



FACULTAD de CIENCIAS
de la COMPUTACIÓN

BENMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

**FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN**

“COACH VIRTUAL PARA METS”

TESIS

Para Obtener el Grado de
Licenciatura en Ciencias de la Computación.

PRESENTA:

RUBÉN ESPINOSA RODRÍGUEZ

ASESOR:

**DRA. MARÍA DE LA CONCEPCIÓN PÉREZ DE CELIS
HERRERO**

Puebla, Pue. Noviembre 2020



Dedicatoria

El adquirir conocimiento nunca es suficiente, ese conocimiento se vuelve vacío si no se aplica, lo cual se logra con perseverancia y esfuerzo. Este trabajo de investigación no es la excepción, y es por ello que agradezco inmensamente a aquellos que me acompañaron durante esta travesía y a ellos es a quienes se lo dedico.

Primeramente a mi Madre que ella es la base de todo lo que soy hoy día y se lo agradezco infinitamente, a mi Abuela Carmen, que es parte de mi aunque ella ya no esté, a los chicos y chicas del laboratorio que pusieron su granito de arena, a mis amigos como E. Rodríguez e I. Machorro, que entre risas, ideas y críticas sirvieron de apoyo al igual que X. Aline por su ayuda y comprensión durante este tiempo, a mis profesores cercanos, el profesor Ríos Acevedo y mi tutora la Dra. Pérez de Celis. Y, por último, pero no menos importante a la Dra. M. Morales que me brindo inspiración y ganas necesarias para continuar gracias a su dedicación y profesionalismo.

“To live is to risk it all; otherwise you're just an inert chunk of randomly assembled molecules drifting wherever the universe blows you...” -R.S.

Índice

Introducción -----	5
Capítulo 1. Descripción del problema a tratar y propuesta de solución -----	6
1.1 Teorías de Cambio de Comportamiento -----	6
1.2 Terapias cognitivo-conductuales -----	7
1.3 Modelo de Adopción de las Precauciones -----	9
1.4 Utilización de la inteligencia artificial en el cambio de comportamiento -----	10
1.5 E-terapias con Chatbot-----	11
1.6 Descripción del problema a tratar-----	12
1.7 Propuesta de solución -----	13
1.8 Alcanzables y Restricciones -----	14
Capítulo 2. Fundamentos Teóricos -----	14
2.1 ¿Qué es un Chatbot? -----	14
2.2 ¿Qué es una aplicación web? -----	15
2.3 ¿Cómo se compone una aplicación web?-----	15
2.4 ¿Qué son las APIs?-----	15
2.5 ¿Qué es una API REST? -----	16
2.6 ¿Qué es el procesamiento de lenguaje natural?-----	16
2.7 APIs de procesamiento de lenguaje natural -----	17
2.8. Herramientas de desarrollador de Facebook-----	17
2.9 Messenger de Facebook como herramienta de desarrollador de Facebook -----	18
2.10 ¿Cómo se hace un Chatbot? -----	18
2.11 ¿Para qué sirven los Chatbots?-----	19
2.12 Diseño de un Chatbot -----	19
2.13 Integración de la psicología en un Chatbot -----	20
2.14 SLOGAN y memoria a largo plazo -----	21
Capítulo 3. Desarrollo de la solución -----	22
3.1 Requerimientos de la solución -----	22
3.2 Un healthbot llamado Metsbot y su psicología -----	22
3.3 Canvas para Metsbot -----	23
3.4 Investigación de campo-----	24
3.4.1 Resultados del análisis del cuestionario aplicado.-----	25
3.5 Consideraciones de los resultados de la encuesta aplicables al diseño de Metsbot -----	30
3.6 Diagramas para el flujo de la conversación -----	31
Figura 7. Flujo de la conversación de Metsbot-----	31
Capítulo 4. Implementación de Metsbot -----	32

4.1 La anatomía de Metsbot	32
4.2 Diseño de la interfaz de usuario de Metsbot	33
4.2.1 Los personajes	33
4.3 Conectividad a la Base de Datos de la plataforma METS	35
4.4 Creación de la API y configuración del Servidor	36
4.4.1 Reglas de Negocio	36
4.5 Funciones del Chatbot	38
4.6 Funciones de la API	41
4.7 Creación de una app en Facebook	42
4.8 Chatfuel la plataforma líder de Chatbot para Messenger Facebook	43
4.9 Descripción de la anatomía de Metsbot	44
4.9.1 Componente de Usuario	45
4.9.2 Componente de Chatfuel	46
4.9.3 Estructura de la conversación para Metsbot en Chatfuel.	53
4.9.4 Mensajes de la API	104
4.9.5 Subcomponente de Chatfuel, el lenguaje y la conversación	105
4.9.6 Componente de la API	107
4.9.7 Ejemplo de la integración de todos los componentes	114
4.9.8 Ejemplo de la integración de todos los componentes para complementar sus limitaciones.	124
4.10 Limitantes de Metsbot	129
4.11 Actualizaciones	132
Capítulo 5. Resultados	133
5.1 Plataforma Interactiva METS	137
5.2. Resultados seguimiento de uno de los usuarios de Mets	139
Capítulo 6. Conclusiones	142
Capítulo 7. Referencias	144

Introducción

Los factores conductuales desempeñan un papel determinante en cada una de las doce principales causas de muerte [1], incluidas las enfermedades crónicas como la enfermedad cardíaca, el cáncer y el accidente cerebrovascular, que son las principales causas de muerte en los países desarrollados. Los factores conductuales más comunes que contribuyen a la mortalidad o muerte incluyen el uso de alcohol, tabaco, y la conducción irresponsable de un vehículo; patrones de dieta y actividad; comportamiento sexual; y el uso ilícito de drogas.

Un estilo de vida saludable incluye alimentación saludable, actividad física diaria, reducción de la conducta sedentaria, protección contra el sol y exámenes regulares de detección del cáncer. Además de las elecciones de estilo de vida que realiza, la salud está determinada, por el tipo de ambiente en el que el individuo y su familia viven, trabajan, aprenden y disfrutan; así como por las políticas públicas, que impactan en su vida diaria [2].

La actividad física ha demostrado ser un factor de prevención importante, en al menos 35 patologías crónicas y además constituye un elemento esencial, para mantener la calidad de vida de las personas a lo largo de los años. Sin embargo, en los últimos treinta años, ha habido un preocupante descenso de los niveles de actividad física en la población general que difícilmente alcanza los niveles recomendables para lograr beneficios sobre la salud. El cómo propiciar que los individuos incrementen su actividad física se ha convertido en un tema de estudio importante de la salud pública ya que la actividad física, es un factor conductual de las enfermedades no transmisibles (ENT).

Estar físicamente activo y minimizar el tiempo que se pasa sentado (comportamiento sedentario) es una forma importante de mantener la buena salud y reducir el riesgo de muchas enfermedades crónicas. De acuerdo a los resultados de ENSANUT 2018 [3], en nuestro país en lo relacionado a las ENT que tienen como factor de riesgo la falta de actividad física y el sedentarismo existen 15 millones de personas, con Hipertensión, cabe señalar que si bien estos datos consideran a las personas mayores de 20 años, Conforme se incrementa la edad, crece el porcentaje de población con diagnóstico previo de hipertensión, principalmente a partir de los 50 años, llegando al 26.7% en el grupo de 70 a 79 años en 2018. Por otra parte, la Prevalencia de sobrepeso y obesidad en la población de 5 a 11 años de edad aumentó del 32.4% en el 2012 a 35.6% en el 2018; para el grupo poblacional de 12 a 19 años varió de 33.2% a 35.8% para los mismos años y finalmente para la población de 20 años y más pasó del 71.3% al 75.2%. Cabe señalar si consideramos estos datos por sexo para la población de 20 años y mas encontramos que la prevalencia de sobrepeso y obesidad en las mujeres paso de 73% en el 2012 a 76.8% en el 2018 y para los hombres esta prevalencia se incremento también pasando del 69.4% en el 2012 al 73% en el 2018.

En lo relacionado al panorama nacional de la práctica de actividad física se tiene que l población de 20 a 69 años el 29% realiza menos de 150 minutos/semana, el 17.9 menos de 420 minutos/semana, el 16.1% menos de 840 minutos/semana, el 17.0% menos de 1680 minutos/semana y solamente el 20% realiza más de 1680 minutos/semana [3].

Adicionado a lo anterior la OMS, ha evidenciado que la inactividad física es el cuarto factor de riesgo en lo que respecta a la mortalidad mundial (6% de las muertes registradas en todo el mundo). Además, se estima que la inactividad física es la causa principal de aproximadamente un 21%-25% de los cánceres de mama y de colon, el 27% de los casos de diabetes y aproximadamente el 30% de la carga de cardiopatía isquémica.

A pesar de contar actualmente, con evidencia confiable de los efectos positivos y la importancia de los hábitos saludables, existen sectores de la sociedad en los que persiste la falta de determinación para abandonar aquellos comportamientos perjudiciales para la salud como pueden ser: el tabaquismo, alimentación no

saludable, el sedentarismo o la falta de actividad física. Por otra parte, podemos también identificar personas que ya iniciaron su travesía para lograr el cambio deseado pero que no son constantes.

En este estudio nos concentramos en dos de estos comportamientos: la falta de actividad física y el sedentarismo. Para los profesionales de la salud es importante buscar herramientas que ayuden a motivar y acompañar a los pacientes en este desafío al cambio; El reto actual es conseguir que las posibilidades que las TIC ponen a nuestra disposición contribuyan a una mejora de la calidad de vida y el bienestar de las personas y ayuden en el ámbito de prevención y promoción de la salud en particular en el proceso de adquirir hábitos saludables. Es bajo esta de premisa surge la inquietud, de diseñar una estrategia, que incentive, una vez que se ha tomada la decisión de modificar una conducta, a paulatinamente abandonar los hábitos no deseados y sustituirlos por hábitos protectores de la salud.

Capítulo 1. Descripción del problema a tratar y propuesta de solución

Previo a la descripción del problema y la presentación de la propuesta de solución consideramos que es importante presentar los diferentes componentes teóricos relacionados con las terapias de cambio de comportamiento y como estos, son considerados en nuestra propuesta de solución; para poder implementar una terapia cognitivo conductual utilizando tecnologías de la información y la comunicación.

1.1 Teorías de Cambio de Comportamiento

Cualquier estrategia que se proponga, para motivar al cambio de comportamiento o para incrementar la adherencia a comportamientos saludables, debe estar sustentada por una Teoría de Cambio de Comportamiento [4],[5]; pues éstas pueden ayudarnos a comprender la índole de los comportamientos en los que se quiere actuar y diseñar soluciones específicas.

Existen diferentes aproximaciones teóricas para abordar el cambio de comportamiento o conducta, además pueden tener un enfoque individual o grupal; y finalmente son aplicables tanto en entornos privados, como públicos. El marco general para comprender las diferentes visiones pasa por definir dos polos dentro de la teoría *Social Cognitiva*: el individual-biosociológico y el social-ambiental.

Existen múltiples teorías y modelos sociopsicológicos en polo individual-biopsicológico del comportamiento, muchos de los cuales se han empleado para diseñar y poner en marcha programas de promoción de la salud con mayor o menor éxito. A continuación, presentamos algunos de los modelos y teorías que utiliza la psicología de la salud¹ para la promoción de la salud:

Teoría cognitiva social

De acuerdo a las teorías cognitivas, los humanos somos más complejos que una serie de respuestas a estímulos externos. La teoría cognitiva social [6], enfatiza que el comportamiento está influenciado por el ambiente y los factores personales. El comportamiento de una persona puede ser influenciado por sus creencias, actitudes y percepciones. Los conceptos centrales en la teoría cognitiva social son aquellos relacionados con las habilidades y la auto-eficacia. Si una persona recibe un incentivo relacionado con una conducta específica, cree que es capaz de desempeñarlo (Auto-eficacia). El éxito en el desempeño del comportamiento aumenta la probabilidad de que este comportamiento se repita en el futuro.

¹ Aunque la Psicología de la Salud no tiene una definición oficial, Matarazzo (1982) la describió como un campo que **engloba aportaciones diversas de la psicología** en relación a la educación, la ciencia y la profesión, aplicadas a la salud y a la enfermedad. https://www.ecured.cu/Psicolog%C3%ADa_de_la_Salud

Modelo de Creencia en la Salud

Modelo de Creencias de Salud [7], creado en la década de los cincuenta como marco teórico para explicar comportamientos de salud y preventivos de la enfermedad, Se trata de un modelo de inspiración cognitiva que considera a los comportamientos como resultados de una serie de creencias y valoraciones que tienen las personas de manera interna. De acuerdo al Modelo de Creencias de Salud las creencias y valoraciones son las siguientes 1) El comportamiento depende de si las personas consideran que son susceptibles a un problema de salud en particular. 2) Consideran que se trata de un problema serio. 3) Están convencidos de que el tratamiento o las actividades de prevención son eficaces. 4) El cuarto concepto se relaciona con la percepción de una barrera que impide la toma de acciones; las barreras pueden variar y pueden incluir aspectos financieros, culturales o de formación.

Teoría de la Acción Razonada

El enfoque primario de esta teoría está en las intenciones, actitudes y creencias [8]. De acuerdo a esta teoría, el intento de una persona de llevar a cabo un comportamiento específico está motivado principalmente por la intención. Las intenciones de una persona, sus creencias en salud, son influenciadas por dos factores clave. Uno, su nivel de intención es mayor si tiene una actitud positiva con respecto al comportamiento. Segundo, su nivel de intención es mayor si están motivados a cumplir una norma social.

Modelo Transteórico

Este modelo explica el cambio en el comportamiento como una serie de seis etapas que atraviesa una persona relacionado con su preparación para el cambio. Las etapas son: i) Pre-contemplación; ii) Contemplación; iii) Preparación; iv) Acción; v) Mantenimiento; vi) Terminación. Utilizando esta teoría, se deben utilizar intervenciones específicas para influenciar en el cambio de comportamiento de acuerdo a la etapa en la que se encuentre la persona.

Existen también otros modelos que utilizan como punto de partida el Modelo de Creencias de Salud como son: La teoría del aprendizaje social y el Modelo de etapas en el cambio individual de conducta, los cuáles consideran que las diversas variables demográficas, socio psicológicas y estructurales son las que afectan la percepción de las personas y de esa forma influyen indirectamente en sus conductas de salud. Es importante destacar que el Modelo de Creencias de Salud y sus variantes, han permitido el desarrollo de otros modelos modernos que son más eficaces y específicos como el Modelo del Proceso de Aceptación de las Precauciones (PAP) con una perspectiva cognitivo-conductual y la Terapia Cognitivo-Conductuales (TCC) los cuales se discutirán en los apartados siguientes.

1.2 Terapias cognitivo-conductuales

La terapia cognitivo conductual (TCC) es un tipo de tratamiento psicoterapéutico que se focaliza en los patrones de pensamiento o “cogniciones” y en el comportamiento, de esta forma ayuda a los pacientes a comprender pensamientos y sentimientos que están influyendo en su comportamiento. La TCC combina la *Terapia Cognitiva* y *Terapias de Comportamiento*. El enfoque se centra en los pensamientos, emociones, sensaciones físicas y acciones, y enseña a los individuos cómo cada uno puede tener un efecto sobre el otro. La forma en la que funciona la TCC es entendiendo los problemas complejos y desglosándolos en partes más pequeñas viendo como están conectadas entre sí y como afectan, de ahí se derivan pensamientos, emociones,

sensaciones físicas y comportamientos, los problemas que se tienen se pueden esquematizar como un “círculo vicioso” como se muestra en la figura 1. La TCC es útil para tratar muchos trastornos, incluyendo la depresión, la ansiedad y las fobias. La premisa detrás de la TCC es que tanto nuestros pensamientos como nuestros comportamientos tienen un efecto sobre nosotros mismos y también sobre los demás. La terapia examina los comportamientos aprendidos y patrones de pensamiento negativos para transformarlos en positivos. A diferencia de otras terapias, la TCC se centra en el presente y mira hacia el futuro. Aunque los eventos y experiencias pasadas son considerados durante la terapia, la atención se centra más en las cuestiones y dilemas actuales.

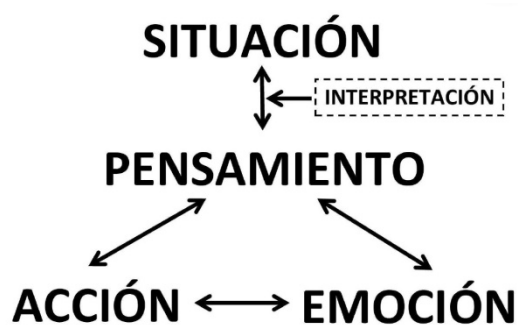


Figura 1. Ciclo elemental de comportamiento en el que una situación dependiendo de la interpretación genera una emoción que provoca una acción. Este proceso se repite en forma recursiva

La TCC ayuda a romper este círculo vicioso de pensamientos, sentimientos y comportamientos negativos. Tras ver las partes con claridad se pueden cambiar; para así lograr un cambio en “cómo se siente”. Pero la TCC tiene como objetivo lograr que este cambio lo haga la persona involucrada, el paciente; y de esta forma pueda elaborar sus propios métodos de afrontar estos problemas siguiendo una dinámica que le permita transformar su comportamiento asociado a situaciones o problemas cotidianos.

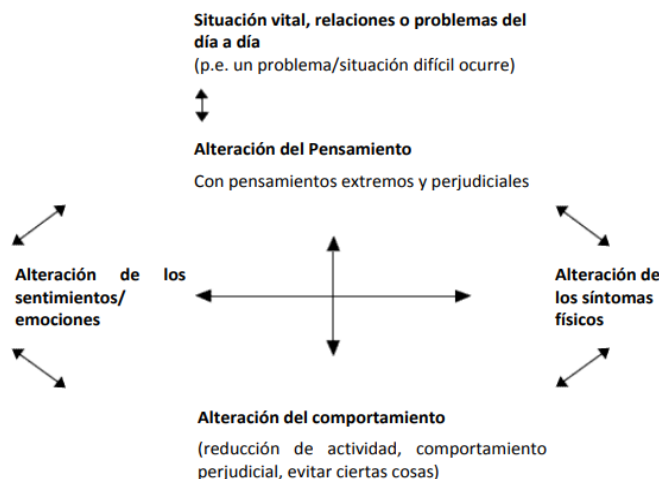


Figura 2. Dinámica de las TCC.

La TCC y el plan de acciones mostrado en la figura 2, se han utilizado para otro tipo de trastornos que tienen que ver con la salud; la bulimia y anorexia por ejemplificar los más comunes, pero también se plantea el uso de TCC para el problema de la obesidad e incluso en ámbitos más específicos como la adherencia al ejercicio en personas con tratamiento contra la obesidad, para los cuales la terapia cognitivo-conductual es de gran utilidad [9].

Otros estudios mencionan que los resultados de la TCC son eficaces, para mejorar la adherencia y disminuir la deserción de un programa de abordaje integral para la reducción de peso en adolescentes con obesidad. En el corto y largo plazo la incorporación de la TCC podría constituir una herramienta útil para hacer frente al complejo problema que es la obesidad, lo que se tendría que traducir en la disminución de su prevalencia [10].

La TCC aplicada en tratamientos de activación física especialmente, en el área de la adherencia a estos tratamientos, explica que la incorporación de ejercicio como un comportamiento constante en su estilo de vida no es fácil para los pacientes, debido a la poca tolerancia que tienen al ejercicio, la falta de motivación para hacerlo u otros factores como pueden ser: experiencias haciendo ejercicio, tiempo y hasta presión social [11]. Es ahí donde se pone en práctica la TCC para lograr cambios combatiendo problemas como la determinación propia para adoptar estos hábitos que tienen que ver con la relación entre lo que la persona cree puede hacer y el esfuerzo que pone al hacerlo. Cabe señalar que es común identificar que las personas con expectativas más altas, cuando tratan de adoptar nuevos hábitos, tienden a rendirse provocando la falta de adherencia dejando como hipótesis que el bienestar psicológico de la persona también es relevante a la adherencia ya que el desempeño puesto en sus esfuerzos por realizar cambios de comportamiento probablemente se ven limitados por el estrés mental que generan. El propósito de los estudios de investigación en psicoterapia es demostrar que las personas pueden cambiar conductas o estilos de vida indeseables. Estos trabajos reportan que muchos individuos cambian y luego vuelven a recaer, o que el proceso de cambio en alguna toma un tiempo demasiado largo y que, por último, otros no logran cambiar en absoluto, razón por la cual se requiere utilizar modelos de cambio de comportamiento *Ad hoc* a los objetivos previstos, como lo es el modelo PAP que describimos a continuación.

1.3 Modelo de Adopción de las Precauciones

La importancia de la psicología en las estrategias de educación en salud es muy importante y esto es algo que se contempló con mayor formalidad en 1986 durante la segunda conferencia Internacional sobre Promoción de la Salud realizada en ese mismo año donde se empezó a plantear elementos de psicología para el apoyo directo de la promoción de la salud entre ellos se formula lo que posteriormente se nombraría como proceso de adopción de precauciones [12].

El Proceso de Adopción de Precauciones (PAP), modelo desarrollado por Weinstein, cuestiona la forma como una persona adopta una precaución [13]. El PAP asume que se reduce un riesgo cuando la persona tiene una motivación para hacerlo. Los casos en los que se mantiene el riesgo, a pesar de reconocer el peligro, probablemente ocurren porque tienen barreras como la falta de información y motivación, la pobre percepción de susceptibilidad y vulnerabilidad propia, la alta percepción de los costos y, finalmente, la falta de intención que les impide emitir la conducta. Además, concibe el cambio como resultado primero de un funcionamiento cognitivo y luego de la planeación. Para que se presente el progreso a través de las etapas se necesita conocer la severidad de los riesgos, la susceptibilidad personal, la percepción de los costos y tener clara percepción de los beneficios.

Este modelo menciona que a través de 7 etapas se puede lograr el cambio de comportamiento.

- Etapa 1. La persona no reconoce que tiene un riesgo y está desprevenida por ende esta etapa se centra solo en informar.
- Etapa 2. La persona reconoce el riesgo, pero no hace nada, se asume que la persona no se ve a sí misma como vulnerable por ende no le interesa cambiar. Esta etapa se centra en hacer a la persona consciente de que es susceptible al riesgo y lo severo que puede ser el problema.
- Etapa 3. La persona ve como algo posible el realizar un cambio, pero no lo hace, se le ayuda a la persona a ver los pros y contras de tomar esa decisión para que la persona parezca estar convencida.

El balance decisional se presenta cuando la decisión es el producto del reconocimiento entre los aspectos favorables (pros) y los desfavorables (contras) con sus pesos relativos [14]

- Etapa 4. La persona ya decide que tiene que hacer, esta etapa es en la que se encuentra más vulnerable a desistir.
- Etapa 5. La persona decide actuar y realiza un plan de acción de cómo y cuándo va a realizarlo. Con el estímulo correcto aumenta la probabilidad de que se logre el comportamiento sano.
- Etapa 6. En esta etapa la persona no solo tiene la intención, sino que la lleva a cabo, el trabajo se centrará en el autocontrol y manejo de recaídas.
- Etapa 7. Se mantiene la conducta saludable por bastante tiempo y se controlan las situaciones que pueden generar recaídas.

Este modelo es una metodología para la promoción de salud siguiendo los postulados del modelo de competencia. El modelo de adopción de precauciones se lleva a cabo, generalmente a través de talleres con grupos vulnerables, diseñados y dirigidos por especialistas diversos que han sido previamente entrenados (psicólogos, médicos, enfermeras y pedagogos) [15].

1.4 Utilización de la inteligencia artificial en el cambio de comportamiento

El cambiar comportamientos resulta ser un proceso que toma tiempo y requiere siempre de apoyo donde la parte difícil es comprometerse al cambio y seguir adelante. Incluso la Asociación Americana de Psicología (APA), sugiere cinco consejos para ayudar a realizar cambios positivos y duraderos en el comportamiento [16]: Hacer un plan duradero, empezar con moderación, cambiar un comportamiento a la vez, involucrar a un compañero y pedir apoyo. Es normal tener tropiezos en el camino a las metas, pero el recuperarse y seguir adelante es lo que hace la diferencia entre lograr el cambio o desistir.

Vamos a centrarnos en los últimos dos consejos donde se habla de involucrar a alguien más y pedir apoyo, estos consejos son fundamentales según la APA, pero lamentablemente no siempre se logran seguir u obtener, ahí es donde entra una de las ramas que componen la inteligencia artificial en su amplia gama de aplicaciones y en esa rama existen muchas aplicaciones que intentan ayudar a cambiar comportamientos y eliminar malos hábitos, por ejemplo la aplicación *Freedom* que bloquea internet para concentrarse en el trabajo, o *Lift* donde se permite elegir qué comportamiento lograr como “correr” o “desayunar”, estas aplicaciones llegan a ser eficientes cuando hay un sistema de “recompensas” o de ver un progreso en algún gráfico [17]. "Lo que hemos aprendido en los últimos 10 a 15 años es que hay un comportamiento automático y luego hay una recompensa después, que es realmente importante porque así es como nuestros cerebros se adhieren a los comportamientos", dice Charles Duhigg, reportero de negocios en el New York Times y autor del libro *El Poder de los hábitos* [18]. Pero, ¿funcionan las aplicaciones?, la respuesta se reduce al comportamiento que se trataba de cambiar ya que según Arun Sundararajan, profesor de la escuela de Negocios Stern de NYU, hay tres tipos de cambios de comportamiento: 1) El que incluye comportamientos cambiantes aprendidos por la experiencia, 2) Volver a capacitar el sistema biomecánico para comportarse de manera diferente y 3) Comportamientos fisiológicos [19]. Siendo estos últimos dos los que mayor probabilidad tienen de funcionar, ya que no es un cambio tal cual, sino un entrenamiento para que se haga algo diferente y es por ello que este tipo de aplicaciones llega a tener tanta popularidad entre los usuarios ya que no hay una aplicación que evite tener malos hábitos pero la tecnología ayuda a seguir nuestro progreso y recordarnos cuando hay que hacer las cosas como apoyo a la decisión final que solo nosotros podemos tomar.

Entonces ya tenemos conocimiento que las apps por sí solas pueden ayudar, aunque la tendencia tecnológica es hacia las cosas inteligentes, es decir los objetos que con inteligencia artificial y aprendizaje automático pueden interactuar de forma “inteligente” con las personas y el entorno. Consultoras como Gartner, empresas como *PatientsLikeMe* entre otros grandes del sector salud y tecnología concuerdan que toda mejora de salud implica cambiar comportamientos y que se necesitan “conversaciones sustanciales” sobre salud y es ahí,

donde la inteligencia artificial entra en juego, ya sea en una app o en una compleja conexión entre dispositivos. La inteligencia artificial ya está en el sector salud y en labores variadas, por ejemplo, en cirugías asistidas, así como seguimiento de pacientes además de complementarse con la popularidad de los *wearables*, dispositivos *vestibles/wereables*, que ayudan a monitorear la actividad de sus usuarios obteniendo información valiosa que puede aumentar la efectividad en tratamientos o incentivar hábitos saludables [20]. Utilizar la IA para cambiar comportamientos resulta algo sumamente complejo, aunque también resulta más viable utilizarla para ayudar a las personas a formar objetivos, a formar hábitos [21]. Hay muchos modelos de cambio de comportamiento, aunque vamos a centrarnos para ejemplificar en el modelo de cambio de *Sepah* que compone 2 bucles uno de formación de objetivos y otro de formación de hábitos. Utilizando este modelo se planifica el objetivo tomando en cuenta la intención inicial por ejemplo hacer más ejercicio, se actúa sobre el objetivo y se recompensa el hábito ya que las acciones que se toman según la intención inicial se convierten en hábitos solamente cuando son recompensadas repetidamente con el tiempo, incluso si esto sucede de forma ocasional, las recompensas pueden ser intrínsecas (por ejemplo, el impulso del estado de ánimo del entrenamiento) o extrínsecas (por ejemplo, que el entrenador personal te menciona que has hecho un buen trabajo) aquí entra en juego el concepto de gamificación, que es la aplicación de principios de comportamiento a la mecánica de juegos, donde la IA puede jugar un papel fundamental ya que todo consiste en recrear lo que sucede cuando se juega algún videojuego en el cual se te dan recompensas tras lograr ciertos objetivos del mismo y esto se puede trasladar a la creación de un hábito. Después de hacerlo y recompensarlo hay que recordar el hábito, esto también se conoce como señal o activador, el cual surge como parte fundamental para volver al ciclo de actividad y recompensa, un extra muy útil es el poder reflexionar sobre el objetivo aquí se muestra el progreso, que y como se han estado ejecutando los objetivos, cuando se tiene contacto con un entrenador personal, por ejemplo, se logra empatía y ayuda para la motivación que se necesita para continuar con el ciclo. Como podemos ver, en este modelo de cambio de comportamiento, la inteligencia artificial puede integrarse prácticamente en cada aspecto siendo una extensión del humano para lograr estos cambios y generar hábitos, así como en este modelo hay muchos más donde la tecnología y la IA se convierten en el complemento perfecto para poder lograr estos nuevos hábitos, recordar, recompensar, reflexionar e incentivar son cosas importantes esto sumado a que una IA puede también significar una diferencia económica incluso en los usuarios ya que sin duda alguna el alcance de los dispositivos es mayor a obtener un apoyo humano como un entrenador personal.

Sin duda alguna, cuando se trata de cambio de comportamiento enfocado en salud como hemos estado analizando, las herramientas tecnológicas y la IA resultan una herramienta clínicamente efectiva y viable basada en dar información, automatizar recordatorios, dar recompensas e incluso poder “platicar” para reflexionar en lo que se ha logrado; aunque si se desea algo más complejo como formación de objetivos donde se toma en cuenta las diferencias de cada persona resulta más viable seguir utilizando la inteligencia humana, por lo cual siempre se puede obtener lo mejor de ambas partes para lograr algo efectivo.

1.5 E-terapias con Chatbot

Un Chatbot es, una aplicación computacional multiplataforma con una interfaz diseñada para reconocer aquello que le escribimos y que es capaz de contestar de una forma coherente al contexto y al tema. Estos sistemas están diseñados para establecer un diálogo basado en reglas y técnicas de procesamiento de lenguaje natural, ya sea hablado o leído, que transforman la interacción entre el usuario y el sistema mediante interfaces cuasi-conversacionales con la finalidad de mejorar la experiencia usuario [22]. Las aplicaciones de este tipo de sistemas son actualmente muy variadas, ya que pueden implementarse para cualquier servicio que se pueda resolver con una lógica conversacional y no requiera de excesiva información visual, además los Chatbots conversacionales tienen la capacidad de retener información para ser utilizada en otro momento de la plática.

El uso de Chatbots, ha ido en aumento, aunque se menciona que el primer *healthbot* surgió como un script que hacía una llamada al programa ELIZA en 1960 y solo hacía una coincidencia de patrón para simular la

interacción entre un paciente y un psicoterapeuta. Tomando en cuenta la evolución de estos mismos tenemos como gran ventaja, el poder hacer uso de estos Chatbots en el ámbito de la salud mental debido a que su forma de interacción es a través de una conversación el cual puede escalar con facilidad para satisfacer las necesidades de muchas personas en distintos lugares a la vez; un ejemplo de esto es el Chatbot llamado *Karim* el cual funciona a través de mensajes de texto y está centrado en ayudar a víctimas de las guerras en medio oriente especialmente a los refugiados ya que un Chatbot no necesita boletos de avión, alimento, protección ni paga ofreciendo un recurso para aquellos que no pueden tener acceso a los recursos más tradicionales [23].

Las *e-terapias* mezclan y hacen uso de dos herramientas para atacar problemáticas de toda índole; desde cambios de hábitos hasta dejar el cigarro y muchos más. Esto consiste en tomar algún tipo de herramienta psicológica llámese, cuestionario o terapia y nuestro Chatbot como herramienta tecnológica conversacional brindando así nuevas posibilidades de los tratamientos de salud mental existentes con un menor costo. Un factor fundamental para estas implementaciones es el tener bien definido la estrategia a seguir como en una intervención y el público meta ya que con este último, se podrá definir la estrategia y el cómo implementarse, en la bibliografía del tema encontramos que se ha demostrado que utilizando el modelo de UTAUT (The Unified Theory of Acceptance and Use of Technology) en los públicos metas en países de habla inglesa principalmente, el cual puede determinar la aceptación o la intención de usar un coach virtual como en el caso de la organización STIVORO con un coach virtual para fomentar el no consumo de tabaco y ayudar a las personas a dejar el tabaco, ellos utilizan este modelo y llegan a la conclusión que dentro de la comunidad universitaria un 49% está dispuesta a usar el coach, el 37% no lo acepta y el resto se encuentra dudoso al respecto. Estas métricas son buenas para medir con mayor precisión la viabilidad del Chatbot en el rubro deseado [24].

También existe el caso de *Woebot* que inicia como un proyecto para poder brindar apoyo a estudiantes universitarios con problemas de ansiedad o depresión comprobando que crear terapias con Chatbots es factible pero es necesaria la intervención de las herramientas psicológicas validadas para poder realizar mediciones y conteos de que es lo que pide el público, modelos como el PHQ-9, SVM, LDA son algunos de los que alimentan la conversación de *Woebot* para detección de síntomas de depresión y poder llevar un registro del estado anímico de sus usuario y saber cómo actuar posteriormente [25].

Las e-terapias en general demuestran viabilidad en una serie de pruebas clínicas y otorgan la acumulación de datos en cualquier momento, visibilidad de patrones y una importante tendencia a generar más cosas en gran escala, las e-terapias y su integración con los Chatbots es algo que no desaparecerá ya que la tendencia de su uso en diferentes aplicaciones va en aumento. Existen importantes limitaciones, sin embargo, ya que los seres humanos responden mejor a otros seres humanos que a un programa informático. En un futuro próximo, la Chatbot terapia deberá demostrar ser un complemento valioso o incluso indispensable al terapeuta humano, pero no un reemplazo.

1.6 Descripción del problema a tratar

El desarrollo tecnológico que se ha vivido durante los últimos siglos ha traído un sinnúmero de beneficios a la sociedad, así como nuevos retos a superar que antes no eran tan comunes y se convierten en una situación de interés social entre las que destacan a nivel mundial el incremento de sedentarismo y la falta de actividad física en la población que juegan como factores de riesgo para enfermedades no transmisibles. El mercado de las TIC tiene también como propósito acompañar a sus usuarios en el proceso de adquisición de comportamientos más saludables, sin embargo, aún es complicado, para estos sistemas lograr una motivación efectiva y empática con sus usuarios como se realizaría con un entrenador personal.

El problema de sobrepeso y obesidad en México no es ninguna novedad tan solo del 2015 al 2018 México sigue ocupando el segundo lugar según la OCDE incluso se proyecta que para el 2030 la obesidad en el país aumente de 32.4% a 39% [26], la misma OCDE se ha encargado de tomar medidas con otros organismos para

intentar que esas cifras no sigan en aumento entre ellas: aumento de impuestos, mostrar el valor nutricional de manera legible en los productos e incluso han realizado campañas masivas para promover los alimentos saludables [27], por otra parte hay otros medios de información y lugares que carecen de esta atención por ejemplo, no existen campañas o algo parecido en entornos laborales así mismo las redes sociales y aplicaciones son medios masivos de información que no han sido explotados por las organizaciones que quieren promover la salud. También se sabe que menos de la mitad de la población hace ejercicio y sin importar el género a medida que se incrementa la edad, los porcentajes de población activa físicamente disminuyen [28], esto principalmente porque cuando se es joven por lo general acudimos a la escuela y ahí se imponen materias donde puedan realizarse físicamente esto sumado a las actividades extra escolares como fútbol, taekwondo, etc. Pero fuera del rango de edad en el cual acudes a la escuela si llegara a existir un plan de activación física la adherencia a estos programas es precaria. En contraste, México tiene un alto número de usuarios de redes sociales con una tendencia en aumento donde al menos un 63% de la población son internautas y el 50% tienen un rango de edad entre 18 a 54 años donde el 83% lo usa para acceder a redes sociales siendo la más utilizada Facebook [29] , [30] , [31]; es por ello que de esta forma surgió la idea de poder atacar la problemática del sobrepeso y adherencia a la activación física por el lado de redes sociales donde la penetración social de nuestro grupo de interés es muy sólido.

1.7 Propuesta de solución

De acuerdo a los resultados de ENSANUT 2018 [3] tenemos dos escenarios en México; El primero es que en un rango de edad entre 18 y más de 50 años la población en general, no hace suficiente actividad física y que a medida que incrementa la edad tiende a disminuir la actividad física incrementándose de este modo aún más el sedentarismo en la población. El segundo escenario hace referencia a la construcción de hábitos ya que también existe evidencia que muchas de las personas, que intentan realizar actividad física en la mayoría de las ocasiones no logran continuar con su plan de activación por falta de adherencia en los mismos. Por otra parte, existe también evidencia de la penetración de internet en el rango de edad de 18 a 60 años en particular en redes sociales, entre las cuales destaca Facebook; por lo cual surge la idea de utilizar esta red social como medio para motivar a sus usuarios a incrementar su actividad física. Nuestra propuesta es entonces, diseñar e implementar un “coach” virtual que actúe como agente motivador, este coach virtual se materializara a través de la creación de un Chatbot. Pensamos en un Chatbot para materializar el coach virtual, ya que su interfaz tipo chat nos permitirá simular una conversación entre los entrenados y su coach. En particular utilizaremos el Messenger de Facebook para la comunicación del coach con el público meta y el plan de activación física es el propuesto por la plataforma METS [32], el cual contiene un programa integral asesorado por expertos en salud y diseñado para mejorar la actividad física diaria de una forma sencilla ya que solo se requiere de caminar lo cual es accesible casi para todas las personas sin importar la condición física inicial.

De esta manera se crea *Metsbot* que se ensambla a la plataforma METS para poder servir como coach a los usuarios que siguen las metas progresivas propuestas por METS para incrementar su actividad física. *Metsbot* se tiene pensado que sirva de elemento motivador para incentivar la adherencia, por una parte y por otra parte para servir como puente con los nuevos posibles usuarios que se encuentren en Facebook, donde al alcance de un clic podrán informarse sobre METS. Esto último ya que *Metsbot* al iniciar conversación con los nuevos posibles usuarios les habla acerca de su salud, del sedentarismo y les invita a usar la plataforma para que puedan ser ahora usuarios de METS con su coach personalizado que podrán consultar 24/7. *Metsbot* es así el primer punto de contacto amigable con METS. Para propiciar la empatía con los usuarios, en el Chatbot se implementan interfaces que le sean agradables al usuario para interactuar, intentando así que no sea el texto el único recurso; ya que se toma en cuenta que la comunicación actual, en los chats ésta no está basada solo en texto. *Metsbot* hará uso de alguno de recursos como *emojis*, botones de guía para mantener al usuario dentro de la conversación y además se implementarán avatares que serán los coach que el usuario ha escogido dentro de la plataforma de METS para que el acercamiento con esta figura virtual sea mayor, cabe señalar que los coach están provistos también de emociones faciales. Otra forma de apoyar a la plataforma de METS es el

poder construir un corpus propio del api Metsbot que contendrá los datos generados entre el Chatbot y los usuarios los cuales se podrán utilizar posteriormente para otros fines. El último objetivo que se quiere lograr con Metsbot es el de ejemplificar como un Chatbot puede ser de utilidad para plataformas con fines similares a METS que también tratan de mejorar la salud siendo así un referente para ideas similares.

1.8 Alcanzables y Restricciones

El campo de la psicología de la salud y sus implicancias en el estilo de vida de una persona, establece los puntos de partida para comprender la estructura necesaria para el cambio conductual, y su mantenimiento a largo plazo. La pregunta fundamental puede ser presentada de la siguiente manera: ¿Cuáles son las condiciones necesarias para crear un cambio conductual en el individuo?

El objetivo del presente trabajo es tratar de responder a esta pregunta, así como también ofrecer puntos de vista susceptibles de evaluación empírica. Responder a esta pregunta implica hacer un análisis de factores motivacionales, conscientes y fuera de la conciencia, así como de ciertos aspectos de interpretación cognitiva que limitan la utilización de recursos y habilidades personales por parte del individuo que conduzcan a lograr este cambio. Con este propósito, voy a asumir las siguientes hipótesis de trabajo: (1) La conducta humana está dirigida con base a metas. (2) Los logros de un individuo no están en proporción a su potencial y a sus recursos. En este trabajo nos limitaremos a presentar un prototipo y seguiremos a un grupo reducido de usuarios Mets.

Capítulo 2. Fundamentos Teóricos

2.1 ¿Qué es un Chatbot?

Los Chatbots, son un producto de la inteligencia artificial cuyo objetivo principal es establecer una conversación coherente con los seres humanos. De hecho, los Chatbots son un intento de superar el test de Turing. Este test indica que una máquina es inteligente cuando al conversar con ella no podemos distinguirla de un ser humano [33], [34].

Un Chatbot es entonces, una aplicación computacional multiplataforma con una interfaz diseñada para reconocer, aquello que le escribimos, es decir, un chat el cual podemos encontrar en cualquier plataforma de conversación desde el antiguo Messenger de Microsoft hasta el actualmente muy popular WhatsApp, el Chatbot debe ser capaz de contestar de una forma coherente al contexto y al tema. Estos sistemas están diseñados para establecer un diálogo basado en reglas y técnicas de procesamiento de lenguaje natural, ya sea hablado, leído e incluso dependiendo de la situación hasta visual, por ejemplo, el famoso Clip de Microsoft Office, SIRI y Cortana. Esto transforma la interacción entre el usuario y el sistema mediante interfaces cuasi-conversacionales con la finalidad de mejorar la experiencia de usuario [35]. Las aplicaciones de este tipo de sistemas son actualmente muy variadas ya que pueden implementarse para cualquier servicio que se pueda resolver con una lógica conversacional y no requiera de excesiva información visual, además los Chatbots conversacionales tienen la capacidad de retener información para ser utilizada en otro momento de la plática. Otra de las ventajas del uso de esta tecnología, es que puede integrarse no solo en plataformas web y móviles, sino también dentro de plataformas que han sido creadas para ello como Slack, Alexa, Skype, Telegram, Discord e incluso en redes sociales como Twitter y Messenger (Facebook); razón por la cual se encuentran en sitios web comerciales, en los cuales tienen la función de promocionar productos, ayudar en la navegación al usuario y obtener información del cliente a través de la plática [36]. Debido a la gran cantidad de aplicaciones, los Chatbots conversacionales se han convertido en herramientas muy populares en el ámbito educativo-administrativo [37].

Los Chatbots cumplen las mismas funciones que la web, pero de manera más cómoda, ágil, sencilla y rápida pensadas muchas veces para ser utilizadas desde el Smartphone por la misma naturalidad de la experiencia al chatear, por lo cual podemos también definirlos como una aplicación web que contiene su frontend y su backend además de poderlo complementar con el uso de APIs externas.

Un Chatbot está pensado para satisfacer ciertas necesidades del usuario el cual puede ser un trabajador, alguien que necesita asesoría, una persona que usa cierto sistema o incluso un paciente. Tomando como ejemplo estos últimos cabe mencionar, que la popularidad de los Chatbots ha alcanzado en el área de la salud es creciente e importante; los usuarios de los Chatbots en el ámbito de la salud, no solo se les considera como usuarios, sino que generalmente son tratados como pacientes, es en el área de la medicina que surge el término de Healthbot [38], [39] como una ramificación de los Chatbots, pero dedicados a la salud como veremos más adelante. Por lo tanto, podemos decir que Metsbot al ser una extensión de METS y al estar ambos, enfocados al área de la salud y considerando a sus usuarios como pacientes lo podemos definir como un Healthbot.

2.2 ¿Qué es una aplicación web?

Una aplicación web es una aplicación informática distribuida cuya interfaz de usuario es accesible desde un cliente web, normalmente un navegador web [40]. Tiene como características habituales:

- Comunicación mediante HTTP sobre TCP/IP.
- Procesamiento en servidor.
- Acceso a bases de datos.
- Arquitectura por capas.
- Distintos tipos de usuarios.

En pocas palabras una aplicación web es un programa de computadora el cual a través del protocolo de comunicación HTTP hace uso de un servidor y una interfaz de usuario para realizar alguna tarea, se menciona que es distribuida, ya que al ser web está pensada para poder ser usado por más de un usuario en cualquier momento.

2.3 ¿Cómo se compone una aplicación web?

La arquitectura de una aplicación define cómo se organizan los distintos módulos que la componen. Una aplicación web tiene tres niveles: interfaz de usuario, lógica de negocio y datos donde el nivel de interfaz tiene las páginas HTML que el usuario solicita a un servidor web y que ve en un cliente web, el nivel de lógica de negocio está compuesto por los módulos que implementan la lógica de la aplicación y que se ejecutan en un servidor por último el nivel de datos está compuesto por los datos, mejor conocido como el sistema de gestión de bases de datos de aquí nacen dos conceptos para el desarrollo de aplicaciones web que es el frontend y el backend.

El Frontend es aquello que contiene la interfaz de usuario y la usabilidad de la aplicación (estructura, colores, fondos, tamaños, botones, animaciones, etc.), es donde se recogerá las entradas e instrucciones del usuario, se prepara en cierto formato y se envía al servidor donde será procesado por el Backend.

El Backend es el que hace funcionar todos los engranajes de la aplicación, es la programación del lado del servidor y se compone principalmente de un servidor, una aplicación y una base de datos, es donde se encuentran las funciones que tendrá la aplicación web además de ser la encargada de optimizar los recursos y/o la seguridad de la misma. En el backend se procesa aquellas interacciones o peticiones que solicitó el frontend con los datos para acomodarlos nuevamente en cierto formato y mandarlo como respuesta que interpretará el frontend, cabe mencionar que el Frontend también se debe modelar para poder procesar y mostrar al usuario una respuesta dependiendo de lo que el servidor haya mandado ya sea información o algún error e incluso si es que el servidor no mando nada.[41], [42], [43].

2.4 ¿Qué son las APIs?

El término API significa Interfaz para la Programación de Aplicaciones (Application Programming Interfaz), y se refiere a una librería que implementa cierta funcionalidad mediante una interfaz con la cual el programador puede acceder a dicha funcionalidad [44], en Web existen las API web que como bien dice la definición funcionan como librerías que hacen ciertas funciones que van desde procesar datos, seguridad hasta

acceder a datos de alguna otra aplicación para complementar las peticiones del usuario por ejemplo alguna geolocalización, las API se implementan dentro de las aplicaciones web a través de “llamados” es decir de peticiones para acceder a la información que procesa el servidor que aloja esa API y de esta forma se complementa la funcionalidad de la aplicación web que se está desarrollando construcción.

En resumen, una interfaz una capa que conecta dos sistemas siendo así una API es una interfaz que conecta aplicaciones para que compartan información, por ejemplo, el API de mapas de Google entre otras [45].

2.5 ¿Qué es una API REST?

Ya conocemos que es una API, en la creación de este tipo de aplicaciones existe un modelo, una arquitectura llamado REST que por sus siglas significa Representational State Transfer siendo así una arquitectura de API que se identifica por tener un conjunto ordenado de restricciones en sus componentes, conexiones y datos creando una interfaz que se conectan vía web con el protocolo HTTP para su comunicación. Esta arquitectura permite acceder a la información con identificadores únicos mejor conocidos como Uniform Resource Identifiers o por sus siglas URI. Estas URI son utilizadas para obtener datos o generar operaciones con los datos en todos los formatos posibles como XML y JSON, es una alternativa a otros protocolos estándar de intercambio de datos como SOAP donde brilla debido a que sus peticiones HTTP ya contienen toda la información necesaria para ejercitarse permitiendo que ni el cliente ni el servidor necesiten recordar estados previos para satisfacer la petición, de ahí el nombre de State Transfer [46], [47].

2.6 ¿Qué es el procesamiento de lenguaje natural?

El Procesamiento de lenguaje natural o por sus siglas PLN (NLP en inglés) consiste en la utilización de un lenguaje natural para comunicarnos con la computadora, donde esta debe entender las oraciones que le sean proporcionadas, el uso de estos lenguajes naturales, facilita el desarrollo de programas que realicen tareas relacionadas con el lenguaje o bien, desarrollar modelos que ayuden a comprender los mecanismos humanos relacionados con el lenguaje.

La arquitectura del sistema de procesamiento del lenguaje natural de la figura 3, muestra como la computadora interpreta y analizar las oraciones que se le proporcionan [48]. Se tiene entonces que la entrada en un sistema PLN es una expresión en LN que el usuario quiere comunicar al sistema. En el siguiente paso, de acuerdo con Cortez, Vega y Pariona [48], el sistema utiliza un analizador lexicográfico y sintáctico. El primero, denominado scanner, se encarga de identificar los componentes léxicos definidos a priori. El segundo, denominado parser, se encarga de verificar si se cumple un orden gramatical entre los elementos identificados por el scanner. A continuación, se realiza un análisis semántico para saber cuál es el significado de cada oración y asignar dicho significado a una expresión lógica (true o false). Por último, se hace un análisis pragmático de la información, es decir, el sistema toma todas las oraciones ya analizadas y, dependiendo de la situación, obtiene una expresión resultante, que es la que se envía al usuario

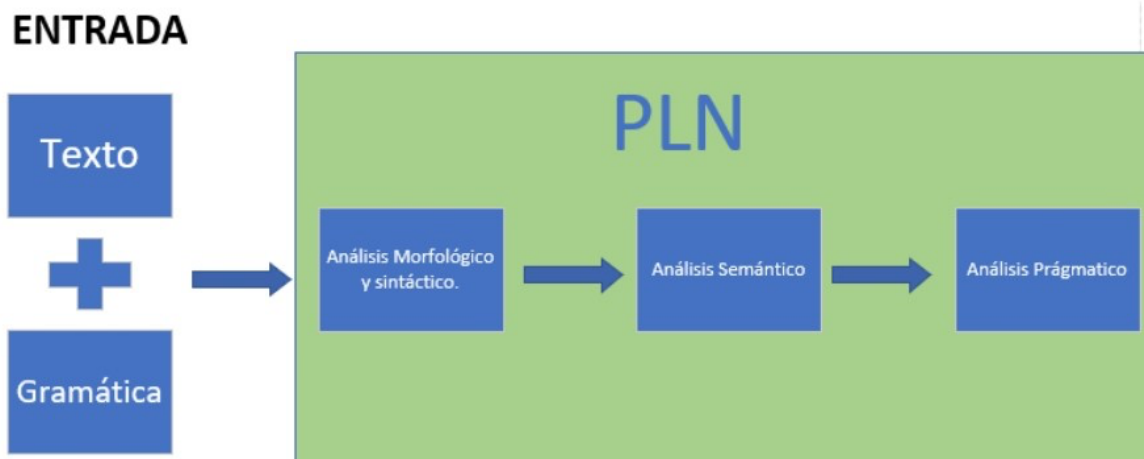


Figura 3. Arquitectura de un sistema de procesamiento de lenguaje natural. Fuente: Cortez, Vega y Pariona [48]

2.7 APIs de procesamiento de lenguaje natural

De tal manera podemos comprender que el PLN se llega a utilizar para la clasificación de textos en un conjunto de temáticas por ejemplo el texto: “No voy a comprar este acondicionador porque es más barato en otras tiendas” pertenece a una temática de Precio-calidad por lo cual el PLN lo clasificaría en una categoría correspondiente a Precio-calidad todo de una forma automatizada. Por lo tanto podemos definirlo como aplicaciones que se dedican a realizar esas funciones, darnos temáticas, un contexto o “a que se refiere” el texto, para lograrlo hay muchas formas de hacerlo y de mejorarlo donde existe la clasificación supervisada, semi-supervisada y no supervisada que hace referencia a cuando este proceso limpia el texto y se vuelve a “alimentar” al programa para que pueda realizarlo de mejor manera en una segunda ocasión también está la opción de automatizar todo esto con técnicas de inteligencia artificial más complejas y modernas como lo sería el Machine Learning que es un área de la inteligencia artificial donde a través de ciertos procesos el mismo sistema o aplicación va “aprendiendo” de lo que realizan sus funciones retro alimentándose para realizar su tarea cada vez mejor o de una forma más óptima y aquí es donde entra la comercialización los programas que son capaces de realizar estas cosas para diversos fines y al ser una aplicación que hace una cierta funcionalidad esto se puede traspasar a una librería o interfaz en la cual los programadores podremos hacer uso de esas funciones para agregar funcionalidad de un PLN en nuestras aplicaciones lo cual podemos traducir como la utilización de una aplicación de PLN como API web, entonces estas se pueden integrar a nuestras aplicaciones web para que podamos trabajar con el contexto de las entradas que dan los usuarios. Existen APIs de PLN de paga como la que ofrece IBM Bluemix o Microsoft Azure y otras gratuitas como Dialog Flow de Google, popularizando así su uso y que a su vez sirve para una mejora cualitativa para los mismos sistemas de PLN que las componen al permitir que su base de datos se puedan actualizar y mejorar, la popularidad de estas API aumento debido a que a la par la popularidad de los Chatbots iba en aumento también; tal que actualmente es difícil conceptualizar un Chatbot sin un proceso de PLN. Por otra parte también se han creado plataformas de desarrollo de Chatbots que de una forma u otra integran herramientas de PLN de una forma más básica, entre ellos destacan Manychat o Chatfuel, una herramienta de creación de Chatbots se puede definir como una aplicación o software que a través de su plataforma te permite crear entidades como si de un sistema experto se tratara con funciones extras como el PLN, conexión con APIs y funciones de métricas para propósitos de monitoreo [49].

2.8. Herramientas de desarrollador de Facebook

Facebook como red social también mantiene un crecimiento e innovación constante gracias al desarrollo de terceros que utilizan los datos obtenidos a través de la misma, por ello desde 2013 Facebook creó una

plataforma con el nombre de “Facebook Developers” donde tienes acceso a los productos para desarrollo que la red social ofrece como: Inteligencia artificial, Realidad aumentada, juegos, código abierto, Integraciones sociales, presencia en redes sociales, realidad virtual, herramientas para publicar y herramientas para empresas.

Básicamente es una serie de productos basada en APIS donde creas una aplicación que funcionara en Facebook o con Facebook para obtener información que estos mismos te proveen creando así una comunidad de desarrollo para Facebook [50].

2.9 Messenger de Facebook como herramienta de desarrollador de Facebook

Entre los productos de desarrollo que ofrece Facebook existe la API de desarrollo para poder utilizar las funciones de Messenger de Facebook el cual es la extensión de mensajería de la red social, esta plataforma que ofrece el desarrollo sobre Messenger de Facebook está compuesta por una serie de herramientas para crear Chatbots, en sí lo que ofrece es una API, plugins web y una vista web para poder tener una interfaz de chat en la cual conversar además de realizar acciones varias como pagos, integración a APIs propias, NLP, estadísticas, etc. [51].

2.10 ¿Cómo se hace un Chatbot?

En su composición más básica un Chatbot solo requiere de una programación lógica en la cual se pueda de algún modo interpretar lo que el usuario escribió y con base en eso dar una respuesta, todo a través de una interfaz de CHAT la cual consiste en una ventana donde se vean los mensajes enviados y recibidos así mismo un apartado donde el usuario pueda escribir y la acción de “Enviar Mensaje” la cual generalmente está compuesta de un botón de enviar con el cual se envía el texto del usuario y se recibe en el demonio del programa el cual toma el texto lo procesa según esté programado y con base en el resultado de ese procesamiento es como responde en forma de mensaje al usuario, dicho mensaje aparecerá en la ventana de mensajes. Actualmente la creación de Chatbots es un poco más sofisticada y por lo general no se inicia desde cero gracias a la ayuda de diversos frameworks y APIs que existen como se mencionó anteriormente. Los Chatbots como su definición menciona son programas de computacionales y están destinados actualmente a funcionar como programas web o aplicaciones web (webapps) por lo cual su composición actual es de un frontend que contiene todo lo visual e interactivo, que sería la ventana de mensajería, y un backend, el cual se encarga de las conexiones web para que la información vaya del servidor al usuario, aquí en el backend es donde encontramos también las llamadas a las APIs, estas tienen tareas muy diversas pero en el mundo de los Chatbots serán las encargadas de tener conexión con la base de datos, procesar las respuestas del usuario, enviar respuestas al usuario como imagen, video, mapas, etc. pero además existen las APIs que están dedicadas al procesamiento de lenguaje natural (PLN) las cuales se convierten en la herramienta que ayuda a entender que es lo que dice el usuario facilitando la tarea del programador para implementar un Chatbot, ya que solo se hace su integración en el backend el cual llevará las respuestas del usuario a la API de PLN esta nos dirá que es lo que el usuario está diciendo o requiriendo para que el resto del backend realice los procedimientos necesarios y así finalmente se le responda al usuario de una forma natural para continuar con la conversación.

La estructura de un Chatbot actual la podemos esquematizar como se muestra en la figura 4.

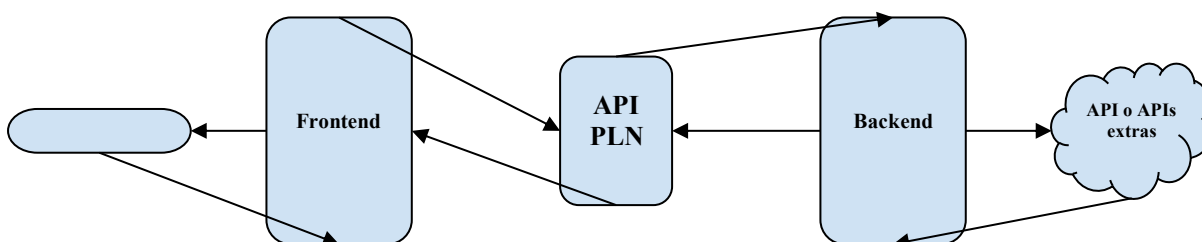


Figura 4. Arquitectura de un Chatbot

Actualmente la parte del frontend incluso puede ser proporcionada por la plataforma en donde se quiere implementar el Chatbot por ejemplo Messenger de Facebook, Skype, etc. como se mencionó con anterioridad. Es decir, no es necesario preocuparse por el frontend para la interfaz donde el usuario va a conversar con el Chatbot en estos casos, ya que solo a través de la plataforma deseada se realizan las configuraciones correspondientes para tener un punto de acceso, una conexión por la cual la API (backend) podrá conectarse para recibir los inputs es decir, las entradas de texto de lo que el usuario está solicitando.

De lo contrario se tendría que realizar una aplicación frontend para crear esta interfaz y colocarla en algún servidor para poder acceder a esa interfaz desde internet.

2.11 ¿Para qué sirven los Chatbots?

Resumiendo lo que anteriormente se mencionó, los Chatbots se pueden definir como aplicaciones web que funcionan como asistentes virtuales creados para mantener algún tipo de conversación, así como ofrecer respuestas preconcebidas ayudándonos a realizar una tarea en concreto; con el único diferenciador de que la interfaz en una aplicación web es muy variante y en un Chatbot siempre es una ventana de chat o una interfaz orientada a la conversación. Gracias a este tipo de interfaz, que utilizan los Chatbots, es que se puede hacer uno para prácticamente de cualquier cosa desde una alarma, lista del supermercado, juegos, el clima, ponerle filtros a una imagen, agendar un vuelo, compras en línea, procesos de acompañamiento para pacientes, concientización de problemas de salud, así como acercamiento y acompañamiento para trastornos psicológicos, ya que genera empatía en el usuario por medio de la conversación multimedia. Casos como Poncho, Joy, Pizza Hut, Cinépolis, Cleverbots y GYANT solo por mencionar a algunos, son Chatbots que existen en la plataforma de Messenger de Facebook y ayudan dando el clima, detectando si sufres de ansiedad o depresión, pedir una pizza, ver la cartelera del cine y comprar tu boleto o realizar una consulta médica por si te sientes mal ayudándote a saber que tienes y que puedes hacer. En resumen, la utilidad de un Chatbot está limitada por la imaginación y la tecnología disponible de quien desea crearlo [52], [53], [54].

2.12 Diseño de un Chatbot

Para la creación de un Chatbot como cualquier aplicación web o cualquier aplicación a desarrollar debe de crearse un diseño o un modelado de cómo será el asistente web que van desde las partes técnicas hasta cuestiones que están involucradas con el lenguaje.



Figura 5. Metodología de modelado utilizada en Metsbot

Actualmente no hay un marco de trabajo o una metodología general para el diseño de los Chatbots, ya que la creación de estos varía entre desarrolladores, sin embargo, por parte de la comunidad se recomiendan ciertos procesos que simulan una estandarización del diseño y modelado de Chatbots, una de estas variantes y en especial la que se utilizó en Metsbot es la que se muestra en la figura 5 y se describe a continuación:

1. Modelo Canvas para Chatbot, para aterrizar la idea del asistente y definir cómo se obtendrá la información, que plataforma o tecnología se utilizará, definición de la personalidad del asistente.
2. Modelos conversacionales, brinda un esquema general de como funcionara el asistente, indica cuáles serán los intentos que pondrá el usuario y cómo responder a estos, se crea un guion el cual el asistente conocerá para la conversación de acuerdo a su objetivo.
3. Diseño y Pruebas, el modelo conversacional se implementa en la plataforma anteriormente seleccionada y se realizan pruebas para corroborar que el guion se está utilizando de manera correcta y se crea una arquitectura general donde se define cuáles son los componentes que utilizará el asistente, por ejemplo: multimedia, llamadas a API, Base de datos, etc.
4. Desarrollo del Chatbot, sobre la primera implementación se integran las funcionalidades del asistente a través de una arquitectura general y se realizan pruebas unitarias.
5. Análisis y Mejora Continua, el Chatbot se encuentra en producción y se analiza la información obtenida, así como la interacción de los usuarios para realizar mejoras en cualquiera de las partes que componen al asistente [55], [56].

2.13 Integración de la psicología en un Chatbot

Como se mencionó en el punto anterior, la creación de Chatbots lleva un proceso de planeación y estructuración que varía dependiendo del creador, sin embargo, todos convergen en el mismo punto, la selección y definición del lenguaje a emplear, es decir, el cómo se va a expresar, qué actitud debe tener, cómo será el flujo de las acciones y/o diálogo a seguir durante la conversación. En este punto es donde se integra la parte lingüística y a su vez se puede integrar la parte psicológica en un Chatbot. En otras palabras, es el proceso de conocer cómo se realizará la conversación para obtener un resultado similar al que generaría un psicólogo durante una consulta esto suele estar hecho a base de ciertos “scripts” o respuestas comunes, así

como el lenguaje que un profesional de la psicología utilizaría. Por ejemplo, Hello Joy día a día monitorea tu estado de ánimo a través de una breve conversación diaria y te alienta cuando te sientes mal pero también se alegra contigo cuando todo parece estar bien[57]. El Chatbot después de ciertos niveles es capaz de identificar signos sintomatológicos cuando el usuario está propenso a presentar cuadros depresivos o de ansiedad; ya que de manera interna construye un perfil del usuario para posteriormente hacer *recomendaciones basadas en contenido*. Si Joy acierta en la identificación de los signos, es decir, estos son confirmados por el usuario, Joy le presenta un test psicológico validado al usuario y dependiendo del resultado le proporciona la opción de realizar ciertas rutinas para mejorar esa condición, o bien solicita la intervención de un psicólogo.

En general la técnica de la psicología, implica interacción y está ligada fuertemente con el flujo de acciones/conversación a través de los cuales el usuario (paciente) adquiere las herramientas para continuar el proceso de forma autónoma, es así que, para implementar este tipo de rutinas, el Chatbot deberá poder conversar con el usuario para lograr el propósito de su funcionalidad. Cabe mencionar que por lo general nunca se busca como objetivo el desplazamiento de la profesión realizada por psicólogos reales, al contrario, se busca poder canalizar a posibles pacientes proporcionando un seguimiento o una atención todo el tiempo lo cual resultaría complicado para un humano real.

2.14 SLOGAN y memoria a largo plazo

Como proceso de la adquisición de un hábito se encuentra la tarea de poderle recordar a la persona el “qué está haciendo” y “por qué” lo está haciendo comúnmente se utilizan palabras o frases repetitivas, de preferencia pegajosas, que pretenden quedarse en la mente de las personas teniendo una función similar a la de un slogan por ejemplo el “No puede comer solo una” de Sabritas o el “Me encanta” de McDonald's.

En experimentos controlados se les presentó a una audiencia una serie de conversaciones para preguntarles posteriormente qué palabras o frases recuerdan después del experimento, así descubriendo que las palabras mostradas en diferentes contextos eran las que más recuerdan ellos ponen como ejemplo que si tratas de recordar la palabra “chocolate” al verla y tenerla presente sobre etiquetas o frases como “leche con chocolate”, “barra de chocolate” y “pastel de chocolate” se vuelve más sencillo de recordar a solo presentar la palabra “chocolate” repetidas veces; De ahí la importancia de crear una frase corta y simple que no sea muy cliché ayudará a que las personas recuerden el producto.

En la terapia funciona de manera similar el crear un “slogan” o una frase para que sea más sencillo al usuario recordar el qué o para que está realizando su proceso de cambio de comportamiento [58].

Capítulo 3. Desarrollo de la solución

Retomando y sintetizando lo visto en los puntos anteriores, sabemos que el sedentarismo es un problema grave que provoca enfermedades no transmisibles y con ello el deterioro de la salud, también hay una relación directamente proporcional entre la edad de las personas y el aumento de sedentarismo en su vida este rango se ve marcado desde los 18 años de edad en adelante, además conocemos que esta parte de la población mayor a los 18 años de edad principalmente, tienen una penetración muy grande en internet donde la mayoría de su tiempo lo dedican al uso de redes sociales. Con estos datos se formula la idea de crear un Chatbot que resida en la red social líder con mayor número de usuarios: *Facebook*. El Chatbot deberá funcionar como un complemento para la plataforma METS el cual ya contiene la planificación y estrategia necesaria para generar o aumentar la actividad física en sus usuarios y de esa forma combatir el sedentarismo. El Chatbot tiene como objetivo ser un primer contacto, amigable, para conseguir nuevos usuarios de METS y a los usuarios registrados en METS servirles como un “coach virtual” que se alimentara del diálogo de sus pacientes/usuarios y de la base de datos de la plataforma METS. Esto resulta posible gracias a la forma en la que se puede interactuar con un Chatbot que incluso nos permite integrar aspectos de terapia psicológica que ya han sido aplicadas con anterioridad como apoyo al sector de la salud, en este caso se ha optado por alimentar la parte conversacional de la Chatbots con el modelo de adaptación de las precauciones tanto en la fase de posibles nuevos usuarios como de usuarios registrados y en este último el uso de la terapia cognitivo conductual. Con esta combinación se pretende animar a más usuarios y a los actuales a combatir el sedentarismo además de ser un soporte para evitar recaídas que provoquen que dejen el plan de activación que proporciona METS en su totalidad.

3.1 Requerimientos de la solución

Parte de la solución a brindar es que este coach virtual pueda ser parte de METS y no un sustituto de la aplicación, que sirvan de complemento, pero de manera independiente, es por ello que el Chatbot utilizará recursos de la misma base de datos como sería la información de los usuarios. Datos personales y datos generados por la aplicación, así mismo usará la información para mostrar los mismos avatares que la aplicación pero con otra funcionalidad y diseños, también será diferente la manera en la que se consulta la información por último el Chatbot tendrá su propia base de datos que no solo ayudará a que se reduzca la cantidad de consultas a la base de datos de la plataforma sino que guardará información que solo le es útil a esta parte, al Chatbot además que de esta manera se puede establecer y respetar la independencia de su funcionalidad al no requerir modificar la base de datos principal de la plataforma.

3.2 Un healthbot llamado Metsbot y su psicología

Como mencionamos anteriormente un Chatbot que tiene funcionalidades enfocadas en el área de la salud recibe el nombre de Healthbot y en efecto, eso es Metsbot, un Chatbot que funciona como coach virtual con el objetivo de incrementar la adherencia de los usuarios de METS a la actividad física. Un Chatbot no puede obligarte a realizar una acción, sin embargo, como se ha estado mencionando anteriormente hay ciertos aspectos psicológicos que ayudan o facilitan el cambio de comportamiento lo cual concuerda con la finalidad de Metsbot. La creación de los diálogos y su flujo fueron supervisados por una psicóloga² con quien se han creado estos mismos bajo los principios del PAP.

Como hemos visto en el capítulo 1, el PAP tiene varias etapas en el cambio de comportamiento y el Chatbot implementado tiene injerencia tanto en las etapas uno a seis que involucra animar al usuario y alertarle de lo que sucede si no realiza actividad física si continúa con su vida sedentaria; es decir, le brinda información al

² Ana Rosa Carrillo Urrutia, Lic. En Psicología con cedula: 11663966.

usuario con el propósito de que el usuario tome la decisión activarse físicamente³. Por otra parte, una vez aceptado el uso de la app y del coach la interacción con Metsbot cambia para dar incentivos y recordatorios; además siguiendo el modelo de las terapias cognitivas-conductuales, genera empatía con el usuario, soportando así la etapa siete del PAP en la que se espera que el usuario pueda tener recaídas en sus comportamientos no saludables, pero al contar con un entrenador personal, este lo animó a seguir adelante hasta lograr cambios duraderos en su comportamiento.

Los diálogos y como se da el flujo conversacional de Metsbot desde el inicio de la conversación siguen el PAP y se incluyen funciones de la terapia cognitiva conductual en el momento que el usuario ya ha iniciado el proceso de cambio de comportamiento y requiere soporte para incrementar la adherencia al mismo. Se recurre también al uso de frases repetitivas como “es para mejorar tu salud” haciendo uso de la teoría repetitiva para que el usuario tenga claro qué es lo que hace este Healthbot a forma de slogan y de esta manera reforzar la estrategia del PAP, el slogan se usa como una indicación directa de cuál sería uno de los beneficios (pros) de usar la plataforma de METS y a su vez el coach que es Metsbot. Es importante indicar que en Metsbot los usuarios interactúan de forma personalizada con un avatar que ellos eligieron como entrenador personal, de ahí que en adelante hablemos del entrenador personal o coach para referirnos a Metsbot. La parte de la planificación se simplifica para el usuario, ya que en lugar de que él tenga que planear que hacer para activarse físicamente, al registrarse y usar la plataforma de METS, esta se encargará de llevar el control del plan de activación física personalizado de acuerdo a las características del usuario y con ayuda de su coach el usuario tiene el soporte, que este requiere; cuando coach detecte que el usuario está teniendo una posible recaída o intente desistir del cambio de hábito, trata de generar empatía con el usuario. Una de las técnicas que se implementan para empatizar con el usuario es preguntarle el motivo de las barreras que le eviten continuar, por el momento el coach solo tratará de alentar a la persona sin dar una respuesta relacionada con el motivo, pero la información que el usuario proporciona es almacenada para poder analizarla y saber cómo atacar estos inconvenientes, además de poder estudiar el comportamiento de nuestros usuarios.

3.3 Canvas para Metsbot

No existe una metodología totalmente validada que determine cómo se deben realizar los Chatbots, sin embargo los investigadores del área que se dedica a la creación y comercializarlos de Chatbots utilizan herramientas de otras metodologías para gestión de proyectos adaptadas a este concepto, entre estas herramientas destaca el uso del modelo *Canvas* de proyectos al cual se le ha realizado una serie de ajustes para poder conceptualizar como debe ser el Chatbot a desarrollar[59], aplicando este modelo Canvas para la creación de Metsbot el modelo se visualiza en la tabla 1.

³ Cabe mencionar que al estar en combinación con el sistema METS el chatbot lo que promueve inicialmente es que los usuarios se enrolen en METS es decir inicien la estrategia para incrementar su actividad física y reducir el sedentarismo y con esto, de acuerdo al PAP, iniciar su cambio conductual.

Tabla 1. Modelado para la creación de Metsbot usando Canvas

Analytics A través de la pagina web What analytics will be collected? How? Why?	Marketing Publicidad en Facebook y a través de la misma promoción de la herramienta principal How will the user get to know?	Value proposition 1) Aplicar la metodología de adaptación de las precauciones al público para que sean conscientes de que pueden padecer/tener problemas de salud 2) Acercarlos a la herramientas que les van a ayudar. 3) Generar adherencia a estos métodos para mejorar su salud What pain point is the chatbot solving? What value is it providing the user?	Users -Personas de cualquier genero -20+ -En ciudad de Puebla -Parte de la comunidad universitaria Who are the users? Age? Location? Interests?	Devices Accesible desde cualquier gadget con conexión a Facebook What devices do your users have access to?
Development platforms En su propia pagina de Facebook y posiblemente en la misma pagina web/app How will the chatbot be implemented?	Third party services - Dialogflow - Api de la herramienta What 3rd party services are you planning to use?		Modalities Interaccion via: -Texto -Recursos Graficos (botones, etc.) How would your users like to interact? Voice/text?	Deployment Platforms Messenger de Facebook Where will your users interact with the chatbot?
Background Tasks - Chequeo Diario del perfil del usuario para poder notificarle su progreso. - Estar alerta a posibles notificaciones que se deseen enviar What tasks should the chatbot carry out in the background?		Personality Una personalidad de COACH pero de una forma no estricta sino amigable y elantador Will your chatbot have a personality?	Conversational Tasks El chatbot hablara acerca de la Diabetes, prevencion de la misma y bajo el modelo de adaptación de las precauciones. What will your chatbot talk about?	

Project name: _____

Chatbot Design Canvas
Version 1 © Srin Janarthanam

3.4 Investigación de campo

Para la construcción de los diálogos, se realizó una investigación de campo para poder conocer cuáles son los motivos por los cuales el internauta que está en redes sociales mexicano promedio, no realiza actividad física suficiente como para combatir su sedentarismo, así como, identificar qué es lo que este público meta desconoce de la problemática. Lo anterior para poder contar herramientas para poder integrar el PAP y la terapia cognitivo conductual según las necesidades del público al que será dirigido Metsbot.

Originalmente el estudio se pensó para habitantes, principalmente de México entre 18 y 40 años de edad, que posiblemente sean sedentarios, aunque no estén conscientes de ello ni de su vulnerabilidad a enfermedades no transmisibles, que tuvieran disposición de un dispositivo electrónico; sin embargo al proporcionar esta aplicación por un medio digital como Google Forms y difundirlo también por medios digitales no se restringió el acceso a ningún participante solo posteriormente se filtraron los resultados a lo deseado.

Al final se contó con la participación de 223 personas, 118 mujeres y 105 hombres en el lapso de un mes comprendido entre el 6 de Julio del 2018 al 6 de agosto del mismo año.

El formulario está basado en el test BBAQ (Barriers to Being Active Quiz) [60], [61] es una medida de autoinforme compuesta por 21 preguntas que evalúan 7 barreras percibidas para hacer actividad física: falta de tiempo, influencia social, falta de energía, falta de voluntad, miedo a lastimarse, falta de habilidades y falta de recursos. Un puntaje superior a 5 en cada barrera, permite identificar que esta es percibida por la persona

como importante. El cuestionario se validó en población universitaria colombiana y se encontró una alta confiabilidad (coeficiente alfa de Cronbach= 0,832) y validez de constructo de este formulario se puede consultar en [62], [63].

Se solicitaron cómo datos socio-demográficos como edad, género, escolaridad, ocupación. Se agregaron también dos preguntas para conocer si los participantes saben la diferencia entre actividad física y ejercicio, así como si consideraban que realizaban suficiente actividad física diariamente. A todos los participantes se les solicitó un consentimiento informado para hacer uso de sus datos en forma anónima.

3.4.1 Resultados del análisis del cuestionario aplicado.

Se contó con la participación de 223 personas, 118 mujeres y 105 hombres. La edad de la muestra oscila entre 15 y 65 años. Se aplicó el instrumento a personal académico, administrativo, de apoyo y estudiantes, quienes conforman la comunidad universitaria en la tabla 2 se detalla las características de la población que respondió la encuesta.

Los resultados de la encuesta se clasificaron según las categorías: falta de tiempo, influencia social, falta de energía, falta de voluntad, miedo a lastimarse, falta de habilidades que contiene la encuesta BBAQ.

Tabla 2. Clasificación por edad y sexo

Rangos de edad	15-24	25-34	35-44	45-54	55 – 65
Sexo					
Mujeres	50 (55,6%)	26 (44,1%)	20 (60,6%)	18 (54,5%)	4 (50,0%)
Hombres	40 (44,4%)	33 (55,9%)	13 (39,4%)	15 (45,5%)	4 (50,0%)
Total	90 (40,3%)	59 (26,4%)	33 (14,7%)	33 (14,7%)	8 (3,9%)

En cuanto a su escolaridad, el 57,9% de la muestra tiene estudios profesionales, el 24,5% estudió un posgrado, el 17,2% estudió bachillerato o preparatoria, y en cuanto a la ocupación; 124 personas, es decir, el 55,4% tiene un trabajo remunerado, mientras que el 35,7% de la población son estudiantes y el 8,9% se encuentra actualmente sin empleo, dicha información se muestra detallada en la Tabla 3.

Tabla 3. Detalles de la muestra relacionadas con su ocupación

Población	Total	Mujeres	Hombres
Estudiantes	79 (35,4%)	48 (60,8%)	31 (39,2%)
Trabajadores	124 (55,6%)	61 (49,2%)	63 (50,8%)
Sin empleo	20 (9,0%)	9 (45,0 %)	11 (55,0%)
Total	223	118 (52,9%)	105 (47,1%)

Los resultados de la encuesta se muestran en la tabla 4 y están clasificados de acuerdo a las categorías del instrumento (BBAQ): falta de energía, falta de voluntad, falta de recursos, falta de tiempo, influencia social, miedo a lastimarse y falta de habilidades. Por otra parte, Los resultados de las primeras dos preguntas donde se hace referencia al conocimiento que tienen respecto a la diferencia entre actividad física y ejercicio, así como la pregunta que corresponde a si consideran que realiza suficiente actividad física se presentan en la tabla 5.

Tabla 4. Respuestas a las preguntas por categorías

Categorías	Preguntas	Población	Muy poco probable	Algo improbable	Algo probable	Muy probable
Falta de tiempo	Pregunta 1: Mi día es tan ocupado ahora que no creo que pueda apartar tiempo para realizar actividad física en mi horario normal	Estudiantes	21 (26,9%)	15 (19,2 %)	33 (42,3%)	10 (11,5%)
		Trabajadores	30 (25,0 %)	23 (18,5 %)	40 (32,3 %)	31(24,2 %)
		Sin empleo	9 (45,0 %)	7 (35,0 %)	3 (15,0 %)	1 (5,0%)
	Pregunta 8:Hacer actividad física me quita mucho tiempo de las otras obligaciones que tengo como mi trabajo, familia, horario, etc.	Estudiantes	30 (38,0 %)	12 (15,2 %)	28 (35,4 %)	9 (11,4 %)
		Trabajadores	28 (22,8%)	42 (33,3%)	34 (27,6 %)	20 (16,3%)
		Sin empleo	7 (35,0%)	7 (35,0%)	6 (30,0 %)	0
	Pregunta 15: Tengo muy poco tiempo libre durante el día para hacer ejercicio.	Estudiantes	20 (25,3 %)	22 (27,8 %)	19 (24,1 %)	18 (22,8 %)
		Trabajadores	25 (20,2 %)	23 (18,5 %)	50 (40,3 %)	26 (21,0 %)
		Sin empleo	8 (40,0%)	8 (40,0%)	2 (10,0%)	2 (10,0%)
Influencia social	Pregunta 2: A ninguno de mis familiares o amigos les gusta realizar actividad física, así que no tengo oportunidad de hacer ejercicio.	Estudiantes	38 (48,7%)	21 (25,6 %)	15 (19,2 %)	5 (6,4%)
		Trabajadores	68 (55,3%)	21 (16,3 %)	26 (21,1 %)	9 (7,3 %)
		Sin empleo	7 (35,0 %)	2 (10,0 %)	8 (40,0 %)	3 (15,0 %)
	Pregunta 9: Me da pena cómo me voy a ver cuándo haga ejercicio en frente de otras personas	Estudiantes	34 (43,0 %)	8 (10,1 %)	15 (19,0 %)	22 (27,8 %)
		Trabajadores	72 (58,5%)	28 (22,0%)	12 (9,8 %)	12 (9,8%)
		Sin empleo	6 (30,0%)	5 (25,0%)	7 (35,0%)	2 (10,0%)
	Pregunta 16: Mis actividades sociales habituales con mis familiares y amigos no incluyen actividades físicas.	Estudiantes	21 (26,6 %)	13 (16,5 %)	21 (26,6 %)	24 (30,4 %)
		Trabajadores	25 (20,2 %)	22 (17,7 %)	33 (26,6 %)	44 (35,5 %)
		Sin empleo	2 (10,0%)	3 (15,0%)	7 (35,0%)	8 (40,0%)
Falta de energía	Pregunta 3: Estoy muy cansado después del trabajo como para hacer ejercicio.	Estudiantes	21 (26,9%)	9 (11,5%)	22 (28,2 %)	27 (33,3 %)
		Trabajadores	22 (17,9 %)	24 (18,7%)	43 (35,0 %)	35 (28,5%)
		Sin empleo	4 (20,0 %)	6 (30,0 %)	4 (20,0 %)	6 (30,0 %)
	Pregunta 10: Yo ni siquiera duermo lo suficiente. No me podría levantar más temprano o acostarme más tarde para hacer ejercicio	Estudiantes	24 (30,4 %)	14 (17,7 %)	21 (26,6 %)	20 (25,3 %)
		Trabajadores	38 (30,9%)	37 (29,3%)	27 (22,0 %)	22 (17,9%)
		Sin empleo	5 (25,0%)	4 (20,0%)	6 (30,0%)	5 (25,0%)
	Pregunta 17: Estoy muy cansada (o) durante la semana y necesito descansar durante el fin de semana para recuperarme	Estudiantes	18 (22,8 %)	22 (27,8 %)	24 (30,4 %)	15 (19,0 %)
		Trabajadores	31 (25,0 %)	25 (20,2 %)	42 (33,9 %)	26 (21,0 %)
		Sin empleo	7 (35,0 %)	3 (15,0 %)	5 (25,0 %)	5 (25,0 %)
Falta de voluntad	Pregunta 4: He estado pensando en empezar a hacer ejercicio, pero no he sido capaz de dar el primer paso.	Estudiantes	20 (25,6 %)	12 (14,1 %)	19 (24,4 %)	28 (35,9 %)
		Trabajadores	35 (28,5 %)	15 (11,4 %)	30 (24,4 %)	44 (35,8 %)
		Sin empleo	1 (5,0 %)	5 (25,0 %)	7 (35,0 %)	7 (35,0 %)
	Pregunta 11: Es más fácil para mí encontrar excusas para no hacer ejercicio que ponerme a hacerlo.	Estudiantes	20 (25,3 %)	12 (15,2 %)	25 (31,6 %)	22 (27,8 %)
		Trabajadores	31 (25,0%)	20 (16,1%)	41 (33,1 %)	32 (25,8 %)
		Sin empleo	1 (5,0%)	4 (20,0%)	6 (30,0%)	9 (45,0%)
	Pregunta 18: Quiero hacer más ejercicio, pero parece que no puedo obligarme a hacerlo.	Estudiantes	12 (15,2 %)	15 (19,0 %)	24 (30,4 %)	28 (35,4 %)
		Trabajadores	25 (20,2 %)	22 (17,7 %)	42 (33,9 %)	35 (28,2 %)
		Sin empleo	2 (10,0 %)	1 (5,0 %)	7 (35,0 %)	10 (50,0 %)

Categorías	Preguntas	Población	Muy poco probable	Algo improbable	Algo probable	Muy probable
Miedo a lastimarse	Pregunta 5: Hacer ejercicio puede ser riesgoso a mi edad.	Estudiantes	66 (84,6%)	12 (14,1%)	1 (1,3%)	0
		Trabajadores	95 (76,6%)	18 (14,5%)	10 (8,1%)	1 (0,8%)
		Sin empleo	13 (65,0%)	6 (30,0%)	1 (5,0%)	0
	Pregunta 12: Conozco muchas personas que se han lastimado porque han hecho mucho ejercicio.	Estudiantes	44 (55,7 %)	15 (19,0 %)	18 (22,8 %)	2 (2,5 %)
		Trabajadores	74 (59,7%)	31 (25,0%)	16 (12,9 %)	3 (2,4%)
		Sin empleo	7 (35,0%)	5 (25,0%)	6 (30,0%)	2 (10,0%)
	Pregunta 19: Me da miedo lesionarme o que me dé un ataque cardíaco,	Estudiantes	50 (63,3 %)	17 (21,5 %)	9 (11,4 %)	3 (3,8 %)
		Trabajadores	78 (62,9 %)	33 (26,6 %)	11 (8,9 %)	2 (1,6 %)
		Sin empleo	13 (65,0 %)	4 (20,0 %)	2 (10,0 %)	1 (5,0 %)
Falta de habilidades	Pregunta 6: No hago ejercicio lo suficiente porque nunca he aprendido ningún deporte.	Estudiantes	42 (53,8%)	16 (18,2%)	11 (14,1%)	10 (12,8%)
		Trabajadores	85 (68,5%)	26 (21,0%)	6 (4,0%)	8 (6,5%)
		Sin empleo	7 (35,0%)	6 (30,0%)	4 (20,0%)	3 (15,0%)
	Pregunta 13: Realmente no me veo aprendiendo un nuevo deporte a mi edad,	Estudiantes	51 (64,6 %)	16 (20,3 %)	6 (7,6 %)	6 (7,6 %)
		Trabajadores	85 (68,5 %)	25 (20,2%)	8 (6,5 %)	6 (4,8 %)
		Sin empleo	6 (30,0%)	9 (45,0%)	4 (20,0%)	1 (5,0%)
	Pregunta 20: No soy lo suficientemente bueno en ninguna actividad física como para entretenerme.	Estudiantes	29 (36,7 %)	16 (20,3 %)	21 (26,6 %)	13 (16,5 %)
		Trabajadores	70 (56,5 %)	28 (22,6 %)	17 (13,7 %)	9 (7,3 %)
		Sin empleo	7 (35,0 %)	5 (25,0 %)	3 (15,0 %)	5 (25,0 %)
Falta de recursos	Pregunta 7: No tengo acceso a caminos para trotar, piscinas, ciclo pistas para bicicleta, etc.	Estudiantes	30 (38,0%)	16 (20,3 %)	22 (27,8 %)	11 (13,9 %)
		Trabajadores	75(61,0%)	19 (14,6 %)	17 (13,8 %)	13 (10,6 %)
		Sin empleo	8 (40,0 %)	6 (30,0 %)	2 (10,0%)	4 (20,0 %)
	Pregunta 14: Es simplemente muy costoso. Uno tiene que tomar una clase, inscribirse en un club o comprar el equipo adecuado.	Estudiantes	32 (40,5 %)	6 (7,6 %)	24 (30,4 %)	17 (21,5 %)
		Trabajadores	60 (48,4%)	28 (22,6 %)	26 (21,0 %)	10 (8,1 %)
		Sin empleo	4 (20,0%)	3 (15,0%)	7 (35,0%)	6 (30,0%)
	Pregunta 21: Si hubiese un lugar para hacer ejercicio y regaderas en el trabajo, tendría más probabilidad de hacer ejercicio.	Estudiantes	19 (24,1 %)	10 (12,7 %)	23 (29,1%)	27 (34,2 %)
		Trabajadores	29 (23,4 %)	16 (12,9 %)	29 (23,4 %)	50 (40,3 %)
		Sin empleo	3 (15,0 %)	2 (10,0 %)	7 (35,0 %)	8 (40,0 %)

De acuerdo a los resultados mostrados en la tabla 4, encontramos que destacan las categorías: falta de energía principalmente en estudiantes, falta de voluntad, que es más notable en trabajadores y falta de recursos en personas sin empleo y estudiantes.

Entre los ítems que más respuestas afirmativas recibieron, está el dieciocho que dice “Quiero hacer más ejercicio, pero parece que no puedo obligarme a hacerlo”, donde el 65,8% de los estudiantes, el 62,1% de los trabajadores y el 85% de las personas sin empleo respondieron algo probable o muy probable; el cuarto ítem dice “He estado pensando en empezar a hacer ejercicio, pero no he sido capaz de dar el primer paso”, a lo que el 60,3% de los estudiantes, el 60,2% de los trabajadores y el 70% de las personas sin empleo respondió muy probable o algo probable. Otros ítems que obtuvieron mayor porcentaje en los indicadores muy probable y algo probable son el uno, once, dieciséis y veintiuno con más del 50%.

Tabla 5. Conocimiento y realización de AF

Preguntas	Recuento de si	Recuento de no
¿Conocías la diferencia entre ejercicio y AF?	108 (48,4%)	115 (51,6%)
¿Consideras que realizas suficiente AF?	103 (46,2%)	120 (53,8%)

AF: Actividad física. Resultados del total de la muestra

Según los datos obtenidos en este estudio, la población objetivo no presenta las mismas barreras que las que identificó el Módulo de Práctica Deportiva y Ejercicio Físico-2018 [64], además, se encontró que menos de la mitad de la muestra conocía la diferencia entre actividad física y ejercicio (véase la tabla 5), el cual puede ser un factor importante, ya que al confundirse, las personas pueden pensar que para combatir el sedentarismo solo existe una solución y es hacer ejercicio, lo que implica invertir tiempo y dinero, esto se podría combatir solo con informar a la población, de esta forma podrían empezar implementando acciones sencillas durante el día, tales como subir escaleras, caminar, como explicaba Pairoj [65]. Por otra parte, se encontró que a pesar de brindarle a la muestra la definición de actividad física, más del 50% dijo no realizarla, por lo tanto, se entiende que el problema tiene raíces más allá del desconocimiento. Se presume que hay factores como el contexto y el sistema de creencias de una persona que influyen directamente en su estilo de vida.

Como se observa en la figura 6, tanto estudiantes, como trabajadores y personas sin empleo coinciden en que no realizan actividad física por falta de voluntad. La mayoría de las conductas de salud son hábitos que se han aprendido en la infancia, y se consolidan durante la adolescencia y la juventud, y los escasos datos disponibles permiten sospechar un proceso similar para las creencias en salud [66], es decir, se desglosa una subcategoría de la falta de voluntad, que es la transmisión generacional de creencias que se tienen en torno a la salud, por lo que podría afirmarse que por varias generaciones no existió una educación para fomentar el autocuidado mediante la actividad física, lo que dio como resultado, una población sedentaria que no se percibe vulnerable. Por años, se han promovido otro tipo de pensamientos, específicamente mitos sobre salud y sobre algunas enfermedades como la diabetes, tales como, “la diabetes solo les da a personas mayores”, “la diabetes puede dar por un susto”, por mencionar algunos, y dichos pensamientos están tan arraigados que es complejo combatirlos únicamente brindando información.

Por otro lado, los estudiantes manifestaron que su inactividad física se debe a la falta de energía; hecho que podría relacionarse con lo que se conoce como astenia[67]. Castillero [68] explica que es un cuadro en el cual se produce una reducción en el nivel de energía y fuerza en el organismo, disminuyendo asimismo la motivación y produciéndose sensación de agotamiento tanto a nivel físico como mental.

Por otra parte, Young, Finn, Bruetman, Pellegrini y Kremer [69] afirman que la astenia es una verdadera epidemia de la época actual. Ellos clasifican sus causas en dos tipos: las funcionales y las orgánicas. Las funcionales, a su vez, se dividen en psiquiátricas, psicosociales (sobre exigencia psicológica y social, problemas familiares como duelo, enfermedades, separación, mudanza, jubilación), alteraciones del sueño, medicaciones, y otras [67]. Las orgánicas se clasifican en causas endócrinas-metabólicas, enfermedades crónicas-inflamatorias y otras. Dentro de estas se ubican la obesidad y el sedentarismo.

Castillero [68] afirma que, en más de la mitad de los casos, la astenia se debe a causas puramente psicológicas y explica que se trata a nivel general a partir de la resolución de la causa concreta que la ha provocado; sin embargo, resulta de gran utilidad la realización de ejercicio físico, ya que ayuda a disminuir el estrés, a relajarse y además ayuda a generar endorfinas de forma endógena [68].

Por otra parte, las personas sin empleo, al igual que los estudiantes, alcanzaron un nivel significativo en la categoría de falta de recursos, lo que podría indicar que no identifican la diferencia entre actividad física y ejercicio, ya que realizar ejercicio o practicar un deporte implica adquirir equipo, o un espacio determinado, y esto conlleva un gasto; sin embargo, la actividad física se realiza durante todo el día y no es preciso salir de la rutina o gastar. O bien, tienen el conocimiento, pero por falta de recursos se refieren a espacios públicos. Si bien, la Ciudad de Puebla cuenta con parques, áreas verdes, jardines y ciclo pistas, actualmente el 81.4% de sus habitantes de 18 y más años percibe inseguridad en su entidad [64]. El INEGI reporta también, algunas de las conductas delictivas que la población ha atestiguado a los alrededores de su vivienda, en primer lugar, el consumo de alcohol en la calle, así como robos o asaltos frecuentes, consumo y venta de drogas, pandillerismo o bandas violentas, e incluso homicidios y en menor escala, pero no menos importante, secuestros, riñas entre vecinos y prostitución por mencionar algunos. Por lo que el 33.4% de la población dejó de salir a caminar por temor a ser víctima de un delito [64].

Por último, todos los hallazgos resultaron de suma importancia para poder implementar nuevas vías para llegar a las personas que llevan una vida sedentaria.

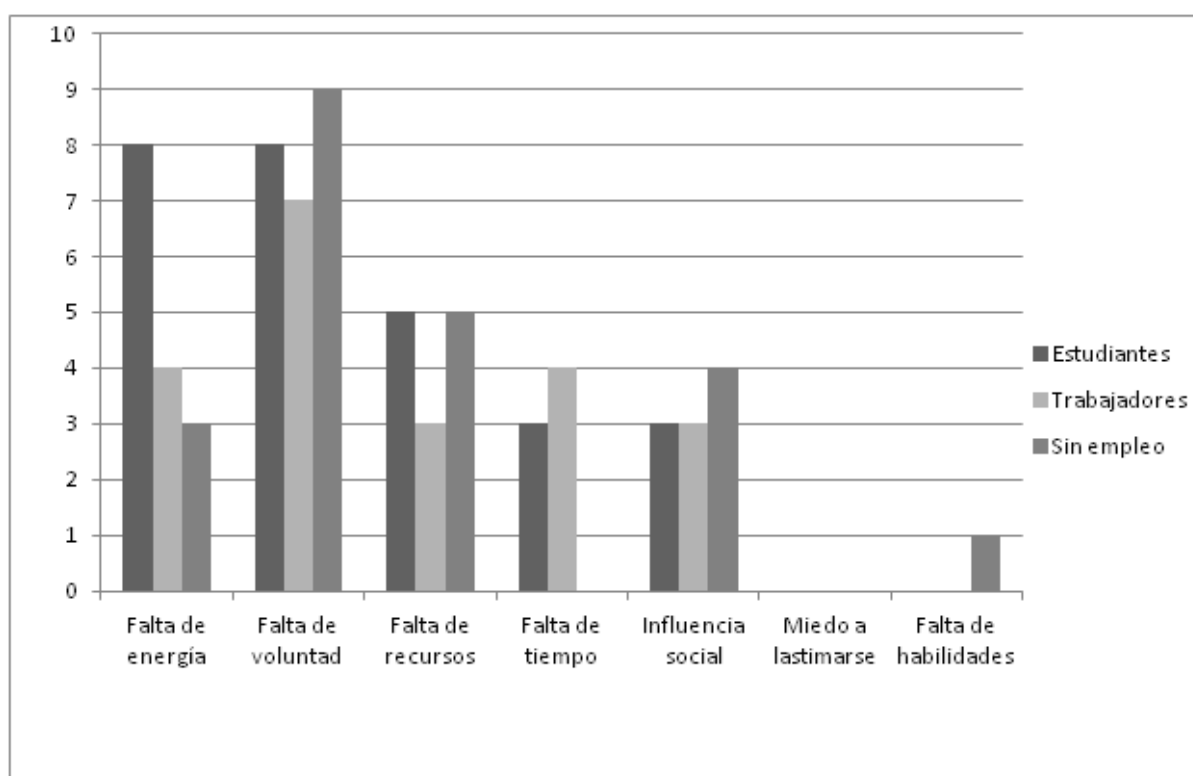


Figura 6. Resultados por categoría de la encuesta realizada que perfila los puntos a fortalecer utilizando el Chatbot

3.5 Consideraciones de los resultados de la encuesta aplicables al diseño de Metsbot

Desglosando y analizando los datos de nuestra muestra, el 55.4% son personas con trabajo remunerado, el 35.7% son estudiantes y el 8.9% se encuentra actualmente sin empleo, esto nos da un esbozo de la situación que presentan nuestro público objetivo donde se quiere aplicar el Chatbot de manera inicial y a su vez nos orienta sobre tipo del lenguaje a emplear.

El 51.6% desconoce la diferencia entre ejercicio y actividad física, confundiendo el significado de ambos términos y el 53% indicó que no realiza suficiente actividad física, lo cual nos da a entender que los usuarios del Chatbot necesitan una guía para que puedan conocer la diferencia entre términos y que podemos abordar desde una etapa no tan temprana de la terapia cognitivo-conductual y aceptación de las precauciones, ya que se les informara pero a su vez ya tienen conciencia

de que no realiza suficiente actividad física por ende solo faltaría concientizar de su posible vulnerabilidad para posteriormente desglosar en problemas más pequeños el porqué no inician un cambio de hábito, esto último se ha obtenido gracias a los resultados del cuestionario BBAQ.

Como se mencionó anteriormente el cuestionario BBAQ cuenta con una forma de ponderación específica según los resultados individuales por lo cual tras haber definido el porcentaje de respuesta de nuestra población por ítem y área estas respuesta del colectivo se han unificado para ponderar a todos como un solo individuo, tras esta acción vimos que destacan las categorías de falta de energía (principalmente en estudiantes), falta de voluntad (principalmente en trabajadores) y falta de recursos (notable en desempleados y estudiantes) por lo cual tenemos una población que dice no tener energía, no tiene voluntad suficiente y no tiene los recursos para poder un cambio de hábito pero con mayor afinidad en los ítems 18 y 4 que corresponde a “Quiero hacer más ejercicio, pero parece que no puedo obligarme a hacerlo” y “He estado pensando en empezar a hacer ejercicio” sucesivamente, pero no he sido capaz de dar el primer paso”. De esta forma podemos validar la viabilidad del healthbot, ya que al funcionar como coach virtual podemos atacar la falta de voluntad e incentivar a que no haya esa falta de energía, por otra parte, la plataforma de METS según su composición y plan de activación física resulta ser el healthbot un complemento muy viable, ya que no obliga a los usuarios a realizar gastos, tener una membresía de gimnasio o algo por el estilo, lo único que se necesita, es caminar.

Por otra parte tras ver las barreras principales de nuestra muestra se podrá realizar una implementación dentro del flujo de conversación en el healthbot para que cuando este tipo de personas recaigan en no estar interesadas por ser parte de METS se le preguntará sus motivos con base en estas barreras y de acuerdo a la respuesta el healthbot responderá de tal manera que esa barrera se vea como un inconveniente menor para concientizar a la persona en cuestión que esa barrera es algo superable y que puede lograrlo cómo se realizaría en una terapia cognitiva-conductual además de poder continuar con un registro de motivos por los cuales no inician su cambio de hábito.

3.6 Diagramas para el flujo de la conversación

Gracias al análisis de datos obtenidos con el estudio de campo se pudo fortificar y consolidar la estructura del flujo conversacional en el Chatbot, cabe mencionar que el realizar este diagrama del flujo conversacional resulta ser una buena práctica para visualizar como el Chatbot cumple con la funcionalidad requerida. Tal así que las herramientas existentes como BOT FRAMEWORK de Microsoft, Dialog Flow, Chatfuel, IBM Watson, Many Chat entre otros proporcionan una interfaz gráfica, en la cual se alimenta su propio modelo de un flujo conversacional para crear la conversación a la cual responderá el Chatbot, dando como resultado un flujo de conversación único, esto es importante, ya que por ejemplo a un Chatbot de agencia de vuelos no le deberían preguntar cuál es el siguiente partido de tenis, esa entrada del usuario no corresponde a la funcionalidad de ese Chatbot en particular y sin embargo se debe procesar esa entrada del usuario.

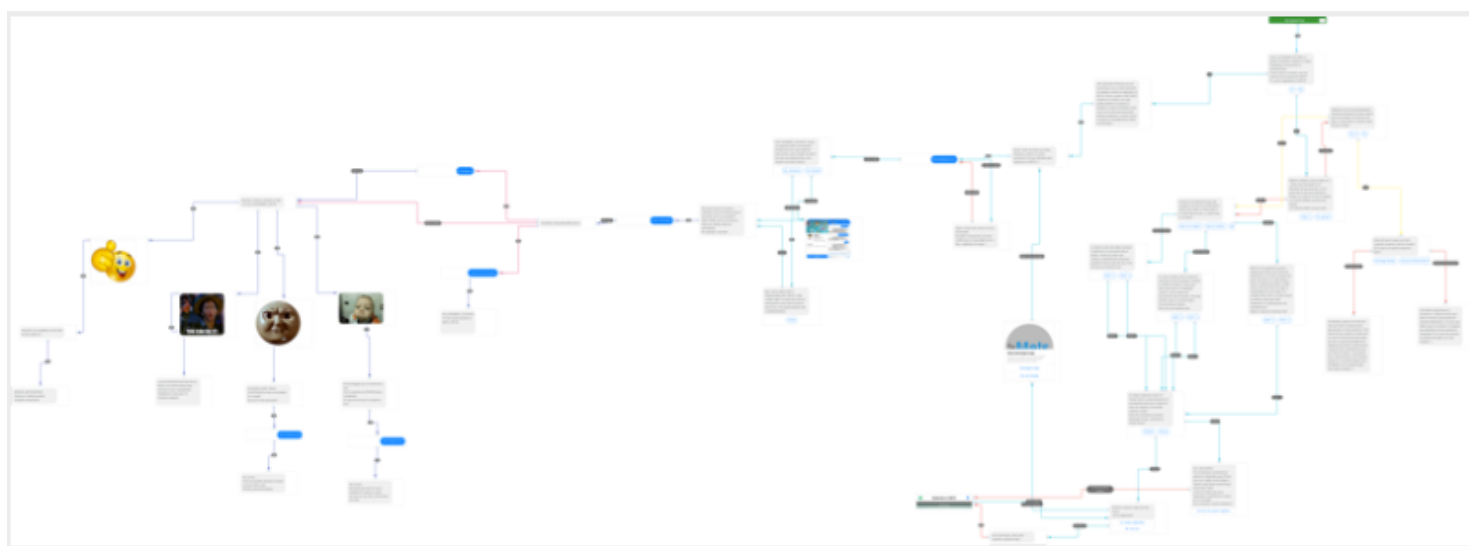


Figura 7. Flujo de la conversación de Metsbot

El flujo de la conversación de forma teórica suele representarse en forma de un diagrama con la finalidad facilitar la explicación del funcionamiento que tendrá el Chatbot, así como un modelo de casos de uso en UML, y aquí en este diagrama del flujo de la conversación es donde se ve integrada la funcionalidad, personalidad y lenguaje del mismo, en el caso de Metsbot este modelo se pensó con anterioridad con base al modelo canvas, en el mismo proceso del modelado de una conversación resulta pertinente también poder integrar la técnica de psicología que aplica el Chatbot en la conversación, ya que esto debe estar incluido en el diálogo. Por último, se ha de mencionar que todo modelo de conversación en un Chatbot contiene una o más ramificaciones con temas fuera del contexto funcional del mismo y estos se infieren, por lo cual rara vez es necesario, colocarlas en el diagrama, ya que las respuestas a dichos temas, fuera de tópico, suelen ser las mismas. Retomando el ejemplo anterior, si un usuario le pregunta al Chatbot de agencia de viajes sobre el próximo partido de tenis este responderá, gracias a la creación de estas ramificaciones, con algo similar a: “Lo siento, no entiendo a que se refiere, ¿desea buscar algún vuelo u hotel?”. El diagrama de flujo conversacional para Metsbot se presenta en la figura 7 y se desglosa en el Anexo 1 para mejorar su comprensión.

Capítulo 4. Implementación de Metsbot

En este capítulo expondremos como se realizó el Chatbot, así como los aspectos del diseño de la interfaz y la integración de Metsbot con la e-terapia METS.

4.1 La anatomía de Metsbot

Esta sección de la anatomía detallamos cuáles son las tecnologías necesarias para poder crear Metsbot y dejarlo en modo de producción lo cual vamos a dividir en hardware y software.

Hardware:

- Servidor: Se utilizará un equipo de cómputo con (características) bajo el sistema de Windows Server 2008 el cual está ubicado en el laboratorio de Tecnologías para la Salud

Software:

- Sistema operativo (Windows, Linux) el cual funcionara como entorno virtual para el desarrollo e implementación del Chatbot.
- GitHub para el control de versionado.
- Nodejs, funcionará como el entorno de desarrollo que funciona con lenguaje JavaScript el cual nos permite crear un servidor virtual sobre el servidor real y poder realizar las conexiones pertinentes.
- Express, librería de Nodejs es el middleware que crea funcionalidad segura y conectividad entre el servidor en node y el frontend.
- NPM, es el registro de software más grande del mundo donde developers Open Source de todo el mundo comparten y utilizan paquetes de librerías de software para todo tipo de uso.
- Request, librería de Nodejs que funciona para los llamados a la plataforma de METS y realizar las peticiones.
- Mongoose, librería de Nodejs que crea un entorno para la comunicación y creación de la base de datos funcional entre Nodejs y Mongodb.
- Timer, librería de Nodejs que permite implementar un ciclo que tiene como funcionalidad crear los recordatorios a los usuarios.
- Mongodb, es el gestor de base de datos el cual es necesario para la base de datos que manipula solamente Metsbot.
- API de integración con Messenger de Facebook, este api es la que nos permite conectarnos con el frontend proporcionado por Facebook para poder hacer uso de su chat a través de una página de la misma red social.

- API de Chatfuel, este api es donde se traspasa el diagrama de flujo conversacional para que posteriormente se conecte con el api de Facebook y el api principal de Metsbot con Nodejs, prácticamente aquí es donde se realiza la conversión y llamado al api.
- Página en Facebook, se crea una página en Facebook a la cual se le configura el API de integración de Messenger de Facebook, ya que al hacer uso de su plataforma el Chatbot debería estar alojado en alguna parte de la red social.

4.2 Diseño de la interfaz de usuario de Metsbot

En la plataforma de METS durante el proceso de registro, se permite a los usuarios seleccionar un avatar el cuál funcionará como la imagen de su coach, e interactuará con el usuario en las diferentes interfaces de la plataforma acompañando al usuario en todo momento. Utilizando este concepto, de que las personas escogen quién sería su coach como lo harían en la vida real, por ejemplo, en un gimnasio donde la persona tiene la libertad de escoger a su entrenador en Metsbot, mediante estos personajes que el usuario seleccionó en la plataforma se continuará su uso, siendo los avatares el canal de comunicación entre el usuario y el sistema. De este modo el avatar seleccionado en la plataforma METS, será quien siempre interactúe a través del Chatbot (Metsbot) con el usuario, por ejemplo, si el usuario *A* al registrarse seleccionó a “Javi” como su coach, al usar Metsbot en la plataforma de chat de Facebook será Javi quien interactúe con él. De esta forma Metsbot puede generar cierta empatía y vínculo entre los usuarios con el Chatbot con la ayuda de la plataforma, ya que tratan de transmitir cierta personalidad y sentimientos para simular a una persona, un coach, lo cual se logra apoyándonos de la ficción de los personajes, que ya conocieron, pero de una forma animada y pasan de ser imágenes estáticas a personajes animados, con “vida”. Esta funcionalidad consideramos permite establece la empatía entre el Metsbot y el usuario, ya que como anteriormente se menciona los Chatbots tienen personalidad y para Metsbot conforme al flujo de su conversación, la comunicación cambia de la fase número uno donde solo se invita al usuario a registrarse en la plataforma (se presenta el canal de comunicación) a la fase número dos donde ya se está realizando un seguimiento con el coach personalizado, simulando lo que realizaría un coach en la vida real, es decir que este te felicite, se moleste o entristezca cuando no cumples los objetivos; es por ello que para agregarle ese toque de realismo, a la personalidad del Chatbot, la interacción de los personajes animados utilizados como coaches tienen cuatro emociones básicas (alegría, neutra, enojo y tristeza) mediante las cuales se pueden matizar su interacción con el usuario.

4.2.1 Los personajes

Se tienen cuatro personajes dos hombre y dos mujeres en ambos casos tanto en las mujeres como en los hombres se cuenta con un personaje joven y otro mayor (que se distinguen por su vestimenta y en el caso de los mayores por las canas en su pelo), en la figura 7 presentamos a los cuatro personajes cuyos nombres son Javi, Victoria, Constanza y Pedro, nombres elegidos en el caso de las mujeres por su significación y en el de los hombres por votación.

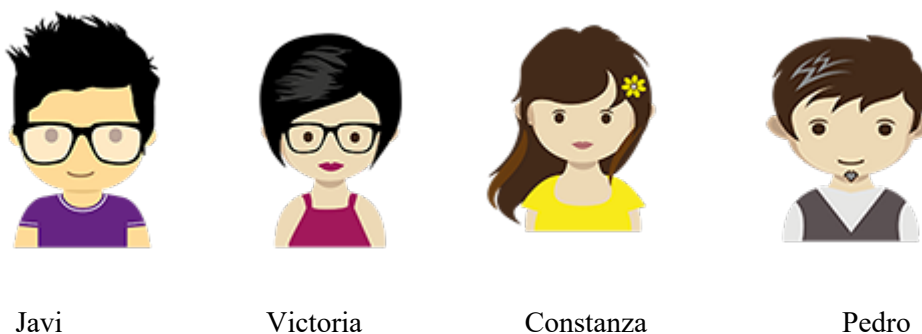


Figura 7. Entrenadores que trabajan en METS e interactúan con los usuarios en Metsbot

Los estados animados que se le da dado a cada personaje son:

- Enojado: Sirve para mostrar molestia al usuario.
- Felicitación (muy bien): sirve para felicitar al usuario.
- Neutro en este caso la neutralidad se identifica por el diálogo con el usuario. Se utiliza para mostrarse, presentarse, saludar.
- Triste: sirve para mostrar decepción al usuario.

En las figuras 8a y 8b se muestran los fotogramas de los diferentes personajes con la imagen de estos de acuerdo a su estado de ánimo

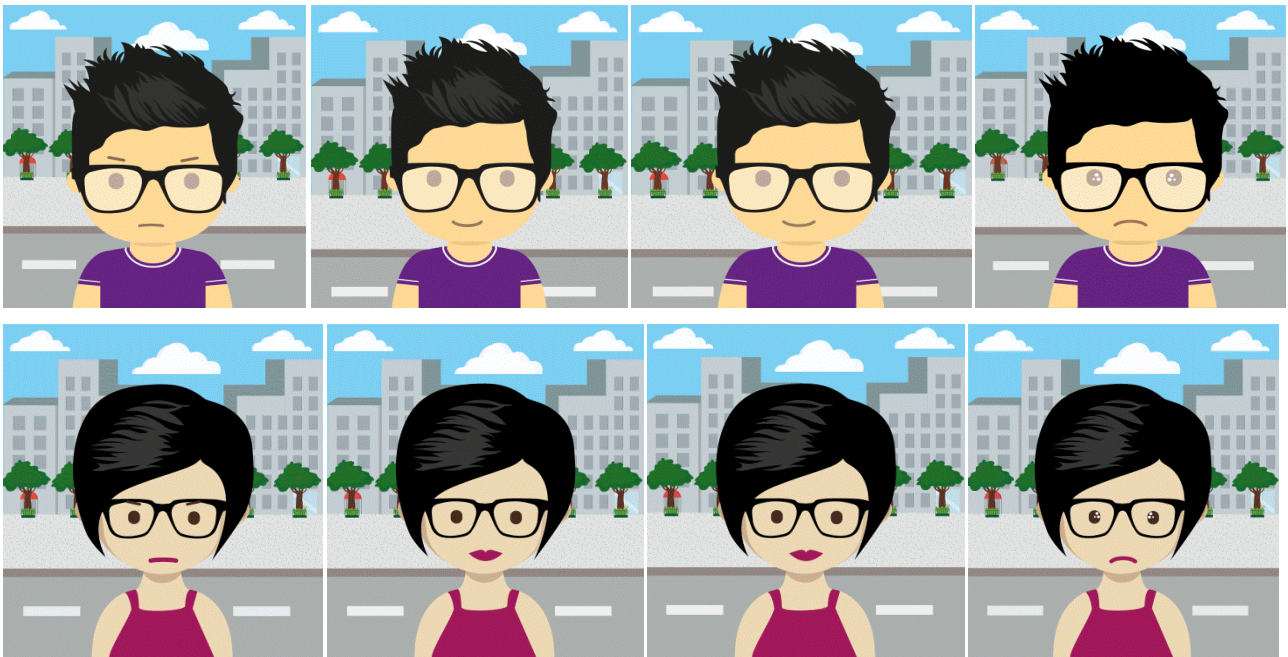


Figura 8a. Estados anímicos de Javi y Victoria

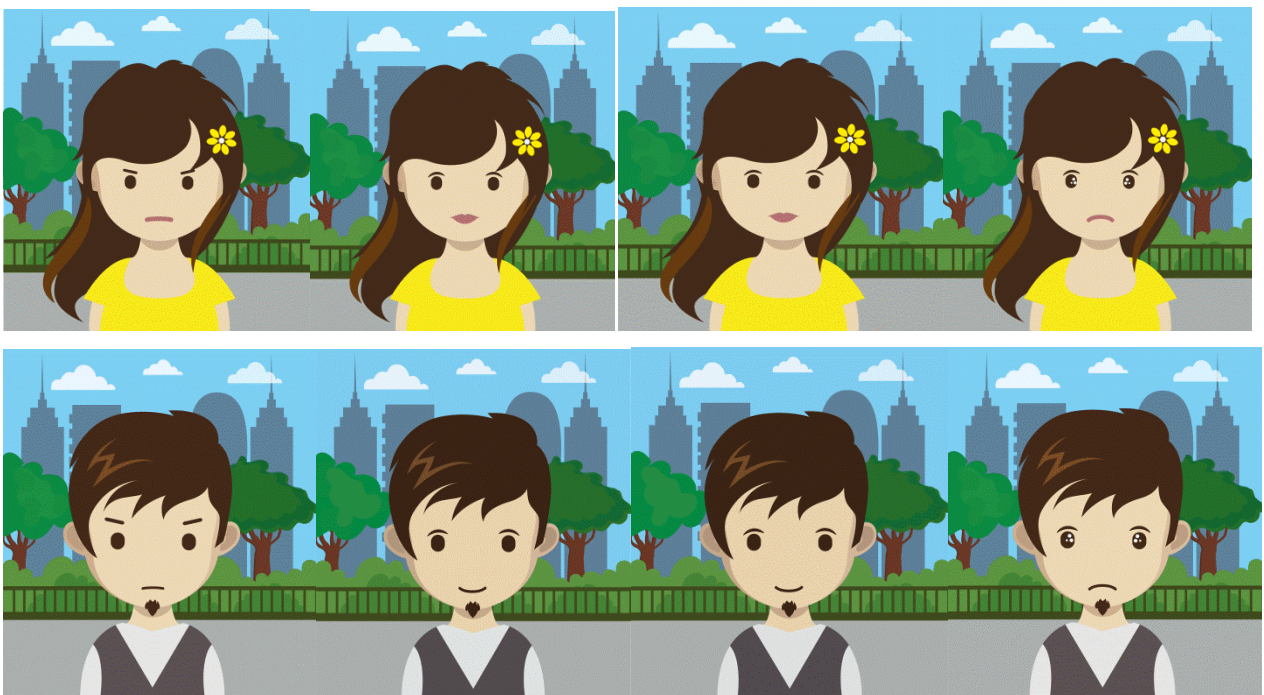


Figura 8b. Estados anímicos de Constanza y Pedro

4.3 Conectividad a la Base de Datos de la plataforma METS

Metsbot se creó también con la finalidad de ser un puente entre usuarios o posibles usuarios y la plataforma, es por ello que el Chatbot también necesita acceder para obtener información de la misma base de datos que la plataforma, en sí Metsbot cuenta con su propia base de datos sin embargo para evitar la redundancia de la información se optó mejor por hacer posible que la API se conectara a la misma base de datos lo cual se realizó de la siguiente manera.

Nodejs necesita de la instalación del NPM para permitir el uso de varias librerías ya existentes por lo cual en este caso se utilizó la librería de Mongoose para la conexión con la base de datos en MongoDB de la API y MYSQL como la librería que conectará a la base de datos en SQL que tiene la plataforma, esta librería cuenta ya con las funciones que permiten la conexión a la base de datos proporcionando únicamente el host, puerto, usuario, contraseña y el nombre de la base de datos a la cual se quiere acceder, de manera adicional se puede gestionar el número de conexiones límite simultáneas que puede generar, ya que esta conexión se realiza una sola vez en la API para N usuarios sin embargo las consultas a la misma generan otra conexión por usuario y ahí es donde se puede aprovechar este límite, para fines de pruebas se ha fijado este límite a 10 conexiones suponiendo que el máximo de usuarios realizando consultas a través de la API serían 10, esta conexión unitaria es un objeto de JavaScript que funciona como un POOL de conexiones. Una vez creado nuestro objeto POOL este cuenta con una función llamada QUERY la cual nos va a permitir pasarle por parámetro un STRING con la consulta a la base de datos en lenguaje SQL ya sean lecturas, escrituras, actualizaciones o eliminaciones. El usuario que se le ha creado a Metsbot solo cuenta con permisos de lectura, ya que su finalidad es solo realizar consultas por lo cual también las funciones que hacen uso de la función “query” en el objeto “pool” están diseñadas para realizar consultas y eliminaciones a través de la función SELECT y DELETE respectivamente del lenguaje SQL, así de esta manera obtenemos el resultado de la consulta a la base de datos que usa la plataforma en la API esta se construye a manera de respuesta para la API de Chatfuel para que finalmente esta información estructurada se muestre como una respuesta en la interfaz del Chatbot proveída por Messenger de Facebook.

Adicionalmente en código la conexión con la base de datos de Mets ya sabemos que se logra con una librería llamada “mysql”, pero para que se logre esta conexión se necesita crear un objeto que contenga el conector brindado por la función createPool al cual se le brinda como parámetros la ubicación de la base datos y sus accesos, esta declaración se ve de la siguiente forma:

```
var pool = mysql.createPool({
  connectionLimit: 10,
  host: 'localhost',
  port: '3306',
  user: 'xxxxxxx',
  password: 'xxxxxx',
  database: 'mets3'
})
```

Donde “pool” será el nombre de la variable, ya que sobre ella se podrán hacer las funciones “query” las cuales crearán un hilo que realizará una query a la base de datos con el query en formato String como atributo.

Host, es la ubicación de la base de datos, como la API de Metsbot se encuentra en el mismo servidor que la base de datos entonces solo indicamos que será en “localhost”, esto se realizó así debido a que por cuestiones de seguridad del laboratorio, la facultad y la universidad los puertos se encuentran cerrados para acceder al

servidor y habría que pedir permisos para que este se habilite entonces posteriormente se tendrían que configurar los permisos de conexión en la base de datos la cual está protegida para que ninguna query externa al servidor sea ejecutada, por ende resulta efectivo el alojar tanto la API como la base de datos en el mismo servidor.

port, es el puerto al cual se accede a la base de datos, por lo general en a los motores de base de datos en SQL creados por Apache se accede a través del puerto 3306 y esta no es la excepción. **user**, es el nombre de usuario que tratará de acceder a la base de datos. **password**, es la contraseña del usuario indicado anteriormente. **database**, es el nombre de la base de datos a la cual se tratará de acceder.

Estos parámetros se pasan a la función “createPool” de nuestro objeto mysql que es la instancia de la librería y tras ejecutarse creara esa conexión que guardará en la variable “pool”, en caso de que la conexión sea errónea entonces se lanzará un mensaje de error y la API estará inactiva.

4.4 Creación de la API y configuración del Servidor

Para crear una API en el framework de Nodejs se necesita solamente crear un objeto servidor y utilizar su función de escucha con el puerto a donde estará escuchando, como cualquier otra aplicación cliente servidor, sin embargo por para obtener mayor robustez para una API como la que queremos crear utilizamos una estructura de una API REST la cual está compuesta por sus funciones de Modelo, Vista y Controlador que son el backend además de usar como un Middleware la librería de Express el cual se encarga de las conexiones, seguridad y la transportación de los datos que reciben y envían el frontend y el backend respectivamente. Siendo así, se instala Express dentro del proyecto y se estructura el mismo en carpetas que contendrán los archivos de JavaScript correspondiente al controlador, el modelo de la base de datos. Las vistas y las rutas, siendo este último el contenedor de los archivos con código que con ayuda del middleware crean estas conexiones para que una aplicación externa o el frontend pueda comunicarse con el servidor a través del protocolo HTTP con sus respectivos llamados POST ,GET, PUT, DELETE, este mismo archivo es el que se encarga de gestionar la seguridad de las conexiones en caso de que así se requiera y recibe las solicitudes para redireccionarlas al controlador indicado que hará uso de las funciones codificadas para manejar los datos de la solicitud y mandar estos mismos a través de la misma ruta que enviara lo que realizó el controlador como una respuesta. Por otra parte, el Modelo se encargará de tener la “plantilla” de cómo es que crea la información en la base de datos además de funciones específicas que se requieran para la interacción con la misma, esta es llamada por un controlador que a su vez fue llamado por una ruta. En efecto hay formas de crear vistas utilizando el mismo framework sin embargo en esta ocasión para Metsbot no hace falta tal función, ya que todo se realiza a través de peticiones de la API de Chatfuel o como un llamado a la API de Messenger de Facebook. Por último, en el proyecto del api existe una carpeta llamada **config** la cual contiene solamente dos archivos, uno de configuración para que el servidor detecte al iniciar en donde se encuentra la base de datos de MongoDB alojada y otro con la creación y configuración del objeto “pool” que permite la conexión con la base de datos de la plataforma y así de esta forma pueda ser instanciada desde los controladores.

4.4.1 Reglas de Negocio

Las reglas de negocio tras definir las nos permiten establecer los límites de la funcionalidad de nuestro software, en este caso al tratarse de un Chatbot con una API integrada que se conecta a una plataforma ya establecida es importante delimitar las funcionalidades que tendrá esta integración, ya que de lo contrario se podría brindar una atención no adecuada por parte del Chatbot o incluso poder afectar la funcionalidad de la plataforma de METS con esta nueva integración de Software porque tendremos acceso a la información de la base de datos por ejemplo.

Las reglas de negocio para Metsbot, entiéndase como Metsbot la integración de todos sus componentes, son:

- Permisos solo de Lectura y Escritura limitados:
La API que alimenta al Chatbot debe acceder a la data que contiene la base de datos de Mets y es por ello que sin importar el usuario con el cual se acceda a la base de datos de manera interna en la API solo se tendrán programadas funciones que NO sobrescriba información existente y tampoco CREAR nuevas instancias, relaciones, campos o tablas nuevas en la base de datos.
- La data adicional será almacenada aparte y no sobrecargar las consultas:
Con relación a lo anterior, Metsbot cuenta con su propia base de datos en la cual incluye parte de la información que no se edita ya existente de la plataforma METS además de nueva información que utiliza el Chatbot para su funcionamiento esto en una base de datos NOSQL. De esta forma el Chatbot cuando requiere información lo cual será con mayor constancia que la plataforma la API que le alimenta ya cuenta con información necesaria para generar queries de SQL que consultará a la base de datos de la plataforma y de esta forma se reduce el número de consultas a la base de datos de la plataforma METS a la mitad.
- Metsbot NO utilizará usuarios ya existentes con privilegios para acceder a la base de datos:
También con relación a lo anterior para seguir garantizando la seguridad de la plataforma de METS cada persona o sistema que quiera acceder a su información debe de contar con su propio usuario, los administradores tienen su propio usuario, la app utiliza otro y Metsbot no será la excepción ante esta regla de negocio heredada de la plataforma, por ende a Metsbot hablando específicamente de su API se le ha creado su propio usuario con sus respectivos permisos y este se hace responsable de manejar y proteger sus propias credenciales.
- El Chatbot es un aditamento a la plataforma, no un sustituto de algún componente:
Como se mencionó anteriormente las funcionalidades de un Chatbot con casi ilimitadas porque incluso estas pueden igualar y/o superar a las funcionalidades que propone una app, es por ello que en ocasiones y dependiendo de la idea es que se opta por crear un Chatbot en lugar de una aplicación móvil, de esta forma también Metsbot podría recrear las funcionalidades que proporciona la aplicación de celular que existe de METS, sin embargo es el objetivo ni la meta de Metsbot sino ser una parte integradora para que se impulse el uso de la plataforma siendo así que la app funcione para brindar información del usuario, información estadística del usuario y un podómetro mientras que Metsbot brinda información concisa del usuario sin gráficas para apoyar de una nueva forma al usuario con la finalidad de no solo mostrarle su información sino de utilizar esa información para alentarle y apoyarlo a que realice sus retos los cuales ya están establecidos y monitoreados por el resto de la plataforma.
- Metsbot no brindará soporte para todos los tipos de usuario:
En la plataforma de METS al registrarte puedes indicar cómo utilizas el monitoreo de tu actividad física, ya sea con el celular como podómetro o con un podómetro convencional para después registrar los pasos de forma manual en la plataforma, esta última es para quienes acceden a METS por medio de su portal web, el Chatbot podría realizar esta tarea también, sin embargo está destinado para complementar el uso de la aplicación y a que sea toda esta interacción totalmente móvil, así que por lo menos en esta versión del Chatbot esa funcionalidad queda descartada debido al poco uso que se le podría dar, ya que de todas formas el ingreso del progreso es manual, este se puede hacer sin problema desde la página web en lugar que desde el Chatbot.

- La información de Metsbot será accesible para los administradores de la plataforma: Metsbot al ser una integración adicional a la plataforma de METS ya existente necesita tener acceso y control por parte de los administradores de la misma, ya que en este documento nos referimos a él como algo adicional, pero recordemos que la plataforma ahora estará compuesta por la integración de web, aplicación móvil y un Chatbot. Debido a ello se facilitan accesos al repositorio, código de la API y base de datos a los administradores de la plataforma, así mismo se crean rutas dentro de la API de las cuales solo tendrán conocimiento los administradores y así poder utilizar funcionalidades del Chatbot en modo administrador es decir que podrán ver todo lo que tiene almacenado, crear, editar o eliminarlo.
- El Chatbot no modificará las rutas de acceso existentes a la base de datos de la plataforma METS: Mets tanto en Web como desde su aplicación cuando realizan una petición a la base de datos lo hacen de formas distintas, uno de forma directa y el otro a través de un Backend, sin embargo la similitud entre estos dos procesos es que ambos se encuentran alojados en el mismo servidor con su configuración de firewall específica en la que por ejemplo solo se permite el acceso al Backend pero no a la base de datos, no existe puerto ni acceso con privilegios que permita acceder a la base de datos de forma remota, es por ello que esto siempre se realizará desde una plataforma que exista en el servidor, incluso la página de Mets está en ese servidor es por ello que ahí podemos ver dicha información que nos regresa.
En el caso de Metsbot no será diferente y la API se integrará en el mismo servidor, en su propia sesión. Así solo se configura un puerto que sea accesible a la API y esta se encargará de hacer lo suyo con el acceso a la base de datos, es decir la API no puede estar alojada en otro servidor o servicio.

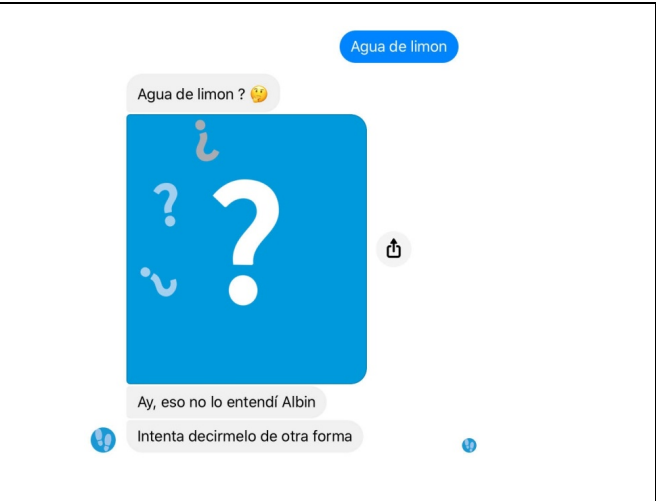
4.5 Funciones del Chatbot

Las funciones que realiza el Chatbot/healthbot que hemos nombrado como Metsbot están basadas en las posibles peticiones que realizaría un usuario y que estas a su vez sean permitidas por las líneas de conversación anteriormente definidas en el diagrama del flujo conversacional. Es decir, las funciones que hace el Chatbot están dadas por aquello que puede “entenderle” a un usuario cuando tengan una conversación que se detallan en la tabla 6.

Tabla 6. Funcionalidades Metsbot para usuarios aún lo registrados

Funcionalidad-Respuesta
Saludo por botón: cuando se abre una ventana de chat en MSN de Facebook con un Chatbot este proporciona como interfaz únicamente un botón que al ser apretado inicia la conversación.
Saludo por texto: todo usuario que utilice a Metsbot forzosamente tendrá que activar el botón anteriormente mencionado “Iniciar conversación” por ende se sobreentiende que saludos posteriores en forma de texto que brinde el usuario como por ejemplo “Hola” o “¿Qué tal?” y texto similar será clasificado como un intento de saludo, el cual Metsbot procesa e iniciara una consulta para saber si el usuario ya utiliza la plataforma o no, de acuerdo a esa sentencia es como inicia una conversación diferente para cada caso

Sentencias de texto no reconocido: cuando el usuario escribe algo que Metsbot no entiende o sale del tema que le involucra este se reconocerá como un intento desconocido lo cual activará una sentencia compuesta por tres mensajes indicando al usuario que Metsbot no entiende lo que trata de decir, un mensaje por cada uno de los tres intentos fallidos siendo el último un mensaje pidiendo al usuario que si experimenta algún inconveniente puede ponerse en contacto con el laboratorio de tecnologías de la salud a través de un email, después de este intento si se registra un cuarto intento desconocido esta sentencia se reiniciará con el primer texto nuevamente.



Sentencia para nuevos usuarios de Mets: tras el saludo inicial ejecutada por el botón de “Iniciar conversación” se le pregunta al usuario si ya es usuario de la plataforma de METS, cuando el usuario indica que no se activa esta sentencia que inicia una nueva conversación para este usuario.

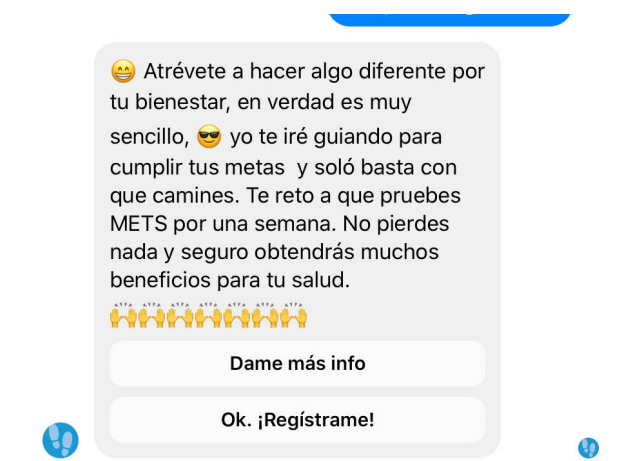
Sentencia “Me identifico”: durante la conversación para nuevos usuarios se les presenta una serie de imágenes con las cuales el usuario podrá pulsar sobre el botón de “Me identifico”, según la imagen seleccionada cambia la conversación, ya que esto tiene relación con su nivel de sedentarismo.

Sentencia de usuario no interesado: después de continuar con la conversación de acuerdo a su nivel de sedentarismo se vuelve a converger en el flujo principal de la conversación llegado a este punto se le propone al usuario utilizar la plataforma de METS y el healthbot, si este menciona de forma negativa a la propuesta se inicia esta sentencia para preguntarle al usuario el motivo de la respuesta la cual considera los principales motivos de los encuestados con respecto a la falta de actividad física.

Sentencia usuario si interesado: con respecto a la sentencia anterior si el usuario si muestra interés se inicia esta sentencia donde se le da más información al usuario y se le da a escoger entre conocer más información, registrarse o mejor no continuar, donde esta última opción iniciaría la sentencia de usuario no interesado.


Sentencia de Información: si en la sentencia anterior el usuario desea más información se activa esta sentencia que le proporciona más información acerca de METS y el sedentarismo proporcionando adicionalmente links de información que se encuentran en la página de Ubisalud, por último, se le da la opción de registrarse de una vez.

Sentencias de excusa: estas series de sentencias son un complemento a la sentencia de usuario no interesado donde según la respuesta dada se trata de convencer al usuario con diálogo estructurado según la terapia de aceptación de las precauciones, dando como última opción el darle más información, lo cual activa la sentencia de información, o de registrarse de una vez en Mets.

<p>Sentencia de usuario definitivamente no interesado: dentro de la conversación guiada por texto y botones no se le da mucha libertad al usuario de no querer utilizar el Chatbot por ende es probable que este se comunique con el Chatbot mencionando que no desea registrarse o que no quiere utilizar la aplicación, este tipo de comentarios se clasifican como un intento de rechazo a la plataforma y por ende solo se responde invitando al usuario que cuando desee puede conversar con Metsbot nuevamente.</p>	 <p>Atrévete a hacer algo diferente por tu bienestar, en verdad es muy sencillo, 😊 yo te iré guiando para cumplir tus metas y soló basta con que camines. Te reto a que pruebes METS por una semana. No pierdes nada y seguro obtendrás muchos beneficios para tu salud.</p> <p>👏👏👏👏👏👏👏👏👏👏👏👏</p> <p>Dame más info</p> <p>Ok. ¡Regístrate!</p>
--	---

En cualquiera de las sentencias anteriores menciona que si se quiere registrar en la plataforma se activa esta sentencia la cual le anima y le pregunta si ya es usuario o no. En la tabla 7 se presentan las sentencias para los usuarios ya registrados.

Tabla 7. Funcionalidades Metsbot para usuarios ya registrados

Funcionalidad-Respuesta	
<p>Sentencia ya es usuario: si ya es usuario se activa esta sentencia que inicia al coach y le pide el correo con el que se registró en METS haciendo una petición a la API.</p>	
<p>Sentencia usuario encontrado: si la API responde que sí existe el usuario entonces se activa esta sentencia llamando al usuario por su nombre y preguntarle si desea iniciar con el coaching o desea ver un tutorial.</p>	
<p>Sentencia usuario no encontrado: esta sentencia se inicia cuando la API responde que el usuario no existe en la plataforma y le pide al usuario que se registre o que lo intente nuevamente de forma cíclica.</p>	
<p>Sentencia usuario encontrado: si la API responde que sí existe el usuario entonces se activa esta sentencia llamando al usuario por su nombre y preguntarle si desea iniciar con el coaching o desea ver un tutorial.</p>	
<p>Sentencia usuario encontrado: si la API responde que sí existe el usuario entonces se activa esta sentencia llamando al usuario por su nombre y preguntarle si desea iniciar con el coaching o desea ver un tutorial.</p>	
<p>Sentencia ver tutorial: sentencia que se activa cuando se le propone al usuario ver un tutorial y este acepta la cual solo presenta un video y da la opción de continuar, lo cual activa la sentencia de inicio del coaching.</p>	
<p>Sentencia de usuario nuevo en METS: cuando en la sentencia de registro a METS el usuario indica que aún no se ha registrado se activa esta sentencia la cual le brinda un botón para que vaya al link y se registre en la plataforma de METS dentro de la página de Ubisalud para posteriormente mencionarle al usuario que termine su registro y que Metsbot estará en espera a que termine.</p>	
<p>Sentencia registrada: cuando el usuario se ha registrado y le menciona a Metsbot que ya se registró, este tipo de oraciones se clasificaron como una intención de que el registro se ha terminado y activará esta sentencia donde le preguntará al usuario si ya descargó la aplicación, en caso de que ya la haya descargado se activa la sentencia ya es usuario, de lo contrario se activa la sentencia de descarga.</p>	 <p>Ok, eso fue algo rápido 😊 Aún no me acabo mi café ☕</p> <p>Albin ¡FELICIDADES YA ESTÁS EN METS! ¿Ya descargaste la app?</p> <p>Aún no Voy a usar podómetro Ya 😊</p>
<p>Sentencia de descarga: esta sentencia le pregunta al usuario qué tipo de celular usa, Android o iPhone.</p>	

Sentencia Android: le proporciona al usuario el link de descarga de la app para Android y se queda en espera a que ya haya iniciado sesión el usuario en la app. (Cuando el usuario indica que ya inicio sesión se activa la Sentencia de ya es usuario).
Sentencia iPhone: le proporciona al usuario el link de descarga de la app para iOS y se queda en espera a que ya haya iniciado sesión el usuario en la app (Cuando el usuario indica que ya inició sesión se activa la Sentencia de ya es usuario).
Sentencia inicia Coach: cuando un usuario da su correo y la API si reconoce que es un usuario se activa esta sentencia donde el coach que se seleccione al registrase en la plataforma le da la bienvenida al usuario y le pregunta por la hora a la que quiere ser recordado de su progreso.
Sentencia menú: cuando el usuario ha definido la hora inmediatamente se activa esta sentencia donde se le proporciona al usuario un menú con las opciones disponibles en todo momento de las cuales puede hacer uso.
Sentencia progreso del día: sentencia activada por el menú que le brinda al usuario el progreso que lleva el día de hoy y de acuerdo al porcentaje se le proporciona un mensaje diferente.
Sentencia Progreso total del reto: sentencia activada por el menú donde se le muestra al usuario una respuesta personalizada de acuerdo al porcentaje del reto cumplido, este en caso de ser muy bajo y que ya han pasado días sin que aumente le pregunta al usuario el motivo de su poca actividad física la cual se almacena a través de la API y brinda un mensaje genérico comprensivo para que el usuario no se sienta molesto o apenado con su respuesta.
Sentencia detalle del reto: sentencia activada por el menú que le brinda al usuario solo información de como es su reto actual.
Sentencia cambiar hora: sentencia activada por el menú donde se le permite al usuario cambiar la hora en la que desea ser recordado de su progreso.
Sentencia activa menú: si el usuario a través de su petición se identifica como un intento de que se le brinde el menú este se activara y solo activará el menú sin texto para que el usuario siga interactuando.
Sentencia de usuario baja: esta sentencia se activa cuando se detecta un intento relacionado con ya no querer usar Mets o Metsbot, el cual solo muestra un diálogo donde se le indica comunicarse con Ubisalud con ciertos datos para que el usuario quede desvinculado de ambas aplicaciones.

4.6 Funciones de la API

Las funciones que realiza la API están basadas en las peticiones que permite realizar el Chatbot y estas son:

A. Funciones hacen referencia al controlador de nombre “email”

- getMessage, la cual permite testear la conexión con el servidor para saber si está activo.
- getText, el cual permite testear que la API tiene establecida una conexión con la base de datos de la plataforma Mets.
- save Email, verifica que el correo electrónico proporcionado por el usuario exista en la base de datos de la plataforma como un usuario y llama a la función de crear un usuario para la base de datos interna de Metsbot.
- setReminder, guarda en la base de datos interna de Metsbot la hora a la que el usuario indica que le gustaría ser recordado de su progreso con la ayuda de la función setValues
- setValues, es la función llamada por setReminder la cual hace un update en la base de datos interna de Metsbot con la hora que el usuario eligió y la meta de sus pasos.

B. Funciones hacen referencia al controlador coachingTools

- insertMotivo, ingresa un string referente a una respuesta del usuario en el array de motivos que se almacena en la base de datos.
- consultaProgresoHoy, checa en la base de datos de la plataforma cuanto ha caminado el día de hoy en forma de porcentaje.
- infoReto, consulta en la base de datos cuál es el reto al cual está inscrito el usuario.

- consultaReto, obtiene de la base de datos de la plataforma y regresa la información sobre cuánto tiene que lograr el usuario para cumplir su reto y por cuantos días.
 - sendAvatarInit, sendAvatarMad, sendAvatarBien, sendAvatarSad, sendAvatar Angry, son las funciones que obtienen de la base de datos de interna de la API, cuál es el número de su avatar y así poder mandar como respuesta el link de la imagen correspondiente al estado de ánimo del avatar específico del usuario.
 - lastDayReminder, se activa a través de un cronjob para indicar el último recordatorio al usuario de que no ha cumplido su reto.
 - checkDays, se activa a través de un cronjob y verifica los días que no ha cumplido los retos el usuario para advertirle o darlo de baja de la plataforma.
 - sedentarismoAlert, se activa a través de un cronjob para alertar a los usuarios que llevan horas sin actividad física.
 - reminderAlert, es la alerta que se activa de acuerdo a un cronjob para verificar si ya es la hora a la que el usuario indicó que le gustaría ser notificado de su progreso.
 - reminderGenerator, es la función llamada por reminder alert en caso de que si sea la hora indicada por el usuario y realiza el proceso de notificación.
 - generaMsg, recibe información para parsearla y llamar a sendRequest
 - sendRequest, realiza un POST a la API de Messenger de Facebook con para enviar un mensaje.
 - coachDelete, función que elimina al usuario de ambas bases de datos.
 - botCreateUser, función que toma el email enviado y lo registra dentro de la base de datos propia de Metsbot.
- C. Funciones hacen referencia al controlador AdminTools, la cual se compone de un CRUD dentro del api con acceso a las rutas de administrador el cual está pensado para ser conectado con un sistema de administrador donde se pueda gestionar a los usuarios del Chatbot.
- createUser, se crea un usuario para Metsbot.
 - getAllUsers, obtiene todos los usuarios de la base de datos de Metsbot.
 - getUser, obtiene un usuario en particular de la base de datos de Metsbot.
 - updateUser, modifica a un usuario en particular de la base de datos de Metsbot.
 - deleteUser, elimina un usuario en particular de la base de datos de Metsbot.
- D. Funciones hacen referencia al controlador phrases, el cual está compuesto por un CRUD para las frases motivacionales que puede decir el Chatbot para cuando este menciona el motivo por el cual no ha podido cumplir con el reto.
- getPhrase, obtiene todas las frases.
 - createPhrase, se inserta una nueva frase.
 - updatePhrase, modifica una frase.
 - deletePhrase, elimina una frase.
 - getRandomPhrase, obtiene una frase de manera aleatoria.

4.7 Creación de una app en Facebook

Para utilizar la plataforma de MSN de Facebook en especial para poder mandar mensajes desde un servicio externo a Facebook como por ejemplo nuestra API, entonces, es necesario crear una “aplicación de Facebook” como se muestra en la figura 9. Esta app se crea a través del portal de developers.facebook.com donde pedirán cierta información, tras crearla hay que definir qué tipo de app es, en este caso se activan los productos de

Messenger, para utilizar la plataforma de mensajería y Webhooks, para tener en escucha nuestra app a las peticiones que realice nuestro servidor y viceversa.

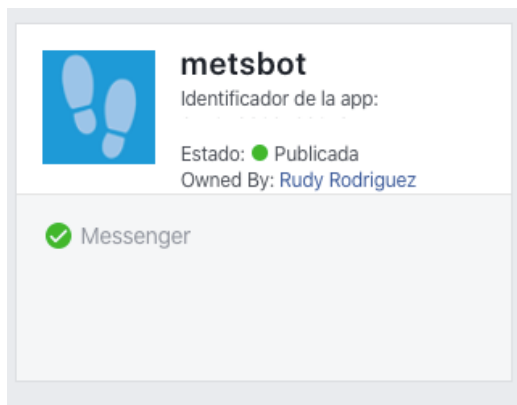


Figura 9. Ventana de Facebook que muestra la aplicación creada.

Para finalizar la implementación se realiza una configuración básica y una avanzada que consiste en llenado de información sobre la app como: nombre, dominios, política de privacidad, categoría e icono de la app. Esto va de la mano con la revisión de la app la cual consiste en crear una solicitud mediante un formulario y posteriormente enviarlo a Facebook, donde indicaras aquellos permisos y funciones que deseas utilizar o tener acceso, se envía para que luego ellos verifiquen que se le haga buen uso a esos permisos, por último al aprobarlo te darán acceso a dichos permisos y funciones, los cuales se mostraran con un círculo de color verde indicando que están activos y tienes acceso como se muestra en la figura 9a. Estos permisos y funciones contienen túneles de acceso a la información de los usuarios de Facebook que utilicen la aplicación creada.



Figura 9a. Ventana de permisos y funciones que tiene activas la aplicación de Facebook.

4.8 Chatfuel la plataforma líder de Chatbot para Messenger Facebook






Chatfuel nace en el verano del 2015 con la meta de facilitar a cualquiera la creación de bots lo cual la convirtió en la plataforma de creación de bots para Facebook Messenger líder a nivel mundial, la plataforma a través de su interfaz intuitiva permite de manera sencilla crear Chatbots e implementarlos en una página de Facebook incluso sin necesidad de configurar una app de Facebook, algunas de las marcas más reconocidas que utilizan la plataforma son: Adidas, TechCrunch, T-Mobile, LEGO, Golden State Warriors [57, 58].


Cabe mencionar que esta plataforma en efecto permite la construcción de un Chatbot de forma sencilla, pero este se verá limitado a funciones de diálogo y respuesta más las funciones extras que ofrece Facebook de forma nativa como las “Cards”, “List”, “Buttons” y “Quick Replies”. En nuestro caso necesitamos algo más robusto para la conexión con METS, almacenar información y poder enviar mensajes desde la API, por ello es que Chatfuel se utilizó solo como plataforma para modelar nuestro diagrama de flujo conversacional en un Front-end de Chatbot dentro de la plataforma de Messenger de Facebook, también como clasificador de texto y se adaptará para que a su vez tenga la funcionalidad de hacer llamados a los endpoints de nuestra API.

4.9 Descripción de la anatomía de Metsbot

Con lo mencionado anteriormente ahora concebimos a Metsbot como la unión de una plataforma que nos brinda el Front-end, un Middleware que aparte de poder crear la comunicación entre Front-end y Back-end también podremos crear la conversación del Chatbot y por último, nuestra API como el Back-end para lograr la conectividad con la plataforma de METS. Siendo así que podemos visualizarlo como el siguiente diagrama mostrado en la figura 10 apoyado de la tabla 8 donde se describen cuáles son sus componentes.

Tabla 8. Descripción de las partes que componen el diagrama de la anatomía de Metsbot.

Componente	Nombre de la herramienta	Imagen que lo identifica
Front-End: Donde interactúa el usuario.	Messenger de Facebook.	
Middleware: Capa oculta que permite la configuración y comunicación del front-end con el back-end.	Plataforma de Chatfuel.	
Back-End: Capa oculta que contiene la lógica funcional para la comunicación con el servidor y las bases de datos.	API REST.	
Base de datos de la plataforma Mets: La base de datos de Mets.	No contiene herramienta, es comunicación directa con la base de datos de la plataforma Mets con SQL a través de la API REST.	
Base de datos de Metsbot: La base de datos de Metsbot.	No contiene herramienta, es comunicación directa con la base de datos de Metsbot con MongoDB a través de la API REST.	

Comunicación: las flechas azules indican “Se comunica con:” y las flechas verdes “Recibe respuesta de:”	Protocolos de comunicación HTTPS y HTTP.	
---	--	---

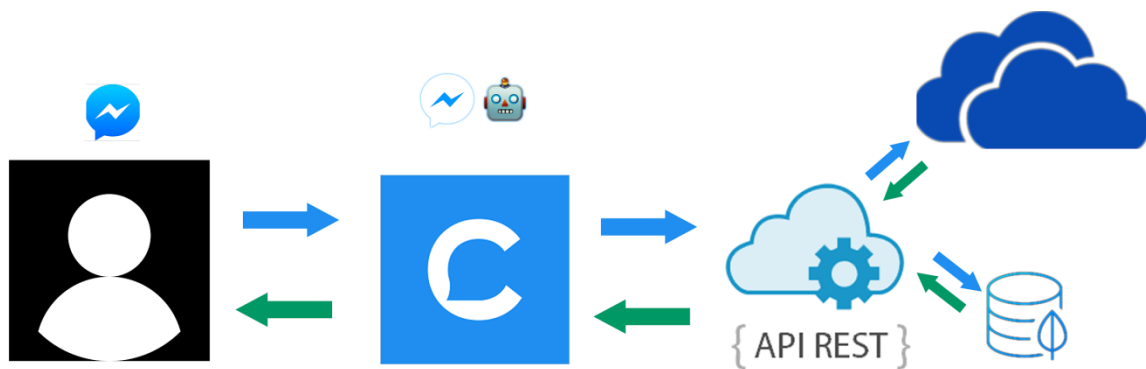


Figura 10. Diagrama de la anatomía de Metsbot y su comunicación.

Resulta importante mencionar que de forma interna cada parte del diagrama tiene su propia estructura y construcción, la cual vamos a explorar a continuación, explicando que contiene cada una. De ahora en adelante a cada una de estas partes (Front-end, Middleware, Backend y sus subpartes) las denominaremos como *COMPONENTES* siendo estos:

- Componente de usuario.
- Componente de Chatfuel.
- Componente de la API.

Esto con la finalidad de poder englobar las partes de todo Metsbot (Front-end, Middleware, Backend) en sus respectivas secciones mientras describimos de que están compuestas cada una. Cabe mencionar que la división entre Front-end y Middleware no se encuentra marcada debido a como están integrados los componentes de cada una de estas partes.

4.9.1 Componente de Usuario

Recordando que Mets y Metsbot son plataformas distintas debemos diferenciar entre los tipos de entes que interactúan con ambas, en este caso particular las personas. Por ellos vamos a describirles de la siguiente manera:

- Usuario: una persona que utiliza Facebook, Messenger de Facebook y Metsbot, pero no necesariamente utiliza la plataforma de Mets.
- Usuario de Metsbot: una persona que utiliza Facebook, Messenger de Facebook, Metsbot y aparte utiliza la plataforma de Mets (app).
- Posible usuario de Metsbot: una persona que utiliza Facebook y Messenger de Facebook, pero no utiliza Metsbot y no necesariamente utiliza la plataforma de Mets.

Como podemos visualizar, aquello que todos los tipos de usuario tienen en común es el hecho de que forzosamente tienen que ser usuarios de Facebook y Messenger de Facebook, independientemente de cual usen con frecuencia, ya que, si una persona se registra en Facebook, pero solo utiliza Messenger de Facebook con su cuenta no afecta en nada ya que aun así puede encontrar y utilizar Metsbot.

Debido a esta similitud el componente de usuario partirá desde un posible usuario de Metsbot en donde la persona pertenece a la red social y utiliza la plataforma de mensajería, con estos dos requerimientos ya existe la posibilidad de que puedan conocer de la existencia de Metsbot. En cuanto a la cuestión del cómo se va a lograr esto, como las personas van a encontrar y/o utilizar Metsbot son temas referentes a la difusión donde interactúa el área de la publicidad, marketing digital y difusión de redes sociales que no corresponden al desarrollo del Chatbot y por ende no se ahondará más en el tema.

Para que los posibles usuarios se conviertan en usuarios y posteriormente usuarios de Metsbot es necesario ser parte de la red social de Facebook, es por ello que se crea una página de Facebook como se mencionó anteriormente y la cual se encuentra alojada en el siguiente dominio <https://www.facebook.com/metsbot>. Tras haber creado la página ahora tenemos una instancia de la red social la cual nos da posibilidades de publicar dentro de la página, crear elementos para su publicidad, etc. aunque la que nos importa es el componente de chat, toda página de Facebook al igual que los perfiles de sus usuarios tienen la posibilidad de acceder a la plataforma de mensajería llamada Messenger de Facebook, en este caso no es la excepción, para facilitarle el acceso al usuario dentro de la página se colocó un botón de “enviar mensaje” que por defecto se encuentra oculto, sin embargo como la finalidad de la página es que ahí sea donde reside nuestro Chatbot se optó por esta opción. Con estos elementos ya tenemos un acceso para atraer a los posibles usuarios todo dentro de la red social.

4.9.2 Componente de Chatfuel

Chatfuel como plataforma permite trasladar estrategias de conversación o diagramas de flujo conversacionales, en concreto para este caso, en un Chatbot funcional e integrado ya en una página y aplicación de Facebook, donde las entradas de texto del usuario ya se responden. Esto facilita la programación de la interfaz e interacción, ya que se basa en una estrategia similar a la programación visual por bloques como usa la plataforma de Scratch (<https://scratch.mit.edu/about>) que es multipropósito para programación, pero en este caso Chatfuel está dedicado a los Chatbots que funcionaran en una interfaz de Messenger de Facebook.

Esto último de la integración con Facebook es parte del ciclo de vida en la creación de un Chatbot en Chatfuel, como ya se mencionó anteriormente esta herramienta funciona exclusivamente para Messenger de Facebook, por lo cual tras registrarse en la página de Chatfuel lo primero que te permite hacer es crear un nuevo Chatbot en tu tablero de inicio o Dashboard como se muestra en la figura 11, en donde estarán almacenados todos los Chatbots que hagas con la plataforma, tras crear un nuevo Chatbot lo primero que te piden hacer, además de nombrar a tu Chatbot, es “conectarlo”, en un inicio la plantilla o template del Chatbot te aparece como desconectado, esto es normal debido a que no hemos entrado todavía a su estructura y solamente lo creamos como un Chatbot desde cero o con una plantilla vacía, sin embargo lo interesante de ello es que aparece de forma inmediata el botón de conectar, obligando al usuario a que desde un inicio conecte o ligue su Chatbot a una página de Facebook de la cual sea propietario el creador para que este Chatbot se pueda alojar ahí y así posteriormente poder crear esa plantilla en un Chatbot funcional dentro del Messenger correspondiente a esa página.

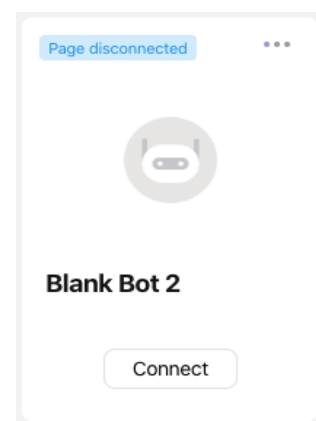


Figura 11. Ventana para crear un nuevo Chatbot dentro de Chatfuel.

Esta interfaz para llenar la plantilla de contenido del Chatbot contiene su propio tipo de componentes que al utilizarlos y combinarlos crean un Chatbot funcional. A continuación, se muestran los componentes que existen dentro de la herramienta, así como su funcionalidad.

Los componentes están acomodados en capas donde en el exterior tenemos los grupos, secuencias y grupos. Donde los grupos son quienes contienen los componentes más atómicos que permiten la funcionalidad del todo.

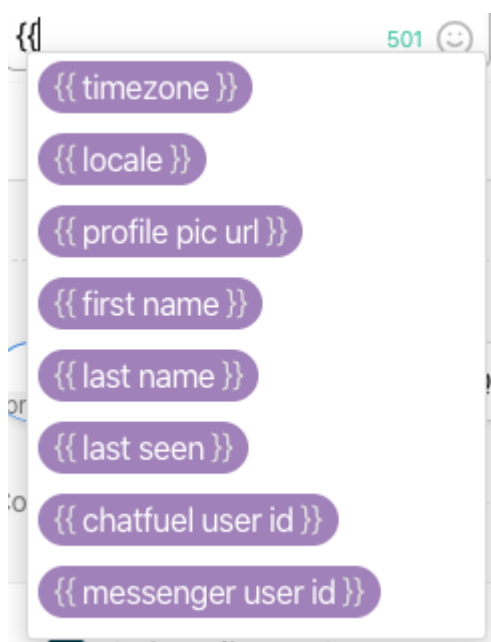
4.9.2.1 Bloques

Los bloques en Chatfuel son el componente central y fundamental de la herramienta, ya que sin bloques no hay Chatbot, un bloque se convertirá en un “pedazo” de la conversación porque este contiene todas las funcionalidades posibles a realizar, cabe mencionar que estas funcionalidades están limitadas a lo que Messenger de Facebook tiene permitido enviar, las funcionalidades/componentes que se pueden utilizar en los bloques ya sea de forma independiente o en conjunto son:

- Texto: es un bloque de texto que el creador escribe como respuesta textual que visualizará el usuario, esta puede contener variables internas de la herramienta o externas que la herramienta puede almacenar.
- Foto: Permite una inserción de imagen o imágenes consecutivas que se cargan en la herramienta.

4.9.2.2 Atributos del Chatbot

Chatfuel dentro de su interacción con los grupos y bloques contiene un componente llamado “Atributos”, los atributos son en pocas palabras variables. Variables que ayudan a la creación del Chatbot dándole dinamismo a este. Todos los atributos dentro de las funciones de Chatfuel se pueden distinguir porque estos están representados por el nombre de la variable y este encerrado entre dos corchetes. Por ejemplo “{{NombreDeUsuario}}”.



Por defecto Chatfuel te proporciona con una serie de atributos por defecto a las cuales se puede acceder gracias a la conexión con una página de Facebook como se muestra en la figura 11a, estas son: zona horaria, localidad, url de la foto de perfil, nombre del usuario, apellido, última fecha de interacción, id del usuario en Chatfuel, id del usuario en Messenger, suscripción, género, nombre del último bloque visitado, nombre del último botón al que se le hizo click, origen, último texto escrito por el usuario como respuesta, referencia, fuente original, referencia original.

Figura 11a. Ventana que muestra como se crean los atributos del Chatbot ó se selecciona uno previo.

Los atributos que se brindan por defecto se pueden distinguir porque Chatfuel los coloca dentro de su interfaz con color morado, a diferencia de los atributos personalizados que han sido creados/declarados por el creador del Chatbot los cuales aparecerán de color café.

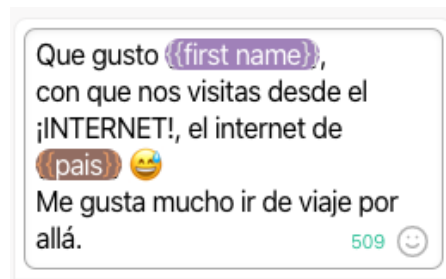


Figura 11b. Bloque de conversación con texto que contiene atributos.

Conociendo el hecho de que el creador del Chatbot también puede declarar sus propios atributos, estos le servirán como variables no solo para almacenar y mostrar información obtenida sino para poder realizar otro tipo de operaciones comunes en la programación como condicionales o hasta mandar información a alguna API externa. Para lograr esto al creador le basta con utilizar alguno de los cuatro bloques dedicados a estas operaciones: Guardar teléfono del usuario, Guardar Email del usuario, Guardar entrada del usuario y Definir atributo del usuario. Siendo los últimos dos los más importantes por ser los más genéricos y por ende adaptables a las necesidades de la conversación que tendrá el Chatbot.

A. Save User Input (Guardar entrada del usuario)

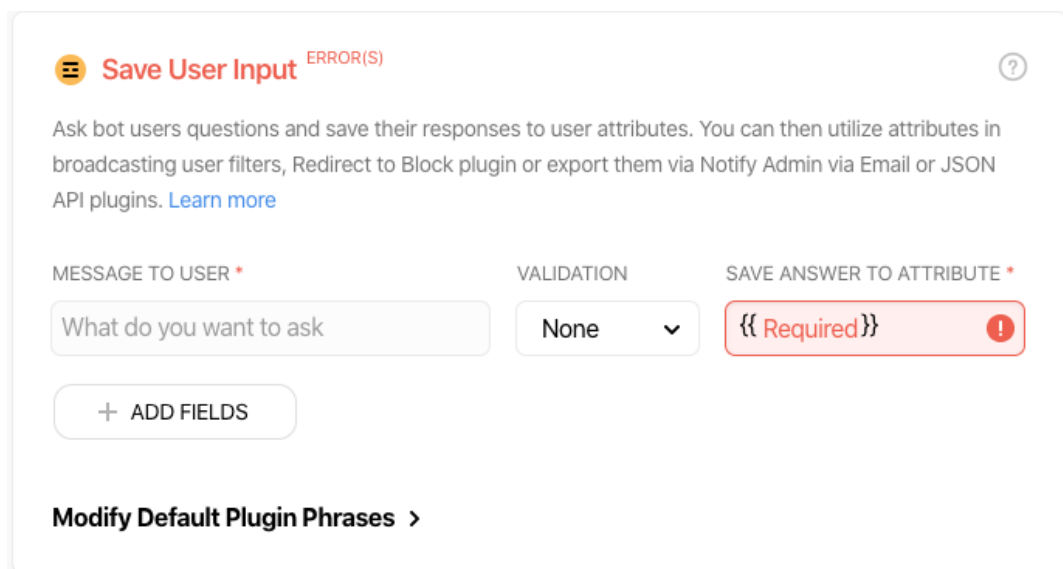


Figura 11c. Ventana del componente que muestra como se crea un atributo “Save User Input”.

Este bloque nos permite crear una pregunta al usuario para que la respuesta sea almacenada en un atributo que se declarara en el mismo, por ejemplo, si preguntamos “¿Cuál es tu edad?” guardaremos la respuesta del usuario en el atributo “{{Edad}}”. Como se puede apreciar en la imagen este contiene una validación simple que solo funciona para poder verificar si la respuesta que dio el usuario es debe ser un Email, número

telefónico o un número, en el caso de que se tenga una validación se podrá configurar una respuesta que indique el error y regresar al usuario a la pregunta inicial hasta que ingrese un dato válido. En el ejemplo anterior el usuario podría responder a la pregunta de la edad con un “Treinta años” lo cual es un texto más no un número, si colocamos la validación de número y tras definir la respuesta al error el ejemplo anterior podría quedar de la siguiente forma: “¿Cuál es tu edad?”, validación: number, atributo: “{{Edad}}”. Lo cual nos daría un flujo como el siguiente expresado en la tabla 9:

Ejemplo de conversación.	
Usuario:	Chatbot:
	“¿Cual es tu edad?”
“Veintitrés años”	
	“Disculpa, ¿podrías poner tu edad con números?, así me seria mas fácil, Gracias.”
	“¿Cual es tu edad?”
“ 23 ”	*Captura y guarda el atributo de {{Edad}}*
	“Perfecto, vaya que eres joven aun...”
(continua el flujo de la conversación)	

Tabla 9. Ejemplo de conversación con un atributo “Save User Input”.

B. Set User Attribute (Definir atributo del usuario)

Figura 11d. Ventana del componente que muestra como se crea un atributo “Set User Input”.

En este bloque nosotros declaramos como se llamara el atributo y seguido pondremos cualquier valor de cualquier tipo con el cual será definido ese atributo o incluso si queremos que no tenga valor al principio podremos definirlo como “not set” que, a modo de ejemplo, sería como definir una variable como “undefined” en JavaScript. Un ejemplo de esto sería lo siguiente: atributo: “{{EdadFutura}}” valor: “{{Edad}} + 10”

De esta forma tendremos la edad del usuario más diez almacenada en nuestro atributo “EdadFutura” a la cual el Chatbot podrá ya acceder, por ejemplo, retomando el caso anterior y suponiendo que el Chatbot sirve para vender seguros:

Ejemplo de conversación.	
Usuario:	Chatbot:
	“{{Firstname}} dentro de 10 años tendrás {{EdadFutura}}, ¿has pensado en ello?, ¿alguna vez has pensado en que será de tu vida a los {{EdadFutura}}?, yo tampoco, pero si seria bueno prevenir ciertos posibles casos...”
(continúa con la conversación)	

Tabla 9a. Ejemplo de conversación con un atributo “Set User Input”.

4.9.2.3 Grupos

Es una forma de clasificación con la que cuenta Chatfuel en la cual se puede agrupar una selección de bloques y asignarles una etiqueta que permitirá distinguir al grupo de bloques, así como poderlo ocultar y desplegarlo en la interfaz como se puede ver en la figura 11e. La finalidad de este componente es poder clasificar los bloques de acuerdo a un criterio que el creador tenga establecido, por ejemplo, todos los bloques que tengan que ver con imágenes se asignaron a un grupo llamado “respuestas en imagen” para saber qué todas las respuestas en imagen se encuentran ahí y así poder manipular más fácil los bloques, este componente es meramente para organizar la lógica del Chatbot.

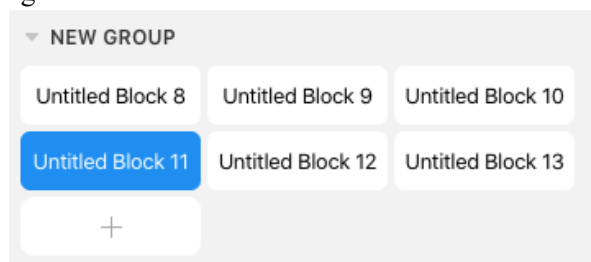


Figura 11e. Ventana que muestra como se ve un Grupo en Chatfuel.

4.9.2.4 Secuencias

Chatfuel te permite crear un bloque en específico o una serie de bloques con la funcionalidad de activarse y por ende manifestarse en Chatbot a un determinado tiempo ya sean segundos, minutos, horas o días como podemos visualizar en la figura 11f.

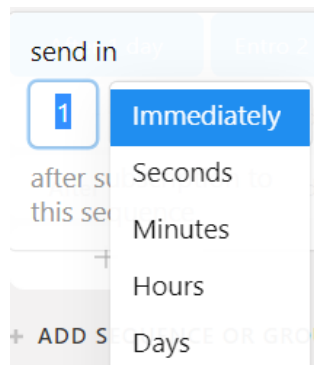


Figura 11f. Ventana que muestra dentro de cuanto tiempo se enviara la Secuencia en Chatfuel.

Esta serie de bloques recibe el nombre de “Secuencia” el cual es una parte del flujo conversacional aparte de o de los flujos principales a la cual un usuario puede “suscribirse” para recibir esos mensajes en determinado tiempo, esto sucede cuando en el o los flujos principales de la conversación existen ciertos bloques que al mostrarse con el usuario esté entre sus instrucciones, cómo mostrar texto, imagen, JSON API, etc., se declara que en ese momento el usuario será suscrito a una secuencia.

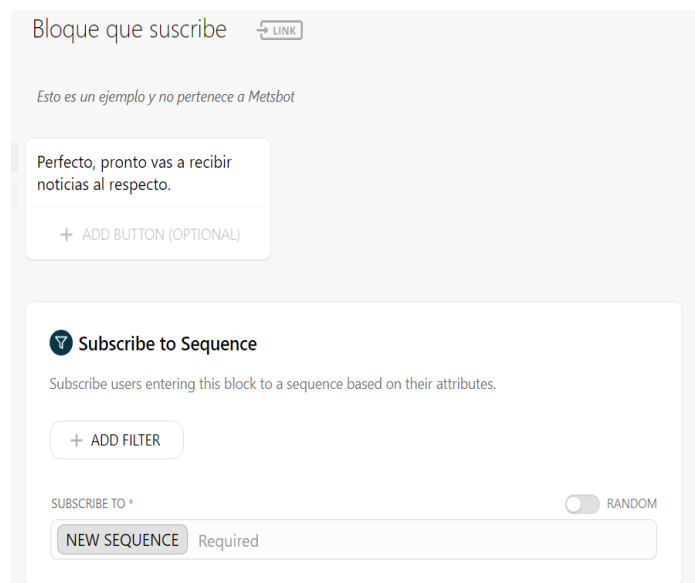


Figura 11g. Ventana que muestra como se ve una suscripción a una Secuencia en Chatfuel.

Como el ejemplo de la figura 11g lo muestra, el usuario solo verá el texto “Perfecto, pronto vas a recibir noticias al respecto.” Pero de manera interna ya está suscrito a que recibirá mensajes o un mensaje dentro de un determinado tiempo y de forma automática los cuales se encuentran dentro de la secuencia “NEW SEQUENCE”.

Las secuencias de Chatfuel como ya se explicó, de forma interna solo son una serie de bloques como se muestra en la figura 11h que tras suscribirse como se vio en la figura 11g, esta se activa y se enviará el contenido de la secuencia y del bloque o bloques al usuario después de cierto tiempo previamente asignado, lo cual se ve de la siguiente forma:

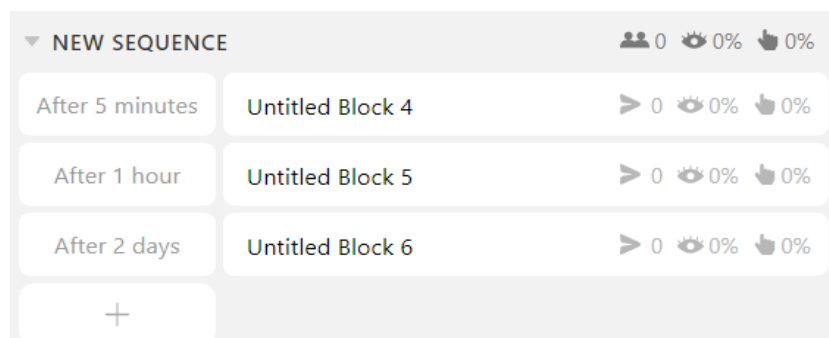


Figura 11h. Ventana que muestra dentro de cuanto tiempo se enviara la Secuencia en Chatfuel.

Como podemos ver en la secuencia nombrada “NEW SEQUENCE” de la figura 11h a la cual se suscribió el usuario en el bloque anterior contiene de la figura 11g, tenemos tres bloques, cada uno con sus respectivas acciones, pero uno se activará a los 5 minutos de haberse suscrito, el siguiente a la hora de haberse suscrito y el último después de dos días de haberse suscrito.

Cabe mencionar que todo usuario que ya haya sido suscrito a una secuencia recibirá mensajes de los nuevos bloques agregados siempre y cuando se cumpla el tiempo y que también, así como hay bloques para suscribirse existe la opción de “quitar suscripción” disponible entre los elementos de un bloque.

4.9.2.5 Setup AI, la Inteligencia artificial para el procesamiento de lenguaje natural

Este componente se encuentra fuera de los principales debido a su funcionalidad que agrega valor al Chatbot, pero no es fundamental para su creación. El valor que le agrega al Chatbot es como lo dice su nombre, la configuración de una AI (Artificial Intelligence o Inteligencia Artificial en español) que funciona como un glosario de palabras y/u oraciones con las cuales se alimenta al Chatbot y aumenta su entendimiento para cuando el usuario tiene la libertad de interactuar en texto, este al igual que los bloques se pueden organizar en grupos de acciones, donde cada acción contiene las frases, palabras, expresiones, etc. En texto colocado por el creador y la respuesta a ese texto en particular, este nodo de inteligencia artificial que brinda la herramienta realmente es una AI dedicada al procesamiento de lenguaje natural la cual toma estas acciones como un glosario para poder detectar textos similares a los del glosario y brindar la respuesta definida anteriormente.

Por ejemplo, si se meten a una acción frases como “gracias”, “muchas gracias” y a ello se le asigna la respuesta de responder con el texto “de nada” entonces la AI configura esa acción de tal forma que si el usuario escribe algo como “Oye muchas gracias, Chatbot” la IA detecta que pertenece a la acción definida anteriormente y responderá “de nada” a pesar de que esa frase no se definió en un inicio.

Por último, cabe mencionar que este componente permite que las respuestas sean un texto definido, una serie de textos definidos que se activarán de manera aleatoria ya sea una o la otra y/o un bloque definido o una serie de bloques de forma aleatoria.

4.9.2.6 Extras

Esta sección no muestra componentes funcionales de la plataforma sino más sobre partes para la configuración y gestión del Chatbot en general siendo estos los siguientes:

- **Conectividad:** Sirve para desconectar el Chatbot de la página de Facebook o conectarlo con alguna otra página.
- **Sube a PRO:** ventana que permite contratar el servicio PRO de Chatfuel con más funcionalidades en cuanto a métricas y atención al cliente.

- Menú persistente, funciona para colocar un menú dentro del Chatbot con el cual el usuario podrá interactuar todo el tiempo.
- Zona horaria: funciona para definir la zona horaria en la cual estará ubicado el Chatbot.
- Pagos: funciona para activar un wizard con el cual podremos configurar métodos de pago para que se realicen compras dentro del Chatbot.
- Extensiones: funciona para definir un link el cual servirá como extensión al Chatbot para que los usuarios desde el perfil del Chatbot puedan acceder a otro contenido.
- Equipo: funciona para gestionar qué perfiles de Facebook tienen acceso y de qué tipo al Chatbot ya sea en su configuración o en la plantilla de conversación.
- Token para broadcasting: Es un token que funciona para crear opciones externas que tengan que ver con broadcast.
- Lista blanca de dominios: Funciona para indicar qué dominios son seguros de acceder desde el Chatbot, esto sirve por ejemplo para ciertas extensiones integrables al Chatbot donde se realizan redireccionamientos automáticos, estos necesitan validar que son dominios seguros permitidos y asignados por el creador del Chatbot.
- Url a extensiones de Messenger: funciona como una lista donde se declaran las URLs de las extensiones que tenga Messenger con respecto al Chatbot.
- Plugin para web: funciona para poder integrar un diálogo hacia tu Chatbot por medio de Messenger, pero incrustado en una página web cualquiera.
- Adquiere usuarios por comentarios: funciona para habilitar el plugin que permite iniciar un chat desde los comentarios en las publicaciones.
- Landing Page: crear una página web que redirija al Chatbot.
- Adquiere usuarios desde tu página web: habilito plugin para adquirir conversaciones desde tu página web.
- Documentación: redirige a la documentación de Chatfuel al respecto de crecimiento y adquisición de usuarios.
- Análisis: funciona como un tablero donde se muestra estadística básica sobre la usabilidad del Chatbot ya sea, actividad de usuarios, usuarios bloqueados, total de usuarios, fuentes, retención, urls populares, entradas no reconocidas, bloques populares, botones populares, herramientas de analítica integradas.

4.9.3 Estructura de la conversación para Metsbot en Chatfuel.

Este componente de Metsbot al estar creado en Chatfuel, como se viene explicando, este compuesto en una estructura a como la plataforma lo permite, es decir, que Metsbot por este lado contiene todo su diagrama de flujo conversacional mostrado en la figura 7, desglosado a como la herramienta lo permite según los componentes que tiene vistos en la sección 4.9.2, lo cual se puede ver seccionado en bloques y grupos. Lo cual se visualiza a continuación en la figura 11i:

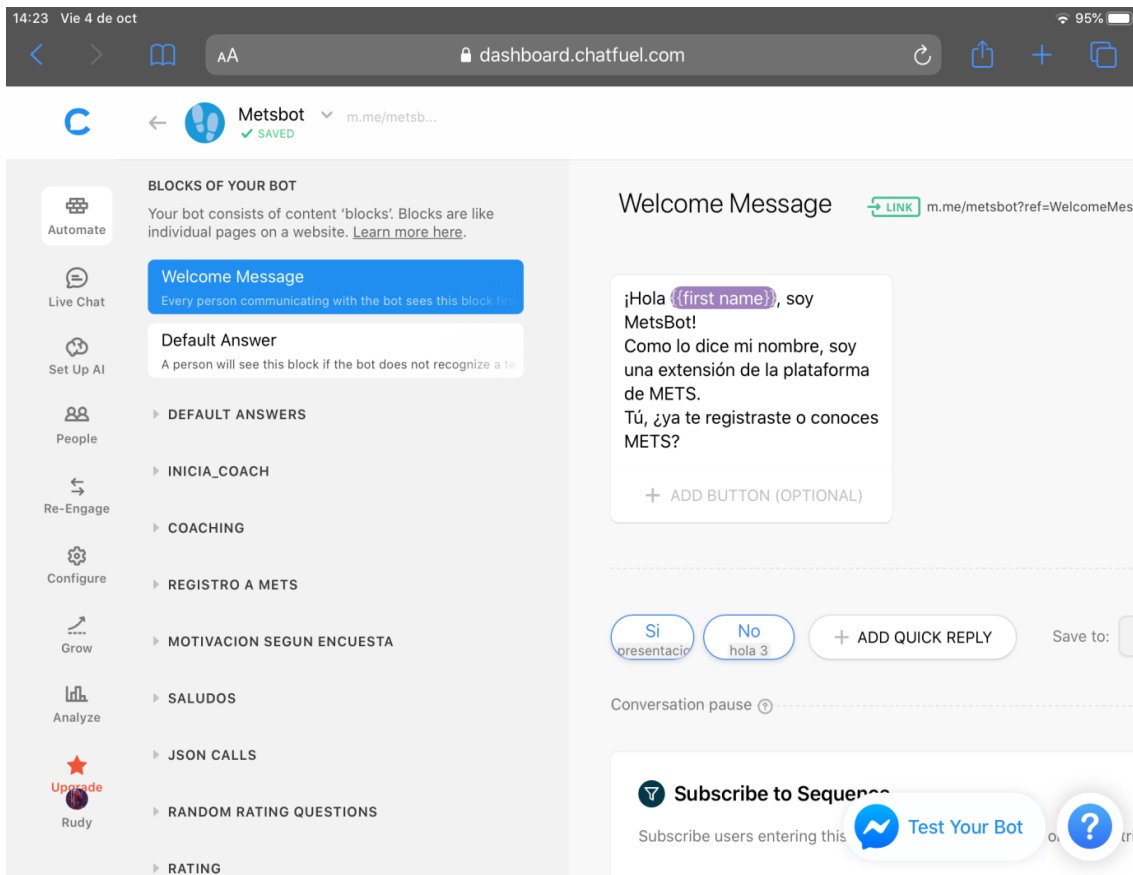


Figura 11i. Ventana que muestra como se ve el diagrama de la figura 7 desglosado en grupos dentro de Chatfuel.

Como podemos ver la estructura empírica de Metsbot ya se encuentra dividido en grupos de acuerdo a las etapas de la conversación con el Chatbot en Chatfuel. Cada grupo contiene sus respectivos bloques quienes a su vez tienen sus respectivas funciones y atributos para poder crear una conversación con el usuario además de mantener esa comunicación con el mismo, ahora entraremos en detalle sobre cada Grupo y Bloque que contiene nuestro Chatbot de Chatfuel.

4.9.3.1 Grupo de Inicio

Este grupo contiene las dos funcionalidades iniciales del Chatbot, el inicio de la conversación y cuando el Chatbot no entiende qué quiso decir el usuario los cuales se muestran en la figura 11j.

Se nombra como bloque de inicio porque la primera vez que un usuario interactúa con el Chatbot la conversación siempre debe ser iniciada por el usuario, por otra parte, siempre se debe conocer las limitantes en cuanto a la conversación que el Chatbot tiene permitido, es por ello que al inicio también se define como va a reaccionar cuando este no entienda una entrada del usuario.

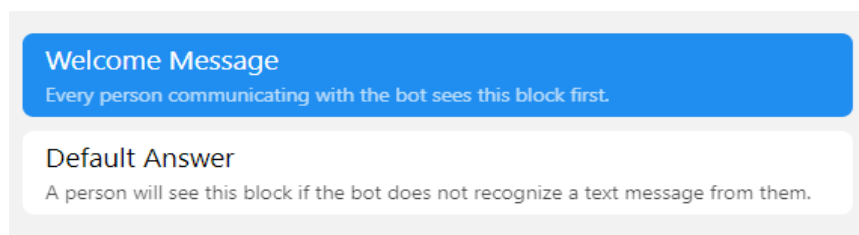


Figura 11j. Ventana que muestra como se ve el contenido del grupo de inicio.

Welcome Message es el bloque que contiene el texto inicial que dará paso al resto de la conversación por lo cual en Chatfuel este siempre se mostrara de color azul a comparación del resto de bloques los cuales su color por defecto es blanco y este no se puede cambiar. El bloque de Welcome Message será activado cuando el usuario abra un chat con Metsbot y de click en el botón de “Empezar” como se muestra en la figura 11k.

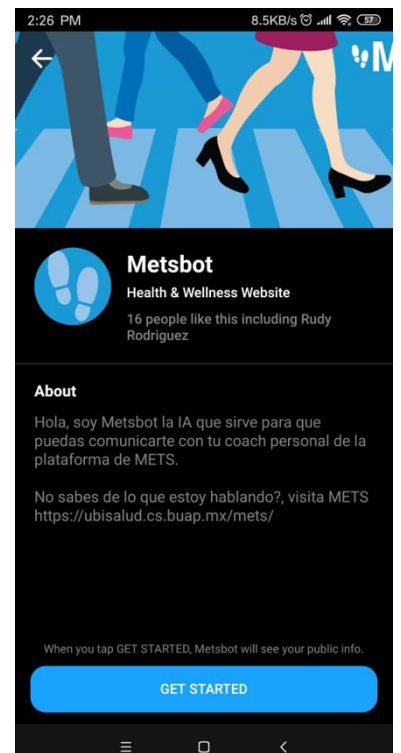


Figura 11k. Captura de pantalla de un celular con idioma por defecto en inglés para ya que el texto de ese botón inicial lo define Facebook según el idioma del teléfono.

También contiene dos Quickreply que guiarán al usuario a través de diferentes flujos activando otros bloques como se muestra en la imagen 11l. Por último, independientemente de la respuesta que elija el usuario se crea una suscripción a una secuencia la cual detonará el bloque de nombre ENCUESTA a los 30 días posteriores que el usuario inició este bloque de bienvenida.

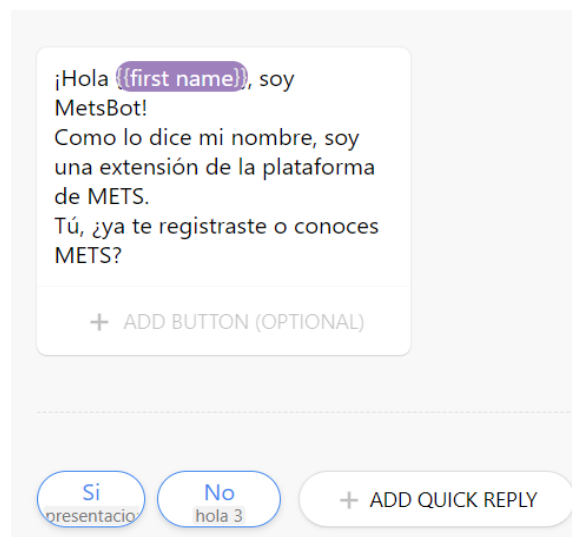


Figura 11l. Ventana con el contenido del bloque Welcome Message.

Default Answer mostrado en la figura 11m es el bloque que activará a través de sus funciones el grupo llamado “Default answers” por mientras este bloque va a repetir la última entrada del usuario y agrega un

signo de interrogación para indicarle al usuario que fue lo que no le entendió, posteriormente creará una variable interna de Chatfuel llamada “no entiende” y la incrementa en 1 según las veces que el bloque sea activado.

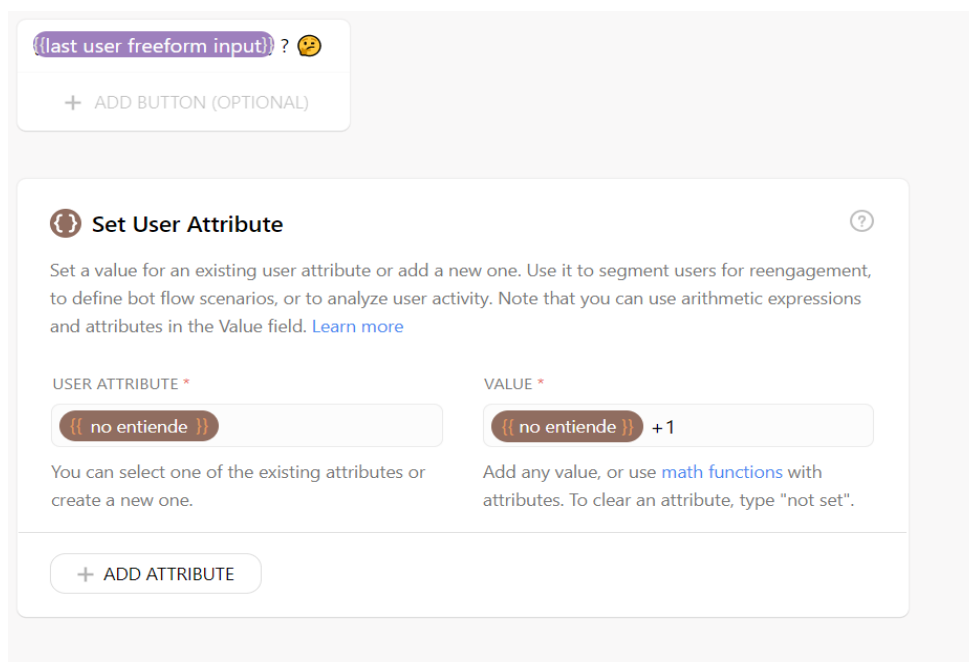


Figura 11m. Ventana con el contenido del bloque Default Answer.

Esto con la finalidad de que cuando el contador sea mayor a dos, quiere decir que de plano el usuario y el Chatbot no se han logrado entender la petición del usuario y decide enviarlo a un bloque donde se tomara ese caso, si el contador no es mayor a dos simplemente será enviado de manera aleatoria a alguno de los dos bloques existentes con respuestas para indicarle al usuario que el Chatbot no ha entendido y que podría hacer este como se muestra en la figura 11n.

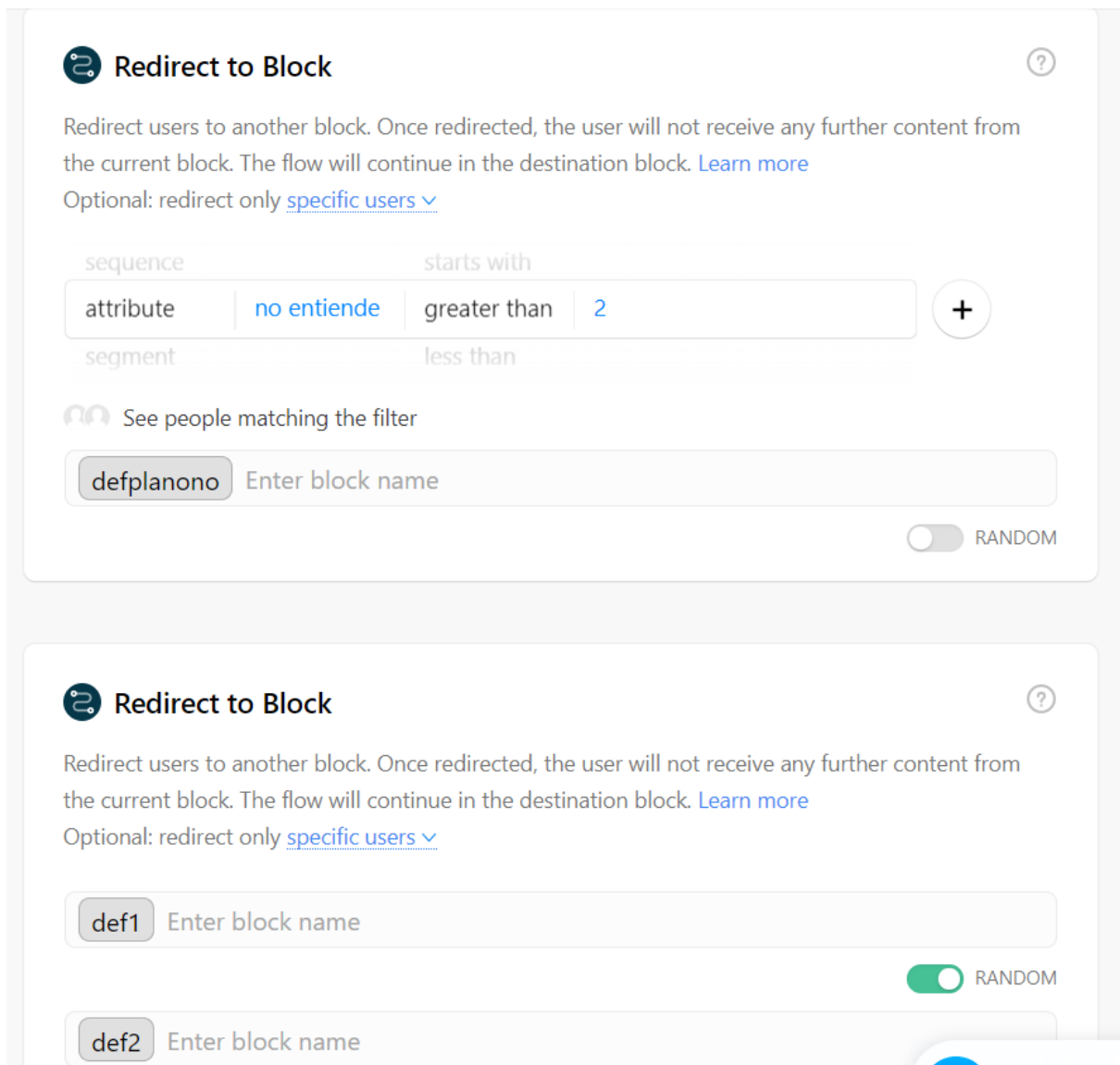


Figura 11n. Ventana con el contenido del bloque Default Answer.

4.9.3.2 Grupo “DEFAULT ANSWERS”

Este grupo mostrado en la figura 11ñ, contiene los bloques a ciertas peticiones concretas que puede tener el usuario como por ejemplo los textos para cuando el Chatbot no entiende la entrada del usuario, inicios de conversación que puedan ser activados no desde el botón porque existe una conversación, una petición no completa, e incluso para guardar valores de entradas como stickers la cual es una función que solo es accesible para los administradores. El detalle de los bloques se muestra en la tabla 10.

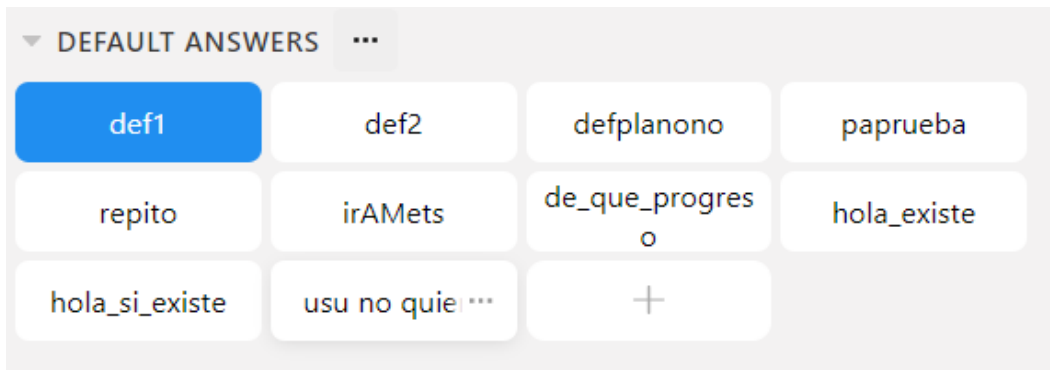


Figura 11ñ. Ventana con el contenido del bloque Default Answer.

Bloque con su descripción – imagen del bloque.

def1: es uno de los bloques que se pueden activar de forma aleatoria para brindar al usuario la respuesta de que el Chatbot no ha entendido lo que quiere decir, compuesto de una imagen, un texto con su nombre, un tiempo de espera y otro texto con una recomendación.



Ay, eso no lo entendí ((first name))
 + ADD BUTTON (OPTIONAL)

Show "typing..." for at least
 1.3 sec ————— 20 sec

Intenta decírmelo de otra forma

def2: es el otro de los bloques que se pueden activar de forma aleatoria para brindar al usuario la respuesta de que el Chatbot no ha entendido lo que quiere decir, compuesto de una imagen, un texto con su nombre, un tiempo de espera y otro texto con una recomendación.

({first name}), no he captado eso.

+ ADD BUTTON (OPTIONAL)

Show "typing..." for at least

1.3 sec 20 sec

Usa palabras cortas o que tengan relación con lo que estábamos hablando, evita respuestas abiertas.

+ ADD BUTTON (OPTIONAL)

defplanono: es un bloque que se muestra al usuario cuando ya el número de intentos es igual a 3 y se da por entendido que hay un problema de comunicación con el usuario por ende se reinicia este contador de intentos, se le menciona al usuario que no le entiende el Chatbot y se le invita a acudir a Mets en la sección de contacto por si necesita reportar algún error o duda que sale de la funcionalidad del Chatbot.

({first name}), sigo sin entender a qué te refieres

+ ADD BUTTON (OPTIONAL)

Show "typing..." for at least

1.8 sec 20 sec

En la página de mets hay más información y también nos puedes contactar por correo.

+ ADD BUTTON (OPTIONAL)

paprueba: es un bloque de prueba que sirve para registrar acciones no en texto que puede realizar un usuario en una ventana de chat de Facebook como dar un like, un sticker, etc.

paprueba [LINK](#)

Save User Input

Ask bot users questions and save their responses to user attributes. You can then utilize attributes in broadcasting user filters, Redirect to Block plugin or export them via Notify Admin via Email or JSON API plugins. [Learn more](#)

MESSAGE TO USER *

oprime hands up

VALIDATION

None

SAVE ANSWER TO ATTRIBUTE *

{handup}

+ ADD FIELDS

Modify Default Plugin Phrases >

{handup}

+ ADD BUTTON (OPTIONAL)

repito: es un bloque que sirve para crear respuestas repetidas, es decir que depende del caso donde queramos que el Chatbot repita lo ultimo que escribió el usuario.

repito [LINK](#)

{last user freeform input}

+ ADD BUTTON (OPTIONAL)

irAMets: es un bloque compuesto por una tarjeta donde se le llama por su nombre al usuario con la indicación de acceder a Mets, este bloque funciona como respuesta cuando hubo algún error con la API ya que no esta encontrando al usuario o su información y por ende le pide ir a la página de Mets para que esta información exista.

irAMets [LINK](#)



{first name} da click en el botón de **IR A BITÁCORA e insíbete a un RETO**

Subtitle or description

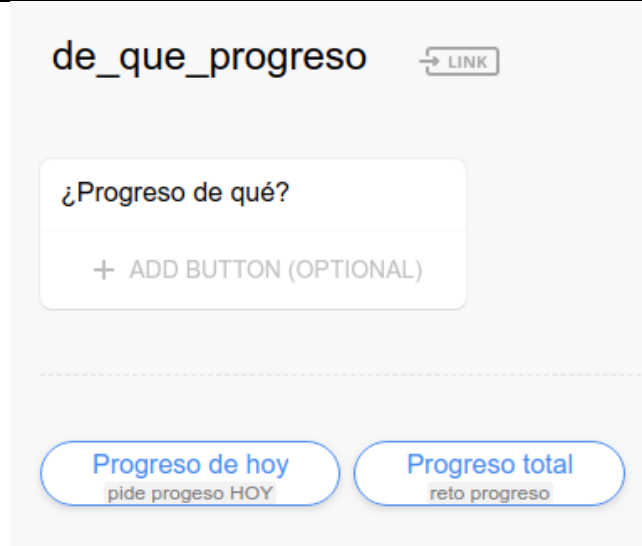
URL

IR A BITACORA

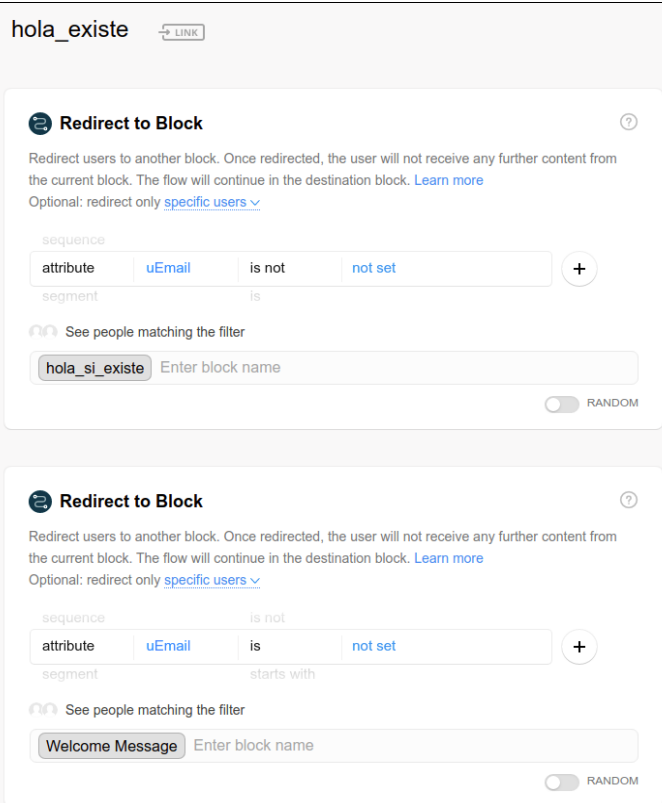
<https://ubisalud.cs.buap.mx/mets/inde1.p...>

+ ADD BUTTON (OPTIONAL)

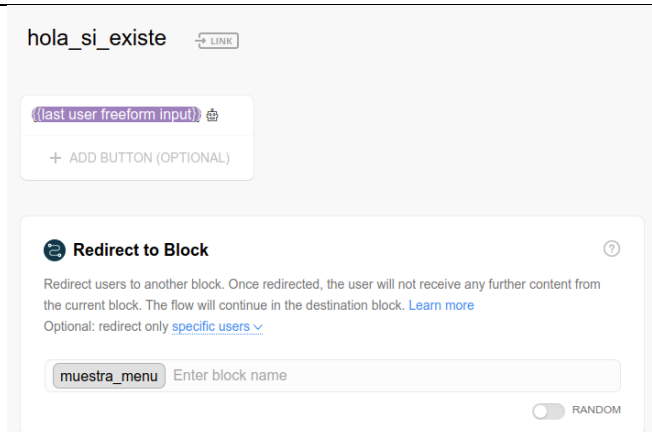
de_que_progreso: es un bloque que se activa cuando el usuario de forma escrita le indica al bot que “quiere ver el progreso” sin embargo como el contexto puede referirse a dos tipos diferentes de progreso ya sea el progreso total del reto o el progreso del día actual en su defecto, entonces se le pregunta al usuario a qué progreso se refiere y este indica cual con alguna de las dos posibles opciones con un botón de quick reply.



hola_existe: es un bloque que se activa cuando el usuario saluda al Chatbot de forma textual pero en el escenario de que pudo o no registrarse en Mets por ende la función de este bloque reside checar una variable llamada uEmail que corresponde a saber si ese atributo ya contiene el email de la persona o no, de ser así entonces ya es un usuario de Mets y del Chatbot por ende se le redirige al bloque correspondiente llamado hola_si_existe de lo contrario se reinicia la conversación con el Chatbot redirigiendo al usuario al bloque de Welcome Message.



hola_si_existe: es un bloque que se activó porque el correo del usuario ya está en Mets como en la API del Chatbot por ende es un usuario ya registrado, entonces solo se repite el saludo del usuario y se le redirige al menú.



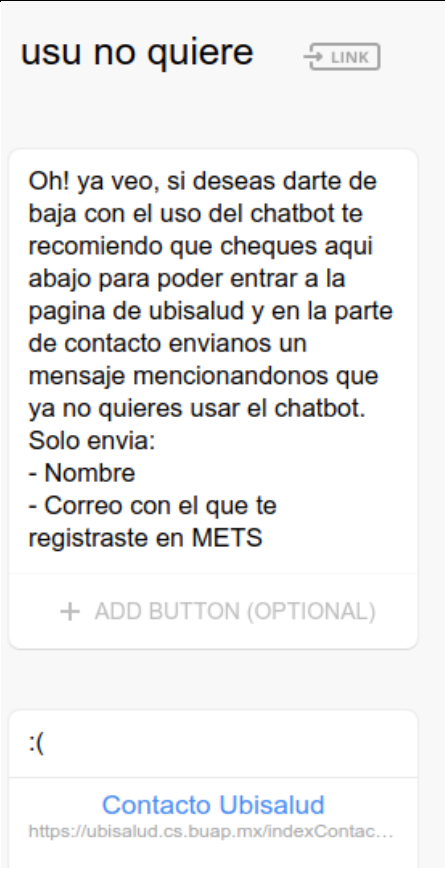
<p>usu no quiere: es un bloque con texto donde se le menciona al usuario como puede contactarse con ubisalud para enviar su solicitud de darse de baja de la plataforma completamente, app y Chatbot.</p>	
--	---

Tabla 10. Tabla con el contenido de cada bloque del grupo Default Answers.

4.9.3.3 Grupo SALUDOS

El grupo de bloques llamado “SALUDOS” mostrado en la figura 110, contiene una serie de bloques que corresponden a la etapa de brindar información al usuario y de concientizarse de que puede tener un problema de sedentarismo a través de diferentes mensajes posibles para evitar que sea tan similar lo que le dice a un usuario como lo que le dice a otro. Los bloques con su funcionalidad serán descritos en la tabla 11.

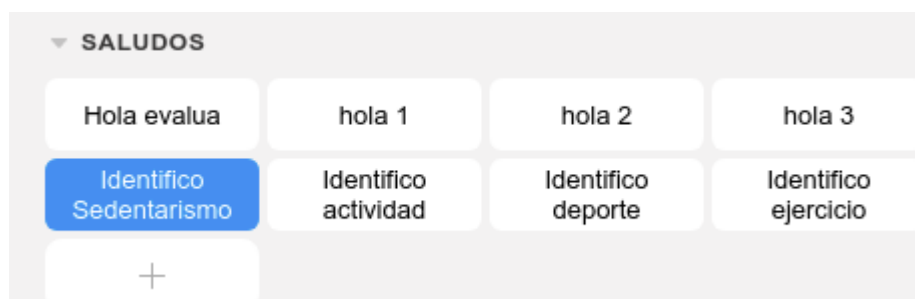


Figura 110. Ventana del grupo SALUDOS y sus bloques.

Bloque con su descripción-Imagen

Hola evalúa: es un bloque que de manera aleatoria ejecutara alguno de los 3 bloques de saludo, con mayor inclinación a que se ejecute el bloque llamado “hola 3” debido a que este es el que contiene mayor contenido e interacción.

Hola evalua [LINK](#)

Redirect to Block

Redirect users to another block. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. [Learn more](#)

Optional: redirect only [specific users](#) ▾

hola 1 Enter block name

RANDOM

hola 2 Enter block name

hola 3 Enter block name

hola 3 Enter block name

+ ADD RANDOM BLOCK

hola1: es un bloque informativo donde Metsbot se presenta, le indica su función y le brinda información al usuario para que considere continuar con la conversación y en cómo puede hacer algo con respecto al sedentarismo que ahora él conoce o solo indicar que no le interesa.

hola 1 [LINK](#)

Bien {{first name}}, déjame contarte un poco sobre mí 😊 , estoy muy interesado en el bienestar de las personas, estoy aquí para mostrarte una forma de cuidar tu salud mediante acciones muy sencillas.

+ ADD BUTTON (OPTIONAL)

Show "typing..." for at least




0.1 sec 4.3 sec 20 sec

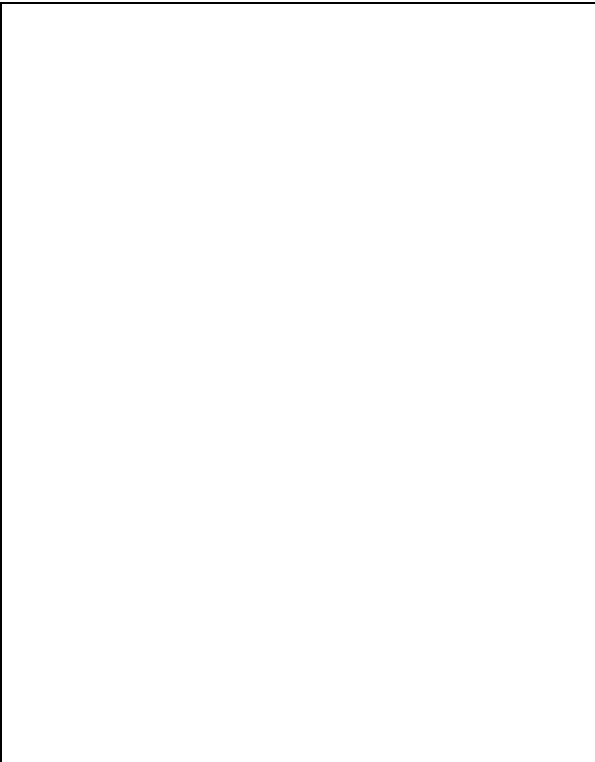
¿Sabías que la mayoría de las personas confunde la actividad física con el ejercicio?

+ ADD BUTTON (OPTIONAL)

Show "typing..." for at least

0.1 sec 4.1 sec 20 sec

	<p>Sin embargo, TODOS hacemos actividad física ☺ ¡Veámos unos ejemplos!</p> <p>+ ADD BUTTON (OPTIONAL)</p> <p>... Show "typing..." for at least 0.1 sec 9.9 sec 20 sec</p>  <p>... Show "typing..." for at least 3.1 sec 20 sec</p>
	 <p>... Show "typing..." for at least 3.4 sec 20 sec</p>  <p>... Show "typing..." for at least 0.1 sec 4.4 sec 20 sec</p>



📄 Actividad física es cualquier movimiento de nuestro cuerpo, ya sea natural o planificado, que produce un gasto energético. Si buscas más información, puedes entrar aquí.

[Página web](https://ubisalud.cs.buap.mx/sedentarismo/)
https://ubisalud.cs.buap.mx/sedentarismo/

+ ADD BUTTON (OPTIONAL)

... Show "typing..." for at least
0.1 sec 6.8 sec 20 sec

Incrementar la actividad física sólo depende de una decisión y esa decisión la tienes TÚ, {{last name}} 🙌 !!
¿Te interesa?

+ ADD BUTTON (OPTIONAL)

Claro 😊 SI_INTERESA No, gracias No_gracias_pregunta + ADD QUICK REPLY

hola 2: es un bloque informativo donde de forma resumida se le indica algo de información al usuario e inmediatamente se le invita a continuar con la conversación o tratar de dejar el flujo antes del registro.

hola 2 → LINK

Bien {{first name}} 😊, estoy aquí para mostrarte una forma de cuidar tu salud mediante acciones muy sencillas.


+ ADD BUTTON (OPTIONAL)

... Show "typing..." for at least
3.8 sec 20 sec

¿Sabías que la mayoría de las personas confunde la actividad física con el ejercicio?

+ ADD BUTTON (OPTIONAL)

... Show "typing..." for at least
0.1 sec 4 sec 20 sec

	<p>La actividad física es cualquier movimiento corporal producido por los músculos y que resulta en un gasto energético, por ejemplo andar en bicicleta, subir escaleras, hacer las tareas del hogar, ir de compras, y otras que forman parte de nuestra rutina cotidiana. Si buscas más información, puedes entrar aquí.</p> <p>Página web https://ubisaud.cs.buap.mx/sedentarismo/</p> <p>+ ADD BUTTON (OPTIONAL)</p>  <p>... Show "typing..." for at least 0.1 sec 6.0 sec 20 sec</p>
	<p>Incrementar la actividad física sólo depende de una decisión y esa la decisión la tienes TÚ, ((last name)) 🤖♂⚡ ¿Te interesa?</p> <p>+ ADD BUTTON (OPTIONAL)</p> <hr/> <p>Claro 😊 SI_INTERESA</p> <p>No, gracias No_gracias_pregunta</p>

hola 3: es un bloque informativo más completo donde el Chatbot se presenta, indica su funcionalidad y le pone al usuario una pequeña actividad donde él indica con qué imagen se identifica más y tras escoger una se le mostrará de forma más personal el resto de la información.

hola 3 [→ LINK](#)

Bien **{{first name}}**, déjame contarte un poco sobre mí 🤖 , estoy muy interesado en el bienestar de las personas y estoy aquí para mostrarte una forma de cuidar tu salud mediante acciones muy sencillas.






+ ADD BUTTON (OPTIONAL)



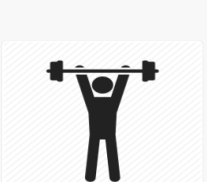
Show "typing..." for at least

0.1 sec 9.5 sec 20 sec

¿Sabías que todos hacemos actividad física? 🤖 La diferencia es la intensidad con la que se realiza ¿No me crees? 🤖 ¿Qué tal si te muestro unos ejemplos y me dices con cuál te identificas?

+ ADD BUTTON (OPTIONAL)

				
Actividad física	Deporte	Actividad física	Sedentarismo	Deporte
Subtítulo o descripción	Subtítulo o descripción	Subtítulo o descripción	Subtítulo o descripción	Subtítulo o descripción
URL	URL	URL	URL	URL
Me identifico Identifico actividad	Me identifico Identifico deporte	Me identifico Identifico actividad	Me identifico Identifico Sedentarismo	Me identifico Identifico deporte
+ ADD BUTTON (OPTIONAL)	+ ADD BUTTON (OPTIONAL)	+ ADD BUTTON (OPTIONAL)	+ ADD BUTTON (OPTIONAL)	+ ADD BUTTON (OPTIONAL)

		
Deporte	Sedentarismo	Ejercicio físico
Subtítulo o descripción	Subtítulo o descripción	Subtítulo o descripción
URL	URL	URL
Me identifico Identifico deporte	Me identifico Identifico Sedentarismo	Me identifico Identifico ejercicio
+ ADD BUTTON (OPTIONAL)	+ ADD BUTTON (OPTIONAL)	+ ADD BUTTON (OPTIONAL)

Identificó Sedentarismo: es un bloque que activa cuando el usuario indica que se identifica con alguna de las actividades que corresponden a una acción sedentaria, de esta forma el Chatbot le hace saber a qué se refiere el sedentarismo y le alerta que eso resulta ser problemático, pero le presenta la posibilidad de continuar la conversación para ayudar o indicar que no está interesado.

Identifico Sedentarismo

🗣️ 📌 El sedentarismo son aquellas actividades caracterizadas por estar sentado o en posición recostada y que no requieren un gasto de energía. Si buscas más información, puedes entrar aquí.

[Página web](https://ubisalud.cs.buap.mx/sedentarismo/)
<https://ubisalud.cs.buap.mx/sedentarismo/>

+ ADD BUTTON (OPTIONAL)

Es urgente que incrementes tu actividad física, y eso depende de una decisión, y esa la decisión la tienes TÚ, {{last name}} 🗣️ 📌
 ¿Te interesa?

+ ADD BUTTON (OPTIONAL)

[claro](#) [No gracias](#)
 SI_INTERESA No_gracias_pregunta

Identificó actividad: es un bloque que se activa cuando el usuario elige una de las imágenes que corresponden a actividad física, de esta manera le menciona al usuario a que se refiere y le hace una invitación a poder incrementar su actividad física a sabiendas de que el sedentarismo puede ser problemático, así como le brinda una opción para tratar de no continuar con la conversación.

Identifico actividad [LINK](#)

🗣️ 📌 Actividad física es cualquier movimiento de nuestro cuerpo, ya sea natural o planificado, que produce un gasto energético. Si buscas más información, puedes entrar aquí.

[Página web](https://ubisalud.cs.buap.mx/sedentarismo/)
<https://ubisalud.cs.buap.mx/sedentarismo/>

+ ADD BUTTON (OPTIONAL)

Incrementar la actividad física sólo depende de una decisión y esa la decisión la tienes TÚ, {{last name}} 🗣️ 📌
 ¿Te interesa?

+ ADD BUTTON (OPTIONAL)

[claro](#) [No gracias](#)
 SI_INTERESA No_gracias_pregunta

Identificó deporte: es un bloque que activa cuando el usuario indica que se identifica con alguna de las actividades que corresponden a una acción de deporte es decir, que el usuario realiza algún deporte y es consciente de ello como actividad física, por ende se le invita al usuario a que pueda incrementar su ritmo de actividad física para evitar lapsos de sedentarismo cuando no está practicando algún deporte, el convencimiento existe al decirle a la persona que se le será más fácil cumplir con el plan de Mets debido a que ya es una persona que practica alguna disciplina deportiva.

Identifico deporte [LINK](#)

📌 El deporte implica movimiento corporal planificado y además involucra reglas establecidas que deben cumplirse dentro del contexto o área específica (campo de juego, cancha, etc.) y, frecuentemente, consiste en competir. Si buscas más información, puedes entrar aquí.

[Página web](https://ubisalud.cs.buap.mx/sedentarismo/)
https://ubisalud.cs.buap.mx/sedentarismo/

+ ADD BUTTON (OPTIONAL)

Quiere decir, que eres una persona disciplinada y te será muy sencillo incrementar la actividad física durante el día, sólo depende de una decisión y esa la decisión la tienes TÚ, [\(last name\)](#) 📌
¿Te interesa?

+ ADD BUTTON (OPTIONAL)

[claro](#) [No gracias](#)
SI_INTERESA No_gracias_pregunta

Identificó ejercicio: es un bloque que activa cuando el usuario indica que se identifica con alguna de las actividades que corresponden a una acción de ejercicio. Por ende, solo se le explica al usuario a que se refiere ese tipo de actividad y se le hace una ligera invitación a que continúe con el Chatbot debido a que probablemente haga ejercicio este tipo de persona y sus lapsos de sedentarismo no presentan un riesgo real para su salud.

Tabla 11. Tabla con el contenido de cada bloque del grupo Default Saludos.

4.9.3.4 Grupo MOTIVACIÓN SEGÚN ENCUESTA

El grupo llamado “Motivación según encuesta” mostrado en la figura 11p, es un grupo de bloques que sirven de respuesta es decir que son accionados por la encuesta del grupo anterior donde se retoma al usuario para que a través de una situación donde se utiliza la terapia cognitivo conductual después de una terapia aceptación de las precauciones, esto en el caso de que el usuario no se muestre convencido de continuar, de lo contrario solo procederemos con el registro. El detalle de cada bloque se especifica en la tabla 12.

Figura 11p. Ventana del grupo Motivación según encuesta con sus bloques.

Bloque con su descripción-Imagen

SI_INTERESA: es un bloque donde se da por entendido que al usuario le interesa continuar con Mets y la conversación con Metsbot por ende le proporciona una breve explicación de qué va Mets y luego se introduce la funcionalidad del Chatbot como coach personal para finalmente proporcionar las opciones de dar más información, registrarse o de plano salir de la interacción.

The screenshot shows the configuration for the 'SI_INTERESA' block. It includes a text message: 'Hemos creado una aplicación para que cumplas el reto de los 10,000 pasos diarios de la OMS, se llama METS y es súper fácil de usar.' Below the message is a 'Show "typing..." for at least' slider set to 1.9 seconds. The block also contains three buttons: 'Dame más info' (with label 'Quiere_Info'), 'Perfecto, regístrate' (with label 'registro_a_mets'), and 'Mejor no' (with label 'No_gracias_pregunta').

No_gracias_pregunta: es un bloque diseñado con las preguntas más comunes para decir que no a la actividad física esto basado en la encuesta de motivos que se realizó anteriormente y se muestra más adelante. Tomando en cuenta estos casos comunes se consultó con la psicóloga para no salir de la terapia cognitivo conductual sino retomarla y buscar una posible contraoferta a través del motivo/excusa que la persona presentó.

A continuación, se mostrarán estas respuestas (motivos/excusas) de los usuarios y cómo se retoma.

The screenshot shows the configuration for the 'No_gracias_pregunta' block. It includes a 'Show "typing..." for at least' slider set to 2.2 seconds. The text message reads: 'No hay problema ((first name)), sólo me gustaría saber ¿por qué?'. Below the message is an 'ADD BUTTON (OPTIONAL)' option. At the bottom, there are four buttons representing different reasons: 'No tengo tiempo' (label 'No_tiempo_rply'), 'Estoy cansado(a)' (label 'cansado_rply'), 'No tengo ganas' (label 'No_ganas_rply'), and 'Es costoso' (label 'costoso_rply').

cansado_reply: bloque que selecciona el usuario cuando indica que es por cansancio por el cual no está interesado.

cansado_rply [LINK](#)

Show "typing..." for at least
1.9 sec 20 sec

Yo también me la vivía cansado de la vida 😞, pero cuando realizas actividad física 🏃‍♂️ incrementas el flujo de sangre a tu cerebro 🧠 y tu cuerpo libera endorfinas que te provocan sensaciones de felicidad, ☺ serenidad y creatividad. Con METS para lograr esto solo debes caminar y listo. 🚶‍♂️🏃‍♂️

[Ok, dame más info](#)
Quiere_info

[Muy bien, regístrate](#)
registro_a_mets

+ ADD BUTTON (OPTIONAL)

costoso_rply: bloque que selecciona el usuario cuando indica que es por la idea de que resulta “costosa” económicamente hablando el realizar actividad física, aunque principalmente esto es así ya que se tiene un concepto de que realizar actividad física es tener una rutina de ejercicio en el gimnasio.

costoso_rply [LINK](#)

Show "typing..." for at least
1.8 sec 20 sec

Ir al Gym 🏋️‍♂️, tener ropa deportiva, pagar un entrenador, en efecto, es muy costoso 💸💸💸

Pero, METS es gratis 😊 y solo necesitas caminar 🚶‍♂️, la app registra tu recorrido diario y con eso sumas pasos para lograr las metas diarias, de tal forma que tu salud mejorará gradualmente.

[Dame más info](#)
Quiere_info

[Wow, ¡Regístrate!](#)
registro_a_mets

+ ADD BUTTON (OPTIONAL)

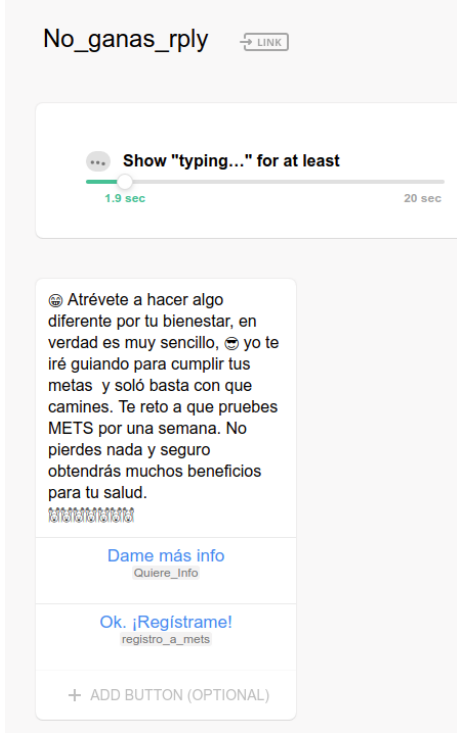
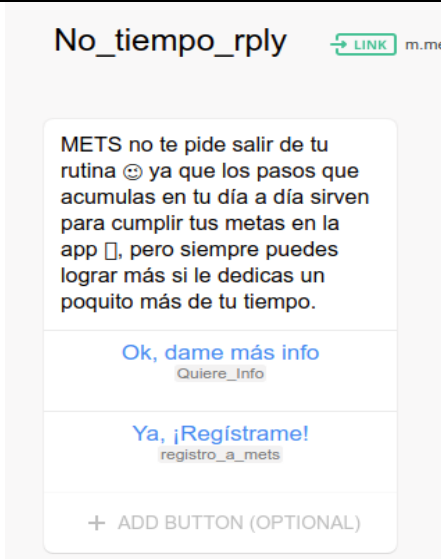
<p>No_ganas_rply: bloque que selecciona el usuario por mencionar que “no tiene ganas” de realizar actividad física.</p>	
<p>No_tiempo_rply: bloque que selecciona el usuario cuando indica que es por la falta de tiempo que no realiza actividad física.</p>	

Tabla 12. Tabla con el contenido de cada bloque del grupo Motivación según encuesta.

4.9.3.5 Grupo REGISTRO A METS

Como se pudo ver en el anterior bloque se trata de retomar al usuario, de convencerlo de continuar con la conversación activando el bloque que indica que si está interesado, este lo mandara a este grupo de bloques mostrados en la figura 11q donde procedemos a indicarle al usuario que se registre en Mets y con base en ello poder utilizar al Chatbot como coach que sería un homólogo de registrarse en Metsbot. Los bloques con su funcionalidad serán descritos en la tabla 13.

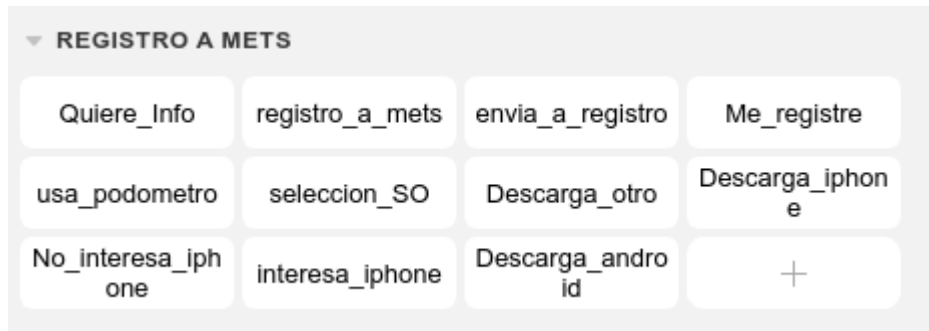


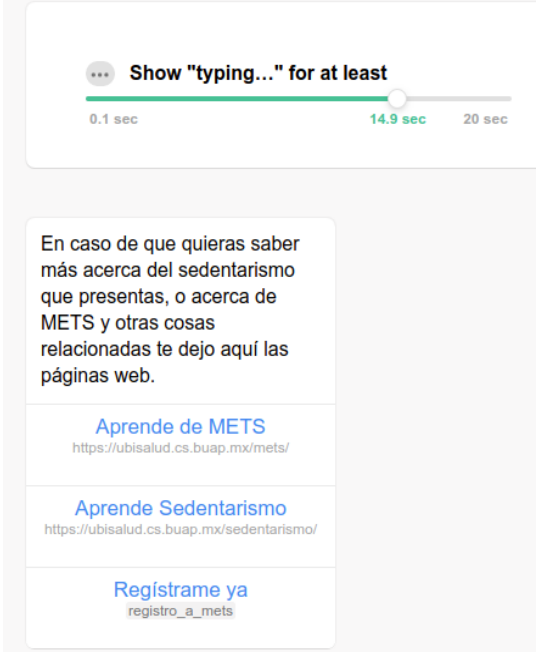
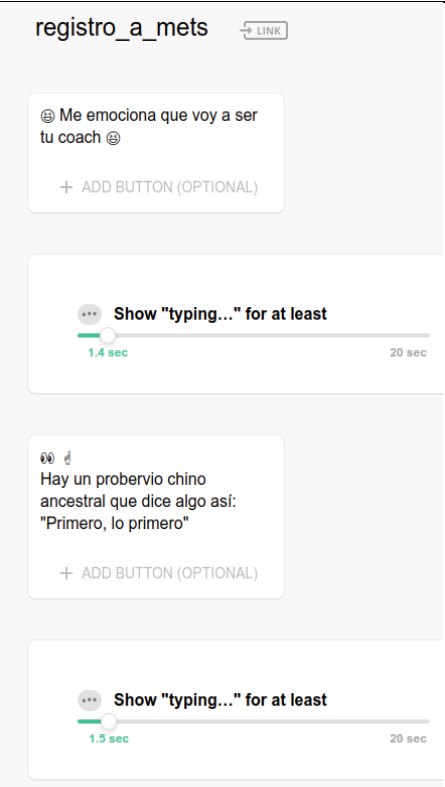
Figura 11q. Ventana del grupo Registro a METS con sus bloques.

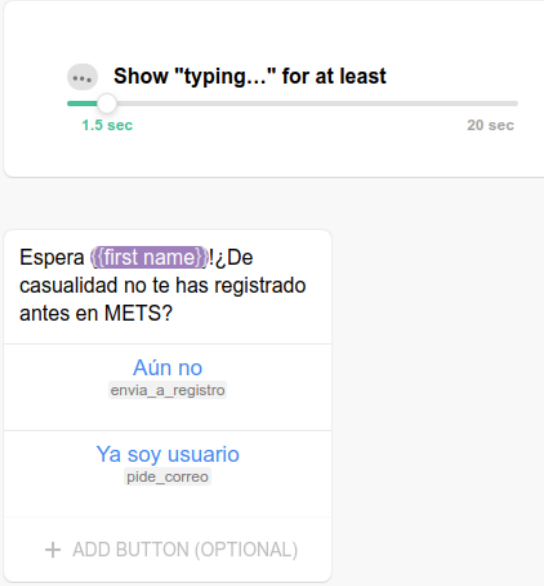
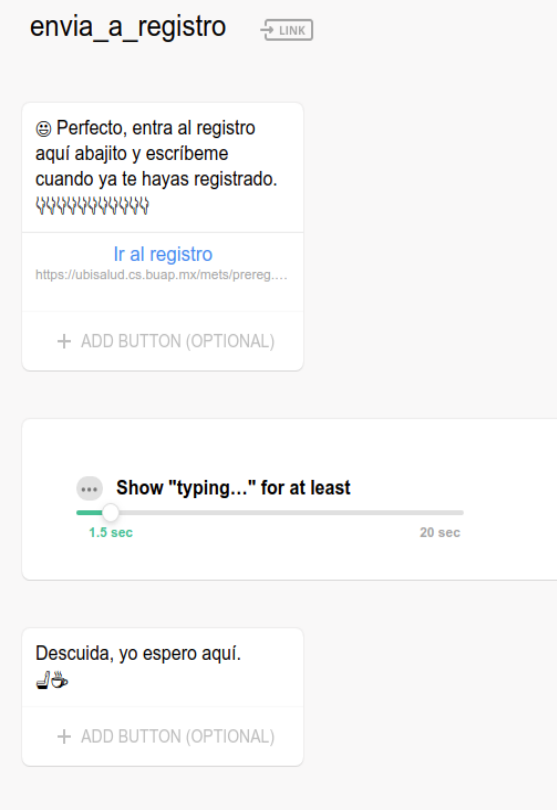
Bloque con su descripción-Imagen

Quiere_Info: parte de las acciones para retomar al usuario y que este se registre es en el caso cuando el usuario está interesado(a) en conocer más acerca de la plataforma de METS o incluso sobre el sedentarismo lo cual es información que se le ha estado presentando pero siempre puede proporcionarse más información por lo cual se le da un último resumen y se le proporciona los links de la página de Ubisalud donde tendrá más a detalle esa información evitando tener mensajes de texto muy extensos entre el usuario y el Chatbot, a demás de que al presentarse en forma de botones este puede fácilmente acceder a los links, regresar y tener presente la opción de ya ir al bloque que inicia el registro.

The screenshot displays a chatbot interface for a block titled "Quiere_Info". At the top right, there is a "LINK" button. The interface contains three main components:

- Typing Indicator 1:** A bubble with three dots and the text "Show 'typing...' for at least". A progress bar below it shows a green segment for "0.1 sec" and a grey segment for "4 sec".
- Text Block:** A rounded rectangle containing the text "Muy bien, a grandes rasgos" and a button labeled "+ ADD BUTTON (OPTIONAL)".
- Typing Indicator 2:** A bubble with three dots and the text "Show 'typing...' for at least". A progress bar below it shows a green segment for "2.3 sec".
- Text Block:** A rounded rectangle containing a paragraph of text: "🌐 METS es una plataforma web y también una app de celular con la cuál podremos contar los pasos que das en tu rutina diaria, y con metas sucesivas logres cumplir el reto de los 10000 pasos de la OMS, es más fácil de lo que parece. Simple, tu caminas, nosotros registramos, medimos tu progreso y cuando menos lo esperes tu salud estará mejorando gracias a la actividad física. :o".

	 <p>The screenshot shows a chatbot interface. At the top, there is a typing indicator with the text "Show 'typing...' for at least" and a progress bar showing 0.1 sec, 14.9 sec, and 20 sec. Below this, there is a text block: "En caso de que quieras saber más acerca del sedentarismo que presentas, o acerca de METS y otras cosas relacionadas te dejo aquí las páginas web." Underneath the text are three buttons: "Aprende de METS" with the URL "https://ubisalud.cs.buap.mx/mets/", "Aprende Sedentarismo" with the URL "https://ubisalud.cs.buap.mx/sedentarismo/", and "Regístrate ya" with the URL "registro_a_mets".</p>
<p>registro_a_mets: este bloque inicia el proceso de registrar al usuario y contiene dos casos:</p> <ol style="list-style-type: none"> 1) El usuario ya se ha registrado anteriormente en la plataforma y desea utilizar el Metsbot. 2) El usuario es nuevo, tanto para la plataforma de Mets como en Metsbot. 	 <p>The screenshot shows a chatbot interface titled "registro_a_mets" with a "LINK" icon. It contains three main blocks: <ul style="list-style-type: none"> A user message: "Me emociona que voy a ser tu coach" with a plus sign and "ADD BUTTON (OPTIONAL)". A typing indicator: "Show 'typing...' for at least" with a progress bar showing 1.4 sec and 20 sec. A text block: "Hay un proverbio chino ancestral que dice algo así: 'Primero, lo primero'" with a plus sign and "ADD BUTTON (OPTIONAL)". </p>

	
<p>envia_a_registro: es el bloque que se activa cuando el usuario indica que es nuevo tanto para Mets como para Metsbot entonces se le proporciona el link que lo dirige exactamente a la página de registro de Mets y el Chatbot queda en espera de que el usuario menciona que ya se registró.</p>	

Me_registre: este bloque es activado mediante texto en la conversación del usuario, está ligado al acervo de expresiones relacionadas a la respuesta que dejó abierta el Chatbot donde indica que se le avise cuando haya terminado de registrarse, por ejemplo: “Ya me registre”, esa frase activará este bloque donde le indicará posteriormente que debe tener la aplicación de Mets instalada.

Me_registre [LINK](https://m.me/metsbot?ref=Me_registre) m.me/metsbot?ref=Me_registre

... Show "typing..." for at least
3.6 sec 20 sec

Ok, eso fue algo rápido 😊
Aún no me acabo mi café ☕

+ ADD BUTTON (OPTIONAL)

... Show "typing..." for at least
2.1 sec 20 sec

{first name}
¡FELICIDADES YA ESTÁS EN METS!
¿Ya descargaste la app?

+ ADD BUTTON (OPTIONAL)

Aún no seleccion_SO Voy a usar podómetro usa_podometro Ya 😊 pide_correo

usa_podometro: al registrarse en la plataforma de Mets se da la opción de poder usar la aplicación desde web y con un podómetro, sin embargo por motivos de usabilidad que se establecieron en los requisitos de la solución Metsbot no tiene mucho sentido que sea usado si es con podómetro, por lo cual en el remoto caso de que exista este tipo de usuario existe esta opción donde se le dan unas indicaciones finales al usuario y Metsbot se despide indicando de que no le es posible funcionar con un podómetro.

usa_podometro [LINK](#)

Me parece perfecto, te recuerdo que si usas podómetro, tendrás que registrar tu progreso desde la plataforma web en: <https://ubisalud.cs.buap.mx/mets/>

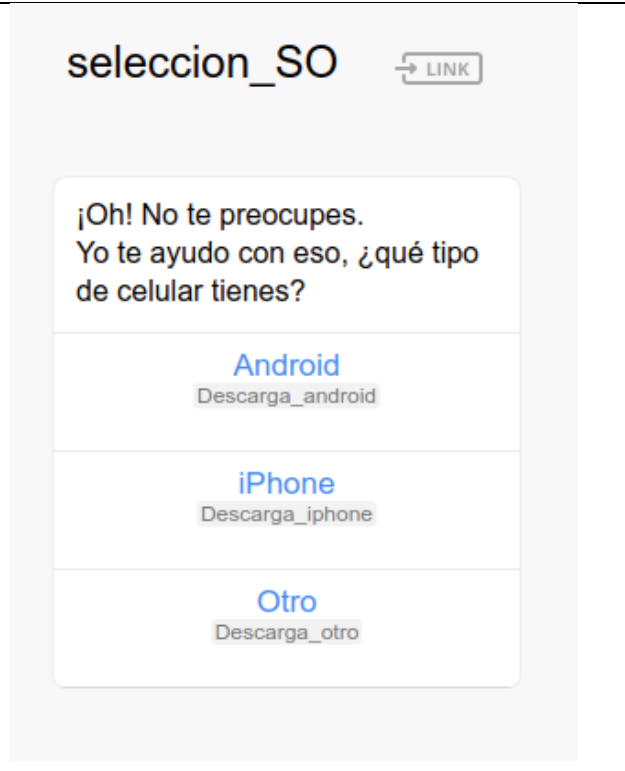
+ ADD BUTTON (OPTIONAL)

... Show "typing..." for at least
0.1 sec 4 sec 20 s

Por el momento me es muy difícil llevar un control de tu progreso si usas el podómetro, pero puedes usar la plataforma sin problemas.
☹☹☹☹
Nos vemos luego.

+ ADD BUTTON (OPTIONAL)

seleccion_SO: bloque que se activa cuando el usuario indica que aún no se ha descargado la aplicación móvil, por ende, se le preguntara cuál es el sistema operativo de su dispositivo móvil.



seleccion_SO LINK

¡Oh! No te preocupes.
Yo te ayudo con eso, ¿qué tipo de celular tienes?

Android
Descarga_android

iPhone
Descarga_iphone

Otro
Descarga_otro

Descarga_otro: es la opción para usuarios que no cuentan con iPhone o un teléfono celular con Android como sistema operativo, por motivos de la plataforma hasta la fecha no existe aplicación más que para Android y otra no en tiendas para iPhone por ende se le hace conocer esta situación al usuario y se le termina redirigiendo al uso del podómetro para que quede como una invitación abierta a utilizar Mets, aunque Metsbot no pueda utilizarlo, sin embargo si existen más versiones para otros sistemas operativos móviles u opciones de adaptarlo esta secuencia de bloques puede cambiar su contenido sin mayor dificultad.



Descarga_otro LINK

☹️☹️☹️☹️☹️
¡Uy! Perdón, pero actualmente solo funcionamos con:
1) Podómetro desde la web
2) Android
3) IOS
Así que si gustas puedes usar METS desde la plataforma web con un podómetro.

Ok, usaré podómetro
usa_podometro

+ ADD BUTTON (OPTIONAL)

Descarga_iphone: bloque que se activa si el usuario indica que tiene un equipo móvil con iOS. A la fecha la app no se encuentra aún en la tienda oficial de aplicaciones para iOS, la Appstore, sin embargo, se sigue trabajando en ello y cuando esté disponible solo se cambia el link a donde será redirigido el usuario.

Descarga_iphone → LINK

A mí también me encanta iPhone, la verdad es mejor. Aquí abajo te dejo el enlace para que puedas descargar la app. Me avisas cuando ya la hayas instalado e iniciado sesión.

XXXXXXXXXXXXXXXXXXXX

[Descargar IOS](#)
www.appstore.com

[Ya inicié sesión :\)](#)
pide_correo

+ ADD BUTTON (OPTIONAL)

... Show "typing..." for at least

0.1 sec 8 sec 20 sec

Hey! momento, me acaban de informar que la app para iOS aun sigue en producción, pero si gustas te puedo avisar para cuando la app ya esté disponible.

[Si, avísame](#)
interesa_iphone

[No, gracias](#)
No_interesa_iphone

+ ADD BUTTON (OPTIONAL)

No_interesa_iphone: bloque que se activa por si el usuario desiste de utilizar Metsbot debido a que no puede tener la app en su dispositivo con iOS.

No_interesa_iphone → LINK

Ok, no pasa nada.
Si quieres volver a hablar conmigo sólo salúdame e iniciamos nuevamente.
Bye.

+ ADD BUTTON (OPTIONAL)

Interesa_iphone: bloque se que activa cuando el usuario está enterado de la situación con la app para su dispositivo con iOS, pero desea ser notificado cuando esto suceda, para ello se le asigna un atributo al usuario con su ID de Facebook dentro de la plataforma para que luego de forma manual se pueda filtrar a los usuarios que cuenten con ese atributo y mandarles un mensaje con el link para descargar la app.

interesa_iphone → LINK

Set User Attribute

Set a value for an existing user attribute or add a new one. Use it to segment users for reengagement, to define bot flow scenarios, or to analyze user activity. Note that you can use arithmetic expressions and attributes in the Value field. [Learn more](#)

USER ATTRIBUTE • VALUE •

{{ notificaApplos }} si

You can select one of the existing attributes or create a new one. Add any value, or use [math functions](#) with attributes. To clear an attribute, type "not set".

+ ADD ATTRIBUTE

Subscribe to Sequence

Subscribe users entering this block to a sequence based on their attributes. [Learn more](#)

sequence is not
attribute notificaApplo is si +
segment starts with

See people matching the filter

SUBSCRIBE TO • RANDOM

IPHONE NOTIFICA Required

<p>Descarga_android: bloque que indica el poder descargar la aplicación de Mets e indica al usuario que inicie sesión para que hasta entonces de click en el botón de “Ya inicié sesión” y de esta forma continuar con el flujo de la conversación.</p>	
--	---

Tabla 13. Tabla con el contenido de cada bloque del grupo Registro a METS.

4.9.3.6 Grupo INICIA_COACH

EL grupo de bloques llamado INICIA_COACH mostrado en la figura 11r, contiene los bloques con las acciones continuas e iniciales cuando el usuario ya está registrado en Mets y va a utilizar el Chatbot como su coach. Los bloques con su funcionalidad serán descritos en la tabla 14.

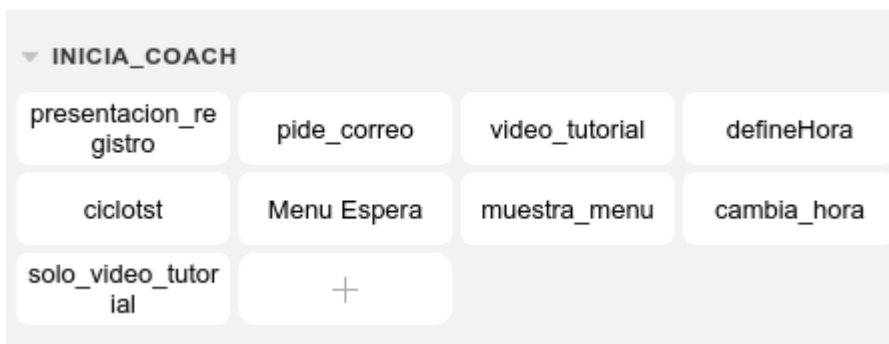


Figura 11r. Ventana del grupo INICIA_COACH con sus bloques.

Bloque con su descripción-Imagen

presentacion_registro: es un bloque que se activa cuando el usuario indica que si ya conoce de Mets que ya está registrado desde el saludo inicial y este responde sorprendido, se presenta a grandes rasgos le indica su funcionalidad igual y le redirige al bloque de pide_correo para como su nombre lo dice, pedirle su correo y validar.

presentacion_registro [LINK](#)

¡Oh! ¡Qué bien! Entonces ya nos conocemos, soy tu coach personal 😊, me elegiste cuando te registraste en METS y ahora, gracias a METSBOT 🤖 ; estamos en contacto.

+ ADD BUTTON (OPTIONAL)

... **Show "typing..." for at least**
2 sec 20 sec

Por aquí podrás pedirme tu progreso y también te estaré recordando cómo vas con tus retos para que juntos podamos lograrlos 🏆 📅 y cuando menos lo esperes tu actividad física habrá incrementado. 🏃‍♂️

+ ADD BUTTON (OPTIONAL)

... **Show "typing..." for at least**
0.1 sec 4 sec 20 sec

Redirect to Block ⓘ

Redirect users to another block. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. [Learn more](#)
Optional: redirect only [specific users](#) ▾

pide_correo Enter block name

RANDOM

pide_correo: es un bloque intermedio que muestra un texto y manda a llamar al bloque con la petición JSON en su respectivo bloque, este puente se creó para mostrar un mensaje y así obtener un tiempo de respuesta no tan inmediato sin necesidad de un elemento de “typing”.

pide_correo [LINK](#)

Bueno, antes de iniciar.

+ ADD BUTTON (OPTIONAL)

Redirect to Block ⓘ

Redirect users to another block. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. [Learn more](#)
Optional: redirect only [specific users](#) ▾

pide mail JSON Enter block name

RANDOM

video_tutorial: es un bloque que se activa cuando se solicita mostrar el tutorial del Chatbot a través del botón con el mismo nombre. Este desplegará un video que se encuentra alojado en el servidor y la API de Chatfuel lo enviará al usuario para que este le aparezca dentro de su chat y pueda reproducirlo.

video_tutorial [LINK](#)

Esperame tantito, estoy buscando el video.

+ ADD BUTTON (OPTIONAL)

Show "typing..." for at least
0.1 sec 20 sec

Video ⓘ

Send a video file in the chat.

From Facebook: Upload a video to your profile or page, [get the video link](#) and paste it here.

From Cloud Storage: Upload a video to cloud storage, obtain the [direct download link](#) and paste it here. Maximum file size is 20 MB. Supported format is MP4.

✓

Listo :) Continuemos.

+ ADD BUTTON (OPTIONAL)

Redirect to Block ⓘ

Redirect users to another block. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. [Learn more](#)

Optional: redirect only [specific users](#) v

Enter block name

RANDOM

defineHora: este es un bloque que se activa a través del flujo inicial o desde su botón donde tras ser activado mostrará al usuario una serie de botones (quick-reply) con las horas disponibles en las que el usuario podrá ser recordado, estas se encuentran limitadas por la cantidad máxima de botones disponibles y un rango de horario en el que las personas no solo se encuentran despiertas sino más activas.

defineHora [LINK](#)

Show "typing..." for at least
0.1 sec 5.9 sec 20 sec

JSON API ⓘ

Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. [Learn more](#).

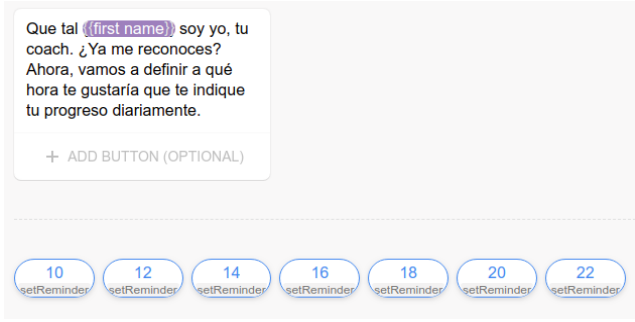
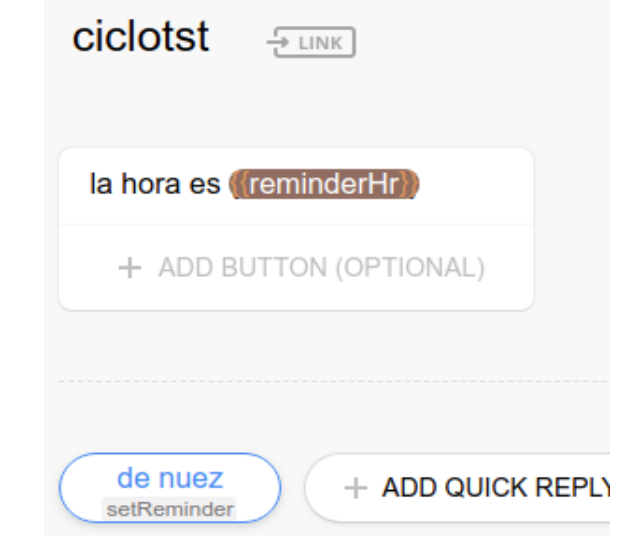
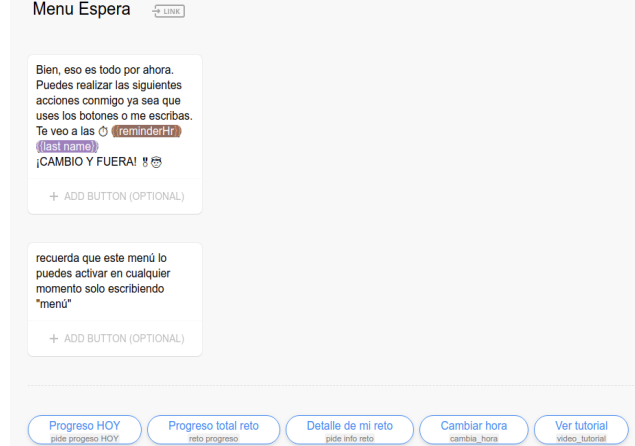

Type: GET URL*

Headers **Pro**
Set a custom header for your JSON call. [Upgrade to Pro](#)

+ Add header

Report errors in the bot (bot users will see the error messages)

Modify Default Plugin Phrases >

	
<p>Ciclotst: es un bloque de prueba el cual se encuentra oculto para ser solo activado con cierto comando y funciona para verificar cual es la hora que definió el usuario y así hacer pruebas.</p>	
<p>Menu Espera: este bloque es activado solamente en la secuencia del registro siendo este el último paso donde el Chatbot indica que han finalizado la primera parte y está listo para iniciar con el coaching y el monitoreo mostrando un menú que contiene botones (quick-reply) con las acciones principales que se estarían realizando en esta etapa de la conversación.</p>	
<p>muestra_menu: este bloque también muestra el mismo menú que el bloque mostrado anteriormente sin embargo este puede ser mandado a llamar en cualquier momento ya sea desde el menú desplegable o escribiendo directamente al Chatbot pidiéndole mostrar el menú.</p>	

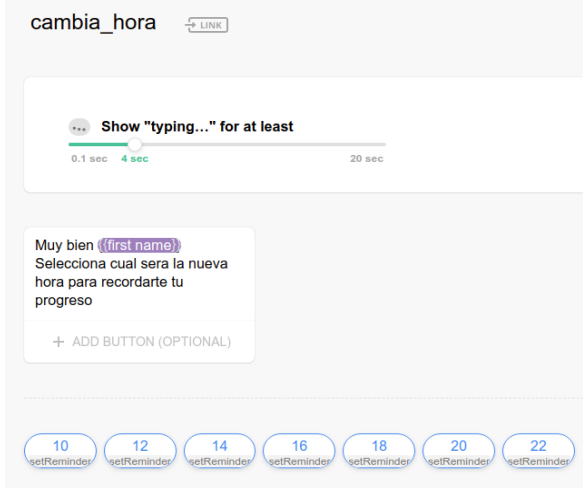
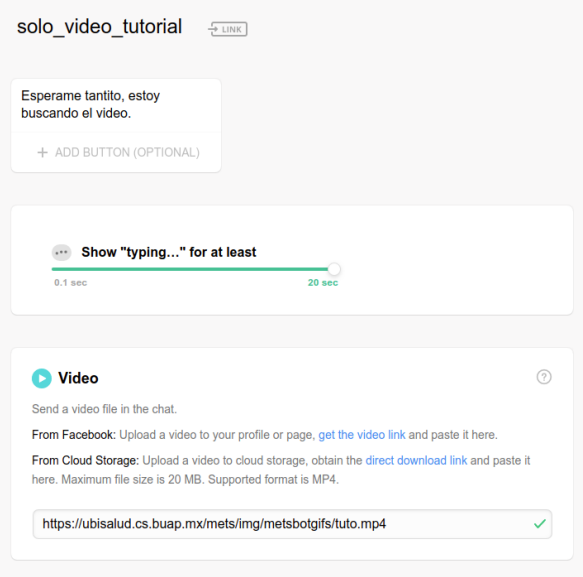
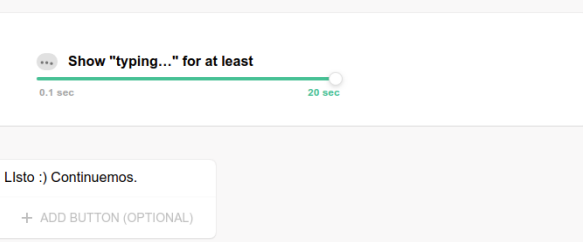
<p>cambia_hora: este bloque similar al de “Define hora” mostrará las horas disponibles y hará el cambio del recordatorio.</p>	
<p>solo_video_tutorial: este bloque como su nombre lo dice solo mostrará nuevamente el video que sirve como tutorial al Chatbot, pero es invocado desde el botón únicamente.</p>	
	

Tabla 14. Tabla con el contenido de cada bloque del grupo Registro a METS.

4.9.3.7 Grupo COACHING

Descripción, este grupo de bloques está caracterizado por contener aquellos bloques que se utilizan sobre la etapa de la conversación donde se le da seguimiento al usuario y en ello se involucra la parte del Coach donde este resulta ser el mismo avatar que seleccionaron en la plataforma y aquí estará mostrándose para alentarlos, regañarlos, felicitarlos, entre otros como se puede apreciar en la figura 11s y que posteriormente cada uno de sus bloques será descrito en la tabla 15.



Figura 11s. Ventana del grupo Coaching con sus bloques.

El grupo y sus bloques son iniciados como antes se pudo mostrar, cuando son llamados de manera textual o activadas por algún botón del menú.

Esto se logra a través de bloques gatillo que son los que mandan a llamar al resto y estos son: Pide Info Reto, reto progreso y pide progreso HOY.

Bloque con su descripción-Imagen

Pide info Reto: este bloque lo que realiza es saludar al usuario y posteriormente realizar un llamado al api para obtener la imagen de su coach, una vez mandado el gif se hace el segundo llamado al api para obtener la información acerca del reto que tiene activo el usuario para mostrárselo en el chat.

pide info reto [LINK](#)

OK, vamos a checar la información de tu reto 🤖👁️

+ ADD BUTTON (OPTIONAL)

... Show "typing..." for at least

2.8 sec 20 sec

PROCESANDO

+ ADD BUTTON (OPTIONAL)

JSON API

Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. [Learn more.](#)

Type URL*

Headers **Pro**
Set a custom header for your JSON call. [Upgrade to Pro](#)

Report errors in the bot (bot users will see the error messages)

[Modify Default Plugin Phrases >](#)

Qué tal {first name}

+ ADD BUTTON (OPTIONAL)

JSON API

Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. [Learn more.](#)

Type URL*

Headers **Pro**
Set a custom header for your JSON call. [Upgrade to Pro](#)

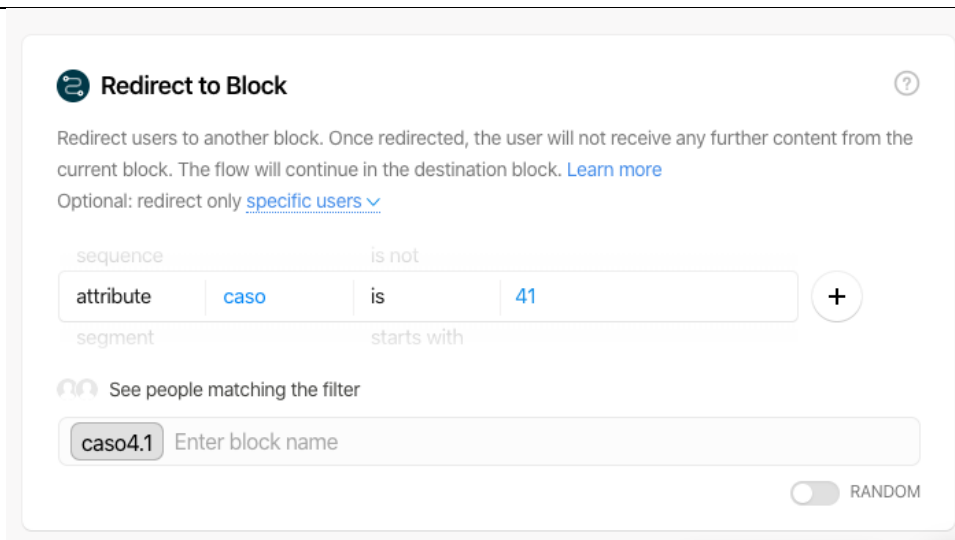
Reto progreso: este es un flujo un poco más complejo y se inicia en este bloque con un mensaje de que se va a realizar algo para posteriormente hacer un llamado a la API y obtener la data específica del reto además de corroborar que el usuario tiene un reto. Si el usuario no tiene reto, accedió ahí sin tener los requisitos previos o si da un error al consultar entonces le enviará a Mets, de lo contrario prosigue con el siguiente bloque.

The screenshot shows a chatbot flow block titled "reto progreso" with a "LINK" icon. Below the title is a message box containing the text "Es hora de checar tu progreso" followed by a placeholder for a variable: "{first name}". Below the message box is a button labeled "+ ADD BUTTON (OPTIONAL)". Below the message box is a "JSON API" block. The "JSON API" block has a description: "Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. Learn more." Below the description are two input fields: "Type" with a dropdown menu set to "GET" and "URL" with a text input field containing "http://ubisalud.cs.buap.mx:5002/api/coach/consultaReto/" followed by a blue button labeled "metsId".

checa porcentaje reto: este bloque en particular tiene dos funcionalidades, pero el mismo objetivo que es, indicarle al usuario como va su progreso y con base en ello continuar con el conductivismo que tiene el coach ya sea con una felicitación o un regaño, en una primera versión del Chatbot se utilizará esto con base en los pasos, días y porcentaje calculado del reto que regresaba la API de acuerdo a la base de datos de Mets. Y a través de condicionales solo pasamos al siguiente bloque de acuerdo al caso, estas condiciones se mantienen para una futura mejora de este flujo en combinación probablemente con la implementación de esta versión.

The screenshot shows a chatbot flow block titled "Redirect to Block" with a question mark icon. Below the title is a description: "Redirect users to another block. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. Learn more". Below the description is an optional filter: "Optional: redirect only specific users". Below the filter are two conditional rows. The first row has "attribute" set to "porcentajeH", "less than" set to "50", and an "and" button. The second row has "attribute" set to "diasCumplid", "less than" set to "5", and a "+" button. Below the conditional rows is a "segment" input field with the text "reto menor 50 y dias pocos" and a placeholder "Enter block name". Below the segment input field is a "RANDOM" toggle switch.

La versión actual del Chatbot cuenta con la misma idea sin embargo tras realizar cambios en la API lo que se regresa ya no es data de la base de datos sino que en la misma API se realiza una serie de condicionales para poder enviar una variable que se utilizara a nivel del Chatfuel y así disminuir la complejidad de ese componente, esta variable se llama CASO el cual indica a qué caso corresponde la información del usuario y según sea el valor de CASO se dirige al bloque correspondiente.



La implementación de actualizar estos casos se logró gracias al uso que algunos usuarios estaban teniendo con el Chatbot y la App de Mets, al final los casos quedaron con el siguiente orden:

caso1.1 ...	caso1.2	caso1.3
caso2.1	caso2.1.1	caso2.2
caso2.3	caso2.4	caso3
caso4	caso4.1	pide progreso HOY

Caso 1: El usuario ha cumplido con su reto diariamente desde la fecha de inicio del reto, donde:

- Caso 1.1: El atributo de sus días por hacer está al corriente o igual a 1.
- Caso 1.2: El atributo de sus días por hacer es mayor al corriente es decir mayor a 1.
- Caso 1.3: El atributo de sus días por hacer es menor que 1.

Caso 2: El usuario está activo porque tiene información sobre días cumplidos, pero ha sido penalizado por no cumplir bien el reto, donde:

- Caso 2.1: La cantidad de días que han pasado en el reto son mayor a los días que ha cumplido con el reto y además ese valor es mayor a cero días.
- Caso 2.1.1: Con base en el caso anterior vemos que la cantidad de días pasados está justificada porque está iniciando el reto apenas.
- Caso 2.2: Sus días pendientes a realizar es estrictamente 1 basándonos en el caso anterior.
- Caso 2.3: Sus días pendientes a realizar son mayor que 1 con base en el caso 2.1
- Caso 2.4: No tiene días pendientes a realizar respecto al caso 2.1

Caso 3: El usuario hoy empieza un reto, donde:

Caso 3: sus días que han pasado del reto corresponde a los días que ha cumplido y estos están en el valor inicial.

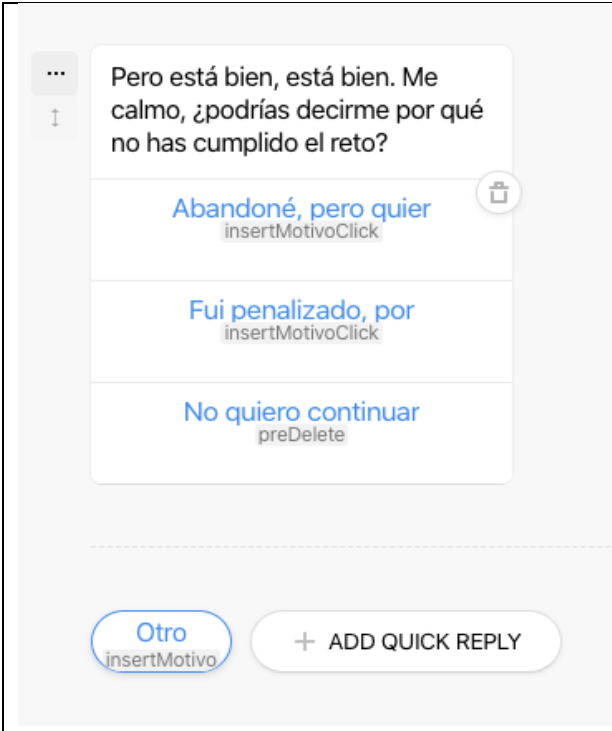
Caso 4: El usuario inició el reto hace varios días y no lo ha cumplido (no ha caminado o bien fue penalizado), donde:

Caso 4.1: Los días que han pasado del reto son mayores a los días en los que ha cumplido el reto, pero porque los días cumplidos están en un valor inicial.

Caso 4: Pero con respecto al caso anterior si los días que han pasado es mayor que los días pendientes a cumplir.

El objetivo principal de tener este flujo de bloques tan definido como se muestra es porque primeramente es probablemente el flujo conversacional que más se estará utilizando en todas las guías de conversación con el Chatbot y con base en ello el lenguaje empleado para contribuir a la idea conductual de crearles un hábito o aumentar su adherencia al programa de METS se puede ver merma por cuestiones de vocabulario empleado según los datos obtenidos, ya que recordemos que en teoría quien le felicita y regaña al usuario es el propio coach virtual y esto se veía muy polarizado creando un sentimiento de que se estaba siendo bastante estricto siendo algo molesto o poco atractivo para la finalidad. Así entonces al crear los casos estos reenvían a los diferentes bloques con una gama un poco más amplia de regaños o felicitaciones además de ser más acertado respecto a los datos, es como si hubiéramos ampliado el espectro de alegría y enojo del Chatbot por medio de una mayor cantidad de bloques y de una forma más específica.

De estos casos uno que vale la pena resaltar es el Caso 4, ya que este es nuestro peor de los casos, es decir quiere decir que el usuario ha abandonado el reto por completo y muy probablemente ha perdido la adherencia al programa por si el usuario decide conocer su progreso ya sea porque recordó que utiliza a Metsbot o a través de las notificaciones que el mismo Metsbot manda, captamos al usuario y le preguntamos que nos pueda explicar por qué no ha cumplido con su reto.

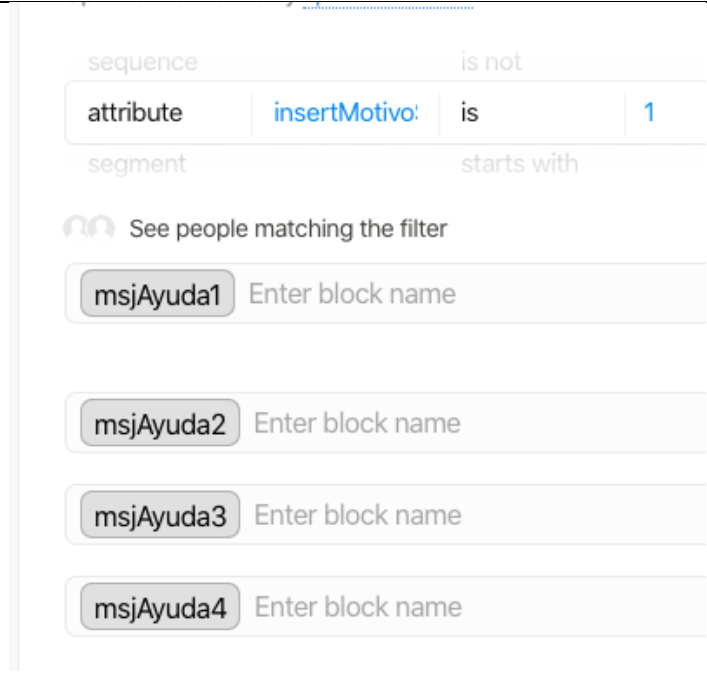


Partimos con un gif del coach mostrando molestia acto seguido esta muestra que se calma y pregunta al usuario los tres primeros motivos medibles para METS como plataforma que podrían ser causas de la falta de adherencia:

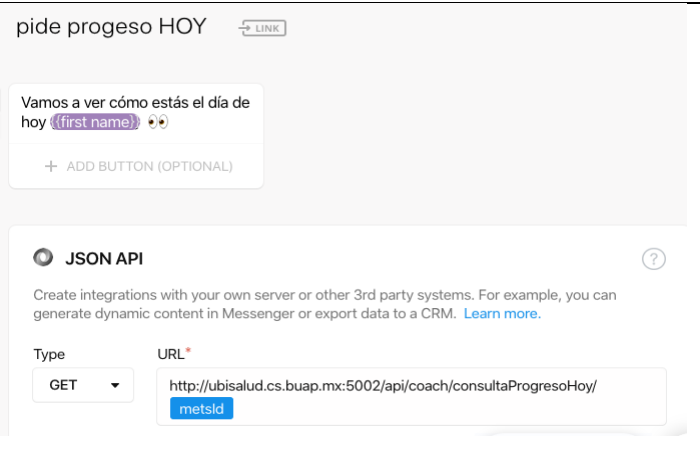
- Abandono, pero quiere retomar
- Lo penalizaron y abandonó.
- Ya no quiere continuar

La última opción de no querer continuar lleva a un bloque donde el usuario será eliminado si así lo desea. De esa forma será dado de baja de la API para que no tenga interacción con el Chatbot. Sin embargo, su información dentro de Mets se conservará con fines analíticos.

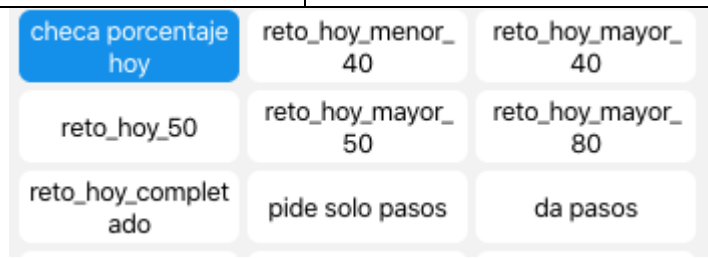
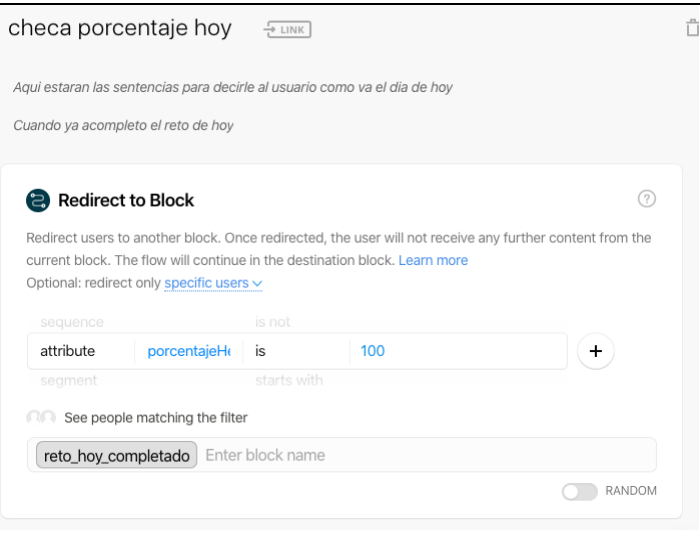
También existe la posibilidad de agregar otro motivo el cual se guardara, pero no contabilizará para tener una lista de motivos varios por los cuales los usuarios pierden adherencia a este tipo de programas y/o implementaciones. Después de que haya insertado su motivo el Chatbot de manera aleatoria le mencionara alguna frase con la finalidad de indicarle al usuario que entiende sus motivos y le alienta a que no vuelva a decaer si gusta regresar.



pide progreso HOY: es un bloque que imita bastante el comportamiento del bloque reto progreso.
 Donde también envía un mensaje al usuario para indicarle que realizara algo, acto seguido consulta con la API si hay información disponible, de ser así lo reenvía al bloque siguiente que es checa porcentaje hoy.



checa porcentaje hoy: este bloque es invocado por el bloque de “pide progreso HOY” donde la API ya regreso la información correspondiente sobre los pasos de hoy según el reto en el que está inscrito el usuario y solo evalúa el atributo “porcentaje Hecho Hoy” para indicar una acción de acuerdo a su progreso, esto se logra a través de los siguientes bloques: “reto hoy_menor_40”, “reto_hoy_mayor_40”, “reto_hoy_50”, “reto_hoy_mayor_50”, “reto_hoy_mayor_80”, “reto hoy completado”.



Estos bloques de respuesta contienen una composición de elementos internos similares que es el GIF del coach correspondiente y un texto personalizado que responde al resultado arrojado por la API.

Por ejemplo, en el bloque “reto hoy menor 40”
Tenemos el llamado a la API para que nos regrese la imagen del coach correspondiente al usuario según su atributo “method”, en este caso un coach molesto.

reto_hoy_menor_40 [LINK](#)

JSON API ⓘ

Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. [Learn more.](#)

Type **URL***

GET

Headers **Pro**

Set a custom header for your JSON call. [Upgrade to Pro](#)

Report errors in the bot (bot users will see the error messages)

[Modify Default Plugin Phrases >](#)

Seguido del texto con el dato entregado por la API almacenado en el atributo “Pasos Hechos Hoy”.

{{first name}} no puedo creer que sólo lleves **PasosHechosHoy**. Eso es menos del 40%, por eso me molesté un poco. Te sugiero que camines más.

Segundo ejemplo con el bloque de “reto hoy completado”:
Tenemos el llamado a la API para que nos regrese la imagen del coach correspondiente al usuario según su atributo “method”, en este caso un coach alegre indicando que se hizo bien.

reto_hoy_completado [LINK](#)

JSON API ⓘ

Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. [Learn more.](#)

Type **URL***

GET

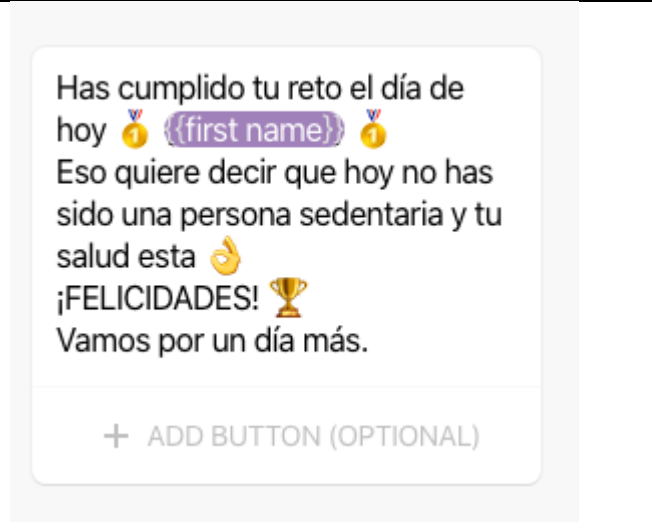
Headers **Pro**

Set a custom header for your JSON call. [Upgrade to Pro](#)

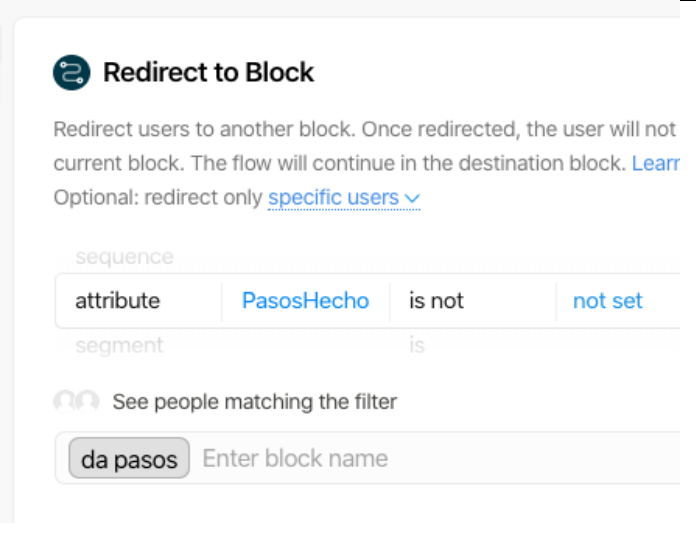
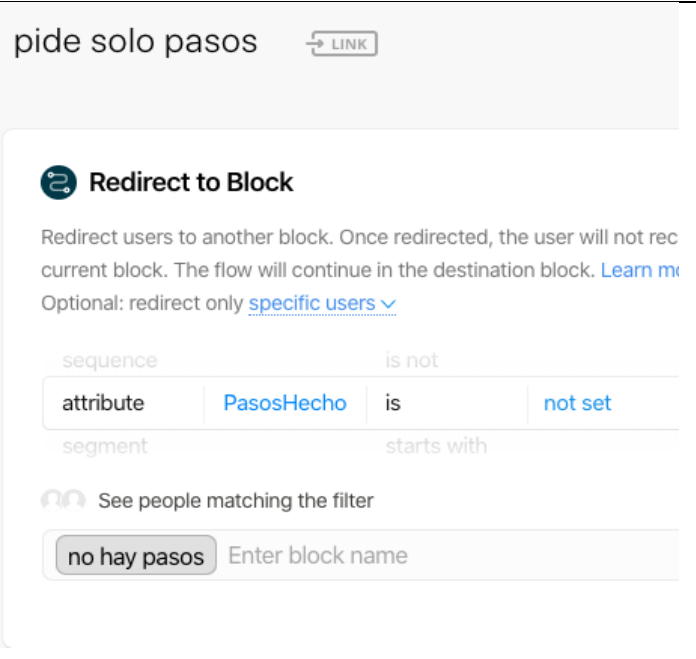
Report errors in the bot (bot users will see the error messages)

[Modify Default Plugin Phrases >](#)

Posteriormente el texto correspondiente, solo que esta vez el atributo entregado por la API desde el bloque de “checa porcentaje hoy” nos dice que se ha completado el reto, es decir está al 100% por ende no hay necesidad de tomar este último dato, almacenarlo en un atributo y mostrarlo, así que simplemente felicitamos el logro.



pide solo pasos: es un bloque que verifica con la API si hay pasos registrados por parte del usuario, en caso de que no haya les enviará al bloque de “no hay pasos”, en caso contrario enviará al bloque de “da pasos”.

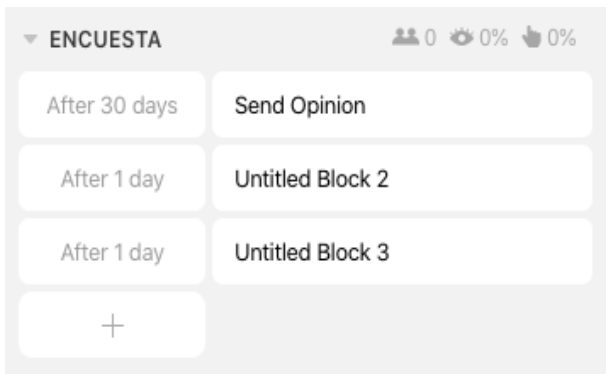


<p>no hay pasos: es el bloque invocado por el bloque anterior de “pide solo pasos” e indica que la API al hacer la consulta no ha encontrado información sobre los pasos del usuario, ya sea porque no ha realizado ni uno o por algún error al obtener la información.</p>	
<p>da pasos: es el bloque invocado por el bloque de “pide solo pasos” cuando si ha encontrado la API información sobre los pasos del usuario y en este bloque simplemente se muestra.</p>	

Tabla 15. Tabla con el contenido de cada bloque del grupo Coaching.

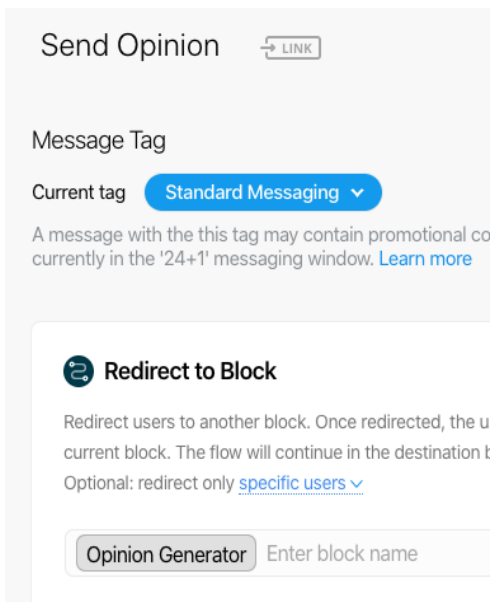
Estos últimos tres bloques corresponden al mismo flujo donde el usuario solo quiere conocer su progreso en pasos sin ningún mensaje por parte del coach, función similar a la de abrir la APP de Mets para visualizar el avance.

4.9.3.8 Secuencia Encuesta



Esta secuencia de bloques mostrada en la figura 11, es activada en los primeros bloques que inician la conversación con el Chatbot y en este caso pasado 30 días desde esa interacción se activa el bloque llamado “Send Opinión”.

Figura 11t. Ventana de la secuencia Encuesta con sus bloques.



Send Opinión es un bloque pivote que simplemente sirve para redireccionar a un bloque llamado Opinión Generator como podemos ver en la figura 11u.

Figura 11u. Ventana de el bloque Send Opinión.

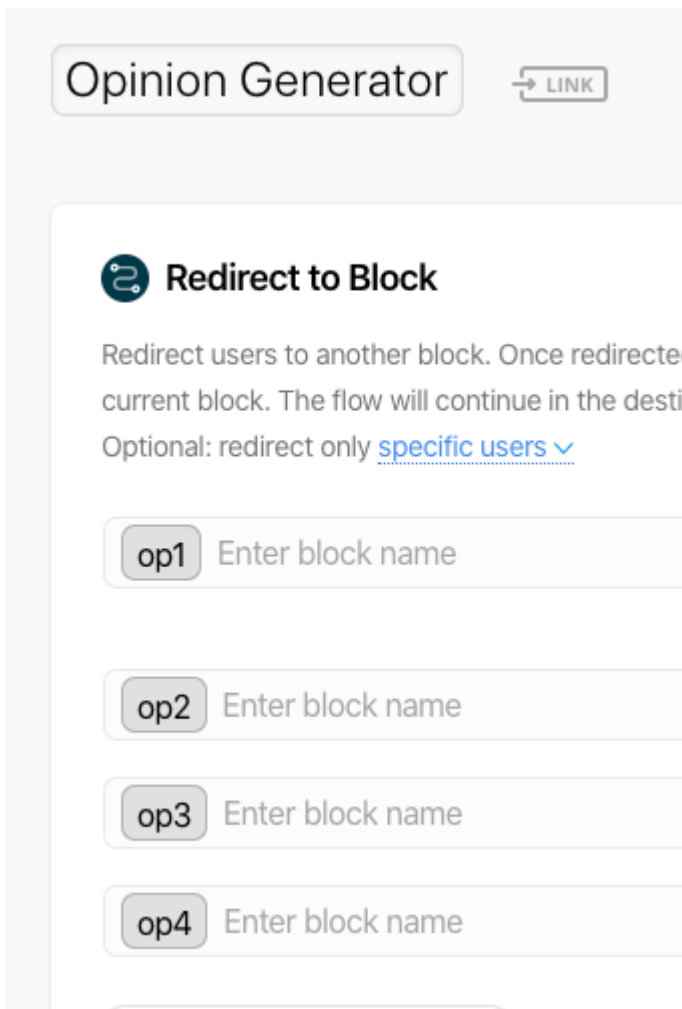


Figura 11v. Ventana de el bloque Opinion Generator.

4.9.3.9 Grupo Random Rating Questions

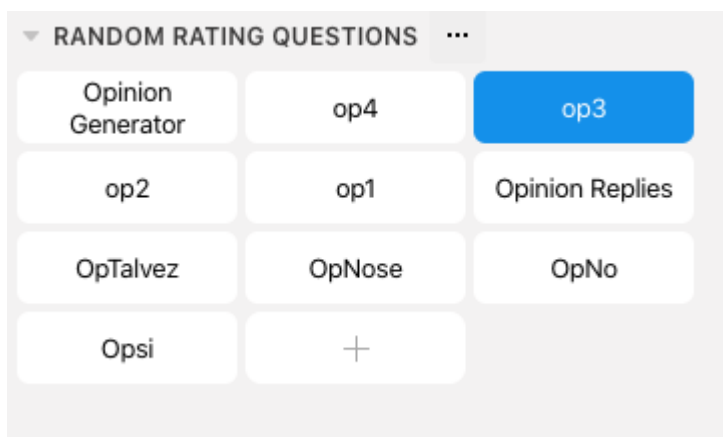


Figura 11w. Ventana de el grupo Random Rating Questions y sus bloques.

Opinión Generator es un bloque el cual tiene activada

la opción de “Random” RANDOM la cual permite direccionar de manera aleatoria a cualquiera de los bloques declarados dentro del Card de “Redirect to Block” en este caso tenemos cuatro bloques diferentes llamados: “op1”, “op”, “op3”, “op4” correspondientemente como se muestra en la figura 11v, cada uno contiene una forma diferente de pedir una opinión al usuario con respecto al Chatbot. Entonces este bloque lo que realiza es de manera aleatoria escoger de qué forma va a pedirle al usuario una opinión a través de los bloques declarados en el grupo llamado “Random Rating Questions”.

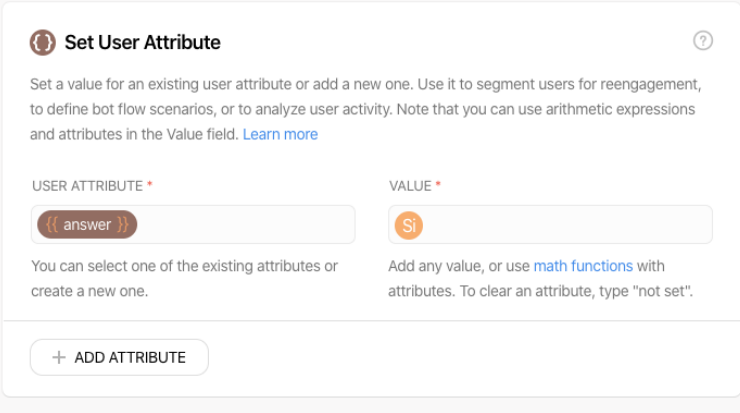
Este grupo ilustrado en la figura 11w, crea el flujo de conversación donde el Chatbot después de 30 días de su primera interacción le indicará al usuario que brinde una opinión acerca de su utilidad.

Los bloques “op1”, “op”, “op3”, “op4” tienen la misma composición que es: “Texto” y una redirección al bloque “Opinion Replies” que sirve para encapsular las respuestas.

Los textos que contienen las diferentes opiniones son:

- Op1: “Hola {{firstname}}, tengo una pregunta 🙋 ¿Sientes que te he sido útil? ”
- Op2: “Hey {{first name}} siento que llevamos ya un tiempo de conocernos y como ha sido tu progreso.” -2 segundos de espera- “Por ello me gustaría que me respondieras si crees que te he sido de utilidad para continuar con tus retos.”
- Op3: “Me da un poco de pena preguntarte esto, pero, después de tanto tiempo que nos conocemos... 🙄” -2 segundos de espera- “¿Crees que te he sido de utilidad?, ósea que te he ayudado a incrementar tu actividad física, a caminar más y así.”
- Op4: “Hola {{firstname}} una pregunta rápida.”, “¿Te he sido de utilidad?
Es decir, notas que conmigo has podido cumplir tus retos, algo que tan solo con la app no hubieras logrado. ¿Sientes que has aumentado tu actividad física gracias a mí? ósea que caminas más, por ejemplo.”

Cabe mencionar que una vez terminado el texto se activa y muestra lo correspondiente al bloque “Opinion Replies” el cuál esta desglosado en la tabla 16.

Bloque-imagen	
<p>Opinión Replies: es un bloque que muestra un emoji y las opciones disponibles a la pregunta que se realizó anteriormente en los bloques de “Op”</p>	
<p>Los bloques de respuesta con los nombres de “Opsi”, “OpNo”, “OpNose”, “OpTalvez” tienen la misma estructura donde se guarda en un atributo de nombre “answer” la respuesta que contiene cada botón, es decir el texto de “SI”, “NO”, “No se”, “Tal vez” e inmediatamente los manda al grupo de bloques llamado “rating generator”.</p> <p>Estos bloques de respuesta de se ven así:</p>	

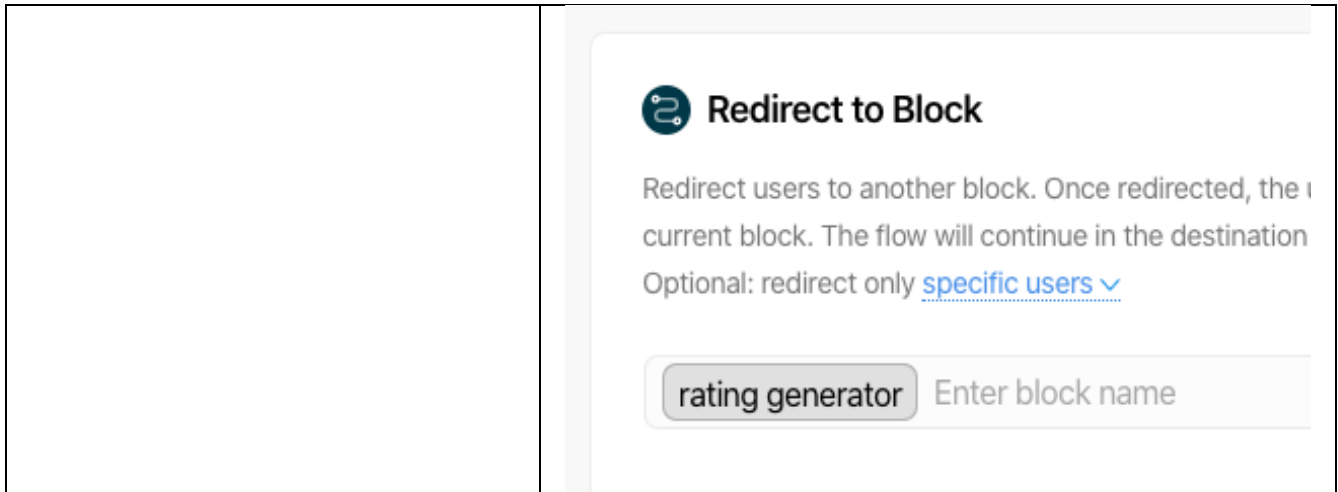


Tabla 16. Tabla con el contenido del bloque de Opinion Replies y su sub flujo.

4.9.3.10 Grupo Rating

Este grupo de bloques mostrado en la figura 11x contienen la parte final del flujo sobre la encuesta donde después de dar una respuesta concreta también se califica el servicio con estrellas. Posteriormente el detalle de sus bloques se encuentra desglosado en la tabla 17.

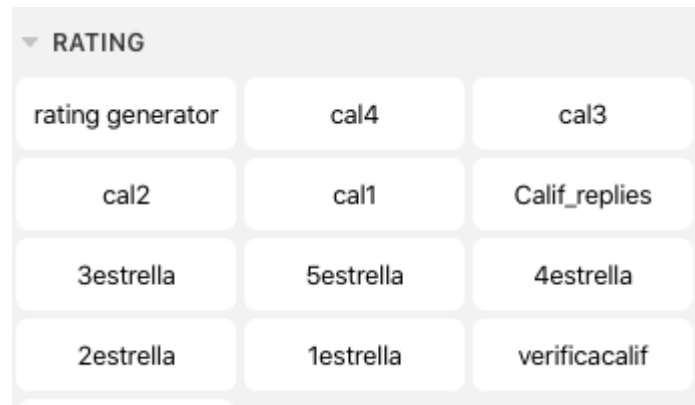



Figura 11x. Ventana del grupo Rating y sus bloques.

Bloque y su descripción-Imagen

rating generator: este bloque similar al bloque de “Opinión Generator” contiene la opción de “Random” activada para que de manera aleatoria escoja alguna de las 4 formas diferentes de decirle al usuario que califique con estrellas el servicio de Metsbot.

rating generator [LINK](#)

Redirect to Block

Redirect users to another block. Once redirected, the flow will continue in the current block. The flow will continue in the current block. Optional: redirect only [specific users](#) 

cal1 Enter block name

cal2 Enter block name

cal3 Enter block name

cal4 Enter block name

Los bloques “cal1”, “cal2”, “cal3”, “cal4” contienen la misma composición de un texto y el redirigirlos al bloque de “Calif Replies”, como se muestra en la siguiente imagen.

cal3 

Pstt!

+ ADD BUTTON (OPTIONAL)

No le vayas a decir a nadie pero estoy buscando un aumento y para ello necesito que me digas con cuantas 🌟 calificas mi servicio en general.

+ ADD BUTTON (OPTIONAL)

Redirect to Block

Redirect users to another block. Once recurrent block. The flow will continue in the current block. The flow will continue in the current block. Optional: redirect only [specific users](#) v

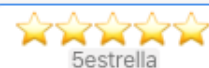
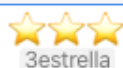
Calif_replies Enter block name

Calif Replies: es un bloque que solo contiene las opciones de calificación con los emojis de estrellas los cuales redireccionan a su propio bloque donde guardará ese valor.

Calif_replies  m.me/metsbot?ref=Pregunta_calif

:)

+ ADD BUTTON (OPTIONAL)



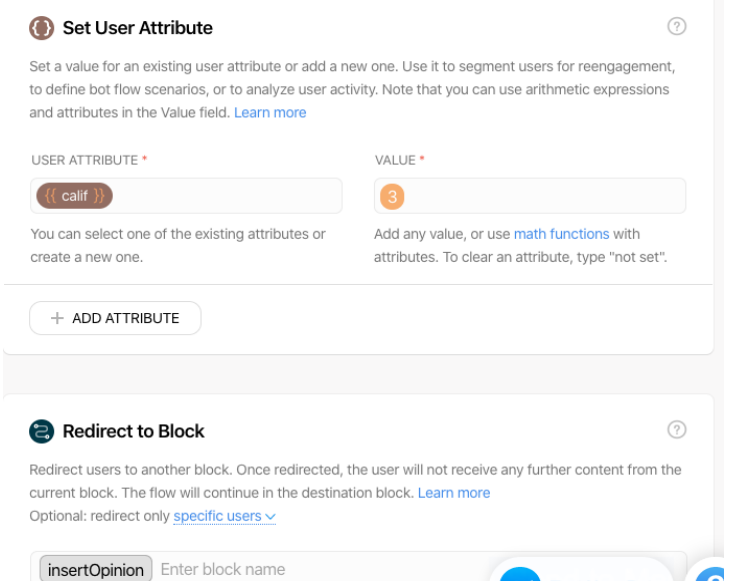

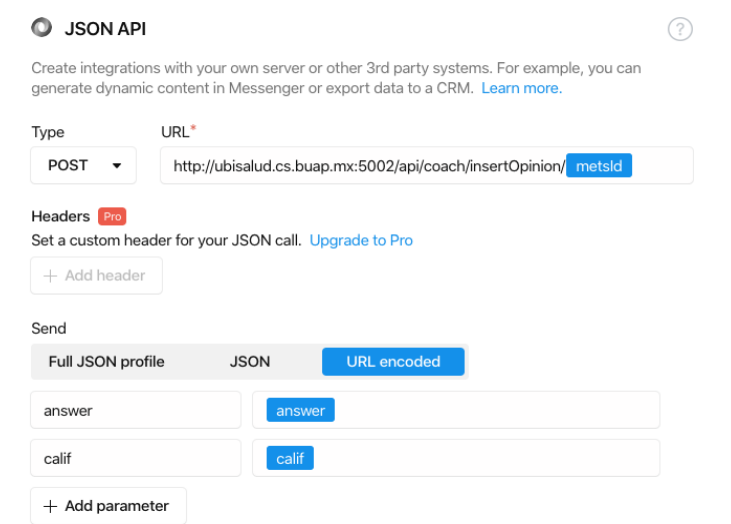
<p>Los botones con estrella redireccionan a su bloque correspondiente y estos contienen la misma estructura la cual solo es guardar el valor en un atributo de nombre “calif” y redirecciona al bloque de “insertOpinion” como se muestra a continuación:</p>	 <p>Set User Attribute</p> <p>Set a value for an existing user attribute or add a new one. Use it to segment users for reengagement, to define bot flow scenarios, or to analyze user activity. Note that you can use arithmetic expressions and attributes in the Value field. Learn more</p> <p>USER ATTRIBUTE * <input style="border: 1px solid #ccc;" type="text" value="{{ calif }}"/> VALUE * <input style="border: 1px solid #ccc;" type="text" value="3"/></p> <p>You can select one of the existing attributes or create a new one. Add any value, or use math functions with attributes. To clear an attribute, type "not set".</p> <p><input type="button" value="+ ADD ATTRIBUTE"/></p> <hr/> <p>Redirect to Block</p> <p>Redirect users to another block. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. Learn more</p> <p>Optional: redirect only specific users v</p> <p><input type="text" value="insertOpinion"/> Enter block name <input type="button" value="Test Your Bot"/></p>
<p>insertOpinion: este bloque perteneciente al grupo de “JSON CALLS” lo que realiza es indicarle al usuario que hará algo, en este caso será enviar los atributos que se estuvieron obteniendo y mandarlos a la API donde se almacenarán y este mismo confirma de recibido, en caso de que haya algún error al ingresar la información entonces se le enviará al usuario a la página de Mets indicando que hubo un error.</p>	 <p>insertOpinion <input type="button" value="LINK"/></p> <p>Bien, procesando tus opiniones.</p> <p><input type="button" value="+ ADD BUTTON (OPTIONAL)"/></p> <p></p> <p><input type="button" value="+ ADD BUTTON (OPTIONAL)"/></p>
	 <p>JSON API</p> <p>Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. Learn more.</p> <p>Type <input type="text" value="POST"/> URL * <input type="text" value="http://ubisalud.cs.buap.mx:5002/api/coach/insertOpinion/metsid"/></p> <p>Headers Pro Set a custom header for your JSON call. Upgrade to Pro</p> <p><input type="button" value="+ Add header"/></p> <p>Send <input type="button" value="Full JSON profile"/> <input type="button" value="JSON"/> <input checked="" type="button" value="URL encoded"/></p> <p><input type="text" value="answer"/> <input type="button" value="answer"/></p> <p><input type="text" value="calif"/> <input type="button" value="calif"/></p> <p><input type="button" value="+ Add parameter"/></p>

Tabla 17. Tabla con el contenido del grupo Rating y sus bloques.

4.9.3.11 Grupo JSON CALLS

Este grupo en particular mostrado en la figura 11y contiene bloques que solo realizan llamados a la API y a pesar de ser una parte de un flujo no es parte fundamental de uno es por ello que debido a esto y por fines prácticos de tener en grupo los llamados a la API que son más específicos, es por ello que también aquí se encuentran bloques que sirven para realizar pruebas y verificar el estado de la comunicación entre el componente de Chatfuel y el componente de la API.



Figura 11y. Ventana de el grupo JSON Calls y sus bloques.

4.9.4 Mensajes de la API

Existen respuestas o mensajes que no se encuentran en el componente de Chatfuel y este no los envía directamente debido a que son respuestas o mensajes que se envían directo desde la API, ya que el mandar una respuesta que ligase a otro bloque para qué se mostrará resultaba en una serie de bloques para respuestas con un brinco innecesario de API a bloque y de bloque a otro bloque.

Los mensajes que envía la API en cuanto al registro del email del usuario en el flujo donde el usuario se da de alta en Mets son:

- "Este email " + email + " no lo he encontrado registrado en METS :o "
- "Parece que aún no te has inscrito a ningún reto, ve a tu bitácora en Mets, aquí abajo en el menú está el enlace, inscríbete a un reto y puedes regresar para configurar tu recordatorio sólo dime DEFINIR RECORDATORIO :) "
- "Perfecto, tu meta actual diaria es alcanzar los " + updatedUser.metaPasos + " yo te creo capaz de ello. También te mandaré tu progreso a las " + updatedUser.timeReminder + " para ver cómo andas de ese sedentarismo :D"
- "Oops al parecer no te he encontrado en la plataforma Mets, revisa que estés correctamente en ella o entra a la pagina y en la sección de contacto mándanos un mensaje con este error."

Los mensajes que envía la API en los flujos que tienen que ver con el proceso de coach y demás son:

- "No te veo como usuario en Metsbot, ingresa a la página de ubisalud para reportar este problema"
- "😭 ¡AY!, intentare hacerlo mejor, gracias por tu opinión."
- "Gracias por tu opinión sigo mejorando para darte un mejor servicio. 🤗 <3"
- "No te veo como usuario en Metsbot, ingresa a la página de ubisalud para reportar este problema"
- "Wow!! hoy has logrado hacer: " + results[0].PasosHechos + " pasos."
- "Bien, analizando tu información tu reto es lograr " + userData.metaPasos + " al día, déjame ver..."
- "🎉 BIP BUP BIP 🎉"

- "Parece que aun no te has inscrito a ningún reto, ve a tu bitácora en Mets, aquí abajo en el menú está el enlace, inscríbete a un reto y puedes regresar para configurar tu recordatorio, sólo dime DEFINIR RECORDATORIO :)"
- "Bien, analizando tu información tu reto es lograr " + results[0].MetaPasos + " al día, durante " + results[0].DiasTotales
- "Bien, estoy checando tu información."
- "Llevas 25 días sin progresar, aún hay tiempo de que te recuperes, pasando los 30 estarías perdiendo totalmente los retos en METS"
- "... WOW, llevas mas de 30 días sin cumplir el reto, debido a ello me veo en la penosa necesidad de eliminar tu usuario de METS, sin embargo, recuerda que puedes volver a iniciar borrando el chat y será como borrón y cuenta nueva."
- "Oops, ALERTA DE SEDENTARISMO! \n Sólo has logrado " + pasitos + " pasos en las últimas 2 horas, te recomiendo que vayas a dar una vuelta por ahí"
- "Wow!! hoy has logrado hacer: " + results[0].PasosHechos + " pasos. \n Has cumplido tu reto el día de hoy 🏆 y vencido el sedentarismo el día de hoy 🙌 ¡FELICIDADES! 🏆 Vamos por un día más. "
- "Hey!, vas bastante bien, llevas " + results[0].PasosHechos + " pasos 🙌 \n Un " + porcentajeHecho + "% no te detengas que estás apunto de completar tu reto de hoy! 😎"
- "Ya es la hora y vas bien, ya pasaste la mitad del reto 😎 \n lo difícil ya pasó, no es hora de detenerse, sigue así."

La composición de los diálogos presentados en el componente de Chatfuel se complementan con estos últimos enviados desde la API que como se puede apreciar la mayoría parecen textos de diálogos incompletos, sin embargo solo es el inicio o fin de la conversación presentada en un texto, también están los mensajes de error que se pueden suscitar por fallos en los datos o accesos en la API, por último tenemos los mensajes que se envían de forma automática sin pasar por el componente de Chatfuel sino realizando una comunicación directa con la API de Messenger de Facebook, ya que estas tareas o mejor dicho mensajes automatizados no se podrían programar desde el componente de Chatfuel. Sin embargo, como se mencionó anteriormente esta serie de combinaciones permiten una completitud y flexible en cuanto a la comunicación que se quiere lograr a través del coach.

4.9.5 Subcomponente de Chatfuel, el lenguaje y la conversación

Resumiendo varias de las secciones anteriores hay un punto en particular que se ha comentado sin embargo se le ha restado un poco de importancia, esto debido a que entonces no se conocían los componentes de la plataforma ni del Chatbot sin embargo con este contexto ya podemos comprender el tema de la integración psicológica en el Chatbot, esto es el componente clave del proyecto donde el Chatbot no solo adquiere personalidad sino que se convierte en un tipo de coach realmente, ya que contiene componentes humanos. Para lograr esto ya conocemos que existe un diagrama de conversación sin embargo el diagrama de conversación solo conocemos que debería estar orientado a que la gente pueda conocer y utilizar la plataforma de Mets a su vez de intentar lograr una adhesión al programa de Mets de forma más prolongada. Lo que queríamos lograr no es nada nuevo, es algo que ya se ha estudiado, realizado y logrado en diferentes contextos desde tomar medicamentos, ejercitarse o quitar malos hábitos como el fumar. Sin embargo estamos hablando de ejemplos que utilizan de componente humano a una persona real, un experto y psicólogos, los métodos que emplean estos últimos por lo general para este tipo de problemáticas con las terapias de tipo cognitivo-conductual y/o sus técnicas complementadas con algunas otras, para nuestro caso haremos lo mismo, emplear técnicas de la terapia cognitivo-conductual para intentar lograr el punto de que los usuarios usen el sistema de Mets y se pueda prolongar la adhesión al programa que este implementa, solo nos queda el cómo lograr que la gente llegue a este paso y para ello combinamos a la fórmula la terapia mejor conocida como aceptación de las

precauciones, ambas técnicas ya fueron descritas anteriormente y por ello no ahondaremos más en ellas. Una vez definiendo qué es lo que vamos a utilizar como idea, con el apoyo de la psicóloga Ana Carrillo se empezó a crear el diagrama de flujo donde en su estructura más básica se podría visualizar como:

Inicio -> Aceptación de las precauciones para convencer -> integración a la plataforma -> Técnica de terapia cognitivo conductual para adherencia.

Sin embargo, como ya pudimos apreciar anteriormente esto creció y se expandió al colocar los bloques de conversación. Estos están redactados con el apoyo de la psicóloga para mantener un lenguaje claro y que se preste para público en general, pero con un lenguaje orientado a estas técnicas también.

Llegados a este punto encontramos que fijándonos en el flujo de la conversación tenemos el inicio y de forma inmediata abarcamos los primeros pasos de la técnica de aceptación de las precauciones informando al usuario de que trata este Chatbot, cuál es la problemática e indicarle explícitamente que él podría ser propenso a tener ese problema, en este caso sedentarismo, esto lo podemos apreciar en los grupos de bloques titulados: saludos y motivación según encuesta.

Continuando con ello hacemos unos saltos en los pasos de la terapia cognitiva conductual debido al apoyo y dinamismo que tuvimos con la conversación anterior, llegando al punto donde se requiere una constancia, una repetición, un estilo de sistema de recompensas para ir logrando desvanecer el hábito de llevar un ritmo sedentario e iniciar un hábito nuevo más orientado a tener y mantener una actividad física de manera constante, esto último que se mencionó sabemos, ya que es el último paso de la Terapia cognitiva conductual por ende también estamos dando por hecho de que la inmediatez que presenta Metsbot lo logra, sin embargo como ya vimos también esto suele tener recaídas por ende se crean líneas extras sobre el flujo de conversación para intentar recuperar al usuario y que este pueda acceder a ellos para tener un apoyo extra y que él pueda después internamente adaptar esos hábitos y saber cómo actuar en alguna recaída donde incluso Metsbot puede recopilar información de este acto para conocer motivos y posteriormente analizar cómo encontrarle una solución.

4.9.6 Componente de la API

Al estar hablando de una API REST como se mencionó anteriormente parte de su distintivo es la estructura de sus componentes, a lo cual podremos definir como árbol de archivos, el cual está compuesto de la siguiente manera:

- ❑ API
 - ❑ controllers
 - ❑ adminTools.js
 - ❑ coachingTools.js
 - ❑ email.js
 - ❑ phrases.js
 - ❑ shared.js
 - ❑ models
 - ❑ phrases.js
 - ❑ users.js
 - ❑ routes
 - ❑ index.js
- ❑ node_modules
- ❑ index.js
- ❑ package.json

La carpeta “controllers” como su nombre lo indica contiene los archivos que forman parte de los controladores de la API es decir aquellos que realizan las operaciones, cálculos y acceden a la base de datos tanto a la de METS como a las de Metsbot, el contenido de estos archivos ya ha sido mencionado anteriormente en las funciones de la API a excepción del archivo shared.js que simplemente es una función que se exporta para poder ser usada en los otros archivos del controlador, la funcionalidad de esa función es brindar una respuesta http a través de un JSON, este se usa principalmente en adminTools.js y phrases.js

La carpeta routes contiene un solo archivo llamado index.js ahí se contiene la configuración de las rutas para los llamados HTTP que le realizan a la API y así este pueda redireccionar la data de la solicitud al controlador correspondiente.

La estructura de las rutas se ve de la siguiente manera:

```
module.exports= (app) =>{
  const express = require('express')
  const email_controller = require('../controllers/email')
  const adminTools_controller = require('../controllers/adminTools')
  const phrases_controller = require('../controllers/phrases')
  const coachingTools_controller = require('../controllers/coachingTools')
  const api_routes = express.Router()
  const email_routes = express.Router()
  const adminTools_routes = express.Router()
  const phrases_routes = express.Router()
  const coach_routes = express.Router()
  api_routes.use('/mail', email_routes)
```

```

    email_routes.get('/',email_controller.getMensaje)
    email_routes.post('/:uEmail',email_controller.saveEmail)
    email_routes.get('/tst/:dato', email_controller.getTEST)
api_routes.use('/reminder', email_routes)
    email_routes.post('/:metsId/:msnId', email_controller.setReminder)

api_routes.use('/admincrud', adminTools_routes)
    adminTools_routes.get('/', adminTools_controller.getAllUsers)
    adminTools_routes.get('/:idType/:usuId', adminTools_controller.getUser)
    adminTools_routes.post('/', adminTools_controller.createUser)
    adminTools_routes.put('/:usuId', adminTools_controller.updateUser)
    adminTools_routes.delete('/:usuId', adminTools_controller.deleteUser)

api_routes.use('/frases', phrases_routes)
    phrases_routes.get('/',phrases_controller.getPhrase)
    phrases_routes.post('/:listId',phrases_controller.createPhrase)
    phrases_routes.put('/:listId',phrases_controller.updatePhrase)
    phrases_routes.delete('/:listId',phrases_controller.deletePhrase)

api_routes.use('/coach', coach_routes)
    coach_routes.get('/gifinit/:uId', coachingTools_controller.sendAvatarInit)
    coach_routes.get('/sendAvatarBien/:uId',
coachingTools_controller.sendAvatarBien)
    coach_routes.get('/sendAvatarSad/:uId',
coachingTools_controller.sendAvatarSad)
    coach_routes.get('/sendAvatarAngry/:uId',
coachingTools_controller.sendAvatarAngry)
    coach_routes.get('/sendAvatarMad/:uId',
coachingTools_controller.sendAvatarMad)
    coach_routes.get('/infoReto/:metsId', coachingTools_controller.infoReto)
    coach_routes.get('/consultaReto/:metsId',
coachingTools_controller.consultaReto)
    coach_routes.get('/consultaProgresoHoy/:metsId',
coachingTools_controller.consultaProgresoHoy)
    coach_routes.post('/insertMotivo/:metsId/',
coachingTools_controller.insertMotivo)
    app.use('/api',api_routes)
}

```

Donde:

```
module.exports= (app) => {}
```

Representa una función anónima de JavaScript la cual se le pasa como parámetro un objeto llamado app el cual es un objeto que contiene funciones de la librería Express para el manejo de la comunicación HTTP con la API

```
const express = require('express');
```

La constante express contiene funciones de la librería Express sin modificar como en app, lo cual permitirá hacer uso de sus funciones para declarar objetos rutas con las que cuenta la librería.

```
const email_controller = require('../controllers/email');
const adminTools_controller = require('../controllers/adminTools');
const phrases_controller = require('../controllers/phrases');
const coachingTools_controller = require('../controllers/coachingTools');
```

Estas constantes harán referencia a las funciones que contienen los controladores en la carpeta de “controllers”, cabe mencionar que en los controladores las funciones que manejan la información deben ser exportadas para que puedan usarse desde estas constantes que al igual que la constante express ahora son objetos constantes que pueden hacer uso de las funciones de otros archivos.

```
const api_routes = express.Router();
const email_routes = express.Router();
const adminTools_routes = express.Router();
const phrases_routes = express.Router();
const coach_routes = express.Router();
const coach_msg = express.Router();
```

Este conjunto de constantes son las rutas personales de cada tipo de controlador, como podemos ver ya hacemos uso de la constante express la cual a través de su función “Router()” esto regresa un objeto “ruta” o “router” de express la cual será declarada de forma diferente en cada una de estas constantes para que cada una tenga su objeto ruta personal y así pueda hacer referencia a su propio controlador por separado. Esto funciona para que el soporte a varias peticiones se pueda manejar de manera adecuada. La constante “api_routes” será aquella con la que declararemos posteriormente la ruta de acceso general a la API y los demás serán un anexo a la ruta de forma personal.

```
app.use('/api', api_routes);
```

Siendo así está la ruta principal para acceder a la API la cual se vería así: “<http://server:5002/api>” en este caso al ser un servidor propio de ubisalud la ruta inicial de acceso a la api es “<http://ubisalud.cs.buap.mx:5002/api>”

Cabe mencionar que en los objetos rutas existe la función “use” y las funciones de llamado “get, post, put, delete” que hacen referencia al CRUD.

La función “use” recibe como parámetro un String que contiene la nueva concatenación en la ruta, es decir la dirección o link y un objeto ruta al cual se le heredarán las funciones del CRUD.

Las funciones CRUD reciben como parámetro un String que contiene la nueva concatenación en la ruta, es decir la dirección o link y un objeto con la función a ejecutar cuando esta ruta se active.

```
api_routes.use('/mail', email_routes)
    email_routes.get('/',email_controller.getMensaje)
    email_routes.post('/:uEmail',email_controller.saveEmail)
    email_routes.get('/tst/:dato', email_controller.getTEST)
api_routes.use('/reminder', email_routes)
    email_routes.post('/:metsId/:msnId', email_controller.setReminder)
```

Este bloque contiene la declaración de las rutas que acceden a las funciones de “email_controller”, primero anexamos a la ruta principal el nuevo bloque con “/mail” y luego con “/reminder” para crear una ruta que se ve así “<http://server:5002/api/mail>” y “<http://server:5002/api/reminder>”; entonces es donde declaramos las rutas de los controladores a través de email.routes y su función de tipo de llamado que son get y post junto con el llamado al controlador y cuál función del controlador.

```
api_routes.use('/admincrud', adminTools_routes)
    adminTools_routes.get('/', adminTools_controller.getAllUsers)
    adminTools_routes.get('/:idType/:usuId', adminTools_controller.getUser)
    adminTools_routes.post('/', adminTools_controller.createUser)
    adminTools_routes.put('/:usuId', adminTools_controller.updateUser)
    adminTools_routes.delete('/:usuId', adminTools_controller.deleteUser)
```

Este bloque contiene la declaración de las rutas que acceden a las funciones de “adminTools_routes”, primero anexamos a la ruta principal el nuevo bloque con “/admincrud” para crear una ruta que se ve así “<http://server:5002/api/admincrud>”; Entonces es donde declaramos las rutas de los controladores a través de “adminTools_controller ”y su función de tipo de llamado que son get, post, put y delete creando un CRUD junto con el llamado al controlador y su función correspondiente.

```
api_routes.use('/frases', phrases_routes)
    phrases_routes.get('/',phrases_controller.getPhrase)
    phrases_routes.post('/:listId',phrases_controller.createPhrase)
    phrases_routes.put('/:listId',phrases_controller.updatePhrase)
    phrases_routes.delete('/:listId',phrases_controller.deletePhrase)
```

Este bloque contiene la declaración de las rutas que acceden a las funciones de “phrases_routes”, primero anexamos a la ruta principal el nuevo bloque con “/frases” para crear una ruta que se ve así “<http://server:5002/api/frases>”; Entonces es donde declaramos las rutas de los controladores a través de “phrases_controller” y su función de tipo de llamado que son get, post, put y delete creando un CRUD junto con el llamado al controlador y su función correspondiente.

```
api_routes.use('/coach', coach_routes)
    coach_routes.get('/gifinit/:uId', coachingTools_controller.sendAvatarInit)
    coach_routes.get('/sendAvatarBien/:uId',
coachingTools_controller.sendAvatarBien)
```

```

    coach_routes.get('/sendAvatarSad/:uId',
coachingTools_controller.sendAvatarSad)
    coach_routes.get('/sendAvatarAngry/:uId',
coachingTools_controller.sendAvatarAngry)
    coach_routes.get('/sendAvatarMad/:uId',
coachingTools_controller.sendAvatarMad)
    coach_routes.get('/infoReto/:metsId', coachingTools_controller.infoReto)
    coach_routes.get('/consultaReto/:metsId',
coachingTools_controller.consultaReto)
    coach_routes.get('/consultaProgresoHoy/:metsId',
coachingTools_controller.consultaProgresoHoy)
    coach_routes.post('/insertMotivo/:metsId/',
coachingTools_controller.insertMotivo)

```

Este bloque contiene la declaración de las rutas que acceden a las funciones de “coach_routes”, primero anexamos a la ruta principal el nuevo bloque con “/coach” para crear una ruta que se ve así “<http://server:5002/api/coach>”; Entonces es donde declaramos las rutas de los controladores a través de “coachingTools_controller” y su función de tipo de llamado que son get, post ya que son accesos de lectura y escritura a las bases de datos, esto junto con el llamado al controlador y su función correspondiente.

La carpeta models contiene los archivos que dan la estructura de cómo se guardará la información en la base de datos propia de Metsbot siguiendo una estructura NOSQL, hecha con MongoDB precisamente.

La creación de un usuario de Metsbot se ve así:

```

const mongoose = require('mongoose')
const Schema = mongoose.Schema
const motivos= new Schema({
  date: {
    type: Date,
    default: Date.now
  },
  description: {
    type: String
  }
})

const userSchema = new Schema({
  email: {
    type: String,
    lowercase: true,
    unique: true,
    required: true
  },
  metsId: { type: Number, required: true },
  metsName: { type: String, required: true },
  metaPasos: {
    type: Number},

```

```

    timeReminder: {type: Number},
    avatar: {type: Number},
    msnId: {type: String},
    motives: [motivos]
  })
module.exports = mongoose.model('User', userSchema)

```

Mandamos a llamar la librería de mongoose para utilizar sus funciones como Schema el cual nos proporciona crear un objeto que se puede almacenar en la base de datos. El Usuario en Metsbot contiene email, metsId (el id que tiene el usuario en la base de datos SQL de METS), metsName (nombre del usuario en Facebook), metaPasos (dato obtenido de la base de datos de METS), timeReminder (hora que define el usuario ser recordado), avatar(referencia sobre que avatar es el que usa el usuario en la base de datos de METS), msnId (el id del usuario en Facebook para comunicarse con él), motives (un arreglo donde se almacenan los motivos del usuario).

La creación del arreglo de frases se ve así:

```

const mongoose = require('mongoose')
const Schema = mongoose.Schema
const phraseSchema = new Schema({
  frases: [String]
})
module.exports = mongoose.model('Phrases', phraseSchema)

```

Mandamos a llamar la librería de mongoose para utilizar sus funciones como Schema el cual nos proporciona crear un objeto que se puede almacenar en la base de datos. Las frases se almacenan en un arreglo dentro de la base de datos que solamente es una nueva entrada de tipo String.

La carpeta node_modules contiene todos los archivos con las librerías que hacen funcionar nuestro código, aquellas que fueron descargadas a través del NPM como Mongoose o Express por ejemplo.

El archivo package.json contiene la declaración del proyecto con las librerías que utiliza, esta es un archivo que se crea de forma automática cuando se inicia un nuevo proyecto con Nodejs y funciona para que sin necesidad de tener la carpeta node_modules el comando NPM entienda que archivos necesitará el proyecto.

Por último tenemos el archivo index.js el cual es el encargado que crear el pool de hilos que usará la API dentro del servidor, también conocido como el archivo que crea el servicio o el servidor por ende es el archivo principal del proyecto como en una página web que de forma automática se manda a llamar al index.html como el archivo principal análogamente sucede lo mismo con el archivo index.js

El archivo index.js luce así:

```

const express = require('express');
const app = express();
const port = process.env.PORT || 5002;
const mysql = require('mysql');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({extended:true}));
app.use(bodyParser.json());
mongoose.Promise = global.Promise; // declarando promise

```

```

mongoose.connect("mongodb://localhost:27017/Metsbot"); // mongodb connection
let routes = require("./api/routes/index");
routes(app);
app.listen(port, () => {
  console.log("Ready at: " + port);
});

const express = require('express');
const app = express();
const port = process.env.PORT || 5002;
const mysql = require('mysql');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

```

Primero declaramos las constantes con las que vamos a trabajar y que contienen:

- express que contiene las funciones de la librería express
- app que ya es una función de express
- port el cual buscará si en el servidor está disponible el puesto 5002, de lo contrario tomara otro.
- mysql el cual contiene las funciones de la librería mysql para podernos conectar a la base de datos de METS.
- mongoose el cual contiene las funciones de la librería mongoose y poder hacer uso de la base de datos en Mongodb
- bodyParser el cual ya contiene las funciones de la librería body-parser y así poder manejar la data en cierto formato.

```

app.use(bodyParser.urlencoded({extended:true}));
app.use(bodyParser.json());

```

Como podemos observar la constante app es nuestro objeto de Express el cual modificamos, ya que él contendrá toda la funcionalidad de nuestro api y así mismo la posibilidad de estar conectado para brindar los servicios correspondientes. En este bloque le agregamos los atributos de que usará la librería body-parser para aceptar data con el formato urlencoded y manejarlo todo en formato JSON.

```

mongoose.Promise = global.Promise; // declarando promise
mongoose.connect("mongodb://localhost:27017/Metsbot"); // mongodb connection

```

Le indicamos a nuestra extensión de la librería Mongoose que cree una conexión al motor de la base de datos y a que base de datos va a acceder.

```

let routes = require("./api/routes/index");
routes(app);

```

Ahora declaramos una variable llamada `routes`, el cual es un objeto que contendrá todas las funciones de nuestro archivo de rutas que ya se explicó como se declaró y posteriormente mandamos a llamar esta función anónima la cual ahora está dentro de la variable `routes` y le pasamos nuestra constante `app`.

```
app.listen(port, () => {
  console.log("Ready at: " + port);
});
```

En este bloque ya con nuestra constante `app` lista y configurada usamos su función `listen` para crear el servicio o servidor que esté pendiente de escuchar en el puerto que contenga la constante `port`.

Adicionalmente para que el manejo de la API pueda soportar peticiones paralelas a nivel función con respecto a los llamados de la base de datos de Mets, ya que la API realiza un llamado a la base de datos lo cual provoca un tiempo de espera en lo que se obtiene la data y así poderla mostrar al usuario, para ello se las funciones se declararon como funciones asíncronas que nos permiten hacer uso del `async/await` el cual es una sintaxis especial que tiene JavaScript la cual nos permite manejar los “promises” de otra forma, de tal manera que una función asíncrona, es decir que contiene la palabra “`async`” lo que regresará será una promesa ya sea resuelta o no, adicional a esta sintaxis se utiliza la palabra reservada “`await`” la cual nos permite hacer una pausa a la respuesta de una promesa de tal manera que el código que continúa ese llamado no se ejecuta hasta que la promise haya devuelto algo.

De esta forma en los llamados a la base de datos nuestra librería de SQL que se utiliza en la API crea sus llamados regresando una promesa por ende podemos utilizar dentro de nuestras funciones la sintaxis especial de `async/await` para pausar el flujo del Chatbot hasta que tengamos la data de esa consulta que activo el usuario, posteriormente acomodamos la data en un objeto de tipo JSON y hacemos un “`response`” de tipo JSON a la API de Chatfuel de nombre JSON API.

4.9.7 Ejemplo de la integración de todos los componentes

Chatfuel como plataforma que conecta con Facebook Messenger para poder crear los flujos de diálogos de un Chatbot a su vez también es el intermediario con nuestra API propia (Metsbot API) como se vio anteriormente pero esto se logra gracias a que Chatfuel cuenta con una api propia llamada JSON API como se explicó anteriormente, de tal forma que cuando se declara un llamado externo a Chatfuel, como en este caso a Metsbot api, se declara una instancia del JSON API de Chatfuel para indicarle que tipo de llamado va a hacer (POST, GET) y con qué parámetros. Por lo cual la API a la que se le realiza el llamado debe estar creado de tal forma que reciba los parámetros y pueda responder al llamado con una respuesta en formato JSON, ya que como su nombre lo indica JSON API solo acepta objetos de tipo JSON y con una estructura específica para que se muestre en el chat del usuario una respuesta acorde a lo que permite mostrar Facebook. [59]

Esto lo podremos apreciar en el siguiente flujo de pasos que sucede cuando un usuario quiere conocer cuál es su reto, es decir cuál es la información del reto al cual está inscrito.

- 1) Primero el usuario indica a través del Chatbot que quiere saber información de su reto, por ejemplo: pidiéndolo directamente desde el botón proporcionado en el menú como en la figura 12.



Figura 12. Ventana de chat que muestra las opciones tras que el usuario dijera “Menú”.

- 2) Esto activará el bloque de conversación llamado “pide info reto” el cual se ve en Chatfuel como lo mostrado en la figura 12a y 12b:

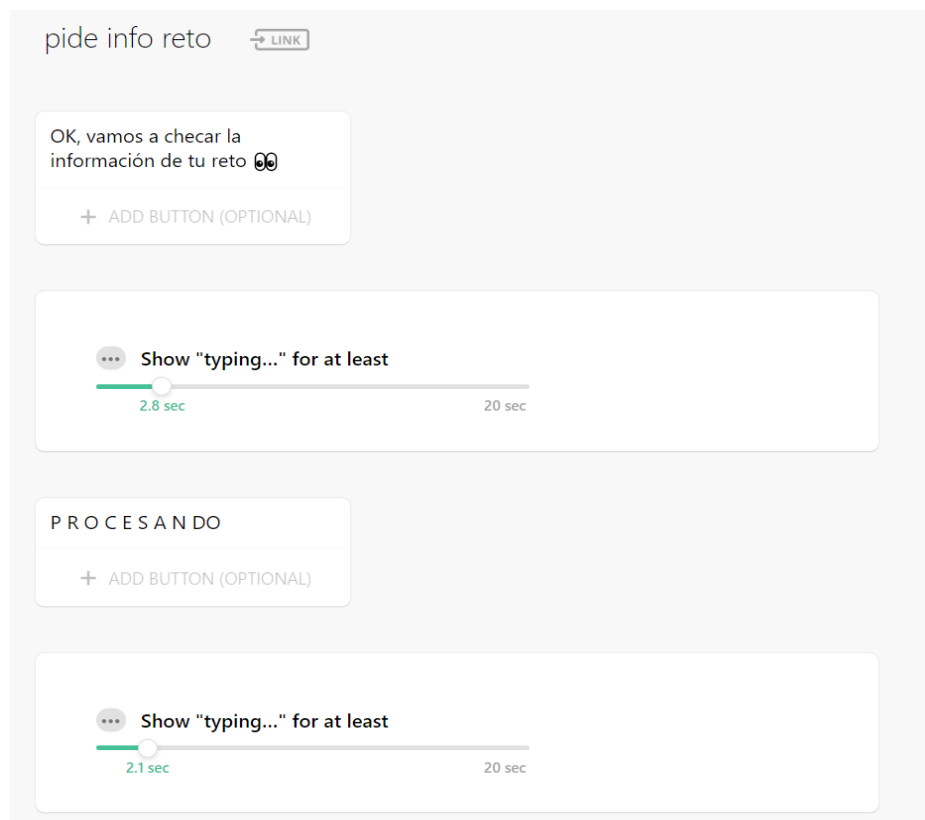


Figura 12a. Ventana de Chatfuel que muestra el bloque de “pide info reto”.

Compuesto por texto y un par de “delays” para que el usuario tome como “normal” que la respuesta no sea inmediata a la hora de realizar la petición.

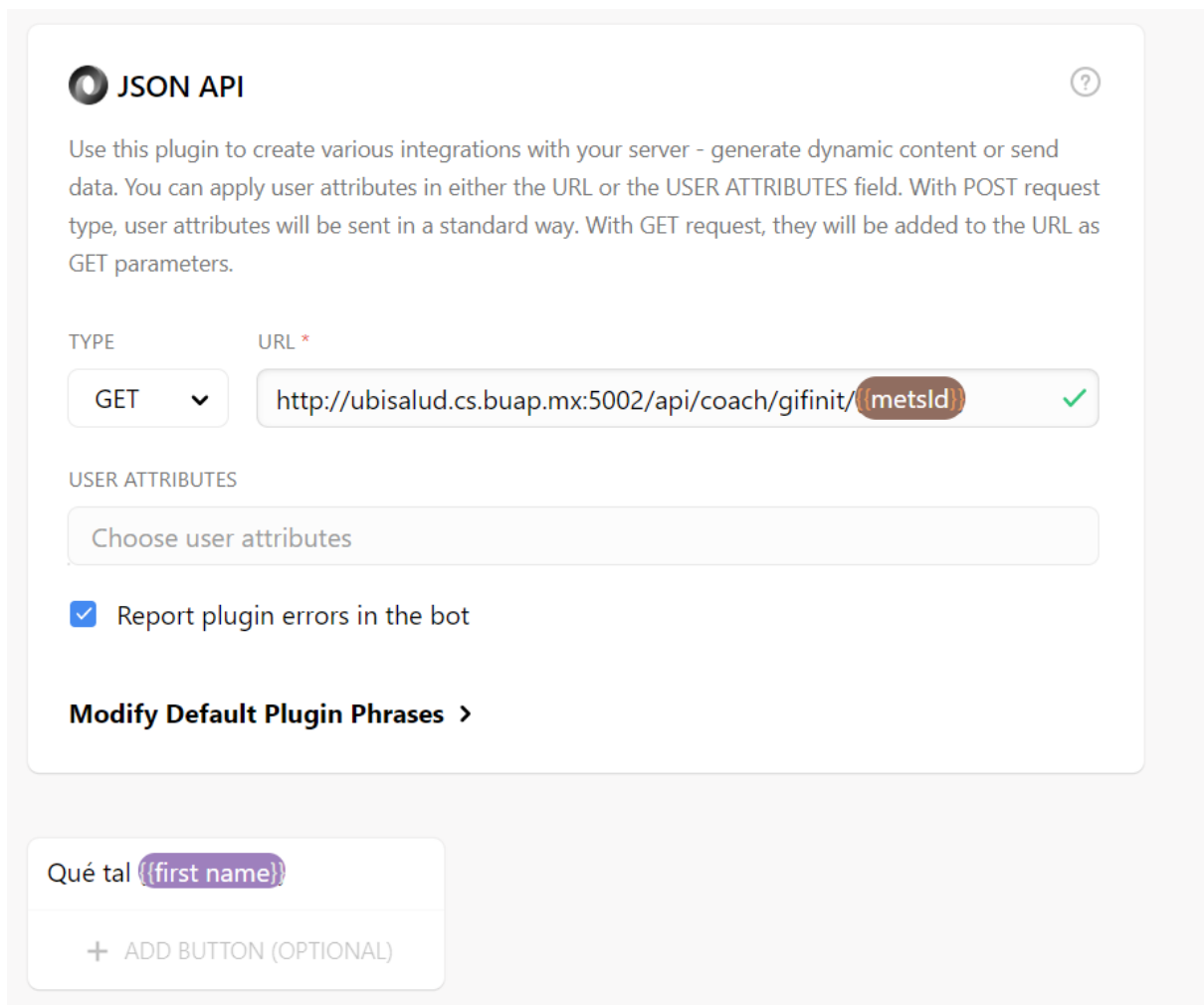


Figura 12b. Ventana de Chatfuel que muestra el resto bloque de “pide info reto”.

- 3) Después se crea la primera instancia de JSON API donde se realiza una petición de tipo GET a nuestra API (Metsbot API) que se encuentra alojada en la dirección: <http://ubisalud.cs.buap.mx:5002/api/coach/gifinit/#####> donde los ##### representan el número que contiene la variable “metsId” que es una variable que anteriormente ya se obtuvo de la base de datos de Mets a través de Metsbot API en un llamado anterior que permitió almacenar en Chatfuel esa variable y ligar con el usuario.

Como podemos ver el parámetro lo estamos mandando como un “Query Param” es decir que lo enviamos con el link a donde se hace el llamado y no desde el body lo cual sería con la sección que dice “USER ATTRIBUTES” esto es debido a que es una URL que no se muestra en navegador y es una variable almacenada por ende no hay tanto riesgo de enviarlo de esta forma, ya que tampoco es información delicada.

Mientras el llamado se ejecuta en nuestra API lo que sí se muestra es el diálogo que contiene “Qué tal {{first_name}}” donde “first_name” es una variable que tiene almacenado el primer nombre del usuario en Facebook (el nombre que tiene en Facebook) de tal forma que si mi usuario en Facebook tiene como nombre Roberto Gómez Bolaños, la variable “first_name” contiene solo la parte de “Roberto” en el String que es el nombre.

Mientras este diálogo se muestra, como bien sabemos ya se hizo un llamado a nuestra API, en la ruta de “/api/coach/gifinit” por lo cual dentro de nuestro api estará entrando aquí:

```
api_routes.use('/coach', coach_routes)
  coach_routes.get('/gifinit/:uId', coachingTools_controller.sendAvatarInit)
```

Donde “:uId” es nuestra variable con el valor numérico del ID del usuario en la base de datos de Mets y se ejecuta la función de nombre “sendAvatarInit” contenida en el controlador de “coachingTools” La función sendAvatarInit se ve así:

```
async function sendAvatarInit(req, res) {
  let metsWebId = req.params.uId
  let gifUrl
  let resbody

  console.log("Check avatar params Mets webId : ", metsWebId)
  try {
    let userData = await User.findOne({ "metsId": metsWebId })
    if (userData) {
      switch (userData.avatar) {
        case 1:
          {
            gifUrl = "https://ubisalud.cs.buap.mx/mets/img/metsbotgifs/1feliz.gif"
            resbody = {
              "messages": [{
                "attachment": {
                  "type": "image",
                  "payload": {
                    "url": gifUrl
                  }
                }
              ]
            } //resbody
            break
          }
        case 2:
          {
            gifUrl = "https://ubisalud.cs.buap.mx/mets/img/metsbotgifs/2feliz.gif"
            resbody = {
              "messages": [{
                "attachment": {
                  "type": "image",
                  "payload": {
```

```

        "url": gifUrl
    }
}
    ]]
} ///resbody
break
}
case 3:
{
    gifUrl
    "https://ubisalud.cs.buap.mx/mets/img/metsbotgifs/3feliz.gif"
    resbody = {
        "messages": [{
            "attachment": {
                "type": "image",
                "payload": {
                    "url": gifUrl
                }
            }
        }
    ]
} ///resbody
break
}
case 4:
{
    gifUrl
    "https://ubisalud.cs.buap.mx/mets/img/metsbotgifs/4feliz.gif"
    resbody = {
        "messages": [{
            "attachment": {
                "type": "image",
                "payload": {
                    "url": gifUrl
                }
            }
        }
    ]
} ///resbody
break
}
default:
{
    console.log("Bad number of GIF")
}
} ///switch
res.json(resbody)

```

```

    } ///if userData exists
  else {
    console.log("userData BAD, is not a user in Metsbot database")
    console.log(userData)
  } ///if user dnt exists

} catch (err) {
  console.log("Hubo un error al conseguir el Avatar ", err)
}
} /// sendAvatarInit

```

Cabe mencionar que esta función no accede a la base de datos de METS sino a la propia base de datos de Metsbot API la cual se encuentra hecha con Mongoddb y por ende resulta más rápido realizar este tipo de consultas y consumiendo menos recursos.

Como podemos ver es una función declarada como “async”, guardamos nuestro uid dentro de una variable de nombre “metsWebId” y como es un valor que viene como parámetro en el link entonces accedemos a ella desde el objeto “req” que es el objeto “require” es decir el objeto que llegó como petición desde JSON API, donde “req.params.uid” es la ruta dentro del objeto donde se encuentra nuestro valor numérico, dentro del requerimiento en la sección de parámetros y con el nombre que le hemos puesto en la ruta.

Una vez con el valor que nos ayudará a hacer el query, entonces procedemos a hacer el query:

```
let userData = await User.findOne({ "metsId": metsWebId })
```

let userData, es una variable que declaramos y le asignamos el valor que retorne el llamado. await, es nuestra palabra reservada para decirle al código que no se siga ejecutando lo siguiente hasta que tengamos una respuesta del promise que le prosigue, en este caso el promise es un query a la base de datos propia de Metsbot.

“User.findOne({ "metsId": metsWebId })”, este es el query que se realiza a la base de datos de Metsbot en Mongoddb donde la estructura de tener una variable “User” y usar su función “findOne” proviene del uso de la librería mongoose dejando el query en Mongoddb como “{ "metsId": metsWebId }” que quiere decir, busca y tráeme el usuario (Solo uno o el primero, por ello se usa el findOne que quiere decir “busca uno”) de la base de datos que contenga en su atributo “metsId” el valor que tiene la variable “metsWebId”, de esta forma buscamos al usuario que corresponde con el usuario que hizo la petición desde Facebook.

Si nuestra variable “userData” es diferente de “undefined” es porque el query si encontró un usuario y lo almaceno en esa variable, por ende ya podemos acceder a su atributo “avatar” el cual contiene el número de avatar correspondiente a ese usuario. Por ende solo usamos un switch que según el número de avatar accederá a la ruta del servidor donde se encuentran las imágenes, almacenamos esa ruta y creamos un objeto con formato tipo JSON:

Suponemos que al usuario le corresponde el avatar numero 4.

```
    gifUrl
"https://ubisalud.cs.buap.mx/mets/img/metsbotgifs/4feliz.gif"
    resbody = {
        "messages": [{
            "attachment": {
                "type": "image",
                "payload": {
                    "url": gifUrl
                }
            }
        ]
    }
```

Este formato en particular es el que menciona la documentación de JSON API para que este se reconozca y se le muestre al usuario como una imagen. Una vez teniendo este objeto utilizaremos nuestro objeto de nombre “res” el cual es un objeto response que viene con la petición a la API así como el objeto require, solo que response se utiliza para indicar a la petición que ya se tiene una respuesta a esa petición, nuestra respuesta será de tipo JSON por eso se usa “res.json” y esa respuesta contiene nuestro objeto JSON. El JSON API recibe esta respuesta y la interpreta, si no es correcta la estructura no mostrará nada, pero si la estructura es correcta le enviará al usuario su avatar correspondiente con el saludo anterior véase en la figura 12c.



Figura 12c. Ventana de Messenger que muestra la respuesta del Chatbot enviada desde la API.

4) Por último en el bloque que de Chatfuel activo que es pide info reto, se crea la segunda y última instancia de la función JSON API en el bloque como se muestra en la figura 12d y por último se hará un redireccionamiento al bloque UpdateReto:

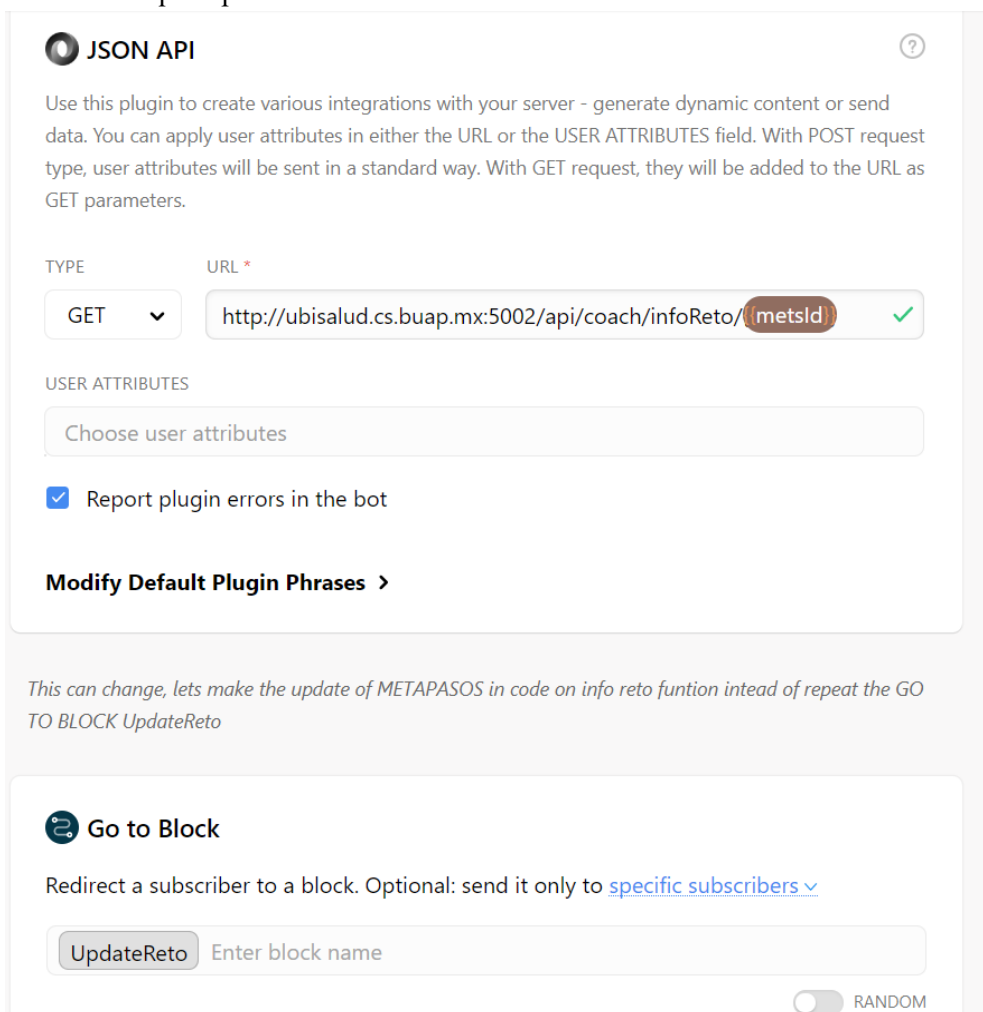


Figura 12d. Ventana de Chatfuel que muestra la parte final del bloque de” pide info reto”.

Este realiza una petición a nuestra API (Metsbot API) de tipo GET en la dirección: <http://ubisalud.cs.buap.mx:5002/api/coach/infoReto/#####> donde “#####” hace referencia a nuestra variable “metsId” que ya sabemos que contiene el id de usuario en la base de datos de Mets.

El llamado se realizó a “api/coach/infoReto” por lo cual llegará a esta ruta de nuestra API:

```
api_routes.use('/coach', coach_routes)
coach_routes.get('/infoReto/:metsId', coachingTools_controller.infoReto)
```

Nuestro valor numérico se queda en el parámetro con el nombre de “metsId” y con ello entrará a función “InfoReto” de nuestro controlador “coachingTools”

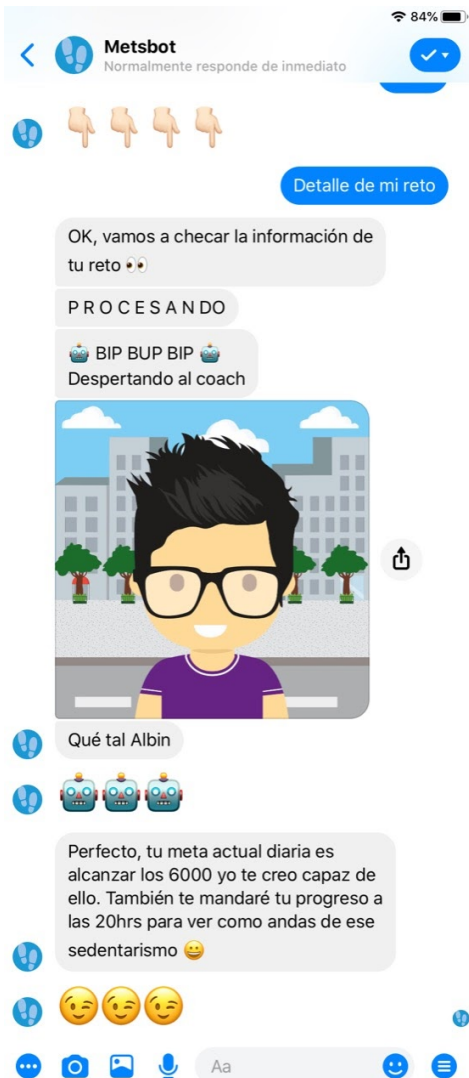
La función InfoReto se ve así:

```
async function infoReto(req, res) {
  console.log("Info Reto")
  let metsUID = req.params.metsId
  let resbody
  let quer = 'SELECT * FROM `bitacoraretosusuario` WHERE (`IdUsuario` =' +
metsUID + ' AND `Status`="En Proceso")'
  await pool.query(quer, function(error, results, fields) {
    if (error) throw error;
    if (results[0] == null) {
      console.log("No se encontró cosas de la bitácora")
      resbody = {
        messages: [{
          text: "Parece que aun no te has inscrito a ningún
reto, ve a tu bitácora en Mets, aquí abajo en el menú está el enlace, inscríbete
a un reto y puedes regresar para configurar tu recordatorio, sólo dime DEFINIR
RECORDATORIO :) "
        }],
        set_attributes: {
          metsId: metsUID,
          infoRetoStatus: 0
        }
      } //resbody
    } else {
      console.log("Se encontró info de la bitácora")
      resbody = {
        messages: [{
          text: "👉 BIP BUP BIP 👉"
        }],
        set_attributes: {
          metsId: metsUID,
          infoRetoStatus: 1,
        }
      }
    }
    res.json(resbody)
  }) //query }
```

Una vez más utilizamos la sintaxis de Async/Await para detener la corrida del código cuando se utilice el objeto “pool” con la función “query” lo cual creará un query que llamara ya no a la base de datos en MongoDB sino a la base de datos SQL de Mets directamente, la query la podemos ver en la variable “quer”, posteriormente vemos que si no hay resultados se crea un objeto con estructura JSON de acuerdo al JSON API para mandar un mensaje, de lo contrario usamos la misma estructura para enviar otro mensaje, lo cual llega como respuesta al JSON API y este mostrará al usuario como un mensaje de texto y activará el bloque de UpdateReto, este bloque ya se sale del flujo principal que se está explicando sin embargo solo es otro

llamado a nuestra api para actualizar en la base de datos propia de la api y así finalmente con los nuevos datos crear el objeto JSON para enviar como respuesta a JSON API con un mensaje compuesto que se visualiza de la siguiente forma:

```
let resbody = {
  messages: [{
    text: "Perfecto, tu meta actual diaria es alcanzar los " + updatedUsu.metaPasos + " yo te creo capaz de ello. También te mandaré tu progreso a las " + updatedUsu.timeReminder + " para ver cómo andas de ese sedentarismo :D"
  }],
  set_attributes: {
    usuEmail: updatedUsu.email,
    metsId: updatedUsu.metsId,
    botID: botID,
    savedReminderStatus: 1
  }
}
```



Donde “updatedUsu” es un objeto con los valores que se modificaron en la base de datos propia de la API la cual contiene en el mensaje la meta de pasos al día que debe lograr el usuario y la hora a la que el usuario indicó que le gustaría ser recordado de su progreso.

La sección del objeto que tiene el nombre “set_attributes” contiene variables con sus respectivos valores, estas variables como “metsId”, “usuEmail”, “botID” y “savedReminderStatus” son variables que llegaran a JSON API y este los almacenará en Chatfuel para usarlas posteriormente.

Al final de este flujo el usuario tiene los datos actualizados de forma interna y recibe en el chat un saludo de su coach y el mensaje correspondiente a la información de su reto como se puede apreciar en la figura 12e.

Figura 12e. Ventana de Messenger que muestra la respuesta al usuario según lo obtenido por la

API

4.9.8 Ejemplo de la integración de todos los componentes para complementar sus limitaciones.

Así con el flujo de conversación anterior ejemplificamos cómo funcionan aspectos del código mencionados como los llamados al JSON API y la funcionalidad asíncrona de la misma. Sin embargo esto no abarca la funcionalidad que tiene el Chatbot cuando se trata de una conversación que inicia el Chatbot, es decir que el mismo envía un mensaje sin necesidad de una interacción del usuario anteriormente de forma inmediata. Un flujo de conversación que se ejecuta sin interacción inicial del usuario es aquella que tiene una tarea automatizada por parte del Chatbot, para ser más específicos en este ejemplo, desde la API.

Como se explicó anteriormente en la descripción del componente de Chatfuel existe esta funcionalidad de las SECUENCIAS que como vimos es un conjunto de bloques que se activan tras realizar cierta acción y que desplegará la funcionalidad de cada bloque o de uno solo tras pasar cierta cantidad de tiempo después de que se activó la secuencia para el usuario específico, siendo así una de las limitantes de la plataforma de Chatfuel que no se mencionó anteriormente es esta funcionalidad de enviar mensajes automáticos de forma recurrente y personalizada. El inconveniente se presenta en el momento donde le damos la libertad al usuario de poder personalizar la hora en la que quiere ser recordado de su progreso para evitar molestia en un caso por ejemplo donde el usuario necesita ser recordado a las 19 hrs porque es su hora de salida del trabajo, no podemos suscribir a una secuencia que exactamente active el bloqueo a las 19hr ya que sería necesario que el usuario se suscribiera a las 19hr y en el flujo de conversación establecido el usuario puede mover entre sus ramificaciones en cualquier momento a cualquier hora.

Una posible solución sería administrar bloques por horas de tal forma que obteniendo la hora del usuario podemos crear un bloque que le suscriba según la hora que eligió, por ejemplo si el usuario se dio de alta a las 14hrs y quiso activar su recordatorio a las 19hr esto activaría un bloque que toma las 14hrs y lo suscribe a una secuencia para ser recordado dentro de 5 horas, esto la primera vez, de tal forma que al recibir el mensaje proveniente de la suscripción activada pasadas las 5 horas le suscribiera a otra secuencia donde ya ahora al ser las 19hr se activará la petición a la API “after 1 Day” es decir al día siguiente a la misma y crear una secuencia circular que mande a la anterior, de esta forma ya el usuario sería recordado a la hora que eligió, sin embargo no solo por la complejidad de la estructura sino por la cantidad de bloques y secuencias alternas a crear por cada hora del día, o del rango de horas seleccionado, con sus diferencias de horas entre la hora elegida más las secuencias circulares saturarían la plataforma de Chatfuel que es otra de sus grandes desventajas actualmente, a mayor número de bloques por cargar, la plataforma pierde “performance” y resulta lento además de engorroso querer navegar entre los bloques, grupos y secuencias ya creados para hacer cambios.

Debido a esa problemática de esa limitante que tiene la plataforma de Chatfuel para nuestro uso de Metsbot se optó por realizar ese proceso desde la API, para ello cuando el usuario está finalizando su registro con el Chatbot o cuando solicita un cambio de hora este realiza un llamado a la API (Metsbot API) a través del JSON API de Chatfuel con el valor de la hora que seleccione el usuario, este se almacena en la base de datos propia de la API y ahí se queda almacenado el dato, la API cuenta con un cronjob proporcionado por la librería ‘node-Schedule’ el cual se activa cada 25 minutos una tarea que consulta a la base de datos propia de la API para obtener a todos los usuarios que coincidan con la hora actual, solo la hora, y posteriormente realiza una query pero ahora a la base de datos de Mets para obtener el progreso de los usuarios y así de forma individual para generar un mensaje dependiendo del progreso y mandarlo de forma personal a cada usuario pero ya no a través de JSON API, sino realizando un POST de forma directa a nuestra app de Facebook previamente creada.

El flujo se está operación dando por entendido que el usuario ya selecciono anteriormente la hora a la que desea ser recordado luce así:

```

var rA = schedule.scheduleJob('25 * * * *', function() {
  console.log("Hora de checar los usuarios con recordatorio")
  console.log("Son las " + getCurrentHour())
  reminderAlert()
});

```

“rA” es una variable que almacena un cronjob, este se crea con la constante ‘schedule’ que contiene las funciones de la librería ‘node-schedule’ y utilizamos en particular la función llamada ‘scheduleJob’ la cual instancia el cronjob por lo cual en sus parámetros pasamos un String con la sintaxis de un cronjob en Unix y un callback el cual ejecutará según se cumpla el cron, cuando esto sucede se manda a llamar a la función “reminderAlert” la cual luce así:

```

async function reminderAlert() {
  try {
    console.log("Entrando a ciclo de reminder Alert")
    let HoraActual = getCurrentHour()
    let usersInHour = await User.find({ "timeReminder": HoraActual })
    if (usersInHour) {
      for (const i in usersInHour) {
        console.log("Usuario #" + i)
        reminderGenerator(usersInHour[i])
      }
    } else
      console.log("Not users in this hour")
  } catch (error) {
    console.log("an Error occur while reminderAlert", error)
  }
}

```

Utilizamos nuevamente la sintaxis de Async/Await, obtenemos la hora actual, solo el valor de la hora porque no nos interesan los minutos, se hace un query a la base de datos en MongoDB usando la función “find” que nos regresa a todos los que cumplan con el parámetro que en este caso es que en su atributo “timeReminder” tengan el mismo valor que la variable “HoraActual”, si es así entonces recorro por usuario y lo envié a la función “reminderGenerator” que luce así:

```

async function reminderGenerator(userToNotify) {
  let metsUIId = userToNotify.metsId
  let resbody
  let quer = 'SELECT * FROM `detalle_diario` WHERE `Status`= 1 AND `IdUsuario`='
+ metsUIId
  console.log("reminderGenerator:", metsUIId)
  if (metsUIId == undefined) {
    console.log("Usuario no registrado en Metsbot")
    throw error
  }
}

```

```

}
try {
  await pool.query(quer, function(error, results, fields) {
    if (error) throw error;
    if (results[0] == null) {
      console.log("No se encontró el cosas de la bitácora")
    } else {
      if (results[0].DiaCumplido == 1) {
        resbody = {
          sender: userToNotify.msnId,
          msg: "Wow!! hoy has logrado hacer: " +
results[0].PasosHechos + " pasos. \n Has cumplido tu reto el día de hoy 🏆 y
vencido el sedentarismo el día de hoy 🙌 ;FELICIDADES! 🏆 Vamos por un día más.
",
          avatarMood: "bien",
          avatarNo: userToNotify.avatar
        }
        generaMsg(resbody)
      } else {
        let porcentajeHecho = (results[0].PasosHechos * 100) /
userToNotify.metaPasos
        if (porcentajeHecho >= 80) {
          resbody = {
            sender: userToNotify.msnId,
            msg: "Hey!, vas bastante bien, llevas " +
results[0].PasosHechos + " pasos👊 \n Un " + porcentajeHecho + "% no te detengas
que estás apunto de completar tu reto de hoy! 😎",
            avatarMood: "bien",
            avatarNo: userToNotify.avatar
          }
          generaMsg(resbody)
        } else if (porcentajeHecho > 50) {
          resbody = {
            sender: userToNotify.msnId,
            msg: "Ya es la hora y vas bien, ya pasaste la
mitad del reto 😎 \n lo difícil ya pasó, no es hora de detenerse, sigue así.",
            avatarMood: "gifinit",
            avatarNo: userToNotify.avatar
          }
          generaMsg(resbody)
        } else if (porcentajeHecho == 50) {
          resbody = {
            sender: userToNotify.msnId,

```

```

        msg: " Qué curioso, estás justo a la mitad del
reto. NO ES HORA DE DETENERSE, la otra mitad es más fácil.💪😄 ",
        avatarMood: "none",
        avatarNo: userToNotify.avatar
    }
    generaMsg(resbody)
} else if (porcentajeHecho >= 30) {
    resbody = {
        sender: userToNotify.msnId,
        msg: " Ya es hora de checar cuantos pasos llevas y
pues sólo has hecho " + results[0].PasosHechos + " pasos, eso no es ni la mitad
😞 \n Pero bueno aún puedes salir y caminar un poco más para cumplir el reto de
hoy. ",
        avatarMood: "sad",
        avatarNo: userToNotify.avatar
    }
    generaMsg(resbody)
} else {
    resbody = {
        sender: userToNotify.msnId,
        msg: "Chequé tu progreso y me encuentro con que
sólo has hecho " + results[0].PasosHechos + " pasos 🧑 \n 😞 mmm... te
recomiendo que camines más porque te falta mucho para cumplir tu reto de hoy.
CAMINA! 🤔👉 ",
        avatarMood: "bien",
        avatarNo: userToNotify.avatar
    }
    generaMsg(resbody)
}
}
}
})
} catch (err) {
    console.log("Error en consultaProgresoHoy: ", err)
}
}

```

Lo primero es obtener el id correspondiente en la base de datos de Mets, habiendo hecho esto se estructura un String con la sintaxis de un query en SQL anexando este id del usuario, entonces ahora sí a través de nuestro objeto “pool” utilizamos su función “query” para realizar una consulta a la base de datos de Mets el cual nos va a regresar todos los atributos del usuario con ese id de la tabla “detalle_diario”, una vez habiendo obtenido estos datos verificamos que el atributo “DiaCumplido” sea diferente de ‘1’, ya que si es ‘1’ esto quiere decir que ya cumplió su meta del día y por ende se manda un mensaje al respecto a través de la función “generaMsg”. De lo contrario sacamos el porcentaje haciendo un cálculo con el atributo “PasosHechos” multiplicado por ‘100’ y dividiendo ese resultado entre el atributo ‘metaPasos’ (he aquí la importancia de que

se actualicen los datos como en el flujo anterior donde el usuario consulta la información de su reto). Una vez habiendo obtenido el porcentaje a través de un “if anidado” se envía un mensaje correspondiente al porcentaje obtenido (si es mayor o igual que 80, si es mayor que 50, si es exactamente 50, si es mayor o igual que 30, menor que 30) para alentar, felicitar o mostrarse decepcionado. En todos los casos armamos un objeto llamado “resbody” el cual contiene los atributos:

- sender: el cual contiene el “id” de Facebook del usuario para enviarle el mensaje al usuario correspondiente.
- msg: el cual contiene el mensaje a mostrar.
- avatarMood: que contiene un string indicando que tipo de avatar corresponde al mensaje.
- avatarNo: el cual contiene el número de avatar correspondiente al usuario.

Este objeto “resbody” se envía a la función “generaMsg”:

```
function generaMsg(resbody) {
  messageData = {
    "text": resbody.msg
  }
  sendRequest(resbody.sender, messageData)
}
```

Donde se crea una estructura de tipo JSON en el objeto “messageData” como indica la estructura de respuesta en la documentación de Facebook Developers, posteriormente se envía ese “messageData” y el “id” a la función sendRequest:

```
function sendRequest(sender, messageData) {
  /// sendGif(resbody.avatarMood, resbody.avatarNo)
  request({
    url: 'https://graph.facebook.com/v2.6/me/messages',
    qs: { access_token: token },
    method: 'POST',
    json: {
      recipient: { id: sender },
      message: messageData,
    }
  }, function(error, response, body) {
    if (error) {
      console.log('Error sending messages: ', error)
    } else if (response.body.error) {
      console.log('Error: ', response.body.error)
    }
  })
}
```

Lo primero a destacar de esta función es que se encuentra comentado una función llamada “sendGif” el cual recibe como parámetro los atributos del avatar correspondiente al usuario y generaba otro “messageData” con la estructura de enviar una imagen, sin embargo se optó por comentarlo, ya que durante la fase inicial del Chatbot resultaba tedioso que los avatares con su expresión se mostrarán cuando el usuario consulta su

progreso y que esto se viera repetido cuando el Chatbot te enviaba mensaje lo cual provocaba ver una conversación saturada de imágenes, así que se optó por descartar al menos de forma momentánea esa función siendo que si es requerida o reinventada en algún otro momento solo hay que quitarle lo comentado y volver a enviar el “resbody” como parámetro.

Continuando utilizamos la función “request” de la librería “request” la cual nos permite hacer peticiones http y https desde nuestra API, en este caso le haremos un POST a la URL de Facebook con nuestro TOKEN de acceso personal de la página que hemos generado en la aplicación de Facebook desde el portal de Facebook Developers, construimos un JSON como lo pide la documentación que contenga el “recipient” el “id” del usuario de Facebook al que se le mandará el mensaje y en “message” nuestro “messageData”.

El hecho por el cual la función “sendRequest” está aparte es debido a que esa es la estructura general de cómo enviar un POST para mandar un mensaje desde servidor a un usuario de Facebook donde la única variante el “message” el cual dependiendo de la estructura que se le ponga enviará un tipo diferente de mensaje como una imagen, texto, quick reply, etc.

De forma similar funciona el cronjob alojado en la variable “sA” solo que esta se activa cada 55 minutos.

```
var sA = schedule.scheduleJob('55 * * * *', function() {
  console.log("Hora de checar los usuarios para SEDENTARIO ALERT")
  let hora = getCurrentHour()
  console.log("Son las " + hora)
  if (hora > 12 && hora <= 20) /// definimos que las alertas de sedentarismo no
ocurran de las 20hr a las 12hr
    sedentarioAlert()
  if (hora == 21)
    dayLastReminder()
  if (hora == 23)
    checkDays()
});
```

La función que ejecuta el callback primero obtiene la hora actual y realiza una serie de comparaciones:

- Si la hora actual no ocurre en el intervalo de las 20hrs a las 12hrs entonces empieza a ejecutar cada 55 minutos la función “sedentarioAlert” que como su nombre lo indica hace un recorrido de los usuarios para saber cuántos pasos llevan y si este menor de 100 es muy probable que tenga una actitud sedentaria durante esa hora, por ende se le manda un mensaje incentivando a que camine un poco.
- Si la hora actual son las 22hrs entonces activa la función “dayLastReminder” el cual realiza un query por usuario y revisa la actividad de las personas si aún no cumplen su reto del día actual les avisa que el día esta por acabar y que aún pueden lograr su meta.
- Si la hora actual son las 23hrs se activa la función “checkDays”, esta función realiza un query por usuario y determina cuántos días lleva sin cumplir su reto, de tal forma que si lleva casi un mes sin cumplir su reto se le notifica que puede reponerse de lo contrario será dado de baja de la aplicación, si se detecta que el usuario lleva 30 días o más sin cumplir su reto se le notifica al usuario que será dado de baja y que utilizar el Chatbot y Mets cuando desee volver, después de ello procedió a eliminar al usuario de ambas base de datos.

4.10 Limitantes de Metsbot

Metsbot como Chatbot y a través de los componentes que le conforman presenta una serie de limitantes que no afectan su funcionamiento sin embargo si crea ciertas dificultades para su

mantenimiento o incluso crecimiento, por el momento y de acuerdo a las necesidades del proyecto y de la tecnología empleada se cuenta por el momento con las siguientes limitantes:

- **Conectividad con METS:**

Como bien hemos podido observar cómo es que se encuentran todos los componentes que conforman a Metsbot pudimos observar que el componente respecto a la API la cual no almacena la todos los datos que muestra al usuario, más bien funcionan como un intermediario entre la base de datos de METS y el componente de Chatfuel donde realiza la consulta a la base de datos directamente para después solo manipular los datos, siendo así que solo realiza meramente lecturas en su mayoría. Todo esto es realizado gracias a un usuario creado para tener ciertos accesos a la base datos, sin embargo, resulta una limitante para Metsbot en caso de que se requiera obtener más permisos que puedan ir acorde a sus necesidades con algún crecimiento del Chatbot. Aunado a esto también se encuentra la dependencia de que la plataforma de METS funcione correctamente, ya que si por alguna actualización o alguna otra situación METS al tener todos los permisos pudiese modificar estructuras o datos de misma base datos entonces eso causaría problemas en las peticiones que se realicen, ya que podría ser data equivocada o que los nombres a buscar en las consultas no existan. Actualmente existen rutinas que pueden responder en caso de que alguno de estos casos suceda con los errores posibles al realizar una consulta, sin embargo, de ser algo como lo que se mencionó anteriormente se necesitaría realizar cambios en el código de la API y volver a colocarlo en funcionamiento.

- **Estabilidad del servidor:**

Tanto la plataforma de Mets como Metsbot se encuentran alojados en un equipo de cómputo servidor que funciona bajo el sistema operativo de Linux al igual que otras aplicaciones que se encuentran con acceso a internet tras una red universitaria. La configuración del equipo como de la red tienen seguridad y rutinas para iniciar nuevamente en caso de cualquier inconveniente, sin embargo no se puede garantizar el acceso continuo a ambas plataformas porque la red puede tener inconsistencias o incluso el suministro de electricidad lo cual ha sucedido en otras ocasiones y esto presenta una limitante para Metsbot en particular, ya que la API que se encuentra alojada ahí no cuenta con los permisos necesarios para levantarse nuevamente por lo cual para ponerlo en línea nuevamente y permitir el uso correcto del Chatbot tendría que realizarse de forma manual desde la consola ya sea en el equipo o tras una conexión remota por SSH, sin embargo no hay forma de tener notificación alguna cuando esto suceda más que con el constante monitoreo de la API a través de sus “log”.

- **Chatfuel y la versión Premium:**

El componente de Chatfuel está compuesto por una plataforma de creación de Chatbots del mismo nombre, el limitante que se tiene por este lado es que la plataforma funciona bajo un esquema de Freemium y Premium donde te permite tener Chatbots de forma gratuita sin embargo esto viene con su respectivas limitantes para el cual Metsbot resulta afectado porque la plataforma deja de funcionar de forma correcta después de tener 5,000 usuarios, ya que el Chatbot solo funcionara de forma correcta para este número de usuarios, los siguientes verán limitado el uso. Por otra parte, todo nuevo usuario recibe un mensaje de que Metsbot está creado con la plataforma de Chatfuel, esto sucede por utilizar la versión gratuita en la cual la misma plataforma se promociona a través de los mismos Chatbots, con la versión Premium o

Pro, como se ve en la imagen. En cuanto al resto de las funciones de paga extra que ofrecen no son importantes para Metsbot al menos no en esta versión.

Free plan	Pro plan	Premium account
Basic functionality to get started	Most powerful features to level up with	Premium service and expert guidance to get results fast
\$0	\$15 _{/mo}	Let's talk
Current Plan	Upgrade To PRO	Contact Us
<input type="checkbox"/> I accept Terms of Service	<input checked="" type="checkbox"/> I accept Terms of Service	
Basic Features Up to 5,000 Subscribers Chatfuel Branding	Full-Featured Bot No Subscriber Limit Remove Branding Precise Targeting with FB Ads Multiple Teammates Priority Support	Full-Featured Bot No Subscriber Limit Remove Branding Precise Targeting with FB Ads Multiple Teammates VIP Support Synced Cloning New

Figura 13. Ventana de Chatfuel que muestra los precios y beneficios de su plataforma.

- **Sistema no supervisado:**

También una parte de lo que permite la herramienta de Chatfuel como se mencionó es que permite crear un glosario de palabras en las cuales van a ingresarse en el sistema de procesamiento de lenguaje natural que tiene el cual toma ese glosario e intuye formas similares de esas palabras y expresiones en el glosario. La misma plataforma guarda las palabras y oraciones a las cuales no pudo darles una respuesta definida fuera de bloque “Default” el cual es el que aparece cuando el Chatbot “no entiende”, esta lista de palabras y oraciones que no entendió el Chatbot necesitan un contexto el cual es una ligadura a cualquiera de los bloques ya declarados para ser parte de un flujo de la conversación, existen sistemas de PLN que toman esta “lista” y la reinsertan de forma automática a lo cual se le dice que cuenta con un sistema no supervisado, ya que todo lo realiza de forma automática, cuando es semi-supervisado gran parte de esa lista se insertan pero aquellas que no puede lo realizara alguien más de forma manual y un sistema supervisado es donde toda la lista se agrega de forma manual el cual es el caso de Metsbot con Chatfuel y esto no presenta tanto una limitante para el Chatbot siempre y cuando se mantenga un constante monitoreo de la plataforma de Chatfuel.

- **Actualizaciones de las políticas de Facebook:**

Desde el año 2018 Facebook ha estado en constante actualizaciones con respecto a la seguridad, acceso y manipulación de la información de los usuarios que utilizan la red social, esto debido a los sucesos con Cambridge Analítica por lo cuales se les ha estado presionando a realizar estos cambios y como menciono al principio ha sido de forma constante por lo cual un gran número de aplicaciones, Chatbots, juegos y demás se ven forzados a estar realizando adaptaciones de código e incluso presentar documentación oficial para que Facebook pueda acceder ciertos privilegios a estas mismas según la categoría a la que pertenezca la aplicación de acuerdo a sus lineamientos. Metsbot claramente no es la excepción y contantemente pierde ciertos permisos que permiten por ejemplo el enviar las alertas de sedentarismo, ya que para

ello se necesita permiso de poder enviar mensajes a los “identificadores” de los usuarios en Messenger de Facebook, para ello era necesario realizar cambios a nivel código y enviar información acerca de ¿quién esta detrás del proyecto?, ¿para qué sirve? y ¿Qué propósito tiene?, agregar scripts en el servidor para validar el dominio, además de credencial de estudiante y matrícula para que nos brindaran una categoría de “medios de comunicación” con la cual esta alerta puede pasar como un mensaje que se puede enviar en lo que existen nuevas categorías en las que Mets sea más ad hoc como aplicaciones para la salud por ejemplo.

4.11 Actualizaciones

Parte del ciclo de vida que tiene cualquier software es el mantenimiento en el cual también se ve incluida la planificación e implementación de actualizaciones, Metsbot como plataforma y sus componentes no son la excepción. El Chatbot ya se encuentra en una fase de producción, está en línea y con usuarios que lo utilizan por ende Metsbot ya se encuentra en una fase de mantenimiento la cual se seguirá planificando conforme la cantidad de usuarios este incrementando, por el momento estas actualizaciones son las siguientes:

- Alimentar el sistema de PLN que tiene Chatfuel por medio del glosario para mantener de manera más efectiva las conversaciones.
- Actualización constante de los links de descarga para la aplicación de Mets, ya que por el momento solo se cuenta una aplicación alojada en el servidor para Android, pero no todavía en la Google Play store, por otro lado las aplicaciones para dispositivos con iOS no funcionan de esa manera, no se pueden descargar e instalar de otra parte que no sea la App Store oficial de Apple, por eso es que se tiene un flujo para avisar a los usuarios que usan iPhone cuando la app esté lista, esto cuenta como una actualización porque cuando las aplicaciones están en las tiendas oficiales se tendrá que eliminar dicho flujo y redirigir a la descarga de la aplicación oficial.
- La siguiente actualización está encaminada a los permisos con la base de datos para poder crear consultas que puedan realizar inserciones y eliminaciones directamente en la base de datos.
- La creación de un software Front-end para los Endpoints que existen dentro de la API destinada a la gestión de la base de datos de Metsbot como administrador.
- Mantener actualizado los permisos de acceso a la API de Facebook Messenger. Como se mencionó anteriormente Facebook desde 2018 ha realizado una serie de actualizaciones periódicas para restringir el acceso a la información que ellos proveen, una de estas actualizaciones que se realizó en Noviembre de 2019 indica que se proporcionará acceso a los identificadores de Messenger de los usuarios para mandar mensajes a estos mismos por medio de una suscripción, es decir el envío constante de mensajes y esto solo será posible si la entidad, en este caso la página de Facebook, que envía los mensajes está registrada como un medio de noticias, lo cual implica que hay que pedir ciertos accesos para METS y realizar las configuraciones en el servidor que pidan los requerimientos de las nuevas normas de Facebook. Actualmente desde finales del 2019 se ha proporcionado la información correspondiente e incluso se realizaron instalaciones en el servidor para que ellos puedan validar la información del dominio, sin embargo por alguna inconsistencia con su plataforma

se indica que ya se pasó el proceso de validación de la página de Facebook, sin embargo después de unos minutos el permiso es revocado y pidiendo nuevamente a que se vuelva a realizar este proceso, la idea es que Metsbot obviamente no es un medio de noticias, sin embargo por el momento es la única categoría a la cual se le da ese permiso, esperamos que pronto existan nuevas categorías con esos mismos permisos para volver a mandar una solicitud.

Capítulo 5. Resultados

Para poder analizar y comentar respecto a los resultados obtenidos con Metsbot nos basaremos en unas métricas que analizan el performance del Chatbot [72] y sumándoles la actividad reportada por la plataforma de METS, ya que recordando la finalidad de Metsbot es que el asistente virtual (Coach) genere una mayor adherencia al programa que METS ofrece, por ende en la plataforma estos resultados se ven beneficiados en los usuarios que utilicen Metsbot esto podría ser un indicador de que la adherencia si está aumentando. Retomando las métricas que se mencionaron estas servirán para poder medir efectividad en el otro propósito de Metsbot que es acercar a través de otros medios, en este caso redes sociales, en particular Facebook, a las personas con METS, por lo tanto si hay un alcance de personas que no conocían METS pero se han registrado a la plataforma por medio de una primera interacción con el Chatbot esto puede ser un indicador de que esta parte está funcionando, además si en las mismas métricas puede observar que hay una interacción constante entre Chatbot y usuario esto puede significar a su vez que hay una adherencia a la comunicación con el Coach y como el Coach virtual está enfocado a mantener una conversación que les invite a seguir con el programa de METS también se podrían brindar índices de que se está aumentando la adherencia al programa mismo lo cual sería cumplir con la finalidad planteada como solución a la problemática inicial, otras métricas brindadas por la plataforma de Chatfuel nos permitirá conocer cuáles son los flujos que más utilizan los usuarios y de esa forma poder determinar más información como cuantos fueron usuarios que no conocían de Mets, cuantos si mantienen la adherencia al programa, cuantos no entienden, no conocen sobre sedentarismo, entre otras cosas.

Tomaremos a modo de prueba a un usuario de Metsbot para poner a prueba alguna de estas métricas y conocer los resultados que obtuvieron para analizarlos.

Descripción del usuario.

Su nombre es María y ya conocía acerca de Mets, esto se lo hace saber al Chatbot y también indica que ya se ha registrado anteriormente por ende solo se le solicita su correo para validar esa información, se confirma e inicia desde donde tenía su reto.

A continuación se mostrará una serie de imágenes seleccionadas que corresponden a parte de la conversación que la usuario María ha tenido con Metsbot, estas se encuentran agrupadas, cada grupo leído de izquierda a derecha corresponde a un momento de interacción entre María y Metsbot en la tabla 18. También en dichas imágenes se muestra que el resultado que se ha tenido con esta usuario ha sido positivo, ya que inicia su reto y parece llevar un ritmo normal pero por momentos lo deja o falla en sus retos diarios, el reto general lo va haciendo mal sin embargo tras constantes alertas de sedentarismo, los recordatorios diarios vemos que provoca que la usuario constantemente pregunte también por su progreso ya sea del día y también de su reto en general y tras un constante intercambio de mensajes donde hay felicitaciones y regaños podemos ver como poco a poco la usuario va aumentando de reto logrando superar el previo y todos con un ciclo muy similar

donde empieza el reto, le cuesta mantener el nuevo reto pero después de un tiempo se va recuperando y aunque falla al principio se logran avances, de tal forma que cuando ya falta poco para terminar el reto al 100% se nota la mejoría y las felicitaciones del Coach de tal forma que de un momento a otro ya cumplió su reto general e inicia el nuevo. La conversación es bastante larga y constante por ende podemos decir que hay muchas sesiones con el usuario donde interactúan y por ende se convierte en un usuario activo del Chatbot lo cual se pudo traducir a una mayor adherencia al programa de METS, ya que no solo se interactuó con el Chatbot constantemente sino que si estaba avanzando en sus retos logrando un resultado positivo por ambas partes.

a) Bloque de conversación: Inicio del registro en Metsbot

Que tal Maria soy yo, tu coach. ¿Ya me reconoces? Ahora, vamos a definir a qué hora te gustaría que te indique tu progreso diariamente.

Da click <https://m.me/metsbot?ref=WelcomeMessage>

Metsbot
Sitio web de salud y bienestar
Enviar mensaje

¡Hola Maria, soy MetsBot! Como lo dice mi nombre, soy una extensión de la plataforma de METS. Tú, ¿ya te registraste o conoces METS?

Si

Por favor indicame ¿cuál es tu correo electrónico? (el que utilizaste para registrarte en METS) 📧

mcpcelish@gmail.com

Bonito nombre en Mets eh!
👋 HOLA USUARIO: conchita 🥰 jijji por comodidad te seguiré llamando por tu nombre de Facebook

Maria, si así está mejor. Ahora sí continuemos.

Que tal Maria soy yo, tu coach. ¿Ya me reconoces? Ahora, vamos a definir a qué hora te gustaría que te indique tu progreso diariamente.

Perfecto, tu meta actual diaria es alcanzar los 4750 yo te creo capaz de ello. También te mandaré tu progreso a las 16 para ver como andas de ese sedentarismo 😊

b) Bloque de conversación: Ver el progreso del día (caso negativo y caso positivo posterior)

Progreso HOY

Vamos a ver cómo estás el día de hoy Maria 👁️

Bien, analizando tu información tu reto es lograr 4750 al día, déjame ver...

Maria no puedo creer que sólo laves 723. Eso es menos del 40%, por eso me molesté un poco. Te sugiero que camines más.

Ver progreso de hoy

Vamos a ver cómo estás el día de hoy Maria 👁️

Wow!! hoy has logrado hacer: 5694 pasos.

Has cumplido tu reto el día de hoy 🎉 Maria 🎉 Eso quiere decir que hoy no has sido una persona sedentaria y tu salud esta 🥰
¡FELICIDADES! 🏆
Vamos por un día más.

c) Bloque de conversación: Regaño y felicitación según el progreso

<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Wow Maria te has pasado 25 día con el reto de 4750, esta un poco mal pero te felicito por no abandonar. Hoy puedes vencer el reto de los 4750 pasos, tu puedes 😊</p> <p>28/09/19 10:01</p>	<p>Maria no puedo creer que sólo lleves 1009. Eso es menos del 40%, por eso me molesté un poco. Te sugiero que camines más.</p> <p>No te enojos</p>
<p>Ya es la hora y vas bien, ya pasaste la mitad del reto 🤓 lo difícil ya pasó, no es hora de detenerse, sigue así.</p>	<p>WOW! bastante bien Maria 🏃 llevas un 83.12 % con 3948 pasos YA! Te falta muy poco, tú puedes, lo difícil ya pasó 🤓 sigue así para completar el reto de hoy 🤓</p>

d) Bloque de conversación: Progreso gradual con altos y bajos, tomado de varios días.

<p>Oops, ALERTA DE SENDERISMO! Sólo has logrado 1 pasos en las últimas 2horas, te recomiendo que vayas a dar una vuelta por ahí</p> <p>Gracias</p> <p>Gracias a ti.</p>	<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Felicidades Maria hoy estas iniciando un nuevo reto. El reto de lograr 5500 pasos al día. Yo se que tu puedes 🏃</p> <p>29/09/19 13:55</p>	<p>Chequé tu progreso y me encuentro con que sólo haz hecho 55 pasos 🤓 mmm... te recomiendo que camines más porque te falta mucho para cumplir tu reto de hoy. CAMINA! 🏃👉</p>	<p>Maria llevas apenas 2108 pasos. 🤓 todavía no llegas ni a la mitad. Pero bueno aún puedes salir y caminar un poco para terminar el reto de hoy.</p>
<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Muy bien Maria parece que estas iniciado tu reto de 5500 pasos al diarios. Yo se que tu puedes, recuerda que debes cumplirlo por 5 días. ¡Éxito!</p>	<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Maria no has caminado todos los dias como deberias. Sinceramente eso me enfurece un poco pero te felicito por no abandonar el reto e intentarlo de nuevo. Por ahora llevas el 20.00 % de tu reto 🏃</p>	<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Vas muy bien, sigue así llevas el 40.00 % de tu reto!!</p>	<p>WOW! bastante bien Maria 🏃 llevas un 86 % con 4760 pasos YA! Te falta muy poco, tú puedes, lo difícil ya pasó 🤓 sigue así para completar el reto de hoy 🤓</p>
<p>Ver progreso de hoy</p> <p>Vamos a ver cómo estás el día de hoy Maria 🏃</p> <p>Wow!! hoy has logrado hacer: 5707 pasos.</p>	<p>Has cumplido tu reto el día de hoy 🏃 Maria 🏃 Eso quiere decir que hoy no has sido una persona sedentaria y tu salud esta 🏃 ¡FELICIDADES! 🏆 Vamos por un día más.</p>	<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p>	<p>Maria no has caminado todos los dias como deberias. Sinceramente eso me enoja un poco pero te felicito por no abandonar el reto e intentarlo de nuevo. Por ahora llevas el 60.00 % de tu reto 🏃🏃</p> <p>Si olvidaste cuanto deberias caminar, puedes preguntarme sobre la información de tu reto.</p>

<p>Oops ⚠️ ALERTA DE SEDENTARISMO ⚠️ Sólo has logrado 0 pasos en las últimas 2 horas, ve a estirar las piernas un poco 🦶🦶</p>	<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Muy bien Maria parece que estas iniciado tu reto de 6250 pasos al diarios. Yo se que tu puedes, recuerda que debes cumplirlo por 5 días. ¡Éxito!</p>	<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Maria me entristece que no hayas cumplido tu retos diario, haz el esfuerzo por cumplirlo hoy</p>	<p>Progreso total reto</p> <p>Es hora de checar tu progreso Maria 🏃</p> <p>Bien, estoy checando tu informacion.</p> <p>Felicidades Maria hoy estas iniciando un nuevo reto. El reto de lograr 7000 pasos al día. 🏃 Yo se que tu puedes</p>
<p>Hola María una pregunta rapida.</p> <p>¿Te he sido de utilidad? Es decir, notas que conmigo has podido cumplir tus retos, algo que tan solo con la app no hubieras logrado. ¿Sientes que has aumentado tu actividad fisica gracias a mi? osea que caminas mas por ejemplo. 🤔</p> <p>Si</p>			

Tabla 18. Tabla con capturas de la conversación entre María y Metsbot, donde se muestran las etapas desde que se registró en Metsbot y como pudo progresar en los retos impuestos por Mets logrando así sus objetivos.

También cabe mencionar que este es el único usuario completo que ha mantenido una comunicación constante con el Chatbot, otros usuarios sirvieron para conocer palabras, posibles preguntas y errores en los flujos de la conversación que podían ocurrir. A algunos se les mencionó que serian usuarios de prueba, a otros se les intentó invitar a METS pero no se concretó el registro por decisión propia de ellos, también algunos usuarios iniciaron conversación con el Chatbot pero al conocer que este no estaba finalizado dejaron de usarlo por ende se les invitó a usar el servicio en el futuro si es que así lo desean. Algunas de estas capturas de pantalla se muestran en la tabla 18a.

Bloques de conversación con diversos usuarios.

Qué tal Mary

🎉 BIP BUP BIP 🎉

Perfecto, tu meta actual diaria es alcanzar los 7000 yo te creo capaz de ello. También te mandaré tu progreso a las 12 para ver como andas de ese sedentarismo 😊

😊😊😊

Menú

👉👉👉👉

Ver progreso de hoy

Vamos a ver cómo estás el día de hoy Mary 🏃

Wow!! hoy has logrado hacer: 6640 pasos.

Has cumplido tu reto el día de hoy 🎉 Mary

🎉 Eso quiere decir que hoy no has sido una persona sedentaria y tu salud esta 🦶

¡FELICIDADES! 🏆

Vamos por un día más.

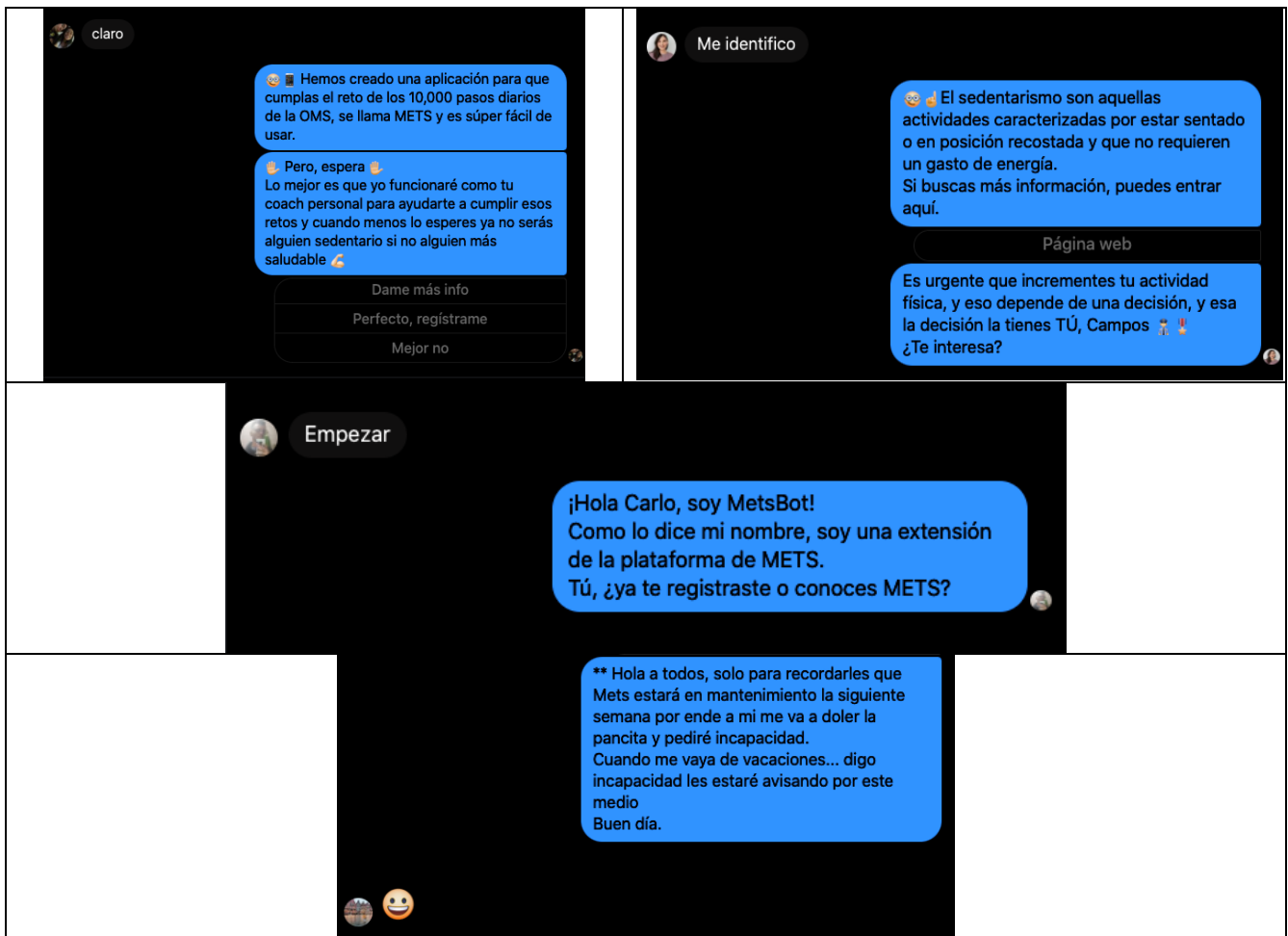


Tabla 18a. Tabla con capturas de la conversación entre usuarios y Metsbot, donde se muestran diversos escenarios.

5.1 Plataforma Interactiva METS

Con base en los estudios sobre actividad física consideramos la caminata como la actividad más accesible y viable para la población objetivo. Elegimos la caminata, ya que existe evidencia importante que relaciona la salud de las personas con la cantidad de pasos que camina al día. Según se plantea en diversas investigaciones las personas que cumplen el reto de caminar alrededor de 10 mil pasos diarios poseen una mejor salud general y una mejor calidad de vida que aquellas personas con poco movimiento durante el día. En METS se propone una serie de retos graduales que el usuario debe vencer hasta alcanzar el hábito de caminar 10,000 pasos diarios. Para situar el nivel del usuario se utiliza el cuestionario RAPA de actividad física que debe realizarse al registrarse en el sistema. De este modo el sistema de recomendación basado en las respuestas del usuario puede ubicarlo en el nivel adecuado para él y asignarle un primer reto de actividad física. El monitoreo constante de los usuarios debe hacerse a través de sensores por lo que asociada a la plataforma web se propone una aplicación móvil que el usuario pueda instalar en su teléfono móvil personal, permitiendo recolectar información acerca de sus actividades diarias: momentos de actividad, tiempo en reposo. De esta manera se espera atender, particularmente a los usuarios registrados recomendando: ejercicios, actividades, artículos sobre la importancia de la actividad física constante, ya que es fundamental fortalecer en el usuario la auto-gestión de su salud lo que redundará en la prevención de enfermedades no transmisibles y sus factores de riesgo. Para realizar la retroalimentación entre el sistema los usuarios contarán con un monitor (coaching) personal que pueda apoyarlo a moldear su estilo de vida a partir de su estado actual. Por otro lado el usuario cuenta con una bitácora en donde puede ver su progreso diariamente, proporcionándole un estímulo para

realizar el reto diario y conseguir más compitiendo con el mismo, de tal manera que busque superar la meta anterior y así controlar los factores de riesgo sin tenerlo presente a lo largo de las actividades el sistema puede consultarse en <http://ubisalud.cs.buap.mx/mets/>.

El usuario de METS debe cumplir una serie de metas progresivas (computación persuasiva) que tienen como objetivo ayudarlo a generar el hábito de caminar diariamente. METS es una plataforma multiusuario, lo cual permite que diversos pacientes interactúen entre sí mediante retos grupales propiciando el cumplimiento de las metas y la modificación del estilo de vida mediante juegos competitivos. La monitorización de la actividad física de los usuarios se lleva a cabo mediante el acelerómetro del teléfono inteligente, que es utilizado como un podómetro (inteligencia ambiental). La codificación del podómetro utilizado en METS se realizó utilizando las librerías y los algoritmos de la aplicación Google Fit, los cuales han sido empleados en estudios clínicos previamente. Dichas librerías y algoritmos se encuentra disponibles de forma gratuita en línea y se pueden utilizar como un soporte para la creación de otras aplicaciones y herramientas web.

El teléfono inteligente, a través de la aplicación, almacena los pasos del paciente cada hora, lo cual permite conocer a detalle las actividades que se realizan durante todo un día. Los datos generados por el paciente son almacenados de forma local por la aplicación y posteriormente son enviados al servidor METS. Esta sincronización se lleva a cabo de forma automática aproximadamente cada 2 horas utilizando un código de alerta incorporado a la aplicación. Adicionalmente, cada vez que el usuario ingresa a la aplicación y se cuenta con conexión a Internet, los pasos son sincronizados automáticamente con el servidor. En caso de que el teléfono no se encuentre conectado a Internet, la aplicación está programada para almacenar los pasos hasta que se cuenta con conexión nuevamente, y en ese momento se realiza la sincronización.

Los datos generados por los usuarios, incluyendo el detalle del número de pasos por hora y por día, así como las horas sin actividad física, son almacenados en el servidor METS. Estos datos pueden ser visualizados por el médico o el investigador a través de Internet (computación ubicua) como se muestra en la figura 14.



Figura 14. Visualización de datos del paciente en Internet a través de Computación Ubicua.

5.2. Resultados seguimiento de uno de los usuarios de Mets

A continuación se presentan es seguimiento del usuario 373. Desde su registro a Metsbot (figura 15 y figura 16); en la tabla 19, marcamos con rojo los retos en los que utilizó la asesoría de su entrenador vía Metsbot cómo puede verse al usar metsbot venció los retos en el tiempo en contraste con los retos en los que no tuvo la asesoría del Metsbot. Por otra parte en la tabla 19, se marca con verde el reto siete que tardó 182 días en cumplir. Cómo se muestra en la gráfica 1, el usuario no cumplía con los 7000 pasos diarios a realizar o bien lo hacía irregularmente, situación que corresponde a la etapa siete del PAP en la que se controlan las situaciones que pueden generar recaídas. Ya que con el apoyo de los mensajes de metsbot (figura 17), el usuario retomó el ritmo y concluyó su reto. Dado lo observado podemos afirmar que con el uso de metsbot los usuarios incrementan su eficiencia un 73%. Es decir pueden cumplir el reto de los 10000 pasos de la OMS en 30 días.

Seguimiento: Estrategia de para incrementar la adherencia Chatbot – Terapia cognitivo conductual



Metsbot

- Mensajes automáticos
- Sitio web de salud y bienestar

A 5 personas les gusta esto, incluidos Rudy Rodríguez y un amigo

Hola, soy Metsbot la IA que sirve para que puedas comunicarte con tu coach personal de la plataforma de METS.

No sabes de lo que estoy hablando?, visita METS <https://ubisalud.cs.buap.mx/mets/>

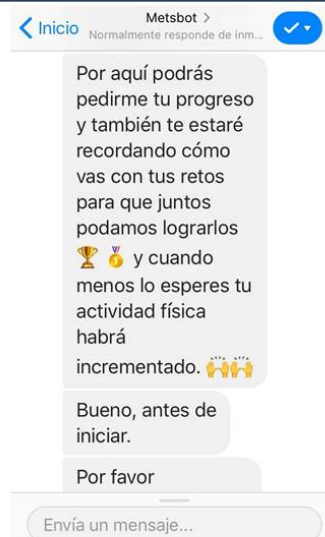


Figura 15. Diálogo inicial del usuario 373 con Metsbot

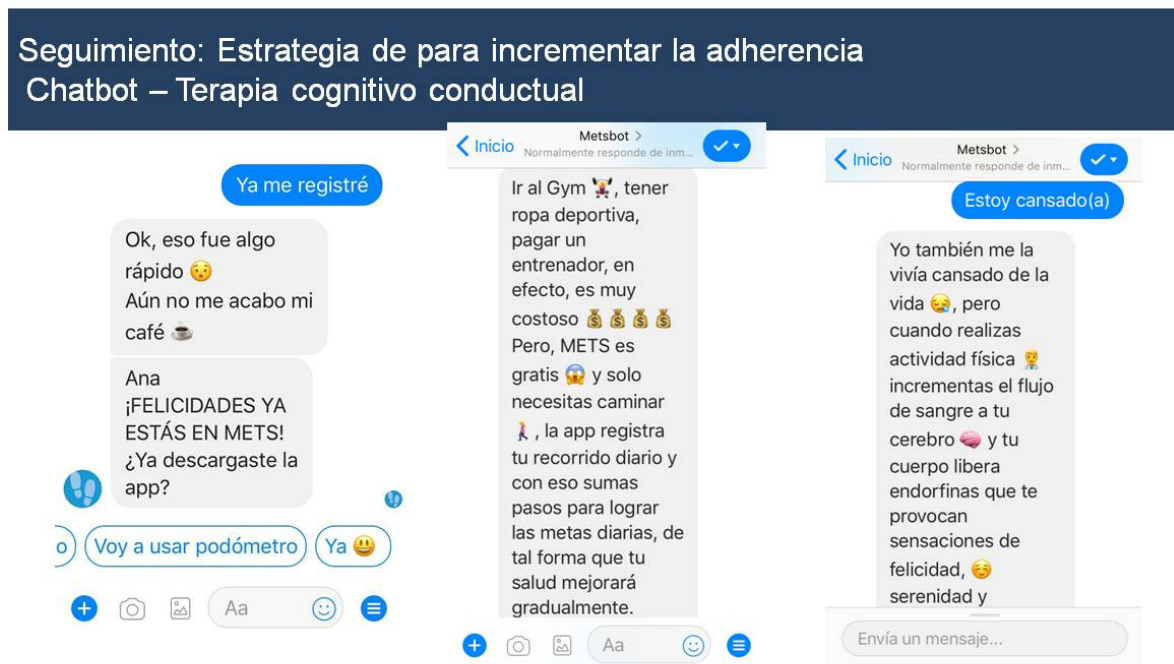
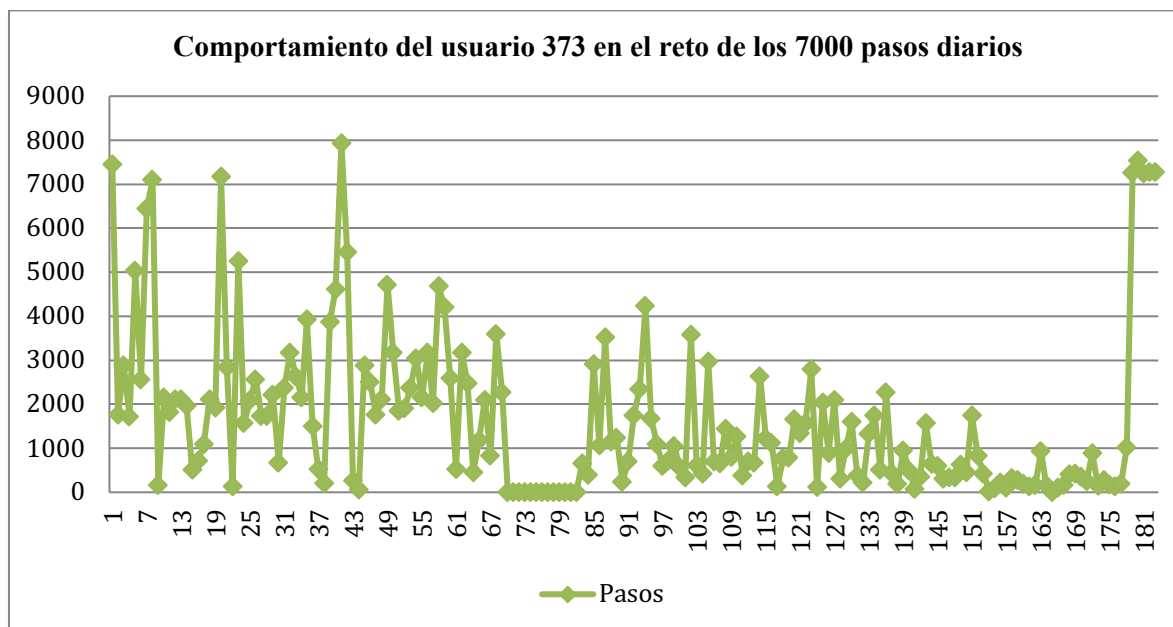


Figura 16. Diálogos motivadores ente Metsbot y el usuario



Gráfica 1. Pasos realizados por el usuario 373, cómo puede observarse si realizaba actividad física pero no cumplía el número de pasos requeridos para de 7000 pasos cinco días continuos para superar el reto

Tabla 19. Comportamiento del usuario restos 7 a 17

Número de Reto	Estatus	Fecha de Inicio	Fecha de Fin	Pasos a cumplir diariamente	Número de días continuos	Días empleados para cumplir el reto
7	Terminado	14/08/2019	25/08/2019	3250	5	11
8	Terminado	26/08/2019	01/09/2019	4000	5	6
9	Terminado	02/09/2019	28/09/2019	4750	5	26
10	Terminado	29/09/2019	04/10/2019	5500	5	5
16	Terminado	05/10/2019	11/10/2019	6250	5	6
17	Terminado	12/10/2019	11/04/2020	7000	5	182

Seguimiento: Estrategia de para incrementar la adherencia Chatbot – Terapia cognitivo conductual

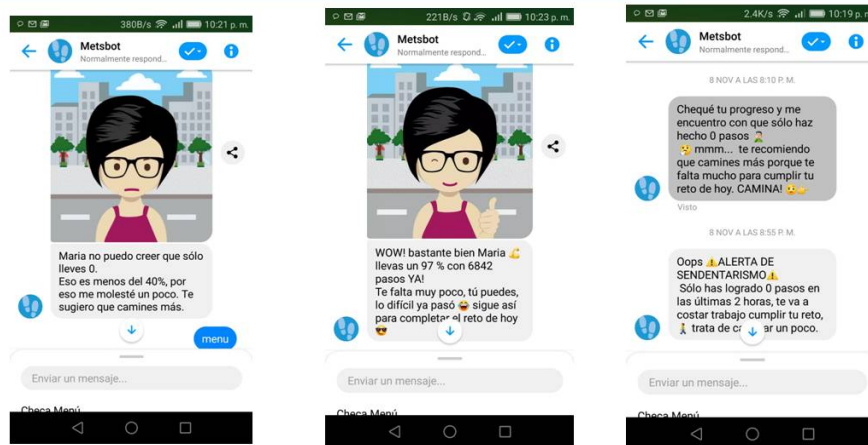


Figura 17. Mensajes de seguimiento para que el usuario retome su actividad y concluya su reto

Capítulo 6. Conclusiones

Dentro del análisis expuesto, es posible apreciar dos aspectos muy importantes: Que las problemáticas sobre la adherencia hacia hábitos, en específico aquellos que tienen que ver con la salud, es un tema complejo y con pocas implementaciones tecnológicas donde no que requiere de una intervención constante por parte de un profesional de la psicología o similar [73]. Y que la penetración de la población en internet no ha parado y conforme se hace más accesible, la penetración aumenta. Esto permite que propuestas de solución como Metsbot resulten factibles. Ese planteamiento aunado al análisis expuesto, la experimentación de este trabajo y otros similares nos refuerzan que la unión entre la tecnología y la psicología[74] resulta efectivo a través de un medio tan primitivo como lo es el lenguaje, el cual permite una serie de posibles beneficios para apoyar a la resolución de diversas problemáticas ya sean de mercado, de información o cuestiones de salud como lo que hemos estado observando con Mets. Por lo tanto tener Chatbots más que una idea que resurge como una “moda tecnológica”, es algo valioso de estudiar, analizar e implementar ya sea por canales de texto o voz, ya que con un lenguaje y objetivo definidos, retomando aspectos de las terapias cognitivas conductuales y expuestos de una forma amigable para con el usuario donde nunca se pretendió superar una “Prueba de Turing” sino más bien integrar un ente tecnológico con el cual se pudiera conversar podemos lograr resultados de adherencia. Esto último surge por la idea de “dar vida” a las imágenes que representaban los coaches de METS para convertirlos en *coaches reales pero virtuales*.

Retomando la factibilidad y viabilidad sobre la implementación de Chatbots recorramos sus aspectos por los cuales esta herramienta resulta tan versátil a forma de conclusión de la investigación:

La escalabilidad de esta tecnología está limitada a la solamente a la tecnología existente y la imaginación humana, ya que hoy puedes *textear* con tu coach virtual de Mets pero también podrías *hablar* con el mismo a través de una bocina inteligente o celular mientras de forma visual ves algún resultado en un dispositivo *wearable* como un *smartwatch* o que abra la puerta para salir a correr/caminar, entre muchas cosas más. Solo hablando de Metsbot tenemos una pieza de software que constantemente se actualiza según los requerimientos de sus usuarios para arreglar sus fallos o agregar nuevas funcionalidades de forma ligeramente sencilla por la forma en la que fue modelado. Lo cual nos lleva a mostrar que no es un sustituto de la psicología en esta área o de las terapias cognitivo conductuales sino un medio más para poderlo llevar a más personas, en cuanto a

difusión y alcance de este software para los usuarios es un factor que pudimos observar durante la implementación de Metsbot, ya que sin necesidad de realizar una campaña de difusión o similar usuarios nuevos que no conocen de METS se siguen acercando por medio del uso de Facebook, no se experimentó integrando herramientas de publicidad para tener una actividad aun mayor, pero el pronóstico es favorable a que habrá una cantidad de adquisición de usuarios nuevos y que estos pueden generar adherencia al modelo. También como se ha mencionado anteriormente pudimos descubrir lo complejo que puede resultar la implementación del lenguaje que empleara un Chatbot y como este debe acoplarse a su público objetivo y al objetivo del mismo software, el cómo implementar funciones que permitan que el uso del Chatbot sea amigable como activar acciones/flujos de conversación por medio de texto o botones, el papel importante de los recursos gráficos para transmitir sentimientos. Retomando el caso particular de Metsbot como ya se comentó su ambiente se compone de herramientas gratuitas que pueden resultar en limitantes a mediano y largo plazo por lo cual habría también que replantear el poder costear el servicio o realizar una migración, otra limitante es la necesidad de iterar constantemente sobre el glosario de palabras o frases que puede entender, ya que este proceso es manual y no automatizado. Los usuarios de prueba y usuarios reales no resultaron ser buenas fuentes para una retroalimentación de la misma, sin embargo, esa retroalimentación fue una constante lo cual nos puede indicar que aun alguno de los flujos de conversación puede no estar totalmente pulidas. Por último resulta benéfico señalar que a pesar de toda la información y experimentación que se ha realizado sigue siendo un área medianamente explorada con una limitante tan grande como lo es el idioma, que a su vez es su punto más fuerte para el alcance e interacción con las personas, lo cual resulta complejo porque se está hablando de que el medio de comunicación principal para con el usuario es el mismo lenguaje humano y es por ello que será necesario realizar demasiadas iteraciones sobre los proyectos individuales para que a nivel conceptual se puedan seguir obteniendo modos de realizar cada vez mejores Chatbots como soluciones comunicativas.

Capítulo 7. Referencias

1. TAFANI, R. et al. Factores de riesgo y determinantes de la salud. *Revista de Salud Pública*, [S.l.], v. 17, n. 3, p. 53-68, Mar. 2014. ISSN 1852-9429. Disponible en: <<https://revistas.psi.unc.edu.ar/index.php/RSD/article/view/6855>>. Fecha de acceso: 23 Feb. 2020 doi:<http://dx.doi.org/10.31052/1853.1180.v17.n3.6855>.
2. Organización Mundial de la Salud (OMS), Informe sobre la salud en el mundo 2008: La atención primaria de salud, más necesaria que nunca; Capítulo 4. Política de salud. ISBN 978 92 4 356373 2. Disponible en: <https://www.who.int/whr/2008/es/>
3. Instituto Nacional de Salud Pública (INSP), Encuesta Nacional de Salud 2018: Presentación de Resultados. Disponible en: <https://ensanut.insp.mx/encuestas/ensanut2018/informes.php>
4. Noreña, M. (n.d.). Teorías de cambio de comportamiento, resumen Preparado. [pdf] Bogotá, Colombia, pp.1-3. Disponible en: <http://proinapsa.uis.edu.co/redcups/Biblioteca/Modelo%20Participativo/Teor%C3%ADas%20de%20cambio%20de%20comportamiento.pdf> [Accedido 28 Sep. 2018].
5. Quillas Benites RK, Vásquez Valencia C, Cuba Fuentes MS. Promoción de cambios de comportamiento hacia estilos de vida saludable en la consulta ambulatoria. *Acta Med Peru*.2017;34(2):125-31
6. Tejada Zabaleta, Alonso. Agenciación humana en la teoría cognitivo social: Definición y posibilidades de aplicación. *Pensamiento Psicológico*, vol. 1, núm. 5, julio-diciembre, 2005, pp. 117-123 Pontificia Universidad Javeriana Cali, Colombia
7. Emilio Moreno San Pedro, Jesús Gil Roales-Nieto, El modelo de creencias de salud revisión teórica, consideración crítica y propuesta alternativa : hacia un análisis funcional de las creencias en salud. *International journal of psychology and psychological therapy*, ISSN 1577-7057, Vol. 3, Nº. 1, 2003, págs. 91-109
8. Stefani, D. (2005). Teoría de la Acción Razonada: Una propuesta de evaluación cuali-cuantitativa de las creencias acerca de la institucionalización geriátrica. *Revista Evaluar*, 5(1). <https://doi.org/10.35670/1667-4545.v5.n1.538>
9. Berra Ruiz, E., & Muñoz Maldonado, S. I. (2018). El modelo transteórico aplicado al cambio de conductas relacionadas con la reducción del peso corporal. *Revista Digital Internacional De Psicología Y Ciencia Social*, 4(2), 153-170. <https://doi.org/10.22402/j.rdiipycs.unam.4.2.2018.165.153-170>
10. TAFANI, R. et al. Factores de riesgo y determinantes de la salud. *Revista de Salud Pública*, [S.l.], v. 17, n. 3, p. 53-68, Mar. 2014. ISSN 1852-9429. Disponible en:

- <<https://revistas.psi.unc.edu.ar/index.php/RSD/article/view/6855>>. Fecha de acceso: 23 Feb. 2020
doi:<http://dx.doi.org/10.31052/1853.1180.v17.n3.6855>.
11. Organización Mundial de la Salud (OMS), Informe sobre la salud en el mundo 2008: La atención primaria de salud, más necesaria que nunca; Capítulo 4. Política de salud. ISBN 978 92 4 356373 2. Disponible en: <https://www.who.int/whr/2008/es/>
 12. Instituto Nacional de Salud Pública (INSP), Encuesta Nacional de Salud 2018: Presentación de Resultados. Disponible en: <https://ensanut.insp.mx/encuestas/ensanut2018/informes.php>
 13. Noreña, M. (n.d.). Teorías de cambio de comportamiento, resumen Preparado. [pdf] Bogotá, Colombia, pp.1-3. Disponible en:
<http://proinapsa.uis.edu.co/redcups/Biblioteca/Modelo%20Participativo/Teor%C3%ADas%20de%20cambio%20de%20comportamiento.pdf> [Accedido 28 Sep. 2018].
 14. Quillas Benites RK, Vásquez Valencia C, Cuba Fuentes MS. Promoción de cambios de comportamiento hacia estilos de vida saludable en la consulta ambulatoria. *Acta Med Peru.* 2017;34(2):125-31
 15. Tejada Zabaleta, Alonso. Agenciación humana en la teoría cognitivo social: Definición y posibilidades de aplicación. *Pensamiento Psicológico*, vol. 1, núm. 5, julio-diciembre, 2005, pp. 117-123 Pontificia Universidad Javeriana Cali, Colombia
 16. Emilio Moreno San Pedro, Jesús Gil Roales-Nieto, El modelo de creencias de salud revisión teórica, consideración crítica y propuesta alternativa : hacia un análisis funcional de las creencias en salud. *International journal of psychology and psychological therapy*, ISSN 1577-7057, Vol. 3, Nº. 1, 2003, págs. 91-109
 17. Stefani, D. (2005). Teoría de la Acción Razonada: Una propuesta de evaluación cuali-cuantitativa de las creencias acerca de la institucionalización geriátrica. *Revista Evaluar*, 5(1).
<https://doi.org/10.35670/1667-4545.v5.n1.538>
 18. Berra Ruiz, E., & Muñoz Maldonado, S. I. (2018). El modelo transteórico aplicado al cambio de conductas relacionadas con la reducción del peso corporal. *Revista Digital Internacional De Psicología Y Ciencia Social*, 4(2), 153-170.
<https://doi.org/10.22402/j.rdi.2018.165.153-170>
 19. Rodríguez M., Mendoza E., Cumplido A. Terapia cognitivo-conductual en el manejo integral de la obesidad en adolescentes. *Rev Med Inst Mex Seguro Soc* [internet] 2014 [consultado 22 Nov 2018]; volumen (52): 568-573. Disponible en: <https://www.medigraphic.com/pdfs/imss/im-2014/ims1411.pdf>
 20. Dalle, R., Calugi, S., Centis, E. El Ghoch, M, y Marchesini, G. (2011). Cognitive-Behavioral Strategies to Increase the Adherence to Exercise in the Management of Obesity. *Journal of Obesity*, (2011). Recuperado desde: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2968119/>
 21. Flores, L. El proceso de adopción de precauciones en la promoción de la salud. *Rev. Argén. Clín. Píscol.* [Internet] 2002 [consultado 22 Nov 2018]; volumen (11): 23-33 Disponible en:
<https://pesquisa.bvsalud.org/portal/resource/pt/lil-402961>

22. Rodríguez I. y Lodoño C. (2010). El proceso de adopción de precauciones en la prevención secundaria del consumo de cigarrillo en estudiantes universitarios. *Acta Colombiana de Psicología*, (13). Recuperado desde: <https://repository.ucatolica.edu.co/bitstream/10983/169/1/v13n1a08.pdf>
23. Guibert W., Grau J. y Prendes Labrada, M. (1999). ¿Cómo hacer más efectiva la educación en la atención primaria?. *Rev Cubana Med Gen Integr*, (15). Recuperado desde: http://scielo.sld.cu/scielo.php?pid=S0864-21251999000200010&script=sci_arttext&tlng=en
24. American Psychological Association. Cómo hacer cambios de comportamientos duraderos. [Internet] (2010). [consultado: 22 Nov 2018]. Disponible en: <https://www.apa.org/centrodeapoyo/duraderos>
25. Vivian Giang. Can technology really change your habits? [Internet] (2014). [consultado: 10 DIC 2018]. Disponible en: <https://www.fastcompany.com/3033986/can-technology-really-change-your-habits>
26. G. Urdiales. Inteligencia artificial para la atención médica del futuro [Internet] (2018). [consultado: 11 DIC 2018]. Disponible en: <https://www.economista.es/salud-innovacion/noticias/9097308/04/18/La-inteligencia-artificial-perfila-la-salud-del-futuro.html>
27. Dr. Cameron Sepah. Augmented Intelligence: How to combine human and artificial intelligence to change behavior [Internet] (2017). [consultado: 11 Dic 2018]. Disponible en: <https://medium.com/goactualize/augmented-intelligence-81b3e7f3489>
28. E-terapia. Chatbot terapia: la inteligencia artificial y la psicoterapia [Internet] (2017). [consultado: 11 Dic 2018]. Disponible en: <https://www.e-terapia.com/2017/05/chatbot-terapia-la-inteligencia.html>
29. Grolleman J., Van Dijk B., Nijholt A. y Van Emst A. (2006). Break the habit! Designing an e-therapy intervention using a virtual coach in aid of smoking cessations. *Persuasive Technology*, (2006). Recuperado desde: https://link.springer.com/chapter/10.1007%2F11755494_19
30. S. Lim, D. (2017). Predicting outcomes in online chatbot-mediated therapy. *Stanford* (2017). Recuperado desde: <https://pdfs.semanticscholar.org/d613/015f7f9c6cf9d0f886f20319815cf2af564a.pdf>
31. OMENT. México ocupa el 2do lugar en obesidad en adultos según la OCDE [Internet] (2015). [consultado: 18 Jul 2017]. Disponible en: <http://oment.uanl.mx/mexico-ocupa-el-2o-lugar-en-obesidad-en-adultos-segun-la-ocde/>
32. SEGOB. Lineamientos por los que se dan a conocer los criterios nutrimentales y de publicidad que deberán observar los anunciantes de alimentos y bebidas no alcohólicas para publicitar sus productos en televisión abierta y restringida, así como en salas de exhibición cinematográfica, conforme a lo dispuesto en los artículos 22 Bis, 79, fracción X y 86, fracción VI, del Reglamento de la Ley General de Salud en Materia de Publicidad [Internet] (2014). [consultado: 18 Jul 2017]. Disponible en: http://www.dof.gob.mx/nota_detalle.php?codigo=5340694&fecha=15/04/2014
33. Instituto Nacional de Estadística y Geografía. Boletín de Prensa Numero 22/17 Modulo De Práctica Deportiva y Ejercicio Físico [Internet] (2017).[consultado: 18 Jul 2017] Disponible en: http://www.inegi.org.mx/saladeprensa/boletines/2017/moprade/moprade2017_01.pdf

34. Rubén Vázquez. México, primer lugar en penetración Social Media [Internet] (2014). [consultado: 18 Jul 2017]. Disponible: <https://www.forbes.com.mx/mexico-primer-lugar-en-penetracion-social-media/>
35. Islas O. Cifras sobre jóvenes y redes sociales en México. Entre textos. 2017 [consultado: 23 Jul 2018]; volumen (19): 1-16. Disponible en: <http://entretextos.leon.uia.mx/num/19/PDF/ENT19-1.pdf>
36. Jorge Tamayo. Hábitos de usuarios de internet en México 2017 [Internet] (2017). [consultado: 23 Jul 2018]. Disponible en: <https://webmarketingtips.mx/local/habitos-de-usuarios-en-mexico-2017-7-258/>
37. A. P. Saygin, I. Cicekli, V. Akman, “Turing test: 50 years later,” in *The Turing Test*, Springer, 2003, pp. 23–78.
38. M. L. Mauldin, “Chatterbots, tinymuds, and the Turing test: Entering the Loebner prize competition”, *Proceedings of Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, Washington, 1994, pp. 16-21.
39. S. B Abu., y E. Atwell, , “Chatbots: Are they Really Useful”, *Journal for Language Technology and Computational Linguistics (JLCL) (formerly LDVForum)*, vol 22, pp. 29-49, 2007
40. Matt Schlicht. The complete beginners guide to chatbots [Internet] (2016). [consultado: 22 Jul 2018]. Disponible en: <https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca>
41. J. López-Martínez, M. García, L. Valle-Rosado. ”Desarrollo e implementación de un chatbot conversacional como apoyo a los estudiantes en su proceso de titulación”, *Memorias 1er Congreso Internacional de Robótica y Computación (CIRC)*, La Paz, México 2013, pp. 361-365.
42. Jayawaderna C., Kuo K., Datta C., Stafford R. Q., Broadbent E. y Macdonald B. A. (2014). Design, implementation and field test of a socially assistive robot for the elderly: Healthbot version 2. *IEEE, (2012 4th)*. Recuperado desde: <https://ieeexplore.ieee.org/abstract/document/6290890>
43. Hanna. Health bots and their pursuit of creating a healthier society [Internet] (2018). [consultado: 13 Oct 2018]. Disponible en: <https://botsociety.io/blog/2018/03/health-bots/>
44. Escuela técnica superior de ingeniería informática. Introducción a las aplicaciones web [internet] (2004). [consultado: 13 Oct 2018]. Disponible en: <http://www.lsi.us.es/docencia/get.php?id=854>
45. Abdullah H. y Zeki A. (2014). Frontend and Backend technologies in Social Networking Sites: Facebook as an Example. *IEEE, (2014 3rd)*. Recuperado desde: <https://ieeexplore.ieee.org/abstract/document/7076874>
46. W3Schools. Web development roadmaps [internet]. [consultado: 13 Oct 2018]. Disponible en: <https://www.w3schools.com/whatis/>
47. Nicole Chapaval. Qué es Frontend y Backend [Internet] (2018). [consultado: 20 Nov 2018]. Disponible en: https://platzi.com/blog/que-es-frontend-y-backend/?gclid=EAIaIQobChMIkKef29Wb4QIVlbXCh1lbwESEAAAYASAAEgLq-PD_BwE

48. Hémida R. Desarrollo de un motor gráfico para el recorrido de mundos tridimensionales [Internet]. Lugar de publicación: colección de tesis digitales UDLAP. (2006). [consultado: 20 Nov 2018]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/hermida_t_r/capitulo2.pdf
49. Marssè M. REST API Design Rulebook [Internet]. USA: O'REILLY; 2012 [revisión: 2011; citado: 2018 Nov 20]. Disponible en: <https://books.google.com.mx/books?id=eABpzyTcJNIC&lpg=PR3&ots=vzTy22leKG&dq=API%20rest&lr&hl=es&pg=PR3#v=onepage&q=API%20rest&f=false>
50. Thomas R. Architectural Styles and the Design of Network-based Software Architectures [Internet]. Lugar de publicación: University of California, Irvine. (2000). [consultado: 20 Nov 2018]. Disponible en: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
51. Puerta J. M. Desarrollo de una API para la descripción y gestión de Servicios Web REST [Internet]. Lugar de publicación: Universitat Jaume. (2015). [consultado: 20 Nov 2018]. Disponible en: http://repositori.uji.es/xmlui/bitstream/handle/10234/156006/TFM_2014_puertaJ.pdf?sequence=1&isAllowed=y
52. Cortez A., Vega H., y Pariona J. (2009). *Procesamiento de lenguaje natural. Revista de ingeniería de sistemas e informática, volumen 6* (2da edición). Recuperado de: http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/risi/2009_n2/v6n2/a06v6n2.pdf
53. Dale R., Moisl H. y Somers H. Handbook of Natural Language Processing [Internet]. USA: Basel; 2000 [revisión: 2000; consultado: 20 Nov 2018]. Disponible en: <https://books.google.com.mx/books?id=VoOLvxyX0BUC&lpg=PP1&dq=natural%20language%20processing&hl=es&pg=PA2#v=onepage&q=natural%20language%20processing&f=true>
54. Facebook. Documentos para desarrolladores [Internet] Disponible en: <https://developers.facebook.com/docs>
55. Facebook. Documentos para desarrolladores [Internet] Disponible: <https://developers.facebook.com/docs/messenger-platform/>
56. elEconomista.es. ¿Qué son exactamente los Chatbots y para qué sirven? [Internet]. 2016. [consultado: 20 Nov 2018]. Disponible en: https://biblioguias.uam.es/citar/estilo_vancouver
57. Charlan J. Qué es un Chatbot y para qué sirve [Internet]. Economía Digital; 2018 [consultado: 20 Nov 2018]. Disponible en: <https://www.esic.edu/rethink/2018/08/04/que-es-un-chatbot-y-para-que-sirve/>
58. Slotnisky D. Transformación Digital: Cómo las personas y las empresas deben adaptarse a esta revolución. Argentina:Digital House;2016. [revisión: 2016; consultado: 20 Nov 2018]. Disponible en: <https://books.google.com.mx/books?id=9dBJDQAAQBAJ&lpg=PT108&dq=Para%20qu%C3%A9%20sirven%20los%20chatbots&hl=es&pg=PT108#v=onepage&q=Para%20qu%C3%A9%20sirven%20los%20chatbots&f=false>
59. Lopez L. Metodología para desarrollar Chatbots [Internet] Medium: (2018). [consultado: 20 Nov 2018]. Disponible en: <https://medium.com/@llpzl.edu/metodolog%C3%ADa-para-desarrollar-chatbots-part-1-eb10f97d6ffb>

60. Cortés D. Las bases de diseñar un buen Chatbot (o de por qué no soy el enemigo) [Internet] Medium: (2018). [consultado: 20 Nov 2018]. Disponible en: <https://medium.com/mejorindustriati/las-bases-de-dise%C3%B1ar-un-buen-chatbot-o-de-por-qu%C3%A9-no-soy-el-enemigo-ed1545ec5b53>
61. Freen D. Hello Joy [Internet]. Chatbots Magazine:Medium; 2016. [consultado: 20 Nov 2018]. Disponible en: <https://chatbotsmagazine.com/hello-joy-b816ca10c57f>
62. Handel S. How to plant powerful ideas in people’s minds that actually take root and grow [Internet]. The emotion machine: 2016. [consultado: 20 nov 2018]. Disponible en: <https://www.theemotionmachine.com/how-to-plant-powerful-ideas-in-peoples-minds-that-actually-take-root-and-grow/>
63. Sanhueza. M. Metodología UX conversacional para Chatbots ¿Cómo comprender y centrarte mejor en tus usuarios antes de crear un Chatbot? [Internet]. Planeta Chatbot;Medium: 2019. [consultado: 10 May 2019]. Disponible en: <https://planetachatbot.com/metodolog%C3%ADa-ux-conversacional-para-chatbots-c%C3%B3mo-comprender-y-centrarte-mejor-en-tus-usuarios-45bbebacbbdb>
64. Henao R., Fernando R., Arevalo V. y Teresa M. Barreras percibidas en jóvenes universitarios para realizar actividad física. Rev Cub de Salud Pub [Internet] 2016 [consultado: 10 May 2019]; 42(1): 61-69. Disponible en: <https://www.redalyc.org/pdf/214/21444931007.pdf>
65. Henao R., Correa J. y Ramirez R. Psychometric properties of spanish version of the “Barriers to Being Active Quiz” among university students in Colombia. Europe PMC [Internet] 2015 [consultado: 10 May 2019]; 31(4): 1708-1716. Disponible en: <https://europepmc.org/article/med/25795962>
66. Chatfuel. About Us [Internet] 2018 [consultado: 10 May 2019]. Disponible en: <https://chatfuel.com/about-us/>
67. BBVAOPEN4U. Cómo desarrollar un Chatbot para Facebook Messenger con Chatfuel [Internet] 2016 [consultado: 10 May 2019]. Disponible en: <https://bbvaopen4u.com/es/actualidad/como-desarrollar-un-chatbot-para-facebook-messenger-con-chatfuel>
68. Chatfuel. JSON API [Internet] 2016 [consultado 20 Jun 2019]. Disponible en: <https://docs.chatfuel.com/articles/735122-json-api>
69. Asociación Americana de Psicología (APA). Cómo hacer cambios de comportamiento duraderos. [internet] Disponible en: <https://www.apa.org/centrodeapoyo/duraderos>
70. Duhigg Charles. El Poder de Los Hábitos: Por Qué Hacemos Lo Que Hacemos En La Vida Y Los Negocios. Editorial Urano. 2012.
71. Flórez, L. (2003). Cuestionario de etapas integrado a un programa (CEMA - PEMA) para la modificación del consumo abusivo de alcohol: evaluación de etapas y variables intermedias. Acta Colombiana de Psicología. 9. 83-104.
72. 14 métricas para monitorear el rendimiento de tu Chatbot [Internet] 2018 [consultado: 26 Jul 2019]. Disponible en: <https://planetachatbot.com/14-metricas-para-monitorear-rendimiento-de-chatbot-df4c6e9aadf4v-df4c6e9aadf4>

73. Cómo motivar el cambio conductual [Internet] 2014 [consultado: 3 Agosto 2019]. Disponible en: <https://www.eufic.org/es/healthy-living/article/motivating-behaviour-change>
74. Barman, L., Mukhopadhyay, D. K., & Bandyopadhyay, G. K. (2018). Use of Social Networking Site and Mental Disorders among Medical Students in Kolkata, West Bengal. *Indian journal of psychiatry*, 60(3), 340–345. https://doi.org/10.4103/psychiatry.IndianJPsychiatry_210_18