchar(255),

fecha\_creacion datetime

);

-- tabla bots

create table bots (

bot\_id int primary key,

usuario\_id int,

nombre varchar(255),

activo boolean,

fecha\_creacion datetime,

horario\_activo varchar(255),

foreign key (usuario\_id) references usuarios(usuario\_id)

on delete cascade

on update cascade

);

-- tabla mensajes

create table mensajes (

mensaje\_id int primary key,

bot\_id int,

mensaje text,

enviado\_por\_bot boolean,

fecha\_creacion datetime,

foreign key (bot\_id) references bots(bot\_id)

on delete cascade

on update cascade

);

-- tabla plantillas

create table plantillas (

plantilla\_id int primary key,

bot\_id int,

contenido text,

nombre varchar(255),

foreign key (bot\_id) references bots(bot\_id)

on delete cascade

on update cascade

);

-- tabla programacion

create table programacion (

programacion\_id int primary key,

bot\_id int,

hora\_envio datetime,

mensaje\_id int,

destinatario\_id int,

foreign key (bot\_id) references bots(bot\_id)

on delete cascade

on update cascade,

foreign key (mensaje\_id) references mensajes(mensaje\_id)

on delete cascade

on update cascade,

foreign key (destinatario\_id) references clientes(cliente\_id)

on delete cascade

on update cascade

);

-- tabla clientes

create table clientes (

cliente\_id int primary key,

nombre varchar(255),

contacto varchar(255),

fecha\_creacion datetime

);

-- tabla palabras\_clave

create table palabras\_clave (

palabra\_clave\_id int primary key,

bot\_id int,

palabra\_clave varchar(255),

foreign key (bot\_id) references bots(bot\_id)

on delete cascade

on update cascade

);

-- tabla respuestas\_automaticas

create table respuestas\_automaticas (

respuesta\_id int primary key,

bot\_id int,

plantilla\_id int,

palabra\_clave\_id int,

foreign key (bot\_id) references bots(bot\_id)

on delete cascade

on update cascade,

foreign key (plantilla\_id) references plantillas(plantilla\_id)

on delete cascade

on update cascade,

foreign key (palabra\_clave\_id) references palabras\_clave(palabra\_clave\_id)

on delete cascade

on update cascade

);

Mensaje recibido desde twingo:

CombinedMultiDict([ImmutableMultiDict([]), ImmutableMultiDict([('SmsMessageSid', 'SM526c1b26af62e848e4411ea4f2e7be34'), ('NumMedia', '0'), ('ProfileName', 'Fernando Moreno'), ('SmsSid', 'SM526c1b26af62e848e4411ea4f2e7be34'), ('WaId', '34680683625'), ('SmsStatus', 'receirralNumMedia', '0'), ('MessageSid', 'SM526c1b26af62e848e4411ea4f2e7be34'), ('AccountSid', 'AC88003bb9a878075d4f0ee2cc1777bf69'), ('From', 'whatsapp:+34680683625'), ('ApiVersion', '2010-04-01')])])

Información del usuario obtenida de auth google

{"email": "luisfergoza@gmail.com", "family\_name": "Moreno", "given\_name": "Luis Fernando", "id": "10193355041924035", "locale": "es", "name": "Luis Fernando Moreno", "picture": "https://lh3.googleusercontent.com/a/AAcHTtc4dsCshAhX7lD0ibxa-Wwl9aBBl47je0Yh-x3KSSs=s96-c", "verified\_email": true}

            # Verificar el formato de las horas (HH:MM:SS)

            # hora\_inicio\_valida = re.match(r'^\d{2}:\d{2}:\d{2}$', hora\_inicio\_actividad)

            # hora\_fin\_valida = re.match(r'^\d{2}:\d{2}:\d{2}$', hora\_fin\_actividad)

            # patrón para comprobar solo horas y minutos:

            # r'^([01]\d|2[0-3]):([0-5]\d)$'

            # tendremos que asegurarnos de en qué formato nos llegan las horas para poder

            # comprobar de una manera u otra