



Facultad 4

Módulo virtual médico para el monitoreo de las variables fisiológicas a través de un asistente personal

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Arlín Verdecia Castillo

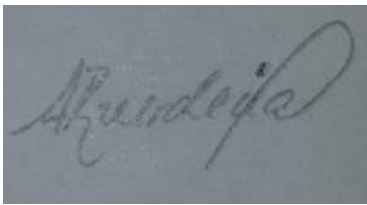
Tutor: Ing. David Pino González

La Habana, diciembre de 2022

“Año 64 del triunfo de la Revolución”

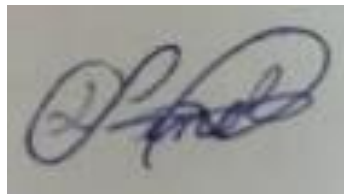
DECLARACIÓN DE AUTORÍA

Declaro ser la autora del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo. Para que así conste firmo la presente el día 1 del mes de diciembre del año 2022.



Firma del Autor

Arlín Verdecia Castillo



Firma del Tutor

Ing. David Pino Gonzáles

DATOS DE CONTACTO

Ing. David Pino Gonzáles :Graduado de nivel: Universitario

Especialidad: Ingeniería en Ciencias Informáticas

Septiembre 2016 – noviembre 2019 fecha: Universidad de las Ciencias Informáticas. Facultad 4. Centro de Informática Industrial (CEDIN). Especialista B en Ciencias Informáticas.

Noviembre 2019 – Febrero 2021: Universidad de las Ciencias Informáticas. Facultad 4. Docente José Antonio Echeverría. Centro de Tecnologías Interactivas. Especialista B en Ciencias Informáticas. Asesor de Seguridad Informática.

Febrero 2021 – hasta la fecha: Universidad de las Ciencias Informáticas. Facultad 4. Docente José Antonio Echeverría. Centro de Tecnologías Interactivas. Especialista A en Ciencias Informáticas. Jefe de proyecto del Asistente de voz personal.

Desde la fecha en que comenzó a trabajar en el CEDIN actual Centro de Tecnologías Interactivas se desempeña en el rol de desarrollador participando así en el desarrollo, despliegue y mantenimiento de diferentes sistemas informáticos.

DEDICATORIA

Esta tesis se la dedico con amor y cariño a mis a mis queridos padres por ser mi motor impulsor en mi vida, por inculcarme una buena educación y valores desde niña, por sus oraciones y bendiciones todos los días. Espero siempre contar con su apoyo y su amor, los amo mucho.

AGRADECIMIENTOS

Agradezco a Dios por guiarme durante el proceso de mi carrera y cumplir mi objetivo y el deseo de mis padres que es el de convertirme en ingeniera.

A mis padres adorados por su apoyo, consejos, y motivarme a seguir mis estudios cuando en algún momento sentía ganas de abandonarlos.

A mis hermanas bellas, Annelis y Annaliet, por su amor y sus consejos para seguir superándome cada día.

A mi pareja por estar conmigo y apoyarme, darme ánimo para seguir adelante. ¿Qué bendición que te hayas cruzado en mi camino?

A todas mis amistades de la universidad que se convirtieron en mi familia durante todos estos años. Siempre los llevaré en mi corazón.

A los profesores que de una forma u otra han intervenido en mi formación profesional y personal.

RESUMEN

El desarrollo de aplicaciones móviles en la salud ofrece la oportunidad de recopilar una cantidad ingente de información sobre el estado de salud de las personas y que puede ser utilizada, no solo para mejorar, sus resultados de salud sino también en la prevención e investigación de enfermedades. Una de las maneras actuales para llevar el control de las variables fisiológicas desde el hogar se hacen en forma de nota, redactando el comportamiento de las mismas para luego ingresar dicha información en historias clínicas electrónicas correspondiente a cada paciente, siendo un proceso muy denso donde se pueden cometer ciertos errores. Para esto se requiere de un proceso de automatización de las variables fisiológicas en el hogar, basado una aplicación móvil que permita el monitoreo de las mismas, estas variables pueden ser Temperatura Corporal, Ritmo cardíaco y Tensión arterial. Para guiar el desarrollo de esta aplicación se utilizó la metodología AUP variación UCI enmarcada en el escenario 4 de Historias de Usuario, generando los artefactos que define cada una de sus fases. Se empleó Dart v2.17.0 como lenguaje de programación para la creación de la aplicación y Java v8.0 para la API; Visual Paradigm en su v17.0 y Flutter v3.0 como herramienta para modelar diagramas y entorno de desarrollo respectivamente. Como resultado, los usuarios podrán contar con un APK que mediante el consumo de servicios sea capaz de intercambio información, además. Se validó el funcionamiento de la aplicación a través de pruebas de software detectando y corrigiendo las no conformidades existentes.

Palabras Claves

APK, Monitorear, Servicio

TABLA DE CONTENIDOS

Índice

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Telemedicina y su influencia en el mundo de las TIC.....	4
1.1.1 Áreas de aplicación de la telemedicina.....	5
1.1.2 Asistente de salud en el hogar.....	6
1.2 Dispositivo <i>wearable</i>	7
1.2.1 Dispositivo <i>wearable para medir la temperatura corporal</i>	8
1.2.2 Dispositivo <i>wearable para medir tensión arterial y frecuencia cardíaca</i>	9
1.3 Dispositivo móvil.....	10
1.4 Aplicaciones móviles.....	11
1.5 Estudio de aplicaciones homólogas para el monitoreo y control de las variables fisiológicas.....	12
1.6 Herramientas y tecnologías utilizadas.....	13
1.7 Metodología de desarrollo.....	16
1.8 Conclusiones del capítulo.....	18
CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN.....	19
2.1 Descripción del contexto del negocio de la propuesta de solución.....	19
2.2 Modelo de dominio.....	19
2.3 Análisis de la Solución Propuesta.....	20
2.4 Levantamiento de Requisitos Funcionales.....	20
2.4.1 Requisitos Funcionales.....	21
2.4.2 Requisitos no funcionales.....	22
2.5 Historias de usuario.....	23
2.6 Análisis y diseño.....	25
2.6.1 Patrón de Diseño.....	25
2.7 Diseño arquitectónico.....	26
2.8 Diagrama de Despliegue.....	27
2.9 Conclusiones del capítulo.....	28
CAPÍTULO III: IMPLEMENTACIÓN Y DESARROLLO.....	29
3.1 Modelo de implementación.....	29
3.1.1 Estándares de codificación.....	29

3.2 Validación del módulo	30
3.2.2 Pruebas del software	30
3.3 Conclusiones del capítulo	32
CONCLUSIONES FINALES	33
RECOMENDACIONES	34
REFERENCIAS BIBLIOGRÁFICAS	35
ANEXO 1 HISTORIAS DE USUARIO	38
ANEXO 2: Entrevista realizada al cliente Julio Alberto Leyva Durán.	49

ÍNDICE DE TABLAS

Tabla 1. Comparación de las aplicaciones homólogas	12
Tabla 3. Listado de Requisitos no Funcionales	22
Tabla 4. HU Mostrar lectura de los signos vitales en tiempo real	23
Tabla 6. HU Registrar Usuario	38
Tabla 7. HU Editar Usuario	39
Tabla 8. HU Eliminar Usuario.....	40
Tabla 9. HU Añadir lectura	41
Tabla 10. HU Añadir Alarma	42
Tabla 11. HU Eliminar Alarma.....	43
Tabla 12. HU Activar usuario.	44
Tabla 13. HU Mostar Historial de los Signos Vitales	44
Tabla 14. HU Listar usuario.....	45
Tabla 15. HU Eliminar Historial de Lectura.....	46
Tabla 16. HU Conectar Api-Rest con el dispositivo móvil	47

ÍNDICE DE FIGURAS

Ilustración 1. Dispositivo Fitness Band KSix Thermometer HR	9
Ilustración 2. Dispositivo wearable ASUS VivoWatch BP.....	10
Ilustración 3. Modelo Conceptual	20
Ilustración 4. Arquitectura Cliente-Servidor	27
Ilustración 5. Diagrama de despliegue	28
Ilustración 6. Comportamiento de las No Conformidades	31
Ilustración 7. Comportamiento de las No Conformidades	32

INTRODUCCIÓN

Los asistentes de salud en el hogar dedican su tiempo y talento profesional al cuidado de quienes padecen de enfermedades crónicas, discapacidades graves, ancianos y cualquier persona que necesite atención continua. Además, brindan servicios básicos que incluyen desde la administración de los medicamentos, hasta llevar el control de los signos vitales. Gracias al avance tecnológico han desplazado a nuestros hogares aplicaciones que permiten mejorar la calidad vida de las personas y que ayudan optimizar la carga de trabajo de los profesionales médicos y cerrar la brecha entre pacientes y el personal médico de la salud, tanto digital como económicamente.

En Cuba a raíz de la Covid 19, el ingreso en el hogar fue una alternativa para el aislamiento de los pacientes con sintomatologías leves y comorbilidades asociadas, siendo la enfermera y médico de la familia los encargados de darle seguimiento, muchos de estos pacientes pertenecen a la tercera edad lo que conlleva a cierta vulnerabilidad, pues la mayoría padecen de enfermedades cardiovasculares como hipertensión, insuficiencia cardíaca que en su momento tienden a complicarse. En el proceso de ingreso en el hogar el médico y enfermera de la familia se encargan de llevar un control de los signos vitales de cada paciente, donde la información correspondiente a cada constante vital es anotada en un papel para luego ingresar los datos en registros clínicos electrónicos asociado a cada paciente, esto resulta ser una tarea muy densa y propensa a tener muchos errores. Precisamente el Centro de Entornos Interactivos 3D (VERTEX) de la Facultad 4 vinculados al proyecto de asistente personal cubano, encomendó desarrollar un Módulo Virtual Médico que permita monitorear las variables fisiológicas como tensión arterial, ritmo cardíaco y temperatura a través de un asistente personal.

Sobre la base de los elementos expuestos anteriormente se formula el siguiente **problema de la investigación:**

¿Cómo obtener la información del estado de salud de las personas sobre la base de monitoreo de las variables fisiológicas?

Para la realización de la investigación se define como **objeto de estudio:** el monitoreo de los signos vitales de las personas y el **campo de acción** que se encuentra enmarcado en el uso de aplicaciones informáticas para el monitoreo de las variables fisiológicas.

Para dar solución al problema antes planteado, se define como **objetivo general:** Desarrollar un módulo virtual médico para el monitoreo de las variables fisiológicas de la persona.

Para dar cumplimiento al objetivo general antes mencionado, se dará cumplimiento a los siguientes **objetivos específicos:**

1. Elaborar la fundamentación teórica sobre el proceso de monitoreo de los signos vitales.
2. Diseñar una aplicación para dispositivos móviles que permita la automatización de los signos vitales.
3. Implementar una aplicación para dispositivos móviles que permita automatizar el proceso para el monitoreo de las constantes vitales de las personas.
4. Evaluar la aplicación para dispositivos móviles que permita llevar a cabo el proceso de monitoreo de los signos vitales.

Para el desarrollo de las tareas científicas se han combinado diferentes métodos teóricos y empíricos de la investigación en la búsqueda y procesamiento de la información. Estos son:

Métodos teóricos:

- **Análítico-Sintético:** se empleó en la búsqueda de los elementos esenciales para el modelo de monitoreo de las variables fisiológicas a través de dispositivos móviles. También en el análisis de la estructura de sistema operativo Android para valorar, conocer sus particularidades y lograr comprender su funcionamiento.
- **Inductivo-Deductivo:** se aplicó para la determinación de las generalidades y se parte del análisis de casos particulares, para arribar a razonamientos que permitan la fundamentación teórica y elaboración del sistema que se desea.

Métodos empíricos:

- **Entrevistas:** Permitted realizar el levantamiento de requisitos, así como un mejor entendimiento acerca del funcionamiento de la tecnología a utilizar. Conocer cómo se realiza el proceso de monitoreo de los signos vitales, así como las tecnologías, lenguajes y plataformas que vamos a utilizar, las ventajas pueden tomar el desarrollo de la aplicación en la actualidad para el bienestar y salud de las personas.
- **Observación:** fue empleada en los distintos momentos de la investigación para la recogida de la información precisa, real y confiable que ayuden a comprender lo principal de la problemática. Al mismo tiempo esto permitió el planteamiento del problema, dando paso a enmarcar el objeto de estudio y el campo de acción, propiciando enfocar la investigación hacia lo que se necesita alcanzar y cómo alcanzarlo.

El presente trabajo se encuentra estructurado en 3 capítulos:

Capítulo 1: Fundamentación teórica: en este capítulo se abordan los elementos teóricos y metodológicos asociados a la visualización de información mediante gráficas. Se realiza un análisis del estado actual de las soluciones informáticas que se corresponden con el campo de acción. Se fundamenta la elección de la metodología de desarrollo a utilizar, teniendo en cuenta los elementos y fases que la componen. Además, se definen las herramientas y lenguajes de programación necesarios para la implementación de la propuesta de solución.

Capítulo 2: Análisis y diseño de la propuesta de solución: en este capítulo se realiza una descripción de la aplicación propuesta como solución y sus principales funcionalidades. Se argumenta el uso de los patrones de diseño y se detalla las particularidades de la misma, en correspondencia con la metodología seleccionada.

Capítulo 3: Implementación y validación de la solución: en este capítulo se describen los estándares de codificación empleados en la implementación de la propuesta de solución. Se describe la estrategia de prueba definida para validar la aplicación y se realiza un análisis de los resultados obtenidos luego de aplicadas las mismas.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se esboza el marco teórico, donde se abordan los principales conceptos y definiciones necesarios para el desarrollo de la investigación. Se describen los sistemas existentes que responden al objeto de estudio y campo de acción de la investigación, ya sea en el escenario nacional o internacional, realizando posteriormente una valoración de los mismos. Además, se explica la selección de la metodología de desarrollo de software y se fundamenta mediante un estudio los lenguajes, herramientas y tecnologías utilizadas para el desarrollo de la propuesta de solución.

1.1 Telemedicina y su influencia en el mundo de las TIC

Inicialmente, la telemedicina se desarrolló para aproximar los servicios sanitarios a la población residente en lugares remotos con escasez de recursos sanitarios y, con ello, mejorar la accesibilidad a los mismos. Posteriormente, se transformó como un medio para mejorar la calidad asistencial al permitir la formación y el apoyo a la toma de decisiones de profesionales sanitarios ubicados en zonas alejadas. Más recientemente, se está planteando como una herramienta para la mejora de la eficiencia en los servicios sanitarios, ya que permite compartir y coordinar recursos geográficamente alejados o rediseñar servicios sanitarios para optimizar recursos. Actualmente, se pueden encontrar servicios de telemedicina para la mayor parte de las especialidades.

Siguiendo la clasificación propuesta por Hersh y colaboradores en el año 2006, los servicios o programas de telemedicina se agrupan en servicios basados en imágenes almacenadas (*store and forward*) como la teleradiología, telehistopatología, teledermatología, etc.; programas y sistemas de monitorización domiciliaría (*home-based services*) y programas de asistencia especializada en tiempo real* (*hospital-based services*). No obstante, otros autores hacen otras clasificaciones, como por ejemplo la que considera el tipo de servicio que se provee, teleconsulta, telemonitorización, teleradiología, para diferentes especialidades. En general, los principales servicios de telemedicina son (MINISTERIO DE SALUD PÚBLICA Hospital Clínico Quirúrgico Hermanos Ameijeiras, 2020):

- **Servicios de asistencia remota:** pueden referirse tanto a las teleconsultas de seguimiento, diagnóstico o tratamiento a distancia del paciente, como a los servicios de telemonitorización de pacientes –a menudo crónicos–, que incluyen en muchas ocasiones registros de parámetros biológicos. Estos servicios también incluyen la comunicación electrónica entre profesionales para llevar a cabo acciones coordinadas. A menudo, dentro de los servicios de asistencia remota se realiza una distinción entre telecuidado y telemonitorización. Los servicios de telemonitorización amplían las opciones para los

pacientes y permiten una atención continua en el hogar. Son impulsados por profesionales de la salud y también contribuyen a empoderar a los ciudadanos y pacientes a tomar un papel activo en la gestión.

- de su enfermedad. Además, se reduce la duración de la estancia hospitalaria de los pacientes, se proporciona un nuevo papel para los médicos como segunda línea de soporte en unos entornos deservicio profesional multiprofesional, a menudo coordinados por profesionales de enfermería y los pacientes pueden responsabilizarse sobre su enfermedad y tomar el control sobre ella.
- **Servicios de gestión administrativa de pacientes:** estos incluyen tanto la solicitud de pruebas analíticas como aspectos relacionados con la facturación por la prestación de servicios.
- **La formación a distancia para profesionales:** que tiene como objetivo suministrar pautas y evidencias sobre salud que faciliten la educación continua de los profesionales de salud.
- **La evaluación y la investigación colaborativa en red:** el uso de las TIC para compartir y difundir buenas prácticas, así como crear conocimiento a través de las acciones y reacciones de sus miembros.

1.1.1 Áreas de aplicación de la telemedicina

Las principales áreas de aplicación de la telemedicina tienen como objetivo el cuidado del paciente en el hogar, los servicios de emergencia o bien servicios de información. Así, las teleconsultas se refieren a la utilización de recursos de telemedicina para obtener una segunda opinión de un profesional de la salud mediante el intercambio de información clínica. Por mencionar los más comunes, se utilizan recursos de telemedicina para transmitir o intercambiar imágenes de radiología diagnóstica y similares (teleradiología); también recursos relacionados con el laboratorio clínico y la gestión de los registros y la historia clínica electrónica (telepatología); el uso de recursos de telemedicina en dermatología mediante la ayuda de la videoconferencia o la transmisión de imágenes (tele dermatología); el uso de recursos para ayudar a los pacientes en psiquiatría mediante videoconferencias y chats (telepsiquiatría); o el uso de recursos de telemedicina para el tratamiento de las afecciones cardiovasculares (telecardiología); y en general en la práctica totalidad de especialidades médicas. Por otro lado, también se utilizan recursos de telemedicina combinados con recursos procedentes de la realidad virtual, la robótica y la inteligencia artificial para apoyar y supervisar los procedimientos de cirugía o incluso llevar a cabo directamente cirugía a distancia (telecirugía). La implementación de las TIC a nivel de la atención socio asistencial nos brinda aquí la posibilidad de mejorar la atención integral y seguimiento no solo de los pacientes crónicos, sino también de las enfermedades con baja

prevalencia, y facilita además su educación en medicina preventiva y salud pública la telemedicina facilita la equidad en el acceso a los servicios asistenciales independientemente de la localización geográfica; reduce los tiempos de espera (tanto en la realización del diagnóstico como en el tratamiento), evitando problemas derivados mayores; posibilita el realizar consultas remotas desde atención primaria al hospital de referencia, reduciendo el número de derivaciones; e incide en términos de formación y competencia tanto a nivel de atención primaria como hospitalario(MINISTERIO DE SALUD PÚBLICA Hospital Clínico Quirúrgico Hermanos Ameijeiras, 2020).

1.1.2 Asistente de salud en el hogar

Los asistentes de salud en el hogar ayudan a sus pacientes a vivir una vida independiente ayudándolos con las actividades diarias que van desde las necesidades de atención médica de rutina hasta las tareas personales. Además de brindar compañía y velar por el bienestar general del paciente, los asistentes de salud en el hogar son la clave esencial para mantener una calidad de vida que los pacientes no pueden disfrutar en los centros de atención fuera de su hogar. Ser asistente de salud en el hogar es un trabajo tan exigente como enriquecedor. Lo más probable es que pase la mayor parte de su tiempo con el mismo paciente y se convierta en su compañero constante mientras navega por un período desafiante en su vida. Los asistentes de salud en el hogar exitosos deben ser confiables para que sus pacientes bajen la guardia y les permitan ingresar a sus vidas. Con la confianza establecida, los asistentes de atención médica domiciliaria pueden observar con mayor facilidad y precisión la condición en evolución del paciente e informar esos cambios a la familia, así como a los médicos y enfermeras que supervisan la atención médica del paciente. Los asistentes de salud en el hogar realizan una amplia variedad de tareas todos los días. Estas tareas variarán según las necesidades del paciente. Las responsabilidades típicas del día a día de los asistentes de salud en el hogar incluyen(Jordan, 2021):

- Administrar medicamentos recetados
- Tomando y monitoreando los signos vitales
- Informar la condición del paciente al médico o enfermero
- Ayudar al paciente con ejercicios o terapia
- Preparar las comidas del paciente
- Vestir al paciente
- Bañar al paciente
- Proporcionar primeros auxilios para heridas leves

- Ayudar con el uso del baño
- Unirse a los pacientes durante las visitas al médico
- Acompañar al paciente durante los viajes fuera del hogar
- Realización de tareas de limpieza livianas

1.2 Dispositivo *wearable*

Un dispositivo *Wearable* (una tecnología que pasa desapercibida entre los accesorios o el vestuario) sirve generalmente para ayudar a mejorar la salud y el bienestar de las personas, animales o incluso las plantas, además van camino en convertirse en todo un asistente para la salud. Con un dispositivo *Wearable* podemos motivarnos más a la hora de hacer algo de ejercicio, las pulseras de actividad nos invitan a conseguir objetivos de pasos, distancia y evitar el sedentarismo. Con una pulsera de actividad podemos obligarnos a andar más diariamente, a levantarnos de vez en cuando de la silla si nuestro trabajo es sedentario, muy habitual cuando teletrabajamos, también nos permite llevar un control de la distancia recorrida a diario. Como es un dispositivo que podemos llevar cómodamente en la muñeca no nos molestará para realizar este tipo de acciones. Este dispositivo se va sincronizando con la aplicación de nuestro móvil o incluso podemos ver alguna información en la propia pantalla de la pulsera o reloj. Con un *Wearable* también puedes tener una mejor comunicación con otras personas, gracias a un *Wearable* que disponga de pantalla, sonido y/o vibración, al recibir una notificación podemos enterarnos mediante algún tipo de aviso, pudiendo así dirigirnos a nuestro teléfono móvil y atender esta comunicación que puede ser un correo electrónico, una notificación de alguna aplicación de mensajería, una llamada o incluso una videollamada. Dependiendo del tipo de *Wearable* que nos haga de dispositivo avisador de estas comunicaciones, incluso podemos contestar algunas de ellas.

Otro de los usos de los dispositivos *Wearables* es para la localización de personas, animales u objetos, así un dispositivo *Wearable* nos permite conocer la localización mediante GPS incrustado en un llavero para colocar en una mochila, en un reloj para un niño, o en el collar de nuestra mascota. Mediante una tarjeta sim con datos móviles es capaz de enviar la localización recogida por el chip GPS a un dispositivo inteligente u ordenador y traducir estas coordenadas en un mapa para en caso de extravío, saber dónde se encuentra. El uso de los dispositivos *wearables* es cada vez más popular. Su tamaño reducido y su accesibilidad contribuyen a que cada vez más personas opten por adquirir este tipo de dispositivos. Las principales ventajas que aportan los *wearables* en la salud son:

- Seguimiento y monitorización de parámetros relevantes: Los valores aportados por los dispositivos *wearables* presentan cada vez mayor fiabilidad, hasta el punto que muchos cuentan con la

aprobación de la comunidad científica. En la actualidad, estos dispositivos son prescritos o recomendados por profesionales médicos en patologías como las cardiovasculares.

- **Mejor comprensión y concienciación:** Los dispositivos wearables han facilitado una mejor comprensión de parámetros relevantes en nuestra calidad de vida, por ejemplo, la calidad del sueño.
- **Contribuyen a la figura del paciente experto:** Al disponer los usuarios de mayor información sobre su salud, este tipo de dispositivos también ha propiciado un mayor autocontrol por parte de los pacientes.
- **Promueven hábitos de vida saludable:** 8 de cada 10 usuarios de *wearables* afirman que estos dispositivos han mejorado su estilo y calidad de vida. Al estar más concienciados con aspectos como su calidad del sueño, propician las condiciones necesarias para que este sea de calidad y reparador.
- **Adherencia terapéutica:** Los pacientes que usan *wearables* presentan tasas de cumplimiento terapéutico y estilo de vida saludable más elevadas.
- Los registros que hacen estos dispositivos permiten compartir información con el médico, favoreciendo, por tanto, la comunicación médico-paciente. Según el ONTSI, 4 de cada 10 usuarios de *wearables* comparten la información con su equipo médico.
- Esa monitorización constante favorece la detección de posibles alteraciones y, por tanto, contribuyen al diagnóstico precoz de ciertas patologías.
- La facilidad de uso y la accesibilidad que proporcionan los *wearables* permiten optimizar el abordaje de pacientes crónicos o personas mayores.
- **Accesibilidad:** Los wearables se incluyen dentro de lo que se conoce como “EHealth”. Estos dispositivos, junto con otras nuevas funcionalidades como la historia clínica digital o la teleasistencia, permiten a los pacientes disponer en todo momento de todo lo referente a su estado de salud (Juan Antonio Soto, 2021).

1.2.1 Dispositivo wearable para medir la temperatura corporal

Fitness Band KSix Thermometer HR: La Fitness Band Thermometer HR incorpora un sensor de temperatura para monitorizar la temperatura corporal durante todo el día. La temperatura se muestra en la pantalla claramente pudiendo llevar un control exhaustivo de la fiebre o ausencia de ella en todo momento. Además de la temperatura, también dispone de un monitor de ritmo cardíaco dinámico de 24h que permite llevar un registro continuo de las pulsaciones durante todo el día. También puede vincularse a la app gratuita Wearfit 2.0, para sincronizar el Smartphone con la información de la actividad

física registrada por la pulsera y aprovechar aún más las posibilidades de monitorización que ofrece. La Wearfit 2.0 permite establecer metas deportivas diarias y comprobar cuánto falta para alcanzarlas. También permite registrar las horas de sueño y la calidad de las mismas, además de configurar alarmas y notificaciones.



Ilustración 1. Dispositivo Fitness Band KSix Thermometer HR

1.2.2 Dispositivo wearable para medir tensión arterial y frecuencia cardíaca

ASUS VivoWatch BP está equipado con sensores médicos ECG, PPG y tecnología ASUS HeahAI. Es un reloj que permite hacer un seguimiento de la salud, bienestar y actividad, con funciones smartwatch estándar y un formato compacto y ligero. Con la aplicación móvil ASUS HealthConnect, puedes controlar tu presión sanguínea y frecuencia cardíaca de manera sencilla y rápida. Y cuenta con 28 días de autonomía de uso normal, por lo que el VivoWatch BP se convierte en un smartwatch con tensiómetro que proporciona consejos de salud y bienestar que puede ayudarte a mejorar tu estilo de vida. Los dos sensores ECG miden las ondas de pulso para determinar tu frecuencia cardíaca. Y por su parte, el sensor óptico PPG mide el flujo sanguíneo para determinar tu pulso. A partir de esos datos, el VivoWatch BP calcula con precisión tu tiempo de tránsito de pulso (PTT) y los datos relacionados con tu salud. Entre ellos, la presión sanguínea, frecuencia cardíaca, calidad del sueño, datos de actividad y nivel de estrés (puntuación HRV). Pero para un funcionamiento adecuado, el reloj debe calibrarse con un tensiómetro de brazo. Debes tomar varias medidas e ingresarlas manualmente en la aplicación. Y como otros relojes que miden la presión arterial, estas calibraciones deben repetirse de vez en cuando.

Haciendo uso de estos dispositivos wearables encargado de velar por la salud de las personas se tomarán en cuenta variables como Temperatura, Tensión Arterial y Ritmo Cardíaco para realizar el proceso de monitoreo en tiempo real a través de una aplicación móvil.



Ilustración 2. Dispositivo wearable ASUS VivoWatch BP

1.3 Dispositivo móvil

El concepto de dispositivo móvil engloba una gran cantidad de dispositivos electrónicos de consumo. Habitualmente, por dispositivo móvil entendemos que se trata de un dispositivo que tiene la capacidad de conectarse a Internet. No obstante, en ocasiones también se tiende a englobar dentro la categoría de dispositivos móviles a cámaras digitales, videocámaras, etc., ya que a veces tienen capacidad de conectarse a otros dispositivos vía Internet o redes wifi o bluetooth. Existen diferentes tipos de dispositivos móviles, creados para cubrir diferentes funcionalidades (protecciondatos.org, 2019).

- **Teléfonos inteligentes o smartphones:** Podemos decir que son como teléfonos móviles convencionales pero contruidos sobre una estructura informática móvil, que les proporciona mayor capacidad de almacenamiento y de procesamiento. Se asemejan bastante a un ordenador. Son usados a menudo tanto para el entretenimiento como herramientas de trabajo.
- **Tabletas:** Son ordenadores portátiles personales integrados en una pantalla táctil, con acceso a Internet y capacidad para ejecutar aplicaciones instaladas sobre un sistema operativo. Su manejo es fácil e intuitivo.
- **Relojes inteligentes:** Son relojes de pulsera que poseen funcionalidades como acceso a Internet, recibir llamadas o enviar mensajes.
- **Reproductores digitales:** Son dispositivos destinados a almacenar y reproducir archivos digitales como audio o vídeo.
- **Cámaras:** Dispositivos que permiten captar imágenes, audio y videos. Además, pueden tener capacidades de conectividad, que permiten enviar y compartir los archivos capturados.
- **GPS:** Son dispositivos que permiten a los usuarios determinar la posición de forma precisa de un vehículo o una persona que use este dispositivo.

- **Ordenadores portátiles:** Son ordenadores completos que, debido a su tamaño reducido y compacto, permiten su transporte y gracias a su batería, su uso mientras no esté conectado el cargador a la red eléctrica.

1.4 Aplicaciones móviles

Los smartphones están presentes en todos los aspectos de nuestra vida cotidiana. También en temas de salud y bienestar. Y eso es, en gran parte, por la gran cantidad de apps que se desarrollan. El mercado de aplicaciones móviles de salud y bienestar ha crecido de manera constante durante los últimos años. De acuerdo con IQVIA, en 2017 había más de 318.000 aplicaciones de salud y bienestar al alcance de los usuarios y esta oferta creció exponencialmente a raíz de la pandemia mundial. Otro dato relevante: según el informe 2020 Consumer Survey on Digital Health de Accenture, el 35% de los consumidores norteamericanos utilizan este tipo de aplicaciones desde sus teléfonos móviles o tabletas. La lista de ventajas de las aplicaciones móviles de salud y bienestar se hace año tras año más larga, pero a continuación puedes encontrar 3 de las más destacadas:

- **Reducción de los costes sanitarios:** Contribuyen de muchas formas a la reducción de costes, pero la prevención es una de las más destacables. Usando este tipo de aplicaciones aumenta la probabilidad de mantener un estilo de vida saludable y, consecuentemente, prevenir muchas de las enfermedades por las que anteriormente se acudía al hospital.
- **Mayor comodidad para los pacientes:** Sin lugar a dudas, una de las grandes mejoras que ofrecen las aplicaciones móviles de salud y bienestar hasta la fecha es la facilidad de acceso a la atención médica, evitando horas y horas en salas de espera.
- **Facilidad de recopilación de datos para proveedores de atención médicas:** Se ha vuelto increíblemente fácil para los pacientes y los hospitales compartir datos con los proveedores de atención médica. Los hospitales ahora pueden compartir la información de los pacientes de forma remota. De este modo, los proveedores de servicios de salud no tienen que rellenar formularios una y otra vez.

A pesar de que todas las aplicaciones de salud y bienestar tienen en común el objetivo de ayudar a las personas a vivir una vida más saludable, podemos clasificarlas según el tipo de prestaciones que ofrecen:

- **Aplicaciones de salud y bienestar general:** Aplicaciones de seguimiento de la nutrición que nos ayudan a calcular las calorías, aplicaciones de sueño que rastrean nuestros patrones de sueño y aplicaciones de manejo del estrés que nos ayudan a detectar picos de alto estrés.

- **Aplicaciones de telemedicina:** Ofrecen una atención virtual al paciente por parte de profesionales autorizados.
- **Las aplicaciones de administración de la salud:** Esta categoría engloba todas aquellas aplicaciones que ayudan a monitorear sus condiciones de salud (como enfermedades cardíacas, diabetes, embarazo, salud mental, etc.), las que ayudan a rastrear medicamentos y las que permiten que los proveedores de atención médica puedan compartir de forma remota los registros de los médicos privados de los pacientes (Doonamis, 2021).

1.5 Estudio de aplicaciones homólogas para el monitoreo y control de las variables fisiológicas.

- **AirStrip ONE: Ayuda** a tu médico a **monitorearte** a través de un teléfono inteligente o iPad. Con esta app el médico sigue tus signos vitales, alergias, medicamentos y resultados de análisis de laboratorio.
- **iCare Health Monitor (Android) y Care Monitor de Salud (iOS)**. Completísima aplicación para medir y contabilizar múltiples constantes vitales, ayuda a valorar también la salud mental, lanza alertas médicas cuando lo considera preciso y propone ejercicios físicos y mentales. Captura los datos a través del móvil, por aportación del usuario o conectándose por bluetooth a dispositivos medidores.
- **S Healt (Android)**. Samsung desarrolló esta aplicación que, además de medir la actividad física, el sueño y el estrés, puede integrar la información procedente de otras apps y dispositivos medidores de glucosa, peso, oxígeno en sangre, presión sanguínea y otros, y establecer objetivos y retos dentro de una comunidad de usuarios.

Tabla 1. Comparación de las aplicaciones homólogas

Aplicaciones	Monitoreo de los signos vitales	Monitoreo de alarmas para los medicamentos	Registro de los datos de los signos vitales
AirStrip ONE	✗	✗	
iCare Health Monitor	✗		
S Healt (Android).	✗		

Luego de realizar un análisis en los sistemas seleccionados, observando sus características y contribuciones se puede decir que sirvieron como base para monitorear las variables fisiológicas siendo así una característica en común de todas las aplicaciones homólogas encontradas. Por otra parte, se puede decir que no cumplen en su totalidad con funcionalidades que resultan de gran importancia para el usuario, como llevar el control de medicación de los pacientes en el hogar y un registro de los variables fisiológicas. Por todo lo antes planteado se evidencia la necesidad de implementar una solución basada en la telefonía móvil que sirva como un asistente en el hogar capaz de monitorear en tiempo real las variables fisiológicas de cada paciente desde casa, guardar la información de esas variables en un historial y llevar la medicación a través del monitoreo de alarmas.

1.6 Herramientas y tecnologías utilizadas

Estas herramientas CASE modelan la información de negocios cuando ésta se transfiere entre distintas entidades organizativas en el seno de una compañía. El objetivo primordial de las herramientas de esta categoría consiste en representar objetos de datos de negocios, sus relaciones, y ayuda a comprender mejor la forma en que fluyen estos objetos de datos entre distintas zonas de negocio en el seno de la compañía. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas no satisfacen las necesidades de la organización(um, 2020). En la elaboración de los diferentes diagramas y modelos empleados en la investigación se utilizó Visual Paradigm. Para el modelado se utiliza el lenguaje Unificado de Modelado(UML) el mismo abstrae y visualiza sistemas de la programación orientada a objetos. El lenguaje de modelado es, por lo tanto, una herramienta práctica para los desarrolladores de programas y sistemas. Por un lado, permite crear fotocalcos claros para proyectos de software, y por otro, presentar sistemas de programación complejos de forma comprensible para las personas que no están familiarizadas con la temática. Por ejemplo, si deseas presentar un proyecto de software de la aplicación de la empresa al responsable de marketing, por medio de un diagrama UML puedes ofrecer una visión general de las características más importantes de la aplicación (IONOS, 2021).

Visual Studio Code v1.61.1

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación(frankier Flores, 2022).

Flutter v3.0

Flutter es un framework de código abierto, creado por Google y que permite desarrollar aplicaciones para distintos dispositivos a partir de un mismo código. Es decir, codificando nuestra aplicación una única vez, esta sería compatible para dispositivos Android e IOS e incluso para aplicaciones web o de escritorio. Es uno de los frameworks con más crecimiento para el desarrollo de apps móviles, aparte de la compatibilidad con los dispositivos móviles, permite adaptarse tanto al mundo web como al escritorio de ciertos dispositivos, siendo la adaptación completa y funcional. El SDK (kit de desarrollo de software) de Flutter se basa en el lenguaje de programación Dart, con muchas semejanzas con Java o C. Este lenguaje ha sido, también, desarrollado por Google con el fin de convertirse en un sucesor de JavaScript. Dart se ejecuta directamente en el navegador, por lo que, en un servidor, los programas Dart se pueden ejecutar de forma directa mediante el compilador Dart2js, lo que nos indica que no es la primera ni última vez que escucharemos hablar sobre este lenguaje debido a su versatilidad y potencialidad (Alfredo Barragán, 2022). Esta nueva versión trae muchas características nuevas, incluida una versión estable en Linux y macOS, versiones simplificadas de iOS, API de ciclo de vida web y mejoras de rendimiento, compatibilidad con Material 3 y mucho más. Se utiliza Dart como lenguaje de programación ya que se centra en el desarrollo de aplicaciones para el usuario y puedes utilizarlo para crear aplicaciones web y móviles. Si sabes un poco de programación, Dart es un lenguaje de programación de objetos escritos, puedes comparar la sintaxis de Dart con JavaScript.

Spring Boot v4

Spring Boot contiene una infraestructura ligera que elimina la mayor parte del trabajo de configurar las aplicaciones basadas en Spring. El objetivo de Spring Boot es proporcionar un conjunto de herramientas para construir rápidamente aplicaciones de Spring que sean fáciles de configurar. Spring Boot es una tecnología que nos permite crear aplicaciones autocontenidas, con esto nos podemos olvidar de la arquitectura y enfocarnos únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias, desplegar nuestro servicio o aplicación a un servidor de aplicaciones y enfocarnos únicamente en crear nuestro código. Para esto Spring Boot utiliza internamente un servidor de aplicaciones embebido, por defecto utiliza Tomcat, pero también lo podemos hacer con jetty o undertow. Y no solo esto, Spring Boot también nos provee un completo gestor de dependencias como maven o gradle, configuraciones automáticas y mucho más para que nuestra aplicación quede a la medida (Marcos Vázquez Gonzales, 2021). Esta herramienta se empleó para el desarrollo de la API Rest con el framework Spring, más utilizado para la plataforma Java. Además, utiliza como lenguaje de programación java en su versión v8.0 utilizado para codificar aplicaciones web. Ha sido una opción popular entre los desarrolladores durante más de dos décadas, con millones

de aplicaciones Java en uso en la actualidad. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo. Es un lenguaje de programación rápido, seguro y confiable para codificarlo todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del servidor (AWS, 2022). Se utiliza Java como lenguaje de programación debido a que es un lenguaje relativamente fácil de aprender y se puede hacer de manera gradual. Cuenta con una gran comunidad de desarrolladores y foros lo que posibilita la aclaración de dudas con bastante rapidez. Además, por su gran utilización en el mundo, la bibliografía crece cada día más. Es un lenguaje que permite crear sistemas multiplataforma, tiene incluida nuevas características a diferencia de versiones anteriores.

Sistemas Gestores de Base de Datos

Un sistema gestor de base de datos (SGBD) o Database Management System (DBMS) es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos. Los gestores de datos o gestores de base de datos permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones. Entre sus funciones se encuentran la de permitir a los usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la organización. Asimismo, el gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes. Además, los sistemas de gestión de base de datos pueden entenderse como una colección de datos interrelacionados, estructurados y organizados en el ecosistema formado por dicho conjunto de programas que acceden a ellos y facilitan su gestión (intelequia, 2021).

PostgreSQL

Es un sistema de base de datos relacional de alta disponibilidad. Es capaz de funcionar de manera estable en el servidor y, por lo tanto, resulta robusto, una de las principales características que buscan las empresas. Además, es consistente y tolerante a fallos. Es compatible con el modelo relacional, ya que asegura siempre su integridad referencial.

Algunas de sus principales características son:

- **Alta concurrencia:** Es capaz de atender a muchos clientes al mismo tiempo y entregar la misma información de sus tablas, sin bloqueos.
- **Soporte para múltiples tipos de datos de manera nativa:** Ofrece los tipos de datos habituales en los sistemas gestores, pero además muchos otros que no están disponibles en otros competidores, como direcciones IP, direcciones MAC, Arrays, números decimales con precisión configurable, figuras geométricas, etc.
- **Soporte a triggers:** Permite definir eventos y generar acciones cuando estos se disparan.

- **Trabajo con vistas:** Esto quiere decir que pueden consultar los datos de manera diferente al modo en el que se almacenan.
- **Objeto-relacional:** Otra de sus principales características, que permite trabajar con sus datos como si fueran objetos y ofrece mecanismos de la orientación a objetos, como herencia de tablas.
- **Soporte para bases de datos distribuidas:** Donde el trabajo con transacciones asegura que estas tendrán éxito cuando han podido realizarse en todos los sistemas involucrados.
- **Soporte para gran cantidad de lenguajes:** PostgreSQL es capaz de trabajar con funciones internas, que se ejecutan en el servidor, escritas en diversos lenguajes como C, C++, Java, PHP, Python o Ruby. Además, ofrece interfaces para ODBC y JDBC, así como interfaces de programación para infinidad de lenguajes de programación.

Todas las anteriores características y muchas otras convierten a PostgreSQL en una elección ideal para la mayoría de proyectos, en los que su funcionalidad, la seguridad o la integridad referencial nos resultan de gran importancia (Arsys, 2018). Como herramienta para administrar la base de datos contamos con pgAdmin 4, nos permite acceder a todas las funcionalidades de la base de datos.

1.7 Metodología de desarrollo

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de informático. Suele estar documentada y promovida por algún tipo de organización ya sea esta pública o privada. Tienen como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto, exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se clasifican en dos clases: las metodologías tradicionales o robustas y las ágiles o ligeras (Presman, 2010).

Las metodologías se pueden clasificar en dos grupos. Las metodologías tradicionales, que buscan siempre una fuerte planificación y documentación durante todo el desarrollo, y las metodologías ágiles, en las que se enfoca al desarrollo de software el cual es incremental, cooperativo, sencillo y adaptado. Para el desarrollo de la solución se utilizará la metodología definida por el centro Vertex de la Universidad para el desarrollo de proyectos. La misma consiste en una versión desarrollada por la UCI basada en la metodología de desarrollo de software AUP.

Metodología de desarrollo de software AUP versión UCI

La metodología utilizada en la universidad es una variante de AUP (*Agile Unified Process, Proceso Ágil Unificado*). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software, usando técnicas ágiles. Se adapta al ciclo de vida definido para la actividad productiva de la institución, logra estandarizar los productos de trabajo elaborados en los diferentes centros de desarrollo de la universidad.

AUP UCI establece tres fases (Sánchez, Tamara Rodríguez, 2016):

- **Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- **Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

En la presente investigación se empleó en la fase de Ejecución, pues la misma propicia que se ejecuten las actividades necesarias para desarrollar el software, de los 4 escenarios que propone la metodología para la descripción de requisitos se emplea el cuarto, ya que es factible aplicarlo a proyectos bien definidos; también se ha de tener en cuenta que el cliente siempre acompaña al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

1.8 Conclusiones del capítulo

- La descripción de los principales conceptos permitió adquirir una mayor comprensión de los temas relacionados con el objeto de estudio de la presente investigación.
- El análisis de las aplicaciones descritas permitió abordar todo lo referente a las tecnologías wearables, así como su empleo en la medicina para mejorar la calidad de vida de las personas.
- Además, se evidenció que tipos de dispositivos wearables son empleados para monitorear las variables fisiológicas y como se conectan a través de dispositivos móviles para sincronizar en tiempo real el estado de las variables mediante una aplicación.
- A partir del estudio del arte se evidenció la necesidad de implementar una aplicación que contribuya a la automatización de las variables fisiológicas en el hogar, variables como tensión arterial, ritmo cardíaco y temperatura corporal.
- El análisis de diferentes herramientas, lenguajes y tecnologías permitió alcanzar los conocimientos necesarios para seleccionar las adecuadas para el desarrollo de la solución. Se selecciona como metodología a utilizar AUP-UCI en su escenario número 4.

CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En el presente capítulo se muestra el modelo de dominio para representar la estructura de la solución, así como la relación entre las diferentes clases con el objetivo de vincular los conceptos asociados a la investigación. Se realiza el levantamiento y especificación de requisitos, tanto funcionales como no funcionales necesarios a cumplir para la elaboración de la solución. Se formula la propuesta de solución, así como la arquitectura y los patrones empleados en el desarrollo de la aplicación.

2.1 Descripción del contexto del negocio de la propuesta de solución

La descripción del contexto del negocio presentada en el epígrafe se realizó utilizando un modelo conceptual. El mismo le permitió a la autora de la investigación comprender el funcionamiento del dominio del negocio informatizado.

2.2 Modelo de dominio

Consiste en uno o más diagramas de clase UML que muestran (prezi, 2022):

1. Los conceptos básicos del dominio del problema.
2. Las propiedades más importantes del problema.
3. Las relaciones importantes entre dichos conceptos.

Es una herramienta eficaz, para que el usuario valide nuestro conocimiento del problema, es por eso que los elementos de notación deben de ser sencillos y fácil de comprender. El propósito de la perspectiva conceptual es (prezi, 2022):

- Reflejar lo más fielmente posible del problema:
- Sus conceptos, propiedades de los conceptos, relaciones entre los conceptos y no se trata de una solución. Los aspectos como eficiencia, eficacia, economía de conceptos están fuera de lugar.
- El modelo del dominio que a continuación se describe constituye una representación de gráfica de los principales conceptos que forman parte del módulo médico para un asistente personal.

A continuación, se esclarecen los conceptos asociados al dominio, definiendo los términos más importantes que serán de utilidad para la posterior confección del modelo de dominio.

Persona: La persona es la que desencadena el proceso de chequeo de los signos vitales y la que interactúa con la aplicación.

Dispositivo *Wearable*: como su nombre lo indica es un dispositivo que lo lleva la persona, a través de él, se obtienen los signos vitales de la persona.

Signos vitales: Los signos vitales reflejan funciones esenciales del cuerpo de la persona, donde se llevará a cabo un monitoreo de las mismas.

Aplicación: La aplicación móvil se encargará de visualizar los signos vitales.

Dispositivo móvil: Dispositivo en el cual va estar conectado por wifi a el dispositivo móvil e instalada la aplicación para visualizar los signos vitales.

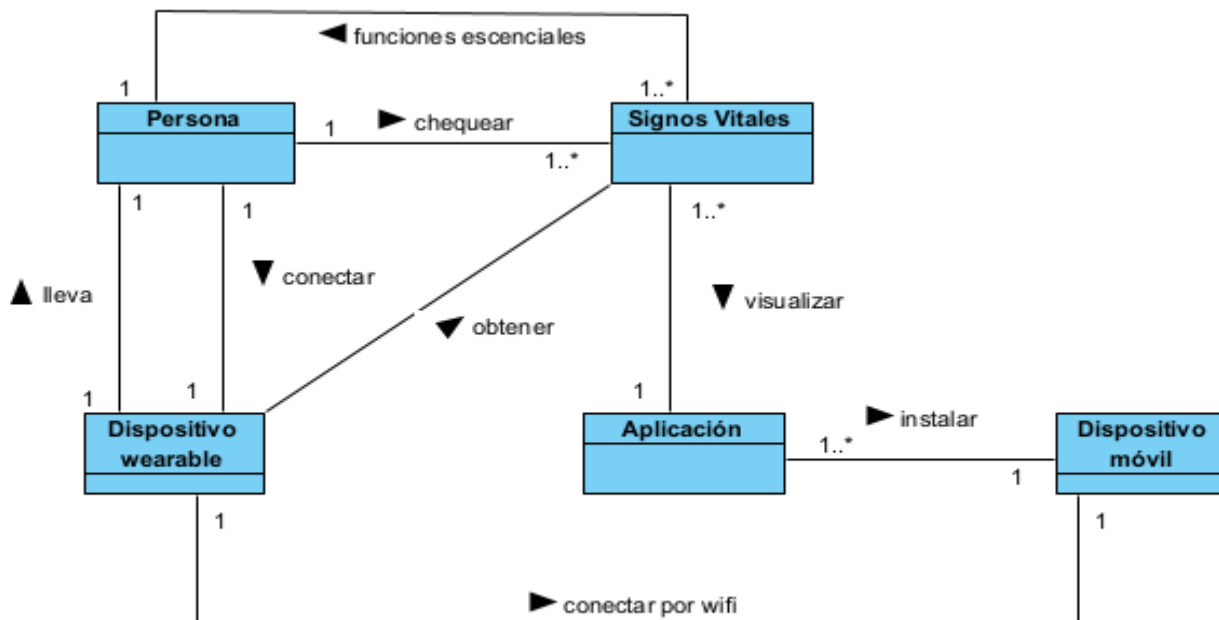


Ilustración 3. Modelo Conceptual

2.3 Análisis de la Solución Propuesta

La propuesta de solución consiste en la creación de una aplicación móvil que monitoree la información de las variables fisiológicas de la persona. Dicha información será recopilada en una base de datos y el usuario permitirá consultar y recibir datos de estas variables fisiológicas en tiempo real a través de la API-Rest.

2.4 Levantamiento de Requisitos Funcionales

El flujo de trabajo Requerimientos, constituye uno de los flujos más importante que propone la metodología AUP-UCI, pues en el mismo se capturan los requisitos del sistema. Los mismos tienen como objetivo fundamental guiar el desarrollo hacia un sistema correcto.

Técnicas de identificación de requisitos

Entrevistas (pmoinformatica, 2016).

- Se realizan con los usuarios o interesados clave.
- Direccionan al usuario hacia aspectos específicos del requerimiento a levantar.
- Son útiles para obtener y documentar información detallada sobre los requerimientos y sus niveles de granularidad.
- Pueden ser entrevistas formales o informales.
- Una clave es mantenerse enfocado en los objetivos de la entrevista.
- Las preguntas abiertas son útiles para identificar información faltante.
- Las preguntas cerradas son útiles para confirmar y validar información.
- El éxito de las entrevistas depende del grado de conocimiento del entrevistador y entrevistado, disposición del entrevistado de suministrar información, buena documentación de la discusión y en definitiva de una buena relación entre las partes.

Observación (pmoinformatica, 2016)

- Consiste en estudiar el entorno de trabajo de los usuarios, clientes e interesados de proyecto (Stakeholders).
- Es una técnica útil cuando se está documentando la situación actual de procesos de negocio.
- Puede ser de dos tipos, pasiva o activa.
- En observación pasiva, el observador no hace preguntas, limitándose solo a tomar notas y a no interferir en el desempeño normal de las operaciones.
- En observación activa, el observador puede conversar con el usuario.

2.4.1 Requisitos Funcionales

Los requerimientos funcionales son las descripciones explícitas del comportamiento que debe tener una solución de software y que información debe manejar (pmoinformatica, 2018).

Tabla 2. Requisitos Funcionales

Código	Nombre del requisito	Descripción del requisito
RF1	Registrar usuario	La aplicación permite registrar el usuario

RF2	Editar usuario	La aplicación permite al usuario editar el nombre del usuario
RF3	Eliminar usuario	La aplicación permite eliminar el usuario
RF4	Añadir lectura	La aplicación permite registrar los signos vitales(tensión arterial, ritmo cardíaco, temperatura corporal) para realizar su lectura
RF5	Añadir alarma	La aplicación permite al usuario añadir la alarma para la toma de cualquier medicamento
RF6	Eliminar Alarma	La aplicación permite al usuario eliminarla alarma
RF7	Mostar en un historial la lectura de los signos vitales.	La aplicación permite visualizar a través de un historial la lectura de los signos vitales
RF8	Monitorear los signos vitales en tiempo real	La aplicación permite monitorear los signos vitales en tiempo real
RF9	Activar usuario	La aplicación seleccionar a un usuario y este y lo cambia a modo activo
RF10	Listar usuarios	La aplicación permite listar todos los usuarios que se han registrado
RF11	Eliminar historial de lectura	Permite eliminar la(s) lectura(s) del historial
RF12	Conectar aplicación móvil con la Api-Rest	Permite conectarla API-Rest con la aplicación móvil

2.4.2 Requisitos no funcionales

Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. En palabras más sencillas, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace (Blog, 2018).

Tabla 2. Listado de Requisitos no Funcionales

Número	Requisito	Descripción
--------	-----------	-------------

RnF 1	Requisitos de Software	El dispositivo móvil debe tener sistema operativo Android 6.0 o superior para poder utilizar la aplicación móvil
RnF 2	Requisito de diseño e implementación	Como lenguaje de programación para el desarrollo de la aplicación se deberá utilizar en Dart
RnF 3	Requisito de usabilidad	La aplicación debe adaptarse a diferentes resoluciones de pantalla
RnF 4	Apariencia o interfaz externa	El diseño de las interfaces será sencillo, con pocas imágenes y colores y la información aparecerá correctamente organizada de forma tal que el usuario, pueda encontrar lo que busca rápidamente
RnF 5	Requisito de Integridad de datos	La aplicación debe garantizar la persistencia de los datos insertados por el usuario

2.5 Historias de usuario

Una historia de usuario es una explicación informal de una función de software, escrita desde la perspectiva del usuario final. Estas historias deben escribirse utilizando un lenguaje no técnico para brindar contexto al equipo de desarrollo (Asana, 2022).

Tabla 3.HU Mostrar lectura de los signos vitales en tiempo real

Historias de Usuarios	
Número:	HU_8
Nombre historia:	Monitorear los signos vitales en tiempo real
Prioridad en negocio:	Alta

Descripción:

1-Objetivo: Monitorear en tiempo real los signos vitales de la persona tales como:

- Tensión Arterial
- Ritmo Cardíaco
- Temperatura

2- Acciones para lograr el objetivo (precondiciones y datos):

Para monitorear los signos vitales de la persona hay que:

- Estar registrado en la aplicación.
- El usuario debe llevar el dispositivo *wearable* encargado de medir los signos vitales.
- La API-Rest tiene que estar pública en la red poder obtener los datos de dichas variables.

3- Comportamientos válidos y no válidos (flujo central y alternos):

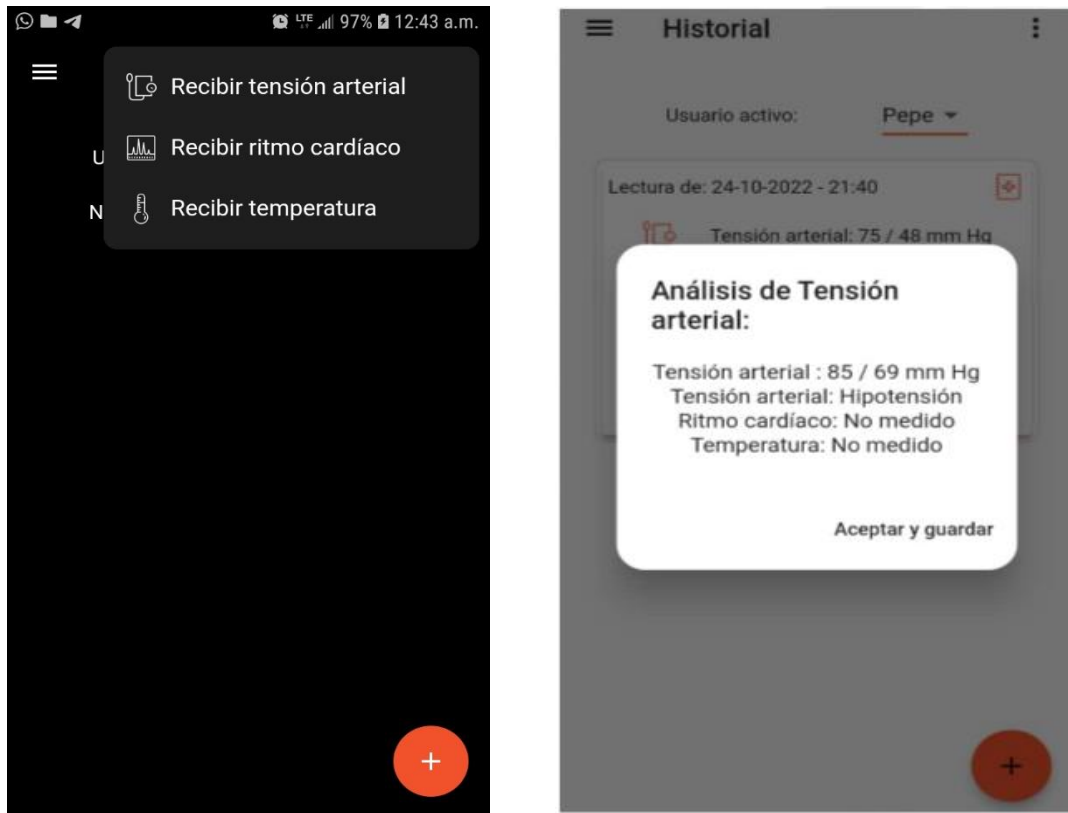
4- Flujo de la acción a realizar:

Cuando el usuario selecciona el menú de acciones que se encuentra en la parte superior derecha se muestran las siguientes opciones:

- Recibir tensión arterial
- Recibir ritmo cardíaco
- Recibir temperatura

Selecciona la constante vital que desea monitorear y luego se muestra el un análisis de lectura asociada a la constante vital que fue seleccionada, seguidamente el usuario selecciona aceptar y guardar dicha información e un historial. Ej. Si se desea conocer la presión arterial.

Prototipo:



2.6 Análisis y diseño

La parte del análisis y diseño, te brinda la opción refinar los requisitos funcionales para ayudar e comprenderlos mejor de una forma más precisa y a la estructuración del sistema.

2.6.1 Patrón de Diseño

Los patrones de diseño son programas reutilizables orientados a objetos para los desafíos más comunes en el desarrollo de software. Son respuestas probadas en el tiempo a problemas de diseño recurrentes. Describen tanto la descripción de una solución como su uso para abordar un problema concreto (Silk, 2022).

Patrones GRASP

Estos patrones GRASP significan según sus siglas en inglés General Responsibility Assignment Software Patterns, es decir, patrones generales de asignación de responsabilidades Patrones GRASP (JnjSite, 2022). En el desarrollo de la aplicación se utilizarán los siguientes patrones GRASP:

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Este patrón se puede observar en la **AddLecturaCubit** que es la encargada de controlar en la aplicación todas las lecturas de los signos vitales.

Controlador: Representa al sistema completo y es el manejador artificial de los eventos. Consiste en la asignación de responsabilidades para controlar el flujo de los eventos que genera el sistema a clases específicas. Su uso se evidencia en la clase **Cubit**, pues en esta se manejan la mayoría de los eventos realizados por la aplicación.

Patrones GOF

Dentro de los patrones clásicos tenemos los GoF (Gang of Four), estudiados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en su mítico libro Design Patterns se contemplan 3 tipos de patrones (Gracia, 2013):

- **Patrones de creación:** tratan de la inicialización y configuración de clases y objetos
- **Patrones estructurales:** Tratan de desacoplar interfaz e implementación de clases y objetos.
- **Patrones de comportamiento** tratan de las interacciones dinámicas entre sociedades de clases y objetos.

En el desarrollo de la aplicación se hizo uso del **Patrón de diseño iterador de categoría comportamiento:**

- Este patrón pertenece a la categoría de comportamiento y permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos. Este patrón se utiliza en la mayoría de las clases a la hora de iterar sobre listas y arreglos personalizados.

2.7 Diseño arquitectónico

En el desarrollo de la solución propuesta se utilizó el patrón arquitectónico cliente-servidor como se muestra en la **ilustración 4**. Su principal objetivo es procesar la información de un modo distribuido, se encarga de repartir las tareas entre proveedores de recursos o servicios llamados servidores y los

demandantes llamados clientes. Para esto se cuenta con una aplicación desde la parte del cliente conectado a un servidor de base de datos.

Para conectarse directamente al servidor se utiliza una API-Rest como interfaz de comunicación a través de solicitudes HTTP. Esta interfaz se desarrolla con una arquitectura n-capas llamadas modelos y controladores respectivamente. Por otro lado desde la parte del cliente, se cuenta con una aplicación desarrollada con flutter que hace uso de una arquitectura n-capas, la capa domain donde se encuentra la lógica de la aplicación y la lógica del negocio, la capa presentación que permitirá encargada de mostrar la información al usuario y la capa data capa que incluye todas las dependencias de los eventos y de persistencia de datos. Y por último la capa repositorio la que implementará los repositorios y las fuentes de datos (data sources).

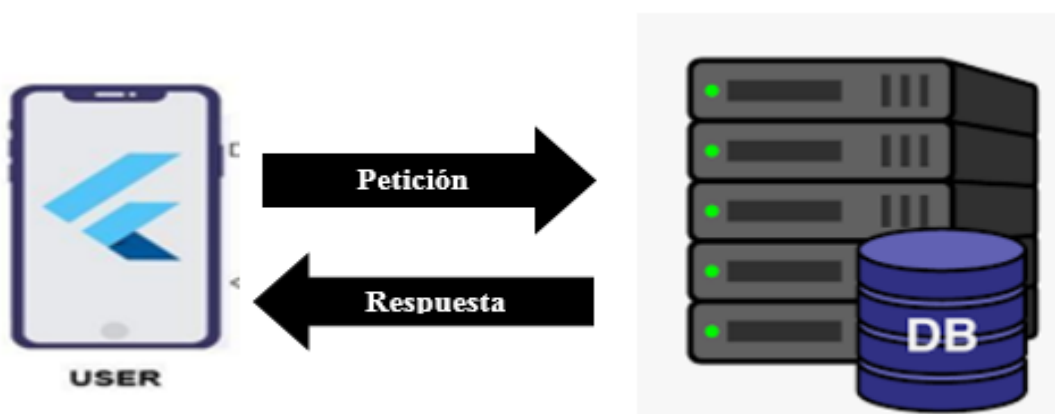


Ilustración 4.Arquitectura Cliente-Servidor

2.8 Diagrama de Despliegue

Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta (Siriwardhana, 2020).

A continuación, se presenta el diagrama que representa la distribución física del sistema a través de nodos, el mismo está compuesto por un dispositivo *wearable* en cargado de medir las variables fisiológicas de la persona, este dispositivo se conectará mediante el protocolo TCP/IP a la capa de

servicios para enviar en tiempo real las variables fisiológicas .Luego la capa de servicios se comunicará con el dispositivo móvil que tiene la aplicación instalada donde el usuario permitirá consultar el estado de las variables fisiológica a través del protocolo de comunicación HTTP (Protocolo de Transferencia de Hipertexto, del inglés Hypertext Transfer Protocol) .

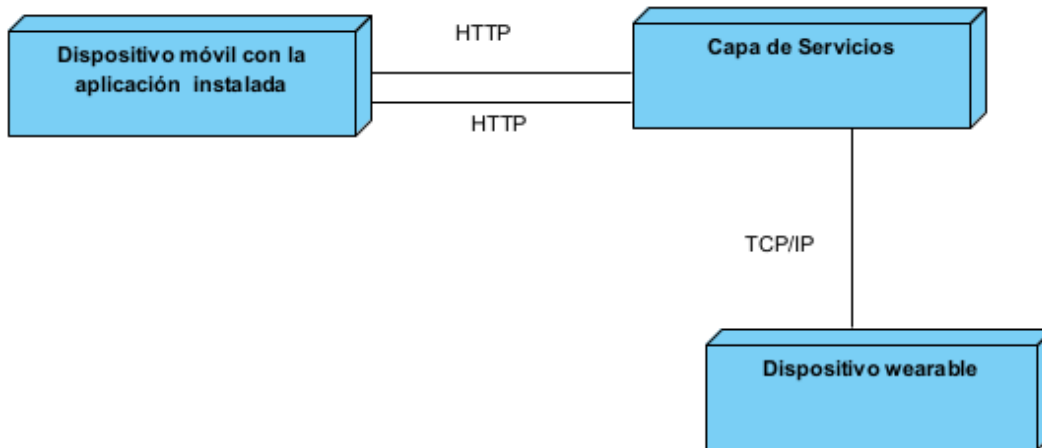


Ilustración 5. Diagrama de despliegue

2.9 Conclusiones del capítulo

Luego de haber concluido el presente capítulo se arribó a las siguientes conclusiones:

- La realización del modelo de dominio permitió representar y comprender los conceptos más significativos que intervienen en el desarrollo de la aplicación.
- La identificación de los requisitos funcionales y no funcionales y la elaboración de las correspondientes historias de usuarios propiciaron las funcionalidades que brindará la aplicación y la guía para la implementación de las mismas.
- Se seleccionó el patrón arquitectónico y los patrones de diseño con el objetivo de facilitar las tareas de implementación.
- Finalmente se realizó el diagrama de despliegue con el propósito de conocer la representación física que va a tomar nuestro sistema.

CAPÍTULO III: IMPLEMENTACIÓN Y DESARROLLO

En el presente capítulo se describen los diferentes mecanismos utilizados para llevar a cabo el desarrollo y validación del sistema propuesto. Así como los estándares de codificación que debe seguir el equipo de desarrollo para implementar el software para un mejor entendimiento y organización del mismo. Luego se realiza la fase de prueba, cuyo objetivo es comprobar si el sistema cumple con los requerimientos.

3.1 Modelo de implementación

El modelo de implementación garantiza la implementación de la propuesta de solución, basándose en su arquitectura, sus componentes, de acuerdo a su mecanismo de estructuración, y los lenguajes de programación viables.

3.1.1 Estándares de codificación

Los estándares de codificación incorporan principios de ingeniería sólidos para la programación en sus respectivos lenguajes y forman la base de cualquier enfoque preventivo. El costo de un buen software es menor que el costo de un software malo (Hamilton, 2020).

Seguidamente se especifican los estándares de codificación a utilizar en la construcción de la solución:

- Los nombres de las clases y las funciones adoptarán la notación CamelCase y no se utilizará el guion bajo como delimitador entre palabras.

```
int tensionArterialMin = 0;
```

- Número de declaraciones por línea: se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

```
• int tensionArterialMin = 0;  
• int tensionArterialMax = 0;  
• int ritmoCardiaco = 0;  
• int temperatura = 0;
```

- Las sentencias pertenecientes a un bloque de código estarán ubicadas a un nivel más a la derecha con respecto a la sentencia que las contiene. El carácter inicio de bloque "{" debe situarse al final de la línea que inicia el bloque y "}" debe situarse en una nueva línea tras la última línea del bloque y alineada con respecto al primer carácter de dicho bloque. Todas las sentencias de un bloque deben encerrarse entre llaves "{...}", aunque el bloque contenga solo

una sentencia. Esta práctica permite añadir código sin cometer errores accidentalmente al olvidar añadir las llaves.

3.2 Validación del módulo

En la ingeniería de software la validación es el proceso de revisión que verifica que el sistema de software producido cumple con las especificaciones y que logra su cometido, se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales.

3.2.2 Pruebas del software

Las pruebas de software son una parte integral del ciclo de vida del desarrollo de software (SDLC). Las pruebas de software se pueden dividir en dos tipos diferentes: pruebas funcionales y no funcionales. Diferentes aspectos de una aplicación de software requieren diferentes tipos de pruebas, como pruebas de rendimiento, pruebas de escalabilidad, pruebas de integración, pruebas unitarias y muchos más. Cada uno de estos tipos de pruebas de software ofrece una excelente visibilidad de la aplicación, desde el código hasta la experiencia del usuario (Lee, 2020).

La estrategia a seguir para la realización de las pruebas a la aplicación móvil implementada, contemplan dos niveles de pruebas: pruebas unitarias y aceptación.

Pruebas Unitarias

Las pruebas unitarias de *software*, conocidas también como unit testing o test unitarios, pueden definirse como un mecanismo de comprobación del funcionamiento de las unidades de menor tamaño de un programa o aplicación en específico (KeepCoding, 2022).

Las pruebas de unidad siempre están orientadas a caja blanca y le permiten al programador conocer si determinada funcionalidad se puede agregar al sistema sin afectar su funcionamiento. Facilitan que el programador cambie el código para mejorar su estructura (refactorización), puesto que permiten hacer pruebas sobre los cambios y asegurarse de que no han introducido errores.

Resultado de las pruebas unitarias

Para la realización de las pruebas unitarias es utilizado el marco de prueba JUnit para verificar una parte del código de la API-Rest con el objetivo de obtener y corregir los errores existentes antes de realizar la entrega al cliente. Para probar el correcto funcionamiento del código de la API se realizaron tres iteraciones de pruebas. En la tabla que a continuación se presenta, se muestran los resultados

obtenidos en cada iteración de prueba a una porción del código de la API-Rest y así como la corrección de cada uno de los errores.

Tabla 5. Cantidad de no conformidad por cada iteración

No conformidades	Primera iteración	Segunda iteración	Tercera iteración
Detectadas	2	1	0
Resueltas	2	1	0
Pendientes	0	0	0

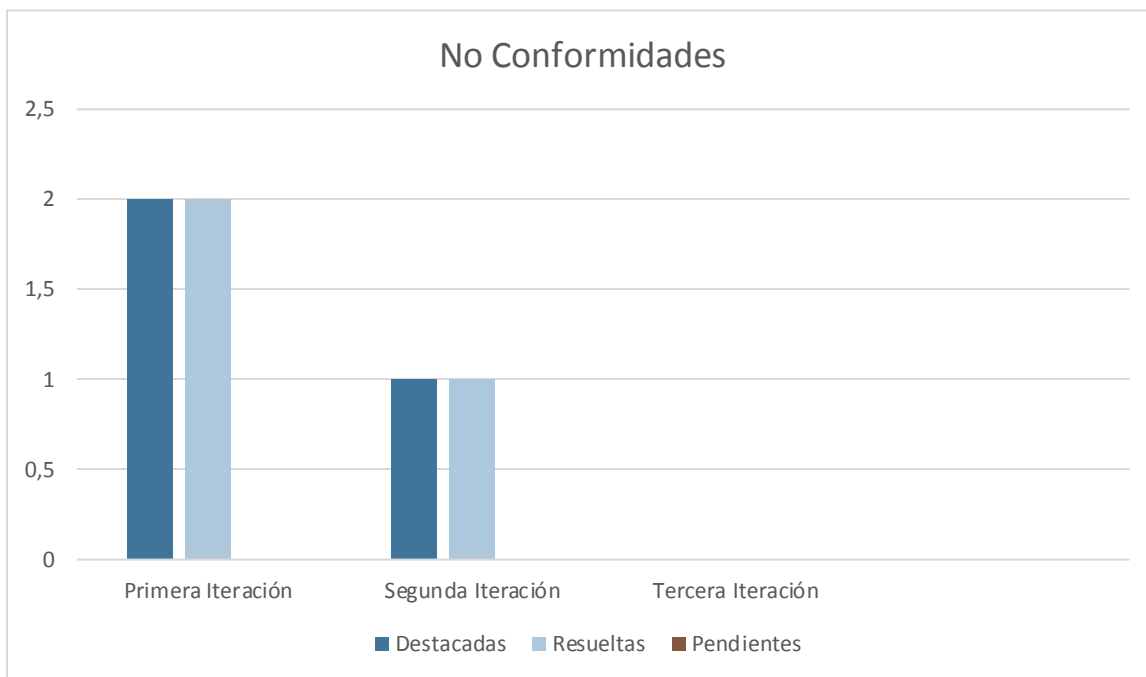


Ilustración 6. Comportamiento de las No Conformidades

Prueba de Aceptación

La prueba de aceptación es la fase más importante de la prueba, ya que decide si el cliente aprueba o no la aplicación o el software. Puede involucrar la funcionalidad, la usabilidad, el rendimiento y la interfaz de usuario de la aplicación (Fabian, 2022).

Resultados de las pruebas

Con el objetivo de comprobar el cumplimiento de los requisitos propuestos se hace uso de las pruebas de caja negra teniendo en cuenta las pruebas de aceptación detectando y corrigiendo algunas no conformidades. Para ello se efectuaron tres iteraciones, en las cuales se detectaron 9 no conformidades;

de estas 3 se clasificaron de complejidad alta, 2 de complejidad media y 4 de complejidad baja.

Las no conformidades fueron clasificadas según su complejidad por lo que se definen de la siguiente forma, arrojando los resultados que se exponen:

- Alta son errores de funcionalidad
- Media son errores de interfaz
- Baja son errores de ortografía

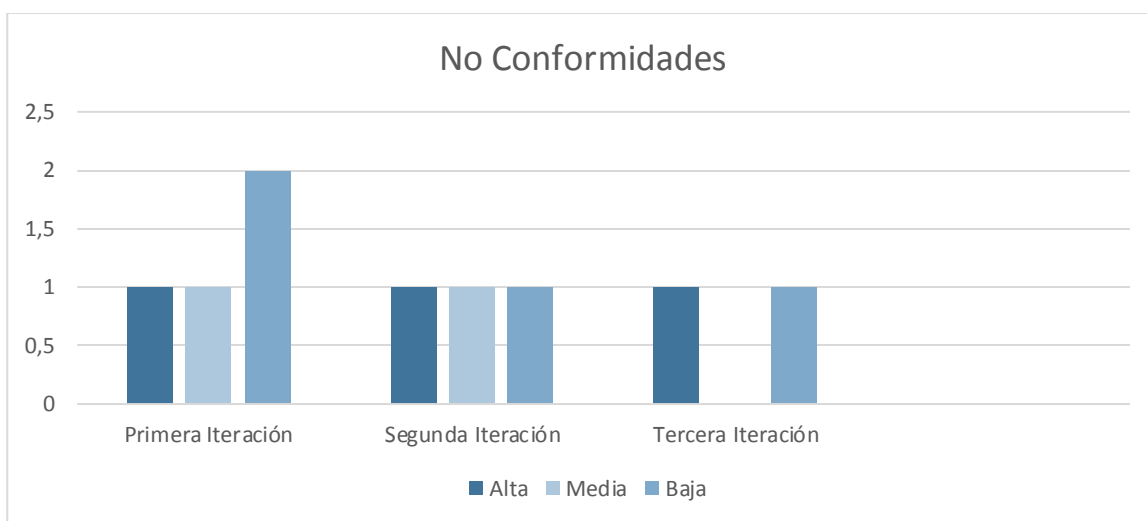


Ilustración 7. Comportamiento de las No Conformidades

3.3 Conclusiones del capítulo

Al concluir este capítulo se termina la realización de la aplicación, obteniendo los siguientes resultados:

- En el presente capítulo se especificó la utilización de estándares de código para la implementación de la propuesta de solución posibilitó adoptar una estructura homogénea que facilita la comunicación y una menor cantidad de errores; logrando un código más limpio y fácil de mantener.
- Las pruebas unitarias utilizando JUnit y la de caja negra empleado las técnicas de casos de prueba de aceptación permitieron detectar los errores presentes, corregirlos en el menor tiempo posible y entregar al cliente una aplicación con mayor calidad.

CONCLUSIONES FINALES

Luego de finalizada la investigación, se pudo arribar a las siguientes conclusiones:

- Con el resultado de la investigación se obtuvo un módulo para obtener la información de las variables fisiológicas de la persona, logrando, además, automatizar el proceso de obtención de dichas variables.
- El uso de la metodología AUP-UCI permitió generar los artefactos de la fase de análisis y diseño que sirvieron como guía del proceso de desarrollo de software.
- La realización de las pruebas unitarias y de caja negra para las pruebas de aceptación permitieron detectar los errores presentes, corregirlos en el menor tiempo posible y entregar al cliente una aplicación con mayor calidad.

RECOMENDACIONES

- Crear el asistente personal virtual.
- Integrar el módulo virtual médico para monitorear las variables fisiológicas al asistente personal cuando sea creado.

REFERENCIAS BIBLIOGRÁFICAS

Alfredo Barragán. (2022, junio 17). *Flutter, SDK para controlar los Sistemas Operativos móviles.*

OpenWebinars.net. <https://openwebinars.net/blog/flutter-un-sdk-para-controlar-todos-los-sistemas-operativos-moviles/>

Arsys. (2018, junio 13). *¿Qué es PostgreSQL y por qué llevarlo a Cloud? | Arsys. Blog de arsys.es.*

<https://www.arsys.es/blog/postgresql-servidores>

Asana. (2022). *Historias de usuario: 3 ejemplos para generar valor para el usuario • Asana. Asana.*

<https://asana.com/es/resources/user-stories>

AWS. (2022). *¿Qué es Java? - Guía de Java empresarial para principiantes - AWS. Amazon Web*

Services, Inc. <https://aws.amazon.com/es/what-is/java/>

Blog, R. (2018, noviembre 29). *Requerimientos Funcionales y No Funcionales, ejemplos y tips.*

Medium. <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>

Doonamis. (2021, noviembre 10). *Apps de salud y bienestar: ¿Cómo mejoran nuestras vidas?*

Doonamis. <https://www.doonamis.es/aplicaciones-moviles-de-salud-y-bienestar-como-mejoran-nuestras-vidas/>

Fabian. (2022, enero 17). *Definición de «Pruebas de aceptación». ElAdminis.*

<https://eladminis.com/definicion-de-pruebas-de-aceptacion/>

frankier Flores. (2022, julio 22). *Qué es Visual Studio Code y qué ventajas ofrece. OpenWebinars.net.*

<https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

Gracia, L. (2013, enero 2). *Un poco de Patrones de Diseño GoF (Gang of Four). Un poco de Java.*

<https://unpocodejava.com/2013/01/02/un-poco-de-patrones-de-diseno-gof-gang-of-four/>

Hamilton, D. (2020, abril 24). *Estándares de codificación de software y pautas de programación.*

Parasoft. <https://es.parasoft.com/blog/an-ounce-of-prevention-software-safety-security-through-coding-standards/>

- INISTERIO DE SALUD PÚBLICA Hospital Clínico Quirúrgico Hermanos Ameijeiras. (2020).
MINISTERIO DE SALUD PÚBLICA Hospital Clínico Quirúrgico Hermanos Ameijeiras Normas y procedimientos para la Telemedicina.
<http://webcache.googleusercontent.com/search?q=cache:N4oYrBjVnhAJ:convencionhha2020.sld.cu/index.php/IMHHHA/im2020/paper/download/665/297&cd=2&hl=es-419&ct=clnk&gl=cu>
- intelequia. (2021). *Gestor de Base de datos: Qué es, Funcionalidades y Ejemplos.*
<https://intelequia.com/blog/post/2949/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos>
- IONOS. (2021). *6 herramientas UML para cualquier ocasión.* IONOS Digital Guide.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/las-mejores-herramientas-uml/>
- JnjSite. (2022, febrero 3). *Los patrones del diseño software GRASP • JnjSite.com.* JnjSite.com.
<https://jnjsite.com/los-patrones-del-diseno-software-grasp/>
- Jordan, A. (2021, agosto 4). *Cómo convertirse en asistente de salud en el hogar» Wiki Útil Brookline College.* Brookline College. <https://www.brooklinecollege.edu/es/blog/how-to-become-a-home-health-aide/>
- Juan Antonio Soto. (2021). *¿Qué es un Wearable y para qué sirve? - Definición.* GEEKNETIC.
<https://www.geeknetic.es/Wearable/que-es-y-para-que-sirve>
- KeepCoding, R. (2022, agosto 1). *¿Qué son las pruebas unitarias de software?*
<https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>
- Lee, G. (2020, octubre 16). *Tipos de pruebas de software: Diferencias y ejemplos.* LoadView.
<https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>
- Marcos Vazquez Gonzales. (2021, marzo 17). *¿Qué es Spring Boot?* Codmind.
<https://blog.codmind.com/que-es-spring-boot/>

pmoinformatica. (2016). *7 Técnicas de levantamiento de requerimientos software—La Oficina de Proyectos de Informática*. <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>

pmoinformatica. (2018). *¿Qué es un requerimiento funcional? - La Oficina de Proyectos de Informática*. <http://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>

Presman, R. S. (2010). *Ingeniería del Software*.

prezi. (2022). *Diagrama de Modelo del Dominio*. [prezi.com. https://prezi.com/zjx8wzhs59kj/diagrama-de-modelo-del-dominio/](https://prezi.com/zjx8wzhs59kj/diagrama-de-modelo-del-dominio/)

protecciondatos.org. (2019, abril 4). Tipos de dispositivos móviles. *Protección de datos*. <https://www.protecciondatos.org/tipos-de-dispositivos-moviles/>

Sánchez, Tamara Rodríguez. (2016). *Metodología de desarrollo para la Actividad productiva de la UCI*.

Silk, J. (2022, marzo 31). Por qué los patrones de diseño de desarrollo de software son importantes para su empresa. *Startechup Inc*. <https://www.startechup.com/es/blog/software-development-design-patterns/>

Siriwardhana, S. (2020, octubre 26). Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue. *Blog de Creately*. <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>

um. (2020). *Herramientas CASE. Ingeniería del software. Informática Aplicada a la gestión Pública. Universidad de Murcia*. https://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html

Victor Diví. (2020, mayo 26). *¿Qué es el lenguaje de programación Dart?* inLab FIB. <https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart>

ANEXO 1 HISTORIAS DE USUARIO

Tabla 4. HU Registrar Usuario

Historias de Usuarios	
Número:	HU_1
Nombre historia:	Registrar usuario
Prioridad en negocio:	Media
Descripción:	<p>1-Objetivo: Permitir que la aplicación deje registrar un usuario ingresando su nombre.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): El usuario tiene que dirigirse a la parte de "secciones" que se encuentra al seleccionar el menú de opciones que aparece en la izquierda superior de la aplicación.</p> <p>4- Flujo de la acción a realizar: Luego de haber seleccionado la opción llamada "usuarios" procede a registrarse tocando el símbolo "más" que aparece en la parte inferior derecha y procede a ingresar su nombre. Acto seguido selecciona "añadir" y se muestra el siguiente mensaje. "Se ha añadido correctamente" en caso contrario selecciona Cancelar".</p>
Prototipo:	

Tabla 5. HU Editar Usuario

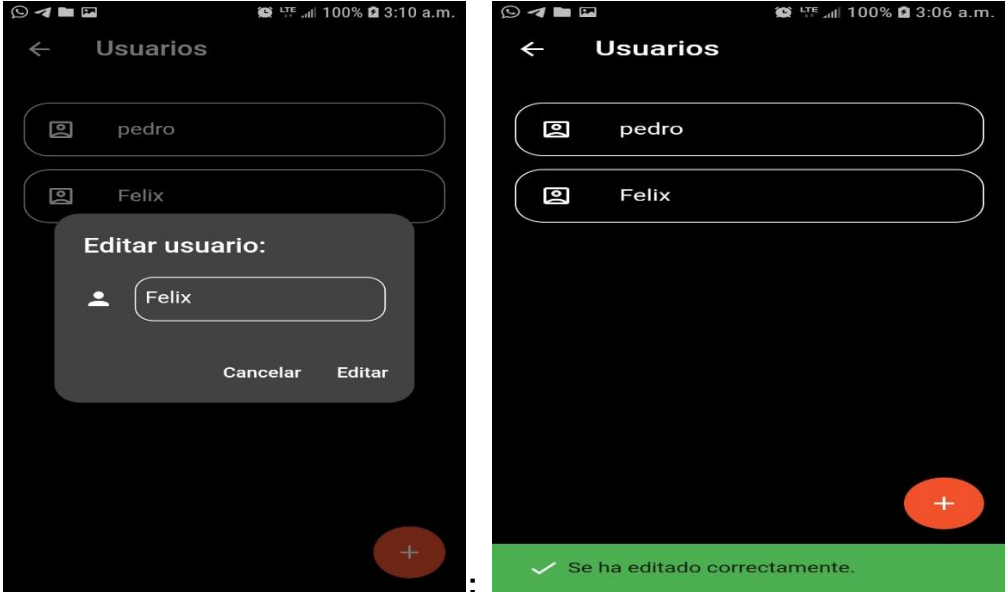
Historias de Usuarios	
Número: HU_2	
Nombre historia: Editar Usuario	
Prioridad en negocio: Media	
Descripción: 1-Objetivo: Permitir editar el nombre del usuario. 2- Acciones para lograr el objetivo (precondiciones y datos): Para editar el nombre del usuario inicialmente debe estar registrado en la aplicación: 3- Comportamientos válidos y no válidos (flujo central y alternos): 4- Flujo de la acción a realizar: Se selecciona el usuario que se dese editar, se hace todas las modificaciones y selecciona "editar". Luego de esto se muestra un mensaje "Se ha editado correctamente". En caso contrario presiona la opción "cancelar".	
Prototipo	

Tabla 6. HU Eliminar Usuario


Historias de Usuarios	
Número:	HU_3
Nombre historia:	Eliminar usuario
Prioridad en negocio:	Media
Descripción:	<p>1-Objetivo: Permitir eliminar el usuario.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): El usuario tiene que estar registrado en la aplicación y no puede estar activo.</p> <p>4- Flujo de la acción a realizar: Se debe seleccionar el usuario que se desea eliminar y acto seguido se muestra un cartel de Eliminar usuario y este selecciona "eliminar ", en caso contrario selecciona "cancelar" Ej. "Seguro que desea eliminar el usuario Pedro" .</p>
Prototipo:	

Tabla 7.HU Añadir lectura


Historias de Usuarios	
Número:	HU_4
Nombre historia:	Añadir lectura
Prioridad en negocio:	Media
Descripción:	<p>1-Objetivo: Permitir añadir lectura de los signos vitales, esta funcionalidad es opcional.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>El usuario tiene que estar registrado en la aplicación.</p> <p>4- Flujo de la acción a realizar:</p> <p>El usuario selecciona el botón "más" que aparece en la parte inferior derecha de la aplicación y aparecen los siguientes campos para registrar los valores correspondientes a las constantes vitales tales como:</p> <ul style="list-style-type: none">➤ Tensión Arterial➤ Ritmo Cardíaco➤ Temperatura <p>Acto seguido selecciona "Analizar" para ver el estado de su (s) contrastantes vitales. En caso contrario presiona "Cancelar".</p>
Prototipo	

Tabla 8. HU Añadir Alarma

Historias de Usuarios	
Número:	HU_5
Nombre historia:	Añadir Alarma
Prioridad en negocio:	Alta
Descripción:	<p>1-Objetivo: Permitir añadir una alarma.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>El usuario tiene que dirigirse al menú de acciones que se encuentra en la parte superior izquierda y seleccionar la opción de recordatorios de medicina.</p> <p>4- Flujo de la acción a realizar:</p> <p>El usuario:</p> <ul style="list-style-type: none">➤ Presiona el botón rojo de las alarmas que aparece en la parte inferior derecha de la aplicación.➤ Ingresa el nombre de la alarma y luego presiona “Añadir”. En caso contrario presiona el botón “Cancelar”.➤ Acto seguido de presionar la opción de “Añadir” muestra la opción de seleccionar la hora, se mostrarán las opciones de “Confirmar ” y “Cancelar” se presiona confirmar para luego determinar el/los días (s) en que se ejecutará la alarma. En el caso que no desee continuar selecciona “Cancelar” y se termina el flujo de acciones.
Prototipo:	

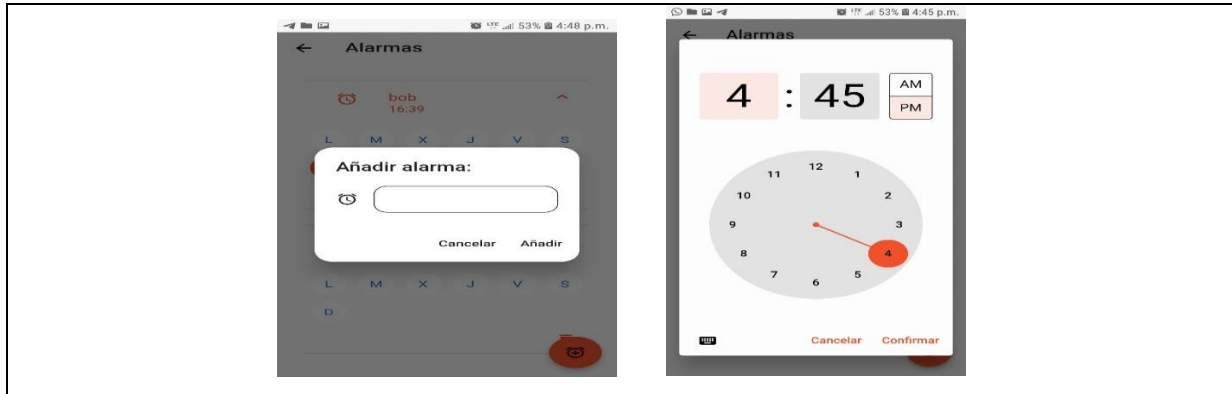


Tabla 9. HU Eliminar Alarma

Historias de Usuarios	
Número:	HU_6
Nombre historia:	Eliminar Alarma
Prioridad en negocio:	Media
Descripción:	<p>1-Objetivo: Permitir eliminar una alarma.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): El usuario tuvo que crear una alarma.</p> <p>4- Flujo de la acción a realizar: El usuario presiona el iconito de eliminar y acto seguido se muestra un cartel con el siguiente mensaje "Seguro que desea eliminar la alarma ... " además de que el usuario puede seleccionar la opción "Eliminar" y "Cancelar".</p>
Prototipo:	

Tabla 10. HU Activar usuario.


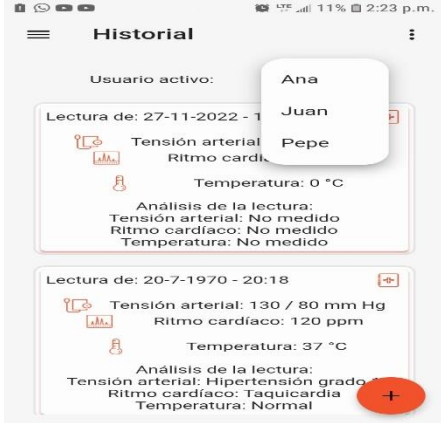
Historias de Usuarios	
Número: HU_7	
Nombre historia: Activar usuario	
Prioridad en negocio: Media	
Descripción:	
1-Objetivo: Permitir seleccionar un usuario y ponerlo a modo activo.	
2- Acciones para lograr el objetivo (precondiciones y datos):	
El usuario tiene que estar registrado en la aplicación.	
4- Flujo de la acción a realizar:	
El usuario se dirige al menú de acciones que se encuentra en la parte superior derecha y seleccionar la opción de "cambiar usuario". Acto seguido se muestran todos los usuarios que están registrados en la aplicación y se selecciona el que va estar activo.	
Prototipo:	
	

Tabla 11. HU Mostar Historial de los Signos Vitales

Historias de Usuarios	
Número: HU_9	
Nombre historia: Mostrar Historial de los Signos Vitales	

Prioridad en negocio: Media

Descripción:

1-Objetivo: Permitir visualizar el historial de las constantes vitales de los usuarios.

2- Acciones para lograr el objetivo (precondiciones y datos):

El usuario tiene que estar registrado en la aplicación.
Debe existir una lectura asociada al usuario que este activo.

4- Flujo de la acción a realizar:

Una vez el usuario se encuentre en la página principal, en un desplegable aparecen los usuarios y asociados a ellos su historial correspondiente a sus signos vitales.

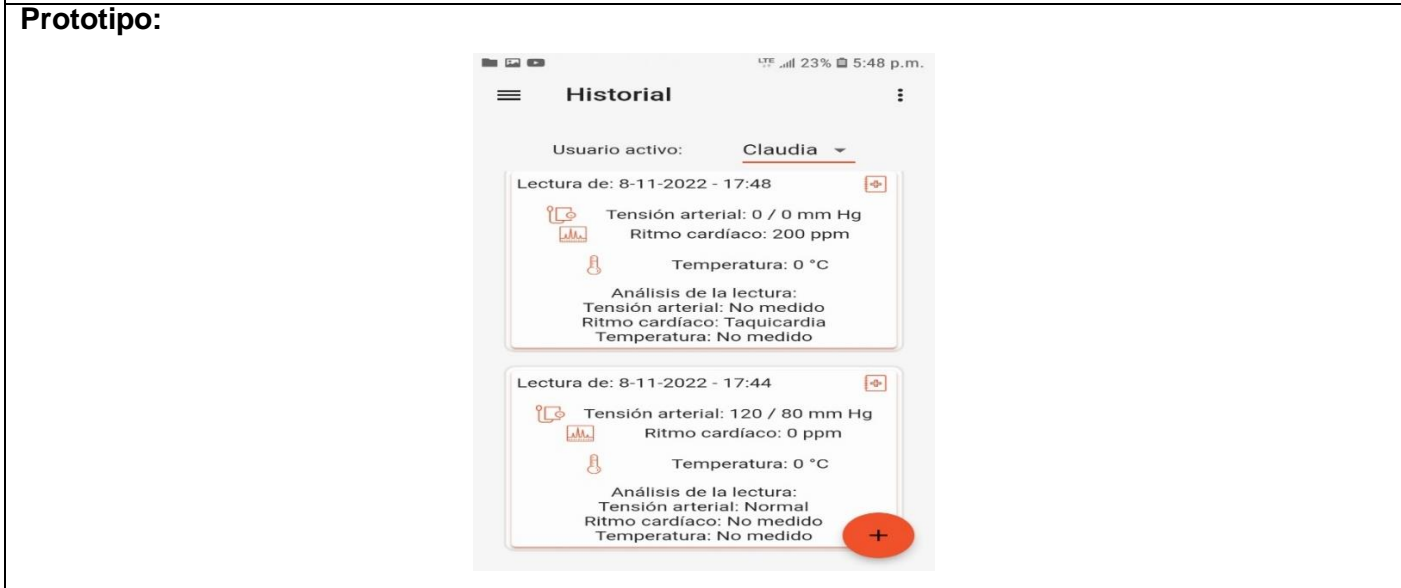


Tabla 12. HU Listar usuario

Historias de Usuarios	
Número:	HU_10
Nombre historia:	Listar Usuario
Prioridad en negocio:	Media
Descripción:	1-Objetivo: Permitir consultar los usuarios que están registrados en la aplicación.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para listar los usuarios tienen que estar registrados en la aplicación.

4- Flujo de la acción a realizar:

Cuando el usuario selecciona la acción de listar usuarios. Se muestra una lista de todos los usuarios registrados en la aplicación.

Prototipo:



Tabla 13. HU Eliminar Historial de Lectura

Historias de Usuarios	
Número:	HU_11
Nombre historia:	Eliminar Historial de lectura
Prioridad en negocio:	Media
Descripción:	<p>1-Objetivo: Permitir eliminar la(s) lectura(s) correspondiente a cada signo vital de un usuario determinado.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para eliminar la(s) lectura(s) tienen que estar las lecturas en el historial.</p> <p>4- Flujo de la acción a realizar: Cuando el usuario se encuentra en el historial, presiona la(s) lectura(s) que desea eliminar y acto seguido selecciona el iconito de eliminar que aparece en la parte derecha superior y muestra el siguiente mensaje " Seguro que desea eliminar la (s)lectura(s)" en caso de que ya no estime necesario</p>

eliminarla(s) selecciona “cancelar” y luego la “X” para desmarcar la lectura.

Prototipo:

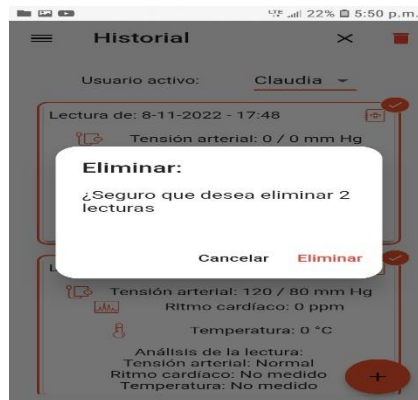


Tabla 14. HU Conectar Api-Rest con el dispositivo móvil

Historias de Usuarios	
Número:	HU_12
Nombre historia:	Conectar Api-Rest con el dispositivo móvil
Prioridad en negocio:	Alta
Descripción:	<p>1-Objetivo: Conectar la Api-Rest con el dispositivo móvil.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>La Api tiene que estar pública en la red.</p> <p>4- Flujo de la acción a realizar:</p> <p>La aplicación se conecta a la Api-Rest para solicitar y recibir estado de los signos vitales.</p>
Prototipo:	

Tabla 16. Caso de Prueba Aceptación

Caso de Prueba Aceptación
CU: HU_8
Nombre: Monitorear los signos vitales en tiempo real
Nombre de la persona que realiza la prueba: Julio Alberto Leyva Durán
Descripción de la Prueba: La aplicación permite al usuario monitoree los signos vitales(Tensión arterial temperatura, ritmo cardíaco)correspondiente a solicitud hecha por el usuario en tiempo real.
Condiciones de Ejecución: El usuario selecciona en la parte superior derecha correspondiente al menú de acciones el usuario selecciona la constante vital que desee monitorear en ese momento y se mostrará los valores y la lectura asociada al signo vital que decidió monitorear.
Entradas/ Pasos de Ejecución: Verificar que la Api-Rest este pública en la red para que la aplicación que el usuario pueda hacer la petición y recibir los signos vitales. Para lograr el proceso de monitoreo de los signos vitales el usuario debe dirigirse al menú de acciones que se encuentra en la parte superior derecha y realizar la petición.
Resultado esperado: El usuario recibió en pantalla el comportamiento de las variables fisiológicas en tiempo real.
Evaluación de la prueba: Prueba satisfactoria

ANEXO 2: Entrevista realizada al cliente Julio Alberto Leyva Durán.

Objetivo: conocer cómo se realiza el proceso de monitorear los signos vitales.

¿Considera usted necesario llevar a cabo un proceso de monitoreo y control de los signos vitales de las personas? ¿Por qué?

1. ¿Qué características tiene este proceso?
2. ¿Qué propuesta de solución sería más asequible para el proceso de monitoreo de las variables fisiológicas?
3. ¿Qué problemática existen en la vida cotidiana a la que hay que darle solución?
4. ¿Consideras que el desarrollo de una aplicación para dispositivos móviles contribuye a mejorar el proceso de monitoreo de las constantes vitales?