Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado
Facultad de Ciencias Matemáticas
Unidad de Posgrado

Sistema de ubicación en áreas terrestres basada en mapas a escala, utilizando geo localización de dispositivos móviles

TESIS

Para optar el Grado Académico de Magister en Computación e Informática

AUTOR

Enrique Gustavo CHALA MENA

ASESOR

Luzmila Elisa PRÓ CONCEPCIÓN

Lima, Perú

2026



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Chala, E. (2026). *Sistema de ubicación en áreas terrestres basada en mapas a escala, utilizando geo localización de dispositivos móviles*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Enrique Gustavo Chala Mena
Tipo de documento de identidad	D.N.I.
Número de documento de identidad	07001403
URL de ORCID	https://orcid.org/0009-0009-7686-3615
Datos de asesor	
Nombres y apellidos	Luzmila Elisa Pró Concepción
Tipo de documento de identidad	DNI
Número de documento de identidad	08862360
URL de ORCID	https//orcid.org/0000-0003-0622-1173
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Carlos Edmundo Navarro Depaz
Tipo de documento	DNI
Número de documento de identidad	08482690
Miembro del jurado 1	
Nombres y apellidos	Virginia Vera Pomalaza
Tipo de documento	DNI
Número de documento de identidad	09813434
Miembro del jurado 2	
Nombres y apellidos	Augusto Parcemón Cortez Vásquez
Tipo de documento	DNI
Número de documento de identidad	08634618
Miembro del jurado 3	
Nombres y apellidos	Rosa Menéndez Mueras

Tipo de documento	DNI
Número de documento de identidad	10246770
Datos de investigación	
Línea de investigación	C.0.3.5. Comunicaciones inalámbricas.
Grupo de investigación	No aplica.
Agencia de financiamiento	Sin financiamiento.
Ubicación geográfica de la investigación	Edificio: Ciudad Universitaria de la Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Lima Urb. Los Cipreses Calle: Avenida Carlos Germán Amezaga 375 Latitud: -12.055916 Longitud: -77.084541
Año o rango de años en que se realizó la investigación	2023 -2024
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones **https://purl.org/pe-repo/ocde/ford#2.02.04**

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA

FACULTAD DE CIENCIAS MATEMÁTICAS
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO
UNIDAD DE POSGRADO

ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO

Siendo las 11:00 horas del día miércoles dieciocho de marzo de dos mil veintiséis, en el Laboratorio 2 de la Facultad de Ciencias Matemáticas de la Universidad Nacional Mayor de San Marcos, el Jurado Examinador de tesis, presidido por el Dr. Carlos Edmundo Navarro Depaz e integrado por los siguientes miembros: Mg. Virginia Vera Pomalaza (Jurado Informante), Mg. Rosa Menendez Mueras (Jurado Informante), Mg. Augusto Parcemón Cortez Vásquez (Jurado Evaluador) y la Dra. Luzmila Elisa Pró Concepción como Miembro Asesora; se reunieron para la sustentación de la tesis titulada: **«SISTEMA DE UBICACIÓN EN ÁREAS TERRESTRES BASADA EN MAPAS A ESCALA, UTILIZANDO GEO LOCALIZACIÓN DE DISPOSITIVOS MÓVILES»**, presentado por el Br. Enrique Gustavo Chala Mena, para optar el Grado Académico de Magíster en Computación e Informática.

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, según tabla adjunta, habiendo obtenido el Bachiller **ENRIQUE GUSTAVO CHALA MENA**, el calificativo de ...DIECISIETE (17) MUY BUENO

Habiendo sido aprobada la sustentación de la Tesis, el Jurado Examinador recomienda que el Consejo de Facultad apruebe el otorgamiento del **Grado Académico de Magister en Computación e Informática** al Bachiller **ENRIQUE GUSTAVO CHALA MENA**.

Siendo las 12:20 horas, se levantó la sesión, firmando para constancia la presente Acta:

Dr. Carlos Edmundo Navarro Depaz.
Presidente

Mg. Virginia Vera Pomalaza.
Miembro

Mg. Rosa Menendez Mueras.
Miembro

Mg. Augusto Parcemón Cortez Vásquez.
Miembro

Dra. Luzmila Elisa Pró Concepción.
Miembro Asesora

UNIVERSIDAD NACIONAL MAYOR DE
SAN MARCOS
Universidad del Perú, Decana de América

CERTIFICADO DE SIMILITUD

Yo, **Luzmila Elisa Pró Concepción** en mi condición de asesora acreditado con la Resolución Decanal N° 0663/FCM-D/2017 del 14 de Setiembre del 2017 y Resolución Decanal N° 001315-2025-D-FCM/UNMSM, de fecha 22 de agosto 2025, de la tesis, cuyo título es: «**SISTEMA DE UBICACIÓN EN ÁREAS TERRESTRES BASADA EN MAPAS A ESCALA, UTILIZANDO GEO LOCALIZACIÓN DE DISPOSITIVOS MÓVILES**», presentado por el bachiller **Enrique Gustavo Chala Mena**, para optar el Grado Académico de Magíster en Computación e Informática, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el **5 %** de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional.**

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para optar del Grado Académico de Magíster en Computación e Informática.

Lima, 20 de mayo de 2025



Firma del Asesor

Nombres y Apellidos del asesor: **Luzmila Elisa Pró Concepción**
DNI: **08862360**

AGRADECIMIENTO

Agradezco a la Universidad que me ha proporcionado el material que he requerido para mi investigación, especialmente al área de infraestructura que aportó el material gráfico del plano del Campus universitario de la U.N.M.S.M. Así mismo, extiendo mi agradecimiento a la Dra. Luzmila Elisa Pro Concepción, mi asesora, me brindó parte de su valioso tiempo en apoyarme con su asesoría y consejos durante toda la investigación.

INDICE

CARÁTULA 1 i
PAGINA EN BLANCO ii
CARÁTULA 2 iii
HOJA DE ACEPTACIÓN iv
AGRADECIMIENTO v
INDICE vi
ÍNDICE DE FIGURAS xi
ÍNDICE DE CUADROS xiii
RESUMEN xiv
ABSTRACT xv
CAPITULO I. INTRODUCCIÓN 1
1.1. Situación problemática 1
1.2. Formulación del problema 2
1.2.1. Problema general 2
1.2.2. Problemas específicos 2
1.3. Justificación de la investigación 3
1.4. Solución propuesta 3
1.5. Alcance 4
1.6. Limitaciones 4
1.6.1- Del dispositivo 4
1.6.2. Del plano cartográfico 5
1.7. Objetivo de la investigación 5
1.7.1. Objetivo general 5
1.7.2. Objetivos específicos 6
CAPITULO II. MARCO TEÓRICO 7
2.1. Marco Filosófico o epistemológico de la investigación 7
2.2. Antecedentes de investigación 8

2.2.1. Posicionamiento en interiores basado en dispositivos móviles 8
2.2.2. Análisis de los mecanismos de posicionamiento en interiores basados en huellas: caso de estudio sobre Zigbee 8
2.2.3. Aplicación móvil Android con sistema de posicionamiento en interiores para mapas y cercanías con tecnologías BLE 9
2.2.4. Aplicativo móvil para localizar y guiar personas a través del Campus Universitario de la Pontificia Universidad Javeriana 9
2.2.5. Diseño e implementación de un prototipo de un sistema de localización en espacios cerrados (Indoor) 9
2.3. Bases teóricas 10
2.3.1. Elementos del sistema de posicionamiento y navegación 10
2.3.2. Infraestructura de posicionamiento y navegación 11
2.3.3. Algoritmos de posicionamiento 14
2.3.4. Aseguramiento de la calidad e interferencias de la señal 16
2.3.5. Marco de trabajo del sistema operativo Android 17
CAPITULO III. METODOLOGÍA 21
3.1 Métodos y procedimientos a ser utilizados 21
3.3. Unidad de análisis. 21
3.4. Población de estudio 22
3.5. Tamaño de la muestra 25
3.6. Selección de la muestra 26
3.7. Materiales o instrumentos que se emplean 27
3.7.1. Dispositivos 27
3.7.2. Herramientas de desarrollo 28
CAPITULO IV. INGENIERÍA DE DESARROLLO DEL SOFTWARE 30
4.1. Formulación de los requerimientos 30
4.1.1. Historia de usuario 30
4.1.2. Requerimientos funcionales 30
4.1.2. Requerimientos no funcionales 31
4.2. Diseño de la interface visual 34
4.2.1. Modelo conceptual de la interface visual 35
4.2.2. Interface visual «Ubícame» 35

4.2.3. Interface visual «Categorías» 42
4.2.4. Interface visual «Locales» 43
4.2.5. Interface visual «Detalle local» 44
4.3. Diseño de procesos del sistema 45
4.3.1. Diagrama de contexto 45
4.3.2. Definición y diseño de procesos 45
4.4. Diseño de arquitectura del software 51
4.4.1. Arquitectura del sistema 51
4.4.2. Estilo del diseño de la arquitectura 51
4.5. Diseño de implementación 53
4.5.1. Diagrama de paquetes 53
4.5.2. Clase Mainmapa 53
4.5.3. Clase Ubícame 54
4.5.4. Clase CategoríaLocal 56
4.5.5. Clase Locales 56
4.5.6. Clase DetalleLocal 57
4.5.7. Clase CoreDB 59
4.5.8. Clase Conecta 59
4.5.9. Clase ImagenCore 60
4.5.10. Clase MapCore 60
4.5.11. Clase AdaptadorListaCategoria 61
4.5.12. Clase AdaptadorListaLocal 61
4.5.13. Clase AdaptadorListaItemSearch 62
4.5.14. Clase EntConfiguracion 62
4.5.15. Clase EntCategoria 63
4.5.16. Clase EntCoordenada 64
4.5.17. Clase EntLocal 64
4.5.18. Clase EntProveedor 65
4.6. Diseño de estructura de Base de datos 66
4.6.1. Categoría 66
4.6.2. Configuración 66
4.6.3. Local 67

4.7. Diseño de variables de control del sistema 68
4.8. Diseño de algoritmos 70
4.8.1. Control de área visible del plano digital en el dispositivo 70
4.8.2. Algoritmo de conversión Coordenadas geográficas a gráficas X,Y 72
4.8.3. Algoritmo para el cálculo del Rumbo 77
4.8.4. Algoritmo de intersección entre 2 puntos 78
4.9. Desarrollo del sistema 79
4.9.1 Definición de datos del sistema en repositorio persistentes 79
4.9.2. Pruebas de rango de precisión de los sensores 84
4.9.3. Formulación de los niveles de precisión 87
4.9.4. Conversión del formato gráfico del plano cartográfico 88
4.9.5. Conversión del sistema de coordenadas del plano 89
4.9.6. Desarrollo del codigo fuente del sistema 89
4.10. Pruebas unitarias y pruebas integrales 89
4.10.1 Módulo CoreDB 89
4.10.2 Módulo Conecta 90
4.10.3 Módulo ImagenCore 90
4.10.4 Módulo MapaCore 90
4.10.5 Módulo Ubicame 91
4.10.6. Módulo CategoriaLocal 91
4.10.7. Módulo Locales 92
4.10.8. Módulo DetalleLocal 92
4.10.11. Pruebas integrales 92
4.11. Pruebas de precisión del sistema móvil 93
4.11.1. Precisión en el reconocimiento del área de interés 93
4.11.2. Precisión de la trayectoria seleccionada 95
CAPITULO V. RESULTADOS Y DISCUSIÓN 103
CAPITULO VI. IMPACTOS 107
CONCLUSIONES 108
RECOMENDACIONES 109
REFERENCIAS BIBLIOGRÁFICAS 110
ANEXOS 113

1. Conversión plano cartográfico formato DWG a formato PNG 113
1.1. Antecedentes 113
1.2. Conversión del plano 114
2. Conversión del plano a coordenadas geográficas grados decimales 116
2.1. Transferir a Google Earth 116
2.2. Establecer límites 116
3. Código fuente 119
3.1. Manifiesto 119
3.2. Package: com.echm.geolocation 121
3.3. Package: echm.db 160
3.4. Package: echm.generales 167
3.5. Package: echm.entidades 172
3.6. Package: echm.entidades 176
3.7. Layout 180
3.8. Definiciones String 187
3.9. Hoja de estilos CSS 188

ÍNDICE DE FIGURAS

Figura 1. Modelo conceptual de interface visual. Fuente propia 35
Figura 2. Entrada principal del sistema. Fuente propia. 37
Figura 3. Buscador y sugerencia. Fuente propia. 39
Figura 4. Posicionamiento del Local. Fuente propia. 40
Figura 5. Posicionamiento del dispositivo y rumbo a destino. Fuente propia 41
Figura 6. Lista de Categorías de Local. Fuente propia. 42
Figura 7. Lista de Locales por Categoría indicada. Fuente propia. 43
Figura 8. Detalle del Local y posición. Fuente propia. 44
Figura 9. Diagrama de contexto. Fuente propia 45
Figura 10. Diagrama de actividad de la clase Ubícame. Fuente propia. 47
Figura 11. Diagrama de actividad de Categoría. Fuente propia. 48
Figura 12. Diagrama de actividad de Locales. Fuente propia. 49
Figura 13. Diagrama de actividad de DetalleLocal. Fuente propia. 50
Figura 14. Arquitectura del sistema. Fuente propia 51
Figura 15. Patrón de diseño de la arquitectura. Fuente propia 52
Figura 16. Diagrama general de clases. Fuente propia 53
Figura 17. Diagrama de Clases de la clase Ubícame. Fuente propia. 55
Figura 18. Diagrama de Clase de Categoría. Fuente propia. 56
Figura 19. Diagrama de Clase de Locales. Fuente propia. 57
Figura 20. Diagrama de Clase de DetalleLocal. Fuente propia. 58
Figura 21. Área del plano con puntos límites y puntos de locales. Fuente propia. 70
Figura 22. Plano digital del sistema «Geolocation». Fuente propia. 72
Figura 23. Diagrama de proceso de conversión. Fuente propia 73
Figura 24. Diagrama de obtención de coordenadas geográficas a X,Y . Fuente propia 75
Figura 25. Diagrama de actividad del Algoritmo de posición de coordenada. Fuente propia 77

Figura 26. Diagrama de precisión en metros. Fuente propia 87
Figura 27. Puntos aleatorios del área de interés sobre Google Earth. Fuente propia .. 94
Figura 28. Puntos aleatorios del área de interés sobre plano sistema. Fuente propia. 95
Figura 29. Validación de trayectoria sobre Google Earth. Ruta 1. Fuente propia..... 97
Figura 30. Validación de trayectoria sobre plano del sistema. Ruta 1. Fuente propia.. 97
Figura 31. Validación de trayectoria sobre Google Earth. Ruta 2. Fuente propia..... 99
Figura 32. Validación de trayectoria sobre plano del sistema. Ruta 2. Fuente propia.. 99
Figura 33. Validación de trayectoria sobre Google Earth. Ruta 3. Fuente propia... 101
Figura 34. Validación de trayectoria sobre plano del sistema. Ruta 3. Fuente propia.. 102
Figura 35. Plano cartográfico Campus universitario. Fuente. UNMSM (2023) 113
Figura 36. Plano cartográfico convertido Campus universitario. Fuente propia 115
Figura 37. Plano del Campus universitario. Fuente U.N.M.S.M. (2023). 116
Figura 38. Plano del Campus universitario en Google Earth. Fuente propia. 117
Figura 39. Área del Campus universitario en Google Earth. Fuente propia..... 118

ÍNDICE DE CUADROS

Tabla 1. Área del plano Cartográfico del Campus universitario. Fuente propia. 22
Tabla 2. Definición de categorías. Fuente propia. 23
Tabla 3. Definición de los puntos de interés. Fuente propia. 24
Tabla 4. Definición de los niveles de precisión en metros. Fuente propia. 27
Tabla 5. Definición de los atributos de la clase EntConfiguracion. Fuente propia. ... 62
Tabla 6. Definición de los atributos de la clase EntCategoria. Fuente propia. 63
Tabla 7. Definición de los atributos de la clase EntCoordenada. Fuente propia. 64
Tabla 8. Definición de los atributos de la clase EntLocal. Fuente propia. 64
Tabla 9. Definición de los atributos de la clase EntProveedor. Fuente propia. 65
Tabla 10. Definición de las categorías de los locales. Fuente propia. 66
Tabla 11. Definición de los parámetros del sistema. Fuente propia. 66
Tabla 12. Definición de los locales. Fuente propia. 67
Tabla 13. Tabla de secuencias para obtener las direcciones. Fuente propia 74
Tabla 14. Lista de categorías. Fuente propia 80
Tabla 15. Lista de Locales. Fuente propia 80
Tabla 16. Lista de la configuración. Fuente propia 83
Tabla 17. Tabla de resultado pruebas de sensores. Escenario 1. Fuente propia 85
Tabla 18. Tabla de resultado pruebas de sensores. Escenario 2. Fuente propia 86
Tabla 19. Tabla de resultado pruebas de sensores. Escenario 3. Fuente propia 86
Tabla 20. Tabla de puntos de interés con selección aleatoria. Fuente propia 93
Tabla 21. Tabla de ruta 1. Fuente propia 96
Tabla 22. Tabla de ruta 2. Fuente propia 98
Tabla 23. Tabla de ruta 3. Fuente propia 100

RESUMEN

Las empresas e instituciones que cuentan con áreas físicas amplias o complejas construcciones, requieren ofrecer a sus usuarios una herramienta que les permita obtener su posición dentro del recinto o área terrestre. El objetivo es el desarrollo de un sistema móvil bajo la plataforma ANDROID que permite demostrar el proceso de localización en coordenadas geográficas en grados decimales del dispositivo sobre un mapa cartográfico a escala real, proporcionar la localización y seguimiento a puntos de interés, consultar en una base de información los puntos de interés y su localización sobre el mapa cartográfico a escala real. El dispositivo debe contar con la capacidad de detección de puntos de referencia por proximidad para el proceso de localización y sensores que permiten asegurar la posición y rumbo del dispositivo. Se aplican algoritmos de posicionamiento y navegación del dispositivo sobre el mapa y procedimientos de acceso a las señales de radio frecuencia con bajo consumo de energía.

Palabras claves: Android, Proximidad, Posición, Coordenada geográfica, sensor, algoritmo, señal de radio frecuencia.

ABSTRACT

Companies and institutions that have large physical areas or complex buildings need to offer their users a tool that allows them to obtain their position within the premises or land area. The objective is the development of a mobile system under the ANDROID platform that allows demonstrating the location process in geographic coordinates in decimal degrees of the device on a real-scale cartographic map, providing the location and tracking of points of interest, consulting in a database of information about points of interest and their location on the real-scale cartographic map. The device must have the ability to detect proximity reference points for the localization process and sensors that ensure the position and heading of the device. Device positioning and navigation algorithms are applied to the map and access procedures to radio frequency signals with low energy consumption.

Key words: Android, Proximity, Position, Geographic coordinate, sensor, algorithm, radio frequency signal.

CAPITULO I. INTRODUCCIÓN

1.1. Situación problemática

Las empresas e instituciones que cuentan con áreas físicas amplias o complejas construcciones, requieren ofrecer a sus usuarios una herramienta que les permita obtener su posición dentro del recinto o área terrestre. Actualmente se requiere en distintas actividades como centros comerciales, entidades educativas, empresas mineras u otras que desean brindar información a sus usuarios de la localización de puntos de interés. GOOGLE MAPS no tiene acceso a estos lugares privados, solo muestra mapas sobre áreas geográficas públicas.

Actualmente existen soluciones que permiten la localización en recintos con un costo de despliegue involucrado. Se basan en receptores localizados en el interior del recinto. El proceso de localización se basa en la triangulación mediante antenas internas instaladas en el recinto, pero con un alcance Por ejemplo los BEACON que utilizan señales RF Bluetooth Low Energy (BLE), conforme aumenta la el área física, aumenta el costo de despliegue de la solución. Implementar una solución de este tipo conlleva costos adicionales de mantenimiento de los dispositivos receptores. Estos llevan una fuente de energía que es una limitante para períodos con alta carga.

A las soluciones con BEACON, existen soluciones mediante posicionamiento en tiempo real por Identificador de Radio Frecuencia RFID, con tarjetas activas que son los emisores de señal para enviar su posición. Esta solución requiere antenas receptoras para efectuar la localización. Como en el caso de los BEACON, el costo de

despliegue se incrementa para áreas física mayores y se requiere que cada persona cuente con una tarjeta activa para la localización, inviable en lugares con mucha afluencia de personas.

1.2. Formulación del problema

1.2.1. Problema general

Las soluciones para resolver geo localización de interiores se basan en implementar sensores de radio frecuencia instalados en el recinto que con dispositivos móviles o tarjetas son usadas para geo localizarse. ¿ se puede desarrollar un sistema para dispositivos de celulares en plataforma ANDROID como una solución de geo localización en interiores de grandes áreas que evite implementar infraestructura adicional?

1.2.2. Problemas específicos

1. ¿El sistema operativo ANDROID cuenta con la capacidad para implementar una solución de geo localización en interiores?

2. ¿Es posible obtener la geo localización del dispositivo móvil próxima a tiempo real considerando la actual infraestructura de comunicaciones móviles?

3. ¿Cómo se puede relacionar una imagen en mapa de bits a una posición por latitud y longitud?

4. ¿Cómo afecta la perdida eventual de señal para la obtención de la geo localización. Invalida el sistema propuesto?

5. ¿Cuál es el proceso óptimo de precisión las coordenadas de geo localización que maximice el ahorro de energía del dispositivo, considerando que un menor margen de error conlleva a un mayor consumo de energía?

1.3. Justificación de la investigación

El presente estudio se justifica porqué va a proporcionar una solución que no requiere costos de inversión en infraestructura de sensores para la geo localización en interiores, solo se requiere contar con un dispositivo celular y un sistema de geo localización en interiores. El sistema móvil geo localiza la posición del dispositivo sobre un plano digital del área física de forma continua.

Las necesidades de entidades públicas y privadas de aplicar localización en áreas de interés. Ofrecer a los usuarios el servicio de localización con la finalidad obtener la información de los puntos de interés e información complementaria en un área construida (Empresas mineras, Universidades, áreas hospitalarias, Aeropuertos, etc.). Por ejemplo, Los centros comerciales requieren ofrecer a sus clientes información de tiendas por departamentos y centros de venta por proximidad. Estos ven la necesidad de promover con publicidad u otra información de los productos que se ofrecen.

Transmitir el conocimiento a los estudiantes y profesionales de Informática, los procedimientos de desarrollo de sistemas para implementar soluciones de geo localización en áreas físicas o recintos privados, utilizando las librerías y componentes estándar bajo la plataforma operativa ANDROID 10.0 o superior.

1.4. Solución propuesta

Un sistema para dispositivos móviles bajo la plataforma operativa ANDROID que localiza la posición por geo localización sobre un plano de área física de un recinto .

1.5. Alcance

Un sistema móvil de localización y posicionamiento sobre el plano terrestre de distribución de la Ciudad Universitaria de la Universidad Nacional Mayor de San Marcos, que corresponde con una gran área construida. La solución se construye en plataforma operativa ANDROID 10 o superior en lenguaje de programación JAVA y utiliza la imagen digitalizada del plano del Campus universitario en mapa de bits y en coordenadas geográficas UTM.

1.6. Limitaciones

1.6.1- Del dispositivo

Dispositivo móvil celular con la capacidad de determinar dirección o rumbo, velocidad, orientación respecto al Norte magnético y posición en coordenadas de latitud y longitud en coordenadas geográficas a grados decimales. Considerar que las señales de radio frecuencia de salida y entrada pueden verse bloqueadas por columnas de concreto y acero.

1.6.1.1. Sensores de conectividad. Los dispositivos móviles deben contar con sistema de posicionamiento global o posicionamiento global asistido (A-GPS), ademas, deben contar con señal Wi-Fi. Estos dispositivos deben contar con:

1. Acceso a la RED del operador de telefonía.
2. Conexión inalámbrica por radio frecuencia para enlace con dispositivo Enrutador Wi-Fi y/o Punto de acceso Wi-Fi.
3. Sensor de conexión inalámbrica a antenas de radio frecuencia GSM.
4. Sensor de conexión por dispositivo GPS para conexión satelital GNSS.

1.6.1.2. Sensores inerciales. Los dispositivos deben contar con sensores de inercia. Estos comprenden acelerómetro y giroscopio. Un acelerómetro mide la fuerza específica y un giroscopio mide la velocidad angular o velocidad de rotación.

1.6.1.3. Plataforma de desarrollo. Aplicación móvil sobre la plataforma operativa ANDROID 10.0 o superior. El lenguaje de desarrollo JAVA y las librerías que proporcione el sistema operativo de ser compatible a nivel de interface de programación JAVA.

Solo se considera en el desarrollo los componentes programables proporcionados por la plataforma SDK Android 10.0, no se utiliza librerías o componentes externos ni hardware fuera del dispositivo, tal como accesorios de radio frecuencia que se acoplen al dispositivo móvil.

1.6.2. Del plano cartográfico

Imagen vectorial del plano cartográfico a escala real en coordenadas terrestres UTM del Campus de la Ciudad Universitaria. La imagen final es un archivo PNG de alta resolución para poder re-dimensionar la imagen acorde con la dimensión de la pantalla del dispositivo móvil.

1.7. Objetivo de la investigación

1.7.1. Objetivo general

Desarrollar un sistema de ubicación en áreas terrestres basada en mapas a escala, utilizando geo localización de dispositivos móviles.

1.7.2. Objetivos específicos

1. Obtener la posición con coordenadas geográficas en grados decimales sobre un plano cartográfico. Establecer los procedimientos de asignación de las coordenadas de geo localización en un mapa de bits o vectorial.

2. Localizar posiciones de sitios de interés en el Campus universitario.

3. Consultar la información de sitios de interés por el usuario cuya información de localización se almacene en repositorio local del dispositivo.

4. Realizar la navegación posicionando el dispositivo y el rumbo hacia un local de destino ajustando la posición del dispositivo conforme se realice el desplazamiento.

5. Mostrar de forma visual el posicionamiento y desplazamiento en el plano digital.

CAPITULO II. MARCO TEÓRICO

2.1. Marco Filosófico o epistemológico de la investigación

Enfoque desde un pensamiento positivista, se considera que el conocimiento humano se fundamenta en el conocimiento científico, con hechos comprobables y contrastable por observación y experimentación. Con la premisa establecida considera el tipo de investigación APLICADA y modo EXPERIMENTAL, se recolecta la información de las coordenadas geográficas del plano cartográfico, se realizan muestras de la posición obtenida por el sistema móvil considerando como aceptados los algoritmos matemáticos de posicionamiento respecto a puntos de referencia. Estos puntos de referencia se definen como puntos del polígono del plano cartográfico. El sistema obtiene la posición del dispositivo respecto a estos puntos de referencia. Los valores obtenidos y comparados con los puntos predefinidos según la muestra tomada previamente sirven para validar el comportamiento del sistema. Pensamiento crítico del procedimiento de investigación.

Respecto a los algoritmos , estos están sustentado por bibliografía documental de investigaciones científicas previas, comprobadas y comparada con la bibliografía adicional. Argumento de selección.

Existen distorsiones en obtener los puntos de posicionamiento, especialmente en áreas cerradas y/o en áreas de construcción que interfieren con la señal de Radio frecuencia, en este caso se toma en cuenta que los algoritmos tienen ciertas requerimientos restrictivos en su aplicación. Considerar que la tecnología es

cambiante, sobre todo en la tecnología de comunicaciones inalámbricas. Argumento de restricción.

2.2. Antecedentes de investigación

Tecnologías que usan la Radio frecuencia (RF) como medio de localización: WLAN/Wi-Fi, Zigbee, Ultra-Wideband (UWB), Bluetooth LE, GPS, A-GPS , RED de Celular y RFID. Consideremos que los dispositivos celulares pueden utilizar mas de un medio de radio frecuencia para poder obtener la posición, como es el caso de la combinación de tecnología UWB, A-GPS, GPS y WLAN. Si bien no se mencionan en la lista, pero los dispositivos móviles usan los sensores magnéticos para obtener la orientación y la altitud.

2.2.1. Posicionamiento en interiores basado en dispositivos móviles

Desarrollo de sistema móvil sobre plataforma Android, se aplicó en el Metro de Madrid usando señal de RF GSM. El objetivo es el desarrollo de una aplicación móvil de bajo costo para posicionamiento en interiores.

Autor. Miguel Ruiz Cuesta (2012). Universidad Autónoma de Madrid, Madrid, España.

2.2.2. Análisis de los mecanismos de posicionamiento en interiores basados en huellas: caso de estudio sobre Zigbee

Análisis de posicionamiento de interiores con tecnología de Radio frecuencia (RF) Wi-Fi, Bluetooth y ZigBee. Aplicación de algoritmos y técnicas de detección basadas en la intensidad de la señal de RF. Utiliza un simulador basado en huellas dactilares (Fingerprinting) que registra los dispositivos de inalámbricos para detectar la

posición en base a la localización del dispositivo inalámbrico. Evalua la aplicación del protocolo 802.15.4 ZigBee para posicionamiento en interiores.

Autor. José Antonio Gómez Martín (2014). Universidad de Sevilla, Sevilla, España.

2.2.3. Aplicación móvil Android con sistema de posicionamiento en interiores para mapas y cercanías con tecnologías BLE

Desarrollo de un sistema en plataforma Android para posicionamiento en interiores con balizas (BEACON) de RF Bluetooth Low Energy (BLE), una aplicación Móvil que usa como recinto de interior la Universidad que incluye una consola centralizada de administración de información.

Autor. Juan Pablo Arévalo Alvarez (2018). Universidad de los Andes, Bogotá, Colombia.

2.2.4. Aplicativo móvil para localizar y guiar personas a través del Campus Universitario de la Pontificia Universidad Javeriana

Desarrollo de un sistema móvil en plataforma Android para localización de interiores que posiciona con Radio frecuencia Wi-Fi. Se evalua el método de huellas dactilares (Fingerprinting) para almacenar los parámetros del dispositivo inalámbricos.

Autores. David Eduardo Cruz Gómez, Javier Camilo Mendoza Ortiz (2016). Pontifica Universidad Javeriana, Bogotá, Colombia.

2.2.5. Diseño e implementación de un prototipo de un sistema de localización en espacios cerrados (Indoor)

Sistema móvil en plataforma Android para localización en interiores que usa balizas (BEACON) de Radio frecuencia para obtener el posicionamiento. Es un sistema con costo de despliegue, requiere la infraestructura de balizas para lograr obtener la posición.

Autores. Martin Pereira Tapiro, William Alejandro Polo Poveda (2015). Universidad Autónoma de occidente, Valle del Cauca, Colombia.

2.3. Bases teóricas

2.3.1. Elementos del sistema de posicionamiento y navegación

Los sistemas de posicionamiento y navegación constan de un conjunto de elementos necesario para determinar visualmente sobre un plano o mapa la posición en tiempo real con la mayor precisión:

1. Sensores que obtienen la posición, el rumbo y la altitud del dispositivo. La posición se obtiene en base a un punto de referencia y depende del tipo de sensor. El servicio de satélites GNSS se utiliza la posición e información complementaria llamada efemérides, adicionalmente se mejora la precisión combina de los satélites GNSS o en combinación con antenas de Red móvil (estación base) para mejorar la precisión mediante el proceso de corrección de error (Diferencial GNSS), Bhatta, B. (2021). El servicio de Red móvil asistido GPS (A-GPS), acelera el proceso de obtención de posición ya que realiza en cálculo en el operador de la Red móvil, García, A. et. al (2020) en asistido GNSS, Groves, P. (2013). El servicio de comunicación con los enrutadores y puntos de acceso Wi-Fi, García, A. et. al (2020) en posicionamiento terrestre en redes no celular. Los sensores inerciales como el acelerómetro, magnetómetro y barómetro, se utilizan para determinar la velocidad, la dirección y la altura de un dispositivo, Samama, N. (2019).

2. Dependiendo de los sensores utilizados, se aplican algoritmos de posicionamiento que consta de la obtención de la distancia del dispositivo usando la intensidad de la señal o el tiempo de respuesta y los cálculos adicionales para obtener la posición, por ejemplo los métodos de Trilateración, Multilateración, Fingerprinting o Triangulación, García, A. et. al (2020) en fundamentos de posicionamiento, Samama, N. (2019) en técnicas de posicionamiento.

3. Un sistema de visualización que ilustra la ubicación del objetivo en un plano de áreas interiores que emplea coordenadas locales o relativas, Pahlavan, K. (2019).

4. Componente de software del sistema que calcula las coordenadas locales en el plano y gestiona la navegación del individuo con el dispositivo. Se denomina «motor de inteligencia para interpretar los datos obtenido por los sensores, Pahlavan, K. (2019)».

2.3.2. Infraestructura de posicionamiento y navegación

2.3.2.1. Infraestructura del medio. La infraestructura de comunicaciones por Radio frecuencia solo involucra a los medios que pueden ser usados por un dispositivo móvil celular con sensores incorporados para detectar e interpretar las señales de radio frecuencia.

1. ***Posicionamiento por Satélites GNSS*** . GNSS (global navigation satellite systems) es una red de satélites controlados por estaciones terrestres. Los dispositivos receptores captan las señales de radio para geolocalizar con precisión al receptor. Actualmente existente 4 constelaciones o grupo de satélites. GPS propiedad de EEUU, GLONASS propiedad de Rusia, BEIDOU propiedad de China y GALILEO

propiedad de la Unión Europea. Bhatta, B. (2021), Groves, P. (2013) en GNSS: fundamentos, señales y satélites.

2. ***Posicionamiento por Redes de celular - Asistido GNSS*** . Utiliza la red móvil para acelerar el proceso de obtención de la posición del receptor, el cálculo se realiza en la Red móvil. Adicionalmente se aprovecha las señales de comunicación con las antenas de Red móvil para localizar un dispositivo móvil , Morton, Y. et. al. (2021). Con el despliegue de más sistemas satelitales, A-GNSS se amplió para incluir otras constelaciones GNSS como GLONASS, BEIDOU, Galileo, QZSS (Sistema por Satélite Cuasi cenital) e IRNSS (Indian Regional Navigation Satellite System), García, A. et. al (2020) en GNSS asistido Gentile, C. et. al (2013) en Localización celular.

3. ***Posicionamiento por Redes no celulares*** . Las tecnologías de posicionamiento basadas en redes no celulares, usan enrutadores Wi-Fi, puntos de acceso Wi-Fi y Bluetooth LE (BLE) para localizar y posicionar el dispositivo móvil. Una base de datos centralizada llamada Serving Mobile Location Center - (E-SMLC) tiene la información de estos puntos de acceso Wi-Fi la cual es consultada para obtener la ubicación. La base de datos mantiene una relación de las direcciones MAC de Wi-Fi y BLE y su ubicación física respectiva, y usan esta información para estimar la dirección actual del dispositivo. García, A. et. al (2020) en posicionamiento terrestre en redes no celular, Zekavat, R. et. al. (2019) en técnicas de localización por huella dactilar.

4. ***Posicionamiento por WSN (Wireless sensor networks)*** . «Las WSN se pueden describir como sistemas autónomos y auto organizados que consiste en sensores de bajo costo y que funcionan con baterías, que generalmente se implementa aleatoriamente (Jondhale, S. et. al. 2022)». Es una RED inalámbrica que trabaja con la capa física definida en el estándar IEEE 802.15.4 (Zigbee networks). Existe una variante con el estándar IEEE 802.15.4a que trabaja con la capa física Ultra-

Wide Band (UWB). Zekavat, R. et. al. (2019) en Localización inalámbrica usando ultra banda ancha, Wagh, S. et. al. (2023) en WSN e IOT.

2.3.2.2. Infraestructura dispositivos receptores. Involucra a los sensores de recepción y envío de ondas de radio frecuencia y sensores inerciales.

1. ***Sensores de radio frecuencia (RF)***

   a. ***Dispositivo wireless para conexión a dispositivos inalámbricos como enrutadores Wi-Fi y puntos de acceso Wi-Fi.*** Las WLAN y las redes de área personal inalámbricas (WPAN). Estas redes, diseñadas para fines de telecomunicaciones locales, generalmente se instalan en ambientes interiores. García, A. et. al (2020).

   b. ***Dispositivo de radio frecuencia para conexión a la Red móvil.*** Un sistema de comunicación celular proporciona servicios de comunicación inalámbrica para el proceso de envío de voz y datos. El dispositivo establece una comunicación con la estación base que proporciona información a la red móvil. Al tener una conexión permanente con la Red móvil se puede localizar el dispositivo con el procedimiento de Trilateración entre las antenas y estación base como puntos de referencia para determinar las coordenadas del dispositivo. García, A. et. al (2020) en redes no celular Wi-Fi y Bluetooth, Morton, Y et. al. (2021) en Navegación con Señales Celulares de Oportunidad, Gentile, C. et. al. (2013) en localización celular.

   c. **Dispositivo de posicionamiento global para satélites GNSS**. Los receptores GNSS se usan ampliamente para obtener información de la posición del dispositivo con acceso al servicio Location Based Services - LBS. Groves, P. (2013) en GNSS: Procesamiento y errores del equipo de usuario.

2. ***Sensores inerciales***

Sistema de Referencia de movimiento, altitud y dirección. Los sistemas inerciales incluyen todas las técnicas que aprovechan las propiedades inerciales de cualquier movimiento. Groves, P. (2013) en Sensores inerciales.

a. ***Giroscopio***. Mide la velocidad de rotación en cada uno de sus ejes. Se usa para medir los cambios de dirección del dispositivo móvil. Pahlavan, K. (2019) en Fusión de sensores para localización híbrida, Sharp, I. , Yu, K. (2019) en Uso de sensores para determinar la posición.

b. ***Acelerómetro***. Mide la aceleración o cambio en la velocidad del desplazamiento de un cuerpo. Se usa para medir movimientos del dispositivo móvil. Pahlavan, K. (2019) en Fusión de sensores para localización híbrida, Sharp, I. , Yu, K. (2019) en Uso de sensores para determinar la posición.

c. ***Magnetómetro***. Mide la fuerza magnética en tres direcciones ortogonales. Obtiene una dirección absoluta. Pahlavan, K. (2019) en Fusión de sensores para localización híbrida, Sharp, I. , Yu, K. (2019) en Uso de sensores para determinar la posición.

d. ***Barómetro***. Mide la altitud basada en la presión atmosférica. Obtiene la posición en altitud del dispositivo móvil. [Pahlavan, K. (2019) en Fusión de sensores para localización híbrida.

2.3.3. Algoritmos de posicionamiento

El algoritmo de posicionamiento procesa la información recolectada por los sensores y obtiene la posición del dispositivo, a mayor incidencias de errores al obtenerla los algoritmos se vuelven mas complejos con resultado menos preciso. Las mediciones no son exactas, las diferencias en la precisión ocurren en todo proceso de cálculo basado en radio frecuencias. Bhatta, B. (2021) en Métodos de

posicionamiento, Samama, N. (2019) en ¿Cuál es exactamente el problema de posicionamiento en interiores?.

2.3.3.1. Intensidad de la señal recibida (RSS) . Medir la distancia entre un punto de referencia y el dispositivo de destino mediante el proceso de la intensidad de la señal entre ambos puntos. La localización basada en RSS se usa para la localización Wi-Fi de los dispositivos aplicada a interiores. García, A. , et. al. (2020) en Fundamentos de posicionamiento, Pahlavan, K. (2019) en Fundamentos del rango RSS, Yu, K. (2018) en Robustez en el sistema de posicionamiento interior basado en huellas dactilares.

2.3.3.2. Posicionamiento TOA (Time of Arrival) . Obtiene la posición midiendo el tiempo que tarda la señal para llegar desde el punto de origen al punto de destino, considere que debe existir un reloj interno que este sincronizado entre los puntos. Se usa especialmente en las redes móviles de los operadores de telefonía celular. García, A. , et. al. (2020) en Fundamentos de posicionamiento, Pahlavan, K. (2019) en Fundamentos del posicionamiento TOA.

2.3.3.3. Posicionamiento sobre un plano 2D . En base de los algoritmos de la obtención de la posición dado por el punto de referencia (A-GPS) o el dispositivo (GPS), el objetivo es estimar la distancia entre ambos y dependiendo del algoritmo, el ángulo de llegada de la señal que el dispositivo puede obtener mediante el algoritmo DOA (Direction of Arrival). Pahlavan, K. (2019) en Introducción a los algoritmos de posicionamiento.

1. ***Métodos básicos de triangulación para posicionamiento***. Se mide la distancia R que es el radio de la circunferencia y el punto de referencia como centro de la circunferencia y el ángulo α que se forma entre el punto de referencia, el objetivo y el norte geográfico (Azimut o TRUE HEADING) para posicionar la ubicación en un mapa 2D. Pahlavan, K. (2019) en Introducción a los algoritmos de posicionamiento.

2. ***Método LS (Least Square) para posicionamiento***. Tomando mas de 2 puntos de referencia con el objetivo, se puede utilizar DOA para obtener el ángulo entre los puntos de referencia y el objetivo y la distancia entre ellos. Los radios generados entre los puntos de referencia y el objetivo es llevado a un proceso de buscar la distancia mas corta por el método de mínimos cuadrados. Este método es usado en distintos métodos de posicionamiento cuando se obtienen múltiples puntos de aproximación para localizar el objeto. Pahlavan, K. (2019) en Introducción a los algoritmos de posicionamiento.

2.3.4. Aseguramiento de la calidad e interferencias de la señal

2.3.4.1. Selección de Radio Frecuencia y Ancho de banda . La selección de la frecuencia requiere considerar las restricciones como el espacio libre que causa la dispersión de la señal, los obstáculos como columnas y paredes que limitan la propagación de la señal y un costo mayor a frecuencias muy altas. «El aumento de la frecuencia también aumentará las pérdidas de propagación en el espacio libre, aumentará las pérdidas por dispersión y penetración (como a través de paredes) y, en frecuencias muy altas (>10 GHz), resultará en componentes electrónicos más caros (Sharp, I. et. al. 2019)» . (Sharp, I. , Yu, K. 2019) en Diseñando sistemas de posicionamiento.

2.3.4.2. Características de propagación de las ondas de radio frecuencia en el entorno. La forma como se propaga la señal en un entorno cerrado y los obstáculos tiene efecto en el rendimiento de localización y tiempos de espera para medir adecuadamente la distancia. (Sharp, I., Yu, K. 2019) en Diseñando sistemas de posicionamiento.

2.3.4.3. Cálculos de la distancia del dispositivo. Los procesos de localización de un dispositivo mediante ondas de radio se ven afectado por las interferencias de la señal y la concurrencia en la banda que afecta en la

obtención de la distancia y demoras en obtener una precisión muy cercana del dispositivo. Jondhale, S. et. al. (2022) en Localización y seguimiento de objetivos mediante WSN, (Sharp, I. , Yu, K. 2019) en Diseñando sistemas de posicionamiento.

2.3.4.4. Sincronización de tiempo y frecuencia . Los medidores de control de tiempo entre el emisor y el receptor deben estar sincronizados para aplicar mediciones de tiempo de tal forma que en un proceso bidireccional se envía una pequeña trama con la marca de tiempo para poder obtener el tiempo de llegada al receptor de la trama. Morton, Y et. al. (2021) en Descripción general de las técnicas de navegación en interiores, Yu, K. (2018) en Estimación de posición para sistemas IR-UWB.

2.3.5. Marco de trabajo del sistema operativo Android

2.3.5.1. Plataforma operativa.

Android 10 o superior
Lenguaje de desarrollo: JAVA
SDK API Level 29
Entorno de desarrollo: Android Studio Giraffe | 2022.3.1

1. ***Características***. Las Herramientas de la plataforma del SDK de Android son un componente del SDK de Android. Se incluyen herramientas que interactúan con la plataforma de Android, principalmente ADB y FASTBOOT.
2. ***Lenguaje de desarrollo***. Android SDK se aplica a 2 lenguajes de desarrollo: KOTLIN y JAVA. En el presente estudio se usa el lenguaje de programación JAVA lo que involucra utilizar la interface en Lenguaje de programación JAVA.

2.3.5.2. Componentes de software . Librerías para el acceso a los sensores y proveedores de localización. La plataforma ofrece todas las Interfaces para acceder a los recursos del dispositivo.

1. ***Android Location Framework. Interfaces***:

    a. GnssStatus.Callback. Se utiliza para recibir notificaciones cuando ocurren eventos GNSS. Es aplicable cuando el dispositivo usa el proveedor GPS.

    b. LocationListener. Se utiliza para recibir notificaciones cuando la ubicación del dispositivo ha cambiado.

    c. LocationManager. Servicios de actualizaciones periódicas de ubicación geográfica del dispositivo.

    d. LocationProvider. Información sobre las propiedades del proveedor respecto a los requerimientos del servicio GPS, RED , Satélite, Antenas de celulares, y otros como aseguramiento de la precisión. Contiene los algoritmos para el cálculo de posicionamiento y dependiendo de la selección de mejor proveedor (Passive/Network / GPS / FUSED). Por defecto y de forma estándar el sistema operativo solo considera como proveedores a Network y GPS. El proveedor de localización fusionada «FUSED» solo es proveido por las APIS GOOGLE PLAY SERVICES, propiedad de Google. Este servicio de componentes utiliza las señales de distintos dispositivos (Bluetooth, Wi-Fi, GPS) para obtener la posición del dispositivo optimizando el consumo de energía.

    e. Criteria. Devuelve el nombre del proveedor (Passive, Network / GPS / FUSED) que mejor cumple con los criterios como parámetro dado. Solo se devolverán los proveedores a los que se pueda acceder de acuerdo al criterio de mayor precisión.

f. GnssMeasurement. Clase para obtener mediciones GNSS sin procesar.

2. ***Android Hardware, Sensor manager***. Proporciona la clase de acceso a los sensores del dispositivo. Utiliza los servicios del Acelerómetro y el magnetómetro.

3. ***Android.hardware. Interface***. Brinda soporte para funciones de hardware como la rotación del dispositivo e interface de control de la pantalla del dispositivo.

2.3.5.3. Hardware de posicionamiento y localización . Considerar los sensores que permiten poder posicionar un dispositivo, considere que los enrutadores Wi-Fi y punto de acceso pueden brindar información de posición física y distancia mediante la intensidad de la señal.

1. ***Tarjeta de RED inalámbrica - Wi-Fi*** . Con el protocolo de comunicación IEEE 802.11mc permite al dispositivo calcular la distancia de los enrutadores y puntos de acceso que lo soporten mediante el envío de señal y tiempo de respuesta. Con mas de un Enrutador Wi-Fi se puede establecer para el cálculo de su posición se puede establecer el mecanismo de trilateración para obtener la posición del dispositivo en un espacio cerrado. El cálculo de la distancia es realizado por el dispositivo solicitante, el Enrutador solo devuelve la respuesta de reconocimiento.

2. ***Receptor GNSS*** . Sensor GNSS que obtiene la información de señales GNSS para el cálculo de su posición. Los satélites GNSS brindan información de posición, tiempo y distancia para que el dispositivo realice el cálculo de su posición.

3. Sensores

a. ***Sensores de movimientos***. La plataforma Android proporciona 2 grupos de sensores, por Software y por Hardware:

i. Por Software. Sensores que retornan información procesada y se basa en sensores de Hardware. Estos son sensores de aceleración, de gravedad, de rotación y de movimiento, permiten detectar la vibración, inclinación y balanceo.

ii. Por Hardware. Los sensores del acelerómetro y del giroscopio siempre están basados en hardware.

b. ***Sensores de posición***. La plataforma Android proporciona dos sensores que permiten determinar la posición de un dispositivo: el Magnetómetro y el acelerómetro.

CAPITULO III. METODOLOGÍA

3.1 Métodos y procedimientos a ser utilizados

El tipo de investigación es APLICADA y modo EXPERIMENTAL, utiliza el estado de las artes del conocimiento científico (I+D) para implementar el proceso de geo localización y posicionamiento geográfico utilizando la plataforma Android en un dispositivo móvil. La investigación no requiere una hipótesis porque es un caso de estudio que explora el desarrollo e implementación del procedimiento para la geo localización y posicionamiento geográfico usando los Framework e interface programables del sistema operativo Android para dispositivos móviles.

Las actividades del proyecto se estructuran siguiendo el ciclo de vida del desarrollo de software (Software Development Life Cycle - S.D.L.C.), se aplica el modelo ágil como proceso de gestión del desarrollo del software. Se utiliza la metodología UML (Unified Modeling Language) como técnica de modelado del software. El propósito es guiar paso a paso el desarrollo de la Tesis Maestría en Computación e Informática; sobre cómo realizar investigaciones científicas, para desarrollar la tesis titulada «Sistema de ubicación en áreas terrestres basada en mapas a escala, utilizando Geo localización de dispositivos móviles».

3.3. Unidad de análisis.

1. Framework e Interfaces de componentes programables JAVA del sistema operativo Android que ejecutan las funciones de posicionamiento (representadas por las coordenadas geográficas de Latitud y Longitud) y control

de sensores del dispositivo para obtener el desplazamiento del dispositivo móvil.

a. android.location. Framework. Conjunto de procedimientos especializados de acceso al servicio de navegación satelital GNSS y servicio de localización por coordenadas terrestres.

b. android.hardware. Interface. Acceso a los sensores hardware como el acelerómetro, giroscopio y sensor de intensidad de la señal inalámbrica.

2. Área terrestre de la Ciudad Universitaria de la Universidad Nacional de San Marcos. El área es representada por Plano cartográfico de la Ciudad Universitaria de la Universidad Nacional de San Marcos a coordenadas geográficas UTM.

3.4. Población de estudio

Es el área del plano del plano cartográfico del Campus universitario sobre el plano terrestre, se toma 4 puntos (Posición 1 inferior izquierdo, Posición 2 inferior derecho, Posición 3 superior derecho y Posición 4 superior izquierdo) representados en coordenadas geográficas grados decimales. (Ver tabla 1).

Tabla 1. Área del plano Cartográfico del Campus universitario. Fuente propia.

Nombre del Campo	Valor de campo	Descripción
Posición 1 Latitud,Longitud	-12.062220,-77.088526	Posición 1 inferior izquierdo
Posición 2 Latitud,Longitud	-12.062292,-77.078894	Posición 2 inferior derecho
Posición 3 Latitud,Longitud	-12.050543,-77.078803	Posición 3 superior derecho
Posición 4 Latitud,Longitud	-12.050471,-77.088435	Posición 4 superior izquierdo

El área contiene un conjunto de coordenadas geográficas, los elementos de la población se obtienen con el Software «Google Earth Pro». El sistema Google Earth proporciona coordenadas geográficas estáticas sobre la superficie terrestre (Estándar World Geodetic System WGS84), en nuestro caso, todos los puntos dentro del área del plano proyectado sobre el plano terrestre. El área se denomina en la investigación como «Área de interés».

De ha definido una categorizacion para los puntos de interés, esta clasificación se denomina «Categoría» y se define como un atributo del punto de interés. (Ver tabla 2).

Tabla 2. Definición de categorías. Fuente propia.

Identificador	Nombre
1	Facultad
2	Administración
3	Talleres
4	Comedor
5	Vivienda
6	Biblioteca
7	Clínica y/o Consultorio
8	Seguridad
9	Laboratorio
10	Deportes
11	Otros

Se selecciona un sub conjunto de coordenadas geográficas estáticas del área de interés con el sistema de Software Google Earth Pro, las coordenadas corresponden a los locales de interés en el plano del Campus universitario proporcionado por el área de Infraestructura de la Universidad Nacional Mayor de San Marcos del año 2023. Estos puntos terrestres se denominan «Puntos de interés» que son utilizados por el sistema para visualizar el posicionamiento del dispositivo móvil en el mapa digital y mostrar la trayectoria de llegada al punto de interés como destino. (Ver tabla 3).

Tabla 3. Definición de los puntos de interés. Fuente propia.

Número Local	Nombre Facultad
1	Ingeniería de Minas
2	Facultad de Psicología
4	Microbiología y crematorio del Instituto de Medicina Tropical
8	Facultad de Ingeniería Electrónica y Eléctrica
9	Facultad de Ingeniería de Sistemas e informática
10	Facultad de Odontología
11	E.A.P. de Educación Física
13	Facultad de Educación
19	E.A.P. Mecánica de fluidos
20	Facultad de Letras y Ciencias
21	Facultad de Ciencias Administrativas
22	Facultad de Ciencias Sociales
23	Facultad de Ciencias Económicas y Contabilidad
24	Facultad de Derecho y Ciencias Políticas
27	Laboratorio de la Facultad de Ciencias Físicas
28	Facultad de Ingeniería Industrial
29	Facultad de Ciencias Matemáticas
30	Facultad de Química e Ingeniería Química
31	E.A.P. Ingeniería Metalúrgica
32	Facultad de Geología
36	Facultad de Ciencias Físicas
43	Facultad de Ciencias Biológicas
44	E.A.P. de Geografía
45	E.A.P. de Ingeniería Geográfica
46	Pabellón aulas Facultad de Ciencias Biológicas
48	Ambientes de E.A.P. Ingeniería Metalúrgica
52	Facultad de Ingeniería Electrónica y Eléctrica
54	E.A.P. Ingeniería Civil
61	Helicario
62	Apiario
65	Pabellón nuevo Facultad de Ciencias económicas
66	Pabellón nuevo Facultad de Letras
3	Instituto de Medicina tropical
12	Ambientes académicos administrativos Educación
15	Sede administrativa central "Jorge Basadre" y Auditorium

26	Red Telemática y Tecnología de información
47	Centro de Producción
51	Ambiente Sindicato de Docentes
55	Oficina Central Admisión
56	Instalación de Centro Pre universitario
57	Instalación de escuela de Post grado
59	Ambiente de auto seguro UNMSM
7	Talleres y Maestranza
37	Oficina y Talleres de Maestranza
18	Comedor de Facultad de Odontología
34	Comedor Universitario y Bienestar
41	Comedor Facultad de Letras
16	Vivienda universitaria EX-Cinfo
17	Vivienda universitaria
14	Biblioteca central "Pedro Zulen"
5	Consultorio del Instituto de Medicina Tropical
53	Clínica Universitaria
58	Clínica de Psicología
60	Oficina de seguridad y vigilancia
63	Laboratorio de Geología
33	Gimnasio UNMSM
35	Estadio Olímpico San Marcos
50	Piscina Universitaria
6	Bioterio del Instituto de Medicina Tropical
25	Librería y distribución - Cenprolid
38	Modulo de servicios múltiples
39	Terreno vacio
40	Pozo de Tanque de Agua y Caseta
42	Modulo de Ventas
49	Modulo de Comercio
64	Planta piloto JICA

3.5. Tamaño de la muestra

1. Selección de 11 coordenadas geográficas aleatorias que son 11 puntos dentro del área de interés, la finalidad es validar si el sistema detecta que la coordenada geográfica calculada sea reconocida para activar el proceso de posicionamiento en tiempo real.

2. Selección de 3 puntos de interés como destino y cada uno con un punto de partida seleccionado de forma aleatoria. Para cada trayectoria se seleccionan de forma dirigida 10 coordenadas geográficas. El total de la muestra: 3 trayectorias * 10 coordenadas geográficas = 30 coordenadas geográficas.

El total de coordenadas geográficas seleccionadas son: 41

3.6. Selección de la muestra

La selección de la muestra es no probabilistico, el investigador dirige la selección de de las coordenadas geográficas por cada trayectoria obtenidas por el sistema móvil. La comparación de los valores resultantes se contrasta contra los valores obtenidos por el Sistema de Software Google Earth. Se infiere que el «Google Earth Pro» contiene los valores reales de las coordenadas geográficas y el sistema móvil retorna las coordenadas calculadas en tiempo real.

Las variables de un elemento de la muestra consta de 3 valores especificados como:

1. Coordenada geográfica decimal estática obtenida con el Software Google Earth Pro, llamada «Coordenada geográfica estática».

2. Coordenada geográfica decimal obtenida por el sistema móvil instalado en el dispositivo móvil, llamada «coordenada geográfica calculada».

3. Distancia en metros entre los 2 puntos de coordenadas geográficas, la coordenada geográfica estática y la coordenada geográfica calculada. La diferencia entre ambas determinan la precisión en metros.

Los margenes se agrupan según la clasificación de niveles que se establecerán en el trabajo de campo. Se agrupa por nivel de precisión porque los sensores pueden varian la posición en el transcurso del tiempo, el motivo se origina porque el algoritmo del Framework de localización refresca la localización de forma aleatoria por un cambio en la información de los sensores (Calidad de servicio del sensor), cambios en la señal de radio frecuencia Wi-Fi, cambio de información de la red del operador de telefonía «Network Based», por ingreso de información del servicio satelital GNSS. Cada nivel tiene un rango de precisión en metros respecto al punto geográfico objetivo. Se definen 4 niveles y se describen los posibles motivos (ver tabla 4).

Tabla 4. Definición de los niveles de precisión en metros. Fuente propia.

Nivel de precisión	Motivo
A	Puede ocurrir en un entorno de área física abierta, sin columnas de concreto o edificaciones de concreto.
B	Puede ocurrir en un entorno de área física abierta, con columnas de concreto y edificaciones.
C	Puede ocurrir en un entorno de área física cerrada o interiores semi abiertos, edificaciones y columnas de concreto.
D	Puede ocurrir en un Entorno de área física interior, con techos, edificaciones y columnas de concreto sin disponibilidad de puntos Wifi o puntos Bluetooth, pero con disponibilidad de servicio de telefonía. Bajo este escenario el posicionamiento puede ser cubierta por el servicio del operador de telefonía llamado GPS asistido A-GPS.

3.7. Materiales o instrumentos que se emplean

3.7.1. Dispositivos

1. SAMSUNG GALAXY A50. Sistema operativo ANDROID 11.0

2. SAMSUNG GALAXY S21. Sistema operativo ANDROID 13.0

3.7.2. Herramientas de desarrollo

3.7.2.1. Android Studio Giraffe | 2022.3.1

Android Studio Giraffe | 2022.3.1
Build #AI-223.8836.35.2231.10406996, built on June 29, 2023
Runtime versión: 17.0.6+0-b2043.56-10027231 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.
Windows 11 10.0
GC: G1 Young Generation, G1 Old Generation
Memory: 2048M
Cores: 6

3.7.2.2. SQL Browser para SQLite

Herramienta para la generación de la microbase de datos e inserción de los datos.
DB Browser for SQLite Versión 3.12.2
Built for x86_64-little_endian-llp64, running on x86_64
Qt Versión 5.12.8
SQLite Versión 3.35.5.

3.7.2.3. Google Earth Pro 7.3.6.9345 (64-bit)

Herramienta usada para obtener la información de coordenadas geográficas del área del Campus universitario.
Fecha de la compilación jueves, 29 de diciembre de 2022 10:50:09 P. M. UTC
Procesador: OpenGL
Sistema operativo: Microsoft Windows (6.2.9200.0)
Controlador de gráficos: Intel (00030.00000.00100.09805)
Tamaño máximo de textura: 16384×16384
Memoria de video disponible: 16446 MB
Servidor: kh.google.com

3.7.2.4. DWG TrueView Autodesk

Herramienta de AutoCAD para la conversión a imagen vectorial, edición del plano con pinceles a 0.5 de ancho para presentación en el dispositivo y edición del contenido del plano para eliminar contenido no necesario.
DWG True View (TM) for Windows (64-Bits) .
2023 Autodesk

3.7.2.5. BeckerCAD 3D PRO 14.0 Rev. 8273.317

Herramienta de edición de archivos DWG de AutoCAD vectoriales.
AutoCAD compatible format
Powered by DataSolid GmbH Copyright 2022

3.7.3.6. Inkscape 1.3 (0e150ed6c4, 2023-07-21)

Herramienta de edición de archivos vectoriales SVG, usado para generar las áreas delimitadas de los locales en el mapa ya convertido a SVG. Se importa el archivo vectorial generado por DWG TrueView en formato PDF a la herramienta de edición y luego se genera el archivo SVG para la edición mencionada.
Open source vector graphics editor for Windows (64-Bits)
Inkscape uses the standardized SVG file format as its main format

CAPITULO IV. INGENIERÍA DE DESARROLLO DEL SOFTWARE

4.1. Formulación de los requerimientos

4.1.1. Historia de usuario

El usuario con el sistema móvil llamado «Geolocation» instalado en un dispositivo móvil, obtiene su ubicación y posición dentro del plano digital del Campus universitario que es gestionado por el sistema móvil y se denomina «Área de interés», permite localizar locales definidos como «Punto de interés» que se encuentran dentro del «Área de interés», cada local se referencia por una coordenada geográfica en grados decimales.

Los locales se agrupan por función que se denomina «Categoría», por ejemplo: Facultad, Administración, Taller, etc. El sistema móvil permite al usuario poder consultar los puntos de interés por «Nombre» del local o por la «Categoría». El sistema móvil puede mostrar el punto de interés en el mapa digital y se denomina «Localización». El sistema móvil cuenta con la capacidad de guiar la trayectoria del usuario al local seleccionado que se denomina «Punto de destino». El sistema móvil puede guiar la trayectoria mediante una brújula hacia el punto de interés como punto de destino en tiempo real y se denomina «Posicionamiento».

4.1.2. Requerimientos funcionales

1. Mostrar el mapa cartográfico digital del Campus de la Ciudad Universitaria de la Universidad Nacional Mayor de San Marcos por defecto como inicio. Se define como «Área de interés»

2. Buscar «Punto de interés» por nombre del Local o sub cadena del nombre del local en una barra de búsqueda que permita retornar los registros encontrados.

3. Mostrar los resultados de la búsqueda del local en una lista, estos pueden ser mas de uno si existieran pero solo un registro debe seleccionado.

4. Visualizar en el mapa digital la posición del punto de interés seleccionado de la barra de búsqueda indicando un icono en el mapa y centrando la imagen del plano digital respecto a punto de interés.

5. Consultar la lista de categorías de de los puntos de interés y mostrar la lista de Locales por cada categoría seleccionada, estos pueden ser mas de uno si existieran pero solo un registro debe seleccionado.

6. Capacidad del sistema de guiar visualmente la trayectoria hacia el punto de interés como punto de destino. La guía visual debe mostrar un icono con un indicativo que oriente la trayectoria al punto de destino.

7. Informar visualmente la ubicación geográfica del dispositivo sobre el plano digital. Actualizar en intervalos de tiempo el cambio de ubicación que se denomina posicionamiento.

8. Indicar con un icono la posición en tiempo real del dispositivo. El indicador visual del dispositivo se denomina «Brújula» y solo se activa si el dispositivo se encuentra dentro del área de interés.

4.1.2. Requerimientos no funcionales

4.1.2.1. Del sistema móvil

1. Los componentes usados como librerías, Framework y recursos son nativos del sistema operativo Android 10.0 y soportado en versiones superiores.
2. Los procesos de acceso a recursos externos se ejecutan de forma asíncrona con mensaje al usuario. Ocurre en el acceso a la base de datos y en la carga de recursos como la carga de las vistas.
3. Las cadenas alfanuméricas que forma parten de las vistas y los textos de los iconos se gestionan en el recurso String.xml.
4. Los intervalos 300 mili segundos para la llamada a la exploración de la localización.

4.1.2.2. Plataforma operativa y base de datos

1. Soporte de plataforma operativa . Android 10 o superior.
2. Micro servicio de Base de datos. SQLite 3.12 o superior.

4.1.2.3. Lenguaje de programación y entorno de desarrollo

1. Lenguaje de programación. Java development Kit 8.0.
2. Entorno de desarrollo. Android Studio Giraffe 2022.3.1

4.1.2.4. Datos pre definidos

1. Lista de puntos de interés definidos como «Local». Cada elemento con atributo de Nombre y coordenada geográfica decimal.
2. Lista de Categorías que agrupan los locales por función definido como «Categoría». Cada elemento con atributo de Nombre e Identificador de Categoría.

3. Lista de las coordenadas geográficas grados decimales del mapa cartográfico del Campus universitario que define el área del Campus universitario en el plano terrestre. Cada elemento es un vértice del plano para un total de 4 puntos geográficos en grados decimales.

4. Las coordenadas que procesa el sistema son coordenadas geográficas grados decimales usando el formato «double», el formato de la coordenada geográfica almacenada en la base de datos usa el formato doble con 6 decimales.

5. Las coordenadas del plano cartográfico del Campus universitario están definidos en coordenadas UTM (Universal Transversal de Mercator).

6. Los datos pre definidos se almacenan en la micro base de datos SQLite reconocido por defecto por los componentes nativos Android de acceso a datos.

7. Los datos pre definidos se almacenan sobre una estructura de base de datos previamente modelada para SQLite.

8. La micro base de datos se instala con el sistema en el dispositivo móvil, definido como un activo del sistema móvil.

9. Los procedimientos de gestión de la micro base de datos se realizan con las librerías estándar de base de datos Java para Android 10.0 o superior.

4.1.2.5. Plano cartográfico digital

1. Formato de imagen convertido. PNG con transparencia ALFA.

2. Dimensión. 4489 pixeles de ancho x 5561 pixeles de alto.

3. Peso. 6.66 MB

4. Nombre. Ciudaduniversitariaoptvertical.png

5. Definición en el proyecto. Recurso del sistema móvil.
6. Fuente. Área de infraestructura de la Universidad Nacional Mayor de San Marcos.
7. Formato de origen. AutoCAD DWG
8. Descripción. Plano cartográfico digital del Campus universitario en unidad de coordenada UTM.

4.1.2.6. Componentes y librerías estándar Android

1. android.location. Framework.
2. android.hardware. Interface.

4.1.2.7. Permisos en la plataforma Android

1. android.permission.ACCESS_FINE_LOCATION
2. android.permission.ACCESS_COARSE_LOCATION
3. android.permission.WRITE_EXTERNAL_STORAGE

4.1.2.8. Generación del instalador

Formato APK de instalación libre con soporte Android 10 o superior.

4.1.2.9. Precisión del dispositivo móvil

Usar la recomendación de la plataforma operativa Android 10 o superior para el proceso de localización condicionado a la evaluación del investigador.

4.2. Diseño de la interface visual

4.2.1. Modelo conceptual de la interface visual

Distribución de las definición de interface visual y/o procesos principales que interactúan con las Interfaces. «Ir» y «Recargar» con opciones disponibles en la navegación en el mapa digital, la primera para ir al punto de destino y la segunda para actualizar la información y visualización de los datos en la interface visual. (ver Figura 1)

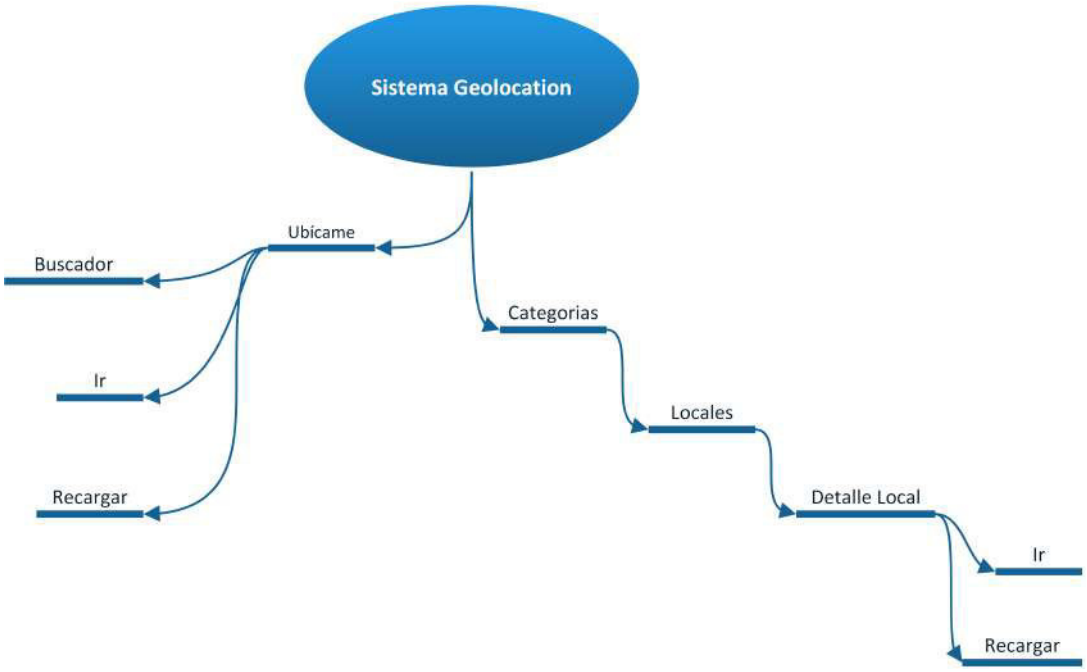

Figura 1. Modelo conceptual de interface visual. Fuente propia

4.2.2. Interface visual «Ubícame»

Localización de las facultades y otros locales del Campus universitario según definición del plano. Cada punto localizado utiliza las coordenadas geográficas de

Latitud y Longitud y se visualiza en el plano posicionado en la pantalla del dispositivo usando como referencia la imagen del plano.

 Buscador de locales del Campus universitario por nombre del local, con ejecución de localización del punto geográfico del local encontrado. Tiene opción de sugerencia conforme ingresa el texto en la caja de búsqueda.

 Menú superior derecho con acceso corto a las opciones del sistema, el menú llama al proceso «Categorías».

 Recarga de pagina para inicializar la pantalla con los valores por defecto, implica que deshabilita los servicios de localización y posicionamiento, de igual forma oculta los íconos en el mapa visual.

 Un botón llamado «IR» que permite de forma dinámica posicionar el dispositivo y activar la brújula del local de destino para orientar al usuario donde se localiza el destino. Este botón solo se visualiza cuando ha seleccionado un local de destino y el sistema entra en modo de localización.

El proceso muestra 3 puntos visibles dentro de la pantalla del dispositivo:

Ícono de punto que indica la posición del local	
Ícono circular giratorio que indica la posición del dispositivo	
Ícono de Brújula que apunta al destino conforme se desplaza el usuario con el dispositivo.	

La interface principal del sistema contiene en la parte central el mapa digitalizado del Campus universitario, en la parte de la cabecera dispone de un buscador de locales y en el superior derecho de un menú de acceso a las categorías y la presentación principal (ver Figura 2).

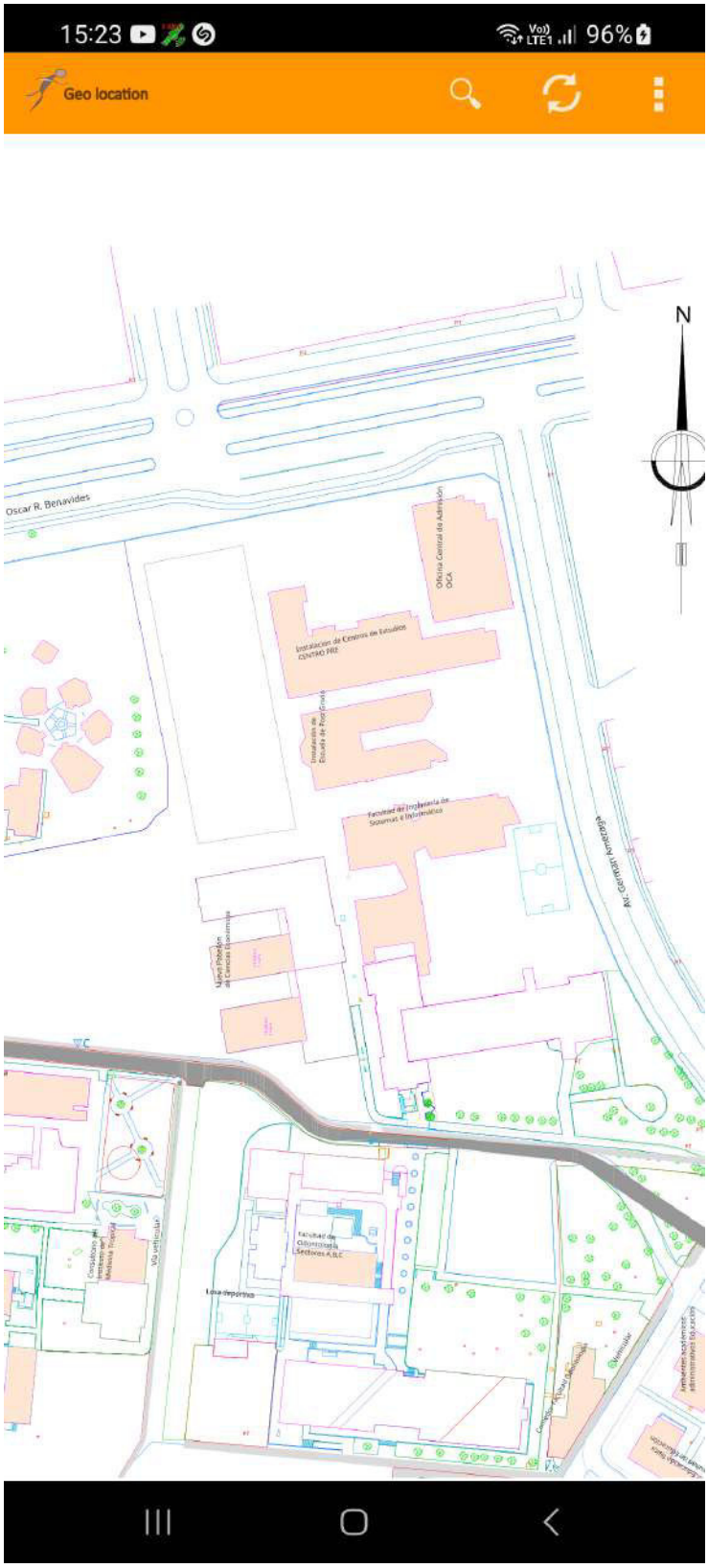

Figura 2. Entrada principal del sistema. Fuente propia.

En la pantalla principal existe la opción de búsqueda que es la barra superior con el icono de la lupa. La búsqueda realiza la visualización en una lista despegable filtrando los resultados según el texto ingresado. Los elementos de la lista desplegada son seleccionables, solo se puede seleccionar un elemento de la lista (Figura 3). Al seleccionar el elemento, indica al sistema que ubique el local en el mapa digital indicando con un icono su posición (ver Figura 4). Visualizado el icono de la posición en el mapa, el sistema activa el icono de «Ir» que es el evento que seleccionado activa el posicionamiento en tiempo real y orienta con la brújula la trayectoria al punto de interés como punto de destino, tomando como punto de partida la posición actual del dispositivo (ver Figura 5). En la sección inferior se muestra la caja de información en tiempo real de las coordenadas geográficas, rumbo, método de acceso a la información y la disponibilidad de servicio de satélite GNSS (ver Figura 5).

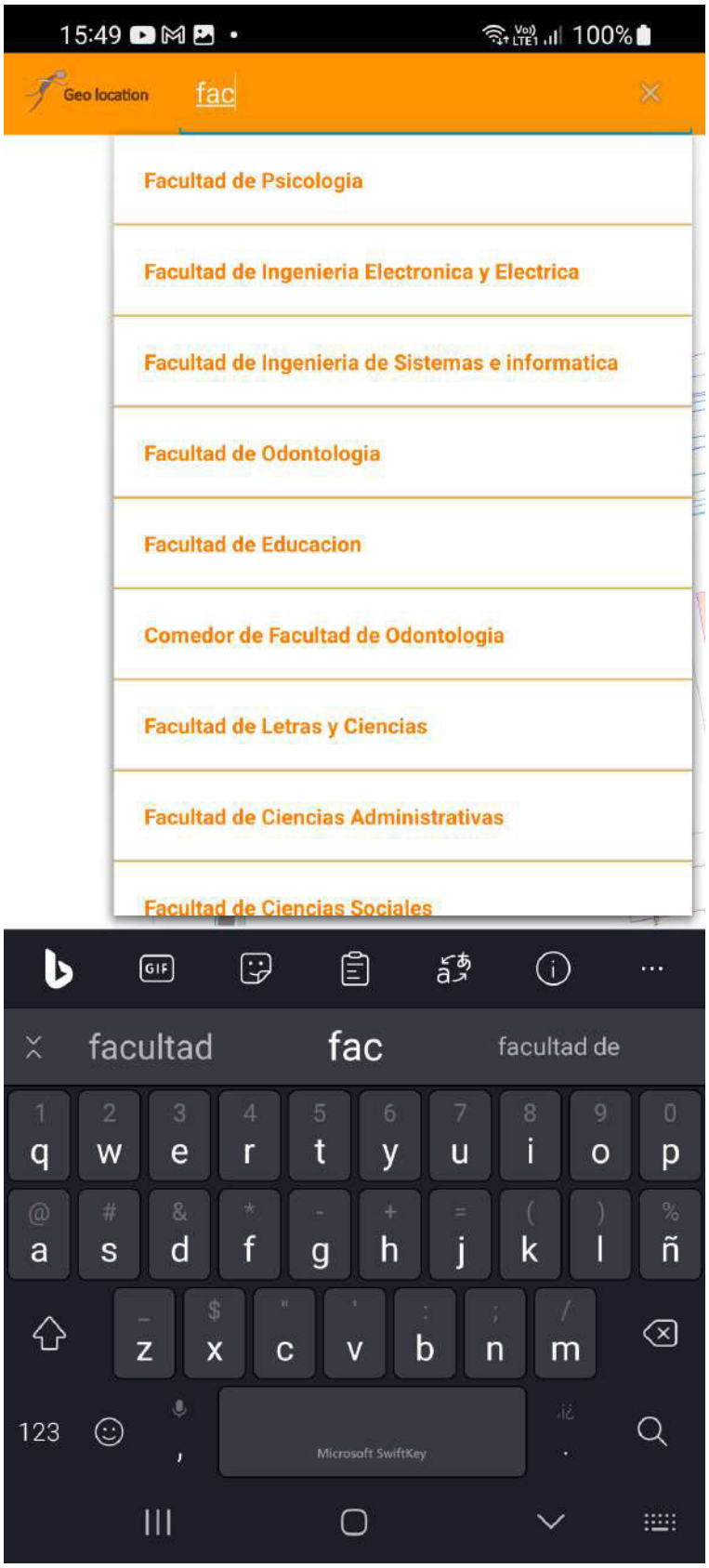

Figura 3. Buscador y sugerencia. Fuente propia.

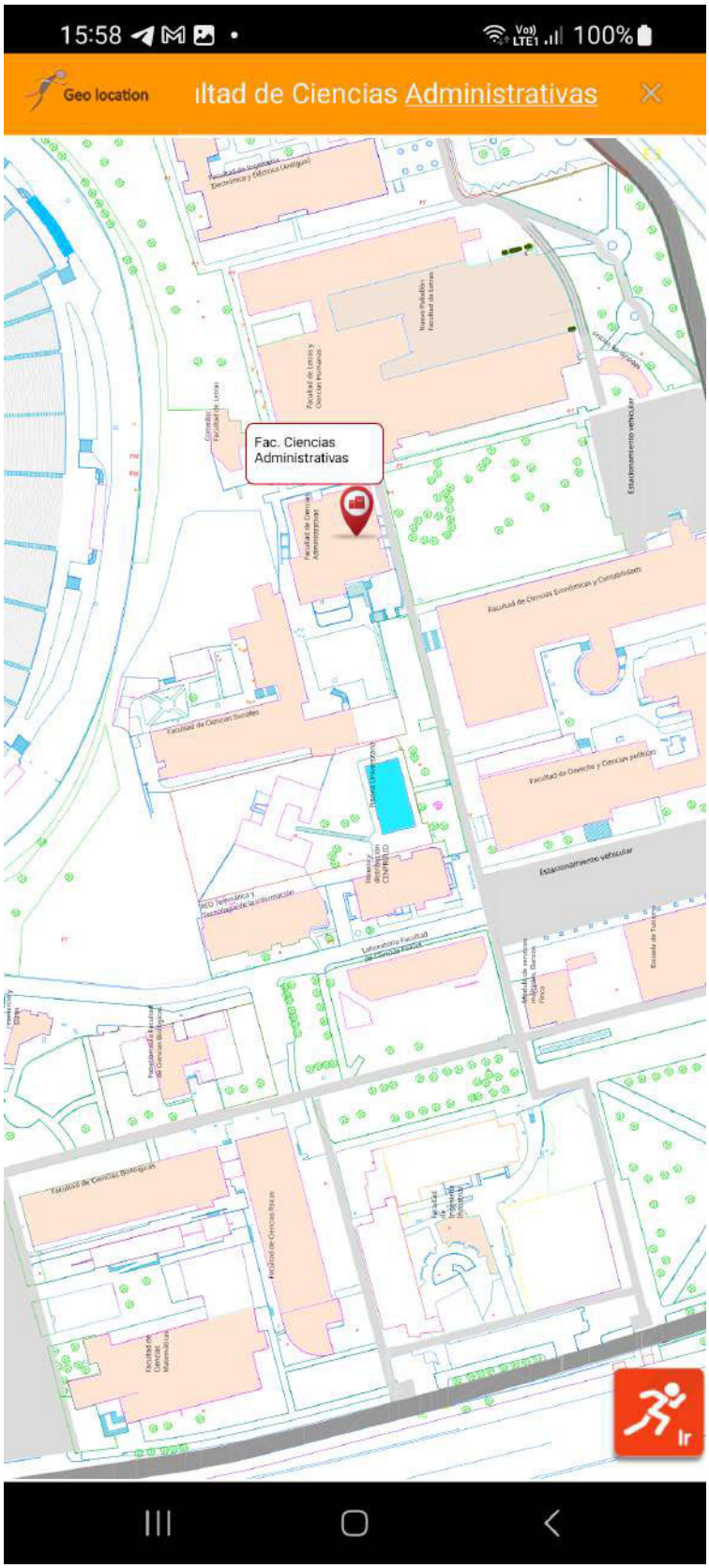

Figura 4. Posicionamiento del Local. Fuente propia.

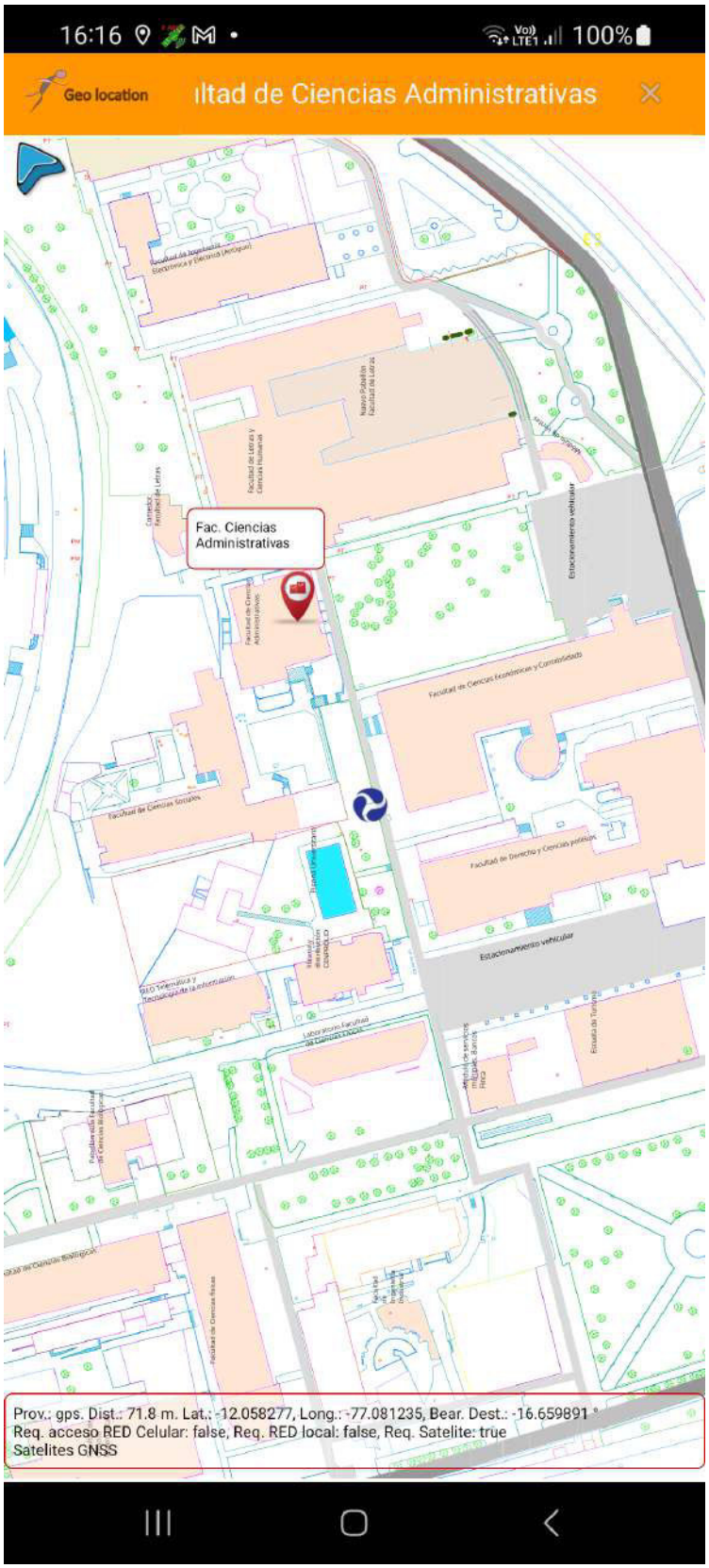

Figura 5. Posicionamiento del dispositivo y rumbo a destino. Fuente propia

4.2.3. Interface visual «Categorías»

Lista las categorías de los locales. Cada categoría agrupa un conjunto de locales. Al acceder a algún elemento de la lista muestra la lista de locales (ver Figura 6).



Figura 6. Lista de Categorías de Local. Fuente propia.

4.2.4. Interface visual «Locales»

Lista los locales. Cada local lleva a un enlace al detalle en el mapa y posicion visual (ver Figura 7). Muestra el nombre del local, número de local según plano UNMSM, Latitud y Longitud.

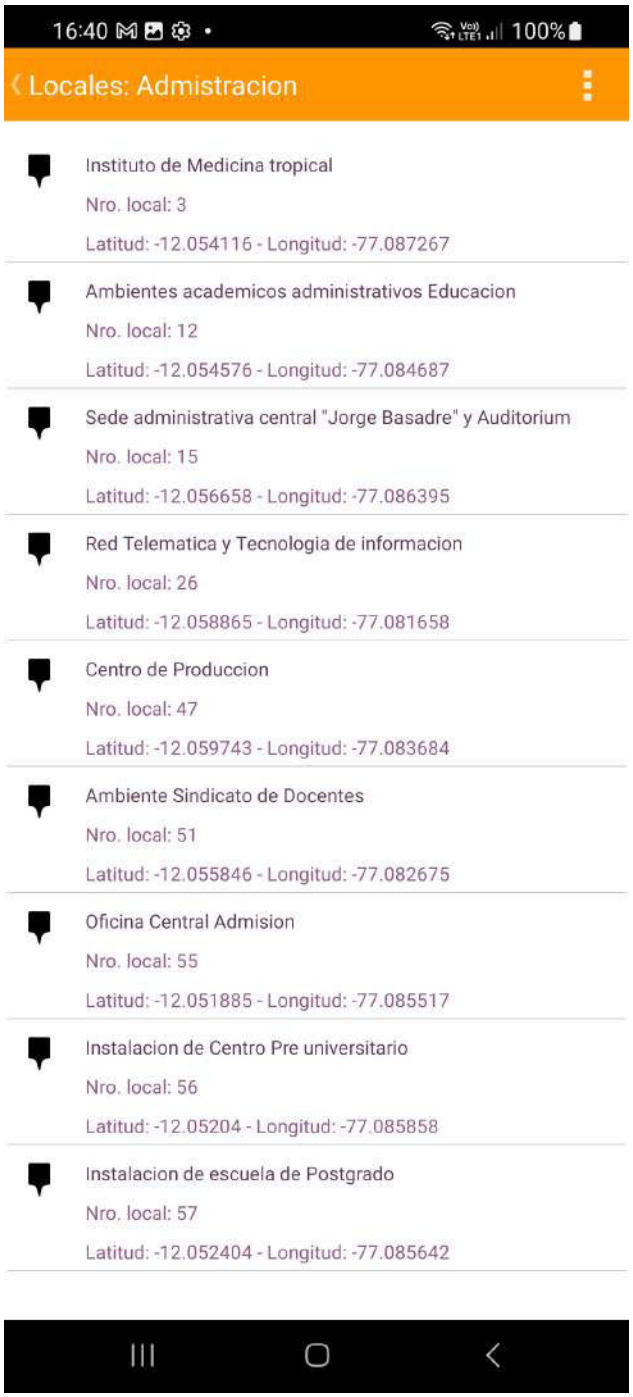

Figura 7. Lista de Locales por Categoría indicada. Fuente propia.

4.2.5. Interface visual «Detalle local»

Muestra el detalle visual del local posicionado en el mapa (ver Figura 8). Tiene la opción de posicionamiento en tiempo real y trayectoria desde la posición del dispositivo al punto de destino.



Figura 8. Detalle del Local y posición. Fuente propia.

4.3. Diseño de procesos del sistema

4.3.1. Diagrama de contexto

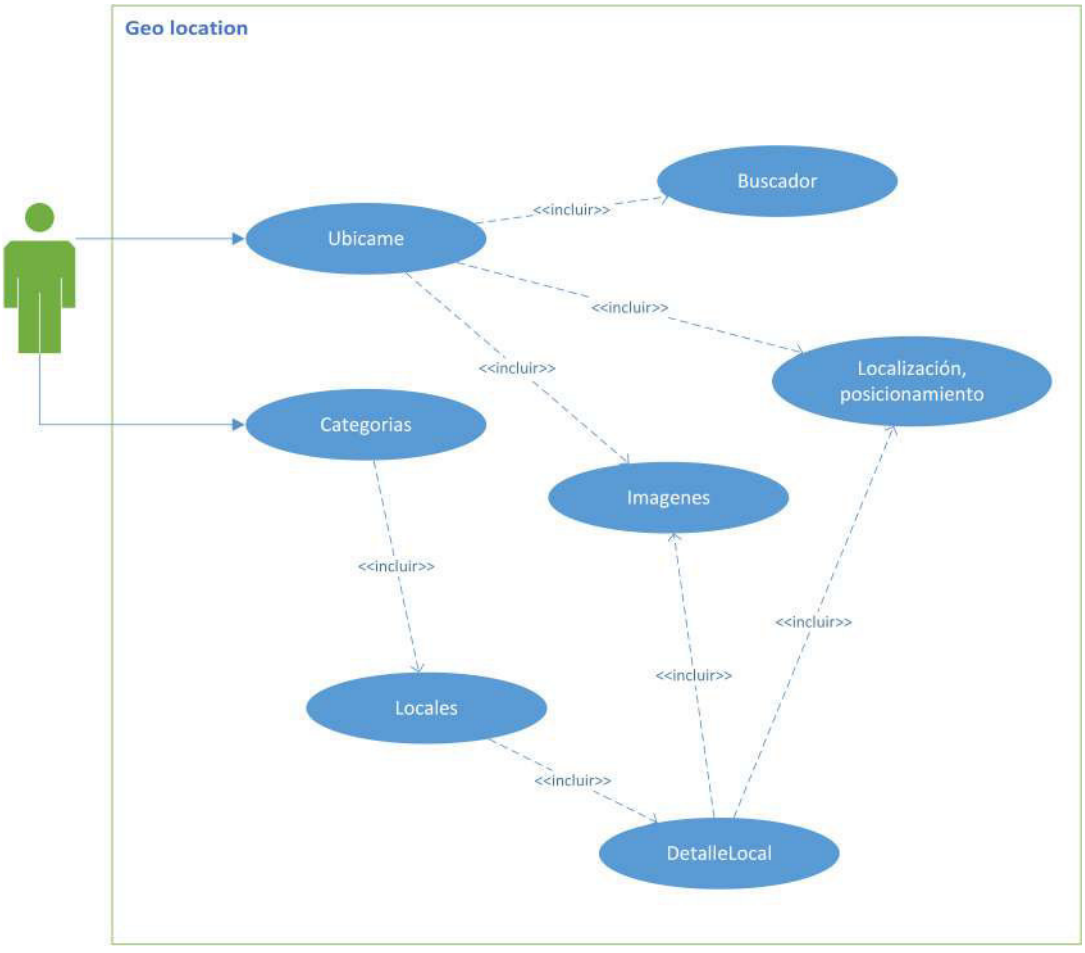

Figura 9. Diagrama de contexto. Fuente propia

4.3.2. Definición y diseño de procesos

4.3.2.1. Proceso Ubícame

Descripción. Proceso principal por defecto, activo en la carga del sistema. Cuenta con opción de posicionamiento en tiempo real y trayectoria al punto de destino. Cuenta con la opción de consulta de locales y muestra el resultado en una lista seleccionable para posicionar el local seleccionado en el mapa digital.

El proceso se divide en 3 bandas de ámbito. La primera banda es la inicialización de la vista, la carga de los iconos, asignación de los eventos y la carga de la imagen plano digital. La segunda banda es la activación del posicionamiento en tiempo real. La tercera banda activa el proceso de posicionamiento al punto de destino. Diagrama de actividad (ver Figura 10).

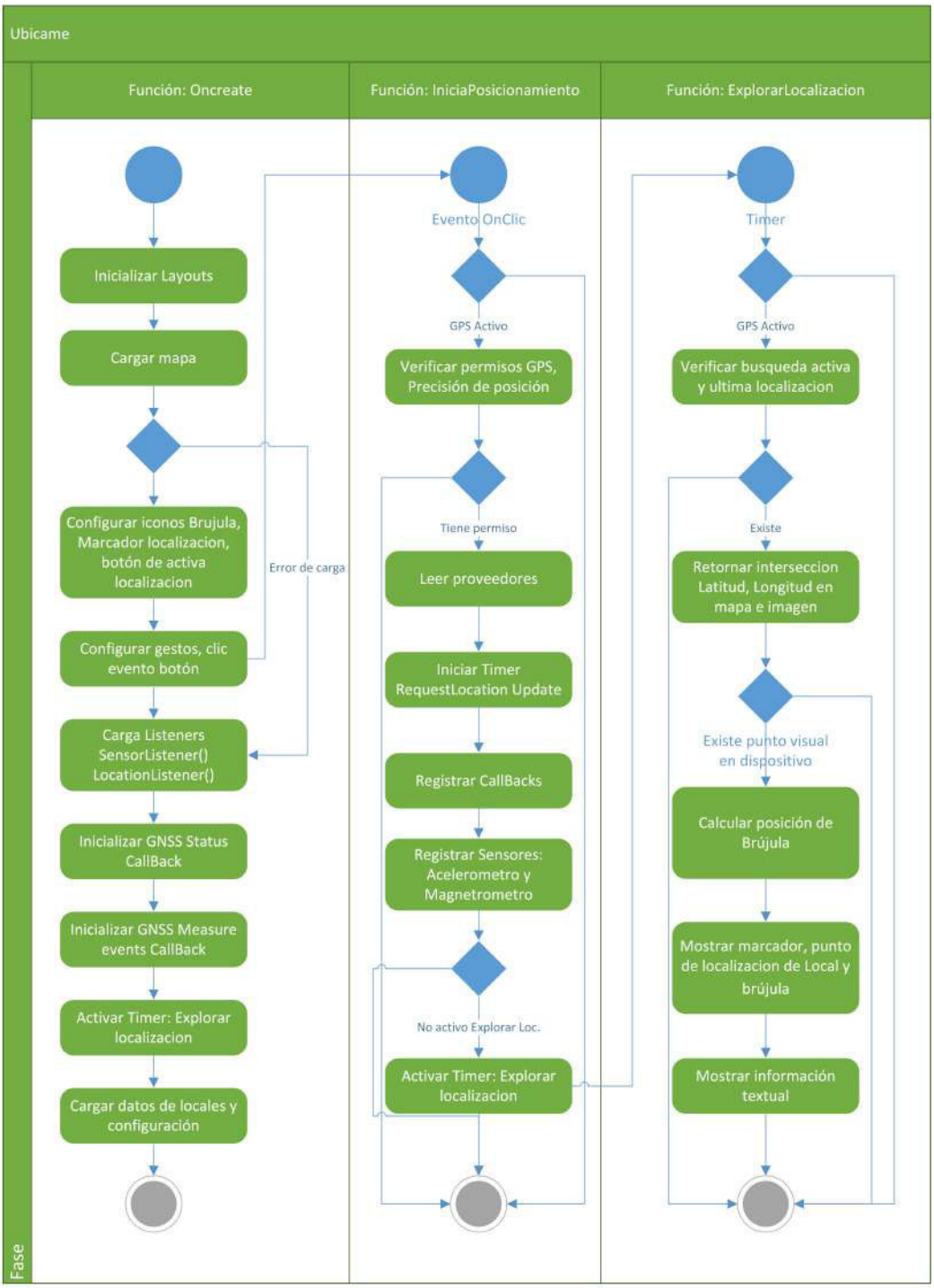

Figura 10. Diagrama de actividad de la clase Ubícame. Fuente propia.

4.3.2.2. Proceso CategoriaLocal

Descripción. Carga y visualiza la lista de categorías en una grilla, cada elemento de la lista es seleccionable para cargar la lista de locales de la categoría seleccionada. Diagrama de actividad (ver Figura 11).



Figura 11. Diagrama de actividad de Categoría. Fuente propia.

4.3.2.3. Proceso Locales

Descripción. Carga y visualiza la lista de locales en una grilla, cada elemento de la lista es seleccionable para cargar el detalle del local. Diagrama de actividad (ver Figura 12).



Figura 12. Diagrama de actividad de Locales. Fuente propia.

4.3.2.4. Proceso DetalleLocal

Descripción. Carga y visualiza el detalle del local con la posición de la ubicación en el mapa digital. Cuenta con opción de posicionamiento en tiempo real y trayectoria al punto de destino.

El proceso se divide en 3 bandas de ámbito. La primera banda es la inicialización de la vista, la carga de los iconos, asignación de los eventos y la carga de la imagen plano digital. La segunda banda es la activación del posicionamiento en tiempo real. La tercera banda activa el proceso de posicionamiento al punto de destino. Diagrama de actividad. (ver Figura 13).

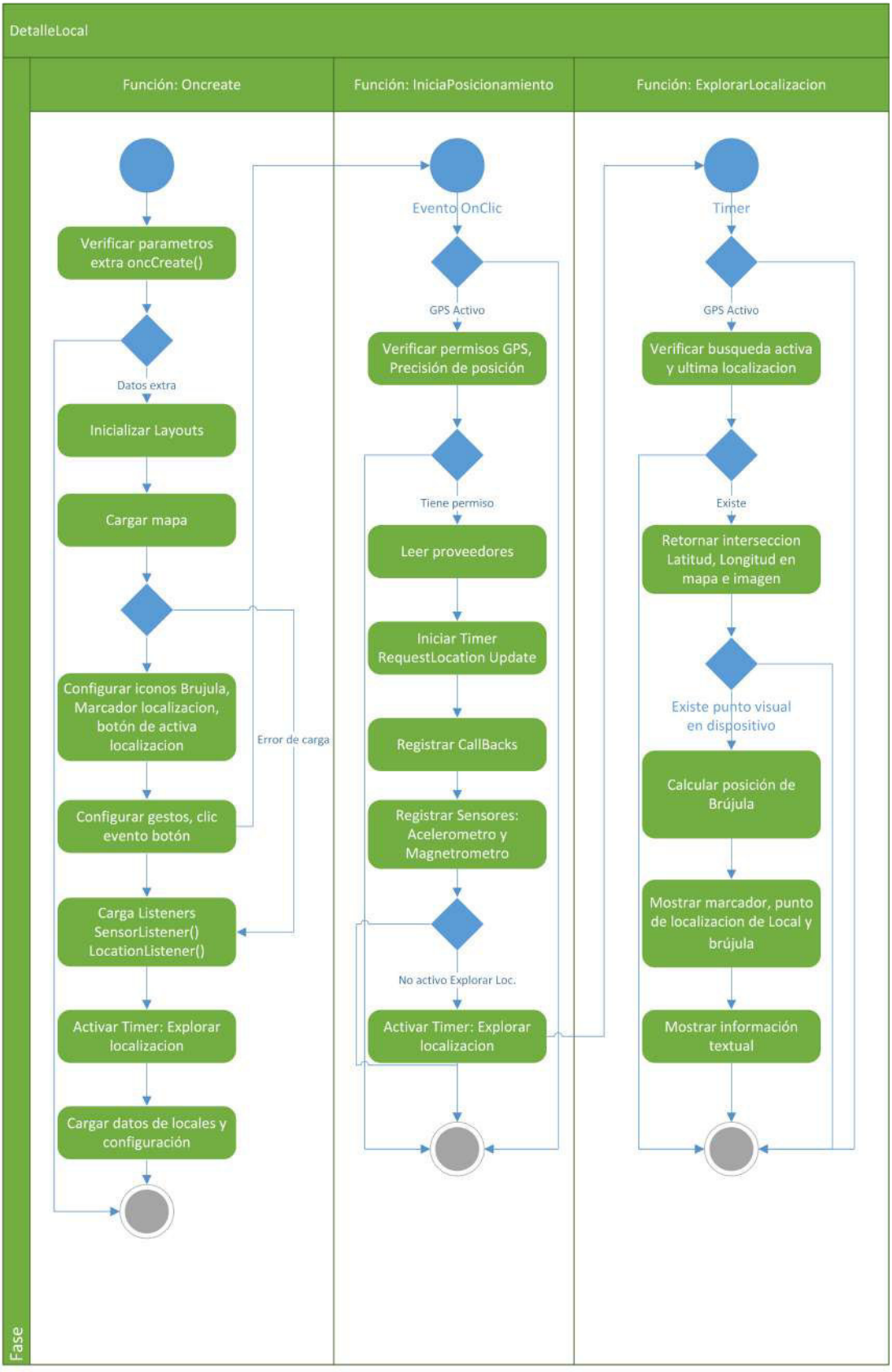

Figura 13. Diagrama de actividad de DetalleLocal. Fuente propia.

4.4. Diseño de arquitectura del software

4.4.1. Arquitectura del sistema

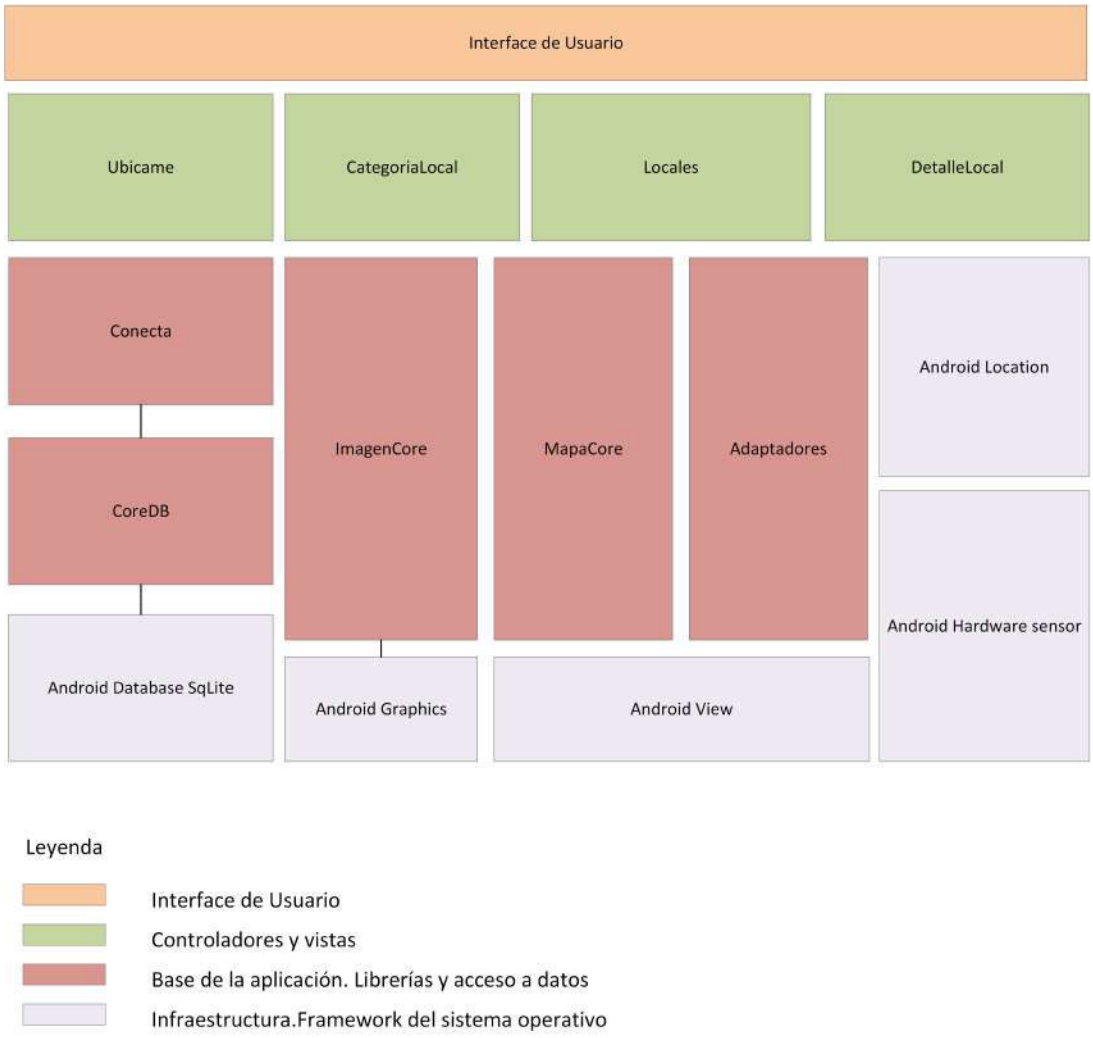

Figura 14. Arquitectura del sistema. Fuente propia

4.4.2. Estilo del diseño de la arquitectura

El estilo se diseña basado en el patrón de desarrollo Modelo Vista Controlador (MVC). El controlador gestiona el enlace de datos y la vista, los adaptadores son módulos de vista usado por el controlador para personalizar la visualización de los datos en la vista, así mismo, el controlador implementa los eventos de la vista e implementa el acceso a la base de datos y el proceso de acceso al «Android Location Framework» y «Android Hardware Sensor» si es requerido por el controlador. Las configuraciones de vista son los objetos XML y las constantes se definen en el objeto «String XML» que es requerido por el controlador (ver Figura 15).

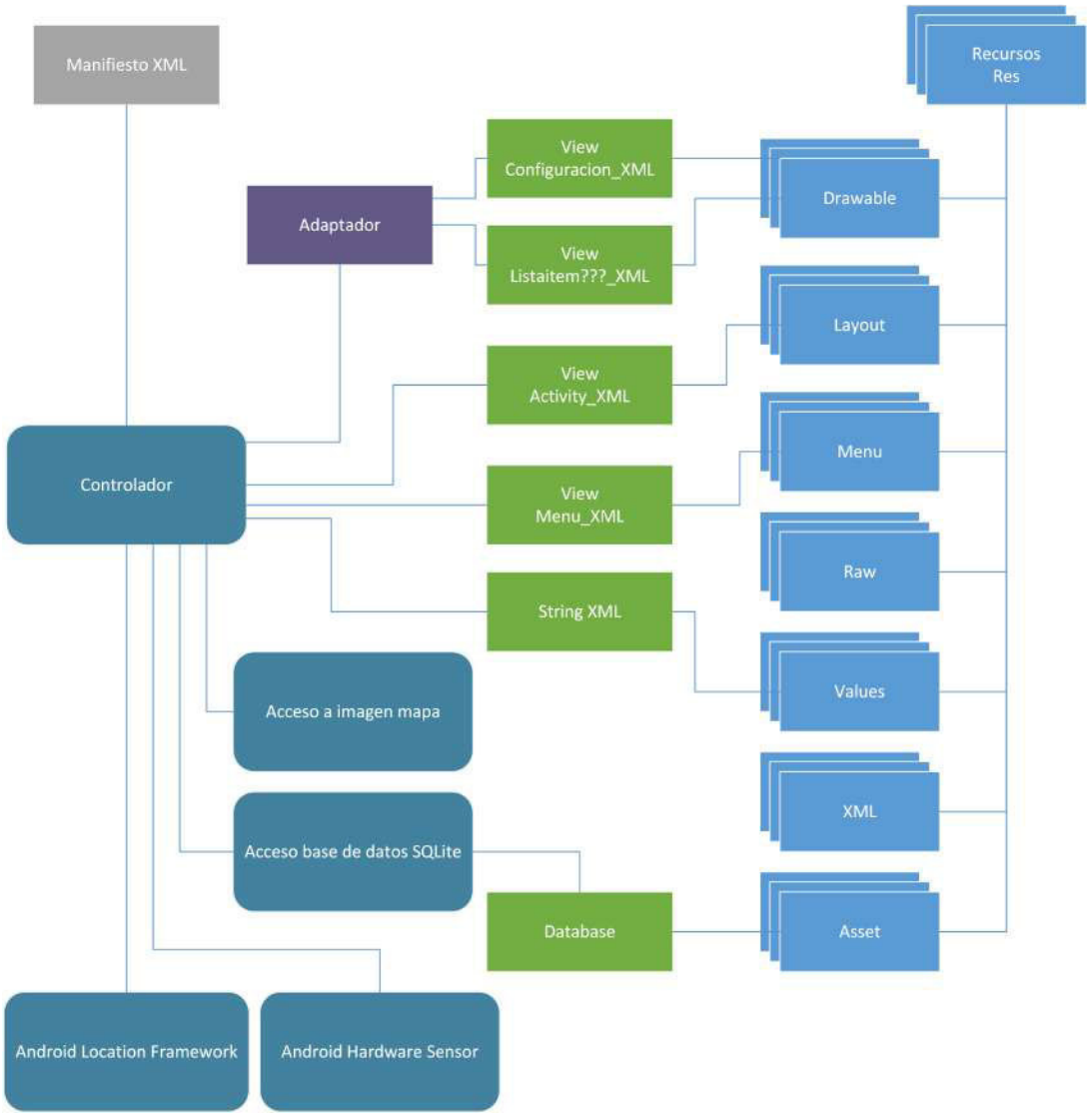

Figura 15. Patrón de diseño de la arquitectura. Fuente propia

4.5. Diseño de implementación

4.5.1. Diagrama de paquetes

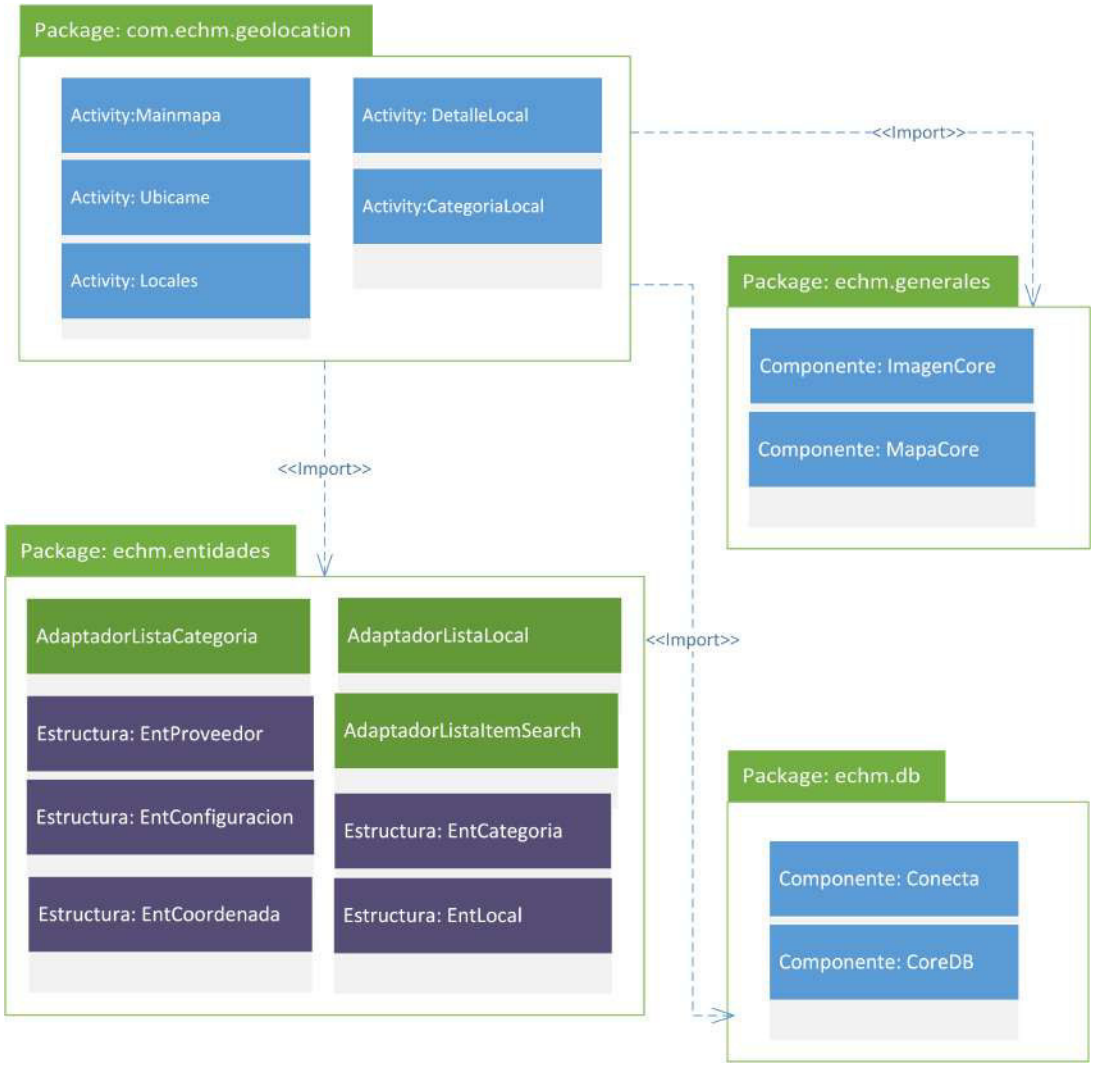

Figura 16. Diagrama general de clases. Fuente propia

4.5.2. Clase Mainmapa

1. Descripción. Vista de entrada al inicio de la carga del sistema, es la vista de arranque en la carga de los parámetros iniciales del sistema e información básica, también denominada «Splash».
2. Package: com.echm.geolocation
3. Extiende: Activity
4. Vista principal: splash.xml

4.5.3. Clase Ubícame

1. Descripción. Vista principal del sistema, muestra el plano digital con la opción de búsqueda, localización y posicionamiento en tiempo real. Diagrama de Clase (ver Figura 17).
2. Package: com.echm.geolocation
3. Implementación: Clase controlador
4. Extiende: Activity
5. Vista principal: activity_ubicame.xml

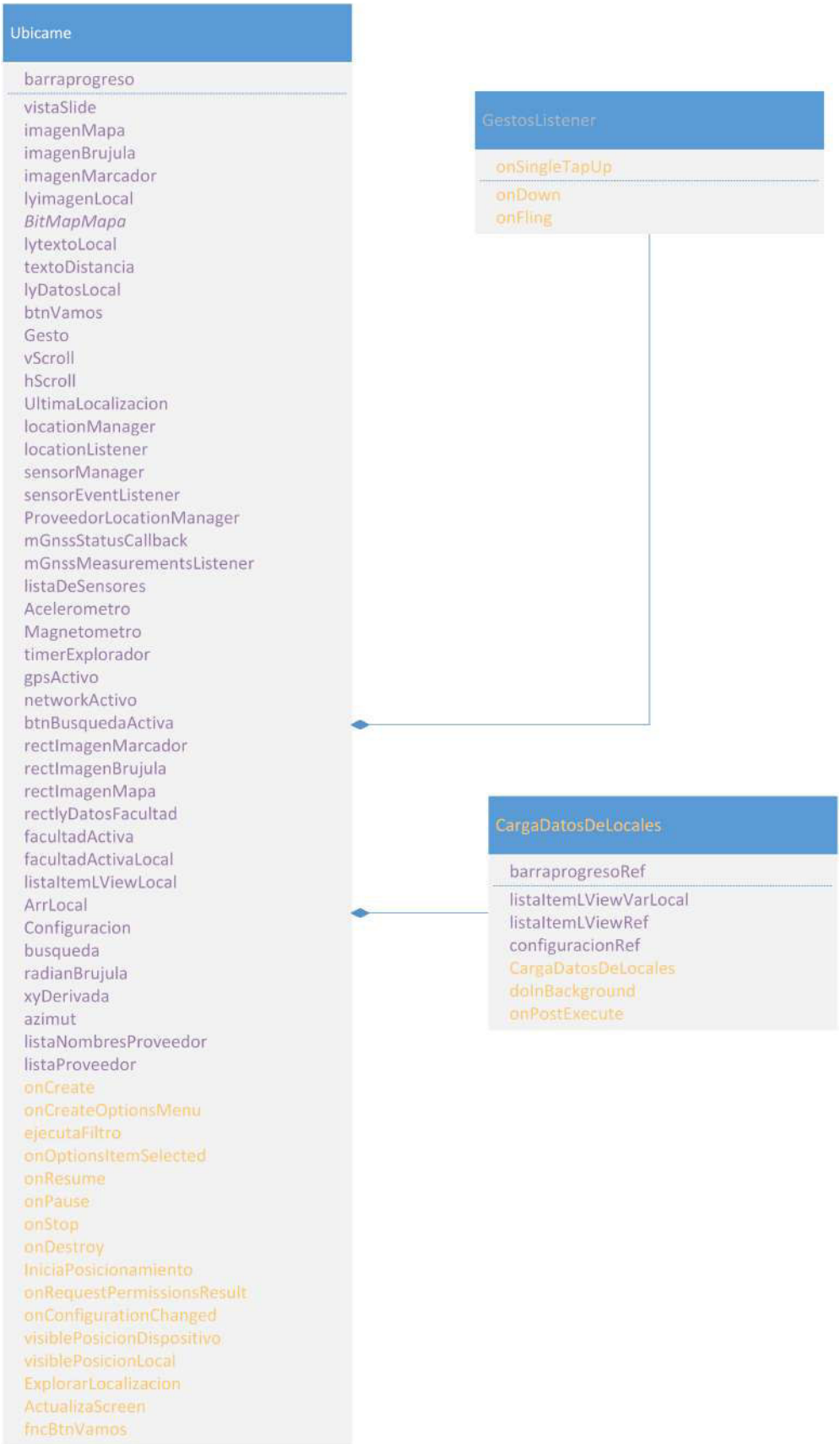

Figura 17. Diagrama de Clases de la clase Ubícame. Fuente propia.

4.5.4. Clase CategoríaLocal

1. Descripción. Visualiza la lista de categorías, cada elemento de la lista es seleccionable si existe. El evento «onCreate()» asigna el listener en el procedimiento «setOnItemClickListener()» para visualizar la lista de locales al seleccionar la Categoría. Diagrama de Clase (ver Figura 18).
2. Package: com.echm.geolocation
3. Implementación: Clase controlador
4. Extiende: Activity
5. Vista: activity_categorialocal.xml

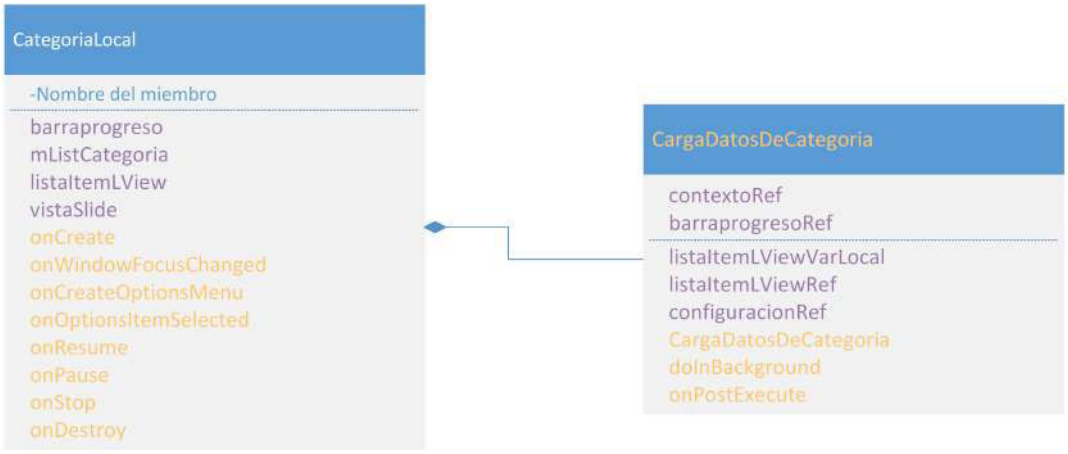

Figura 18. Diagrama de Clase de Categoría. Fuente propia.

4.5.5. Clase Locales

1. Descripción. Visualiza la lista de locales para la categoría seleccionada con el controlador «CategoriaLocal», cada elemento de la lista es seleccionable si existe. El evento «onCreate()» asigna el listener en el procedimiento «setOnItemClickListener()» para visualizar la vista con el detalle del local al seleccionar el local. Diagrama de Clase (ver Figura 19).

2. Package: com.echm.geolocation
3. Implementación: Clase controlador
4. Extiende: Activity
5. Vista: activity_locales.xml

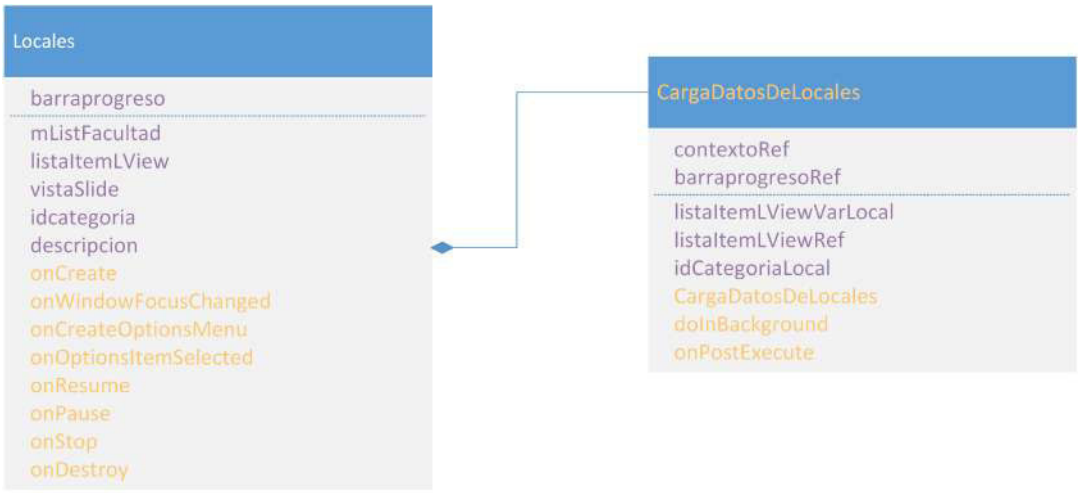

Figura 19. Diagrama de Clase de Locales. Fuente propia.

4.5.6. Clase DetalleLocal

1. Descripción. Visualiza la vista del detalle del local y permite ubicar el local de destino. Contiene las funcionalidades de localización y posicionamiento en tiempo real. Diagrama de Clase (ver Figura 20).
2. Package: com.echm.geolocation
3. Implementación: Clase controlador
4. Extiende: Activity
5. Vista: activity_detallelocal.xml

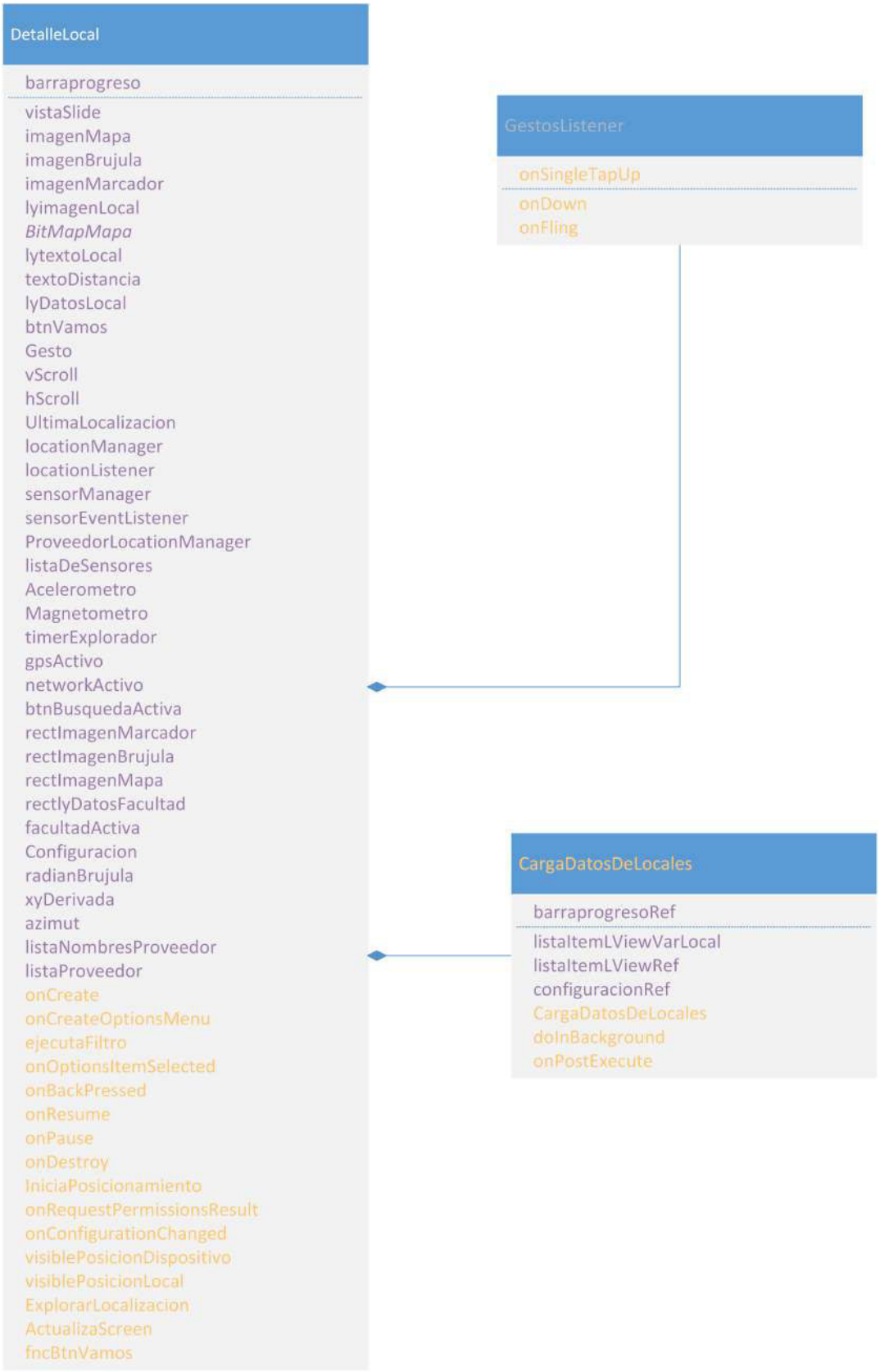

Figura 20. Diagrama de Clase de DetalleLocal. Fuente propia.

4.5.7. Clase CoreDB

1. Descripción. Clase base de configuración de la micro base de datos. Extiende la clase SQLiteOpenHelper{}. Usada para generar la instancia de la base de datos y configurar el repositorio de acceso.
2. Package: echm.db
3. Implementación: Librería
4. Procedimientos y funciones:
    a. CoreDB(Context context). Constructor
    b. crearBaseDeDatos(). Verifica la base de datos si existe llamando al procedimiento verificaBaseDeDatos(). Si no existe , llama al procedimiento copiarBaseDeDatos()
    c. verificaBaseDeDatos(). Verifica la existencia en el dispositivo de la Base de datos en el directorio: /data/data/com.echm.geolocation/databases/
    d. copiarBaseDeDatos(). Copia binaria de la base de datos desde el directorio de recursos al directorio de producción del dispositivo «/data/data/com.echm.geolocation/databases/», será usado como repositorio de acceso a los datos desde la aplicación.
    e. abrirBaseDeDatos(). Abre la base de datos para procesamiento y la asigna a la variable de conexión de base de datos.
    f. close(). Cierra la base de datos.

4.5.8. Clase Conecta

1. Descripción. Clase de procesamiento de datos de la micro base de datos.
2. Package: echm.db
3. Implementación: Librería
4. Procedimientos y funciones:
    a. Conecta(Context context). Constructor

b. cierraDB(). Cierra la conexión a la Base de datos, llama a CoreDB.close()
c. leerConfiguracion(EntConfiguracion configuracion). Lee los datos de configuración del sistema.
d. leerCategoria(). Lee los datos de las categorías de los locales.
e. leerLocal(int idlocal, EntLocal local). Lee el local del Campus Universitario identificado por el id del local.
f. leerLocales(int idcategoria). Lee los locales por el Identificador de categoría.

4.5.9. Clase ImagenCore

1. Descripción. Procesa la carga de la imagen en memoria.
2. Package: echm.generales
3. Implementación: Librería
4. Procedimientos y funciones:
   a. CargaImagenMapa(Context context) . Carga la imagen del mapa.

4.5.10. Clase MapCore

1. Descripción. Procesa la obtención de coordenadas X,Y , obtiene la dirección o rumbo entre 2 puntos, los grados desplazados cuando cambia el HEADING del dispositivo en modo LANDSCAPE o PORTRAIT.
2. Package: echm.generales
3. Implementación: Librería
4. Procedimientos y funciones:
   a. retornaCoordenadaXYConLatitudLongitudXInterseccion(double Latitud, double Longitud, EntConfiguracion Configuracion) . Retorna las coordenadas cartesianas X,Y de un punto en el plano 2D del Campus universitario. El objetivo es posicionar el punto de localización de un Local o del dispositivo

dentro de la imagen del plano del Campus universitario, por ende solo serán validas las coordenadas que se encuentren dentro del plano.

b. interseccionDeCoordenadasConDireccion(double inicialDireccion1, Location partida1, Location partida2, double inicialDireccion2). Es la intersección entre 2 puntos de partida con dirección medida en grados. Intersección de 2 vectores en el plano 2D.
c. direccionDesdeCoordenadaDeOrigen(Location origen, Location destino) . Retorna en grados la dirección o rumbo entre 2 puntos respecto al norte geográfico, norte terrestre o norte verdadero. TRUE HEADING.
d. rtnGradosDelHeadingDesplazados(int Rotacion) . Retorna el desplazamiento en grados LANDSCAPE - PORTRAIT con respecto a la rotación del dispositivo.

4.5.11. Clase AdaptadorListaCategoria

1. Descripción. Procesa la lista de categorías y los eventos de selección. Clase utilidad que se aplica como adaptador de lista de Categorías de Locales.
2. Package: echm.entidades
3. Extiende: ArrayAdapter
6. Vista: listaitemcategoria.xml

4.5.12. Clase AdaptadorListaLocal

1. Descripción. Procesa la lista de locales y los eventos de selección. Clase utilidad que se aplica como adaptador de lista de Locales.
2. Package: echm.entidades
3. Extiende: ArrayAdapter
4. Vista de adaptador: listaitemfacultad.xml

4.5.13. Clase AdaptadorListaItemSearch

1. Descripción. Procesa la lista de sugerencias del sistema a la búsqueda en la barra de búsqueda. Clase utilidad que se aplica como adaptador de lista de sugerencias en el buscador.
2. Package: echm.entidades
3. Extiende: CursorAdapter
4. Vista: listaitemsearch.xml

4.5.14. Clase EntConfiguracion

1. Descripción. Entidad configuración. Clase para definir la entidad configuracion del sistema (ver Tabla 5).
2. Package: echm.entidades
3. Implementación: Estructura de Datos.

Tabla 5. Definición de los atributos de la clase EntConfiguracion. Fuente propia.

Nombre	Tipo	Descripción
IdConfiguracion	public int	Identificador de Configuración
Nombre	public String	Nombre
PolLat1	public double	Polígono latitud límite 1 inferior izquierdo
PolLon1	public double	Polígono longitud límite 1 inferior izquierdo
PolLat2	public double	Polígono latitud límite 2 inferior derecho
PolLon2	public double	Polígono longitud límite 2 inferior derecho
PolLat3	public double	Polígono latitud límite 3 superior derecho
PolLon3	public double	Polígono longitud límite 3 superior derecho
PolLat4	public double	Polígono latitud límite 4 superior izquierdo
PolLon4	public double	Polígono longitud límite 4 superior izquierdo
PoGeoUtmRelCuadN1	public double	Posición UTM relativa a cuadrante Norte límite 1
PoGeoUtmRelCuadE1	public double	Posición UTM relativa a cuadrante Este límite 1

PoGeoUtmRelCuadN2	public double	Posición UTM relativa a cuadrante Norte límite 2
PoGeoUtmRelCuadE2	public double	Posición UTM relativa a cuadrante Este límite 2
PoGeoUtmRelCuadN3	public double	Posición UTM relativa a cuadrante Norte límite 3
PoGeoUtmRelCuadE3	public double	Posición UTM relativa a cuadrante Este límite 3
PoGeoUtmRelCuadN4	public double	Posición UTM relativa a cuadrante Norte límite 4
PoGeoUtmRelCuadE4	public double	Posición UTM relativa a cuadrante Este límite 4
PoGeoUtmN1	public double	Posición UTM Norte límite 1
PoGeoUtmE1	public double	Posición UTM Este límite 1
PoGeoUtmN2	public double	Posición UTM Norte límite 2
PoGeoUtmE2	public double	Posición UTM Este límite 2
PoGeoUtmN3	public double	Posición UTM Norte límite 3
PoGeoUtmE3	public double	Posición UTM Este límite 3
PoGeoUtmN4	public double	Posición UTM Norte límite 4
PoGeoUtmE4	public double	Posición UTM Este límite 4
ImgAnchoPixel	public double	Ancho imagen mapa
ImgAltoPixel	public double	Alto imagen mapa

4.5.15. Clase EntCategoria

1. Descripción. Entidad categoría de locales. Clase para definir la entidad categoría del Local, por ejemplo Facultad, Administración, Deporte, etc. Ver tabla (Tabla 6).
2. Package: echm.entidades
3. Implementación: Estructura de Datos.

Tabla 6. Definición de los atributos de la clase EntCategoria. Fuente propia.

Nombre	Tipo	Descripción
IdCategoria	public int	Identificador de categoría
Descripcion	public String	Descripción de la categoría

4.5.16. Clase EntCoordenada

1. Descripción. Entidad coordenadas X,Y gráficas. Clase para definir la entidad configuracion del sistema (ver Tabla 7).
2. Package: echm.entidades
3. Tipo: Estructura de Datos.

Tabla 7. Definición de los atributos de la clase EntCoordenada. Fuente propia.

Nombre	Tipo	Descripción
X	public int	Coordenada X
Y	public int	Coordenada Y

4.5.17. Clase EntLocal

1. Descripción. Entidad local. Clase para definir la entidad Local del Campus definido en el plano. La UNMSM ha definido 66 locales entre locales administrativos, facultades y otros. Ver tabla (Tabla 8).
2. Package: echm.entidades
3. Path: echm/entidades
4. Implementación: Estructura de Datos.

Tabla 8. Definición de los atributos de la clase EntLocal. Fuente propia.

Nombre	Tipo	Descripción
IdLocal	public int	Identificador de local
IdCategoria	public int	Identificador de categoría

Nombre	public String	Nombre del local
Descripcion	public String	Descripción del local
Corto	public String	Nombre corto
Telefono	public String	Teléfono
NroLocal	public int	Número de local
Cx	public float	Coordenada X. No usado
Cy	public float	Coordenada Y. No usado
imagenIcono	public Bitmap	Imagen del local. No usado
icoVisible	public boolean	Si el ícono del local es visible.
enMapaAddSubView	public boolean	Es una capa de imagen en img. mapa.
Latitud	public double	Latitud. Coordenada geográfica.
Longitud	public double	Longitud. Coordenada geográfica.
cXImagen	public int	Coordenada de la imagen
cYImagen	public int	Coordenada de la imagen

4.5.18. Clase EntProveedor

1. Descripción. Entidad proveedor de localización. Clase para definir la entidad proveedor del servicio para posicionamiento. Utiliza los proveedores proporcionados por el sistema operativo Android. Ver tabla (Tabla 9).
2. Package: echm.entidades
3. Implementación: Estructura de Datos.

Tabla 9. Definición de los atributos de la clase EntProveedor. Fuente propia.

Nombre	Tipo	Descripción
nombreProveedor	public int	Nombre del proveedor
identificadorProveedor	public int	Identificador de proveedor.
Nombre	public String	Nombre del proveedor para posicionamiento
esMejor	public boolean	Mejor proveedor

4.6. Diseño de estructura de Base de datos

4.6.1. Categoría

Definición de las categorías de los locales. Ver tabla (Tabla 10).

Tabla 10. Definición de las categorías de los locales. Fuente propia.

Nombre	Tipo	Descripción
IdCategoria	INTEGER	Identificador de Categoría. Incremental, Unique ID Key. Identificador de categoría
Descripcion	TEXT NOT NULL	Descripción. Nombre de la categoría

4.6.2. Configuración

Definición de la configuracion del sistema (ver Tabla 11).

Tabla 11. Definición de los parámetros del sistema. Fuente propia.

Nombre	Tipo	Descripción
IdConfiguracion	INTEGER	Identificador de Configuración
Nombre	TEXT NOT NULL	Nombre
PolLat1	NUMERIC NOT NULL	Polígono latitud límite 1 inferior izquierdo
PolLon1	NUMERIC NOT NULL	Polígono longitud límite 1 inferior izquierdo
PolLat2	NUMERIC NOT NULL	Polígono latitud límite 2 inferior derecho
PolLon2	NUMERIC NOT NULL	Polígono longitud límite 2 inferior derecho
PolLat3	NUMERIC NOT NULL	Polígono latitud límite 3 superior derecho
PolLon3	NUMERIC NOT NULL	Polígono longitud límite 3 superior derecho
PolLat4	NUMERIC NOT NULL	Polígono latitud límite 4 superior izquierdo

PolLon4	NUMERIC NOT NULL	Polígono longitud límite 4 superior izquierdo
PoGeoUtmRelCuadN1	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Norte límite 1
PoGeoUtmRelCuadE1	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Este límite 1
PoGeoUtmRelCuadN2	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Norte límite 2
PoGeoUtmRelCuadE2	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Este límite 2
PoGeoUtmRelCuadN3	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Norte límite 3
PoGeoUtmRelCuadE3	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Este límite 3
PoGeoUtmRelCuadN4	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Norte límite 4
PoGeoUtmRelCuadE4	NUMERIC NOT NULL	Posición UTM relativa a cuadrante Este límite 4
PoGeoUtmN1	NUMERIC NOT NULL	Posición UTM Norte límite 1
PoGeoUtmE1	NUMERIC NOT NULL	Posición UTM Este límite 1
PoGeoUtmN2	NUMERIC NOT NULL	Posición UTM Norte límite 2
PoGeoUtmE2	NUMERIC NOT NULL	Posición UTM Este límite 2
PoGeoUtmN3	NUMERIC NOT NULL	Posición UTM Norte límite 3
PoGeoUtmE3	NUMERIC NOT NULL	Posición UTM Este límite 3
PoGeoUtmN4	NUMERIC NOT NULL	Posición UTM Norte límite 4
PoGeoUtmE4	NUMERIC NOT NULL	Posición UTM Este límite 4
ImgAnchoPixel	NUMERIC NOT NULL	Ancho imagen mapa
ImgAltoPixel	NUMERIC NOT NULL	Alto imagen mapa

4.6.3. Local

Definición de los locales del mapa del Campus universitario (ver Tabla 12).

Tabla 12. Definición de los locales. Fuente propia.

Nombre	Tipo	Descripción

IdLocal	INTEGER NOT NULL	Identificador de Local. corresponde al Identificador asignado en el plano cartográfico del Campus universitario UNMSM asignada por el área de infraestructura
IdCategoria	INTEGER NOT NULL	Identificador foráneo de categoría.
Nombre	INTEGER NOT NULL DEFAULT 0	Nombre del local. Nombre asignado en la Leyenda del plano cartográfico del Campus universitario UNMSM
Descripcion	TEXT	Descripción del local
Corto	TEXT	Nombre corto del local
NroLocal	INTEGER	Numero de local según plano Campus
Latitud	NUMERIC NOT NULL DEFAULT 0	Coordenada Latitud. Coordenada geográfica en grados decimales.
Longitud	NUMERIC NOT NULL DEFAULT 0	Coordenada Longitud. Coordenada geográfica en grados decimales
Telefono	TEXT	Teléfono local (No usado)
Estado	INTEGER NOT NULL	Estado , Defecto 1 = Existe. Booleano

4.7. Diseño de variables de control del sistema

El plano cartográfico digital es una imagen en formato PNG, ha tenido un proceso de conversión de coordenadas UTM a coordenadas geográficas grados decimales, posterior al cual se ha delimitado el área con los vértices del plano cartográfico sobre el plano terrestre en la siguiente secuencia: Límite 1 inferior izquierda, Límite 2 inferior derecha, Límite 3 superior izquierda, Límite 4 superior izquierda. El plano se orienta con el Norte hacia arriba de la imagen (ver Figura 21). El proceso de conversión se detalla en los anexos 1 y 2.

Se definen las siguientes variables para evaluar el comportamiento del sistema y determinar el grado de acercamiento a las medidas adecuadas según se ha obtenido del trabajo de campo de toma de coordenadas geográficas. En la imagen

están marcados los puntos con Google Earth Profesional indicado por un PIN, cada punto marcado es una coordenada geográfica estática en grados decimales.

El sistema activara la localización cuando el dispositivo con el usuario se encuentre dentro de área indicada por el rectángulo con contorno color negro. Cuando el dispositivo se desplace el sistema mediante la brújula situada en la parte superior izquierda de la pantalla del dispositivo móvil, indicará el rumbo para llegar al destino, el punto que indica la posición actual del dispositivo que se desplazará conforme el dispositivo sea desplazado por el usuario. La diferencia medida en metros entre la coordenada geográfica estática y la coordenada geográfica calculada por el dispositivo es el margen de error del sistema para un punto indicado.

Este procedimiento sera usado para validar la precisión del sistema móvil en los puntos que conforman la trayectoria del dispositivo hacia el punto de destino. Son 3 variables a considerar:

1. Coordenada geográfica decimal estática obtenida con el Software Google Earth Pro, llamada «Coordenada geográfica estática».

2. Coordenada geográfica decimal obtenida por el sistema móvil instalado en el dispositivo móvil, llamada «coordenada geográfica calculada».

3. Distancia en metros entre los 2 puntos de coordenadas geográficas, la coordenada geográfica estática y la coordenada geográfica calculada. La diferencia entre ambas determinan la precisión en metros.

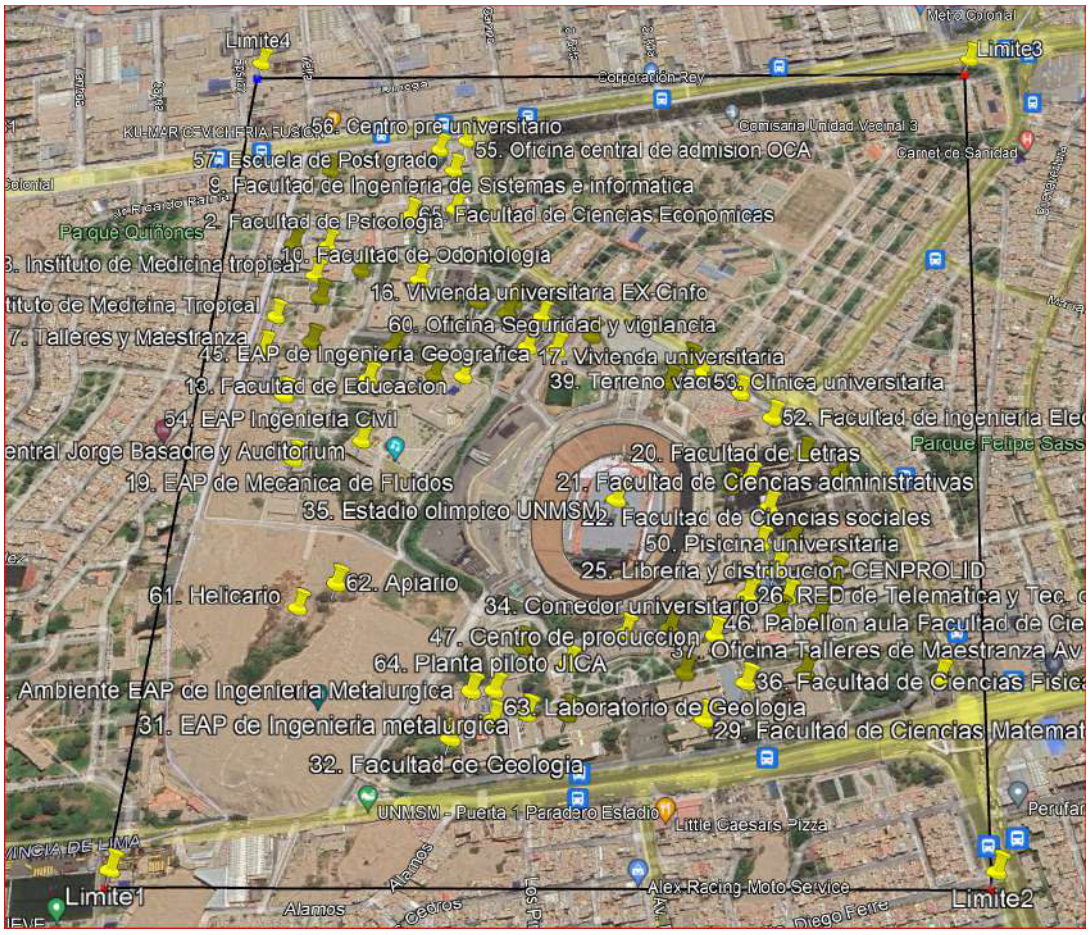

Figura 21. Área del plano con puntos límites y puntos de locales. Fuente propia.

4.8. Diseño de algoritmos

4.8.1. Control de área visible del plano digital en el dispositivo

Con el área delimitada por la unión de los vértices, estos se usan como puntos de referencia para obtener el desplazamiento del dispositivo móvil dentro del área, ademas, se usa para obtener el factor de la imagen PIXELES/METRO para posicionar en el plano de la imagen (ver Figura 22), la coordenada de (Latitud,Longitud) del local o «Punto de interés» y la coordenada de (Latitud,Longitud) del dispositivo. Considerando que la imagen del mapa está en coordenadas geográficas en grados

decimales, existe un factor requerido para posicionar cualquier punto dentro del mapa, este factor es:

factorX = ImgAnchoPixel / (Limite1 Este - Limite2 Este)

factorY = ImgAltoPixel / (Limite4 Norte - Limite1 Norte)

Aplicar el factor para obtener coordenadas X,Y gráficas:

```
xyDerivada.X = (int) (gpsTriangulacion.distanceTo(gpsLadoX) * factorX);

xyDerivada.Y = (int) (gpsTriangulacion.distanceTo(gpsLadoY) * factorY);
```

Fragmento de código :

```
Location gpsLadoX = new Location("gpsLadoX");

gpsLadoX.setLatitude(coordenadasCalculadaX.getLatitude());

gpsLadoX.setLongitude(coordenadasCalculadaX.getLongitude());

Location gpsLadoY = new Location("gpsLadoY");

gpsLadoY.setLatitude(coordenadasCalculadaY.getLatitude());

gpsLadoY.setLongitude(coordenadasCalculadaY.getLongitude());

// Es el Ancho orientación Norte coeficiente:

// Ancho imagen / Longitud metros del ancho.

double factorX = Configuracion.ImgAnchoPixel /
(Configuracion.PoGeoUtmRelCuadE1 - Configuracion.PoGeoUtmReCuadlE2);

// Es el Alto orientación Norte : Alto imagen / Longitud metros del alto.

double factorY = Configuracion.ImgAltoPixel / (Configuracion.PoGeoUtmRelCuadN4
- Configuracion.PoGeoUtmRelCuadN1);

xyDerivada.X = (int) (gpsTriangulacion.distanceTo(gpsLadoX) * factorX);

xyDerivada.Y = (int) (gpsTriangulacion.distanceTo(gpsLadoY) * factorY);
```

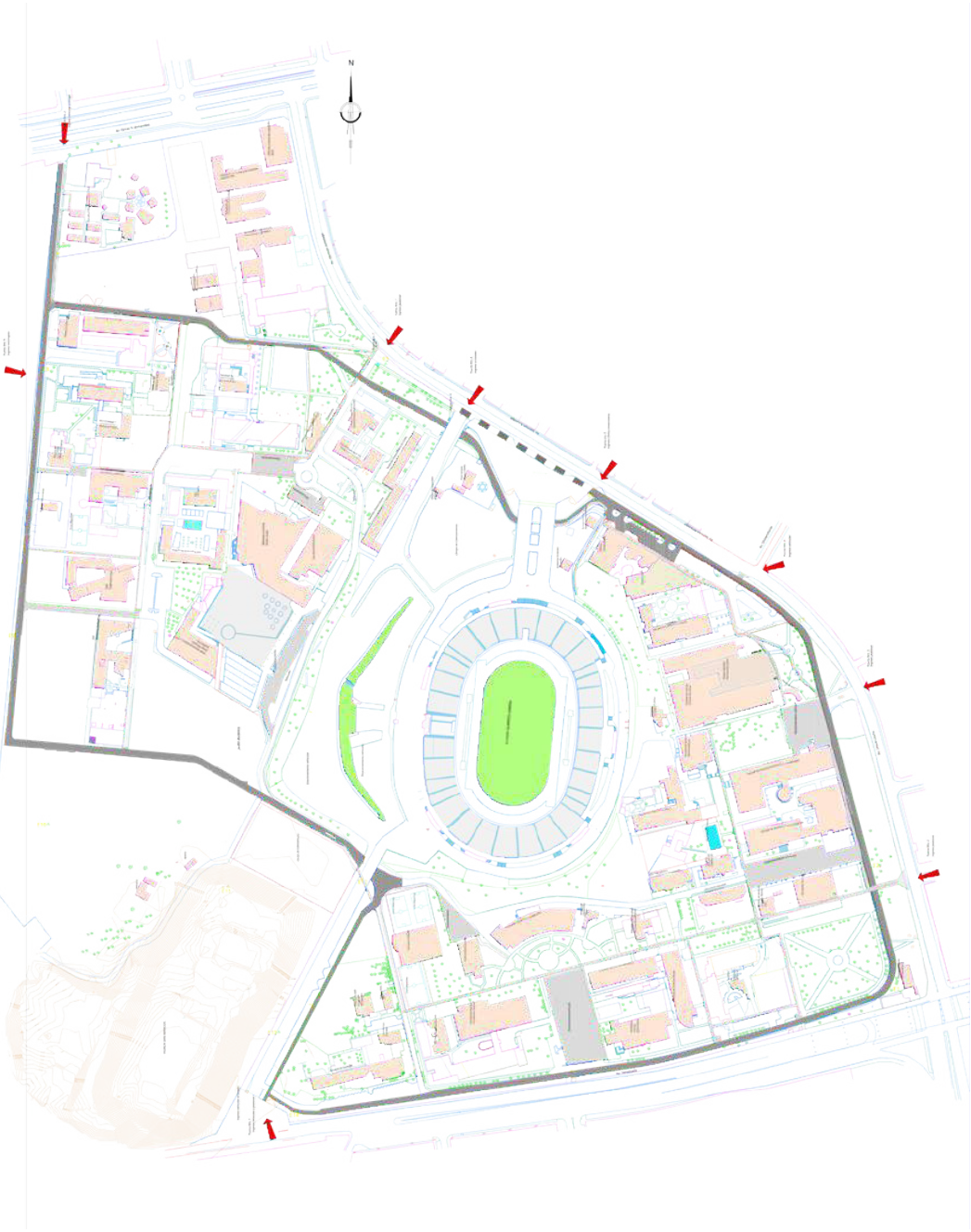

Figura 22. Plano digital del sistema «Geolocation». Fuente propia.

4.8.2. Algoritmo de conversión Coordenadas geográficas a gráficas X,Y

Algoritmo de conversión para el cálculo de la posición en coordenadas X,Y en pixeles de una coordenada geográfica en la imagen del plano. Convierte un punto de coordenadas geográficas en el plano cartográfico a coordenadas X,Y en pixeles de la imagen del plano digital. Como se muestra la imagen (ver Figura 23), indica la secuencia gráfica de como desarrolla el algoritmo el proceso de conversión. En el «diagrama A» se muestra como se encuentra en forma inicial los datos del plano con los vértices en coordenadas geográficas en grados decimales y se lee según la secuencia indicada por la numeración asignada, números del 1 al 4.

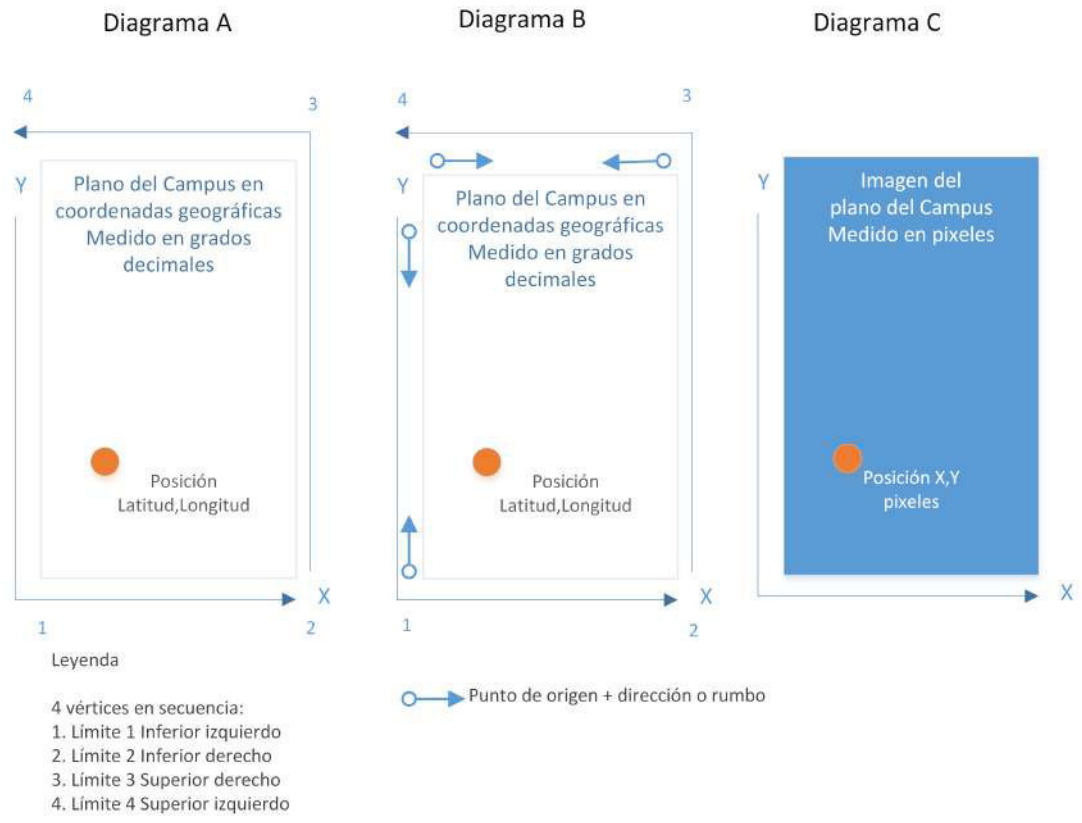

Figura 23. Diagrama de proceso de conversión. Fuente propia

El «Diagrama B» en la imagen (ver Figura 23) es el resultado intermedio que el algoritmo obtiene y se refiere a los 3 vértices 1, 4, 3 . Estos son las coordenadas geográficas con la dirección o rumbo al vértice opuesto (ver Tabla 13).

Tabla 13. Tabla de secuencias para obtener las direcciones. Fuente propia

Vértice inicio	Vértice fin	Descripción
Limite 1	Límite 4	Coordenada geográfica Limite 1 con dirección medida en grados hacia la coordenada geográfica Limite 4
Límite 4	Límite 1	Coordenada geográfica Limite 4 con dirección medida en grados hacia la coordenada geográfica Limite 1
Límite 4	Límite 3	Coordenada geográfica Limite 4 con dirección medida en grados hacia la coordenada geográfica Limite 3
Límite 3	Límite 4	Coordenada geográfica Limite 3 con dirección medida en grados hacia la coordenada geográfica Limite 4

El «Diagrama B» de la imagen (ver Figura 23) tiene como objetivo llevarlo a un procedimiento de calculo de coordenadas (X,Y) sobre un plano cartesiano pero con coordenadas geográficas y los cálculos no son lineas rectas de pares ordenados, son arcos.

En la imagen (ver Figura 24) se muestra como se obtiene la distancia en metros para cada eje en función de las coordenadas geográficas, con la distancia en metros desde el vértice Limite 4 se llega con en el siguiente paso a la obtención en pixeles de la imagen del plano.

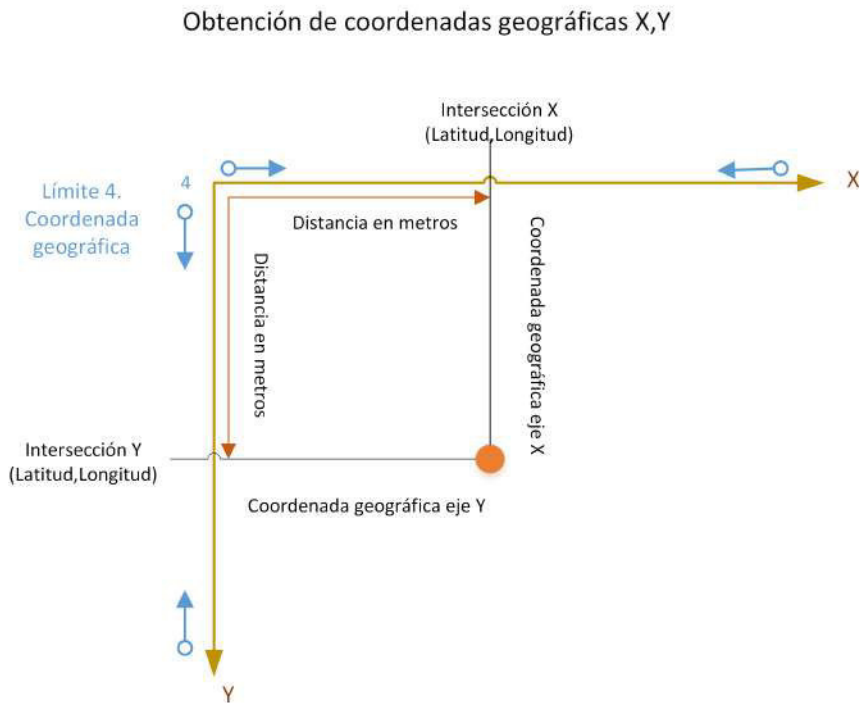

Figura 24. Diagrama de obtención de coordenadas geográficas a X,Y . Fuente propia

Obtiene las coordenadas geográficas del límite 1 inferior izquierdo, límite 2 inferior derecho, límite 3 superior derecho, límite 4 superior izquierdo que conforma el área de interés:

```
//- Limite 1
Location gpsInferiorIzquierdo = new Location("gpsInferiorIzquierdo");
gpsInferiorIzquierdo.setLatitude(Configuracion.PolLat1);
gpsInferiorIzquierdo.setLongitude(Configuracion.PolLon1);
//- Limite 2
Location gpsInferiorDerecho = new Location("gpsInferiorDerecho");
gpsInferiorDerecho.setLatitude(Configuracion.PolLat1);
gpsInferiorDerecho.setLongitude(Configuracion.PolLon2);
//- Limite 3
Location gpsSuperiorDerecho = new Location("gpsSuperiorDerecho");
gpsSuperiorDerecho.setLatitude(Configuracion.PolLat3);
gpsSuperiorDerecho.setLongitude(Configuracion.PolLon3);
//- Limite 4
Location gpsSuperiorIzquierdo = new Location("gpsSuperiorIzquierdo");
gpsSuperiorIzquierdo.setLatitude(Configuracion.PolLat4);
gpsSuperiorIzquierdo.setLongitude(Configuracion.PolLon4);

Location gpsTriangulacion = new Location("gpsTriangulacion");
gpsTriangulacion.setLatitude(Latitud);
gpsTriangulacion.setLongitude(Longitud);
```

Obtiene el factorX y Factor Y:

```
factorX = ImgAnchoPixel / (Limite1 Este - Limite2 Este)
factorY = ImgAltoPixel / (Limite4 Norte - Limite1 Norte)

Donde (Limite1 Este - Limite2 Este) igual a longitud en metros ancho
Donde (Limite4 Norte - Limite1 Norte) igual a longitud en metros alto
```

Obtiene las coordenadas gráficas X,Y:

```
xyDerivada.X = (int) (gpsTriangulacion.distanceTo(gpsLadoX) * factorX);
xyDerivada.Y = (int) (gpsTriangulacion.distanceTo(gpsLadoY) * factorY);
```

Donde :

gpsTriangulacion.distanceTo(gpsLadoX) es la distancia en metros

gpsTriangulacion.distanceTo(gpsLadoY) es la distancia en metros

En la imagen (ver Figura 25) se muestra el diagrama de actividad del algoritmo

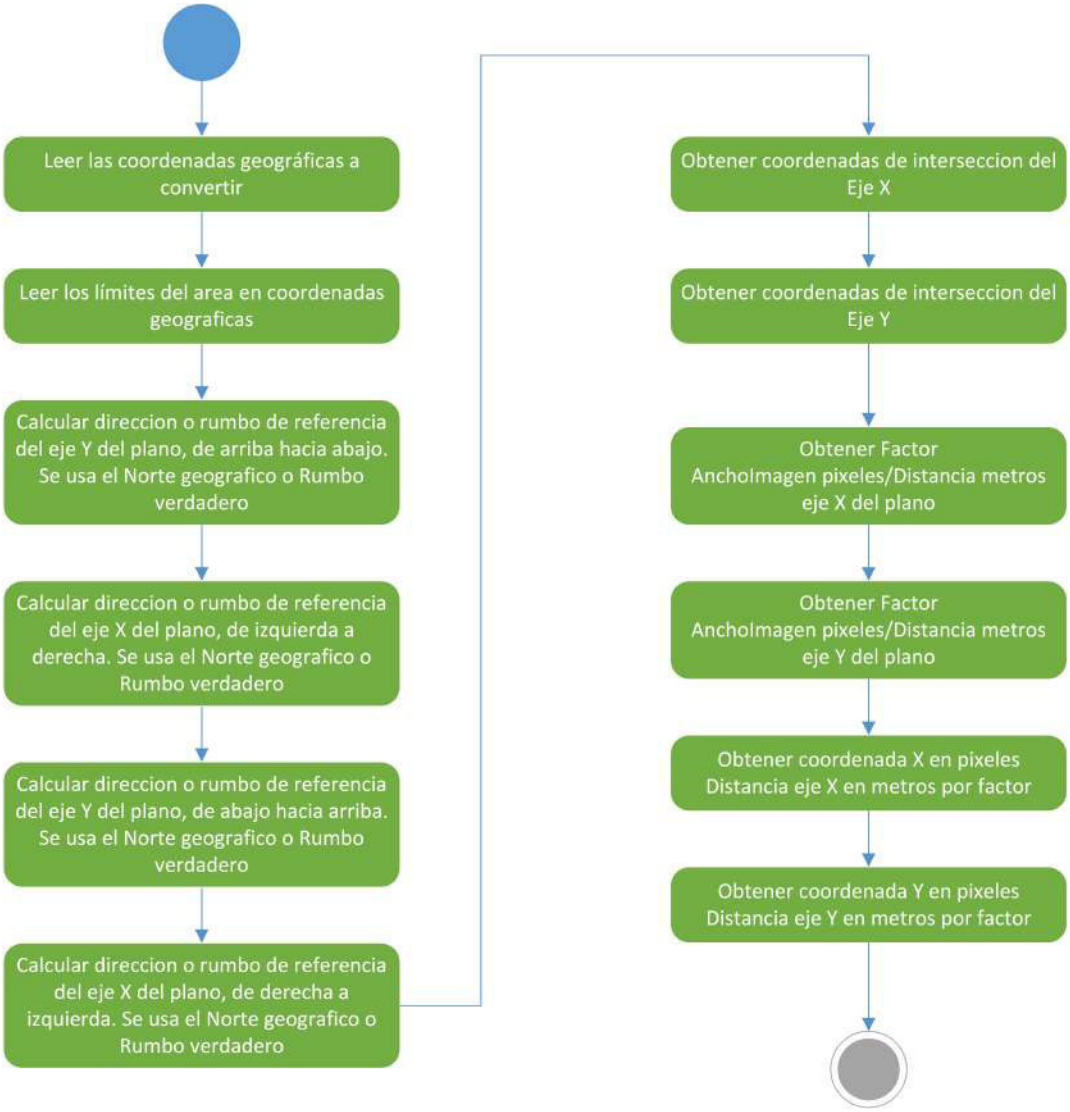

Figura 25. Diagrama de actividad del Algoritmo de posición de coordenada. Fuente propia

4.8.3. Algoritmo para el cálculo del Rumbo

Retorna en grados la dirección o rumbo (Bearing) entre 2 puntos respecto al norte geográfico o norte verdadero. TRUE HEADING. Se usa el meridiano como cero grados, hacia el lado izquierdo del meridiano es negativo , hacia el lado derecho del meridiano es positivo. El algoritmo toma las coordenadas geográficas de origen y las coordenadas geográficas de destino con la finalidad de obtener el ángulo. Se llama la función:

direccionDesdeCoordenadaDeOrigen(Location origen, Location destino)

Fragmento de código:

```
    double lat1 = Math.toRadians(origen.getLatitude());
    double lon1 = Math.toRadians(origen.getLongitude());
    double lat2 = Math.toRadians(destino.getLatitude());
    double lon2 = Math.toRadians(destino.getLongitude());
    double dLon = lon2 - lon1;
   double dPhi = Math.log(Math.tan(lat2 / 2 + (Math.PI / 4)) / Math.tan(lat1 /
2 + (Math.PI / 4)));
    if (Math.abs(dLon) > Math.PI) {
       dLon = (dLon > 0.0) ? -(2.0 * Math.PI - dLon) : (2.0 * Math.PI + dLon);
    }
   //-- Retorna el ángulo de la conversion coordenadas X,Y a coordenadas
polares
   //-- Arco tangente de y/x = Sen(ß)/Cos(ß) en el rango de -pi to pi entre el
   //-- 1er y 4to cuadrante con la funcion atan2(). Retorna el ángulo en
radianes
   double direccion = Math.atan2(dLon, dPhi);
   return Math.IEEEremainder((Math.toDegrees(direccion) + 360.0), 360.0);
```

4.8.4. Algoritmo de intersección entre 2 puntos

El objetivo es obtener el punto de intersección de 2 vectores en coordenadas geográficas. cada punto tiene una coordenada geográfica de inicio y un rumbo o grado de dirección con respecto al norte geográfico. la intersección es conseguida en coordenadas geográficas en grados decimales. Se llama la función:

interseccionDeCoordenadasConDireccion(double inicialDireccion1, Location partida1, Location partida2, double inicialDireccion2)

Fragmento de código:

```
Location interseccion = new Location("interseccion");
double lat1 = Math.toRadians(partida1.getLatitude());
double lon1 = Math.toRadians(partida1.getLongitude());
double lat2 = Math.toRadians(partida2.getLatitude());
double lon2 = Math.toRadians(partida2.getLongitude());
double brng13 = Math.toRadians(inicialDireccion1);
double brng23 = Math.toRadians(inicialDireccion2);
double dLat = lat2 - lat1;
```

```
  double dLon = lon2 - lon1;
 double dist12 = 2.0 * Math.asin(Math.sqrt(Math.sin(dLat / 2.0) * Math.sin(dLat /
2.0) + Math.cos(lat1) * Math.cos(lat2) * Math.sin(dLon / 2.0) * Math.sin(dLon /
2.0)));
   if (dist12 == 0.0) return partida1;
   double bearingA = Math.acos((Math.sin(lat2) - Math.sin(lat1) * Math.cos(dist12)) /
(Math.sin(dist12) * Math.cos(lat1)));
   if (Double.isNaN(bearingA)) bearingA = 0.0;
  double bearingB = Math.acos((Math.sin(lat1) - Math.sin(lat2) * Math.cos(dist12)) /
(Math.sin(dist12) * Math.cos(lat2)));
  double brng12, brng21;
  if (Math.sin(dLon) > 0.0) {
     brng12 = bearingA;
     brng21 = 2.0 * Math.PI - bearingB;
  } else {
     brng12 = 2.0 * Math.PI - bearingA;
     brng21 = bearingB;
  }
 double alpha1 = Math.IEEEremainder((brng13 - brng12 + Math.PI), (2.0 * Math.PI)) -
Math.PI;
        double alpha2 = Math.IEEEremainder((brng21 - brng23 + Math.PI), (2.0 *
Math.PI)) - Math.PI;
  if (Math.sin(alpha1) == 0.0 && Math.sin(alpha2) == 0.0) {
       interseccion.setLatitude(Math.toDegrees(-91.0));
       interseccion.setLongitude(Math.toDegrees(-181.0));
       return interseccion;
  }
  if (Math.sin(alpha1) * Math.sin(alpha2) < 0.0) {
       interseccion.setLatitude(Math.toDegrees(-91.0));
       interseccion.setLongitude(Math.toDegrees(-181.0));
       return interseccion;
  }

  double alpha3 = Math.acos(-Math.cos(alpha1) * Math.cos(alpha2) +
           Math.sin(alpha1) * Math.sin(alpha2) * Math.cos(dist12));
  double dist13 = Math.atan2(Math.sin(dist12) * Math.sin(alpha1) * Math.sin(alpha2),
           Math.cos(alpha2) + Math.cos(alpha1) * Math.cos(alpha3));
  double lat3 = Math.asin(Math.sin(lat1) * Math.cos(dist13) +
            Math.cos(lat1) * Math.sin(dist13) * Math.cos(brng13));
  double dLon13 = Math.atan2(Math.sin(brng13) * Math.sin(dist13) * Math.cos(lat1),
            Math.cos(dist13) - Math.sin(lat1) * Math.sin(lat3));
  double lon3 = lon1 + dLon13;
  lon3 = Math.IEEEremainder((lon3 + Math.PI), (2.0 * Math.PI)) - Math.PI;
  interseccion.setLatitude(Math.toDegrees(lat3));
  interseccion.setLongitude(Math.toDegrees(lon3));
  return interseccion;
```

4.9. Desarrollo del sistema

4.9.1 Definición de datos del sistema en repositorio persistentes

4.9.1.1. Categorías

Lista de categorías definidas por el investigador. La información se ha tabulado considerando los nombre de los campos según definición de Tabla de Base de datos (ver Tabla 14).

Tabla 14. Lista de categorías. Fuente propia

IdCategoria	Descripcion
1	Facultad
2	Administración
3	Talleres
4	Comedor
5	Vivienda
6	Biblioteca
7	Clínica y/o Consultorio
8	Seguridad
9	Laboratorio
10	Deportes
11	Otros

4.9.1.2. Locales

Coordenadas geográficas grados decimales de las áreas de interés, en la investigación presente corresponde solo las áreas físicas de las facultades e instalaciones de administrativas definidas en el plano cartográfico. Se transfirió la información a la Micro base de datos definida. La información de ha tabulado considerando los nombre de los campos según definición de Tabla de Base de datos (ver Tabla 15).

Tabla 15. Lista de Locales. Fuente propia

IdLocal	IdCategoría	Nombre	Latitud	Longitud
1	1	Ingeniería de Minas	-12.05239	-77.08727
2	1	Facultad de Psicología	-12.05363	-77.08716
4	1	Microbiología y crematorio del Instituto de Medicina Tropical	-12.05446	-77.08716
8	1	Facultad de Ingeniería Electrónica y Eléctrica	-12.05516	-77.08718
9	1	Facultad de Ingeniería de Sistemas e informática	-12.05305	-77.08557
10	1	Facultad de Odontología	-12.05418	-77.08593
11	1	E.A.P. de Educación Física	-12.05475	-77.08476
13	1	Facultad de Educación	-12.0557	-77.08524
19	1	E.A.P. Mecánica de fluidos	-12.0569	-77.0872
20	1	Facultad de Letras y Ciencias	-12.05725	-77.08166
21	1	Facultad de Ciencias Administrativas	-12.05766	-77.08142
22	1	Facultad de Ciencias Sociales	-12.05815	-77.08152
23	1	Facultad de Ciencias Económicas y Contabilidad	-12.05806	-77.08112
24	1	Facultad de Derecho y Ciencias Políticas	-12.05861	-77.08063
27	1	Laboratorio de la Facultad de Ciencias Físicas	-12.0591	-77.08128
28	1	Facultad de Ingeniería Industrial	-12.05987	-77.08099
29	1	Facultad de Ciencias Matemáticas	-12.06045	-77.08211
30	1	Facultad de Química e Ingeniería Química	-12.06038	-77.08366
31	1	E.A.P. Ingeniería Metalúrgica	-12.06037	-77.08412
32	1	Facultad de Geología	-12.06068	-77.08495
36	1	Facultad de Ciencias Físicas	-12.05999	-77.08165
43	1	Facultad de Ciencias Biológicas	-12.05986	-77.08235
44	1	E.A.P. de Geografía	-12.05526	-77.08615
45	1	E.A.P. de Ingeniería Geográfica	-12.05573	-77.08641
46	1	Pabellón aulas Facultad de Ciencias Biológicas	-12.05939	-77.08202
48	1	Ambientes de E.A.P. Ingeniería Metalúrgica	-12.06009	-77.08477
52	1	Facultad de Ingeniería Electrónica y Eléctrica	-12.05638	-77.0814
54	1	E.A.P. Ingeniería Civil	-12.05597	-77.08743

61	1	Helicario	-12.05897	-77.08688
62	1	Apiario	-12.05866	-77.08647
65	1	Pabellón nuevo Facultad de Ciencias económicas	-12.05315	-77.08614
66	1	Pabellón nuevo Facultad de Letras	-12.05689	-77.08101
3	2	Instituto de Medicina tropical	-12.05412	-77.08727
12	2	Ambientes académicos administrativos Educación	-12.05458	-77.08469
15	2	Sede administrativa central "Jorge Basadre" y Auditorium	-12.05666	-77.0864
26	2	Red Telemática y Tecnología de información	-12.05887	-77.08166
47	2	Centro de Producción	-12.05974	-77.08368
51	2	Ambiente Sindicato de Docentes	-12.05585	-77.08268
55	2	Oficina Central Admisión	-12.05189	-77.08552
56	2	Instalación de Centro Pre universitario	-12.05204	-77.08586
57	2	Instalación de escuela de Post grado	-12.0524	-77.08564
59	2	Ambiente de auto seguro UNMSM	-12.05552	-77.08244
7	3	Talleres y Maestranza	-12.05526	-77.08777
37	3	Oficina y Talleres de Maestranza	-12.05994	-77.07945
18	4	Comedor de Facultad de Odontología	-12.05456	-77.08514
34	4	Comedor Universitario y Bienestar	-12.05934	-77.08306
41	4	Comedor Facultad de Letras	-12.05735	-77.08188
16	5	Vivienda universitaria EX-Cinfo	-12.0548	-77.08432
17	5	Vivienda universitaria	-12.05527	-77.08447
14	6	Biblioteca central "Pedro Zulen"	-12.05569	-77.0856
5	7	Consultorio del Instituto de Medicina Tropical	-12.05404	-77.08668
53	7	Clínica Universitaria	-12.05568	-77.08225
58	7	Clínica de Psicología	-12.0536	-77.08761
60	8	Oficina de seguridad y vigilancia	-12.05532	-77.08411
63	9	Laboratorio de Geología	-12.06045	-77.08449
33	10	Gimnasio UNMSM	-12.05951	-77.08422
35	10	Estadio Olímpico San Marcos	-12.05749	-77.08326
50	10	Piscina Universitaria	-12.05851	-77.08133
6	11	Bioterio del Instituto de Medicina Tropical	-12.05476	-77.08768
25	11	Librería y distribución - Cenprolid	-12.05888	-77.0812
38	11	Modulo de servicios múltiples	-12.05915	-77.08083

39	11	Terreno vacío	-12.056	-77.08179
40	11	Pozo de Tanque de Agua y Caseta	-12.05512	-77.08367
42	11	Modulo de Ventas	-12.05717	-77.08058
49	11	Modulo de Comercio	-12.05928	-77.08255
64	11	Planta piloto JICA	-12.06008	-77.08451

4.9.1.3. Configuración

Vértices del plano cartográfico del Campus Universitario con la finalidad de determinar la delimitación del «área de interés» para la visualización y control de posicionamiento del dispositivo. La información se ha tabulado La información de ha tabulado considerando los nombre de los campos según definición de Tabla de Base de datos (ver Tabla 16).

Tabla 16. Lista de la configuración. Fuente propia

Nombre del Campo	Valor de campo	Significado
IdConfiguracion	1	Identificador
Nombre	U. Nacional Mayor de San Marcos	Nombre asignada a la configuración
PolLat1	-12.062220	Posición latitud LIMITE 1 inferior izquierdo
PolLon1	-77.088526	Posición longitud LIMITE 1 inferior izquierdo
PolLat2	-12.062292	Posición latitud LIMITE 2 inferior derecho
PolLon2	-77.078894	Posición longitud LIMITE 2 inferior derecho
PolLat3	-12.050543	Posición latitud LIMITE 3 superior derecho
PolLon3	-77.078803	Posición longitud LIMITE 3 superior derecho
PolLat4	-12.050471	Posición latitud LIMITE 4 superior derecho
PolLon4	-77.088435	Posición longitud LIMITE 4 superior derecho
PoGeoUtmRelCuadN1	915.4073	Coordenada UTM relativa al plano Norte LIMITE 1 inferior izquierdo
PoGeoUtmRelCuadE1	856.3343	Coordenada UTM relativa al plano Este LIMITE 1 inferior izquierdo

PoGeoUtmRelCuadN2	915.4073	Coordenada UTM relativa a cuadrante Norte LIMITE 2 inferior derecho
PoGeoUtmReCuadlE2	-192.079	Coordenada UTM relativa a cuadrante Este LIMITE 2 inferior derecho
PoGeoUtmRelCuadN3	2214.1154	Coordenada UTM relativa a cuadrante Norte LIMITE 3 superior derecho
PoGeoUtmReCuadlE3	-192.079	Coordenada UTM relativa a cuadrante Este LIMITE 3 superior derecho
PoGeoUtmRelCuadN4	2214.1154	Coordenada UTM relativa a cuadrante Norte LIMITE 4 superior izquierdo
PoGeoUtmRelCuadE4	856.3343	Coordenada UTM relativa a cuadrante Este LIMITE 4 superior izquierdo
PoGeoUtmN1	8665700	Coordenada UTM Norte LIMITE 1 inferior izquierdo
PoGeoUtmE1	272651	Coordenada UTM Este LIMITE 1 inferior izquierdo
PoGeoUtmN2	8665700	Coordenada UTM Norte LIMITE 2 inferior derecho
PoGeoUtmE2	273700	Coordenada UTM Este LIMITE 2 inferior derecho
PoGeoUtmN3	8667000	Coordenada UTM Norte LIMITE 3 superior derecho
PoGeoUtmE3	273700	Coordenada UTM Este LIMITE 3 superior derecho
PoGeoUtmN4	8667000	Coordenada UTM Norte LIMITE 4 superior izquierdo
PoGeoUtmE4	272651	Coordenada UTM Este LIMITE 4 superior izquierdo
ImgAnchoPixel	4489	Ancho en pixeles de la imagen del plano
ImgAltoPixel	5561	Alto en pixeles de la imagen del plano

4.9.2. Pruebas de rango de precisión de los sensores

Prueba de discrepancia entre coordenadas de los sensores con posición fija durante 60 segundos. Se realizan 2 pruebas, en cada prueba previamente se sacude el

dispositivo para re calibrar los sensores, considerar que la evaluación se realiza en un posición de descanso y el dispositivo tiene poco movimiento. La coordenada geográfica estática llamada «punto de referencia» se obtiene con Google Earth y la coordenada geográfica calculada con el dispositivo móvil.

1. Escenario en interior, con paredes, con techo, sin Wi-Fi. En este escenario se computa los cambios en los resultados por la exploración que realizan los sensores cada intervalo de tiempo, el rango máximo de la precisión es de 9.85 metros. Entre la primera toma y la segunda aumenta la distancia al punto de referencia, esto se debe a que el sensor toma como valor inicial la posición histórica del dispositivo provista por la función del Framework «getLastKnownLocation()», es la forma de obtener la ultima posición provista por el sensor, luego actualiza los valores porque realiza una exploración real. El escenario indica que la posición busca obtenerla de las antenas de los operadores de telefonía llamada «Network Based», en este escenario el posicionamiento por GNSS no esta disponible por la interferencia de los techos. Considerar que es posible que la antena de los operadores este localizada en un punto lejano respecto al punto geográfico de evaluación puede no ser exacto dado que la posición la recibe del operador de telefonía si esta usando GPS asistido (A-GPS) (ver Tabla 17).

Tabla 17. Tabla de resultado pruebas de sensores. Escenario 1. Fuente propia

Registro	Tiempo Segundos	Punto referencia Coordenada geográfica estática	Punto calculado Coordenada geográfica calculada	Precisión Metros
1	0 - 20	-12.054982 , -77.074469	-12.054983 , -77.074448	2.18
2	21 - 40	-12.054982 , -77.074469	-12.054933 , -77.074391	9.85
3	41 - 60	-12.054982 , -77.074469	-12.054998 , -77.074416	6.11

2. Escenario en exterior, sin paredes, sin techo, no se puede asumir Wi-Fi porque puede o no existir un punto de acceso en el rango de alcance de los sensores. En este escenario se toma una pausa de 20 segundos para estabilizar el dispositivo porque se ha tenido el dispositivo en movimiento para llegar a los nuevos puntos de

evaluación. Se realizan 2 tomas, cada punto de referencia con desplazamiento de 2 metros y se calcula la precisión. En este caso podemos asumir que la posición es estable y muy cercana al punto de referencia (ver Tabla 18).

Tabla 18. Tabla de resultado pruebas de sensores. Escenario 2. Fuente propia

Registro	Tiempo Segundos	Punto referencia Coordenada geográfica estática	Punto calculado Coordenada geográfica calculada	Precisión Metros
1	21 - 60	-12.054982 , -77.074469	-12.054981 , -77.074476	0.74
2	21 - 60	-12.054983 , -77.074422	-12.054971 , -77.074410	1.73

3. Escenario en exterior, sin paredes, sin techo, no se puede asumir Wi-Fi porque puede o no existir un punto de acceso en el rango de alcance de los sensores. Se realiza con movimiento manual del dispositivo los primeros 10 segundos, luego se mantiene en reposo hasta el segundo 41 y se vuelve a realizar un movimiento manual del dispositivo. Considerando que los sensores detectan movimiento y el proveedor de localización del sistema operativo vuelve a realizar la lectura, hay un rango máximo de 2.61 metros con respecto al punto de referencia y se vuelve mas estable en el siguiente movimiento del dispositivo a un rango máximo de 1.04 metros. En el registro 4 los puntos coinciden (ver Tabla 19).

Tabla 19. Tabla de resultado pruebas de sensores. Escenario 3. Fuente propia

Registro	Tiempo Segundos	Punto referencia Coordenada geográfica estática	Punto calculado Coordenada geográfica calculada	Precisión Metros
1	0 - 10	-12.054970,-77.074464	-12.054973,-77.074489	2.61
2	11 - 20	-12.054970,-77.074464	-12.054969,-77.074459	0.62
3	21 - 30	-12.054970,-77.074464	-12.054972,-77.074465	0.22
4	31 - 40	-12.054970,-77.074464	-12.054970,-77.074464	0.00
5	41 - 50	-12.054970,-77.074464	-12.054965,-77.074455	1.04
6	51 - 60	-12.054970,-77.074464	-12.054973,-77.074456	0.85

4.9.3. Formulación de los niveles de precisión

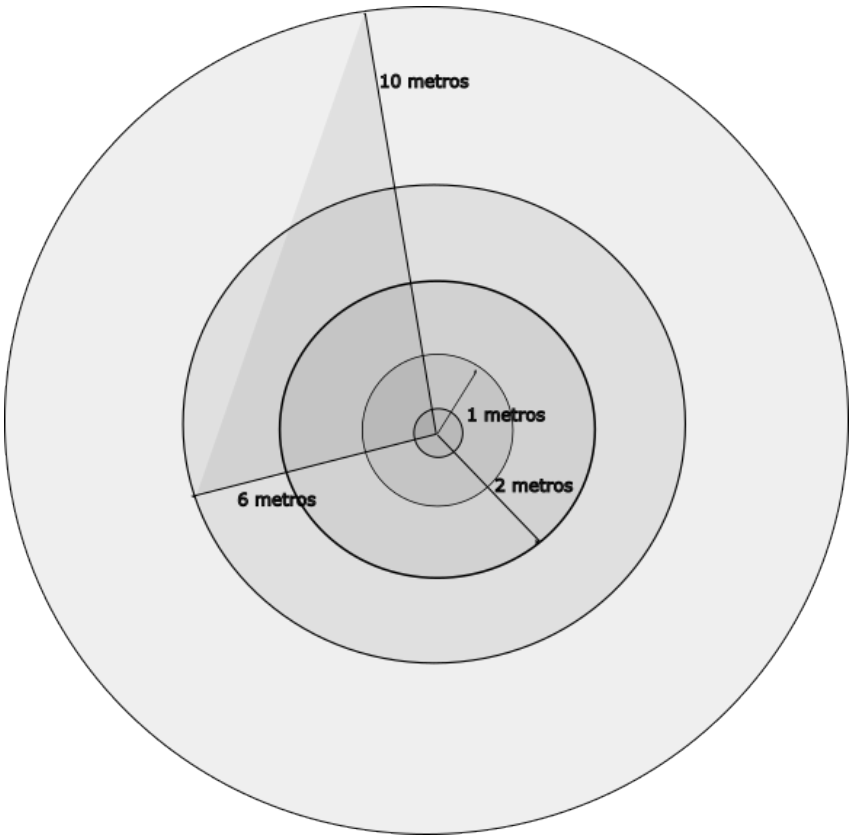

Figura 26. Diagrama de precisión en metros. Fuente propia

Considerando la pruebas del rango de precisión de los sensores en el apartado 4.9.2, la imagen (ver Figura 26) muestra rangos de precisión en metros con los distintos escenario de la coordenada geográfica estática como punto central. El sistema móvil muestra la coordenada geográfica calculada, en las pruebas se calcula el nivel de precisión en metros entre la coordenada geográfica estática (punto de referencia) y la coordenada geográfica calculada (Obtenida por el dispositivo móvil). El dispositivo esta configurado con alta precisión usando la RED del operador de telefonía «Network Based», GNSS, los puntos de acceso Wi-Fi y los puntos Bluetooth. El sistema móvil puede volver a explorar y obtener las coordenadas

geográficas calculadas cuando los sensores detectan un cambio o cuando cubre un intervalo de tiempo y puede ocasionar que los valores cambien, esto se debe que el proveedor de localización fusionada «FUSED» del sistema operativo Android tiene algoritmos internos que deciden el momento de nueva exploración, se ejecutan los listener: LocationListener() y SensorEventListener(). Si se activa el servicio GNSS se ejecuta la llamada a GnssMeasurementsEvent.Callback().

Volviendo a la imagen (ver Figura 26), se indica los niveles de aproximación, de mayor precisión al menor nivel de precisión. Se definen 4 niveles de precisión:

1. Nivel de precisión «A». Rango de 0 - 2 metros. Puede ocurrir en un entorno de área física abierta, sin columnas de concreto o edificaciones de concreto.

2. Nivel de precisión «B». Rango de 0 - 6 metros . Puede ocurrir en un entorno de área física abierta, con columnas de concreto y edificaciones.

3. Nivel de precisión «C». Rango de 0 - 10 metros. Puede ocurrir en un entorno de área física cerrada o interiores semi abiertos, edificaciones y columnas de concreto.

4. Nivel de precisión «D». Rango mayor a 10 metros. Puede ocurrir en un entorno de área física interior, con techos, edificaciones y columnas de concreto sin disponibilidad de puntos Wi-Fi o puntos Bluetooth, pero con disponibilidad de servicio de telefonía. Bajo este escenario el posicionamiento puede ser cubierta por el servicio del operador de telefonía llamado GPS asistido A-GPS.

4.9.4. Conversión del formato gráfico del plano cartográfico

Conversión del plano cartográfico del Campus universitario de la Universidad Nacional Mayor de San Marcos a formato gráfico PNG (Picture Network Group). (Ver anexo 1).

4.9.5. Conversión del sistema de coordenadas del plano

Conversión del plano cartográfico del Campus universitario de la Universidad Nacional Mayor de San Marcos de coordenadas geográficas UTM (Universal Transversa Mercator) a coordenadas geográficas en grados decimales. (Ver anexo 2).

4.9.6. Desarrollo del codigo fuente del sistema

Codigo fuente del sistema móvil «Geolocation». (Ver anexo 3).

4.10. Pruebas unitarias y pruebas integrales

Se realizan las pruebas unitarias por módulo del proyecto enfocado «De abajo hacia arriba», en cada módulo se realizan pruebas de los principales procedimientos que afectan al sistema en la instalación u originan resultados o procedimiento de localización y posicionamiento o procedimientos de acceso a la base de datos. Del dispositivo se prueban el funcionamiento de los sensores para la localización y procedimientos de calculo de la posición en tiempo real, considere que esto implica que los sensores que causan que el dispositivo vuelva obtener la localización del dispositivo como un proceso de nueva exploración de las coordenadas geográficas calculadas.

4.10.1 Módulo CoreDB

Se ejecuta y valida el proceso de carga e instalacion de la base de datos en la plataforma operativa en los procedimientos : verificaBaseDeDatos() y copiarBaseDeDatos(). Se verifica si existe, en caso contrario, copia la base de datos en el directorio de la aplicación a la instalacion del instalador APK.

4.10.2 Módulo Conecta

Se ejecuta y valida la base de datos este activa en el procedimiento Conecta(). Se ejecuta y valida el procedimiento de cierre de cursores de acceso a los datos en modo de depuración de variables. Se ejecuta y valida las funciones: leerCategoria(), leerLocal(int idlocal, EntLocal local), leerLocales(int idcategoria)

4.10.3 Módulo ImagenCore

Se valida la carga de la imagen digital del plano cartográfico que contiene el área de interés. Se verifica la carga en memoria con la ejecución del procedimiento : CargaImagenMapa(Context context)

4.10.4 Módulo MapaCore

Se ejecuta y valida los procedimientos en el entorno de llamada desde el módulo «Ubicame» y «DetalleLocal»:

1. retornaCoordenadaXYConLatitudLongitudXInterseccion(). Mide la distancia en posición de imagen en el plano.

2. interseccionDeCoordenadasConDireccion(). Obtiene la intersección entre 2 puntos dado el grado de inicio o salida.

3. direccionDesdeCoordenadaDeOrigen(). Retorna en grados la dirección o rumbo (Bearing) entre 2 puntos respecto al norte geográfico o norte verdadero. TRUE HEADING. Se usa el meridiano como cero grados, hacia el lado izquierdo del meridiano es negativo , hacia el lado derecho del meridiano es positivo.

4.10.5 Módulo Ubicame

Se ejecuta y valida el procedimiento de búsqueda en la barra de búsqueda con los procedimientos: busqueda.setOnQueryTextListener(), ejecutaFiltro(String query)

Se ejecuta y valida los procedimiento de posicionamiento y posicion del punto de interés:

1. IniciaPosicionamiento(). Inicia la gestión de posicionamiento

2. visiblePosicionDispositivo(int cXImagen, int cYImagen). Activa la posición visible del icono de lugar

3. visiblePosicionLocal(int cXImagen, int cYImagen). Activa la posicion visible del icono del local

4. ExplorarLocalizacion(). Explora la localización y la visualización en el mapa

5. ActualizaScreen(). Actualiza la pantalla del dispositivo móvil

4.10.6. Módulo CategoriaLocal

Se ejecuta y valida el procedimiento de carga de la lista de categorías: new CargaDatosDeCategoria(getApplicationContext(),barraprogreso,listaItemLView).execute().

4.10.7. Módulo Locales

Se ejecuta y valida el procedimiento de carga de locales: new CargaDatosDeLocales(this,barraprogreso,idcategoria,listaItemLView).execute()

4.10.8. Módulo DetalleLocal

Se ejecuta y valida los procedimiento de posicionamiento y posición del punto de interés:

1. IniciaPosicionamiento(). Inicia la gestión de posicionamiento

2. visiblePosicionDispositivo(int cXImagen, int cYImagen). Activa la posición visible del icono de lugar

3. visiblePosicionLocal(int cXImagen, int cYImagen). Activa la posición visible del icono del local

4. ExplorarLocalizacion(). Explora la localización y la visualización en el mapa

5. ActualizaScreen(). Actualiza la pantalla del dispositivo móvil

4.10.11. Pruebas integrales

Las pruebas integrales se ejecutan con el dispositivo móvil realizando búsquedas de locales en el área de interés y realizando búsqueda por posicionamiento en tiempo real. Se ejecutan las pruebas de usabilidad seleccionando las opciones de instalación, ejecución del sistema y realizando la navegación en las opciones del menú del sistema, estas pruebas tomaron parte en el proceso de correcciones del

sistema. Las pruebas de precisión que se denomina «Prueba de precisión del sistema móvil» se ejecuta en el punto 4.11.

4.11. Pruebas de precisión del sistema móvil

4.11.1. Precisión en el reconocimiento del área de interés

Validación de la precisión de reconocimiento del área de interés por el sistema móvil instalado en el dispositivo móvil. El sistema debe reconocer que se localiza dentro del área de interés y permite activar el proceso de posicionamiento del dispositivo en tiempo real. Se selecciona 11 puntos aleatorios y verifica que se active el proceso de localización (ver Tabla 20).

Tabla 20. Tabla de puntos de interés con selección aleatoria. Fuente propia

Registro	Latitud	Longitud	Nivel de precisión	Descripción del área
Punto 1	-12.060953	-77.085621	A	Entrada área de ingreso vehicular Av. Venezuela.
Punto 2	-12.060831	-77.084219	A	Área de Geología e ingeniería Química. Av. Venezuela.
Punto 3	-12.059752	-77.082775	A	Área frente al comedor universitario.
Punto 4	-12.058690	-77.079944	A	Área Facultad de derecho y ciencias políticas
Punto 5	-12.058354	-77.082269	A	Área Facultad de Ciencias sociales.
Punto 6	-12.055571	-77.082445	A	Área Clínica universitaria.
Punto 7	-12.055319	-77.085436	A	Área externa Biblioteca Pedro Zulen.
Punto 8	-12.054790	-77.085551	A	Área de estacionamiento 2 Biblioteca Pedro Zulen.
Punto 9	-12.055361	-77.086707	A	Área de ingeniería geográfica.
Punto 10	-12.052597	-77.085804	A	Área de Escuela de post grado
Punto 11	-12.054052	-77.085456	A	Estacionamiento de Facultad de Odontología.

En la imagen (ver Figura 27) se muestra el área de interés sobre el mapa geográfico Google Earth del Campus universitario, cada punto referenciado de la tabla (ver Tabla 20) se marca con un icono color Naranja.



Figura 27. Puntos aleatorios del área de interés sobre Google Earth. Fuente propia

La imagen (ver Figura 28) muestra los puntos aleatorios seleccionados sobre el plano digital del sistema móvil, como se observa los puntos geográficos calculados por el sistema móvil tienen alto nivel de precisión sobre el plano 2D X,Y. Las localizaciones de los puntos geográficos validan el sistema con un nivel de precisión «A». Cada punto se marca con un PIN color celeste.

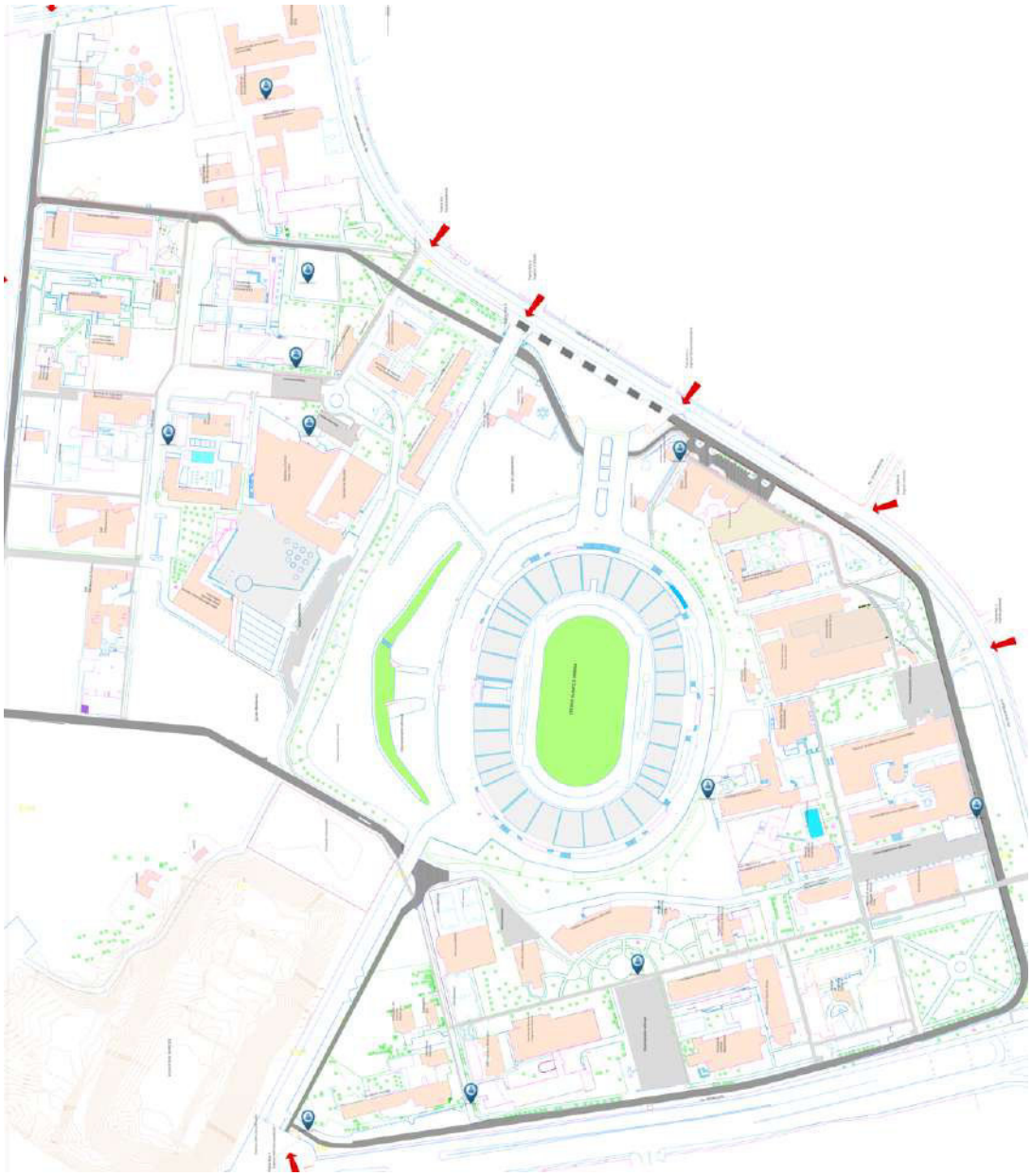

Figura 28. Puntos aleatorios del área de interés sobre plano sistema. Fuente propia

4.11.2. Precisión de la trayectoria seleccionada

Se ejecuta 3 pruebas de validación de la precisión de la trayectoria porque permite al sistema orientar la trayectoria con la brújula hacia el punto de interés. La prueba consiste en seleccionar 3 puntos de interés y seguir la ruta definida desde un punto de origen definido para cada ruta. En cada ruta se seleccionan 10 puntos o coordenadas geográficas calculada como muestra a validar.

4.11.2.1. Partida: Estacionamiento comedor universitario. Destino: Facultad de derecho y ciencias políticas

Tabla 21. Tabla de ruta 1. Fuente propia

Registro	Nivel de precisión	Latitud	Longitud
Ruta 1.1	A	-12.059835	-77.082888
Ruta 1.2	A	-12.060186	-77.082584
Ruta 1.3	A	-12.060692	-77.082497
Ruta 1.4	A	-12.060592	-77.081865
Ruta 1.5	A	-12.060446	-77.081142
Ruta 1.6	A	-12.060170	-77.080256
Ruta 1.7	A	-12.059631	-77.079654
Ruta 1.8	A	-12.059489	-77.079700
Ruta 1.9	A	-12.059013	-77.079828
Ruta 1.10	A	-12.058753	-77.080479

La imagen (ver Figura 29) se muestra el área de interés sobre el mapa geográfico Google Earth del Campus universitario, cada punto referenciado de la tabla (ver Tabla 21) se marca con un PIN color Verde, cada punto enlazado con linea color negro para indicar la trayectoria.

La imagen (ver Figura 30) muestra la ruta sobre el plano digital del sistema móvil, como se observa los puntos geográficos calculados por el posicionamiento del sistema móvil tiene alto nivel de precisión sobre el plano 2D X,Y. Cada punto se marca con un PIN color azul, cada PIN enlazado con linea color rojo para indicar la trayectoria.

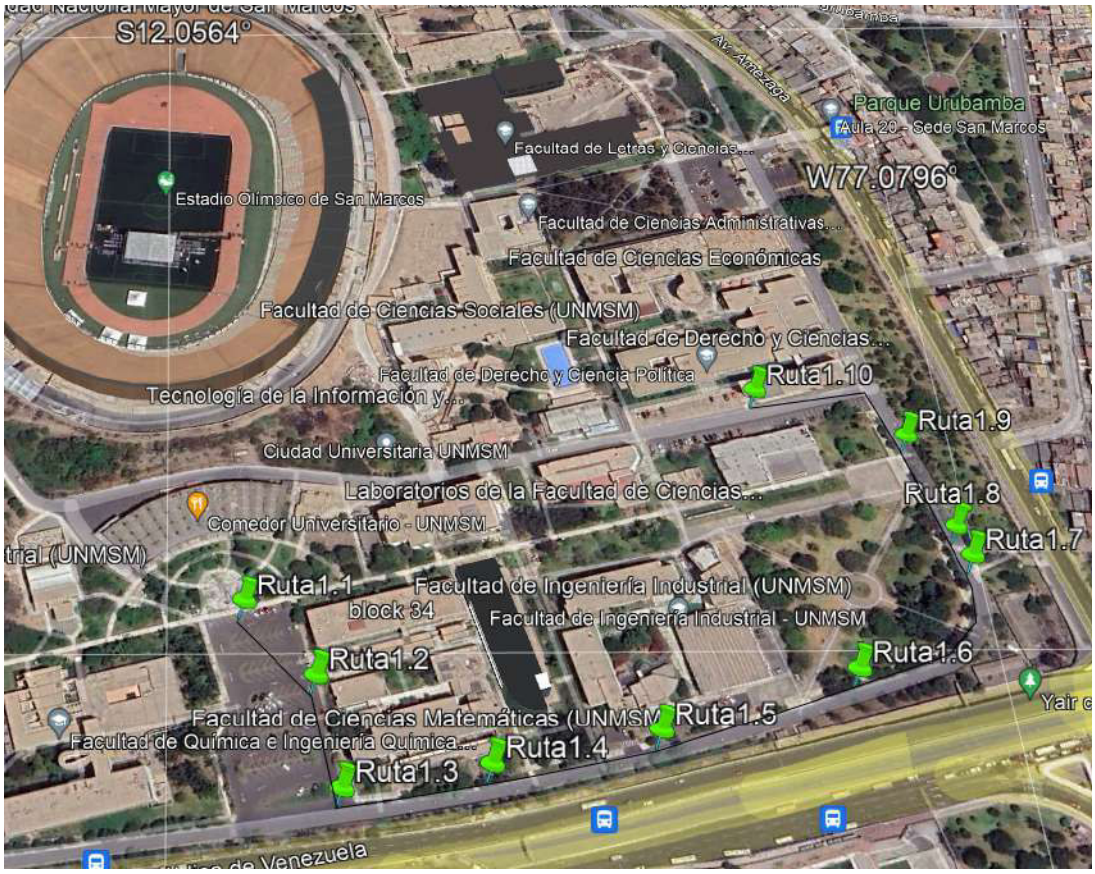

Figura 29. Validación de trayectoria sobre Google Earth. Ruta 1. Fuente propia



Figura 30. Validación de trayectoria sobre plano del sistema. Ruta 1. Fuente propia

4.11.2.2. Partida: Facultad de Ciencias económicas. Destino: Comedor universitario

Tabla 22. Tabla de ruta 2. Fuente propia

Registro	Nivel de precisión	Latitud	Longitud
Ruta 2.1	B	-12.058155	-77.081208
Ruta 2.2	B	-12.058273	-77.081161
Ruta 2.3	A	-12.058538	-77.081111
Ruta 2.4	A	-12.058827	-77.081049
Ruta 2.5	A	-12.058985	-77.081005
Ruta 2.6	A	-12.059128	-77.080967
Ruta 2.7	A	-12.059296	-77.080903
Ruta 2.8	A	-12.059367	-77.081227
Ruta 2.9	A	-12.059445	-77.081532
Ruta 2.10	A	-12.059552	-77.081945
Ruta 2.11	A	-12.059630	-77.082273
Ruta 2.12	A	-12.059434	-77.082934

La imagen (ver Figura 31) se muestra el área de interés sobre el mapa geográfico Google Earth del Campus universitario, cada punto referenciado de la tabla (ver Tabla 22) se marca con un PIN color Rojo, cada punto enlazado con una linea color negro para indicar la trayectoria.

La imagen (ver Figura 32) muestra la ruta sobre el plano digital del sistema móvil, como se observa los puntos geográficos calculados por el posicionamiento del sistema móvil tiene alto nivel de precisión sobre el plano 2D X,Y. Cada punto se marca con un PIN color azul, cada PIN enlazado con linea color rojo para indicar la trayectoria.



Figura 31. Validación de trayectoria sobre Google Earth. Ruta 2. Fuente propia



Figura 32. Validación de trayectoria sobre plano del sistema. Ruta 2. Fuente propia

4.11.2.3. Partida: Estacionamiento odontología . Destino: Post grado

Tabla 23. Tabla de ruta 3. Fuente propia

Registro	Nivel de precisión	Latitud	Longitud
Ruta 3.1	A	-12.054150	-77.085437
Ruta 3.2	A	-12.053948	-77.085452
Ruta 3.3	A	-12.053788	-77.085478
Ruta 3.4	A	-12.053746	-77.085609
Ruta 3.5	A	-12.053682	-77.085755
Ruta 3.6	B	-12.053528	-77.085883
Ruta 3.7	A	-12.053289	-77.085915
Ruta 3.8	A	-12.053002	-77.085947
Ruta 3.9	B	-12.052756	-77.085996
Ruta 3.10	B	-12.052579	-77.086015

La imagen (ver Figura 33) se muestra el área de interés sobre el mapa geográfico Google Earth del Campus universitario, cada punto referenciado de la tabla (ver Tabla 23) se marca con un PIN color Marrón, cada PIN enlazado con una linea color negro para indicar la trayectoria.

La imagen (ver Figura 34) muestra la ruta sobre el plano digital del sistema móvil, como se observa los puntos geográficos calculados por el posicionamiento del sistema móvil tiene alto nivel de precisión sobre el plano 2D X,Y en pixeles. Cada punto se marca con un PIN color azul, cada PIN enlazado por una linea color rojo para indicar la trayectoria.

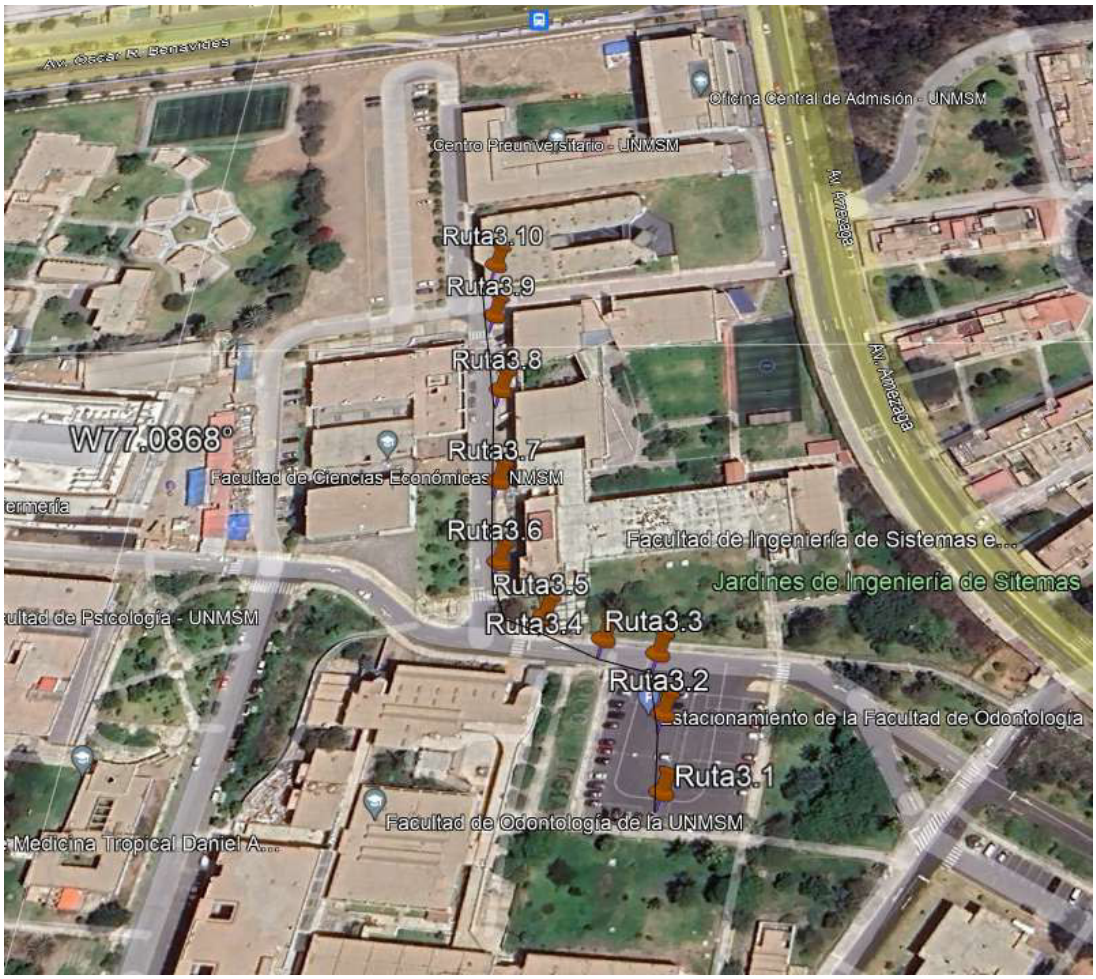

Figura 33. Validación de trayectoria sobre Google Earth. Ruta 3. Fuente propia



Figura 34. Validación de trayectoria sobre plano del sistema. Ruta 3. Fuente propia

CAPITULO V. RESULTADOS Y DISCUSIÓN

Se realizó el trabajo de validación de localización, posicionamiento, calidad de los sensores, capacidad de los proveedores de localización y criterio de selección de la plataforma operativa para proporcionar la información de las localización geográfica.

1. Rango de precisión de los sensores. El rango de precisión depende de varios factores que se enumeran: La disponibilidad de las antenas de los operadores de telefonía «Network Based», del alcance del sensor de radio frecuencia Wi-Fi, de los sensores inerciales que puedan detectar cambios en el estado del dispositivo, del sensor del servicio GNSS que detecte un cambio de posición, de la infraestructura de la zona geográfica que indique la presencia de edificaciones que bloqueen la señal del sensor, de otros dispositivos que interfieran con la señal del sensor, de reflejos que causen rebotes de la señal de radio frecuencia.

2. En las pruebas para determinar los niveles de precisión de los sensores, ver sección 4.9.2. «Pruebas de rango de precisión de los sensores», aplica escenarios que puedan afectar la precisión de las ubicaciones definidas, el algoritmo aplicado por el Framework de localización retorna en una primera instancia la ultima posición conocida provista por la función «getLastKnownLocation()», existe una fase de ajuste de las coordenadas hasta estabilizar la posición que se mantiene estable en zonas abiertas con un alto nivel de precisión. En el caso de zonas cerradas o con edificaciones los requerimientos de posición son mas frecuentes y la posición toma mas tiempo estabilizarse con un rango mayor en metros respecto al punto de referencia, pasado un tiempo puede ocurrir que la distancia sea mayor respecto al punto de referencia.

3. En las pruebas de detección del área de interés, ver sección 4.11.1. «Precisión en el reconocimiento del área de interés», que son zona abierta, el sistema detecta con nivel de precisión «A» la ubicación del área de interés. Como se menciona, el área es un cuadrante que no solo involucra al área del Campus, también a un área cercana y externa al Campus de la Universitario.

4. En las pruebas de posicionamiento del sistema móvil, ver 4.11.2. «Precisión de la trayectoria seleccionada», el sistema retorna un nivel de precisión «A», en casos de zonas donde existen edificaciones muy cercanas a la ruta de las pruebas como es el caso de la ruta 2, se presenta un nivel «B» de precisión y ocurre en los 2 punto iniciales de la ruta 2, se puede deber a la presencia de edificaciones muy cercanas que forman un pasaje. Igual es el caso de la ruta 3 con 2 puntos finales con nivel de precisión «B».

5. Localización. En los puntos de localización realizados en las pruebas búsqueda y visualización en el plano digital de locales del Campus universitario, coincide con la localización de obtenida por el software de «Google Earth Pro» en zonas abiertas. Esto indica que los cálculos para localizar con las coordenadas geográficas sobre el plano digital del sistema, tiene una alta aproximación respecto al software «Google Earth Pro». Es importante no solo obtener un nivel de precisión alto de las coordenadas, también el algoritmo de posicionamiento sobre el plano es muy preciso en la visualización. Ver 4.11.2. «Precisión de la trayectoria seleccionada».

6. Posicionamiento. El seguimiento del cambio de posición del dispositivo se ejecuta en intervalos de 300 mili segundos programado en un «TIMER», esto favorece en el ahorro de la batería del celular. Se configura con máxima precisión «Criteria.ACCURACY_FINE» que indica por defecto ACCURACY = HIGH, solo considera asegurara máxima precisión. Las edificaciones afecta la precisión, esto involucra desplazarse dentro de la edificación que dispone de techos o columnas de acero y concreto que obstruyen parcialmente la señal de radio frecuencia, pero se recupera pasado el obstáculo. Se ha procedido ha dejar la

selección del proveedor al sistema operativo llamando a la función «getBestProvider()». Para efectuar el acceso al proceso de selección del proveedor, se accede previamente con una solicitud de permisos ACCESS_FINE_LOCATION y ACCESS_COARSE_LOCATION en el manifiesto de la aplicación.

7. Calidad de los sensores. se probó el sistema en los dispositivos SAMSUNG A50 y SAMSUNG S21+. La precisión en el SAMSUNG S21+ es mayor dentro de los locales porqué recupera la señal GPS mas rápido que el SAMSUNG A50. La calidad de los sensores puede ser determinante en la precisión , así como también la capacidad de la Batería. Devuelve el nombre del proveedor que mejor cumple con los criterios dados. Android solo devolverá los proveedores a los que se permita el acceso mediante la configuración de la función Criteria(). Si varios proveedores cumplen los criterios, se devuelve el de mayor precisión. Si ningún proveedor cumple con los criterios se aplica la siguiente secuencia de abandono de requisitos: Energía, precisión, rumbo o dirección, desplazamiento, altitud.

8. Proveedor de localización fusionada «FUSED». Es la recomendación de Google para su plataforma Android, indica que cuenta con opciones de Fingerprint, cache de posicionamiento, triangulación con sensores GPS, Wi-Fi, Bluetooth, Bluetooth LE y sensores inerciales, para tener mejor aproximación dentro de los locales. Es el proveedor que tiene por defecto en el aseguramiento de la mejor precisión.

9. El mejor criterio. Si bien el proveedor de localización fusionada «FUSED» es considerado el mejor proveedor, se deja que el sistema operativo tome las decisiones según el criterio requerido de mayor precisión «Criteria.ACCURACY_FINE» para retornar el mejor proveedor «getBestProvider()», considerar que la lista de proveedores que retorna la función son: «Passive», «Network», «FUSED», «GPS». , hay que tomar en cuenta que una buena aplicación debe considerar muchos factores que no

necesariamente tienen que estar directamente ligado a la aplicación, por ejemplo , el recurso de batería, el conflicto con otras aplicaciones que están activas en segundo plano, sensores que dependiendo del dispositivo pueden consumir mayor nivel de energía.

CAPITULO VI. IMPACTOS

Con las mejoras en los dispositivos de celulares, nos referimos a los sensores y mayor capacidad de la batería, se puede implementar una solución totalmente integrada en el dispositivo. Los modernos dispositivos GPS, Wi-Fi, Bluetooth y la mayor capacidad de las antenas de los operadores permite el desarrollo de un sistema de bajo costo sin requerir infraestructura adicional , tal como sensores en los recintos con protocolo Bluetooth de bajo consumo «Low Energy».

Los dispositivos que emiten señal de radio frecuencia permiten poder informar al dispositivo ciertas características únicas de posición e identificación para registrar el dispositivo tal como un Enrutador Wi-Fi o un punto de acceso Wi-Fi con una huella digital (Fingerprint), esto permite al dispositivo poder posicionar en áreas que no cuenten con señal GPS o no cuenten con señal de radio de antenas de operadores de telefonía.

Actualmente existe una mejora continua en la calidad de los sensores de radio frecuencia, sensores inerciales y capacidad de la batería que permiten poder desarrollar sistemas móviles de bajo costo.

En la presente investigación se desarrollo adicionalmente, la capacidad de ofrecer información desde el dispositivo al usuario, tal como datos del local de forma simplificada, pero puede ser escalado a mayor volumen de información relevante para el usuario del aplicativo móvil.

CONCLUSIONES

1. Los dispositivos móviles cuentan con las mismas capacidades de otros dispositivos de geolocalizar en interiores. Los dispositivos de geo localización están condicionados por su entorno, en zonas abiertas el nivel de precisión es alto, en zonas cerradas siempre se presenta el posible bloqueo de señal de radio frecuencia, excepto que existan dispositivos de RED y puntos Bluetooth que pueden ser usados para obtener la posición.

2. Se obtiene la posición del dispositivo en tiempo real, las pruebas del sistema muestran un alto nivel de precisión en las pruebas de campo. Implementa en el posicionamiento sobre el mapa digital del Campus universitario, usa el procedimiento de conversión de coordenadas geográficas en grados decimales a coordenadas X,Y en el plano 2D de mapa de bits. El proceso de conversión es eficiente y no causa demora de procesamiento.

3. Con el procedimiento de conversión de coordenadas geográficas en grados decimales a coordenadas X,Y en el plano digital 2D, se puede localizar un punto usando coordenadas geográficas.

4. Siempre se puede presentar perdida eventual de la intensidad de la señal, esto no invalida la solución, el proceso de lectura de la posición en un intervalos de tiempo reajusta los valores resultantes y se ajustan a nivel de mayor precisión.

5. Nunca habrá un punto de equilibrio entre la precisión y el ahorro de energía, los algoritmos del sistema Android controla la exploración para maximizar el ahorro de batería, inclusive en se ha configurado con máxima precisión como es el caso del proveedor de localización fusionada «FUSED LOCATION PROVIDER».

RECOMENDACIONES

1. Ampliar el desarrollo del sistema móvil para proporcionar otros servicios de información complementaria de los locales o facultades. Estos pueden ser : Horarios de atención, información de actividades administrativas dentro del local como el tipo de servicio y horarios de atención, información a publicar por la Facultad tal como avisos a los estudiantes y eventos a realizar.

2. El sistema se puede personalizar para el usuario con información relevante de su Facultad. En este caso la Universidad puede participar proporcionando información pública del estudiante para que solo llegue a él la información de los cursos inscritos e información de horario de prácticas, clases y exámenes.

REFERENCIAS BIBLIOGRÁFICAS

Bhatta, B. (2021). Global Navigation Satellite Systems New Technologies and Applications (2nd ed.). Boca Ratón, Florida, EE.UU.: CRC Press, Taylor & Francis Group.

Cao. J. , Yang, Y. (2022). Wireless Sensing. Principles, Techniques and Applications. Cham, Suiza: Springer Nature Switzerland AG.

Dey, I. (2022). Propagation Modeling for Wireless Communications. Boca Ratón, Florida, EE.UU.: CRC Press.

García, A. , Maier, S. , Phillips, A. (2020). Location-Based Services in Cellular Networks from GSM to 5G NR. Boston, Londres, Inglaterra: Artech House.

Gentile, C. , Alsindi, N. , Raulefs, R. , Teolis, C. (2013). Geolocation Techniques. Principles and Applications. Heidelberg, Alemania: Springer Science + Business Media.

Groves, P. (2013).Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems (2nd ed.). Boston, Londres, Inglaterra: Artech House.

Jondhale, S. , Maheswar, R. , Lloret, J. (2022). Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks. Cham, Suiza: EAI/Springer Innovations in Communication and Computing, Springer Nature Switzerland AG.

Khamis, A. (2021). Smart mobility. Exploring foundational technologies and wider impacts. Courtice, ON, Canadá: APress Media.

Morton, Y , Van Diggelen, F. , Spilker, J. , Parkinson, B. (2021). Position, Navigation, and Timing Technologies in the 21st Century. Integrated Satellite Navigation, Sensor Systems, and Civil Applications (Volumen 1). Piscataway, Nueva Jersey, EE.UU.: IEEE Press, John Wiley & Sons, Inc.

Morton, Y , Van Diggelen, F. , Spilker, J. , Parkinson, B. (2021). Position, Navigation, and Timing Technologies in the 21st Century. Integrated Satellite Navigation, Sensor Systems, and Civil Applications (Vol. 2). Piscataway, Nueva Jersey, EE.UU.: IEEE Press, John Wiley & Sons, Inc.

Pahlavan, K. (2019). Indoor Geolocation. Science and Technology At the Emergence of Smart World and IoT. Gistrup, Dinamarca: River Publishers.

Panigrahi, N. (2014). Computing in Geographic Information Systems. Boca Ratón, Florida, EE.UU.: CRC Press, Taylor & Francis Group.

Samama, N. (2019). Indoor Positioning Technologies and performance. Piscataway, Nueva Jersey, USA: IEEE Press, John Wiley & Sons, Inc.

Sharp, I. , Yu, K. (2019). Wireless Positioning: Principles and Practice. Gateway East, Singapore: Springer Nature Singapore Pte Ltd.

Tian, X. , Tong, X. , Wang, X. (2023). Wireless Localization Techniques. Cham, Suiza: Springer Nature Switzerland AG.

Van Sickle, J. (2017). Basic GIS Coordinates (3ra. ed.). Boca Ratón, Florida, USA: CRC Press, Taylor & Francis Group.

Wagh, S. , Bhende, M., Thakare, A. (2023). Energy optimization protocol design for sensor networks in IOT domains. Boca Ratón, Florida, EE.UU.: CRC Press.

Yu, K. (2018). Positioning and Navigation in Complex Environments. Hershey, Pensilvania, USA: IGI Global.

Zekavat, R. , Buehrer, R. M. (2019). Handbook of position location theory, practice, and advances (2nd ed.). Piscataway, Nueva Jersey, USA: IEEE Press Editorial Board.

ANEXOS

1. Conversión plano cartográfico formato DWG a formato PNG

1.1. Antecedentes

El plano cartográfico del Campus universitario fue proporcionado por el área de infraestructura de la U.N.M.S.M. en formato DWG propietario generado por la herramienta AUTOCAD. El plano se ha diseñado en modo horizontal (LANDSCAPE) con el Norte orientado a la izquierda (ver Figura 35).

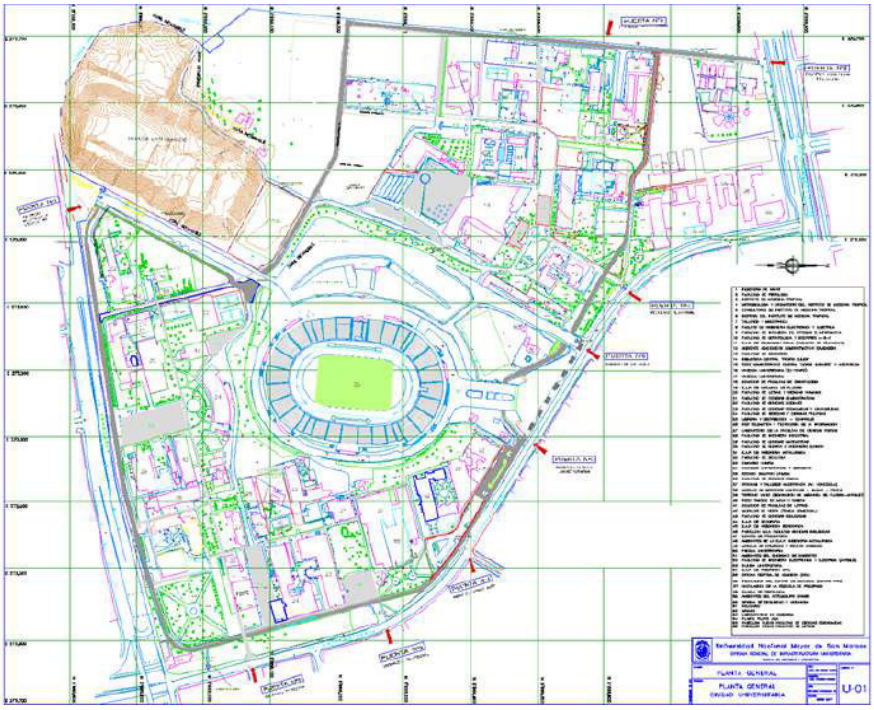

Figura 35. Plano cartográfico Campus universitario. Fuente. UNMSM (2023)

1.2. Conversión del plano

Proceso de conversión del plano vectorial en DWG AUTOCAD a formato PNG en alta resolución.

1. Con la herramienta «DWG TrueView Autodesk» realiza el proceso de edición.

    1.1. Se genera previamente un nuevo grupo de pinceles de 0.5 mm porqué los pinceles por defecto usado en el plano son muy anchos

    1.2. Se eliminan los elementos de indicativos de texto, leyendas y cuadricula. De esta forma el plano queda limpio de elementos no necesarios para ser usado en el sistema.

    1.3. Se exporta en formato PDF con imagen vectorial.

2. Con la herramienta «Inkscape versión 1.3» se realiza el proceso de edición.

    2.1. Se importa el archivo PDF generado previamente, InkScape extrae la imagen vectorial del plano.

    2.2. Se edita el archivo asignando color a las área de los locales y asignando un titulo a cada local.

    2.3. Se rota la imagen en modo vertical, con el norte orientado hacia arriba, el Este a la izquierda, el Oeste a la derecha y el Sur hacia abajo.

    2.4. Se guarda el archivo vectorial en formato SVG. Por defecto InkScape administra el proyecto con formato SVG.

    2.5. Se exporta en formato PNG con dimensiones 4489 pixeles ancho x 5561 alto (ver Figura 36).

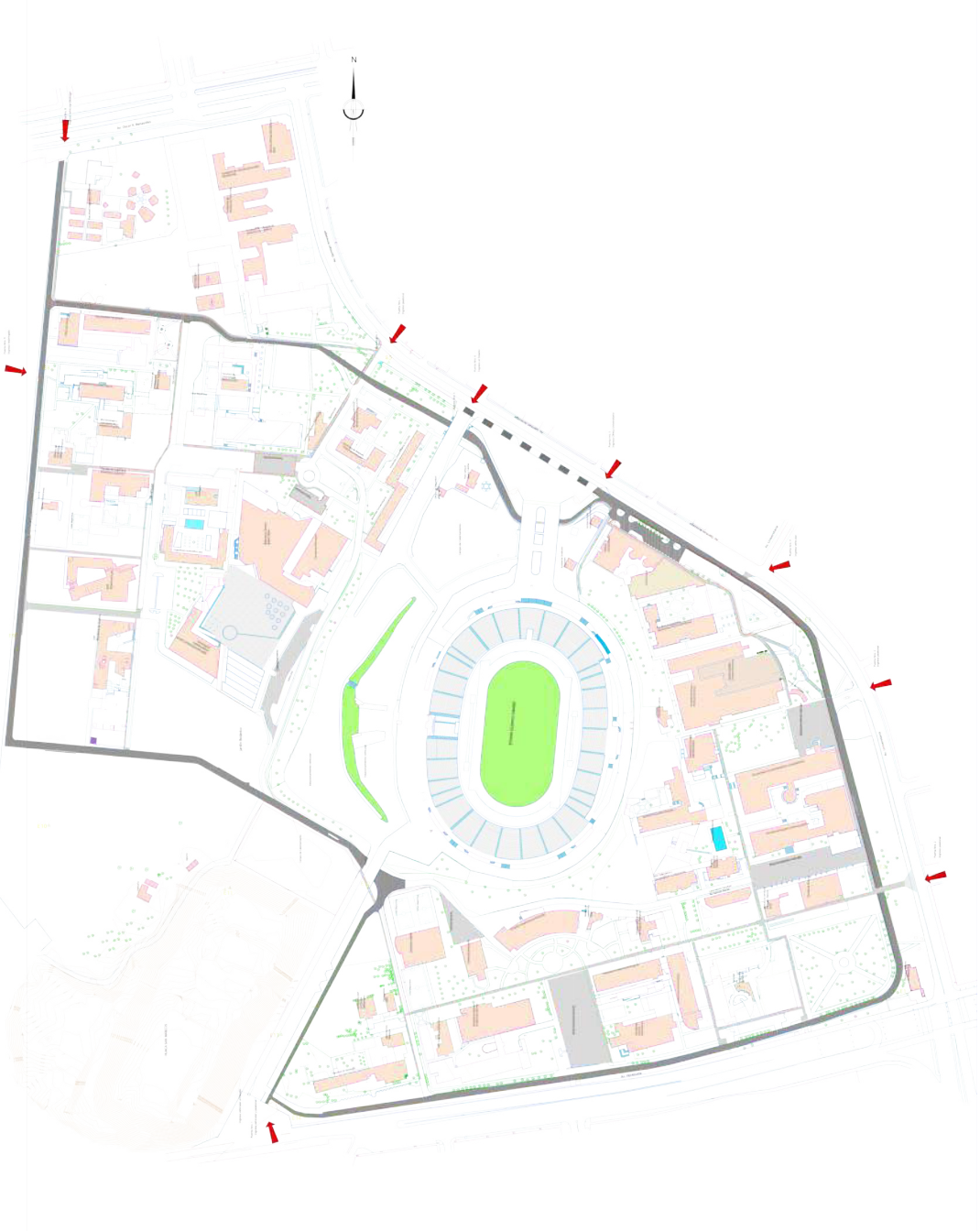

Figura 36. Plano cartográfico convertido Campus universitario. Fuente propia

2. Conversión del plano a coordenadas geográficas grados decimales

2.1. Transferir a Google Earth

El plano digital original utiliza coordenadas geográficas UTM y esta diseñado de forma horizontal, se debe llevar a un plano cartográfico universal proporcionado por Google Earth considerando coordenadas geográficas en grados decimales y el diseño en sentido vertical, con el norte indicando hacia arriba (ver Figura 37).

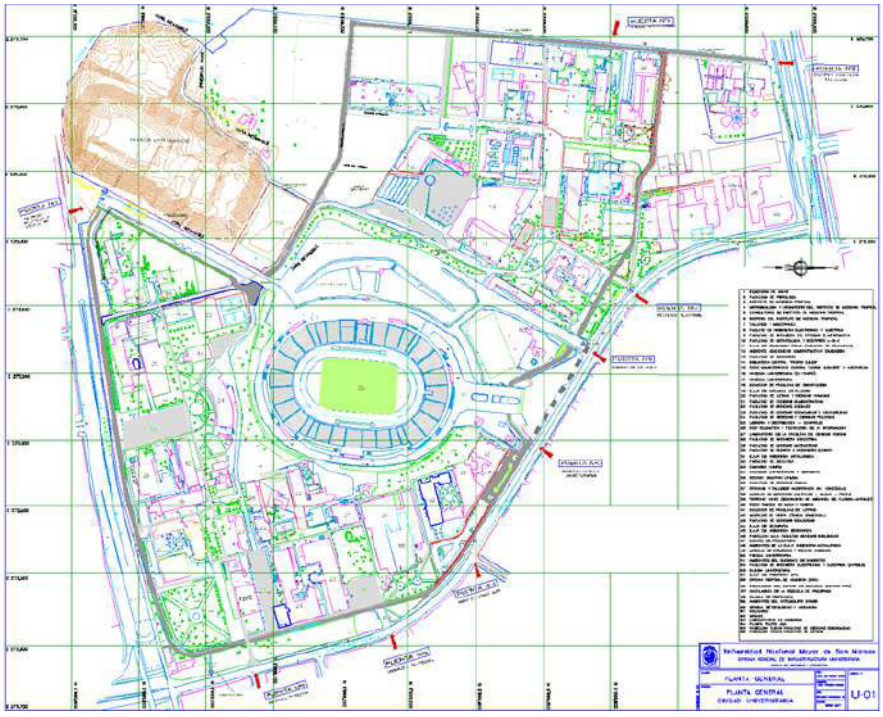

Figura 37. Plano del Campus universitario. Fuente U.N.M.S.M. (2023).

2.2. Establecer límites

La imagen del plano (ver Figura 37) contiene las coordenadas en unidad de medida UTM (Universal Transversa Mercator). Con Google Earth se establece los limites considerando las coordenadas UTM para cada uno de los 4 vértices, en nuestro caso lo llamamos «Límite».

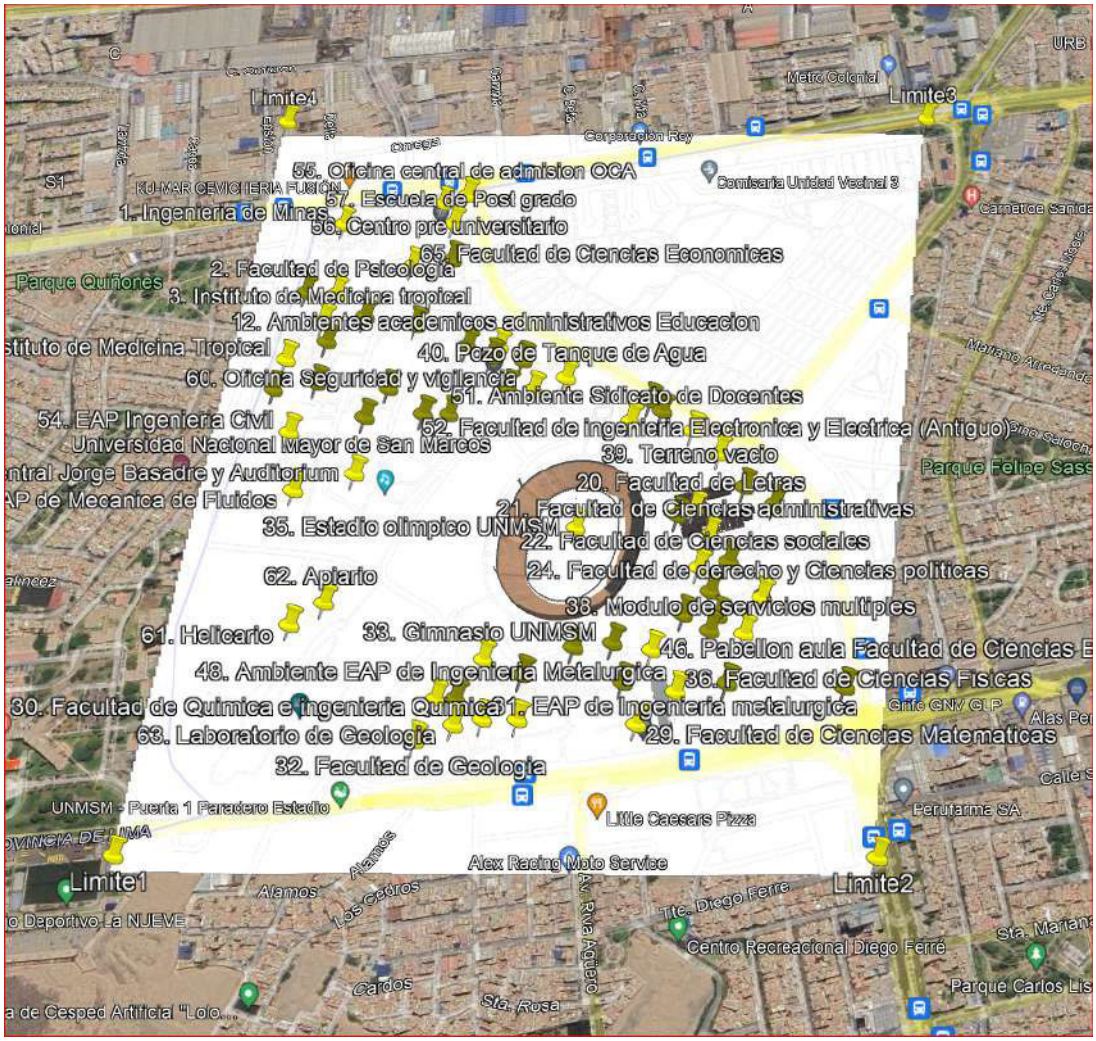

Figura 38. Plano del Campus universitario en Google Earth. Fuente propia.

En la imagen (ver Figura 38) se distingue el área en el mapa cartográfico universal con color de fondo blanco, el plano 2D real del Campus universitario usando coordenadas UTM con Norte hacia arriba y Este hacia a la derecha y Oeste a la izquierda y el Sur abajo, es la forma correcta de mostrarlo para poder obtener los limites del plano como área rectangular, son los 4 puntos indicados marcados con un PIN de color amarillo y con nombres: «Limite1», «Limite2», «Limite3»,

«Limite4». Estos limites siguen la secuencia «Limite1» inferior izquierda, «Limite2» inferior derecha, «Limite3» superior derecha, «Limite4» superior izquierda.

Dentro del área se han establecido las coordenadas geográficas de los locales referenciados en la leyenda del plano cartográfico del Campus universitario.

No se selecciona el área del Campus universitario porque es un polígono con lados irregulares como se muestra en la imagen (ver Figura 39), no se aplica para el cálculos de coordenadas geográficas basadas en cuadrantes.

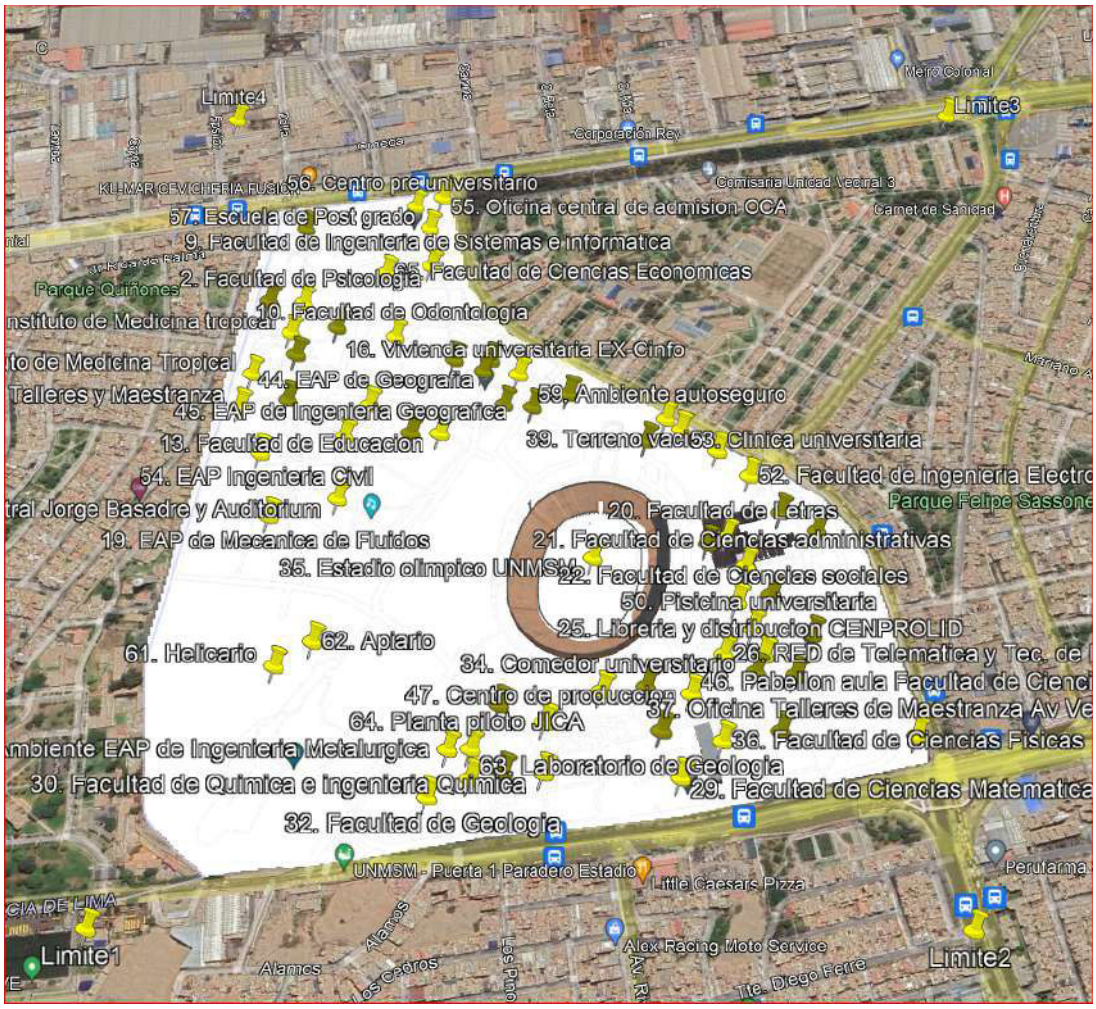

Figura 39. Área del Campus universitario en Google Earth. Fuente propia.

3. Código fuente

3.1. Manifiesto

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="3"
    android:versionName="1.0.2" >
       <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:largeHeap="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AdnTemaApp" >
        <activity
            android:name="com.echm.geolocation.MainMapa"
            android:theme="@style/SplashTema"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.echm.geolocation.CategoriaLocal"
            android:label="@string/title_activity_categorialocal"
            android:configChanges="orientation|screenSize" >
        </activity>
        <activity
            android:name="com.echm.geolocation.Locales"
            android:label="@string/title_activity_locales"
            android:configChanges="orientation|screenSize" >
        </activity>
        <activity
            android:name="com.echm.geolocation.DetalleLocal"
            android:label="@string/title_activity_detallelocal"
            android:configChanges="orientation|screenSize" >
        </activity>
        <activity
            android:name="com.echm.geolocation.Ubicame"
            android:label="@string/title_activity_ubicame"
            android:configChanges="orientation|screenSize">
            <meta-data
```

```
                android:name="android.app.searchable"
                android:resource="@xml/searchable" />
        </activity>
    </application>
</manifest>
```

3.2. Package: com.echm.geolocation

3.2.1. Clase: MainMapa

```
package com.echm.geolocation;

import android.os.Bundle;
import android.os.Handler;
import android.app.Activity;
import android.content.Intent;
import android.view.MotionEvent;
import android.view.Window;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.LinearLayout;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * MainMapa. Entrada al sistema. Main.
 *
 */
public class MainMapa extends Activity {
        LinearLayout vistaSlide;
        Handler lHandler;
        Runnable endSplash;

    /**
     * @author ECHM/.
     * Carga la aplicacion.
     */
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                getWindow().requestFeature(Window.FEATURE_NO_TITLE);
                super.onCreate(savedInstanceState);
         setContentView(R.layout.splash);
         vistaSlide =  findViewById(R.id.EfectoSlideMain);
         endSplash = () -> {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
```

```
            Intent iMenuMain = new Intent(getBaseContext(),Ubicame.class);
            iMenuMain.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(iMenuMain);
            finish();
        };
        lHandler=new Handler();
        lHandler.postDelayed(endSplash, 1000L);
       }

    /**
     * @author ECHM/.
     * Proceso de toma de touch en el screen
     * @return boolean
     */
    @Override  public boolean onTouchEvent(MotionEvent event) {
      if(event.getAction() == MotionEvent.ACTION_DOWN) {
       lHandler.removeCallbacks(endSplash);
       Animation out = AnimationUtils.makeOutAnimation(getBaseContext(), true);
       vistaSlide.startAnimation(out);
        Intent iUbicame = new Intent(getBaseContext(),Ubicame.class);
        startActivity(iUbicame);
        return true;
      }
      return super.onTouchEvent(event);
    }
}
```

3.2.2. Clase: CategoriaLocal

```
package com.echm.geolocation;

import java.util.ArrayList;
import echm.db.Conecta;
import echm.entidades.AdaptadorListaCategoria;
import echm.entidades.EntCategoria;
import android.annotation.SuppressLint;
import android.content.Context;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.Toast;

/**
```

```
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * CategoriaLocal
 *
 */
public class CategoriaLocal extends Activity {
    ProgressBar barraprogreso;
    ListView mListCategoria;
    ArrayList<EntCategoria> listaItemLView = null;
    RelativeLayout vistaSlide;

    /**
     * @author ECHM/.
     * Carga de formulario
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_categorialocal);
        getActionBar().setDisplayShowHomeEnabled(false);
        getActionBar().setDisplayHomeAsUpEnabled(true);
        vistaSlide = findViewById(R.id.SlideListaCategoriaslocales);
        listaItemLView= new ArrayList<>();
        barraprogreso = new ProgressBar(CategoriaLocal.this, null,
android.R.attr.progressBarStyleLarge);
        barraprogreso.setIndeterminate(true);
        barraprogreso.setVisibility(View.VISIBLE);
        RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(100,
100);
        params.addRule(RelativeLayout.CENTER_IN_PARENT);
        vistaSlide.addView(barraprogreso, params);
        new CargaDatosDeCategoria(getApplicationContext(), barraprogreso,
listaItemLView).execute();
        mListCategoria = findViewById(R.id.listaCategorias);
        mListCategoria.setAdapter(new
AdaptadorListaCategoria(getApplicationContext(), R.layout.listaitemcategoria,
listaItemLView));
        mListCategoria.setOnItemClickListener((parent, view, position, id) -> {
            if (listaItemLView.get((int) id).IdCategoria >0) {
                Intent iDetalle = new Intent(getBaseContext(), Locales.class);
                iDetalle.putExtra("idcategoria", listaItemLView.get((int)
id).IdCategoria);
                iDetalle.putExtra("descripcion", listaItemLView.get((int)
id).Descripcion);
                startActivity(iDetalle);
            }
        });
    }
```

```
    /**
     * @author ECHM/.
     * @param hasFocus Si tiene el foco, refresca la lista
     */
    @Override
    public void onWindowFocusChanged(boolean hasFocus) {
        if (hasFocus) {
            mListCategoria.invalidateViews();
        }
    }

    /**
     * @author ECHM/.
     * @param menu Almacen de item del menu superior.
     * @return boolean
     */
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.categorialocal, menu);
        return true;
    }

    /**
     * @author ECHM/.
     * Seleccion de item de Menu overflow.
     * @return boolean
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            Intent iHome = new Intent(getBaseContext(), Ubicame.class);
            iHome.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(iHome);
            finish();
        }
        if (item.getItemId() == R.id.action_ubicame) {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
            startActivity(new Intent(getBaseContext(), Ubicame.class));
            finish();
        }
        if (item.getItemId() == R.id.action_locales) {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
            startActivity(new Intent(getBaseContext(), CategoriaLocal.class));
            finish();
        }
        return true;
    }

    @Override
    protected void onResume() {
        super.onResume();
```

```
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onStop() {
        super.onStop();
        if (barraprogreso != null) {
            if (barraprogreso.getVisibility() == View.VISIBLE) {
                barraprogreso.setVisibility(View.GONE);
            }
        }
    }

    /**
     * @author ECHM/.
     * Cierra la ventana
     */
    @Override
    public void onDestroy() {
        super.onDestroy();
    }

    /**
     * @author ECHM/.
     * Ejecucion en Background la carga de categoria.
     */
    static class CargaDatosDeCategoria extends AsyncTask<Object, Object, Object> {
        @SuppressLint("StaticFieldLeak")
        private final Context contextoRef;
        @SuppressLint("StaticFieldLeak")
        private final ProgressBar barraprogresoRef;
        private ArrayList<EntCategoria> listaItemLViewVarLocal;
        private final ArrayList<EntCategoria> listaItemLViewRef;
        public CargaDatosDeCategoria(Context contexto,ProgressBar
barraprogreso,ArrayList<EntCategoria> listaItemLView){
            contextoRef =contexto;
            barraprogresoRef =barraprogreso;
            listaItemLViewRef=listaItemLView;
        }

        /**
         * @author ECHM/.
         * Carga los parametros.
         */
        @Override
        protected Object doInBackground(Object... params) {
            Conecta vConecta = new Conecta(contextoRef);
            listaItemLViewVarLocal = vConecta.leerCategoria();
            vConecta.cierraDB();
            return true;
```

```
        }

        /**
         * @author ECHM/.
         * Carga la data obtenida del servidor.
         */
        @Override
        protected void onPostExecute(Object result) {
            if (barraprogresoRef.getVisibility() == View.VISIBLE) {
                barraprogresoRef.setVisibility(View.GONE);
            }
            if (listaItemLViewVarLocal.size() > 0) {
                listaItemLViewRef.clear();
                listaItemLViewRef.addAll(listaItemLViewVarLocal);
            } else
                Toast.makeText(contextoRef, "No existe disponibilidad de datos",
Toast.LENGTH_LONG).show();
        }
    }
}
```

3.2.3. Clase: DetalleLocal

```
package com.echm.geolocation;

import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
import java.util.Optional;
import java.util.Timer;
import java.util.TimerTask;
import echm.db.Conecta;
import echm.entidades.Constantes;
import echm.entidades.EntProveedor;
import echm.entidades.EntConfiguracion;
import echm.entidades.EntCoordenada;
import echm.entidades.EntLocal;
import echm.generales.ImagenCore;
import echm.generales.Mapacore;
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.content.res.Configuration;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
```

```
import android.graphics.Rect;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.util.DisplayMetrics;
import android.view.GestureDetector;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager;
import android.view.GestureDetector.SimpleOnGestureListener;
import android.view.animation.AnimationUtils;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.HorizontalScrollView;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Detallelocal.
 *
 */
public class DetalleLocal extends Activity {
    RelativeLayout vistaSlide;
    ProgressBar barraprogreso;
    ImageView imagenMapa;
    ImageView imagenBrujula;
    ImageView imagenMarcador;
    ImageView lyimagenLocal;
    TextView lytextoLocal, textoDistancia;
    LinearLayout lyDatosLocal;
    Button btnVamos;
    GestureDetector Gesto;
    HorizontalScrollView hScroll;
```

```
    ScrollView vScroll;
    Location UltimaLocalizacion;
    LocationManager locationManager;
    LocationListener locationListener;
    SensorManager sensorManager;
    SensorEventListener sensorEventListener;
    List<Sensor> listaDeSensores;
    Sensor Acelerometro, Magnetometro;
    static Bitmap BitMapMapa = null;
    Timer timerExplorador;
    boolean GpsActivo = false, btnBusquedaActiva = false;
    Rect rectImagenMarcador, rectImagenBrujula, rectImagenMapa, rectlyDatosFacultad;
    String tituloAproximacion;
    EntLocal facultadActiva;
    EntConfiguracion Configuracion;
    double radianBrujula = 0;
    EntCoordenada xyDerivada;
    float azimut = 0;
    int idfacultad = 0;
    String Nombre="";
    List<String> listaNombresProveedor;
    ArrayList<EntProveedor> listaProveedor = null;

    /**
     * @author ECHM/.
     * Carga de formulario
     */
    @SuppressLint("ClickableViewAccessibility")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detallelocal);
        vistaSlide = findViewById(R.id.SlideDetalleLocal);
        Bundle extras = getIntent().getExtras();
        Configuracion= new EntConfiguracion();
        listaProveedor= new ArrayList<>();
        if (extras != null) {
            idfacultad = extras.getInt("idfacultad");
            Nombre=extras.getString("nombre");
            barraprogreso = new ProgressBar(DetalleLocal.this, null,
android.R.attr.progressBarStyleLarge);
            barraprogreso.setIndeterminate(true);
            barraprogreso.setVisibility(View.VISIBLE);
            RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(100,
100);
            params.addRule(RelativeLayout.CENTER_IN_PARENT);
            vistaSlide.addView(barraprogreso, params);
            facultadActiva= new EntLocal();
            facultadActiva.IdLocal =idfacultad;
            new DetalleLocal.CargaDatosDeLocales(this, barraprogreso, facultadActiva,
Configuracion).execute();
        }
        getActionBar().setDisplayShowHomeEnabled(false);
        getActionBar().setDisplayHomeAsUpEnabled(true);
```

```
        getActionBar().setTitle(String.format("Local: %s",Nombre));

        vistaSlide = findViewById(R.id.SlideDetalleLocal);
        hScroll = findViewById(R.id.scrollHzUbicame);
        vScroll = findViewById(R.id.scrollVTUbicame);
        btnVamos = findViewById(R.id.btnVamos);
        imagenMapa = findViewById(R.id.imagenMapa);
        imagenBrujula = findViewById(R.id.imagenBrujula);
        imagenMarcador = findViewById(R.id.imagenMarcador);
        lyimagenLocal = findViewById(R.id.lyimagenFacultad);
        lytextoLocal = findViewById(R.id.lytextoFacultad);
        lyDatosLocal = findViewById(R.id.lyDatosFacultad);
        textoDistancia = findViewById(R.id.textoDistancia);

        rectImagenBrujula = new Rect();
        rectImagenMarcador = new Rect();
        rectImagenMapa = new Rect();
        rectlyDatosFacultad = new Rect();
        xyDerivada = new EntCoordenada(0, 0);

        BitMapMapa=new ImagenCore().CargaImagenMapa(getApplicationContext());
        if (facultadActiva != null && BitMapMapa!=null) {
            imagenMapa.setImageBitmap(BitMapMapa);
            lyDatosLocal.bringToFront();
            imagenBrujula.bringToFront();
            imagenMarcador.bringToFront();
            btnVamos.bringToFront();
            Gesto = new GestureDetector(DetalleLocal.this, new GestosListener());
            imagenMapa.setOnTouchListener((arg0, arg1) -> {
                ((InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE)).hideSoftInputFromWindow(imagenMapa.ge
tWindowToken(), 0);
                return Gesto.onTouchEvent(arg1);
            });

        }
        //-- Boton Ir.... Activacion de los listeners.
        btnVamos.setOnClickListener(v -> {
            IniciaPosicionamiento();
            if(btnBusquedaActiva)
                fncBtnVamos(false);
        });
        //-- Inicializa el Location listener. No se activa hasta que el manager se
instancie.
        locationListener = new LocationListener() {
            public void onLocationChanged(Location location) {
                UltimaLocalizacion = location;
                //-- Este codigo es solo para pruebas en desarrollo
                //Location gpsOrigen = new Location("gpsOrigen");
                //gpsOrigen.setLatitude(-12.059873);
                //gpsOrigen.setLongitude(-77.080987);
                //UltimaLocalizacion=gpsOrigen;
                //--- ** ---
            }
```

```
            public void onStatusChanged(String provider, int status, Bundle extras) {
            }
            public void onProviderEnabled(String provider) {
            }
            public void onProviderDisabled(String provider) {
                btnBusquedaActiva = false;
            }
        };
        sensorEventListener = new SensorEventListener() {
            float[] Gravedad;
            float[] Geomagnetica;

            @Override
            public void onAccuracyChanged(Sensor arg0, int arg1) {
                // TODO Auto-generated method stub
            }
            @Override
            public void onSensorChanged(SensorEvent arg0) {
                if (arg0.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
                    Gravedad = arg0.values;
                if (arg0.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
                    Geomagnetica = arg0.values;
                if (Gravedad != null && Geomagnetica != null) {
                    float[] R = new float[9];
                    float[] I = new float[9];
                    if (SensorManager.getRotationMatrix(R, I, Gravedad,
Geomagnetica)) {
                        float[] orientacion = new float[3];
                        SensorManager.getOrientation(R, orientacion);
                        azimut = orientacion[0];
                    }
                }
            }
        };
        locationManager = (LocationManager)
this.getSystemService(Ubicame.LOCATION_SERVICE);
        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

        if (timerExplorador == null) {
            timerExplorador = new Timer();
            timerExplorador.scheduleAtFixedRate(new TimerTask() {
                public void run() {
                    ExplorarLocalizacion();
                }
            }, 0, Constantes.TIEMPO_TIMER);
        }
    }

    /**
     * @author ECHM/.
     * @param menu Almacen de item del menu superior.
     * @return boolean
     */
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.detallelocal, menu);
        return true;
    }

    /**
     * @author ECHM/.
     * Seleccion de item de Menu overflow.
     * @return boolean
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            finish();
        }
        if (item.getItemId() == R.id.action_ubicame) {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
            startActivity(new Intent(getBaseContext(), Ubicame.class));
            finish();
        }
        if (item.getItemId() == R.id.action_locales) {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
            startActivity(new Intent(getBaseContext(), Locales.class));
            finish();
        }
        return true;
    }

    @Override
    public void onBackPressed() {
        finish();
        super.onBackPressed();
    }

    /**
     * @author ECHM/.
     * Dispositivo en primer plano
     */
    @Override
    protected void onResume() {
        if (btnBusquedaActiva) {
            IniciaPosicionamiento();
        }
        super.onResume();
    }

    /**
     * @author ECHM/.
     * Dispositivo en pausa o segundo plano
     */
    @Override
    protected void onPause() {
```

```
        if (sensorManager != null) {
            sensorManager.unregisterListener(sensorEventListener);
        }
        if (locationManager != null) {
            locationManager.removeUpdates(locationListener);
        }
        if (timerExplorador != null) {
            timerExplorador.cancel();
            timerExplorador = null;
        }
        super.onPause();
    }

    /**
     * @author ECHM/.
     * Cierra la ventana
     */
    @Override
    public void onDestroy() {
        super.onDestroy();
    }

    @Override
    public void onWindowFocusChanged(boolean hasFocus) {
        if (hasFocus && facultadActiva != null) {
            visiblePosicionLocal(facultadActiva.cXImagen, facultadActiva.cYImagen);
            ActualizaScreen();
            fncBtnVamos(true);
        }
    }

    /**
     * @author ECHM/.
     * Inicia la gestion de posicionamiento
     */
    private void IniciaPosicionamiento() {
        GpsActivo=locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
        if (GpsActivo) {
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
                    ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(DetalleLocal.this,new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,Manifest.permission.ACCESS_COARSE_L
OCATION},1);
            }else{
                listaNombresProveedor = locationManager.getProviders(true);
                listaNombresProveedor.forEach(p-> listaProveedor.add(new
EntProveedor(p)));
                Criteria criterio = new Criteria();
                criterio.setAccuracy(Criteria.ACCURACY_FINE);
                String Proveedor = locationManager.getBestProvider(criterio, false);
                if(Proveedor != null){
                    Optional<EntProveedor> rtnValor= listaProveedor.stream().filter(p
-> p.nombreProveedor.contains(Proveedor)).findFirst();
```

```
                    rtnValor.ifPresent(p -> p.esMejor = true);
                    locationManager.requestLocationUpdates(Proveedor, 1000, 1,
locationListener);
                    UltimaLocalizacion =
locationManager.getLastKnownLocation(Proveedor);
                    listaDeSensores =
sensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER);
                    if (listaDeSensores.size() > 0) {
                        Acelerometro =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
                        Magnetometro =
sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
                        if (Acelerometro != null && Magnetometro != null) {
                            sensorManager.registerListener(sensorEventListener,
Acelerometro, SensorManager.SENSOR_DELAY_FASTEST);
                            sensorManager.registerListener(sensorEventListener,
Magnetometro, SensorManager.SENSOR_DELAY_FASTEST);
                            btnBusquedaActiva = true;
                        } else {
                            Toast.makeText(getApplicationContext(),"Este dispositivo
no cuenta con los recursos para el servicio de posicionamiento",
Toast.LENGTH_LONG).show();
                        }
                    } else {
                        Toast.makeText(getApplicationContext(),"Este dispositivo no
dispone de sensores", Toast.LENGTH_LONG).show();
                    }
                }
            }
        } else {
            Toast.makeText(getApplicationContext(),"No esta disponible el servicio de
localizacion", Toast.LENGTH_LONG).show();
        }
        if (btnBusquedaActiva && timerExplorador == null) {
            timerExplorador = new Timer();
            timerExplorador.scheduleAtFixedRate(new TimerTask() {
                public void run() {
                    ExplorarLocalizacion();
                }
            }, 0, Constantes.TIEMPO_TIMER);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults) {
        if(requestCode==1){
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(getApplicationContext(),"Se ha habilitado, vuelva a
intentarlo", Toast.LENGTH_LONG).show();
            }
        }
    }

    @Override
    public void onConfigurationChanged(@NonNull Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
```

```
        if(btnVamos.getVisibility()==View.VISIBLE){
            if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE ||
                    newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
                DisplayMetrics metrics = new DisplayMetrics();
                getWindowManager().getDefaultDisplay().getMetrics(metrics);
                btnVamos.setX(metrics.widthPixels - btnVamos.getWidth());
                btnVamos.setY(metrics.heightPixels - (btnVamos.getHeight() * 2));
            }
        }
    }

    /**
     * @author ECHM/.
     * Activa la posicion visible del icono de lugar
     */
    private void visiblePosicionDispositivo(int cXImagen, int cYImagen) {
        rectImagenBrujula.top = cYImagen;
        rectImagenBrujula.left = cXImagen;
        rectImagenBrujula.bottom = cYImagen + imagenBrujula.getHeight();
        rectImagenBrujula.right = cXImagen + imagenBrujula.getWidth();

        rectImagenMarcador.top = cYImagen;
        rectImagenMarcador.left = cXImagen;
        rectImagenMarcador.bottom = cYImagen + imagenMarcador.getHeight();
        rectImagenMarcador.right = cXImagen + imagenMarcador.getWidth();
        vScroll.scrollTo(cXImagen - (hScroll.getWidth() / 2), cYImagen -
(vScroll.getHeight() / 2));
        hScroll.scrollTo(cXImagen - (hScroll.getWidth() / 2), cYImagen -
(vScroll.getHeight() / 2));
    }

    /**
     * @author ECHM/.
     * Activa la posicion visible del icono de la facultad.
     */
    private void visiblePosicionLocal(int cXImagen, int cYImagen) {
        lytextoLocal.setText(facultadActiva.Corto);
        rectlyDatosFacultad.top = cYImagen - lyDatosLocal.getHeight();
        rectlyDatosFacultad.left = cXImagen - lyDatosLocal.getWidth();
        rectlyDatosFacultad.bottom = cYImagen;
        rectlyDatosFacultad.right = cXImagen;
        int PosicionXRelativa = cXImagen - (hScroll.getWidth() / 2);
        int PosicionYRelativa = cYImagen - (vScroll.getHeight() / 2);
        vScroll.scrollTo(PosicionXRelativa, PosicionYRelativa);
        hScroll.scrollTo(PosicionXRelativa, PosicionYRelativa);
        lyDatosLocal.setVisibility(View.VISIBLE);
    }

    /**
     * @author ECHM/.
     * Explora la localizacion y la visualizacion en el mapa
     */
    private void ExplorarLocalizacion() {
        if (GpsActivo) {
```

```
            if (btnBusquedaActiva && UltimaLocalizacion != null) {
                //-- Posiciona la ubicacion del dispositivo si se encuentra dentro de
la area visible del mapa en la pantalla del dispositivo.
                //-- Considere el mapa el area valida de visualizacion del
dispositivo y si se encuentra dentro del area visible en el dispositivo.
                xyDerivada =
Mapacore.retornaCoordenadaXYConLatitudLongitudXInterseccion(UltimaLocalizacion.getLat
itude(), UltimaLocalizacion.getLongitude(),Configuracion);
                if (xyDerivada.X > 0 && xyDerivada.Y > 0) {
                    visiblePosicionDispositivo(xyDerivada.X, xyDerivada.Y);
                }
                //-- *** --
                if (azimut != 0 && facultadActiva != null) {
                    //-- Direccion o rumbo hacia el destino respecto a la ultima
localizacion
                    Location gpsDestino = new Location("gpsDestino");
                    gpsDestino.setLatitude(facultadActiva.Latitud);
                    gpsDestino.setLongitude(facultadActiva.Longitud);
                    double direccionCoodenada =
Mapacore.direccionDesdeCoordenadaDeOrigen(UltimaLocalizacion, gpsDestino);
                    double desplazamiento =
Mapacore.rtnGradosDelHeadingDesplazados(((WindowManager)
getSystemService(Context.WINDOW_SERVICE)).getDefaultDisplay().getRotation());
                    if (direccionCoodenada < (Math.toDegrees(azimut) +
desplazamiento)) {
                        radianBrujula = (Math.PI * (360 - Math.abs(direccionCoodenada
- (Math.toDegrees(azimut) + desplazamiento))) / 180.0);
                    } else {
                        radianBrujula = (Math.PI * (direccionCoodenada -
(Math.toDegrees(azimut) + desplazamiento)) / 180.0);
                    }
                    if (xyDerivada.X > 0 && xyDerivada.Y > 0) {
                        String ProveedorMejor="";
                        Optional<EntProveedor> rtnValor=
listaProveedor.stream().filter(p -> p.esMejor).findFirst();
                        if(rtnValor.isPresent()){
                            ProveedorMejor=rtnValor.get().nombreProveedor;
                        }
                        tituloAproximacion = String.format(Locale.getDefault(),
"Prov.: %s. Dist.: %.1f m. Lat.: %f, Long.: %f, Bear. Dest.: %f °",
ProveedorMejor,UltimaLocalizacion.distanceTo(gpsDestino),

UltimaLocalizacion.getLatitude(),UltimaLocalizacion.getLongitude(),UltimaLocalizacion
.bearingTo(gpsDestino));
                    } else {
                        tituloAproximacion = "";
                    }
                }
            }
        }
        runOnUiThread(this::ActualizaScreen);
    }

    /**
     * @author ECHM/.
     * Actualiza la pantalla
     */
    public void ActualizaScreen() {
```

```
        imagenMapa.getLocalVisibleRect(rectImagenMapa);
        if (btnBusquedaActiva && UltimaLocalizacion != null) {
            if (xyDerivada.X > 0 && xyDerivada.Y > 0 &&
Rect.intersects(rectImagenMapa, rectImagenBrujula)) {
                imagenBrujula.setRotation((float) Math.toDegrees(radianBrujula));
                imagenBrujula.setVisibility(View.VISIBLE);
                imagenBrujula.setX(hScroll.getLeft());
                imagenBrujula.setY(hScroll.getTop());

                imagenMarcador.setRotation((float) Math.toDegrees(radianBrujula));
                imagenMarcador.setVisibility(View.VISIBLE);
                imagenMarcador.setX((float) (rectImagenMarcador.left -
(imagenMarcador.getWidth() / 2.0) - rectImagenMapa.left));
                imagenMarcador.setY((float) (rectImagenMarcador.top -
(imagenMarcador.getHeight() / 2.0) - rectImagenMapa.top));
            } else {
                imagenBrujula.setVisibility(View.INVISIBLE);
                imagenMarcador.setVisibility(View.INVISIBLE);
            }
        } else {
            imagenBrujula.setVisibility(View.INVISIBLE);
            imagenMarcador.setVisibility(View.INVISIBLE);
        }
        if (facultadActiva != null) {
            if (Rect.intersects(rectImagenMapa, rectlyDatosFacultad)) {
                lyDatosLocal.setVisibility(View.VISIBLE);
                lyDatosLocal.setX((20 + rectlyDatosFacultad.left -
rectImagenMapa.left));
                lyDatosLocal.setY((20 + rectlyDatosFacultad.top -
rectImagenMapa.top));
            } else {
                lyDatosLocal.setVisibility(View.INVISIBLE);
            }
        } else {
            lyDatosLocal.setVisibility(View.INVISIBLE);
        }
        if (btnBusquedaActiva) {
            textoDistancia.setText(tituloAproximacion);
            textoDistancia.setVisibility(View.VISIBLE);
            textoDistancia.setX(hScroll.getWidth() - textoDistancia.getWidth() - 10);
            textoDistancia.setY(vScroll.getHeight() - textoDistancia.getHeight() -
10);
        } else {
            textoDistancia.setVisibility(View.INVISIBLE);
        }
    }

    /**
     *  @author ECHM/.
     * Control del boton Vamos con busqueda activa
     */
    private void fncBtnVamos(boolean activo){
        if(activo){
            DisplayMetrics metrics = new DisplayMetrics();
            getWindowManager().getDefaultDisplay().getMetrics(metrics);
```

```
            btnVamos.setX(metrics.widthPixels - btnVamos.getWidth());
            btnVamos.setY(metrics.heightPixels - (btnVamos.getHeight() * 2));
            btnVamos.setVisibility(View.VISIBLE);
        }else{
            btnVamos.setVisibility(View.INVISIBLE);
        }
    }

    /**
     * @author ECHM/.
     * Ejecucion de gestos de la imagen.
     */
    private static class GestosListener extends SimpleOnGestureListener {
        @Override
        public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float
velocityY) {
            int SWIPE_MIN_DISTANCE = 120;
            int SWIPE_THRESHOLD_VELOCITY = 200;
            if (e1.getX() - e2.getX() > SWIPE_MIN_DISTANCE && Math.abs(velocityX) >
SWIPE_THRESHOLD_VELOCITY) {
                return true;
            } else if (e2.getX() - e1.getX() > SWIPE_MIN_DISTANCE &&
Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY) {
                return true;
            }
            if (e1.getY() - e2.getY() > SWIPE_MIN_DISTANCE && Math.abs(velocityY) >
SWIPE_THRESHOLD_VELOCITY) {
                return true;
            } else
                return e2.getY() - e1.getY() > SWIPE_MIN_DISTANCE &&
Math.abs(velocityY) > SWIPE_THRESHOLD_VELOCITY;
        }
    }

    /**
     * @author ECHM/.
     * Ejecucion en Background la carga de locales.
     */
    static class CargaDatosDeLocales extends AsyncTask<Object, Object, Object> {
        @SuppressLint("StaticFieldLeak")
        private final Context contextoRef;
        @SuppressLint("StaticFieldLeak")
        private final ProgressBar barraprogresoRef;
        private final EntLocal facultadActivaRef;
        private final EntConfiguracion configuracionRef;
        public CargaDatosDeLocales(Context contexto, ProgressBar barraprogreso,
EntLocal facultadActiva, EntConfiguracion Configuracion){
            contextoRef =contexto;
            barraprogresoRef =barraprogreso;
            facultadActivaRef =facultadActiva;
            configuracionRef =Configuracion;
        }

        /**
         * @author ECHM/.
```

```
         * Carga los parametros
         */
        @Override
        protected Object doInBackground(Object... params) {
            Conecta vConecta = new Conecta(contextoRef);
            vConecta.leerLocal(facultadActivaRef.IdLocal,facultadActivaRef);
            vConecta.leerConfiguracion(configuracionRef);
            vConecta.cierraDB();
            return true;
        }

        /**
         * @author ECHM/.
         * Cierra la carga
         */
        @Override
        protected void onPostExecute(Object result) {
            if (barraprogresoRef != null) {
                if (barraprogresoRef.getVisibility() == View.VISIBLE) {
                    barraprogresoRef.setVisibility(View.GONE);
                }
            }
            if (facultadActivaRef!=null) {
                EntCoordenada xyDerivada =
Mapacore.retornaCoordenadaXYConLatitudLongitudXInterseccion(facultadActivaRef.Latitud
, facultadActivaRef.Longitud, configuracionRef);
                facultadActivaRef.cXImagen = xyDerivada.X;
                facultadActivaRef.cYImagen = xyDerivada.Y;
            } else
                Toast.makeText(contextoRef, "No existe disponibilidad de datos",
Toast.LENGTH_LONG).show();
        }
    }
}
```

3.2.4. Clase: Locales

```
package com.echm.geolocation;

import java.util.ArrayList;
import echm.db.Conecta;
import echm.entidades.AdaptadorListaLocal;
import echm.entidades.EntLocal;
import android.annotation.SuppressLint;
import android.content.Context;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
```

```
import android.view.animation.AnimationUtils;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.Toast;

/**
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @date 2023-06-01
 * @apiNote Locales.
 * @since SDK 29.0
 */
public class Locales extends Activity {
    ProgressBar barraprogreso;
    ListView mListFacultad;
    ArrayList<EntLocal> listaItemLView = null;
    RelativeLayout vistaSlide;
    int idcategoria = 0;
    String descripcion;

    /**
     * @author ECHM/.
     * Carga de formulario
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getActionBar().setDisplayShowHomeEnabled(false);
        getActionBar().setDisplayHomeAsUpEnabled(true);
        setContentView(R.layout.activity_locales);
        vistaSlide = findViewById(R.id.SlideListaLocales);
        Bundle extras = getIntent().getExtras();
        listaItemLView= new ArrayList<>();
        if (extras != null) {
            idcategoria = extras.getInt("idcategoria");
            descripcion = extras.getString("descripcion");
            getActionBar().setTitle(String.format("Locales: %s",descripcion));
            barraprogreso = new ProgressBar(Locales.this, null,
android.R.attr.progressBarStyleLarge);
            barraprogreso.setIndeterminate(true);
            barraprogreso.setVisibility(View.VISIBLE);
            RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(100,
100);
            params.addRule(RelativeLayout.CENTER_IN_PARENT);
            vistaSlide.addView(barraprogreso, params);
            new
CargaDatosDeLocales(this,barraprogreso,idcategoria,listaItemLView).execute();
        }
        mListFacultad = findViewById(R.id.listaLocales);
```

```
        mListFacultad.setAdapter(new AdaptadorListaLocal(getApplicationContext(),
R.layout.listaitemfacultad, listaItemLView));
        mListFacultad.setOnItemClickListener((parent, view, position, id) -> {
            if (listaItemLView.size()>0 && listaItemLView.get((int) id).IdLocal > 0)
{
                Intent iDetalle = new Intent(getBaseContext(), DetalleLocal.class);
                iDetalle.putExtra("idfacultad", listaItemLView.get((int)
id).IdLocal);
                iDetalle.putExtra("nombre", listaItemLView.get((int) id).Nombre);
                startActivity(iDetalle);
            }
        });
    }

    /**
     * @author ECHM/.
     * @param hasFocus Refresca la lista
     */
    @Override
    public void onWindowFocusChanged(boolean hasFocus) {
        if (hasFocus) {
            mListFacultad.invalidateViews();
        }
    }

    /**
     * @author ECHM/.
     * @param menu Almacen de item del menu superior.
     * @return boolean
     */
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.locales, menu);
        return true;
    }

    /**
     * @author ECHM/.
     * Seleccion de item de Menu overflow.
     * @return Valor boleano
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == android.R.id.home) {
            finish();
        }
        if (item.getItemId() == R.id.action_ubicame) {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
            startActivity(new Intent(getBaseContext(), Ubicame.class));
            finish();
        }
        if (item.getItemId() == R.id.action_locales) {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
```

```
            startActivity(new Intent(getBaseContext(), CategoriaLocal.class));
            finish();
        }
        return true;
    }

    /**
     * @author ECHM/.
     * Dispositivo en primer plano
     */
    @Override
    protected void onResume() {
        super.onResume();
    }

    /**
     * @author ECHM/.
     * Dispositivo en pausa o segundo plano
     */
    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onStop() {
        super.onStop();
        if (barraprogreso != null) {
            if (barraprogreso.getVisibility() == View.VISIBLE) {
                barraprogreso.setVisibility(View.GONE);
            }
        }
    }

    /**
     * @author ECHM/.
     * Cierra la ventana
     */
    @Override
    public void onDestroy() {
        super.onDestroy();
    }

    /**
     * @author ECHM/.
     * Ejecucion en Background la carga de locales. Se suprime la advertencia por
solo es interna
     */
    static class CargaDatosDeLocales extends AsyncTask<Object, Object, Object> {
        @SuppressLint("StaticFieldLeak")
        private final Context contextoRef;
        @SuppressLint("StaticFieldLeak")
        private final ProgressBar barraprogresoRef;
        private ArrayList<EntLocal> listaItemLViewVarLocal;
```

```
        private final ArrayList<EntLocal> listaItemLViewRef;
        int idCategoriaLocal;


        public CargaDatosDeLocales(Context contexto,ProgressBar barraprogreso,int
idcategoria, ArrayList<EntLocal> listaItemLView){
            contextoRef =contexto;
            barraprogresoRef =barraprogreso;
            listaItemLViewRef =listaItemLView;
            idCategoriaLocal=idcategoria;
        }

        /**
         * @author ECHM/.
         * Carga los parametros
         */
        @Override
        protected Object doInBackground(Object... params) {
            Conecta vConecta = new Conecta(contextoRef);
            listaItemLViewVarLocal = vConecta.leerLocales(idCategoriaLocal);
            vConecta.cierraDB();
            return true;
        }

        /**
         * @author ECHM/.
         * Cierra la carga
         */
        @Override
        protected void onPostExecute(Object result) {
            if (barraprogresoRef != null) {
                if (barraprogresoRef.getVisibility() == View.VISIBLE) {
                    barraprogresoRef.setVisibility(View.GONE);
                }
            }
            if (listaItemLViewVarLocal.size() > 0) {
                listaItemLViewRef.clear();
                listaItemLViewRef.addAll(listaItemLViewVarLocal);
            } else
                Toast.makeText(contextoRef, "No existe disponibilidad de datos",
Toast.LENGTH_LONG).show();
        }
    }
}
```

3.2.5. Clase: Ubicame

```
package com.echm.geolocation;

import java.util.ArrayList;
import java.util.List;
```

```
import java.util.Locale;
import java.util.Optional;
import java.util.Timer;
import java.util.TimerTask;
import echm.db.Conecta;
import echm.entidades.AdaptadorListaItemSearch;
import echm.entidades.Constantes;
import echm.entidades.EntProveedor;
import echm.entidades.EntConfiguracion;
import echm.entidades.EntCoordenada;
import echm.entidades.EntLocal;
import echm.generales.ImagenCore;
import echm.generales.Mapacore;
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.content.res.Configuration;
import android.location.Criteria;
import android.location.GnssMeasurement;
import android.location.GnssMeasurementsEvent;
import android.location.GnssStatus;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.location.LocationProvider;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.ActionBar;
import android.app.Activity;
import android.app.SearchManager;
import android.content.Context;
import android.content.Intent;
import android.database.MatrixCursor;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.Rect;
import android.support.v4.app.ActivityCompat;
import android.util.DisplayMetrics;
import android.view.GestureDetector;
import android.view.GestureDetector.SimpleOnGestureListener;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager;
import android.view.animation.AnimationUtils;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.HorizontalScrollView;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
```

```
import android.widget.RelativeLayout;
import android.widget.ScrollView;
import android.widget.SearchView;
import android.widget.TextView;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.support.annotation.NonNull;
import android.widget.Toast;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Ubicame. Mapa.
 *
 */
public class Ubicame extends Activity {
    ProgressBar barraprogreso;
    RelativeLayout vistaSlide;
    ImageView imagenMapa;
    ImageView imagenBrujula;
    ImageView imagenMarcador;
    ImageView lyimagenLocal;
    static Bitmap BitMapMapa = null;
    TextView lytextoLocal, textoDistancia;
    LinearLayout lyDatosLocal;
    Button btnVamos;
    GestureDetector Gesto;
    HorizontalScrollView hScroll;
    ScrollView vScroll;
    Location UltimaLocalizacion;
    LocationManager locationManager;
    LocationListener locationListener;
    SensorManager sensorManager;
    SensorEventListener sensorEventListener;
    LocationProvider ProveedorLocationManager;
    GnssStatus.Callback mGnssStatusCallback;
    GnssMeasurementsEvent.Callback mGnssMeasurementsListener;
    List<Sensor> listaDeSensores;
    Sensor Acelerometro, Magnetometro;
    Timer timerExplorador;
    boolean gpsActivo = false, networkActivo = false, btnBusquedaActiva = false;
    Rect rectImagenMarcador, rectImagenBrujula, rectImagenMapa, rectlyDatosLocal;
    String tituloAproximacion, resumenMedicionesGNSSSinProcesar="",
datosEstadoGNSS="";
```

```
    EntLocal facultadActiva = null, facultadActivaLocal;
    ArrayList<EntLocal> listaItemLViewLocal = null;
    ArrayList<EntLocal> ArrLocal = null;
    EntConfiguracion Configuracion;
    SearchView busqueda;
    double radianBrujula = 0;
    EntCoordenada xyDerivada;
    float azimut = 0;
    List<String> listaNombresProveedor;
    ArrayList<EntProveedor> listaProveedor = null;

    /**
     * @author ECHM/.
     * Carga de formulario
     */
    @SuppressLint("ClickableViewAccessibility")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ubicame);
        ActionBar actionBar = getActionBar();
        actionBar.setIcon(R.drawable.geolocationlogohome);
        actionBar.setDisplayOptions(ActionBar.DISPLAY_SHOW_HOME);
        vistaSlide = findViewById(R.id.SlideUbicame);
        listaProveedor = new ArrayList<>();
        listaItemLViewLocal = new ArrayList<>();
        Configuracion = new EntConfiguracion();
        hScroll = findViewById(R.id.scrollHzUbicame);
        vScroll = findViewById(R.id.scrollVTUbicame);
        btnVamos = findViewById(R.id.btnVamos);
        rectImagenBrujula = new Rect();
        rectImagenMarcador = new Rect();
        rectImagenMapa = new Rect();
        rectlyDatosLocal = new Rect();
         xyDerivada = new EntCoordenada(0, 0);
        imagenMapa = findViewById(R.id.imagenMapa);
        imagenBrujula = findViewById(R.id.imagenBrujula);
        imagenMarcador = findViewById(R.id.imagenMarcador);
        lyimagenLocal = findViewById(R.id.lyimagenFacultad);
        lytextoLocal = findViewById(R.id.lytextoFacultad);
        lyDatosLocal = findViewById(R.id.lyDatosFacultad);
        textoDistancia = findViewById(R.id.textoDistancia);
        barraprogreso = new ProgressBar(Ubicame.this, null,
android.R.attr.progressBarStyleLarge);
        barraprogreso.setIndeterminate(true);
        barraprogreso.setVisibility(View.VISIBLE);
        RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(100,
100);
        params.addRule(RelativeLayout.CENTER_IN_PARENT);
        vistaSlide.addView(barraprogreso, params);
        BitMapMapa = new ImagenCore().CargaImagenMapa(getApplicationContext());
        if (BitMapMapa != null) {
            imagenMapa = findViewById(R.id.imagenMapa);
            imagenMapa.setImageBitmap(BitMapMapa);
```

```
            lyDatosLocal.bringToFront();
            imagenBrujula.bringToFront();
            imagenMarcador.bringToFront();
            btnVamos.bringToFront();
            Gesto = new GestureDetector(Ubicame.this, new GestosListener());
            imagenMapa.setBackgroundColor(Color.WHITE);
            imagenMapa.setOnTouchListener((arg0, arg1) -> {
                ((InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE)).hideSoftInputFromWindow(imagenMapa.ge
tWindowToken(), 0);
                return Gesto.onTouchEvent(arg1);
            });

            //-- Boton Ir.... Activacion de los listeners.
            btnVamos.setOnClickListener(v -> {
                if (facultadActiva != null) {
                    IniciaPosicionamiento();
                    if (btnBusquedaActiva)
                        fncBtnVamos(false);
                }
            });
        }
        new CargaDatosDeLocales(this, barraprogreso, listaItemLViewLocal,
Configuracion).execute();
        //-- Inicializa el Location manager.
        locationListener = new LocationListener() {
            public void onLocationChanged(Location location) {
                UltimaLocalizacion = location;
                //-- Este codigo es solo para pruebas en desarrollo
                //Location gpsOrigen = new Location("gpsOrigen");
                //gpsOrigen.setLatitude(-12.059873);
                //gpsOrigen.setLongitude(-77.080987);
                //UltimaLocalizacion=gpsOrigen;
                //--- ** ---
            }

            public void onStatusChanged(String provider, int status, Bundle extras) {
            }

            public void onProviderEnabled(String provider) {
            }

            public void onProviderDisabled(String provider) {
                btnBusquedaActiva = false;
            }
        };

        sensorEventListener = new SensorEventListener() {
            float[] Gravedad;
            float[] Geomagnetica;

            @Override
            public void onAccuracyChanged(Sensor arg0, int arg1) {
                // TODO Auto-generated method stub
```

```
            }

            /**
             * @author ECHM/.
             * Azimut, ángulo de rotación sobre el eje -z. Este valor representa el
ángulo entre el eje y del dispositivo y el polo norte magnético.
             * @param arg0 the {@link android.hardware.SensorEvent SensorEvent}.
             */
            @Override
            public void onSensorChanged(SensorEvent arg0) {
                if (arg0.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
                    //-- Aceleracion Gx, Gy, Gz
                    Gravedad = arg0.values;
                if (arg0.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
                    //-- Campo magnético ambiental en los ejes X, Y y Z
                    Geomagnetica = arg0.values;
                if (Gravedad != null && Geomagnetica != null) {
                    float[] R = new float[9];
                    float[] I = new float[9];
                    if (SensorManager.getRotationMatrix(R, I, Gravedad,
Geomagnetica)) {
                        float[] orientacion = new float[3];
                        SensorManager.getOrientation(R, orientacion);
                        azimut = orientacion[0];
                    }
                }
            }
        };
        locationManager = (LocationManager)
this.getSystemService(Ubicame.LOCATION_SERVICE);
        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        gpsActivo = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
        networkActivo =
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (timerExplorador == null) {
            timerExplorador = new Timer();
            timerExplorador.scheduleAtFixedRate(new TimerTask() {
                public void run() {
                    ExplorarLocalizacion();
                }
            }, 0, Constantes.TIEMPO_TIMER);
        }



        mGnssStatusCallback = new GnssStatus.Callback() {
            @Override
            public void onSatelliteStatusChanged(GnssStatus status){
                StringBuilder listaSatNombres=new StringBuilder();
                int satelliteCount = status.getSatelliteCount();
                datosEstadoGNSS=String.format(Locale.getDefault(),"Nro. Satelites: %d
. ",satelliteCount);
                if(satelliteCount>0){
                    for (int cnt=0;cnt<satelliteCount;cnt++){
                        int idSat=status.getSvid(cnt);
```

```
                        int idConstelacion=status.getConstellationType(cnt);
                        switch (idConstelacion){
                            case GnssStatus.CONSTELLATION_GPS:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"GPS"));
                                break;
                            case GnssStatus.CONSTELLATION_SBAS:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"SBAS"));
                                break;
                            case GnssStatus.CONSTELLATION_GLONASS:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"GLONASS"));
                                break;
                            case GnssStatus.CONSTELLATION_QZSS:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"QZSS"));
                                break;
                            case GnssStatus.CONSTELLATION_BEIDOU:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"BEIDOU"));
                                break;
                            case GnssStatus.CONSTELLATION_GALILEO:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"GALILEO"));
                                break;
                            case GnssStatus.CONSTELLATION_IRNSS:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"IRNSS"));
                                break;
                            case GnssStatus.CONSTELLATION_UNKNOWN:

listaSatNombres.append(String.format(Locale.getDefault(),"  . Id.:%d/Nombre
Sat.:%s",idSat,"UNKNOWN"));
                                break;
                        }
                    }
                    datosEstadoGNSS+=listaSatNombres.toString();
                }
            }
        };

        //-- Valor en RAW de mediciones GNSS sin procesar, solo para evaluar los
valores, no produccion
        mGnssMeasurementsListener = new GnssMeasurementsEvent.Callback() {
            @Override
            public void onGnssMeasurementsReceived(GnssMeasurementsEvent event) {
                for (GnssMeasurement m : event.getMeasurements()) {
                    resumenMedicionesGNSSSinProcesar =
                            String.format(Locale.getDefault(),"1:%s, 2:%s, 3:%s,
4:%s, 5:%s, 6:%s, 7:%s, 8:%s, 9:%s, 10:%s, 11:%s, 12:%s, 13:%s",
```

```
                                    m.getSvid(), m.getTimeOffsetNanos(),
m.getState(), m.getReceivedSvTimeNanos(),
                                    m.getReceivedSvTimeUncertaintyNanos(),
m.getCn0DbHz(),
                                    m.getPseudorangeRateMetersPerSecond(),
m.getPseudorangeRateUncertaintyMetersPerSecond(),
                                    m.hasCarrierFrequencyHz() ?
m.getCarrierFrequencyHz() : "",
                                    m.getMultipathIndicator(),
m.getConstellationType(),
                                    m.hasAutomaticGainControlLevelDb() ?
m.getAutomaticGainControlLevelDb() : "",
                                    m.hasCarrierFrequencyHz() ?
m.getCarrierFrequencyHz() : "");
                }
            }

            @Override
            public void onStatusChanged(int status) {
            }
        };
    }

    /**
     * @author ECHM/.
     * @param menu Almacen de item del menu superior.
     * @return boolean
     */
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.ubicame, menu);
        SearchManager manager = (SearchManager)
getSystemService(Context.SEARCH_SERVICE);
        busqueda = (SearchView) menu.findItem(R.id.action_search).getActionView();
        busqueda.setSearchableInfo(manager.getSearchableInfo(getComponentName()));
        busqueda.setImeOptions(busqueda.getImeOptions() |
EditorInfo.IME_ACTION_SEARCH
                | EditorInfo.IME_FLAG_NO_EXTRACT_UI |
EditorInfo.IME_FLAG_NO_FULLSCREEN);
        busqueda.setOnSuggestionListener(new SearchView.OnSuggestionListener() {
            @Override
            public boolean onSuggestionClick(int arg0) {
                facultadActivaLocal = ArrLocal.get(arg0);
                busqueda.setQuery(facultadActivaLocal.Nombre, true);
                facultadActiva = facultadActivaLocal;
                EntCoordenada xyDerivada =
Mapacore.retornaCoordenadaXYConLatitudLongitudXInterseccion(facultadActiva.Latitud,
facultadActiva.Longitud, Configuracion);
                facultadActiva.cXImagen = xyDerivada.X;
                facultadActiva.cYImagen = xyDerivada.Y;
                visiblePosicionLocal(facultadActiva.cXImagen,
facultadActiva.cYImagen);
                fncBtnVamos(true);
                if (btnBusquedaActiva) {
                    sensorManager.unregisterListener(sensorEventListener);
                    locationManager.removeUpdates(locationListener);
                }
```

```
                btnBusquedaActiva = false;
                return false;
            }

            @Override
            public boolean onSuggestionSelect(int arg0) {
                // TODO Auto-generated method stub
                return false;
            }
        });
        busqueda.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextChange(String query) {
                if (query != null && query.length() > 0) {
                    ejecutaFiltro(query);
                }
                return true;
            }

            @Override
            public boolean onQueryTextSubmit(String arg0) {
                // TODO Auto-generated method stub
                busqueda.clearFocus();
                return false;
            }
        });
        return true;
    }

    /**
     * @author ECHM/.
     * Ejecuta el filtro de busqueda.
     */
    private void ejecutaFiltro(String query) {
        ArrLocal = new ArrayList<>();
        String[] columns = new String[]{"_id", "text"};
        Object[] temp = new Object[]{0, "default"};
        MatrixCursor cursor = new MatrixCursor(columns);
        listaItemLViewLocal.forEach(p -> {
            if (p.Nombre.toLowerCase().contains(query.toLowerCase())) {
                temp[0] = p.NroLocal;
                temp[1] = p.Nombre;
                cursor.addRow(temp);
                ArrLocal.add(p);
            }
        });
        SearchManager manager = (SearchManager)
getSystemService(Context.SEARCH_SERVICE);
        busqueda.setSearchableInfo(manager.getSearchableInfo(getComponentName()));
        busqueda.setSuggestionsAdapter(new AdaptadorListaItemSearch(this, cursor,
ArrLocal));
    }

    /**
```

```
     * @author ECHM/.
     * Seleccion de item de Menu overflow.
     * @return Valor boleano
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.action_ubicame) {
            Intent iHome = new Intent(getBaseContext(), Ubicame.class);
            iHome.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(iHome);
            finish();
        }
        if (item.getItemId() == R.id.action_refresh) {
            if (facultadActiva != null) {
                EntCoordenada xyDerivada =
Mapacore.retornaCoordenadaXYConLatitudLongitudXInterseccion(facultadActiva.Latitud,
facultadActiva.Longitud, Configuracion);
                facultadActiva.cXImagen = xyDerivada.X;
                facultadActiva.cYImagen = xyDerivada.Y;
                visiblePosicionLocal(facultadActiva.cXImagen,
facultadActiva.cYImagen);
                fncBtnVamos(true);
            }
            if (btnBusquedaActiva) {
                sensorManager.unregisterListener(sensorEventListener);
                locationManager.removeUpdates(locationListener);
            }
            btnBusquedaActiva = false;
        }
        if (item.getItemId() == R.id.action_locales) {

vistaSlide.startAnimation(AnimationUtils.makeOutAnimation(getBaseContext(), true));
            startActivity(new Intent(getBaseContext(), CategoriaLocal.class));
            finish();
        }
        return true;
    }

    /**
     * @author ECHM/.
     * Dispositivo en primer plano
     */
    @Override
    protected void onResume() {
        if (btnBusquedaActiva) {
            IniciaPosicionamiento();
        }
        super.onResume();
    }

    /**
     * @author ECHM/.
     * Dispositivo en pausa o segundo plano
     */
    @Override
```

```
    protected void onPause() {
        if (btnBusquedaActiva) {
            sensorManager.unregisterListener(sensorEventListener);
            locationManager.removeUpdates(locationListener);
            if (timerExplorador != null) {
                timerExplorador.cancel();
                timerExplorador = null;
            }
        }
        super.onPause();
    }

    @Override
    protected void onStop() {
        super.onStop();
        if (barraprogreso != null)
            barraprogreso.setVisibility(View.GONE);
    }

    /**
     * @author ECHM/.
     * Cierra la ventana
     */
    @Override
    public void onDestroy() {
        super.onDestroy();
    }

    /**
     * @author ECHM/.
     * Inicia la gestion de posicionamiento
     */
    private void IniciaPosicionamiento() {
        if (gpsActivo) {
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
                    ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(Ubicame.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION}, 1);
            } else {
                listaNombresProveedor = locationManager.getProviders(true);
                listaNombresProveedor.forEach(p -> listaProveedor.add(new
EntProveedor(p)));
                Criteria criterio = new Criteria();
                criterio.setAccuracy(Criteria.ACCURACY_FINE);
                String Proveedor = locationManager.getBestProvider(criterio, false);

                //-- Para el caso de FUSED_PROVIDER se usa por defecto en el mejor
criterio.
                //-- Seleccion de pruebas, de forma indicativa se trabaja con Network
y GPS:
                //-- String Proveedor = LocationManager.NETWORK_PROVIDER;
                //-- String Proveedor = LocationManager.GPS_PROVIDER;
                //-- *** ---
```

```
                if (Proveedor != null) {
                    Optional<EntProveedor> rtnValor =
listaProveedor.stream().filter(p ->
p.nombreProveedor.contains(Proveedor)).findFirst();
                    rtnValor.ifPresent(p -> p.esMejor = true);
                    locationManager.requestLocationUpdates(Proveedor, 1000, 1,
locationListener);
                    UltimaLocalizacion =
locationManager.getLastKnownLocation(Proveedor);
                    listaDeSensores =
sensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER);


                    //-- Solo se ha usado para evaluar los valores con el CallBack
mGnssMeasurementsListener
                    //--
locationManager.registerGnssMeasurementsCallback(mGnssMeasurementsListener);


                    //-- Informacion del los satelites GNSS. String Proveedor="gps"
para controlar la info del estado.
                    //-- Como con la disposicion actual (FUSED_PROVIDER) solo se
aplica de forma optimizada, no se visualiza.
                    if(Proveedor.contains(LocationManager.GPS_PROVIDER)){

locationManager.registerGnssStatusCallback(mGnssStatusCallback);
                    }
                    //-- Administracion de proveedor de locacion.
                    ProveedorLocationManager =
locationManager.getProvider(Proveedor);


                    if (listaDeSensores.size() > 0) {
                        Acelerometro =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
                        Magnetometro =
sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
                        if (Acelerometro != null && Magnetometro != null) {
                            sensorManager.registerListener(sensorEventListener,
Acelerometro, SensorManager.SENSOR_DELAY_FASTEST);
                            sensorManager.registerListener(sensorEventListener,
Magnetometro, SensorManager.SENSOR_DELAY_FASTEST);
                            btnBusquedaActiva = true;
                        } else {
                            Toast.makeText(getApplicationContext(), "Este dispositivo
no cuenta con los sensores para posicionamiento", Toast.LENGTH_LONG).show();
                        }
                    } else {
                        Toast.makeText(getApplicationContext(), "Este dispositivo no
dispone de sensores", Toast.LENGTH_LONG).show();
                    }
                }
            }
        } else {
            Toast.makeText(getApplicationContext(), "No esta disponible el servicio
de localizacion", Toast.LENGTH_LONG).show();
        }
        if (btnBusquedaActiva && timerExplorador == null) {
            timerExplorador = new Timer();
            timerExplorador.scheduleAtFixedRate(new TimerTask() {
                public void run() {
```

```
                    ExplorarLocalizacion();
                }
            }, 0, Constantes.TIEMPO_TIMER);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults) {
        if (requestCode == 1) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(getApplicationContext(), "Se ha habilitado el acceso,
vuelva a intentarlo", Toast.LENGTH_LONG).show();
            }
        }
    }

    @Override
    public void onConfigurationChanged(@NonNull Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
        if (btnVamos.getVisibility() == View.VISIBLE) {
            if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE ||
                    newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
                DisplayMetrics metrics = new DisplayMetrics();
                getWindowManager().getDefaultDisplay().getMetrics(metrics);
                btnVamos.setX(metrics.widthPixels - btnVamos.getWidth());
                btnVamos.setY(metrics.heightPixels - (btnVamos.getHeight() * 2));
            }
        }
    }

    /**
     * @author ECHM/.
     * Activa la posicion visible del icono de lugar
     */
    private void visiblePosicionDispositivo(int cXImagen, int cYImagen) {
        rectImagenBrujula.top = cYImagen;
        rectImagenBrujula.left = cXImagen;
        rectImagenBrujula.bottom = cYImagen + imagenBrujula.getHeight();
        rectImagenBrujula.right = cXImagen + imagenBrujula.getWidth();

        rectImagenMarcador.top = cYImagen;
        rectImagenMarcador.left = cXImagen;
        rectImagenMarcador.bottom = cYImagen + imagenMarcador.getHeight();
        rectImagenMarcador.right = cXImagen + imagenMarcador.getWidth();
        vScroll.scrollTo(cXImagen - (hScroll.getWidth() / 2), cYImagen -
(vScroll.getHeight() / 2));
        hScroll.scrollTo(cXImagen - (hScroll.getWidth() / 2), cYImagen -
(vScroll.getHeight() / 2));
    }

    /**
     * @author ECHM/.
     * Activa la posicion visible del icono de la Facultad.
```

```
     */
    private void visiblePosicionLocal(int cXImagen, int cYImagen) {
        if (facultadActiva != null) {
            lytextoLocal.setText(facultadActiva.Corto);
            rectlyDatosLocal.top = cYImagen - lyDatosLocal.getHeight();
            rectlyDatosLocal.left = cXImagen - lyDatosLocal.getWidth();
            rectlyDatosLocal.bottom = cYImagen;
            rectlyDatosLocal.right = cXImagen;
            int PosicionXRelativa = cXImagen - (hScroll.getWidth() / 2);
            int PosicionYRelativa = cYImagen - (vScroll.getHeight() / 2);
            vScroll.scrollTo(PosicionXRelativa, PosicionYRelativa);
            hScroll.scrollTo(PosicionXRelativa, PosicionYRelativa);
            lyDatosLocal.setVisibility(View.VISIBLE);
        }
    }

    /**
     * @author ECHM/.
     * Explora la localizacion y la visualizacion en el mapa
     */
    private void ExplorarLocalizacion() {
        if (gpsActivo) {
            if (btnBusquedaActiva && UltimaLocalizacion != null) {
                //-- Obtiene las coordenadas X,Y referidas a la imagen del plano en
funcion de la ultima localizacion del dispositivo y posiciona la ubicacion
                //-- del dispositivo dentro del area visible del mapa en la pantalla
del dispositivo. Considere que las coordenadas validas son las que se encuentran
                //-- dentro de la imagen del plano.
                xyDerivada =
Mapacore.retornaCoordenadaXYConLatitudLongitudXInterseccion(UltimaLocalizacion.getLat
itude(), UltimaLocalizacion.getLongitude(), Configuracion);
                if (xyDerivada.X > 0 && xyDerivada.Y > 0) {
                    visiblePosicionDispositivo(xyDerivada.X, xyDerivada.Y);
                }
                //-- *** --
                if (azimut != 0 && facultadActiva != null) {
                    //-- Direccion o rumbo hacia el destino respecto a la ultima
localizacion
                    Location gpsDestino = new Location("gpsDestino");
                    gpsDestino.setLatitude(facultadActiva.Latitud);
                    gpsDestino.setLongitude(facultadActiva.Longitud);
                    double direccionCoodenada =
Mapacore.direccionDesdeCoordenadaDeOrigen(UltimaLocalizacion, gpsDestino);
                    double desplazamiento =
Mapacore.rtnGradosDelHeadingDesplazados(((WindowManager)
getSystemService(Context.WINDOW_SERVICE)).getDefaultDisplay().getRotation());
                    if (direccionCoodenada < (Math.toDegrees(azimut) +
desplazamiento)) {
                        radianBrujula = (Math.PI * (360 - Math.abs(direccionCoodenada
- (Math.toDegrees(azimut) + desplazamiento))) / 180.0);
                    } else {
                        radianBrujula = (Math.PI * (direccionCoodenada -
(Math.toDegrees(azimut) + desplazamiento)) / 180.0);
                    }
                    if (xyDerivada.X > 0 && xyDerivada.Y > 0) {
                        String ProveedorMejor = "";
```

```
                        Optional<EntProveedor> rtnValor =
listaProveedor.stream().filter(p -> p.esMejor).findFirst();
                        if (rtnValor.isPresent()) {
                            ProveedorMejor = rtnValor.get().nombreProveedor;
                        }
                        //-- UltimaLocalizacion.bearingTo(). Devuelve el rumbo
inicial aproximado en grados al este del norte verdadero cuando se viaja por la ruta
                        //-- más corta entre esta ubicación y la ubicación dada. El
camino más corto se define utilizando el elipsoide WGS84.
                        tituloAproximacion = String.format(Locale.getDefault(),
"Prov.: %s. Dist.: %.1f m. Lat.: %f, Long.: %f, Bear. Dest.: %f °", ProveedorMejor,
                                UltimaLocalizacion.distanceTo(gpsDestino),
UltimaLocalizacion.getLatitude(), UltimaLocalizacion.getLongitude(),
                                UltimaLocalizacion.bearingTo(gpsDestino));
                        //-- Requerimientos de acceso con el proveedor
                        if(ProveedorLocationManager!=null){
                            tituloAproximacion += String.format("\nReq. acceso RED
Celular: %s, Req. RED local: %s, Req. Satelite: %s",
ProveedorLocationManager.requiresCell(),

ProveedorLocationManager.requiresNetwork(),ProveedorLocationManager.requiresSatellite
());
                        }
                        //-- tituloAproximacion += String.format("\n Medidas GNSS no
proc. %s", resumenMedicionesGNSSSinProcesar);
                        if(ProveedorMejor.contains(LocationManager.GPS_PROVIDER)){
                            tituloAproximacion += String.format("\nSatelites GNSS
%s", datosEstadoGNSS);
                        }
                    } else {
                        tituloAproximacion = "";
                    }
                }
            }
        }
        runOnUiThread(this::ActualizaScreen);
    }

    /**
     * @author ECHM/.
     * Actualiza la pantalla
     */
    public void ActualizaScreen() {
        imagenMapa.getLocalVisibleRect(rectImagenMapa);
        if (btnBusquedaActiva && UltimaLocalizacion != null) {
            if (xyDerivada.X > 0 && xyDerivada.Y > 0 &&
Rect.intersects(rectImagenMapa, rectImagenBrujula)) {
                imagenBrujula.setRotation((float) Math.toDegrees(radianBrujula));
                imagenBrujula.setVisibility(View.VISIBLE);
                imagenBrujula.setX(hScroll.getLeft());
                imagenBrujula.setY(hScroll.getTop());

                imagenMarcador.setRotation((float) Math.toDegrees(radianBrujula));
                imagenMarcador.setVisibility(View.VISIBLE);
                imagenMarcador.setX((float) (rectImagenMarcador.left -
(imagenMarcador.getWidth() / 2.0) - rectImagenMapa.left));
                imagenMarcador.setY((float) (rectImagenMarcador.top -
(imagenMarcador.getHeight() / 2.0) - rectImagenMapa.top));
```

```
            } else {
                imagenBrujula.setVisibility(View.INVISIBLE);
                imagenMarcador.setVisibility(View.INVISIBLE);
            }
        } else {
            imagenBrujula.setVisibility(View.INVISIBLE);
            imagenMarcador.setVisibility(View.INVISIBLE);
        }
        if (facultadActiva != null) {
            if (Rect.intersects(rectImagenMapa, rectlyDatosLocal)) {
                lyDatosLocal.setVisibility(View.VISIBLE);
                lyDatosLocal.setX((20 + rectlyDatosLocal.left -
rectImagenMapa.left));
                lyDatosLocal.setY((20 + rectlyDatosLocal.top - rectImagenMapa.top));
            } else {
                lyDatosLocal.setVisibility(View.INVISIBLE);
            }
        } else {
            lyDatosLocal.setVisibility(View.INVISIBLE);
        }
        if (btnBusquedaActiva) {
            textoDistancia.setText(tituloAproximacion);
            textoDistancia.setVisibility(View.VISIBLE);
            textoDistancia.setX(hScroll.getLeft());
            textoDistancia.setY(vScroll.getHeight() - textoDistancia.getHeight() -
10);
        } else {
            textoDistancia.setVisibility(View.INVISIBLE);
        }
    }

    /**
     *  @author ECHM/.
     * Control del boton Vamos con busqueda activa
     */
    private void fncBtnVamos(boolean activo) {
        if (activo) {
            DisplayMetrics metrics = new DisplayMetrics();
            getWindowManager().getDefaultDisplay().getMetrics(metrics);
            btnVamos.setX(metrics.widthPixels - btnVamos.getWidth());
            btnVamos.setY(metrics.heightPixels - (btnVamos.getHeight() * 2));
            btnVamos.setVisibility(View.VISIBLE);
        } else {
            btnVamos.setVisibility(View.INVISIBLE);
        }
    }

    /**
     * @author ECHM/.
     * Ejecucion en Background la carga de locales. Se suprime la advertencia por
solo es interna
     */
    static class CargaDatosDeLocales extends AsyncTask<Object, Object, Object> {
         @SuppressLint("StaticFieldLeak")
```

```
         private final Context contextoRef;
        @SuppressLint("StaticFieldLeak")
        private final ProgressBar barraprogresoRef;
        private ArrayList<EntLocal> listaItemLViewVarLocal;
        private final ArrayList<EntLocal> listaItemLViewRef;
        EntConfiguracion configuracionRef;
        public CargaDatosDeLocales(Context contexto, ProgressBar barraprogreso,
ArrayList<EntLocal> listaItemLView, EntConfiguracion Configuracion){
            contextoRef =contexto;
            barraprogresoRef =barraprogreso;
            listaItemLViewRef =listaItemLView;
            configuracionRef =Configuracion;
        }

        /**
         * @author ECHM/.
         * Carga los parametros.
         */
        @Override
        protected Object doInBackground(Object... params) {
            Conecta vConecta = new Conecta(contextoRef);
            listaItemLViewVarLocal = vConecta.leerLocales(0);
            vConecta.leerConfiguracion(configuracionRef);
            vConecta.cierraDB();
            return true;
        }

        /**
         * @author ECHM/.
         * Cierra la carga
         */
        @Override
        protected void onPostExecute(Object result) {
            if (barraprogresoRef != null) {
                if (barraprogresoRef.getVisibility() == View.VISIBLE) {
                    barraprogresoRef.setVisibility(View.GONE);
                }
            }
            if (listaItemLViewVarLocal.size()>0){
                listaItemLViewRef.clear();
                listaItemLViewRef.addAll(listaItemLViewVarLocal);
            }else
                Toast.makeText(contextoRef,"No existe disponibilidad de datos",
Toast.LENGTH_LONG).show();
        }
    }

    /**
     * @author ECHM/.
     * Ejecucion de gestos de la imagen
     */
    private static class GestosListener extends SimpleOnGestureListener {
        @Override
        public boolean onSingleTapUp(MotionEvent e) {
```

```
            return true;
        }

        @Override
        public boolean onDown(MotionEvent e) {
            return true;
        }

        @Override
        public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float
velocityY) {
            int SWIPE_MIN_DISTANCE = 120;
            int SWIPE_THRESHOLD_VELOCITY = 200;
            if (e1.getX() - e2.getX() > SWIPE_MIN_DISTANCE && Math.abs(velocityX) >
SWIPE_THRESHOLD_VELOCITY) {
                return true;
            } else if (e2.getX() - e1.getX() > SWIPE_MIN_DISTANCE &&
Math.abs(velocityX) > SWIPE_THRESHOLD_VELOCITY) {
                return true;
            }
            if (e1.getY() - e2.getY() > SWIPE_MIN_DISTANCE && Math.abs(velocityY) >
SWIPE_THRESHOLD_VELOCITY) {
                return true;
            } else
                return e2.getY() - e1.getY() > SWIPE_MIN_DISTANCE &&
Math.abs(velocityY) > SWIPE_THRESHOLD_VELOCITY;
        }
    }
}
```

3.3. Package: echm.db

1.3.1. Clase : Conecta

```
package echm.db;

import java.io.IOException;
import java.util.ArrayList;
import android.content.Context;
import android.database.*;
import echm.entidades.EntCategoria;
import echm.entidades.EntConfiguracion;
import echm.entidades.EntLocal;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Capa de datos.
 *
 */
public class Conecta {
    CoreDB corDB;
    public boolean RtnConectaEstado;

    /**
     * @author ECHM/.
     * Constructor
     */
    public Conecta(Context context) {
        RtnConectaEstado=false;
        corDB = new CoreDB(context);
        try {
            corDB.crearBaseDeDatos();
            RtnConectaEstado=true;
        } catch (IOException ioe) {
            throw new Error("No se puede crear la Base de Datos");
        }
        corDB.abrirBaseDeDatos();
    }
```

```
    /**
     * @author ECHM/.
     * Cierra la Base de datos
     */
    public void cierraDB() {
        corDB.close();
    }

    /**
     * @author ECHM/.
     * Retorna la configuracion del plano.
     */
    public void leerConfiguracion(EntConfiguracion configuracion) throws SQLException
{
        Cursor mCursor = corDB.DB.rawQuery("SELECT * FROM Configuracion", null);
        if (mCursor != null) {
            if (mCursor.moveToFirst()) {
                configuracion.IdConfiguracion =
mCursor.getInt(mCursor.getColumnIndex("IdConfiguracion"));
                configuracion.Nombre =
mCursor.getString(mCursor.getColumnIndex("Nombre"));
                configuracion.PolLat1 =
mCursor.getDouble(mCursor.getColumnIndex("PolLat1"));
                configuracion.PolLon1 =
mCursor.getDouble(mCursor.getColumnIndex("PolLon1"));
                configuracion.PolLat2 =
mCursor.getDouble(mCursor.getColumnIndex("PolLat2"));
                configuracion.PolLon2 =
mCursor.getDouble(mCursor.getColumnIndex("PolLon2"));
                configuracion.PolLat3 =
mCursor.getDouble(mCursor.getColumnIndex("PolLat3"));
                configuracion.PolLon3 =
mCursor.getDouble(mCursor.getColumnIndex("PolLon3"));
                configuracion.PolLat4 =
mCursor.getDouble(mCursor.getColumnIndex("PolLat4"));
                configuracion.PolLon4 =
mCursor.getDouble(mCursor.getColumnIndex("PolLon4"));
                configuracion.PoGeoUtmRelCuadN1 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmRelCuadN1"));
                configuracion.PoGeoUtmRelCuadE1 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmRelCuadE1"));
                configuracion.PoGeoUtmRelCuadN2 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmRelCuadN2"));
                configuracion.PoGeoUtmReCuadlE2 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmReCuadlE2"));
                configuracion.PoGeoUtmRelCuadN3 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmRelCuadN3"));
                configuracion.PoGeoUtmReCuadlE3 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmReCuadlE3"));
                configuracion.PoGeoUtmRelCuadN4 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmRelCuadN4"));
                configuracion.PoGeoUtmRelCuadE4 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmRelCuadE4"));
                configuracion.PoGeoUtmN1 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmN1"));
                configuracion.PoGeoUtmE1 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmE1"));
                configuracion.PoGeoUtmN2 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmN2"));
                configuracion.PoGeoUtmE2 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmE2"));
```

```
                configuracion.PoGeoUtmN3 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmN3"));
                configuracion.PoGeoUtmE3 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmE3"));
                configuracion.PoGeoUtmN4 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmN4"));
                configuracion.PoGeoUtmE4 =
mCursor.getDouble(mCursor.getColumnIndex("PoGeoUtmE4"));
                configuracion.ImgAnchoPixel =
mCursor.getInt(mCursor.getColumnIndex("ImgAnchoPixel"));
                configuracion.ImgAltoPixel =
mCursor.getInt(mCursor.getColumnIndex("ImgAltoPixel"));
            }
            mCursor.close();
        }
    }

    /**
     * @author ECHM/.
     * Retorna la lista de categorias.
     * @return Retorna la lista de categorias
     */
    public ArrayList<EntCategoria> leerCategoria() {
        ArrayList<EntCategoria> ArrCategoria = new ArrayList<>();
        Cursor mCursor = corDB.DB.rawQuery("SELECT * FROM Categoria where Estado=1",
null);
        if (mCursor != null) {
            if (mCursor.moveToFirst()) {
                do {
                    EntCategoria categoria = new EntCategoria();
                    categoria.IdCategoria =
mCursor.getInt(mCursor.getColumnIndex("IdCategoria"));
                    categoria.Descripcion =
mCursor.getString(mCursor.getColumnIndex("Descripcion"));
                    ArrCategoria.add(categoria);
                } while (mCursor.moveToNext());
            }
            mCursor.close();
        }
        return ArrCategoria;
    }

    /**
     * @author ECHM/.
     * Retorna la facultad.
     */
    public void leerLocal(int idlocal, EntLocal local) {
        Cursor mCursor = corDB.DB.rawQuery(String.format("SELECT * FROM Local where
Estado=1 And IdLocal=%s", idlocal), null);
        if (mCursor != null) {
            if (mCursor.moveToFirst()) {
                local.Latitud = 0.0;
                local.Longitud = 0.0;
                local.IdLocal = mCursor.getInt(mCursor.getColumnIndex("IdLocal"));
                local.IdCategoria =
mCursor.getInt(mCursor.getColumnIndex("IdCategoria"));
                local.Nombre = mCursor.getString(mCursor.getColumnIndex("Nombre"));
```

```
                local.Descripcion =
mCursor.getString(mCursor.getColumnIndex("Descripcion"));
                local.Corto = mCursor.getString(mCursor.getColumnIndex("Corto"));
                local.NroLocal =
mCursor.getString(mCursor.getColumnIndex("NroLocal"));
                local.Latitud = mCursor.getDouble(mCursor.getColumnIndex("Latitud"));
                local.Longitud =
mCursor.getDouble(mCursor.getColumnIndex("Longitud"));
                local.Telefono =
mCursor.getString(mCursor.getColumnIndex("Telefono"));
                local.icoVisible = false;
                local.enMapaAddSubView = false;
            }
            mCursor.close();
        }
    }

    /**
     * @author ECHM/.
     * Retorna la lista de facultades.
     * @return Retorna la lista de facultades
     */
    public ArrayList<EntLocal> leerLocales(int idcategoria) {
        ArrayList<EntLocal> ArrFacultad = new ArrayList<>();
        Cursor mCursor = corDB.DB.rawQuery(String.format("SELECT * FROM Local where
Estado=1 And (IdCategoria=%s Or %s=0)", idcategoria , idcategoria), null);
        if (mCursor != null) {
            if (mCursor.moveToFirst()) {
                do {
                    EntLocal local = new EntLocal();
                    local.Latitud = 0.0;
                    local.Longitud = 0.0;
                    local.IdLocal =
mCursor.getInt(mCursor.getColumnIndex("IdLocal"));
                    local.IdCategoria =
mCursor.getInt(mCursor.getColumnIndex("IdCategoria"));
                    local.Nombre =
mCursor.getString(mCursor.getColumnIndex("Nombre"));
                    local.Descripcion =
mCursor.getString(mCursor.getColumnIndex("Descripcion"));
                    local.Corto = mCursor.getString(mCursor.getColumnIndex("Corto"));
                    local.NroLocal =
mCursor.getString(mCursor.getColumnIndex("NroLocal"));
                    local.Latitud =
mCursor.getDouble(mCursor.getColumnIndex("Latitud"));
                    local.Longitud =
mCursor.getDouble(mCursor.getColumnIndex("Longitud"));
                    local.Telefono =
mCursor.getString(mCursor.getColumnIndex("Telefono"));
                    local.icoVisible = false;
                    local.enMapaAddSubView = false;
                    ArrFacultad.add(local);
                } while (mCursor.moveToNext());
            }
            mCursor.close();
        }
        return ArrFacultad;
```

```
    }
}
```

3.3.2. Clase: CoreDB

```
package echm.db;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.file.Files;
import java.nio.file.Paths;
import android.content.Context;
import android.database.SQLException;
import android.database.sqlite.*;
import com.echm.geolocation.R;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Modulo: CoreDB.
 * Soporte de conexion a base de datos SQLITE.
 *
 */
public class CoreDB extends SQLiteOpenHelper {
    Context localContext;
    public SQLiteDatabase DB;

    public CoreDB(Context context) {
        super(context, context.getResources().getString(R.string.db_nombre), null,
Integer.parseInt(context.getString(R.string.db_version)));
        localContext = context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }

    /**
```

```
     * @author ECHM/.
     * Crea la base de datos
     */
    public void crearBaseDeDatos() throws IOException {
        if (!verificaBaseDeDatos()) {
            this.getReadableDatabase();
            try {
                copiarBaseDeDatos();
            } catch (IOException e) {
                throw new Error(e.getMessage());
            }
        }
    }

    /**
     * @author ECHM/.
     * Verifica la existencia de la Base de datos
     * @return Estadp de la transaccion
     */
    private boolean verificaBaseDeDatos() {
        SQLiteDatabase checkDB = null;
        String rutaDB;
        try {
            rutaDB = localContext.getResources().getString(R.string.db_path) +
localContext.getResources().getString(R.string.db_nombre);
            checkDB = SQLiteDatabase.openDatabase(rutaDB, null,
SQLiteDatabase.OPEN_READWRITE);
            checkDB.close();
        } catch (SQLiteException e) {
            if (checkDB != null) checkDB.close();
        }
        return checkDB != null;
    }

    /**
     * @author ECHM/.
     * Copiar la base de datos a repositorio
     */
    private void copiarBaseDeDatos() throws IOException {
        InputStream myInput =
localContext.getAssets().open(localContext.getResources().getString(R.string.db_nombr
e));
        String rutaDB = localContext.getResources().getString(R.string.db_path) +
localContext.getResources().getString(R.string.db_nombre);
        OutputStream myOutput = Files.newOutputStream(Paths.get(rutaDB));
        byte[] buffer = new byte[1024];
        int length;
        while ((length = myInput.read(buffer)) > 0) {
            myOutput.write(buffer, 0, length);
        }
        myOutput.flush();
        myOutput.close();
        myInput.close();
    }
```

```
    /**
     * @author ECHM/.
     * Abre la Base de datos
     */
    public void abrirBaseDeDatos() throws SQLException {
        try {
            String rutaDB = localContext.getResources().getString(R.string.db_path) +
localContext.getResources().getString(R.string.db_nombre);
            DB = SQLiteDatabase.openDatabase(rutaDB, null,
SQLiteDatabase.OPEN_READWRITE);
        } catch (SQLException e) {
            throw new Error(e.getMessage());
        }
    }

    /**
     * @author ECHM/.
     * Cierra la base de datos
     */
    @Override
    public synchronized void close() {
        if (DB != null)
            DB.close();
        super.close();
    }
}
```

3.4. Package: echm.generales

3.4.1. Clase: ImagenCore

```
package echm.generales;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Rect;
import com.echm.geolocation.R;
import java.io.IOException;
import java.io.InputStream;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * ImagenCore
 *
 */
public class ImagenCore {

    /**
     * @author ECHM/.
     * Carga la imagen de los mapas.
     */
    public Bitmap CargaImagenMapa(Context context) {
        Bitmap BitMapMapa=null;
        InputStream imageIS;
        imageIS =
context.getResources().openRawResource(R.raw.ciudaduniversitariaoptvertical);
        BitmapFactory.Options options = new BitmapFactory.Options();
        options.inJustDecodeBounds = true;
        options.inPreferredConfig = Bitmap.Config.RGB_565;
        BitmapFactory.decodeStream(imageIS, null, options);
        try {
            imageIS.reset();
            options.inJustDecodeBounds = false;
            options.inSampleSize = 1;
            options.inPreferredConfig = Bitmap.Config.RGB_565;
```

```
            Rect outRect = new Rect(-1, -1, -1, -1);
            BitMapMapa = BitmapFactory.decodeStream(imageIS, outRect, options);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return BitMapMapa;
    }
}
```

3.4.2. Clase: MapaCore

```
package echm.generales;

import echm.entidades.EntConfiguracion;
import echm.entidades.EntCoordenada;
import android.location.Location;
import android.view.Surface;

/**
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @date 2023-06-01
 * @apiNote Mapcore
 * @since SDK 29.0
 */
public class Mapacore {
    /**
     * @return EntCoordenada    Distancia en posicion de imagen en el plano, hay que
considerar que es una imagen
     * @author ECHM/.
     * @param    Latitud Latitud
     * @param    Longitud Longitud
     * @param    Configuracion    Coordenadas del poligono
     */
    public static EntCoordenada
retornaCoordenadaXYConLatitudLongitudXInterseccion(double Latitud, double Longitud,
EntConfiguracion Configuracion) {
        EntCoordenada xyDerivada = new EntCoordenada(0, 0);
        //- Limite 1
        Location gpsInferiorIzquierdo = new Location("gpsInferiorIzquierdo");
        gpsInferiorIzquierdo.setLatitude(Configuracion.PolLat1);
        gpsInferiorIzquierdo.setLongitude(Configuracion.PolLon1);
        //- Limite 2
        Location gpsInferiorDerecho = new Location("gpsInferiorDerecho");
        gpsInferiorDerecho.setLatitude(Configuracion.PolLat1);
        gpsInferiorDerecho.setLongitude(Configuracion.PolLon2);
        //- Limite 3
        Location gpsSuperiorDerecho = new Location("gpsSuperiorDerecho");
```

```
        gpsSuperiorDerecho.setLatitude(Configuracion.PolLat3);
        gpsSuperiorDerecho.setLongitude(Configuracion.PolLon3);
        //- Limite 4
        Location gpsSuperiorIzquierdo = new Location("gpsSuperiorIzquierdo");
        gpsSuperiorIzquierdo.setLatitude(Configuracion.PolLat4);
        gpsSuperiorIzquierdo.setLongitude(Configuracion.PolLon4);

        Location gpsTriangulacion = new Location("gpsTriangulacion");
        gpsTriangulacion.setLatitude(Latitud);
        gpsTriangulacion.setLongitude(Longitud);

        // Respecto al norte geografico  o Norte terrestre (True heading).
        double direccionCoodenadaYPivot =
direccionDesdeCoordenadaDeOrigen(gpsSuperiorIzquierdo, gpsInferiorIzquierdo);
        double direccionCoodenadaXPivot =
direccionDesdeCoordenadaDeOrigen(gpsSuperiorIzquierdo, gpsSuperiorDerecho);
        double direccionCoodenadaY =
direccionDesdeCoordenadaDeOrigen(gpsInferiorIzquierdo, gpsSuperiorIzquierdo);
        double direccionCoodenadaX =
direccionDesdeCoordenadaDeOrigen(gpsSuperiorDerecho, gpsSuperiorIzquierdo);

        Location coordenadasCalculadaX =
interseccionDeCoordenadasConDireccion(direccionCoodenadaYPivot, gpsSuperiorIzquierdo,
gpsTriangulacion, direccionCoodenadaX);
        Location coordenadasCalculadaY =
interseccionDeCoordenadasConDireccion(direccionCoodenadaXPivot, gpsSuperiorIzquierdo,
gpsTriangulacion, direccionCoodenadaY);

        Location gpsLadoX = new Location("gpsLadoX");
        gpsLadoX.setLatitude(coordenadasCalculadaX.getLatitude());
        gpsLadoX.setLongitude(coordenadasCalculadaX.getLongitude());

        Location gpsLadoY = new Location("gpsLadoY");
        gpsLadoY.setLatitude(coordenadasCalculadaY.getLatitude());
        gpsLadoY.setLongitude(coordenadasCalculadaY.getLongitude());
        // Es el Ancho orientacion Norte coeficiente: Ancho imagen / Longitud metros
del ancho.
        double factorX = Configuracion.ImgAnchoPixel /
(Configuracion.PoGeoUtmRelCuadE1 - Configuracion.PoGeoUtmReCuadlE2);
        // Es el Alto orientacion Norte : Alto imagen / Longitud metros del alto.
        double factorY = Configuracion.ImgAltoPixel /
(Configuracion.PoGeoUtmRelCuadN4 - Configuracion.PoGeoUtmRelCuadN1);
        xyDerivada.X = (int) (gpsTriangulacion.distanceTo(gpsLadoX) * factorX);
        xyDerivada.Y = (int) (gpsTriangulacion.distanceTo(gpsLadoY) * factorY);

        return xyDerivada;
    }

    /**
     * @author ECHM/.
     * Es la interseccion entre 2 puntos dado el grado de inicio o salida.-><-
     */
    public static Location interseccionDeCoordenadasConDireccion(double
inicialDireccion1, Location partida1, Location partida2, double inicialDireccion2) {
        Location interseccion = new Location("interseccion");
        double lat1 = Math.toRadians(partida1.getLatitude());
        double lon1 = Math.toRadians(partida1.getLongitude());
```

```
        double lat2 = Math.toRadians(partida2.getLatitude());
        double lon2 = Math.toRadians(partida2.getLongitude());

        double brng13 = Math.toRadians(inicialDireccion1);
        double brng23 = Math.toRadians(inicialDireccion2);
        double dLat = lat2 - lat1;
        double dLon = lon2 - lon1;
        double dist12 = 2.0 * Math.asin(Math.sqrt(Math.sin(dLat / 2.0) *
Math.sin(dLat / 2.0) + Math.cos(lat1) * Math.cos(lat2) * Math.sin(dLon / 2.0) *
Math.sin(dLon / 2.0)));

        if (dist12 == 0.0) return partida1;
        double bearingA = Math.acos((Math.sin(lat2) - Math.sin(lat1) *
Math.cos(dist12)) / (Math.sin(dist12) * Math.cos(lat1)));
        if (Double.isNaN(bearingA)) bearingA = 0.0;
        double bearingB = Math.acos((Math.sin(lat1) - Math.sin(lat2) *
Math.cos(dist12)) / (Math.sin(dist12) * Math.cos(lat2)));
        double brng12, brng21;
        if (Math.sin(dLon) > 0.0) {
            brng12 = bearingA;
            brng21 = 2.0 * Math.PI - bearingB;
        } else {
            brng12 = 2.0 * Math.PI - bearingA;
            brng21 = bearingB;
        }
        double alpha1 = Math.IEEEremainder((brng13 - brng12 + Math.PI), (2.0 *
Math.PI)) - Math.PI;
        double alpha2 = Math.IEEEremainder((brng21 - brng23 + Math.PI), (2.0 *
Math.PI)) - Math.PI;
        if (Math.sin(alpha1) == 0.0 && Math.sin(alpha2) == 0.0) {
            interseccion.setLatitude(Math.toDegrees(-91.0));
            interseccion.setLongitude(Math.toDegrees(-181.0));
            return interseccion;
        }
        if (Math.sin(alpha1) * Math.sin(alpha2) < 0.0) {
            interseccion.setLatitude(Math.toDegrees(-91.0));
            interseccion.setLongitude(Math.toDegrees(-181.0));
            return interseccion;
        }

        double alpha3 = Math.acos(-Math.cos(alpha1) * Math.cos(alpha2) +
                Math.sin(alpha1) * Math.sin(alpha2) * Math.cos(dist12));
        double dist13 = Math.atan2(Math.sin(dist12) * Math.sin(alpha1) *
Math.sin(alpha2),
                Math.cos(alpha2) + Math.cos(alpha1) * Math.cos(alpha3));
        double lat3 = Math.asin(Math.sin(lat1) * Math.cos(dist13) +
                Math.cos(lat1) * Math.sin(dist13) * Math.cos(brng13));
        double dLon13 = Math.atan2(Math.sin(brng13) * Math.sin(dist13) *
Math.cos(lat1),
                Math.cos(dist13) - Math.sin(lat1) * Math.sin(lat3));
        double lon3 = lon1 + dLon13;
        lon3 = Math.IEEEremainder((lon3 + Math.PI), (2.0 * Math.PI)) - Math.PI;

        interseccion.setLatitude(Math.toDegrees(lat3));
        interseccion.setLongitude(Math.toDegrees(lon3));
        return interseccion;
```

```
    }

    /**
     * @author ECHM/.
     * ECHM/. Retorna en grados la direccion o rumbo (Bearing) entre 2 puntos respecto al norte geografico o norte verdadero. TRUE HEADING.
     * Se usa el meridiano como cero grados, hacia el lado izquierdo del meridiano es negativo , hacia el lado derecho del meridiano es positivo.
     */
    public static double direccionDesdeCoordenadaDeOrigen(Location origen, Location destino) {
        double lat1 = Math.toRadians(origen.getLatitude());
        double lon1 = Math.toRadians(origen.getLongitude());
        double lat2 = Math.toRadians(destino.getLatitude());
        double lon2 = Math.toRadians(destino.getLongitude());
        double dLon = lon2 - lon1;
        double dPhi = Math.log(Math.tan(lat2 / 2 + (Math.PI / 4)) / Math.tan(lat1 / 2 + (Math.PI / 4)));

        if (Math.abs(dLon) > Math.PI) {
            dLon = (dLon > 0.0) ? -(2.0 * Math.PI - dLon) : (2.0 * Math.PI + dLon);
        }
        //-- Retorna el angulo de la conversion coordenadas X,Y a coordenadas polares
        //-- Arco tangente de y/x = Sen(ß)/Cos(ß) en el rango de -pi to pi entre el 1er y 4to cuadrante
        //-- con la funcion atan2(). Retorna el angulo en radianes
        double direccion = Math.atan2(dLon, dPhi);
        return Math.IEEEremainder((Math.toDegrees(direccion) + 360.0), 360.0);
    }

    /**
     * @author ECHM/.
     * Retorna el desplazamiento en grados LANDSCAPE - PORTRAIT con respecto a la rotacion del dispositivo.
     */
    public static double rtnGradosDelHeadingDesplazados(int Rotacion) {
        double Desplazamiento = 0;

        switch (Rotacion) {
            case Surface.ROTATION_90:
                Desplazamiento = 90.0;
                break;
            case Surface.ROTATION_270:
                Desplazamiento = -90.0;
                break;
        }
        return Desplazamiento;
    }
}
```

3.5. Package: echm.entidades

3.5.1. Clase: AdaptadorListaCategoria

```
package echm.entidades;

import java.util.ArrayList;
import com.echm.geolocation.R;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Adaptador de la lista de categoria.
 *
 */
public class AdaptadorListaCategoria extends  ArrayAdapter<EntCategoria>{
          private final ArrayList<EntCategoria> itemsLista;
          private final Context lcontexto;

          public AdaptadorListaCategoria(Context context, int textViewResourceId,
ArrayList<EntCategoria> itemsLista)
          {
           super(context, textViewResourceId, itemsLista);
           lcontexto=context;
           this.itemsLista = itemsLista;
          }

          @Override
          public View getView(int posicion, View convertView, ViewGroup parent)
          {
                 View v = convertView;
                 if (v == null) {
                   LayoutInflater vi =
(LayoutInflater)lcontexto.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
```

```
                  v = vi.inflate(R.layout.listaitemcategoria, parent,false);
                }
                EntCategoria o = itemsLista.get(posicion);
                if (o != null) {
                      TextView tt = v.findViewById(R.id.TituloItem);
                      if(tt != null){
                            tt.setText(o.Descripcion);
                      }
                }
                return v;
          }
}
```

3.5.2. Clase: AdaptadorListaItemSearch

```
package echm.entidades;

import java.util.ArrayList;
import android.content.Context;
import android.database.Cursor;
import android.widget.CursorAdapter;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import com.echm.geolocation.R;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Clase: Adaptador de la lista de Item en sugerencia de busqueda.
 *
 */
public class AdaptadorListaItemSearch extends CursorAdapter {
    private final ArrayList<EntLocal> itemsLista;
    private TextView text;

    public AdaptadorListaItemSearch(Context context, Cursor cursor,
ArrayList<EntLocal> itemsLista) {
        super(context, cursor, false);
        this.itemsLista = itemsLista;
    }

    @Override
    public void bindView(View arg0, Context arg1, Cursor arg2) {
```

```
        String Linea = "";
        if (itemsLista.get(arg2.getPosition()).Nombre != null) {
            Linea = itemsLista.get(arg2.getPosition()).Nombre;
        }
        text.setText(Linea);
    }

    @Override
    public View newView(Context arg0, Cursor arg1, ViewGroup arg2) {
        LayoutInflater inflater = (LayoutInflater)
arg0.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View view = inflater.inflate(R.layout.listaitemsearch, arg2, false);
        text = view.findViewById(R.id.TituloItem);
        return view;
    }
}
```

3.5.3. Clase: AdaptadorListaLocal

```
package echm.entidades;

import java.util.ArrayList;
import com.echm.geolocation.R;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Adaptador de la lista de Lista Facultad.
 *
 */
public class AdaptadorListaLocal extends ArrayAdapter<EntLocal> {
    private final ArrayList<EntLocal> itemsLista;
    private final Context lcontexto;

    public AdaptadorListaLocal(Context context, int textViewResourceId,
ArrayList<EntLocal> itemsLista) {
```

```
        super(context, textViewResourceId, itemsLista);
        lcontexto = context;
        this.itemsLista = itemsLista;
    }

    @Override
    public View getView(int posicion, View convertView, ViewGroup parent) {
        View v = convertView;
        if (v == null) {
            LayoutInflater vi = (LayoutInflater)
lcontexto.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            v = vi.inflate(R.layout.listaitemfacultad,parent,false);
        }
        EntLocal o = itemsLista.get(posicion);
        if (o != null) {
            ImageView vd = v.findViewById(R.id.imageItem);
            if (vd != null && o.imagenIcono != null) {
                vd.setImageBitmap(o.imagenIcono);
            }
            TextView tt0 = v.findViewById(R.id.TituloItem);
            if (tt0 != null) {
                tt0.setText(o.Nombre);
            }
            TextView tt1 = v.findViewById(R.id.DireccionItem);
            if (tt1 != null && o.NroLocal != null) {
                tt1.setText(String.format("Nro. local: %s",o.NroLocal));
            }
            TextView tt2 = v.findViewById(R.id.CoordenadasItem);
            if (tt2 != null && o.Latitud != null && o.Longitud != null) {
                tt2.setText(String.format("Latitud: %s - Longitud:
%s",o.Latitud,o.Longitud));
            }
        }
        return v;
    }
}
```

3.6. Package: echm.entidades

3.6.1. Clase: Constantes

```
package echm.entidades;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Entidad Constantes del sistema.
 *
 */
public class Constantes {
    public static final int TIEMPO_TIMER = 300;
}
```

3.6.2. Clase: EntCategoria

```
package echm.entidades;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Entidad Categoria.
 *
 */
public class EntCategoria {
    public int IdCategoria;
    public String Descripcion;
}
```

3.6.3. Clase: EntConfiguracion

```
package echm.entidades;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Entidad Configuracion.
 *
 */
public class EntConfiguracion {
    public int IdConfiguracion;
    public String Nombre;
    public double PolLat1;
    public double PolLon1;
    public double PolLat2;
    public double PolLon2;
    public double PolLat3;
    public double PolLon3;
    public double PolLat4;
    public double PolLon4;
    public double PoGeoUtmRelCuadN1;
    public double PoGeoUtmRelCuadE1;
    public double PoGeoUtmRelCuadN2;
    public double PoGeoUtmReCuadlE2;
    public double PoGeoUtmRelCuadN3;
    public double PoGeoUtmReCuadlE3;
    public double PoGeoUtmRelCuadN4;
    public double PoGeoUtmRelCuadE4;
    public double PoGeoUtmN1;
    public double PoGeoUtmE1;
    public double PoGeoUtmN2;
    public double PoGeoUtmE2;
    public double PoGeoUtmN3;
    public double PoGeoUtmE3;
    public double PoGeoUtmN4;
    public double PoGeoUtmE4;
    public double ImgAnchoPixel;
    public double ImgAltoPixel;
}
```

3.6.4. Clase: EntCoordenada

```
package echm.entidades;
```

```
/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Entidad Coordenadas.
 *
 */

public class EntCoordenada {
 public  int X;
 public  int Y;

 public EntCoordenada(int x, int y){
        this.X=x;
        this.Y=y;

 }
}
```

3.6.5. Clase: EntLocal

```
package echm.entidades;

import android.graphics.Bitmap;

/**
 *
 * @author Enrique Chala Mena
 * @author ECHM/.
 * @author www.enriquechala.com
 * @author Facultad de Ciencias Matematicas
 * @author Universidad Nacional Mayor de San Marcos
 * @version 1.0
 * @since SDK 29.0
 * @date 2023-06-01
 * @apiNote
 * Entidad Local.
 *
 */

@SuppressWarnings("unused")
public class EntLocal {
    public int IdLocal;
    public int IdCategoria;
    public String Nombre;
```

```
    public String Descripcion;
    public String Corto;
    public String Telefono;
    public String NroLocal;
    public float Cx;
    public float Cy;
    public Bitmap imagenIcono;
    public boolean icoVisible;
    public boolean enMapaAddSubView;
    public Double Latitud;
    public Double Longitud;
    public int cXImagen;
    public int cYImagen;
}
```

3.6.6. Clase: EntProveedor

```
package echm.entidades;

import android.location.Location;

@SuppressWarnings("unused")
public class EntProveedor {
    public String nombreProveedor;
    public int identificadorProveedor;
    public boolean esMejor;
    public Location locacionXOrigen;
    public Location ultimaLocalizacion;
    public float distancia;
    public boolean existe;

    public EntProveedor(String nombreProveedor){
        this.nombreProveedor=nombreProveedor;
    }
}
```

3.7. Layout

3.7.1. Activity_CategoriaLocal.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/SlideListaCategoriaslocales"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context=".CategoriaLocal" >
        <LinearLayout
                    android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
                    android:layout_alignParentTop="true"
                    android:paddingTop="10sp"
                    android:paddingBottom="10sp"
                    android:orientation="vertical">
                              <ListView
                                   android:id="@+id/listaCategorias"
                                   android:layout_width="fill_parent"
                                   android:layout_height="0dp"
                                   android:layout_marginBottom="10sp"
                                   android:layout_weight="1"
                                   android:padding="2sp" >
                         </ListView>
        </LinearLayout>
</RelativeLayout>
```

3.7.2. Activity_DetalleLocal.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/SlideDetalleLocal"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context=".DetalleLocal" >
 <HorizontalScrollView
    android:id="@+id/scrollHzUbicame"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ScrollView android:id="@+id/scrollVTUbicame"
       android:layout_width="wrap_content"
```

```
      android:layout_height="match_parent"
      android:layout_marginLeft="10sp"
      android:layout_marginRight="10sp"
      android:layout_marginTop="2sp"
      android:layout_marginBottom="2sp"
      android:orientation="vertical">
      <LinearLayout
             android:layout_width="fill_parent"
             android:layout_height="wrap_content"
             android:orientation="vertical" >
           <ImageView android:id="@+id/imagenMapa"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:focusableInTouchMode="true"
                       android:contentDescription="@string/title_img_mapa"/>
      </LinearLayout>
   </ScrollView>
</HorizontalScrollView>
 <ImageView
      android:id="@+id/imagenLugar"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:visibility="invisible"
      android:src="@drawable/ico_lugar"
         android:contentDescription="@string/title_img_lugar"/>
 <ImageView
      android:id="@+id/imagenBrujula"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:visibility="invisible"
      android:src="@drawable/trianguloruta"
         android:contentDescription="@string/title_img_brujula"/>
      <ImageView
             android:id="@+id/imagenMarcador"
             android:layout_width="wrap_content"
             android:layout_height="wrap_content"
             android:visibility="invisible"
             android:src="@drawable/punto_pos"
             android:contentDescription="@string/title_img_marcador"/>
<LinearLayout
    android:id="@+id/lyDatosFacultad"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:visibility="invisible"
    android:gravity="end"
    android:orientation="vertical"
       tools:ignore="UseCompoundDrawables">
             <TextView
                    android:id="@+id/lytextoFacultad"
                       style="@style/cssTextViewTituloTienda"
                    android:text=""
           />
                 <ImageView
```

```
                        android:id="@+id/lyimagenFacultad"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:src="@drawable/ico_tienda"
                        android:contentDescription="@string/title_img_lugar"/>
 </LinearLayout>
 <Button
       android:id="@+id/btnVamos"
       android:layout_width="80sp"
          android:layout_height="80sp"
          android:background="@drawable/selectorbtnir"
          android:visibility="invisible"/>
 <TextView
          android:id="@+id/textoDistancia"
       style="@style/cssTextViewTituloDistancia"
          android:text=""
          android:maxLines="4"
          android:visibility="invisible"/>
</RelativeLayout>
```

3.7.3. Activity_Locales.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/SlideListaLocales"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context=".Locales" >

        <LinearLayout
                    android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
                    android:layout_alignParentTop="true"
                    android:paddingTop="10sp"
                    android:paddingBottom="10sp"
                    android:orientation="vertical">
                              <ListView
                                   android:id="@+id/listaLocales"
                                   android:layout_width="fill_parent"
                                   android:layout_height="0dp"
                                   android:layout_marginBottom="10sp"
                                   android:layout_weight="1"
                                   android:padding="2sp" >
                         </ListView>
        </LinearLayout>

</RelativeLayout>
```

3.7.4. Activity_Ubicame.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/SlideUbicame"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context=".Ubicame" >
 <HorizontalScrollView
    android:id="@+id/scrollHzUbicame"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ScrollView android:id="@+id/scrollVTUbicame"
       android:layout_width="wrap_content"
       android:layout_height="match_parent"
       android:layout_marginLeft="10sp"
       android:layout_marginRight="10sp"
       android:layout_marginTop="2sp"
       android:layout_marginBottom="2sp"
       android:orientation="vertical">
       <LinearLayout
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:orientation="vertical" >
            <ImageView android:id="@+id/imagenMapa"
                 android:layout_width="fill_parent"
                 android:layout_height="wrap_content"
                 android:focusableInTouchMode="true"
                        android:contentDescription="@string/title_img_mapa"/>
       </LinearLayout>
    </ScrollView>
 </HorizontalScrollView>
 <ImageView
       android:id="@+id/imagenLugar"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:visibility="invisible"
       android:src="@drawable/ico_lugar"
          android:contentDescription="@string/title_img_lugar"/>
  <ImageView
       android:id="@+id/imagenBrujula"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:visibility="invisible"
       android:src="@drawable/trianguloruta"
          android:contentDescription="@string/title_img_brujula"/>
       <ImageView
              android:id="@+id/imagenMarcador"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:visibility="invisible"
              android:src="@drawable/punto_pos"
```

```
            android:contentDescription="@string/title_img_marcador"/>
 <LinearLayout
     android:id="@+id/lyDatosFacultad"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:visibility="invisible"
     android:gravity="end"
     android:orientation="vertical"
        tools:ignore="UseCompoundDrawables">
           <TextView
                  android:id="@+id/lytextoFacultad"
                     style="@style/cssTextViewTituloTienda"
                  android:text=""
            />
                 <ImageView
                      android:id="@+id/lyimagenFacultad"
                      android:layout_width="wrap_content"
                      android:layout_height="wrap_content"
                      android:src="@drawable/ico_tienda"
                      android:contentDescription="@string/title_img_lugar"/>
 </LinearLayout>
 <Button
       android:id="@+id/btnVamos"
       android:layout_width="64sp"
          android:layout_height="64sp"
          android:background="@drawable/selectorbtnir"
          android:visibility="invisible"/>
 <TextView
          android:id="@+id/textoDistancia"
       style="@style/cssTextViewTituloDistancia"
          android:text=""
          android:maxLines="8"
          android:visibility="invisible"/>
</RelativeLayout>
```

3.7.5. listaitemcategoria.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
     android:layout_width="fill_parent"
     android:layout_height="wrap_content"
     android:background="@drawable/selectoritemlist">
       <ImageView android:id="@+id/imageItem"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:layout_marginRight="10sp"
              android:layout_marginLeft="10sp"
              android:layout_marginTop="10sp"
              android:layout_marginBottom="10sp"
              android:src="@drawable/grupo_cat"
              android:contentDescription="@string/title_img_pin"/>
```

```
      <LinearLayout android:id="@+id/linearLayoutTextos"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="vertical">
                    <TextView android:text=""
                              android:id="@+id/TituloItem"
                              android:layout_width="fill_parent"
                              android:textStyle="bold"
                              style="@style/cssListTextViewItemTituloList"/>
         </LinearLayout>
</LinearLayout>
```

3.7.6. listaitemfacultad.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
     android:background="@drawable/selectoritemlist"
     android:layout_width="fill_parent"
     android:layout_height="wrap_content">
     <ImageView android:id="@+id/imageItem"
                 android:layout_width="wrap_content"
                 android:layout_height="wrap_content"
                 android:layout_marginRight="10sp"
                 android:layout_marginLeft="10sp"
                 android:layout_marginTop="10sp"
                 android:layout_marginBottom="10sp"
                 android:src="@drawable/locacion_marker"
                        android:contentDescription="@string/title_img_pin"/>
      <LinearLayout android:id="@+id/linearLayoutTextos"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
                    <TextView android:text=""
                              android:id="@+id/TituloItem"
                              android:layout_width="fill_parent"
                              android:layout_height="wrap_content"
                              android:textStyle="normal"
                              style="@style/cssListTextViewItemTituloInteriorList"/>
                    <TextView android:text=""
                              android:id="@+id/DireccionItem"
                              android:layout_width="fill_parent"
                              android:layout_height="wrap_content"
                              style="@style/cssListTextViewItemDetalleInteriorList"/>
                    <TextView android:text=""
                              android:id="@+id/CoordenadasItem"
                              android:layout_width="fill_parent"
                              android:layout_height="wrap_content"
                              style="@style/cssListTextViewItemDetalleInteriorList"/>
         </LinearLayout>
</LinearLayout>
```

3.7.7. listaitemsearch.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
     android:layout_width="fill_parent"
     android:layout_height="wrap_content"
     android:background="@drawable/selectoritemlist">
       <TextView android:text=""
              android:id="@+id/TituloItem"
              android:layout_width="fill_parent"
              android:textStyle="bold"
              style="@style/cssListTextViewItemTituloList"/>
</LinearLayout>
```

3.7.8. splash.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/EfectoSlideMain"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#FF9500"
    android:orientation="vertical" >
    <RelativeLayout
        android:id="@+id/contenedorlogocenter"
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="center" >
        <ImageView
            android:id="@+id/imageView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
            android:src="@drawable/centralsplashsvg"
            android:contentDescription="@string/title_img_splash"/>
    </RelativeLayout>
</LinearLayout>
```

3.8. Definiciones String

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Geolocation</string>
    <string name="strversion">1</string>
    <string name="db_nombre">dbuniversanmarcos</string>
    <string name="db_path">/data/data/com.echm.geolocation/databases/</string>
    <string name="db_version">1</string>
    <string name="menuroot">Geolocation</string>
    <string name="action_search">Buscar</string>
    <string name="action_categorias">Categorías</string>
    <string name="action_refresh">Recargar</string>
    <string name="titfooter_ubicame">Ubicame</string>
    <string name="titfooter_categorias">Categorias</string>
    <string name="titfooter_locales">Locales</string>
    <string name="title_activity_listadetallecategoria">Categoría</string>
    <string name="title_img_brujula">Brujula</string>
    <string name="title_img_marcador">Posicion</string>
    <string name="title_img_lugar">Posicion</string>
    <string name="title_img_mapa">Mapa</string>
    <string name="title_img_pin">Pin</string>
    <string name="title_img_splash">Geolocation</string>
    <string name="title_activity_categorialocal">Categorias</string>
    <string name="title_activity_locales">Locales</string>
    <string name="title_activity_detallelocal">Local</string>
    <string name="title_activity_ubicame">Ubicame</string>
    <string name="searchInput">Buscar locales</string>
</resources>
```

3.9. Hoja de estilos CSS

```
<resources xmlns:android="http://schemas.android.com/apk/res/android">
    <style name="AdnTemaApp" parent="@android:style/Theme.Holo">
       <item
name="android:dropDownListViewStyle">@style/ActionBarEstiloDropDown</item>
       <item
name="android:spinnerDropDownItemStyle">@style/ActionBarEstiloDropDownSpinner</item>
       <item name="android:actionBarStyle">@style/xActionBar</item>
    </style>
    <style name="SplashTema" parent="@android:style/Theme.Holo">
       <item name="android:windowNoTitle">true</item>
    </style>
    <style name="ActionBarEstiloDropDownSpinner"
parent="@android:style/Widget.Holo.DropDownItem.Spinner">
      <item name="android:textAppearance">@style/TextoDropDownEstilo</item>
      <item name="android:background">#FF9500</item>
    </style>
    <style name="ActionBarEstiloDropDown"
parent="@android:style/Widget.Holo.ListView.DropDown">
      <item name="android:textAppearance">@style/TextoDropDownEstilo</item>
      <item name="android:background">#FF9500</item>
    </style>
    <style name="TextoDropDownEstilo" parent="@android:style/Widget">
      <item name="android:textColor">#FFFFFF</item>
      <item name="android:layout_marginRight">9sp</item>
      <item name="android:layout_marginLeft">9sp</item>
    </style>
    <style name="xActionBar" parent="@android:style/Widget.Holo.ActionBar">
        <item name="android:background">#FF9500</item>
       <item
name="android:titleTextStyle">@style/Theme.AdnTemaApp.ActionBar.TitleTextStyle</item>
    </style>
    <style name="Theme.AdnTemaApp.ActionBar.TitleTextStyle"
parent="android:style/TextAppearance.Holo.Widget.ActionBar.Title">
       <item name="android:textColor">#FFFFFF</item>
    </style>

    <style name="cssTextViewTituloAyuda">
      <item name="android:textColor">#000000</item>
      <item name="android:layout_width">wrap_content</item>
      <item name="android:layout_height">wrap_content</item>
      <item name="android:textStyle">bold</item>
      <item name="android:padding">6sp</item>
      <item name="android:textSize">17sp</item>
    </style>
    <style name="cssTextViewTituloTienda">
      <item name="android:textColor">#000000</item>
      <item name="android:layout_width">90sp</item>
      <item name="android:layout_height">40sp</item>
```

```
  <item name="android:padding">6sp</item>
  <item name="android:textSize">9sp</item>
  <item name="android:background">@drawable/contornotitulolocal</item>
</style>
 <style name="cssTextViewTituloDistancia">
  <item name="android:textColor">#100904</item>
  <item name="android:layout_width">match_parent</item>
  <item name="android:layout_height">wrap_content</item>
  <item name="android:padding">6sp</item>
  <item name="android:textSize">10sp</item>
  <item name="android:textStyle">normal</item>
  <item name="android:background">@drawable/contornotextoinfo</item>
</style>

<style name="cssContainerDialogo">
  <item name="android:layout_width">300sp</item>
  <item name="android:layout_height">180sp</item>
  <item name="android:orientation">vertical</item>
  <item name="android:gravity">center</item>
  <item name="android:background">@drawable/contornodialogo</item>
</style>

<style name="cssListContainerItemEventos">
  <item name="android:layout_width">wrap_content</item>
  <item name="android:layout_height">wrap_content</item>
  <item name="android:orientation">vertical</item>
  <item name="android:gravity">center</item>
  <item name="android:paddingTop">4sp</item>
  <item name="android:paddingBottom">14sp</item>
  <item name="android:paddingLeft">10sp</item>
</style>
 <style name="cssListContainerItemCartelera">
  <item name="android:layout_width">fill_parent</item>
  <item name="android:layout_height">fill_parent</item>
  <item name="android:orientation">vertical</item>
  <item name="android:gravity">center</item>
  <item name="android:background">@drawable/contornocelda</item>
</style>

<style name="cssListTextView">
  <item name="android:textColor">#FFFFFF</item>
  <item name="android:layout_width">wrap_content</item>
  <item name="android:layout_height">wrap_content</item>
  <item name="android:textAppearance">?android:attr/textAppearanceSmall</item>
</style>

<style name="cssListTextViewItemTituloGrid" parent="cssListTextView">
    <item name="android:textColor">#636363</item>
    <item name="android:textStyle">bold</item>
    <item name="android:textSize">12sp</item>
</style>
<style name="cssListTextViewItemTituloForm" parent="cssListTextView">
    <item name="android:textColor">#636363</item>
```

```
        <item name="android:paddingLeft">4sp</item>
        <item name="android:paddingRight">4sp</item>
        <item name="android:paddingTop">8sp</item>
        <item name="android:paddingBottom">8sp</item>
        <item name="android:textStyle">bold</item>
        <item name="android:background">@drawable/contornotexto</item>
    </style>
    <style name="cssListTextViewItemTituloList" parent="cssListTextView">
        <item name="android:paddingLeft">20sp</item>
        <item name="android:paddingTop">20sp</item>
        <item name="android:paddingBottom">20sp</item>
        <item name="android:textColor">@drawable/selectortextcolorlistitem</item>
    </style>
    <style name="cssListTextViewItemTituloInteriorList" parent="cssListTextView">
        <item name="android:paddingLeft">10sp</item>
        <item name="android:paddingTop">10sp</item>
        <item name="android:paddingBottom">5sp</item>
        <item
name="android:textColor">@drawable/selectortextcolorlisttituloitem</item>
    </style>
    <style name="cssListTextViewItemDetalleInteriorList" parent="cssListTextView">
        <item name="android:paddingLeft">10sp</item>
        <item name="android:paddingTop">5sp</item>
        <item name="android:paddingBottom">5sp</item>
        <item
name="android:textColor">@drawable/selectortextcolorlistdetalleitem</item>
    </style>
     <style name="cssListTextViewItemTextImagenList" parent="cssListTextView">
        <item name="android:paddingLeft">10sp</item>
        <item name="android:paddingTop">4sp</item>
        <item name="android:paddingBottom">4sp</item>
        <item name="android:textColor">@drawable/selectortextcolorlistitem</item>
    </style>
    <style name="cssListTextViewItemForm" parent="cssListTextView">
        <item name="android:textColor">#636363</item>
        <item name="android:paddingLeft">4sp</item>
        <item name="android:paddingRight">4sp</item>
        <item name="android:paddingTop">2sp</item>
        <item name="android:paddingBottom">2sp</item>
        <item name="android:background">@drawable/contornotexto</item>
    </style>
</resources>
```