# Construcción de un webhook en Messenger Facebook para la recolección de datos

La construcción y puesta en funcionamiento del bot es relativamente sencilla. Bastaría con seguir los pasos de la guía que ofrece Facebook para construir un webhook.

<https://developers.facebook.com/docs/messenger-platform/getting-started/webhook-setup>

En esta muestra el camino para construir el bot y para sincronizar el webhook con la página de Facebook, de manera que los mensajes recibidos por esta sean redirigidos al software. Para ello será necesario construir un proyecto npm e instalar las librerías de express, copiar el código pertinente y posteriormente desplegar el proyecto en un servidor. El código inicial señalado por Facebook constara de dos partes. La primera será la encargada de importar las dependencias necesarias y abrir un puerto de escucha para recibir las llamadas a nuestro servicio. La segunda constará de dos funciones básicas para el funcionamiento de nuestro bot: una get, y otro post. La primera será la encargada de la verificación del servicio, de manera que Facebook pueda llamarla cuando le digamos el emplazamiento del webhook y verificar que el servicio llamado e el correcto, a través de un código de verificación del que constarán las dos partes. La segunda, la post, será el servicio a través del cual Facebook nos enviará los mensajes recibidos en la página para que sean tratados por nuestro servicio.

Para comprobar el funcionamiento de estas dos será necesario desplegar el servicio en un servidor con un certificado ssl y que soporte https, para lo que elegimos heroku. Para desplegar el servicio en heroku será necesario crearse una cuenta de heroku, crear una aplicación y seguir los pasos indicados en la pestaña deploy del menú de nuestra aplicación.

Una vez tenemos el código desplegado será necesario crear una aplicación en Facebook for developers y añadirle a esta los servicios de messenger. En la configuración de estos existe una opción de configurar tu webhook en que podremos indicar el emplazamiento de nuestro webhook, el código de verificación (también indicado en nuestro servidor) y la pagina por la que queremos que el bot interceda, de manera que los paquetes de datos enviados por Messenger hacia eta página serán también enviados a nuestro webhook que se encargará de interpretarlos.

Posteriormente en la guía muestra como tratar los mensajes de texto y aquellos con archivos adjuntos. Facebook nos enviará los mensajes en forma de paquetes de datos en el body de la petición, body que será interpretado por el método post, el cual se encargará de quedarse con el paquete que nos interesa de este body, y de pasárselo a la función handleMessage o handlePostback según se trate de un paquete de tipo postback o de tipo mensaje.

Estos dos métodos se encargarán de procesar el contenido de los paquetes y responder según este. El primero, hanldeMessage, tratará los paquetes de tipo mensaje y se encargará de chequear si el mensaje se trata de un texto o de un archivo adjunto (imagen, ubicación…) y responder según esto. Para responder llamará a la otra función generada, llamada callSendApi, a la que deberemos pasarle el id del destinatario, que será el mismo que el del sender del paquete de datos recibido; y el mensaje de respuesta, con el formato indicadoporfacebook para cada caso (quick replys, postback, text message…).

El segundo método, handlePostback servirá para procesar los mensajes de este tipo, que serán los diferentes botones (como el de comenzar, los del menú o los de una respuesta de tipo postback con botones adjuntos). De esta manera según el valor del payload del botón se generarña una respuesta y se le pasará a la función callSendApi.

Toda esta información que se vaya recibiendo es la que deberemos enviar a la base de datos de arcgis. Para esto y para el control del dialogo es además posible que sea necesario la implementación con una base de datos.

## Integración con mongoDb

Para esta labor el mayor inconveniente con que nos encontramos es que el bot no tiene ningún tipo de memoria y trata cada mensaje aisladamente. Es por esto que a la hora de tener una conversación será necesario ir guardando esta de alguna manera para saber como contestar en cada caso. Así será necesario conectar al bot con una mongo database en que se irá guardando la información recolectada en objetos “report”, a los que también se les dotara de un campo “step” que nos permitirá conocer por que paso de la conversación estamos en cada momento, de manera que podamos saber a que pregunta de las que le hacemos a los usuarios se corresponde cada una de sus respuestas.

Para el uso de la mongo database primero de todo generaremos una base de datos en mongodb atlas, que nos permitirá levantar un servidor mongo y crear en este colleciones de prueba. Estas tendrán una uri a través de la cual podremos hacer las llamadas desde nuestro código. De hecho lo primero que haremos en este ser abrir una conexión con esta base de datos y generar una colección y un esquema para los objetos que crearemos en esta.

var ReportSchema = {

  sender\_id: { type: Number },

  step: { type: Number },

  response: { type: {} },

  responseAux: { type: {} },

  responseAuxIndicator: { type: Number },

  fromApp: { type: Boolean },

  homeOrComunitty: { type: String },

  cause: { type: String },

  homeDamages: { type: String },

  humansHarmed: { type: Number },

  humansDeath: { type: Number },

  date: { type: Number },

  X: { type: Number },

  Y: { type: Number },

  address: { type: String },

  img: { data: Buffer, contentType: String },

  observation: { type: String },

  imgUrl: { type: String },

  tomarControl: { type: Boolean },

  formatedDate: { type: String }

};

Para la interacción de nuestro código con mongo utilizaremos la libreria de javascript mongoose, cuyas dependencias deberemos instalar en el paquete y en el servidor de heroku. Cada vez que importemos unas nuevas dependencias esto será necesario.

El siguiente paso será crear las funciones:

* getReport(sender\_Id: nos permite obtener el último objeto generado con sender\_id el indicado
* create(sender\_psid, stepNew): crea un objeto report en la base de datos con el sender\_psid referido y el paso indicado
* getStep(sender\_psid): llama a getReport y devuelve el paso del reporte devuelto y el objeto reporte, en caso de que lo encuentre. Si no existe ningún reporte asociado a esesender\_psid, o este ya ha sido terminado o ha expirado, devuelve un -1, que indicaráa handleMessage que será necesario crear un objeto reporte nuevo en mongo.
* nextStep(report): aumenta en uno el campo step
* reset(): borra todas las entradas de la colección
* fillReport(field, value, report): rellena el campo indicado, con el valor indicado del objeto report indicado en mongo

Con esto ya podremos tener un inventario de las diferentes conversaciones y de la etapa de estas, para poder responder adecuadamente y guardar las respuestas recibidas por los usuarios.

En una sección de este documento se dedica a configurar la base de datos. <https://www.sitepoint.com/building-facebook-chat-bot-node-heroku/>

## Elaboración de la lógica conversacional

Esta será llevada principalmente por handleMessage, que se encargará de llamar a getStep cada vez que sea activado y disparar una de las funciones que generaremos para cada uno de los pasos.

En el caso de que el paso recibido por getStep sea -1 será necesario crear una nueva entrada en mongo. En el resto de los casos se llamará a una función según el tipo de objeto recibido y el paso, pasándoles las variables msgText, con el contenido del mensaje; y el objeto report recibido por getStep. Estas funciones se encargarán de generar una acción para los distintos tipos de respuestas esperadas en cada paso y de actualizar el report según sea adecuado.

Para el caso en que el tipo de mensaje no se corresponda con el esperado según el paso se generará el método correctDemand(sender\_psid, step, report), que generará una respuesta acertada para cada caso.

Para el caso en que el paquete recibido sea una imagen se guardara la url de f benque se emplaza y además se convertirá a bloob para poder guardarla en mongo, usando la función getImage(url,callback).

## Botón comenzar

Cuando un usuario pretende escribir a nuestro bot y no ha tenido ninguna conversación previa con él. Se encontrará con un mensaje de saludo y un botón comenzar, que será interpretado como un callback. La creación de este la encontramos en el siguientelink y es a través del comando escrito posteriormente.

<https://www.sitepoint.com/building-facebook-chat-bot-node-heroku/>

curl -X POST -H "Content-Type : application/json" -d '{ "setting\_type" : "call\_to\_actions", "thread\_state" : "new\_thread", "call\_to\_actions" : [{"payload":"Greeting"}]}' [https://graph.facebook.com/v2.6/me/thread\_settings?access\_token= EAAHLYiGPUaMBAP9GLbFuUi4ZCpGCEKoT5tG1hsdOfzZBjnq8hcWQKbErPyO5lxZAa13ItzyRr6TAmpE3w1mYzjnMZAK2DBQaHZBSiuMoIAbqdhBEEK8fHNQLyGQbY5jjgW3cZAaWpZBwY61rO9kGrMw1fxafEiCO9dqPAIYKLTNNwZDZD](https://graph.facebook.com/v2.6/me/thread_settings?access_token=%20EAAHLYiGPUaMBAP9GLbFuUi4ZCpGCEKoT5tG1hsdOfzZBjnq8hcWQKbErPyO5lxZAa13ItzyRr6TAmpE3w1mYzjnMZAK2DBQaHZBSiuMoIAbqdhBEEK8fHNQLyGQbY5jjgW3cZAaWpZBwY61rO9kGrMw1fxafEiCO9dqPAIYKLTNNwZDZD)

## Botones del menú

Para esto utilizaremos <https://developers.facebook.com/docs/messenger-platform/send-messages/persistent-menu/?locale=es_ES> de facebook, cuyo funconamiento se nos indica en elsiguiente link. La llamada necesaria al bot para activar el menú se encuentra al final del script prueballamada.js

## Usar terminal de servidor de heroku

Para instalar los modulos a usar en el paquete se usa npm install xxx. Pero además será necesario instalarlos en la maquin virtual y para ello se correra dentro de la maquina de comandos del servidor de heroku, la cual se lanza con el comando: Heroku run bash.

## Can not find module xxxx error

Cuando ocurra esto, será porque en el proceso de construcción del bot es necesario añadir al proyecto las dependencias de las que se quiere hacer require en el código. Para eso usaremos el comando npm instal xxxx.

Una vez subido a heroku habrá que ejecutar npm install desde la maquina virtual(es decir, tras correr heroku run bash en el cmd para acceder al servidor remoto)

## Tratamiento de las imagenes

En el caso de recibir una imagen esta es contenida en una url, a la que se puede acceder desde el navegador. Esta se encuentra en el objeto recibido. Received\_mesage.attachments[0] y entre sus propiedades tata: Received\_mesage.attachments[0].type = “image”, así como Received\_mesage.attachments[0].payload.url.

Este procedimiento de comprobación aparece en el código inicial sugerido por Facebook, dentro de la finción handle message.

## Tratamiento de locations

Muy parecido a las imágenes. Será un objeto de tipo webhook\_event.message, que tendrá attachments.

Received\_message.attachments[0].type = “location”, Received\_message.attachments[0].payload.coordinates.lat o . long

## Facebook ofrece quik replys

<https://developers.facebook.com/docs/messenger-platform/send-messages/quick-replies/>

Estas permiten crear botones con opciones de respuesta rápida, que será procesados igual que mensajes. Estas respuestas serán declaradas al principio de un script. Cuando un usuario responde utilizando una de las quick replys es interpretado como un mensaje de texto, cuyo texto es el que aparece en el botón de quick replys. En el caso de que el texto no quepa, este aparecerá con puntos suspensivos, y lo recibido en caso de usarse también serña con puntossuspensvis.

## Otras posibilidades de interacción

<https://developers.facebook.com/docs/messenger-platform/introduction/conversation-components/#sender_actions>

## Guardar imagen en base de datos

Para guardar la imagen enbase de datos, se guardara la url de Facebook y también la imagen transformada a blob,en un field de este tipo en mongo.

Para esto utiliza la función.

function getImage(url, callback) {

  https.get(url, res => {

    // Initialise an array

    const bufs = [];

    // Add the data to the buffer collection

    res.on('data', function (chunk) {

      bufs.push(chunk)

    });

    // This signifies the end of a request

    res.on('end', function () {

      // We can join all of the 'chunks' of the image together

      const data = Buffer.concat(bufs);

      // Then we can call our callback.

      callback(null, data);

    });

  })

    // Inform the callback of the error.

    .on('error', callback);

}

El tipo decampo:

var schema = new Schema({

img: { data: Buffer, contentType: String }

});

https://st ackoverflow.com/questions/29780733/store-an-image-in-mongodb-using-node-js-express-and-mongoose

## Ordenar respuestas

No existe manera de asegurarse de que son enviadas por orden. En algunos momentosnuestro bot requiere responder con dos mensajes y para esto anidamos dos llamadas a callSendApi, esperando a recibir una confirmacióndel envio de la primera para llamar de nuevo.

Esto se realizará al final de la función handleMessage, y sabra cuando enviardos respuestas a apartir del valor del campo respAuxIndicator, que en caso de ser 1 enviará primero un mensaje con el valorde respAux

<https://stackoverflow.com/questions/47483190/sending-multiple-reply-messages-on-single-postback-in-facebook-messenger-bots>

## Post automático a partir de link

En la siguiente web se pueden generar enlaces para postear automáticamente <http://www.sharelinkgenerator.com/>. Así queremos que nos genere un link para compartir nuestrapágina defacebook.

## Terminos de uso

Los alojamos usando el siguiente servidor <https://app.netlify.com/sites/laughing-leakey-d5dbf0/overview>

## Conseguir permisos para hacer la app pública

Será necesario seguir los pasos de verificación que Facebook nos comenta

## Coordinar datos con ArcGis

Para esto utilizaremos los servicios rest de ArcGis <https://developers.arcgis.com/documentation/core-concepts/rest-api/> De entre estos usaremos addFeature. Será muy importante que en el objeto que pasmos coom parámetro paraañadir, dentro delcampo de geometría especifiquemos la spatialreference

    "geometry": { "x": report.X, "y": report.Y, "spatialReference": { "wkid": 4326 }

## Conseguir location desde el address suministrado porlos usuarios

Usamos la api de Google geocoding. Esta deberá ser añadida a un proyecto y conseguir unas credenciales como se explicaen un apartado del link.

<https://developers.google.com/maps/documentation/geocoding/intro>

Da algún problema cuando tiene ildes o ñs asi que son eliminadas con el método que aquí aparece <https://es.stackoverflow.com/questions/62031/eliminar-signos-diacr%C3%ADticos-en-javascript-eliminar-tildes-acentos-ortogr%C3%A1ficos>

Tran pedir asistencia Google cloud nos confirman que los servicios de localización tienen un crédito gratuito mensual de 200$

La llamada se hace desde la función getLocationFromAddress

## Dialogo

1. Hola es el asistente de daños de republica dominicana,¿Cómo te ayudamos? [Reportar daños,información]
   1. Si información->información
   2. Si reportar daños ->paso 2
   3. Si otro -> Sino leaparecenlos botones quiere decir que nonos escribe desde la aplicación de Messenger. Sería mejor que nos escribiera desde la app.En caso de que este usando elcelular y no le sea posible, escribanos “No tengo la app”.
   4. Si contiene “no” y “app”->paso 13
2. ¿Esta en un lugar seguro? [Si, No]
   1. Si No->Debería ir a un lugar seguro. En caso de que sea necesario utilice el número de emergencia 911. No dude en escribirnos cuando este seguro
      1. Si escriben->utilice los botones pararesponder
   2. Si Si->paso 3
   3. Si otro->utilice los botones para responder
3. ¿quiere hacer un reportede daños en su hogar o de su comunidad?[Hogar, comunidad]
   1. Si botones se guarda y siguiente
   2. Si otro->utilice los botones para responder
4. Mepodría decir la causa de los daños[Lluvias intensas, deslizamientos, incendios o explosión, huracán,terremoto, violencia, Accidentes de tráfic..., tsunami, Otro]
   1. Si botones se guarda y siguiente
   2. Si otro->guarda en causa y siguiente
5. Ha sufrido daños su vivienda?[No, daños leves, daños graves]
   1. Si botones guarda y siguiente
   2. Si botones se guarda y siguiente
   3. Si otro->utilice los botones para responder
6. ¿Ha habido muertos o heridos?
   1. Si No->guarda heridos y muertos a cero y avanza dos pasos
   2. Si si->¿Cuántas personas han resultado heridads?
   3. Si botones se guarda y siguiente
   4. Si otro->señale el número deheridos utilizándolos números
   5. Si número->guarda en heridos y siguiente
   6. Si no hubo heridos->guarda 0 en heridos ypaso siguiente
7. Si hubomuertos ¿Podría escribirnoscuantos?
   1. Si no hubomuertos->guarda 0 en muertes y siguiente
   2. Si numero->guarda en muertes ysiguiente
   3. Si otro->señale el número demuertos utilizando los números
8. Envienos unafoto de los daños
   1. Si otro aimagen->una foto es de mucha ayuda para ubicar los daños. Envienos una foto de los daños
   2. Si foto->Guardamosimagen y siguiente
9. Envienos su ubicación
   1. Si otro->pasa siguiente
   2. Si ubicación->guarda y avanza dospasos
10. Es importante que nos envie su ubicación paraayudarle. Deberáaceptar esto en el móvil. En otro caso puede escribir su dirección(calle y ciudad).Envienos su ubicación.
    1. Si ubicación->guarda y siguiente paso
    2. Si otro->
       1. Ubicación encontrada->guarda y pasa a siguiente
       2. No encontrada->No hemos encontrado la dirección que nos ha especificado. Por favor compruebe que el nombre esta escrito correctamente evitando el carácter ñ, o díganos la dirección de otro lugar próximo
11. Si quiere hacer alguna observacionañadala en el siguiente mensaje
    1. Guarda y envia a arcgis el reporte. Sigueinte paso
12. Quiere reportar otro daño?[Si, No]
    1. Si no->Muchas gracias por colaborar con el servicio de monitoreo. Su información nos es muy útil para ayudarle. Con el siguiente link podrá avisar a sus amigosde que noshaayudadado con su información (Link compartir)
    2. Reportar->Crea nuevo reporte en el paso 3 y Se sube elanterior al paso 21. Al buscar el reporte con el usuario se sabra que esta ha sido finalizada y tomara el que tenga una fecha más reciente
13. De acuerdo, iniciaremos un reporte sin botones.¿Cual es la causa de los daños producids?
    1. Guarda y siguiente
14. Ha sufrido daños su vivienda? Descríbalos
    1. Guarda y siguiente
15. ¿Ha habido muerto? Indiquenos la cantidad utilizando unnúmero
    1. Si número->guarda y siguiente
    2. Si otro->Indiquenos la cantidad utilizando número
16. ¿Ha habido heridos? Indiquenos la cantidad utilizando unnúmero
    1. Si número->guarda y siguiente
    2. Si otro->Indiquenos la cantidad utilizando número
17. Envienos una imagen de los daños provocados
    1. Si otr>Una foto es de mucha ayuda para ubicar los daños. Envienos una foto de los daños
    2. Si foto->guarda y siguiente
18. Escribanos la dirección del suceso, especificando la calle y ciudad
    1. Si encuentra->guarda y siguiente
    2. Si no->No hemos encontrado la dirección que nos ha especificado, Por favor, compruebe que el nombre esta escrito correctamente, evitando el carácter ñ, o diganos la dirección de otro lugar próximo
    3. Si otro se repite pregunta
19. Si quiere hacer alguna observación añádala en el siguientemensaje
    1. Guarda y siguiente
20. Muchas gracias por colaborar con el servicio de monitoreo. Su informaciónnos es muy útil para ayudarle. Con el siguiente link podráavisar a sus amigos de que nos ha ayudado con su información (Link compartir)
    1. Se sube al paso 21. Al buscar el reporte con el usuario se sabraque esta ha sido finalizada