

**Desarrollo de una aplicación para la recolección de datos generados por sensores
en smartphones**

Sonia Fernanda Mancipe Ramírez

Universidad Antonio Nariño
Facultad de Ingeniería de Sistemas
Bogotá, Colombia
2020

**Desarrollo de una aplicación para la recolección de datos generados por sensores
en smartphones**

Sonia Fernanda Mancipe Ramírez

Trabajo de grado como requisito para optar al título de:
Ingeniera de Sistemas y computación

Director(a):

Juan Camilo Ramírez

Asesora Metodológica:

Rosalba Cruz Cepeda
Licenciada en Educación

Tipo de Proyecto:

Desarrollo de software

Universidad Antonio Nariño
Facultad de Ingeniería de Sistemas
Bogotá, Colombia
2020

CONTENIDO

LISTA DE IMÁGENES	5
LISTA DE TABLAS	6
RESUMEN	7
INTRODUCCIÓN	1
1. PLANTEAMIENTO DEL PROBLEMA.....	2
1.1 DESCRIPCIÓN DEL PROBLEMA.....	2
1.2 FORMULACIÓN DEL PROBLEMA.....	3
1.3 JUSTIFICACIÓN	3
1.4 OBJETIVOS.....	5
1.4.1 Objetivo General	5
1.4.2 Objetivos Específicos	5
1.5 ALCANCE Y LIMITACIONES DEL PROYECTO	5
1.5.1 Alcance	5
1.5.2 Limitaciones	6
2. METODOLOGÍA.....	7
2.1 EQUIPO SCRUM.....	7
2.2 EVENTOS	7
2.3 ARTEFACTOS.....	8
2.4 SPRINT.....	8
2.4.1 Sprint 1.....	8
2.4.2 Sprint 2.....	8
2.4.3 Sprint 3.....	9
2.4.4 Sprint 4.....	9
2.4.5 Sprint 5.....	9
3. MARCO REFERENCIAL	10
3.1 MARCO TEÓRICO	10
3.1.1 Reconocimiento de actividad humana.....	10
3.1.2 Sensores.....	10
3.1.3 Sensor acelerómetro	10
3.1.4 Sensor proximidad	11
3.1.5 Giroscopio	12

3.1.6	GPS.....	13
3.1.7	Magnetómetro.....	13
3.1.8	Aprendizaje automático.....	14
3.2	ANTECEDENTES O ESTADO DEL ARTE.....	16
3.2.1	Sensor Kinetics.....	16
3.2.2	Sensores Multiherramienta.....	17
3.2.3	Prueba de Sensores.....	17
4.	DESARROLLO DEL PROYECTO.....	20
4.1	DESCRIPCIÓN DE LA APLICACIÓN.....	20
4.2	FASE DE INICIACIÓN Y AGENDAMIENTO	20
4.3	PLANEACIÓN Y ESTIMACIÓN.....	23
4.3.1	Roles del sistema.....	23
4.3.2	Arquitectura y herramientas del sistema	23
4.3.3	Listado de historias de usuario	23
4.3.4	Plan de pruebas	24
4.4	IMPLEMENTACIÓN Y DESARROLLO	25
4.4.1	Sprint 1: módulo de inicio	25
4.4.2	Sprint 2: módulo de visualizar sensores	28
4.4.3	Sprint 3: módulo de configuración	39
4.4.4	Sprint 4: módulo para configurar tiempo de registro	43
4.4.5	Sprint 5: módulo para generar archivo de texto	47
5.	RESULTADOS OBTENIDOS.....	52
6.	CONCLUSIONES Y RECOMENDACIONES	55
7.	BIBLIOGRAFÍA.....	56
8.	CIBERGRAFÍA	58

LISTA DE IMÁGENES

Imagen 1. Sistema de coordenadas	11
Imagen 2. Proximidad	12
Imagen 3. Giroscopio disco circular	12
Imagen 4. GPS Asistido	13
Imagen 5. Magnetómetro	14
Imagen 6. Sensor Kinetics	16
Imagen 7. Sensor Kinetics Visualización	16
Imagen 8. Menú App	17
Imagen 9. Sensor GPS	17
Imagen 10. Sensor Proximidad	17
Imagen 11. Menú Sensores	18
Imagen 12. Sensores	18
Imagen 13. Plantilla historia de usuarios	21
Imagen 14. Inicio de Tablero Trello	22
Imagen 15. Proceso de tablero Trello	22
Imagen 16. MVC	23
Imagen 17. Módulo de inicio	26
Imagen 18. Ejecutar aplicación	26
Imagen 19. Diagrama de actividades Sprint 1	27
Imagen 20. Mostrar sensores	32
Imagen 21. Visualizar sensores	32
Imagen 22. Diagrama de actividades Sprint 2	33
Imagen 23. Definición de Sensores para registrar	40
Imagen 24. Seleccionar sensores a registrar	41
Imagen 25. Diagrama de actividades Sprint 3	41
Imagen 26. Parametrización de tiempo para registro	44
Imagen 27. Parametrizar tiempo	44
Imagen 28. Diagrama de actividades Sprint 4	45
Imagen 29. Generar archivo de texto	48
Imagen 30. Ejecutar acciones de la app	48
Imagen 31. Diagrama de actividades Sprint 5	49
Imagen 32 Prueba app SensorReader usuario 1	53
Imagen 33 Prueba app SensorReader usuario 2	54

LISTA DE TABLAS

Tabla 1. Benchmarking	19
Tabla 2. Product Backlog	24
Tabla 3. Requerimiento de la aplicación para Pantalla principal	25
Tabla 4. Caso de prueba funcional pantalla de inicio	28
Tabla 5. Visualizar sensores	29
Tabla 6. Mostrar Acelerómetro	29
Tabla 7. Mostrar Proximidad	30
Tabla 8. Mostrar Giroscopio	30
Tabla 9. Mostrar Magnetómetro	31
Tabla 10. Mostrar GPS	31
Tabla 11. Caso de prueba funcional	34
Tabla 12. Caso de prueba acelerómetro	35
Tabla 13. Caso de prueba proximidad	36
Tabla 14. Caso de prueba giroscopio	37
Tabla 15. Caso de prueba magnetómetro	38
Tabla 16. Caso de prueba GPS	39
Tabla 17. Configuración de Registro Sensores	40
Tabla 18. Caso de prueba funcional definición de Sensores	42
Tabla 19. Configuración de Registro Tiempo	43
Tabla 20. Caso de prueba funcional parametrización de tiempo	46
Tabla 21. Generación de archivo de texto	47
Tabla 22. Caso de prueba funcional Sprint 5	50
Tabla 23 Parte 2 Caso de prueba funcional Sprint 5	51

RESUMEN

Este trabajo de grado presenta el desarrollo de la aplicación móvil SensorReader para la recolección de datos a partir de sensores en teléfonos inteligentes con sistema operativo Android. Con este desarrollo se busca dar solución a una necesidad identificada en el proyecto de investigación “Sistema de comunicación para sobrevivientes de un desastre basado en una red ad-hoc de teléfonos inteligentes”, liderado por el grupo LACSER - Laboratory for Advanced Computational Science and Engineering Research de la Universidad Antonio Nariño, el cual tiene por objetivo desarrollar una aplicación móvil en sistema operativo Android como un canal alternativo de comunicación que permita intercambiar mensajes entre sobrevivientes de un desastre y proporcionar en tiempo real la ubicación geográfica aproximada de los mismos, sin requerir acceso a Internet ni una infraestructura centralizada de red. El proyecto de investigación contempla el uso de aprendizaje automático a partir de las lecturas hechas por los sensores de un dispositivo con el propósito de predecir automáticamente si el mismo está siendo manipulado en ese momento por un ser humano y así decidir si la aplicación de mensajería debe priorizar la comunicación con dicho nodo. La aplicación SensorReader, desarrollada en este trabajo de grado permitirá la recolección de datos para el entrenamiento de los modelos de aprendizaje automático previstos en el proyecto de investigación.

Este desarrollo se propone como una alternativa para la recolección de la información requerida por el proyecto de investigación, dado que las aplicaciones disponibles en el mercado no satisfacen los requerimientos del mismo. La app desarrollada en este trabajo de grado le ofrecerá al usuario varios beneficios, incluyendo la selección de los tipos de sensores que se leerán y el rango de tiempo en el que realizará la captura de datos.

Con el desarrollo de este trabajo de grado se planea como opción en un futuro aplicar aprendizaje automático para la posterior identificación de actividad humana a partir de los datos recolectados.

Palabras claves: android, dispositivos inteligentes, sensores, proximidad, acelerómetro, giroscopio, GPS y magnetómetro.

ABSTRACT

This undergraduate degree project presents the development of a mobile application for the collection of data readings from sensors in Android smartphones. This development seeks to address a need identified in the research project “Communication system for disaster survivors based on an ad-hoc smartphone network”, led by research group LACSER - Laboratory for Advanced Computational Science and Engineering Research of Universidad Antonio Nariño, which aims to develop a mobile application in Android operating system as an alternative communication channel that allows the exchange of messages between survivors of a disaster and provides in real time the approximate geographical location of these without requiring access to the Internet or a centralized network infrastructure. The research project contemplates the use of machine learning from the readings made by the sensors of a device with the purpose of automatically predicting if it is being manipulated at that time by a human being and thus decide if the messaging application should prioritize communication with said node. The application developed in this degree project will allow the data collection for the training of machine learning models provided in the research project.

This development is proposed as an alternative for the collection of the information required by the research project since the applications available in the market do not meet its requirements. The app developed in this undergraduate degree project will offer the user several benefits, including the selection of the types of sensors that will be read and the time range in which the data will be captured.

Optional future work contemplated as part of this undergraduate degree project consists of the application of machine learning for the subsequent identification of human activity from the data collected.

Keywords: android, smart devices, sensors, proximity, accelerometer, gyroscope, GPS and magnetometer

INTRODUCCIÓN

El reconocimiento de actividad humana consiste en identificar acciones que las personas realizan a partir de registros tomados anteriormente sobre los mismos (Anguita, Ghio & Oneto, 2013). Se realiza a través de dispositivos móviles con sensores que permiten identificar posiciones en las que se encuentre una persona como acostado, de pie, de lado, entre otros. Sus aplicaciones se extienden a diferentes sectores como es el caso en el área de la salud, en donde se deben realizar monitoreos constantes a los pacientes. Con los datos obtenidos se analiza la información sistemáticamente para que los doctores tengan información necesaria y oportuna de cada paciente (Chetty, White & Akther., 2015; Capela, 2016).

El proyecto de investigación “Sistema de comunicación para sobrevivientes de un desastre basado en una red ad-hoc de teléfonos inteligentes”, liderado por el grupo LACSER de la Universidad Antonio Nariño, tiene por objetivo desarrollar un canal alternativo de comunicación que permita intercambiar mensajes entre sobrevivientes de un desastre y proporcionar en tiempo real la ubicación geográfica aproximada de los mismos, sin requerir acceso a Internet ni una infraestructura centralizada de red. Además, requieren predecir automáticamente si un dispositivo está siendo manipulado en ese momento por un ser humano, para priorizar la comunicación con los nodos en los que se identifica se está presentando contacto físico con el humano, lo anterior con el objetivo de economizar recursos como la batería. Esto se pretende hacer aplicando aprendizaje automático a partir de la lectura de los sensores de un smartphone. Actualmente el proyecto de investigación no cuenta con una herramienta que le permita recopilar los datos necesarios para hacer el entrenamiento de los modelos de aprendizaje automático previstos.

La aplicación móvil SensorReader desarrollada en este trabajo de grado permitirá la recopilación de datos que posteriormente serán utilizados para el entrenamiento de estos modelos.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCIÓN DEL PROBLEMA

Hoy en día, el recopilar los datos generados por los sensores de dispositivos móviles tienen varias aplicaciones en estudios de reconocimiento de actividad humana, como por ejemplo para el monitoreo de humanos pacientes fuera de un hospital (Capela et al., 2016). También se han realizado investigaciones para implementar kit de sensores (luz, temperatura, proximidad, acelerómetro, frecuencia cardíaca, entre otros) en hogares que permitan la lectura no solo del entorno del hogar sino también de las personas u animales que entren en ella (Munguia, Intille, Lopez & Larson, 2006).

En la actualidad las aplicaciones en dispositivos móviles cuentan con el soporte en cuanto a capacidad, memoria y batería para la recolección de datos en tiempos prolongados (Shoaib et al., 2015; Hammerla, 2016). En la actualidad, estos dispositivos electrónicos han avanzado en tecnología, ampliando su capacidad y haciendo posible la ejecución de este tipo de aplicaciones recolectoras de datos (Shoaib et al., 2015). Adicional, los teléfonos móviles están equipados con varios sensores, como: micrófono, cámara, GPS, giroscopio, Bluetooth, acelerómetro y proximidad, entre otros (Stenneth, 2013). Generalmente estos sensores vienen adaptados en los smartphones para mejorar las experiencias de las personas. Además, su disponibilidad depende del procesador y de la batería (Pires, Garcia, Pombo & Flórez, 2011).

Para este trabajo de grado los sensores que serán usados para recolectar datos por medio del smartphone se basan en los mínimos básicos con los que cuentan los smartphones ya que así se amplía el porcentaje de funcionalidad de los sensores de la app SensorReader, de igual forma se permite definir que sensores o no recolectar en caso de que no sean requeridos y así tener un ahorro en consumo de batería y de almacenamiento. Teniendo en cuenta que el grupo de investigación usará esta app para predecir si un smartphone tiene contacto o no con un humano por medio de aprendizaje automático y que para este tipo de aprendizaje es habitual el uso de archivos de texto plano por ser más aceptable para los lenguajes de manipulación de datos la información se debe recolectar en dicho formato.

Una de las situaciones en donde la recopilación de datos suministrados por los sensores de un smartphone podría impactar positivamente es durante las catástrofes naturales; durante estos sucesos es de vital importancia ubicar personas de manera rápida y precisa, por lo que se implementa el uso de sensores como el acelerómetro y el giroscopio que nos permiten conocer la velocidad u orientación en la que se mueve el smartphone; el sensor proximidad, que nos permite identificar si hay contacto o no con el smartphone por medio de un detector de luz; magnetómetro, que nos indica que tan intenso es o no el campo magnético y por último el GPS, que nos permite acceder a las coordenadas u ubicación del smartphone.

El proyecto de investigación “Sistema de comunicación para sobrevivientes de un desastre

basado en una red ad-hoc de teléfonos inteligentes”, liderado por el grupo LACSER - Laboratory for Advanced Computational Science and Engineering Research de la Universidad Antonio Nariño, tiene por objetivo desarrollar una aplicación móvil en sistema operativo Android como un canal alternativo de comunicación que permita intercambiar mensajes entre sobrevivientes de un desastre y proporcionar en tiempo real la ubicación geográfica aproximada de los mismos sin requerir acceso a Internet ni una infraestructura centralizada de red. El proyecto de investigación **contempla** el uso de aprendizaje automático a partir de las lecturas hechas por los sensores de un smartphone con el propósito de predecir automáticamente si el dispositivo está siendo manipulado en ese momento por un ser humano y así decidir si la aplicación de mensajería debe priorizar la comunicación con dicho nodo.

1.2 FORMULACIÓN DEL PROBLEMA

¿Cómo automatizar la recolección de datos de sensores en un smartphone por medio de una aplicación móvil?

1.3 JUSTIFICACIÓN

En el presente trabajo de grado se desarrolló una aplicación móvil que permite a los usuarios definir inicialmente los sensores de un smartphone que se requieren recolectar y el rango de tiempo en el que se deben leer, de esta forma una vez el usuario inicia la lectura bajo los parámetros que definió se ejecuta la recolección de datos de manera repetitiva hasta que el usuario finalice, una vez finalice se entrega la información en un archivo de texto plano.

Este trabajo de grado contribuirá con el proyecto de investigación “Sistema de comunicación para sobrevivientes de un desastre basado en una red ad-hoc de teléfonos inteligentes” que tiene por objetivo desarrollar una aplicación móvil en sistema operativo Android como un canal alternativo de comunicación que permita intercambiar mensajes y proporcionar en tiempo real la ubicación geográfica aproximada de sobrevivientes sin requerir acceso a Internet. Esta investigación, liderada por el grupo LACSER de la Universidad Antonio Nariño, posteriormente permitirá la búsqueda de métodos efectivos para localizar sobrevivientes de desastres naturales y sus resultados se utilizarán para el desarrollo de mecanismos de comunicación con estos sobrevivientes que faciliten su rescate.

La importancia social y económica de este trabajo de grado radica en el aporte gratuito de una herramienta para la predicción de actividad humana en un smartphone que el grupo de investigación LACSER usará con el fin de localizar sobrevivientes de un desastre y la comunicación con los mismos estará contribuyendo a salvar vidas y a establecer mecanismos de rescate más eficientes.

Teniendo en cuenta que este trabajo de grado aportara un aplicativo movil con los sensores básicos de un smartphone para a futuro aplicar un método de aprendizaje automático sobre los datos que se generan en formato de texto plano que es la estructura más aceptable para manipulación de datos y considerando el consumo de batería y almacenamiento que esta aplicación en ejecución puede consumir se permite al usuario recolectar únicamente los sensores requeridos en el tiempo requerido, dependiendo de los criterios planteados para realizar el estudio de aprendizaje automático sobre frecuencia de las lecturas y especificación de sensores..

Con este trabajo de grado el estudiante encargado aplicará los conocimientos adquiridos durante el pregrado para dar fin a lo propuesto, el estudiante tendrá claro cuál es su conocimiento, qué área de interés le es llamativa y se sabrá también si está preparado para continuar sus metas educativas. Además del grupo de investigación y del estudiante que desarrolla la tesis esta podrá ser útil para futuras tesis o proyectos que requieran información de la que en este trabajo de grado se brinda, de ser el caso que se realice un estudio basado en la rama de este trabajo funcionara como una base o recurso.

1.4 OBJETIVOS

1.4.1 Objetivo General

Diseñar y desarrollar una aplicación en sistema operativo Android que permita la lectura y recolección de los datos generados por los sensores de un dispositivo móvil.

1.4.2 Objetivos Específicos

- Desarrollar una aplicación móvil en sistema operativo Android que realice como mínimo la lectura de sensores de proximidad, acelerómetro, giroscopio, GPS y magnetómetro de un smartphone.
- Realizar pruebas de la aplicación para garantizar la correcta recolección de los datos que se almacenarán en archivo con formato de texto.
- Permitir a los usuarios la selección de los tipos de sensores y rangos de tiempo a recolectar para ser analizados.

1.5 ALCANCE Y LIMITACIONES DEL PROYECTO

1.5.1 Alcance

El alcance planteado para la aplicación móvil es:

- Inicio, la aplicación permite al usuario elegir entre visualizar los sensores en tiempo real o configurar los parámetros para la lectura de sensores, ninguno de estos procesos se relaciona o afecta al otro.
- Visualización, el usuario tiene a disposición los siguientes sensores: acelerómetro, proximidad, giroscopio, magnetómetro o GPS para visualizar los datos generados en ese instante de tiempo.
- Configuración, el usuario puede seleccionar entre los sensores de: aceleración, proximidad, giroscopio, magnetómetro o GPS dependiendo de la necesidad que se presenta para ser registrados sus datos en un archivo de texto.
- Parametrización, le permite al usuario definir cada cuanto tiempo en medida de segundos se van a leer y guardar los datos de los sensores que se definen en la configuración.

-
- La aplicación genera para el usuario un archivo con formato de texto una vez se ha realizado una lectura, dentro de este archivo se recolectan los datos generados por los sensores, hora de la lectura y porcentaje de batería del smartphone.

1.5.2 Limitaciones

Las limitantes del proyecto son:

- La aplicación móvil se realizó en sistema operativo Android, a partir de la versión 5.0.
- Para la correcta ejecución de la app en el smartphone se debe contar por lo menos con los sensores de proximidad, acelerómetro, giroscopio, GPS y magnetómetro.
- La aplicación móvil se desarrolló en el lapso de un semestre (4 meses).
- Durante el desarrollo de este trabajo de grado se contempló como trabajo **opcional** de acuerdo con la cobertura del tiempo el entrenamiento de un modelo de aprendizaje automático (utilizando máquinas de vectores de soporte o redes neuronales, por ejemplo) sobre un conjunto de datos recopilados con la app desarrollada, esta sería una tarea opcional que el estudiante desarrollador podía contemplar realizar si así lo desea y si el tiempo es acorde.

2. METODOLOGÍA

Scrum es una metodología de desarrollo ágil, tiene un proceso que se sigue de manera continua con un conjunto de buenas prácticas, el cual se adaptará a este proyecto de desarrollo. La metodología Scrum es conocida y practicada por el estudiante desarrollador del proyecto. Esta metodología permite trabajar colaborativamente y de manera continua, realizando revisiones para obtener un buen resultado y si existen correcciones ajustar con tiempo (Schwaber & Sutherland, 2013).

2.1 EQUIPO SCRUM

Para el proyecto se establecieron los siguientes roles:

- **Product Owner:** director del proyecto de investigación, María del Pilar Salamanca.
- **Scrum Master:** director del trabajo de grado, Juan Camilo Ramírez. Esta persona se encargó de observar al equipo, no intervenía en los procedimientos, únicamente cuando existieron falencias o cosas que se podrían cambiar, se encargó de verificar el cumplimiento de la metodología y todo lo propuesto.
- **Development Team:** Sonia Mancipe. Realizó las tareas de diseño, desarrollo, implementación y creación de los requisitos planteados al proyecto.

2.2 EVENTOS

Los eventos se utilizan en Scrum para realizar con regularidad reuniones y minimizar la necesidad de algunas no definidas. Con los eventos se define un lapso, así cada evento tiene un tiempo máximo definido. Estos eventos en la metodología Scrum hacen parte de un Sprint. Un evento puede terminarse antes del tiempo establecido si el objetivo del evento ya se cumplió.

Sprint es un conglomerado de eventos que permiten realizar revisiones y seguimientos de manera formal con el equipo para inspeccionar y ajustar algo si corresponde. Además, es posible realizar retroalimentación a las actividades en ejecución.

Para el proyecto se llevaron a cabo las siguientes reuniones:

- **Reunión semanal:** se realizaron una vez por semana revisión con directores de proyecto durante 30 minutos.
- **Reunión de Planificación del Sprint:** durante las reuniones semanales se definieron fechas exactas y horas para sprint y tareas a entregar. Los sprint eran mínimo cada 3 semanas.

- **Revisión del Sprint:** de acuerdo con las fechas agendadas se realizaron las entregas indicadas.
- **Retrospectiva del Sprint:** cada tres semanas se realizaron reuniones con el grupo de investigación y directores para revisar los avances, logros, fallas, nuevas fechas y retroalimentaciones.

2.3 ARTEFACTOS

Los artefactos representan trabajo que es útil para generar oportunidades que requieren ajustes, aunque en cualquier momento pueden surgir nuevas tareas que se suman en los seguimientos de cada sprint. Las fases planeadas son:

- **Configuración entornos de desarrollo:** el estudiante adaptó los recursos necesarios en el equipo sobre el cual realizó el desarrollo de la aplicación.
- **Inicio:** por medio de reuniones con los directores se realiza el levantamiento de los requerimientos para el desarrollo de la aplicación.
- **Agendamiento:** junto con los directores y grupo de investigación se definió el plan de trabajo que se llevó a cabo para las entregas, se definieron los recursos a utilizar y tiempos máximos de entrega.
- **Desarrollo:** se inicia la fase de desarrollo de la aplicación en el lenguaje seleccionado y el requerimiento indicado.
- **Sprint:** durante el sprint se presentaron los acuerdos definidos para la entrega, una vez se cumplió el tiempo de duración.
- **Revisión general:** los resultados obtenidos del desarrollo durante el sprint se presentan y se acuerdan nuevas entregas.
- **Manuales de uso, liberación de desarrollo:** se realizó la documentación final para usuarios de la aplicación y presentación con las funcionalidades definidas inicialmente. El desarrollo se culminó al 100%.
- **Revisión general y cierre:** se presenta la información entregada en la fase anterior al grupo de investigación y directores.

2.4 SPRINT

2.4.1 Sprint 1

Módulo de inicio:

- Presentar la opción de visualizar los datos generados por los sensores en tiempo real.
- Presentar la opción de configurar la recopilación de los datos generados por los sensores.

2.4.2 Sprint 2

Módulo de visualizar sensores (Acelerómetro, Proximidad, Giroscopio, GPS y Magnetómetro):

- Visualizar aceleración en los 3 eje en tiempo real.
- Visualizar proximidad al smartphone en tiempo real.
- Visualizar velocidad de rotación alrededor de los 3 ejes en tiempo real.
- Visualizar orientación del campo magnético de los 3 ejes en tiempo real.
- Visualizar coordenadas de geolocalización en tiempo real si el GPS este activo.

2.4.3 Sprint 3

Módulo de configuración de registro, sensor:

- Seleccionar los sensores a leer entre acelerómetro, proximidad, giroscopio, magnetómetro y/o GPS para registro.

2.4.4 Sprint 4

Módulo para parametrización de tiempo de lectura:

- Ingresar intervalo de tiempo (segundos) para realizar el registro de los sensores.

2.4.5 Sprint 5

Módulo para generación de archivo de texto:

- Crear archivo con registro de los sensores seleccionados por el usuario.
- Compartir en archivo la hora de la lectura más el porcentaje de batería.

3. MARCO REFERENCIAL

3.1 MARCO TEÓRICO

3.1.1 Reconocimiento de actividad humana

Es la identificación de actividades que realizan los humanos a partir de un conjunto de acciones que ya han realizado y son objeto de estudio para identificar una vez se repitan que comportamiento o posición se está realizando (Anguita, Ghio & Oneto, 2013; Han et al., 2014). El reconocimiento de actividad humana se realiza a través de dispositivos móviles que permiten identificar posiciones en las que se encuentre una persona, como: acostado, de pie, de lado, entre otros.

El reconocimiento de actividad humana busca adquirir información sobre el estado de un humano a partir de datos generados en el ambiente que lo rodea, esto permite el monitoreo de los usuarios o paciente. Este reconocimiento es útil en áreas de salud médica, por ejemplo, pues permite tener bajo registro los cambios de un paciente que esté conectado (Chetty, White & Akther., 2015). Para el proyecto de investigación que motiva este trabajo de grado el reconocimiento de actividad humana consistirá en identificar si un smartphone está siendo usado o no por un humano.

3.1.2 Sensores

Los smartphones cuentan con una estructura de hardware que se compone de memoria RAM, cámara, capacidad, velocidad, acelerómetro, giroscopio entre otros. El acelerómetro y giroscopio hacen parte de la estructura de sensores, los cuales reciben señales que se emiten a su alrededor (Pires, García, Pombo & Flórez, 2011). Las señales registradas se pueden recibir de diferentes eventos, como: movimientos o actividades que tienen los usuarios.

Los teléfonos móviles están equipados con varios sensores, como: micrófono, cámara, GPS, giroscopio, Bluetooth, acelerómetro y proximidad (Stenneth, 2013). Generalmente estos sensores vienen adaptados en los smartphones para mejorar las experiencias de las personas. Además, su disponibilidad depende del procesador y de la batería (Pires, Garcia, Pombo & Flórez, 2011). Para este trabajo de grado los sensores serán usados para recolectar datos por medio del smartphone que el grupo de investigación usará para predecir si un smartphone tiene contacto o no con un humano.

3.1.3 Sensor acelerómetro

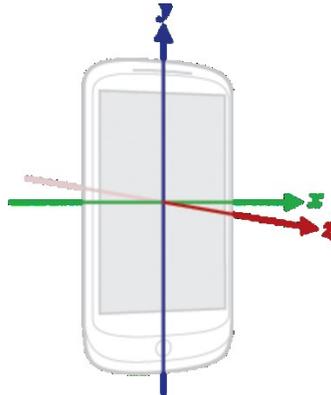
En el desarrollo de este proyecto se tendrán en cuenta sensores que permiten tomar muestra de movimientos, como: el acelerómetro, proximidad, giroscopio y magnetómetro, pero también se tendrá en cuenta el significado de biometría que hace referencia a la

identificación de autenticación por medio de teclas o táctil de un móvil para lo que los teléfonos inteligentes actualmente vienen equipados (Maghsoudi, 2017).

El acelerómetro es un dispositivo capaz de captar las fuerzas de aceleración a las que se enfrenta, permite capturar los datos desde los ejes X, Y Z (imagen 1) (Viana et al., 2017), además con este sensor es posible medir las fuerzas de aceleración estáticas o dinámicas en aplicaciones móviles, lo cual permite identificar movimiento u orientación (Maghsoudi,2017). En el artículo realizado por Federal Institute of Education, Science and Technology of Ceara (IFCE) en el 2017 diseñó una herramienta que tiene como sensor de datos el acelerómetro y se analizan para identificar la superficie de un terreno en el cual se presentan accidentes (Viana et al., 2017).

Este sensor ha sido usado para determinar actividad humana diaria y de los deportes, ya que permite reconocer de manera confiable los movimientos u acciones que hagan (Shoaib et al.,2015). El acelerómetro se caracteriza por proporcionar alta precisión en los datos de reconocimiento generados para detectar eventos, lo cual sería útil en caso de que los datos recolectados se usen para un trabajo de aprendizaje automático (Shoaib et al.,2015).

Imagen 1. Sistema de coordenadas



Fuente 1: tomado de Hernández (2017)

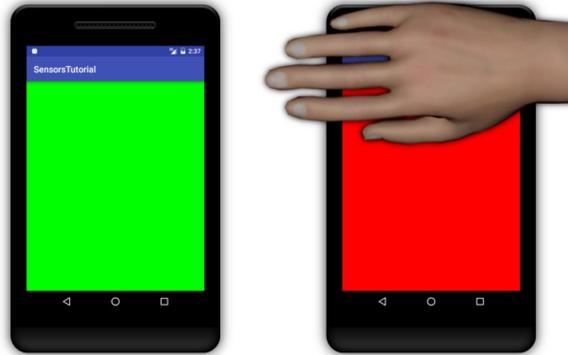
3.1.4 Sensor proximidad

Es una almohadilla o dispositivo táctil usada en sistemas electrónicos que crean una región para su lectura de detección 2D o 3D, dicha región permite identificar la información de un movimiento o acción realizado sobre un smartphone. Con el sensor de proximidad es posible que un usuario realice movimientos con el smartphone dentro de la región creada y capturar el evento. Ejemplos de lo que el sensor permite capturar son: contestar una llamada acercando el equipo al oído o mover el equipo como manecillas del reloj (Hovden et al.,2007).

Por otro lado, este sensor permite identificar los cambios de orientación de un smartphone, el contacto o cercanía entre éste y un cuerpo, como, por ejemplo, cuando se acerca el equipo a una oreja para recibir una llamada se bloquea la pantalla y generalmente se

mantiene hasta no alejarse (imagen 2) (Honores & Vizuite, 2014). Este sensor junto con sensores como temperatura, luz y detectores de corriente electrónica son implementados como parte de un kit que lee datos en entonos (Munguia, Intille, Lopez & Larson, 2006).

Imagen 2. Proximidad



Fuente 2: tomado de Hathibelagal (2017)

3.1.5 Giroscopio

Este sensor trabaja de la mano con el acelerómetro ya que ambos trabajan sobre la orientación del smartphone. A diferencia del acelerómetro este sensor funciona a partir de un disco circular simétrico (imagen 3). Con este sensor se puede mantener un registro de orientación de un teléfono móvil el cual se puede seleccionar gráficamente para mantener o cambiar automáticamente la imagen horizontal o vertical, la dirección y el movimiento angular. En un teléfono móvil, pueden determinar la posición y orientación (Jiménez, 2014).

Este sensor ha tomado fuerza a raíz de la realidad virtual aumentada, juegos y videos en 360. Este sensor permite inclinar el teléfono inteligente con cualquier sentido para cambiar de perspectiva o mover un objeto dentro de un juego (Collado, 2019).

Imagen 3. Giroscopio disco circular



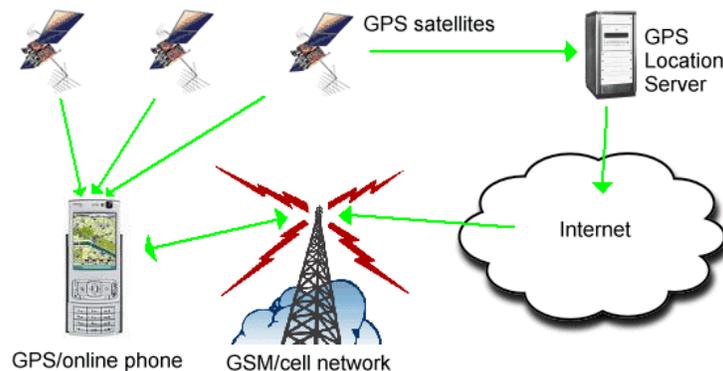
Fuente 3: tomado de Mira Cómo hacerlo (2019)

3.1.6 GPS

El sensor de posicionamiento global (en inglés: Global Positioning System) fue desarrollado por el ejército de los EE.UU. en 1973. Los smartphones tienen disponibles 24 satélites operando el 95% del tiempo, aunque se comunican por medio de los 31 satélites que la fuerza aérea ha estado estudiando y se encuentran operacionales en órbita por medio de una antena interna que es parte del hardware del smartphone. La antena está conectada al smartphone por medio de un controlador de software o Driver que permite la interacción entre la antena y el sistema operativo para la generación de una interfaz o datos (Dalul, 2019).

Para acceder a la ubicación por medio de una aplicación móvil se puede utilizar el GPS y el proveedor de ubicación de red de Android, el proveedor de ubicación de red de Android es un asistente (AGPS) de GPS que logra identificar la ubicación de un usuario por medio de la torre de telefonía celular y las señales de Wi-Fi (imagen 4), proporcionando información de ubicación de manera que funcione en interiores y exteriores ya que el GPS solo no es capaz de recolectar datos en interiores y exteriores, además los datos se podrán recolectar rápidamente y el consumo de batería será menor (estrategias de ubicación, 2019). Para realizar el registro de coordenadas de la posición que tiene el usuario por medio del GPS el smartphone interactúa con tres satélites por lo que puede tardar un poco en la identificación.

Imagen 4. GPS Asistido



Fuente 4: tomado de Kios Buku Gema (2011)

3.1.7 Magnetómetro

El sensor magnetómetro recolecta información sobre la intensidad del campo magnético, generalmente este sensor se implementa junto con el acelerómetro. Además de la rotación, el magnetómetro permite identificar la posición en la que se encuentra con respecto a la tierra que siempre apunta al norte, como se puede observar en la imagen 5.

Imagen 5. Magnetómetro



Fuente 5: tomado de Could an iPhone be a useful field tool? (2009)

3.1.8 Aprendizaje automático

El aprendizaje automático es la disciplina que se encarga de hacer máquinas con la habilidad de aprender a partir de grandes volúmenes de datos a hacer predicciones o tareas que son difíciles o imposibles de hacer para los seres humanos (Castonguay et al., 2017). El uso de aprendizaje automático sobre los datos generados por sensores móviles puede ser usado en ramas como: finanzas, ya que se aplica para el análisis de costos y prevención de impactos futuros o planeación de presupuestos a partir de costos en meses anteriores; en salud, ya que permite tener bajo registro los cambios de un paciente que esté conectado e identificar si es posible patrones de tiempo o afectaciones recurrentes y tecnología, ya que nos permite tener máquinas como dispensadores de alimento para mascota (perro) que cada determinado tiempo o de acuerdo a la dieta que maneja el animal le ocupa el frasco de alimento (Vaughn, Biocco, Liu & Anwar. 2018).

Ahora bien, la clasificación en aprendizaje automático es la predicción de un dato a partir de otros. El dato que se predice se nombra atributo y es categórico; los atributos categóricos son datos preestablecidos como “sí” o “no”, “mujer” o “hombre”; son valores limitados entre los cuales el atributo final solo puede ser uno de los dos (Marin, s.f).

Las clasificaciones de aprendizaje supervisado, no supervisado y por refuerzo aplican para diferentes situaciones. Algunos casos en los que se podría aplicar uno de estos algoritmos es en el área de tecnología, en la identificación de un correo maligno o no deseado, identificación de acceso a un sitio, equipo o red de un intruso (González, 2018; Juárez, 2017).

El método de aprendizaje supervisado consiste en tener un registro de datos que son características de un valor o que con llevan a un solo valor. Este tipo de datos finales son valores etiquetados o etiquetas, por lo que cuando se tenga una serie de características se podrá identificar a que valor hacen referencia. Este método usa algunas técnicas como: redes neuronales que simula comportarse como el cerebro humano, bosques aleatorios, máquinas de vectores de soporte que busca encontrar una combinación de parámetros que den una respuesta o predicción y vecinos más cercanos (González, 2018).

El método de aprendizaje no supervisado igual que el método supervisado se basa en datos de entrenamiento, los cuales no llevan un patrón ni etiquetas de salida. Este método usa algunas técnicas como K-media, mezcla gaussiana y agrupamiento jerárquico (González, 2018). Por último, el método de aprendizaje por refuerzo es diferente ya que tiene un modelo de acción-recompensa, con lo que busca maximizar recompensa de acuerdo con una acción, generalmente este modelo se refleja en reacciones humanas, este método de aprendizaje también usa técnicas como programación dinámica (González, 2018).

Los datos de entrenamiento son la base de información que entrena modelos de aprendizaje automático para lograr mayor precisión en un estudio. Estos datos son fundamentales para la implementación del aprendizaje automático, debe contener diferentes tipos de datos detallados que son las entradas y una etiqueta de valor (si o no, correcto o incorrecto, entre otros) que representan una salida o atributo destino (Smith, 2019).

Los conjuntos de datos de entrenamiento se recopilan de la realización de un evento o eventos en varias ocasiones al azar, dependiendo de la cantidad de datos y la forma en que se recolecten los datos es posible lograr precisión en el resultado de la aplicabilidad de los datos, entre más datos de entrenamiento se recopilen de diferentes entornos como la calle, una casa, un parque entre otras, y situaciones como estar usando el smartphone o no usarlo, recibir una llamada, caminar con el smartphone en un bolsillo, dejar el dispositivo electrónico en un carro, entre otros; permitirán dar una mayor precisión para lograr una predicción valida, pues se tendrán las bases suficientes para comparar diferentes casos y dar una respuesta (Suto, Oniga, Lung & Orha. 2017).

3.2 ANTECEDENTES O ESTADO DEL ARTE

A continuación, se relacionan aplicaciones desarrolladas por compañías de software para la lectura de datos generados por sensores. Cabe resaltar que la app que se desarrollará en este trabajo de grado ofrecerá las funcionalidades que las aplicaciones citadas a continuación no ofrecen.

3.2.1 Sensor Kinetics

La aplicación Sensor Kinetics fue creada por Innoventions Inc., en Houston, con la tecnología de RotoView. Esta aplicación registra la lectura de muchos sensores, incluidos el acelerómetro, el giroscopio, el magnetómetro, entre otros (imagen 6 y 7).

- Únicamente registra los datos cuando se especifica por lo que no permite parametrizar un lapso o secuencias de **tiempo** que especifique la lectura de los sensores en tiempos determinados.
- La aplicación no permite una opción para registrar coordenadas, ni opción de **GPS**.
- La aplicación no aparenta ser lo suficientemente amigable a un usuario ya que contiene demasiados caminos.

Imagen 6. Sensor Kinetics



Fuente 6: captura de app Kinetics (2019)

Imagen 7. Sensor Kinetics Visualización



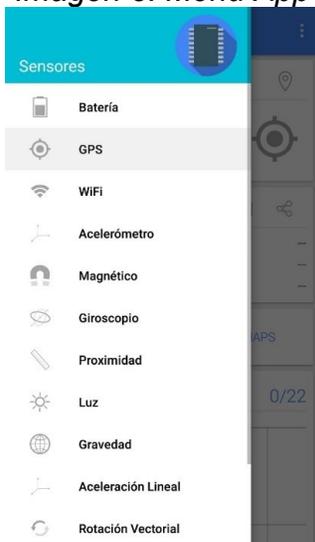
Fuente 7: captura de app Kinetics (2019)

3.2.2 Sensores Multiherramienta

La aplicación Sensores Multiherramienta fue creada por weRed Software. Esta aplicación permite visualizar sensores como: batería, GPS, acelerómetro, giroscopio, magnetómetro, proximidad, entre otros (imagen 8, imagen 9 e imagen 10).

- La aplicación únicamente permite visualizar el dato capturado en ese instante de tiempo.
- Únicamente permite visualizar un dato cuando se pide enviar a otra aplicación la información (eje. WhatsApp, Gmail)
- No permite se parametrica un lapso o secuencias de **tiempo** para la lectura de los sensores en tiempos determinados.
- La aplicación, aunque tiene una opción para lectura de **GPS** no funciona.
- La aplicación no almacena la información en ningún archivo para el usuario.

Imagen 8. Menú App



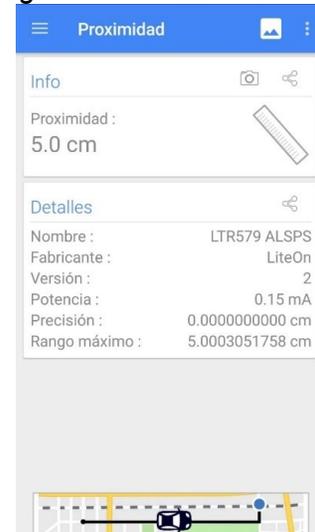
Fuente 8: captura de app Multiherramienta (2019)

Imagen 9. Sensor GPS



Fuente 9: captura de app Multiherramienta (2019)

Imagen 10. Sensor Proximidad



Fuente 10: captura de app Multiherramienta (2019)

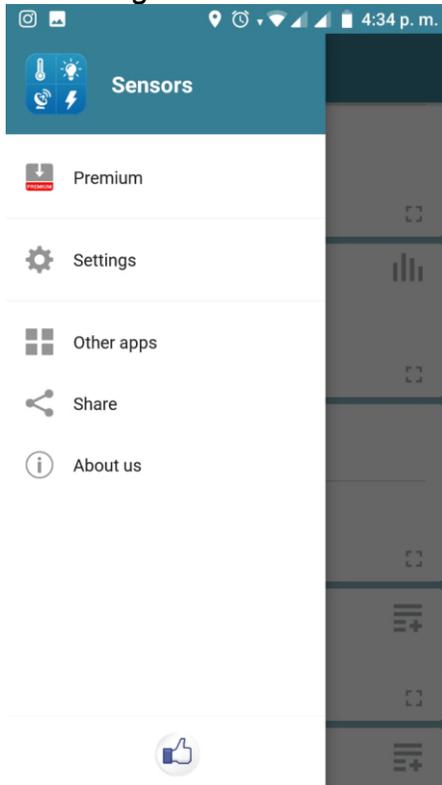
3.2.3 Prueba de Sensores

La aplicación Prueba de sensores fue creada por EXA Tools. Esta aplicación permite visualizar sensores como batería, GPS, acelerómetro, giroscopio, magnetómetro, proximidad, entre otros (imagen 11 y 12).

- La aplicación únicamente permite visualizar el dato capturado en ese instante de tiempo.

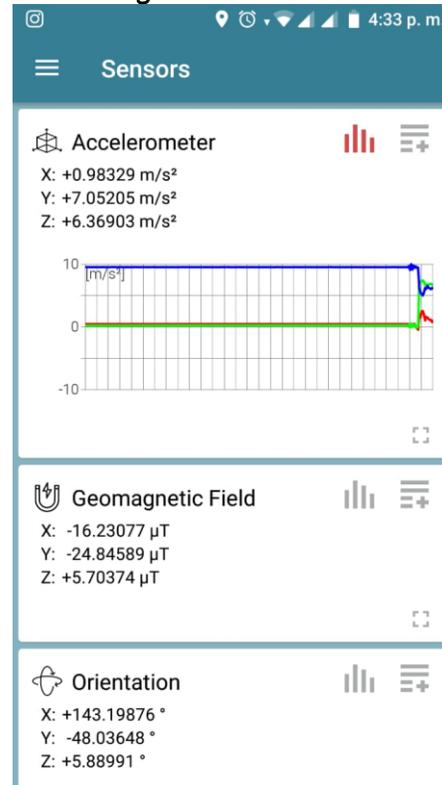
- No permite se parametrize un lapso o secuencias de **tiempo** para la lectura de los sensores en tiempos determinados.
- La aplicación no permite una opción para registrar **magnetómetro**.
- La aplicación no almacena la información en ningún archivo para el usuario, ni le permite compartir un dato

Imagen 11. Menú Sensores



Fuente 11: captura de app prueba de Sensores (2019)

Imagen 12. Sensores



Fuente 12: captura de app prueba de Sensores (2019)

A continuación (Tabla 1), se realiza una comparación por medio del proceso benchmarking entre las aplicaciones relacionadas en el estado del arte con la aplicación SensorReader y sus funciones.

Tabla 1. Benchmarking

App	ACELERÓMETRO	PROXIMIDAD	GIROSCOPIO	GPS	MAGNETOMETRO	BIOMETRIA	REGISTRO DE TIEMPO EN LECTURA	PARAMETRIZACION PARA LECTURA	RECOLECCION DE DATOS EN FORMATO DE TEXTO
Sensor Kinetics	X	X	X		X		X		X
Sensores Multiherramienta	X	X	X		X				
Prueba de Sensores	X	X	X	X					
SensorReader	X	X	X	X	X		X	X	X

Fuente 13: elaboración propia (2019)

4. DESARROLLO DEL PROYECTO

Para la ejecución del trabajo de grado se adaptó la metodología de desarrollo ágil Scrum, se emplearon las fases indicadas en las técnicas y los despliegues correspondientes.

4.1 DESCRIPCIÓN DE LA APLICACIÓN

SensorReader es una aplicación desarrollada en este trabajo de grado para la recolección de datos que permitirán al grupo de investigación LACSER el entrenamiento de los modelos de aprendizaje automático previstos en el proyecto de investigación para implementar al proyecto “Sistema de comunicación para sobrevivientes de un desastre basado en una red ad-hoc de teléfonos inteligentes”. La aplicación SensorReader en este trabajo de grado le ofrecerá al usuario acciones como la selección de los sensores que se leerán, visualización de sensores trabajando y la parametrización de tiempo en el que realizará la captura de datos, además de permitirle la visualización de ubicación del archivo de texto generado.

4.2 FASE DE INICIACIÓN Y AGENDAMIENTO

Para el proyecto se establecieron los siguientes roles:

- **Product Owner:** María del Pilar Salamanca.

- **Scrum Master:** Juan Camilo Ramírez.

- **Development Team:** Sonia Mancipe Ramírez.

Mediante las reuniones coordinadas se definieron las entregas y formas de entrega con el equipo. A continuación, se listan y describen los conceptos usados para el levantamiento de las historias de usuario que dan solución a los requerimientos, en la imagen 13 se puede observar la estructura de datos que lleva la plantilla de historia de usuarios.

Imagen 13. Plantilla historia de usuarios

Id	Nombre
_____	_____
Descripción	
Yo como: _____	
Quiero: _____	
Con el objetivo: _____	
Criterios de aceptación	
Escenario: _____	
Cuando _____	
Entonces _____	

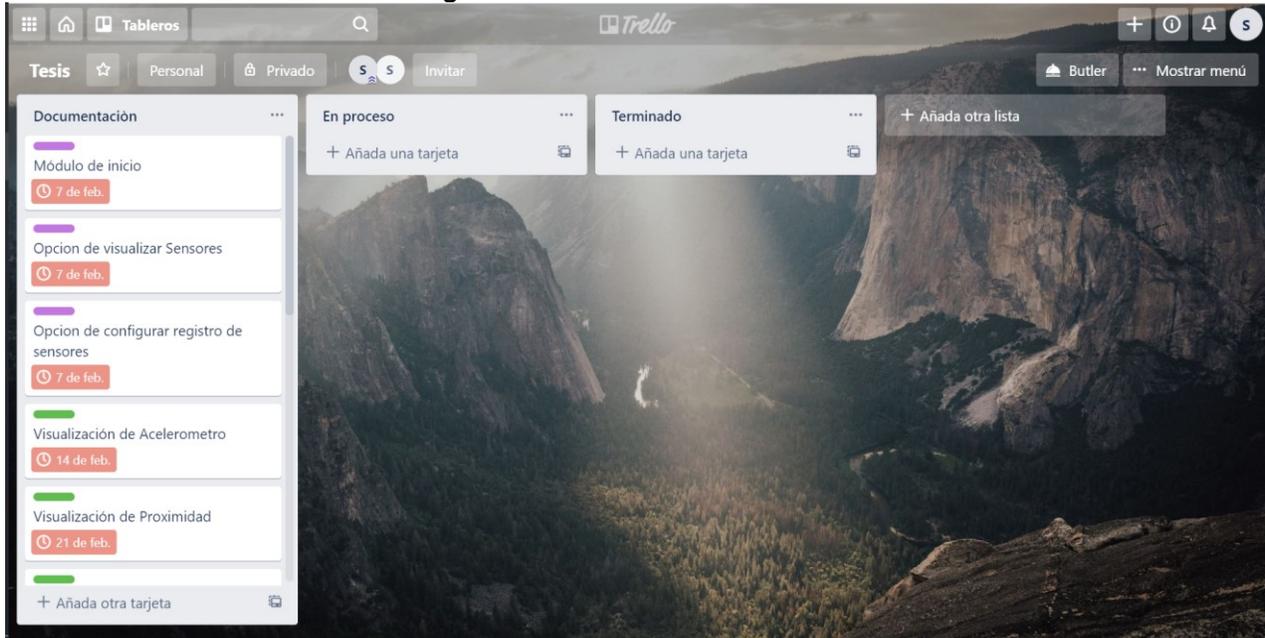
Fuente 14: elaboración propia

- **Id:** código que hace referencia a una única historia de usuario, identificador.
- **Nombre:** texto que hace referencia a la función de la historia de usuario.
- **Descripción:** breve explicación por medio de interrogantes que dan solución a una detallada información sobre lo esperado:
 - **“Yo como:”** ¿Quién usará esta función o para quien es esta funcionalidad?
 - **“Quiero:”** ¿Qué espera como salida el usuario para quien se dirige esta función?
 - **“Con el objetivo:”** ¿Con que intención o con que finalidad se hace esto?
- **Criterios de aceptación:** define el alcance para comparar y concluir si la salida cumple con lo esperado
 - **“Escenario”** Objeto sobre el cual se realiza la funcionalidad. Eje: “Ingreso exitoso”.
 - **“Cuando”** Momento en el que la acción debe responder. Eje: “Cuando el ingreso del usuario y contraseña son correctos”
 - **“Entonces”** Evento o respuesta que el sistema da al usuario. Eje: “Entonces el sistema permitirá el ingreso”

Con el propósito de llevar un orden de las tareas que darían respuesta a los requerimientos, se manejaron las plantillas de Trello (Imagen 14 y 15), donde por spring se generaron etiquetas con colores las cuales se podían organizar en tres casillas de:

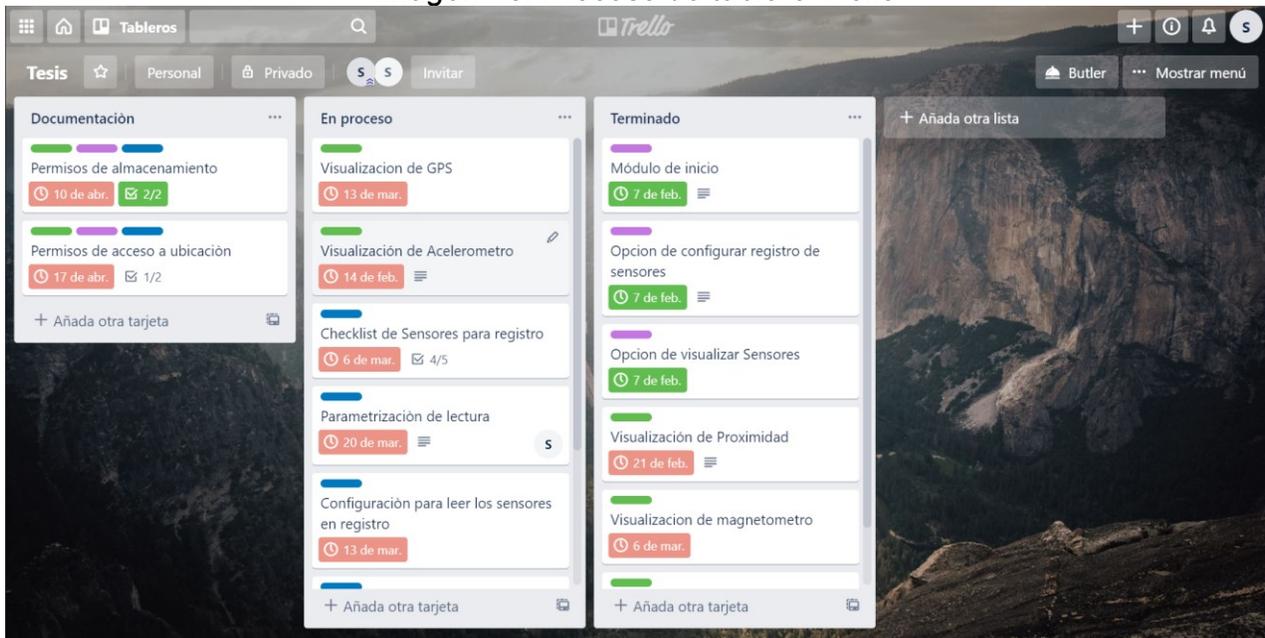
- Documentación, si únicamente está definido el requerimiento
- En proceso, si ya inicio el desarrollo y pruebas del requerimiento.
- Terminado, si se cumplió con el requerimiento.

Imagen 14. Inicio de Tablero Trello



Fuente 15: elaboración propia sobre plataforma

Imagen 15. Proceso de tablero Trello



Fuente 16: elaboración propia sobre plataforma

4.3 PLANEACIÓN Y ESTIMACIÓN

4.3.1 Roles del sistema

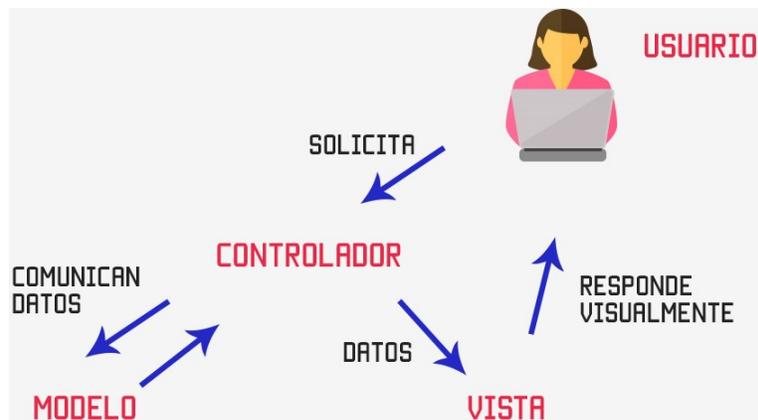
- **Usuario:** el usuario que da uso a la aplicación y selecciona registrar la información generada por los sensores tiene la potestad de ser la única persona con el acceso a los datos en primera instancia ya que quedan registrados en su almacenamiento externo, adicional del usuario depende si en otras aplicaciones o a otras personas se le comparte la información que ha recopilado. Tienen completa disponibilidad de las funciones que incluye la aplicación para poseer como visualizar o parametrizar lecturas.

4.3.2 Arquitectura y herramientas del sistema

Teniendo en cuenta que durante el desarrollo y después de ello se pueden presentar cambios en interfaz o lógica de la aplicación se adaptó el patrón MVC (Modelo, Vista, Controlado), que permitió tener por separado las funcionalidades del programa para dar facilidad en el momento de ejecutar un cambio o entender la codificación.

- Modelo: funciones para generar información (Sensores).
- Vista: interfaz gráfica (XML).
- Controlador: recibe las acciones a ejecutar y envía la solicitud para que se lleve a cabo la función, como se puede observar en la imagen 16 (Selección.class).

Imagen 16. MVC



Fuente 17: tomado de código facilito MVC Model, View, Controller (2015)

4.3.3 Listado de historias de usuario

Con el objetivo de planificar la ejecución de los requerimientos para la aplicación, se realizó el Product Backlog como se puede observar en la tabla 2 donde se listan cada una de las actividades a realizar con prioridad, tiempo estimado e id.

Tabla 2. Product Backlog

ID	Historia	Spring	Tiempo estimado (horas)	Prioridad
1	Pantalla principal - Menú de opciones	1	8	Baja
2	Opción de visualizar sensores	2	12	Baja
3	Mostrar Acelerómetro	2	48	Media
4	Mostrar Proximidad	2	48	Media
5	Mostrar Giroscopio	2	48	Media
6	Mostrar Magnetómetro	2	48	Media
7	Mostrar GPS	2	48	Media
8	Seleccionar sensores a guardar	3	46	Media
9	Ingresar tiempo (Segundos) para registro	4	40	Alta
10	Generar archivo de texto	5	60	Alta

Fuente 18: elaboración propia

4.3.4 Plan de pruebas

El objetivo del plan de pruebas es validar el cumplimiento y funcionamiento adecuado de la aplicación basados en los criterios de calidad establecidos y los resultados obtenidos de los diferentes planes de prueba.

- **Pruebas unitarias:** se verificó la lógica, funcionalidades y especificaciones del software ejecutado. Estas pruebas son una técnica de caja blanda ejecutadas por el desarrollador.
- **Pruebas de integración:** se realizaron pruebas conjuntas/completas sobre la conexión entre las partes del software desarrollado (sensores, interfaces), funcionando conjuntamente y la interacción entre el software y los recursos del hardware. Estas pruebas son técnicas de caja negra/caja gris ejecutadas por el equipo de desarrollo que en este caso es una persona.
- **Pruebas de sistema:** se verificó que las funcionalidades de la aplicación respondieran como lo indicaban los requerimientos. Estas pruebas son una técnica de caja negra ejecutadas por personas particulares que no conocían el desarrollo.
- **Pruebas de aceptación:** se compartió la aplicación para que validaran si la app funcionaba sin novedades de acuerdo con lo que se esperaba. Estas pruebas son una técnica de caja negra ejecutadas por personas particulares que para este caso no conocían la app.

4.4 IMPLEMENTACIÓN Y DESARROLLO

A partir de este punto se plantea la ejecución de las actividades realizadas en cada sprint con el propósito de alcanzar el objetivo propuesto con la aplicación.

4.4.1 Sprint 1: módulo de inicio

Durante el Sprint 1 se llevó acabo el módulo de inicio, el cual permite al usuario dar la perceptibilidad para escoger la actividad a llevar a cabo, entre visualizar los sensores en ejecución o configurar el registro de los sensores.

4.4.1.1 Requerimientos

Con el propósito de ampliar información sobre el módulo del menú de opciones de la aplicación, a continuación, se relaciona el requerimiento correspondiente (Tabla 3).

Tabla 3. Requerimiento de la aplicación para Pantalla principal

Id	Nombre
1	Pantalla principal - Menú de opciones
Descripción	
Yo como: usuario final	
Quiero: poder seleccionar qué hacer en la aplicación	
Con el objetivo: de tener la opción de visualizar primero los sensores para saber qué datos se van a recolectar o hacer directamente el registro	
Criterios de aceptación	
Escenario: opción de acciones	
Cuando: el usuario ingresó a la aplicación	
Entonces: debe poder seleccionar qué hacer	

Fuente 19: elaboración propia

4.4.1.2 Mockups

A continuación, se presenta el diseño de la aplicación correspondiente a la primera pantalla de la app (imagen 17).

Imagen 17. Módulo de inicio

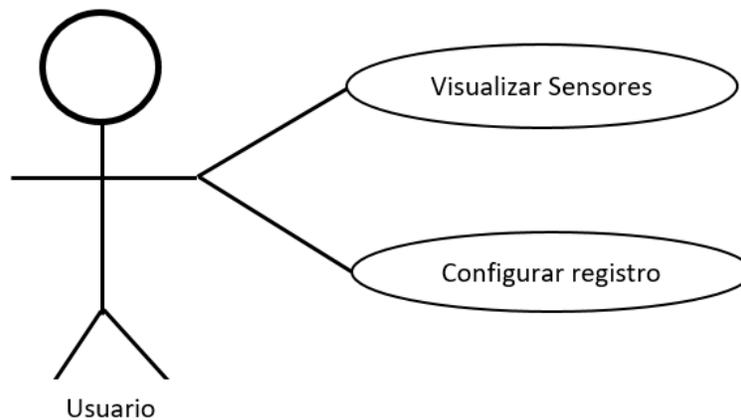


Fuente 20: elaboración propia

4.4.1.3 Casos de uso

Para el módulo de inicio de la aplicación, a continuación, se registra el diseño del caso de uso correspondiente, el cual indica que en la primera pantalla el único perfil (usuario) tiene acceso a los módulos de “Visualizar Sensores” o “Configurar registro” (imagen 18).

Imagen 18. Ejecutar aplicación

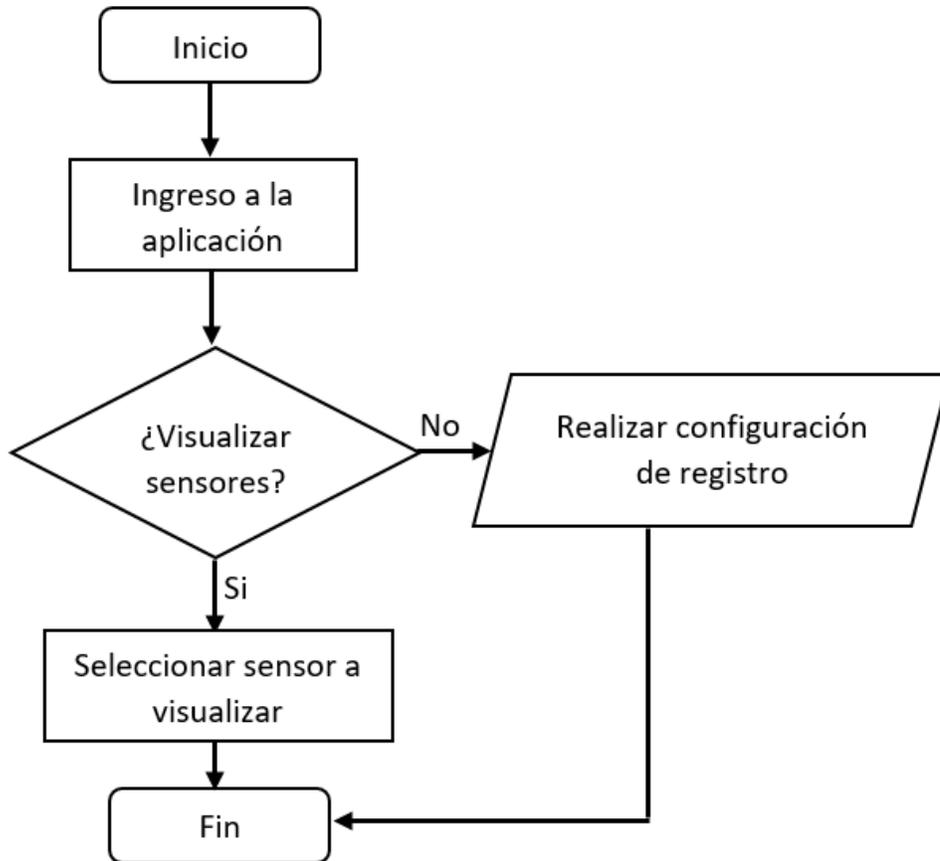


Fuente 21: elaboración propia

4.4.1.4 Diagrama de actividades

A continuación, se diseña el diagrama correspondiente al módulo de inicio de la aplicación (imagen 19), en el que se detalla cómo funciona el flujo de pasos en la aplicación SensorReader y los posibles caminos.

Imagen 19. Diagrama de actividades Sprint 1



Fuente 22: elaboración propia

4.4.1.5 Pruebas funcionales

A continuación, se muestra el caso de pruebas funcionales correspondiente al módulo de inicio, el requerimiento respectivo se observa en la tabla Menú pantalla de inicio Tabla 4.

Tabla 4. Caso de prueba funcional pantalla de inicio

Id	Nombre caso de prueba			
1	Menú pantalla de inicio			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que una vez el usuario ingresa a la aplicación, en la pantalla de inicio debe tener las opciones definidas (Visualización y configuración).			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO Android 5.0 o superior.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Ingresar a la app	La aplicación debe mostrar dos botones habilitados (visualizar/configuración) que apuntan a dos módulos diferentes.	Acertado

Fuente 23: elaboración propia

4.4.2 Sprint 2: módulo de visualizar sensores

Durante el Sprint 2 se llevó acabo el módulo de visualizar sensores, el cual permite al usuario observar los sensores trabajando. El usuario tiene la disponibilidad de visualizar uno a uno los sensores.

4.4.2.1 Requerimientos

Con el propósito de ampliar información sobre el módulo de visualización de sensores de la aplicación, a continuación, se relaciona el requerimiento correspondiente al módulo de visualización que permite visualizar las coordenadas de cada sensor (GPS, Acelerómetro, Magnetómetro, Proximidad y Giroscopio) (Tabla 5, 6, 7, 8, 9 y 10).

Tabla 5. Visualizar sensores

Id	Nombre
2	Visualizar sensores
Descripción	
Yo como: usuario final	
Quiero: poder visualizar cada sensor	
Con el objetivo: de observar que datos se están generando	
Criterios de aceptación	
Escenario: visualizar sensores	
Cuando: el usuario dio clic en el botón 'Visualizar sensores' de la pantalla principal	
Entonces: debe poder seleccionar que sensor observar uno por uno	

Fuente 24: elaboración propia

Tabla 6. Mostrar Acelerómetro

Id	Nombre
3	Mostrar Acelerómetro
Descripción	
Yo como: usuario final	
Quiero: poder seleccionar el sensor Acelerómetro	
Con el objetivo: de verlo en ejecución generando valores en los 3 ejes	
Criterios de aceptación	
Escenario: visualizar Acelerómetro	
Cuando: el usuario dio clic en el botón 'Acelerómetro'	
Entonces: debe poder ver los datos que está generando el sensor en ese instante de tiempo	

Fuente 25: elaboración propia

Tabla 7. Mostrar Proximidad

Id	Nombre
4	Mostrar Proximidad
Descripción	
Yo como: usuario final	
Quiero: poder seleccionar el sensor proximidad	
Con el objetivo: de verlo en ejecución y saber si hay un objeto cerca o no	
Criterios de aceptación	
Escenario: visualizar Proximidad	
Cuando: el usuario dio clic en el botón 'Proximidad'	
Entonces: debe poder ver si el sensor detecta cercanía o no.	

Fuente 26: elaboración propia

Tabla 8. Mostrar Giroscopio

Id	Nombre
5	Mostrar Giroscopio
Descripción	
Yo como: usuario final	
Quiero: poder seleccionar el sensor Giroscopio	
Con el objetivo: de verlo en ejecución generando valores en los 3 ejes	
Criterios de aceptación	
Escenario: visualizar Giroscopio	
Cuando: el usuario dio clic en el botón 'Giroscopio'	
Entonces: debe poder visualizar los datos del sensor en ese instante de tiempo	

Fuente 27: elaboración propia

Tabla 9. Mostrar Magnetómetro

Id	Nombre
6	Mostrar Magnetómetro
Descripción	
Yo como: usuario final	
Quiero: poder seleccionar el sensor Magnetómetro	
Con el objetivo: de ver los valores en los 3 ejes y la unidad de campo magnético.	
Criterios de aceptación	
Escenario: visualizar Magnetómetro	
Cuando: el usuario dio clic en el botón 'Magnetómetro'	
Entonces: debe poder visualizar los datos del sensor en ese instante de tiempo.	

Fuente 28: elaboración propia

Tabla 10. Mostrar GPS

Id	Nombre
7	Mostrar GPS
Descripción	
Yo como: usuario final	
Quiero: poder seleccionar el sensor GPS	
Con el objetivo: de verlo en ejecución generando valores en los 2 ejes.	
Criterios de aceptación	
Escenario: visualizar GPS	
Cuando: el usuario dio clic en el botón 'GPS' (dio permisos de acceso a ubicación)	
Entonces: debe poder ver los datos que está generando el sensor en ese instante de tiempo	

Fuente 29: elaboración propia

4.4.2.2 Mockups

A continuación, se presenta el diseño de la aplicación correspondiente a la pantalla de visualización de sensores donde el usuario puede observar uno a uno cada valor generado por los sensores (imagen 20).

Imagen 20. Mostrar sensores

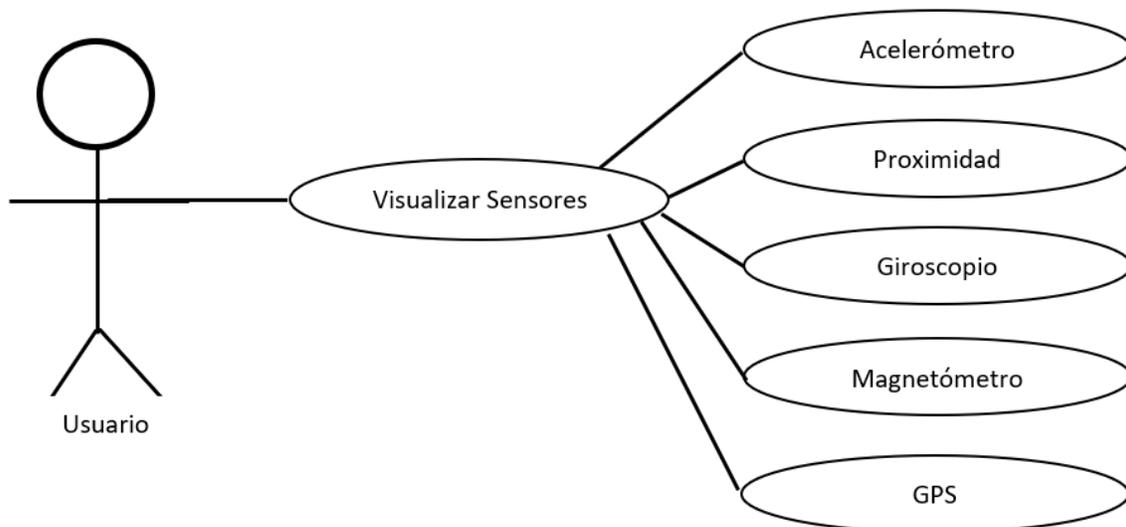


Fuente 30: elaboración propia

4.4.2.3 Casos de uso

Para el módulo de visualización de sensores en la aplicación, a continuación, se registra el diseño del caso de uso correspondiente el cual indica que una vez el usuario ingresa por medio de “visualización sensores” puede ver a mayor detalle los valores de cada sensor (imagen 21) en ese instante de tiempo.

Imagen 21. Visualizar sensores

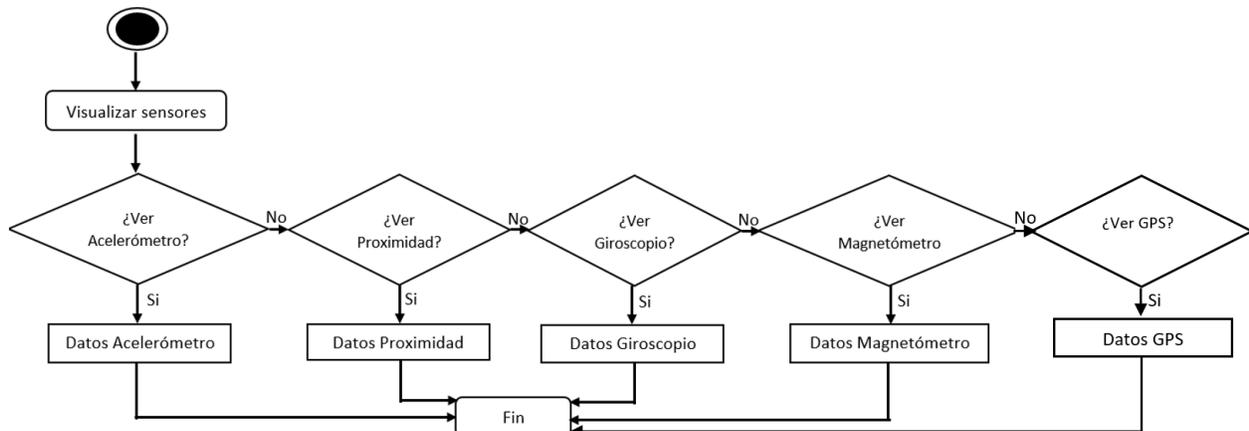


Fuente 31: elaboración propia

4.4.2.4 Diagrama de actividades

A continuación, se diseña el diagrama correspondiente al módulo de visualización de sensores (imagen 22), en el que se indica cómo funciona el flujo de pasos en la aplicación SensorReader para la visualización de valores y los posibles caminos.

Imagen 22. Diagrama de actividades Sprint 2



Fuente 32: elaboración propia

4.4.2.5 Pruebas funcionales

A continuación, se muestra el caso de pruebas funcionales correspondiente al módulo de visualización. El requerimiento respectivo indica paso a paso los módulos probados y sus resultados Tablas 11, 12, 13, 14, 15 y 16.

Tabla 11. Caso de prueba funcional

Id	Nombre caso de prueba			
2	Visualización de sensores			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que el usuario pueda visualizar de manera individual los sensores registrados.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior. Permisos de ubicación.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Visualizar sensores	Listado de sensores a visualizar (Acelerómetro, Proximidad, Giroscopio, Magnetómetro y GPS)	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Dio clic en visualizar sensores, pero no aceptó permisos de ubicación.	Visualizará los botones de los demás sensores menos el GPS y recibe una notificación de que el GPS no está activo.	Acertado

Fuente 33: elaboración propia

Tabla 12. Caso de prueba acelerómetro

Id	Nombre caso de prueba			
3	Visualización de acelerómetro			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que el usuario pueda recibir una respuesta en la aplicación sobre el sensor de acelerómetro			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Seleccionar sensor de acelerómetro en la pantalla de visualización de sensores.	<ul style="list-style-type: none"> ⊕ Descripción corta de la información que puede mostrar el sensor. ⊕ Aceleración en los 3 ejes en tiempo real. 	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Dio clic en visualizar acelerómetro, pero el smartphone no cuenta con el sensor.	Descripción corta de la información que puede mostrar el sensor, pero no habrá información de los ejes, visualizará una nota de "Sensor no disponible".	Acertado

Fuente 34: elaboración propia

Tabla 13. Caso de prueba proximidad

Id	Nombre caso de prueba			
4	Visualización de proximidad			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que el usuario pueda recibir una respuesta en la aplicación sobre el sensor de proximidad.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Seleccionar sensor de proximidad en la pantalla de visualización de sensores.	<ul style="list-style-type: none"> ⊕ La pantalla cambia de color si detecta cercanía (gris claro) o no (gris oscuro). ⊕ Indica 0 cuando hay cercanía. ⊕ Visualiza una nota de detección de contacto o no. 	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Dio clic en visualizar proximidad, pero el smartphone no cuenta con el sensor.	Visualizará una nota de "Sensor no disponible".	Acertado

Fuente 35: elaboración propia

Tabla 14. Caso de prueba giroscopio

Id	Nombre caso de prueba			
5	Visualización de giroscopio			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que el usuario pueda recibir una respuesta en la aplicación sobre el sensor de giroscopio.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Seleccionar sensor de giroscopio en la pantalla de visualización de sensores.	<ul style="list-style-type: none"> ⊕ Descripción corta de la información que puede mostrar el sensor. ⊕ Velocidad de rotación en los 3 ejes en tiempo real. 	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Dio clic en visualizar giroscopio, pero el smartphone no cuenta con el sensor.	Descripción corta de la información que puede mostrar el sensor, pero no habrá información de los ejes, visualizará una nota de "Sensor no disponible".	Acertado

Fuente 36: elaboración propia

Tabla 15. Caso de prueba magnetómetro

Id	Nombre caso de prueba			
6	Visualización de magnetómetro			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que el usuario pueda recibir una respuesta en la aplicación sobre el sensor de magnetómetro.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Seleccionar sensor de magnetómetro en la pantalla de visualización de sensores.	<ul style="list-style-type: none"> ⊕ Descripción corta de la información que puede mostrar el sensor. ⊕ Orientación del campo magnético en los 3 ejes en tiempo real. ⊕ Total de teslas. 	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Dio clic en visualizar magnetómetro, pero el smartphone no cuenta con el sensor.	Descripción corta de la información que puede mostrar el sensor y valor por defecto de teslas, pero no habrá información del campo magnético en los 3 ejes, por lo anterior visualizará una nota de "Sensor no disponible".	Acertado

Fuente 37: elaboración propia

Tabla 16. Caso de prueba GPS

Id	Nombre caso de prueba			
7	Visualización de GPS			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que el usuario pueda recibir una respuesta en la aplicación sobre el GPS.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior. Permisos de ubicación.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Seleccionar GPS en la pantalla de visualización de sensores.	<ul style="list-style-type: none"> ⊕ Descripción corta de la información que puede mostrar el sensor. ⊕ Geolocalización actual, coordenadas en latitud y longitud. 	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Dio clic en visualizar sensores, pero no aceptó permisos de ubicación.	El botón de GPS no estará visible.	Acertado

Fuente 38: elaboración propia

4.4.3 Sprint 3: módulo de configuración

Durante el Sprint 3 se realizó parte del módulo de configuración, el cual permite al usuario definir la configuración para el registro de los datos en el archivo CSV.

4.4.3.1 Requerimientos

Con el propósito de ampliar información sobre el módulo de configuración de la aplicación, a continuación, se relaciona el requerimiento correspondiente (Tabla 17).

Tabla 17. Configuración de Registro Sensores

Id	Nombre
7	Configuración de registro para sensores
Descripción	
Yo como: usuario final	
Quiero: poder seleccionar los sensores a registrar	
Con el objetivo: de recolectar únicamente los sensores necesarios	
Criterios de aceptación	
Escenario: selección de sensores	
Cuando: el usuario dio clic en el checklist del sensor	
Entonces: deben registrar en el archivo solo los sensores seleccionados.	

Fuente 39: elaboración propia

4.4.3.2 Mockups

En la imagen 23 se presenta el diseño de la aplicación correspondiente a la pantalla de configuración de registro para sensores donde el usuario puede seleccionar el checklist de sensores a registrar en el archivo de texto.

Imagen 23. Definición de Sensores para registrar

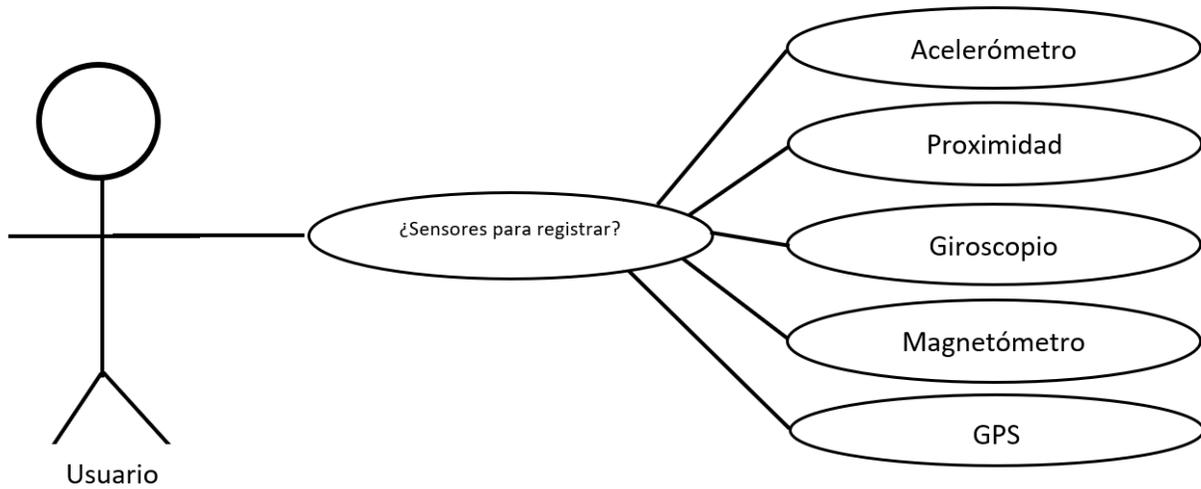


Fuente 40: elaboración propia

4.4.3.3 Casos de uso

Para el módulo de configuración de registro en la aplicación la imagen 24 muestra el diseño del caso de uso correspondiente el cual indica que el usuario puede seleccionar que sensor requiere para registrar.

Imagen 24. Seleccionar sensores a registrar

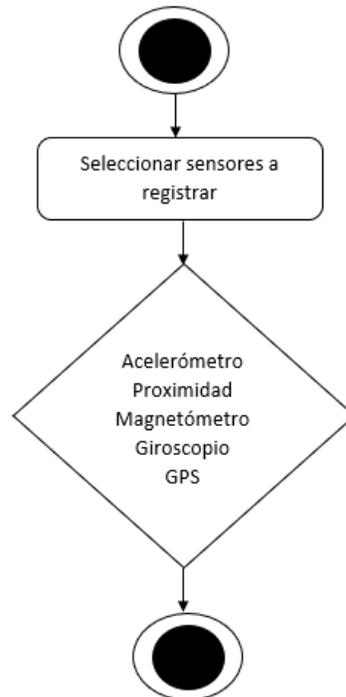


Fuente 41: elaboración propia

4.4.3.4 Diagrama de actividades

A continuación, se diseña el diagrama correspondiente al módulo de configuración de registro de sensores (imagen 25), en el que se indica el orden en el que se llega al checklist de sensores a registrar.

Imagen 25. Diagrama de actividades Sprint 3



Fuente 42: elaboración propia

4.4.3.5 Pruebas funcionales

A continuación, se muestra el caso de pruebas funcionales correspondiente al módulo de configuración de sensores para registro, el requerimiento respectivo se observa en la tabla 18 Caso de prueba funcional definición de Sensores.

Tabla 18. Caso de prueba funcional definición de Sensores

Id	Nombre caso de prueba			
8	Seleccionar sensores a registrar			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que al usuario si se le permita seleccionar los sensores que requiera recolectar.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior. Permisos de almacenamiento y ubicación.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Seleccionar sensores	Una vez se genere el archivo de los sensores, únicamente deben estar los seleccionados en el checklist	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	No seleccionó ningún sensor	No podrá iniciar el registro y la aplicación le notificará que debe seleccionar un sensor a registrar.	Acertado
1	Usuario	No aceptó permisos de ubicación.	No estará disponible para seleccionar en el checklist el GPS.	Acertado
1	Usuario	No aceptó permisos de almacenamiento.	No estará disponible ningún campo para registro. La pantalla de la aplicación estará completamente bloqueada y le notificará: no es posible realizar registro ya que no se aceptaron permisos.	Acertado

Fuente 43: elaboración propia

4.4.4 Sprint 4: módulo para configurar tiempo de registro

Durante el Sprint 4 se realizó la siguiente parte del módulo de configuración, el cual hace referencia a la definición del parámetro de tiempo para registro de los datos de sensores.

4.4.4.1 Requerimientos

Con el propósito de ampliar la información sobre el módulo de configuración de registro de sensores de la aplicación, a continuación, se relaciona el requerimiento correspondiente a la parametrización del tiempo (Tabla 19).

Tabla 19. Configuración de Registro Tiempo

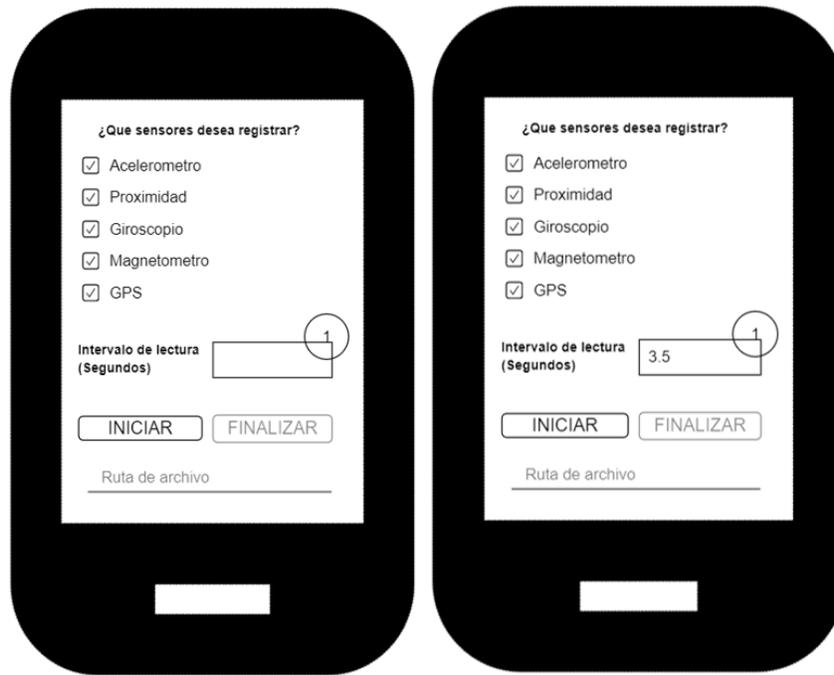
Id	Nombre
8	Parametrización de tiempo para registro
Descripción	
Yo como: usuario final	
Quiero: poder definir cada cuantos segundos se realizarán los registros de sensores	
Con el objetivo: de recolectar los sensores en el tiempo definido.	
Criterios de aceptación	
Escenario: definición de tiempo	
Cuando: el usuario da clic en el campo de intervalo e ingresa cada cuantos segundos se deben realizar las lecturas	
Entonces: debe quedar en el archivo la lectura de sensores solo cuando se cumple el tiempo definido.	

Fuente 44: elaboración propia

4.4.4.2 Mockups

A continuación, se presenta el diseño de la aplicación correspondiente a la pantalla configuración de registro de sensores donde el usuario puede realizar checklist de sensores (Sprint 3) y parametrizar el tiempo de lectura (segundos) como se puede observar en la imagen 26.

Imagen 26. Parametrización de tiempo para registro

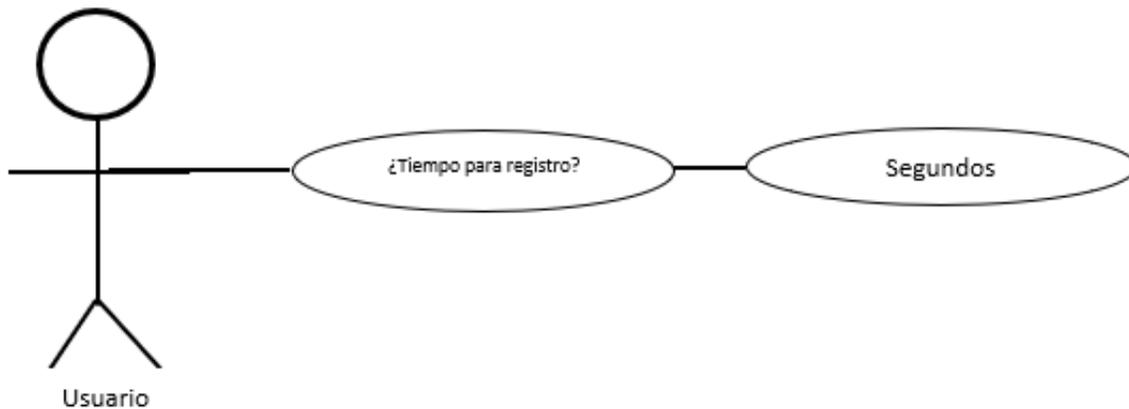


Fuente 45: elaboración propia

4.4.4.3 Casos de uso

Para el módulo de configuración de registro en la aplicación, a continuación, se diseña el caso de uso correspondiente, el cual indica que el usuario puede realizar parametrización de tiempo ingresando la cantidad en segundos, como se observa en la imagen 27.

Imagen 27. Parametrizar tiempo

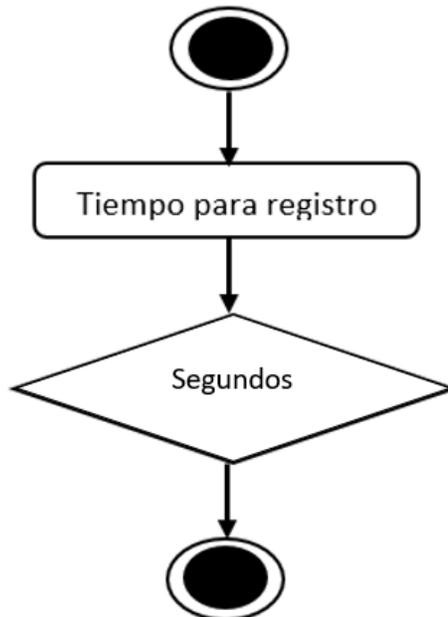


Fuente 46: elaboración propia

4.4.4.4 Diagrama de actividades

La imagen 28 muestra el diseño del diagrama correspondiente al módulo de configuración de registro de sensores en el que se indica el orden en el que se realiza registro de tiempo de lectura de sensores.

Imagen 28. Diagrama de actividades Sprint 4



Fuente 47: elaboración propia

4.4.4.5 Pruebas funcionales

La tabla 20 muestra el caso de pruebas funcionales correspondiente al módulo de configuración de registro de sensores, el requerimiento respectivo el módulo probado y sus resultados.

Tabla 20. Caso de prueba funcional parametrización de tiempo

Id	Nombre caso de prueba			
9	Parametrización de tiempo para registro			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que la parametrización definida por el usuario en la recolección de sensores si se efectúa en el tiempo indicado.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior. Permisos de almacenamiento.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Definición de intervalo de segundos	Una vez se genere el archivo de los sensores únicamente deben estar los registros en el tiempo definido, cada vez que pasaba la cantidad de segundos ingresados se debió generar una nueva línea de información.	Acertado
Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	No ingresó el intervalo se segundos	No podrá iniciar el registro y la aplicación le notificará que debe ingresar el intervalo de segundos para la lectura.	Acertado
1	Usuario	No aceptó permisos de almacenamiento.	No estará disponible ningún campo para registro. La pantalla de la aplicación estará completamente bloqueada y le notificará: no es posible realizar registro ya que no se aceptaron permisos.	Acertado

Fuente 48: elaboración propia

4.4.5 Sprint 5: módulo para generar archivo de texto

Durante el Sprint 5 se realizó la última parte del módulo de configuración, el cual hace referencia a la generación del archivo de registro de sensores.

4.4.5.1 Requerimientos

Con el propósito de ampliar información sobre el módulo de configuración de la aplicación, en la tabla 21 se puede observar el requerimiento correspondiente a la generación de archivo de texto.

Tabla 21. Generación de archivo de texto

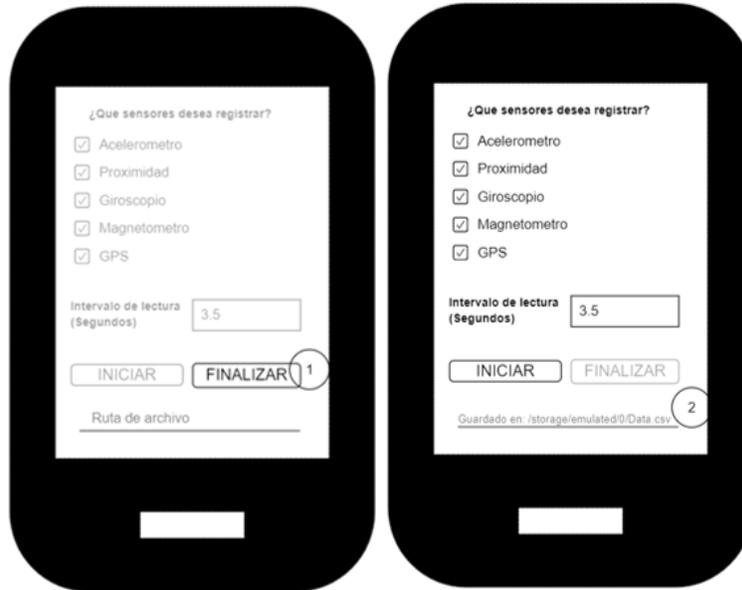
Id	Nombre
9	Generar archivo de texto
Descripción	
Yo como: usuario final	
Quiero: generar un archivo de texto con los sensores seleccionados y datos adicionales como tiempo (hora de lectura) y batería	
Con el objetivo: de tener un registro de la información y guardar el archivo en el almacenamiento externo de mi smartphone.	
Criterios de aceptación	
Escenario: generar registro	
Cuando: el usuario inicia la lectura y da clic en el botón de finalizar	
Entonces: debe generar un archivo con la información solicitada anteriormente y poderlo guardar en el almacenamiento externo del smartphone.	

Fuente 49: elaboración propia

4.4.5.2 Mockups

A continuación, se presenta el diseño de la aplicación correspondiente a la pantalla de configuración de registro donde el usuario debe primero iniciar lectura (cumpliendo con el Sprint 3 y 4) y finalizar la lectura para generar el archivo de texto como se observa en la imagen 29.

Imagen 29. Generar archivo de texto

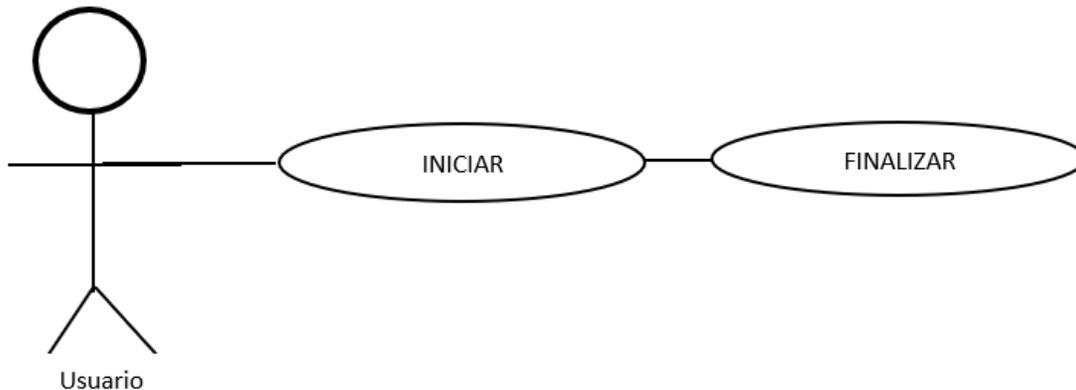


Fuente 50: elaboración propia

4.4.5.3 Casos de uso

La imagen 30 muestra el diseño del caso de uso correspondiente a la generación del archivo en formato de texto, el cual indica que el usuario primero debe iniciar la lectura y finalizar después de haberse cumplido mínimo una vez el tiempo indicado.

Imagen 30. Ejecutar acciones de la app



Fuente 51: elaboración propia

4.4.5.4 Diagrama de actividades

La imagen 31 muestra el diagrama correspondiente al módulo de configuración de registro de sensores, en el que se indica el orden en el que se llega a la generación del archivo de texto.

Imagen 31. Diagrama de actividades Sprint 5



Fuente 52: elaboración propia

4.4.5.5 Pruebas funcionales

Las tablas 22 y 23 muestran el caso de pruebas funcionales correspondiente al módulo de configuración de registro de sensores con la generación del archivo de texto, el requerimiento respectivo el módulo probado y sus resultados.

Tabla 22. Caso de prueba funcional Sprint 5

Id	Nombre caso de prueba			
10	Generar archivo			
Usuario	Usuario final			
Estado de prueba	Pasada			
Desarrollador	Sonia Mancipe			
Descripción	Esta prueba permite validar/verificar que se esté permitiendo la generación del archivo de texto.			
Requisitos para la prueba	La aplicación debe estar instalada en el equipo que se probará con SO 5.0 o superior. Permisos de almacenamiento.			
Flujo principal				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	Clic INICIAR	<ul style="list-style-type: none"> ⊕ Una vez se dé clic en el botón iniciar se bloquean todos los botones de checklist, intervalo de segundos e iniciar. ⊕ Se inició la lectura. ⊕ Se activa el botón de finalizar. 	Acertado
2	Usuario	Clic FINALIZAR	<ul style="list-style-type: none"> ⊕ Una vez se dé clic en el botón finalizar se desbloquean los botones de checklist, intervalo de segundos e iniciar. ⊕ Se generó el archivo de texto y se indica la ruta de almacenamiento. ⊕ Se bloquea el botón de finalizar. 	Acertado

Fuente 53: elaboración propia

Tabla 23 Parte 2 Caso de prueba funcional Sprint 5

Flujo de excepción				
Paso	Actor	Acción	Resultados esperados	Resultado
1	Usuario	No aceptó permisos de almacenamiento.	<ul style="list-style-type: none"> ⊕ No estará disponible ningún campo para registro. ⊕ La pantalla de la aplicación estará completamente bloqueada y le notificará: no es posible realizar registro porque no se aceptaron permisos. 	Acertado
2	Usuario	No dio clic en INICIAR.	No podrá iniciar el registro y la aplicación no ejecutará ningún proceso.	Acertado
3	Usuario	Dio clic en FINALIZAR y no había superado el primer ciclo del tiempo.	La aplicación esperara a que se cumpla el primer ciclo de lectura para generar el archivo.	Acertado
4	Usuario	Cerró completamente la aplicación después de dar clic en INICIAR a una nueva lectura, sin FINALIZAR.	La aplicación no generará ningún archivo del tiempo que había transcurrido.	Acertado

Fuente 54: elaboración propia

5. RESULTADOS OBTENIDOS

Durante el desarrollo del trabajo de grado se logró la aplicación SensorReader la cual está en sistema operativo Android y permite al usuario la recolección de los datos generados por los sensores del smartphone en el que se ejecuta la aplicación. Además, SensorReader permite configurar la recolección de los datos de los sensores de proximidad, acelerómetro, giroscopio, GPS y magnetómetro; parametrizar el tiempo en el que se debe realizar la lectura y almacenarlo en un archivo con formato de texto (CSV), dentro de las pruebas ejecutadas el máximo tiempo que estuvo en ejecución una lectura fue 6 horas. Además de permitirle a los usuarios la selección de los sensores a recolectar para ser analizados.

Como trabajo opcional se ideó inicialmente el entrenamiento de un modelo de aprendizaje automático (utilizando máquinas de vectores de soporte o redes neuronales, por ejemplo) sobre un conjunto de datos recopilados con la app SensorReader, esta fue una tarea opcional que el estudiante desarrollador contempló realizar, pero por la cobertura del tiempo y los retardos durante el desarrollo del proyecto no fue posible, se espera a futuro poder realizar este trabajo de entrenamiento.

En el capítulo de implementación y desarrollo se realizaron las pruebas correspondientes a pruebas unitarias y de integración durante cada spring. Como es necesario probar las funcionalidades al integrar toda la aplicación, se ejecutaron las pruebas sistema y aceptación, en las cuales se validó que la aplicación fuese aceptada por los smartphones, validando también la ejecución de los módulos. Para estas pruebas se compartió la aplicación con dos personas a quienes se les explicó cómo debía responder la aplicación en general y se les pidió instalarla en sus smartphones. Una vez realizadas las pruebas se observó lo siguiente:

1. Estudiante de Ingeniería

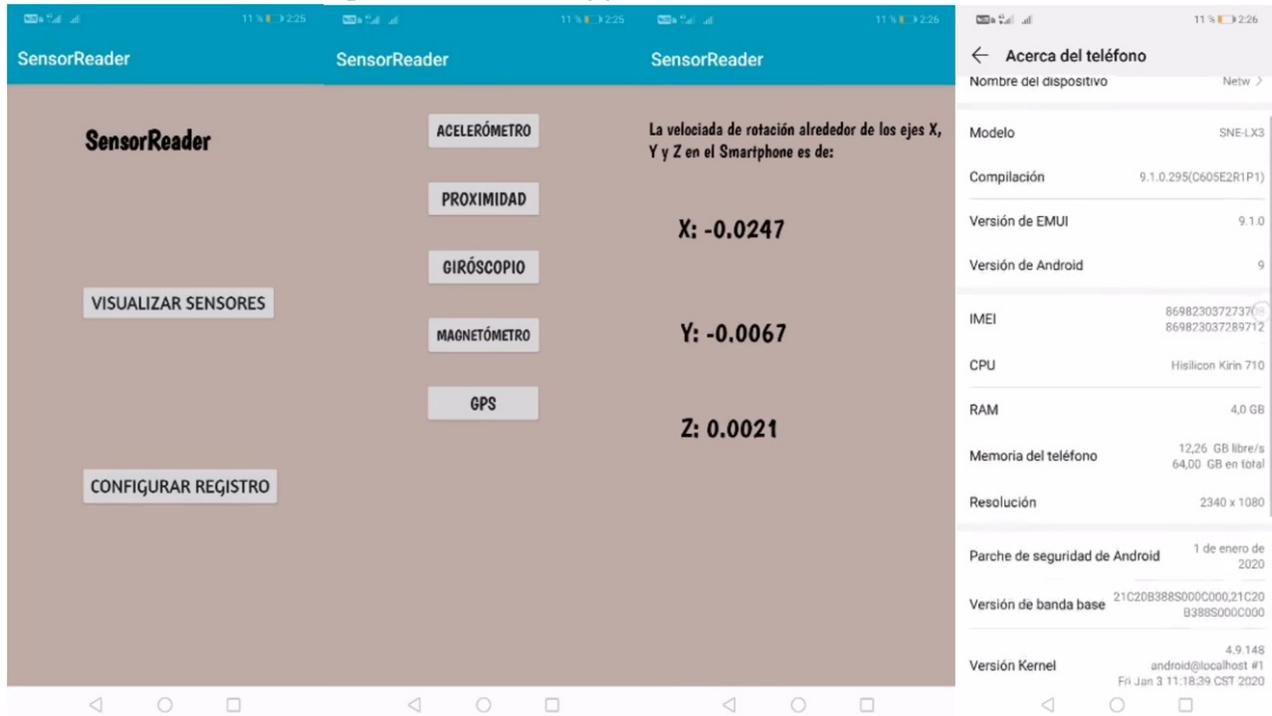
Referencia: Huawei SNE-LX1

Versión de Android: 9

Dimensiones: 2340 x 1080

Resultado: la aplicación se ejecuta correctamente, el usuario tiene uso total de los módulos del aplicativo con correcto funcionamiento y por medio de videos nos permite comprobar la funcionalidad y recolección de evidencias. Ver imagen 32.

Imagen 32 Prueba app SensorReader usuario 1



Fuente 55: elaboración propia

2. Estudiante de administración

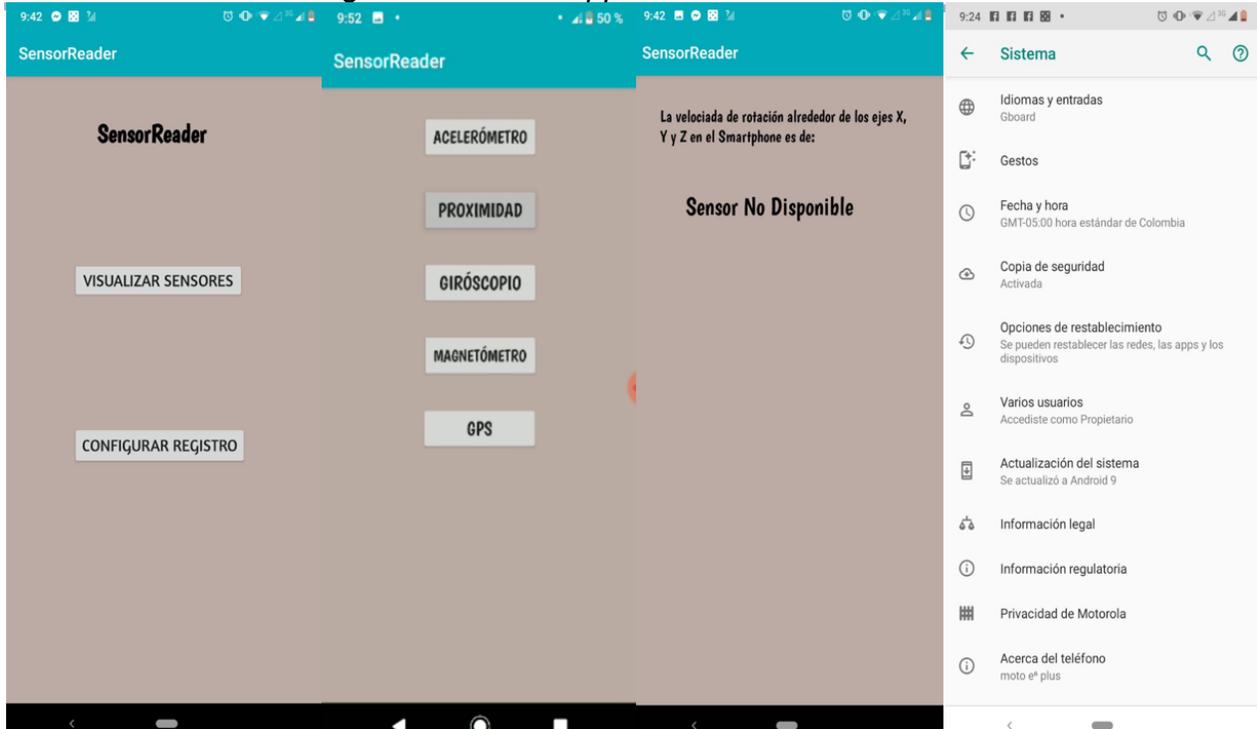
Referencia: Moto e plus

Versión de Android: 9

Dimensiones: 720 x 1280

Resultado: la aplicación se ejecuta correctamente. El usuario tiene uso total de los módulos del aplicativo con correcto funcionamiento y por medio de videos permite comprobar la funcionalidad y recolección de evidencias, ver imagen 33. Con esta prueba se validó la funcionalidad cuando un smartphone no cuenta con un sensor.

Imagen 33 Prueba app SensorReader usuario 2



Fuente 56: elaboración propia

6. CONCLUSIONES Y RECOMENDACIONES

Con la finalización del presente trabajo de grado se llega a las siguientes conclusiones:

- Durante el desarrollo del proyecto, el módulo de generación de archivo fue principalmente el más complejo, el que tomó más tiempo, teniendo en cuenta que el archivo tenía que alimentar más de una línea de datos y que debía realizarse cuando se cumpliera el tiempo determinado. Fue complejo llegar al proceso de lograr incluir la condición del tiempo en la lectura y después de esto, que la lectura se realizara varias veces, pero en diferentes líneas. Como solución se creó un hilo, se usaron funciones como `BufferedWriter` del lenguaje de programación y se almacenó la información en memoria, para que cuando el usuario finalizara la lectura, esa información se grabara en el archivo de texto.
- Se considera un hallazgo el patrón que llevan los sensores, ya que previo al desarrollo de la aplicación se estudiaron de manera independiente cada sensor. Cuando llegó la hora de implementación se identificó cómo todos los sensores requerían activar un mismo servicio, llamar cada uno su tipo de sensor y activar/pausar cada sensor.
- La aplicación `SensorReader` es de gran ayuda para quien esté interesado en saber cómo funcionan los sensores de un smartphone o para quien requiera realizar un análisis de estos datos, como lo es el grupo de investigación LACSER de la Universidad Antonio Nariño.
- La aplicación propone ser clara de entender para los usuarios, generándoles notificaciones o bloqueos/activaciones a partir de pasos anteriores con el propósito de que el usuario no tenga inconvenientes en cuanto al uso de la app.
- Con el módulo de visualización de sensores la aplicación le permite ver al usuario los sensores en tiempo real, además de darle una pequeña introducción de qué significan los valores que se están reflejando.
- Con el módulo de configuración de registro le permite almacenar únicamente los sensores requeridos, parametrizar cada cuanto se debe realizar la lectura y almacenar la información en un archivo de texto.
- La información recopilada se guarda en el archivo de texto ubicado en el almacenamiento.
- Un trabajo futuro podría ser incluir nuevos sensores y realizar un análisis de los datos generados, aplicando aprendizaje automático para la posterior identificación de actividad humana, a partir de los datos recolectados.

7. BIBLIOGRAFÍA

- Anguita, D., Ghio, A., & Oneto, L. (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. University of Genova & Universitat Politecnica de Catalunya.
- Baldominos, A., Cervantes, A., Sáez, Y., & Isasi, P. (2018). A Comparison of Machine Learning and Deep Learning Techniques for Activity Recognition using Mobile Devices. Sensors.
- Capela, N., Lemaire, E., Baddour, N., Goljar, N & Burger, H. (2016). Evaluation of a Smartphone human activity recognition application with able-bodied and stroke participants. Journal of NeuroEngineering and Rehabilitation.
- Castonguay, V., Lavoie, P., Karazivan, P., Morris, J. & Gagnon, R. (2017). P030: Multisource feedback for emergency medicine residents: different, relevant and useful information. Résumés scientifique 2017.
- Challenger, I., Diaz, Y. & Becerra, R. (2014). The programming language Python. Ciencias Holguín
- Chetty, G., White, M., & Akther, F. (2015). Smart phone based data mining for human activity recognition. Procedia Computer Science, 46, 1181–1187.
- Han, M., Hun, Jac., Nugent, Chris., McClean, Sally. & Lee, Sungyoung. (2014). A Lightweight Hierarchical Activity Recognition Framework Using Smartphone Sensors. Molecular Diversity Preservation International.
- Hernández, R., Fernández, C. & Baptista, P. (2017). Metodología de la investigación. Mc Graw Hill.
- Honores, L. & Vizquete, J. (2014). Estudio estadístico comparativo entre sensores android y windows phone aplicado en la detección de movimientos telúricos. Escuela Superior Politécnica de Chimborazo.
- Hovden, T., Huie, M., Bich, T. & Acker, P. (2007). Proximity sensor device and method with activation confirmation. Synaptics Incorporated.
- Maghsoudi, J. (2017). A Behavioral Biometrics User Authentication Study Using Motion Data from Android Smartphones. Pace University.

- Micucci, D., Mobilio, M & Napoleano, P. (2017). A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones. Sensors.
- Munguia, E., Intille, S., Lopez, L. & Larson, K. (2006). The Design of a Portable Kit of Wireless Sensors for Naturalistic Data Collection. Massachusetts Institute of Technology.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research.
- Pires, I., García, N., Pombo, N & Flórez, F. (2011). From Data Acquisition to Data Fusion: A Comprehensive Review and a Roadmap for the Identification of Activities of Daily Living Using Mobile Devices. Sensors.
- Pogorelc, B., Bosnić, Z & Gams M. (2011). Automatic recognition of gait-related health problems in the elderly using machine learning. Springer.
- Shoaib, M., Bosch, S., Durmaz, O., Scholten, H. & Havinga, P. (2015). A Survey of Online Activity Recognition Using Mobile Phones. Sensors.
- Suto, J., Oniga, S., Lung, C. & Orha, I. (2017). Recognition Rate Difference between Real-time and Offline Human Activity Recognition. International Conference on Internet of Things for the Global Community.
- Vaughn, A., Biocco, P., Liu, Y. & Anwar, M. (2018). Activity Detection and Analysis Using Smartphone Sensors. IEEE Computer Society.
- Viana, J., Neto, G., Galindo, I., Oliveira, A., Braga, R. & Oliveira, C. (2017). A visualization and analysis approach of cyclist data obtained through sensors. Federal Institute of Education, Science and Technology of Ceará IFCE, Ceará, Brazil
- Wiley, J. (2014). Machine learning methods in chemoinformatics. WIREs Computational Molecular Scienc, 4:468–481.

8. CIBERGRAFÍA

- Collado, C. (2018). Los 9 sensores más importantes de tu móvil, y para qué sirve cada uno. Recuperado de: <https://andro4all.com/2018/12/sensores-movil>
- Collado, C. (2019). Qué es el giroscopio de un móvil, para qué sirve y cómo saber si el tuyo lo tiene. Recuperado de: <https://andro4all.com/2019/02/giroscopio-movil-que-es>
- Dalul, S. (2019). How to improve GPS accuracy on Android. Recuperado de : <https://www.androidpit.com/how-to-improve-your-gps-signal-on-your-android>
- Developers, (2019). Location Strategies. Recuperado de <https://developer.android.com/guide/topics/location/strategies#Updates>
- KIENYKE (2019). El 67% de la población cuenta con un dispositivo móvil. Recuperado de: <https://www.kienyke.com/tendencias/tecnologia/cuantas-personas-tienen-celular-en-el-mundo>
- Federación Internacional de Sociedades de la Cruz y de la Media Luna Roja (2007). Cruz Roja Colombiana. Recuperado de http://web.cruzrojacolombiana.org/publicaciones/pdf/planes_de_respuesta_y_contingencia_1722011_044520.pdf.
- González, J., (2018). Tipos de aprendizaje automático. Recuperado de <https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-413e3c615e2>
- Hackeando Tec (2015). Redes Neuronales - 1.3 Neuronas Biológicas Superveloces - Hackeando Tec. Recuperado de <https://youtu.be/J4rf00Pex1E>
- Hathibelagal, A. (2017). Android Sensors in Depth: Proximity and Gyroscope. Recuperado de : <https://code.tutsplus.com/tutorials/android-sensors-in-depth-proximity-and-gyroscope--cms-28084>
- Hernández, M. (2017). Android Sensor. Recuperado de <https://elrobotista.com/android-sensor/>
- Jiménez, J. (2014). APLICACIÓN MÓVIL PARA LA CAPTURA DESATENIDA DE DATOS DE SENSORES EN TELÉFONOS INTELIGENTES. (Trabajo de

grado, Universidad Carlos III Escuela politécnica superior). Recuperado de :
<https://core.ac.uk/download/pdf/44310781.pdf>

- Juárez, G. (2017). ¿Cómo funciona el aprendizaje automático (machine learning)?. Recuperado de <http://www.nexolution.com/como-funciona-el-aprendizaje-automatico-machine-learning/>
- Kios Buku Gema. (2011). Using Cellular Telephone Networks for GPS Anywhere. Recuperado de <https://kiosbukugema.blogspot.com/2011/03/assisted-gps.html>
- Marín, H. (S.f). Minería de Datos. Recuperado de <https://www.tamps.cinvestav.mx/~hmarin/Mineria/EC2.pdf>
- Mira Cómo hacerlo. (2019). ¿Qué es el Giroscopio del Móvil y como Funciona?. Recuperado de : <https://miracomohacerlo.com/giroscopio-del-movil-sirve/>
- Python Diario. (2018). Mi diario Python. Recuperado de <http://www.pythondiario.com>.
- Schultebracks, L (2015). Introduction to Support Vector Machines. Recuperado de <https://medium.com/@LSchultebracks/introduction-to-support-vector-machines-9f8161ae2fcb>
- Suárez, R. (2017). Python. Recuperado de <https://www.ecured.cu/Python>
- Tolosa, C. & Giz, Á. (S.f). Sistemas Biométricos. Recuperado de: http://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia%20Bioinformatica/web_BIO/Documentacion/Trabajos/Biometria/Trabajo%20Biometria.pdf
- Training Data (2019). Techopedia. Recuperado de <https://www.techopedia.com/definition/33181/training-data>