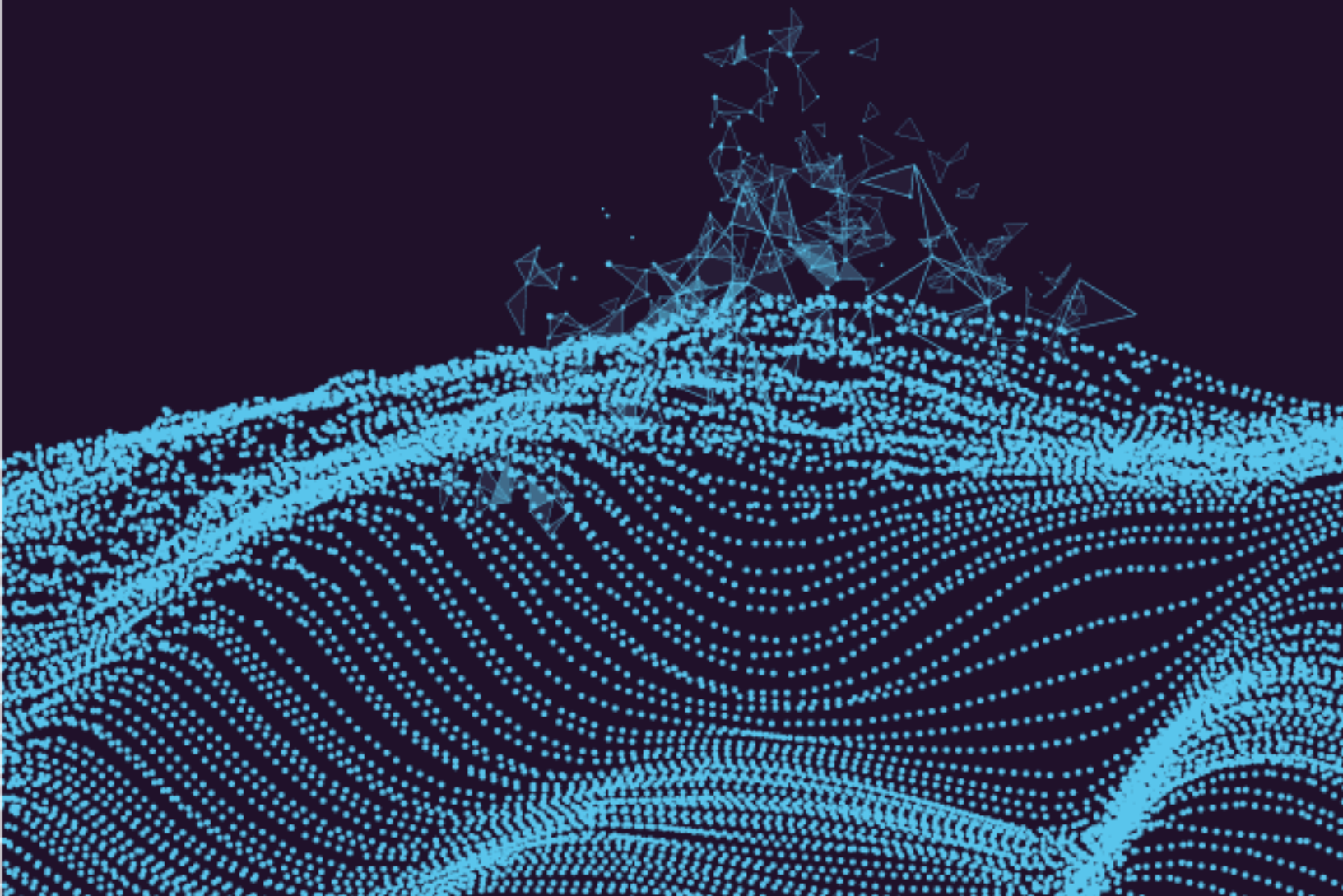


iBest.CLOUD

EVOLUCIÓN INTELIGENTE EN LA NUBE PARA TU EMPRESA

Manual Técnico

FiBi



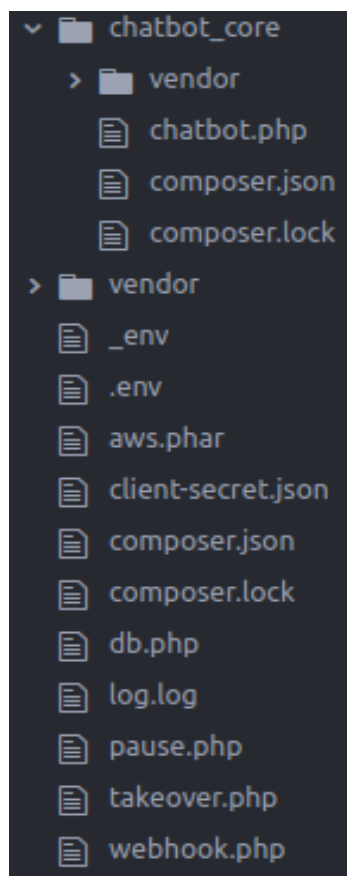


FiBi

En este manual se describe el uso, requerimientos y algunos de los métodos que se desarrollaron para el funcionamiento de FiBi.

FiBi es un bot (interfaz conversacional) para el manejo interno de recursos humanos de Finvivi, se utiliza “Chat-API” como cliente para mandar mensajes por el canal de Whats app, el procesamiento de lenguaje natural es manejado por Dialogflow utilizando “intents” y “text responses”, también esta integrado con “Dashbot” para permitir la entrada de un agente de finvivi a la conversación y detener a FiBi mientras conversa con el usuario. La lógica y la integración están implementadas en php.

A continuación, se muestra la distribución del proyecto.



Chatbot.php

Este archivo se encuentra dentro del directorio chatbot_core.

En este archivo esta implementada la función “reply”, encargada de enviar los mensajes de Whats app al usuario, a travez de Chat-API, mediante http request.

```
public static function reply($toChatId,$message){  
  
    $url = 'https://eu40.chat-api.com/instance36108/message?token=7spqw6hexooumsfu';  
    $data = [  
        'phone' => $toChatId, // Receivers chatId  
        'body' => $message, // Message  
    ];  
  
    $json = json_encode($data); // Encode data to JSON  
    // Make a POST request  
    $options = stream_context_create(['http' => [  
        'method' => 'POST',  
        'header' => "Content-type: application/json",  
        'content' => $json  
    ]  
    ]);  
    // Send a request  
    $result = file_get_contents($url, false, $options);  
}
```

También existe una versión de reply especializada para enviar media, funciona de la misma manera.

```
public static function replyMessageMedia($fileUrl, $toChatId){  
  
    $urlMedia = 'https://eu40.chat-api.com/instance36108/sendFile?token=7spqw6hexooumsfu';  
    $data = [  
        'chatId' => $toChatId, // Receivers chatId  
        'body' => $fileUrl, // Message (Image Url)  
        'filename' => 'file.png' //random name  
    ];  
  
    $json = json_encode($data); // Encode data to JSON  
    // Make a POST request  
    $options = stream_context_create(['http' => [  
        'method' => 'POST',  
        'header' => 'Content-type: application/json',  
        'content' => $json  
    ]  
    ]);  
    // Send a request  
    $result = file_get_contents($urlMedia, false, $options);  
    log_prints("result media--> ".$result);  
    // wait 10 seconds  
    sleep(10);  
}
```



En la clase Chatbot se encuentra el método que envía mensajes al tablero de Dashbot.

```
public static function sendToDashbot($userID, $text){//para mandar los mensajes provenientes del cliente y del bot,  
  
    $url = 'https://tracker.dashbot.io/track?platform=universal&v=10.1.1-rest&type=outgoing&apiKey=6wh6itNRNy7ftoYcC29BD8dZr3S7bqPnY3prYqW';  
    $data = [  
        "text" => $text,  
        "userId" => $userID  
    ];  
  
    $json = json_encode($data); // Encode data to JSON  
    // Make a POST request  
    $options = stream_context_create(['http' => [  
        'method' => 'POST',  
        'header' => "Content-type: application/json",  
        'content' => $json  
    ]  
    ]);  
    // Send a request  
    $result = file_get_contents($url, false, $options);  
}
```



Webhook.php

En webhook.php se reciben los mensajes provenientes del usuario y se reenvían a dialogflow, que se encarga del procesamiento del lenguaje natural y nos regresa un JSON con la información referente al intent que se activo con el mensaje que recibió.

```
$bot = new chatbot;

$projId = 'fibi-5a846'; //código del bot en Dialogflow

$data = json_decode(file_get_contents('php://input'), true);
$to=""; //ChatId or phone
$messageBody="";
$rawMessage="";
$messageResponse = "";

foreach($data['messages'] as $message){
    if($message['fromMe'] === false && $message['body'] != ""){

        $rawMessage = $message['body'];
        $to = $message['author'];

        $intentName = detect_intent_texts($projId, $rawMessage, $to, 'sp-MX');
    }
}
```



En la función “detect_intent_texts” se encuentra la lógica para mantener el flujo de la conversación con FiBi y se envía el mensaje a Dialogflow para su procesamiento.

```
function detect_intent_texts($projectId, $text, $sessionId, $languageCode = 'sp-MX'){  
  // NOTE: @proyect id -> fibi-5a846  
  // NOTE: @$sessionId -> Número de telefono  
  
  //Data base  
  $db = new Db("transformationdb.c9fxro0g2xx9.us-east-1.rds.amazonaws.com", "chatbot", "chatNuevo2018", "finvivirFibi_QA");  
  
  //new session  
  $test = array('credentials' => 'client-secret.json');  
  $sessionsClient = new SessionsClient($test);  
  $session = $sessionsClient->sessionName($projectId, $sessionId ?: uniqid());  
  
  // create text input  
  $textInput = new TextInput();  
  $textInput->setText($text);  
  $textInput->setLanguageCode($languageCode);  
  
  // // create query input  
  $queryInput = new QueryInput();  
  $queryInput->setText($textInput);  
  
  // // get response and relevant info  
  $response = $sessionsClient->detectIntent($session, $queryInput);  
  $queryResult = $response->getQueryResult();  
  $queryText = $queryResult->getQueryText(); //toda la info del json  
  $intent = $queryResult->getIntent(); //lo que el usuario envio al agente  
  $displayName = $intent->getDisplayName(); //objeto intent  
  $confidence = $queryResult->getIntentDetectionConfidence(); //nombre del intent  
  $fulfilmentText = $queryResult->getFulfillmentText(); //confianza de deteccion del intent  
  //text response definida en dialogflow para el intent  
  
  $sessionsClient->close();  
  
  // recuperar de la base de datos  
  $userName = $db->getUserName($sessionId); //current user name  
  $current = $db->getUserSession($sessionId); //current session  
  $correo = $db->getUserEmail($sessionId); //current user email
```



Flujo dentro de “detect_intent_texts” dedicado a pausar el bot, creación de tickets de Zendesk y creación de denuncias encriptadas.

```
if ($current == 'pausa') {
  chatbot::sendToDashbot($sessionId, $queryText);
  if ($displayName == 'play') {
    chatbot::reply($sessionId, "Se ha reanudado la conexión con FiBi");
    chatbot::sendToDashbot($sessionId, "El usuario: " . $userName . " con id: " . $sessionId . " ha puesto en PLAY a FiBi.");
    $db->insertRecord($sessionId, $queryText, "user", "base");
  }
  else {
    $db->insertRecord($sessionId, $queryText, "user", "pausa");
  }
}
else{
  // ***** current *****
  // creacion de ticket //
  if ($current == "ticket") {
    if ($queryText == "cambiar correo" || $queryText == "Cambiar correo" || $queryText == "cambio correo") {
      chatbot::reply($sessionId, "Tu correo actual es: " . $correo . " por favor escribe tu nuevo correo electrónico:");
      $db->insertRecord($sessionId, $queryText, "user", "cambioCorreo");
    }
    else {
      $isValidMail = $db->sendTicket($sessionId, $queryText); // si falla la creacion de ticket, regresa false
      $db->insertRecord($sessionId, $queryText, "user", "base");
      if ($isValidMail == FALSE) {
        chatbot::reply($sessionId, "¡Oh oh! 😞 Ha ocurrido un problema al generar tu ticket, por favor asegúrate de que hayas realizado la verificación de tu correo electrónico.");
      }
      else {
        chatbot::reply($sessionId, "Tu ticket se generó correctamente, en unos momentos mis colegas te brindarán la ayuda necesaria. ¿Te puedo ayudar con algo más? ");
      }
    }
  }
  // creacion de denuncia //
  elseif ($current == "denuncia") {
    chatbot::reply($sessionId, "¡Ya generé tu denuncia con éxito! Mis colegas te ayudarán con ella. ¿Necesitas ayuda con algo más?");
    $db->insertRecord($sessionId, $queryText, "user", "denuncia2", TRUE); //TRUE significa que se encriptará antes de insertar en DB.
    $db->sendMail("denuncia-mail-test-MARY", $queryText);
  }
  elseif ($current == "correo") { //guarda el correo del nuevo usuario
    if (filter_var($queryText, FILTER_VALIDATE_EMAIL)) { // Correo valido
      $db->updateUserEmail($sessionId, $queryText);
      $db->verifyEmail($queryText);
      chatbot::reply($sessionId, "Nuevo correo: " . $queryText . "Por ser tu primer ticket tengo que validar tu cuenta de correo electrónico, ayudame a entrar a tu bandeja de entr");
      $db->insertRecord($sessionId, $queryText, "user", "ticket");
    }
    else { // NOTE: correo invalido
  }
}
```



Continuación del flujo:

```
else { // NOTE: correo invalido
  chatbot::reply($sessionId, '¡Lo siento, pero el correo: ' . $queryText . ' no es un correo valido! Ingresa nuevamente tu correo por favor!');
  $db->insertRecord($sessionId, $queryText, "user", "correo");
}
}
elseif ($current == "cambioCorreo") { //cambio del correo de usuario

  if (filter_var($queryText, FILTER_VALIDATE_EMAIL)) {
    $db->updateUserEmail($sessionId,$queryText);
    $db->verifyEmail($queryText);
    chatbot::reply($sessionId, 'El cambio de correo electrónico implica que tengamos que validar nuevamente tu correo, entra a la bandeja de entrada');
    $db->insertRecord($sessionId, $queryText, "user", "base");
  }
  else {
    chatbot::reply($sessionId, '¡Lo siento, pero el correo: ' . $queryText . ' no es un correo valido! Ingresa nuevamente tu correo por favor!');
    $db->insertRecord($sessionId, $queryText, "user", "cambioCorreo");
  }
}
elseif ($current == "name") { //guarda el nombre del nuevo usuario

  $db->updateUserName($sessionId, $queryText);
  chatbot::reply($sessionId, '¡Gracias ' . $queryText . ' ! ¿En qué te puedo ayudar?');
  $db->insertRecord($sessionId, $queryText, "user", "base");
}
// ***** displayName *****
elseif ($displayName == 'pausa') {
  chatbot::reply($sessionId, "Se ha puesto a FiBi en *pausa*, se le reconectará con un humano");
  chatbot::sendToDashbot($sessionId, "El usuario: " . $userName . " con id: " . $sessionId . " ha puesto en PAUSA a FiBi.");
  $db->insertRecord($sessionId, $queryText, "user", "pausa");
}
elseif ($displayName == 'cambioCorreo') {
  chatbot::reply($sessionId, "Tu correo actual es: " . $correo . " por favor escribe tu nuevo correo electrónico:");
  $db->insertRecord($sessionId, $queryText, "user", "cambioCorreo");
}
elseif ($displayName == 'si' && $current == "denuncia2") {
  chatbot::reply($sessionId, " Dime, ¿En qué te puedo ayudar:
Dudas sobre el negocio 🤔.
Generar un ticket 🎫.
Generar una denuncia 🚨.");
  $db->insertRecord($sessionId, $queryText, "user", "base");
}
}
elseif ($displayName == 'crearTicket') {
```




```

elseif ($displayName == 'crearTicket') {

if ($correo != FALSE) {
    chatbot::reply($sessionId, $fulfilmentText);
    $db->insertRecord($sessionId, $queryText, "user", "ticket");
}
else {
    chatbot::reply($sessionId, "Por favor ingresa tu correo electronico.");
    $db->insertRecord($sessionId, $queryText, "user", "correo");
}
}
elseif ($displayName == 'denuncia') {

    chatbot::reply($sessionId, $fulfilmentText);
    $db->insertRecord($sessionId, $queryText, "user", "denuncia");
}
elseif ($displayName == 'Default Welcome Intent') {

    if ($userName != FALSE) { /// si existe el usuario
        chatbot::reply($sessionId, "¡Hola de nuevo " . $userName . " Dime, ¿En qué te puedo ayudar:
°.- Dudas sobre el negocio 🤔.
°.- Generar un ticket 📄.
°.- Generar una denuncia 🙏σ.");
        $db->insertRecord($sessionId, $queryText, "user", "base");
    }
    else { /// si no existe el usuario

        $db->insertUser($sessionId, "temp");
        chatbot::reply($sessionId, $fulfilmentText . ' Veo que es la primera vez que platicamos ¿cómo te llamas? ');
        $db->insertRecord($sessionId, $queryText, "user", "name");
    }
}
}/// respuestas con fotos
elseif ($displayName == 'vacacionesDias') {

```

```

}/// respuestas con fotos
elseif ($displayName == 'vacacionesDias') {
    chatbot::reply($sessionId, 'Ademas de los días que te corresponden por ley, finvivir te otorga 2 días adicionales por año');
    chatbot::replyMessageMedia('https://s3.amazonaws.com/fibidashbot/diasdevacaciones.PNG', $sessionId);
}
elseif ($displayName == 'finvipesosCalculo') {
    chatbot::reply($sessionId, 'Los finvipesos se asignan conforme a la antigüedad y nivel de puesto, estos estan topados a 2,100. ');
    chatbot::replyMessageMedia('https://s3.amazonaws.com/fibidashbot/calculofinvipesos.PNG', $sessionId);
}
elseif ($displayName == 'escuelaDatosFactura') {
    chatbot::reply($sessionId, 'Recuerda que para programar y pagar una factura de beca escolar, deben venir correctos los datos');
    chatbot::replyMessageMedia('https://s3.amazonaws.com/fibidashbot/datosfactura.PNG', $sessionId);
}
elseif ($displayName == 'incapacidadInfonavitDatos') {
    chatbot::replyMessageMedia('https://s3.amazonaws.com/fibidashbot/datosinfonavit.PNG', $sessionId);
}
elseif ($displayName == 'diasFestivosOficiales') {
    chatbot::replyMessageMedia('https://s3.amazonaws.com/fibidashbot/diasfestivos.PNG', $sessionId);
}
else { //          flujo normal          //

    chatbot::reply($sessionId, $fulfilmentText);
    $db->insertRecord($sessionId, $queryText, "user", "base");
}
}
return $displayName;
}
?>

```



“detect_intent_texts” retorna el nombre del intent que se activo.

Estructura del JSON que recibimos de Dialogflow:

```
{
  "responseId": "ea3d77e8-ae27-41a4-9e1d-174bd461b68c",
  "session": "projects/your-agents-project-id/agent/sessions/88d13aa8-2999-4f71-b233-39cbf3a824a0",
  "queryResult": {
    "queryText": "user's original query to your agent",
    "parameters": {
      "param": "param value"
    },
    "allRequiredParamsPresent": true,
    "fulfillmentText": "Text defined in Dialogflow's console for the intent that was matched",
    "fulfillmentMessages": [
      {
        "text": {
          "text": [
            "Text defined in Dialogflow's console for the intent that was matched"
          ]
        }
      }
    ],
    "outputContexts": [
      {
        "name": "projects/your-agents-project-id/agent/sessions/88d13aa8-2999-4f71-b233-39cbf3a824a0/contexts/generic",
        "lifespanCount": 5,
        "parameters": {
          "param": "param value"
        }
      }
    ],
    "intent": {
      "name": "projects/your-agents-project-id/agent/intents/29bcd7f8-f717-4261-a8fd-2d3e451b8af8",
      "displayName": "Matched Intent Name"
    },
    "intentDetectionConfidence": 1,
    "diagnosticInfo": {},
    "languageCode": "en"
  },
  "originalDetectIntentRequest": {}
}
```



db.php

en este archivo se conecta y se maneja la base de datos.

Cuenta con las funciones:

- encrypt() //encripta los datos de la denuncia para la BD.
- updateUserName() //modificación de nombre de usuario
- updateUserEmail() //modificación del correo de usuario
- getUserId() //retorna el ID del usuario (serial)
- getSession() //retorna el estado actual de la sesión del usuario
- getUsername() //retorna el nombre del usuario
- getUserEmail() //retorna el email del usuario
- connect() //inicia la conexión con la base de datos
- insertUser() //inserta un usuario a la base de datos
- insertRecord() //ingresa un nuevo registro (mensaje)
- verifyEmail() //envia un correo para verificar el user email
- sendMail() //envia el correo de denuncia y llama a encrypt
- sendTicket() //envia un correo al correo de soporte de zendesk.

Webhook.php, pause.php y takeover.php trabajan con la misma base de datos, esto facilita la integración y mantiene las sesiones de los usuarios en orden.



pause.php

Este archivo funciona como un webhook para dashbot, sirve para detener el bot cuando un agente pone pausa a una conversación dentro del tablero de dashbot.

```
//Conexión con dashBot
$dataREST = json_decode(file_get_contents('php://input'), true); //recibimos el post desde dashBot
$idPause = $dataREST['userId']; //al hacer pausa se settea =userId de chatapi (webhook.php)
$paused = $dataREST['paused'];

// Conexión con el chatbot:
$bot = new chatbot;

//Conexión con base de datos
$db = new Db("transformationdb.c9fxro0g2xx9.us-east-1.rds.amazonaws.com", "chatbot", "chatNuevo2018", "finvivirFibi_QA");
$username = $db->getUserName($sessionId); //current user name

if ($paused == TRUE) {
    $bot->sendToDashbot($idPause, "se ha puesto PAUSA al bot, usuario con nombre: " . $username);
    $bot->reply($idPause, "El agente de finvivir ha puesto *PAUSA* a FiBi");
    $db->insertRecord($idPause, 'dashbot: *PAUSA*', "user", "pausa"); //dashbot: anuncia el origen del mensaje
}

if ($paused == FALSE) {
    $bot->sendToDashbot($idPause, "se ha puesto PLAY al bot");
    $bot->reply($idPause, "El agente de finvivir ha puesto *PLAY* a FiBi");
    $db->insertRecord($idPause, 'dashbot: *PLAY*', "user", "base"); //dashbot: anuncia el origen del mensaje
}
```



takeover.php

Este archivo funciona como un webhook para dashbot, cuando un agente manda un mensaje desde el tablero de dashbot, takeover.php lo recibe y utiliza una instancia de chatbot para enviar al usuario el mensaje via whatsapp.

```
//Conexión con dashBot
$dataREST = json_decode(file_get_contents('php://input'), true); //recibimos el post desde dashBot
$idSend = $dataREST['userId']; //al hacer pausa se settea =userId de chatapi (webhook.php)
$text = $dataREST['text']; //mensaje de texto proveniente de dashbot

//Conexión con base de datos
$db = new Db("transformationdb.c9fxro0g2xx9.us-east-1.rds.amazonaws.com", "chatbot", "chatNuevo2018", "finvivirFibi_QA");
$username = $db->getUserName($sessionId); //current user name

//conexión con el bot
$bot = new chatbot;

$bot->reply($idSend, $text); //mandar mensaje de whatsapp con texto de dashbot
$bot->sendToDashbot($idSend, $text);
$db->insertRecord($idSend, 'dashbot: '.$text, "user", "pausa"); //dashbot: anuncia el origen del mensaje
```

Dialogflow

Dialogflow sirve para desarrollar tecnologías de interacción Humano-Computadora basadas en conversaciones en lenguaje natural, desarrollado por Google.

The screenshot shows the Dialogflow console for an agent named 'Fibi'. The interface includes a left sidebar with navigation options like Intents, Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, and Prebuilt Agents. The main area is titled 'Fibi' and has a 'SAVE' button. Below the title are tabs for 'General', 'Languages', 'ML Settings', 'Export and Import', 'Environments', 'Speech', and 'Share'. The 'General' tab is active, showing a description field with the placeholder 'Describe your agent', a 'DEFAULT TIME ZONE' dropdown menu set to '(GMT-6:00) America/Chicago', and a 'GOOGLE PROJECT' section with a table:

Project ID	fibi-5a846
Service Account	dialogflow-jmfyym@fibi-5a846.iam.gserviceaccount.com

Below the table is an 'API VERSION' section.

En Dialogflow, el flujo básico de conversación implica estos pasos:

- El usuario da un input.
- El agente de Dialogflow analiza esa entrada.
- El agente devuelve una respuesta al usuario en forma de JSON.

Para definir cómo funcionan las conversaciones, se crean intenciones (intents) en el agente, que asignan las entradas del usuario a las respuestas. Cada intent define que ejemplos de expresiones (training phrases) pueden desencadenar la intención, que extraer de la expresión y como responder(text responses).



The screenshot displays the Dialogflow 'Intents' management interface. On the left, a sidebar lists various components: 'Fibi' (with a dropdown and settings gear), 'es' (language), 'Intents' (selected), 'Entities', 'Knowledge [beta]', 'Fulfillment', 'Integrations', 'Training', 'History', and 'Analytics'. The main content area is titled 'Intents' and features a 'CREATE INTENT' button. Below the title is a search bar labeled 'Search intents'. A pagination control shows '5 OF 9' items. A list of intents is displayed, each with a blue selection dot. The selected intent is 'garantiasAutorizadas'. Other visible intents include 'faltasCobranzaCanje', 'formatoAN', 'garantiaCuentaLiquidada', 'garantiaResguardo', 'garantiasSemana', 'garantiaVenderLiquidar', 'gastoPorReembolsar', and 'gastoReembolsarSinFactura'.

Las expresiones que le damos como ejemplo se utilizan para entrenar su modelo, por tanto no es necesario que el usuario escriba las frases de forma idéntica. El modelo aprende a identificar que busca el usuario.



Dashbot

Tablero de Dashbot:

dashbot Bots / Skills ▾ Tour Competitive Intelligence Docs Pricing Blog Account Log Out

[Add Chatbot](#)

Your Bots & Skills [Cross Channel Dashboard](#)

Account
Chatbots / Skills
Organizations
Settings

Your Bots

[Add a Bot, Skill, or Action in Your Bots](#)

Name ▾	Platform ▾	Data Received in Last Hour	Lifetime Users ▾
Fibi	Dev Universal		-
Mary	Dev Universal		-

[Add a Bot, Skill, or Action](#)



Sección de live transcripts:

En esta sección el agente puede detener a FiBi y tomar su lugar en la conversación, esto sucede también si el usuario pausa a FiBi.

dashbot Bots / Skills Tour Competitive Intelligence Docs Pricing Blog Account Log Out

Cross Channel Dashboard

521332324969... 11:24:51 AM
El usuario: Martin con id: 5213323249694@c.us Last: 12:17:47 PM
ha puesto en PLAY a FiBi. 0 m / 14 Out

Live FiBi: Started: April 22, 2019 11:24 AM
Send Pause Bot

El usuario: Martin con id: 5213323249694@c.us ha puesto en PLAY a FiBi.
Monday, April 22nd 2019, 12:17:47 PM (-)

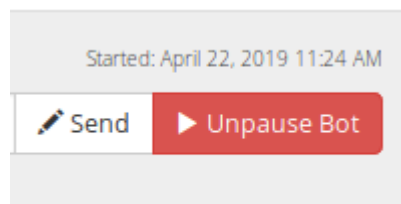
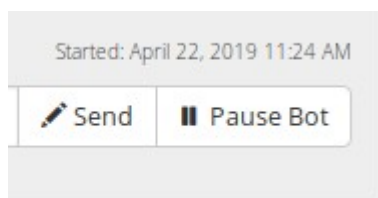
Play
Monday, April 22nd 2019, 12:17:44 PM (-)

El usuario: Martin con id: 5213323249694@c.us ha puesto en PAUSA a FiBi.
Monday, April 22nd 2019, 12:17:34 PM (-)

se ha puesto PLAY al bot
Monday, April 22nd 2019, 11:27:03 AM (-)

El usuario: Martin con id: 5213323249694@c.us ha puesto en PLAY a FiBi.
Monday, April 22nd 2019, 11:26:48 AM (-)

Help



Av. Unión #163 Piso 1 Col. Lafayette Guadalajara, Jalisco, México
+(52) 33 2014 4391 hola@inbest.cloud www.inbest.cloud

