UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

TEMA: IMPLEMENTACIÓN DE SISTEMA DE ALERTA AL MONITOR MULTIPARÁMETROS.

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES: Sigrit Dailin Delgado Vega
Diego Isaac Ruiz Córdova
TUTOR: Franklin Illich Kuonquí Gaínza

Guayaquil - Ecuador
2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Sigrit Dailin Delgado Vega** con documento de identificación Nº **1719040287** y **Diego Isaac Ruiz Córdova** con documento de identificación Nº **0930992383**; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 15 de septiembre del año 2022

Atentamente,

Sigrit Dailin Delgado Vega
1719040287

Diego Isaac Ruiz Córdova
0930992383

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, **Sigrit Dailin Delgado Vega** con documento de identificación Nº **1719040287** y **Diego Isaac Ruiz Córdova** con documento de identificación Nº **0930992383**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **IMPLEMENTACIÓN DE SISTEMA DE ALERTA AL MONITOR MULTIPARÁMETROS**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo a final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Guayaquil, 15 de septiembre del año 2022

Atentamente,

Sigrit Dailin Delgado Vega
1719040287

Diego Isaac Ruiz Córdova
0930992383

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Franklin Illich Kuonquí Gainza**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **IMPLEMENTACIÓN DE SISTEMA DE ALERTA AL MONITOR MULTIPARÁMETROS**, realizado por **Sigrit Dailin Delgado Vega** con documento de identificación N° **1719040287** y por **Diego Isaac Ruiz Córdova** con documento de identificación N° **0930992383**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 19 de septiembre del año 2022

Atentamente,

Ing. Franklin Illich Kuonquí Gainza, Mg.
0909627432

DEDICATORIA

Este trabajo de titulación se lo quiero dedicar, especialmente a la memoria de Miguel Delgado, gracias por inspirarme a seguir adelante, por acompañarme a lo largo de mi vida por las memorias que el tiempo no borrará, por las risas ante la adversidad, el cariño, la dedicación y los consejos que no olvidaré. No me alcanzan las palabras para describir el vació que quedó, pero espero que donde sea que estés te sientas orgulloso. Te estaré agradecida por siempre.

Sigrit Dailin Delgado Vega

Este trabajo de titulación está dedicado a mis padres, quienes son el motor de mi vida y gracias a sus esfuerzos y gran sacrificio lograré terminar mis estudios universitarios. Son quienes me inspiraron a estudiar esta carrera, me enseñaron a trabajar arduamente para conseguir y lograr todos mis objetivos, quienes en vida me enseñaron lo importante que es la perseverancia y lo duro que puede ser la vida, pero a pesar de la dificultades mientras exista un rayo de luz nada malo pasara. Ambos me guiaron con su amor y perseverancia a lo largo de mi vida.

Dedico esta investigación a toda mi familia, que me han enseñado que el mejor regalo que se puede dar en la vida es la educación, el cual es un puente que nos permite aprender e innovar a nuevos mundos en donde los sueños son posibles.

Diego Isaac Ruiz Córdova

AGRADECIMIENTO

Quiero agradecer a mi familia, a Estrella Cedeño y Miguel Delgado, porque a pesar de la distancia siempre me han apoyado. A Betthy Velásquez, por siempre apoyarme, por los consejos y por ser mi modelo a seguir, a mi madre y mi hermana por su motivación a seguir adelante. Quiero agradecer también a Carlos, por su apoyo incondicional a lo largo de estos años.

Agradezco a la Universidad Politécnica Salesiana, a los docentes que me han acompañado en estos cinco años de carrera, a mis compañeros que se convirtieron en colegas, por todo el conocimiento recibido y por hacer de esta experiencia inolvidable.

Un agradecimiento especial al Ing. Franklín Kuonquí por su paciencia y dedicación con nosotros.

Sigrit Dailin Delgado Vega

Agradezco principalmente a Dios y a los docentes de la carrera de ingeniería en mecatrónica que han formado parte de mi formación profesional, quienes con su conocimiento y su apoyo brindaron las herramientas para un futuro exitoso como un ingeniero mecatrónico.

Agradezco a todas esas amistades que me dio directamente como indirectamente la universidad, porque me han demostrado ser grandes seres humanos que siempre me han brindado de su apoyo aun cuando fue difícil avanzar, por acompañarme y hacer de este camino más agradable y enriquecedor.

Diego Isaac Ruiz Córdova

RESUMEN

Dentro del mundo de la medicina el uso del monitor multiparámetros o monitor de pacientes, resulta vital para la obtención de datos del estado actual del paciente. Por ello el presente proyecto tiene como objetivo diseñar un sistema que pueda ayudar a optimizar el tiempo de respuesta del paramédico mediante el uso de comandos de voz a travez de un dispositivo ECHO de AMAZON ALEXA.

El funcionamiento de este consiste en el procesamiento de datos a través de ThingSpeak, plataforma IoT que se encarga de almacenar y permitir una visualización de datos en diversos tipos de dispositivo, posteriormente esta información pasa por un backend por medio de la API propia de ThingSpeak, ahí se crea la aplicación que más adelante será procesada por la plataforma de desarrollo de Skills de Alexa y finalmente utilizada por el usuario.

Para fines demostrativos se desarrolló un monitor multiparámetros utilizando Arduino y sensores, además del uso de un módulo WiFi ESP-01 para implementar la conección a internet de los sensores a la nube.

Palabras claves: signos vitales, monitor de pacientes, ALEXA SKILLS, ALEXA, BACKENDLESS, ThingSpeak, Arduino Uno.

ABSTRACT

In the world of medicine, the use of the multi-parameter monitor or patient monitor is vital for obtaining data on the current state of the patient. Therefore, this project aims to design a system that can help optimize the response time of the paramedic by using voice commands through an AMAZON ALEXA ECHO device.

The operation of this consists of processing data through ThingSpeak, IoT platform that is responsible for storing and allowing data visualization in various types of device, then this information goes through a backend through ThingSpeak's own API, there is created the application that will later be processed by the Alexa Skills development platform and finally used by the user.

For demonstration purposes, a multi-parameter monitor was developed using Arduino and sensors, in addition to the use of a WiFi ESP-01 module to implement the internet connection of the sensors to the cloud.

Keywords: Vital sings, patient monitor, ALEXA SKILLS, ALEXA, BACKENDLESS, ThingSpeak, Arduino Uno.

ÍNDICE

ÍNDICE DE FIGURAS

I. PLANTEAMIENTO DEL PROBLEMA

El paramédico debe estar pendiente en todo momento de la señal que envía el monitor, revisándolo constantemente para cerciorarse del estado actual del paciente. Este proceso ralentiza las acciones que pueden ser tomadas en favor del paciente. Es por esto, que lo ideal sería enviar una señal de forma auditiva que brinde información importante para la toma de decisiones del paramédico. Actualmente, estas alarmas se dan por vía celular o remota, esto podría ser útil en un hospital, pero no en una ambulancia, donde el profesional no puede distraerse revisando su celular frecuentemente.[3]

La presión de la situación, el tiempo en contra y la gravedad del paciente son factores que se interponen al momento en que un paramédico atiende una emergencia. Este momento es crucial, pues es donde el paramédico realiza su mayor esfuerzo para ayudar al paciente y por lo tanto se pueden cometer la mayor cantidad de errores humanos. Debido a esto existe un porcentaje alto de riesgo para el paramédico de poder desarrollar: estrés postraumático, depresión, agotamiento, etc. Llevando un trabajo para nada fácil, el cansancio y la presión que ejercen sobre el paramédico podría poner en riesgo la vida del paciente y por ello se requiere una ayuda que no distraiga al paramédico. [8].

Cuando un monitor no funciona adecuadamente se puede perjudicar a los pacientes, este factor puede deberse a falta de mantenimiento o por mal uso de los paramédicos. Algunas de las causas para que los paramédicos usen inadecuadamente los monitores son, por ejemplo, que muchos están acostumbrados al uso de un monitor específico, alguna marca de preferencia o desconocen nuevas funciones de los actuales monitores. Para evitar su daño temprano y minimizar los riesgos que esto conlleve es muy importante que la mejora sea de agrado del operador, tenga una plataforma amigable y sobre todo de fácil uso. De esta manera para los profesionales será mucho más sencillo entender el funcionamiento y operación del monitor, además de no añadir ni dificultar más su trabajo. [5].

En el tema de alarmas, el monitor multiparámetros se enfoca en reportar el estado actual del paciente, mas no si este está propenso a sufrir complicaciones en su estado. Esta ausencia de alarmas preventivas hace que el estado del paciente dependa solamente del paramédico y su constante monitoreo del mismo. Esta función no es equiparable a las alarmas que en algunos monitores se pueden programar debido a que estas solo sonarán si obtienen un parámetro y por tanto no se puede saber si el paciente tiende a sufrir algún ataque o si se trata de una lectura inadecuada de los sensores.

II. JUSTIFICACIÓN

El presente trabajo de titulación tiene como objetivo diseñar e implementar un sistema de alerta al monitor multiparámetro con el fin de utilizar la tecnología para mejorar el desempeño de los profesionales paramédicos en su labor dentro de las ambulancias.

La toma de signos vitales ejerce un papel fundamental en una situación de emergencia, puesto que es muy importante que el paramédico conozca el estado del paciente y atenderlo de manera más eficaz. Si pudiera recibir esta información de manera oportuna y por medio de un asistente como el Echo Dot, podría prestar un servicio más ágil, sobre todo porque la forma de atención que brinde el paramédico puede marcar una drástica diferencia en el recorrido que tiene el paciente al hospital.

Por ello se creará una mejora para el monitor multiparámetros, uno de los más usados en el servicio hospitalario debido a su costo y portabilidad. BeneView T1 es al mismo tiempo un módulo multiparámetro y un monitor de transporte. Como módulo, conecta con el monitor BeneView T1 a la cabecera del paciente y proporciona todos los parámetros estándar.Adicionalmente, puede desenchufarse de manera rápida para seguir al paciente por los puntos de asistencia, lo que permite su monitorización completa también durante el transporte. Por tanto, BeneView T1 ofrece transferencia de datos inalámbrica y garantiza la continuidad de la información de monitorización. Además, cuando está conectado de forma inalámbrica al sistema de monitorización central Hypervisor VI, puede visualizarse toda la información del BeneView T1 desde la enfermería o desde cualquier monitor de cabecera que esté dentro de la red.

Tanto por la falta de mantenimiento del equipo, como con el mal uso del mismo, el sistema tampoco emite alertas sobre ello. Por eso también se va a implementar funciones que permitan identificar el mal funcionamiento del equipo para que este pueda seguir funcionando de manera óptima y no emita alertas falsas que también figuran como uno de los problemas grandes que se tiene al atender a pacientes en la ambulancia.

Figura 1. Modelo monitor multiparámetros

III. OBJETIVOS

III-A. Objetivos Generales

Implementar un sistema de alerta para un monitor multiparámetro que reduzca el tiempo de respuesta de los paramédicos, mediante mensajes de audio que informe el estado del paciente.

III-B. Objetivos Específicos

- Investigar los sistemas de monitoreo multiparámetros para la obtención de señales con la información del estado del paciente.
- Diseñar un dispositivo electrónico que convierta las señales del monitor multiparámetro en mensajes de audio, utilizando microcontroladores.
- Implementar el sistema de alerta para la verificación de su funcionamiento adecuado.

IV. MARCO TEÓRICO

IV-A. Toma de signos vitales

Así como para los ingenieros los parámetros que envía un sensor pueden brindar un estimado de lo que sucede con el funcionamiento de la máquina, para un médico equivale a la toma de signos vitales. De la misma manera, los signos vitales pueden dar un indicio de un mal funcionamiento del cuerpo, ya que el sistema inmune emite señales de alerta como: fiebre, tos, estornudos, etc. Alertas que pueden ser monitorizadas y ayudar con un mejor diagnóstico por parte del personal médico. [4]

Existen características que pueden causar una variación en los valores obtenidos siendo los principales los siguientes:

1. Edad: La principal diferencia entre un niño y un adulto es su frecuencia cardiaca, ya que los niños tienen una mayor frecuencia cardiaca. Esto también es producto del envejecimiento de los vasos sanguíneos, que con el paso del tiempo pierden su elasticidad y se vuelven más lentos.
2. Género: Un ejemplo de esto es que las mujeres, por una derivación de las hormonas producidas a raíz de la pubertad y de su ciclo menstrual, suelen tener el pulso más alto que los hombres de su misma edad o similar.
3. Ejercicio físico: La forma más rápida de alterar la respiración y el pulso es con la actividad física, ya que esta produce calor y aumenta el metabolismo.
4. Embarazo: No es sorpresa que mientras más avance el embarazo la mujer tenga un pulso y respiración más acelerados, es por eso que se requiere un control más estricto ya que la línea entre lo patológico y lo que podría considerarse normal es muy fina.
5. Estado emocional: Las emociones que generan adrenalina o noradrenalina conllevan a un aumento del metabolismo, al igual pasa con el ejercicio pueden aumentar la respiración y el pulso.
6. Hormonas: Como se explica en el punto anterior la producción de progesterona en las mujeres altera ligeramente sus signos vitales.
7. Medicamentos Existen medicamentos relajantes o estimulantes que pueden producir adrenalina o inducir al estado de reposo.

Para este proyecto se ha tomado en cuenta los parámetros normales de un adulto sano en reposo, que son los siguientes:

- Pulso: 60 a 100 bpm o latidos por minuto
- Temperatura: 36.5°C a 37.3°C

La temperatura suele variar por cuestiones de conversión pero se ha decidido establecerla en un rango que va desde 36°C a 37°C.

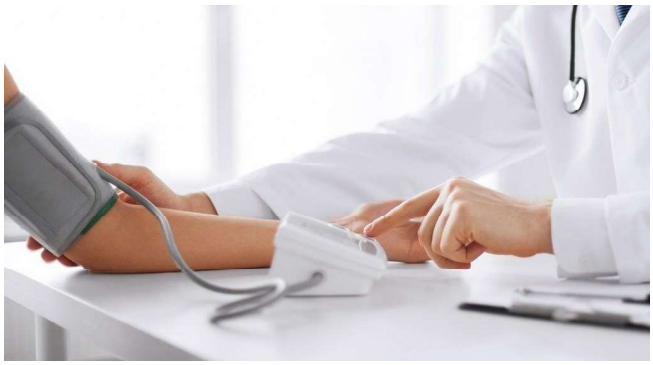


Figura 2. Toma de signos vitales

IV-A1. Pulso cardiaco: Siendo uno de los indicadores más importantes en la toma de signos vitales, la frecuencia cardiaca indica que tan bien funciona el cuerpo humano y el estado de salud del mismo. Este parámetro nos indica que tan bien está circulando la sangre. Por ello tomar mediciones de la frecuencia cardíaca con regularidad puede ayudarlo a controlar su salud y sus niveles de condición física. También puede ayudarlo a evitar problemas en la salud a futuro. [9]

La frecuencia cardíaca normal en reposo es entre 60 y 100 BPM, pero puede ser más baja o más alta dependiendo de la edad, género y salud en general. Una frecuencia cardíaca rápida suele ser una indicación de ansiedad o estrés, mientras que un latido cardíaco lento indica mala salud o problemas relacionados con la edad.**Rajanna2018**

Si se llegan a ignorar el pulso, es decir forzar el cuerpo teniendo un pulso bajo o muy elevado, puede devenir en un sin número de consecuencias graves para la salud del paciente. Si el pulso es muy elevado (taquicardia) es señal de un exceso de estrés o ansiedad en el sistema que puede provocar hipertensión, enfermedades de las arterias coronarias, muerte cardiovascular prematura y otros problemas de salud graves. Por otro lado, si el ritmo cardiaco es bajo señala una falta de oxigenación, lo que provoca una presión arterial baja que provoca un flujo sanguíneo insuficiente a los órganos y tejidos, provocando fatiga, mareos y condiciones de salud más graves conocidas. como 'insuficiencia cardíaca'[1].

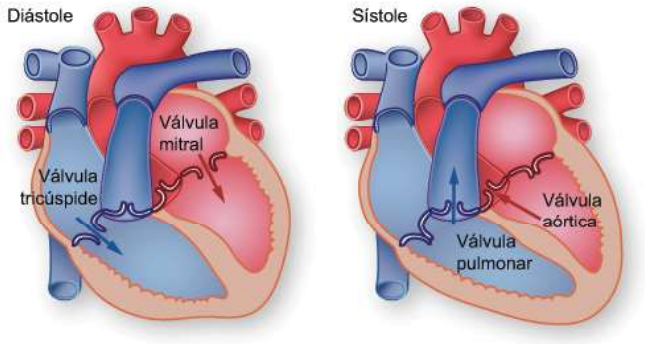


Figura 3. Funcionamiento y partes del corazón

IV-A2. Temperatura corporal: Así como la frecuencia cardiaca era esencial para la toma de signos vitales, la temperatura corporal resulta ser igual de importante. Cabe resaltar que suele ser el primer indicador de la presencia de un gente infeccioso en el cuerpo humano y la fiebre constituye uno de los mecanismos de defensa más usado por el sistema inmune. Debido a esto es esencial a la hora de dar un diagnóstico.

La temperatura corporal normal, de acuerdo a la Asociación Médica Americana (American Medical Association), se encuentra entre 97,8° F (o Fahrenheit, equivalentes a 36,5°C, o Celsius) y 99°F (37,2°C). [2]

Figura 4. Toma de temperatura

IV-B. Monitor Multiparámetro

Conocido también como monitor cardíaco, es un equipo que sirve para medir los signos vitales de un paciente que determinan su estado. Como este lo indica, es un monitor que muestra en una pantalla varios de los signos vitales, como son la frecuencia respiratoria, la presión invasiva y la no invasiva, la saturación de oxígeno, el dióxido de carbono, entre otros. [6]

Figura 5. Monitor multiparámetros

IV-B1. ¿Cómo Funciona?:

Los monitores se encargan de recoger, mostrar y almacenar todas las constantes vitales del paciente. El equipo trabaja de forma diferente para cada signo vital, por ejemplo, para medir la frecuencia cardiaca, recoge mediante electrodos la actividad eléctrica del corazón y la amplifica.

Para medir la frecuencia respiratoria recoge y amplifica los movimientos respiratorios del tórax. Y para determinar la cantidad de oxígeno lo hace a través del pulso.

El monitoreo de signos vitales se lleva a cabo especialmente en pacientes sometidos a anestesia, medicamentos, hemorragias internas o externas, enfermedades que requirieran transfusión de sangre y en aquellos que, por su delicado estado de salud, tienen una variación constante de signos vitales

IV-B2. Signos Vitales que Monitorea:

1. Electrocardiograma
2. Respiración
3. Temperatura
4. Presión no invasiva
5. Saturación de oxigeno
6. Presión invasiva

Figura 6. Monitor Multiparámetros y sensores

IV-C. Interfaz de Programación de Aplicaciones (API)

Por sus siglas en inglés, Application Programming Interfaces, las API's nacieron como una manera de intercambio entre dos o más programas. Alrededor del año 2000 estos intercambios pasaron de aplicaciones intaladas a realizarse en la web. Este intercambio de información entre una o más aplicaciones ha hecho que varias compañías creen sus propios servicios API. Un ejemplo de esto es el propio AWS de AMAZON.

El funcionamiento básico de este intercambio se realiza mediante la deficinición de un cliente (quien envía la solicitud) y el servidor (quien la recibe).

Existen varias formas de funcionamiento según como fueran diseñadas para ser utilizadas, en este caso se utilizar una API de REST.

IV-C1. API de REST: Como su nombre lo indica, esta API utiliza una arquitectura llamada REST. Esta se basa en que mientras se tenga un número de operaciones limitadas mejor será el servicio prestado.

En otras palabras esto se puede explicar haciendo una analogía con una memoria, entre más datos se almacenen más costará procesar o ingresar los mismos.

Las características que se deben cumplir para saber si se necesita utilizar una API de RESTful son las siguientes:

- Si las solicitudes del cliente al servidor se realizan por formato HTTP.
- La información del cliente no se almacena ni se hace referencia dentro de la comunicación. Es decir utiliza una comunicación sin estado.
- La información solicitada debe tener todas las facilidades (hipervínculos, recursos, etc) que permitan que el cliente pueda manipularla o trabajar con los datos adquiridos.
- Si se utiliza un sistema de datos que tenga una jerarquía.
- De ser necesario debe permitir el envío de códigos que se puedan ejecutar.

Esta arquitectura trabaja con identificadores universales únicos, o por sus siglas en inglés URI, quien facilita el acceso a la información que se planea editar según las siguientes operaciones:.

- GET: Como su nombre lo indica su función es obtener, leer o consultar información
- POST: Esta operación está diseñada para publicar, postear o crear información dentro del servidor.
- PUT: Esta operación cambia la información de estado o permite editarla.
- DELETE: Como lo indica su nombre su función es eliminar información.

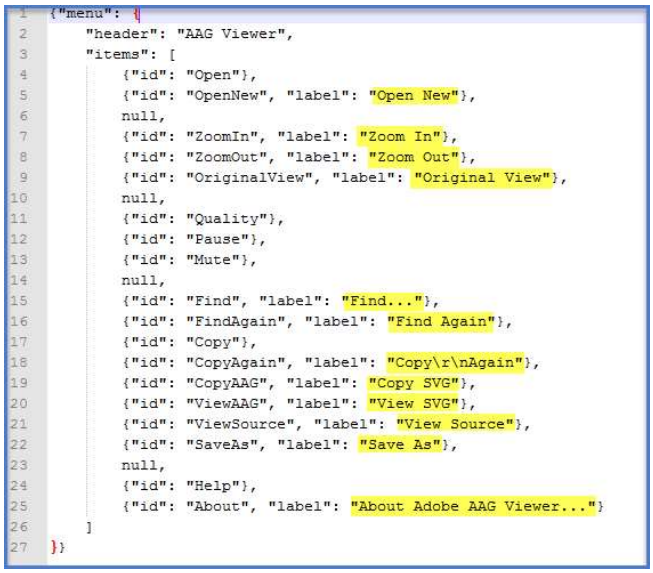
Por todo esto el uso de la arquitectura REST tiene algunas ventajas como que su portabilidad y comunicación con otras plataformas la hacen muy popular, sobre todo si se busca trabajar en un medio digital.

IV-D. JSON file

Ya que se utilizan algunos servidores para este proyecto su comunicación es posible gracias uso de archivos JSON.

Por sus siglas en inglés JavaSript Object Notation (JSON) es el nombre con el que se le conoce a un archivo que guarda estructuras de datos simples y objetos en su propio formato. Este se usa principalmente para trasmitir datos entre una aplicación web y un servidor.

Este file es muy utilizado, esto debido a que está compuesto por archivos de textos simples y es capaz de ser abierto por cualquier editor de texto, por ejemplo el Microsoft Notepad de Windows.



```
{"menu": {
    "header": "AAG Viewer",
    "items": [
        {"id": "Open"},
        {"id": "OpenNew", "label": "Open New"},
        null,
        {"id": "ZoomIn", "label": "Zoom In"},
        {"id": "ZoomOut", "label": "Zoom Out"},
        {"id": "OriginalView", "label": "Original View"},
        null,
        {"id": "Quality"},
        {"id": "Pause"},
        {"id": "Mute"},
        null,
        {"id": "Find", "label": "Find..."},
        {"id": "FindAgain", "label": "Find Again"},
        {"id": "Copy"},
        {"id": "CopyAgain", "label": "Copy\r\nAgain"},
        {"id": "CopyAAG", "label": "Copy SVG"},
        {"id": "ViewAAG", "label": "View SVG"},
        {"id": "ViewSource", "label": "View Source"},
        {"id": "SaveAs", "label": "Save As"},
        null,
        {"id": "Help"},
        {"id": "About", "label": "About Adobe AAG Viewer..."}
    ]
}}
```

Figura 7. Ejemplo de un file.json

IV-E. ThingSpeak

Para los creadores de MATLAB, el tener un servidor que pueda almacenar datos y que al mismo tiempo pueda devolver estos en forma de gráficas resultó en todo un desafío. Pero por esta razón se creó la plataforma de análisis y recolección de datos llamada ThingSpeak.

Esta plataforma abierta esta diseñada para compartir datos entre personas y objetos, es Open Source y se maneja utilizando una API y el prococolo HTTP para la obtención y distribución de datos.

Está pensada en el Internet de las Cosas (IoT), ya que se enfoca en brindar la mayor facilidad posible, minimizando el uso de código de programación y una configuración sencilla con respecto a otros Frameworks. Su objetivo es ayudar en la realización de proyectos como también para mejorar el monitoreo en áreas como energía, ambiente, etc.

En el caso de su uso para estudiantes y educadores, ThingSpeak ha sido una herramienta predilecta en la adquisición de datos por sensores. Teniendo así su propia librería para utilizar en Arduino y facilitar la recolección de datos.Además que no hay que olvidar que al estar creada por MATHWORKS, también cuenta con la capacidad de enviar datos a este programa para ser procesados a continuación.

Por estas razones y para obtener la lectura de los datos de sensores en el dispositivo se utiliza esta plataforma. Además que cuenta con una aplicación móvil donde solo necesita la llave del canal y el ID del mismo para que estos datos puedan ser visualizados en cualquier sitio.

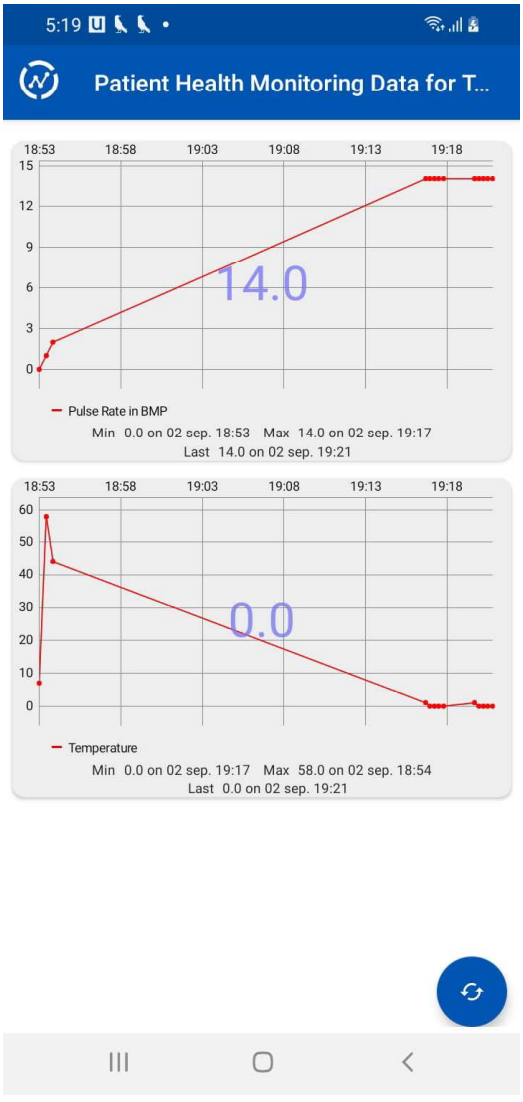


Figura 8. Captura de la interfaz de la aplicación Thingview

IV-F. Amazon Alexa

Alexa es el servicio controlado por voz más famoso, tal así que es muy usada sobre todo para aplicaciones de domótica. También es uno de los más sencillos de utilizar, ya que se parece a los asistentes Siri de Apple y el que pertenece a Google.

Alexa siempre está evolucionando y se van desarrollando muchas más aplicaciones en distintas áreas que van desde la industria de videojuegos hasta proyectos de investigación.

IV-F1. Alexa Echo Dot: El parlante Echo Dot es uno de los más pequeños diseñados para Alexa, por tanto uno de los más portables y fáciles de instalar.Al funcionar con Alexa siempre está evolucionando y agregando muchas más funciones.

Figura 9. Amazon Echo Dot

IV-G. Alexa Skills

Alexa Skills Kit (ASK) es un marco de desarrollo de software que permite crear contenido llamado habilidades o skills en inglés. Las habilidades funcionan como si se trataran de aplicaciones para Alexa. Mediante el uso de comandos de voz se pueden realizar distintas tareas como: revisar mensajes, reproducir música y jugar.

Estas habilidades también controlar los dispositivos conectados a la nube con su voz. Por ejemplo, el usuario pueden pedir a Alexa que encienda las luces o cambie la temperatura de un termostato.

IV-G1. Interacción con el usuario: La manera más sencilla para comenzar a interactuar con la habilidad de Alexa es la ïnvocación". Pero para ello primero se debe llamar a Alexa para así pedirle a esta que abra la aplicación.

El proceso completo se detalla de la siguiente manera:

1. El usuario dice "Alexa" a un dispositivo habilitado para Alexa.
2. El dispositivo envía audio al servicio en la nube de Alexa.
3. Se reconoce el idioma
4. Determinación del requerimiento del usuario.
5. Envío de solicitud de invocación una habilidad que puede cumplir con el requerimiento del usuario.

El servicio Alexa maneja el reconocimiento de voz y el procesamiento del lenguaje natural. Por su parte, las habilidades se ejecutan como servicios en una plataforma en la nube. Alexa se comunica con su skill mediante un

mecanismo de solicitud/respuesta a través de la interfaz HTTPS.

Cuando un usuario llama a su habilidad de Alexa, su habilidad recibe una solicitud POST con un cuerpo JSON. El cuerpo de la solicitud contiene los parámetros que su habilidad necesita para comprender la solicitud, realizar su lógica y generar una respuesta.

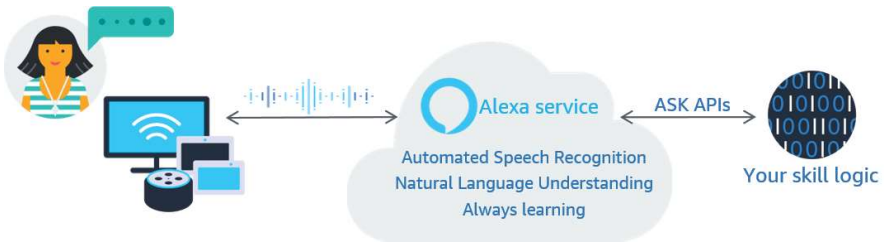


Figura 10. Funcionamiento de una Skill de Alexa

IV-H. Backendless

Es una plataforma de desarrollo de aplicaciones diseñada para servir tanto a los desarrolladores de aplicaciones individuales como a compañías que busquen el servicio. Ya que permite desarrollar aplicaciones de manera eficiente. Esto puede proporcionar soluciones integrales diseñadas para desarrollo web o móvil.

Consta de servicios de desarrollador que permitan facilitar la construcción del backend. Esto incluye administración de implementación, modelado de datos, desarrollo móvil, depuración y control de código fuente.

Cuenta también con un manejo de APIs básicas como: bases de datos, usuarios, archivos, geolocalización, mensajería, etc. Aunque permite al usuario definir las propias.

Figura 11. Backendless

Para este proyecto se utilizara un backend para Alexa que será configurado gracias a la función codeless de Backend.

IV-H1. Backendless Codeless: Es una alternativa para escribir código sin un lenguaje de programación. Se destaca por el uso de una interfaz intuitiva, la lógica de las tareas se expresa mientras se manipulan construcciones básicas de programación, como variables representadas por bucles, objetos y bloques gráficos. Los usuarios no necesitan conocer la sintaxis del lenguaje de programación ni comprender las complejidades de la programación de bajo nivel.

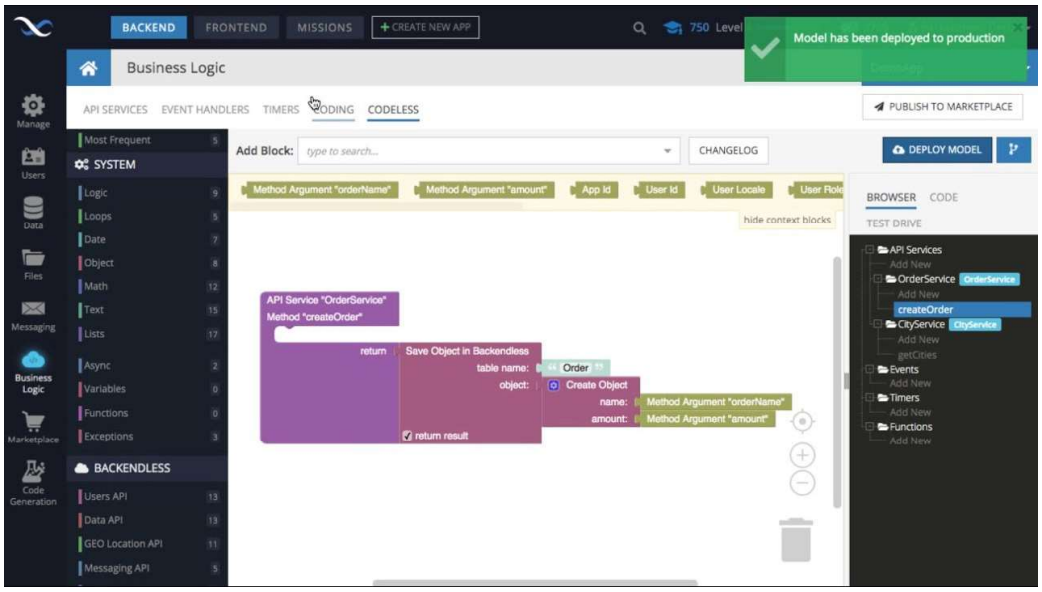


Figura 12. Backendless Codeless

IV-I. Arduino

Para empezar a hablar de Arduino hay que recordar que es actualmente una de las placas electrónicas más vendidas al rededor del mundo. Y es que destaca su precio, facilidad y flexibilidad para programar.

Al ser de software libre permite que cualquier usuario pueda modificar la placa (quitar o agregar componentes) o crear la suya propia utilizando los mismos recursos y así crear un microordenador que satisfaga sus requerimientos.

Esta placa consta de un microcontrolador y varios pines que ayudan a conectar sensores y actuadores.Además de ello el circuito que permite su funcionamiento se encuentra en una placa de PBC.

Existen distintos modelos de placas que van desde la más sencilla como el Arduino UNO hasta la más compleja como el Arduino MKR VIDOR 400. Esta variedad permite crear un sin fin de proyectos desde un pequeño semáforo hasta la automatización de una casa entera.

Figura 13. Logo de Arduino

IV-I1. Microcontrolador de Arduino: Para este proyecto se ha utilizado el microcontrolador que trae la placa UNO por defecto, el ATmega328P fabricado por Atmel.Este controlador tiene la particularidad de utilizar la tecnología ”Picopowerçuyo objetivo es reducir el consumo eléctrico de la placa.

Además presenta la arquitectura AVR, también creada por la compañía Atmel quien busca ser un rival para arquitecturas como la Pic de Microchip.

Figura 14. Microcontrolador ATmega328P

IV-I2. Arduino UNO: Es la placa predilecta para muchos programadores y desarrolladores. Posee catorce pines que pueden ser utilizados como entradas o salidas digitales, seis salidas analógicas, puerto USB, puerto jack de alimentación, terminales para conexión ICSP y un botón para reseteo. Esta tarjeta incluye salidas de 5V y 3.3V, además de trabajar con 5V, una memoria de 32K y para su alimentación se puede utilizar un voltaje que vaya desde 7V hasta 12V.

Para su programación se utilizan lenguajes C y C++ y posee su propia plataforma de para programar, el software más conocido como Arduino (IDE).

Figura 15. Arduino UNO

IV-I3. Arduino IDE: Integrated Development Enviroment, por sus siglas en inglés, es el sofware propio de Arduino. Creado como un conjunto de distas herramientas que permiten el desarrollo de todo tipo de aplicaciones.

Su estructura básica consiste en un editor de código, un compilador, un depurador y un generador de interfaz gráfica de usuario (GUI). Para Arduino, también incluye herramientas para cargar programas compilados en la memoria flash del hardware.

Como Arduino acostumbra esta plataforma está construida en Software Libre, es compatible con varios sistemas operativos como Windows, Mac OS, Linux.

IV-J. Sensores utilizados

IV-J1. Módulo ECG AD8232 Monitor Cardiaco: Para este proyecto se decidió hacer uso del módulo ECG AD8232, por su bajo costo y compatibilidad con Arduino. Este sensor trabaja con señales analógicas y devuelve

el registro de un electrocardiograma.

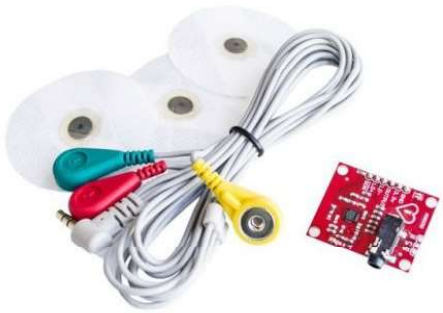


Figura 16. Módulo ECG AD8232 Monitor Cardiaco

IV-J2. Termocupla WZP-PT100: Este sensor es de tipo termoresistivo, es decir corresponde a una resistencia térmica. Tiene una cupla de acero inoxidable que le permite ser sumergible y brinda resistencia contra el agua. Su característica principal es que se maneja en una escala de 0 ohmnios a 100 ohmnios, característica que le da nombre.

Figura 17. Termocupla WZP-PT100

IV-J3. Módulo WiFi Esp-01: Es un módulo WiFi que será usado para enviar los datos de los sensores a la nube. Su característica principal es el poseer un microchip ESP8266, con una arquitectura de 32 bits y la razón principal por la que se realiza la conexión WiFi. Para programar este módulo se hace uso de su firmware AT,aunque cambiando su firmware se puede aumentar su capacidad para utilizarlo como un microcontrolador de 32bits a 80Mhz.

Este módulo trabaja con 3.3V por ello se utilizan dos resistencias, una de 2.2k y otra de 1k para reducir el voltaje de la señal que se obtiene del Aruino.

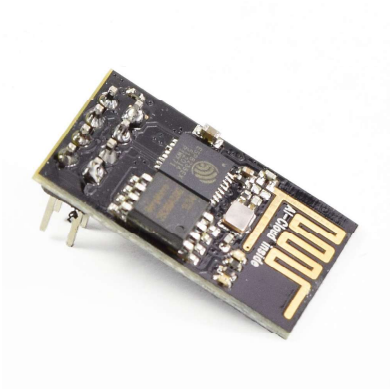


Figura 18. Módulo WiFi

IV-J4. Buzzer: Conocido también como zumbador, funciona como un conversor de energía eléctrica en sonido.Es decir cumple la función de transductor. Se utiliza como alarma y si se ajusta la frecuencia correcta se podría crear melodías.

Para este proyecto se utilizará como seguimiento del pulso que registra el módulo ECG AD8232.

Figura 19. Buzzer

IV-J5. Pluggable Terminal: El terminal enchufable es un sensor de temperatura que se puede usar en muchos lugares, como la temperatura del piso, el monitoreo de la temperatura del tanque de agua caliente, el sensor de temperatura a prueba de agua DS18B20 también debe conectarse a la resistencia pull-up.

Para este proyecto se lo utilizara para medir la temperatura de una persona.

Figura 20. Puggable

IV-J6. MLX90614: El MLX90614 es un sensor de temperatura infrarrojo sin contacto fabricado por Melexis. Estos sensores se pueden conectar a autómatas o procesadores como Arduino para medir la temperatura de los objetos a distancia.

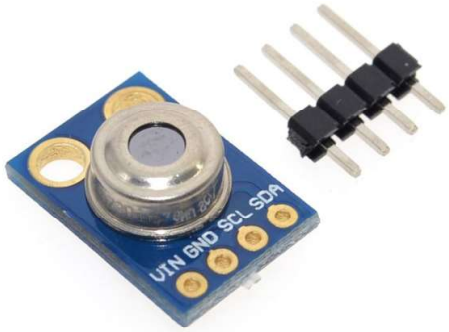


Figura 21. MLX90614

IV-J7. MAX30100: MAX30100: El MAX30100 es un oxímetro de pulso integrado y un monitor de frecuencia cardíaca. Tiene dos LEDs: un LED rojo (660 nm) y un LED infrarrojo (920 nm), un fotodetector, óptica especial, filtro de luz ambiental de 50 a 60 Hz y convertidor ADC delta-sigma de 16 bits, hasta 1000 muestras por segundo. También tiene un sensor de temperatura interno para compensar la temperatura.

Figura 22. MAX30100

V. MARCO METODOLÓGICO

V-A. Instrumentos de Recolección de Datos

Un instrumento de recolección de datos es una herramienta que ayuda al investigador a recopilar de mejor manera la información que necesita para su investigación. Comprenden un sinfín de instrumentos que son mayoritariamente clasificados en cuantitativos y cualitativos. Los cualitativos suelen ser más criticados en base a que son presunta falta de objetividad, la imposibilidad de reproducción de sus datos y la falta de validez. [7]

En este trabajo la recolección de datos será realizada por los sensores que formarán parte del diseño de prototipo en el monitor creado con Arduino.

V-B. Herramientas para el Procesamiento de Datos

Para este trabajo se utilizará la consola de Blackendless, la plataforma para mostrar datos en la nube será Thingspeak y para la recepción y manejo de datos se utilizará el IDE de Arduino. Finalmente para mostrar los datos se usará una skill de Alexa a la que se ha llamado ”TT2_Patient_Monitor

El uso de estas herramientas será descritas a continuación.

V-B1. Arduino: Para este proyecto se ha decidido hacer la adquisición de datos por medio de la placa Arduino UNO, el código utilizado se detallará a continuación:

V-B2. Código:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>


#define RX 2
#define TX 3

#define DEBUG true

String AP = "Sala Arriba"; // "SSID-WiFiname"
String PASS = "Silvia98"; // "password"
String API = "LK8DQBGV3UITAF7F";
String HOST = "api.thingspeak.com";
String PORT = "80";

float temperature;

SoftwareSerial espSerial(RX,TX);

//Crear el objeto lcd  direcci n  0x3F y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x3F,16,2);  //

// Variables
int pulsePin = A0; // Pulse Sensor purple wire connected to analog pin 0
int blinkPin = 6 ; // pin to blink led at each beat
int onPin = 13;
const int pinBuzzer = 4;

String espData(String command, const int timeout, boolean debug)
{
  Serial.print("AT Command ==> ");
  Serial.print(command);
  Serial.println("     ");

  String response = "";
  espSerial.println(command);
```

```
  long int time = millis();
  while ( (time + timeout) > millis())
  {
    while (espSerial.available())
    {
      char c = espSerial.read();
      response += c;
    }
  }
  if (debug)
  {
    //Serial.print(response);
  }
  return response;
}



void setup()
{
Serial.begin(9600);
pinMode(12, INPUT); // Setup for leads off detection LO +
pinMode(11, INPUT); // Setup for leads off detection LO -

 // Inicializar el LCD
  lcd.init();

  //Encender la luz de fondo.
  lcd.backlight();
pinMode(blinkPin,OUTPUT); // pin that will blink to your heartbeat!
pinMode(onPin,OUTPUT); // pin that will fade to your heartbeat!
digitalWrite(onPin,HIGH);
pinMode(pinBuzzer, OUTPUT);

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Patient Health");
lcd.setCursor(0,1);
lcd.print(" Monitoring ");
delay(4000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Initializing....");
delay(5000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Getting Data....");
Serial.begin(9600);
lcd.print("Connecting...");
Serial.begin(9600);
espSerial.begin(115200);
  espData("AT+RST", 1000, DEBUG);                      //Reset the ESP8266 module
  espData("AT+CWMODE=1", 1000, DEBUG);                 //Set the ESP mode as station mode
  espData("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"", 1000, DEBUG);
   delay(1000);
}

void loop() {
lcd.clear();
lcd.setCursor(0,0);
lcd.print("BPM :");
lcd.setCursor(7,0);
lcd.print(pulsePin);
```

```
lcd.setCursor(0,1);
lcd.print("Temp.:");
lcd.setCursor(7,1);
lcd.print(temperature);
lcd.setCursor(13,1);
lcd.print("C");

String readDat="GET https://api.thingspeak.com/channels/1822679/feeds.json?api_key=9
    OOUIBWMZLWCJGVG&results=2";
String sendData = "GET /update?api_key="+ API +"&field1="+getTemperatureValue()+"&field2="+
    getpulsePinValue();
 espData("AT+CIPMUX=1",5,"OK");
 espData("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 espData("AT+CIPSEND=0," +String(sendData.length()+4),1000,DEBUG);
    espSerial.find(">");
    espSerial.println(sendData);

    espData("AT+CIPCLOSE=0",1000,DEBUG);
    delay(10000);
}

String getpulsePinValue(){
  if((digitalRead(11) == 1)||(digitalRead(12) == 1)){
Serial.println("NO PULSO");
}
else{
// send the value of analog input 0:
Serial.println(analogRead(A0));
digitalWrite(blinkPin,HIGH);
delay(1000);
digitalWrite(blinkPin,LOW);
delay(1000);
digitalWrite(pinBuzzer, HIGH);    // Ponemos en alto(5V) el pin del buzzer
delay(1000);                      // Esperamos 1 segundo
digitalWrite(pinBuzzer, LOW);     // Ponemos en bajo(0V) el pin del buzzer
}
//Wait for a bit to keep serial data from saturating
delay(1000);
}

String getTemperatureValue(){
int temp = analogRead(A1);
  temperature = temp*(5/1023);
  delay(1000);
  Serial.println(temperature);
}
```

V-B3. Monitor Serial: Los resultados se han configurado para mostrarse en 9600 baudios y se muestran en el monitor serial como indica la siguiente figura:

Figura 23. Captura de monitor serial

V-C. ThingSpeak

La plataforma de ThingSpeak se caracteriza por su fácil uso y por ello se la ha elegido para que sea un intermediario entre los datos entregados por Arduino y los datos entregados a Alexa.

Para trabajar con esta plataforma el primer paso es contar con una cuenta de MATLAB, caso contrario se debe crear una. Una vez que se tenga una cuenta activa se continuará con el proceso que se detallará a continuación.

V-C1. Creación de canal: El primer paso para utilizar esta plataforma es la creación de un canal.Para ello se irá a la pestaña ”New Channel”.

En ella se encontrarán los parámetros que debe tener el canal que son: nombre, descripción, nombre de campos y cuantos se desean activar, enlaces a proyectos externos, etc.

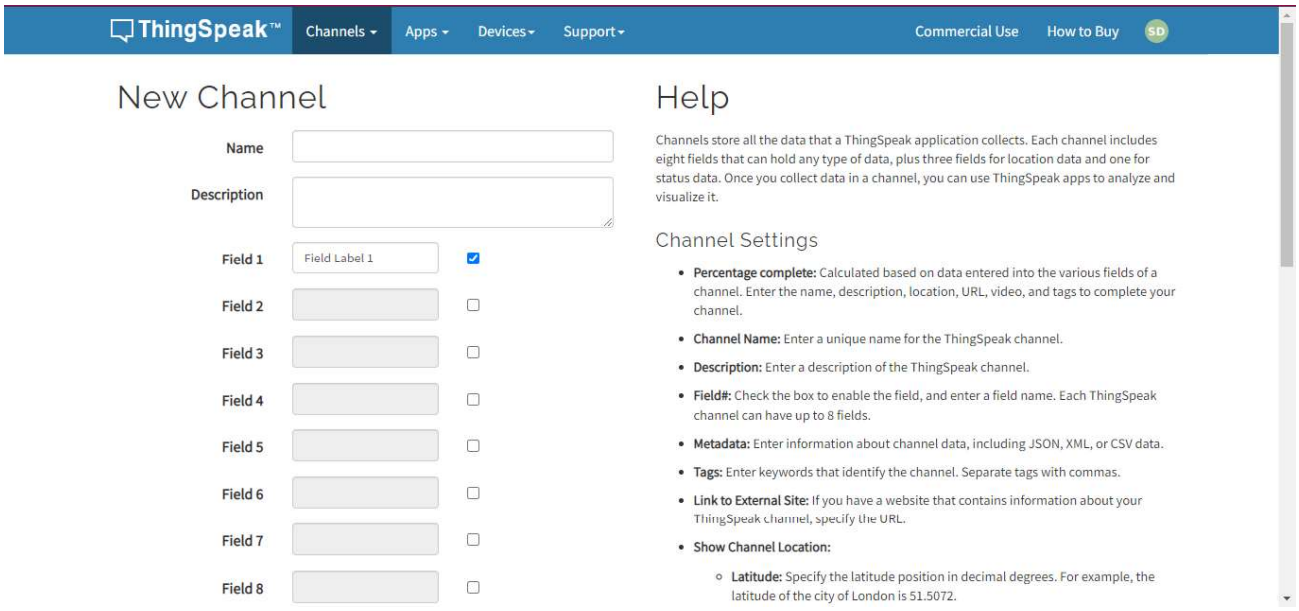


Figura 24. Ventana de creación de Canal

Para el caso del proyecto se realizaron los siguientes cambios en los parámetros del canal:

1. Cambiar el nombre del canal a: Patient Health Monitoring Data for TT2
2. Activar el primer canal y renombrarlo a: Pulse Rate in BMP
3. Activar el segundo canal y renombrar a: Temperature

Los cambios realizados se pueden verificar en ajustes del canal, donde también de ser necesario se pueden agregar nuevos canales dependiendo de las variables con las que se vaya a trabajar.

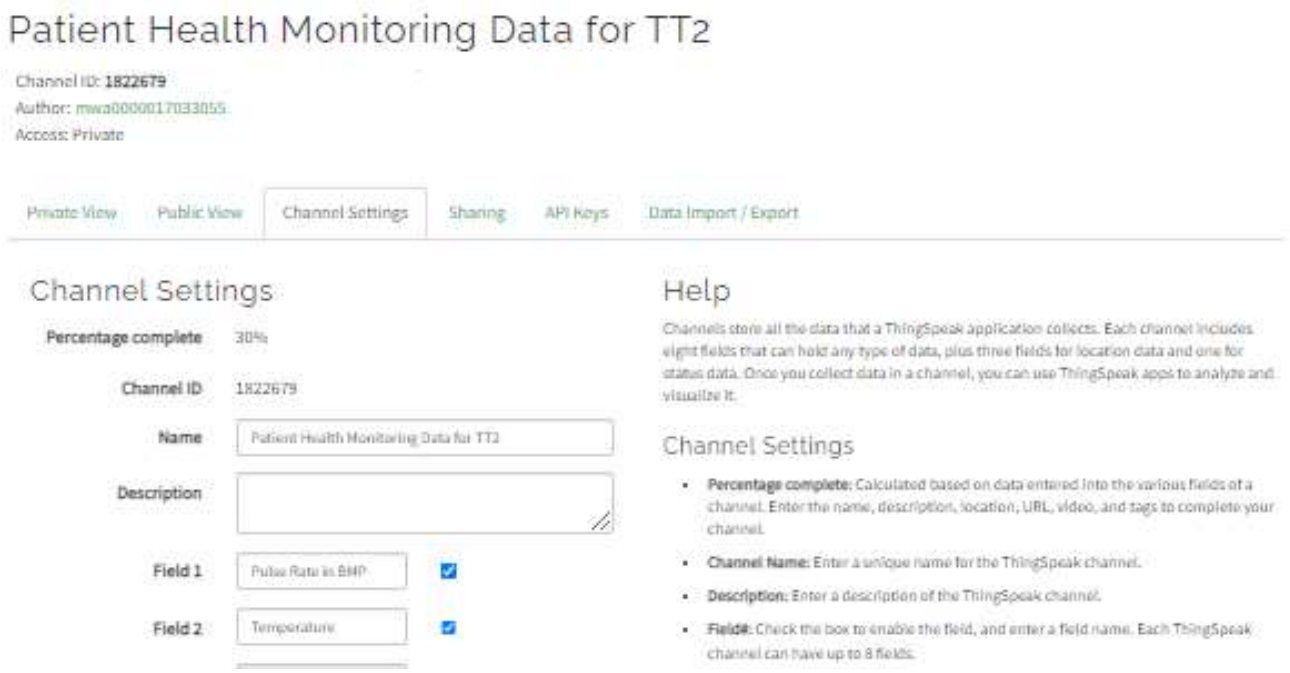


Figura 25. Parámetros del Canal

V-C2. Visualizaciones en el canal: Para poder observar los datos adquiridos, ThingSpeak cuenta con widgets que permiten que los datos se muestren de manera dinámica, además de ser personalizables.
En la siguiente imagen se muestra como se ve el canal creado para el proyecto.

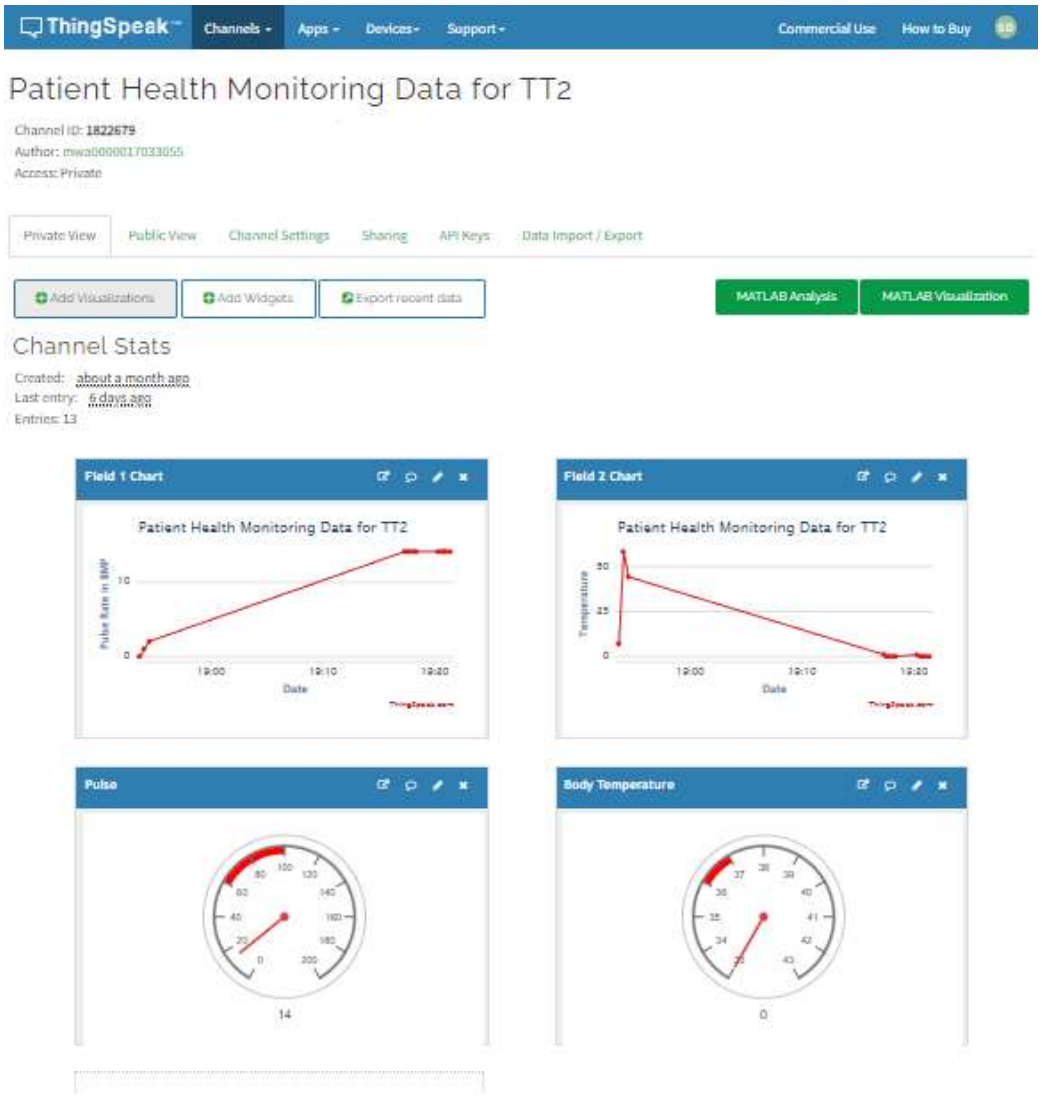
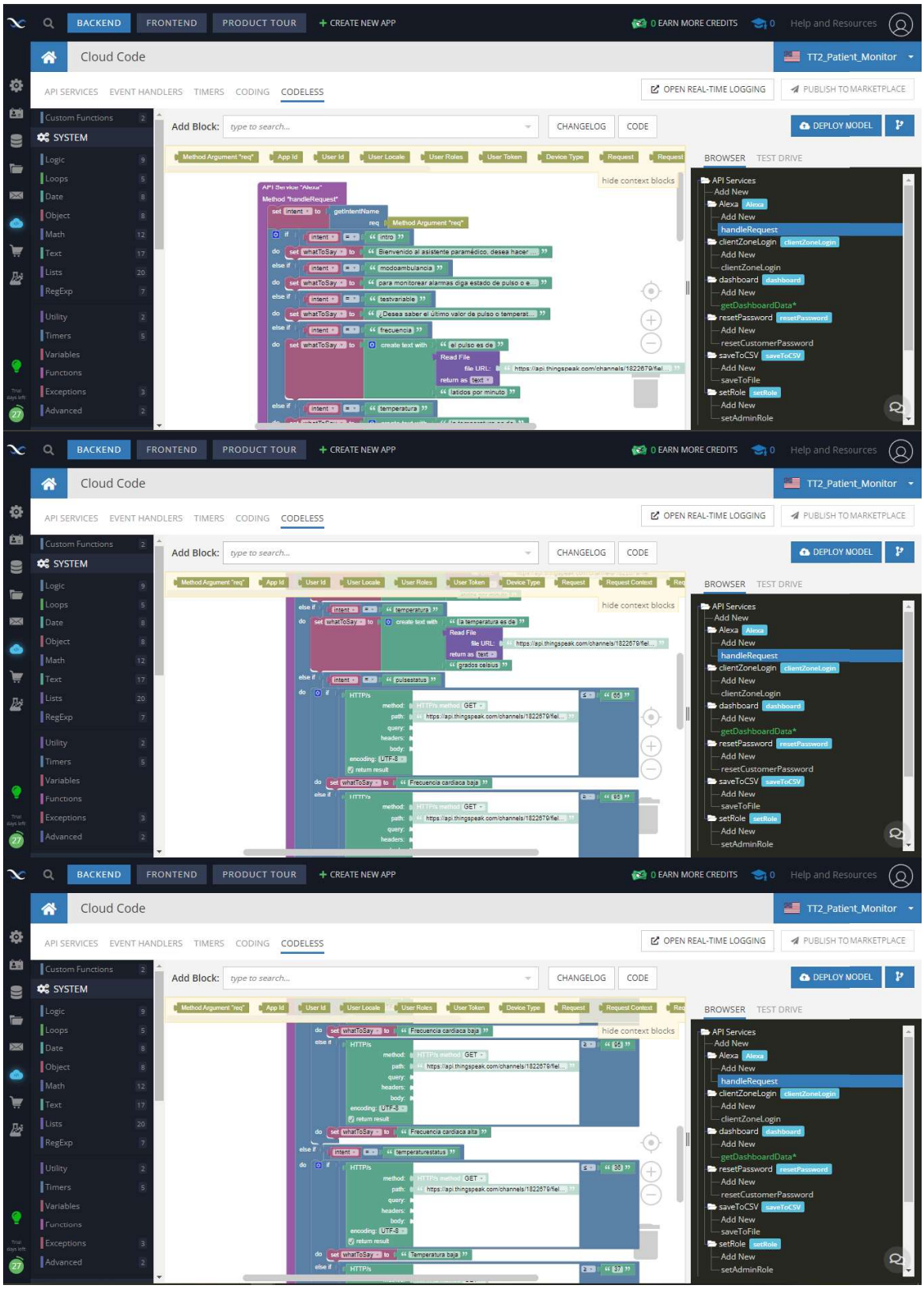


Figura 26. Ventana de visualización de Canal

V-D. Blackenless

El código main en el formato Codeless se mostrará a continuación y como se puede observar se hizo de manera sencilla y que se pudiera acceder a la información sin tener que requerir todo el proceso de una vez. Es decir que el usuario una vez que haya llamado a la skill podrá preguntar el estado de la temperatura y el pulso y su valor sin tener que escuchar de nuevo la introducción o las instrucciones para llevar a cabo.


BACKEND
FRONTEND
PRODUCT TOUR
+ CREATE NEW APP
0 EARN MORE CREDITS
Help and Resources
Cloud Code
TT2_Patient_Monitor
API SERVICES
EVENT HANDLERS
TIMERS
CODING
CODELESS
OPEN REAL-TIME LOGGING
PUBLISH TO MARKETPLACE
Add Block:
type to search...
CHANGELOG
CODE
DEPLOY MODEL
SYSTEM
Logic
Loops
Date
Object
Math
Text
Lists
RegExp
Utility
Timers
Variables
Functions
Exceptions
Advanced
hide context blocks
BROWSER
TEST DRIVE
API Services
Alexa
handleRequest
clientZoneLogin
dashboard
getDashboardData*
resetPassword
resetCustomerPassword
saveToCSV
saveToFile
setRole
setAdminRole

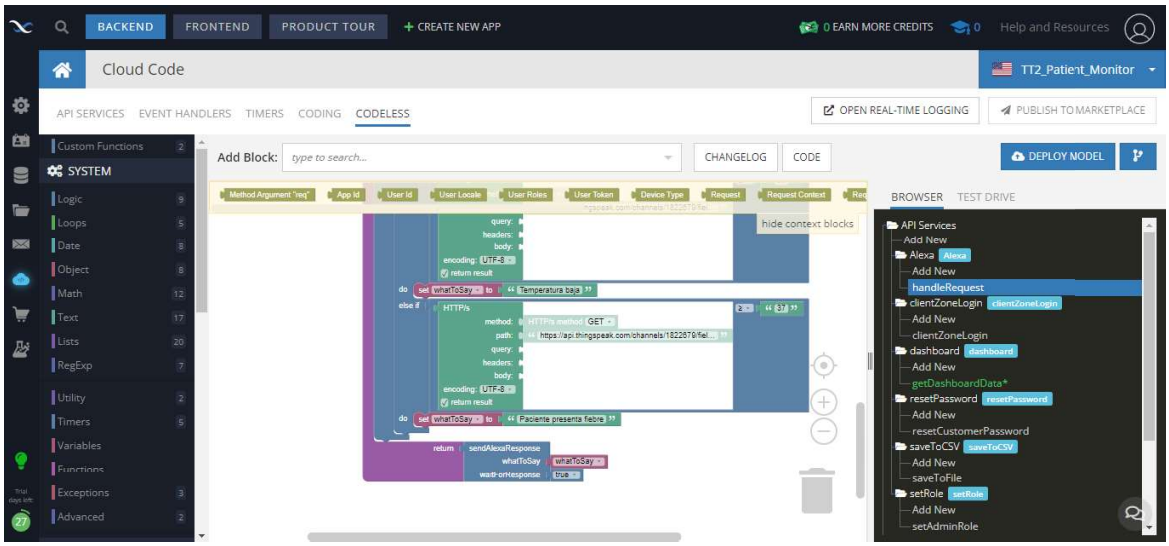


Figura 27. Código main

V-D1. Función getIntentName: Dentro de la programación de la skill de Alexa existe una variable conocida como ïntent.[es]ta sirve más que nada para activar una sub función de la propia Alexa. En este caso los intent se utilizan para poder preguntar sobre el estado del paciente o los valores de las variables del monitor.

Para llamarlos se utiliza la siguiente función.

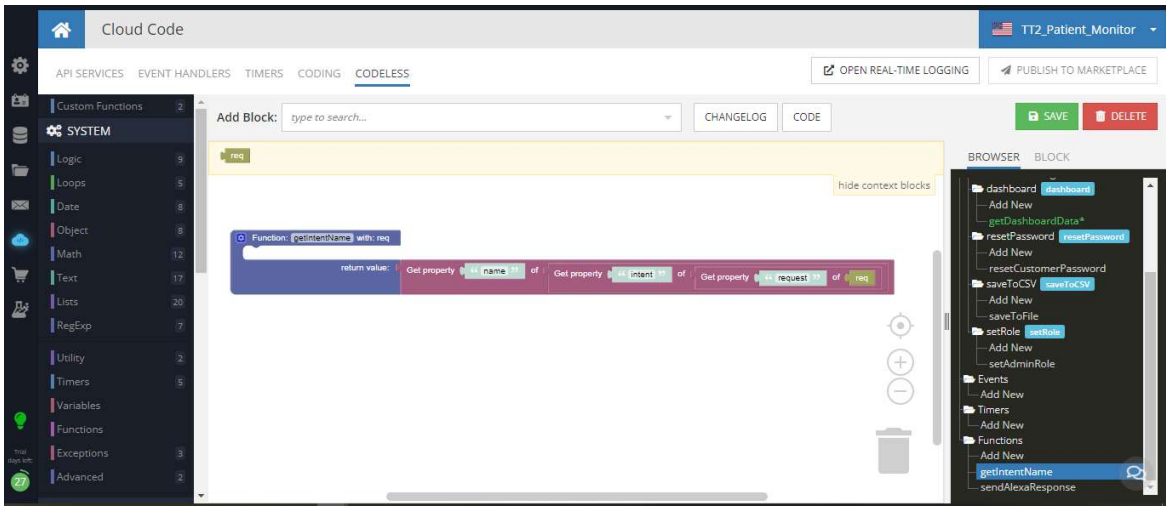


Figura 28. getIntentName

V-D2. Function sendAlexaResponse: Si ya se tienen declarados los intents y la función de cada uno de ellos, el siguiente paso es emitir una respuesta a lo que el usuario pida. Es por ello que esta función se utiliza para hacer que Alexa responda con la configuración descrita en el código principal.

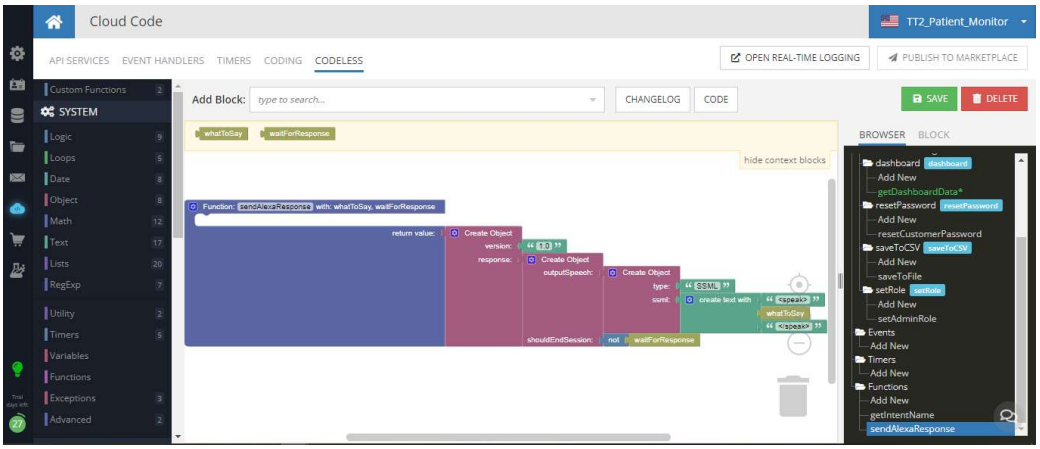


Figura 29. sendAlexaResponse

V-D3. Deploy Model: Este es un botón en forma de nube que se encuentra en la parte derecha del entorno de Blackendless, con ello cuando las funciones estén guardadas y el programa no tenga ningún bloque que no está conectado, se podrá enviar a una dirección en la nube que posteriormente se añadirá al entorno de Alexa Skills para que sea ejecutada.

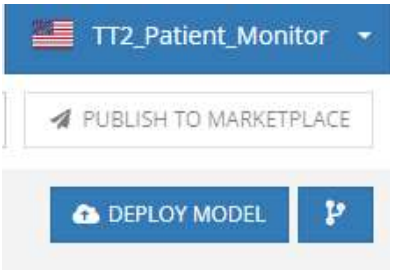


Figura 30. Function Deploy Model

V-E. Alexa Skills Developer

Hay que tener en cuenta que el primer paso para crear una skill es el de crear una cuenta a parte de la que se tiene en Amazon (y que por defecto está añadida a Alexa).

Una vez hecho esto el entorno de creación de skills estará listo para comenzar.

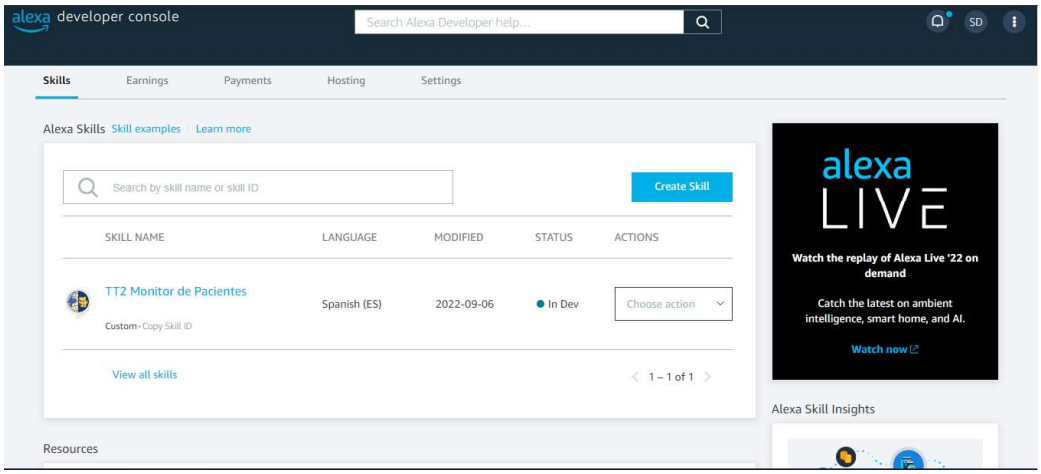


Figura 31. Consola de creación de skills de Alexa

V-E1. Parámetros iniciales: Una vez abierta la consola se procede a crear una nueva skill, precionando el botón en la parte superior derecha de la pantalla. Una vez ahí se nos pedirá llenar algunos parámetros: nombre de la skill, idioma, si se desea utilizar una plantilla ya creada o crear una y la última y más importante para que el proyecto funcione que es el almacenaje.

Ya que la dirección de API y el almacenaje son de Blackendless, se procede a escoger la opción de proporcionado por el usuario como está descrito en las imágenes a continuación.

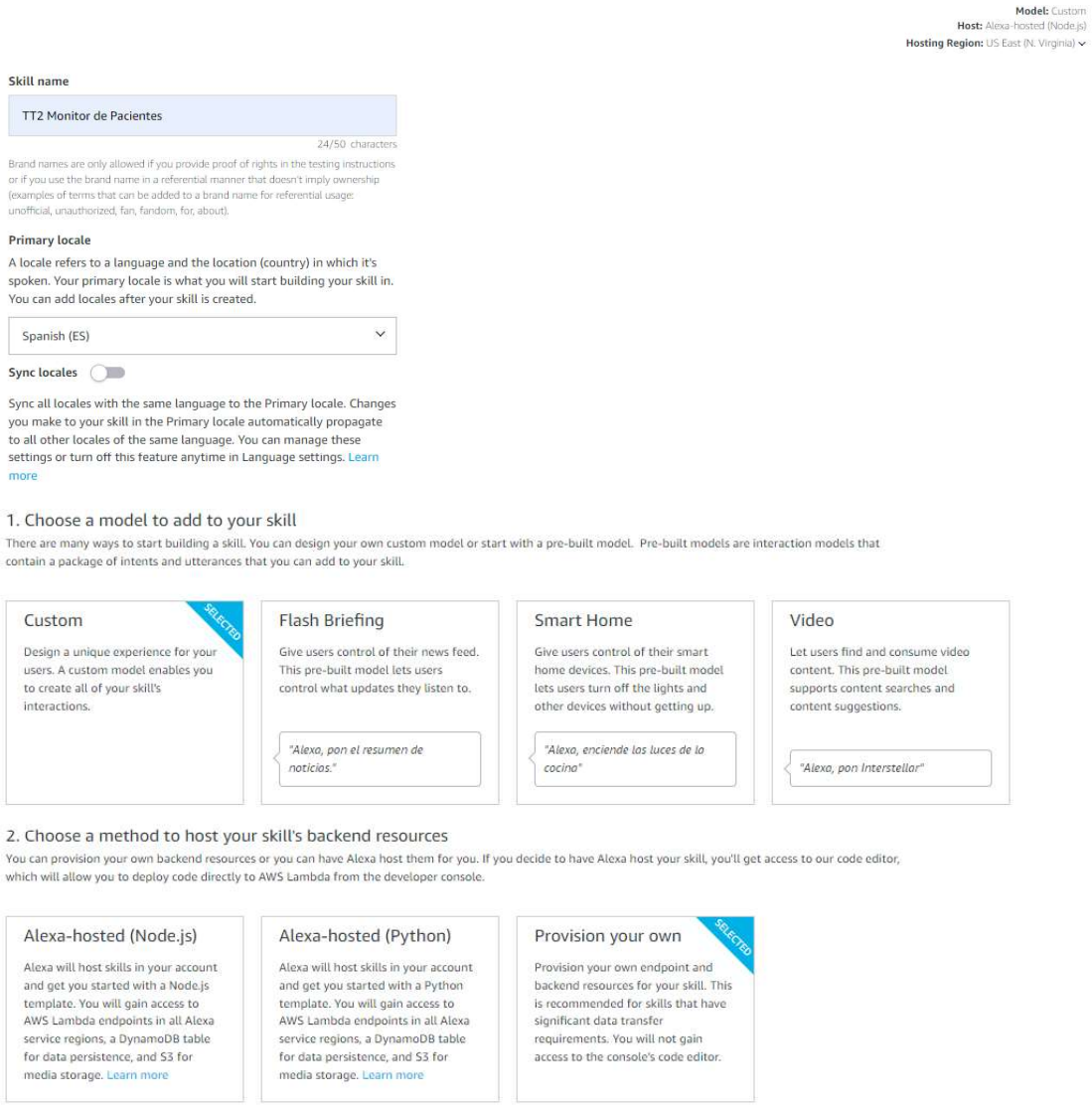


Figura 32. Parámetros de skills de Alexa

V-E2. Nombre de Invocación: El siguiente paso es elegir un nombre para activar la skill, es decir que cuando el usuario diga .ªlexa"le siga un conjunto de palabras que permita que Alexa reconozca la skill y la abra.

Este proyecto utiliza a Alexa como asistente paramédico así que ese fue el nombre que se le asignó a la invocación.

Figura 33. Nombre para invocación de la skill de Alexa

V-E3. Intents: Como se mencionó en el capítulo anterior, los intents son aquellos que permiten a Alexa saber a qué parámetro quiere acceder el usuario

Constan de dos partes: su nombre y un conjunto de expresiones que puede decir el usuario para activarlas. Es decir estas expresiones tienen que estar diseñadas para que el usuario pueda hacer uso de ellas en una conversación fluida.

Alexa ya trae consigo una plantilla para realizar pruebas (HelloWorld) y otros intents por defecto. Para este proyecto se han creado los intents:

- frecuencia
- temperatura
- intro
- testvariable
- modoambulancia
- pulsestatus
- temperaturestatus

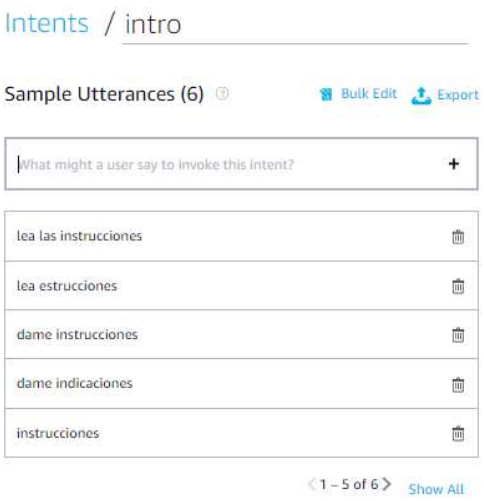


Figura 34. Ejemplo de un intent creado para la skill de Alexa

V-E4. Código para trabajar en Alexa Skills Developer: Debido a que la skill se desarrolla en Backendless no se permite editar el código en la ventana del desarrollador.

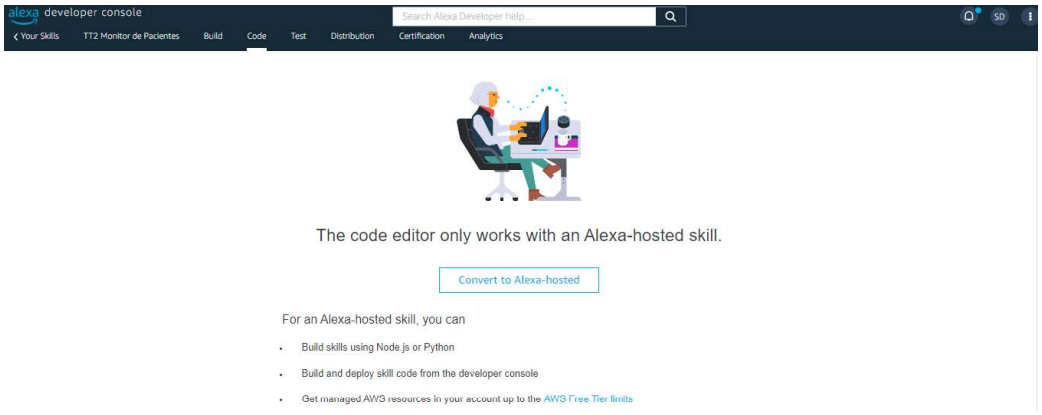


Figura 35. Impedimento de trabajar con código en la ventana del desarrollador de Skills

V-E5. Prueba de Skill: Alexa cuenta con un apartado para realizar pruebas con la skill, a continuación se muestra una prueba del funcionamiento de la skill creada.

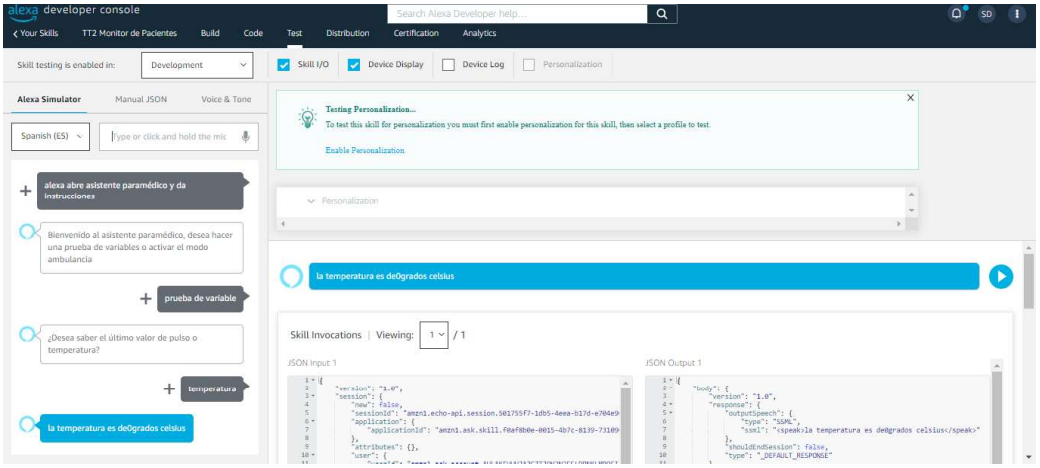


Figura 36. Prueba de funcionamiento de la skill

Para terminar el proceso de creación de skills, se dan parámetros finales como imagen, descripción de la skill, categoría, etc.

V-F. Monitor con Arduino

Para la realización de este proyecto se ha decidido implementar un monitor multiparámetros que tenga como controlador una tarjeta Arduino UNO.

Para ello se utilizan los sensores de pulso ECG AD8232 y el sensor P100, por medio del módulo ESP-01 se realiza la conexión con internet, así como muestra el siguiente gráfico:

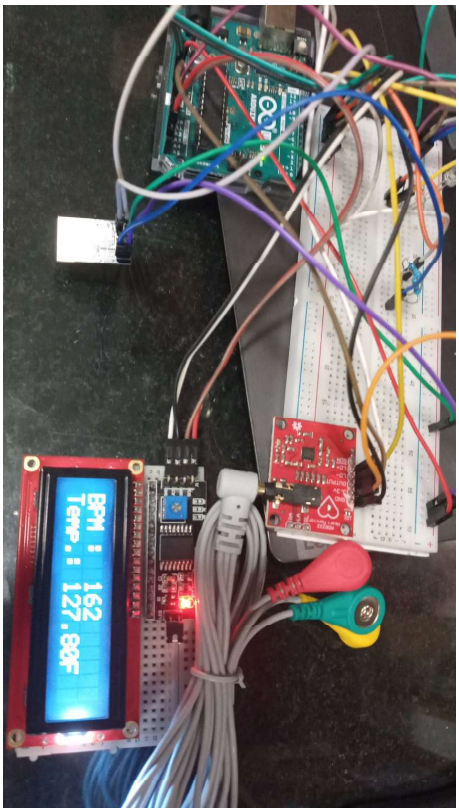


Figura 37. Monitor conectado

Para luego el circuito se colocó en una caja hecha con PLA y diseñada en Autodesk inventor como se ve en la siguiente imagen.

Figura 38. Caja de Monitor

VI. RESULTADOS

VI-A. Respuesta del monitor multiparámetros

Los resultados de los sensores se pueden visualizar en la pantalla LCD del dispositivo como se muestra a continuación:

Figura 39. Valores mostrados en pantalla de dispositivo

VI-B. Conexión a internet

Mediante el módulo Wifi se configuró la red Sala Arriba, para realizar las pruebas correspondientes.

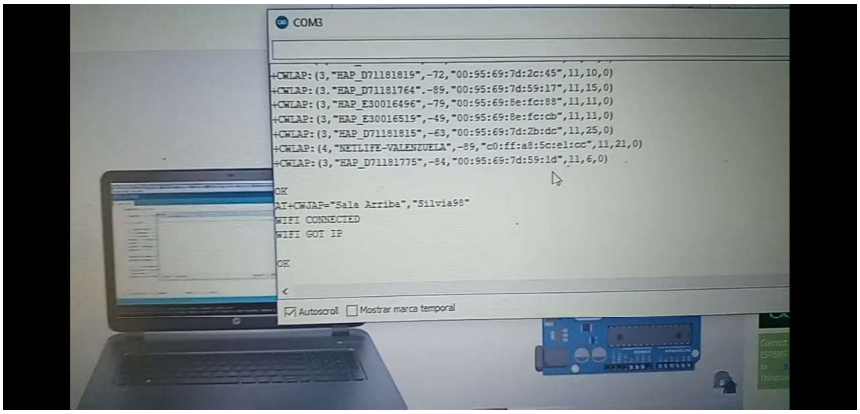


Figura 40. Conexión a Red

VI-B1. Skill finalizada: Después de realizar el proceso descrito anteriormente, lograr las conexiones entre aplicaciones con éxito y testear la skill finalizada, el resultado del proyecto será la skill lista para su funcionamiento y distribución, misma que en la tienda de AMAZON ALEXA SKILLS tendrá la apariencia que se muestra a conticuación.

Figura 41. Skill de Alexa finalizada

VI-B2. Análisis de Vulnerabilidades: Se realizó un escaneo al equipo en búsqueda de algún tipo de vulnerabilidad que pueda afectar directamente al equipo, sin embargo, no se ha encontrado ningún tipo de amenaza existente, para la búsqueda se ha utilizado el sistema Linux de una máquina virtual.

VI-B3. Obtención Resultado (Temperatura): Se realizó la prueba de toma de temperatura con la termocupla y un termometro digital para hacer la comparación de los resultados y compararlos. En las imagenes se muestran los resultados obtenidos.

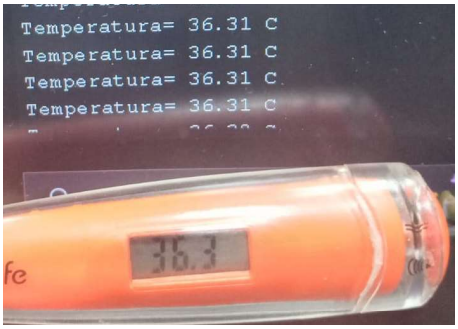


Figura 42. Skill de Alexa finalizada

VII. CRONOGRAMA

A continuación se muestra el cronograma de trabajo en la figura 43.

Figura 43. Cronograma

VIII. PRESUPUESTO

Nombre del elemento	Descripción	Cantidad	Valor total
PT100 Sensor Temp 3h	Sensor de temperatura de 3 hilos WZP3C PT100	1	10,20$
SFM27 3-24V DC	Buzzer de 3 a 24 V	1	1,10 $
HD-PS-9V	Adaptador de voltaje	1	6 $
ESP-01	Módulo Wifi	1	4,50 $
Ad8232	Sensor de pulso	1	16$
LCD	Pantalla	1	3,50 $
Arduino UNO	Placa Arduino y cable	1	11,99 $
Amazon ECHO	Parlante ECHO	1	39,99 $
Impresión 3D	16 horas en PLA	1	25 $
Luces led	5V	2	0,40 $
Jumpers	cables jumpers	40	4 $
Resistencias	2.2K y 1K	2	0.40 $
SUBTOTAL PROTOTIPO			112,88$
Envío de componentes local		1	3,50$
Envío de componentes exterior		1	25$
transporte	transporte publicó y gasolina		200$
Alimentación			250$
Horas de trabajo		350 horas	350$
SUBTOTAL COSTOS LOGÍSTICOS			828,50$
TOTAL(MAQUETA + COSTOS LOGÍSTICOS)			941,38$

IX. CONCLUSIONES

Se diseñó una habilidad de Alexa que fuera capaz de devolver datos a los paramédicos, además se aprovechó las facilidades de ThingSpeak para que estos sean visibles por varios dispositivos al mismo tiempo y que no necesariamente se encuentren en el mismo lugar. Esto puede dar paso a un sistema de monitoreo directamente enlazado con el hospital y de esta forma omitir el paso en donde el hospital pregunte sobre el estado del paciente sino que ya lo sepa de antemano antes de que el paciente arribe.

De momento el dispositivo puede establecer conexión con otros dispositivos que funcionen con Alexa, es decir que esto abarca desde los equipos diseñados por Amazon hasta dispositivos móviles o computadoras en donde opere la aplicación.

Según lo consultado con algunos parámedicos, el proyecto beneficia más al paciente y al hospital. Esto debido a que no se puede tener en cuenta variables como: duración de viaje, diferencias entre enfermedades crónicas y accidentes y demás variables que son tratadas por el paramédico. Por el contrario para el hospital resulta muy útil, ya que les permite tener tanto a especialistas esperando como el conocer del estado del paciente remotamente.

X. RECOMENDACIONES

Este proyecto funciona con internet y este es uno de sus puntos débiles. Cuando el vehículo se encuentre en un túnel, existe un fallo de conexión en la red o en el lugar no se cuenta con internet; el proyecto no va a resultar.

De momento Alexa trabaja con un tiempo de espera de cinco segundos en su modo de seguimiento, esto quiere decir que el usuario tiene cinco segundos para responder. De no ser así la skill se cerrará. Esto no puede ser alterado con un delay, ya que solo incrementaría el tiempo en que Alexa vuelva a preguntar. Se necesita investigar más a fondo este modo.

REFERENCIAS

[1] R. Lucio, N. Villacrés y R. Henríquez, *Sistema de Salud de Ecuador*, jun. de 2011.

[2] R. S. Nair, J. M. Lawrence, A. Vyakaranam y col., «Recent Trends and Opportunities of Remote Multi-Parameter PMS using IoT,» IEEE, abr. de 2021, págs. 325-329, ISBN: 978-1-6654-0514-0. DOI: 10.1109/EIConCIT50028.2021.9431926.

[3] ——, «Recent Trends and Opportunities of Remote Multi-Parameter PMS using IoT,» en *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, IEEE, 2021, págs. 325-329.

[4] S. Piyawat y S. Pullteap, *Fiber Optic Sensor Applications for Vital Signs Monitoring: A Review*, 2019.

[5] J. E. Ramirez Reyes, «Diseño e implementación de guıas rápidas y plan de capacitaciónes para el buen manejo de equipos médicos en la central de urgencias Louis Pasteur,» 2021.

[6] I. P. Romero Hernandez, L. Gaviria Arango y col., «Diseño e implementación de un plan de aseguramiento metrológico para equipos biomédicos de la clınica Colsubsidio Calle 100 y el centro médico de especialistas de la calle 63,» 2018.

[7] M. J. Sánchez, M. Fernández y J. C. Diaz, «Técnicas e instrumentos de recolección de información: análisis y procesamiento realizado por el investigador cualitativo,» *Revista cientıfica UISRAEL*, vol. 8, n.º 1, págs. 107-121, 2021.

[8] A. Surely, «An Affect as Interaction Approach for Stress Management Among Paramedics,» en *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, IEEE, 2021, págs. 1-5.

[9] M. E. C. Valenzuela, M. V. Garcıa, A. B. Jacobo, G. H. Ruiz, N. C. B. Herrejón y C. E. O. Perkins, «Evaluación del efecto de un programa de ejercicio fısico sobre la capacidad cardiorrespiratoria en académicos de la Universidad de Sonora con sındrome metabólico: un estudio piloto,» *Retos: nuevas tendencias en educación fısica, deporte y recreación*, n.º 44, págs. 264-275, 2022.

A. *Código traducido a JavaScript de Blackendless Codeless*

```
1
2   morekeywords={typeof, new, true, false, catch, function, return, null, catch, switch, var, if
        , in, while, do, else, case, break},
3   morecomment=[s]{/*}{*/},
4   morecomment=[l]//,
5   morestring=[b]",
6   morestring=[b]'
7 }
8
9 var whatToSay, intent;
10
11 function encodePath(path) {
12   if(path.startsWith("http://") || path.startsWith("https://")) {
13     return path
14   }
15
16   let decodedPath
17   try {
18     decodedPath = decodeURI(path)
19   } finally {
20     return (decodedPath || path).split("/").map(encodeURIComponent).join("/")
21   }
22 }
23
24 function readFileContent(encoding, result) {
25     if (!encoding) {
26       if (typeof Buffer !== 'undefined') {
27         return result instanceof Buffer
28           ? result.toJSON().data
29           : Buffer.from(JSON.stringify(result)).toJSON().data
30       }
31
32       return result
33     }
34
35     if (result && typeof result === 'object') {
36       return JSON.stringify(result)
37     }
38
39     return result
40 }
41
42
43   intent = (await require('codeless-function-e2047c8fc9df60733a043b01278ec615')(req));
44
45   if (intent == 'intro') {
46     whatToSay = 'Bienvenido al asistente param dico, desea hacer una prueba de variables o
          activar el modo ambulancia';
47
48   } else if (intent == 'modoambulancia') {
49     whatToSay = 'para monitorear alarmas diga estado de pulso o estado de temperatura';
50
51   } else if (intent == 'testvariable') {
52     whatToSay = ' Desea  saber el  ltimo  valor de pulso o temperatura?';
53
54   } else if (intent == 'frecuencia') {
55     whatToSay = ['el pulso es de',(await Backendless.Request.get((function(url){ if( !url ) {
          throw new Error('Url must be specified.')} if( !url.startsWith('http://') && !url.
          startsWith('https://')) { return 'https://' + url } return url})(encodePath('https://
          api.thingspeak.com/channels/1822679/fields/field1/last.html?api_key=9OOUIBWMZLWCJGVG'))
```

```
        ).setEncoding('utf8').then(result => readFileContent('utf8', result))),'latidos por
        minuto'].join('');
56
57    } else if (intent == 'temperatura') {
58      whatToSay = ['la temperatura es de',(await Backendless.Request.get((function(url){ if( !url
            ) { throw new Error('Url must be specified.')} if( !url.startsWith('http://') && !url.
            startsWith('https://')) { return 'https://' + url } return url})(encodePath('https://
            api.thingspeak.com/channels/1822679/fields/field2/last.html?api_key=9OOUIBWMZLWCJGVG'))
            ).setEncoding('utf8').then(result => readFileContent('utf8', result))),'grados celsius
            '].join('');
59
60    } else if (intent == 'pulsestatus') {
61      if ((await Backendless.Request['get']((function(url){ if( !url ) { throw new Error('Url
            must be specified.')} if( !url.startsWith('http://') && !url.startsWith('https://')) {
            return 'https://' + url } return url})('https://api.thingspeak.com/channels/1822679/
            fields/field1/last.html?api_key=9OOUIBWMZLWCJGVG')).setEncoding('utf8').send()) <=
            '65') {
62        whatToSay = 'Frecuencia cardiaca baja';
63
64      } else if ((await Backendless.Request['get']((function(url){ if( !url ) { throw new Error('
            Url must be specified.')} if( !url.startsWith('http://') && !url.startsWith('https://')
            ) { return 'https://' + url } return url})('https://api.thingspeak.com/channels
            /1822679/fields/field1/last.html?api_key=9OOUIBWMZLWCJGVG')).setEncoding('utf8').send()
            ) >= '95') {
65        whatToSay = 'Frecuencia cardiaca alta';
66      }
67
68    } else if (intent == 'temperaturestatus') {
69
70      if ((await Backendless.Request['get']((function(url){ if( !url ) { throw new Error('Url
            must be specified.')} if( !url.startsWith('http://') && !url.startsWith('https://')) {
            return 'https://' + url } return url})('https://api.thingspeak.com/channels/1822679/
            fields/field1/last.html?api_key=9OOUIBWMZLWCJGVG')).setEncoding('utf8').send()) <=
            '36') {
71        whatToSay = 'Temperatura baja';
72
73      } else if ((await Backendless.Request['get']((function(url){ if( !url ) { throw new Error('
            Url must be specified.')} if( !url.startsWith('http://') && !url.startsWith('https://')
            ) { return 'https://' + url } return url})('https://api.thingspeak.com/channels
            /1822679/fields/field1/last.html?api_key=9OOUIBWMZLWCJGVG')).setEncoding('utf8').send()
            ) >= '37') {
74        whatToSay = 'Paciente presenta fiebre';
75      }
76    }
77
78    return (await require('codeless-function-3d9d7da73c54e9461df1de038dbbb0e6')(whatToSay, true))
          //
```

B. Código JSON de la Skill de alexa

```
1  {
2      "interactionModel": {
3          "languageModel": {
4              "invocationName": "asistente param dico",
5              "intents": [
6                  {
7                      "name": "AMAZON.CancelIntent",
8                      "samples": []
9                  },
10                 {
11                     "name": "AMAZON.HelpIntent",
12                     "samples": []
13                 },
```

```
14                  {
15                      "name": "AMAZON.StopIntent",
16                      "samples": [
17                          "alexa para"
18                      ]
19                  },
20                  {
21                      "name": "AMAZON.NavigateHomeIntent",
22                      "samples": []
23                  },
24                  {
25                      "name": "frecuencia",
26                      "slots": [],
27                      "samples": [
28                          "pulso",
29                          "c antos latidos por minuto",
30                          "pregunta cu l es la frecuencia cardiaca",
31                          "frecuencia",
32                          "dime la frecuencia",
33                          "cu l es la frecuencia",
34                          "pregunta cu l es la frecuencia"
35                      ]
36                  },
37                  {
38                      "name": "temperatura",
39                      "slots": [],
40                      "samples": [
41                          "temperatura",
42                          "dime la temperatura",
43                          "pregunta la temperatura"
44                      ]
45                  },
46                  {
47                      "name": "intro",
48                      "slots": [],
49                      "samples": [
50                          "lea las instrucciones",
51                          "lea estrucciones",
52                          "dame instrucciones",
53                          "dame indicaciones",
54                          "instrucciones",
55                          "muestrame las intrucciones"
56                      ]
57                  },
58                  {
59                      "name": "testvariable",
60                      "slots": [],
61                      "samples": [
62                          "variables",
63                          "prueba de variables",
64                          "pueba los sensores",
65                          "prueba el sensor",
66                          "testea el sensor",
67                          "test de sensor"
68                      ]
69                  },
70                  {
71                      "name": "modoambulancia",
72                      "slots": [],
73                      "samples": [
74                          "modo ambulancia"
75                      ]
76                  },
```

```
                {
                    "name": "pulsestatus",
                    "slots": [],
                    "samples": [
                        "estado de pulso",
                        "alarma de pulso",
                        "estado del pulso"
                    ]
                },
                {
                    "name": "temperaturestatus",
                    "slots": [],
                    "samples": [
                        "estado de la temperatura",
                        "estado de temperatura"
                    ]
                }
            ],
            "types": []
        }
    }
}
```