



UNIVERSIDAD TECNOLÓGICA ECOTEC

FACULTAD DE INGENIERÍAS, ARQUITECTURA Y CIENCIAS DE LA NATURALEZA

TÍTULO DEL TRABAJO:

DESARROLLO DE UNA APLICACIÓN PARA ALERTAS DE EMERGENCIA ANTE
CAÍDAS ENFOCADA AL FORTALECIMIENTO DE LA SEGURIDAD PERSONAL
MEDIANTE EL USO DE TECNOLOGÍAS VESTIBLES

LÍNEA DE INVESTIGACIÓN:

TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

MODALIDAD DE TITULACIÓN:

TRABAJO INTEGRADOR CURRICULAR

CARRERA:

INGENIERÍA EN SOFTWARE

TÍTULO A OBTENER:

INGENIERO EN SOFTWARE

AUTOR:

CARLOS XAVIER BERMEO RODRÍGUEZ

TUTOR:

ING. MARCOS ANTONIO ESPINOZA MINA, PHD

GUAYAQUIL – ECUADOR

2023



ANEXO No. 9

**PROCESO DE TITULACIÓN
CERTIFICADO DE APROBACIÓN DEL TUTOR**

Samborondón, 22 de agosto de 2024

Magíster
Erika Ascencio
Facultad de Ingenierías, Arquitectura y Ciencias de la Naturaleza
Universidad Tecnológica ECOTEC

De mis consideraciones:

Por medio de la presente comunico a usted que el trabajo de titulación TITULADO: DESARROLLO DE UNA APLICACIÓN PARA ALERTAS DE EMERGENCIA ANTE CAÍDAS ENFOCADA AL FORTALECIMIENTO DE LA SEGURIDAD PERSONAL MEDIANTE EL USO DE TECNOLOGÍAS VESTIBLES, fue revisado, siendo su contenido original en su totalidad, así como el cumplimiento de los requerimientos establecidos en la guía para su elaboración, por lo que se autoriza al estudiante: BERMEO RODRIGUEZ CARLOS XAVIER, para que proceda con la presentación oral del mismo.

ATENTAMENTE,



ING. MARCOS ANTONIO ESPINOZA MINA, PHD.
Tutor



ANEXO No. 10

**PROCESO DE TITULACIÓN
CERTIFICADO DEL PORCENTAJE DE COINCIDENCIAS
DEL TRABAJO DE TITULACIÓN**

Habiendo sido revisado el trabajo de titulación TITULADO: DESARROLLO DE UNA APLICACIÓN PARA ALERTAS DE EMERGENCIA ANTE CAÍDAS ENFOCADA AL FORTALECIMIENTO DE LA SEGURIDAD PERSONAL MEDIANTE EL USO DE TECNOLOGÍAS VESTIBLES elaborado por CARLOS XAVIER BERMEO RODRÍGUEZ fue remitido al sistema de coincidencias en todo su contenido el mismo que presentó un porcentaje del 1 %, mismo que cumple con el valor aceptado para su presentación que es inferior o igual al 10% sobre el total de hojas del documento. Adicional se adjunta print de pantalla de dicho resultado.

INFORME DE ANÁLISIS
registro

BERMEO-CARLOS

1%
Textos sospechosos

Similitud
De coincidencias entre oraciones
+ 1% entre los fuentes mencionadas
- 4% de palabras no reconocidas

Nombre del documento: BERMEO-CARLOS.docx ID del documento: 953857934e395e42476782ad19b377afab6876d Tamaño del documento original: 2.25 MB	Depositar: MARCOS ANTONIO ESPINOZA MINA Fecha de depósito: 5/8/2024 Tipo de carga: Interfaz Fecha de fin de análisis: 5/8/2024	Número de palabras: 14.629 Número de caracteres: 99.871
---	---	--

ATENTAMENTE,



ING. MARCOS ANTONIO ESPINOZA MINA, PH.D.
Tutor

Agradecimiento

Quisiera expresar mi profundo agradecimiento a mi familia por su constante apoyo y comprensión a lo largo de este viaje académico. Su aliento ha sido fundamental para mi desarrollo personal y profesional en la carrera que me apasiona. Con los conocimientos adquiridos durante estos estudios, aspiro a devolverles en medida todo lo que me han brindado, esperando hacerlos sentir orgullosos de los frutos de su inestimable apoyo.

A mi fiel amigo Tony, mi mascota. Desde su llegada ha sido un constante compañero a lo largo de numerosos momentos, compartiendo conmigo alegrías y consuelos. Su presencia ha sido un apoyo invaluable, ofreciéndome su leal compañía y enseñándome el valor de la amistad incondicional.

Por último, quiero agradecer sinceramente a esos amigos que fueron partícipes en la realización de este trabajo. Su apoyo, colaboración y motivación han sido esenciales para alcanzar este logro. Cada uno de ustedes ha contribuido de manera única a este proyecto, proporcionando tanto ayuda práctica como aliento en los momentos cruciales.

Índice

Resumen	10
Introducción	11
Antecedentes	11
Planteamiento Del Problema	12
Pregunta Científica	13
Objetivos	13
Objetivo General	13
Objetivos Específicos	13
Justificación	14
Marco Teórico	16
Caídas de las personas	16
Clasificación de las caídas	17
Causas de las caídas	17
Consecuencias en adultos mayores	18
Prevención de las caídas	20
Marco legal	22
Tipos de sistemas de detección de caídas	24
Sensores de ambiente	24
Sistemas de visión	24
Avances de sensores en la actualidad	25
Tecnología vestible	26
Algoritmos usados para la detección de caídas	27

	6
Basados en umbrales	27
Basados en aprendizaje automático	28
Situación actual de los algoritmos de detección de caídas	28
Sistema operativo del smartwatch - Wear os	30
Patrones de arquitectura de software - MVC	30
Herramientas de desarrollo	31
Android Studio	31
Github	31
Definición de API	32
Amazon Simple Notification Service	32
Google Maps SDK	33
Sincronización de datos en Wear OS	33
Metodología incremental	36
Etapas	36
Método De Investigación	38
Enfoque de la investigación	38
Alcance de la investigación	38
Delimitación de la investigación	39
Métodos empleados y análisis de la información	40
Elementos metodológicos específicos para TI	40
Diseño del proyecto	40
Recopilación de información	41
Desarrollo y diseño	42

	7
Diseño del sistema	42
Tecnología y herramientas	43
Análisis de datos	43
Cronograma de actividades	44
Análisis de resultados	46
Levantamiento de requerimientos	46
Diseño	47
Desarrollo	51
Algoritmo usado	51
Cambio de aceleración	53
Magnitud de aceleración	53
Cambio de aceleración vertical	54
Código del algoritmo	54
Arquitectura del sistema	56
Código del smartwatch	58
Código del smartphone	59
Envío de alertas mediante SMS	60
Obtención de la geolocalización	62
Pruebas	63
Conclusiones	68
Recomendaciones	69
Referencias	71
Anexos	74

Anexo 1.- Envío de la información entre dispositivos	74
Comparación de clientes	75
Anexo 2.- Obtener ubicación en Wear OS	75
Anexo 3.- Configurar SDK de Google	76
Anexo 4.- Historial de versiones	79
Anexo 5.- Pruebas de funcionalidad en emulador	79

Índice de Figuras

Figura 1	19
Figura 2	20
Figura 3	45
Figura 4	45
Figura 5	46
Figura 6	48
Figura 7	48
Figura 8	49
Figura 9	50
Figura 10	51
Figura 11	54
Figura 12	56
Figura 13	57
Figura 14	58
Figura 15	59
Figura 16	60
Figura 17	61
Figura 18	67
Figura 19	75
Figura 20	76
Figura 21	77

	10
Figura 22	77
Figura 23	77
Figura 24	78
Figura 25	78
Figura 26	78
Figura 27	79
Figura 28	80

Resumen

El presente trabajo de investigación tiene como objetivo abordar el problema crítico de detectar caídas en tiempo real para mejorar la seguridad de personas vulnerables, desarrollando y evaluando un sistema de detección de caídas utilizando dispositivos vestibles. Se selecciona el Galaxy Watch 6 debido a su compatibilidad y capacidad de integración con dispositivos Android, y se profundiza en el funcionamiento de sus sensores de aceleración y giroscopio para establecer umbrales de detección precisos. El diseño del sistema incluye una interfaz de usuario intuitiva en el smartwatch y una aplicación complementaria para smartphones que visualiza los datos de caídas.

Para implementar el sistema, se emplea la metodología incremental, desarrollando un algoritmo basado en umbrales que utiliza datos en tiempo real de los sensores del dispositivo. Además, se integran funcionalidades como notificaciones de emergencia y gestión de contactos de emergencia. La precisión del sistema se evalúa mediante pruebas con cuatro participantes que simulan caídas y realizan actividades normales, recopilando datos detallados para su análisis.

Los resultados de las pruebas indican una alta efectividad en la detección de caídas, demostrando una sensibilidad y especificidad adecuadas para su uso en dispositivos de propósito general. El sistema muestra una buena capacidad para distinguir entre caídas reales y actividades normales, logrando una precisión y fiabilidad que respaldan su utilidad.

Introducción

Antecedentes

En un mundo cada vez más conectado, la tecnología juega un papel muy importante en la vida cotidiana, impactando no solamente en la forma de comunicación, sino también la manera en la que se enfrentan ciertos desafíos cruciales.

Uno de los tantos desafíos actuales es el riesgo de caídas, esta preocupación global representa la segunda causa de muerte por traumatismos involuntarios. Con cifras impactantes, la OMS (2021) revela que anualmente se registran 684 000 caídas mortales y 37,3 millones de caídas que presentan la suficiente gravedad como para necesitar atención médica especializada.

Además, un estudio realizado por Ayoung-Chee et al. (2014), menciona que casi la mitad de pacientes que han sido hospitalizados por una caída sufren otra caída posteriormente, y más de un tercio fallece dentro del primer año. Es común que muchos pacientes de edad avanzada que son dados de alta directamente desde servicios de urgencias debido a lesiones menores provocadas por caídas, sufran caídas repetidas, experimenten una disminución en su capacidad funcional, y regresen para recibir atención médica nuevamente en los siguientes tres meses.

En el contexto ecuatoriano, el portal worldlifeexpectancy (2020) reporta que hubo 1114 muertes por caídas en el Ecuador ubicándolo en el puesto 77 del ranking mundial. A través de la combinación de tecnología vestible y una aplicación móvil complementaria, se aspira a proporcionar a los usuarios una herramienta efectiva para compartir su ubicación en tiempo real con personas de su elección, al tiempo que brinda la capacidad de activar alertas de ayuda en situaciones críticas.

Según las conclusiones de una investigación realizada en Statista publicadas por Fernández (2023), en el año 2022, el 90% de la población es propietaria de al menos un teléfono inteligente. Las aplicaciones móviles instaladas en estos dispositivos ayudan a las

personas a enfrentar situaciones de emergencia, inicialmente se utilizaba la información de la geolocalización obtenida por los dispositivos móviles, pero autores como Repanovici & Nedelcu (2021) indican que información personal, como datos del usuario o referencias de la emergencia son claves y se pueden facilitar a los operadores de emergencia.

La presente investigación se sumerge en el desafío de la detección de caídas, un problema de salud pública que afecta a millones de personas en todo el mundo. Las caídas no solo representan una amenaza para la vida, como lo demuestran las estadísticas de la OMS, sino que también generan un impacto significativo en la calidad de vida de quienes las experimentan y sus seres queridos.

Planteamiento Del Problema

Basados en el contexto anteriormente expuesto en la introducción, el presente trabajo aborda la problemática de las caídas debido a las graves consecuencias que pueden traer consigo. La OMS (2021) define a las caídas como eventos no planeados que resultan en la pérdida de equilibrio y el impacto con el suelo u otra superficie sólida. Estos incidentes pueden variar en severidad, desde tener consecuencias leves como lesiones en las zonas afectadas hasta situaciones más graves como fracturas e incluso la muerte. Además, como se mencionó anteriormente, las caídas ocupan el segundo puesto en las causas mundiales de defunción por traumatismos involuntarios, por detrás de las colisiones de tránsito.

El Dr. Alexandre B. de Moura (2023) , cirujano ortopédico certificado por la Junta Estadounidense de Cirujanos Ortopédicos, ofrece un poco más de profundidad en los tipos más comunes de lesiones que se dan por caídas, estas pueden ser moretones, raspones, cortes, esguinces, fracturas de huesos, estiramiento de los tendones, dislocaciones, entre otros. También menciona que las causas más comunes son las inclemencias del tiempo, aceras rotas, suelo húmedo o irregular, iluminación o calzado inadecuado.

En este proyecto se hace uso de la tecnología vestible debido a las ventajas que otorga, José Santoyo Ramón menciona en su obra (Contribuciones al estudio y análisis de sistemas vestibles de detección de caídas, 2021) que se aprovechan los sensores que se integran fácilmente en la indumentaria del paciente al mismo tiempo que elimina la limitación en la zona de vigilancia como lo hacen otras alternativas; además, estos resultan menos intrusivos y suelen tener costos más asequibles.

Es por esto por lo que se plantea desarrollar una aplicación la cual mediante la detección del cambio brusco de velocidad del dispositivo en cortos períodos de tiempo se generen alertas de seguridad, estas alertas contendrán información del usuario como: geolocalización, fecha y hora cuando se produjo la caída y los contactos a quienes se informó mediante un SMS.

Pregunta Científica

¿Cómo identificar caídas de manera efectiva utilizando sistemas de detección basados en dispositivos vestibles e integrar alertas de emergencia para brindar asistencia oportuna a las personas en riesgo de caídas?

Objetivos

Objetivo General

Desarrollar una aplicación utilizando tecnologías vestibles que permita la detección precisa de caídas y la transmisión automática de alertas a contactos designados, con el fin de mejorar la seguridad personal y proporcionar asistencia inmediata en situaciones de emergencia.

Objetivos Específicos

Definir la tecnología vestible que se usará para recolección de datos y su funcionamiento.

Desarrollar el algoritmo de detección de caída y el sistema de alertas automáticas para notificar a personas elegidas por el usuario.

Programar la aplicación móvil la cual tendrá un registro de las caídas y ofrecerá información de estas mismas.

Realizar pruebas de funcionalidad para verificar el funcionamiento del sistema.

Justificación

Los autores He et al. (2016) pronostican que para el año 2050, la población actual de personas mayores aumentará del 8.5% al 20%. Tomando en consideración la investigación realizada por Yu et al. (2021), las personas con más de 65 años tienen al menos un 28% de probabilidad de sufrir una caída anualmente y entre el 20-30% de estas caídas pueden producir lesiones graves o incluso la muerte. Dada la alta frecuencia de caídas y sus severas consecuencias entre las personas mayores, resulta esencial implementar métodos efectivos de detección y prevención de caídas para salvaguardar su seguridad y mejorar significativamente su calidad de vida.

En respuesta a esta necesidad crítica, el sistema de detección de caídas instalado en la tecnología vestible y el uso de los sensores integrados proporciona una herramienta práctica y efectiva. Este sistema no solo agiliza la respuesta en situaciones de emergencia que pueden surgir al momento de una caída, sino que también contribuye significativamente a la reducción de la mortalidad o lesiones causadas por las mismas. Además de beneficiar a la población mayor, este proyecto impacta positivamente en varios grupos vulnerables, como deportistas activos, trabajadores en entornos peligrosos, personas con condiciones de equilibrio precarias y mujeres embarazadas. Todos estos grupos se benefician de una detección temprana y una respuesta rápida ante posibles caídas, lo que demuestra la versatilidad y el valor extendido de la tecnología vestible en la prevención de caídas.

MARCO TEÓRICO

Marco Teórico

En el capítulo actual se explora los fundamentos fisiológicos relacionados con estos incidentes, así como las tecnologías innovadoras que permiten una detección efectiva. A continuación, se presenta un análisis detallado para comprender a fondo los elementos cruciales de este sistema.

Caídas de las personas

El médico Homero Gac Espínola (2023), instructor del Departamento de Medicina Interna, Geriátrica y Gerontología, define a las caídas como un evento espontáneo no planeado en donde el cuerpo del paciente colapsa en el suelo o a un nivel inferior. También hace mención que la mayor parte de las caídas que se presentan en las personas de mayor edad son productos de una enfermedad y no siempre se deben a la edad o a factores ambientales.

Siguiendo esta línea, la Organización Panamericana de la Salud (OPS, 2023) afirma que las caídas son accidentes frecuentes y que, de cada diez caídas, una resulta en lesiones serias. Este dato resalta la importancia de implementar medidas preventivas efectivas que puedan mitigar el impacto de estos accidentes en la población vulnerable.

Finalmente, la fundación Mapfre de España, organización dedicada al bienestar social y la calidad de vida de la población, realizó un estudio (Caídas en personas mayores, 2022) en el que concluye con ciertos puntos a destacar:

- Solamente el 15.5% de las personas mayores que sufren de caídas logran levantarse por sí mismas.
- Las consecuencias de las caídas provocan un deterioro impactante en las funciones de la vida diaria. Tras sufrir la caída, un 15.1% de las personas comienzan a ser parcialmente dependientes, mientras que un 2.1% queda totalmente dependiente para realizar las funciones cotidianas.

Las caídas no solo resultan en una mortalidad directa inmediata del 1,6% entre las personas mayores, sino que también tienen un impacto significativo a largo plazo. Después de seis meses, aproximadamente el 11% de las personas que sufren caídas han fallecido.

Es común pensar que las caídas son inevitables en las personas mayores, a menudo asociadas con problemas de motricidad similar a los niños durante su desarrollo. Sin embargo, este estudio sugiere que esta percepción es incorrecta. Se encontró que en el 58% de los casos de caídas en personas mayores, existían factores externos conocidos y, por lo tanto, prevenibles. Esto indica que hay oportunidades significativas para mitigar este riesgo.

Clasificación de las caídas

La OPS (2023) menciona que existen tres tipos de caídas existentes

- **Accidental:** Se producen por razones ajenas a las personas y que no vuelven a ocurrir.
- **Repetida:** Indica la presencia de enfermedades, medicamentos, etc.
- **Prolongada:** Son aquellas caídas en donde la persona permanece por más de 15 minutos en el suelo debido a que no se pueden levantar por sí solos y requieren de ayuda externa. Generalmente ocurren en personas de edad más avanzada.

Causas de las caídas

Existen múltiples factores los cuales pueden llevar a que se produzca una caída, se dividirán en dos tipos: internas y externas.

Causas internas de las caídas

Se definen como causas internas a aquellos factores que producen caídas a causa de problemas existentes en la persona. Montse Queral (2021), especialista en Medicina de Familia y Geriátrica en MAPFRE, menciona los factores más comunes.

- **Visuales**

- o Percepción anómala de la profundidad
- o Disminución de la tolerancia de la luz
- o Disminución de la capacidad de acomodación
- o Cataratas
- o Glaucoma
- Sistema nervioso
 - o Enlentecimiento del tiempo de reacción y de los reflejos
 - o Disminución de la sensibilidad propioceptiva
- Cardiovasculares
 - o Infarto agudo de miocardio
 - o Accidente vascular cerebral
 - o Enfermedad de Parkinson
 - o Insuficiencia cardíaca

Causas externas de las caídas

Son factores que se ubican en el entorno de la persona. Montse Queral (2021) también menciona algunos de los factores más comunes.

- Falta de luz
- Indumentaria no adecuada
- Habitaciones mal acomodadas
- Pisos resbalosos

Consecuencias en adultos mayores

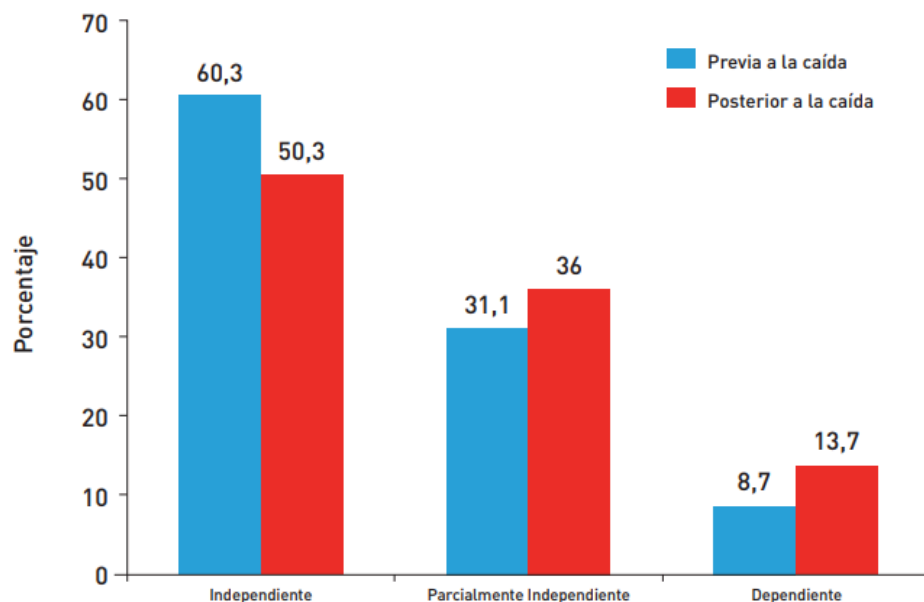
Retomando el estudio realizado por la Fundación MAPFRE (2022), que incluyó la participación de 1610 pacientes, se observa que un 91,4% de las personas presentó algún tipo de lesión. Dentro de este grupo, el 51,9% sufrió heridas, contusiones o laceraciones, mientras que un 38,3% experimentó fracturas. Además, es notable que un 16,6% de los pacientes padeció un traumatismo craneal y un 2,4% presentó lesiones intracraneales,

como hematoma epidural, hematoma subdural o hemorragia intraparenquimatosa o subaracnoidea. Es posible que algunas de estas consecuencias se hayan agravado debido a la atención tardía con la que se informó de la caída, subrayando la importancia de una detección temprana y respuesta rápidas para mitigar los efectos negativos de estos incidentes.

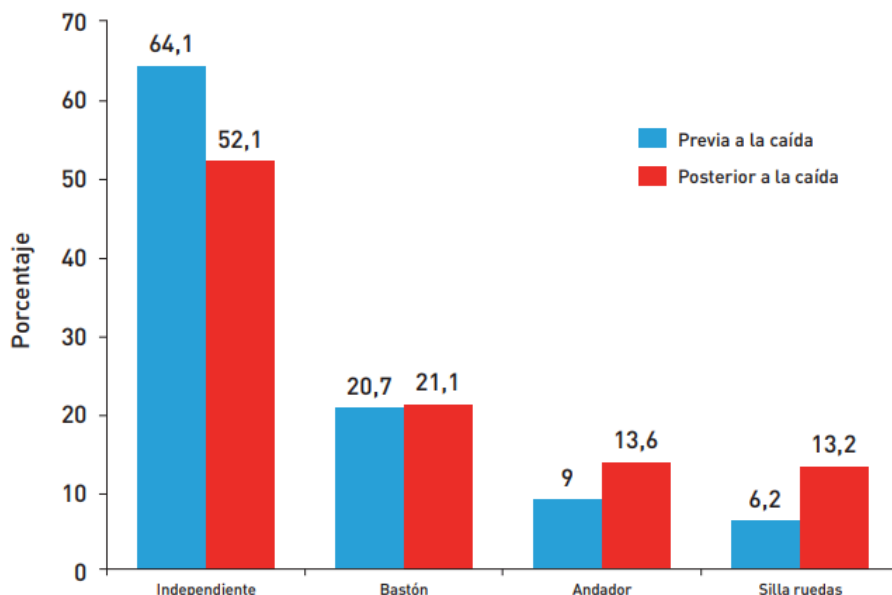
La Figura 1 y Figura 2 presentados destacan el impacto significativo que las caídas tienen en la movilidad y la independencia funcional de los pacientes mayores. Antes de la caída, el 64.1% de los pacientes eran independientes, pero este porcentaje disminuye a 52.1% después del incidente. Asimismo, se observa un aumento en la dependencia de ayudas para la movilidad, como bastones, andadores y sillas de ruedas tras la caída. El estudio también revela que el porcentaje de pacientes que eran completamente independientes pasa del 60.3% al 50.3% después de una caída, mientras que la dependencia total aumenta del 8.7% al 13.7%.

Figura 1

Deterioro funcional agudo secundario a la caída



Fuente: Fundación MAPFRE (Fundación Mapfre, 2022), Estudio FALL-ER

Figura 2*Deterioro agudo de la movilidad secundario a la caída*

Fuente: Fundación MAPFRE (Fundación Mapfre, 2022), Estudio FALL-ER

El deterioro funcional agudo secundario a la caída se refiere a la pérdida inmediata y notable de la capacidad para llevar a cabo las actividades básicas de la vida diaria (ABVD) después de una caída. Esto puede incluir dificultades para realizar tareas como vestirse, bañarse, comer, y otras actividades esenciales para la vida independiente.

Por otro lado, el deterioro agudo de la movilidad secundario a la caída se enfoca específicamente en la pérdida repentina de la capacidad de moverse de manera independiente o con menos ayuda. Esto puede manifestarse como una mayor necesidad de dispositivos de asistencia, como bastones, andadores, o sillas de ruedas, o una dependencia total de la ayuda de otros para desplazarse.

Prevención de las caídas

Las caídas, aunque sean eventos imprevistos, no deben considerarse inevitables. En realidad, existen diversas medidas y consideraciones que podemos adoptar para prevenir o mitigar el riesgo de caídas. Desde la modificación del entorno hasta la atención a factores personales, la prevención de caídas se convierte en un componente esencial para mantener la seguridad y el bienestar. Explorar estrategias y prácticas para reducir este riesgo no solo contribuye a la salud física, sino que también promueve un entorno más seguro y favorable para el desenvolvimiento cotidiano.

El Instituto Nacional del Envejecimiento (2023) menciona algunos de los factores a tomar en consideración para el cuidado de la salud en las personas y así tener un menor riesgo de sufrir una caída.

- **Ajustar el entorno doméstico**

Modificar el entorno doméstico desempeña un papel crucial en la prevención de caídas al crear un espacio adaptado y seguro que minimiza los riesgos inherentes. Al ajustar el entorno, se pueden eliminar obstáculos, mejorar la iluminación, instalar barandas y garantizar superficies antideslizantes, lo que reduce significativamente la probabilidad de accidentes.

- **Realizar actividad física**

La práctica regular de actividad física emerge como un pilar fundamental en la prevención de caídas al fortalecer el cuerpo, mejorar el equilibrio y la coordinación, y mantener la agilidad. El ejercicio contribuye al desarrollo de la fuerza muscular, en especial en las áreas clave que sustentan la estabilidad. Además, al participar en actividades que fomentan la flexibilidad y el control corporal, se fortalecen las habilidades motoras esenciales para prevenir tropiezos y caídas accidentales. La actividad física no solo fortalece el cuerpo, sino que también optimiza la función cognitiva, promoviendo una mayor conciencia espacial y temporal.

- **Alimentación sana**

El consumo adecuado de proteínas favorece el desarrollo y mantenimiento de la masa muscular, esencial para la estabilidad y el equilibrio. Además, una alimentación saludable promueve un peso corporal adecuado, aliviando la carga en las articulaciones y facilitando la movilidad. La atención a la nutrición no solo impacta en la fortaleza física, sino que también incide en la energía y vitalidad.

- **Implementación de avances tecnológicos**

La utilización de tecnologías avanzadas se presenta como una herramienta moderna para prevenir caídas, proporcionando soluciones innovadoras que aumentan la seguridad y permiten un seguimiento personalizado. Dispositivos como sensores de movimiento, alarmas inteligentes y sistemas de domótica pueden alertar sobre posibles riesgos, como cambios repentinos en la actividad o situaciones anómalas en el entorno doméstico. Asimismo, la teleasistencia y las aplicaciones de salud móvil permiten un seguimiento continuo, brindando a los profesionales de la salud y a los cuidadores información valiosa para anticiparse a posibles situaciones de riesgo. La tecnología, al ofrecer herramientas de detección temprana y asistencia remota, no solo proporciona un enfoque proactivo en la prevención de caídas, sino que también ofrece una capa adicional de seguridad que puede ser especialmente beneficiosa para aquellos con mayores riesgos, como las personas mayores.

Marco legal

El presente trabajo de integración curricular propuesto se alinea con varias normativas y proyectos vigentes en Ecuador, destacando la intersección entre tecnología, salud y bienestar social.

Primero, según la Constitución de la República del Ecuador (2008), el Estado tiene la obligación de garantizar el derecho a la salud, la seguridad y una vida digna para todos los ciudadanos. Este proyecto, que busca implementar un sistema de detección de caídas y alertas inmediatas, está directamente alineado con el mandato constitucional de

proporcionar atención integral y protección a los grupos vulnerables, como los adultos mayores. Al facilitar una respuesta rápida a emergencias, la tecnología desarrollada no solo protege la vida de estos ciudadanos, sino que también mejora su calidad de vida, cumpliendo así con los principios establecidos en el artículo 32 de la Constitución.

Segundo, la Agenda 2030 para el Desarrollo Sostenible (CEPAL, 2015), adoptada por Ecuador, subraya la necesidad de garantizar una vida sana y promover el bienestar para todos en todas las edades. El proyecto de detección de caídas contribuye directamente al Objetivo de Desarrollo Sostenible (ODS) 3, que busca asegurar vidas saludables y promover el bienestar para todos. Implementando una tecnología que puede prevenir lesiones y salvar vidas, este proyecto se inserta en los esfuerzos globales de Ecuador para mejorar la salud pública y la calidad de vida de sus ciudadanos, abordando un problema significativo de salud pública como lo son las caídas en la población mayor.

Por último, el Plan de Desarrollo para el Nuevo Ecuador 2024-2025 es una iniciativa aprobada en el gobierno del presidente Daniel Noboa Azín. Este plan se estructura en cuatro ejes principales: social, económico, infraestructura y medio ambiente, e institucional. Su objetivo es delinear las acciones del Estado para mejorar la calidad de vida de la población, promover el desarrollo sostenible y garantizar la eficiencia en la asignación de recursos públicos. Los ejes incluyen 9 objetivos, 70 políticas y 105 metas, todas alineadas con los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030.

Este plan se destaca por su enfoque integral, abarcando desde la mejora de las condiciones de vida y el acceso equitativo a servicios de salud y educación, hasta la promoción de la innovación y la conectividad, y la construcción de un Estado eficiente y transparente. Por ejemplo, el eje social del Plan de Desarrollo busca mejorar las condiciones de vida de la población, promoviendo el acceso equitativo a salud y bienestar. El presente proyecto de integración está alineado con este objetivo, ya que se centra en la mejora de la salud y la seguridad de los usuarios, particularmente de aquellos en riesgo de caídas. La implementación de este sistema puede contribuir significativamente a reducir el

impacto de las caídas, un problema de salud común y potencialmente grave, mejorando así la calidad de vida y el bienestar de los ciudadanos.

Además, el Plan de Desarrollo destaca la importancia de la innovación y el uso de tecnologías avanzadas para fomentar el desarrollo económico y social. El desarrollo de aplicaciones avanzadas para dispositivos vestibles, como esta propuesta, no solo se alinea con este enfoque, sino que también promueve el uso de tecnologías de punta para resolver problemas prácticos y urgentes. Este tipo de innovación es crucial para estimular la economía del conocimiento y apoyar la modernización del sistema de salud pública.

Tipos de sistemas de detección de caídas

Los sistemas de detección de caídas son tecnologías diseñadas para identificar situaciones en las que una persona puede estar experimentando una caída y proporcionar una respuesta o alerta adecuada. Estos sistemas utilizan diversos sensores y dispositivos para monitorear el movimiento y las actividades del usuario, identificando patrones que podrían indicar una caída.

Sensores de ambiente

En el contexto de estos sistemas, los sensores de ambiente se utilizan para recopilar datos sobre el entorno de una persona, como la ubicación, la orientación, la velocidad y la aceleración. Estos datos se pueden utilizar para identificar señales de advertencia de una caída, como un cambio repentino en la dirección o la velocidad de movimiento.

Los sensores de movimiento o Infrarrojos Pasivos (2023) son dispositivos específicamente diseñados para detectar el cambio en la posición o en los movimientos dentro de un área delimitada. Cuando un cuerpo entra el campo de visión del sensor se genera un cambio térmico el cual es interpretado como movimiento. La sensibilidad de estos sensores se puede configurar para no generar falsas alarmas, esto permite adaptarlos a entornos y necesidades específicas.

Sistemas de visión

Los sistemas de visión hacen uso de la tecnología de cámaras y análisis de imágenes para la identificación de patrones visuales los cuales puedan indicar una posible caída y generar una respuesta en base a estos.

Estos sistemas hacen uso de cámaras de vídeo o cámaras especializadas para capturar imágenes dentro del entorno que se encuentre monitoreando. Mediante el análisis de las imágenes se pueden detectar movimientos inusuales como cambios en la postura de las personas o cualquier otro movimiento que indique una posible caída.

Son particularmente útiles en entornos donde se requiere una detección más compleja o donde los sensores de movimiento tradicionales pueden no ser suficientemente precisos. Además, pueden adaptarse a diversas configuraciones y condiciones de iluminación para proporcionar una cobertura más completa.

Avances de sensores en la actualidad

En los últimos años, se han realizado avances significativos en la tecnología de sensores de movimiento, acelerómetros y giroscopios, que han transformado la funcionalidad y precisión de los dispositivos vestibles. Estos avances han permitido una mayor integración y miniaturización de los sensores, mejorando la capacidad de monitoreo y la comodidad del usuario.

- **Mejora en la precisión y sensibilidad de los sensores:**

Los sensores de movimiento modernos, incluidos acelerómetros y giroscopios, han experimentado mejoras en su precisión y sensibilidad. Esto se debe a la implementación de nuevos materiales y técnicas de fabricación que permiten medir movimientos con mayor exactitud. Por ejemplo, los sensores MEMS (sistemas microelectromecánicos) han reducido su tamaño a micrómetros, lo que permite su integración en dispositivos vestibles sin comprometer la precisión.

- **Integración y miniaturización:**

La miniaturización de los sensores ha sido un avance crucial. Los dispositivos vestibles actuales, como los relojes inteligentes y pulseras de fitness, incorporan múltiples sensores en un espacio reducido sin aumentar significativamente el tamaño del dispositivo. Esta integración permite que los dispositivos vestibles sean más ligeros y cómodos para el uso diario.

- **Mejora en la duración de la batería:**

Otro avance importante es la optimización del consumo de energía de los sensores. Los acelerómetros y giroscopios modernos están diseñados para consumir menos energía, lo que prolonga la duración de la batería de los dispositivos vestibles. Esto es fundamental para aplicaciones que requieren monitoreo continuo, como el seguimiento del sueño o la actividad física a lo largo del día.

- **Conectividad y sincronización:**

Los avances en conectividad, como el uso de Bluetooth Low Energy (BLE), han permitido que los dispositivos vestibles se sincronicen fácilmente con teléfonos inteligentes y otros dispositivos. Esto facilita la transferencia de datos en tiempo real, permitiendo a los usuarios acceder a sus estadísticas y recibir alertas instantáneas.

Tecnología vestible

Las tecnologías vestibles se basan en una amplia gama de tecnologías, incluyendo microelectrónica, sensores, actuadores, comunicaciones y software. Estas se utilizan en una variedad de aplicaciones, incluidas salud, fitness, entretenimiento, seguridad y productividad.

La tecnología vestible, también conocida como ropa inteligente o electrónica textil, son dispositivos electrónicos inteligentes integrados en la vestimenta o usados como accesorios que pueden actuar como extensión del cuerpo.

Algunos dispositivos vestibles, como relojes inteligentes o pulseras de actividad, están equipados con sensores de movimiento y acelerómetros que pueden detectar patrones de movimiento asociados con caídas. Estos dispositivos pueden monitorear continuamente la actividad del usuario y, en caso de detectar una posible caída, activar alertas automáticas.

Smartwatch

Los smartwatches son dispositivos electrónicos que realizan más funciones que un reloj tradicional, debido a que incorpora diversas capacidades que lo asemejan a un teléfono inteligente. Estos dispositivos cuentan con pantallas táctiles, conectividad inalámbrica y sensores integrados para diversas funciones, entre estas se encuentran la detección de actividades físicas que realice el usuario y la salud de este.

Algunos de los sensores más comunes que contienen los smartwatches son: acelerómetros, giroscopios, sensor de frecuencia cardíaca, entre otros. Es posible realizar la detección de eventos que coincidan con movimientos similares a una caída y realizar una alerta mediante notificaciones visuales en la pantalla, alertas de sonido e incluso el envío automático de mensajes de emergencia a contactos preseleccionados por el usuario.

Algoritmos usados para la detección de caídas

Basados en umbrales

La principal característica de estos algoritmos radica en que se utiliza una o más de las magnitudes inerciales para poder identificar patrones característicos de una caída. Estos patrones se comparan con valores umbrales predefinidos y si son sobrepasados, se considera que ha ocurrido una caída. Una de las principales ventajas de estos algoritmos está en la simplicidad y el bajo coste tecnológico.

Entre los algoritmos más comunes se encuentran:

- Monitorización de umbral básico: se realiza una comprobación del módulo de aceleración para determinar si superó un umbral máximo definido y considerar que se produjo una caída.
- iFall: los autores Sposaro y Tyson (2009) definen que este algoritmo comprueba si el módulo de aceleración pasa por una caída libre y subida brusca se considera una caída. Se hacen uso de dos umbrales: umbral inferior y umbral superior.
- Fall Index: el autor Yoshida (2005) también considera la existencia de la fase de caída libre y subida brusca, a diferencia del algoritmo anterior, este hace uso del módulo de diferencia de aceleración de algunas muestras consecutivas.

Basados en aprendizaje automático

Son un tipo de algoritmo que aprende a identificar patrones en los datos y se utilizan para entrenar modelos de detección de caídas que pueden identificar una caída con mayor precisión que los métodos tradicionales. Estos algoritmos se alimentan de un conjunto de datos de entrenamiento que contiene ejemplos de caídas y no caídas y aprende a identificar las características que distinguen entre las dos clases de datos. Una vez que el algoritmo ha sido entrenado, se puede utilizar para detectar caídas en nuevos datos.

Utilizar tecnología de aprendizaje automático en un smartwatch para la detección de caídas puede presentar desafíos significativos debido a los requerimientos de potencia de cómputo, consumo de energía y complejidad asociados. Los algoritmos de aprendizaje automático demandan recursos computacionales considerablemente altos tanto para el entrenamiento como para la inferencia, lo cual puede sobrepasar los límites de los dispositivos portátiles como los smartwatches, que tienen recursos limitados de CPU, memoria y batería. Además, el consumo energético adicional podría reducir drásticamente la duración de la batería del dispositivo, lo cual resulta poco práctico en situaciones de emergencia.

Situación actual de los algoritmos de detección de caídas

En la actualidad, los algoritmos de detección de caídas se desarrollan principalmente a partir de dos enfoques: algoritmos basados en umbrales y algoritmos de aprendizaje automático. Según el estudio realizado por Yu et al (2021)., la detección de caídas antes del impacto (pre-impacto) requiere la identificación precisa del inicio del movimiento de caída, diferenciando entre actividades normales y el comienzo de la caída. Este estudio demostró que un algoritmo de aprendizaje profundo logró la mejor combinación de sensibilidad (99.32%) y especificidad (99.01%). Le siguió un algoritmo de máquina de vectores de soporte (SVM), que alcanzó una sensibilidad del 99.77% y una especificidad del 94.87%. Por otro lado, el algoritmo basado en umbrales obtuvo resultados inferiores, con una sensibilidad del 95.50% y una especificidad del 83.43%.

A pesar de que los algoritmos de aprendizaje automático (AA) muestran una mayor precisión en la detección de caídas, la decisión de optar por un algoritmo basado en umbrales para este proyecto responde a varios factores críticos que buscan garantizar un sistema más compacto y ampliamente compatible con una mayor variedad de smartwatches. Los wearables modernos son dispositivos de propósito general, diseñados para realizar una amplia gama de funciones simultáneamente, lo cual implica que el poder de procesamiento y la memoria disponibles son limitados.

En este contexto, los algoritmos de aprendizaje automático, aunque precisos, requieren una capacidad de cómputo significativa y un consumo de energía elevado, lo cual puede ser prohibitivo en dispositivos con recursos limitados. Estos dispositivos, al manejar múltiples procesos y aplicaciones de manera concurrente, no pueden permitirse dedicar grandes cantidades de recursos a una sola tarea sin afectar el rendimiento general y la duración de la batería. Por lo tanto, la simplicidad y eficiencia de los algoritmos basados en umbrales los convierten en una opción más viable para ser implementados en estos dispositivos.

Además, los algoritmos basados en umbrales son más fáciles de ajustar y modificar en función de las necesidades específicas del usuario y las características del dispositivo. La capacidad de realizar ajustes rápidos y precisos es crucial en un entorno donde las condiciones y los requisitos pueden variar considerablemente. Esta flexibilidad permite que los algoritmos se adapten mejor a diferentes modelos de smartwatches y a las particularidades de cada usuario, garantizando una mayor adaptabilidad.

Otra consideración importante es la estabilidad y la predictibilidad del rendimiento. Los algoritmos de aprendizaje automático pueden requerir entrenamientos continuos y ajustes para mantener su precisión, lo cual no siempre es práctico en un entorno de dispositivo wearable. En cambio, los algoritmos basados en umbrales, una vez calibrados adecuadamente, pueden ofrecer un rendimiento consistente sin la necesidad de un mantenimiento constante.

Sistema operativo del smartwatch - Wear os

Fue presentado por primera vez en marzo de 2014 como una extensión del sistema operativo Android, adaptado para funcionar en dispositivos con pantallas más pequeñas y diseñado para proporcionar una experiencia de usuario intuitiva y centrada en la información en tiempo real. A lo largo de los años, Wear OS ha experimentado varias actualizaciones y mejoras para mejorar su rendimiento, funcionalidad y compatibilidad con una amplia gama de dispositivos portátiles de diferentes fabricantes. La versión más reciente, Wear OS 4, continúa evolucionando para ofrecer una experiencia completa y enriquecedora para los usuarios de dispositivos portátiles, con un enfoque en la salud, el bienestar y la conectividad con otros dispositivos.

La elección de este sistema operativo se justifica por el reciente cambio de Samsung al abandonar Tizen OS (Trokko, 2021), en favor de Wear OS 4 en sus dispositivos, lo que indica una tendencia hacia esta plataforma en la industria.

Además de Samsung, otras empresas prominentes como Huawei y Mobvoi también han adoptado Wear OS en sus dispositivos, lo que amplía significativamente el alcance y la compatibilidad de las aplicaciones desarrolladas para esta plataforma. Además, facilita su adopción y uso por parte de los usuarios finales.

Patrones de arquitectura de software - MVC

Los patrones de diseño se refieren a soluciones recurrentes a problemas comunes que surgen durante el desarrollo de software. Estos patrones son prácticas establecidas y probadas que proporcionan una estructura y un enfoque para resolver ciertos tipos de problemas de diseño de software. Los patrones de diseño no son códigos o soluciones directas, sino más bien descripciones generales de cómo abordar y resolver problemas específicos.

El patrón de diseño Modelo-Vista-Controlador (MVC) es una arquitectura de software que separa una aplicación en tres componentes principales: el modelo, la vista y el controlador. El modelo se encarga de la lógica de negocio y la gestión de los datos, proporcionando una interfaz para acceder y manipular la información. La vista es la responsable de la presentación de los datos y la interfaz de usuario, mostrando la información del modelo de manera interactiva y visualmente atractiva. El controlador actúa como un intermediario entre el modelo y la vista, manejando la lógica de la aplicación, las entradas del usuario y las actualizaciones a la vista en respuesta a los cambios en el modelo.

Herramientas de desarrollo

Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para la plataforma Android, desarrollado por Google. Basado en IntelliJ IDEA de JetBrains, este IDE ofrece un conjunto completo de herramientas para el desarrollo de aplicaciones Android, facilitando tanto la escritura de código como el diseño de interfaces gráficas. Entre sus características

más destacadas se encuentran un editor de código avanzado con autocompletado inteligente, un emulador de Android para pruebas en diferentes dispositivos virtuales, y un sistema de construcción basado en Gradle que gestiona dependencias y configuraciones de compilación.

Github

El autor Gustavo B. (Hostinger, 2023) describe a GitHub como una plataforma de desarrollo colaborativo basada en la web que utiliza el sistema de control de versiones Git. Permite a los desarrolladores alojar sus proyectos, gestionar versiones de código, y colaborar en equipo de manera eficiente. Además de ofrecer almacenamiento para repositorios, GitHub proporciona herramientas para la revisión de código, seguimiento de problemas, y la integración continua, facilitando la gestión y el desarrollo de proyectos de software de cualquier escala.

Definición de API

Sydle (2023) define una API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés) como un conjunto de reglas y protocolos que permiten la comunicación entre diferentes aplicaciones de software. Las APIs definen los métodos y datos que los desarrolladores pueden usar para interactuar con una aplicación, servicio o sistema operativo, facilitando la integración y la funcionalidad entre distintas plataformas. A través de las APIs, es posible acceder a las funciones y datos de otras aplicaciones de manera segura y controlada, promoviendo la interoperabilidad y la reutilización de código en el desarrollo de software.

Amazon Simple Notification Service

Según Amazon (2024), Amazon Simple Notification Service (Amazon SNS) se define como un servicio gestionado que facilita la entrega de mensajes desde quienes los publican a quienes los reciben (también conocidos como emisores y receptores). Los emisores interactúan de manera asíncrona con los receptores al enviar mensajes a un "tema", que

sirve como un canal de comunicación. Los usuarios pueden suscribirse a estos temas de SNS y recibir los mensajes publicados a través de varios tipos de puntos de conexión compatibles.

Amazon SNS ofrece dos tipos principales de mensajería: aplicación a aplicación (A2A) y aplicación a persona (A2P). En A2A, los mensajes pueden ser entregados a suscriptores como Amazon Data Firehose, AWS Lambda, Amazon SQS, endpoints HTTP/S y canalizaciones de Event Fork. En A2P, las notificaciones son enviadas a suscriptores como aplicaciones móviles, números de teléfono móvil y direcciones de correo electrónico.

Google Maps SDK

El Google Maps SDK (Software Development Kit) es un conjunto de herramientas y APIs desarrollado por Google que permite integrar funcionalidades avanzadas de mapas en aplicaciones para dispositivos móviles y web. Este SDK proporciona acceso a mapas interactivos y enriquecidos, permitiendo visualizar y manipular mapas en tiempo real. Se puede incorporar características como la visualización de ubicaciones, rutas y puntos de interés, así como la integración con servicios de geolocalización y navegación.

Además de las capacidades básicas de visualización de mapas, el Google Maps SDK ofrece funcionalidades avanzadas como la personalización de estilos de mapas, la superposición de capas informativas como datos de tráfico en tiempo real, y la integración con Street View para vistas panorámicas de ubicaciones específicas. También facilita la creación de aplicaciones basadas en la ubicación mediante el uso de API para la detección de ubicación del usuario y la geocodificación de direcciones.

Sincronización de datos en Wear OS

En el contexto de Wear OS, la sincronización de datos es un componente crucial para la integración efectiva entre dispositivos portátiles y teléfonos inteligentes. La

sincronización de datos en Wear OS permite que las aplicaciones compartan información en tiempo real, mejorando la funcionalidad y la experiencia del usuario. En la documentación oficial para desarrolladores (Android Developer, 2024) se puede encontrar que los datos pueden sincronizarse desde la red, lo que incluye la transferencia de datos entre el smartwatch y un servidor en la nube, permitiendo así que las aplicaciones en el dispositivo portátil se mantengan actualizadas con la información más reciente, independientemente de la ubicación del usuario.

La Data Layer API en Wear OS es un conjunto de herramientas diseñadas para facilitar la comunicación y sincronización de datos entre dispositivos portátiles y teléfonos inteligentes. Esta API permite crear aplicaciones que pueden compartir datos en tiempo real, lo que es esencial para aplicaciones que requieren una comunicación constante y precisa entre dispositivos. La Data Layer API incluye varios componentes clave, como `DataClient`, `MessageClient`, y `ChannelClient`, cada uno con funciones específicas para manejar diferentes tipos de comunicación y sincronización de datos.

DataClient

`DataClient` es uno de los componentes más versátiles y esenciales de la Data Layer API en Wear OS, diseñado para gestionar la sincronización de datos estructurados entre dispositivos portátiles y teléfonos inteligentes. Este componente se basa en el uso de objetos `Dataltem`, que actúan como contenedores para los datos que necesitan ser compartidos y sincronizados.

Un `Dataltem` es una estructura de datos que contiene una colección de pares clave-valor y opcionalmente, un archivo binario. Cada `Dataltem` se identifica de manera única mediante una URI, que incluye el nombre del paquete y una ruta específica. Esta URI garantiza que los datos se puedan identificar y acceder de manera consistente en todos los dispositivos conectados.

Cuando se actualiza un DataItem en un dispositivo, DataClient se encarga de replicar esa actualización en todos los dispositivos conectados a través de la red Wear OS. Este proceso de replicación se realiza de manera eficiente y transparente para el usuario y el desarrollador, asegurando que los datos estén siempre sincronizados sin necesidad de intervención manual.

Una de las principales ventajas de DataClient es su capacidad para mantener los datos persistentes y sincronizados en un almacén de datos distribuido. Esto significa que cualquier cambio realizado en un DataItem se guarda automáticamente y se replica en todos los dispositivos emparejados, incluso si uno de los dispositivos está temporalmente fuera de línea. Una vez que el dispositivo vuelva a estar en línea, DataClient sincronizará automáticamente los datos pendientes, garantizando la coherencia en todo momento.

MessageClient

El MessageClient en la API de DataLayer para Wear OS proporciona una interfaz crucial para la comunicación asíncrona entre dispositivos Wear OS y dispositivos conectados, como teléfonos Android. Este cliente facilita el intercambio de datos entre la aplicación del teléfono y la aplicación asociada en el smartwatch a través de mensajes. Utiliza un modelo basado en eventos para permitir la transmisión eficiente de datos, crucial para aplicaciones que requieren actualizaciones en tiempo real o la sincronización de información entre dispositivos.

El MessageClient maneja la creación, envío y recepción de mensajes entre el teléfono y el smartwatch mediante un sistema de colas y listeners. Esto permite que las aplicaciones envíen datos estructurados, como textos o bytes, entre los dispositivos de manera confiable y eficiente. Además, proporciona métodos para gestionar la conexión y la desconexión, asegurando que la comunicación entre los dispositivos sea estable y robusta incluso en condiciones de red variables o intermitentes.

ChannelClient

El ChannelClient en la API de DataLayer para Wear OS ofrece una infraestructura avanzada para la transferencia de datos bidireccional entre dispositivos conectados, como teléfonos Android y smartwatches. A diferencia del MessageClient, que se centra en la comunicación asíncrona a través de mensajes individuales, el ChannelClient facilita la creación y gestión de canales persistentes de comunicación. Estos canales permiten un flujo continuo de datos en tiempo real entre los dispositivos, ideal para aplicaciones que requieren una sincronización constante de información o la transmisión de flujos de datos complejos.

Al utilizar el ChannelClient, se puede establecer canales de comunicación personalizados con configuraciones específicas de calidad de servicio y priorización de datos. Esto es crucial para optimizar el rendimiento y la eficiencia de las aplicaciones, asegurando que los datos críticos se transmitan de manera confiable y sin interrupciones significativas. Además, el cliente gestiona automáticamente la reconexión y la gestión de errores, lo que mejora la robustez y la disponibilidad de la comunicación entre los dispositivos en diversas condiciones de red.

Para conocer sobre la sincronización de datos entre el smartwatch y el smartphone se recomienda visitar el Anexo 1.- Envío De La Información Entre Dispositivos.

Metodología incremental

La plataforma Nimble (2022) menciona que la metodología incremental es un enfoque de gestión de proyectos que divide el proceso de desarrollo en varias fases o segmentos menores, conocidos como incrementos. Cada uno de estos segmentos se planifica, diseña, implementa y revisa de manera individual, permitiendo que el proyecto evolucione y se adapte a los cambios de requisitos o a nuevas ideas que surjan durante el desarrollo.

Etapas

1. **Planificación y diseño inicial:** Antes de comenzar el desarrollo, se establecen los requisitos generales y se desarrolla un esquema global del proyecto.
2. **Desarrollo por fases:** El proyecto se desglosa en múltiples módulos o incrementos. Cada uno de estos segmentos se desarrolla siguiendo un ciclo completo de análisis de requisitos, diseño, codificación e implementación.
3. **Implementación y evaluación:** Una vez completado, cada incremento se integra con los elementos ya existentes del sistema para formar un conjunto cohesivo y funcional.
4. **Retroalimentación y ajustes:** Tras cada lanzamiento de incremento, se recoge la retroalimentación de los usuarios o partes interesadas. Este feedback es vital para detectar fallos, hacer ajustes y refinar el producto.
5. **Iteración y refinamiento:** Con base en la retroalimentación y los resultados de las pruebas, se planifican y desarrollan nuevos incrementos. Este proceso iterativo permite ajustar el producto continuamente a las necesidades cambiantes del mercado o del entorno del usuario.

MÉTODO DE INVESTIGACIÓN

Método De Investigación

Enfoque de la investigación

Para este proyecto, se adopta un enfoque metodológico cuantitativo que se caracteriza por la recopilación y el análisis de datos numéricos objetivos obtenidos a través de pruebas y mediciones precisas. Este enfoque se justifica en la necesidad de evaluar de manera precisa y sistemática la efectividad del sistema de detección de caídas desarrollado. A través de la recolección de datos cuantitativos, se pueden establecer métricas claras y reproducibles para medir la precisión y la fiabilidad del algoritmo de detección, sin sesgos subjetivos. Esto permite obtener resultados robustos y verificables que respalden las conclusiones del estudio.

Además, el enfoque cuantitativo facilita la objetividad en la evaluación de variables clave como la precisión de detección y la tasa de falsos positivos y negativos. Esto es esencial para comprender el rendimiento del sistema en diferentes escenarios y condiciones. Al emplear métodos estadísticos adecuados para analizar los datos cuantitativos recogidos, se podrán identificar patrones significativos y realizar inferencias válidas sobre la efectividad del sistema.

Alcance de la investigación

El proyecto cuenta con un alcance exploratorio-descriptivo, ya que primero se explora la tecnología disponible en dispositivos vestibles y su potencial para detectar y alertar caídas de manera efectiva y no invasiva para el usuario. Esta exploración implica investigar y analizar las capacidades técnicas de los acelerómetros y otros sensores integrados en estos dispositivos, así como evaluar las diferentes metodologías y algoritmos de detección de caídas que pueden ser implementados. El objetivo es identificar las características y ventajas de estas tecnologías para su aplicación práctica en la resolución del problema de la detección de caídas.

Posteriormente, se procedió a una fase descriptiva donde se detalla la solución tecnológica propuesta. Esto incluye una descripción exhaustiva de cómo el sistema desarrollado utiliza los datos de los sensores para identificar caídas y enviar alertas de ayuda. Se explican las funcionalidades específicas del sistema, su arquitectura, y el flujo de datos desde la detección de la caída hasta la notificación de emergencia. Además, se analiza la utilidad y efectividad de la solución en términos de su capacidad para proporcionar asistencia oportuna y reducir riesgos para los usuarios. En esta etapa, la documentación se centra en demostrar cómo la tecnología implementada no solo cumple con los requisitos técnicos, sino que también ofrece un valor significativo en términos de seguridad y bienestar para los usuarios finales.

Delimitación de la investigación

En este proyecto se establece un plazo de un mes de duración, durante el cual se realizaron pruebas exhaustivas con un grupo selecto de participantes. La investigación se enfocó en simular escenarios de caídas y actividades de la vida diaria (ADL) para evaluar la precisión y efectividad del sistema de detección de caídas. Este enfoque permitió obtener datos variados y relevantes que ayudaron a diferenciar entre eventos normales y potenciales caídas, garantizando que el sistema responda adecuadamente en situaciones reales.

La muestra seleccionada proporciona una base sólida para el análisis inicial de la funcionalidad del sistema. Los datos recogidos de estos individuos, aunque limitados en diversidad demográfica, ofrecen una primera visión de cómo el sistema de detección de caídas responde a diferentes tipos de movimientos.

Aunque la muestra no es representativa de la totalidad de la población, por ejemplo, adultos mayores, los resultados obtenidos durante ese mes de pruebas proporcionan información valiosa sobre el rendimiento y la fiabilidad del sistema. Estos resultados sirven

como base para futuras investigaciones que podrán incluir muestras más amplias y diversas, incluyendo adultos mayores.

Métodos empleados y análisis de la información

La población del estudio se compone de cuatro adultos jóvenes, con edades alrededor de los 20 años. Aunque estos participantes no representan directamente a la población de adultos mayores o mujeres embarazadas, se eligieron deliberadamente para evitar exponer a estos individuos a riesgos innecesarios durante las pruebas. Se implementaron pruebas sistemáticas utilizando un enfoque iterativo para ajustar los umbrales de detección del sistema de manera dinámica durante las simulaciones de caídas y actividades de la vida diaria (ADL).

Las simulaciones incluyeron caídas inducidas de manera segura y movimientos cotidianos como caminar, sentarse y levantarse, lo que permitió obtener una gama completa de datos para el análisis. Este método permitió una adaptación continua y refinada de los parámetros de detección, asegurando que el sistema respondiera de manera óptima a diferentes escenarios y movimientos. Durante las pruebas, se registraron y analizaron cuidadosamente los datos obtenidos de los sensores para evaluar el rendimiento del algoritmo en términos de precisión y eficacia.

En cuanto al procesamiento y análisis de la información, se utilizó un enfoque cuantitativo para examinar los resultados de las pruebas. Los datos recopilados fueron procesados para calcular métricas clave como la tasa de detección de caídas, la tasa de falsos positivos y negativos, y otras estadísticas relevantes. Estas métricas proporcionaron una medida objetiva del rendimiento del sistema bajo diversas condiciones.

Elementos metodológicos específicos para TI

Diseño del proyecto

La metodología incremental se caracteriza por dividir el proyecto en ciclos de desarrollo iterativos, cada uno con objetivos específicos y entregas tangibles, como lo indica Pérez (2016). En este proyecto, esta metodología se aplicó dividiendo el desarrollo del sistema de detección de caídas en fases claramente definidas. Cada fase culminó en una versión funcional del sistema que pudo ser evaluada y refinada en función de la retroalimentación recibida. Esto permitió una mejora continua y una adaptación constante a las necesidades y expectativas de los usuarios.

El uso de la metodología incremental en este proyecto ofreció la ventaja de detectar y resolver posibles inconvenientes desde las primeras etapas de desarrollo. Al identificar y abordar estos problemas de manera temprana, se redujo significativamente el impacto negativo en la calidad y funcionalidad del producto final. Este enfoque aseguró que el sistema de detección de caídas fuera coherente y efectivo, proporcionando una solución robusta y confiable. Además, facilitó la integración de nuevas ideas, requisitos o cambios a lo largo del ciclo de desarrollo, asegurando que el producto final estuviera alineado con las necesidades emergentes de los usuarios.

Además, el diagrama de Gantt jugó un papel crucial en la implementación de esta metodología, proporcionando una herramienta esencial para la planificación y seguimiento meticuloso de cada ciclo de desarrollo. La estructura clara del diagrama, con plazos definidos y objetivos específicos, facilitó significativamente la gestión del tiempo y los recursos. La retroalimentación continua recabada en cada iteración fue esencial para realizar ajustes oportunos y perfeccionar el sistema en fases subsiguientes. De esta manera, se aseguró que el producto final no solo cumpliera con los requisitos iniciales, sino que también se adaptara a las expectativas cambiantes de los usuarios, optimizando así la funcionalidad y la recepción del producto.

Recopilación de información

Fuentes de datos

La recopilación de datos para el desarrollo del sistema de detección de caídas se centró principalmente en la captura y análisis de los valores X, Y, Z del acelerómetro en tiempo real de dispositivos portátiles. Estos datos fueron fundamentales para comprender las variaciones de aceleración durante actividades normales y eventos de caída simulados.

Herramientas de recopilación

Para la captura en tiempo real de los valores del acelerómetro, se emplearon herramientas de desarrollo específicas proporcionadas por el entorno de desarrollo integrado (IDE) utilizado. Esto incluyó la visualización de datos en tiempo real directamente desde el sensor del dispositivo, lo cual fue crucial para ajustar y validar el algoritmo de detección de caídas.

Procedimientos de recopilación

Inicialmente, se realizó una fase de visualización de datos en tiempo real para entender el comportamiento del acelerómetro durante diferentes actividades cotidianas y simulaciones de caídas. Posteriormente, se procedió con la implementación iterativa del algoritmo de detección de caídas, que implicó la observación y análisis de cómo los cambios en los datos del acelerómetro podían identificar con precisión eventos de caída. Este proceso permitió ajustar los umbrales y condiciones del algoritmo para mejorar su precisión y eficacia.

Desarrollo y diseño

Planificación

Cada ciclo incluyó objetivos específicos como la implementación inicial del algoritmo de detección de caídas, la creación de interfaces de usuario para alertas y gestión de contactos de emergencia, y la optimización del rendimiento del sistema.

Diseño del sistema

El diseño del sistema se enfocó en la implementación de una arquitectura modular que permitiera la fácil integración de componentes como el algoritmo de detección de caídas, la gestión de notificaciones y la interfaz de usuario. Se utilizó un enfoque basado en componentes para asegurar la escalabilidad y la flexibilidad del sistema frente a futuras actualizaciones y requisitos adicionales.

Tecnología y herramientas

Para el desarrollo del sistema, se emplearon tecnologías como Kotlin para la lógica de aplicación en dispositivos Android Wear, XML para el diseño de interfaces de usuario, y herramientas de desarrollo integradas como Android Studio para la codificación, depuración y pruebas del software. Estas tecnologías fueron seleccionadas por su compatibilidad con dispositivos vestibles y su robustez en el desarrollo de aplicaciones móviles.

Análisis de datos

El análisis de los datos recopilados se centra en evaluar la efectividad del sistema de detección de caídas mediante varios parámetros clave: sensibilidad, precisión, especificidad, verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Estos indicadores son fundamentales para determinar la capacidad del sistema de detectar correctamente las caídas y minimizar las alertas falsas.

Términos

- **Verdaderos positivos:** Caídas detectadas correctamente.
- **Falsos positivos:** Actividades normales incorrectamente identificadas como caídas.
- **Verdaderos negativos:** Actividades normales correctamente identificadas como tales.
- **Falsos negativos:** Caídas no detectadas por el sistema.

Sensibilidad

La sensibilidad del sistema se calcula como la proporción de caídas detectadas correctamente (verdaderos positivos) con respecto al total de caídas simuladas durante las pruebas. Este indicador es crucial para asegurar que el sistema pueda identificar la mayoría de las caídas reales, minimizando así la posibilidad de no detectar eventos críticos.

Precisión

La precisión del sistema se definirá como la proporción de caídas detectadas correctamente (verdaderos positivos y verdaderos negativos) con respecto a todas las detecciones realizadas, ya sean correctas o incorrectas. Este parámetro proporciona una medida de qué tan confiables son las alertas generadas por el sistema, asegurando que las acciones tomadas en respuesta a las alertas sean apropiadas.

Especificidad

La especificidad se calculará como la proporción de actividades normales o no caídas que son correctamente identificadas como tales (verdaderos negativos) con respecto al total de actividades normales simuladas. Esta métrica es esencial para garantizar que el sistema no genere alertas innecesarias durante actividades diarias no riesgosas.

Cronograma de actividades

En el diagrama de Gantt, se realizó una serie de tareas que incluyeron la definición y selección de la tecnología vestible a utilizar, véase Figura 3, Figura 4 y Figura 5, así como la investigación exhaustiva sobre diversas tecnologías y sistemas de ayuda existentes para la detección de caídas. Se estudió sobre arquitecturas, algoritmos, sensores de aceleración y giroscopios. Además, se desarrollaron funcionalidades específicas como la lógica del algoritmo de detección de caídas, sistemas de geolocalización, y mecanismos para evitar falsos positivos y negativos. Se realizaron pruebas de detección de caídas y evaluación del

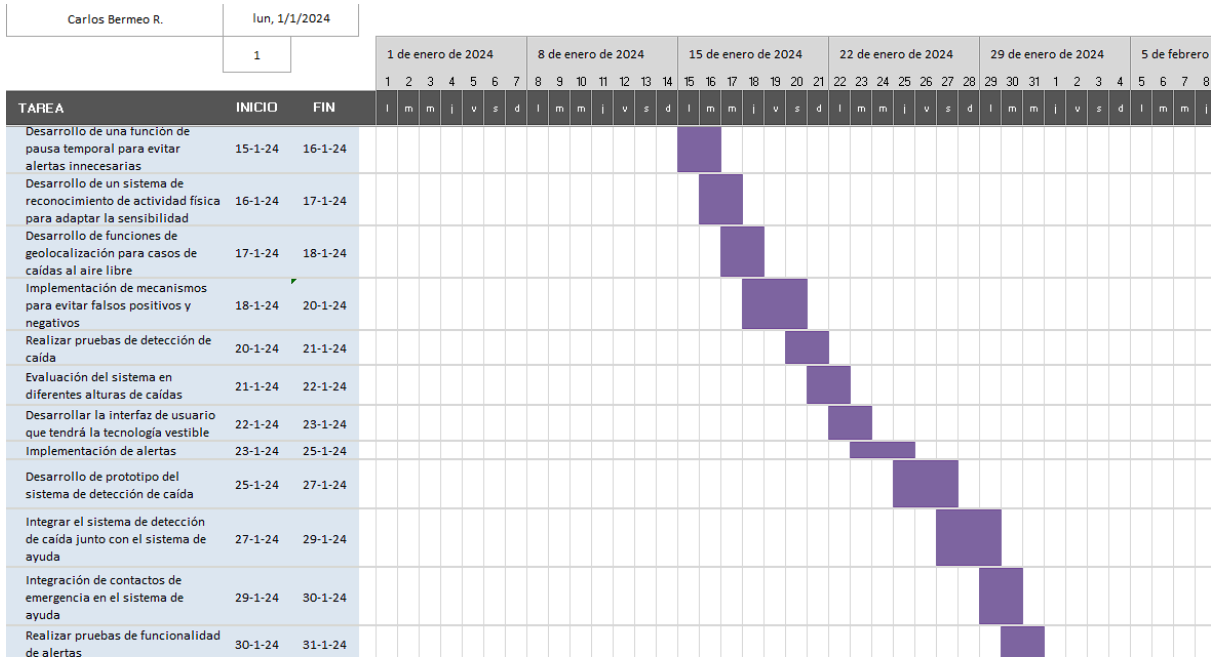
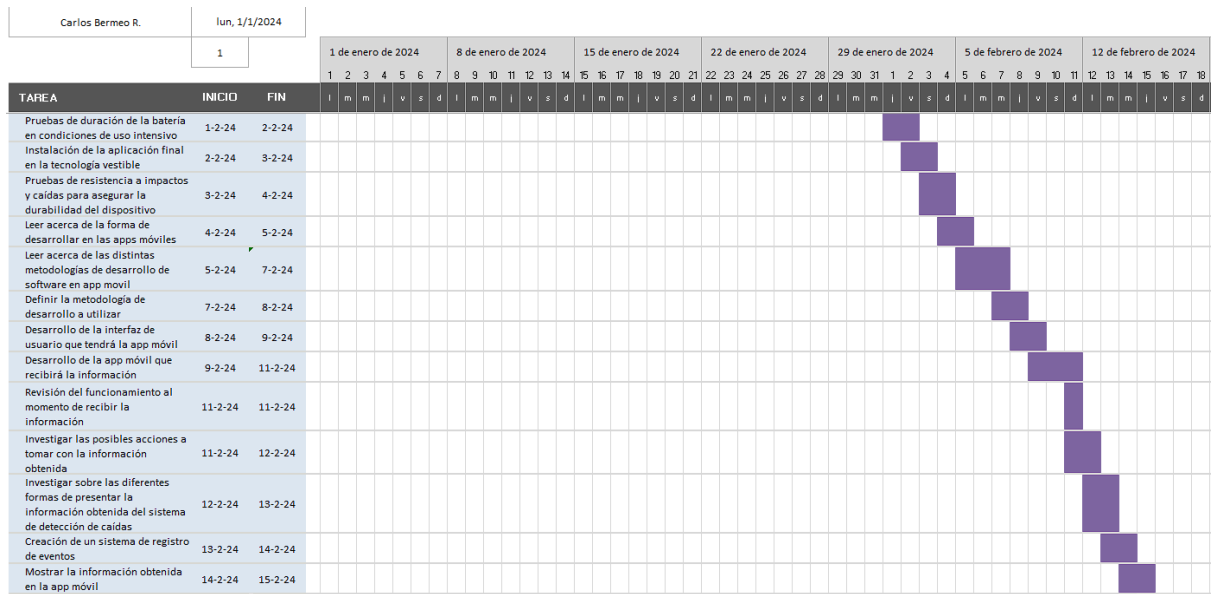


Figura 5

Diagrama de Gantt Semana 5-7



Análisis de resultados

Levantamiento de requerimientos

El primer paso en el levantamiento de requerimientos fue identificar la tecnología vestible adecuada para el proyecto. Se seleccionó el Galaxy Watch 6, gracias a su capacidad para integrarse fácilmente con otros dispositivos de la misma marca y a sus

avanzados sensores de aceleración y giroscopio. La elección de este dispositivo se basó en una cuidadosa evaluación de su compatibilidad, facilidad de uso y aceptación por parte de los usuarios, asegurando que el dispositivo cumpliera con los estándares necesarios para un uso cotidiano fiable y preciso.

Con la tecnología seleccionada, el siguiente enfoque fue comprender en profundidad el funcionamiento de los sensores en la detección de caídas. La precisión del algoritmo dependía de conocer cómo los sensores de aceleración y giroscopio podían distinguir entre caídas y actividades diarias normales. Se investigaron detalladamente los umbrales y las métricas clave, como la magnitud de la aceleración y el cambio en la aceleración vertical, esenciales para la identificación precisa de eventos de caída. Este análisis incluyó una revisión exhaustiva de la literatura y estudios previos sobre algoritmos de detección de caídas para identificar las mejores prácticas y las configuraciones de sensores más efectivas.

Para validar y ajustar estos umbrales teóricos, se llevó a cabo la recolección de información directamente de los sensores del dispositivo. Durante esta fase, se registraron en tiempo real los datos de aceleración en los ejes X, Y y Z, junto con las marcas de tiempo correspondientes. Estos datos se exportaron a un archivo XLS para su análisis detallado. Este proceso permitió mapear los momentos exactos en los que se generaban las caídas y observar cómo los valores de los sensores cambiaban durante estos eventos. Este análisis detallado de los datos fue fundamental para ajustar los umbrales y configurar el algoritmo de detección con mayor precisión.

Diseño

En la Figura 6 se puede apreciar la interfaz principal del sistema, diseñada para ofrecer una experiencia de usuario clara y accesible en el dispositivo vestible. La pantalla presenta dos botones principales que conducen a los módulos esenciales del sistema. El primer botón, etiquetado como "SEND ALERT!", permite a los usuarios generar una alerta

de manera manual, véase Figura 7. Esta funcionalidad es crucial en situaciones donde el usuario anticipa una caída o emergencia y necesita notificar a sus contactos de manera proactiva.

Figura 6

Pantalla principal del sistema

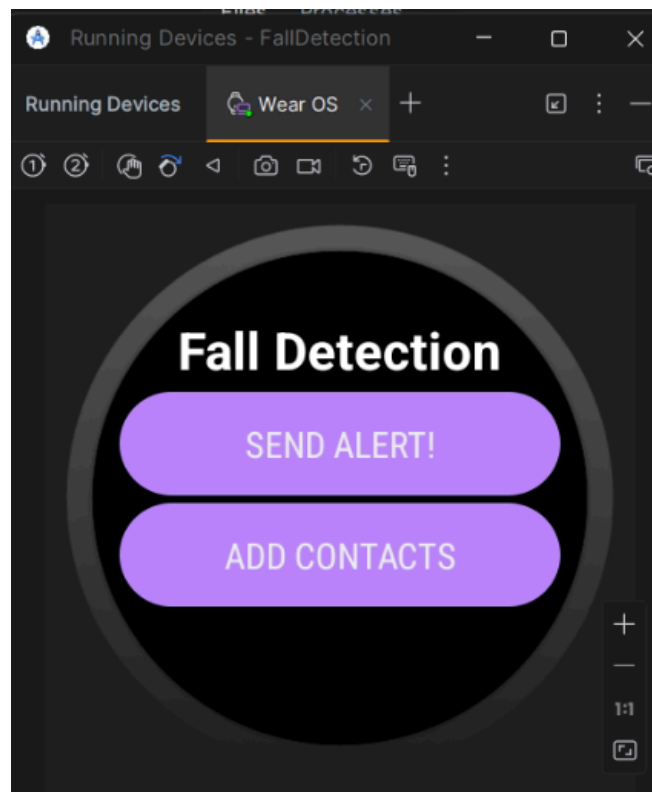
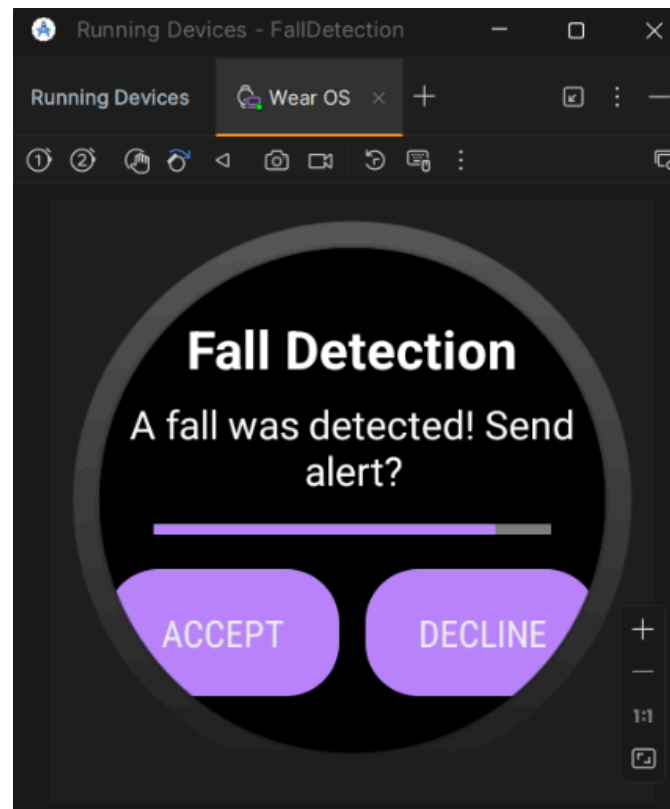


Figura 7

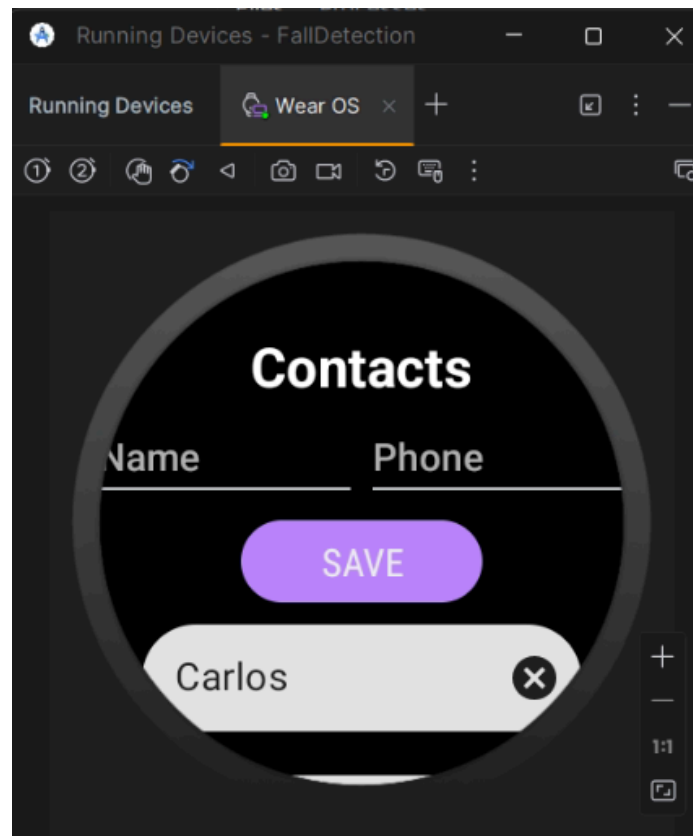
Módulo de alertas



El segundo botón, etiquetado como "ADD CONTACTS", ofrece la posibilidad de gestionar los contactos de emergencia, véase Figura 8. A través de esta opción, los usuarios pueden añadir o eliminar los contactos que recibirán las notificaciones en caso de una emergencia. Este módulo es fundamental para asegurar que la ayuda adecuada esté disponible rápidamente, personalizando la lista de contactos según las necesidades del usuario.

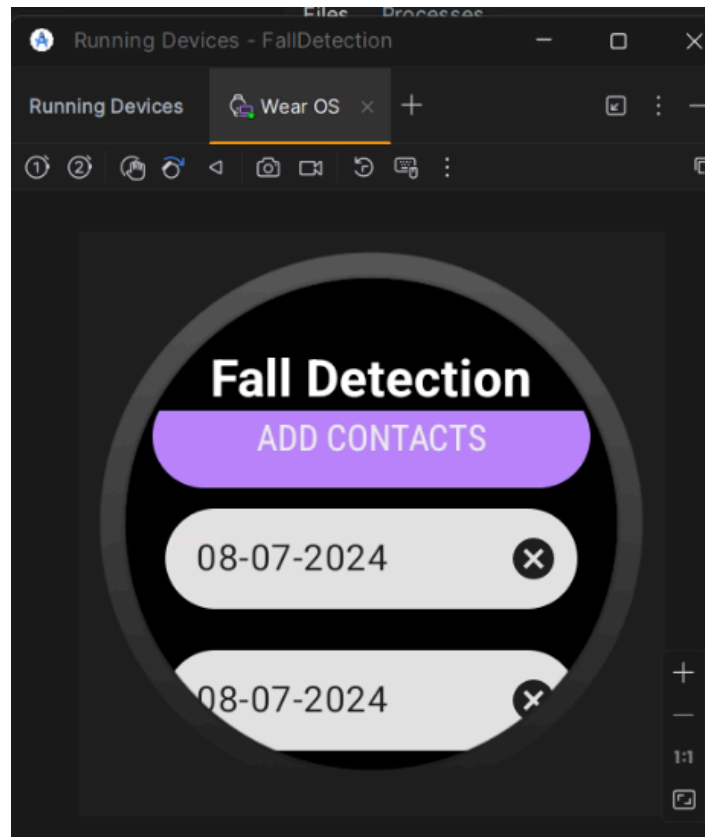
Figura 8

Módulo de contactos de emergencia



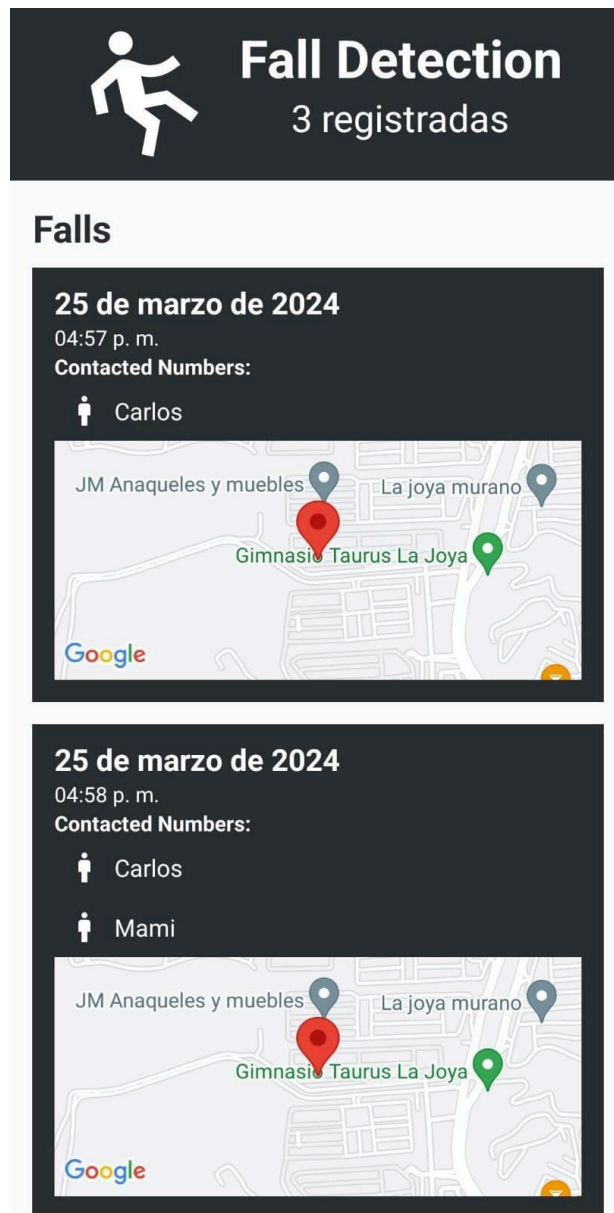
Además, en la misma pantalla principal, se muestra un registro de las caídas detectadas, véase Figura 9, indicando la fecha en la que se identificaron. Esto permite al usuario y a sus contactos de emergencia revisar el historial de eventos, proporcionando información valiosa para el seguimiento y análisis de las caídas. La inclusión de estas fechas facilita la identificación de patrones o frecuencias de caídas, lo que puede ser útil para la toma de decisiones médicas y para ajustar el sistema según las necesidades del usuario.

La disposición y el diseño de estos elementos están pensados para maximizar la facilidad de uso y la accesibilidad, permitiendo una interacción rápida y eficiente con el sistema, incluso en situaciones de emergencia.

Figura 9*Caídas registradas*

El módulo de detección de caídas, tal como se presenta en la Figura 10, está diseñado para proporcionar una interfaz clara y detallada de los eventos de caída registrados por el sistema. Este módulo en el smartphone sirve exclusivamente para obtener información adicional acerca de las caídas registradas anteriormente, facilitando el seguimiento y la gestión de estos eventos tanto para el usuario como para sus contactos de emergencia.

Figura 10*Módulo de visualización - smartphone*



Desarrollo

Algoritmo usado

El algoritmo de detección de caídas se ejecuta dentro de un servicio que monitorea continuamente los eventos del acelerómetro. Cada vez que el acelerómetro detecta un cambio en los movimientos del usuario, el algoritmo calcula las magnitudes y los cambios necesarios, evaluando así las condiciones que indicarían una posible caída. En este contexto, una caída se detecta si se cumplen simultáneamente todas las siguientes condiciones:

$$\Delta A > 10.0 \text{ (umbral de cambio de aceleración)}$$

$$A_{magn} > 20.0 \text{ (umbral de magnitud de aceleración)}$$

$$\Delta a_z < 9.0 \text{ (umbral de cambio de aceleración vertical)}$$

Para calcular el cambio en la magnitud de la aceleración se utiliza la siguiente fórmula:

$$\Delta A = A_{magn} - A_{magn}^{prev}$$

Este cálculo mide la variación en la magnitud de la aceleración entre dos muestras consecutivas del acelerómetro. La magnitud de la aceleración en un momento dado es una medida de la fuerza total ejercida sobre el sensor.

La magnitud de la aceleración se calcula utilizando la fórmula:

$$A_{magn} = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

La fórmula de la magnitud de la aceleración se basa en los principios de la geometría euclidiana y representa la distancia euclidiana en el espacio tridimensional. Cada uno de los componentes a_x , a_y y a_z , corresponde a la aceleración medida en los ejes x, y, y z, respectivamente, en un sistema de coordenadas cartesianas. Según la segunda ley de Newton, la aceleración total experimentada por un objeto es el resultado de todas las fuerzas actuando sobre él. En el caso de aplicaciones como la detección de caídas, es más práctico trabajar con una medida única de aceleración total en lugar de analizar las componentes individuales por separado. La magnitud de la aceleración proporciona precisamente eso: una medida escalar que captura la intensidad del movimiento, sin tener en cuenta su dirección. Esto resulta crucial para evaluar eventos como caídas, donde la fuerza total del impacto es de mayor interés que la dirección específica del movimiento.

Para calcular el cambio en la aceleración vertical se utiliza la siguiente fórmula

$$\Delta a_z = a_z - a_z^{prev}$$

El concepto de cambio en la aceleración vertical se deriva de la física newtoniana y la ley de la inercia. Según esta ley, cualquier cambio en el movimiento de un objeto está relacionado con la acción de una fuerza externa. En el caso de una caída, la fuerza gravitatoria y la resistencia del aire (en el caso de una caída libre) provocan una desaceleración rápida al impactar con una superficie, lo que se refleja en un cambio brusco en la aceleración vertical.

Las medidas de umbral elegidas para la detección de caídas se basan en la necesidad de equilibrar la sensibilidad y la especificidad del algoritmo para asegurar una detección precisa de caídas mientras se minimizan las falsas alarmas.

Cambio de aceleración

$$\Delta A > 10.0$$

Este umbral se selecciona para detectar cambios significativos en la aceleración. Un cambio de aceleración de 10 m/s² o más suele indicar un evento brusco, como una caída. Este valor se elige para distinguir entre los movimientos cotidianos normales y aquellos que son abruptos y potencialmente peligrosos, como una caída. Ajustar este umbral ayuda a evitar que movimientos menores y cotidianos generen falsas alarmas, mientras que sigue siendo lo suficientemente sensible como para detectar una caída real.

Magnitud de aceleración

$$A_{\text{magn}} > 20.0$$

La magnitud total de la aceleración que supera los 20 m/s² sugiere un impacto significativo. Durante una caída, es común experimentar una aceleración total elevada debido a la combinación de la aceleración de la gravedad y la desaceleración súbita al impactar con el suelo u otra superficie. Este umbral ayuda a asegurar que solo los eventos con suficiente fuerza se consideren caídas, eliminando así la posibilidad de que movimientos más suaves, como agacharse o sentarse rápidamente, sean clasificados como caídas.

Cambio de aceleración vertical

$$\Delta a_z < 9.0$$

Un cambio en la aceleración vertical menor que 9 m/s² se utiliza para detectar la desaceleración rápida típica al golpear el suelo. Durante una caída, la aceleración vertical cambia bruscamente cuando el cuerpo deja de moverse verticalmente al impactar. Este umbral se elige para capturar ese cambio rápido, asegurando que el algoritmo detecte el impacto de la caída. Este umbral también ayuda a diferenciar entre una caída y otros movimientos rápidos que no resultan en un impacto significativo.

Código del algoritmo

Una vez que se han definido los valores necesarios, se establecen como constantes en el código, tal como se puede observar en la Figura 11. Estas constantes incluyen los umbrales de cambio de aceleración, magnitud de aceleración y aceleración vertical, que son cruciales para la detección de caídas. Junto a estas constantes, se inicializan variables encargadas de almacenar tanto la magnitud de aceleración previa como la aceleración vertical previa. Estas variables son fundamentales para comparar los valores actuales con los anteriores, permitiendo así determinar si se ha producido una caída basándose en los umbrales establecidos.

Figura 11

Magnitudes definidas y aceleraciones

```
//Magnitudes
private var previousAccelerationMagnitude = 0.0f
private var previousVerticalAcceleration = 0.0f
private val CHANGE_THRESHOLD = 10.0f
private val MAGNITUDE_THRESHOLD = 20.0f
private val VERTICAL_THRESHOLD = 9.0f
```

El algoritmo implementado en el objeto `sensorEventListener`, véase Figura 12, se encarga de monitorear continuamente los eventos del sensor de aceleración para detectar posibles caídas. Cuando se detecta un cambio en el sensor, el método `onSensorChanged` se ejecuta. Primero, se verifica si la precisión del sensor es confiable; si no lo es, el método retorna inmediatamente sin realizar más acciones.

A continuación, se obtiene la aceleración vertical actual del eje Z (`verticalAcceleration`) y se calcula la magnitud de la aceleración total utilizando la función `calculateMagnitude`. Se determina el cambio en la aceleración calculando la diferencia entre la magnitud de la aceleración actual y la magnitud de la aceleración previa (`accelerationChange`). De manera similar, se calcula el cambio en la aceleración vertical comparando la aceleración vertical actual con la previa (`verticalAccelerationChange`).

El algoritmo evalúa tres condiciones para determinar si se ha detectado una caída: el cambio en la aceleración debe ser mayor que un umbral predefinido (`CHANGE_THRESHOLD`), la magnitud actual de la aceleración debe superar otro umbral (`MAGNITUDE_THRESHOLD`), y el cambio en la aceleración vertical debe ser menor que un tercer umbral (`VERTICAL_THRESHOLD`). Si todas estas condiciones se cumplen y la actividad de alerta no está en ejecución (`!FallDetectionManager.isAlertActivityRunning()`), se activa la alerta de caída mediante la función `triggerFallAlert`.

Finalmente, los valores de aceleración vertical previa y magnitud de aceleración previa se actualizan con los valores actuales para ser utilizados en la siguiente iteración.

Figura 12

Algoritmo de detección de caída

```

private val sensorEventListener = object : SensorEventListener{
    override fun onSensorChanged(event: SensorEvent) {
        if(event.accuracy == SensorManager.SENSOR_STATUS_UNRELIABLE){
            return;
        }

        val verticalAcceleration = event.values[2]
        val currentAccelerationMagnitude = calculateMagnitude(event.values)
        val accelerationChange = currentAccelerationMagnitude - previousAccelerationMagnitude
        val verticalAccelerationChange = verticalAcceleration - previousVerticalAcceleration

        val fallDetected = (accelerationChange > CHANGE_THRESHOLD &&
            currentAccelerationMagnitude > MAGNITUDE_THRESHOLD &&
            verticalAccelerationChange < VERTICAL_THRESHOLD)

        if (fallDetected && !FallDetectionManager.isAlertActivityRunning()) {
            triggerFallAlert()
        }

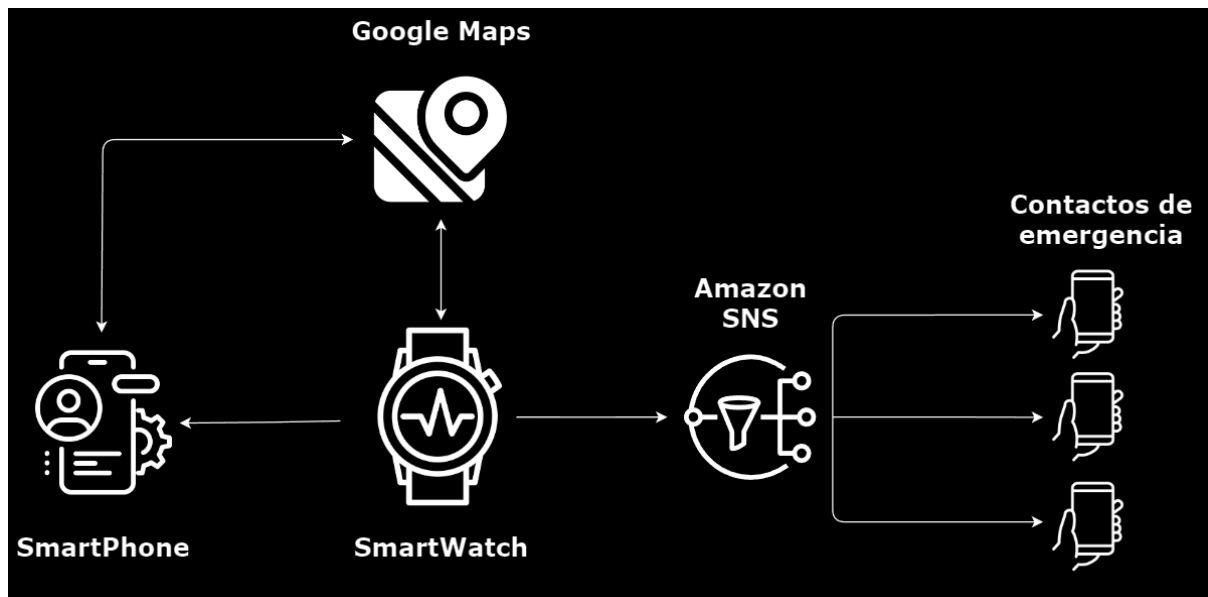
        previousVerticalAcceleration = verticalAcceleration
        previousAccelerationMagnitude = currentAccelerationMagnitude
    }

    override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {
        //None
    }
}

```

Arquitectura del sistema

La arquitectura propuesta en esta investigación, véase Figura 13, se enfoca en el desarrollo de un sistema integral de monitoreo y notificación de caídas, utilizando dispositivos inteligentes conectados y servicios en la nube.

Figura 13*Arquitectura del software*

El smartwatch, como componente central del sistema, emplea sensores avanzados para detectar caídas y activar automáticamente un proceso de geolocalización a través de Google Maps. Esta funcionalidad no solo permite obtener la ubicación precisa del usuario en caso de emergencia, sino que también facilita la rápida notificación a través de Amazon SNS (Simple Notification Service).

Cuando el smartwatch detecta una caída, se inicia una solicitud POST a Amazon SNS, que está configurado para gestionar este tipo de alertas. Amazon SNS, en respuesta, distribuye de manera eficiente la información relevante sobre la caída y la ubicación del usuario a los contactos de emergencia designados, utilizando SMS como canal de comunicación. Esta capacidad de distribución instantánea garantiza que los contactos pertinentes estén informados rápidamente y puedan actuar con prontitud para asistir al usuario en problemas.

Además de la detección automática de caídas, al abrir la aplicación complementaria en el smartphone se realiza consultas al smartwatch para obtener actualizaciones sobre cualquier incidente reciente.

Si se detecta una caída, la aplicación muestra los detalles pertinentes junto con un mapa actualizado proporcionado nuevamente por Google Maps.

En este proyecto, el uso de MessageClient permite implementar una arquitectura basada en eventos, donde la información se envía y recibe únicamente cuando es necesario, optimizando así la comunicación entre el smartwatch y el smartphone.

Código del smartwatch

El MessageClient en el smartwatch está configurado para escuchar mensajes entrantes a través del método onMessageReceived. Cuando se recibe un mensaje con la ruta "/request_fall_data", se desencadena un evento que indica que el smartphone ha solicitado datos de caídas, véase Figura 14. En respuesta, el smartwatch llama al método sendFallDataToPhone, el cual convierte la lista de datos de caídas (fallList) a formato JSON y la envía al smartphone a través de sendMessage.

Figura 14

Envío de información mediante MessageClient

```
override fun onMessageReceived(messageEvent: MessageEvent) {  Pegassuo
    if (messageEvent.path == "/request_fall_data") {
        Log.d( tag: "Wear", msg: "Received request for fall data")
        val dataFallList = storeData.getData( context: this, DataFall::class.java)
        if (dataFallList != null) {
            sendFallDataToPhone(dataFallList)
        }
    }
}

private fun sendFallDataToPhone(fallList: List<DataFall>){  Pegassuo *
    val request = PutDataMapRequest.create("/fall_data")
    val jsonString = Gson().toJson(fallList)
    val requestMessage = request.asPutDataRequest()
    mMessageClient.sendMessage(
        requestMessage.uri.host.toString(),
        requestMessage.uri.path.toString(),
        jsonString.toByteArray(StandardCharsets.UTF_8)
    )
}
```

Código del smartphone

Para iniciar la comunicación, el smartphone puede solicitar los datos de caídas al smartwatch mediante el método `requestFallDataFromWear`. Este método crea una solicitud con la ruta `"/request_fall_data"` y envía el mensaje, tal como la demuestra la Figura 15. Esta solicitud desencadena el evento en el smartwatch para enviar los datos de caídas de vuelta al smartphone.

Figura 15

Creación de solicitud de datos

```
private fun requestFallDataFromWear() {  Pegassuo *
    val request = PutDataMapRequest.create("/request_fall_data")
    val requestMessage = request.asPutDataRequest()
    mMessageClient.sendMessage(
        requestMessage.uri.host.toString(),
        requestMessage.uri.path.toString(),
        requestMessage.data
    )
}
```

El `MessageClient` también escucha mensajes entrantes a través del método `onMessageReceived`, véase Figura 16. Cuando se recibe un mensaje con la ruta `"/fall_data"`, se procesa el mensaje convirtiendo los datos JSON de vuelta a una lista de objetos `DataFall`. Estos datos se guardan en un archivo local mediante `saveFallDataToLocalFile`, y la interfaz de usuario se actualiza con los nuevos datos de caídas a través del adaptador de la lista.

Figura 16

Recepción de la información

```

override fun onMessageReceived(messageEvent: MessageEvent) {  Pegassuo
    if (messageEvent.path == "/fall_data") {
        val message = String(messageEvent.data, StandardCharsets.UTF_8)
        val dataList = Gson().fromJson(message, Array<DataFall>::class.java).toList()
        saveFallDataToLocalFile(dataList)
        fallList = dataList
        fallAdapter.updateData(fallList)
    }
}

private fun requestFallDataFromWear(){  Pegassuo *
    val request = PutDataMapRequest.create("/request_fall_data")
    val requestMessage = request.asPutDataRequest()
    mMessageClient.sendMessage(
        requestMessage.uri.host.toString(),
        requestMessage.uri.path.toString(),
        requestMessage.data
    )
}

```

Envío de alertas mediante SMS

Cuando se envía un mensaje SMS utilizando Amazon Simple Notification Service (SNS), el proceso implica varios pasos coordinados para garantizar la entrega eficiente del mensaje al destinatario, véase Figura 17. Inicialmente, el remitente configura un mensaje a través de la consola de administración de AWS o mediante la API de SNS, especificando el contenido del mensaje y el número de teléfono del destinatario.

Una vez que el mensaje es enviado, SNS lo recibe y gestiona su enrutamiento hacia los proveedores de servicios SMS designados. Amazon SNS coordina con estos proveedores, que pueden ser operadores de telecomunicaciones o plataformas de mensajería SMS, para entregar el mensaje al número de teléfono especificado. Durante

este proceso, SNS monitorea el estado del mensaje, registrando si fue entregado con éxito o si ocurrió algún error. Para optimizar la entrega, SNS implementa políticas de reintentos automáticos en caso de fallos temporales, asegurando que el mensaje se entregue incluso en condiciones de red variables.

Figura 17

Envío de SMS

```
suspend fun pubTextSMS(messageVal: String, phoneNumberVal: String){ ▲ Pegassuo *
    val dotenv = dotenv {
        directory = "/assets"
        filename = "env"
    }

    val stacticCredentials = StaticCredentialsProvider{
        accessKeyId = dotenv["ACCESSKEYID"]
        secretAccessKey = dotenv["SECRETACCESSKEY"]
    }

    val attributes = mutableMapOf<String, MessageAttributeValue>()

    attributes["AWS.SNS.SMS.SMSType"] = MessageAttributeValue.invoke {
        stringValue = "Promotional"
        dataType = "String"
    }

    val request = PublishRequest{
        message = messageVal
        phoneNumber = phoneNumberVal
        messageAttributes = attributes
    }

    SnsClient{
        credentialsProvider = stacticCredentials
        region = "sa-east-1"
    }.use { snsClient →
        snsClient.publish(request)
    }
}
```


El código de la Figura 17 es una función suspendida que envía un mensaje SMS utilizando el servicio AWS SNS. Primero, se carga el archivo de configuración .env que contiene las credenciales necesarias para acceder a los servicios de AWS. Estas credenciales incluyen el ACCESSKEYID y SECRETACCESSKEY, que son esenciales para la autenticación y autorización.

Una vez obtenidas las credenciales, se configuran como un StaticCredentialsProvider, asegurando que el cliente SNS pueda autenticarse correctamente. Luego, se definen los atributos del mensaje, específicamente el tipo de SMS, que en este caso está configurado como "Promotional". Estos atributos se almacenan en un mapa para ser incluidos en la solicitud de publicación.

El siguiente paso es crear una solicitud de publicación (PublishRequest), que contiene el mensaje a enviar, el número de teléfono del destinatario, y los atributos del mensaje previamente definidos. Finalmente, se inicializa el cliente SNS con las credenciales y la región especificada. En este caso para las pruebas realizadas el servicio AWS requería que la región sea aquella en la que los destinatarios se encontraban, en un ambiente de producción se puede hacer uso de cualquier región

El cliente SNS se utiliza para publicar la solicitud de mensaje, enviando el SMS al número de teléfono especificado. Al usar use, se asegura que el cliente SNS se cierre correctamente después de que la operación de envío haya concluido. Esta estructura permite una gestión segura y eficiente de los recursos mientras se realiza la operación de envío de SMS.

Obtención de la geolocalización

En Wear OS, la obtención de la posición geográfica se facilita a través de la integración de tecnologías como el GPS (Global Positioning System) y otros métodos de geolocalización disponibles en los dispositivos portátiles. Cuando una aplicación en un smartwatch necesita determinar la ubicación actual del usuario, utiliza los sensores de

ubicación disponibles en el dispositivo, para más información sobre cómo se proporcionan los datos de ubicación en Wear OS se recomienda visitar el Anexo 2.- Obtener Ubicación En Wear OS.

El smartwatch puede emplear el GPS incorporado para obtener coordenadas precisas de longitud y latitud. Este método es ideal para entornos al aire libre y áreas abiertas donde el dispositivo pueda recibir señales GPS satelitales de manera directa. Además del GPS, algunos smartwatches también pueden utilizar redes Wi-Fi cercanas y datos de torres de telefonía móvil para mejorar la precisión de la ubicación cuando la señal GPS es limitada, como en interiores o áreas urbanas densamente pobladas donde la recepción satelital puede ser intermitente.

Una vez que el smartwatch obtiene las coordenadas de ubicación, estas pueden ser utilizadas directamente por la aplicación para diversos fines, como notificar la ubicación en caso de emergencia, registrar la ruta en aplicaciones de fitness o proporcionar indicaciones en aplicaciones de mapas. La precisión y disponibilidad de la geolocalización en Wear OS depende tanto de la calidad de los sensores del dispositivo como de la infraestructura de soporte, como la conectividad de red y la disponibilidad de señales satelitales.

Versionamiento de la aplicación

El versionamiento del sistema se gestiona mediante GitHub, aprovechando sus funcionalidades para asegurar un desarrollo colaborativo y controlado. Todos los cambios en el código se registran mediante commits que describen de manera clara y concisa las modificaciones realizadas en cada iteración del proyecto. Cada commit está asociado a una tarea específica o a la solución de un problema identificado, lo que facilita la trazabilidad y el seguimiento de los avances realizados, el historial de versiones del proyecto puede ser revisado en el Anexo 4.- Historial De Versiones.

Pruebas

En la fase de pruebas, se evaluó exhaustivamente la funcionalidad y precisión del sistema de detección de caídas. Para ello, los participantes realizaron una serie de actividades de la vida diaria (ADL) y simulaciones de caídas, siguiendo un protocolo basado en los resultados exitosos del estudio KFall (Yu, Jang, & Xiong, 2021). Las actividades incluyeron:

- Estar de pie
- Caminar
- Correr
- Trotar
- Sentarse en una silla
- Levantarse de una silla
- Caerse hacia adelante de pie
- Caerse hacia atrás de pie
- Caerse hacia adelante al sentarse/levantarse
- Caerse hacia atrás al sentarse/levantarse
- Caerse al tropezarse caminando
- Caerse al tropezarse trotando
- Caerse al tropezarse corriendo
- Acostarse
- Subir y bajar escaleras

Cada uno de estos ejercicios fue realizado cinco veces por cada uno de los cuatro participantes, asegurando una recopilación de datos robusta y diversa. El propósito de incluir tanto actividades cotidianas como simulaciones de caídas fue garantizar que el sistema pudiera diferenciar de manera precisa entre movimientos normales y eventos de caída. En dado caso de no contar con un smartwatch se puede hacer uso del emulador integrado en Android Studio, véase Anexo 5.- Pruebas De Funcionalidad En Emulador

Se realizaron un total de 140 simulaciones de caídas, con cada participante ejecutando siete tipos diferentes de caídas, cinco veces cada uno. Paralelamente, se llevaron a cabo 160 actividades normales, distribuidas entre los cuatro participantes, cada uno realizando ocho tipos de actividades normales, también cinco veces cada una.

De las 140 caídas simuladas, el sistema logró detectar correctamente 130, resultando en una sensibilidad del 92.9%. La sensibilidad, también conocida como recall, mide la capacidad del sistema para identificar caídas reales y se calcula con la fórmula:

$$\text{Sensibilidad} = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos negativos}}$$

En este caso, la sensibilidad se calcula como:

$$\text{Sensibilidad} = \frac{130}{140} = 92.9\%$$

En cuanto a las actividades normales, de las 160 realizadas, el sistema identificó correctamente 145 como no caídas, lo que contribuye a una especificidad del 90.6%. La especificidad mide la capacidad del sistema para distinguir correctamente las actividades cotidianas de las caídas, minimizando las falsas alarmas, y se calcula con la fórmula:

$$\text{Especificidad} = \frac{\text{Verdaderos negativos}}{\text{Verdaderos negativos} + \text{Falsos positivos}}$$

El resultado de este cálculo fue:

$$\text{Especificidad} = \frac{145}{160} = 90.6\%$$

La precisión del sistema, que mide la proporción de verdaderos positivos entre todas las detecciones de caídas, se calculó en un 89.66%. Esta métrica es crucial para entender la exactitud del sistema en la identificación de caídas verdaderas, considerando tanto los verdaderos positivos como los falsos positivos, y se calcula con la fórmula:

$$\text{Precisión} = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos positivos}}$$

Como resultado se obtuvo:

$$\text{Precisión} = \frac{130}{145} = 89.66\%$$

Además, se calculó la tasa de falsos positivos, que resultó ser del 9.4%. Este indicador muestra la proporción de actividades normales que fueron incorrectamente identificadas como caídas, proporcionando una visión sobre el nivel de falsas alarmas que genera el sistema, y se calcula con la fórmula:

$$\text{Tasa de Falsos positivos} = \frac{\text{Falsos positivos}}{\text{Verdaderos negativos} + \text{Falsos positivos}}$$

Como resultado tenemos

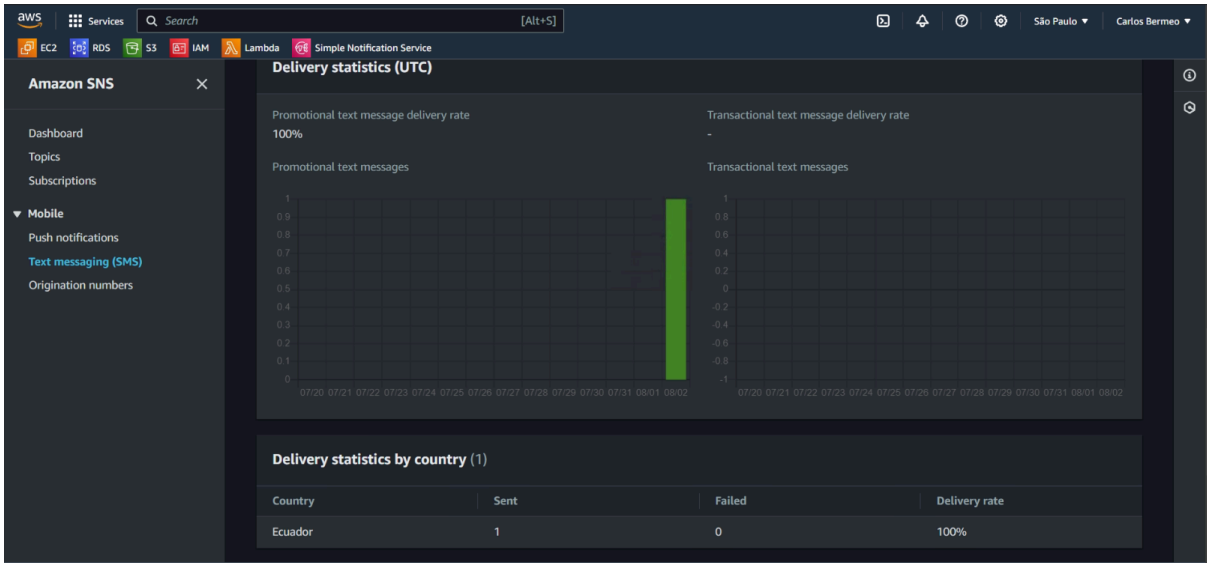
$$\text{Tasa de Falsos positivos} = \frac{15}{160} = 9.4\%$$

Para realizar las pruebas de envío de mensajes, se utiliza la consola proporcionada por el servicio AWS SNS. En esta consola, es posible configurar los números de teléfono que se utilizarán para las pruebas o suministrarlos directamente desde el smartwatch, incluso en modo sandbox. AWS SNS ofrece un modo sandbox con un límite de mensajes a enviar, donde los números de teléfono deben ser verificados previamente mediante códigos de autenticación. Como se muestra en la Figura 18, los mensajes aparecen clasificados como transaccionales, ya que así fueron configurados en el código del smartwatch, y se envían de manera exitosa.

Además, la consola de AWS SNS permite rastrear a qué países se han enviado los mensajes y obtener estadísticas sobre la cantidad de mensajes exitosos y fallidos. Esta funcionalidad es útil para monitorear y asegurar el correcto funcionamiento del sistema de mensajería. Una vez que se migre al entorno de producción, los números de teléfono serán gestionados directamente por el smartwatch y distribuidos automáticamente por el servicio AWS SNS, eliminando la necesidad de verificaciones manuales y aumentando la eficiencia del sistema.

Figura 18

Consola de AWS SNS



Conclusiones

En la evaluación del sistema de detección de caídas, se obtuvieron resultados que muestran una alta efectividad en la identificación de caídas y una capacidad aceptable para minimizar falsas alarmas. Específicamente, el sistema alcanzó una sensibilidad del 92.9%, una precisión del 89.66% y una especificidad del 90.6%. Aunque estos resultados son buenos, se observó una diferencia significativa en comparación con estudios previos que reportaron una sensibilidad del 99.32% y una especificidad del 99.01%. Esta diferencia puede atribuirse al uso de un dispositivo de propósito general, lo cual permite que la aplicación sea compatible con una amplia variedad de dispositivos Android, pero limita la capacidad del dispositivo para dedicarse exclusivamente a la detección de caídas.

La elección de utilizar dispositivos de propósito general, en lugar de dispositivos específicos optimizados para la detección de caídas, permite que la aplicación sea accesible y utilizable en una mayor variedad de entornos y por un mayor número de usuarios. Esta flexibilidad es una ventaja significativa, ya que permite configurar umbrales específicos para diferentes dispositivos y condiciones de uso, adaptando el sistema a diversas situaciones. Sin embargo, esta misma flexibilidad puede resultar en una menor eficiencia en comparación con dispositivos dedicados exclusivamente a esta tarea.

Además, el sistema demostró ser robusto y fiable en el envío de mensajes de emergencia siempre y cuando exista conectividad a internet, asegurando que los contactos de emergencia sean notificados oportunamente. La visualización de la información en el smartphone también fue exitosa, permitiendo a los usuarios revisar el historial de caídas y ver la ubicación de cada incidente en un mapa de manera efectiva. Estos resultados, junto con la alta sensibilidad y precisión del sistema, validan la efectividad y confiabilidad del sistema en diversas condiciones y escenarios de uso, ofreciendo una solución práctica y accesible para la detección de caídas.

Recomendaciones

Para mejorar la efectividad del sistema de detección de caídas, sería beneficioso explorar el uso de algoritmos de aprendizaje automático optimizados específicamente para dispositivos vestibles. Estos algoritmos, especialmente los modelos de aprendizaje profundo, pueden ser entrenados con grandes volúmenes de datos sobre caídas y actividades normales, lo que permitiría una detección más precisa y adaptativa. Utilizar aprendizaje automático puede ayudar a identificar patrones complejos en los datos de aceleración y giroscopio, superando las limitaciones de los umbrales estáticos actuales y adaptándose mejor a las variaciones individuales de los usuarios.

Otra recomendación clave es realizar un análisis continuo y detallado de los datos recopilados durante las pruebas y el uso en el mundo real. Almacenar estos datos en una base de datos centralizada facilitaría la aplicación de técnicas avanzadas de análisis de datos, como el análisis de series temporales. Esto permitiría identificar tendencias y ajustar los umbrales de detección para mejorar la precisión del sistema. Además, implementar un sistema de retroalimentación que permita a los usuarios reportar falsos positivos y negativos podría ayudar a ajustar el algoritmo de detección en función de esta retroalimentación, mejorando su fiabilidad y exactitud.

Para asegurar que el sistema de envío de mensajes funcione de manera confiable incluso en situaciones de conectividad limitada, se podrían considerar mejoras en la infraestructura de conectividad. Tecnologías como el almacenamiento en caché de mensajes y el envío diferido cuando la conexión se restablezca podrían aumentar la robustez del sistema. Estas mejoras garantizarían que los contactos de emergencia sean notificados oportunamente, incluso si la conexión a internet se interrumpe temporalmente. Además, integrar opciones de notificación alternativas, como mensajes SMS directos, podría mejorar la eficacia en escenarios de baja conectividad.

Finalmente, se recomienda mejorar la interfaz de usuario en el smartphone para hacer la visualización de la información más intuitiva y detallada. Proporcionar a los usuarios y a sus contactos de emergencia una mejor comprensión de los eventos de caída, incluyendo recomendaciones claras sobre las acciones a seguir, puede aumentar significativamente la usabilidad y la efectividad del sistema en situaciones críticas. Adicionalmente, añadir funcionalidades como el seguimiento en tiempo real y la integración con otros dispositivos de salud podría ofrecer una experiencia más completa y enriquecedora para los usuarios, potenciando la utilidad del sistema en la vida cotidiana.

Referencias

Amazon. (5 de Enero de 2024). AWS. Obtenido de AMAZON:

https://docs.aws.amazon.com/es_es/sns/latest/dg/welcome.html

Ayoung-Chee, P., McIntyre, L., E Ebal, B., & McCormick, W. (Febrero de 2014). National Library of Medicine. Obtenido de PubMed:

<https://pubmed.ncbi.nlm.nih.gov/24458057/>

B., G. (10 de Enero de 2023). Hostinger. Obtenido de Hostinger:

<https://www.hostinger.es/tutoriales/que-es-github>

CEPAL. (25 de Septiembre de 2015). CEPAL. Obtenido de CEPAL:

<https://repositorio.cepal.org/server/api/core/bitstreams/cb30a4de-7d87-4e79-8e7a-ad5279038718/content>

de Moura, A. (23 de Noviembre de 2023). NYSPINE. Obtenido de New York Spine Institute:

<https://www.nyspine.com/blog/que-buscar-despues-de-un-resbalon-y-una-caida/?lang=es>

Fernández, R. (09 de Agosto de 2023). Statista. Recuperado el 27 de Septiembre de 2023, de

<https://es.statista.com/temas/10145/industria-y-consumo-mundial-de-smartphones/#topicOverview>

Fundación Mapfre. (Mayo de 2022). Fundación Mapfre. Obtenido de Fundación Mapfre:

<https://documentacion.fundacionmapfre.org/documentacion/publico/es/media/group/115317.do>

Gac Espínola, H. (28 de Noviembre de 2023). Pontificia Universidad Católica de Chile.

Obtenido de Facultad de medicina:

<https://medicina.uc.cl/publicacion/caidas-adulto-mayor/>

Google. (4 de Enero de 2024). Android Developer. Obtenido de Android Developer:

<https://developer.android.com/training/wearables/data/data-layer?hl=es-419>

He, W., Goodkind, D., & Kowal, P. (Marzo de 2016). CENSUS. Obtenido de CENSUS:

<https://www.census.gov/content/dam/Census/library/publications/2016/demo/p95-16-1.pdf>

MICROSEGUR. (19 de Julio de 2023). MICROSEGUR. Obtenido de ADVANCED

SECURITY SOLUTIONS:

<https://microsegur.com/que-es-un-sensor-de-movimiento-pir/>

Ministerio de Defensa Nacional del Ecuador. (25 de Enero de 2021). Ministerio de Defensa

Nacional del Ecuador. Obtenido de Ministerio de Defensa Nacional del Ecuador:

https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/02/Constitucion-de-la-Republica-del-Ecuador_act_ene-2021.pdf

NIA. (15 de Julio de 2023). National Institute on Aging. Obtenido de Instituto Nacional del

envejecimiento: <https://www.nia.nih.gov/espanol/caida/prevenga-caidas-fracturas>

OMS. (26 de 4 de 2021). OMS. Obtenido de Organización Mundial de la Salud:

<https://www.who.int/es/news-room/fact-sheets/detail/falls>

OPS. (28 de Noviembre de 2023). aesculapseguridaddelpaciente. Obtenido de

aesculapseguridaddelpaciente:

<http://www.aesculapseguridaddelpaciente.org.mx/docs/seguridad-del-paciente/accion-esencial6/Guia-de-diagnostico-y-manejo-de-caidas.pdf>

Pérez, A. (16 de Agosto de 2016). OBS. Obtenido de Business School:

<https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental>

Queralt, M. (3 de Noviembre de 2021). Salud - Blogs Mapfre. Obtenido de mapfre:

<https://www.salud.mapfre.es/salud-familiar/mayores/enfermedades-mayores/factores-de-riesgo/>

Ramón, J. A. (Diciembre de 2021). riuma. Recuperado el 4 de Octubre de 2023, de https://riuma.uma.es/xmlui/bitstream/handle/10630/24216/TD_SANTOYO_RAMON_Jose_Antonio.pdf?sequence=1&isAllowed=y

Repanovici, R., & Nedelcu, A. (2021). iopscience. Recuperado el 27 de Septiembre de 2023], de <https://iopscience.iop.org/article/10.1088/1757-899X/1009/1/012049/pdf>

Sposaro, F., & Tyson, G. (Septiembre de Septiembre de 2009). fsu. Obtenido de fsu: <https://ww2.cs.fsu.edu/~sposaro/mobile/presentations/iFall.pdf>

SYDLE. (26 de Septiembre de 2023). SYDLE. Obtenido de SYDLE: <https://www.sydle.com/es/blog/api-6214f68876950e47761c40e7>

Trokko. (19 de Mayo de 2021). SAMSUNG. Obtenido de SAMSUNG: <https://eu.community.samsung.com/t5/wearables/samsung-se-despide-de-tizen/td-p/3448112>

worldlifeexpectancy. (2020). WORDLIFEEXPECTANCY. Obtenido de WORDLIFEEXPECTANCY: <https://www.worldlifeexpectancy.com/es/ecuador-falls>

Yoshida. (Enero de 2005). ResearchGate. Obtenido de ResearchGate: https://www.researchgate.net/publication/287854544_A_wearable_computer_system_for_a_detection_and_prevention_of_elderly_users_from_falling

Yu, X., Jang, J., & Xiong, S. (16 de Julio de 2021). National Library of Medicine. doi:10.3389/fnagi.2021.692865

Anexos

Anexo 1.- Envío de la información entre dispositivos

En el proyecto, se emplea MessageClient para facilitar la comunicación entre un smartwatch y un smartphone de manera eficiente y confiable. Este componente se utiliza específicamente para enviar y recibir datos estructurados entre dispositivos, en este caso, para manejar eventos de detección de caídas y la sincronización de información.

El motivo principal por el cual se opta por utilizar MessageClient sobre otras alternativas radica en su capacidad para gestionar la comunicación asincrónica de manera robusta dentro del ecosistema de Wear OS. A diferencia de otras opciones de comunicación, como DataClient que se enfoca en la sincronización de datos persistentes o ChannelClient que maneja flujos continuos de información, MessageClient es ideal para situaciones donde se requiere enviar notificaciones instantáneas y eventos discretos entre dispositivos de forma directa y rápida.

En este contexto específico, el MessageClient permite que el smartwatch envíe alertas inmediatas sobre caídas detectadas al smartphone, solicitando datos actualizados sobre incidentes recientes. Además, su integración con otros servicios como PutDataMapRequest y sendMessage facilita la transmisión eficiente de datos estructurados, como listas de eventos de caídas almacenadas, garantizando una respuesta rápida ante situaciones críticas sin comprometer la eficiencia ni la confiabilidad del sistema.

Comparación de clientes

Figura 19

Comparación de clientes

	Cliente de datos	Cliente de mensajes	Cliente de canales
Tamaño de datos superior a 100 KB	Sí	No	Sí
Puede enviar mensajes a nodos que no están conectados actualmente.	Sí	No	No
Patrón de comunicación	Recurso compartido basado en la red	Transmisión de mensajes 1:1 (con respuesta)	Transmisión 1:1

Fuente: Developers Android

Anexo 2.- Obtener ubicación en Wear OS

En Wear OS, existen dos métodos principales para obtener la geolocalización en un momento determinado: FLP (la API de servicios de Google) y WHS (Servicios de salud de Wear). Cada método ofrece diferentes características y ventajas, es por esto que las necesidades específicas de la aplicación son las que definen la elección del método más apropiado.

FLP proporciona acceso a una gama más amplia de datos de ubicación, incluyendo latitud, longitud, altitud, precisión, velocidad y dirección. También permite la configuración de notificaciones de ubicación y la realización de seguimiento de ubicación en tiempo real. Sin embargo, FLP consume más batería que WHS y requiere una conexión de red activa para funcionar. Para hacer uso del SDK de Google véase Anexo 3.- Configurar SDK de Google.

WHS está diseñado para proporcionar datos de ubicación de bajo consumo de batería, lo que lo hace ideal para aplicaciones que necesitan realizar un seguimiento de la ubicación del usuario durante períodos prolongados. WHS también puede proporcionar datos de ubicación en interiores, incluso cuando no hay una conexión de red activa.

Tabla 1

Diferencia entre FLP y WHS

Característica	FLP	WHS
Precisión	Alta	Baja
Consumo de batería	Alto	Bajo
Conectividad de red	Requerida	Opcional

Anexo 3.- Configurar SDK de Google

Primero se necesita crear un nuevo proyecto en la Consola de Google Cloud

Figura 20

Creación de proyecto nuevo en la consola de Google

☰ Google Cloud Buscar (/) recursos,

Proyecto nuevo

⚠ Tienes 23 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#) ↗

[MANAGE QUOTAS](#) ↗

Nombre del proyecto * ?

ID del proyecto: proyecto-de-prueba-428722. No se puede cambiar más adelante.

[EDITAR](#)

Ubicación * [EXPLORAR](#)

Organización o carpeta superior

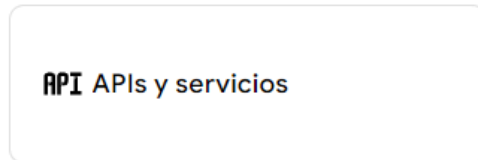
CREAR CANCELAR

Una vez dentro del proyecto se necesita acceder a la parte de APIs y servicios

Figura 21

Sección de APIs y servicios

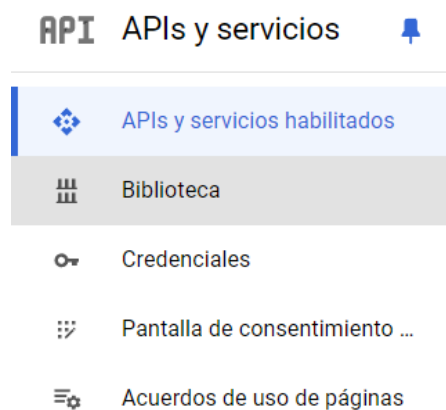
Acceso rápido



Posterior a esto se debe acceder a la sección de Biblioteca

Figura 22

Biblioteca de APIs

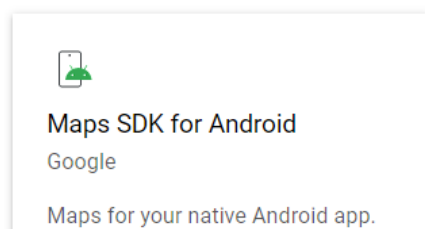


Para poder habilitar los permisos de mapas del SDK se entra a la sección de Maps SDK for Android

Figura 23

Detalles del Maps SDK

Mapas



Una vez dentro se tiene que Administrar los permisos dentro de la API, se deben habilitar los permisos que se requieran para la aplicación

Figura 24

Administración de los permisos de la API

API Name	Status	Description
Address Validation API	ENABLE	The Address Validation API allows developers to verify the accuracy of addresses.
Geocoding API	DISABLE	Convert between addresses and geographic coordinates.
Geolocation API	DISABLE	Location data from cell towers and WiFi nodes.
Places API	DISABLE	Get detailed information about 100 million places
Places API (New)	DISABLE	Next generation of the Places API with access to more than 200 million places
Time Zone API	DISABLE	Time zone data for anywhere in the world.

Finalmente hay que obtener la API KEY que será la cual permita la comunicación entre la app y los servicios de Google

Figura 25

Menú de la consola de Google

- Google Maps Platform
- Descripción general
- API APIs y servicios**
- Métricas
- Cuotas
- Claves y credenciales**
- Asistencia
- Biblioteca de soluciones
- Administración de mapas
- Diseños de mapa
- Conjuntos de datos

Figura 26

Obtención de la API Key

Claves de API

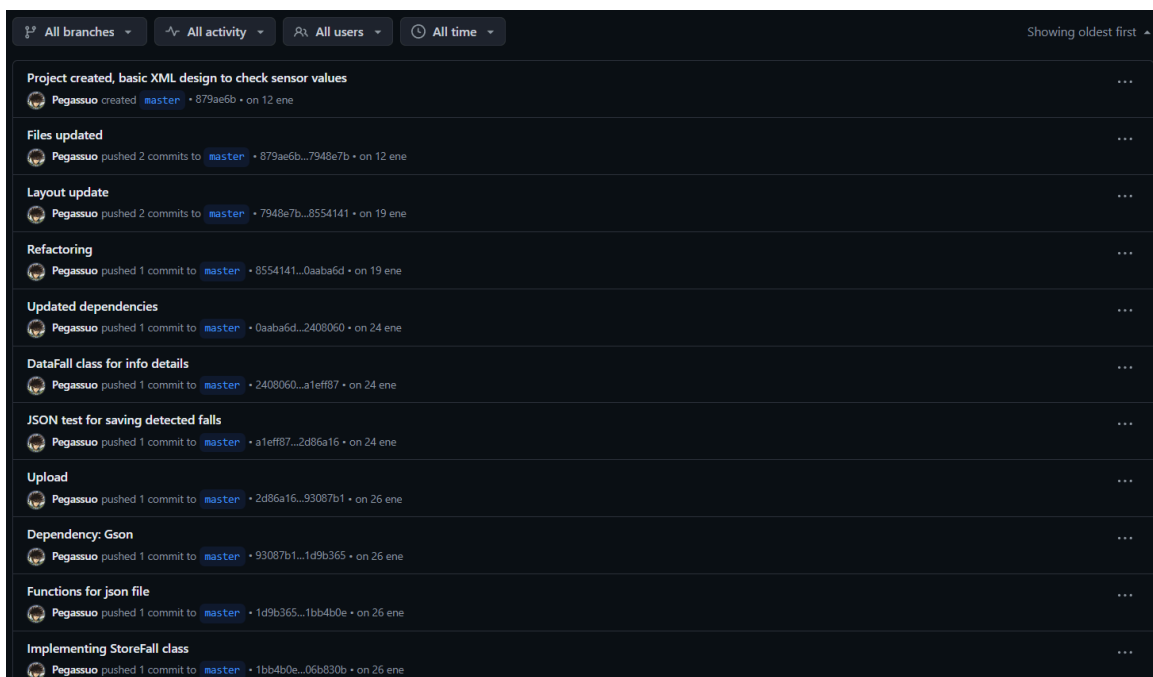
Nombre	Fecha de creación	Restricciones ↑	Acciones
✓ Maps API Key	4 mar 2024	7 API ...	MOSTRAR CLAVE ⋮

Anexo 4.- Historial de versiones

En la Figura 27 se observa claramente la evolución del repositorio a lo largo del tiempo. Desde el inicio del proyecto, se pueden seguir los commits que han registrado cada cambio y mejora realizada en el código. Esta visualización detallada permite identificar cómo se han incorporado progresivamente nuevas funcionalidades y mejoras al sistema. Desde los primeros commits iniciales, donde se establecieron las bases del proyecto, hasta las versiones más recientes que incluyen implementaciones específicas como la detección de caídas y el manejo de alertas, cada paso refleja el progreso y refinamiento continuo del sistema.

Figura 27

Historial de versiones del proyecto



Anexo 5.- Pruebas de funcionalidad en emulador

Las pruebas de funcionalidad se pueden realizar utilizando los controles extendidos disponibles en la plataforma de desarrollo, que permiten modificar los valores de los sensores para simular escenarios de caída. Al ajustar manualmente estos valores, se puede recrear una variedad de situaciones de caída, lo que facilita la evaluación y ajuste del

algoritmo de detección de caídas. Cabe mencionar que esta prueba no es confiable para simular caídas de verdad, pueden servir para verificar si nuestros valores obtenidos son similares o iguales a los ofrecidos por el emulador.

Figura 28

Simulación de caída

