



UCASAL

UNIVERSIDAD CATÓLICA DE SALTA

"Desarrollo de un wearable para monitoreo de signos vitales de adultos mayores"

Leal Olivares, Agustina
INGENIERÍA EN INFORMÁTICA
FACULTAD DE INGENIERÍA
AÑO 2022

Título:

Ingeniería en Informática

Profesor Guía:

Nombre: Guaimas, Eduardo Mauricio

Firma: _____

Tribunal Evaluador:

Nombre: _____

Firma: _____

Nombre: _____

Firma: _____

Nombre: _____

Firma: _____

Alumna:

Nombre: Leal Olivares, Agustina

Firma: _____

Fecha de Exposición:

__/__/__

Dedicatoria

A mis abuelos y a mi papá que guían
y protegen desde el cielo.

Agradecimientos

Agradezco especialmente a todas y cada una de las personas que me acompañaron, apoyaron y me dieron fuerzas a lo largo de la carrera, a quienes me dieron palabras de aliento para lograr este objetivo.

A mis padres Maria Esther y Francisco por acompañarme y guiarme siempre en la vida y que con mucho esfuerzo me incentivaron siempre y me apoyaron incondicionalmente a lo largo de toda mi carrera.

A mi hermano por ser siempre brindarme palabras de aliento cuando, por darle valor a cada pasito que daba en la carrera y motivarme siempre a continuar a pesar de las adversidades.

A mis sobrinos por ser mi paz y mi alegría en los días más difíciles.

Y por último agradecer de todo corazón a mis profesores de la Universidad que me inspiraron a ser mejor cada día, por cada una de sus enseñanzas, la paciencia y dedicación a la hora de transmitirnos sus conocimientos.

ÍNDICE

ÍNDICE	4
ÍNDICE DE ILUSTRACIONES, GRÁFICOS Y TABLAS	8
1. INTRODUCCIÓN	12
2. ESTADO DE LA CUESTIÓN	14
2.1. Signos vitales SV en personas adultas mayores.....	15
2.1.1. COVID-19 y los Adultos Mayores.....	16
2.2. Situación actual respecto a la solución informática	17
2.2.1. Wearables	17
2.2.2. Wearables en el sector de la salud.....	18
2.2.3. Wearables y adultos mayores	19
2.3. Scrum	19
2.3.1. ¿Por qué utilizar SCRUM?.....	19
2.3.2. Principios SCRUM.....	21
2.3.3. Proceso Scrum.....	21
2.4. PMBOK.....	22
2.4.1. Componentes de la guía PMBOK	23
2.5. Antecedentes.....	24
3. DEFINICIÓN DEL PROBLEMA	26
3.1. Población a la cual va dirigida el proyecto	26
3.2. Objetivos	27
3.2.1. Objetivo General.....	27
3.2.2. Objetivos Específicos	27
3.3. Alcances	28

3.4. Alternativas tecnológicas para solucionar el problema.....	29
3.4.1. Oxímetro de pulso	29
3.4.2. Celular con botón SOS	29
4. SOLUCIÓN PROPUESTA	30
4.1. Recolección de información sobre requerimientos y necesidades de adultos mayores en su hogar	30
4.1.1. Marco metodológico.....	30
4.2. Análisis de factibilidad del proyecto	36
4.2.1. Factibilidad técnica.....	36
4.2.2. Factibilidad operativa	36
4.2.3. Factibilidad económica.....	36
4.3. Análisis FODA.....	36
4.3.1. Estrategias por implementar	37
4.4. Marco de trabajo.....	38
4.4.1. Aplicación del marco de trabajo.....	40
4.5. Formulación del proyecto.....	46
4.5.1. Acta de Constitución del proyecto	47
4.5.2. Estructura de Desglose del Trabajo (EDT)	48
4.5.3. Gestión de los recursos del proyecto	50
4.5.4. Gestión de Recursos Humanos.....	55
4.5.5. Gestión de Tiempos del proyecto	57
4.5.5.1. Diagrama de Gantt.....	59
4.5.6. Gestión de Riesgos del proyecto	61
4.5.7. Gestión de las comunicaciones del proyecto.....	62
4.5.8. Gestión de los interesados	64

4.5.9. Costos del proyecto	65
4.6. Diseño de la solución	68
4.6.1. Diseño de la solución de software	68
4.7. Desarrollo de la solución.....	74
4.7.1. Desarrollo de la solución de hardware	74
4.7.2. Desarrollo de la solución de software	80
5. VERIFICACIÓN EXPERIMENTAL.....	93
5.1. Prueba del sensor MAX 30100, botón y notificaciones.....	94
6. CONCLUSIONES	98
6.1. Futuras líneas de investigación	99
7. BIBLIOGRAFIA	100
8. ANEXOS	104
8.1. Encuesta	104
8.2. Gestión de proyectos	105
8.2.1. ¿Qué es un proyecto?	105
8.2.2. ¿Qué es la gestión de proyectos?.....	106
8.3. Desarrollo ágil de proyectos.....	106
12 principios del agilismo	107
8.4. Matriz RACI o RASI.....	107
8.4.1. El significado de las siglas RACI.....	108
8.4.2. Reglas a tener en cuenta en las matrices RACI.....	108
8.4.3. Variantes de la matriz RACI	108
8.5. ¿Por qué Flutter?	109
8.5.1. ¿Qué es Flutter?.....	109

8.5.2. ¿Qué es Ionic?	109
8.5.3. Flutter vs Ionic.....	109
8.6. Firebase cómo BaaS	112
7.6.1. Beneficios del backend como servicio	113
8.7. Código de sensor y botón con Firebase en Arduino IDE	113
8.8. Configuración para crear aplicación Flutter en Android Studio	118
8.9. Creación proyecto Firebase y configuración para agregarlo a la aplicación Flutter. 119	
1. Crear un proyecto Firebase	119
2. Registrar app con Firebase para iOS	119
3. Registrar app con Firebase para Android.....	119
4. Agregar archivo de configuración Firebase a la aplicación para iOS	119
5. Agregar archivo de configuración Firebase a la aplicación para iOS	120
7.10. Buenas prácticas para manejo de repositorios Github.....	121
1. Probar los cambios antes de hacer commit	121
2. Usar diferentes ramas.....	121
3. Realizar commits frecuentemente	121
4. Escribir mensajes de confirmación útiles.....	121

ÍNDICE DE ILUSTRACIONES, GRÁFICOS Y TABLAS

Ilustración 1: Oxímetro de pulso.....	29
Ilustración 2: Celular con botón SOS.....	29
Ilustración 3: Wemos D1 Mini ESP8266.....	52
Ilustración 4: Sensor MAX30102	52
Ilustración 5: Botón y tapa de botón	53
Ilustración 6: Diseño de la solución de software	68
Ilustración 7: Pantalla de inicio.....	69
Ilustración 8: Diseño de Dashboard	70
Ilustración 9: Diseño de historial diario	71
Ilustración 10: Diseño de pantalla de perfil	72
Ilustración 11: Diseño modal de telemedicina	73
Ilustración 12: Pulsioximetría	75
Ilustración 13: Esquemas de conexión.....	76
Ilustración 14: Esquema de conexión protoboard.....	76
Ilustración 15: Esquema de conexión Wemos D1mini y MAX30100.....	77
Ilustración 16: Prueba de conexión de componentes	77
Ilustración 17: Datos obtenidos del sensor Arduino IDE	78
Ilustración 18: Wemos 1-Button Shield.....	78
Ilustración 19: Esquema de conexión Wemos D1 y botón	78
Ilustración 20: Funcionamiento del botón en monitor serie de Arduino IDE.....	79
Ilustración 21: Código para subir datos del sensor a Firebase	79
Ilustración 22: Código con condiciones para subir datos del botón a Firebase	80
Ilustración 23: Comandos GIT.....	82
Ilustración 24: Colección Firebase de usuarios logueados	82
Ilustración 25: Resultados de pantalla de inicio de sesión.....	83
Ilustración 26: Datos de sensor RTBD.....	84
Ilustración 27: Obtención de datos RTBD	84
Ilustración 28: Resultado de pantalla de Dashboard	85

Ilustración 29: Datos en Cloud Firestore	86
Ilustración 30: Resultado de pantalla de historial diario	87
Ilustración 31: Resultados de la pantalla de perfil	88
Ilustración 32: Resultados	89
Ilustración 33: Funciones en Firebase	90
Ilustración 34: Código con botón SOS.....	91
Ilustración 35: Código de notificaciones.....	92
Ilustración 36: Prototipo.....	93
Ilustración 37: Pulsera wearable	93
Ilustración 38: Prueba de sensor en tiempo real	94
Ilustración 39: Datos del sensor RTBD	94
Ilustración 40: Datos en tiempo real en la aplicación	95
Ilustración 41: Datos del botón RTBD	96
Ilustración 42: Prueba de notificaciones	96
Ilustración 43: Prueba de e-mail	97
Ilustración 44: Configuración Inicial en Arduino IDE	114
Ilustración 45: Configuraciones IDE - Firebase.....	115
Ilustración 46: Código para obtener fecha y hora	116
Ilustración 47: Código para subir datos	116
Ilustración 48: Código para subir datos II.....	117
Ilustración 49: Loop de carga a Firebase	118
Ilustración 50: Archivo Gradle a nivel proyecto.....	120
Ilustración 51: Archivo Gradle a nivel app	121
Gráfico 1: Principios SCRUM	21
Gráfico 2: Indicadores demográficos seleccionados por censos nacionales. Total del país. Años 1869 – 2010.	26
Gráfico 3: Edad	32
Gráfico 4: Sexo	32
Gráfico 5: Convivencia	33

Gráfico 6: Obra Social	33
Gráfico 7: ¿Conoce cuáles deberían ser los valores adecuados de oxígeno en sangre? ...	34
Gráfico 8: ¿Tiene conocimiento sobre sus valores de oxígeno en su sangre?	34
Gráfico 9: ¿Realiza controles sanguíneos regularmente?	34
Gráfico 10: ¿Cuál es su opinión acerca de un dispositivo parecido a un reloj que controle su nivel de oxígeno en sangre y su ritmo cardíaco?	35
Gráfico 11: ¿Cuál es su opinión sobre un control remoto (a distancia) sobre ciertos indicadores personales de su salud física?	35
Gráfico 12: ¿Cómo considera tener un botón que le avise a un familiar o centro de salud mediante telemedicina si tiene un problema de salud?	35
Gráfico 13: Matriz FODA.....	37
Gráfico 14: Marco de trabajo Scrum.....	40
Gráfico 15: Estructura de Desglose del Trabajo (EDT).....	48
Gráfico 16: Resumen de tiempo del proyecto.....	57
Gráfico 17: Desagregación del tiempo del proyecto.....	58
Gráfico 18: Diagrama de Gantt	59
Tabla 1: Consolidado de las cifras normales de los signos vitales según la edad.....	16
Tabla 2: Procesos SCRUM	21
Tabla 3: Componentes PMBOK 6° Edición	23
Tabla 4: Backlog Item.....	40
Tabla 5: Historia de Usuario - Inicio de sesión del Usuario	41
Tabla 6: Historia de Usuario - Cierre de sesión del Usuario.....	41
Tabla 7: Historia de Usuario - Visualización de medida de ritmo cardíaco	42
Tabla 8: Historia de Usuario - Visualización de saturación de oxígeno	42
Tabla 9: Historia de Usuario - Bitácora de medidas por rango de tiempo	42
Tabla 10: Historia de Usuario - Notificaciones alerta de valores anómalos	43
Tabla 11: Historia de Usuario - Notificación alerta de botón SOS presionado	43
Tabla 12: Historia de Usuario - Envío de mail a centro de telemedicina	44
Tabla 13: Sprint 1.....	44

Tabla 14: Sprint 2.....	45
Tabla 15: Sprint 3.....	45
Tabla 16: Sprint 4.....	46
Tabla 17: Acta de Constitución del proyecto.....	47
Tabla 18: Recursos.....	56
Tabla 19: Matriz de responsabilidad.....	57
Tabla 20: Probabilidad e impacto	62
Tabla 21: Planificación de comunicaciones.....	63
Tabla 22: Registro de interesados	64
Tabla 23: Inversión en RRHH.....	66
Tabla 24: Inversión en Hardware.....	66
Tabla 25: Costos indirectos	67
Tabla 26: Costos totales	67

1. INTRODUCCIÓN

En el siglo XXI, en la República Argentina, el proceso de envejecimiento de la población ha avanzado sostenidamente, siendo uno de los países más envejecidos de la región. La Encuesta Nacional sobre Calidad de Vida de Adultos Mayores (ENCaViAM) realizada por el Instituto Nacional de Estadísticas y Censos, INDEC (2012) arrojó datos tales como que la población de Adultos Mayores (65 años y más) pasó de un 2,2% sobre la población total en el año 1869 (primer censo nacional) a un 10,2% sobre la población total en el año 2010. Este último porcentaje indica una alta proporción de personas mayores con relación a la población total.

Dicha encuesta provee también información acerca de las características de los hogares de los adultos mayores, entendiendo por hogar a la persona o grupo de personas (parientes o no) que comparten los gastos de alimentación y residen bajo el mismo techo. Se evaluaron 3 criterios con respecto a los hogares donde habitan los adultos mayores: hogares unipersonales de adultos mayores, hogares unigeneracionales de adultos mayores y hogares multigeneracionales con al menos un adulto mayor.

A partir de esa evaluación se analizó la distribución de los hogares con adultos mayores por criterio, obteniendo que 2 de cada 10 son unipersonales, 3 de cada 10 son unigeneracionales y 5 de cada 10 son multigeneracionales (Instituto Nacional de Estadísticas y Censos, INDEC, 2012).

Recopilando esta información se puede observar que el 50% de los adultos mayores viven solos o con personas de su misma generación, lo que reduce las capacidades de cuidado a los adultos con mayores necesidades. Este fenómeno provoca el aumento de las necesidades de atención de los adultos mayores por parte de sus familiares, que generalmente no disponen del tiempo necesario para satisfacer todo el requerimiento de atención (Organización Mundial de la Salud, 2021).

Asimismo, González, Sosa y Reboiras (2021) señalan que en las personas mayores, se puede visibilizar, con gran ímpetu, el deseo de permanecer en su domicilio, realizando allí sus actividades cotidianas de manera independiente y en caso de requerir asistencia, recibirla en su propio domicilio y solo en casos de extrema necesidad recurrir a centros especializados o a la asistencia de terceras personas.

Por lo mencionado anteriormente y la situación pandémica mundial, se considera menester brindar herramientas tecnológicas para colaborar con los adultos mayores y sus familiares,

cubriendo las necesidades de constante atención, haciendo de su hogar un lugar más seguro y fomentando su autonomía e independencia.

Estas herramientas considerarán ciertas necesidades de este grupo etario a tener en cuenta en su construcción:

- Necesidades de cuidados de salud, monitoreando y obteniendo constantemente información acerca del estado del adulto mayor.
- Necesidades de comunicación, especialmente en casos de emergencia, para informar a familiares y personas cercanas, Asimismo, para los centros de salud que realizan telemedicina.

Para atender dichas necesidades se pretende diseñar un prototipo móvil y *wearable* para brindar una solución a las necesidades descritas en el párrafo anterior.

Este trabajo propone un diseño e implementación de un dispositivo *wearable*, similar a lo que se denomina un smartwatch (reloj inteligente) además de una aplicación con la cual abordar las distintas vías de interacción, recabando datos, obteniendo información sobre la salud de la persona en el mismo momento que suceden los hechos.

Este dispositivo no necesitará de la asistencia de la persona que lo posee, todo será de forma remota, ya que los ancianos no son nativos digitales, por lo tanto, se tiene en consideración esta cualidad del grupo social.

Adicionalmente, se pretende brindar información a los centros de salud que realizan telemedicina (prestación de servicios médicos a distancia a través de tecnologías de la información y las comunicaciones) para monitorear pacientes por ejemplo con COVID-19 de bajo riesgo, monitoreándolos desde sus hogares evitando y reduciendo contagios y complicaciones.

La solución de este prototipo constará de software y de hardware. Este último para que sea utilizado por los adultos mayores, quienes interaccionarán con el mismo y se recopile datos acerca de su estado general de salud y sus actividades diarias. El software, que será una aplicación móvil, será utilizado por los familiares de los adultos mayores, quienes, a partir de esos datos recopilados, obtendrán información en el momento del estado de sus seres queridos. Dicha aplicación permitirá, además de la información en el momento, tener una bitácora del estado de salud de la persona que utiliza el hardware.

2. ESTADO DE LA CUESTIÓN

Cabe resaltar, antes de iniciar el capítulo, tres conceptos relevantes a la situación planteada en el proyecto, los mismos son:

- **Vejez:** la OMS considera que la edad cronológica no siempre es el mejor indicador de los cambios que acompañan el envejecimiento, en todas las precisiones se usa la edad de 60 o más para describir las edades más avanzadas. La vejez no es una enfermedad, es un estado de graduales cambios degenerativos, de lentos desgastes. Es otra etapa del ciclo vital.
- **Envejecimiento:** Según la OMS es el proceso fisiológico que se inicia en el momento de la concepción y se hace más evidente después de la madurez.
- **Adulto Mayor:** Denominación elegida entre otras como “viejos”, “ancianos”, “personas de la tercera edad” para referirse al colectivo de las personas que, habiendo alcanzado la adultez, transitan la tercera etapa de este ciclo vital (Maina, 2004).

Además, se puede destacar el informe realizado por la Lic. en Sociología Arlegui (2009) denominado “*Situación de los adultos mayores en la Argentina, Activos y capacidad de respuesta frente a la vejez en contextos vulnerables*” (pág.9), donde destaca que:

El envejecimiento demográfico en una transformación estructural de la composición etaria de la población, provocada por el acelerado incremento de las personas de edades avanzadas respecto del resto de los grupos de edad. Este fenómeno es producto de la convergencia de dos factores propios de la modernidad y el desarrollo económico: Una menor tasa de natalidad y una mayor longevidad (Arlegui, 2009, pág. 11).

Según Organización Panamericana de la Salud (2020) define a la vejez como:

El conjunto de modificaciones morfológicas y fisiológicas que aparecen como consecuencia de la acción del tiempo sobre los seres vivos, que supone una disminución de la capacidad de adaptación en cada uno de los órganos, aparatos y sistemas, así como de la capacidad de respuesta a los agentes lesivos que inciden en el individuo (pág. 4).

Por su parte, la Enciclopedia Británica (2016) la define como:

El cambio gradual e intrínseco en un organismo que conduce a un riesgo reciente de vulnerabilidad, pérdida de vigor, enfermedad y muerte. Tiene lugar en una célula, un órgano o en la totalidad del organismo durante el período vital completo como adulto de cualquier ser vivo (pág. 875).

Una definición biológica del envejecimiento es:

El cambio gradual en un organismo que conduce a un mayor riesgo de debilidad, enfermedad y muerte. Tiene lugar en una célula u órgano o el organismo total en el lapso de vida de un adulto o cualquier ser vivo. Hay una disminución en las funciones biológicas y en la capacidad para adaptarse al estrés metabólico. Cambios en órganos incluyen el reemplazo de las células funcionales cardiovasculares con tejido fibroso. Efectos generales del envejecimiento incluyen inmunidad reducida, pérdida de fuerza muscular, disminución de la memoria y otros aspectos de la cognición, y la pérdida del color en el cabello y la elasticidad en la piel (Lozano Poveda, 2012, pág. 19).

Según el informe “*Envejecimiento*”¹, publicado por la Organización Mundial de la Salud (OMS) en su sitio web, se puede destacar algunas cifras importantes a tener en cuenta.

- En 2019, el número de personas de 60 años o más ascendía a 1000 millones.
- Para 2030, el número de personas de 60 años o más ascenderá a 1400 millones.
- En 2050, el número de personas de 60 años o más ascenderá a 2100 millones.
- La pauta de envejecimiento de la población es mucho más rápida que en el pasado.
- Todos los países se enfrentan a retos importantes para garantizar que sus sistemas sanitarios y sociales estén preparados para afrontar ese cambio demográfico.

Donde nuevamente se destaca el creciente envejecimiento demográfico que se está viviendo no solo en Argentina, sino en el mundo, y la urgencia que genera esta situación por afrontarla de la mejor manera posible en pos de evitar el deterioro de la calidad de vida los ancianos y prolongar sus expectativas de vida.

En conclusión, es preciso señalar que el acelerado incremento de personas de edades avanzadas respecto del resto de los grupos de edad viene acompañado del incremento en la necesidad de cuidado de estas personas por otras de rangos etarios menores al mismo, que generalmente son sus familiares.

2.1. Signos vitales SV en personas adultas mayores

Los signos vitales (SV) son valores que permiten estimar la efectividad de la circulación, de la respiración y de las funciones neurológicas basales y su réplica a diferentes estímulos

¹ Envejecimiento: <https://www.who.int/es/health-topics/ageing> - 06/04/21

fisiológicos y patológicos. Por lo que es imperativo, que los médicos revisen acuciosamente estas medidas. Apoyados en las nuevas tecnologías, se detectan fácilmente las alteraciones en los SV que demandan del médico intervenciones propias y oportunas.

Los SV son la cuantificación de acciones fisiológicas, como la frecuencia y ritmo cardíaco (FC), la frecuencia respiratoria (FR), la temperatura corporal (TC), la presión arterial (PA o TA) y la oximetría (OXM), que indican que un individuo está vivo y la calidad del funcionamiento orgánico. Los SV normales cambian de un individuo a otro y en el mismo ser en diferentes momentos del día; pero también por las variables a referenciar más adelante. Cualquier alteración de los valores normales, orienta hacia un mal funcionamiento orgánico y por ende se debe sospechar de un estado mórbido (Villegas González, Villegas Arenas, & Villegas González, 2012).

Tabla 1: Consolidado de las cifras normales de los signos vitales según la edad

Edad	Temperatura	Respiración	Pulso	Tensión arterial
Recién nacido	36.6° C a 37.8° C	30 a 40/ min	120-160/min	70/50
Primer año	36.6° C a 37.8° C	26 a 30/ min	120-130/ min	90/50
Segundo año	36.6° C a 37.8° C	25/ min	100-120/min	De 2 a 10 años: Sistólica: # años x 2 + 80. Diastólica: mitad de la sistólica + 10
Tercer año	36.6° C a 37.8° C	25/ min	90-100/min	
4 a 8 años	36.5° C a 37° C	20 a 25/ min	86-90/min	
8 a 15 años	36.5° C a 37° C	18 a 20/ min	80-86/min	De 10 a 14 años: Sistólica: # de años + 100. Diastólica: mitad de sistólica + 10
Edad adulta	36.5° C	16 a 20/ min	60-80/min	120/ 80 +6- 10
Vejez	36.0 C ó menos	14 a 16/ min	60 ó menos	

Fuente: (Villegas González, Villegas Arenas, & Villegas González, 2012)

2.1.1. COVID-19 y los Adultos Mayores

Respecto del estado general de los adultos mayores en Latinoamérica y como repercutió el COVID en ellos, se puede destacar el informe de la Organización Panamericana de la Salud (2021) titulado “Las personas mayores de 60 años han sido las más afectadas por la COVID-19 en las Américas” y publicado el 30 de septiembre de 2020.

En este se informan algunos datos relevantes como: “Si bien todos están el riesgo de contraer la COVID-19, las personas mayores tienen mayor probabilidad de enfermar gravemente si se infectan, con los mayores de 80 años muriendo a una tasa cinco veces mayor que la media.” (pág.37).

El informe de la Organización de las Naciones Unidas (2020) el impacto de la COVID-19 en las personas mayores sugiere que esto puede ser debido a condiciones subyacentes que afectan

al 66% de las personas mayores de 70 años. Este también es el caso de las Américas, donde la mayoría de las muertes por la COVID-19 ocurren en personas de 70 años o más, seguidas de personas entre 60 y 69 años.

Si bien las residencias o centros de atención a largo plazo han sido las más afectadas, y representan entre el 40% y el 80% de las muertes por la COVID-19 en todo el mundo, en las Américas, donde es más probable que el cuidado de los adultos mayores se lleve a cabo en el hogar, el distanciamiento físico constituye un desafío particular.

Además, en el informe se concluye que:

Incluso antes de la pandemia de la COVID-19, más del 50% de las poblaciones mayores en algunos países de ingresos bajos y medianos carecían de acceso a algunos servicios de salud esenciales, un problema que la pandemia ha exacerbado. Pero no basta con garantizar que los adultos mayores tengan acceso a los servicios básicos de salud, señaló Enrique Vega, jefe de la Unidad de Curso de Vida Saludable de la OPS. Los servicios también deben adaptarse a las necesidades específicas de las personas mayores (pág. 108).

Basándonos en dicho informe se puede observar que la pandemia afectó principalmente a los adultos mayores y estos no tienen la posibilidad de acceder a servicios básicos de salud. Se podría mejorarse aliviando la carga de los centros de salud con la utilización de telemedicina.

2.2. Situación actual respecto a la solución informática

2.2.1. Wearables

Wearable hace referencia al conjunto de aparatos y dispositivos electrónicos que se incorporan en alguna parte de nuestro cuerpo, interactuando continuamente con el usuario y con otros dispositivos con la finalidad de realizar alguna función específica.

La palabra *wearable* tiene una raíz inglesa cuya traducción significa “llevable” o “vestible”. En la jerga tecnológica hace referencia a computadoras corporales. Todos estos productos incorporan un microprocesador que se encuentra interactuando continuamente y que acompaña al usuario a todas partes.

La tecnología *wearable* se encuentra presente en un amplio abanico de sectores que satisfacen nuestras necesidades y deseos con la finalidad de aumentar nuestra calidad de vida.

2.2.2. Woreables en el sector de la salud

En el área de prevención, dispositivos a bajo precio y análisis personalizados podrían dar solución a dos de las principales causas de muerte: las enfermedades cardiovasculares y la diabetes. La supervisión, junto con incentivos individualizados, ayudaría a cambiar progresivamente el comportamiento de los individuos, combatiendo desde el inicio las causas profundas de las enfermedades, en particular la obesidad y la falta de ejercicio (Gutiérrez Martínez & Estrada, 2018).

Las compañías de seguros y programas de bienestar corporativo están incentivando a la gente a utilizar dispositivos de seguimiento para reducir los costos de atención médica (Rodríguez Gómez, 2017).

Muertes adicionales se podrían evitar al alertar a médicos y equipos de respuesta de emergencia, y entonces de esta forma poder ayudar rápidamente a una persona que pudiera tener un ataque al corazón u otra crisis de salud (Gutiérrez Martínez & Estrada, 2018).

En 2020, el 30% de las muertes en el mundo desarrollado podría ser evitada mediante soluciones de salud de monitoreo. Las Naciones Unidas predicen que la vida se extendería a 79 años en el mundo desarrollado hacia el 2020. Con las soluciones de vigilancia de la salud, la esperanza media de vida podría aumentar a 79,5 años (Hernández Velasquez, 2020).

Los dispositivos de monitoreo remoto proporcionarían un acceso continuo de información modificando la relación paciente/profesional. Según un estudio realizado por Chong (2016) citado en De Arriba Pérez (2019) existen 5 P's que marcarán tendencia en la nueva etapa en el cuidado de nuestra salud.

- 1) Preventiva, relacionado con los *wearables* o gadgets tecnológicos de registro de información emitida por nuestro cuerpo (ej.: ritmo cardíaco), y cómo estos podrían evitar que la salud de los pacientes empeorase (especialmente los crónicos).
- 2) Proactiva, transformando el rol de los pacientes y capacitándoles para decidir qué hacer después de recibir información precisa sobre su estado de salud.
- 3) Penetrante, materializando la posibilidad de programar *wearables* para aplicar tratamientos de baja complejidad lejos de los hospitales, en determinado tipo de pacientes y situaciones concretas.

- 4) Predictiva, trazando la evolución de la salud de las personas a través de la monitorización sistemática de su estado de salud.
- 5) Personalizada, ajustado a las características y necesidades de los diferentes tipos de paciente (pág.59-60).

2.2.3. Wearables y adultos mayores

La consultora e-Marketer (2020) realizó un estudio, que permitió medir el impacto de productos como los smartwatches y similares en el mercado europeo, Estados Unidos y Canadá. En el año 2020, arrojó una importante tendencia de los adultos mayores a este tipo de tecnología. Esta “tecnología para vestir”, si bien sigue siendo más utilizada en jóvenes (con el 38% del consumo), para los adultos mayores pasó, en 4 años, de 6.5%, a 22%, según el informe de la consultora.

La misma consultora, proporcionó una tabla predictiva, en donde se permite ver que la tendencia de los adultos mayores crecerá en los próximos tres años casi un 5% más. De igual forma, el informe arrojó que en 2020 fueron aproximadamente 56.7 millones de personas las que utilizaron, al menos una vez al mes, los ‘wearables’.

2.3.Scrum

Scrum es uno de los métodos ágiles más populares. Es un framework adaptable, iterativo, rápido, flexible y eficaz, diseñado para ofrecer un valor considerable en forma rápida a lo largo del proyecto. Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo.

Una fortaleza clave de Scrum radica en el uso de equipos interfuncionales (crossfunctional), autoorganizados y empoderados que dividen su trabajo en ciclos de trabajo cortos y concentrados llamados Sprints.

2.3.1. ¿Por qué utilizar SCRUM?

Algunas de las ventajas principales del uso de Scrum en cualquier proyecto son:

1. Adaptabilidad: El control del proceso empírico y el desarrollo iterativo hacen que los proyectos sean adaptables y abiertos a la incorporación del cambio.
2. Transparencia: Todos los radiadores de información tales como un Scrumboard y el Sprint Burndown Chart se comparten, lo cual conduce a un ambiente de trabajo abierto 3.

Retroalimentación continua: La retroalimentación continua se proporciona a través de los procesos de Realizar Daily Standup y Demostrar y validar el sprint.

4. Mejora continua: Los entregables se mejoran progresivamente sprint por sprint a través del proceso de Refinar el Backlog Priorizado del Producto.

Entrega continúa de valor: Los procesos iterativos permiten la entrega continua de valor tan frecuentemente como el cliente lo requiere a través del proceso de Envío de entregables.

Ritmo sostenible: Los procesos Scrum están diseñados de tal manera que las personas involucradas pueden trabajar a un ritmo sostenible que, en teoría, puede continuar indefinidamente.

Entrega anticipada de alto valor: El proceso de Crear el Backlog Priorizado del Producto asegura que los requisitos de mayor valor del cliente sean los primeros en cumplirse.

Proceso de desarrollo eficiente: El Timeboxing y la reducción al mínimo del trabajo que no es esencial conducen a mayores niveles de eficiencia.

Motivación: Los procesos de Realizar Daily Standup y Retrospectiva del sprint conducen a mayores niveles de motivación entre los empleados.

Resolución de problemas de forma más rápida: La colaboración y co-ubicación de equipos interfuncionales conducen a la resolución de problemas con mayor rapidez.

Entregables efectivos: El proceso de Crear el Backlog Priorizado del Producto, y las revisiones periódicas después de la creación de entregables aseguran entregas eficientes al cliente.

Centrado en el cliente: El poner énfasis en el valor del negocio y tener un enfoque de colaboración con los stakeholders asegura un framework orientado al cliente.

Ambiente de alta confianza: Los procesos de Realizar Daily Standup y la Retrospectiva del Sprint promueven la transparencia y colaboración, dando lugar a un ambiente de trabajo de alta confianza que garantiza una baja fricción entre los empleados.

Responsabilidad colectiva: El proceso de Comprometer Historias de Usuarios permite que los miembros del equipo hagan suyo el proyecto y su trabajo.

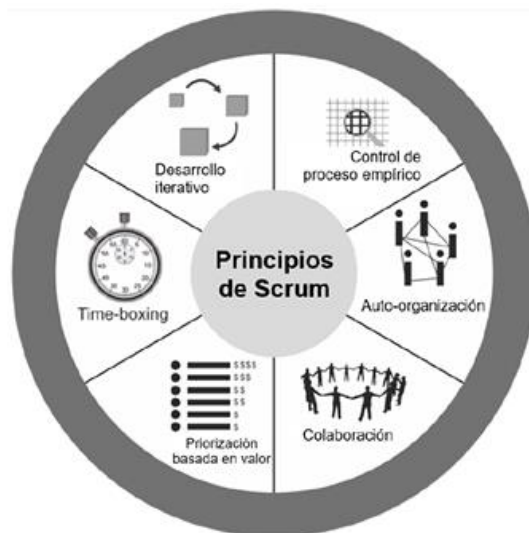
Alta velocidad: Leve a una mejor calidad. Un framework de colaboración permite a los equipos interfuncionales altamente cualificados alcanzar su potencial y una alta velocidad.

Ambiente innovador: Los procesos de Retrospectiva Sprint y Retrospectiva de Proyecto crean un ambiente de introspección, aprendizaje y capacidad de adaptación que conllevan a un ambiente de trabajo innovador y creativo (Sutherland, 2021, págs. 41-42).

2.3.2. Principios SCRUM

Los principios de Scrum son las pautas básicas para aplicar el framework de Scrum y deben implementarse en forma obligatoria en todos los proyectos Scrum.

Gráfico 1: Principios SCRUM



Fuente: (Sutherland, 2021)

2.3.3. Proceso Scrum

Los procesos de Scrum abordan las actividades específicas y el flujo de un proyecto de Scrum. En total hay diecinueve procesos fundamentales de Scrum que aplican a todos los proyectos (Scrum Study, 2018)

Tabla 2: Procesos SCRUM

FASE	PROCESOS FUNDAMENTALES DE SCRUM
INICIO	1. Crear la visión del proyecto 2. Identificar al Scrum Master 3. Formar Equipos Scrum y Stakeholder(s) 4. Desarrollar épica(s) 5. Crear el Backlog Priorizado del Producto 6. Realizar la planificación de lanzamiento
PLANIFICACIÓN Y ESTIMACIÓN	1. Crear historias de usuarios 2. Estimar historias de usuarios

	3. Comprometer historias de usuarios 4. Identificar tareas 5. Estimar tareas 6. Crear el Sprint Backlog
IMPLEMENTACIÓN	1. Crear entregables 2. Realizar Daily Standup 3. Refinar el Backlog Priorizado del Producto
REVISIÓN Y RETROSPECTIVA	1. Demostrar y valorar el Sprint 2. Retrospectiva del Sprint
LANZAMIENTO	1. Enviar entregables 2. Retrospectiva del proyecto

Fuente: (Scrum Study, 2018)

2.4.PMBOK

A mediados del siglo XX, se buscó reconocer la gestión de proyectos como una profesión, motivo por el cual un grupo de profesionales unieron sus conocimientos sobre los fundamentos para la dirección de proyectos. Posteriormente el PMI desarrolló y publicó la Guía de los Fundamentos para la Dirección de Proyectos, reconocido por recopilar las buenas prácticas describiendo técnicas y herramientas aplicables en la mayoría de los proyectos (Guía del PMBOK 6° Edición, 2017).

Para Assaf (2019) la guía del PMBOK 6° Edición, es un documento que reúne conocimientos de los profesionales involucrados en el Project Management describiendo conceptos, herramientas y técnicas para el uso en los diversos proyectos.

Es importante considerar que la Guía del PMBOK 6° Edición (2017) es diferente de una metodología, ya que ésta es un conjunto técnicas, procedimientos y reglas basadas en las buenas prácticas para ser empleadas por profesionales que se desempeñan en esa disciplina. Dicha guía debe ser considerada por las organizaciones para emplear metodologías englobando políticas, procedimientos, reglas, herramientas y técnicas.

La aplicación de esta guía permitirá a las organizaciones controlar de manera correcta sus proyectos, asegurando tiempo, costos y calidad que garantiza la satisfacción de los clientes logrando aumentar la productividad de la empresa.

Los estudios confirman que esta guía se puede aplicar en el ámbito público y privado donde se encontrará un marco referencial formal, interesante y posible de entender en cuanto a las actividades involucradas durante el ciclo de un proyecto, que puede ser de cualquier tipo como, por ejemplo: construcción, tecnológicos, electromecánicos, etc. El PMBOK, es reconocido mundialmente como guía para la gestión de proyectos (Assaf, 2019)

2.4.1. Componentes de la guía PMBOK

La Guía PMBOK 6° Edición (2017) identifica varios componentes claves que se interrelacionan unos con otros conformando un proyecto, al gestionarlos de manera correcta permitirá la conclusión exitosa del proyecto. A continuación, se explica estos componentes en la siguiente tabla:

Tabla 3: Componentes PMBOK 6° Edición

Componentes Clave de la Guía del PMBOK	Breve descripción
Ciclo de vida del proyecto	Fases de proyecto considerado de inicio a fin.
Fase del Proyecto	Conjunto de actividades en los cuales se obtienen uno o más entregables.
Punto de revisión de fase	Revisión en la culminación de una fase dónde se podrá tomar la decisión proseguir con siguiente, pudiéndose presentar modificaciones o dar por concluido un programa o proyecto.
Procesos de la dirección de proyectos	Sistema de actividades que producen resultado final, tomando en cuenta las entradas y salidas.
Grupo de procesos de la dirección de proyectos	Los grupos de procesos incluyen procesos de inicio, planificación, ejecución, monitoreo y control, y cierre. No son considerados fases del proyecto

Área de conocimiento de la dirección de proyectos	Son 10 áreas de conocimientos que describen individualmente los procesos, prácticas, datos iniciales, resultados, herramientas y técnicas.
--	--

Fuente: (Guía del PMBOK 6° Edición, 2017)

2.5. Antecedentes

Proyectos e investigaciones enfocadas al mismo ámbito de aplicación

1. Artículo “Desarrollo e incorporación de tecnologías avanzadas en la prestación de servicios sociales”

Se da a conocer cuál es el nivel de desarrollo e incorporación de las tecnologías avanzadas en la prestación de los servicios sociales centrándose en la población que, por edad o dependencia, requiere de apoyos. Allí plantea cuestiones como las diferentes concepciones sobre la tecnología y la tecnología avanzada, o sobre el porqué de los casos de éxito. Asimismo, plantea los principales retos y recomendaciones que la sociedad debe abordar si quiere que la tecnología sirva para proveer de mejor manera los servicios con los que hoy se cuenta y los que se están diseñando de cara al futuro.

2. Artículo “HELPER: A home assisted and cost-effective living system for people with disabilities and homebound elderly”

Propone un sistema de vida asistida reconfigurable, simple, eficiente y rentable, de tecnología punta, que satisfaga las necesidades de los pacientes en cama, las personas con discapacidad y las personas mayores. La característica distintiva que hace que esta implementación sea única es su bajo costo, su bajo consumo de energía y su control fácil de usar, por lo tanto, el usuario puede operar el sistema sin asistencia externa.

3. Artículo llamado "Sistemas domóticos para adultos mayores con movilidad reducida" de Ricardo Contreras y Mónica Flores publica para la Revista Investigatio UEES

Donde se muestra la evolución de los sistemas de automatización para el hogar; la aparición de los estándares Zigbee, KNX y el protocolo X10; recopila varios proyectos domóticos orientados a la asistencia del adulto mayor y explica la contribución de estos a su independencia. También se realiza una comparación de los estándares bajo diferentes

parámetros y se concluye que para implementar estos proyectos es mejor la utilización de sensores y actuadores inalámbricos por no ser intrusivos. Siendo Zigbee la mejor opción por su costo, ahorro de energía, alcance, adaptabilidad, interoperabilidad y escalabilidad. En base a los casos de éxito mostrados y necesidades de los usuarios se proponen requerimientos mínimos para hogares inteligentes que brinden mayor independencia al adulto mayor.

3. DEFINICIÓN DEL PROBLEMA

Según la antes mencionada Encuesta Nacional sobre Calidad de Vida de Adultos Mayores, en América Latina y el Caribe el envejecimiento demográfico se ha expresado con celeridad desde la segunda mitad del siglo XX. En Argentina, según los datos publicados por el Centro Latinoamericano y Caribeño de Demografía, para el año 2050 una de cada cinco personas tendrá 65 años y más.

A partir de los años '70, el incremento de la población de mayor edad continuó con su tendencia creciente, pero a un ritmo mucho menor. Simultáneamente, se produjo una reducción de la proporción de la población joven en un 30% de la población total.

Gráfico 2: Indicadores demográficos seleccionados por censos nacionales. Total del país. Años 1869 – 2010.

Censos nacionales	Adultos mayores (65 años y más)	Jóvenes (0 a 14 años)	Esperanza de vida al nacer	Tasa global de fecundidad	Porcentaje de extranjeros	
					Población total	Población adulta mayor
	%		Años	Hijos/mujer		%
1869	2,2	42,8	32,9	6,8	12,1	17,1
1895	2,5	40,3	40,0	7,0	25,4	27,1
1914	2,3	38,4	48,5	5,3	29,9	51,0
1947	3,9	30,9	61,1	3,2	15,3	56,6
1960	5,6	30,8	66,4	3,1	13,0	49,3
1970	7,0	29,3	65,6	3,1	9,5	39,6
1980	8,2	30,3	68,9	3,3	6,8	25,2
1991	8,9	30,6	71,9	2,9	5,0	15,9
2001	9,9	28,3	73,8	2,4	4,2	11,0
2010	10,2	25,5	75,2	2,2	4,5	9,2

Fuente: INDEC. Estructura demográfica y envejecimiento poblacional en Argentina, Serie Análisis Demográfico N°14 y Censos Nacionales de Población.

Fuente: INDEC 2010

En la actualidad, el país se encuentra en una etapa de envejecimiento avanzada. Según el censo 2010, un 10,2% de la población tiene 65 años y más; es decir, existe una alta proporción de personas mayores con relación a la población total.

3.1. Población a la cual va dirigida el proyecto

El pico máximo de la buena funcionalidad de los órganos internos se sitúa hacia los treinta años. A partir de esa edad se inicia un descenso gradual pero continuo. A pesar de esta pérdida, la mayoría de las funciones continúan siendo adecuadas durante el resto de la vida, porque la capacidad funcional de casi todos los órganos es superior a la que el cuerpo necesita; pero se observa los siguientes cambios:

- La oxigenación en sangre disminuye, ya que el corazón no actúa al mismo ritmo que una persona joven.

- La frecuencia cardíaca máxima disminuye, pero la frecuencia en reposo no sufre cambios.
- Disminuye el volumen máximo de sangre que pasa a través del corazón.
- Disminuye la capacidad pulmonar.
- Se observa un aumento de la cantidad de aire remanente en los pulmones después de respirar.

Los cambios que se van dando en la medida que envejecemos se pueden ver alterados por el estilo de vida que ha llevado la persona. Varios órganos pueden sufrir daños en un grado mucho mayor que el causado por el envejecimiento, como en el caso de las personas que llevan un estilo de vida sedentario, una dieta inadecuada, que fuman y abusan del alcohol y de las drogas.

Por lo descrito hasta ahora, es que la población son aquellas personas que tienen más de 60 años y se encuentran en una situación de salud con problemáticas que deben ser monitorizadas.

3.2. Objetivos

3.2.1. Objetivo General

Diseñar e implementar un prototipo y aplicación, con el fin de monitorear y registrar, algunas variables con relación al estado de salud físico, en una persona de la tercera edad, en el primer semestre del año 2022, en la ciudad de Salta Capital.

3.2.2. Objetivos Específicos

El proyecto de grado tendrá los siguientes objetivos:

- Comprender el funcionamiento de los distintos sensores y placas involucrados en el proyecto, evaluando cuáles son las mejores prácticas para hacer uso de estos, obteniendo datos y controlándolos a través de una aplicación móvil.
- Diseñar y desarrollar un prototipo de una aplicación móvil, que permita interactuar con las placas Arduino y los sensores conectados para monitorear el estado de salud de adultos mayores.
- Obtener un dispositivo wearable y móvil que sea económico, es decir que utilice los recursos de manera eficiente, que sea fiable ya que los datos que recopilará serán de vital importancia para los usuarios, y que pueda ser implementado para su uso en la vida cotidiana de los adultos mayores.

3.3. Alcances

El alcance permite representar la totalidad del trabajo necesitado para dar por terminado el trabajo. En detalle, este proyecto involucrará varios aspectos que llevarán a la conclusión de este:

- En una primera parte se llevará a cabo un estudio de factibilidad del proyecto, la definición del marco de trabajo para su desarrollo y la posterior formulación del mismo. Para el marco de trabajo se utilizará Scrum y para la formulación del proyecto se utilizará PMBOK 6ta Edición. En esta etapa se incluye también, la recolección de información acerca de las necesidades relacionadas al proyecto del público hacia el cual va dirigido el proyecto.
- Como segunda etapa se realizará el análisis y diseño de la aplicación, en la que se incluyen varias funcionalidades, entre ellas:
 - Monitorización del estado general de salud por medio de sensores de ritmo cardiaco y saturación de oxígeno,
 - Implementación de un botón SOS o botón de emergencia para que el usuario pueda usar en caso de que lo considere necesario y la persona que está a cargo del adulto reciba una alerta,
 - Notificaciones de alerta para valores anómalos de ritmo cardiaco y saturación de oxígeno,
 - Envío de mail a centros de telemedicina informando el promedio de saturación de oxígeno del paciente (diagnosticado con Covid-19 y atendido por un centro de telemedicina) en el día.
- La tercera y última etapa del proyecto será la construcción de un prototipo a escala que pueda ser probado en un usuario donde a través de los sensores y una aplicación móvil se obtengan datos del estado general de salud del usuario en cuestión.

En cuanto al alcance en lo que respecta específicamente a la salud de los mayores, es que el prototipo está diseñado, al menos en esta etapa del proyecto, en monitorear el ritmo cardiaco y el nivel de oxígeno en sangre.

Quedan exentos en su medición la presión arterial, el nivel de glucosa en sangre, la temperatura corporal del adulto mayor, calorías en su alimentación y morbilidad.

3.4. Alternativas tecnológicas para solucionar el problema

3.4.1. Oxímetro de pulso

Un oxímetro² de pulso es un dispositivo pequeño que parece una especie de horquilla o pinza para ropa, pero grande. Te lo puedes poner en el dedo cómodamente (la mayoría exige tener la uña hacia arriba) y en unos segundos se encienden unas luces con números que indican tu nivel de oxígeno en el cuerpo y tu ritmo cardíaco. La mayoría de la gente sana tendrá una lectura de oxígeno de entre el 95 y el 98 por ciento. Algunas personas con padecimientos preexistentes podrían tener una lectura normal más baja.

Ilustración 1: Oxímetro de pulso



Fuente: The New York Times

3.4.2. Celular con botón SOS

Teléfono fácil de usar con el teclado grande y acceso de emergencia con tecla "SOS".

Ilustración 2: Celular con botón SOS



Fuente: Mercado Libre

² ¿Para qué sirve un oxímetro?: <https://www.nytimes.com/es/2020/04/29/espanol/estilos-de-vida/oximetro-para-que-sirve.html> - 25/5/21

4. SOLUCIÓN PROPUESTA

El desarrollo del proyecto se realizó de una manera lógica y ordenada, determinada por los siguientes pasos:

- 1) Recolección de información sobre requerimientos y necesidades de adultos mayores en su hogar.
- 2) Análisis de factibilidad del proyecto.
- 3) Aplicación de marco de trabajo Scrum en el proyecto.
- 4) Formulación del proyecto utilizando la guía de buenas prácticas PMBOK 6.^a Edición.
- 5) Diseño de las pantallas del proyecto.
- 6) Desarrollo de la solución hardware y software.
- 7) Construcción e integración de las diferentes partes del prototipo, acorde a los diseños ya desarrollados.
- 8) Pruebas de funcionamiento del prototipo.

4.1. Recolección de información sobre requerimientos y necesidades de adultos mayores en su hogar

4.1.1. Marco metodológico

La metodología de campo permitió obtener información real del lugar de los hechos, la presente investigación fue enfocada a personas adultas mayores que requieran monitoreo constante de su estado de salud en su residencia.

4.1.1.1. Población y Muestra

Para el desarrollo del presente proyecto se trabajó con un adulto mayor (Señora de más de 70 años que vive sola), debido a que la población es muy pequeña, la muestra es la misma que la población.

4.1.1.2. Recolección de información

Para el presente proyecto se recopiló información mediante la revisión de textos, también se realizaron una observación a una persona de la tercera edad, que resultó de guía para establecer las necesidades y posibles soluciones a su diario vivir, siendo de gran utilidad para la implementación del proyecto.

4.1.1.3. Procesamiento y Análisis de Datos

Una vez que se ha obtenido la información necesaria para llevar a cabo el proyecto, los datos recopilados fueron revisados y validados, de forma ordenada y sistemática. Posteriormente, se tabuló la información, y se realizó un análisis de esta, para de esta manera proporcionar los resultados obtenidos y cubrir los objetivos del proyecto planteado.

Los instrumentos para recolectar datos elegidos fueron:

4.1.1.3.1. Observación

En este sentido, Sabino (1994) sostiene que por ser fundamental en todos los campos de la ciencia; la observación consiste en el uso sistemático de nuestros sentidos orientados a la captación de la realidad que queremos estudiar. Más adelante distingue entre observación cotidiana y observación científica y aclara que observar científicamente, es percibir activamente la realidad exterior con el propósito de obtener los datos que, previamente, han sido definidos como de interés para la investigación.

Para esto, la observación que se realiza cotidianamente, como parte de nuestra experiencia vital, no puede ser considerada como científica, pues no está orientada hacia objetos precisos de estudio, no es sistemática y carece de controles o de mecanismos que nos pongan al resguardo de los errores que podemos cometer cuando la realizamos.

La observación posibilita registrar tanto los hechos objetivos como los subjetivos; esto nos permite realizar el enriquecimiento de los datos y; de esa manera obtener un cuerpo de información más variado y completo de cómo y en qué medida facilita el monitoreo la implementación de un dispositivo *wearable* en una persona de la tercera edad.

4.1.1.3.2. Encuesta

El instrumento a través del cual se realiza esta indagación es el cuestionario que consiste en un proceso estructurado de recolección de información a través de la respuesta a una serie predeterminada de preguntas.

Por lo cual, la finalidad del cuestionario es obtener información, de manera sistemática y ordenada, respecto de lo que las personas son, hacen, opinan, piensan, sienten, esperan, desean, aprueban o desaprueban respecto del tema objeto de investigación.

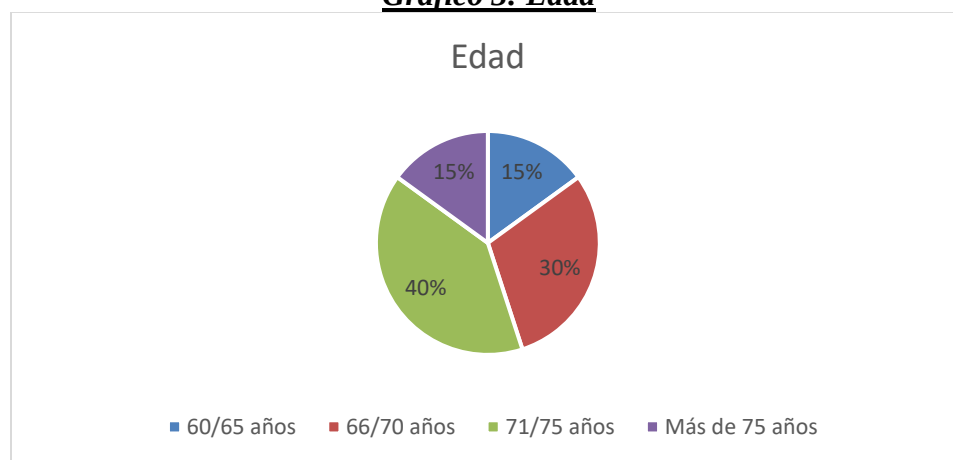
La encuesta es realizada para conocer los conocimientos de los ancianos sobre si estuviese de acuerdo o no con utilizar este prototipo.

La encuesta se desarrolla, en 3 partes:

- La primera para conocer los datos sociodemográficos de los encuestados.
- La segunda, mediante preguntas dicotómicas para indagar acerca del estado de salud de los encuestados.
- Finalmente, la tercera, se realiza una escala de Likert con el fin de indagar acerca de la predisposición de la muestra para usar el prototipo.

Datos sociodemográficos:

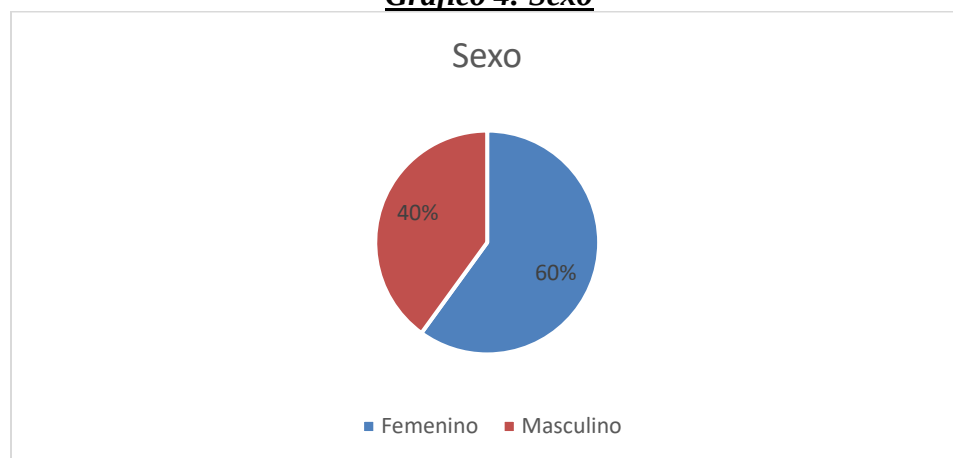
Gráfico 3: Edad



Fuente: Elaboración propia con base en la encuesta

Análisis del gráfico: El 70% de los encuestados se encuentra en un rango etario de entre 66 y 75 años, siendo la mayor cantidad de ancianos con jubilación.

Gráfico 4: Sexo



Fuente: Elaboración propia con base en la encuesta

Análisis del gráfico: El 60% de la muestra menciona ser del sexo femenino, mientras el 40% restante del sexo masculino.

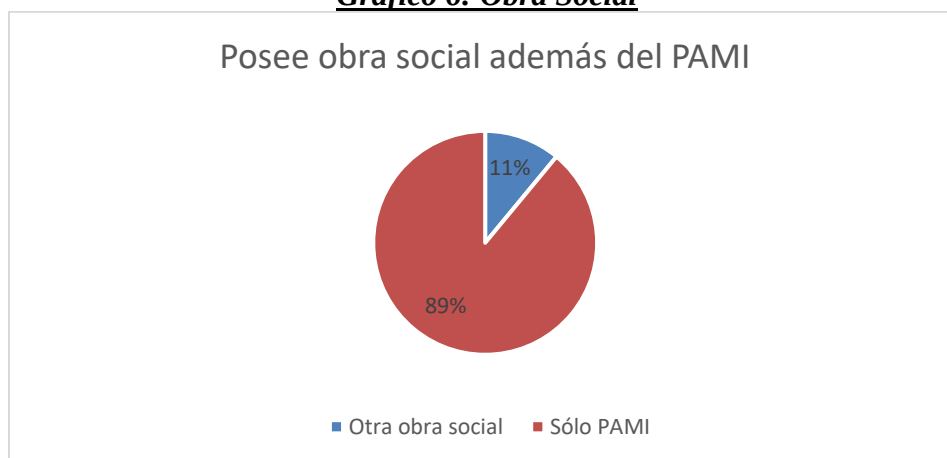
Gráfico 5: Convivencia



Fuente: Elaboración propia con base en la encuesta

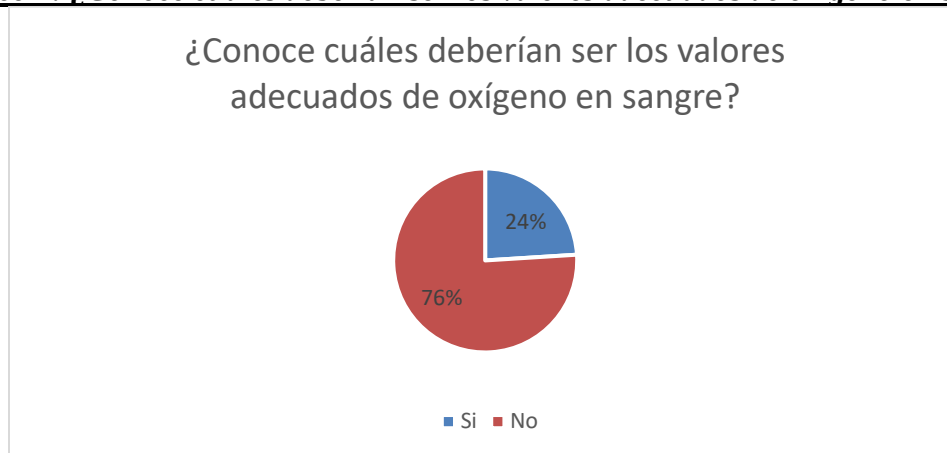
Análisis del gráfico: El 82% de los encuestados señala que en su hogar convive con otras personas, esposa, esposo, hijos, nietos, etc. Solo el 18% refiere que vive de manera solitaria, sin convivencia.

Gráfico 6: Obra Social

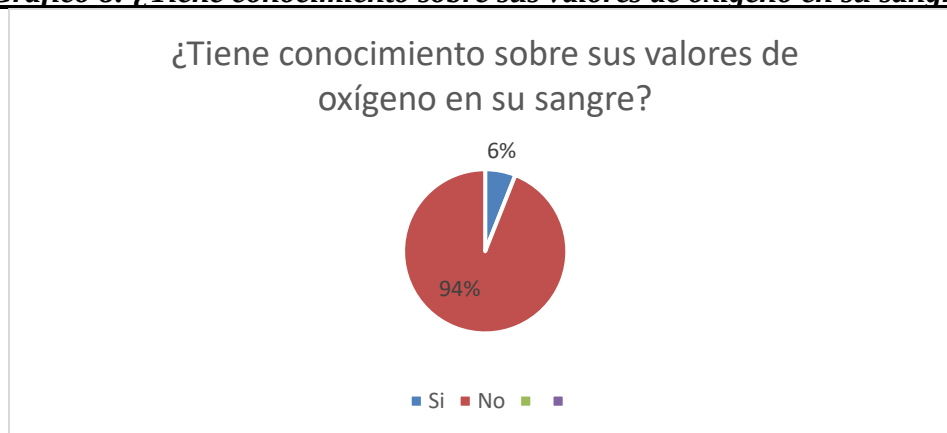


Fuente: Elaboración propia con base en la encuesta

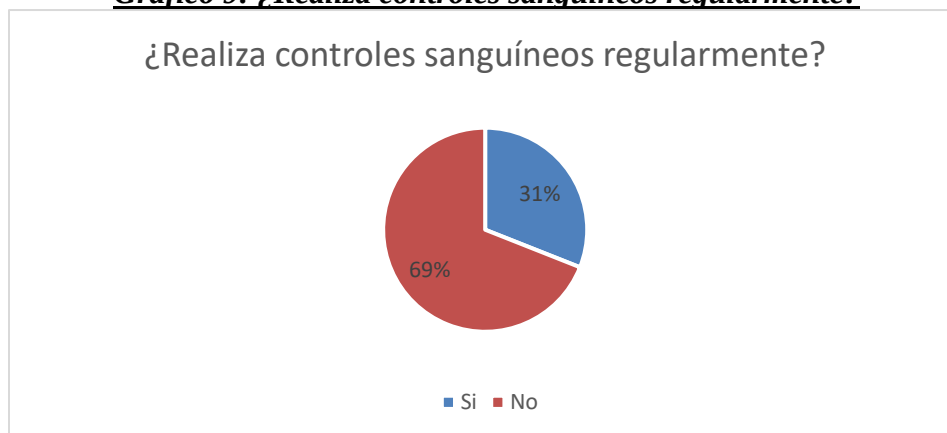
Análisis del gráfico: El 89% de la muestra menciona que solo tiene la obra social del PAMI, un 11% admite poseer otra obra social aparte de la estatal pública. Con el fin de indagar acerca de qué prestadoras podrían ser las que brinden el servicio del proyecto.

Gráfico 7: ¿Conoce cuáles deberían ser los valores adecuados de oxígeno en sangre?

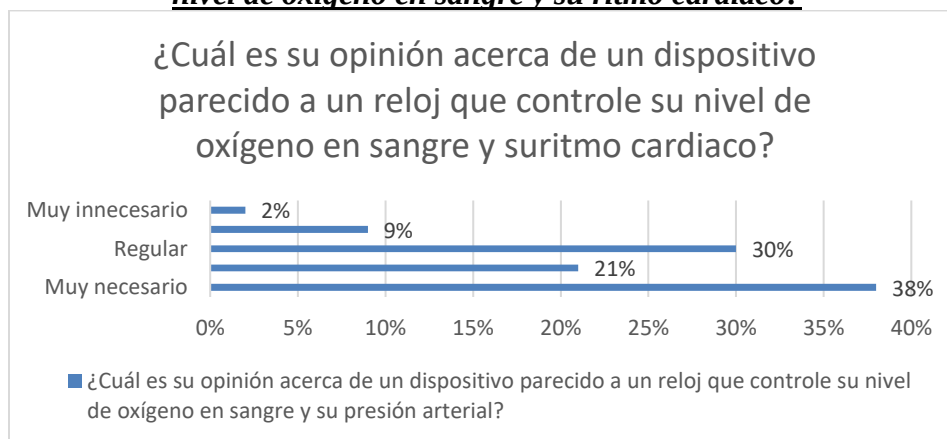
Fuente: Elaboración propia con base en la encuesta

Gráfico 8: ¿Tiene conocimiento sobre sus valores de oxígeno en su sangre?

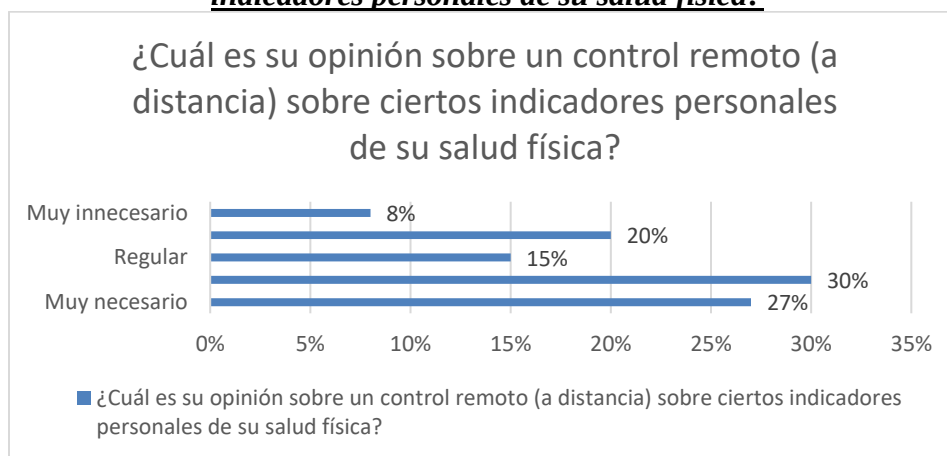
Fuente: Elaboración propia con base en la encuesta

Gráfico 9: ¿Realiza controles sanguíneos regularmente?

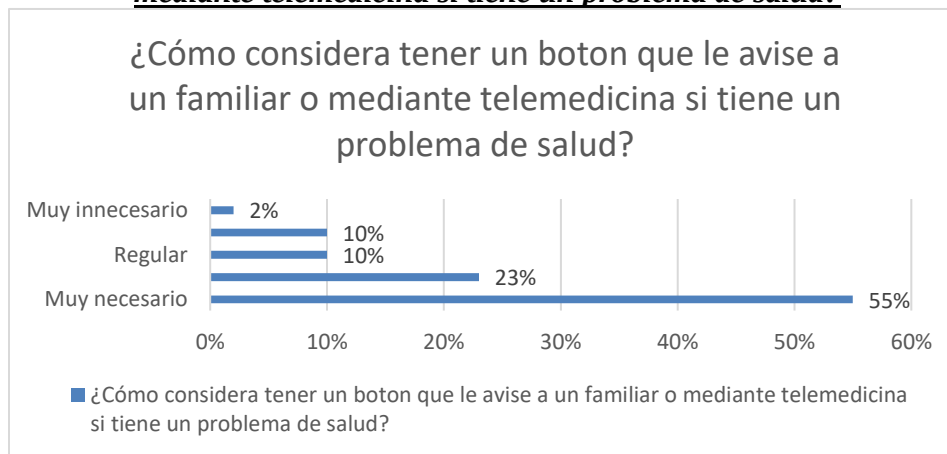
Fuente: Elaboración propia con base en la encuesta

Gráfico 10: ¿Cuál es su opinión acerca de un dispositivo parecido a un reloj que controle su nivel de oxígeno en sangre y su ritmo cardiaco?

Fuente: Elaboración propia con base en la encuesta

Gráfico 11: ¿Cuál es su opinión sobre un control remoto (a distancia) sobre ciertos indicadores personales de su salud física?

Fuente: Elaboración propia con base en la encuesta

Gráfico 12: ¿Cómo considera tener un botón que le avise a un familiar o centro de salud mediante telemedicina si tiene un problema de salud?

Fuente: Elaboración propia con base en la encuesta

4.2. Análisis de factibilidad del proyecto

Un estudio de factibilidad permitió determinar si se cuenta con los recursos suficientes para cumplir los objetivos de la investigación, por lo que se analizó la factibilidad con base en los siguientes parámetros:

- Técnica.
- Operativa.
- Económica.

4.2.1. Factibilidad técnica

La implementación del presente proyecto de investigación es factible técnicamente, debido a que se cuenta con las herramientas tecnológicas existentes en el mercado para el desarrollo de una aplicación móvil y dispositivo *wearable* dirigido a personas de la tercera edad, además se posee los conocimientos requeridos para diseñar y construir el prototipo electrónico.

4.2.2. Factibilidad operativa

En este punto el proyecto es factible, dado que el impacto de este dispositivo electrónico aplicado a los adultos mayores fue positivo porque el sistema se enfoca en monitorear el desarrollo de las tareas del diario vivir, de una manera más cómoda, mejorando la calidad de vida del usuario, por lo que este prototipo no presenta inconvenientes al momento de utilizarlo.

4.2.3. Factibilidad económica

El proyecto de investigación es factible económicamente debido al costo total para el desarrollo del prototipo, tanto el hardware como el software es accesible económicamente. El prototipo y los recursos adicionales que se requiera se encuentran cubiertos en su totalidad por el investigador.

El presente proyecto de grado pretende dar solución a la necesidad de cuidado de los adultos mayores, permitiéndoles manejarse de manera independiente en su hogar, pero con una constante monitorización de sus familiares o cuidadores.

4.3. Análisis FODA

En una primera instancia se realiza un análisis FODA para poder estudiar en mayor profundidad las características del proyecto y justificar el esfuerzo de este.

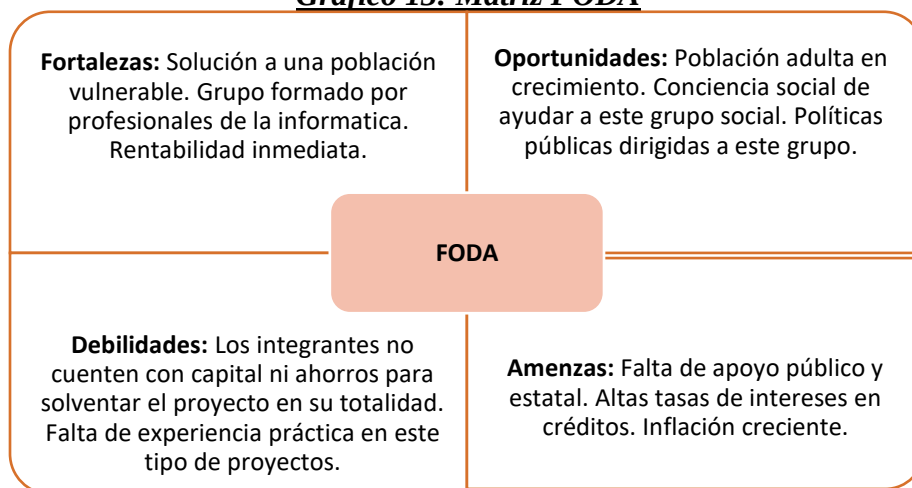
Con esto el lector va a poder identificar rápidamente los aspectos internos y externos de este trabajo, comprender mejor los riesgos asociados y conocer su importancia e impacto en nuestra sociedad.

El análisis FODA es una herramienta que permite evaluar las fortalezas, oportunidades, debilidades y amenazas con respecto a la situación actual de cualquier objeto de estudio (persona, empresa, proyecto, organización, etc.).

Tanto las fortalezas como las debilidades son internas de la organización, por lo que es posible actuar directamente sobre ellas. En cambio, las oportunidades y las amenazas son externas, y solo se puede tener injerencia sobre las ellas modificando los aspectos internos.

Este análisis aporta una visión global de la capacidad competitiva de nuestro proyecto tanto a nivel interno como externo.

Gráfico 13: Matriz FODA



Fuente: Elaboración propia, de acuerdo con el análisis previo del trabajo

4.3.1. Estrategias por implementar

Estrategias FO: Maximizan las fortalezas para aprovechar las oportunidades.

- Promoción de la aplicación en distintas redes sociales destacando sus ventajas para ganar el mercado.
- Atraer al público en conferencias o workshops, mostrando y permitiendo que realicen pruebas de su confortabilidad y usabilidad.

Estrategias FA: Maximizan las fortalezas para enfrentar las amenazas

- Hacer campañas del producto destacando su bajo costo e interfaz amigables para que los usuarios lo prefieran antes de las alternativas.
- Buscar alternativas de los componentes utilizados para evitar o disminuir la no disponibilidad del producto.

Estrategias DO: Mejoran las debilidades ante las oportunidades

- Notificar a los usuarios de la aplicación móvil que el adulto mayor monitorizado se desconectó de la red o del dispositivo, promoviendo de esta forma la permanencia y el control pasivo del adulto mayor en sus hogares, previniendo y/o evitando posibles contagios del virus COVID-19.
- Mejorar, en el corto plazo, las medidas de seguridad de la información para proporcionar mayor confiabilidad en el producto.
- Evaluar otros ámbitos de aplicación del dispositivo y sus posibilidades de mejora.

Estrategias DA: Mitigan las debilidades para resistir a las amenazas

- Búsqueda de nuevos componentes o proveedores de estos, aunque esto implique aumentar los costos del producto.
- Publicitar el producto mostrando sus beneficios haciendo hincapié en los beneficios y diferencias positivas respecto de las alternativas ya conocidas en el mercado.

4.4. Marco de trabajo

El marco de trabajo seleccionado para llevar a cabo el desarrollo del proyecto es Scrum³, debido a que es de fácil adaptación a cualquier tipo de proyectos.

Scrum es un modelo de desarrollo ágil caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada.

³ Libro de Scrum Manager: https://www.scrummanager.net/files/sm_proyecto.pdf - 19/9/21

En un sentido amplio, el marco de trabajo de Scrum se compone de una serie de reglas, que definen roles que integran los equipos, artefactos necesarios para los procesos, bloques de tiempo preestablecidos y ceremonias que deben respetarse.

Los equipos de Scrum definen tres roles bien diferenciados:

1. El Scrum Máster, responsable de asegurar los procesos;
2. El Dueño de Producto, responsable de maximizar el valor del producto;
3. El Equipo, responsable de realizar el trabajo.

Además, emplea 3 artefactos para llevar a cabo sus procesos:

1. Pila del producto: (product backlog) lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.
2. Pila del sprint: (sprint backlog) lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
3. Incremento: resultado de cada sprint.

La definición de bloques de tiempo iterativos (de 1 a 4 semanas), está destinada a crear continuidad y regularidad en las cuales, se basarán las seis ceremonias (reuniones) que aseguran el cumplimiento de objetivos:

1. Reunión de Planificación de la Entrega
2. Reunión de Planificación del Sprint
3. El Sprint
4. Reunión diaria
5. Reunión de revisión
6. Reunión de Retrospectiva

Y la pieza clave es el sprint.

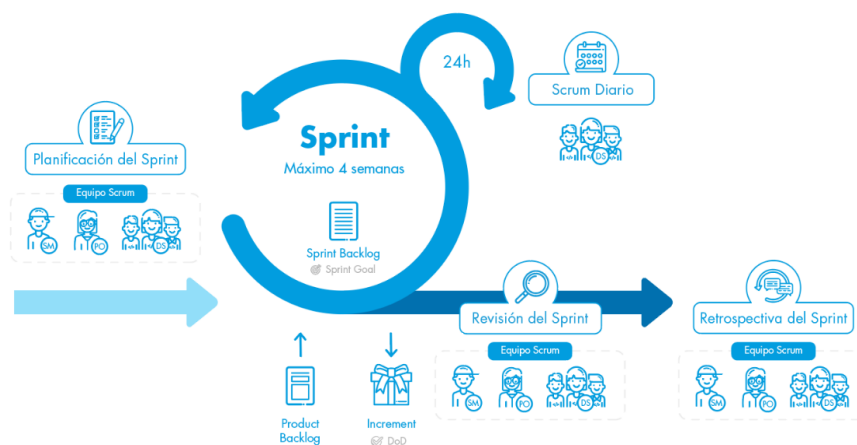
Se denomina sprint a cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente operativa (incremento).

Proceso:

Para el uso de la metodología Scrum en un proyecto se va a ejecutar en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, según convenga). Las iteraciones deben proporcionar un resultado completo, un incremento de producto final que sea capaz de ser entregado sin necesidad de emplear demasiado esfuerzo al cliente para cuando él lo

solicite. Para iniciar el proceso se debe partir de la lista de objetivos/requisitos debidamente priorizada para el producto, debe actuar como un plan para proyecto. El cliente es el encargado de listar y priorizar los objetivos, procurando que el valor de aporte respecto a su coste quede repartido en iteraciones y entregas. El cliente tiene la capacidad de maximizar la utilidad de lo que se encuentre en desarrollo y retornar la inversión mediante la replanificación de objetivos del producto.

Gráfico 14: Marco de trabajo Scrum



Fuente: Scrum.org

4.4.1. Aplicación del marco de trabajo

4.4.1.1. Pila de producto (Product Backlog)

Tabla 4: Backlog Item

Id	Backlog Ítem	Complejidad
1	Inicio de sesión del Usuario	3
2	Cierre de sesión del Usuario	3
3	Visualización de medida ritmo cardíaco	5
4	Visualización de medida de saturación de oxígeno	5
5	Bitácora de medidas por hora	13
6	Notificaciones alerta de valores anómalos	8
7	Notificación alerta de botón SOS presionado	8
8	Envío de mail a centro de telemedicina	8

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

4.4.1.2. Historias de Usuario

En entornos ágiles como Scrum, las historias de usuario son pequeñas descripciones de los requerimientos de un proyecto o cliente. Al redactar las historias de usuario se debe tener en cuenta describir el Rol, la funcionalidad y el resultado esperado en una frase corta. Debe venir acompañada de los criterios de aceptación, hasta un máximo de 4 por historia, redactado también en una frase que indique el contexto, el evento y el comportamiento esperado ante ese evento.

Siguiendo esa guía de redacción de historias de usuario, establecimos las siguientes para este proyecto:

Tabla 5: Historia de Usuario - Inicio de sesión del Usuario

1 Historia de Usuario: Inicio de sesión del Usuario
DESCRIPCIÓN: Como usuario necesito iniciar sesión en la aplicación para poder visualizar las funcionalidades del sistema.
Criterios de aceptación: - Los usuarios podrán acceder al sistema registrándose de 2 maneras: - Con usuario y contraseña. - Con una cuenta de Gmail. - La validación de los datos de sesión y su posterior relación con la información de la aplicación debe ser gestionada a través de Firebase.

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Tabla 6: Historia de Usuario - Cierre de sesión del Usuario

2 Historia de Usuario: Cierre de sesión del Usuario
DESCRIPCIÓN: Como usuario deseo poder cerrar sesión en el sistema para evitar que la sesión quede abierta y en el caso de ser necesario poder iniciar sesión en otro dispositivo.
Criterios de aceptación: El usuario debe haber iniciado sesión en el dispositivo.

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Tabla 7: Historia de Usuario - Visualización de medida de ritmo cardíaco

3 Historia de Usuario: Visualización de medida de ritmo cardíaco
DESCRIPCIÓN: Como familiar/persona a cargo del usuario que utiliza el hardware necesito poder visualizar la medida de ritmo cardíaco tomadas por el sensor.
Criterios de aceptación: • El usuario debe estar logueado. • Se debe visualizar el último valor tomado en un dashboard. • En el caso de que el valor este fuera de los valores normales, disparar una alerta (especificada en las notificaciones).

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Tabla 8: Historia de Usuario - Visualización de saturación de oxígeno

4 Historia de Usuario: Visualización de saturación de oxígeno
DESCRIPCIÓN: Como familiar/persona a cargo del usuario que utiliza el hardware necesito poder visualizar la medida de saturación de oxígeno tomadas por el sensor.
Criterios de aceptación: • El usuario debe estar logueado. • Se debe visualizar el último valor tomado en un dashboard. • En el caso de que el valor este fuera de los valores normales, disparar una alerta (especificada en las notificaciones).

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Tabla 9: Historia de Usuario - Bitácora de medidas por rango de tiempo

5 Historia de Usuario: Bitácora de medidas por rango de tiempo
DESCRIPCIÓN: Como familiar/persona a cargo del usuario que utiliza el hardware necesito disponer de una bitácora de los valores obtenidos por los sensores dentro de un cierto rango de tiempo.

Criterios de aceptación:

- El usuario debe estar logueado.
- Se debe visualizar el valor tomado de saturación de oxígeno y ritmo cardíaco, y el horario en que fueron tomados dichos valores.

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Tabla 10: Historia de Usuario - Notificaciones alerta de valores anómalos

6 Historia de Usuario: Notificaciones alerta de valores anómalos

DESCRIPCIÓN: Como familiar/persona a cargo del usuario que utiliza el hardware solicito me llegue una notificación en caso de que se dispare algún valor anormal en los sensores.

Criterios de aceptación:

- El usuario debe estar logueado.
- Se debe disparar una notificación al celular del usuario con el valor anormal tomado en tiempo real.

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Tabla 11: Historia de Usuario - Notificación alerta de botón SOS presionado

7 Historia de Usuario: Notificación alerta de botón SOS presionado

DESCRIPCIÓN: Como familiar/persona a cargo del usuario que utiliza el hardware solicito me llegue una notificación en caso de que se presione el botón SOS del dispositivo.

Criterios de aceptación:

- El usuario debe estar logueado.
- Se debe disparar una notificación al celular del usuario informando que se presionó el botón SOS.

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Tabla 12: Historia de Usuario - Envío de mail a centro de telemedicina

8 Historia de Usuario: Envío de mail a centro de telemedicina
DESCRIPCIÓN: Como familiar/persona a cargo del usuario que utiliza el hardware, quiero enviar un mail al centro de telemedicina a cargo de monitorear al usuario/paciente con el reporte de sus valores de saturación de oxígeno.
Criterios de aceptación: 1. El usuario debe estar logueado. 2. Se debe enviar un mail diario con el promedio de los valores de saturación de oxígeno tomados ese día.

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

4.4.1.3. Pila de Sprint (Sprint Backlog)

El proyecto está planificado para llevarse a cabo en 4 sprints con una duración de una semana cada uno. En los mismos se llevarán a cabo las 7 historias de usuario antes mencionadas.

En resumen, se realizará:

- Sprint 1: Inicio y cierre de sesión del usuario
- Sprint 2: Visualización de medidas
- Sprint 3: Bitácora de medidas
- Sprint 4: Notificaciones y mail

Sprint 1

Tabla 13: Sprint 1

Tarea	Responsable	Prioridad	Duración	Estado
Planificación del Sprint	Product Owner	1	1 día	Finalizado
Desarrollo interfaz de login	Programador	3	2 días	Finalizado
Desarrollo interfaz de logout	Programador	1	1 día	Finalizado
Desarrollo de autenticación con Firebase	Programador	3	2 días	Finalizado

Pruebas	Tester	1	1 día	Finalizado
		Total	7 días	

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Sprint 2

Tabla 14: Sprint 2

Tarea	Responsable	Prioridad	Duración	Estado
Planificación del Sprint	Product Owner	1	1 día	Finalizado
Desarrollo interfaz de visualización de medidas	Programador	3	1 día	Finalizado
Desarrollo de método de obtención de datos del sensor con Arduino IDE	Programador	3	1 día	Finalizado
Desarrollo de método para subida de datos a Firebase desde el sensor	Programador	3	1 día	Finalizado
Desarrollo de método de obtención de medidas desde Firebase	Programador	3	1 día	Finalizado
Desarrollo de método de asociación de medidas obtenidas con usuario logueado	Programador	3	1 día	Finalizado
Pruebas	Tester	1	1 día	Finalizado
		Total	7 días	

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Sprint 3

Tabla 15: Sprint 3

Tarea	Responsable	Prioridad	Duración	Estado
Planificación del Sprint	Product Owner	1	1 día	Finalizado
Desarrollo interfaz de visualización de la bitácora	Programador	3	2 días	Finalizado

Desarrollo de método de asociación de datos desde Firebase con tabla de bitácora	Programador	3	3 días	Finalizado
Pruebas	Tester	1	1 día	Finalizado
		Total	7 días	

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Sprint 4

Tabla 16: Sprint 4

Tarea	Responsable	Prioridad	Duración	Estado
Planificación del Sprint	Product Owner	1	1 día	Finalizado
Desarrollo de método que identifique valores anómalos	Programador	3	1 día	Finalizado
Desarrollo de método que envíe datos desde Arduino IDE a Firebase cuando se presiona el botón	Programador	3	1 día	Finalizado
Desarrollo de método en aplicación para que reciba datos cuando se presiona el botón	Programador	3	1 día	Finalizado
Desarrollo de interfaz de notificaciones	Programador	3	1 día	Finalizado
Desarrollo de interfaz de mail	Programador	3	1 día	Finalizado
Pruebas	Tester	1	1 día	Finalizado
		Total	7 días	

Fuente: Elaboración propia de acuerdo con el análisis del proyecto

4.5. Formulación del proyecto

La formulación del proyecto se basará en la guía de buenas prácticas (PMBOK) versión 6. En esta sección se desarrollará la adaptación de algunas áreas de conocimiento relevantes al proyecto.

4.5.1. Acta de Constitución del proyecto

Tal como lo establece la Guía PMBOK, el primer paso a dar en todo proyecto es el de realizar el Acta de Constitución del proyecto. Que es el punto de partida y se encuentra relacionado con los objetivos de este.

Aquí se establece la colaboración entre los diferentes miembros y las etapas de planificación.

Tabla 17: Acta de Constitución del proyecto

Acta de Constitución del proyecto	
Justificación y propósito del proyecto	Desarrollar un prototipo <i>wearable</i> con el cual se envíe datos de manera remota a un familiar del adulto mayor donde se monitorearán oxígeno en sangre y ritmo cardiaco. Con este sistema proyectamos ayudar a familiares de adultos mayores que necesitan constante monitorización de los mismos debido a que comparten el mismo hogar o bien permanecen mucho tiempo alejados de ellos.
Descripción del Producto/Servicio	Prototipo <i>wearable</i> conectado a una red wi-fi con envío de datos en tiempo real. La finalidad del prototipo es brindar una solución desde la tecnología al control de determinadas variables en lo que respecta a la salud de los adultos mayores. Comprobando en el momento su saturación de oxígeno en sangre y ritmo cardiaco.
Objetivos del Proyecto	Diseñar e implementar un prototipo y aplicación, con el fin de monitorear y registrar, algunas variables con relación al estado de salud físico, en una persona de la tercera edad.
Hitos Principales	1. Plan de gestión del proyecto y del producto. 2. MVP (Mínimo Producto Viable) 3. Plan de implementación. 4. Testeo del prototipo.
Director de Proyecto Asignado y sus responsabilidades	Agustina Leal Olivares: Responsabilidades: 1. Planificar el proyecto y sus objetivos. 2. Definir y delimitar la estrategia y los alcances del proyecto.

	3. Coordinar el equipo y los recursos disponibles.
	4. Desarrollar planes de comunicación.

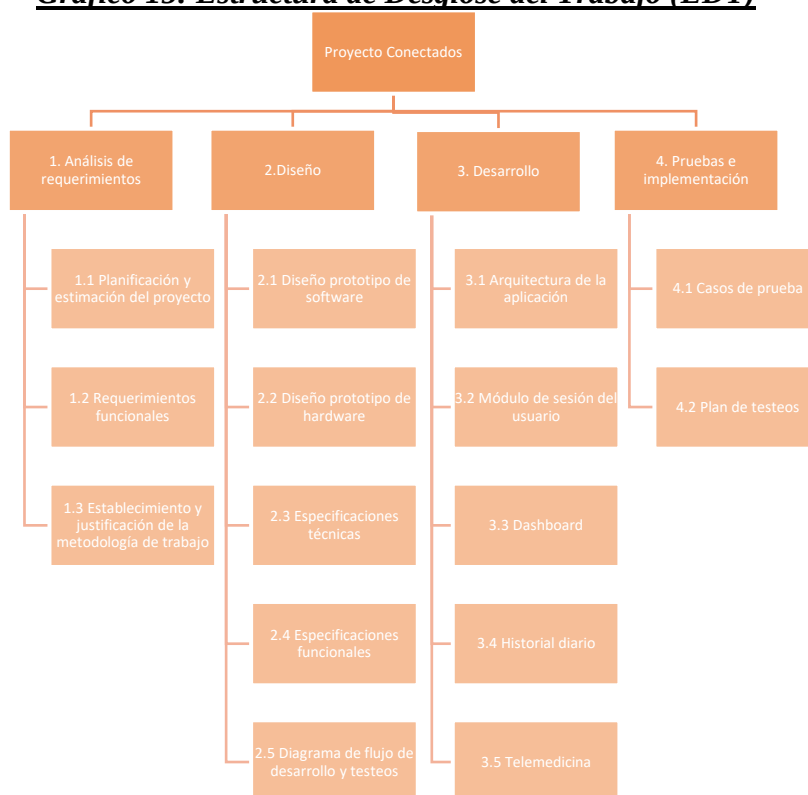
Fuente: Elaboración propia de acuerdo con el análisis del proyecto

4.5.2. Estructura de Desglose del Trabajo (EDT)

Este proceso consiste en dividir las tareas del proyecto en partes o componentes más pequeñas y fáciles de manejar.

La EDT nos sirve para organizar y definir el alcance total del proyecto. En este caso, en el gráfico siguiente se puede observar la estructura de desglose del trabajo del proyecto.

Gráfico 15: Estructura de Desglose del Trabajo (EDT)



Fuente: Elaboración propia de acuerdo con el análisis del proyecto

Diccionario de la EDT

1. Análisis de requerimientos:

1.1. Planificación y estimación del proyecto: Se tiene en cuenta el periodo que demanda el proyecto, como también cada una de las fases o eslabones de la cadena hasta cumplir con los objetivos

1.2. Requerimientos funcionales: Todas las exigencias que hacen al normal desarrollo de las tareas dentro del proyecto.

1.3. Establecimiento y justificación de la metodología de trabajo: Definir cuáles serán los pasos y los objetivos en cada fase del proyecto.

2. Diseño:

2.1. Diseño prototipo de software: Se diseña la maqueta del software a desarrollar.

2.2. Diseño prototipo de hardware: Se establecen los componentes físicos necesarios para el dispositivo.

2.3. Especificaciones técnicas: Son los documentos en los cuales se definen las normas, exigencias y procedimientos a ser empleados y aplicados en cualquier instancia del trabajo.

2.4. Especificaciones funcionales: Son los documentos que amplían en detalle los requerimientos de desarrollo, incluyendo una descripción completa de la lógica esperada, los datos y los criterios de prueba, con el fin de que los miembros del equipo dispongan de la información suficiente para poder diseñar una especificación técnica.

2.5. Diagrama de flujo de desarrollo y testeos: Aquí se describen los pasos a seguir en el trabajo, de manera gráfica de cada fase de este, como también sobre los testeos realizados en cada paso.

3. Desarrollo:

3.1. Arquitectura de la aplicación: Aquí se describen los patrones y las técnicas que se utilizan para diseñar y desarrollar aplicaciones. La arquitectura le proporciona un plan y las prácticas recomendadas que debe seguir para diseñar una aplicación bien estructurada.

3.2. Módulo de sesión del usuario: Permite a los usuarios iniciar sesión en la aplicación con los distintos métodos de autenticación.

3.3. Dashboard: Es la gráfica de los indicadores de medidas tomadas por los sensores.

3.4. Historial diario: Bitácora sobre los indicadores de las medidas tomadas al usuario del *wearable* en un día.

3.5. Telemedicina: Módulo de envío de indicadores tomados por la aplicación a centros de telemedicina.

4. Pruebas e implementación:

4.1. Casos de prueba: Son el conjunto de condiciones o variables bajo las cuales se determinará si una aplicación, un sistema de software o una característica o comportamiento de estos resulta o no aceptable.

4.2. Plan de testeos: Cronograma de las actuaciones de cada hecho sucedido o acontecimiento que se analiza.

4.5.3. Gestión de los recursos del proyecto

La Gestión de Recursos incluye los procesos para identificar, adquirir y gestionar los recursos necesarios para la finalización con éxito del proyecto.

El objetivo clave de esta área de conocimiento, la Gestión de los Recursos, es garantizar que los recursos adecuados estarán disponibles para el director del proyecto y el equipo del proyecto en el momento y lugar oportunos

4.5.3.1. Elección de los recursos

Para el presente proyecto se hace uso de una placa Arduino, modelo: ESP8266 WeMos D1 Mini. Entre las ventajas que se encuentran en esta placa, y que fue motivo de elección:

- Tienen un consumo de corriente muy bajo entre 15 μ A y 400 mA que se puede reducir aún más con el modo de sueño profundo activado a 0,5 μ A. Por lo tanto, el consumo de corriente es 70.000 más alto en el modo de suspensión profunda.
- El precio, que no supera los 3 dólares.
- La programación es similar al IDE Arduino.

Como todo producto dentro de un proyecto, también tiene sus desventajas, tales como:

- En este modelo hay problemas en cuanto es la falta de consenso al designar los pines.
- El modelo es importado.
- El precio varía de acuerdo a los movimientos del dólar.

La conexión seleccionada es a través de Wi Fi. Entre sus ventajas podemos mencionar:

- Más movilidad: Una de las ventajas más claras es la movilidad, el dispositivo conectado puede transportarse dentro del radio de acción del wi fi sin ningún problema.

- Menos cables: Al conectarse no se requieren cables, lo que es bueno para personas de la tercera edad.
- Más compatibilidad: También hay más compatibilidad, especialmente al conectarse a terceros dispositivos, como el caso de en un futuro conectar el dispositivo a otro artefacto inteligente.
- Sin límites de dispositivos: Un router Wi-Fi permite conectar muchos equipos de forma inalámbrica, aunque limitados por cable. Por tanto, se puede decir que no habría problemas en tener muchos dispositivos al mismo tiempo.

Entre las desventajas, se pudo encontrar:

- Problemas de estabilidad: Un dispositivo conectado por Wi-Fi puede sufrir más problemas con la calidad de la conexión en determinadas circunstancias.
- Seguridad: Sin dudas, la seguridad es algo que trae siempre problemas a las redes inalámbricas. Siendo necesario proteger nuestras redes mejor para evitar intrusos. Además, se deben revisar constantemente que no hay brechas que puedan ser aprovechadas.
- Depender del alcance del Wi-Fi: Se va a estar atado siempre al alcance de la red inalámbrica. Se pueden utilizar amplificadores, pero es un problema que está siempre presente.

4.5.3.2. Recursos de hardware seleccionados

Mini NodeMCU D1

El Wemos D1 mini ESP8266 es una tarjeta de desarrollo similar a Arduino, especialmente orientada al Internet de las cosas (IoT). Está basado en el SoC (System on Chip) ESP8266, un chip altamente integrado, diseñado para las necesidades de un mundo conectado. Integra un potente procesador con Arquitectura de 32 bits (más potente que el Arduino Due) y conectividad Wifi.

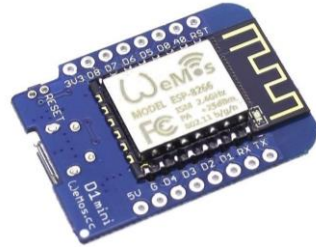
Para el desarrollo de aplicaciones se puede elegir entre los lenguajes Arduino y Lua. Al trabajar dentro del entorno Arduino podremos utilizar un lenguaje que ya conocemos y hacer uso de un IDE sencillo de utilizar, además de hacer uso de toda la información sobre proyectos y librerías disponibles en internet. La comunidad de usuarios de Arduino es muy activa y da soporte a plataformas como el ESP8266.

Wemos está diseñado especialmente para trabajar en protoboard o soldado sobre una placa. Posee un regulador de voltaje en placa que le permite alimentarse directamente del puerto USB.

Los pines de entradas/salidas trabajan a 3.3 V. El chip CH340G se encarga de la comunicación USB-Serial.

Esta placa está diseñada para permitir que los “shields” compatibles con WeMos mini se conecten a la placa de una manera similar a la plataforma de desarrollo Arduino.

Ilustración 3: Wemos D1 Mini ESP8266



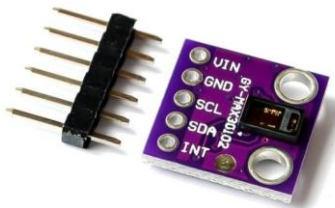
Fuente: Amazon

Sensor MAX30102

La pulsioximetría es un método no invasivo, que permite medir el porcentaje de saturación de oxígeno de la hemoglobina (SaO₂) en sangre de un paciente utilizando un circuito fotoeléctrico. Para esto se emplea un pulsioxímetro, que es un dispositivo que integra emisores de luz y el sensor que mide la cantidad de luz reflejada por el dedo del paciente. La luz detectada por el sensor varía de acuerdo a la concentración de oxígeno en la sangre, la sangre oxigenada absorbe mayor cantidad de luz infrarroja, mientras que la sangre poco oxigenada absorbe mayor luz roja.

El MAX30102 es un dispositivo que integra un pulsioxímetro y un monitor de frecuencia cardíaca. Posee dos LED: un led rojo (660 nm) y un led infrarrojo (920 nm), un fotodetector, óptica especializada, filtro de luz ambiental entre 50 y 60 Hz, y un conversor ADC delta sigma de 16 bits y de hasta 1000 muestras por segundo. Además, posee un sensor de temperatura interno para compensar los efectos de la temperatura en la medición.

Ilustración 4: Sensor MAX30102



Fuente: Amazon

Botón pulsador

Un interruptor o pulsador eléctrico es un dispositivo que permite desviar o interrumpir el curso de una corriente eléctrica. Para el prototipo se utilizará un pulsador convencional y una tapa de botón roja.

Ilustración 5: Botón y tapa de botón



Fuente: Amazon

4.5.3.3. Recursos de Software

Arduino IDE

Arduino IDE es un entorno de desarrollo y en él se realiza la programación de microcontroladores. Tiene como base el entorno de Processing al igual que un lenguaje de programación fundamentado en Wiring.

Flutter con Dart

Flutter es un *framework* de Dart⁴ para crear aplicaciones multiplataforma con un único código. A diferencia de otros *frameworks* multiplataforma como por ejemplo Ionic, el código de una aplicación de Flutter se compila a código nativo, por lo que el rendimiento alcanzado es superior a aplicaciones basadas en *web-views*. Además, a diferencia de React Native, Flutter no utiliza componentes nativos, sino que viene con sus propios componentes, llamados *widgets*, por lo que la misma aplicación se verá igual en cualquier dispositivo, independientemente de su sistema operativo o la versión. Gracias a ello, el desarrollador no tiene que preocuparse porque el diseño de su aplicación se vea mal en dispositivos antiguos.

⁴ ¿Qué es el lenguaje de programación Dart?: <https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart> - visitada el 25/5/21

Firebase

Firebase⁵ de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Firebase dispone de diferentes funcionalidades, que se pueden dividir básicamente en 3 grupos: Desarrollo (Develop), Crecimiento (Grow) y Monetización (Earn), a los que hay que sumar la Analítica (Analytics).

En este proyecto solo utilizaremos la función de desarrollo, en específico:

- **Realtime database**

Una de las herramientas más destacadas y esenciales de Firebase son las bases de datos en tiempo real. Estas se alojan en la nube, son No SQL y almacenan los datos como JSON. Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados, aunque el usuario no realice ninguna acción.

Firebase envía automáticamente eventos a las aplicaciones cuando los datos cambian, almacenando los datos nuevos en el disco. Aunque no hubiera conexión por parte de un usuario, sus datos estarían disponibles para el resto y los cambios realizados se sincronizarían una vez restablecida la conexión.

- **Autenticación de usuarios**

La identificación de los usuarios de una app es necesaria en la mayoría de los casos si estos quieren acceder a todas sus características.

Firebase ofrece un sistema de autenticación que permite tanto el registro propiamente dicho (mediante email y contraseña) como el acceso utilizando perfiles de otras plataformas externas (por ejemplo, de Facebook, Google o Twitter), una alternativa muy cómoda para usuarios reacios a completar el proceso.

Así, este tipo de tareas se ven simplificadas, considerando también que desde aquí se gestionan los accesos y se consigue una mayor seguridad y protección de los datos. Se debe mencionar que Firebase puede guardar en la nube los datos de inicio de sesión con total seguridad, evitando que una persona tenga que identificarse cada vez que abra la aplicación.

⁵ ¿Qué es Firebase? Funcionalidades – Ventajas – Conclusiones: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/> - visitada el 25/5/21

- **Almacenamiento en la nube**

Firebase cuenta con un sistema de almacenamiento, donde los desarrolladores pueden guardar los ficheros de sus aplicaciones (y vinculándolos con referencias a un árbol de ficheros para mejorar el rendimiento de la app) y sincronizarlos. Al igual que la mayoría de las herramientas de Firebase, es personalizable mediante determinadas reglas.

Este almacenamiento es de gran ayuda para tratar archivos de los usuarios (por ejemplo, fotografías que hayan subido), que se pueden servir de forma más rápida y fácil. También hace la descarga de referencias a ficheros más segura.

- **Cloud Messaging**

Su utilidad es el envío de notificaciones y mensajes a diversos usuarios en tiempo real y a través de varias plataformas.

4.5.4. Gestión de Recursos Humanos

La gestión de recursos humanos⁶ incluye los procesos necesarios para hacer más efectivo el uso de las personas involucradas en el proyecto. Esta gestión incluye a todos los implicados en el proyecto (clientes, sponsors, etc.).

En el presente trabajo, se estiman recursos humanos teniendo en cuenta el desarrollo de una solución final integral comercial de este proyecto.

Para dicha estimación, los roles involucrados serán analistas de sistema, diseñadores, desarrolladores y testers.

Sus responsabilidades son:

- **Analista:** Tiene a su cargo el relevamiento y análisis de las necesidades del negocio. Documentan procesos y requerimientos, considerando los objetivos y lineamientos del proyecto.
- **Diseñador:** Está involucrado en el proceso de diseño en cada una de sus etapas: investigación, definición, prototipado, desarrollo y medición. Tiene la intención de comprender un problema, proponer una solución y ayudar a llevarla a cabo.

⁶ Gestión de recursos en proyectos informáticos: <https://pmquality.files.wordpress.com/2009/08/gestion-de-rr-hh-en-proyectos-informaticos.pdf> - 28/5/21

- **Desarrollador:** Es responsable del diseño técnico, desarrollo y confección de test unitarios de las soluciones que implementamos. Investigación y aprendizaje de nuevos lenguajes, frameworks y tecnologías.
- **Tester:** Posee como principal responsabilidad identificar posibles fallos de implementación, calidad o usabilidad de un producto o servicio. Siendo las principales tareas, definir la estrategia, planificar, diseñar y ejecutar las pruebas, teniendo en cuenta el negocio de la aplicación, para asegurar la máxima calidad posible del producto.
- **Técnico Electrónico:** responsable de la construcción del *wearable* y desarrollo del código para tomar datos de las placas y sensores. También es responsable de la investigación, análisis y elección de alternativas hardware.

En la siguiente tabla se pueden visualizar los recursos humanos necesarios.

Tabla 18: Recursos

Recurso	Cantidad
Analista	1
Diseñador	1
Desarrollador	2
Técnico Electrónico	1
Tester	1

Elaboración propia de acuerdo con el análisis del proyecto

4.5.4.1. Matriz de Responsabilidad

Basada en Matriz RASI (Más información sobre esta matriz en el anexo 8.4)

Siglas:

- **“R”** – Responsable. Es el rol encargado de realizar la tarea.
- **“A”** – Aprobador. Es el rol que aprueba el trabajo realizado por el responsable.
- **“S”** – Soporte. Son aquellas personas que son consultadas sobre la cuestión, personas a las cuales se les pregunta su opinión sobre algún aspecto de la tarea, ya bien sea porque deben tenerse en cuenta o porque son expertos en la materia.
- **“I”** – Informado. Son aquellas personas a las que hay que mantener informadas sobre la evolución de la tarea.

Tabla 19: Matriz de responsabilidad

Rol	Análisis	Diseño	Desarrollo	Prueba	Implementación
Analista	R	A	A	A	A
Diseñador	I	R	I	S	I
Desarrollador	S	S	R	S	R
Técnico Electrónico	S	S	R	S	R
Tester	I	I	I	R	I

Elaboración propia de acuerdo con el análisis del proyecto

4.5.5. Gestión de Tiempos del proyecto

El proceso de gestión del tiempo contiene los procesos de Dirección de Proyectos necesarios para estudiar y establecer que tiempo es necesario para garantizar que el Proyecto posea un plazo viable para ejecutar la entrega del resultado esperado del Proyecto.

Para el proyecto en cuestión se realizó un diagrama de Gantt para poder visualizar de manera ordenada las distintas etapas del mismo y sus correspondientes tiempos. Se espera que, según los plazos establecidos, el proyecto tenga una duración de aproximadamente 84 días.

En resumen, las tareas realizadas serán las siguientes:

Gráfico 16: Resumen de tiempo del proyecto

Task Name	Duration	Start	Finish	% Complete
✓ Proyecto Conectados	84 days	Tue 3/5/22	Fri 26/8/22	100%
✓ Documentación	84 days	Tue 3/5/22	Fri 26/8/22	100%
✓ ▸ Análisis de requerimientos	5 days	Tue 3/5/22	Mon 9/5/22	100%
✓ ▸ Diseño	15 days	Tue 3/5/22	Mon 23/5/22	100%
✓ ▸ Desarrollo del prototipo hardware	6 days	Tue 24/5/22	Tue 31/5/22	100%
✓ ▸ Desarrollo de aplicación Conectados	25 days	Wed 1/6/22	Tue 5/7/22	100%
✓ ▸ Pruebas de integración de HW y SW	14 days	Tue 5/7/22	Fri 22/7/22	100%

Elaboración propia de acuerdo con el análisis del proyecto

La desagregación de estas tareas la podremos ver en las siguientes imágenes:

Gráfico 17: Desagregación del tiempo del proyecto

	Task Name	Duration	Start	Finish	% Complete
✓	▸ Proyecto Conectados	84 days	Tue 3/5/22	Fri 26/8/22	100%
✓	Documentación	84 days	Tue 3/5/22	Fri 26/8/22	100%
✓	▸ Análisis de requerimientos	5 days	Tue 3/5/22	Mon 9/5/22	100%
✓	Definición de objetivos y alcances	1 day	Tue 3/5/22	Tue 3/5/22	100%
✓	Planificación del proyecto	1 day	Wed 4/5/22	Wed 4/5/22	100%
✓	Análisis de alternativas y definición de recursos necesarios para el proyecto	1 day	Thu 5/5/22	Thu 5/5/22	100%
✓	Análisis del proyecto aplicando metodología Scrum	2 days	Fri 6/5/22	Mon 9/5/22	100%
✓	▸ Diseño	15 days	Tue 3/5/22	Mon 23/5/22	100%
✓	Diseño de interfaz de la aplicación	5 days	Tue 10/5/22	Mon 16/5/22	100%
✓	Diseño del prototipo	5 days	Tue 17/5/22	Mon 23/5/22	100%
✓	▸ Desarrollo del prototipo hardware	6 days	Tue 24/5/22	Tue 31/5/22	100%
✓	Conexión del sensor MAX30102 con ESP8266	1 day	Tue 24/5/22	Tue 24/5/22	100%
✓	Codificación para leer valores del sensor	2 days	Wed 25/5/22	Thu 26/5/22	100%
✓	Conexión de botón con ESP8266	1 day	Fri 27/5/22	Fri 27/5/22	100%
✓	Codificación para leer valores del botón	2 days	Mon 30/5/22	Tue 31/5/22	100%
✓	Subida de datos obtenidos a Firebase	1 day	Tue 31/5/22	Tue 31/5/22	100%
✓	▸ Desarrollo de aplicación Conectados	25 days	Wed 1/6/22	Tue 5/7/22	100%
✓	▸ Sprint 1	6 days	Wed 1/6/22	Wed 8/6/22	100%
✓	Desarrollo de interfaz de Inicio de Sesión	2 days	Wed 1/6/22	Thu 2/6/22	100%
✓	Desarrollo de interfaz de Cierre de Sesión	1 day	Fri 3/6/22	Fri 3/6/22	100%
✓	Desarrollo de autenticación con Firebase	2 days	Mon 6/6/22	Tue 7/6/22	100%
✓	Pruebas	1 day	Wed 8/6/22	Wed 8/6/22	100%
✓	▸ Sprint 2	6 days	Wed 8/6/22	Wed 15/6/22	100%
✓	Desarrollo interfaz de visualización de medidas (Dashboard)	2 days	Wed 8/6/22	Thu 9/6/22	100%
✓	Desarrollo de método de obtención de datos del sensor desde Firebase	2 days	Fri 10/6/22	Mon 13/6/22	100%
✓	Desarrollo de método de asociación de medidas obtenidas con usuario logueado	1 day	Tue 14/6/22	Tue 14/6/22	100%
✓	Pruebas	1 day	Wed 15/6/22	Wed 15/6/22	100%
✓	▸ Sprint 3	6 days	Thu 16/6/22	Thu 23/6/22	100%
✓	Desarrollo de interfaz de bitácora (Historial Diario)	2 days	Thu 16/6/22	Fri 17/6/22	100%
✓	Desarrollo de método de asociación de datos desde Firebase	3 days	Thu 16/6/22	Mon 20/6/22	100%
✓	Pruebas	1 day	Thu 16/6/22	Thu 16/6/22	100%
✓	▸ Sprint 4	7 days	Fri 24/6/22	Mon 4/7/22	100%
✓	Desarrollo de método que identifique valores anómalos	1 day	Fri 24/6/22	Fri 24/6/22	100%
✓	Desarrollo de método de obtención de datos desde Firebase en tiempo real	2 days	Mon 27/6/22	Tue 28/6/22	100%
✓	Desarrollo de método en la aplicación que reciba datos cuando se presione el botón	1 day	Wed 29/6/22	Wed 29/6/22	100%
✓	Configuración en Firebase para permitir enviar y recibir notificaciones	1 day	Thu 30/6/22	Thu 30/6/22	100%
✓	Desarrollo de interfaz de mail	1 day	Fri 1/7/22	Fri 1/7/22	100%
✓	Pruebas	1 day	Mon 4/7/22	Mon 4/7/22	100%
✓	▸ Pruebas de integración de HW y SW	14 days	Tue 5/7/22	Fri 22/7/22	100%
✓	Distintas pruebas	14 days	Tue 5/7/22	Fri 22/7/22	100%

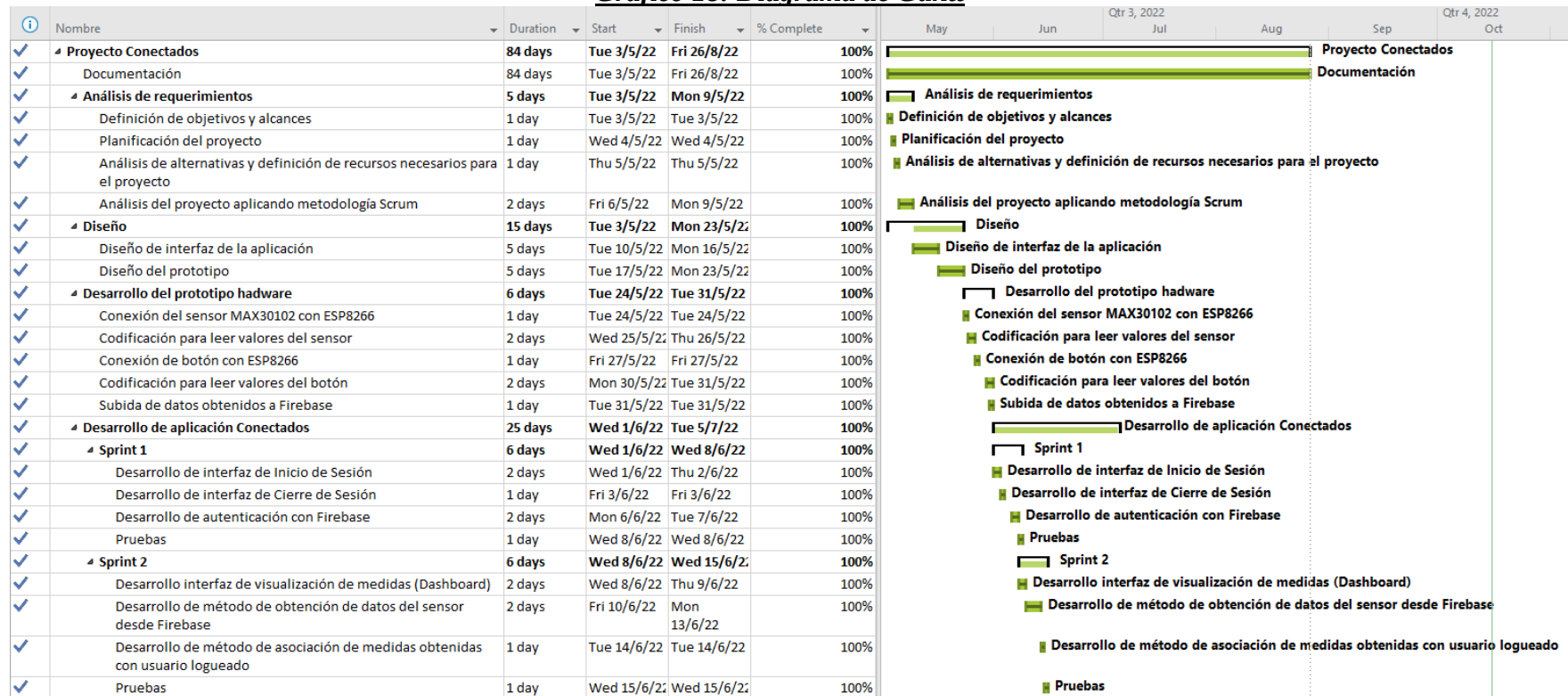
Elaboración propia de acuerdo con el análisis del proyecto

4.5.5.1. Diagrama de Gantt

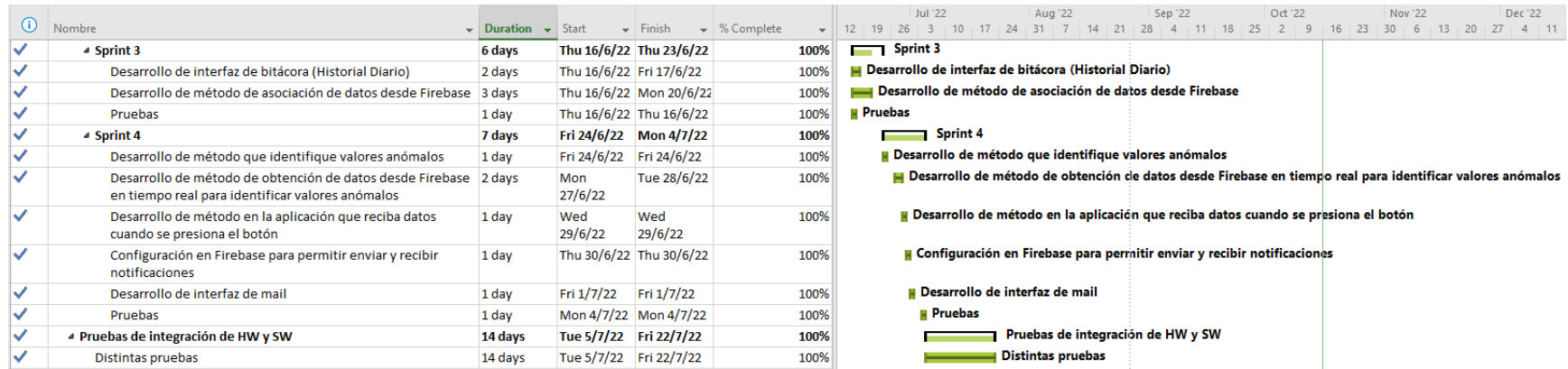
Un diagrama de Gantt es una herramienta útil para planificar proyectos. Al proporcionarte una vista general de las tareas programadas, todas las partes implicadas sabrán qué tareas tienen que completarse y en qué fecha.

El mismo se realizó con la herramienta Project de paquete Office, el resultado fue el siguiente:

Gráfico 18: Diagrama de Gantt



Elaboración propia de acuerdo con el análisis del proyecto



Elaboración propia de acuerdo con el análisis del proyecto

4.5.6. Gestión de Riesgos del proyecto

Según PMBOK⁶, el riesgo de un proyecto es un evento o condición incierta que, de producirse, tiene un efecto positivo o negativo en uno o más objetivos del proyecto, tales como el alcance, el cronograma, el costo y la calidad. Tienen su origen en la incertidumbre inherente a todos los proyectos.

La gestión de riesgos es una disciplina dentro de la Dirección de Proyectos que a través de la aplicación de procesos estandarizados busca aumentar la probabilidad y el impacto de los riesgos positivos y disminuir la probabilidad y el impacto de los riesgos negativos en un proyecto.

Se hace uso de la séptima versión PMBOK.

4.5.6.1. Identificación de riesgos

Los riesgos identificados en este proyecto son los siguientes:

- **Cambios en los requerimientos:** al ser un proyecto que aplica la metodología Scrum, los cambios en los requerimientos son muy frecuentes asociados a las necesidades que se vayan evaluando en el transcurso del proyecto. Es importante que estos cambios no afecten, o en la menor medida, a los costos y tiempos del proyecto.
- **Disponibilidad de componentes de hardware:** los componentes utilizados por el proyecto no se consiguen fácilmente en cualquier negocio de la ciudad, por ende, puede ocurrir que sea necesario adquirirlo vía internet a otra provincia o a otro país.
- **Componentes defectuosos o estropeados:** Al ser muy frágiles los componentes que se utilizan, es probable que estos se estropeen cuando se realizan las pruebas o bien tengas fallas de fábrica y sea necesario reemplazarlos, esto puede provocar retrasos en el proyecto.
- **Fallos en el software por actualización:** El framework elegido para desarrollar el proyecto es muy reciente y se encuentra en constante actualización, lo que provoca que el código cambie constantemente y puede generar retrasos, tener que migrar a nuevas versiones código que ya se encontraba funcionando.
- **Aumento en los costos del proyecto:** los costos de los componentes involucrados en el proyecto, así como también, los costos de los RRHH, están expresados en pesos, pero la suba constante del dólar en Argentina hace que estos valores fluctúen demasiado y se pueda ver afectado el proyecto a grandes aumentos en sus costos.

- **Personal no capacitado en la tecnología a utilizar:** al ser una tecnología relativamente nueva la que involucra este proyecto, un riesgo muy probable es el de que el personal no esté capacitado en dicha tecnología y se deba invertir tiempo y dinero en capacitación para los recursos.

4.5.6.2. Matriz de probabilidad e impacto

Vincula la probabilidad de ocurrencia de cada riesgo con su impacto en los objetivos del proyecto.

Tabla 20: Probabilidad e impacto

Riesgo	Ocurrencia	Impacto	Valoración	Prioridad
Cambios en los requerimientos	2	3	6	3
Disponibilidad de componentes de hardware	3	4	12	4
Componentes defectuosos o estropeados	3	4	12	4
Fallos en el software por actualización	3	3	9	3
Aumento en los costos del proyecto	5	5	25	5
Personal no capacitado en la tecnología a utilizar	2	4	8	4

Elaboración propia de acuerdo con el análisis del proyecto

Las columnas de la matriz de probabilidad de impacto representan:

- **Riesgo:** nombre del riesgo
- **Ocurrencia:** grado de ocurrencia del riesgo. Puede tener valores del 1 al 5, siendo 1 el grado más bajo de ocurrencia y 5 el más alto.
- **Impacto:** grado de afectación del riesgo al proyecto. Puede tener valores del 1 al 5, siendo 1 el grado más bajo y 5 el más alto.
- **Valoración:** es el resultado del producto entre el grado de ocurrencia y el grado de impacto del riesgo.
- **Prioridad:** varía del 1 al 5, donde el 1 representa menor prioridad y el 5 una mayor prioridad.

4.5.7. Gestión de las comunicaciones del proyecto

La gestión de las comunicaciones del proyecto es de suma importancia, ya que con esta etapa se pueden diseñar y planificar las siguientes funciones básicas para la gestión de la

comunicación: anticipar, organizar, dirigir, coordinar, controlar. Por lo tanto, gestionar la comunicación es:

- Definir los objetivos de la comunicación
- Identificar los medios para alcanzar estos objetivos
- Planificar acciones específicas para alcanzar los objetivos, teniendo en cuenta los recursos y las situaciones
- Organizar la aplicación de estos planes
- Coordinar la interacción de los componentes del proceso de comunicación, así como las actividades de los actores de primera línea
- Supervisar
- Ajustar el proceso en función de los resultados de la comunicación y la retroalimentación

El objetivo de la gestión de la comunicación es lograr una comunicación eficaz tanto dentro de la organización como entre esta y su entorno externo. Esta actividad consiste en el uso profesional de técnicas, métodos, procedimientos, tecnologías, que aseguran una interacción informativa y emocional eficaz entre los diferentes niveles de comunidades e individuos, que representan a la organización y su entorno.

Tabla 21: Planificación de comunicaciones

Interesados	Información por comunicar	Responsable	Formato	Periodicidad
Ministerio de Salud	Avances del proyecto	Analista funcional	Mail/presencial/ notas	Mensual
Centros de jubilados	Prototipos del proyecto	Analista funcional	Mail/presencial	Cuando se requiera
Fabricantes, vendedores de productos similares a SmartWatch	Ventajas del prototipo y resultados de testeos	Analista funcional	Mail/presencial	Cuando se requiera
Equipo de trabajo	Novedades del proyecto	Analista funcional	Mail / Chats del equipo	En cada iteración

Desarrolladores	Requerimientos funcionales	Analista funcional	Mail / Herramientas de planificación (TFS, Jira)	En cada iteración
Gobernación/Intendencia	Avances del proyecto	Analista funcional	Mail/presencial/ notas	Cuando se requiera

Elaboración propia de acuerdo con el análisis del proyecto

4.5.8. Gestión de los interesados

La Gestión de los Interesados del Proyecto incluye los procesos necesarios para identificar a las personas, grupos u organizaciones que pueden afectar o ser afectados por el proyecto, para analizar las expectativas de los interesados y su impacto en el proyecto, y para desarrollar estrategias de gestión adecuadas a fin de reconocer el logro de los objetivos o si se requieren cambios sobre la marcha

Tabla 22: Registro de interesados

Interesados	Grado de Involucramiento Actual	Grado de Involucramiento Deseado	Impacto del Interesado en el Proyecto
Ministerio de Salud	Apoya	Apoya	Alto
Centros de jubilados	Neutral	Apoya	Alto
Fabricantes, vendedores de productos similares a SmartWatch	Desconocedor	Apoya	Medio
Equipo de trabajo	Lidera	Lidera	Alto
Desarrolladores	Lidera	Lidera	Alto
Gobernación/Intendencia	Desconocedor	Apoya	Medio

Elaboración propia de acuerdo con el análisis del proyecto

4.5.9. Costos del proyecto

4.5.9.1. Costos

Los costos son todos aquellos gastos en los que incurre una empresa para realizar una tarea, un trabajo o un proyecto determinado. Las dos principales clases de costos que se conocen son los costos directos e indirectos.

El principal rasgo distintivo de los costos directos e indirectos de un proyecto tiene que ver con la relación que guardan con el objeto, que en este caso son los proyectos de los que dependen.

Los costos directos

Son los que guardan una relación estrecha con el producto, proyecto o servicio. Es decir, son costos que se asocian de forma directa a la elaboración y terminación de un producto.

Para este proyecto, los costos directos que analizaremos serán los siguientes:

- **Inversión en Recursos Humanos:**

Cómo se mencionó anteriormente, se requerirán un total de 6 recursos humanos para llevar a cabo este proyecto en su totalidad. Los costos relacionados con su actividad serán:

Con base en los datos proporcionado por la encuesta en OPENQUBE sobre empleos en el sector TI.

Opencube es una plataforma colaborativa en la que empleados y exempleados (usuarios) pueden calificar y escribir reseñas anónimas para brindar información de calidad sobre las empresas.

Allí los usuarios pueden detallar los “pros” y “contras” de trabajar en esa empresa. Además, se le pide que evalúen su conformidad (de 1 a 10, siendo 1 lo menos satisfactorio y 10 lo más satisfactorio) de determinadas situaciones en la empresa, a saber: Su ambiente, los beneficios, el equipamiento, la posibilidad de crecimiento, el salario, la calidad del código y las políticas de inclusión.

En esta plataforma se detallan los salarios o la tendencia sobre los costos en proyectos, brindando información de las empresas más importantes en la actualidad.

Estimaremos los costos suponiendo que los recursos serán semi senior. Los costos fueron extraídos de la plataforma en el mes de abril del año 2022 Los sueldos serían:

Calculando 84 días hábiles de trabajo, entonces:

Tabla 23: Inversión en RRHH

Rol	Sueldo x hora	Cantidad	Horas x día	Días	Total
Analista Funcional	\$ 687,5	1	4	22	\$ 60.500
Programador	\$ 750	2	4	44	\$ 264.000
Diseñador	\$ 625	1	2	12	\$ 15.000
Técnico Electrónico	\$ 687,5	1	4	18	\$ 49.500
Tester	\$ 437,5	1	2	12	\$ 10.500
				Total	\$ 399.500

Elaboración propia de acuerdo con el análisis del proyecto

- **Inversión en Hardware**

Tabla 24: Inversión en Hardware

Hardware	Cantidad	Precio en dólares*
Sensor Max30102	1	US\$ 2
Botón con pulsador	1	US\$ 1
Esp8266 Wemos D1 Mini	1	US\$ 6
Fuente para protoboard	1	US\$ 3
Cables	20	US\$ 2
	Total	US\$ 14

Elaboración propia de acuerdo con el análisis del proyecto

*Los precios fueron proporcionados desde la plataforma Ali Express, en dólares, debido a los constantes cambios por los procesos inflacionarios en nuestro país, complicando la realización de costos incluso en el corto plazo.

- **Inversión en software**

El software requerido para el proyecto en su mayoría es software libre. Los programas que se utilizarán son:

1. Arduino IDE
2. Andriod Studio
3. Firebase

Firestore en su mayoría es libre, pero, al integrar las notificaciones con los datos en tiempo real se utiliza una función de Firestore que es paga y se paga por uso. Actualmente el costo es insignificante, pero podría ser relevante en algún momento.

La inversión en software no requiere de erogaciones monetarias, lo único que podría ser considerado como un costo, es el tiempo que le demanda al equipo, la búsqueda, adquisición, demoras en estos procesos y la implementación de los programas, sumados a su verificación.

Los costos indirectos

Por el contrario, estos costos son los que se relacionan de manera tangencial con los proyectos o las tareas previstas. Es decir, no son aplicables a un producto específico.

Los costos indirectos aproximados a considerar en este proyecto son:

Tabla 25: Costos indirectos

Costos	Precio mensual	Precio total*
Inversión en electricidad	\$ 5000	\$ 10000
Inversión en internet	\$ 1500	\$ 3000
	Total	\$ 13000

Elaboración propia de acuerdo con el análisis del proyecto

*Estos costos fueron proporcionados por el Consejo Profesional de Agrimensores, Ingenieros y Profesionales Afines – COPAIPA, al mes de abril de 2022 según los costos de sus proyectos o los costos de ese momento que ellos disponían.

Costo total del proyecto

Tabla 26: Costos totales

Costos	Subtotal
Costos de RRHH	\$ 399.500
Costos de Hardware	\$ 4.080 (14 dólares)*
Costos de software	\$ 0
Costos indirectos	\$ 13.000
Total	\$ 416.580

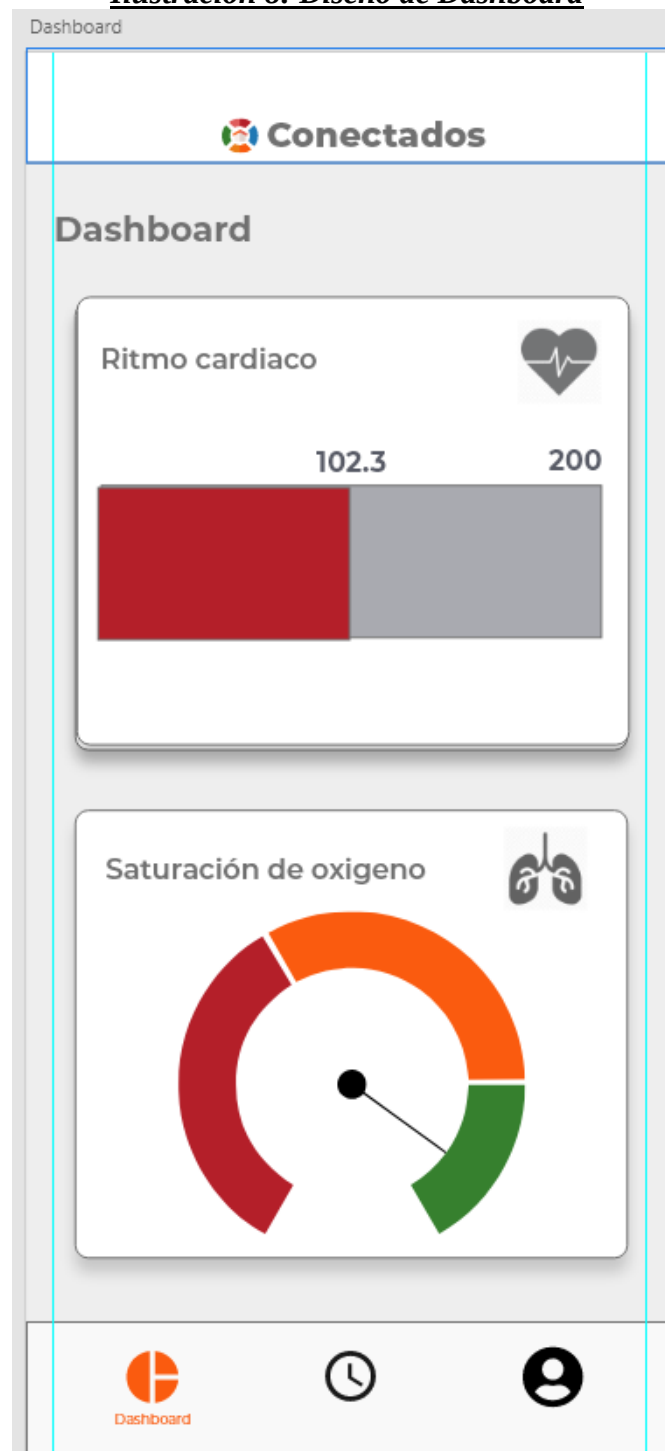
Elaboración propia de acuerdo con el análisis del proyecto

*Según cotización del dólar blue al día 25/10/22

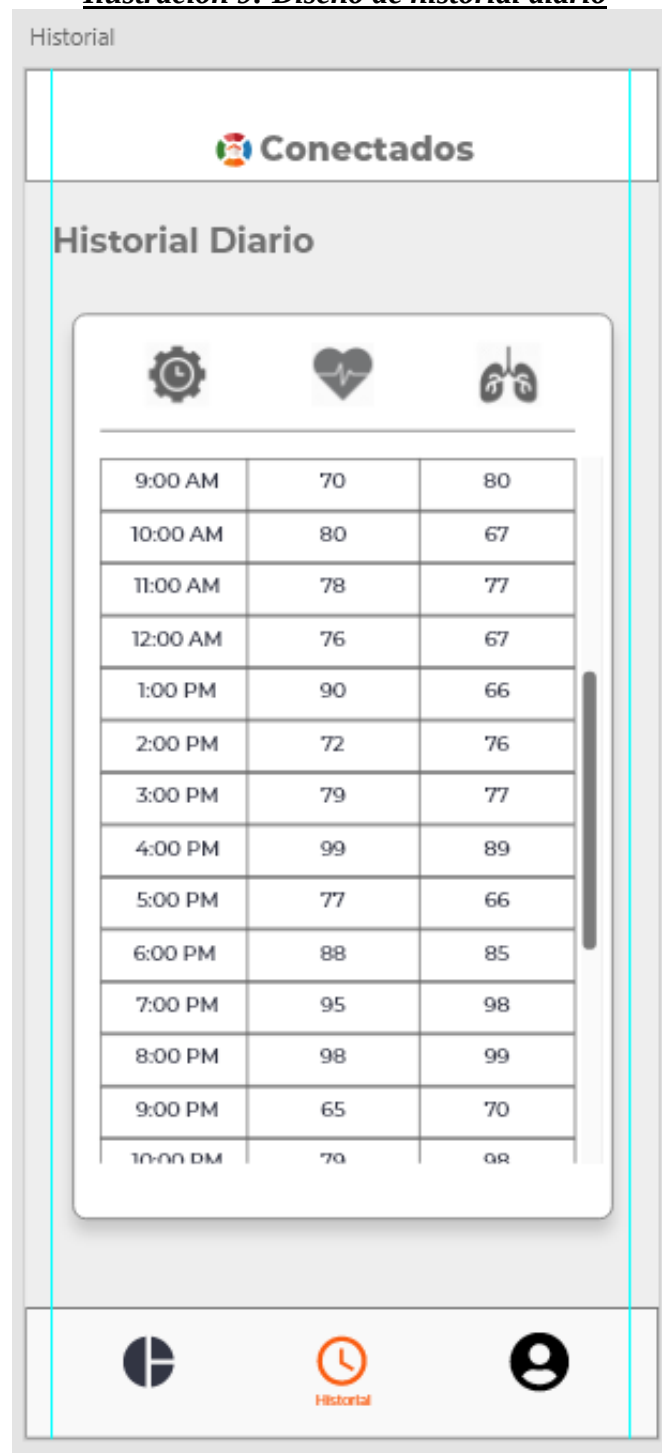
4.6.1.1. Diseño de pantalla de Inicio de Sesión**Ilustración 7: Pantalla de inicio**

The mockup shows a mobile application interface for logging in. At the top, the title 'Iniciar sesión' is displayed in a small, grey font. Below the title, the word 'Conectados' is prominently shown in a large, bold, orange font. Underneath this, there is a circular logo composed of five colored segments (red, green, blue, orange, and grey) arranged around a central white house icon. The login form consists of two input fields: 'EMAIL' with the text 'agustina.leal@hotmail.com' and 'CONTRASEÑA' with masked characters '*****'. At the bottom of the form, there are three orange buttons with white text: 'Registrarse', 'Iniciar Sesión', and 'Google Sign In'.

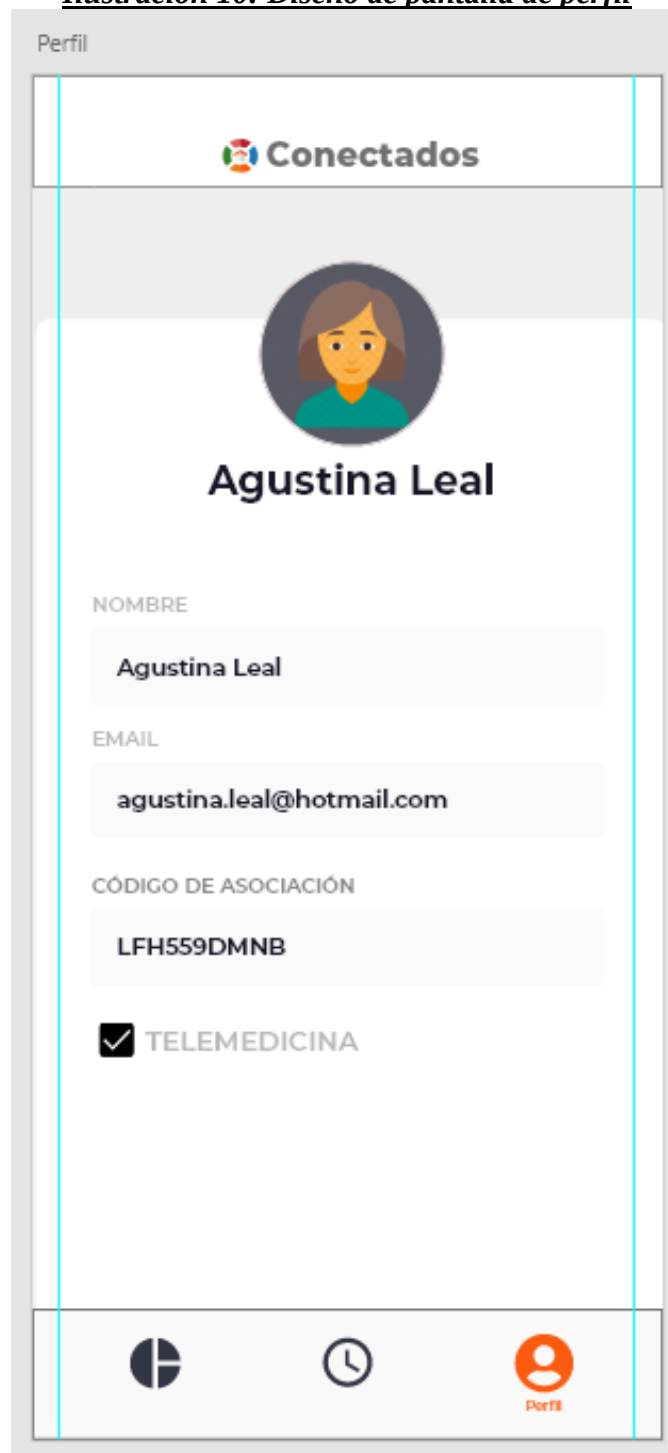
Fuente: Elaboración propia. De acuerdo con el diseño del proyecto

4.6.1.2. Diseño de pantalla de Dashboard/Visualización de datos en tiempo real**Ilustración 8: Diseño de Dashboard**

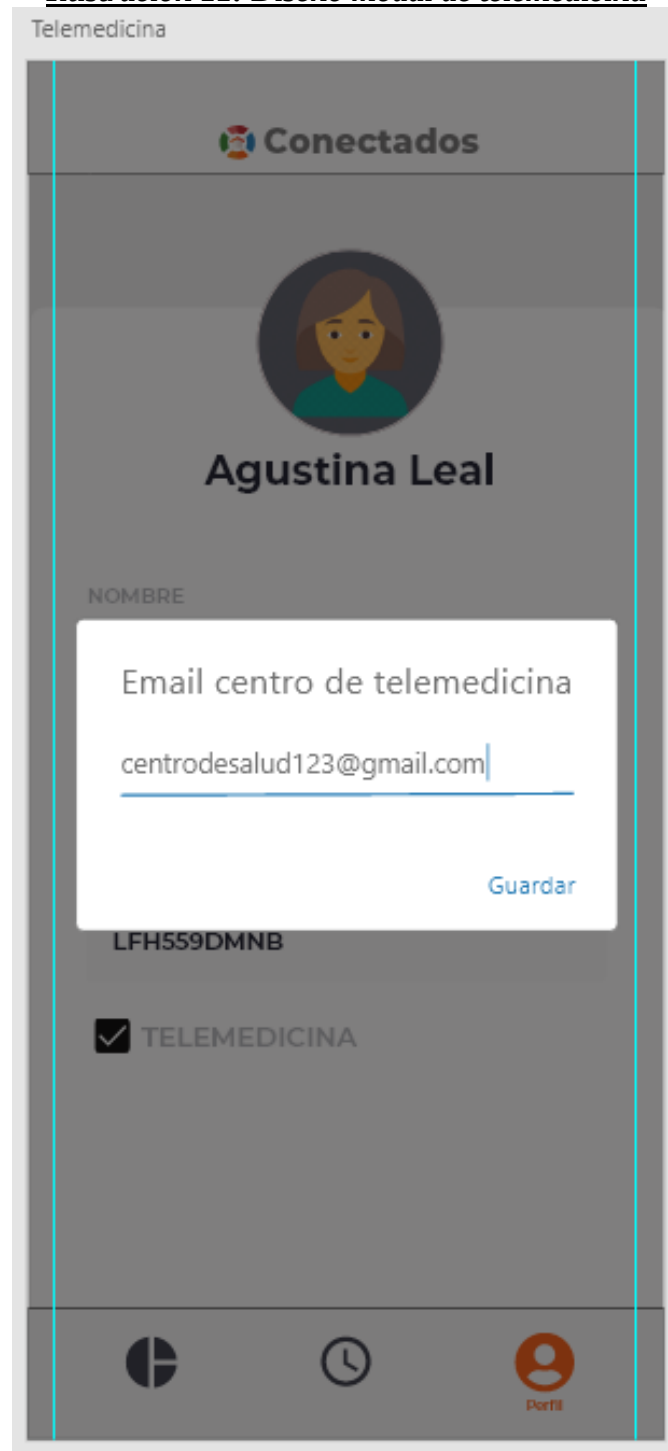
Fuente: Elaboración propia. De acuerdo con el diseño del proyecto

4.6.1.3. Diseño de pantalla de Historial Diario***Ilustración 9: Diseño de historial diario***

Fuente: Elaboración propia. De acuerdo con el diseño del proyecto

4.6.1.4. Diseño de pantalla de Perfil***Ilustración 10: Diseño de pantalla de perfil***

Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

4.6.1.5. Modal de telemedicina en pantalla de perfil***Ilustración 11: Diseño modal de telemedicina***

Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

4.7. Desarrollo de la solución

El desarrollo de la solución de la solución se llevó a cabo en varias partes, realizando pruebas en cada una de ellas. Las cuales comprendieron el desarrollo de la solución de hardware, de la cual se probó primero el funcionamiento del sensor mencionado en capítulos anteriores y luego el funcionamiento del Botón SOS y su correspondiente transmisión de los datos a la base de datos en Firebase. Además, el desarrollo de la solución de software, la cual se basó en el diseño visualizado en la sección 4.3.1 y se efectuaron las correspondientes pruebas para obtener los datos subidos por el sensor y el botón a Firebase.

4.7.1. Desarrollo de la solución de hardware

4.7.1.1. Sensor MAX 30102

El MAX30102⁸ es un sensor que incorpora las funciones de pulsímetro y oxímetro en un único integrado que podemos usar junto con un procesador como Arduino.

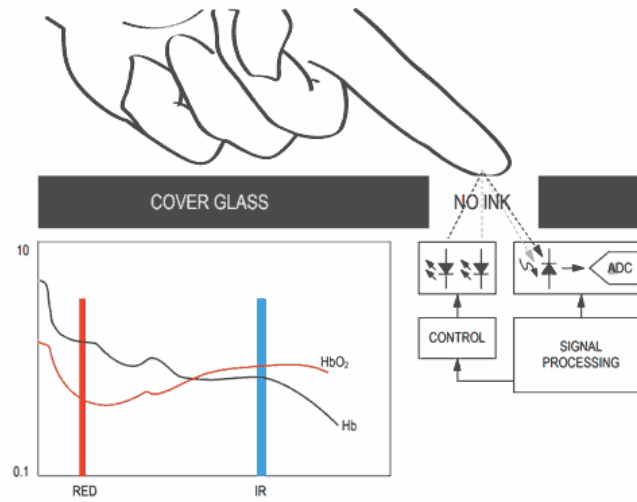
La serie MAX3010x es un sensor óptico, que basa su funcionamiento en el distinto comportamiento que la sangre tiene ante la luz, en función de su grado de saturación de oxígeno.

Para ello, el MAX30102 incorpora dos LED, uno de espectro rojo y otro de infrarrojo. El MAX30102 se pone sobre la piel, por ejemplo, en el dedo o la muñeca. El sensor detecta la luz reflejada, y determina el grado de saturación.

¿Cómo funciona un MAX30102?

La pulsioximetría es un método no invasivo para determinar el porcentaje de saturación de oxígeno en sangre. Su funcionamiento se basa en que la hemoglobina (Hb) y la hemoglobina saturada (oxihemoglobina, HbO₂) tienen distintos coeficientes de absorción de la luz para distintas longitudes de onda.

⁸ ¿QUÉ ES UN MAX30102? <https://www.luisllamas.es/pulsimetro-y-oximetro-con-arduino-y-max30102/> - 19/9/21

Ilustración 12: Pulsioximetría

Fuente: luisllamas.es

La sangre oxigenada absorbe mayor cantidad de luz infrarroja, mientras que la sangre poco oxigenada absorbe mayor luz roja. En partes del cuerpo donde la piel es suficiente, fina y bajo la que pasan vasos sanguíneos, **es posible emplear esta diferencia para determinar el grado de saturación.**

El MAX30102 incorpora dos LED, uno de espectro rojo (660 nm) y otro de infrarrojo (880 nm), así como fotodiodos para medir la luz reflejada y un ADC de 18 bits y frecuencia de muestreo de 50 sps (samples per second) a 3200 sps.

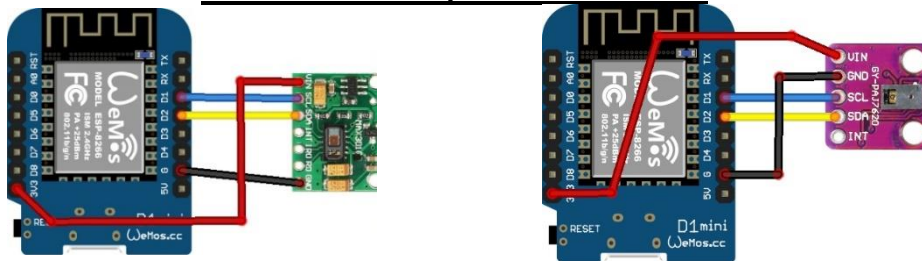
También dispone de la electrónica necesaria para la amplificación y filtrado de la señal, cancelación de luz ambiental y rechazo a frecuencias de 50-60 Hz (luz artificial) y compensación de temperatura.

El consumo del módulo es de hasta 50 mA durante la medición, aunque la intensidad puede ajustarse programación, con un modo de bajo consumo de 0.7 μ A durante las mediciones.

4.7.1.2. Esquema de conexión de MAX30102 con placa WEMOS D1 Mini

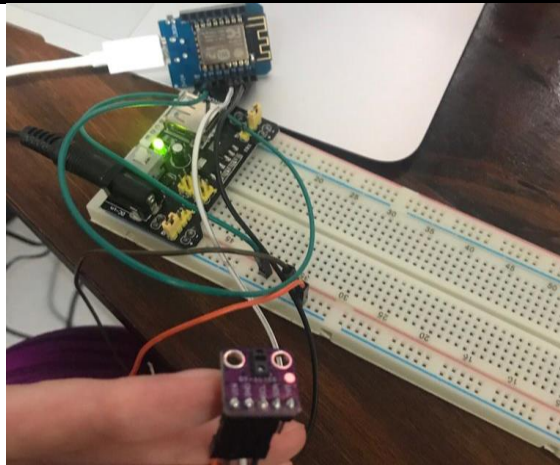
En una primera instancia de investigación se encontraron diferentes esquemas de conexión para los componentes mencionados o bien para algunos similares y se realizó algunas pruebas para verificar que estos esquemas y sus respectivos códigos de prueba en Arduino IDE arrojen datos.

Algunos de ellos fueron:

Ilustración 13: Esquemas de conexión

Fuente: Elaboración propia. Diseño de la solución de hardware

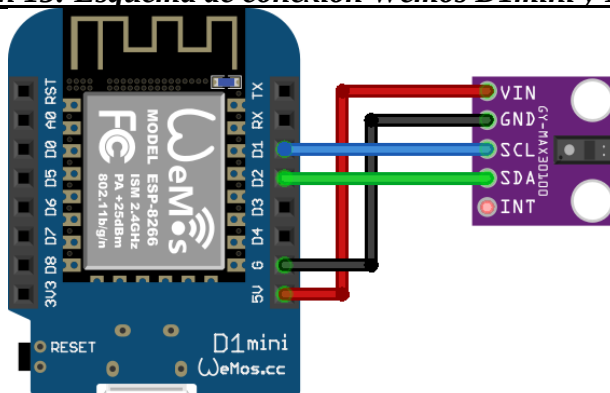
Con la utilización de dicho esquema se procedió al armado del circuito, en una primera instancia se conectó directamente el sensor con la placa con el puerto VIN del sensor en 3.3 V de la placa, pero en el monitor serie del Arduino IDE no se visualizaban datos, se probó también utilizando una fuente de protoboard, la cual se utiliza para alimentar circuitos eligiendo el voltaje de entrada y de salida, en este caso sería de 3.3 V. Pero este procedimiento tampoco arrojó datos.

Ilustración 14: Esquema de conexión protoboard

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

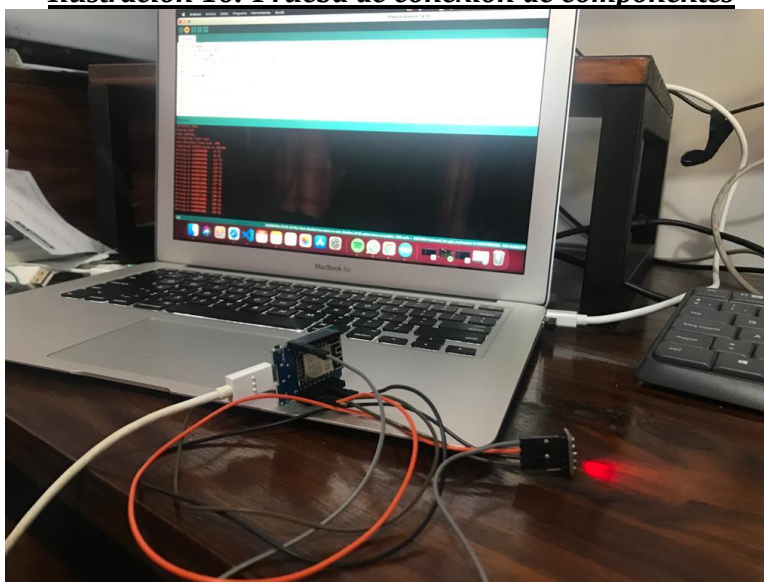
Una muy buena fuente de información para investigar acerca de esquemas de conexión, sensores y demás cuestiones de electrónica, es la web de Luis Llamas (<https://www.luisllamas.es/>), en específico para este proyecto se utilizó el artículo “PULSÍMETRO Y OXÍMETRO CON ARDUINO Y MAX30102”, el cual brindó la información necesaria para entender el funcionamiento del sensor y como debería ser la correcta conexión con nuestra placa Wemos D1 Mini Esp8266

El esquema de conexión utilizado fue el siguiente:

Ilustración 15: Esquema de conexión Wemos D1mini y MAX30100

Fuente: Elaboración propia. Diseño de la solución de hardware

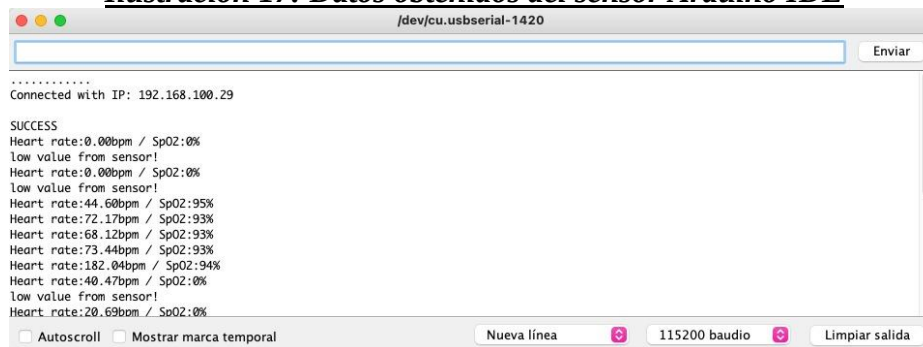
Una vez obtenido este esquema de conexión y se procedieron a realizar las pruebas sobre los componentes reales, a continuación, podremos ver una foto con la placa Wemos conectada al sensor, el mismo se encuentra con su luz encendida, lo que significa que está listo para enviar datos (en los intentos anteriores nunca se había logrado que se encienda la luz).

Ilustración 16: Prueba de conexión de componentes

Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

Los datos que pudimos visualizar en el Monitor Serie del Arduino IDE, acercando el sensor a la muñeca, fueron los siguientes:

Ilustración 17: Datos obtenidos del sensor Arduino IDE



Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

4.7.1.3. Botón SOS

En una primera instancia se había seleccionado como botón SOS un componente de la serie WEMOS, el mismo era el Wemos 1-Button Shield, el cual es un interruptor, conectado al pin D3, diseñado especialmente para la placa Wemos D1 mini.

Ilustración 18: Wemos 1-Button Shield

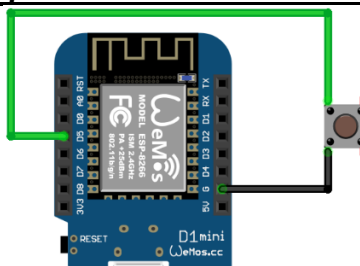


Fuente: Amazon

Pero no se logró hacer funcionar este componente, probablemente por alguna falla en la conexión o bien en el componente mismo. Al no poder adquirirlo de nuevo, ya que no se lo consigue en Argentina y no se están recibiendo importaciones desde China (donde se lo había comprado al componente), se optó por usar un botón común y corriente de circuitos eléctricos.

El esquema de conexión de dicho botón es muy sencillo:

Ilustración 19: Esquema de conexión Wemos D1 y botón



Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Para probar la conexión entre el botón y la placa se escribió un código en Arduino IDE muy simple el cual detecta cuando se ha presionado el botón y muestra un mensaje en el Monitor Serie del IDE.

Ilustración 20: Funcionamiento del botón en monitor serie de Arduino IDE



Elaboración propia. De acuerdo con el desarrollo del proyecto

4.7.1.4. Implementación de Firebase en código de Arduino IDE

Necesitamos poder visualizar los datos enviados por el sensor en nuestra base de datos, que a su vez también funciona como suerte de back-end de la aplicación móvil. Para ello se debieron realizar ciertas configuraciones en Firebase y en Arduino IDE. Dichas configuraciones permiten a la placa ESP8266 conectarse a una red WI-FI, obtener datos del sensor y escribirlos en Firebase.

Con respecto al sensor, cómo este puede tomar datos erróneos o nulos, se tuvieron en cuenta estas condiciones en el código que se compiló en la placa mencionada. Todo esto en búsqueda de obtener lecturas limpias y certeras.

Podemos ver estas condiciones en el siguiente extracto de código:

Ilustración 21: Código para subir datos del sensor a Firebase



Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Además, para el botón SOS, también se definió una condición a la hora de subir datos, la misma es que el valor del botón sea distinto a su ultimo valor, es decir, si en primera instancia el botón no estaba presionado comienza teniendo como valor FALSE, una vez que sea presionado ese valor cambiara a TRUE. Más explícitamente lo podemos ver en el extracto de código que carga datos del botón en Firebase:

Ilustración 22: Código con condiciones para subir datos del botón a Firebase



```
boolean btnPressed = !digitalRead(D5);  
if (btnPressed == true && lastState == false)  
{  
    Firebase.RTDB.setStringAsync(&fbdo, "button/status", "yes");  
}  
lastState = btnPressed;
```

Fuente: Elaboración propia. De acuerdo con el testeado del proyecto

Hay otra sección de código que carga datos en Firebase del sensor, pero no en tiempo real, la sigla RTBD que vemos en los fragmentos de código anteriores significa Base de Datos en Tiempo Real (Real Time Database). Está otra sección mencionada carga datos cada 1 hora y es para la pantalla/funcionalidad de historial diario diseñada en la sección 4.6.1.3.

El código completo para el sensor y el botón en Arduino IDE lo encontraremos en el Anexo 8.7.

4.7.2. Desarrollo de la solución de software

Para el desarrollo de la solución de software se fue realizando de manera conjunta el código basado en framework Flutter con el IDE Android Studio y la configuración de Firebase en el proyecto.

Primero se realizó la creación de la aplicación Flutter con Android Studio, las configuraciones realizadas las podemos ver en el Anexo 8.8. y luego creamos el proyecto en Firebase y lo configuramos para agregarlo a la aplicación, los pasos realizados se encuentran en el Anexo 8.9.

Además, se creó el repositorio de código en Github⁹, este alojador de código permite a los desarrolladores administrar sus proyectos, manejar distintas versiones, no perder estados anteriores de un código cuando se realizan refactorizaciones, entre otras ventajas.

En el desarrollo del código se trataron de respetar las buenas prácticas de manejo de repositorios, pero para entenderlas mejor vamos a definir algunos conceptos:

- Git se basa en **snapshots** (instantáneas) del código en un estado determinado, que viene dado por el autor y la fecha.
- Un **Commit** es un conjunto de cambios guardados en el repositorio de Git y tiene un identificador SHA1 único.
- Las ramas (**branches**) se pueden pensar como una línea de tiempo a partir de los commit. Hay siempre como mínimo una rama principal o predefinida llamada Master.
- **Head** es el puntero al último commit en la rama activa.
- **Remote** se refiere a sitios que hospedan repositorios remotos como GitHub.

Las buenas prácticas para manejo de repositorios se detallan en el anexo 8.10. En específico para el proyecto se utilizaron las siguientes:

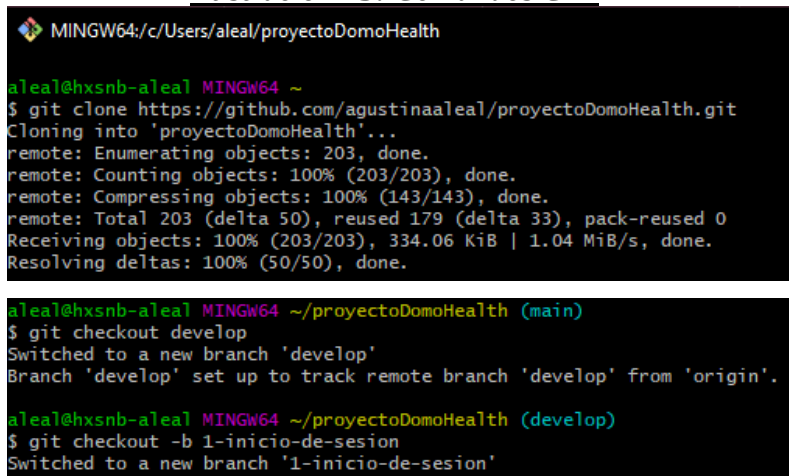
1. Prueba los cambios antes de hacer commit
2. Usa diferentes ramas
3. Realiza commit frecuentemente
4. Escribe mensajes de confirmación útiles

4.7.2.1. Pantalla de Inicio de Sesión

Antes de comenzar a desarrollar la pantalla de login o inicio de sesión se verificó el diseño de la misma y se creó la rama en GitHub desde la rama develop, para ello se ejecutan los siguientes comandos en el bash:

⁹ <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores> - Visitada 7/9/21

Ilustración 23: Comandos GIT



```

MINGW64/c/Users/aleal/proyectoDomoHealth

aleal@hxsnb-aleal MINGW64 ~
$ git clone https://github.com/agustinaaleal/proyectoDomoHealth.git
Cloning into 'proyectoDomoHealth'...
remote: Enumerating objects: 203, done.
remote: Counting objects: 100% (203/203), done.
remote: Compressing objects: 100% (143/143), done.
remote: Total 203 (delta 50), reused 179 (delta 33), pack-reused 0
Receiving objects: 100% (203/203), 334.06 KiB | 1.04 MiB/s, done.
Resolving deltas: 100% (50/50), done.

aleal@hxsnb-aleal MINGW64 ~/proyectoDomoHealth (main)
$ git checkout develop
Switched to a new branch 'develop'
Branch 'develop' set up to track remote branch 'develop' from 'origin'.

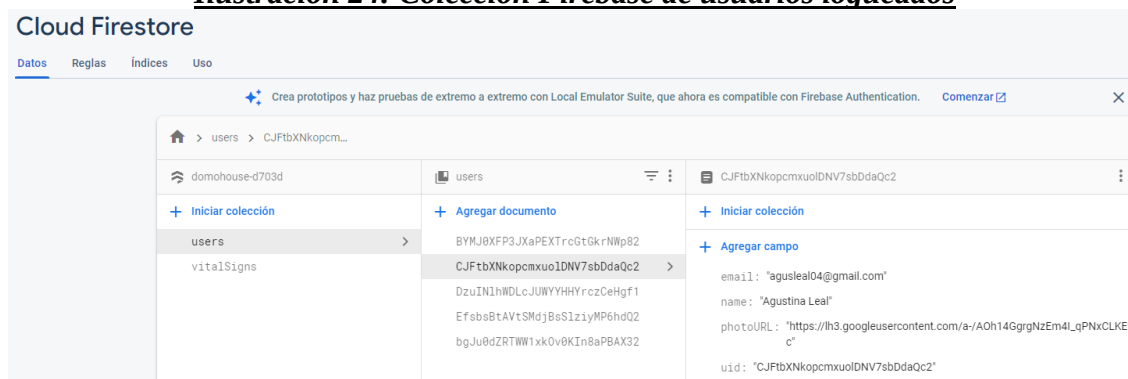
aleal@hxsnb-aleal MINGW64 ~/proyectoDomoHealth (develop)
$ git checkout -b 1-inicio-de-sesion
Switched to a new branch '1-inicio-de-sesion'
  
```

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Una vez creada esta rama y posicionados dentro de ella con el último comando, se comenzó el desarrollo del código sobre el entorno Android Studio. Luego de una serie de varias horas de codificación a prueba y error, se obtuvo la primera pantalla de la aplicación con su conexión a Firebase para la Autenticación.

La autenticación se puede realizar registrándose con un email y contraseña o bien con una cuenta de Google. Estos datos se guardan en Firestore y luego son mostrados en la pantalla de perfil de la aplicación. Además, una vez autenticados, obtenemos un ID que permitirá la conexión con el prototipo.

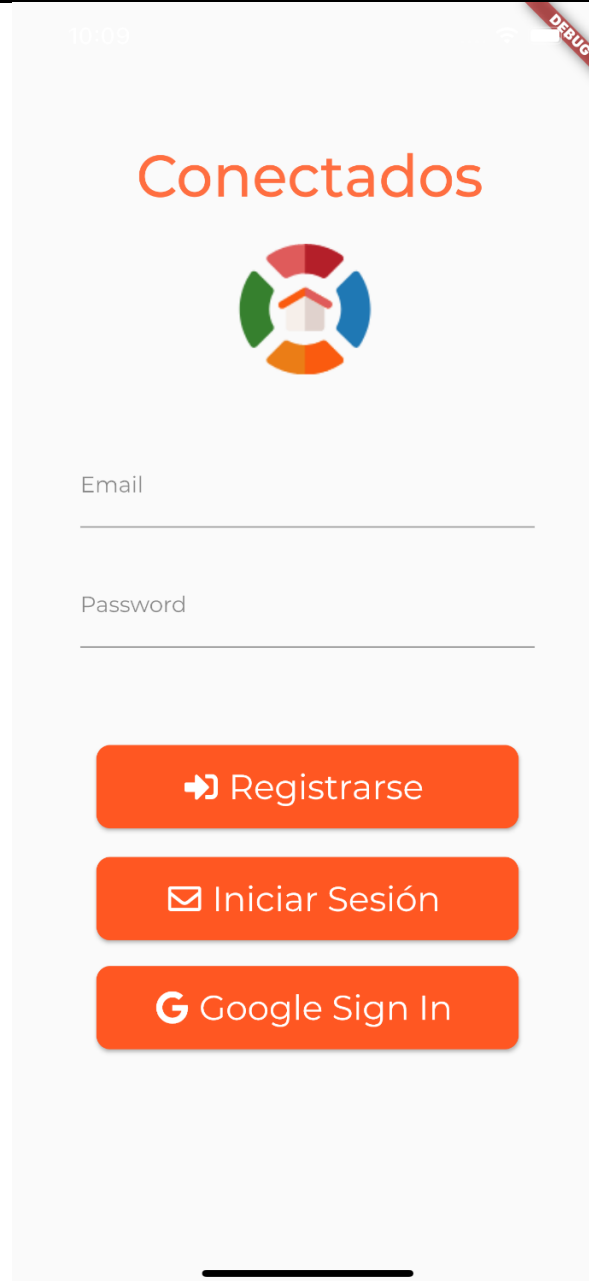
Ilustración 24: Colección Firebase de usuarios logeados



Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

El resultado de la creación de la pantalla fue:

Ilustración 25: Resultados de pantalla de inicio de sesión



Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

4.7.2.2. Pantalla de dashboard/Visualización de datos en tiempo real

Como vimos anteriormente, antes de iniciar el desarrollo de un nuevo feature o funcionalidad debemos crear con los comandos git la nueva rama, para este caso podría haber sido

“2-Dashboard”, pero para esta pantalla primero se desarrollaron dos elementos de la pantalla que se repiten en el resto, estos son conocidos como AppBar y TabBar en desarrollo de aplicaciones móviles. Luego se continuó desarrollando la vista en sí de esta pantalla en una nueva rama.

Primero se diseñó la estructura de la vista, es decir, el título, las tarjetas, cuáles y como iban a ir los gráficos, entre otras cuestiones. Una vez establecida esta estructura y habiendo verificado que coincidiera con el diseño realizado, se procedió a obtener los datos subidos por el sensor a la base de datos en tiempo real de Firebase.

Los datos subidos a la RTDB se mostraban de la siguiente manera:

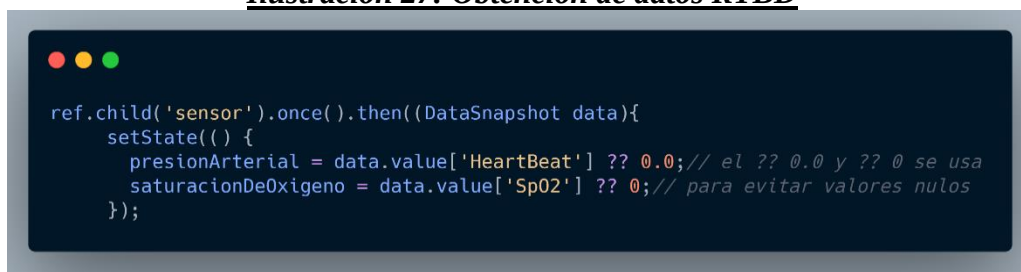
Ilustración 26: Datos de sensor RTBD



Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

En código para obtener estos datos, básicamente se realiza una instantánea (snapshot¹⁰) de los datos y se los guarda en variables para luego ser usado en los gráficos.

Ilustración 27: Obtención de datos RTBD

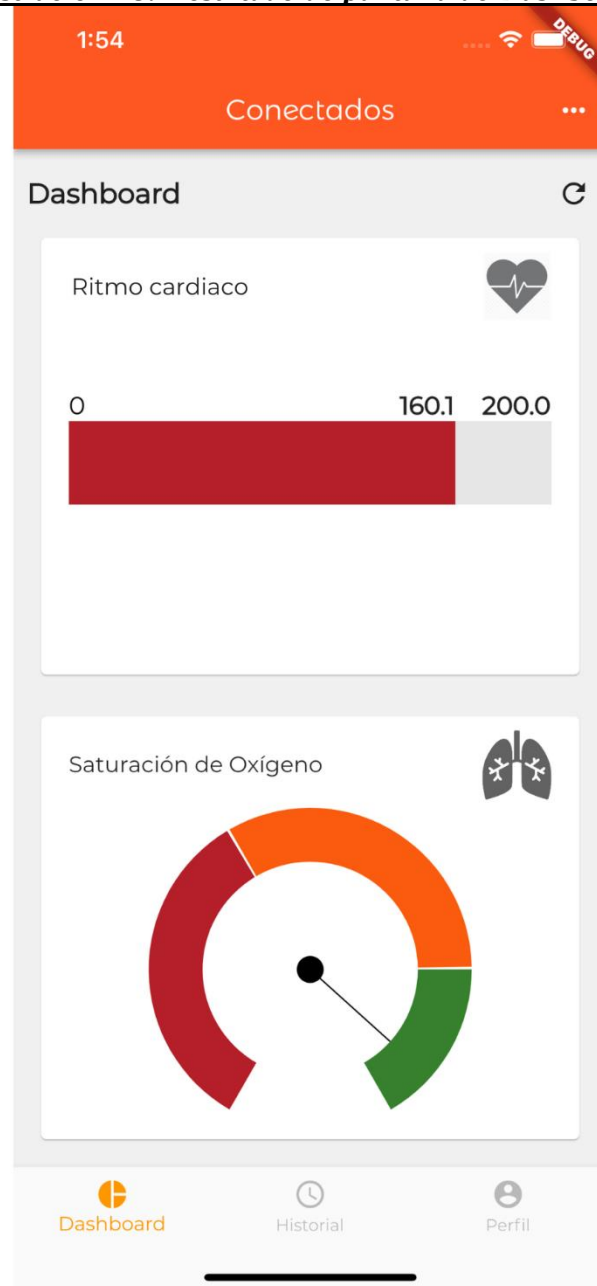


Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

¹⁰ Snapshot: La creación del snapshot consiste en tomar una «fotografía», una instantánea de un volumen lógico concreto (un volumen lógico puede ser una base de datos convencional).

El resultado de esta pantalla con los datos del sensor y el TabBar y Appbar es:

Ilustración 28: Resultado de pantalla de Dashboard



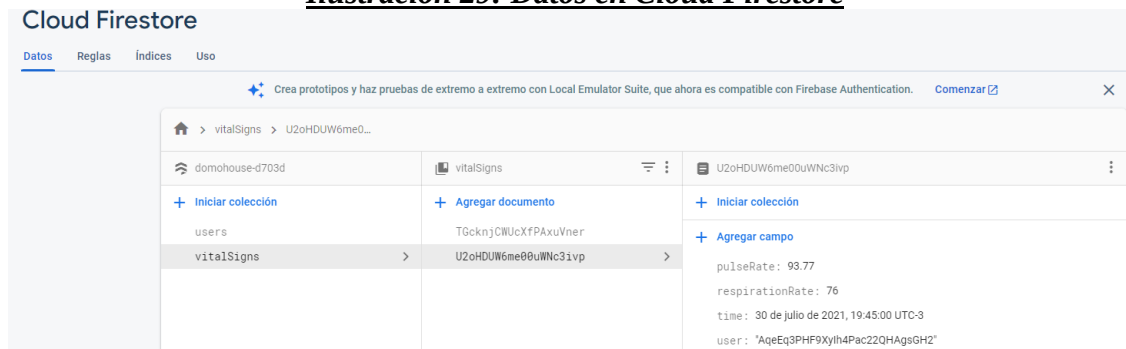
Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

4.7.2.3. Pantalla de Historial Diario

Para el desarrollo de esta pantalla se tuvo en cuenta que los datos obtenidos en Firestore serán cada una hora en el mejor de los casos y que solo se deben visualizar aquellos que hayan sido tomados en el día en curso.

Los datos en Firestore obtenidos del sensor se guardaban de la siguiente manera:

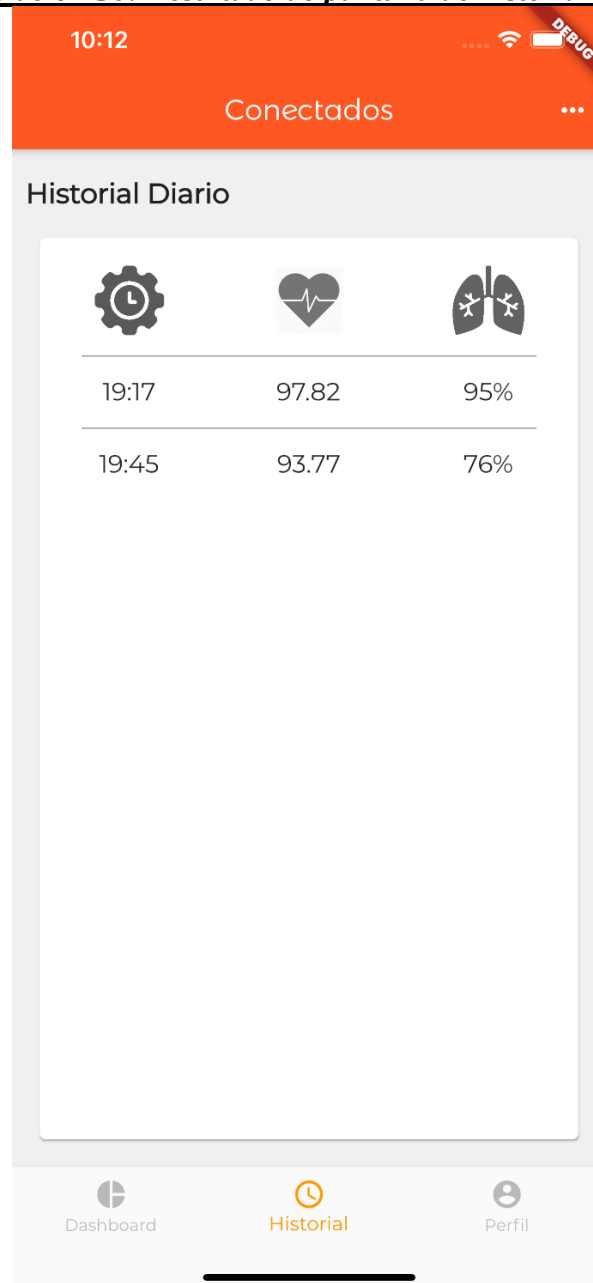
Ilustración 29: Datos en Cloud Firestore



Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Hubo algunos inconvenientes con la zona horaria, debido a que Firestore almacena sus datos con la zona horaria central UTC 0 y desde el sensor llegaban con la zona horaria de Argentina UTC -3. Oportunamente, se pudieron resolver estos inconvenientes.

La pantalla de historial diario es muy sencilla, pero lleva bastante lógica detrás para obtener los datos y plasmarlos en la tabla diseñada. El resultado de esta fue:

Ilustración 30: Resultado de pantalla de historial diario

Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

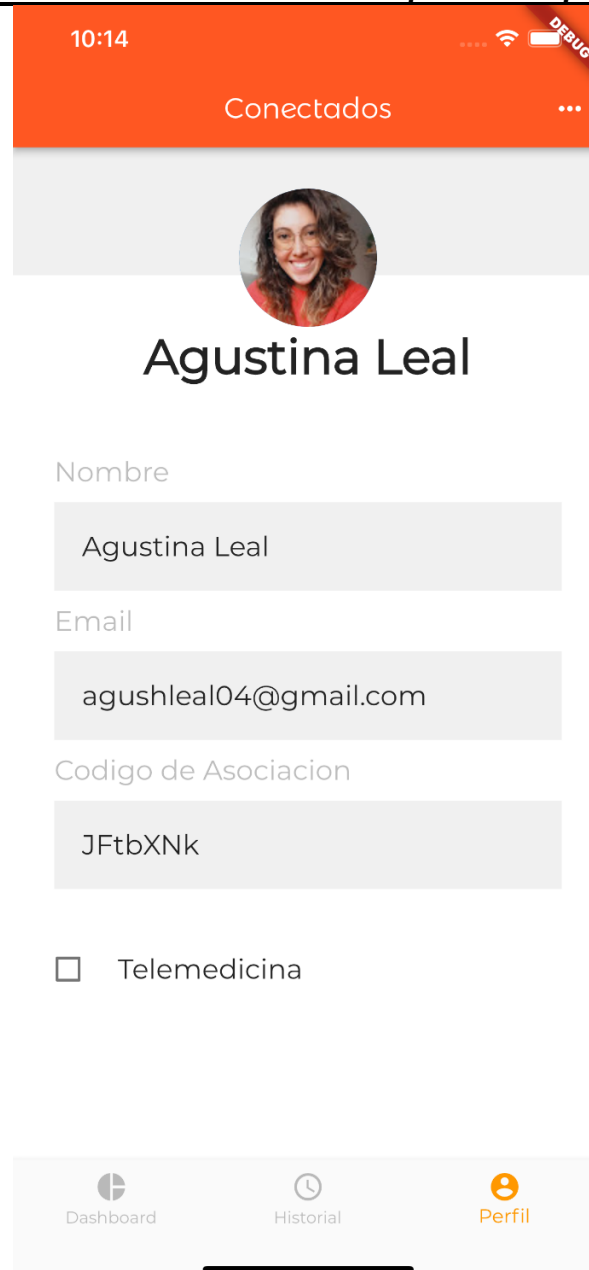
4.7.2.4. Pantalla de Perfil

La última pantalla es la de perfil, esta misma toma los datos guardados en la autenticación que vimos en la captura de pantalla de la colección Users en Firestore de la pantalla de inicio de sesión.

En esta que se puede visualizar el código de asociación que permite asociar la aplicación móvil a la placa ESP8266. Además, también se visualiza un “checkbox” para tildar si el usuario de la aplicación está siendo monitoreado por un centro de telemedicina (ver sección 4.7.2.5. Telemedicina)

El desarrollo de esta pantalla devolvió lo siguiente:

Ilustración 31: Resultados de la pantalla de perfil



10:14

Conectados

Agustina Leal

Nombre

Agustina Leal

Email

agushleal04@gmail.com

Codigo de Asociacion

JFtbXNk

☐ Telemedicina

Dashboard Historial Perfil

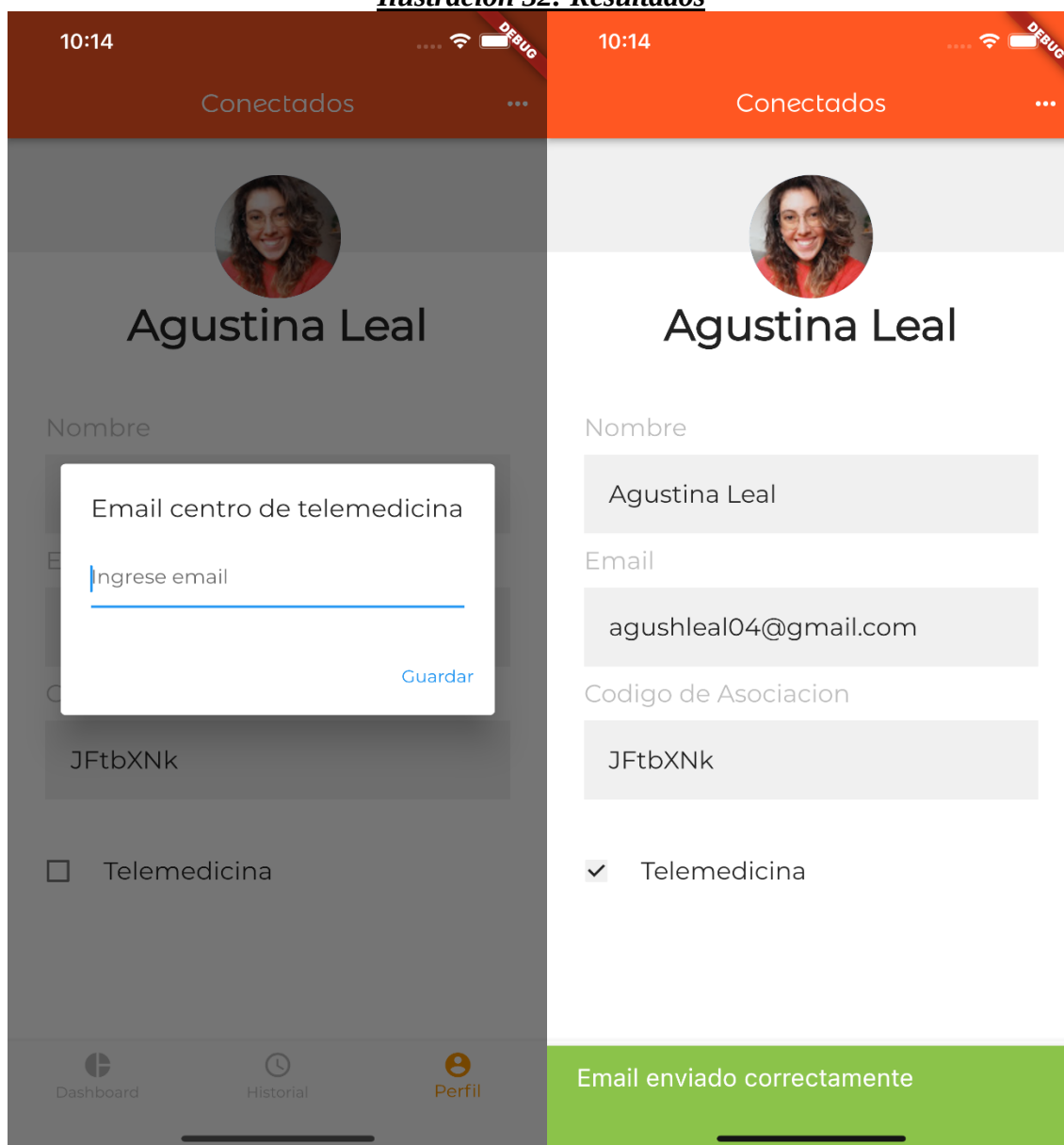
Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

4.7.2.5. Telemedicina

Con respecto al envío de reportes de telemedicina, se especificó hacerlos a través de mails desde la aplicación, mandando diario un mail con el reporte del promedio de saturación de oxígeno del usuario/paciente en ese día.

Desde la pantalla de perfil de la aplicación se puede ingresar el email correspondiente al centro de telemedicina que debe recibir el reporte y se visualiza en la aplicación de la siguiente manera:

Ilustración 32: Resultados



Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

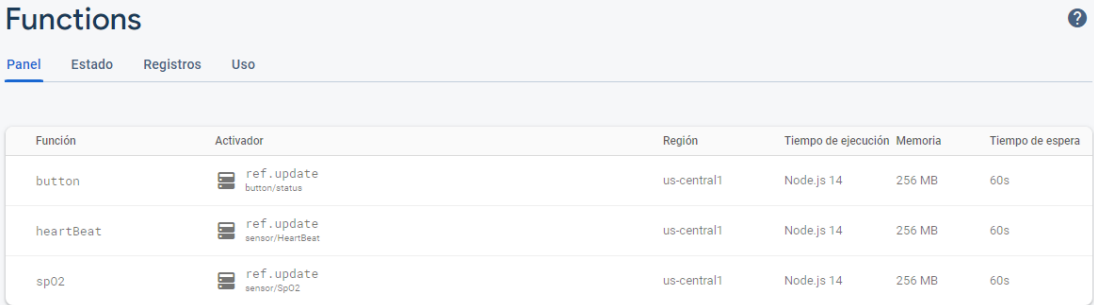
4.7.2.6. Notificaciones

Para el desarrollo y configuración de las notificaciones se utilizó una herramienta de Firebase llamada Cloud Functions que permite integrar las notificaciones propiamente de Firebase con Cloud Messaging y los datos que se van cargando/subiendo en tiempo real a la RTDB.

Este código se realizó con el lenguaje NodeJs, el cual funciona como backend de Firebase, y se despliega las funciones que sean necesarias para las notificaciones. En nuestro caso fueron 2 funciones, una para los valores anómalos del sensor y otra para que notifique cuando se presiona el botón.

Dentro de Firebase podemos ver estas funciones de la siguiente manera:

Ilustración 33: Funciones en Firebase



Functions					
Panel	Estado	Registros	Uso		
Función	Activador	Región	Tiempo de ejecución	Memoria	Tiempo de espera
button	ref.update button/status	us-central1	Node.js 14	256 MB	60s
heartBeat	ref.update sensor/HeartBeat	us-central1	Node.js 14	256 MB	60s
spO2	ref.update sensor/SpO2	us-central1	Node.js 14	256 MB	60s

Fuente: Elaboración propia. De acuerdo con el testeado del proyecto

En el siguiente fragmento de código se puede observar que la función “button” se activa cuando se actualiza el valor de button/status en la RTBD.

Ilustración 34: Código con botón SOS

```
exports.button = functions.database.ref('button/status').onUpdate (evt => {  
  const payload = {  
    notification: {  
      title: 'BOTÓN SOS',  
      body: `El usuario puede estar requiriendo atención`,  
      badge: '1',  
      sound: 'default'  
    }  
  };  
  
  return admin.database().ref('fcm-token').once('value').then(allToken => {  
    if (allToken.val() && (evt.after.val() == 'yes')) {  
      console.log('token available');  
      const token = Object.keys(allToken.val());  
      return admin.messaging().sendToDevice(token, payload);  
    } else {  
      console.log('no token available');  
    }  
  });  
});
```

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Las funciones “HeartBeat” y “SpO2” se activan cuando se actualizan sus valores en la RTBD y estos cumplen con los criterios de anómalos. El código de estas funciones es:

Ilustración 35: Código de notificaciones

```
exports.spO2 = functions.database.ref('sensor/SpO2').onUpdate
((evt: { after: { val: () => number; }; }) => {
  const payload = {
    notification: {
      title: 'Valor Anómalo',
      body: `El valor de saturación de oxígeno es de:
${evt.after.val()}`,
      badge: '1',
      sound: 'default'
    }
  };

  return admin.database().ref('fcm-
token').once('value').then((allToken: { val: () => {}; }) => {
    if (allToken.val() && evt.after.val() < 80) {
      console.log('token available');
      const token = Object.keys(allToken.val());
      return admin.messaging().sendToDevice(token, payload);
    } else {
      console.log('no token available');
    }
  });
});

exports.heartBeat =
functions.database.ref('sensor/HeartBeat').onUpdate ((evt: {
after: { val: () => number; }; }) => {
  const payload = {
    notification: {
      title: 'Valor Anómalo',
      body: `El valor del ritmo cardíaco es de:
${evt.after.val()}`,
      badge: '1',
      sound: 'default'
    }
  };

  return admin.database().ref('fcm-
token').once('value').then((allToken: { val: () => {}; }) => {
    if (allToken.val() && (evt.after.val() < 60 ||
evt.after.val() > 160)) {
      console.log('token available');
      const token = Object.keys(allToken.val());
      return admin.messaging().sendToDevice(token, payload);
    } else {
      console.log('no token available');
    }
  });
});
```

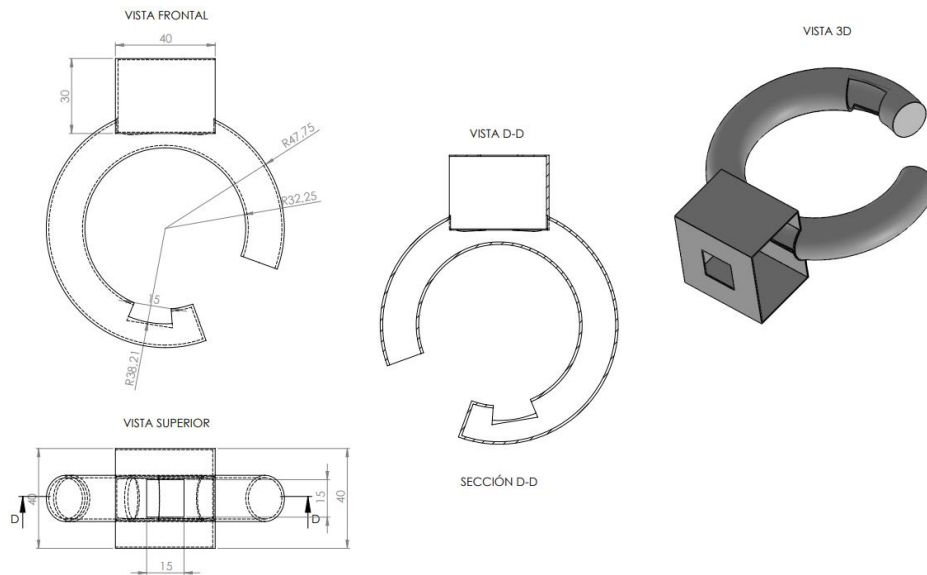
Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

5. VERIFICACIÓN EXPERIMENTAL

Para probar las funcionalidades, la aplicación y los componentes en sí, primero se procedió a la creación de un prototipo de reloj pulsera para conectar los componentes y poder probarlos en la muñeca de una persona.

El prototipo para la impresión en 3D fue el siguiente:

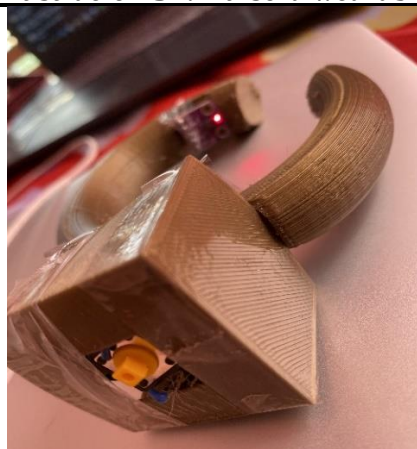
Ilustración 36: Prototipo



Fuente: Elaboración propia. De acuerdo con el diseño del proyecto

Una vez impreso el mismo se procedió ubicar el sensor, el botón y la placa ESP dentro del mismo, obteniendo:

Ilustración 37: Pulsera wearable



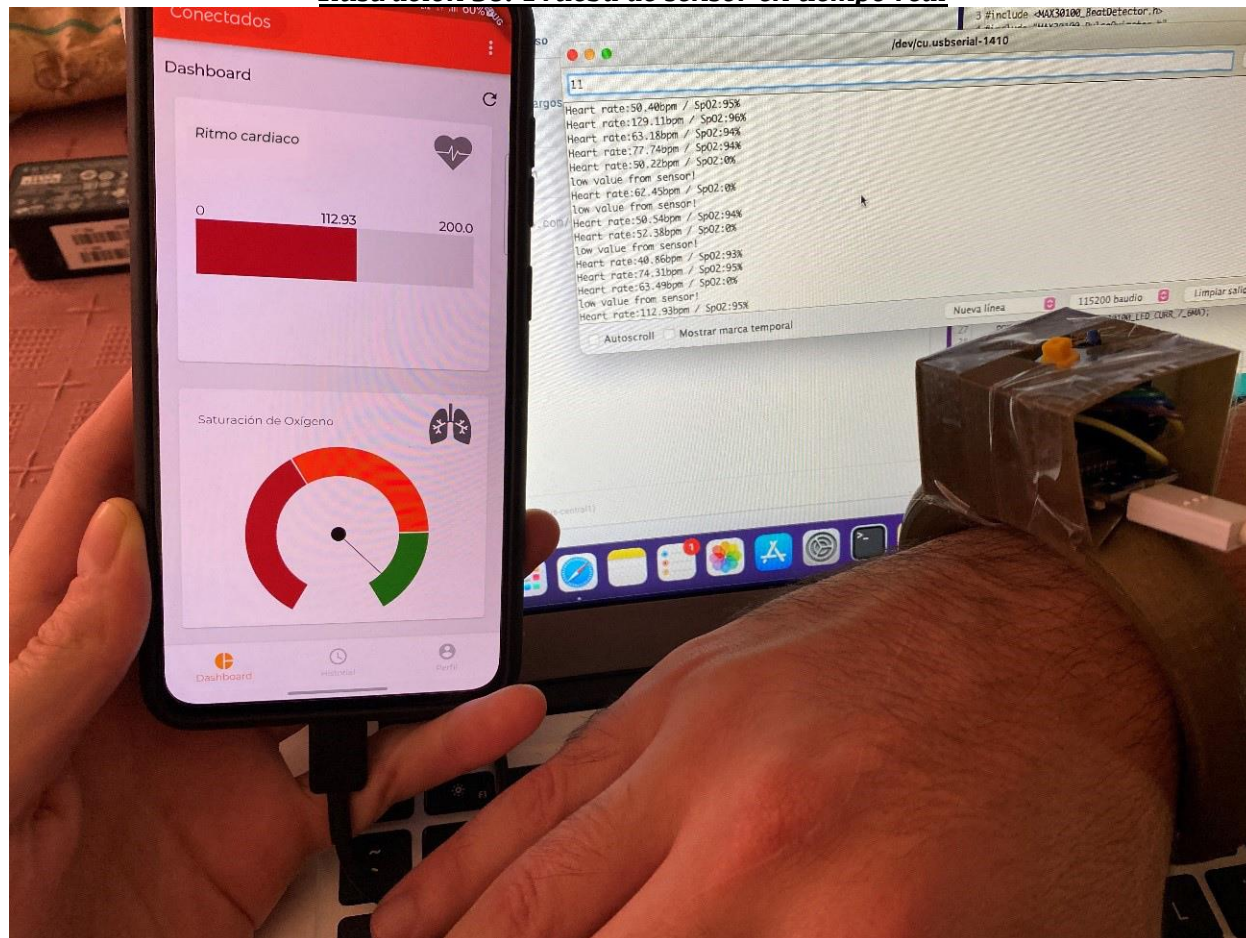
Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

5.1. Prueba del sensor MAX 30100, botón y notificaciones

Una vez armado el prototipo, se procedió a conectarlo a la computadora y a realizar pruebas con el ARDUINO IDE para visualizar si se estaban capturando correctamente los datos, los resultados obtenidos fueron:

En el monitor serial del IDE:

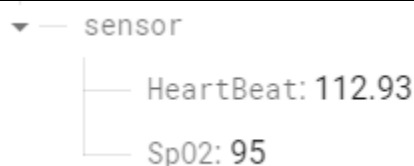
Ilustración 38: Prueba de sensor en tiempo real



Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

En Firebase (se visualiza la última lectura no errónea):

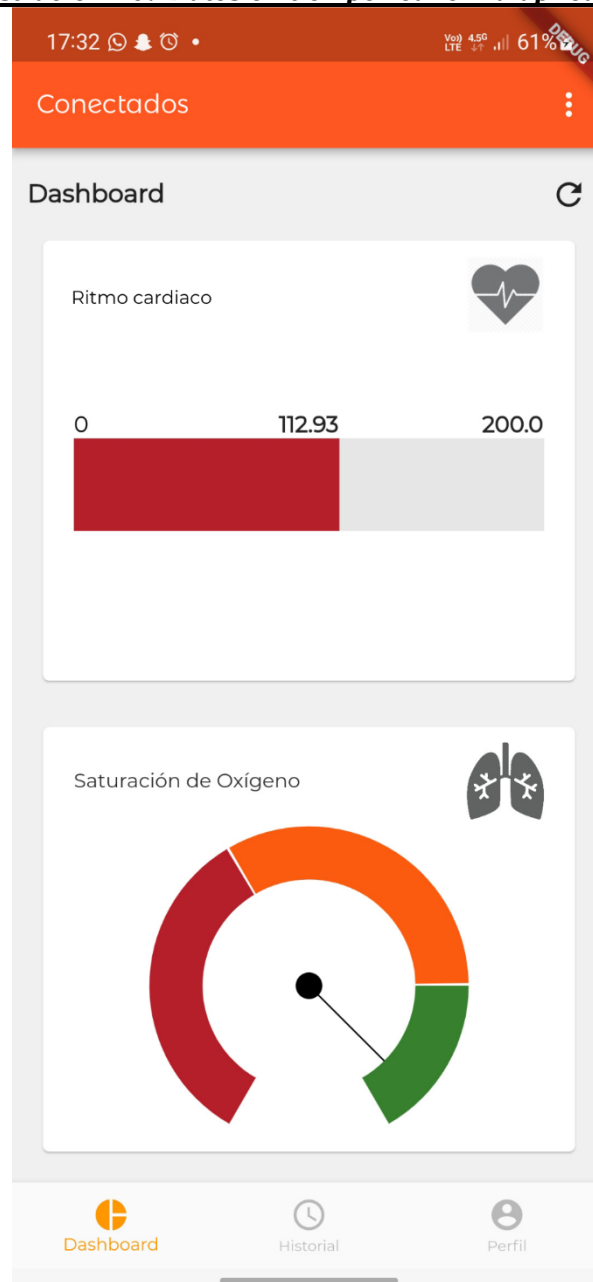
Ilustración 39: Datos del sensor RTBD



Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

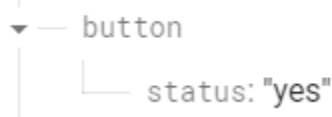
En la aplicación se tomó una captura de pantalla donde se visualiza como se ven los datos tomados desde Firebase:

Ilustración 40: Datos en tiempo real en la aplicación



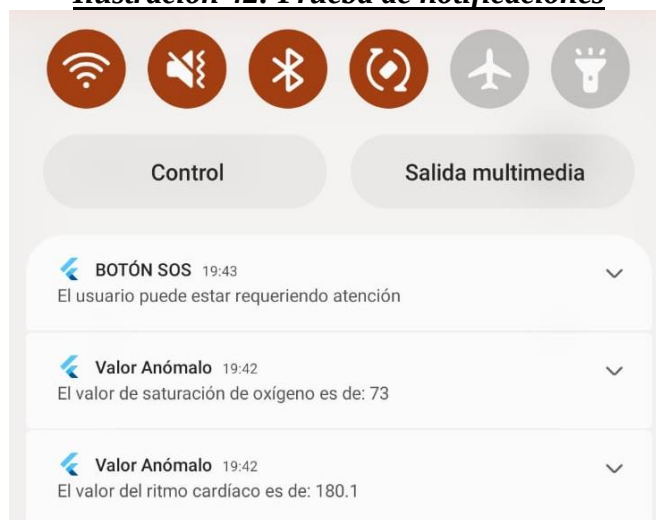
Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

Luego de presionar el botón SOS:

Ilustración 41: Datos del botón RTBD

Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

Estos datos son tomados por la RTBD de Firebase y en el caso del botón, se dispara una notificación al dispositivo móvil.

Ilustración 42: Prueba de notificaciones

Fuente: Elaboración propia. De acuerdo con el testeo del proyecto

Por último, se realizaron pruebas para verificar el correcto envío de mails desde la aplicación.

Mientras se realizaban las pruebas surgió un inconveniente con el servicio nativo de Gmail y Firebase para enviar mail debido faltas de permisos porque la aplicación no se encuentra publicada en dominios privados (estos dominios deben pagarse). Es por ello que se procedió a utilizar un servidor de mails público y gratuito para realizar pruebas. El mismo se denomina “EmailJS¹¹” y permitió enviar el siguiente mail desde la aplicación:

¹¹ <https://www.emailjs.com/>

Ilustración 43: Prueba de e-mail

Fuente: Elaboración propia. De acuerdo con el testeado del proyecto

6. CONCLUSIONES

Cuando se comenzó el desarrollo del presente proyecto de grado ni en las mejores gestiones de riesgo hubiéramos previsto que íbamos a atravesar una pandemia global al nivel de la que aún estamos viviendo. La pandemia nos llevó a reflexionar mucho acerca del valor de nuestra vida en la tierra y con más razón, el valor de nuestros adultos mayores, quienes fueron los más afectados.

El proyecto en su conjunto, aplicación móvil y prototipo de hardware se proponen como una opción cómoda y fácil de utilizar para un monitoreo básico de cualquier usuario, sea un adulto mayor, una persona con una patología respiratoria o como antes se menciona, un paciente con COVID-19. También sería útil para agilizar la recolección de datos para los centros de telemedicina.

En lo que se refiere al objetivo planteado para el proyecto, se puede concluir que, si bien a pesar de las limitaciones encontradas con los sensores y placa por sus complicaciones para ingresarlos al país y su fragilidad a la hora de hacer pruebas con ellos, se logró comprender acabadamente su funcionamiento y obtener datos de los mismos para hacer uso de ellos en la aplicación desarrollada, permitiendo un monitoreo fiable y continuo de una persona. Si bien sólo se desarrolló un prototipo funcional, los precios de los componentes utilizados para el armado del *wearable* se encuentran muy por debajo de cualquier de otras soluciones disponibles en el mercado, por lo tanto, se considera cumplido el objetivo de disponer de un dispositivo económico para las personas.

Con respecto a la aplicación móvil desarrollada, se puede concluir que si bien, Flutter, que es un lenguaje de programación muy potente para el desarrollo de aplicación multiplataforma, hoy en día se encuentra en constantes cambios y evolución, lo que provoca tener que realizar modificaciones y actualizaciones continuamente de librerías y códigos que quedan deprecados. Esta situación provocó bastantes retrasos en el desarrollo de la misma.

Con relación al prototipo hardware, la mayor desventaja se presentó a la hora de conseguir los recursos, debido a que debían ser importados y en Argentina esto suele ser un problema que se incrementó aún más con la pandemia. A pesar de los largos tiempos de demora para adquirir el hardware, el prototipo se pudo llevar a cabo de manera correcta, permitiendo realizar las pruebas pertinentes.

6.1. Futuras líneas de investigación

Como futuras líneas de investigación del proyecto se podría considerar:

- Adaptar la aplicación para recolectar datos de cualquier tipo de smartwatch en el mercado (actualmente hay infinidad de opciones y muchas muy económicas aún aquí en la Argentina, cuando se inició el proyecto esto no era así).
- Expandir los usuarios finales a pacientes de una terapia intermedia, por ejemplo, y que el personal de enfermería pueda monitorear su estado general y recibir alertas.
- Agregar otros sensores al dispositivo para monitorear el estado de salud general de los usuarios del mismo, por ejemplo, presión arterial, glucosa en sangre, sensor de temperatura corporal, etc. Este último también permitiría mejorar la recolección de datos para paciente con COVID-19.
- Cifrado de la información (Integridad, confidencialidad y disponibilidad de la información).

7. BIBLIOGRAFIA

- Arlegui, M. A. (2011). *Situación de los adultos mayores en la Argentina - Activos y capacidad de respuesta frente a la vejez en contextos vulnerables*.
<https://www.gerontologia.org/portal/archivosUpload/Situacion-adultos-mayores-Argentina.pdf>
- Assaf, R. (2019). *El Cuerpo de Conocimientos de la Gestión de Proyectos*. Lima: Ediciones Centenario.
- Asmat Córdova, M. (s. f.). *Gestión de Riesgos en Proyectos según el PMBOK: Lineamientos Generales para su implementación*. SlideShare. Recuperado 2021, de <https://es.slideshare.net/ManuelAsmatCrdovaMBA/gestin-de-riesgos-en-proyectos-segn-el-pmbok-lineamientos-generales-para-su-implementacin>
- *Authenticating Your Client*. (s. f.). Google Developers. Recuperado 22 de agosto de 2021, de <https://developers.google.com/android/guides/client-auth?hl=es>
- Clark, J. (s. f.). *Backend para una aplicación Flutter*. Back4app. <https://blog.back4app.com/es/backend-para-aplicacion-flutter>
- *Configurar un Editor*. (s. f.). Flutter. <https://esflutter.dev/docs/get-started/editor?tab=androidstudio>
- Diví, V. (2020, 30 octubre). *¿Qué es el lenguaje de programación Dart?* inLab FIB. Recuperado 25 de mayo de 2021, de <https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart>
- Fernández, Y. (2019, 30 octubre). *Qué es Github y qué es lo que le ofrece a los desarrolladores*. Xataka. Recuperado 7 de septiembre de 2021, de <https://www.xataka.com/basics/que-github-que-le-ofrece-a-desarrolladores>

- *Firebase: qué es, para qué sirve, funcionalidades y ventajas.* (2020, 17 mayo). DIGITAL55. Recuperado 25 de mayo de 2021, de <https://digital55.com/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- *Firebase's Realtime Database in a Flutter Application.* (s. f.). Engineering Education (EngEd) Program. Recuperado 2021, de <https://www.section.io/engineering-education/implementing-firebase-in-flutter/>
- *Fundamentos de la gestión de proyectos | Guía de gestión de proyectos.* (s. f.). Wrike. <https://www.wrike.com/es/project-management-guide/fundamentos-de-la-gestion-de-proyectos/#que-es-un-proyecto>
- Guía del PMBOK 6° Edición. (2017). *Guía PMBOK*. Nueva York: PMBOK.
- Gómez, J. (2016, 2 febrero). *¿Qué es una matriz RACI y cómo usarla?* El Laboratorio de las TI. Recuperado 2021, de <https://www.laboratorioti.com/2016/02/22/ticcionario-una-matriz-raci-usarla/>
- Gutiérrez Martínez, J., & Estrada, A. F. (2018). Las tecnologías disruptivas y su aplicación en la medicina con vistas al 2030. *Revista Cubana de Salud Pública*, 50(4), 55-68.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación*. México: Mc Graw Hill.
- Hernández Velasquez, L. P. (2020). *Seguridad de la información en dispositivos smartwatch*. Obtenido de <https://repository.ucatolica.edu.co/bitstream/10983/24872/4/Seguridad%20de%20la%20informaci%C3%B3n%20en%20dispositivos%20Smartwatch.pdf>
- Hito Master S.L. (s. f.). *La Gestión del Tiempo del Proyecto*. MDAP. Recuperado 2021, de <https://uv-mdap.com/programa-desarrollado/bloque-i-el-ciclo-de-vida-del-proyecto/modulo-3-planificacion-del-proyecto/gestion-del-tiempo-del-proyecto/>

- INDEC (Instituto Nacional de Estadística y Censos). (2012). *cuesta Nacional sobre Calidad de Vida de Adultos Mayores ENCaViAM - 1a ed.* (N.º 46). <https://www.indec.gob.ar/ftp/cuadros/sociedad/encaviam.pdf>
- Llamas, L. (2020, 12 abril). *Pulsímetro y oxímetro con Arduino y MAX30102*. Luis Llamas. Recuperado 19 de septiembre de 2021, de <https://www.luisllamas.es/pulsimetro-y-oximetro-con-arduino-y-max30102/>
- Maxim Integrated. (2018, octubre). *Datasheets MAX30102*. <https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf>
- Menzinsky, A., López, G., & Palacio, J. (2016, julio). *Scrum Manager*. Iubaris Info 4 Media SL. https://www.scrummanager.net/files/sm_proyecto.pdf
- Mobizt, M. (s. f.-a). *Firestore Arduino Client Library for ESP8266 and ESP32*. GitHub. Recuperado 2021, de <https://github.com/mobizt/Firebase-ESP-Client>
- Mobizt, M. (s. f.-b). *Firestore RTDB Arduino Library for ESP8266*. GitHub. Recuperado 2021, de <https://github.com/mobizt/Firebase-ESP8266>
- Mobizt, M. (s. f.-c). *JSON parser and builder for ESP8266*. GitHub. Recuperado 2021, de <https://github.com/mobizt/FirebaseJson>
- Molina Tejedor, M. A. (2000, mayo). *Gestión de Recursos Humanos en Proyectos Informáticos*. <https://pmquality.files.wordpress.com/2009/08/gestion-de-rr-hh-en-proyectos-informaticos.pdf>
- OMS (Organización Mundial de la Salud). (2020, 5 febrero). *Envejecimiento*. Who.int. Recuperado 6 de abril de 2021, de https://www.who.int/es/health-topics/ageing#tab=tab_1

- OPS (Organización Panamericana de la Salud). (2020, 30 julio). *Las personas mayores de 60 años han sido las más afectadas por la COVID-19 en las Américas*. OPS/OMS | Organización Panamericana de la Salud. Recuperado 6 de abril de 2021, de <https://www.paho.org/es/noticias/30-9-2020-personas-mayores-60-anos-han-sido-mas-afectadas-por-covid-19-americas>
- Parker-Pope, T. (2021, 7 octubre). *¿Qué es un oxímetro de pulso? ¿De verdad necesito uno en casa?* The New York Times. Recuperado 25 de mayo de 2021, de <https://www.nytimes.com/es/2020/04/29/espanol/estilos-de-vida/oximetro-para-que-sirve.html>
- Pérez, A. (2022, 4 mayo). *Costos directos e indirectos de un proyecto*. OBS Business School. Recuperado 2022, de <https://www.obsbusiness.school/blog/costos-directos-e-indirectos-de-un-proyecto>
- *Resultados de la encuesta de sueldos sysarmy 2021.01*. (2021). Openqube. Recuperado 2021, de <https://sueldos.openqube.io/encuesta-sueldos-2021.01/#Salarios-Segun-Puesto>
- TISNÉS, A., & SALAZAR-ACOSTA, L. M. (2016). *Envejecimiento poblacional en Argentina: ¿qué es ser un adulto mayor en Argentina? Una aproximación desde el enfoque de la vulnerabilidad social*. Papeles de POBLACIÓN. <http://www.scielo.org.mx/pdf/pp/v22n88/1405-7425-pp-22-88-00209.pdf>

8. ANEXOS

8.1. Encuesta

Datos sociodemográficos:

1. Edad.
2. Sexo.
3. Vive solo o convive con otra persona.
4. Posee obra social además del PAMI

Datos de salud del encuestado, sobre sus conocimientos y controles en cuanto al ritmo cardíaco y sus niveles de oxígeno en sangre:

5. ¿Conoce cuáles deberían ser los valores adecuados de oxígeno en sangre?
 - Si
 - No
6. ¿Tiene conocimiento sobre sus valores de oxígeno en su sangre?
 - Si
 - No
7. ¿Realiza controles sanguíneos regularmente?
 - Si
 - No

Escala de Likert, sobre la aceptación o no del prototipo wearable:

8. ¿Cuál es su opinión acerca de un dispositivo parecido a un reloj que controle su nivel de oxígeno en sangre y su ritmo cardíaco?
 - Muy necesario
 - Necesario
 - Regular
 - Innecesario
 - Muy innecesario
9. ¿Cuál es su opinión sobre un control remoto (a distancia) sobre ciertos indicadores personales de su salud física?
 - Muy necesario

- Necesario
- Regular
- Innecesario
- Muy innecesario

10. ¿Cómo considera tener un botón que le avise a un familiar o centro de salud mediante telemedicina si tiene un problema de salud?

- Muy necesario
- Necesario
- Regular
- Innecesario
- Muy innecesario

8.2. Gestión de proyectos ¹²

La capacidad de entregar proyectos a tiempo, sin salirse del presupuesto y alineados con los objetivos empresariales, es clave para ganar ventaja en el entorno empresarial mundial altamente competitivo de hoy en día. Aquí es donde entran en juego los gestores de proyectos. Los gerentes de proyectos tienen una labor increíblemente compleja, que combina capacidades de organización, una mente analítica y habilidades interpersonales versadas.

8.2.1. ¿Qué es un proyecto?

El Project Management Institute define un «proyecto» como «un esfuerzo temporal realizado para crear un producto, servicio o resultado único».

Hay algunos aspectos clave de esta definición que han de tenerse en cuenta:

- La palabra «temporal» implica que los proyectos deben tener un principio y un final definidos. Esto significa que cada proyecto debe incluir la cronología, el alcance y los recursos. El hecho de que sea temporal con un principio y un final también implica que no forma parte de las operaciones en curso, lo cual nos lleva al segundo punto...

¹² Fundamentos de la gestión de proyectos: <https://www.wrike.com/es/project-management-guide/fundamentos-de-la-gestion-de-proyectos/#que-es-un-proyecto>

- El propósito de un proyecto debe ser «crear un producto, servicio o resultado único». Esto significa que se iniciará un proyecto para lograr un objetivo específico que generalmente queda fuera del ámbito de las actividades empresariales habituales del día a día. Esto implica que el equipo del proyecto puede estar formado por personas que no suelen trabajar juntas y que necesitan recursos normalmente ajenos al ámbito de las actividades cotidianas.

8.2.2. ¿Qué es la gestión de proyectos?

La gestión de proyectos es la práctica de aplicar conocimientos, habilidades, herramientas y técnicas para completar un proyecto de acuerdo con unos requisitos específicos. Se trata de identificar el problema, elaborar un plan para solucionarlo y, a continuación, ejecutar ese plan hasta que se resuelva el problema. Puede parecer fácil, pero hay muchos factores que intervienen en cada etapa del proceso.

8.3. Desarrollo ágil de proyectos

Basado en el documento escrito por Eugenia Bahit “Scrum & Extreme Programming”¹³, podemos decir que el desarrollo ágil de proyecto es:

El desarrollo ágil de software no es más que una metodología de gestión de proyectos adaptativa, que permite llevar a cabo, proyectos de desarrollo de software, adaptándose a los cambios y evolucionando en forma conjunta con el software.

Si bien las metodologías de gestión podrían resumirse en solo dos enfoques (o tendencias): el enfoque ágil y el enfoque predictivo (propuesto por el Project Management Institute), cada uno de estos enfoques, presenta una diversa variedad de propuestas que pueden aplicarse.

Los enfoques ágiles se diferencian (y también guardan ciertos aspectos en común) de los predictivos, básicamente por la forma de abordaje de un proyecto.

El enfoque predictivo, es aquel que plantea el abordaje estricto de un proyecto, sobre la base del cumplimiento de tres aspectos predefinidos al comienzo del proyecto: alcance, costo y tiempo; mientras tanto, el enfoque ágil, plantea los proyectos desde el cumplimiento de un objetivo más amplio: entregar software con el mayor valor posible.

¹³ Scrum & Extreme Programming – Eugenia Bahit: <http://umh2818.edu.umh.es/wp-content/uploads/sites/884/2016/02/Scrum-y-eXtrem-Programming-para-programadores.pdf>

Esto, puede explicarse si no, comentando que la principal diferencia es que el enfoque predictivo, propone la definición detallada del alcance del proyecto, y la estipulación precisa de tiempo y costo, mientras que el enfoque ágil, plantea la definición de un alcance global al comienzo, para luego ir incrementándolo en las diversas iteraciones (cada una de las cuales, supone la entrega de un software 100% funcional).

12 principios del agilismo

Principio 1: Satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Principio 2: Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Principio 3: Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible

Principio 4: Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto. Este cuarto principio, marca una diferencia radical e inigualable con las metodologías tradicionales.

Principio 5: Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

Principio 6: Conversación cara a cara.

Principio 7: El software funcionando es la medida principal de progreso.

Principio 8: Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

Principio 9: La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

Principio 10: La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Principio 11: Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.

Principio 12: A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

8.4. Matriz RACI o RASI

La matriz RACI es también conocida como matriz de responsabilidades porque sirve para establecer las responsabilidades de cada actor que participa en una tarea.

La matriz se construye con una tabla donde por filas tenemos tareas y por columnas actores.

En la intersección de cada fila con cada columna vamos colocando la responsabilidad de cada uno de estos roles: “R”, “A”, “C” o “I”.

8.4.1. El significado de las siglas RACI

Veamos ahora el significado de las siglas y aquí es importante que pongas atención en la descripción de los roles y en una palabra que puede llevar a confusión: “Responsable”.

“R” – Responsable del inglés Responsible. Es el rol encargado de realizar la tarea. Hay que tener cuidado que en este caso responsable significa responsable de realizar el trabajo no que sea el que tiene capacidad de decidir cuando está completo o no. Esta capacidad pertenece a otro rol.

“A” – Aprobador del inglés Accountable y algunas veces Approver. Es el rol que aprueba el trabajo realizado por el Responsable. Realmente sería el responsable de certificar que la tarea se ha realizado, de ahí su nombre.

“C” – Consultado del inglés Consulted. Son aquellas personas que son consultadas sobre la cuestión, personas a las cuales se les pregunta su opinión sobre algún aspecto de la tarea ya bien sea porque deben tenerse en cuenta o porque son expertos en la materia.

“I” – Informado del inglés Informed. Son aquellas personas a las que hay que mantener informadas sobre la evolución de la tarea. Lo más frecuente es informar de cuando se ha completado la tarea, pero dependiendo del rol y de la implicación pueden solicitar que se les informe de la evolución.

8.4.2. Reglas a tener en cuenta en las matrices RACI

- Al menos debe haber un Responsable dentro de una matriz RACI.
- Sí, aunque te cueste creerlo sólo puede haber un Aprobador por tarea dentro de la matriz RACI.
- Puedes incluir tantos Responsables, Informados y Consultados como necesites.

8.4.3. Variantes de la matriz RACI

La matriz RACI es la forma más común de matriz de responsabilidades, pero existen más variantes que pueden resultarte de utilidad dependiendo de la ocasión.

Las más interesantes son:

Matriz RASCI. Los roles “R”, “A”, “C” e “I” son los mismos que hemos definido más arriba y se añade uno más, “S” Soporte del inglés Support. Es un rol que da soporte para completar la tarea encomendada al Responsable. Este rol ayuda a realizar la tarea.

Matriz RASI. Siguiendo el rol introducido en el caso anterior y si nadie va a ser consultado la matriz RACI puede verse transformada en una matriz RASI, es decir habrá actores que den soporte al desarrollo de la tarea.

Matriz RACIQ. Se añade la participación de un equipo de revisión de Calidad (Q del inglés Quality) para hacer revisión de la tarea.

8.5. ¿Por qué Flutter?

En una primera instancia se había definido como framework de desarrollo de la aplicación a Ionic, luego de una base de investigaciones se decidió ir por Flutter.

Los fundamentos son los siguientes:

8.5.1. ¿Qué es Flutter?

Flutter es un framework de Google. Permite, a partir de un mismo código, el desarrollo de apps compatibles con iOS y Android a una velocidad récord.

Basándonos en nuestra experiencia, Flutter mejora el rendimiento de las aplicaciones. Reduce su tamaño y es en general más estable y produce menos errores.

8.5.2. ¿Qué es Ionic?

Ionic es un framework originalmente basado en Angular. Permite a nuestros programadores el desarrollo de apps con tecnologías web. Utiliza estándares como HTML, CSS y JavaScript. Prepara el código de una app para que pueda funcionar tanto en plataformas iOS como en Android. También, ofrece la posibilidad de realizar una compilación más y hacer una ampliación de escritorio basada en ElectronJS. Dando lugar a una Progressive Web App.

8.5.3. Flutter vs Ionic

Ionic y Flutter comparten una visión común de crear aplicaciones de alto rendimiento que funcionen en todos los dispositivos. Sin embargo, sus filosofías centrales no podrían ser más diferentes.

- El principio de Ionic es utilizar la plataforma web. Adoptando estándares abiertos siempre que sea posible.
- Al programar con Ionic se aprende sobre las herramientas y los lenguajes de la web. Es un framework diseñado para ofrecer un gran rendimiento en dispositivos móviles, equipos de escritorio y, especialmente, en la web.
- Flutter, en cambio, crea un ecosistema propio que está en desacuerdo con los lenguajes comunes, conjuntos de herramientas y estándares que se encuentran en el mundo del desarrollo en general.
- Flutter ofrece un rendimiento muy bueno en dispositivos móviles. Pero las limitaciones fundamentales de su arquitectura hacen que sea una mala elección para implementaciones basadas en web.
- La elección de la solución debe basarse en dónde y cómo se planea implementar la app y qué habilidades se conoce o se quiere aprender en el futuro.

En el desarrollo de apps ¿Flutter o Ionic?

Es importante tener en cuenta los siguientes factores:

- **Rendimiento**

Flutter podría ser una mejor opción en cuanto a rendimiento. Sin embargo, en muchos casos, Ionic da el mismo rendimiento si está escribiendo una aplicación para un consumidor más estándar o para el uso de empleados. Hay que tener en cuenta que el rendimiento casi siempre se reduce a cómo se escribe el código. No depende de la plataforma o framework que se elija.

Ionic utiliza el tiempo de ejecución y los recursos del navegador estándar. Por lo que el tamaño de la aplicación suele ser muy pequeño. Flutter requiere una gran cantidad de código para aplicaciones muy básicas, porque envía todo ese tiempo de ejecución a pesar de que el navegador ya tiene funciones disponibles para casi todo. Por lo tanto, si se está buscando alcanzar los estándares de rendimiento de Google para aplicaciones web progresivas (PWA) o ocupar un lugar destacado en las páginas de resultados del motor de búsqueda, resultará casi imposible con el rendimiento actual de Flutter en la web.

- **Portabilidad de código**

Cuando se trata de implementar una aplicación en dispositivos móviles y de escritorio, tanto Ionic como Flutter aparecen uniformemente emparejados. Con Flutter puedes crear algunas

aplicaciones de iOS y Android con un solo código. Se puede compilar la aplicación para ejecutarse de forma nativa en una serie de plataformas de escritorio.

La pregunta es si se desea implementar la aplicación a través de la web. Ya sea como una aplicación web de escritorio tradicional o como PWA. Las limitaciones inherentes de la implementación web de Flutter probablemente nunca funcionarán para aplicaciones que requieren tiempos de carga rápidos y un rendimiento ágil. Sin mencionar que su enfoque propietario limitará el número de bibliotecas web que se puede aprovechar. Dado que Ionic se basa en la web y se basa completamente en los estándares web, creemos que es justo darle a Ionic la ventaja cuando se trata de dispositivos móviles, equipos de escritorio y la web.

- **Look & feel nativo**

Aunque ninguno de los frameworks utiliza los elementos de la interfaz de usuario nativos de cada plataforma. Flutter e Ionic actualizan automáticamente el diseño de sus elementos de la interfaz de usuario para que coincida con la plataforma en la que se ejecuta la aplicación: Material Design para Android y Cupertino para iOS.

Ambas soluciones le permiten acceder a los servicios de la plataforma y las API nativas a través de una biblioteca de complementos predefinidos, con un conjunto de herramientas para crear sus propios complementos personalizados según sea necesario.

Sin embargo, debe tenerse en cuenta cómo es la implementación móvil nativa de Flutter. Si se está haciendo un trabajo nativo personalizado con Flutter, se debe aprender la manera en que Flutter trabaja con iOS y Android.

- **Conocimientos y habilidades**

Aquí es donde las distinciones entre Ionic y Flutter se hacen realmente evidentes.

Primero, a menos que seas uno de los 1.9% de los desarrolladores que ya conocen Dart, en comparación con el 70% que conoce JavaScript, tendrás que pensar que quieres aprender un nuevo idioma. JavaScript es, por supuesto, una apuesta bastante segura para cualquier desarrollador. La necesidad de los desarrolladores de aprender Dart dependerá únicamente del éxito o fracaso de Flutter como solución viable a largo plazo.

Segundo, debido a que Flutter opera en su propio ecosistema altamente personalizado. Hay que aprender la forma de Flutter de hacer las cosas. Por ejemplo, si está resolviendo problemas de interfaz de usuario. Se aprenderá y dominará el motor de renderizado personalizado de Flutter. No

las aplicaciones web en general. Cuando se interactúe con plataformas de dispositivos nativos, se debe aprender la interpretación de Android y iOS de Flutter, no Android o iOS en general. Este factor es uno de los mayores a considerar, al saltar a un silo de desarrollo que no comparte estándares y herramientas con otras plataformas.

En contraste, cuando estás creando con Ionic, no necesariamente estás aprendiendo Ionic. Estás aprendiendo cómo construir aplicaciones web en general. Se aprende a escribir con JavaScript, con CSS y a crear interfaces de alto rendimiento con componentes web basados en estos estándares.

- **Oportunidades para el futuro**

El último factor para considerar es la vida útil del proyecto, la libertad y flexibilidad que se tendrá a medida que la aplicación madure.

Con Ionic, se apuesta por el desarrollo en la web. Por lo que incluso si eliges construir en otras plataformas en el futuro, todo lo que construyas se basará en estándares web abiertos. Y, dado que Ionic se basa en componentes web, puede usarlo con cualquier framework JS. Con Ionic se tendrá la libertad de aprovechar lo que sea que traiga el mañana.

La tecnología está en constante evolución y cada vez más rápido. Por lo que es necesario seleccionar la tecnología más versátil en cada momento. Que permita crear productos digitales y soluciones reusables, modulares, que puedan acoplar y desacoplar componentes fácilmente sin tener que rehacer el producto.

8.6. Firebase cómo BaaS¹⁴

Backend como servicio (BaaS, por sus siglas en inglés) es un modelo en la nube que ayuda a los desarrolladores a automatizar las funciones de desarrollo de backend y lograr una gestión de la infraestructura de nube sin complicaciones. Un proveedor de servicios de BaaS es responsable de dirigir los servidores y realizar el mantenimiento. El BaaS ofrece a los desarrolladores varias herramientas para ayudarlos a crear código, backend y acelerar el desarrollo. Algunas de las características de BaaS incluyen API, almacenamiento de archivos, administración de datos y notificaciones push.

¹⁴ Backend para aplicación Flutter: <https://blog.back4app.com/es/backend-para-aplicacion-flutter/>

En BaaS, las API y los SDK se utilizan para conectar aplicaciones a servicios de backend en la nube. Las API de BaaS se crean primero y funcionan como la base de las plataformas iOS y Android. Los desarrolladores no tienen que comprar ningún hardware si utilizan un servicio BaaS. Se considera un modelo de servicio en la nube flexible y sin complicaciones. El BaaS ayuda a simplificar la creación de API.

8.6.1. Beneficios del backend como servicio

- **Escalable:** Optar por BaaS puede ser ventajoso, ya que ofrece gran escalabilidad a los desarrolladores. Los usuarios pueden optar fácilmente por escalar sus recursos de BaaS cuando surja la necesidad.
- **Sin complicaciones:** El BaaS es un modelo de servicio en la nube sin complicaciones, puesto que no requiere que los desarrolladores realicen ningún tipo de alojamiento o mantenimiento. El proveedor de servicios de BaaS maneja todas estas acciones.
- **Ahorro de tiempo:** Un BaaS permite un desarrollo rápido, por el hecho de que no se requiere que los usuarios dirijan y administren servidores en la nube. Los recursos fácilmente disponibles en el BaaS hacen que sea conveniente realizar acciones de desarrollo clave. El desarrollo de aplicaciones es más rápido en comparación con otros modelos de servicios en la nube.
- **Más flexible:** El BaaS también ofrece una gran flexibilidad en comparación con otros modelos de servicios en la nube. Es especialmente conveniente para el desarrollo de aplicaciones móviles y también se puede utilizar para el desarrollo web y la implementación de API.

Firebase es una plataforma BaaS popular de Google. Proporciona una amplia gama de funciones y componentes que ayudan en el desarrollo de aplicaciones web y móviles. Firebase es un programa de plataforma de base de datos NoSQL que sigue el protocolo JSON para realizar acciones de almacenamiento de datos.

8.7. Código de sensor y botón con Firebase en Arduino IDE

Este código está escrito en el lenguaje C++, Arduino IDE es meramente el framework de desarrollo.

La configuración inicial para los sensores y Firebase es la siguiente:

Ilustración 44: Configuración Inicial en Arduino IDE

```
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
#include <time.h>
#include <MAX30100.h>
#include <MAX30100_BeatDetector.h>
#include <MAX30100_PulseOximeter.h>
#include "addons/TokenHelper.h"
#include <Wire.h>
#define REPORTING_PERIOD_MS 5000

#define WIFI_SSID "xxxxxx" //Nombre de la red WIFI
#define WIFI_PASSWORD "xxxxxxxxxx"

#define PIN D5

#define API_KEY "AIzaSyC0TM3EioEpEwAxySZFtNX7I5dv36nAgdM" // API Key de Firebase
#define FIREBASE_PROJECT_ID "domohouse-d703d" // Project ID de Firebase
#define DATABASE_URL "domohouse-d703d.firebaseio.com" //RTDB URL de Firebase

// Email y password de usuario registrado en Firebase
#define USER_EMAIL "xxxxxxx@gmail.com"
#define USER_PASSWORD "xxxxxxx"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
FirebaseJson js;

unsigned long dataMillis = 0;
int count = 0;
PulseOximeter pox;
uint32_t tsLastReport = 0;
bool taskCompleted = false;
boolean lastState = false;
```

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Luego se configura el setup de la aplicación donde se inicia la conexión al WIFI, se inicia la conexión con Firebase, se realiza la configuración para obtener día y hora desde internet (la necesitamos para el historial diario), se verifica que esté funcionando el sensor y se define el tamaño máximo del buffer del ESP8266.

Ilustración 45: Configuraciones IDE - Firebase

```
void setup()
{
  Serial.begin(115200);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  // Configuración para iniciar Firebase
  config.api_key = API_KEY;
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  config.database_url = DATABASE_URL;
  config.token_status_callback = tokenStatusCallback;

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);

  // Configuración para tomar datos de la hora y el día desde internet
  configTime(0, 0, "pool.ntp.org", "time.nist.gov");
  setenv("TZ", "<-03>3", 1);

  //Verificación para ver si se inició el sensor
  if (!pox.begin()) {
    Serial.println("FAILED");
    for(;;);
  } else {
    Serial.println("SUCCESS");
  }
  pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

  #if defined(ESP8266)
    fbdo.setBSSLBufferSize(1024, 1024);
  #endif
}
```

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Luego se define el código que se va a ejecutar en Loop sobre la placa, es decir, la obtención y subida de datos del sensor y del botón. Para ello, tenemos 2 métodos primero y una función que se van a ejecutar en distintos momentos de acuerdo lo requiera la aplicación.

La función es la que permitirá obtener la fecha desde internet para poder pasarla como dato cuando se carguen los valores del sensor para el historial diario.

Ilustración 46: Código para obtener fecha y hora

```
String get_time(){
    time_t now;
    time(&now);
    char time_output[30];
    strftime(time_output, 30, "%y-%m-%dT%TZ", localtime(&now));
    return String(time_output);
}
```

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

El primer método es para subir los datos del botón y del sensor en la base de datos en tiempo real de Firebase, este código se ejecuta en loop cada 5 segundos (REPORTING_PERIOD_MS = 5000).

Ilustración 47: Código para subir datos

```
void realtimeData() {
    pox.update();
    if (millis() - tsLastReport > REPORTING_PERIOD_MS) {

        float bpm = pox.getHeartRate();
        float spO2 = pox.getSpO2();

        Serial.print("Heart rate:");
        Serial.print(pox.getHeartRate());
        Serial.print("bpm / SpO2:");
        Serial.print(pox.getSpO2());
        Serial.println("");
        tsLastReport = millis();

        if (bpm < 30 || spO2 == 0) {
            Serial.println(F("low value from sensor!"));
        } else {
            Firebase.RTDB.setFloatAsync(&fbdo, "sensor/HeartBeat", bpm);
            Firebase.RTDB.setIntAsync(&fbdo, "sensor/SpO2", spO2);
        }
    }

    boolean btnPressed = !digitalRead(D5);
    if (btnPressed == true && lastState == false)
    {
        Firebase.RTDB.setStringAsync(&fbdo, "button/status", "yes");
    }
    lastState = btnPressed;
}
```

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

El otro método es el que sube los datos del sensor cada 1 hora a la base de datos de Firebase, conocida como Firestore, esta no es volátil como la RTDB, si no, que los datos perduran en el

tiempo. En este método utilizamos la función mencionada anteriormente para obtener la fecha (get_time).

Ilustración 48: Código para subir datos II

```
void storeData() {
  pox.update();
  if (Firebase.ready() && (millis() - dataMillis > 60000 || dataMillis == 0))
  {
    float bpm = pox.getHeartRate();
    int spO2 = pox.getSpO2();

    dataMillis = millis();

    String content;

    String documentPath = "vitalSigns";
    String referencePath = "users/";

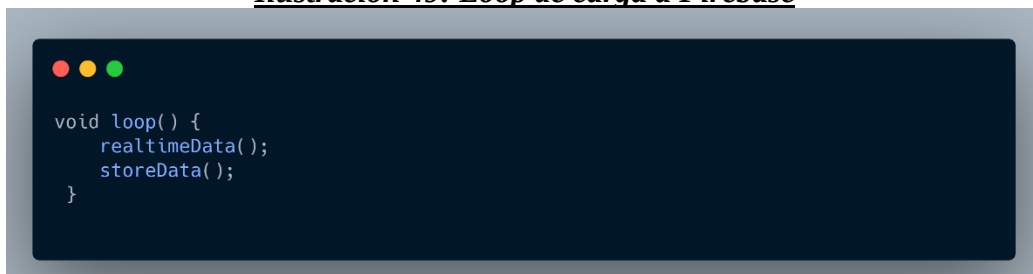
    if (pulseRate < 30 || respirationRate == 0) {
      Serial.println(F("low value from sensor!"));
    } else {
      js.set("fields/pulseRate/doubleValue", bpm);
      js.set("fields/respirationRate/integerValue", spO2);
      js.set("fields/time/timestampValue", get_time());
      js.set("fields/user/stringValue", auth.token.uid.c_str());
      js.toString(content);

      Serial.println("-----");
      Serial.println("Create a document...");

      if (Firebase.Firestore.createDocument(&fbdo, FIREBASE_PROJECT_ID, "",
      documentPath.c_str(), content.c_str()))
      {
        Serial.printf("ok\n%s\n", fbdo.payload().c_str());
      }
      else
      {
        Serial.println(fbdo.errorReason());
      }
    }
  }
}
```

Fuente: Elaboración propia. De acuerdo con el desarrollo del proyecto

Por último, como se mencionó anteriormente, estos métodos se ejecutan en loop, por ende, en el código Arduino escribimos lo siguiente:

Ilustración 49: Loop de carga a Firebase

Fuente: Elaboración propia. De acuerdo con el proyecto

8.8. Configuración para crear aplicación Flutter en Android Studio

Los pasos para comenzar a crear la aplicación se pueden visualizar en el tutorial oficial de Desarrollo con Flutter en su página¹⁵ oficial.

1. Descargar Android Studio
2. Instalar plugins de Dart y Flutter en Android Studio:

Para instalar estos:

- Inicie Android Studio.
- Abra preferencias de complementos (**Preferences>Plugins** en macOS, **File>Settings>Plugins** en Windows & Linux).
- Seleccione **Browse repositories...**, elige el complemento de Flutter y presione install.
- Presione Yes cuando aparezca para instalar el complemento de Dart.
- Presione Restart cuando aparezca.

3. Crear nuevo proyecto:

- En el IDE, haz clic **Create New Project** desde la ventana ‘Welcome’ o **File>New>Project...** desde la ventana principal del IDE.
- Selecciona **Flutter** en el menú, y clic en **Next**.
- Ingresa tu **Project name** y **Project location** deseado.
- Clic en **Finish**.

¹⁵ <https://esflutter.dev/docs/get-started/editor?tab=androidstudio> – Visitada en 22/8/2021

8.9. Creación proyecto Firebase y configuración para agregarlo a la aplicación Flutter.

Los pasos para comenzar a configurar el proyecto Firebase y agregarlo a la aplicación Flutter se encuentran también en la página oficial de Firebase¹⁶.

1. Crear un proyecto Firebase

1. En Firebase console, hacer click en **Agregar Proyecto** y seleccionar o ingresar **Nombre de proyecto**.
2. Hacer clic en **Continuar**.
3. Hacer clic en **Crear Proyecto**.

2. Registrar app con Firebase para iOS

1. En el centro de la página de descripción general del proyecto en Firebase console, hacer clic en el ícono de **iOS** para iniciar el flujo de trabajo de configuración.
2. Ingresar el ID del paquete de tu app en el campo **ID del paquete de iOS**.
3. Hacer clic en **Registrar app**.

3. Registrar app con Firebase para Android

1. En el centro de la página de descripción general del proyecto en Firebase console, hacer clic en el ícono de **Android** o en **Agregar App** para iniciar el flujo de trabajo de configuración.
2. Ingresar el nombre del paquete de tu app en el campo Nombre del paquete de **Android**.
3. Ingresar el **certificado de firma SHA-1** ¹⁷de depuración.
4. Hacer clic en **Registrar app**.

4. Agregar archivo de configuración Firebase a la aplicación para iOS

Hacer clic en **Descargar GoogleService-Info.plist** a fin de obtener el archivo de configuración de Firebase para iOS (GoogleService-Info.plist).

1. Usar Xcode para mover el archivo al directorio Runner/Runner de tu app de Flutter.

¹⁶<https://firebase.google.com/docs/flutter/setup?hl=es&platform=ios> - Visitada en 22/8/2021

¹⁷ Para obtener el certificado de firma SHA-1 debemos seguir los pasos que se encuentran en el siguiente link <https://developers.google.com/android/guides/client-auth?hl=es> – Visitando el 22/8/2021

2. Volver al flujo de trabajo de configuración de Firebase console y haz clic en **Siguiente** para omitir los pasos restantes.

5. Agregar archivo de configuración Firebase a la aplicación para iOS

1. Agregar el archivo de configuración de Firebase para Android a la app, como se indica a continuación:

- a. Haz clic en **Descargar google-services.json** a fin de obtener el archivo de configuración de Firebase para Android (google-services.json).
- b. Transfiere tu archivo de configuración al directorio del módulo (nivel de app) de tu app.

2. Agregar reglas para incluir el complemento de servicios de Google al archivo Gradle (build.gradle) de nivel de raíz (a nivel de proyecto). Además, revisa que tengas el repositorio Maven de Google.

Ilustración 50: Archivo Gradle a nivel proyecto

```
buildscript {  
  
    repositories {  
        // Check that you have the following line (if not, add it):  
        google() // Google's Maven repository  
    }  
  
    dependencies {  
        // ...  
  
        // Add the following line:  
        classpath 'com.google.gms:google-services:4.3.8' // Google Services plugin  
    }  
}
```

Fuente: firebase.google.com

3. En el archivo Gradle (generalmente app/build.gradle) de tu módulo (a nivel de app), aplica el complemento Gradle de los servicios de Google:

Ilustración 51: Archivo Gradle a nivel app

```
apply plugin: 'com.android.application'
// Add the following line:
apply plugin: 'com.google.gms.google-services' // Google Services plugin

android {
    // ...
}
```

Fuente: firebase.google.com

7.10. Buenas prácticas para manejo de repositorios Github**1. Probar los cambios antes de hacer commit**

Siempre es bueno probar los cambios antes de enviarlos al repositorio local. Las pruebas permiten identificar errores y fallas más rápido antes de que vayan al servidor de producción.

2. Usar diferentes ramas

Las ramas pueden permitirte tener múltiples versiones del código en un solo repositorio.

3. Realizar commits frecuentemente

Un commit temprano ayuda a reducir el riesgo de conflictos entre dos cambios simultáneos. Además, tener puntos de control periódicos significa que se puede comprender cuando algo dejó de funcionar. Se puede hacer commit localmente “todo el tiempo” y push cuando todo esté funcionando.

4. Escribir mensajes de confirmación útiles

Hay dos partes importantes de un mensaje de confirmación. El mensaje de confirmación debe ser sencillo. Y debería ser significativo.

Crear mensajes de confirmación útiles es una de las mejores cosas que se puede hacer. Es bueno para los demás, como visitantes, desarrolladores, reclutadores, clientes y también para uno mismo.