UNIVERSIDAD DE SALAMANCA

ESCUELA POLITÉCNICA SUPERIOR DE ÁVILA

MASTER EN GEOTECNOLOGÍAS CARTOGRÁFICAS EN INGENIERÍA Y ARQUITECTURA

PROYECTO FIN DE MASTER

APLICACIONES DE LAS NUEVAS TECNOLOGÍAS GEOMÁTICAS EN LA AGRICULTURA MODERNA: PROGRAMA AGRISOFT

TUTORES

D. JESÚS FERNÁNDEZ HERNÁNDEZ
D. JOSÉ ANTONIO MARTÍN JIMÉNEZ

ALUMNO

JOSÉ MANUEL CASLA FRANCISCO

Septiembre 2018

INDICE DE CONTENIDOS

INDICE DE IMÁGENES

INDICE DE TABLAS

INDICE DE CROQUIS

INDICE DE ESQUEMAS

Capítulo 1. Introducción

Como estación final del viaje que ha supuesto para mí el estudio del Master "Geotecnologías Cartográficas en Arquitectura e Ingeniería", estaba la realización de un Trabajo Fin de Master, en adelante PFM. Mi idea siempre ha sido, en las diferentes titulaciones que he cursado (Ingeniero Técnico en Topografía y Grado en Ingeniería Geomática y Topografía) y que han necesitado de la realización de un trabajo final, el elegir un trabajo que me aportase algo nuevo, huir de lo más accesible que significase únicamente realizar un proyecto para cumplir el expediente. En la titulación de Ingeniero Técnico en Topografía, obtenida en la Universidad Politécnica de Madrid, realice un Trabajo Final de Carrera consistente en un levantamiento gravimétrico de la zona norte de la Comunidad de Madrid y de la zona sur de la provincia de Segovia, alejada su temática de los horizontes profesionales que se veían para mi vida laboral, enmarcada aún hoy en el sector de la construcción. En la titulación de Grado en Ingeniería Geomática y Cartografía cursada también en la Universidad de Salamanca, en el campus de Ávila, elegí como Trabajo Final de Grado la realización de un estudio sobre la evolución de catastro rústico de la localidad de Condado de Castilnovo (provincia de Segovia) empleando para ello el programa de Sistema de Información Geográfica (SIG) gvSIG, por lo tanto también ajeno a mi vida profesional.

Continuando con esa actitud de explorar caminos del conocimiento ajenos a mi vida laboral y que me permitan ampliar mi campo de actividad profesional, llegado a esta última estación del Master, me propuse realizar un PFM que me permitiera ampliar mis conocimientos en el campo de la programación informática, sector hoy en día presente en todos los campos profesionales, económicos y sociales debido a la gran extensión de la informática. Mis conocimientos en programación se inician en el lenguaje C bajo MS-DOS, a mediados de la década de los 90 del siglo pasado cursando la asignatura de Informática en la titulación de Ingeniero Técnico en Topografía, fundamentalmente para elaborar programas para el cálculo de operaciones matemáticas con datos topográficos. A continuación me adentré en el lenguaje Basic, de manera autodidacta, también para realizar operaciones matemáticas con datos topográficos. Por último, y ya cursando el presente Master, la asignatura de Herramientas Informáticas para el Geoprocesado, tuve acceso al lenguaje Visual Basic.Net.

La programación siempre ha sido una materia que me ha atraído. Construir herramientas que me permitieran facilitar mi vida laboral, que completasen las herramientas informáticas contenidas en los equipos topográficos y que en mis inicios profesionales podían carecer de ellas o simplemente eran mejorables.

Como el sector agrícola forma parte de mi entorno, pensé en unir ambos sectores para hacer este PFM. Las aplicaciones informáticas están hoy a la orden del día en el sector agrario para diversas actividades: mapas de rendimiento; herramientas para elaborar la declaración de la PAC (Política Agrícola Común); análisis de costes; seguimiento de enfermedades; guiado de maquinaria agrícola.... Es esta última aplicación en la que me fijé para realizar este PFM.

Una aplicación informática para el guiado de maquinaria agrícola consiste en una herramienta que partiendo de datos de posicionamiento geográfico procedentes de un GPS (Sistema de Posicionamiento Global) ayuda al operario a realizar las operaciones agrícolas con mayor precisión, mayor descanso, menor consumo de inputs y más superficie trabajada por unidad de tiempo. Esta aplicación ha de aportar al operario, en todo momento, información sobre su posición geográfica georreferenciada y su posición en valores relativos respecto de otros puntos que marcan una línea de referencia que el habrá seleccionado previamente y a partir de la cual la aplicación informática traza paralelas por las cuales la máquina agrícola ha de transitar. La aplicación indica al operario los desplazamientos que ha de realizar sobre la máquina (a izquierda o derecha) para posicionarse paralelamente a la traza anterior y a la distancia determinada por la anchura del implemento agrícola.

Los implementos agrícolas (pulverizadores para los tratamientos herbicidas y fungicidas, abonadoras para el abonado del suelo, máquinas para el trabajo del suelo,..) con los que trabajan los más potentes tractores en la actualidad pueden abarcar anchos de trabajo que oscilan desde anchuras de 4-9 metros con máquinas para el laboreo del suelo hasta los 24-32 metros para aquellas máquinas que requieren menos potencia, como pulverizadores y abonadoras. Con estas anchuras de trabajo, la referencia visual de toda la vida o los marcadores de espuma aparecidos en la década de los 90 del siglo pasado se muestran insuficientes. Hace falta una herramienta más precisa que ayude al operario en su trabajo, le aporte mayor descanso en las largas jornadas laborales en las épocas de mayor actividad y evite solapes o zonas sin trabajar con la consiguiente pérdida de rentabilidad económica de la actividad (sobre todo en circunstancias de baja o nula visibilidad como son los trabajos nocturnos, con niebla o con deslumbramiento por sol).

Las circunstancias anteriores me hicieron pensar en hacer una aplicación de guiado agrícola, a la cual he denominado **Agrisoft,** como PFM en Android y ejecutable en dispositivos que tengan este sistema operativo, principalmente tabletas por su pantalla de mayor tamaño.

En el mercado existen aplicaciones similares:

1.- De empresas de equipamiento topográfico (Leica, Topcon, Trimble….) que se caracterizan por su elevado coste y que el producto integra el software y el hardware necesario (receptor GPS y antena de recepción de señal),

2.- Aplicaciones de coste más moderado de empresas de servicios informáticos que venden el producto por módulos: aplicación con o sin receptor GPS; aplicación con o sin soporte visual (tipo Tablet o notebook);…

3.- Aplicaciones gratuitas descargables de internet sin receptor GPS y sin soporte visual.

Las versiones más modernas de estas aplicaciones se denominan autoguiado: una vez iniciado el trabajo, seleccionada la traza de referencia, durante el tránsito de trabajo por las trazas paralelas a la de referencia el operario deja el control de la máquina a un dispositivo que mediante un conjunto de servomotores y sensores mueven el volante de la máquina adecuadamente para mantener el paralelismo de la distintas trazas de trabajo. El operario solo recupera el control de la máquina en los inicios y finales de la traza, coincidentes en la mayoría de las veces con los linderos de la finca, para realizar maniobras que permiten el inicio de una

nueva traza de trabajo. Iniciada una nueva traza el operario al pulsar un botón deja el control del volante a la aplicación y así solo ha de preocuparse de controlar el correcto funcionamiento de la máquina.

Las aplicaciones existentes en el mercado pueden tener como sistema operativo Windows o Android. He elegido el sistema Android al estar más extendido en el mercado de las tablets y ser abundante la información en internet sobre tutoriales, blog y foros de consulta.

Capítulo 2. Las nuevas tecnologías cartográficas aplicadas a la agricultura

La población mundial ha incrementado notablemente durante las últimas décadas y por tanto también ha aumentado la demanda de alimentos, no solo motivada por este aumento de la población sino también por el aumento en el nivel de vida de la sociedad que en su nuevo status reclama mayor cantidad y calidad de alimentos y que además se produzcan con menor consumo de recursos. La tecnología avanza a pasos agigantados en todos los sectores, y el sector agrícola no es una excepción.

Desde los inicios del presente siglo la agricultura ha experimentado una creciente modernización apoyada en la incorporación de la informática a este sector, sufriendo una importante evolución entre la actividad tradicional y la moderna.

Ya existió un salto importante a partir de mediados del siglo pasado con la generalización de los tractores (aunque ya fabricados desde inicios del siglo XX, su alto coste los convertía en prohibitivos para la gran mayoría de los agricultores) que supuso verdaderamente la llegada de la revolución industrial a la agricultura con siglo y medio de retraso. Hasta esa época, como norma general, el trabajo agrícola era fundamentalmente humano y ayudado por animales de tiro (caballos, vacas y bueyes). Hasta esta la llegada de los tractores, como avance únicamente existían herramientas de hierro para trabajar el suelo que eran movidas por tracción animal.

Imagen 1.- Agricultura tradicional

La finalidad de la agricultura tradicional era la de subsistir. Dicha técnica se caracterizaba por la carencia de tecnificación y tecnología. De manera que su producción era escasa y se limitaba para el propio consumo del agricultor y su familia.

Las herramientas básicas de esta labor eran la hoz, la azada o la pala y el rendimiento de la actividad no permitía obtener grandes réditos lo cual impedía también poder invertir en maquinaria.

En la actualidad, las personas que mantienen una agricultura similar (en países subdesarrollados) siguen consiguiendo resultados parecidos, debido a la exclusiva dependencia de las capacidades físicas de los trabajadores.

Esta agricultura se caracteriza por pequeñas propiedades utilizando técnicas rudimentarias, artesanales y antiguas cuyo objetivo principal es el autoconsumo y requiere de mucha mano de obra.

La aparición de las primeras máquinas supuso el aumento de los rendimientos por unidad de superficie, el aumento de los beneficios del agricultor, el aumento de las explotaciones, la disminución de la demanda de mano de obra (con el consiguiente éxodo a zonas urbanas de la mano de obra sobrante también atraída por salarios superiores y mejor calidad de vida).

La agricultura actual nace con la finalidad de responder a las necesidades de los mercados, demandando mayores cantidades de productos y también a la demanda por parte de los agricultores de mejorar su calidad de vida. El empleo de maquinaria supuso crear una actividad mucho más eficaz.

El segundo gran salto en el avance de la agricultura se corresponde con la incorporación de la informática y la ciencia a esta actividad desde los inicios del presente siglo. Ello supone:

1.- Un ahorro de recursos, tiempo y dinero logrando así una mayor producción en cantidad, calidad y beneficios en general.

2.- La automatización que implica reducir el riesgo de la dependencia de la variabilidad del clima.

3.- Ahorro de tiempo en acciones como la cosecha, incorporando máquinas de tipo cosechadoras que trabajan automáticamente y con una alta eficiencia.

4.- Ahorro en el consumo de inputs, como fertilizantes y tratamientos químicos.

5.- Aumento de la producción agrícola lo que repercute en una mayor rentabilidad económica de la actividad y por tanto del nivel de vida del agricultor. (*ver imagen 2*)

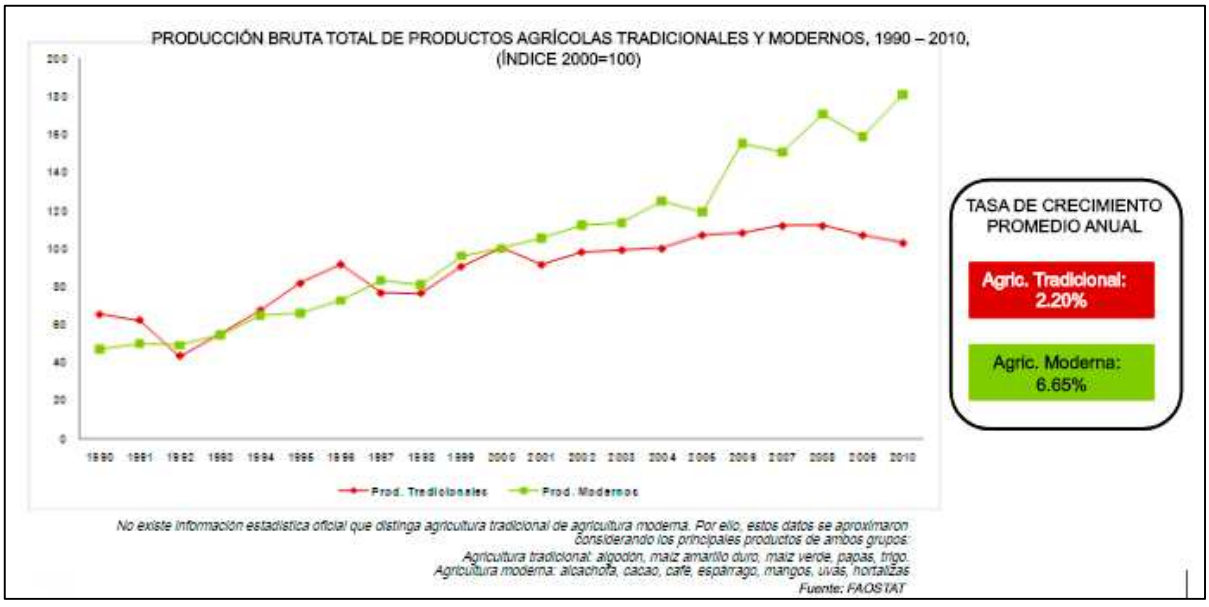


Imagen 2.- Evolución dispar de la producción en función del tipo de agricultura

Las máquinas agrícolas hoy en día disponen, en el peor de los casos, de la misma tecnología que cualquier otra máquina propulsada por los derivados del petróleo. Tractores con transmisión continua, gestión electrónica de la potencia, computadoras a bordo para la configuración de todas las operaciones de trabajo, comunicación ISOBUS con los implementos con los que trabajan para configurar los trabajos, motores de última tecnología sometidos a estrictas normas anticontaminantes….

La gestión de una explotación agrícola se ve apoyada por herramientas informáticas que permiten conocer el estado de los cultivos a distancia (por imágenes de satélites o vuelos de drones); controlar la evolución de los gastos e ingresos; tener una predicción del tiempo que ayuda en la toma de decisiones; conocer cuál ha sido la producción del cultivo mediante un mapa de rendimiento; hacer previsiones de cosecha; controlar la aparición y evolución de plagas y enfermedades; empleo de los Sistema de Posicionamiento Global (en adelante GPS) para operaciones de medición y apoyo a la conducción de las máquinas agrícolas;…..

La incorporación de las nuevas tecnologías geomáticas al sector agrario, no son pocas como puede parecer a simple vista:

1.- La tecnología GPS incorporada a los vehículos agrícolas facilitan la realización de una mayor cantidad de trabajo en menos tiempo y además de una forma más segura y eficiente. Por otro lado, esta eficiencia trae consigo un mayor ahorro de combustible.

Los tractores y cualquier máquina de recolección que se usan en la actualidad disponen de tecnología GPS, en la mayoría de los casos de manera opcional en el momento de la adquisición o también es posible su incorporación a máquinas ya adquiridas. Gracias a ella, se pueden mover con total precisión y se ahorran insumos. Gracias al GPS, se llega a reducir el número de zonas solapadas en torno a un 8 y 12%. Además, pueden trabajar en cualquier hora y con cualquier tipo de condición climatológica. Sin olvidar, por supuesto, la mayor velocidad, seguridad y precisión que supone usar este tipo de cosechadoras o tractores.

Imagen 3.- Tecnología GPS en tareas de recolección de forraje con cosechadora y tractor

El terreno donde actualmente trabaja esta tecnología es en los tractores autónomos. Tractores sin conductor, que gracias a la tecnología GPS podrían realizar las tareas agrícolas convenientemente programadas de manera autónoma.

2.- Lenguajes de programación (Java, Visual Basic, Matlab...) que se emplean en el diseño de software. Estos softwares nacen para el análisis del terreno, mapas de rendimiento, análisis de imágenes aéreas o de satélites, control telemático de las operaciones riego, control de la temperatura, humedad y luz óptimas en los invernaderos, analizar que partes del terreno son más productivas y son empleados por los agricultores para ser mucho más eficientes en su actividad económica.

3.- El uso de drones está cada vez más extendido en muchos sectores económicos y también en el mundo agrícola. Dentro de este campo su empleo se extiende a:

a) Realización de estudios topográficos con precisión centimétrica**,** para planificar, por ejemplo, de forma óptima nuevos sectores de riego o detectar problemas de drenaje en los existentes.

b) Manejo eficiente del agua de riego: el estrés hídrico en los cultivos provoca el cierre de estomas, reduciendo la transpiración y aumentando la temperatura de las hojas. Este aumento de temperatura puede monitorizarse con la cámara térmica del dron, ya que la combinación de diferentes rangos del espectro refleja variaciones espectrales que se pueden relacionar con su nivel de estrés hídrico. De este modo puede estimarse de forma individual los requerimientos hídricos para cada planta y ajustar los riegos a dichas necesidades, con el consiguiente ahorro de agua y energía.

c) Monitorización del estado de los cultivos durante su ciclo fenológico y fisiológico. Las imágenes multiespectrales obtenidas con vuelos dron, en combinación con parámetros medidos en campo, permiten determinar los principales índices agronómicos del cultivo, pudiendo ajustar las dosis óptimas de fertilizantes y fitosanitarios al poder detectar los cambios fisiológicos que las enfermedades causan en los cultivos en estados tempranos.

   Al ir equipados con cámaras, sensores y poder ser controlados, permiten a los propietarios de amplios cultivos realizar un seguimiento sin tener que acceder a ellos, lo cual evita daños innecesarios. Los productos generados por los drones, imágenes aéreas principalmente, entran en competencia con los productos generados por los satélites. Hoy se puede hablar de la teledetección generada con drones.

d) Desde hace un tiempo se vienen aplicando productos fitosanitarios mediante drones, una forma de hacerlo mucho más rápida, eficaz y controlada.

Imagen 4.- Dron en tratamiento de cultivo

4.- Empleo en el sector agrícola de imágenes recogidas por satélites. Europa, aunque un poco más tarde que los EEUU, está desarrollando un nuevo conjunto de satélites que se empezaron a lanzar al espacio en 2014. La ESA (Agencia Espacial Europea) es la encargada del lanzamiento de estos satélites llamados Sentinel, desarrollados dentro del programa de Monitorización Global para la Seguridad y el Medio Ambiente.

El objetivo de estos satélites es tener un control general del mundo desde el espacio. Así se pueden usar para meteorología, geografía, agricultura y muchos otros ámbitos. Para conseguirlo, el conjunto está formado por satélites de dos tipos: satélites radar y satélites de imagen super-espectrales.

El primer satélite Sentinel lanzado fue el Sentinel – 1A que se lanzó en 2014. Este y su hermano, el Sentinel – 1B, proporcionan imágenes de radar terrestres y oceánicas. Estos satélites tienen aplicaciones en vigilancia, pero no en agricultura. En agricultura son necesarias las imágenes multiespectrales.

En 2015 se lanzó el primer satélite Sentinel 2. En marzo de este 2017 se ha lanzado el satélite Sentinel 2B. Estos satélites sí que tienen aplicaciones en el campo de la agricultura ya que proporcionan imágenes de alta resolución como si de una cámara de fotos se tratara. Esto permite captar imágenes de la vegetación y el suelo. De esta forma se pueden obtener datos para controlar la navegación de las vías navegables interiores, tener información sobre el uso del suelo, etc.

En resumen, dentro del lanzamiento de los satélites Sentinel son los satélites Sentinel 2 los que pueden tener una aplicación más directa en agricultura de precisión ya que obtendrán imágenes de la vegetación, casi en tiempo real. Estas imágenes, con programas de procesamiento, se pueden convertir en mapas útiles en agricultura.

Algunas de las ventajas que aporta el SENTINEL 2, respeto LANDSAT (satélites puestos en órbita por los EEUU), es que permite trabajar con mucha más resolución. Landsat tiene una resolución en la que cada píxel son 30 x 30 m, mientras que en Sentinel se llega a 10 x 10 m. Esto abre la puerta a que en zonas donde hay una agricultura de menor escala también se puedan utilizar imágenes de satélite para poder trabajar con ellas.

Otra mejora que aportarán los satélites Sentinel 2 en comparación a Landsat es una mayor frecuencia de imágenes. Esto quiere decir que se dispondrá de imágenes de la misma zona en menos diferencia de tiempo.

Por ejemplo, si ahora Landsat tiene una frecuencia de visitas de 10 días, está previsto que Sentinel la tenga de unos 5 días. Así, por ejemplo, para planificar el abonado de cobertera de un maíz, tener información cada 5 días es fundamental ya que el cultivo puede evolucionar muy rápidamente.

Para poder usar las imágenes obtenidas por el satélite es necesario procesarlas. Con el procesado es posible calcular índices de vigor de la vegetación y representarlos sobre mapas, que permiten detectar la variación entre parcelas. Por ejemplo, el cereal de invierno se suele abonar (abonado basado principalmente en nitrógeno) en enero, febrero o marzo, al final de la parada invernal, para que tenga el abono suficiente para crecer en primavera. Tener imágenes de satélite tomadas solo unos días antes del abonado, permite ajustar mucho más la fertilización, con lo que se consigue un ahorro económico y medioambiental. Con las imágenes de satélite y

mediante programas de procesamiento podemos calcular índices que miden la intensidad de verde de la planta. En concreto se suele calcular el NDVI, un índice que se calcula pixel por pixel de la imagen a partir de la reflexión de la longitud de onda del infrarrojo cercano y del rojo. Calculando este índice pixel por pixel en una imagen, se pueden realizar mapas de intensidad de verde. A partir de ellos se obtienen los mapas de fertilización. Así, se puede decidir aplicar más fertilizante en las zonas con menos intensidad de verde. La intensidad de verde está íntimamente relacionada con la cantidad de nitrógeno en hoja del cereal ya que cuanto más nitrógeno más intenso y oscuro es el verde de la hoja.

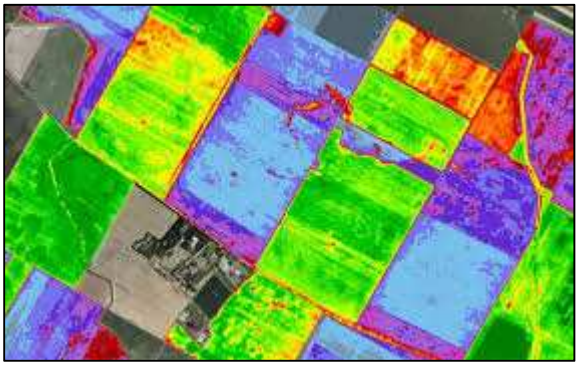


Imagen 5.- Satélite para uso agrícola

Focalizar la agricultura en satélites tiene sus desventajas, por ejemplo las imágenes de satélites pueden resultar caras, o se pueden haber tomado un día de niebla, con lo que no se tendrían datos en ese instante. Las imágenes procedentes de drones se pueden complementar bien con los satélites.

Gracias a estas nuevas tecnologías, se puede sacar un máximo rendimiento al uso de recursos naturales, tales como el agua, la tierra, los fertilizantes o la energía. Además, con este tipo de herramientas se produce un ahorro en los costes. Por lo tanto, se hablaría de aumentar el rendimiento sin tener que aumentar los gastos, lograr un óptimo cultivo para cada tipo de terreno.

Con estas tecnologías nace el concepto de Agricultura de Precisión: aquella que se encarga de que haya una mayor eficiencia a la hora de cuidar los cultivos gracias a la gestión agronómica. La Agricultura de Precisión permite manejar cada insumo (semilla, fertilizante, riego, herbicida, fungicida, etc…) de una manera concreta para cada unidad de superficie, es decir de acuerdo a las necesidades de cada parcela, permitiendo disminuir los costos y reducir el impacto ambiental.

La agricultura de precisión se basa en tres etapas fundamentales: la recolección de información, el análisis de dicha información y el manejo de la unidad de producción y en todos los pasos, como hemos visto anteriormente las nuevas tecnologías geomáticas tienen su intervención.

Gracias a este tipo de agricultura se combina una administración totalmente eficiente con la rentabilidad a la hora de realizar las explotaciones del cultivo.

Este tipo de agricultura abarca todavía muchas más técnicas aún en desarrollo o que, de momento, son algo inasequibles para los particulares.

En definitiva, gracias a la agrotecnología se está consiguiendo que las plantas crezcan más sanas y fuertes, ahorrando en costes y en tiempo

El informe “Cambio climático, impactos y vulnerabilidad en Europa 2016”, publicado por la Agencia Europea del Medio Ambiente (AEMA), señala a la Península Ibérica como uno de los puntos críticos de los efectos del cambio climático. El aumento de temperaturas máximas y la disminución generalizada de precipitaciones ocasionarán un incremento del riesgo de fenómenos extremos, como pueden ser periodos de sequías prolongados. En la práctica esto supondrá que el agua se convierta en un recurso cada vez más escaso y limitante, si cabe, de manera que cada vez sea más necesario el uso eficiente del mismo.

Las nuevas tecnologías geomáticas aplicadas a la agricultura se presentan como una ayuda para mitigar estos problemas. Además proporcionan gran cantidad de información de precisión que puede ser consultada en tiempo real en el ordenador o el móvil, facilitando de este modo la toma de decisiones.

Por tanto, el empleo de estas nuevas tecnologías aplicadas a la agricultura permitirá una utilización más eficiente del agua de riego, fertilizantes y fitosanitarios, incrementando la rentabilidad de las explotaciones mediante una reducción de costes y minimizando el impacto ambiental como consecuencia del ajuste a los requerimientos reales de los cultivos.

Capítulo 3. Sistema de Posicionamiento Global GPS

3.1. Sistema de Posicionamiento Global (GPS)

GPS es la abreviatura del Global Positioning System (en adelante GPS) y utiliza la constelación de satélites NAVSTAR, siendo el acrónimo en inglés de Navigation System for Time And Ranging.

La metodología del sistema GPS se basa en la determinación de la posición de puntos sobre la superficie terrestre, apoyándose en la información radioeléctrica enviada por los satélites.

El GPS es un sistema basado en satélites artificiales activos, formando una constelación con un mínimo de 24 satélites activos. Permite diferentes rangos de precisión según el tipo de receptor utilizado y la técnica aplicada.

El sistema GPS fue desarrollado por el Departamento de Defensa de los Estados Unidos de América. La metodología nació con el objetivo de mejorar el sistema de satélites de navegación militar TRANSIT (efecto Doppler), muy usado en geodesia desde 1967 en todo el mundo. El primer satélite GPS data de 1.978 y la fecha desde la que se considera en funcionamiento el sistema es enero de 1.994.

El sistema de referencia geodésico asociado al sistema GPS se conoce como World Geodetic System (WGS). El primer elipsoide global de referencia se estableció 1.960 y ha sido mejorado hasta su versión actual denominada WGS84 (definido en 1984). Sobre este sistema de referencia se obtienen las coordenadas cartesianas o polares del punto en el que se ha realizado la observación.

Al calcular la posición de un punto por métodos de posicionamiento GPS deberemos tener en cuenta que estás lo son respecto al Sistema Geodésico de Referencia WGS84 y que han de hacerse las oportunas observaciones y transformaciones, que nos permitan obtener los resultados en el sistema de coordenadas deseado, el sistema de referencia local.

A principios de los años 70 se propuso el proyecto GPS, para satisfacer los requerimientos militares del gobierno de los Estados Unidos en la determinación de posiciones terrestres precisas sin importar las condiciones meteorológicas por las que estuviera afectado y bajo un sistema unificado de cobertura. Una vez consolidado militarmente dicho sistema, sus aplicaciones se extendieron para usos comerciales divulgándose entre la comunidad científica. Las aplicaciones del sistema incluyen aplicaciones de navegación, topografía y geodesia abarcando desde la administración de una flota de vehículos hasta la automatización de maquinaria de construcción.

Frente al control del sistema GPS por parte del gobierno de los EEUU, la Unión Europea está desarrollando su propia constelación de satélites para disponer de un sistema de navegación propio. Este sistema de denomina GALILEO y el número de satélites será de 24 a 35. Existen otros sistemas semejantes, como el GLONASS de patente rusa y el BEIDOU de patente china.

El sistema GPS consta de tres sectores: el sector espacial (los satélites), el sector de control terrestre de los mismos y el sector usuario (los receptores de los usuario) que recogen señales enviadas por los satélites y determinan las coordenadas del punto sobre el que se encuentran.

3.1.1. Sector espacial

Está compuesto por la constelación de satélites NAVSTAR (Sistema de Navegación para tiempo y distancia) los cuales transmiten señales (decodificadas por los receptores GPS) de radio con información referida a tiempos sincronizados, parámetros de posición de los satélites, información del estado de salud de los satélites sobre dos portadoras y otros datos adicionales.

La constelación actual consta de entre 27 y 31 satélites distribuidos en seis órbitas con 4 o más satélites en cada una. Los planos orbitales tienen una inclinación de 55 grados y están distribuidas uniformemente en el plano del ecuador separados entre sí 60 grados, por lo tanto hay seis planos orbitales identificados con las letras A, B, C, D, E y F. Dentro de cada órbita los satélites se identifican con los números 1, 2, 3, 4, 5. Las órbitas no son geoestacionarias y son aproximadamente circulares con radio 26.560 kilómetros. El periodo de la órbita, tiempo que emplea un satélite en recorrer una órbita completa es de medio día sidéreo, es decir aproximadamente 11,967 horas. Con este periodo orbital, un satélite estará sobre el horizonte unas cinco horas. El objetivo es que al menos cuatro sean visibles al mismo tiempo, a cualquier hora del día y desde cualquier punto de la superficie terrestre.

La constelación de satélites NAVSTAR es una mezcla de satélites antiguos y modernos:

1.- Bloque IIA: sustituyeron a los satélites del bloque II y comprende un total de 19 satélites desde el SVN22 al SVN40. Se lanzaron desde el año 1.990 al 1.997. En el año 2.016 se retiró del servicio el último satélite activo de este bloque.

2.- Bloque HR: lanzados para sustituir a los satélites del bloque II y bloque IIA. Se lanzaron un total de 13 satélites entre los años 1.997 y 2.004.

3.- Bloque IIR(M): es una versión actualizada del bloque IIR y junto a estos constituyen la base de la constelación actual. Se lanzaron 8 satélites entre los años 2.005 y 2.009.

4.- Bloque HF: amplía las capacidades del bloque HR y suma una tercera señal civil en una frecuencia protegida para aquellos tipos de transporte, como aerolíneas, que han de asegurar de modo especial la seguridad de los pasajeros. Los satélites de este bloque tienen una vida útil más larga y disponen de relojes atómicos de rubidio y cesio, lo que les proporciona mayor exactitud. Incluye 12 satélites desde el SVN62 al SVN73 y se lanzaron entre los años 2.010 y 2.016.

5.- Bloque III: esta generación de satélites se encuentra en desarrollo en la actualidad. Se empezaron a lanzar en el año 2.016 y se prevé completar el bloque con 12 satélites y se denominaran desde el SVN74 en adelante.

Para la identificación de cada satélite se pueden emplear diversas signaturas:

1.- Número NAVSTAR (SVN) o número de lanzamiento de satélite.

2.- Letra de la órbita a la que pertenece y número de la posición dentro de ella.

3.- Número de catálogo NASA.

4.- Identificación internacional: año de lanzamiento, día juliano y tipo.

5.- Número IRON. Número aleatorio asignado por NORAD.

6.- Código seudoaleatorio PRN, que es la forma habitual de identificación.

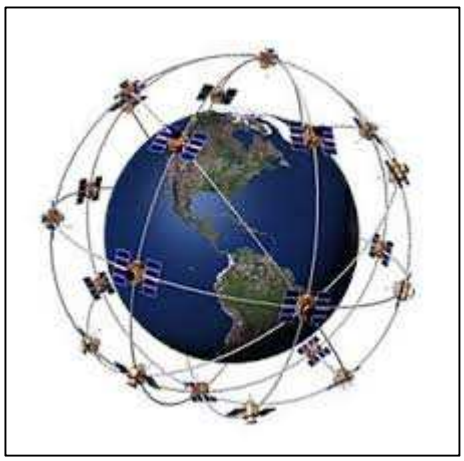


Imagen 6.- Órbitas de los satélites entorno a la tierra

Los satélites tienen una serie de antenas emisoras que funcionan en la banda L del espectro electromagnético, que son las que recibimos en nuestros receptores. Los satélites emiten información sobre dos movimientos ondulatorios que actúan como portadoras de códigos, la primera se denomina L1 (cuya frecuencia es 1.575,42 MHz) y la segunda L2 (con frecuencia 1.227,6 MHz). El poder utilizar las 2 frecuencias permite determinar, por comparación de la diferencia de retardos, el retardo ionosférico consecuencia del tránsito de la señal GPS por la capa de la ionosfera, de lo que se deduce que se consigue mayor precisión en la medición si el receptor de señal GPS mide esta dos frecuencias.

Sobre estas dos ondas portadoras se envía una información modulada compuesta por tres códigos y un mensaje de navegación:

1.- El primer código que envían es el llamado código C/A (coarse/adquisition, e identifica al satélite emisor y permite determinar el tiempo de emisión de la señal) y ofrece precisiones que en la actualidad oscilan entre 3 metros y 10 metros. Modula la fase de la portadora L1. Tiene una longitud de 1.023 chips, una frecuencia de 1.023 MHz y una duración de 1ms.

2.- El segundo código emitido es el código P (precise code) con precisiones métricas. Modula las fases L1 y L2. Tiene una duración de 7 días y una frecuencia de 10,23 MHz. En modo AS (anti-spoofing, anti trampa) este código se encripta para formar el código Y.

3.- El tercer código, el código L2C, es de precisión similar al código C/A

4.- Mensaje de navegación. Formado por bits de datos que el receptor GPS decodifica: órbita del satélite, corrección del reloj, etc.

Existen dos niveles de servicio:

1.- Nivel PPS (precise positioning service): disponible para usuarios autorizados por el gobierno de los EEUU. Proporciona la mayor precisión posible para un único receptor GPS.

2.- SPS (standard positioning service): permite sólo el acceso al código C/A y al mensaje de navegación de la señal L1.

El mensaje enviado consta de 1.500 bits y está dividido en 5 celdas. La información contenida en cada celda es la siguiente:

- Celda 1: parámetros de desfase del reloj y modelo del retardo ionosférico y troposférico.

- Celdas 2 y 3: efemérides de los satélites.

- Celda 4: celda reservada a aplicaciones militares.

- Celda 5: almanaque.

Sobre la portadora L1 se suelen enviar los códigos vistos C/A y P, además del mensaje correspondiente. En la portadora L2 se modula el mensaje de navegación y los códigos L2C y P.

3.1.2. Sector de control

La misión de este sector consiste en el seguimiento continuo de los satélites, calculando su posición, transmitiendo datos y controlando a diario los satélites de la constelación. En la actualidad está formada por: una estación de control principal; una estación de control principal alternativa; 16 estaciones de monitorización y 12 antenas de control.

Además de la información señalada anteriormente procedente de los satélites también se registra otro tipo de información:

1.- Influencia que sobre el satélite tiene el campo magnético terrestre.

2.- Parámetros sobre la presión de la radiación solar.

3.- Posibles fallos de los relojes atómicos.

4.- Operatividad de cada uno de los satélites.

5.- Posición estimada de cada uno de los satélites dentro de la constelación global.

Toda esta información se transmite a la estación principal de control situada en Colorado Sprint (Estados Unidos). En esta estación se procesa la información para obtener las posiciones de los satélites en sus órbitas y el estado de los relojes atómicos que llevan cada uno de los satélites transmitiendo esta información a los satélites. Posteriormente los satélites transmiten esta información a los receptores de los usuarios.

Imagen 7.- Estaciones del sector de control

3.1.3. Sector usuario

Este segmento está formado por todos los equipos utilizados para la recepción de las señales emitidas por los satélites, así como por el software necesario para la comunicación del receptor con el ordenador y el postprocesado de la información para la obtención de los resultados.

Las primeras aplicaciones civiles llegaron de la mano de la navegación, en lo que hoy conocemos como gestión y control de flotas.

3.2. Funcionamiento del sistema GPS

Para determinar la posición de un receptor GPS tan sólo es necesario conocer la distancia entre el receptor y los satélites que están siendo observados y la posición de dichos satélites, de tal manera que por métodos de triangulación se obtiene la posición del receptor. Como la posición de un punto-receptor sobre la superficie terrestre queda definida por la terna latitud, longitud y altura, su posición tridimensional, son tres por lo tanto las incógnitas a calcular y por lo tanto, con la observación a tres satélites se resolvería el sistema de incógnitas.

Las posiciones de los satélites se obtienen a partir de las efemérides radiadas por todos ellos y contenidas en el mensaje de navegación. La distancia se determina a partir de la medida del tiempo que necesita la señal para viajar desde el satélite emisor al receptor y multiplicando dicho tiempo por la velocidad de la luz (velocidad de transmisión de la señal GPS).

Por lo tanto el sistema GPS se basa fundamentalmente en la medida del tiempo de una forma muy precisa. Para ello los satélites contienen varios osciladores de alta precisión, con estabilizadores capaces de dar medidas del tiempo del orden de 10^{-12} y de 10^{-14} segundos en los satélites de última generación (los satélites del bloque III). El satélite junto con su posición transmite el tiempo de emisión de la señal. El receptor resta al tiempo de llegada del mensaje el tiempo de emisión. Es fundamental la sincronización de los relojes del satélite y del receptor pues una defectuosa medición de este afecta a la medida de la distancia satélite-receptor, la cual es función del tiempo y de la velocidad de la luz. Un error en la medida del tiempo del orden de un nanosegundo genera un error en la medida de la distancia de 30 centímetros.

La sincronización entre los relojes no es la mejor posible. Los satélites si usan relojes de alta precisión, relojes atómicos, pero los relojes de los receptores no son de este tipo, por simple cuestión económica, sino que emplean osciladores de cristal mucho más económicos con un cierto error o desfase respecto a los relojes de los satélites. Este desfase se denomina "bias" del reloj del receptor.

La solución a este problema del desfase temporal entre relojes consiste en considerar este "bias" como una cuarta incógnita en el proceso de medida y cálculo y por lo tanto son necesarios al menos la recepción de señal GPS de cuatro satélites. A la medida de la distancia satélite-receptor afectada por este error temporal se le conoce como pseudodistancia.

Cada satélite tiene un reloj-oscilador que provee una frecuencia sobre la que se estructura todo el conjunto de la señal emitida.

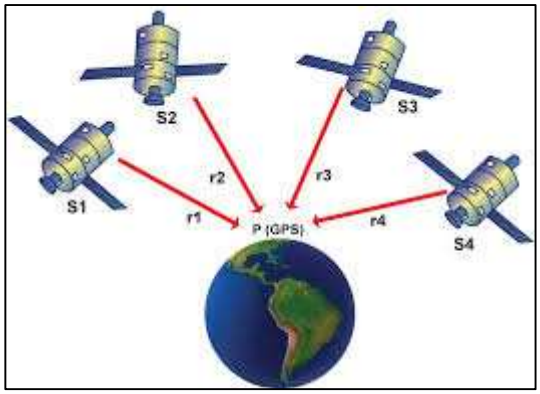


Imagen 8.- Pseudodistancias r entre los satélites y el receptor

3.3. Clasificación de los receptores GPS

Los receptores GPS del sector usuario se pueden clasificar en base a varios criterios:

1.- En función del observable que emplean para determinar su posición:

- Receptores de medida de pseudodistancia (código), que son los navegadores.
- Receptores de medida de pseudistancia y fase, receptores más precisos empleados en topografía y geodesia.

2.- Receptores en función de la frecuencia registrada:

- Receptores que registran la frecuencia L1, monofrecuencia que registran el código C/A.
- Receptores bifrecuencia, que registran la frecuencia L1 y L2.

3.- En función de la precisión que permiten alcanzar:

- Navegadores: únicamente reciben el código C/A por la portadora L1. Son receptores GPS sencillos y de bajo coste. Funcionan autónomamente y consiguen precisiones por debajo de los 10 metros.

- Receptores GPS submétricos: reciben los mismos observables que los navegadores, pero difieren de éstos en que trabajan diferencialmente, es decir, admiten correcciones. Es necesario un receptor GPS de referencia y un receptor GPS móvil, de tal manera que el equipo de referencia envía las correcciones necesarias al receptor móvil para conseguir precisiones por debajo del metro.

- Receptores GPS monofrecuencia de código y fase: son receptores que toman datos de la portadora L1 en sus dos modalidades código C/A y fase. Son equipos que también trabajan en modo diferencial alcanzando precisiones de 1 cm + 2 ppm., lo que permite emplearlos en aplicaciones topográficas.

- Receptores GPS doble frecuencia: son los equipos de mayor precisión y coste empleados en topografía y geodesia. Toman como observables las dos portadoras emitidas por los satélites, realizando medidas de código C/A y P en la portadora L1, de código P y L2C en portadora L2 y medición de fase en L1 y L2. Con ellos se pueden alcanzar precisiones de 5 mm + 1 ppm.

3.4. Descripción del receptor del usuario

Los receptores empleados en topografía y geodesia constan de los siguientes elementos:

1.- **Antena GPS:** recibe y amplifica la señal recibida de los satélites. La posición calculada se refiere a la posición de la antena y su misión es la de convertir la energía electromagnética que reciben en corriente eléctrica que a su vez pasa al receptor mediante cable o bluetooth.

2.- **Receptor GPS:** ordenador que decodifica la señal recibida por la antena y registra las observaciones enviadas por los satélites. Poseen diferentes canales para seguir a varios satélites y un procesador interno.

3.- **Terminal:** es una interface de usuario que permite conocer el estado de la recepción, proceso de cálculo, y llevar a cabo la edición de los datos del receptor. Poseen una unidad de almacenamiento de información, teclado de control, pantalla de comunicación con el usuario, diferentes conectores.

El desarrollo de la tecnología GPS se orienta hacia la mejora en el seguimiento de los satélites, la disminución de efector multipath y la creación de equipos más compactos, ligeros y pequeños.

3.5. Métodos de posicionamiento GPS

Existen distintos criterios a la hora de clasificar los métodos de observación o posicionamiento GPS:

1.- Según el Sistema de Referencia:

- Absoluto: se calcula la posición de un punto utilizando las medidas de pseudodistancias por código por un solo receptor y la precisión está en menos de 10 metros.

- Relativo o diferencial: es necesario observar con dos equipos simultáneamente y las mediciones se pueden hacer por código o por fase. Se determina la distancia o incremento de coordenadas entre las antenas de los receptores GPS. La ventaja de este método radica en que los errores de posicionamiento muy similares en los dos receptores son eliminados en su mayor parte.

2.- Según el movimiento del receptor:

- Estático: se determina la posición de la antena del receptor GPS en base a unas observaciones realizadas durante un periodo de tiempo en el cual el GPS no ha sufrido desplazamientos. Se calcula un único trio de coordenadas (x,y,z). Hay redundancia de observaciones.

- Cinemático: se calcula un conjunto de coordenadas (x,y,z) en función del tiempo y la situación de la antena del receptor GPS la cual estará en movimiento. No hay redundancia en las coordenadas del punto calculado.

3.- Según el observable utilizado:

- Medida de código: se determina a partir de las pseudodistancias entre el satélite y el receptor mediante la utilización del código de la portadora. Se puede medir el código C/A y L2C o el código P.

- Medida de fase de la portadora: se utiliza la fase de la portadora para realizar la medida de la pseudodistancia. Requiere trabajar en modo diferencial o relativo.

4.- Según el momento de la obtención de coordenadas:

- Tiempo real: las coordenadas del receptor se obtienen en el momento de la observación y la precisión depende del observable utilizado (código o fase) y del método utilizado (absoluto o relativo).

- Postproceso: las coordenadas del receptor son obtenidas en postproceso. Una vez finalizada la observación se calculan las posiciones en gabinete. Este método se suele utilizar para posicionamiento estático relativo que con medidas de fase se obtienen soluciones más precisas que en tiempo real.

A partir de la combinación de estos métodos surgen otros métodos de observación:

- Estático absoluto con medición de pseudodistancias.

- Cinemático absoluto con medición de pseudodistancias.

- Estático relativo con medición de pseudodistancias y fase. Este método de observación puede ser estándar o rápido.

- Cinemático relativo con medición de pseudodistancias y fase. Este método de observación puede ser cinemático en postproceso, RTK (Medición de fase en tiempo real, Real Time Kinematic), y DGPS (Medición de código, Real Time Diferencial GPS).

Para el caso de la topografía y geodesia todas las medidas GPS utilizarán el modo diferencial o relativo, es decir, se mide una línea base desde un punto fijo (receptor GPS situado en una estación de referencia de coordenadas conocidas con precisión) a un punto desconocido (receptor GPS móvil).

Los principales métodos de posicionamiento GPS aplicados en topografía y geodesia, que utilizan la medida de fase para la determinación de la línea base entre el receptor fijo y el receptor móvil son los siguientes:

- **Método estático relativo estándar:** se trata del posicionamiento para la medida de distancias con gran precisión (5 mm +1 ppm.) en el que dos o más receptores se estacionan durante un tiempo mínimo de media hora. Los resultados obtenidos alcanzan precisiones elevadas. Este método es empleado para medir distancias mayores de 20 km con total precisión. Sus aplicaciones son fundamentalmente en el campo geodésico y para el seguimiento de movimientos tectónicos.

- **Método estático relativo rápido:** es una variante del método anterior reduciendo los tiempos de observación hasta 5 o 10 minutos por estación manteniendo la misma precisión debido al algoritmo matemático para la resolución estadística de las ambigüedades que permite la disminución de los tiempos de observación. Por el contrario las distancias de observaciones deben ser menores de 20 km. Este método se emplea para el cálculo de redes topográficas locales.

- **Método RTK**: consiste en la obtención de coordenadas en tiempo real con precisión centimétrica (1 o 2 cm +1 ppm.). Es un método diferencial en el que el receptor fijo o referencia estará en modo estático en un punto de coordenadas conocidas, mientras que el receptor móvil del cual se calculan sus coordenadas está en movimiento. La comunicación entre el receptor fijo y el móvil se realiza mediante algún sistema de telecomunicaciones (radio-modem, GSM, GPRS, satélites, etc.). Esta comunicación entre receptores GPS es una restricción en la utilización del método pues es dependiente del alcance de la comunicación. Es común su empleo en el mundo de la topografía para realizar taquimétricos y replanteos en tiempo real.

- **Método DGPS**: consiste en la obtención de coordenadas en tiempo real con precisión métrica o submétrica. Al igual que el método anterior es un método diferencial con un receptor GPS fijo en un punto de coordenadas conocidas y un receptor móvil del cual se calcularán sus coordenadas en tiempo real. Se trabaja con el código, es decir, con la medida de pseudodistancias. En el receptor móvil se realizar una corrección a las pseudodistancias calculadas, mediante los parámetros de corrección que envía el receptor GPS de referencia. También precisa la comunicación por algún sistema de telecomunicaciones entre el receptor GPS fijo y el

receptor GPS móvil con la restricción que conlleva este tipo de comunicación. Este método se aplica fundamentalmente en navegación: actualizaciones cartográficas de pequeña escala, GIS, etc.

3.6. Red de correcciones GPS del ItaCyL

La Red GNSS de Castilla y León, gestionada por el Instituto Tecnológico Agrario de Castilla y León (en adelante ItaCyL) es un servicio libre y gratuito de posicionamiento de alta precisión con receptores GPS dentro del territorio de Castilla y León con el datum ETRS89. Dicha red proporciona correcciones de código y fase para los sistemas de navegación GPS y GLONASS, tanto en tiempo real RTK a través de un caster NTRIP, como en postproceso a través de ficheros RINEX.

Al mismo tiempo, el sistema constituye un marco de referencia geodésico activo en ETRS89, que sustituye y complementa con ventaja a las tradicionales redes geodésicas basadas en vértices fijos

Las correcciones facilitadas por la Red GNSS de CyL son de carácter libre y gratuito, y se ofrecen a través de Internet. Para poder recibir las correcciones es necesario inscribirse en la página web http://www.itacyl.es, asignándose uno mismo el usuario y contraseña que permite acceder a los datos en tiempo real y también recibir comunicaciones de cambios, cortes de red, o actualizaciones importantes.

Existen dos tipos diferentes de correcciones en función del momento de aplicación:

1.- Corrección en diferido o postproceso. Con esta metodología se consigue el posicionamiento preciso una vez que han sido procesados los datos en la oficina. Para ello es necesario descargar del receptor las mediciones realizadas, y desde internet los ficheros RINEX de corrección. Este sistema proporciona precisiones que pueden ser superiores al uso de correcciones en tiempo real. Para usar este servicio el usuario debe disponer de un receptor GPS que almacene los datos de observación en bruto para poder realizar una corrección diferencial y de un software de procesado de datos GPS. Para acceder a los ficheros del ItaCyL que nos permiten realizar un trabajo de precisión en postproceso hay que acceder al servidor ftp://ftp.itacyl.es/RINEX. Son ficheros en formato RINEX 2.11, comprimidos con el software Hatanaka y clasificados en función de la fecha, hora y estación GPS que los ha generado.

2.- Corrección en directo o tiempo real. Este sistema permite conseguir un posicionamiento preciso en el mismo instante en que se realiza la medición. Entre sus principales aplicaciones están los trabajos de replanteo y la navegación de alta precisión. Los datos suministrados por esta red permiten usar las técnicas denominadas RTK (Real Time Kinematic) y DGPS (GPS Diferencial de Código). Para usar este tipo de correcciones el usuario necesita disponer de un receptor GNSS que admita correcciones en el estándar RTCM (versiones 2 o 3) y de conexión a Internet en el instante en el que realiza la medición. Generalmente se usan las redes de telefonía móvil como proveedores de acceso a Internet a través de GPRS o UMTS. El acceso a las correcciones en tiempo real se realiza a través del Caster NTRIP (Networked Transport of RTCM vía Internet Protocol) de la Red GNSS de Castilla y

León. NTRIP es un protocolo estándar diseñado para difundir en tiempo real los datos procedentes de receptores GNSS en Internet. Para poder usar las correcciones suministradas por la Red es necesario que el receptor GNSS tenga un cliente NTRIP configurado con los datos de acceso al Caster NTRIP de Castilla y León que se indican a continuación:

- URL: HYPERLINK"http:/www.ntrip.itacyl.es:2101/"

- IP: 195.76.182.228

- Puerto: 2101

- Usuario y contraseña: los definidos por el usuario del servicio.

La tabla del Caster NTRIP ofrece 44 flujos de datos diferentes (llamados puntos de montaje o mountpoints), con la información de cada uno de ellos que incluye el nombre abreviado (ej: VALA3, VRS3....), el nombre completo, el tipo de datos (RTCM), los mensajes que emite, etc.

Se distinguen dos tipos de puntos de montaje:

1.- Puntos de montaje de estaciones individuales. El usuario necesita conocer cuál es la estación base más cercana y elegirla manualmente. Las observaciones en RTK de cada estación individual son suministradas solo en formato RTCM 3.0 (nombres acabados en 3, ej: BURG3). El uso de estaciones individuales se recomienda solo para conexiones permanentes de otras redes GNSS que utilicen los datos brutos, o para usuarios avanzados que lo necesiten por algún motivo concreto.

2.- Puntos de montaje de solución de Red. El usuario, dentro de la zona de cobertura geográfica de la Red, no necesita conocer la estación más cercana sino que automáticamente envía su posición y recibe una corrección diferencial óptima que integra los datos de los receptores circundantes de la Red. La corrección se envía en distintos formatos o conceptos, a elegir por el usuario:

- VRS (Virtual Reference Station), en 3 formatos distintos: RTCM 2.3 (VRS2), RTCM 3.0 (VRS3), y CMR (VRSC).

- MAC (Master-Auxiliary Concept), en formato RTCM 3.1 (MAC3).

- DGPS (Corrección solo de código), en formato RTCM 2.1 (DGNSS2).

La recomendación desde el ItaCyL es conectarse siempre a las soluciones de red (VRS) por su ventaja frente a una estación individual, ya que son independientes del área de trabajo, y en caso de caída de una estación individual el usuario seguirá recibiendo correcciones de las estaciones circundantes.

Dentro de las soluciones de Red, se recomienda por defecto VRS3, que emite en formato RTCM3, este formato consume menos ancho de banda que el RTCM2.3, es un estándar soportado por todos los fabricantes, y además emite mensajes adicionales que los receptores más modernos aprovechan ventajosamente para mejorar en ciertos casos la fiabilidad.

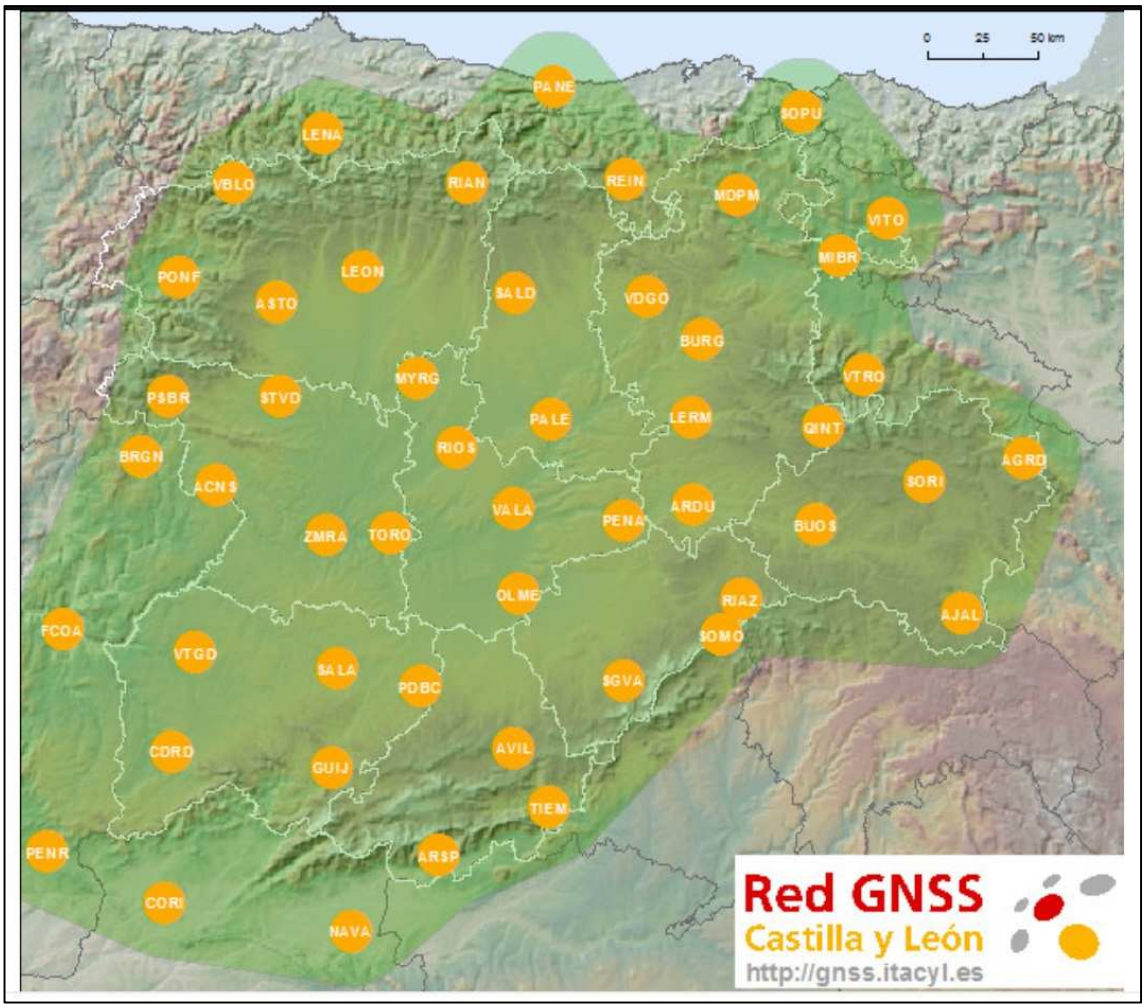


Imagen 9.- Puntos de montaje o estaciones GNSS de la red del ItaCyL

3.7. Conceptos geodésicos relacionados con el sistema GPS

La posición de un punto en el espacio se fija en relación a un objeto físico y se expresa a partir de ciertas cantidades convencionalmente definidas.

El sistema de referencia es el objeto físico real, concepto relacionado con la mecánica. Los sistemas de referencia se distinguen entre sí en función de su estado de movimiento.

Los sistemas de referencia se materializan en el terreno, con lo que se denomina marco de referencia (conjunto de coordenadas de una red de estaciones fijadas por algún convenio).

Las “cantidades” que expresan la posición del punto en el sistema de referencia estarían relacionadas con el sistema de coordenadas. Este es un concepto matemático y dado un sistema de referencia podemos emplear varios sistemas de coordenadas para establecer la posición del punto: coordenadas cartesianas, esféricas, etc.

La forma de la tierra se aproxima a una superficie denominada geoide que coincidiría con la superficie media de los océanos prolongada por debajo de los continentes. La complejidad

de la expresión matemática de esta superficie la hace poco útil, por lo que se aproxima a la superficie de un elipsoide de revolución. Los valores numéricos que definen este elipsoide son publicados periódicamente por la Asociación Internacional de Geodesia (IAG: International Association of Geodesy). Dichos valores sirven como estándar durante un largo período de tiempo para disciplinas como la geodesia, la ingeniería y la navegación.

El concepto geodésico de Datum, necesario para la elaboración de cartografía o representar sobre esta puntos determinados, que definido como el punto tangente al elipsoide y al geoide, donde ambos son coincidentes. Se compone de:

1.- Un elipsoide de referencia.

2.- Un "Punto Fundamental" en el que el elipsoide y la tierra son tangentes. De este punto se han de conocer la longitud, latitud y el azimut de una dirección determinada.

En geodesia existen dos Datum : el horizontal y el vertical, siendo este último la superficie de referencia respecto a la que se definen las altitudes.

3.7.1. Sistema de referencia global WGS84

El marco de referencia empleado con el sistema GPS tiene su origen en el centro de la tierra y gira solidario con ella, es por tanto un marco de referencia no inercial. Es necesario definir un sistema con referencia a un Datum Universal con cobertura para toda la superficie terrestre, evitándose así la territorialidad del resto de Datums existentes.

Los valores que definen el elipsoide de revolución correspondiente al sistema de referencia geodésico mundial WGS84, creado con el propósito antes expuesto y al que está referida la posición proporcionada por los satélites GPS son los siguientes:

- Longitud del semieje mayor: a=6.378.137,0 m
- Longitud del semieje menor: b= 6.356.752,31424 m
- Constante gravitacional geocéntrica de la tierra, incluyendo la atmósfera: $\mu=3986004,418*108\ m^3s^{-2}$.
- Velocidad angular de la tierra: $\omega=7,292115*10^{-5}\ rad*s^{-1}$

El sistema WGS84 se define como un sistema cartesiano geocéntrico del siguiente modo:

- El origen (0, 0, 0) coincide con el centro de la tierra.
- El eje z se encuentra en la dirección del polo norte terrestre convencional, definido por el BIH (Bureau International de l'Heure) para la época 1.984.
- El eje x es la intersección del plano meridiano de Greenwich, definido por el BIH para la época 1.984 y el plano del Ecuador.
- El eje y es perpendicular a los dos anteriores.

Las coordenadas que se obtienen de la constelación de satélites pueden ser cartesianas en el espacio respecto al centro de masas de la Tierra (X, Y, Z) o geográficas. Las coordenadas geográficas son la latitud φ, la longitud λ y la altitud h definidas de la siguiente manera:

- La latitud φ de un punto P es el ángulo entre la normal al elipsoide que pasa por P y el plano ecuatorial. La latitud varía entre +90º (Polo norte) y -90º (Polo sur).

- La longitud λ de un punto P es el ángulo entre el plano meridiano de Greenwich y el plano meridiano que pasa por el punto P medido sobre el plano del Ecuador. La longitud es positiva hacia el este y negativa hacia el oeste respecto del meridiano de Greenwich y sus valores se encuentran entre -180º y 180º.

- La altitud geodésica h de un punto P es la distancia a lo largo de la normal al elipsoide entre la superficie del elipsoide y el punto P. La altitud ortométrica sería la altitud del punto respecto del geoide y la diferencia entre la altitud respecto del geoide y respecto del elipsoide se conoce como ondulación del geoide N.

3.7.2. Sistema de referencia local ETRS89

El elipsoide de revolución que mejor se adapte al geoide en la zona de trabajo, con un punto donde ambos coinciden (también coinciden las verticales al elipsoide y al geoide) es la solución adoptada, constituyendo el concepto de Sistema Geodésico de Referencia Local. A largo de la historia diversos elipsoides se han utilizado para definir el Sistema de Referencia de cada país, de tal forma que se define aquel que mejor se ajuste al geoide.

La Subcomisión de la Asociación Internacional de Geodesia (IAG) para el marco de referencia europeo (EUREF), recomendó que el Sistema de Referencia Terrestre para Europa que debía ser adoptado, fuera el denominado European Terrestrial Reference System 1989 (ETRS89).

El Real Decreto 1071/2007 establece ETRS89 como sistema se referencia geodésico oficial en España para la representación geográfica y cartográfica en el ámbito de la Península Ibérica y las Islas Baleares. Este sistema tiene asociado el elipsoide GRS80 y está materializado por el marco que define la Red Geodésica Nacional por Técnicas Espaciales (REGENTE).

3.7.3. Transformaciones entre coordenadas geográficas WGS84 y cartesianas ETRS89

El uso del sistema GPS en topografía o geodesia implica usar, por motivos de simplicidad, coordenadas cartesianas (x, y) en lugar de geográficas. Si el uso del sistema es la navegación, puede ser más común el empleo de coordenadas geográficas (longitud, latitud).

La transformación de coordenadas geográficas a cartesianas se conoce como problema directo. Existen varios métodos para resolver este problema. Me centraré en el método conocido como las fórmulas de Coticchia-Surace.

Es un método fácil de programar, planteado por Alberto Cottichia y Luciano Surace en el “Bolletino di Geodesia e Science Affini”. La precisión del método ronda el centímetro cuando se utilizan suficientes decimales.

Las coordenadas cartesianas del sistema ETRS89, se corresponden con la proyección UTM (Universal Transverse Mercator). La representación cartográfica del globo terrestre, ya sea considerando este como una esfera o un elipsoide, supone un problema, ya que no existe modo alguno de representar toda la superficie desarrollada sin deformarla e incluso de representarla fielmente, ya que la superficie de una esfera no es desarrollable en su conversión a un soporte papel, a una representación plana.

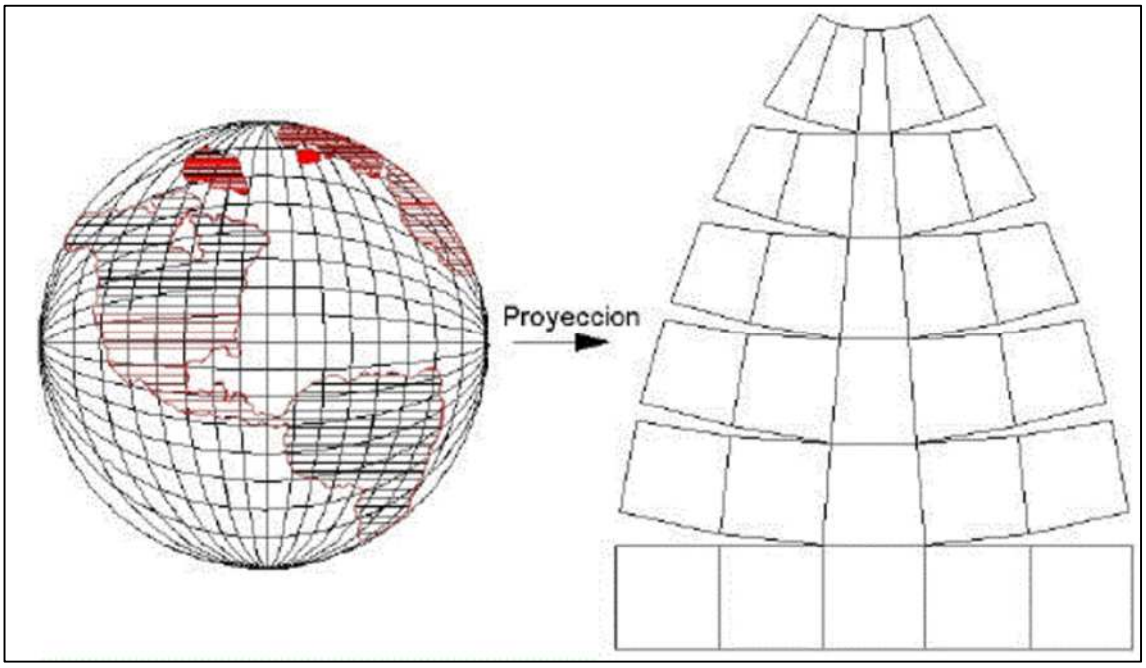


Imagen 10.- Paso de la representación sobre la superficie de la tierra a la proyección plana

La proyección UTM se emplea habitualmente dada su importancia militar, sobre todo desde que el Servicio de Defensa de Estados Unidos la todo como referencia para su empleo a nivel mundial en la década de 1940.

La proyección UTM es una proyección cilíndrica al emplear un cilindro para proyectar las coordenadas geográficas. El cilindro está situado tangente al elipsoide en el ecuador. Los meridianos y paralelos se representan como una cuadrícula de manera que una recta oblicua situada entre dos paralelos forma un ángulo constante con los meridianos.

Se define un Huso como las posiciones geográficas que ocupan todos los puntos comprendidos entre dos meridianos. La proyección UTM emplea husos de 6º de longitud, generando en cada huso un meridiano central equidistante 3º de longitud con los extremos del huso. Los husos se generan a partir del meridiano de Greenwich. Cada huso emplea un cilindro de proyección distinto, siendo cada uno de los cilindros empleados tangente al meridiano central del huso. Esta tangencia del cilindro de proyección al meridiano central del huso, implica que únicamente esta línea de tangencia conserva la distancia real (es por lo tanto automecoica, es decir su deformación lineal k es 1). La distorsión entre las distancias reales y las proyectadas aumenta a medida que nos alejamos del meridiano central de tangencia, aumentando por lo tanto del valor de deformación lineal k. Para evitar que las magnitudes lineales aumenten a medida que aumenta la distancia al meridiano central se considera que la posición del cilindro de proyección sea secante al elipsoide que representa la tierra en dos meridianos, creándose por lo tanto dos líneas para las cuales el coeficiente k sea igual a 1.

Introducidos ciertos conceptos geodésicos necesarios para abordar el tema de la transformación de coordenadas, a continuación paso a explicar el paso de coordenadas geográficas a cartesianas.

Para la resolución del problema directo es necesario conocer los datos básicos de la geometría del elipsoide de revolución (semieje mayor y menor) al que están referidas las coordenadas geográficas, en este caso el elipsoide WGS84 cuya geometría fue enunciada en el punto 3.6.1.

Las fórmulas matemáticas que relacionan las coordenadas geográficas en el sistema WGS84 y las coordenadas rectangulares en el sistema ETRS89 son las siguientes:

- Como primer paso para resolver el problema directo es necesario convertir los grados sexagesimales (grados, minutos y segundos) de la latitud y longitud a grados sexagesimales en notación decimal (lo que se conoce como grados decimales) de la siguiente manera:
  - Grados decimales= grados+minutos/60+segundos/60
- Como la mayor parte de los pasos posteriores se realizan en radianes, hay que transformar los grados decimales a radianes:
  - Radianes=Grados decimales$*\pi/180$
- Cálculo del signo de la longitud: la longitud es positiva si está al este el meridiano de Greenwich y negativa si está al oeste del meridiano de Greenwich
- Cálculo del huso o zona UTM en la que se encuentra el punto a transformar.
  - Huso=número entero de [(Grados decimales/6)+31]
- Cálculo del meridiano central del huso de las coordenadas geodésicas del punto a transformar:
  - λ_0=Huso$*$6-183
- Distancia angular entre el punto a transformar y el meridiano central del huso:
  - $\Delta\lambda=\lambda-\lambda 0$
- Excentricidad del elipsoide WGS84:
  - $e = \sqrt{a^2 - b^2}/a$
  - Segunda excentricidad, $e' = \sqrt{a^2 - b^2}/b$
- Radio polar de curvatura: $c = a^2/b$
- Aplanamiento: $\alpha = {a - b}/{a}$
- Parámetro $A=\cos\varphi * \sin\Delta\lambda$
- Parámetro $\xi=\frac{1}{2} * \ln\frac{1+A}{1-A}$

- Parámetro η=$\tan^{-1}(\tan\varphi/\cos\Delta\lambda)$
- Parámetro υ=$\frac{c}{(1+e'^2*cos^2\varphi)^{1/2}}$
- Parámetro ς=$\frac{e'^2}{2}*\xi^2*cos^2\varphi$
- Parámetro A_1=$\sin 2\varphi$
- Parámetro A2= A1$*\cos^2\varphi$
- Parámetro J_2=$\varphi+\frac{A1}{2}$
- Parámetro J_4=$\frac{3*J2+A2}{4}$
- Parámetro J_6=$\frac{5*J4+A2*cos^2\varphi}{3}$
- Parámetro α=$\frac{3}{4}*e'^2$
- Parámetro β=$\frac{5}{3}*\alpha^2$
- Parámetro ϒ=$\frac{35}{27}*\alpha^3$
- Parámetro B_ϕ=0,9996$*c*(\varphi-\alpha*J_2+\beta*J_4-\Upsilon*J_6)$

Obtenidos todos los parámetros anteriores, se procede a calcular las coordenadas UTM finales:

- x=500.000+(ξ$*$ υ$*(1+\frac{\varsigma}{3})$)
- y= η$*$ υ$*$(1+ ς$)+B_\phi$, si el punto a transformar pertenece al hemisferio sur, al valor obtenido para la coordenada y en la expresión anterior hay que sumar 10.000.000.

Capítulo 4. Entorno de programación Android. Lenguaje Java

4.1. Reseña histórica

Android es el nombre con el que se denomina al sistema operativo creado para el desarrollo de aplicaciones móviles por la compañía Android Inc. (compañía de pequeñas dimensiones creada poco tiempo antes nacida para generar aplicaciones para dispositivos móviles).

El objetivo con el que nació Android fue el de promover estándares abiertos en teléfonos y ordenadores móviles.

Una cronología del desarrollo de Android contiene los siguientes hitos:

1.- En el año 2005 dicha compañía fue adquirida por Google.

2.- En el año 2007 se crea el consorcio Open Handset Alliance, formado por empresas (Google, Intel, Motorola, T-Mobile, Samsung, Ericsson, Toshiba, Vodafone, etc) punteras en aquella época den telefonía móvil, con el objetivo de promover el diseño y la difusión de la plataforma Android.

3.- En el año 2007 se lanza la primera versión de Android SDK (Software Developers Kit) y se abre Android Market, plataforma para la descarga de aplicaciones.

4.- En el año 2008 aparece el primer móvil con Android, el T-Mobile G1.

5.- En el año 2010 Android ya está consolidado como uno de los sistemas operativos ara móviles más utilizados.

6.- En el año 2011 Android alcanza una cuota de mercado superior al 50%.

7.- En el año 2012 se suprime Android Market y aparece Google Play Store. En este portal se unifican la descarga de aplicaciones y de contenido. Android alcanza una cuota de mercado del 70%.

Android destaca por las siguientes características:

1.- Su popularidad: en la actualidad existen más de un millón de aplicaciones que utilizan este sistema operativo.

2.- Su seguridad, por haberse detectado pocas vulnerabilidades en su estructura.

3.- Desde la plataforma Google Play se tiene acceso a todo tipo de aplicaciones, ya sean gratuitas o de pago.

4.- Dispone se soporte para streaming.

5.- Soporta videollamadas.

6.- Permite que el terminal pueda ser empleado como punto de acceso inalámbrico.

7.- Pueda adaptarse a multitud de resoluciones de pantalla.

8.- Soporta conexiones wifi, bluetooth, UMTS, GSM, etc.

9.- Permite el envío de mensajes MMS y SMS.

10.- Tiene capacidad para trabajar con archivos MP3, GIF, JPEG, PNG, BMP, WAV, MPEG-4, etc.

11.- Es una plataforma de desarrollo libre basada en Linux y de código abierto.

12.- Adaptable a cualquier tipo de dispositivo: teléfonos móviles, tabletas, relojes, cámaras, televisores, etc.

13.- Asegurada la portabilidad ya que las aplicaciones se desarrollan en Java y esto nos asegura que las aplicaciones podrán ser ejecutadas en cualquier CPU, presente y futura, gracias al concepto de máquina virtual.

14.- La arquitectura se basa en componentes de internet, por ejemplo, la interfaz de usuario se basa en ficheros XML.

15.- Android tiene como filosofía el que los dispositivos estén conectados siempre a internet.

16.- Dispone de gran cantidad de servicios incorporados: bases de datos SQL, localización por medio de GPS, navegador, reconocimiento de voz, etc.

17.- Android está optimizado para dispositivos de baja potencia y poca memoria.

18.- Alta calidad de gráficos y sonido.

A lo largo de su existencia, desde el año 2.007) Android ha tenido distintas versiones, las cuales se denominan con el nombre de postres y su aparición está ordenada alfabéticamente. De esta manera la secuencia de versiones hasta la fecha es la siguiente (entre paréntesis la versión correspondiente numérica): Apple Pie (1.0), Banana Bread (1.1), Cupcake (1.5), Donut (1.6), Éclair (2.0 y 2.1), Froyo (2.2), Gingerbread (2.3), Honeycomb (desde 3.0 a 3.2), Ice cream sándwich (4.0), Jelly Bean (desde 4.1 a 4.3), Kit Kat (4.4), Lollipop (5.0 y 5.1), Marshmallow (6.0), Nougat (desde 7.0 y 7.1) y Oreo (8.0 y 8.1). La última versión no mantiene la tradición de los nombres de postres y se denomina P (versión 9.0).

Android se ha convertido en el sistema operativo para dispositivos móviles más potente y demandado, con una cuota de mercado del 80%, muy por encima del sistema operativo iOS (sistema operativo de la compañía Apple).

Imagen 11.- Logotipo actual de Android

4.2. Arquitectura de Android

La arquitectura de Android está formada por cuatro capas, todas ellas basadas en software libre:

1.- Aplicaciones: conjunto de aplicaciones instaladas en una máquina Android. Normalmente las aplicaciones Android están escritas en lenguaje de programación Java, para lo cual podemos utilizar Android SDK. El SDK no acaba de ofrecer todo lo disponible del entorno de ejecución Java (JRE) pero es compatible con una gran parte de él. También se pueden desarrollar aplicaciones en C/C++, utilizando Android NDK (Native Development Kit).

2.- Entorno de aplicación: plataforma de desarrollo para aplicaciones diseñada para la reutilización de componentes.

Los servicios más importantes son:

a) Views: conjunto de vistas, para visual de los componentes.

b) Resource Manager: proporciona el acceso a recursos que no son del código.

c) Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.

d) Notification Manager: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.

e) Content Providers: mecanismo para acceder a datos de otras aplicaciones.

3.- Librerías nativas y Runtime de Android.

Android incluye un conjunto de librerías en C/C++ usadas en varios de sus componentes. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son: SGL (motor de gráficos 2D); FreeType (fuentes en bitmap y renderizado vectorial); SQLite (motor de bases de datos relacionales disponible para todas las aplicaciones); SSL (proporciona servicios de encriptación).

El Runtime de Android está basado en el concepto de máquina virtual utilizado en Java. La máquina virtual de Java (JRE, Java Runtime Environment) toma los archivos compilados y los transforma en función del sistema operativo empleado. La máquina virtual de Java no fue posible utilizarla debido a las limitaciones de los dispositivos móviles en los que debían correr las aplicaciones Android (poca memoria y procesador limitado). Por lo tanto, Google tomó la decisión de crear una nueva máquina virtual, Dalvik (sus orígenes se remontan al año 2.005), que superase las limitaciones anteriores. Para ello la nueva máquina virtual optimiza recursos ejecutando ficheros Dalvik (formato .dex optimizado para ahorrar memoria) y está basada en registros. A partir de la versión de Android 5.0 se reemplaza Dalvik por ART, reduciendo el tiempo de ejecución un 33%. En el Runtime también se incluye el módulo Core Libraries, con la mayoría de librerías disponibles en lenguaje Java.

4.- El núcleo Linux: el núcleo de Android está formado por el sistema operativo Linux, versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

4.3. Instalación del entorno de desarrollo Android Studio

Google dispone de un paquete de software Android SDK, que incorpora todas las herramientas necesarias para el desarrollo de aplicaciones en Android, como pueden ser: conversor de código, depurador, librerías, emuladores, documentación, ejemplos de código, etc.

Un número importe de desarrolladores prefieren usar un entorno de desarrollo integrado (IDE), que agrupa un editor de código con todas las herramientas de desarrollo. El IDE actual recomendado por Google es Android Studio. Android Studio dispone de diversas versiones desde su lanzamiento en el año 2.014. Desde este año Google potenció esta IDE en detrimento de otra, la conocía como Eclipse. Actualmente la última versión es la 3.1.

El IDE Android Studio se puede descargar desde la siguiente dirección de internet: http://www.developer.android.com/sdk/. En esta dirección elegiremos el sistema operativo en el que se va a instalar (Mac o Windows) y la versión que queremos descargar. Se descarga un fichero ejecutable que activa la instalación de Android Studio.

Una vez instalado Android Studio, es importante revisar los paquetes instalados. Para ello en SDK Manager se muestran los paquetes instalados y los que se pueden instalar o actualizar.

Es conveniente tener instalados o actualizados de la última versión de Android, los siguientes paquetes: Android X.X Platform, Google APIs Android X, algún System Image para crear emuladores, Android SDK Build.tools, Android SDK Tools, Android SDK Platform-tools, Android Support Repository, Android Support Library, Google Play Services y Android USB driver.

Los emuladores permiten emular en el ordenador los dispositivos basados en Android en los que se ejecutarán las aplicaciones que creemos. Estos emuladores se conocen como dispositivo virtual Android (AVD) y de esta forma Android pone a nuestra disposición una manera de reproducir las aplicaciones en una gran variedad de teléfonos, tabletas, relojes o televisores con cualquier versión de Android.

Antes de empezar a programar en Android es importante elegir la versión del sistema para la que deseamos realizar la aplicación, ya que hay clases y métodos que están disponibles a partir de una determinada versión. Cuando se ha lanzado una nueva versión, esta siempre ha sido compatible con las versiones anteriores, solo se suman nuevas funcionalidades o se modifican las existente, pero no se elimina ninguna funcionalidad, simplemente las modificadas se etiquetan como obsoletas. Es conveniente consultar si necesitamos alguna característica especial que solo esté disponible a partir de una versión, ya que todos los usuarios con versiones de Android inferior a la seleccionada no podrán instalar la aplicación, por lo tanto es conveniente seleccionar la menor versión posible que nuestra aplicación pueda soportar. No obstante existen librerías de compatibilidad para poder incorporar ciertas funcionalidades en cualquier versión de Android.

Un proyecto de Android Studio puede contener varios módulos.

Cada módulo está formado por:

1.- Un descriptor de la aplicación, fichero *manifest.xml*. Este fichero describe la aplicación Android. Se define su nombre, paquete, icono, estilos, etc. Se indican las actividades, las intenciones, los servicios y los proveedores de contenido de la aplicación. También se declaran los permisos que requerirá la aplicación. Se indica la versión mínima de Android para poder ejecutarla, el paquete de Java, la versión de la aplicación, etc.

2.- El código fuente en Java, localizado en la carpeta *java*. Esta carpeta contiene el código fuente de la aplicación.

3.- Una serie de ficheros con recursos empleados en la aplicación, localizados en la carpeta *res*, que a su vez se descompone en otras carpetas:

a) Carpeta *drawable,* donde se almacenan los ficheros de imágenes y descriptores de imágenes en xml.

b) Carpeta *mipmap,* con la misma finalidad que la carpeta *drawable* pero con la diferencia que si ponemos los gráficos en *mipmap*, estos no son rescalados para adaptarlos a la densidad gráfica del dispositivo donde se ejecuta la aplicación, sino que se buscara en las subcarpetas el gráfico con la densidad más parecida y se utilizará directamente. Es aconsejable guardar los iconos de la aplicación en esta carpeta.

c) Carpeta *layout*, que contiene los ficheros xml con las vistas de la aplicación. Las vistas permiten configurar las diferentes pantallas que compondrán la interfaz de usuario de la aplicación y se emplea un formato similar al html.

d) Carpeta *menú,* ficheros xml con los menús de cada actividad.

e) Carpeta *values,* ficheros xml que contienen valores usados en la aplicación de manera que cambiando dichos valores en esos ficheros no es necesario cambiar el código fuente. En el fichero *colors.xml,* se definen los tres colores primarios de la aplicación. En *dimens.xml,* se definen los márgenes horizontal y vertical. En *strings.xml,* se definen todas las cadenas de caracteres de la aplicación. En *styles.xml,* se definen los estilos y temas de la aplicación.

f) Carpeta anim, que contiene ficheros xml con animaciones de vistas.

g) Carpeta animator, que contiene ficheros xml con animaciones de propiedades.

h) Carpeta xml, con otros ficheros xml que necesita la aplicación.

i) Carpeta raw, que contiene ficheros adicionales que no se encuentran en formato xml.

4.- Carpeta Gradle Scripts. Contiene una serie de ficheros Gradle que permiten construir y compilar la aplicación. El fichero más importante es build.gradle que es donde se configuran las opciones de compilación del módulo.

4.4. Componentes de una aplicación

Hay una serie de elementos clave que resultan imprescindibles para desarrollar aplicaciones en Android.

1.- Vistas (view). Las vistas son elementos que componen la interfaz de usuario de una aplicación, por ejemplo, un botón o una entrada de texto. Todas las vistas son objetos descendientes de la clase View de Java y por lo tanto pueden ser definidas utilizando código de Java pero también, y más habitual, es definirlas mediante un fichero xml y dejar que el sistema cree los objetos a partir de este fichero.

2.- Layout. Un layout es un conjunto de vistas agrupadas de una determinada forma. Se disponen de diferentes tipos de layouts para organizar las vistas de forma lineal, en forma de cuadrícula o indicando la posición absoluta de cada vista. Los layouts son objetos descendientes de la clase view. Los layouts se pueden definir mediante código o de manera más habitual y simple mediante código xml.

3.- Actividad (activity). Lo que comúnmente conocemos como pantallas de una aplicación en lenguaje Android se llaman activity (actividad). Son elementos básicos de visualización cuya función es la creación de la interfaz de usuario. Una aplicación suele tener varias activity para relacionarse con el usuario y todas ellas pertenecen a una clase descendiente de Activity.

4.- Servicio (service). Un servicio es un proceso que se ejecuta sin la necesidad de una interacción con el usuario. En Android se disponen de dos tipos de servicios: servicios locales, que se ejecutan en el mismo proceso y servicios remotos que se ejecutan en procesos separados.

5.- Intención (intent). Una intención representa la voluntad de realizar alguna acción. Se utiliza cada vez que queremos lanzar una actividad, lanzar un servicio, comunicarnos con un servicio, intercambiar información entre componentes, etc. Los componentes lanzados pueden ser internos o externos a la aplicación a la que pertenecen.

6.- Fragment. Un fragment está formado por la unión de varias vistas para crear un bloque funcional de la interfaz de usuario. Surgieron con la aparición de las tabletas, cuyas pantallas de mayor tamaño que las de los teléfonos móviles generaban resultados de la aplicaciones nos satisfactorios cuando estas estaban pensadas para ejecutarse en los teléfonos móviles.

7.- Receptores de anuncios (broadcast receiver). Un receptor de anuncios recibe anuncios y reacciona ante ellos. Loa anuncios pueden ser originados por el sistema (anuncio sobre el estado de la batería) o por las aplicaciones.

8.- Proveedores de contenido (content provider). Android define los content provider como mecanismo estándar para que las aplicaciones puedan compartir datos sin necesidad de comprometer la seguridad del sistema de ficheros. Con este sistema podemos acceder a los datos de otras aplicaciones o proporcionar datos a otras aplicaciones.

4.5. Introducción al lenguaje de programación Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1.995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarían a menos que tengamos Java instalado y cada día se crean más. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes. El lenguaje de programación Java se caracteriza por ser un lenguaje simple, familiar al estar basado en el lenguaje C, de rápido aprendizaje y seguro. Además como característica positiva a su favor, la descarga de Java es gratuita.

Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles, juegos, contenido basado en web y software de empresa. Java cuenta con más de 9 millones de desarrolladores en todo el mundo. Su importancia se puede resumir en cifras:

1.- El 97% de los ordenadores de empresas ejecutan Java.

2.- El 89% de los ordenadores en Estados Unidos ejecutan Java.

3.- Java es la primera opción para los desarrolladores y la primera plataforma de desarrollo.

4.- Tres mil millones de teléfonos móviles ejecutan Java.

5.- El 100% de los reproductores de Blu-ray incluyen Java.

6.- Ciento veinticinco millones de televisores ejecutan Java.

Los desarrolladores de software prefieren Java porque ha sido probado, ajustado y ampliado por una amplia comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas de Java. Java está diseñado para el desarrollo de aplicaciones portátiles de elevado rendimiento para el más amplio rango de plataformas informáticas posible. Al poner a disposición de todo el mundo aplicaciones en entornos heterogéneos, las empresas pueden proporcionar más servicios y mejorar la productividad, las comunicaciones y colaboración del usuario final y reducir drásticamente el costo de propiedad tanto para aplicaciones de usuario como de empresa. Java permite al desarrollador: escribir software en una plataforma y ejecutarla virtualmente en otra; crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles; desarrollar aplicaciones de servidor; combinar aplicaciones o servicios con un gran nivel de personalización; escribir aplicaciones para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico.

La seguridad de Java se debe a su máquina virtual, la cual realiza comprobaciones seguridad. Por otra parte el lenguaje de programación de Java carece de características inseguras.

Java permite ejecutar una aplicación desarrollada en su lenguaje, independientemente de la plataforma en la que se ejecute, por lo cual también tiene sus inconvenientes pues es necesario definir la representación, por ejemplo de los números, válida para las distintas plataformas. El chequeo del código de la aplicación se hace tanto en tiempo de depuración como en tiempo de ejecución.

Como inconvenientes de Java cabe mencionar que es un lenguaje interpretado, por lo que la ejecución es poco eficiente, y que no permite el acceso remoto al hardware.

Java es un lenguaje de programación orientado a objetos he incorpora las características más importantes para este tipo de programación: encapsulamiento, polimorfismo, herencia y enlace dinámico. Es un lenguaje compilado, generando ficheros de clases compilados que son interpretadas por la máquina virtual java la cual mantiene el control sobre las clases que se están ejecutando.

Para ilustrar el concepto de programación orientada a objetos podemos tomar ejemplos dela vida real. Podemos apreciar que casi todo se compone de objetos (coche, lapicero, casa, etc). Por ejemplo, en el caso de los coches, a pesar de la diferencia de aspecto entre dos coches, se puede concluir que ambos tienen unas características comunes como son poseer ruedas, un volante, asientos, etc. Es decir, podemos interpretar que el BMW y el SEAT son objetos, y el tipo o clase, de esos objetos es coche.

Como podemos ver esta clase no sólo describe que cosas posee un coche, sino que describe que acciones puede realizar un coche (acelerar, frenar y girar). Es decir, una clase define los atributos y las acciones(o métodos) que puede realizar un objeto de la clase.

Además podemos comprobar que un objeto puede estar formado por otros objetos, por ejemplo el coche posee 4 objetos de la clase rueda.

En el mundo real los objetos se relacionan entre sí, un objeto puede pedir a otro que realice alguna acción por él. En la vida real una persona acelera, pero lo hace pidiéndoselo al coche, que es quien realmente sabe acelerar. El hecho de que un objeto llame a un método de otro objeto se indica diciendo que el primer objeto ha enviado un mensaje al segundo objeto. El nombre del mensaje es el nombre de la función llamada.

Podemos entender que a partir de la clase podemos fabricar objetos. Ese objeto construido se le denomina instancia, y al proceso de construir un objeto se le llama instanciación.

Cuando se construye un objeto es necesario darle un valor inicial a sus atributos. Existe un método especial en cada clase llamado constructor, el cual es ejecutado de forma automática cada vez que es instanciada una variable. Generalmente el constructor se llama igual que la clase y no devuelve ningún valor. Es posible que exista más de un constructor en una clase, diferenciados sólo en los parámetros que recibe, pero en la instanciación sólo será utilizado uno de los constructores.

Es recomendable emplear el constructor para inicializar las variables internas del objeto, o para obtener recursos.

Tomando como ejemplo un televisor, el usuario que dispone del mando a distancia puede manejar el volumen del televisor, pero a dicho usuario no le agradaría que otra persona pudiese manejar el volumen. Lo mismo sucede con objetos: un objeto que no quiere que otro objeto llame a un método o acceda a un atributo debe de definir dichos atributos y métodos como privados. En cambio si los definimos como públicos cualquier objeto podrá utilizarlos.

Retomando el ejemplo de los coches, pensemos en un coche deportivo y en un coche utilitario, ambos son coches, pero hacemos la distinción deportivo y utilitario porque sus diferencias son lo suficientemente importantes para ser tenidas en cuenta, pero no tanto como para pensar que uno es un coche y el otro no. Esta situación se interpreta pensando en que existen dos clases: deportivo y utilitario, pero que ambas, además de sus propias

características (atributos y métodos) poseen las de la clase coche, esto es la herencia. Denominaremos clase base a la clase de la cuál heredamos y derivada a la clase que hereda de la clase base.

Existen dos tipos de herencia sencilla y múltiple. Sencilla significa que sólo heredamos de una clase base, mientras que múltiple indica que tenemos varias clases base (por ejemplo un hidroavión hereda de barco y de avión). Java sólo soporta herencia simple.

Al utilizar la herencia aparecen dos conceptos: super y this. This representa al objeto completo, en cambio super sólo representa la parte heredada de la clase base.

Los constructores no son heredados, pero sí llamados. Es decir, cuando se construye un objeto de la clase derivada se llama al constructor de la clase derivada, pero antes de comenzar a ejecutarse se llama al constructor de la clase base, que tras ejecutarse continua la ejecución del constructor de la clase derivada.

Supongamos ahora que tenemos algo que es capaz de encenderse, de apagarse, de iniciar una reproducción, de parar una reproducción, sin duda todos pensamos en un reproductor, pero por esa descripción encajan objetos como reproductor de casete, reproductor de CD, el vídeo, etc. Una persona que sabe iniciar una reproducción de un CD también sabe iniciar una reproducción de video, en ambos casos debe de encender el objeto, iniciar la reproducción, parar la reproducción y apagar el reproductor. Eso significa que para esa persona es transparente el tipo (clase) real del objeto reproductor que posea, ya que la persona sabe que puede ponerlo en marcha, apagarlo, sin necesidad de conocer la clase real, tan sólo debe de saber que es un objeto del tipo reproductor.

Existe otra característica de Java denominada polimorfismo, consistente en que varios objetos de distintas clases pueden recibir el mismo mensaje y son capaces de responderlo.

Java dispone de varias versiones:

1.- Java EE, conocida como Java Enterprise Edition. Esta versión se utiliza comúnmente para desarrollar proyectos en web, proyecto que han de navegar por internet.

2.- Java Server Enterprise, que es la que normalmente se utiliza para desarrollar aplicaciones móviles.

3.- Java Micro Edition, enfocada para el desarrollo de aplicaciones para equipos de bajo nivel, como mandos a distancia o nuestros electrodomésticos de uso diario.

Otros conceptos asociados a Java, además de su máquina virtual JRE definida anteriormente, son el JDK y JSE.

JDK es al acrónimo de Java Development Kit, y es el componente o kit de Java necesario para desarrollar aplicaciones en Java. Este kit contiene todas las clases y librerías necesarias para desarrollar nuevos programas en Java

JSE, es el acrónimo de Java Standar Edition y es una colección de aplicaciones del lenguaje Java útiles para muchos programas de la plataforma Java.

Capítulo 5. Aplicación Agrisoft

5.1. Recursos empleados

Para la realización de la aplicación Agrisoft, base de este PFM, se han empleado distintos recursos los cuales se describen a continuación:

1.- **Ordenador portátil Acer TravelMate 5730**, con procesador Intel Core 2 Duo T6570. Dispone de 500 GB de disco duro, 4 GB de memoria RAM, sistema operativo de 64 bits y 2,10Ghz de velocidad del procesador. El sistema operativo es Windows 7 Profesional.

2.- **Tablet Samsung Galaxy Tab A modelo T585**, con 32 GB de disco duro y 2 GB de memoria RAM, velocidad de 1,6 Ghz de la CPU con 8 núcleos, versión de Android 7.0, tarjeta SIM de datos para conectarse a internet, tarjeta externa de memoria microSD, batería de 7.300 mAh, una salida de datos mini USB, pantalla de 10,1´´, Wi-Fi, Bluetooth y GPS interno.

3.- **Receptor GPS u-Blox C94-M8P-3**. El receptor GPS empleado es propiedad de la Universidad de Salamanca y me ha sido cedido para la realización de este PFM. La casa U-Blox es un fabricante mundial en receptores GPS con soluciones adaptables a distintos clientes. U-Blox ofrece receptores GPS para sistemas de navegación en el automóvil, sincronización de tiempo para estaciones base celulares, vehículos aéreos no tripulados (UAV), seguimiento de personas y bienes y aplicaciones agrícolas. En kit original incluye dos receptores NEO-M8P, dos antenas UHF externas, dos antenas imantadas GPS externas, dos platos de tierra de antena y dos cables micro USB.

   El kit es doble en todos sus elementos porque está pensado para trabajar con estación base y estación móvil o rover de tal manera que el receptor fijo situado en una estación de coordenadas conocidas pueda emitir correcciones diferenciales vía radio que recibirá el receptor móvil para obtener una posición geográfica más precisa. En el desarrollo de este PFM solo se ha empleado una unidad de cada componente del kit ya que se ha optado por recibir correcciones diferenciales vía internet proporcionadas por la red de estaciones GNSS del ItaCyL.

   El receptor u-Blox C94-M8P-3 tiene la capacidad de aplicar correcciones a las observaciones GPS obtenidas por él. En este PFM dichas correcciones han sido obtenidas vía internet mediante la tablet arriba reseñada y mediante bluetooth han sido enviadas al receptor u-Blox con la aplicación Lefebure (descrita más abajo). El receptor u-Blox C94-M8P-3 no dispone de bluetooth por lo que ha sido necesario adaptarle un dispositivo bluetooth (modelo HC-06, descrito a continuación) para recibir dichas correcciones y de esa manera incorporarlas al cálculo.

Imagen 12.- Receptor GPS u-Blox C94-M8P-3

Dispone de varios interfaces de comunicación: puerto RS232, muerto micro USB como fuente de alimentación de salida y entrada de datos, pins para comunicación UART, conexión para comunicación radio UHF y conexión para antena GPS.

Mediante luces LED permite conocer el estado en el que se encuentra: luz azul fija significa que aún no está calculando posición; luz azul parpadeante significa que está calculando posición; luz verde parpadeante significa que está recibiendo correcciones diferenciales externas.

El receptor GPS u-Blox C94-M8P-3 no dispone de batería por lo que a través del puerto micro USB es necesario dotarle de corriente eléctrica.

El receptor GPS u-Blox C94-M8P-3 soporta los formatos de comunicación NMEA, UBX (propio de la empresa u-Blox) y RTCM versión 3.2. El formato NMEA proporciona a la aplicación desarrollada en este PFM la cadena de datos de la cual extrae determinados valores empleados en la misma: como son la precisión de la observación; la longitud y latitud de la antena del receptor y la velocidad. A continuación se presenta un extracto de del archivo con formato NMEA generado por el receptor u-Blox:

- La línea sombreada en azul proporciona detalles sobre la solución GPS obtenida. Incluye los números de los satélites utilizados en la solución y el DOP (dilución de la precisión, valor que indica el efecto de la geometría de los satélites en la precisión y que cuanto más pequeño sea más precisa será la solución aportada. En el ejemplo expuesto desde 3 a 85 son los 8 satélites que intervienen en el cálculo. 1,41 es el DOP, 0,75 es el HDOP (dilución de la precisión horizontal) y 1,19 es el VDOP (dilución de la precisión vertical). Los espacios entre las comas indican satélites que no intervienen en la solución.

- La línea sombreada en rojo muestra todos los satélites que el receptor podría encontrar en función de la máscara de elevación (ángulo sobre el horizonte a partir del cual se admiten señales de satélites) y los datos del almanaque.

- La línea sombreada en amarillo contienen las coordenadas geográficas (latitud, longitud y altitud) de la antena del receptor que serán leídas por la aplicación Agrisoft.

- La línea sombreada en verde muestra el propio formato que tiene NMEA para la posición GPS de posición (4230.65817,N,00533.33543,W), velocidad sobre el suelo en nudos (0,013) y tiempo (fecha 180818).

- La línea sombreada en magenta muestra otra medida de velocidad más precisa.

- La línea sombreada en gris proporciona datos de ubicación y precisión en 3D y está constituido por hora en la que se corrigió algún problema (hora 10:17:48 en hora UTC), posición (4230.65817,N,00533.33543,W), calidad de la posición (en este caso 5 que quiere decir que la posición era flotante), número de satélites rastreados (12), dilución horizontal de la precisión (0,75), altitud en metros sobre el nivel medio del mar (795,8) y ondulación del geoide (50,5 diferencia de cotas entre el geoide y el elipsoide WGS84).

- La línea sombreada en morado muestra, entre otra información la longitud de la baselínea, la distancia entre el receptor fijo que emite correcciones y el receptor móvil.

$GNGSA,A,3,68,74,73,75,83,84,85,,,,,,1.41,0.75,1.19*12

$GPGSV,3,1,11,04,,,49,07,10,323,27,08,25,283,45,10,30,138,48*40

$GPGSV,3,2,11,16,86,184,41,18,03,227,,20,41,105,47,21,44,052,48*7C

$GPGSV,3,3,11,26,56,151,48,27,57,303,48,31,00,182,*45

$GLGSV,3,1,11,67,03,009,32,68,19,057,47,69,14,110,,73,16,176,39*69

$GLGSV,3,2,11,74,57,214,44,75,43,313,46,76,04,337,32,83,11,056,41*6A

$GLGSV,3,3,11,84,58,014,41,85,43,274,47,,,,48*60

$GNGLL,4230.65817,N,00533.33543,W,101747.00,A,D*6B

$GNTXT,01,01,01,DGNSS baseline big: 11 km*56

$GNRMC,101748.00,A,4230.65817,N,00533.33543,W,0.013,,180818,,,D*77

$GNVTG,,T,,M,0.013,N,0.024,K,D*3C

$GNGGA,101748.00,4230.65817,N,00533.33543,W,5,12,0.75,795.8,M,50.5,M,1.0,0028*77

La empresa u-Blox dispone de un software, llamado u-center, en el sistema operativo Windows, con el cual podemos configurar el receptor además de tareas de control de la señal que recibe. Se trata de un software que da soporte completo a todos los receptores GPS u-Blox. Con el podemos visualizar en tiempo real los satélites observados por el receptor, el código NMEA que genera, configurar los puertos de comunicación, velocidades de comunicación, archivos de grabación de observación en bruto, etc.

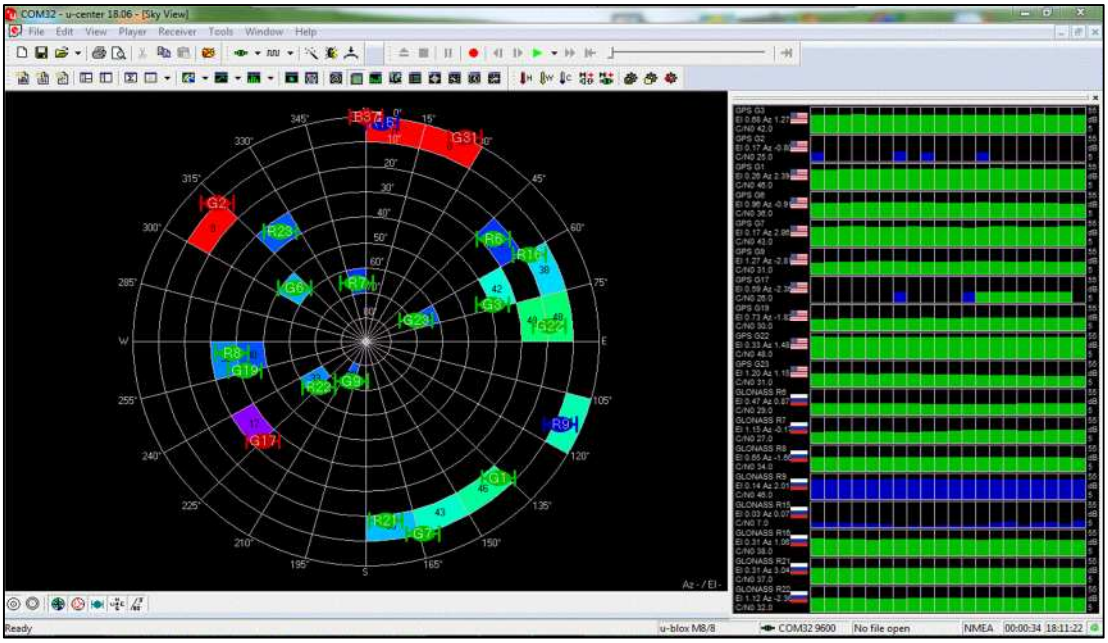


Imagen 13.- Señal recibida por el receptor u-Blox recogida en el software u-center

Las características técnicas del receptor GPS u-Blox C94-M8P-3 son las siguientes:

- Receptor GPS de alta precisión según el fabricante.
- Preparado para recibir señal de satélites de las constelaciones GPS, GLONASS y BeiDou.
- Puede recibir a la vez señal de satélites de dos constelaciones distintas.
- Puede registrar los datos de observación.
- 72 canales de recepción de señal.
- De la constelación GPS recibe las señales L1C/A en la frecuencia 1.575,42 MHz, de la constelación GLONASS recibe la señal L10F en la frecuencia 1.602 MHz+k*562,5 KHz donde k es el número de canal de la frecuencia del satélite y de la constelación BeiDou recibe la señal B1I emitida en la frecuencia 1.561,098 MHz.
- La frecuencia de la recepción de señal es programable y oscila entre 0,25Hz y 10Mhz.
- Puede trabajar en movimiento a velocidades de 500 m/s.
- Los tiempos que emplea en fijar posición oscila entre 1 segundo (arranque en caliente con cualquier constelación) y 29 segundos (para la constelación GPS y arranque en frío).
- La frecuencia de actualización de datos es de 10Hz para la toma de datos crudos de observación (ficheros RAW) y de 5 Hz para mediciones en RTK.
- La precisión en el cálculo, según el fabricante, de la posición horizontal (longitud+latitud o coordenadas X e Y cartesianas) es de 0,025 metros + 1 ppm (partes por millón con un límite de la longitud de la baselínea de 10 kilómetros. La baselínea es la distancia entre el receptor móvil y la estación base que emite correcciones diferenciales).
- Si la frecuencia de actualización de observaciones es importante (por ejemplo para elevadas velocidades de movimiento del receptor) se recomienda configurar el receptor para recibir solo señal de la constelación GPS.
- La antena proporcionada con el receptor es magnética.

En su configuración predeterminada, el receptor móvil NEO-M8P intentará proporcionar la mejor precisión de posicionamiento en función de los datos de corrección recibidos. Pasará al modo flotación RTK tan pronto como reciba una secuencia de entrada de mensajes RTCM 3. Una vez que el receptor haya resuelto las ambigüedades de la fase portadora, pasará al modo fijo RTK con mayor precisión. Una vez en modo RTK fijo las precisiones esperables son del orden del centímetro. Por lo general tarda menos de 60 segundos en resolver las ambigüedades y pasar del modo RTK flotante al modo RTK fijo. Este periodo de tiempo se conoce como convergencia.

El receptor móvil intentará proporcionar el modo fijo RTK cuando se puedan estimar 5 o más ambigüedades. Para los receptores de constelación única, esto significa que deben estar visibles al menos 6 satélites con máscara de elevación de 10 (por debajo de este ángulo sobre el horizonte no se recibe señales de satélites). Con el seguimiento de las constelaciones GPS y GLONASS, se precisa de un mínimo de 7 satélites para resolver ambigüedades y el seguimiento de la constelaciones simultaneas GPS+BeiDou requerirá 8 satélites. El receptor móvil volverá al modo RTK flotante (sin suficiente precisión) si pierde la cantidad mínima de señales necesarias para mantener el modo fijo RTK. El receptor móvil intentará volver al modo fijo RTK una vez que se haya restablecido el número mínimo de señales.

Si las correcciones diferenciales RTCM3.2 procedentes de la estación base RTCM 3 dejan de estar disponibles, el receptor móvil trabaja como un receptor de precisión estándar independiente.

El modo de operación, flotante o fijo, se indica en los mediante mensajes NMEA y UBX-NAV.

El reloj del receptor es impulsado por un oscilador de 32 kHz del tipo TCXO. Este reloj sigue funcionando a pesar de una pérdida de alimentación, lo que permite crear una copia de seguridad para que se guarden en la RAM de datos relevantes que permiten un reinicio de la actividad más rápido.

4.- **Bluetooth HC-06**. Para la transmisión de las correcciones diferenciales procedentes de la red GPS del ItaCyL recogidas vía internet en la tablet es necesario acoplar al receptor u-Blox un dispositivo Bluetooth. Dicho dispositivo es un Bluetooth HC-06 v2.0+EDR (Enhanced Data Rate), con las siguientes características:

- Alcance entre 5 y 10 metros.
- Solo puede operar en modo esclavo.
- Chip de radio CSR BC417143.
- Frecuencia: 2,4 GHz, banda ISM.
- Modulación GFSK.
- Antena PCB incorporada.
- Potencia de emisión: <-80 dBm, clase 2.
- Velocidad: Asincrónica: 2 Mbps (max.)/160 kbps, sincrónica: 1 Mbps/1 Mbps.
- Seguridad: Autenticación y encriptación (contraseña por defecto: 1234).
- Módulo montado en tarjeta con regulador de voltaje y 4 pines suministrando acceso a VCC, GND, TXD, y RXD.
- Consumo de corriente: 30 mA a 40 mA
- Voltaje de operación: 3.6 V a 6 V

- Dimensiones totales: 1.7 cm x 4 cm aprox.
- Temperatura de operación: -25 ºC a +75 ºC.

Las principales aplicaciones de dispositivos de este tipo son la comunicación inalámbrica entre ordenadores, teléfonos móviles o tablets que dispongan de conexión Bluetooth y microcontroladores, en este caso el receptor GPS u-Blox, que como carece de este dispositivo es necesario acoplarle uno para la recepción de correcciones diferenciales.

La comunicación Bluetooth entre dos equipos debe realizarse entre un módulo configurado como maestro (en este caso la tablet) y otro como esclavo (en este caso el dispositivo HC-06).

El módulo HC-06 suele viene con velocidad de transmisión serial de 9.600 bps, 1 bit de parada, y sin bit de paridad, nombre linvor y contraseña 1234.

El módulo HC-06 tiene como ventajas su pequeño tamaño, buenas características de transmisión y recepción que le brindan un alcance muy amplio (por tratarse de un sistema local Bluetooth), bajo consumo de corriente que posee tanto en funcionamiento, como en modo de espera y que una vez enlazado con otro dispositivo es capaz de recordarlo en su memoria y no solicita validación alguna (contraseña “1234” por defecto).

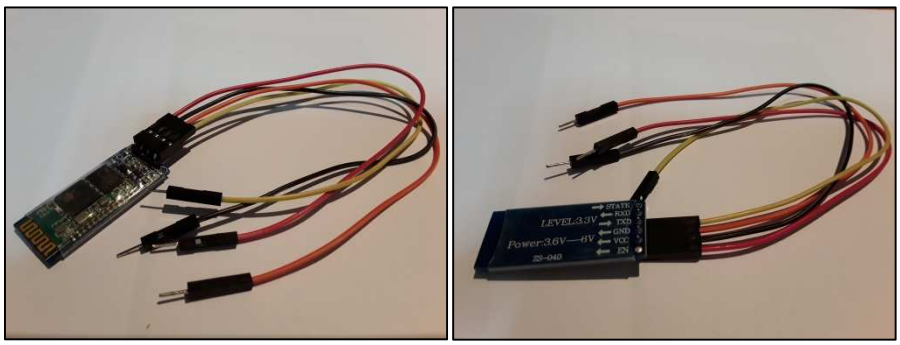


Imagen 14.- Bluetooth HC-06 con los cables de conexión al receptor u-Blox

5.- Aplicación Lefebure. La aplicación Lefebure Ntrip Cliente es una aplicación gratuita desarrollada para el sistema operativo Android. Desarrollada por la empresa estadounidense Lefebure Design especializada en soluciones GPS para el sector agrícola y descargable desde su página de internet http://www.Lefebure.com.

Permite conectar mediante Bluetooth un receptor GPS y el dispositivo en que esté instalada dicha aplicación, en este PFM la tablet Samsung. Mediante la conexión a internet de la tablet se reciben las correcciones diferenciales emitidas por el ItaCyL y son enviadas por la aplicación Lefebure a través de Bluetooth al receptor GPS u-Blox.

Para la configuración de la aplicación Lefebure es necesario seguir los siguientes pasos:

1.- Lo primero que se necesita es que el receptor GPS tenga clave habilitada en su Bluetooth y que este Bluetooth haya sido detectado por la tablet e introducida en esta la contraseña (1234) del Bluetooth una vez haya sido detectado.

2.- A continuación hay que arrancar la aplicación Lefebure Ntrip Cliente en la tablet. Pulsando sobre el icono  se accede al menú de configuración de la aplicación. En *Receiver Settings* configuramos la conexión Bluetooth y los ficheros de salida donde grabar los datos (si es nuestro deseo) NTRIP que envía la aplicación al receptor GPS y los datos de observación GPS en formato NMEA. En *NTRIP Settings* configuramos la conexión a la dirección IP desde la cual descargamos las correcciones diferenciales a enviar al receptor GPS vía Bluetooth. La configuración NTRIP empleada para desarrollar este PFM ha sido la siguiente:

- Network Protocol: NTRIP v1.0.
- Caster IP: 195.76.182.228.
- Caster Port: 2101.
- Usuario y contraseña.
- Data Stream: punto de montaje de la red GNSS del ItaCyL más cercano a la zona geográfica en la que se va a probar la aplicación. Se selecciona RIA13, estación GNSS ubicada en la localidad de Riaza, provincia de Segovia a 21 kilómetros de la zona donde se pondrá a prueba esta aplicación.
- Longitud y latitud aproximadas de la zona de trabajo para la prueba de campo con maquinaria agrícola: en este caso latitud 41º norte y longitud 3,6º oeste

Una vez configurada la aplicación, en la pantalla principal se pulsa el botón *Connect*, la aplicación intenta conectar con el dispositivo Bluetooth y cuando lo consigue muestra en pantalla una barra de color amarillo que muestra la evolución de la transmisión de datos. Durante la transmisión de datos se muestran en pantalla distintos mensajes que informan sobre: tipo de posición que está obteniendo el receptor GPS (fija y precisa o flotante y menos precisa); satélites usados; fallos de conexión; etc.

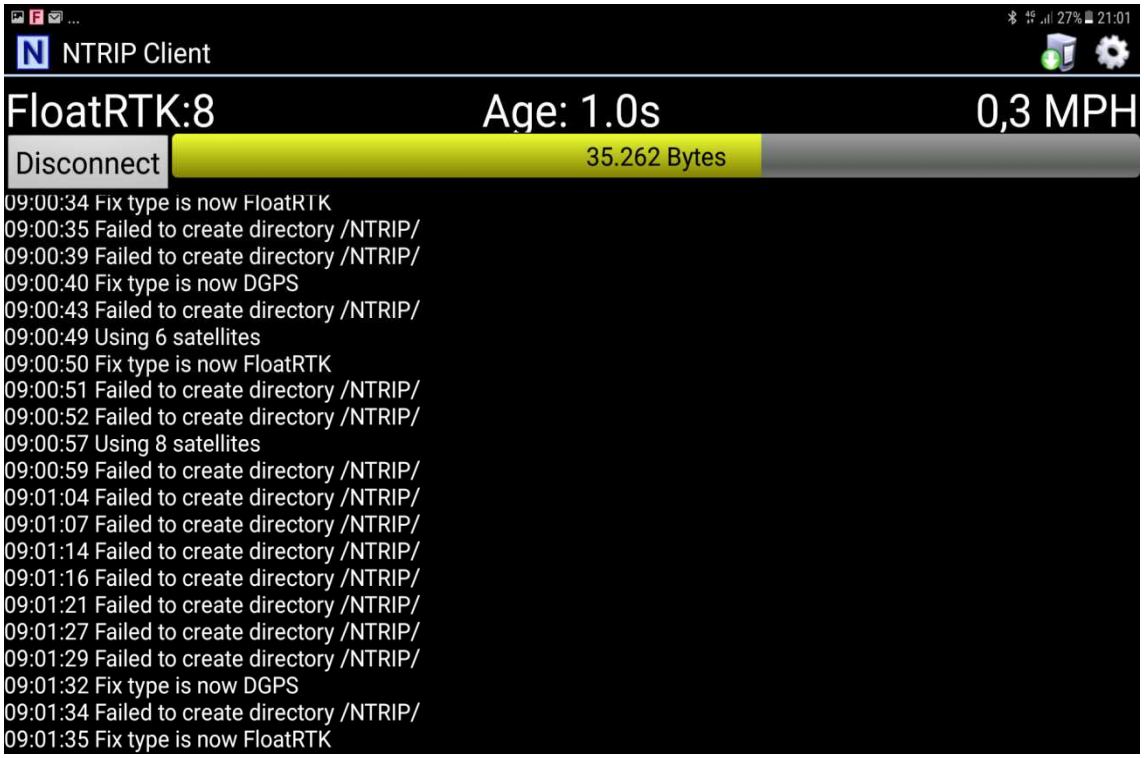


Imagen 15.- Aplicación Lefebure enviando correcciones diferenciales vía Bluetooth al receptor GPS

6.- Entorno de desarrollo Android Studio. Para el desarrollo de la aplicación Agrisoft, base de este PFM, se ha utilizado el entorno de desarrollo de Android Studio 3.1.3, de fecha 4 de junio de 2.018, instalado en el ordenador portátil Acer arriba reseñado. La primera toma de contacto con este entorno de desarrollo data del mes de enero del presente año y desde entonces realice varias instalaciones del mismo motivadas por problemas de compatibilidades o simplemente errores que obtenía en las aplicaciones de prueba y que por prevención me llevaban a reinstalar Android Studio.

No ha sido fácil trabajar en Android Studio sin una base en programación en lenguaje Java. Ha sido necesaria una gran dosis de paciencia y partir de cero alguna que otra vez para conseguir desde unos conocimientos nulos en Java llegar a la meta final que es la aplicación Agrisoft presentada en este PFM.

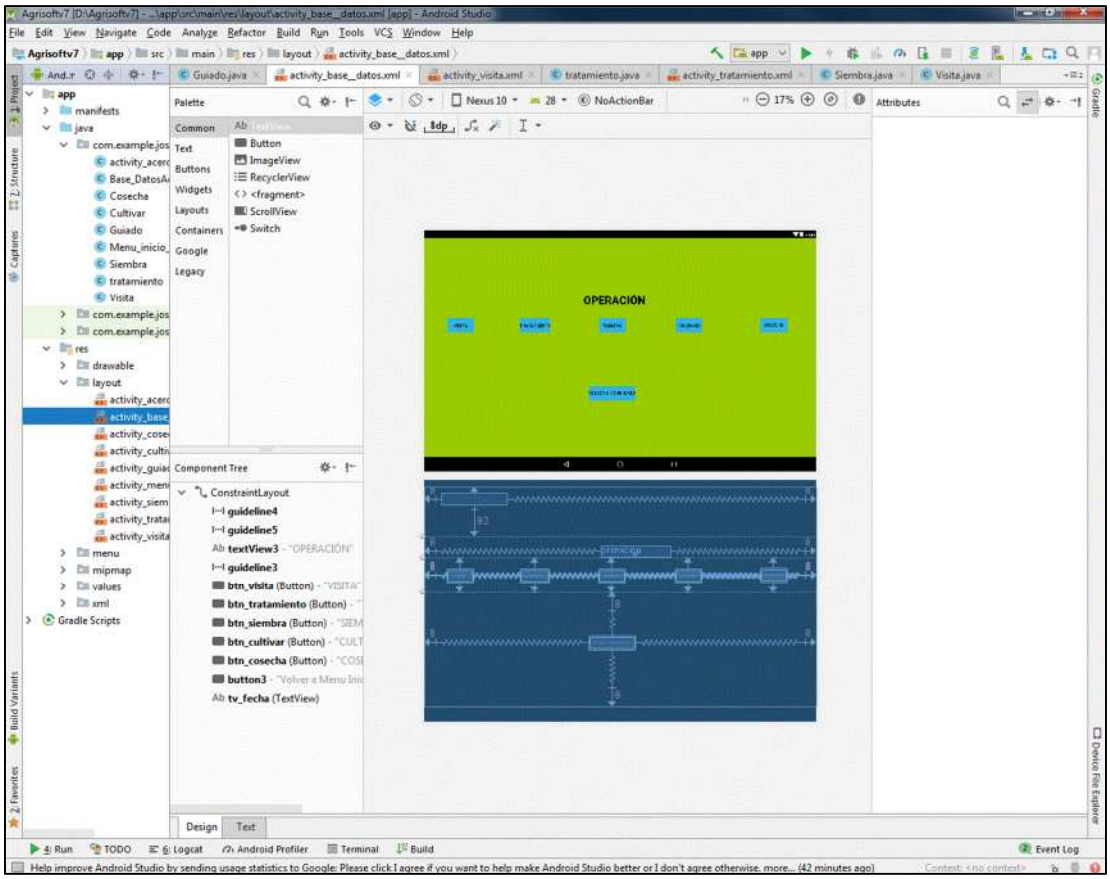


Imagen 16.- Android Studio con una versión de prueba de Agrisoft

Android Studio, a mi juicio y con el equipo informático donde ha sido instalado (el ordenador portátil arriba señalado), se caracteriza por ser lento y genera ficheros de aplicaciones que no solo ocupan varios megas de capacidad sino que contienen miles de archivos (en el caso de la aplicación Agrisoftv1, 36,5 MB y 1.318 archivos).

Una vez arrancado Android Studio, el entorno de trabajo se asemeja a cualquier otro programa desarrollado para Windows. En su parte superior se encuentra el menú principal del mismo y a continuación enunciaré los submenús más empleados:

- File: desde este submenú podemos abrir nuevos proyectos; abrir los ya creados; abrir proyecto recientes; imprimir; importar y exportar librerías; salir; etc.

- Edit: desde este submenú accedemos como en cualquier programa en Windows a funciones de cortar, pegar, borrar, buscar, etc.

- Build: desde este submenú las opciones más empleadas son las que permiten limpiar (*Clean Project*) y reconstruir el proyecto de trabajo (*Rebuild Project*), sobre todo necesario cuando surgen errores en el proyecto.

- Run: desde este submenú accedemos a las opciones que nos permiten ejecutar la aplicación que estamos desarrollando (*Run*), ejecutar y depurar paso a paso la aplicación (*Debugg app*), editar configuraciones (*Edit Configurations*) y establecer puntos de ruptura en la ejecución de la aplicación para a partir de ellos ejecutarla paso a paso pudiendo tener un control en tiempo real del valor de las variables (*Toggle Line BreakPoint*).

- Tools. Desde este submenú podemos acceder a:

  - *SDK Manager:* desde el submenú de esta opción podemos instalar plataformas de Android para las cuales queremos desarrollar la aplicación (*SDK Platforms*) que en el caso de este PFM han sido todas, a partir de la versión 7.0 de Android (API 14) y herramientas que serán necesarias para ejecutarla, como emuladores, servicios de Google, conexiones USB (dentro de *SDK Tools*).

  - *AVD Manager*: desde este submenú podemos crear dispositivos virtuales desde los cuales ejecutar nuestras aplicaciones sin necesidad de tener un dispositivo físico. Desde el inicio he optado por probar las distintas versiones de prueba de Agrisoft en un dispositivo real, la tablet arriba indicada, para tener un control más real de la evolución de la misma. Dicha evaluación de las aplicaciones requieren conexión USB entre el ordenador y la tablet para lo cual fue necesario instalar en el ordenador los drivers correspondientes a la tablet Samsung. La ejecución de las aplicaciones con los dispositivos virtuales se caracteriza por ser lentas inicialmente y requerir de instalaciones de componentes específicos.

En la parte izquierda de Android Studio se sitúa el árbol de trabajo de la aplicación. Desde este árbol de trabajo podemos acceder a distintos archivos y carpetas. Los más importantes empleados en la aplicación son:

- **Carpeta Manifest**. Desde ella accedemos al fichero AndroidManifest.xml donde se solicitan los permisos necesarios para ejecutar la aplicación así como las actividades (Activity) que contiene. Los permisos solicitados para la realización de la aplicación Agrisoft han sido los siguientes:

  - android.permission.WRITE_EXTERNAL_STORAGE: para poder escribir en ficheros de datos alojados en una unidad de almacenamiento externa al disco de la tablet.

  - android.permission.ACCESS_NETWORK_STATE: para obtener información de todas las redes, por ejemplo para saber si la aplicación tiene acceso a internet.

  - android.permission.INTERNET: para tener acceso a internet desde la aplicación.

  - android.permission.ACCESS_COARSE_LOCATION: para tener acceso al sistema de localización basado en redes de telefonía y Wi-Fi.

  - android.permission.ACCESS_FINE_LOCATION: para acceder al cualquier tipo de sistema de localización.

  - android.permission.ACCESS_WIFI_STATE: para acceder al servicio wifi desde la aplicación.

- **Carpeta Java**, subcarpeta com.example.josemanuel.agrisoftv1: esta subcarpeta contiene los ficheros con extensión .java que contienen el código fundamental de la aplicación. Los ficheros .java creados para la aplicación Agrisoft son las siguientes:

  - activity_acercade: en esta clase Java se encuentra el código de que nos permite acceder a información relativa a la aplicación.

  - Base_DatosActivity: clase Java que contiene el código del menú que nos permite seleccionar el tipo de operación de trabajo que se va a realizar en campo. Nos permite acceder a otras activity. Es accesible desde la activity Menu_inicio_Activity.

  - Cosecha: clase Java accesible desde la activity Base_DatosActivity. Contiene el código para generar el fichero de texto Cosecha.xml que nos permite guardar cierta información sobre esta operación agrícola.

  - Cultivar: clase Java a la que se accede desde Base_DatosActivity. Contiene el código para generar el fichero Cultivar.xml donde guardaremos datos sobre esta operación agrícola.

  - Guiado: clase Java accesible desde Menu_inicio_Activity. Contiene el código que nos permite el guiado mediante técnicas GPS de la máquina agrícola donde esté instalado el receptor.

  - Menu_inicio_Activity: contiene el código que nos permite ir a Base_DatosActivity, Guiado, activity_acercade y un botón salir para abandonar la aplicación.

  - Siembra: clase Java accesible desde Base_DatosActivity. Contiene el código que nos permite crear el archivo Siembra.xml donde se guardará cierta información introducida por teclado relativa a esta operación agrícola.

  - Tratamiento: clase Java accesible desde Base_DatosActivity. Contiene el código que nos permite crear el fichero Tratamiento.xml en el cual mediante entradas de teclado guardaremos información relativa a esta operación agrícola.

  - Visita: clase Java accesible desde Base_DatosActivity. Contiene el código por el cual se crea el archivo Visita.xml en el cual se guarda información introducida mediante entradas de teclado.

Para crear una clase Java es necesario hacer doble clic sobre la carpeta java o sobre una clase ya creada y a continuación seleccionar Java Class.

- **Carpeta res**: esta carpeta se compone de las siguientes subcarpetas.

  - **drawable:** contiene las imágenes (en formato png) que se mostrarán cuando la aplicación sea instalada en un dispositivo y que serán el icono que la represente. A su vez también contiene la imagen (en formato .bmp) que es el fondo de pantalla de la aplicación y que se muestra al iniciarse esta. Como muestro más adelante ambas imágenes se han personalizado. En cuanto a la imagen que se muestra como fondo de pantalla en la imagen

inicial de la aplicación, por limitaciones de capacidad tuve que recurrir a reducir la calidad de la imagen mostrada por debajo de un mega y seleccionar un fotograma en blanco y negro pues imágenes de mayor calidad me daban error al depurar la aplicación.

- **layout:** contiene los archivos en formato .xml en los cuales está diseñado el interfaz o pantalla de comunicación con el usuario de la aplicación. Al hacer doble clic sobre alguno de estos archivos nos aparece una pantalla para iniciar el diseño. Se pueden generar mediante código (si seleccionamos la pestaña Text que aparece en la parte inferior) o de manera gráfica (si seleccionamos la pestaña Design que aparece en la parte inferior). En mi caso los he realizado de manera gráfica por la sencillez que representan. Se selecciona sobre una paleta el tipo de elemento que queremos añadir (botón, edittext, texview, etc.) y una vez incorporado a la pantalla virtual que representa la aplicación se le da formato (color, letra, tamaño, etc). A la derecha de la pantalla aparece información editable sobre la configuración del elemento seleccionado sobre la pantalla virtual de la aplicación (en la imagen 17, sobre el fondo de pantalla de la aplicación).

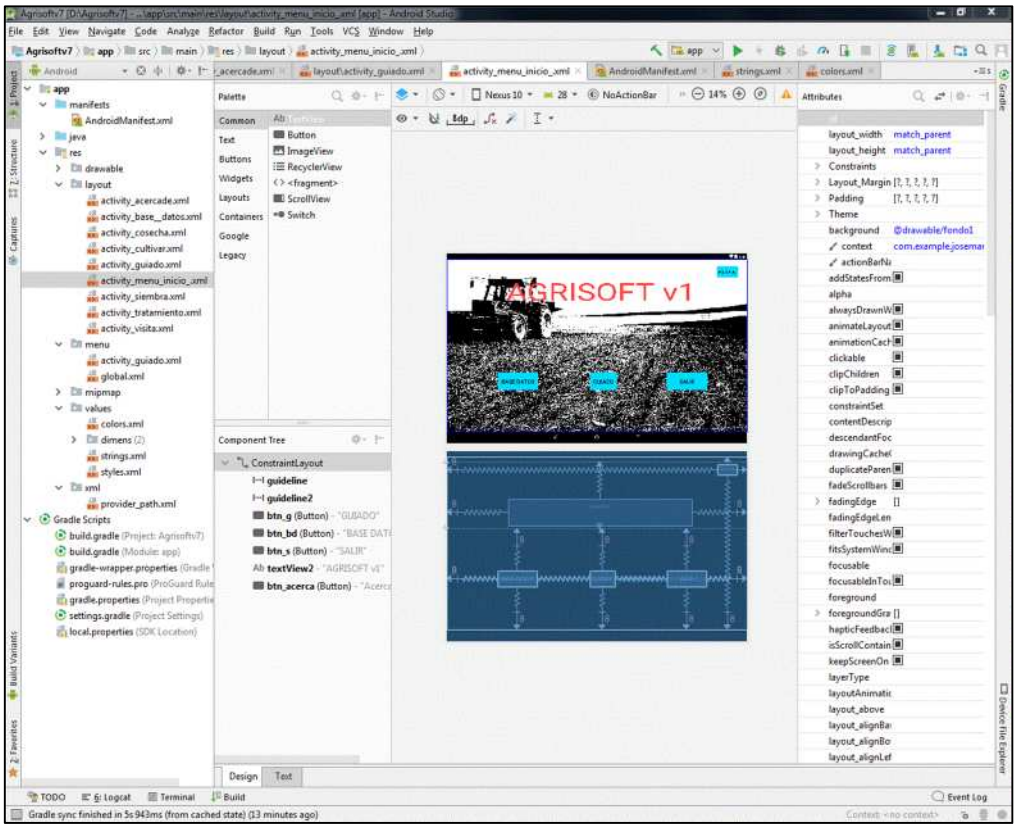


Imagen 17.- Imagen del layout Menu_Inicio_Activity

Por cada clase Java creada en la carpeta java automáticamente se crea un activity (pantalla de comunicación con el usuario de la aplicación). Por lo tanto en consonancia con las clases Java enunciadas en la carpeta java se han generado los siguientes activity: activity_acercade.xml; activity_base_datos.xml; activity-cosecha.xml; activity_cultivar.xml; activity_guiado.xml; activity_menu_inicio.xml; activity_siembra.xml, activity_tratamiento.xml y activity_visita.xml.

Una clase Java contiene el código que nos permite ejecutar las operaciones que queremos hacer con la aplicación mediante los elementos de comunicación (botones, entradas de texto, mensajes, datos numéricos, menús, etc.) entre la aplicación y el usuario de la misma a través de los layout.

Un elemento muy importante en el diseño de aplicaciones en Android Studio son los layouts. Permiten combinar varios elementos o vistas (botones, textos, etc) dentro de una vista y controlar su comportamiento y posición. De los diversos layout que ofrece Android Studio siempre he elegido ConstraintLayout. Este layout permite posicionar en la pantalla los elementos de diseño de tal manera que la posición de estos se mantendrá invariable independientemente del tamaño de la pantalla donde se ejecute la aplicación y sus tamaños se adaptarán a esta.

- **mipmap:** en la subcarpeta ic_launcher, ubicada dentro de mipmap se ubica el icono de la aplicación. Este icono será guardado en esta carpeta con cuatro versiones diferentes para adaptarse a distintas pantallas o calidades de pantallas donde se instale la aplicación.

- **values:** en esta subcarpeta se encuentran archivos .xml que indican valores empleados en la aplicación. Por ejemplo en el archivo strings.xml se definen las cadenas de caracteres que aparecen en la aplicación.

- **Carpeta Gradle Scripts:** de los archivos que contiene esta carpeta el más importante es el fichero **build.gradle** que es donde se configuran las opciones de compilación de la aplicación:

  - El parámetro compileSdkVersion indica la versión del sdk con la que se compila la aplicación, en este caso la 26.

  - El parámetro minsdkVersion especifica el nivel mínimo de API que requiere la aplicación para ser ejecutada. En el caso de la aplicación Agrisoft, puede ejecutarse en dispositivos que cuenten como mínimo con la versión 14.

  - El parámetro targetSdkVersion indica la versión más alta con la que se ha puesto a prueba la aplicación, en el caso de Agrisoft, la 26. Al salir nuevas versiones del SDK hay que comprobar la aplicación con estas nuevas versiones y actualizar el valor.

  - El parámetro versionCode y versionName indican la versión de la aplicación Agrisoft desarrollada. Cuando el diseño y desarrollo de la aplicación evolucione hay que incrementar ambos valores, versionCode a 2 cuando sea una evolución importante y versionName a 1.1 si es un cambio menor.

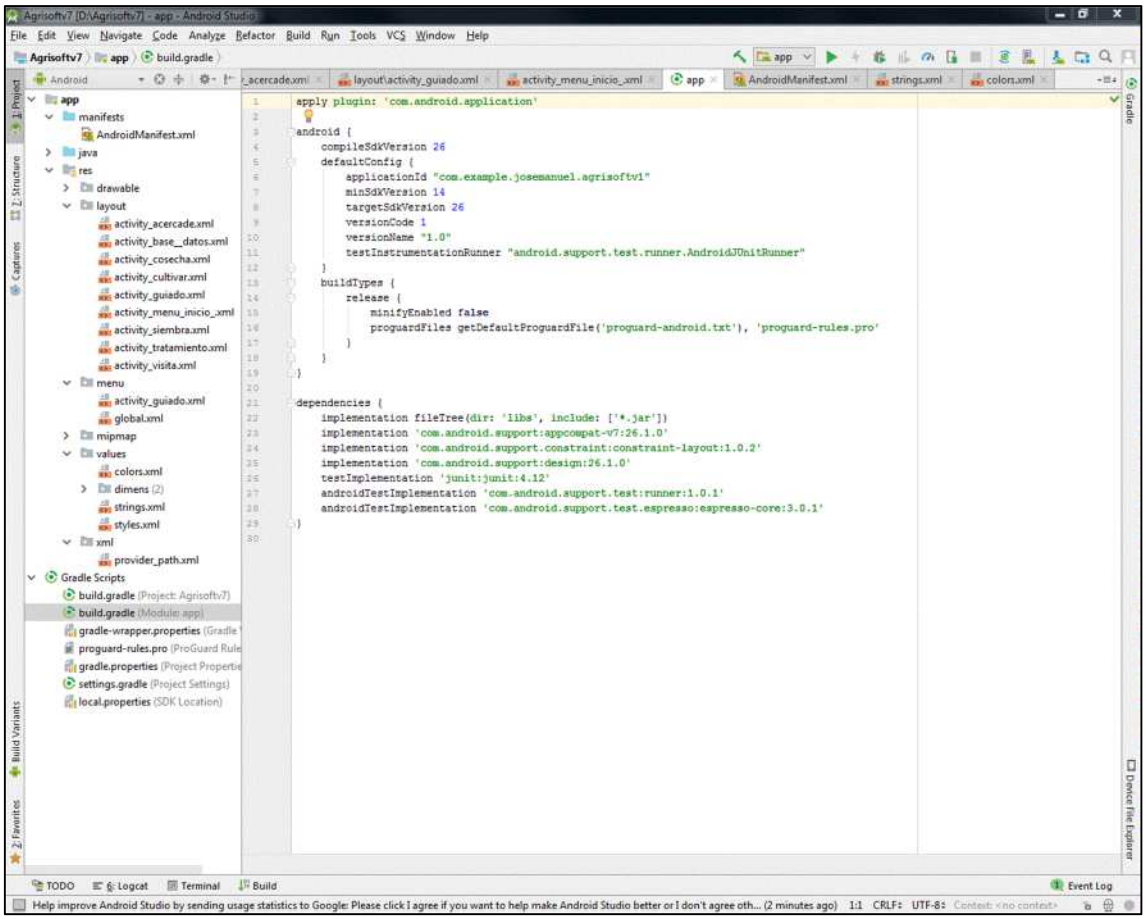


Imagen 18.- Imagen del archivo build.gradle

Como he comentado anteriormente la ejecución en modo prueba de la aplicación se puede realizar de dos maneras. Desde el menú Run con Run “app” (botón ) y con Debugg “app” (botón ).

En ambos casos antes de comenzar la ejecución Android Studio nos pregunta en que dispositivo queremos ejecutarla: un dispositivo real conectado por USB o un dispositivo virtual creado dentro de Android Studio. Siempre he probado la aplicación con un dispositivo real (la tablet arriba comentada): por conocer gráficamente como se representaría la aplicación en una pantalla real; por ser más lenta la ejecución en un dispositivo virtual; y por necesitar de un receptor de señal GPS real que me registrase señal de satélites.

La ejecución con Run “app” en primer lugar verifica si la aplicación tiene algún error y posteriormente la instala en el dispositivo externo mediante la conexión USB. Cualquier error que tenga la aplicación que sea detectado antes de ejecutarla o en el momento de ejecución provoca la salida por pantalla de una descripción del mismo que ayuda en su resolución. La principal fuente de ayuda para la resolución de errores ha sido internet mediante los múltiples foros de programación en Java o Android Studio que existen.

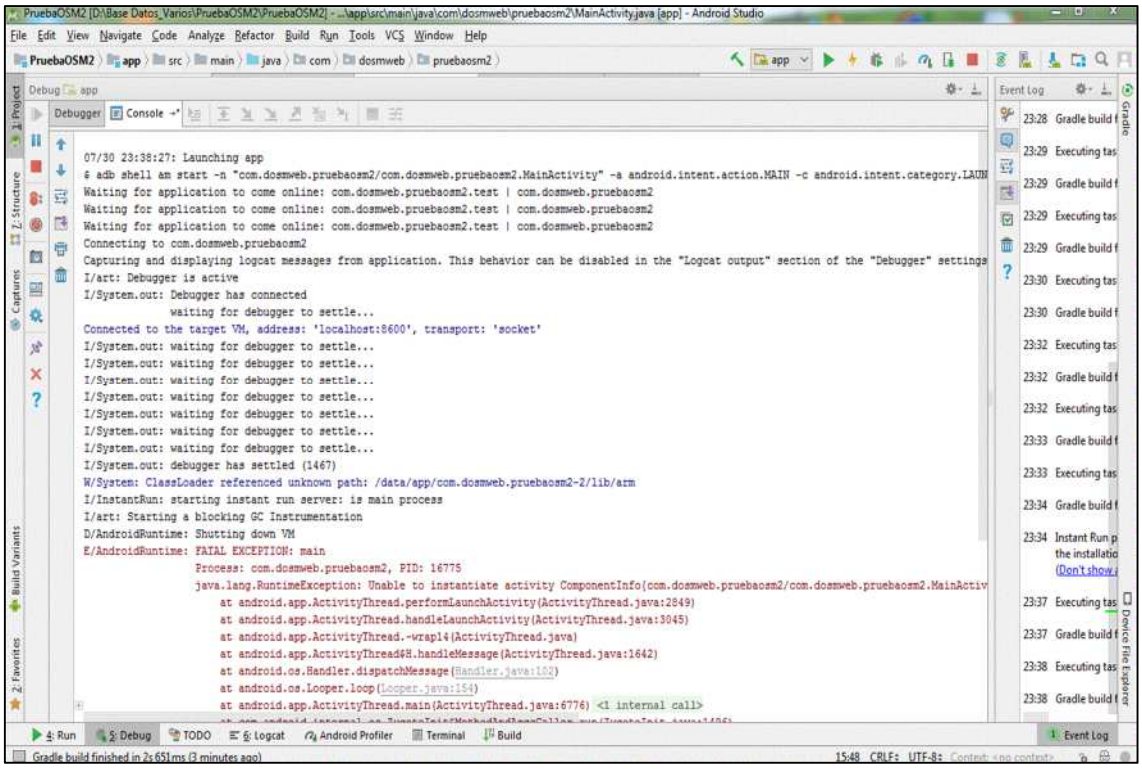


Imagen 19.- Salida por pantalla de un error en la ejecución de la aplicación

La ejecución de la aplicación con Debugg “app” necesita crear un punto de ruptura en la misma. La ejecución de la aplicación se realizará con Run “app” hasta ese punto de ruptura a partir del cual la aplicación correrá paso a paso siempre y cuando el usuario pulse la tecla F7 para que la aplicación siga avanzando. Durante ese proceso de avance la aplicación recorre cada una de las líneas de código y muestra en pantalla el valor de las variables que va generando y por lo tanto se realiza un seguimiento en tiempo real y al detalle de la misma, fundamental para conocer el motivo por el cual las variables pueden adoptar valores erróneos o el punto de la aplicación donde esta queda bloqueada.

Para crear un punto de ruptura existen dos métodos:

1.- Pulsar con el botón izquierdo del ratón sobre la escala numérica que aparece a la izquierda del código y cuyos números representan las líneas del código.

2.- Desde el menú Run, seleccionar View Breakpoints….

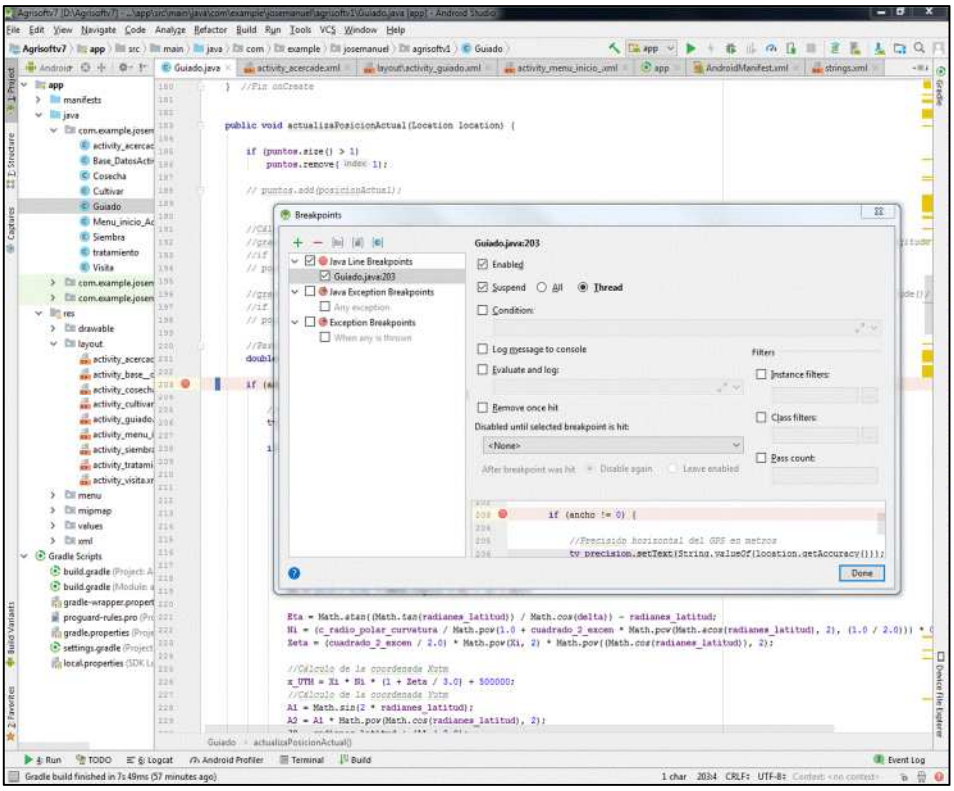


Imagen 20.- Creación de un punto de ruptura en la línea de código 203 dentro de Guiado.java

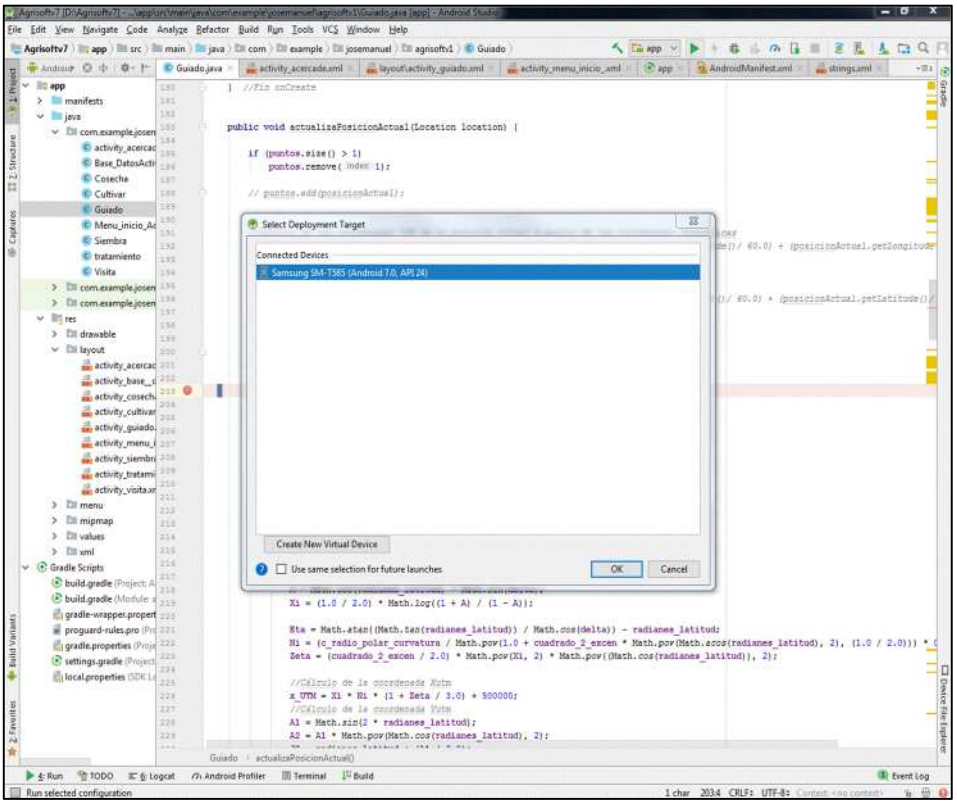


Imagen 21.- Selección del dispositivo para ejecutar la aplicación, real o virtual

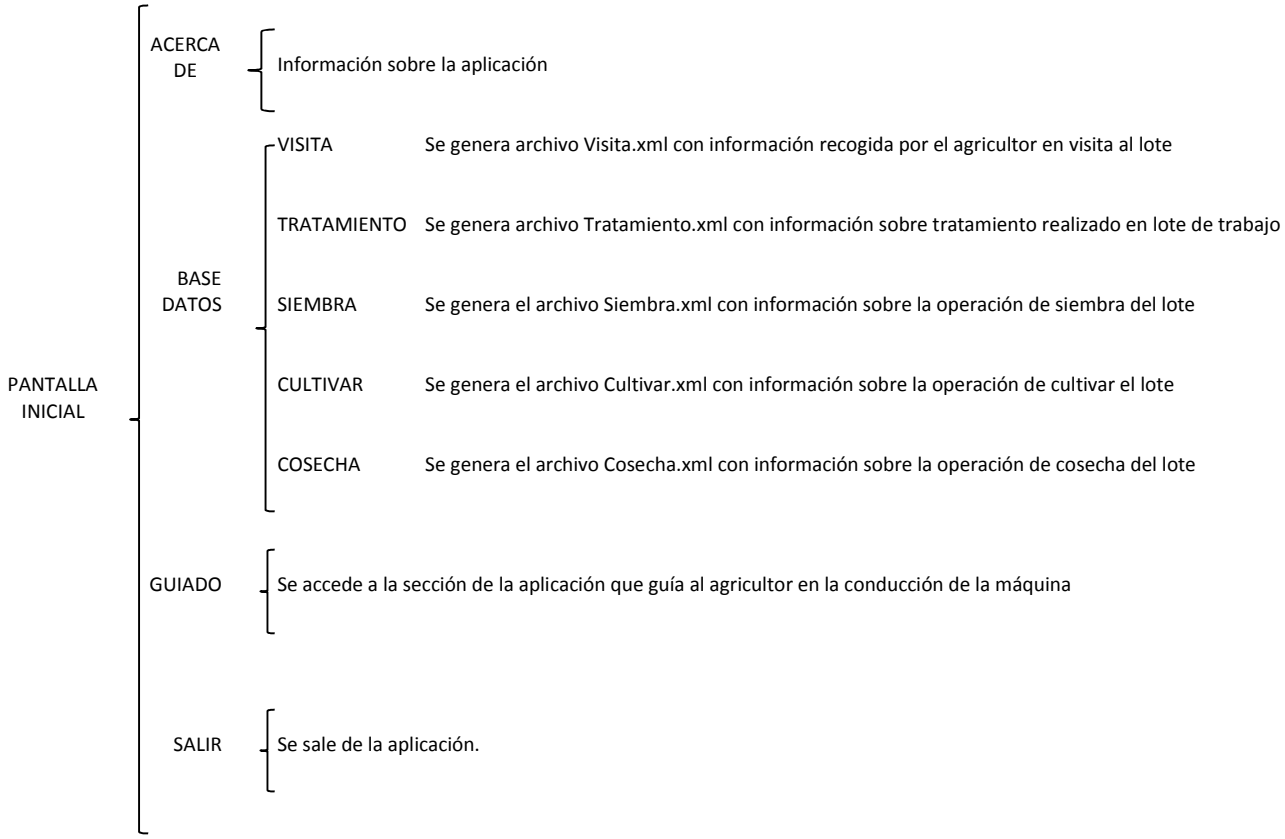
5.2. Aplicación Agrisoft

La aplicación Agrisoft desarrollada en este PFM en su versión 1, constituye una ayuda al agricultor en dos vertientes:

1.- Sirve como base de datos, en archivo con formato xml, para el registro de todos los trabajos que realice en campo.

2.- Sirve como apoyo para aquellos trabajos que por sus características necesitan un asistente en la conducción, puesto que le guía en el manejo de la máquina para que el trabajo que realiza no tenga zonas sin trabajar o solapes en zonas ya trabajadas.

A continuación se representa un esquema de la aplicación Agrisoft:



Esquema 1.- Esquema aplicación Agrisoft

La aplicación Agrisoftv1 se instala automáticamente en el dispositivo Android, en este caso una tablet, con solo conectarlo al ordenador que tiene abierto el proyecto de la aplicación a través de cable USB y ejecutar la aplicación desde Android Studio en un dispositivo real. Durante el proceso de ejecución e instalación de la aplicación en la tablet se genera un icono en el escritorio de esta que representa la aplicación. Dicho icono es configurable, de tal manera que si no se configura personalizado, Android Studio genera un icono por defecto con el logotipo de Android. Para la aplicación Agrisoft se ha generado un icono propio que es el presentado a continuación:

Haciendo doble click sobre el icono arriba indicado entramos en la aplicación.

La pantalla de inicio de la aplicación es la presentada a continuación, correspondiente con el layout activity_menu_inicio.xml y con la clase java Menu_inicio_Activity.java. En ella podemos observar cuatro botones con las siguientes funciones:

Imagen 22.- Pantalla de inicio de la aplicación

1.- Botón **BASE DATOS**: pulsando este botón accedemos al layout activity_base_datos.xml y a la clase java Base_DatosActivity.java y por lo tanto a la parte de la aplicación desarrollada para introducir por teclado las características del trabajo agrícola que va a desarrollar el agricultor y que quedarán registradas en un archivo en formato xml.

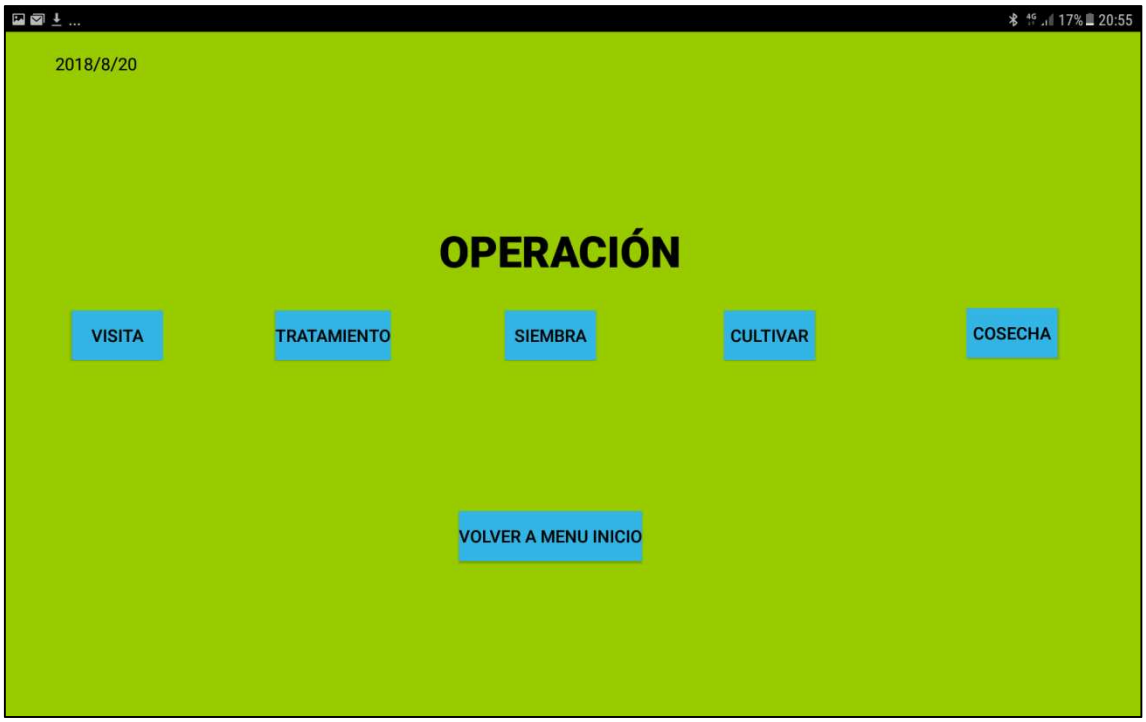


Imagen 23.- Pantalla del layout activity_base_datos, pulsando el botón BASE DATOS

2.- Botón **GUIADO**: pulsando este botón accedemos al layout activity_guiado.xml y a la clase java Guiado.java. Desde esta parte de la aplicación tenemos acceso al guiado de maquinaria agrícola en base a la señal del receptor GPS que recibe la aplicación y que convenientemente procesada servirá de referencia al agricultor en la conducción de su máquina. Su uso se explica al final de este capítulo.

3.- Botón **SALIR**: pulsando este botón abandonamos la aplicación.

4.- Botón **ACERCA DE..:** pulsando este botón se accede al layout activity_acercade.xml (imagen 24) y a la clase java activity_acercade.java. Este botón nos da acceso a información sobre la aplicación. Este layout contiene el botón VOLVER AL MENU INICIO que nos permite volver al layout activity_menu_inicio.xml.

Si pulsamos el botón BASE DATOS accedemos a la pantalla mostrada en la imagen 23 y que contiene los siguientes elementos:

1.- Fecha, en la parte superior izquierda.

2.- Un textView con el texto OPERACIÓN, que sirve de introducción a los botones de selección que aparecen debajo:

3.- Botones de acceso a distintas opciones de la aplicación:

3.1.- Botón **VISITA**: pulsando este botón accedemos al layout activity_visita.xml y a la clase java Visita.java. Dentro de este layout, como se muestra en la imagen 25, tenemos dos textviews, campos o entradas de teclado: lote y observaciones.

El campo lote hace referencia a la parcela que está siendo visitada o inspeccionada por el agricultor y el campo observaciones está reservado a los comentarios que quiera registrar: como puede ser el estado del cultivo, enfermedades registradas, daños de fauna, etc.

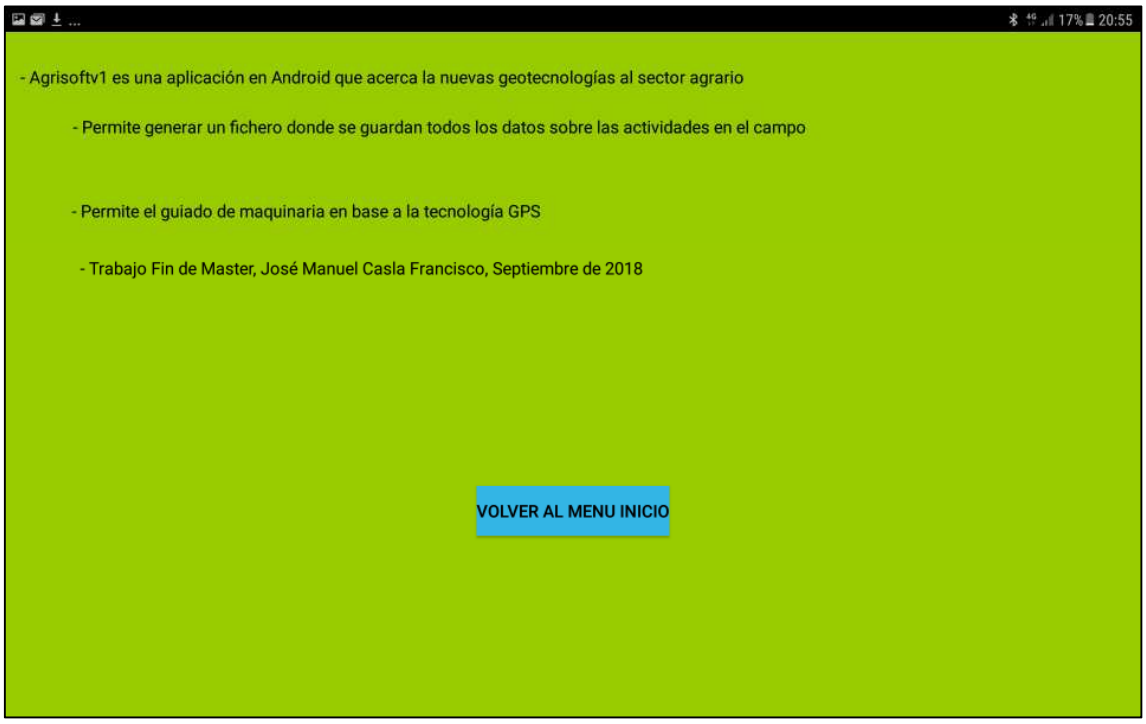


Imagen 24.- Pantalla del layout activity_acercade, pulsando botón ACERCA DE

Como norma general para todos los campos o textviews a rellenar, al pulsar sobre el espacio reservado al texto se activa el teclado, numérico si el campo es a rellenar con números o teclado tipo texto si es a rellenar con letras, dependiendo de cómo haya sido configurado el textview en el código del layout.

Con el botón **GUARDAR**, se guardan las entradas de teclado realizadas en un fichero llamado Visita.xml (nombre por defecto asignado en el código de la aplicación) cuya estructura es la siguiente:

<?xml version="1.0"?>

-<VISITA>

<FECHA>fecha de la visita, dada por la aplicación</FECHA>

<LOTE>lote o parcela visitada</LOTE>

<COMENTARIO>observaciones anotadas</COMENTARIO>

</VISITA>

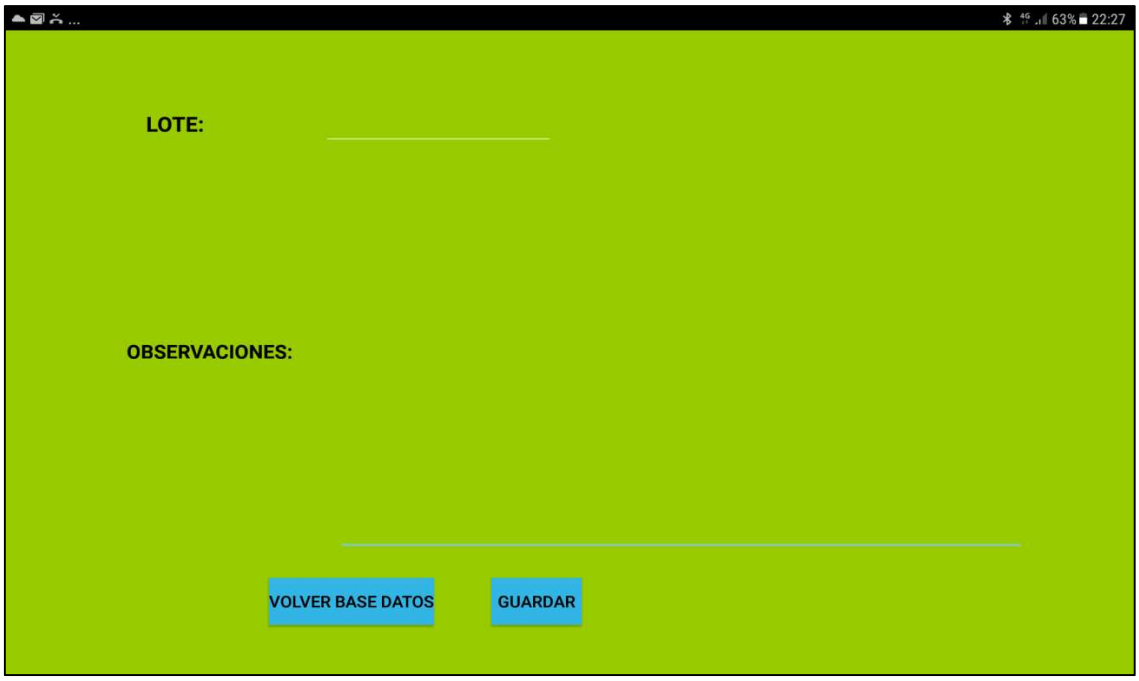


Imagen 25.- Pantalla de la operación visita

Si se pulsa el botón **VOLVER BASE DATOS** se retrocede al layout activity_base_datos.xml.

3.2.- Botón **TRATAMIENTO**: pulsando este botón se accede al layout activity_tratamiento.xml y a la clase java tratamiento.java.

La operación tratamiento se reserva para operaciones de abonado, tratamiento fungicidas y tratamientos con herbicidas. Como se aprecia en la imagen 26 se compone de varios textviews a rellenar por entradas de teclado y da la opción de poder realizar hasta 4 tratamientos a la vez pudiendo especificar las dosis de cada uno de ellos. Se puede indicar la duración de la operación y un campo de observaciones donde anotar comentarios. La situación ideal para rellenar estos campos es a la finalización de la operación, cuando ya se conoce el tiempo que ha sido necesario para ejecutarla.

Con el botón **VOLVER BASE DATOS** se vuelve al layout o pantalla activity_base_datos.xml y con el botón **GUARDAR** se genera el fichero Tratamiento.xml (nombre por defecto asignado en el código de la aplicación) cuya estructura es la siguiente:

<?xml version="1.0"?>

-<TRATAMIENTO>

<FECHA>fecha de realización de la operación</FECHA>

<LOTE>lote o parcela sobre la que se hace el tratamiento</LOTE>

<TRATAMIENTO1>tratamiento 1</TRATAMIENTO1>

<DOSIS1>dosis del tratamiento 1</DOSIS1>

<TRATAMIENTO2>tratamiento 2</TRATAMIENTO2>

<DOSIS2>dosis del tratamiento 2</DOSIS2>

<TRATAMIENTO3>tratamiento 3</TRATAMIENTO3>

<DOSIS3>dosis del tratamiento 3</DOSIS3>

<TRATAMIENTO4>tratamiento 4</TRATAMIENTO4>

<DOSIS4>dosis del tratamiento 4</DOSIS4>

<DURACION>duración del tratamiento</DURACION>

<OBSERVACIONES>comentarios sobre el tratamiento realizado</OBSERVACIONES>

</TRATAMIENTO>

Imagen 26.- Pantalla de la operación tratamiento

3.3.- Botón **SIEMBRA**: pulsando este botón se accede al laypot activity_siembra.xml y a la clase java Siembra.java.

Imagen 27.- Pantalla de la operación siembra

Este layout se compone de varios textviews o campo a rellenar donde se puede especificar la semilla con la que se siembra (cultivo 1 y 2, por ejemplo trigo, cebada, girasol, etc.), dosis de siembra, duración de la operación de siembra y observaciones. Al igual que en la operación tratamiento lo ideal es acceder a este layout después de realizada la operación para así tener conocimiento del campo a rellenar duración.

Con el botón **VOLVER A BASE DE DATOS** volvemos al layout activity_base_datos.xml y con el botón **GUARDAR** se genera el fichero Siembra.xml donde se guardan las entradas de teclado realizadas y cuya estructura es la siguiente:

<?xml version="1.0"?>

-<SIEMBRA>

<FECHA>fecha en la que se realiza la operación de siembra</FECHA>

<LOTE>lote o parcela donde se realiza la siembra</LOTE>

<CULTIVO1>cultivo 1 que se siembra</CULTIVO1>

<DOSIS1>dosis del cultivo 1</DOSIS1>

<CULTIVO2>cultivo 2 que se siembra</CULTIVO2>

<DOSIS2>dosis del cultivo 2</DOSIS2>

<OBSERVACIONES>comentarios sobre la operación siembra</OBSERVACIONES>

</SIEMBRA>

3.4.- Botón **CULTIVAR**: pulsando este botón se accede al layout activity_cultivar.xml y a la clase java Cultivar.java.

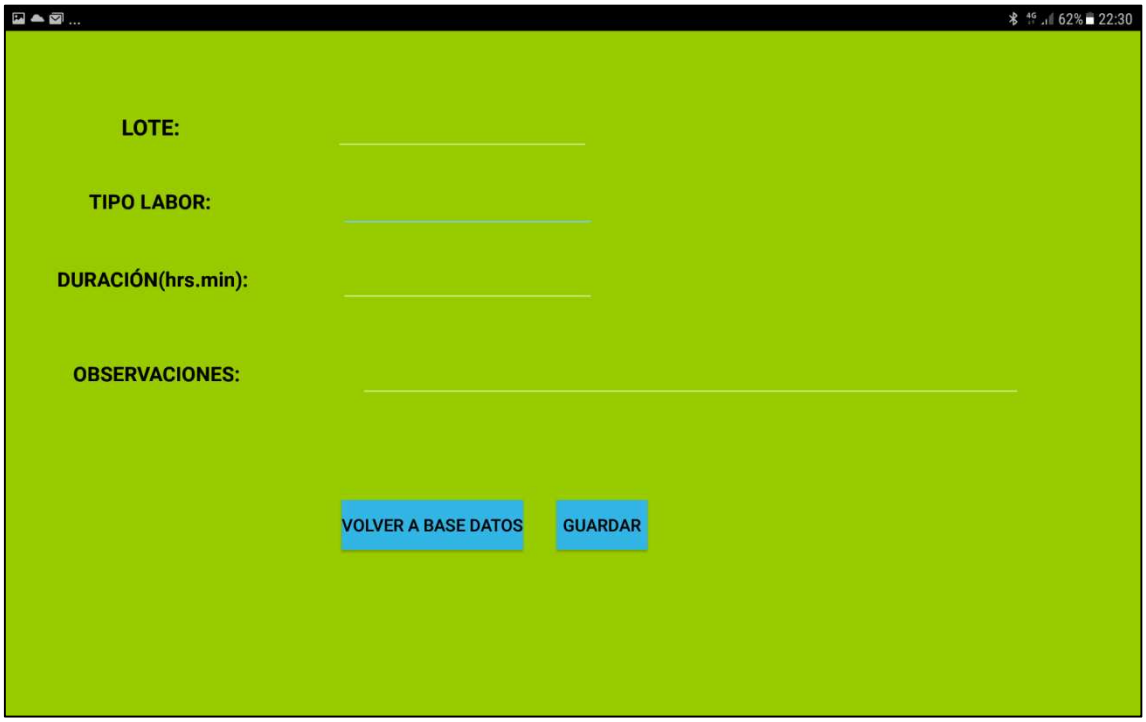


Imagen 28.- Pantalla de la operación cultivar

Este layout está conformado por varios textviews a rellenar mediante entradas de teclado donde se especifica la el lote o parcela que se cultiva, tipo de labor realizada (por ejemplo, alzado con vertedera, gradeo, etc.), duración de la misma y un campo para anotar observaciones.

Pulsando el botón **VOLVER A BASE DATOS** volvemos al layout activity_base_datos.xml y con el botón **GUARDAR** se genera el fichero por defecto Cultivar.xml donde se guardan las entradas de teclado anteriores y que tiene la siguiente estructura:

<?xml version="1.0"?>

-<CULTIVAR>

<FECHA>fecha en la que se realiza la operación de cultivar</FECHA>

<LOTE>lote o parcela que se cultiva</LOTE>

<LABOR>tipo de labor realizada</LABOR>

<DURACIÓN>duración empleada en la operación</DURACIÓN>

<OBSERVACIONES>comentarios sobre la operación cultivar</OBSERVACIONES>

</CULTIVAR>

3.5.- Botón **COSECHA**: pulsando este botón se accede al layout activity_cosecha.xml y a la clase java Cosecha.java.

Imagen 29.- Pantalla de la operación cosecha

El layout al que se accede pulsando el botón **COSECHA** está compuesto por una serie de textviews o campo a rellenar con entradas de teclado relacionadas todas ellas con la operación agrícola cosecha: lote o parcela cosechada; producto cosechado (trigo, cebada, vid, maíz, etc.); cantidad cosechada por unidad de superficie (kilogramos por hectárea); duración de la operación de cosecha y el campo observaciones.

Con el botón **VOLVER A BASE DATOS** se regresa al layout activity_base_datos.xml y con el botón guardar se genera el fichero por defecto Cosecha.xml en el cual se guardan las entradas de teclado anteriores y cuya estructura es la siguiente:

<?xml version="1.0"?>

-<COSECHA>

<FECHA>fecha de la operación cosecha</FECHA>

<LOTE>parcela o lote cosechado</LOTE>

<PRODUCTO>producto cosechado/PRODUCTO>

<CANTIDAD>cantidad por unidad de superficie cosechada</CANTIDAD>

<DURACION>tiempo empleado en la operación cosecha</DURACION>

<OBSERVACIONES>comentarios sobre la operación cosecha</OBSERVACIONES>

</COSECHA>

3.6.- Botón **VOLVER A MENU INICIO**: este botón permite volver al layout activity_menu_inicio.xml y a la clase java Menu_inicio_Activity.java, es decir a la pantalla de inicio de la aplicación.

Dada la importancia que tiene en este PFM la parte de la aplicación referida al guiado de maquinaria agrícola he dejado para la parte final de este capítulo su explicación.

Para acceder a ella, desde la pantalla inicial hay que pulsar el botón GUIADO y se accede al layout activity_guiado.xml.

El layout o pantalla inicial del guiado, sin empezar a trabajar, es la presentada en la imagen 30.

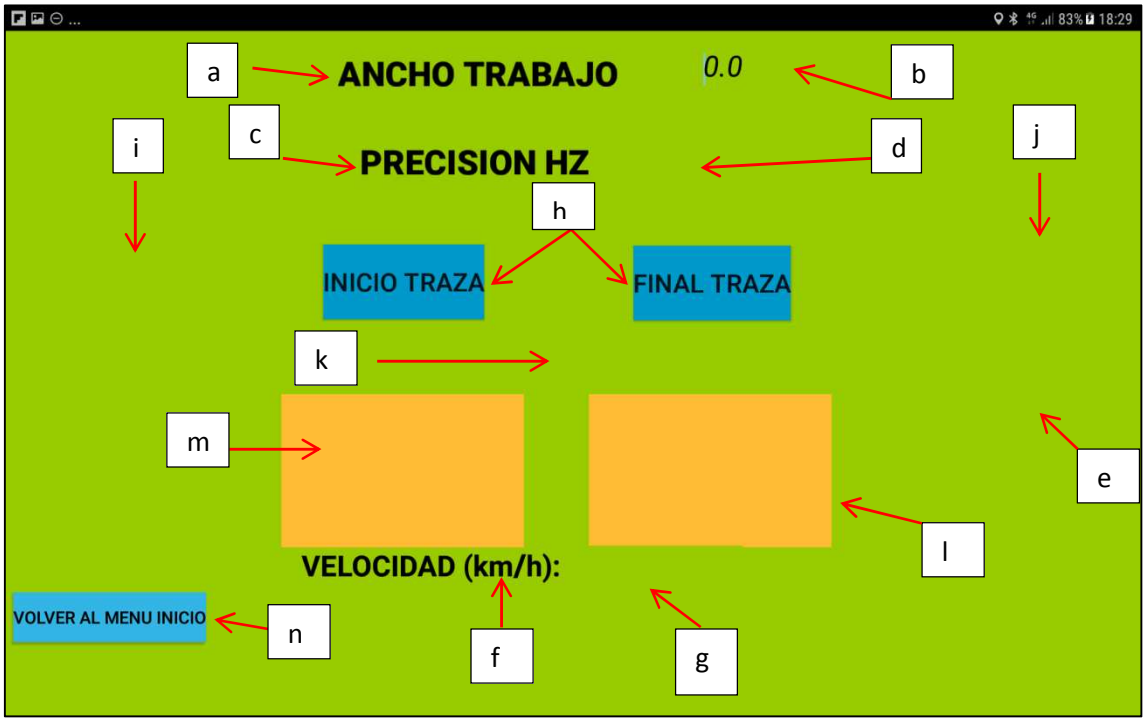


Imagen 30.- Pantalla para el guiado de maquinaria agrícola

Dicho layout se compone de un botón y varios edittext y textviews:

a) Textview **ANCHO TRABAJO**: texto que sirve de anuncio para introducir a continuación, a la derecha, el ancho de la operación agrícola a realizar.

b) Edittext que por defecto aparece 0.0.Mientras no se introduzca desde teclado un valor numérico mayor de 0, no se activan otras opciones. El ancho de trabajo hace referencia al ancho del implemento agrícola que lleve la máquina: el ancho de la barra de tratamiento de la máquina de pulverización; el ancho de trabajo de la abonadora con la que se aplican tratamientos de abonado químico; el ancho de trabajo de la graba con la que se cultiva la parcela; el ancho de trabajo de la sembradora con la que se siembra la parcela; el ancho del corte de la cosechadora con la que se recolecta un cereal; etc.

c) Textview **PRECISION**: texto que sirve de entrada al valor de la precisión de la observación GPS que se obtiene en tiempo real por medio del receptor u-Blox.

d) Edittext: este campo indica la precisión horizontal estimada en metros de la observación GPS que se obtiene en tiempo real por medio del receptor u-Blox. No he podido confirmar si este dato, mediante el método *getAccuracy(),* se extrae del protocolo NMEA y en concreto del parámetro adimensional HDOP. Por diseño de la aplicación, solo cuando este valor está por debajo de 10, se muestra su valor en pantalla y se da paso a otras opciones. Cuanto más bajo sea este parámetro más precisa en la posición obtenida. Según las pruebas realizadas en campo, el valor más bajo alcanzado ha sido de 0,02 pero en muy contadas ocasiones. El valor más habitual y a su vez más estable es de 0,50.

   Solo cuando se ha introducido un ancho de trabajo distinto de cero, la aplicación accede a valorar el valor de *getAccuracy()*.

e) Activado el edittext que contiene la precisión obtenida en el cálculo de la posición, que como he comentado se activa cuando está por debajo de 10, se activa también dos edittext, señalados en la imagen 19 con la letra *e*, que son las coordenadas X e Y en el Sistema de Coordenadas ETRS89, proyección UTM, que tiene la antena del receptor u-Blox. Se muestran las coordenadas en tiempo real y varían cada segundo siempre y cuando no haya cortes en la recepción de la señal de los satélites. El tiempo transcurrido entre dos registros de posición es configurable en Android, y se ha optado por el menor intervalo permitido, que es de un segundo. Las coordenadas X e Y proceden de las coordenadas latitud y longitud geográficas en el Sistema de Referencia WGS84 obtenidas del protocolo NMEA con los métodos de Android *getLatitude()* y *getLongitude().* Dichas coordenadas geográficas han sido transformadas en ETRS89, proyección UTM. La aplicación para este PFM se ha desarrollado dejando visibles estas coordenadas, con un propósito didáctico. Para un uso comercial se suprimiría su visualización, dado que no aporta información al agricultor.

f) Textview **VELOCIDAD (km/h**): texto que sirve de anuncio para introducir a continuación, a la derecha, la velocidad de la máquina en la que está instalada la aplicación y el receptor u-Blox.

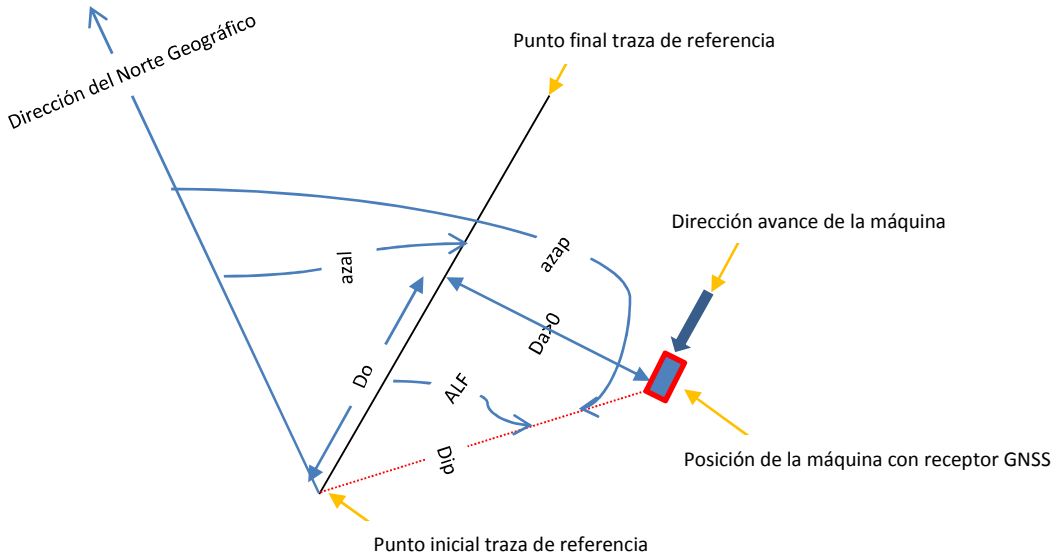
g) Edittext, situado a la derecha de **VELOCIDAD (Km/h**) que muestra el valor de la velocidad de la máquina. Velocidad procedente del protocolo NMEA, que se obtiene mediante el método de Android *getSpeed().* Inicialmente, en el código NMEA, la velocidad se expresa en metros por segundo, pero en pantalla se muestra en kilómetros por hora gracias al algoritmo de cálculo que incorpora la aplicación. Dicho edittext solo se activa cuando la precisión está por debajo de 10, es decir cuando el campo *e* muestra el valor de la precisión.

h) Botones **INICIO TRAZA** y **FINAL TRAZA**. Solo se activan estos botones cuando el parámetro DOP está por debajo de 10. Una vez que el valor de precisión ha alcanzado un valor suficientemente bajo, a valorar por el operador, podemos pulsar ambos botones, uno para seleccionar el punto inicial de la traza de referencia y otro para pulsar el punto final de la traza de referencia. Con el botón **INICIO TRAZA**, y haciendo uso de los métodos de Android *getLatitude()* y *getLongitude(),* se seleccionan las coordenadas geográficas el punto que el operador haya seleccionado como inicio de la traza de referencia. De la misma manera, con el botón **FINAL TRAZA**, y haciendo uso de los métodos de Android, *getLatitude()* y *getLongitude(),* se seleccionan las coordenadas geográficas del punto que el operador haya seleccionado como final de la traza de referencia.

i) Cuando el operador ha pulsado el botón **INICIO TRAZA**, el campo *i*, formado por dos edittext, muestra las coordenadas en la proyección UTM y Sistema de Referencia ETRS89 del punto elegido como inicio de la traza de referencia, a partir de la cual se trazarán paralelas a la distancia marcada por el valor del campo ancho. Al igual que el campo e arriba comentado, la aplicación para este PFM se ha desarrollado dejando visibles estas coordenadas, con un propósito didáctico. Para un uso comercial se suprimiría su visualización, dado que no aporta información al agricultor.

j) La explicación desarrollada para el campo *i* es totalmente válida para el campo *j*. La única diferencia es que con el botón **FINAL TRAZA**, se selecciona el punto final de la traza de referencia.

k) Seleccionados los puntos inicio y final de la traza, el campo *k* se activa y empieza a mostrar la mínima distancia, que en la aplicación se denota como *Da*, entre la posición actual de la antena del receptor u-Blox y la traza de referencia (distancia vertical a la traza de referencia), según se muestra en el croquis 1.



Croquis 1

El cálculo de *Da* se realiza mediante trigonometría de la siguiente manera:

1.- Como datos conocidos obtenidos del receptor GPS tenemos los siguientes:

- Coordenadas cartesianas UTM del punto inicial de la traza: x_inicio, y_inicio
- Coordenadas cartesianas UTM del punto final de la traza: x_final, y_final
- Coordenadas cartesianas UTM de la posición actual del receptor (o máquina agrícola): x_punto, y_punto

2.- Cálculo de los incrementos de coordenadas entre el punto inicial y final de la traza:

ix = x_final – x_inicio
iy = y_final – y_inicio

3.- Cálculo del azimut, parámetro *azal* en la aplicación, de la traza de referencia (ángulo que forma dicha traza con el norte geográfico). Para dicho cálculo es importante calcular en que cuadrante se encuentra la dirección de la cual queremos calcular su azimut. Los siguientes cálculos consideran ángulos en radianes, dado que Android para cálculos trigonométricos trabaja con radianes.

Si (ix > 0 && iy > 0)); si la traza está en el primer cuadrante
azal = atan(ix / iy
Si (ix > 0 && iy <= 0) ; si la traza está en el segundo cuadrante
azal = (PI / 2) + atan(iy / ix)
Si (ix < 0 && iy < 0) ; si la traza está en el tercer cuadrante
azal = PI + atan(ix / iy)
Si (ix < 0 && iy > 0) ; si la traza está en el cuarto cuadrante
azal = (PI * 1.5) + atan(iy / ix)

4.- Cálculo de datos entre el punto inicio de la traza y el punto respecto del cual queremos calcular *Da*, punto donde se encuentra la máquina:

iix = x_punto – x_inicio
iiy = y_punto- y_inicio

5.- Cálculo del azimut (parámetro *azap* de la aplicación) de la dirección formada por el inicio de la traza de referencia y el punto donde se encuentra la máquina:

if(ix > 0 && iy > 0)); si la dirección está en el primer cuadrante
azap = atan(iix / iiy);
if(ix > 0 && iy <= 0) ; si la dirección está en el segundo cuadrante
azap = (PI / 2) + atan(iiy / iix);
if(ix < 0 && iy < 0) ; si la dirección está en el tercer cuadrante
azap = PI + atan(iix / iiy);
if(ix < 0 && iy > 0) ; si la dirección está en el cuarto cuadrante
azap = (PI * 1.5) + atan(iiy / iix);

6.- Cálculo de la distancia (parámetro *Dip* en la aplicación) entre el punto inicial de la traza y el punto donde se encuentra la máquina:

$$Dip=\sqrt{(iix^2+iiy^2)}$$

7.- Cálculo de *Da*:

ALF = -azap + azal; diferencia de azimutes
Da = sin(ALF) * Dip
Do = -1*cos(ALF) * Dip; más adelante, en el apartado m se explicará el parámetro Do

El valor mostrado, es decir el valor de *Da*, será negativo cuando la máquina este a la izquierda de la traza de referencia y positivo cuando esté a la derecha de la traza de referencia.

En la aplicación desarrollada para este PFM se ha dejado visible este campo, con un propósito didáctico. Para un uso comercial se suprimiría su visualización, dado que no aporta información al agricultor.

l) El campo *l* es un textview que muestra el desplazamiento, en valor numérico, que el agricultor tiene que dar a la máquina para situarse, respecto a la traza de referencia, a una distancia igual al ancho de trabajo, es decir a una paralela a la traza de referencia, siendo la distancia de paralelismo el ancho de trabajo. La aplicación calcula la línea paralela más cercana en la que se encuentra la máquina y respecto a esta muestra el valor. En el caso del croquis 2 presentado a continuación, la línea paralela respecto de la cual la aplicación calcula el desplazamiento es la número dos y por lo tanto se encuentra a dos veces el ancho de trabajo respecto de la traza de referencia. En la aplicación el valor de desplazamiento a aplicar a la máquina se denota como *despla* y se muestra en valor absoluto, independientemente de si la máquina está a la derecha o izquierda de la traza paralela respecto a la que calcule este valor.

Para el cálculo de *despla* en Android creo dos ArrayList, uno para las trazas paralelas a la derecha de la traza inicial.

El valor de despla se calcula con la siguiente expresión:

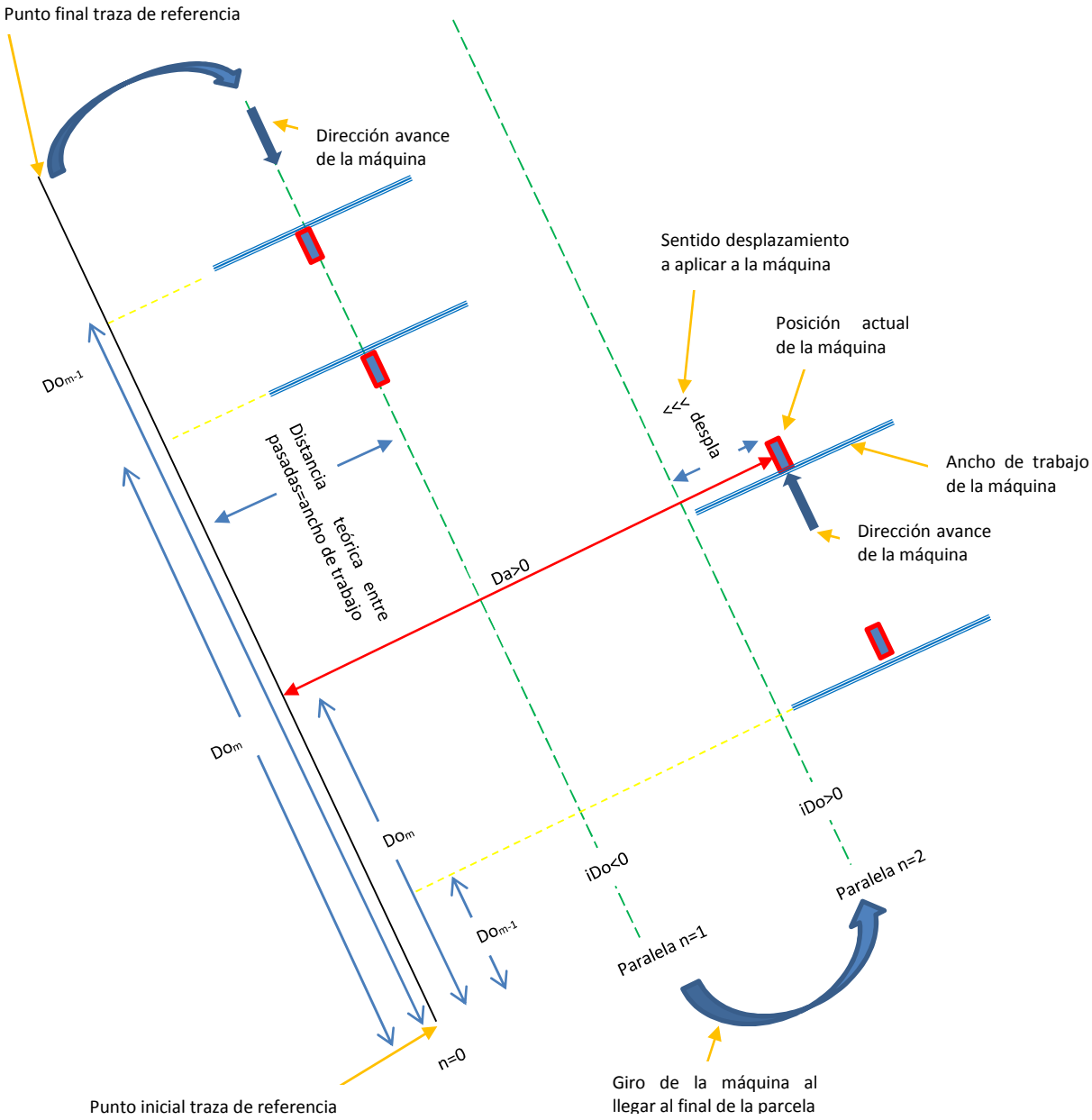
despla = paralelas.get(n) – Da; si Da es mayor que cero
despla = (paralelasnegativas.get(n)) – Da; si Da es menor que cero

paralelas y ***paralelasnegativas*** son dos ArrayList (clase que permite almacenar datos en memoria sin necesidad de declarar su tamaño ya que almacena datos de forma dinámica) creados en la aplicación, que contienen las distancias a la traza inicial, con signo positivo y negativo y por lo tanto a la derecha y a la izquierda respectivamente de esta. Con el método *.get(n)* se extraen los valores de ambos ArrayList que estén situados en la posición n, considerando que la posición 1 es cuando *n* vale 0.

El siguiente código da contenido a ambos ArrayList:

```
for(n=0;n<300;n++){
para= (n*ancho);
paranegativas=(-1*n*ancho);
paralelasnegativas.add(paranegativas);
paralelas.add(para);
}
```

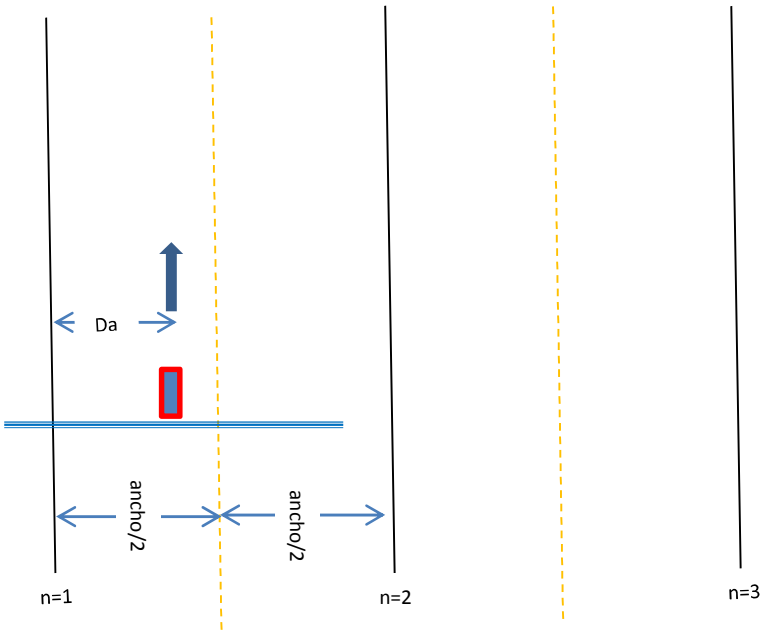
Con el bucle *for* se genera una secuencia de operaciones, mientras la variable *n* sea menor de 300 (número considerado como suficientemente amplio para el trazado de paralelas). Las variables para y *paranegativas* recogen el valor resultado de multiplicar *n* veces el valor del ancho de trabajo. Con el método *.add* se incorporan a los ArrayList correspondientes los valores calculados de para y *paranegativas*.



Croquis 2

El valor de *n*, parámetro que registra la paralela por la que circula la máquina, varía aumentando de uno en uno *(n++)* en función de los valores de *Da* y el ancho de trabajo. Para situaciones en las que la máquina se encuentra a la derecha de la traza de referencia inicial, por lo tanto *Da*>0, la expresión que selecciona el valor de *n* dentro del ArrayList paralelas es la siguiente:

```
n++
if(Da - (ancho*0.5)< paralelas.get(n)<Da+(ancho*0.5)){
Código para el cálculo de despla
Código que muestra en pantalla despla
Código que muestra en pantalla los símbolos de movimiento}
```



Croquis 3

Si por ejemplo, el ancho de trabajo es 10, *n* es 1 y por lo tanto el valor de paralelas.get(1) es también 10, y *Da* son 14 metros, *n* no cambia a 2 ya que:

$$14-5<10<14+5$$

Si el valor de *Da* fuese 16:

$$16-5<10<16+5$$

y al no cumplirse la condición *if* arriba indicada, la aplicación sale de este condicional *if* y el valor n se incrementaría en 1 (para el ejemplo dado pasaría a ser 2) y *paralelas.get*(2) devolvería del ArrayList paralelas el valor de 20 para el cual si se cumple que 16-5<20<16+5.

Para el caso de *Da*<0, es decir la máquina situada a la izquierda de la traza de referencia, el análisis es análogo pero considerando que el condicional *if* pasa a ser:

if (Da + (ancho * 0.5) <paralelasnegativas.get(n) < Da - (ancho * 0.5)){

Código para el cálculo de *despla*
Código que muestra en pantalla *despla*
Código que muestra en pantalla los símbolos de movimiento}

m) El valor mostrado en este textview son tres símbolos de mayor que (>>>) o menor que (<<<) que sirven de orientación al agricultor sobre el movimiento que ha de dar a la máquina para llevarla a una traza paralela a la inicial y separada de ella *n* veces al ancho de trabajo. El sentido de estos símbolos depende de tres parámetros:

1.- El valor de *Da*, ya sea este positivo o negativo.

2.- El valor de *iDo. iDo* es la variable que en la aplicación indica el sentido de avance de la máquina: si avanza por la traza paralela n en dirección hacia el final de la traza de referencia o si avanza por la paralela n en dirección al inicio de la traza de referencia. El cálculo de *iDo* se realiza de la siguiente manera:

2.1.- En primer lugar se calcula *Do* como ya se ha indicado en el punto 7 del apartado *k*. *Do*, como se muestra en el croquis 2, es la distancia medida a lo largo de la traza de referencia entre la proyección sobre esta traza de la posición de la máquina y el punto inicial de la traza de referencia.

2.2.- Se incorporan al ArrayList *distanciaorigen* los valores de *Do* con el método *.add:*

distanciaorigen.add(Do)

2.3.- Con un bucle *for* se calcula el valor de *iDo*, como la diferencia entre las distancias *Do* en un instante m y el instante *m-1* anterior, de tal manera que si *Do* en el instante *m* es mayor que *Do* en el instante *m-1* quiere decir que la máquina se desplaza hacia el final de la traza de referencia (según se observa en la paralela *n=2* del croquis 2). De la misma manera si *Do* en el instante *m* es menor que en el instante m-1, quiere decir que la máquina de desplaza hacia el inicio de la traza de referencia (según se observa en la paralela *n*=1 del croquis 2).

for (int m = 1; m < distanciaorigen.size(); m++) {

iDo = distanciaorigen.get(m) - distanciaorigen.get(m - 1)
..
..
..}

3.- El valor de *despla*. La orientación de los símbolos depende de si *despla* es mayor que 0 o menor que 0.

n) Botón **VOLVER AL MENU INICIO**. Pulsando este botón volvemos al menú inicio y por lo tanto al layout activity_menu_inicio.xml.

La imagen 31 muestra un contenido real del layout activity_guiado.xml durante una prueba de campo realizada con la aplicación Agrisoft. En ella se puede apreciar todos los campos arriba explicados:

- **Campo b)** El valor del ancho de trabajo introducido por entrada de teclado ha sido de 10 metros.

- **Campo d)** El valor de la precisión con la que se obtiene la posición procedente del método de Android *get.Accuracy()* es de 0.5.

- **Campo e)** Como el valor de la precisión es menor que el umbral con el que está configurada la aplicación (10) se activa este campo que recoge las coordenadas UTM en el Sistema de Referencia ETRS89 de la antena de receptor GPS u-Blox y que se actualiza cada segundo. Los valores mostrados son 422.046,863….para la coordenada X y 4.569.325,552…..para la coordenada Y.

- **Campo g)** Como el valor de la precisión es menor de 10 se activa este campo mostrando un a valor de 9,5859….km/h.

- **Campo i)** Una vez pulsado el botón **INICIO TRAZA**, solo permitido cuando la precisión está por debajo de 10, nos aparecen las coordenadas seleccionadas para el inicio de la traza de referencia. En este caso son 422.073,714….para la coordenada X y 4.568.964,734…. para la coordenada Y.

- **Campo j)** De igual manera, una vez pulsado el botón **FINAL TRAZA**, nos parecen las coordenadas seleccionadas para el final de la traza de referencia. En este caso 422.044,673…. para la coordenada X y 4.569.257,589…. para la coordenada Y.

- **Campo k)** Una vez seleccionados los puntos inicio y final de la traza de referencia se activa este campo, cuyo valor, 8,886… indica que la antena del receptor GPS se encuentra a la derecha de la traza de referencia y a 8,88… metros de distancia.

- **Campo l)** Activado el campo k, se activa el campo l, que nos indica la distancia que debe moverse la máquina que tiene el receptor GPS para situarse en una paralela a la traza de referencia. En este caso 1,113 metros lineales para situarse a 10 metros de la traza de referencia, pues esta es la anchura de trabajo. Dado el valor del campo k y el valor del campo l se deduce que el valor de *n* (parámetro que nos indica el número de la paralela en la que se encuentra la máquina) es 1.

- **Campo m)** Este campo se activa a la vez que el campo l y del sentido de los símbolos, >>>, se deduce que la máquina se encuentra a la izquierda de la traza paralela número *n*=1, se desplaza hacia el final de la traza de referencia (*iDo*>0) e indica que el agricultor ha de desplazar la máquina hacia la derecha 1,113 metros para estar a 10 metros (ancho de trabajo) de la traza de referencia y meterse en la paralela número 1.

Imagen 31.- Pantalla guiado durante la prueba

Capítulo 6. Resultados

La aplicación Agrisoft y el receptor u-Blox que la complementa, han sido probados en dos situaciones diferentes. Por un lado, emulando un receptor GPS topográfico, tomando las coordenadas de varios puntos y comparando las coordenadas obtenidas con las tomadas previamente con un receptor Leica comercial. La otra situación es emplearla para el uso que ha sido concebida: su empleo en una operación de campo de tratamiento con un pulverizador.

6.1. Toma de datos en campo y comparación con datos obtenidos con receptor GPS Leica comercial

La finalidad de dicha prueba era obtener las coordenadas planimétricas en el Sistema de Coordenadas ETRS89 proyección UTM, de 4 puntos previamente señalizados. La toma se ha realizado por duplicado: en primer lugar con un receptor GPS comercial de una marca puntera en el sector como es la casa Leica; en segundo lugar se han tomado los mismos puntos con el receptor u-Blox empleado en este PFM y la aplicación Agrisoft que permite transformar las coordenadas geográficas obtenidas por el receptor en coordenadas UTM en el Sistema ETRS89.

La prueba se realizó el 17 de agosto a las 9 de la mañana, en un entorno despejado donde los obstáculos existentes en el entorno (dos casas de dos plantas cada una y a distancias cercanas a los diez metros) no supusieron en ningún momento cortes en la recepción de la señal de los satélites.

En ambos casos se aplicaron correcciones diferenciales procedentes de la red GNSS del ItaCyL, desde su punto de montaje situado en la ciudad de León (denominado dentro de la red como LEON3). Dicho punto de montaje está equipado con un receptor Leica GR25 con antena LEIAR25 NONE y sus coordenadas son las siguientes:

- Latitud: 42º 35´18,259034 N
- Longitud: 5º 39´03,512866 W
- Altura elipsoidal: 970,234

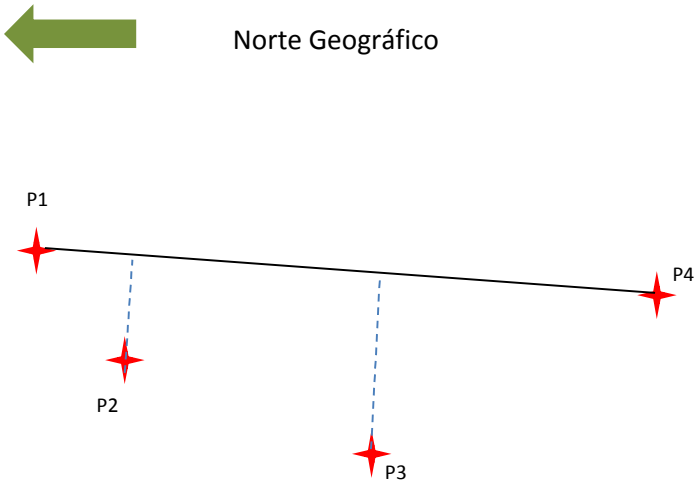
El receptor Leica utilizado ha sido el modelo CS20 de la gama Captivate con antena modelo GS16. La toma de puntos con este receptor se ha realizado durante tres segundos de observación, una sola vez y con una máscara de elevación de 10º sobre el horizonte, por lo tanto señales de satélites GPS que lleguen a la antena por debajo de ese ángulo no entran a formar parte del cálculo y son desechadas.

Para una toma más precisa de los puntos en ambos métodos, se ha empleado un trípode para aplomar con mayor exactitud las antenas.

Imagen 32.- Receptor Leica CS 20 y antena GS16 utilizados

Imagen 33.- Receptor Leica sobre uno de los puntos a tomar

Croquis de la situación de los puntos tomados:



Croquis 4

Las coordenadas planimétricas de los puntos tomados con el receptor GPS Leica CS20 son las siguientes:

PUNTO	PUNTOS CON RECEPTOR LEICA GS20		
	X	Y	HDOP
P1	290.053,369	4.709.680,223	0,008
P2	290.049,733	4.709.680,413	0,008
P3	290.047,792	4.709.676,420	0,008
P4	290.050,524	4.709.669,982	0,007

Tabla 1.- Coordenadas de los puntos tomados con el receptor Leica

De la tabla 1 se puede deducir que la precisión planimétrica en la toma de puntos ha sido muy buena dado que el parámetro HDOP en ningún momento alcanza el centímetro, por lo tanto la toma de datos con el receptor Leica puede afirmarse que ha sido del orden del centímetro.

La toma de los mismos puntos con el receptor u-Blox y la aplicación Agrisoft, aplicando correcciones diferenciales recibidas vía internet (con buena cobertura de internet) desde el punto de montaje de LEON3 perteneciente a la red GNSS del ItaCyL, se hizo en idénticas condiciones atmosféricas, el mismo día pero 15 minutos más tarde. Para conseguir una mayor fiabilidad en la toma de datos se repitió la toma de datos cuatro veces por cada punto, por lo que tenemos redundancia de observaciones. La decisión de tener exceso de observaciones deriva de no poder homogeneizar la configuración de los dos receptores: la aplicación Agrisoft está configurada para tomar un punto y mostrar sus coordenadas en pantalla en el momento de pulsar los botones INICIO TRAZA Y FINAL TRAZA y no para medir un punto durante tres

segundos y obtener sus coordenadas después de este periodo de observación como hace el receptor Leica; en el receptor u-Blox desconozco si se puede configurar la toma de datos con máscara de elevación para evitar señales de satélites que introduzcan un error excesivo en los cálculos. Según el método de Android getAccuracy, mostrado en pantalla en la aplicación Agrisoft, la precisión de los puntos tomados con el receptor u-Blox ha sido de 0,5. Se hizo la prueba de estudiar este valor durante el mismo momento de la toma de puntos sin considerar correcciones procedentes de la red GNSS del ItaCyL y empleando el receptor GPS de la Tablet. Dicho valor no bajo de 3, por lo tanto aplicando correcciones diferenciales se consigue un aumento considerable de la precisión. Para dar corriente al receptor GPS, dado que no tiene batería, ha sido necesario el uso del ordenador portátil, siendo esta la única misión del ordenador. Mediante una de las salidas USB del ordenador y mediante un cable de dos hilos se le abastece de energía. La Tablet, para su funcionamiento, se abastece de su propia batería. Cuando el receptor GPS se conecta al ordenador, se enciende una luz azul en el mismo de manera continua. Cuando dicha luz pasa a parpadeante, significa que ya está recibiendo señal de los satélites y que por lo tanto puede enviar datos de coordenadas geográficas a la Tablet vía Bluetooth. El Bluetooth HC-06 se conecta al receptor GPS mediante 4 cables.

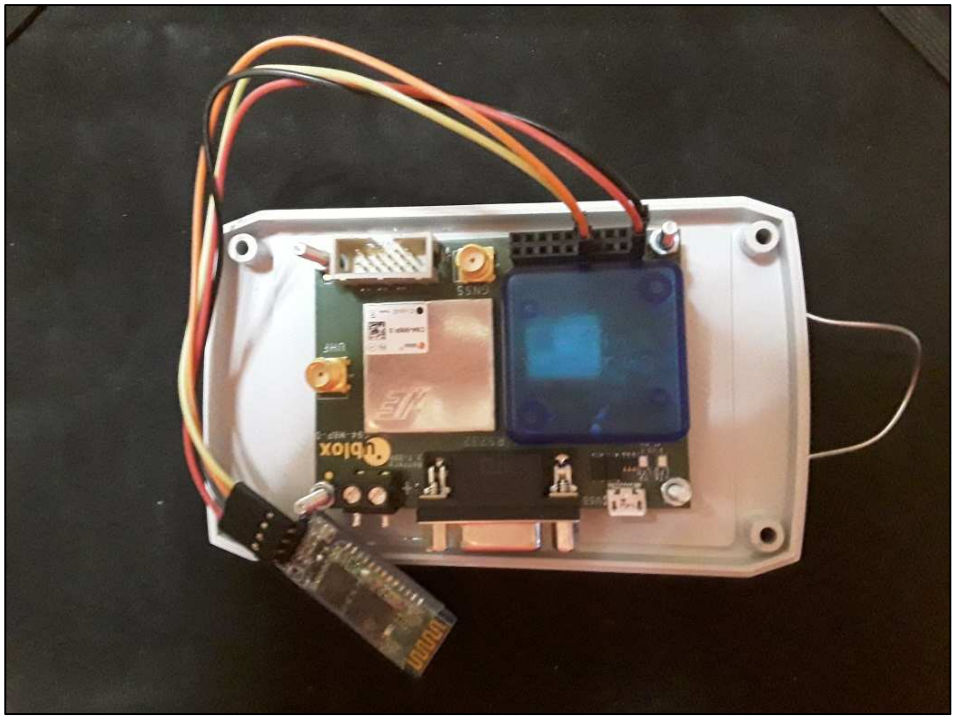


Imagen 34.- Bluetooth conectado al receptor u-Blox

En la Tablet hay que desactivar su receptor GPS. Para ello en *ajustes*, se desactiva la opción *Ubicación* y de esta manera solo tomará datos de posición de las señales de receptores GPS externos a ella. También hay que comprobar que tiene activado el Bluetooth. Emparejado una vez el Bluetooth de la Tablet con el Bluetooth HC-06, ya no será necesario volver a emparejarlos más. A continuación se arranca la aplicación Lefebure con la cual vamos a enviar correcciones diferenciales al receptor GPS el cual las procesará conjuntamente con sus observaciones de satélites para calcular una posición precisa. Arrancada la aplicación Lefebure, y configurada esta como previamente se ha explicado en el capítulo 5, se pulsa el botón *Connect* y después de unos segundos de intentar establecer contacto con el dispositivo Bluetooth HC-06, una vez conseguida la comunicación se muestra una barra de estado que a medida que se envía información se rellena de color amarillo. En el momento que se establece

la comunicación con el Bluetooth HC-06, la luz testigo de este dispositivo pasa a ser roja fija y en el receptor u-Blox se activa una nueva luz de color verde fija que significa que está recibiendo correcciones diferenciales Bluetooth y enviando datos de posición a la Tablet por el mismo medio, en formato NMEA (ver imagen 36).

A continuación salgo de la aplicación Lefebure, sin cerrarla, se queda emitiendo correcciones y arranco la aplicación Agrisoft. Para la acción que queremos realizar, tomar datos de coordenadas de cuatro puntos, en el menú inicio pulso el botón *GUIADO* y accedemos al layout correspondiente. En este, es premisa indispensable introducir un ancho de trabajo para dar acceso a la toma activación de la toma de dato. Se introduce cualquier ancho de trabajo y cuando la precisión baja del valor 10, se activa la transformación de las coordenadas geográficas, a UTM en el Sistema de Coordenadas ETRS89, enviadas vía Bluetooth desde el receptor GPS a la aplicación Agrisoft y por lo tanto podemos pulsar los botones *INICIO TRAZA* y *FINAL TRAZA*. Cuando dicha precisión se considera suficiente, pulsando ambos botones queda registrado en pantalla las coordenadas X e Y de los puntos seleccionados. Como han sido cuatro tomas por cada punto, se ha pulsado dos veces el botón *INICIO TRAZA* y otras dos el botón *FINAL TRAZA*.

Imagen 35.- Receptor u-Blox+Agrisoft en la toma del punto 2

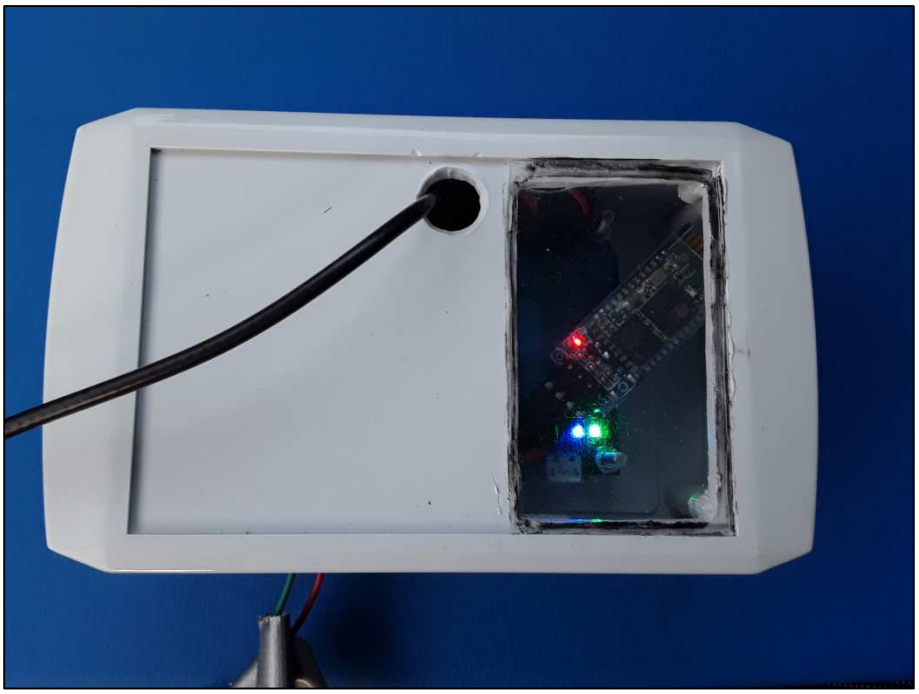


Imagen 36.- Luces encendidas en Bluetooth y receptor u-Blox durante pruebas

Las coordenadas planimétricas de los puntos tomados con el receptor GPS u-Blox más la aplicación Agrisoft son las siguientes:

PUNTO	TOMA	PUNTOS CON RECEPTOR U-BLOX Y APLICACIÓN AGRISOFT		
		X	Y	HDOP
P1	1	290.051,079	47.09.680,56	0,5
	2	290.050,886	4.709.680,529	0,5
	3	290.050,858	4.709.680,511	0,5
	4	290.050,885	4.709.680,49	0,5

PUNTO	TOMA	PUNTOS CON RECEPTOR U-BLOX Y APLICACIÓN AGRISOFT		
		X	Y	HDOP
P2	1	290.047,237	4.709.679,954	0,5
	2	290.047,316	4.709.679,84	0,5
	3	290.047,862	4.709.679,768	0,5
	4	290.047,864	4.709.679,861	0,5

PUNTO	TOMA	PUNTOS CON RECEPTOR U-BLOX Y APLICACIÓN AGRISOFT		
		X	Y	HDOP
P3	1	290.044,629	4.709.675,735	0,5
	2	290.045,197	4.709.676,366	0,5
	3	290.045,046	4.709.676,389	0,5
	4	290.045,33	4.709.676,251	0,5

PUNTO	TOMA	PUNTOS CON RECEPTOR U-BLOX Y APLICACIÓN AGRISOFT		
		X	Y	HDOP
P4	1	290.047,969	4.709.670,577	0,5
	2	290.047,938	4.709.670,429	0,5
	3	290.048,143	4.709.669,979	0,5
	4	290.048,25	4.709.669,883	0,5

Tabla 2.- Coordenadas de los puntos tomadas con receptor u-Blox y aplicación Agrisoft

PUNTO	COORDENADAS PROMEDIO PUNTOS RECEPTOR U-BLOX Y APLICACIÓN AGRISOFT		
	X	Y	HDOP
1	290.050,927	4.709.680,523	0,5
2	290.047,5698	4.709.679,856	0,5
3	290.045,0505	4.709.676,185	0,5
4	290.048,075	4.709.670,217	0,5

Tabla 3.- Coordenadas promedio de los puntos tomadas con receptor u-Blox y aplicación Agrisoft

A continuación, en la tabla 4 se realiza un análisis de las diferencias entre las coordenadas obtenidas en las distintas tomas de cada punto, tabla 2 y las coordenadas promedio obtenidas en la tabla 3. Los valores máximos de dichas diferencias oscilan entre 0,45 metros y -0,36 metros (señalados en color amarillo en la tabla 4), valores por debajo y cercanos a la precisión horizontal estimada mostrada en la aplicación con el método de Android getAccuracy, 0,50.

PUNTO	TOMA	DIFERENCIAS RESPECTO A PUNTO PROMEDIO	
		ΔX	ΔY
P1	1	-0,152	-0,037
	2	0,041	-0,007
	3	0,069	0,012
	4	0,042	0,032

PUNTO	TOMA	DIFERENCIAS RESPECTO A PUNTO PROMEDIO	
		ΔX	ΔY
P2	1	0,333	-0,098
	2	0,254	0,016
	3	-0,292	0,088
	4	-0,294	-0,005

PUNTO	TOMA	DIFERENCIAS RESPECTO A PUNTO PROMEDIO	
		ΔX	ΔY
P3	1	0,421	0,450
	2	-0,146	-0,181
	3	0,005	-0,204
	4	-0,280	-0,066

PUNTO	TOMA	DIFERENCIAS RESPECTO A PUNTO PROMEDIO	
		ΔX	ΔY
P4	1	0,106	-0,360
	2	0,137	-0,212
	3	-0,068	0,238
	4	-0,175	0,334

Tabla 4.- Diferencias de coordenadas entre las diferentes tomas de los puntos tomadas con receptor u-Blox y aplicación Agrisoft y la posición promedio

Las coordenadas absolutas obtenidas por ambos métodos, receptor Leica CS20 y receptor u-Blox+Agrisoft difieren bastante, más de lo que cabría esperar dada la precisión con la que se han obtenido ambas. La tabla 5 muestra dichas diferencias:

PUNTO	DIFERENCIA COORDENADAS ENTRE RECEPTORES U-BLOX+AGRISOFT Y LEICA CS20	
	ΔX	ΔY
1	-2,442	0,299
2	-2,163	-0,557
3	-2,742	-0,235
4	-2,449	0,235

Tabla 5.- Diferencia coordenadas absolutas entre u-Blox+Agrisoft y receptor Leica CS20

Dado que la principal función de la aplicación Agrisoft, es el cálculo de las distancias a una línea de referencia para el guiado de maquinaria, se puede realizar dicho análisis en base a las coordenadas de los puntos 1, 2, 3 y 4 tomados en ambos sistemas. Según se aprecia en el croquis número 4, se considera como traza de referencia la formada por los puntos 1 y 4 y se realiza el análisis de las distancias a dicha traza de los puntos 2 y 3. La tabla 6 muestra las distancias calculadas y se observa unas diferencias de 0,496 metros para el punto 2 y de -0,151 metros para el punto 3. Estas diferencias en coordenadas relativas son bastantes menores que las diferencias en coordenadas absolutas mostradas en la tabla 5 y por debajo de la precisión de 0,50 obtenida en la toma de puntos con el sistema u-Blox+Agrisoft.

PUNTO	DISTANCIAS A LA TRAZA DE REFERENCIA 1-4		
	LEICA	U-BLOX+AGRISOFT	DIFERENCIA
2	3,554	3,058	0,496
3	4,356	4,507	-0,151

Tabla 6.- Distancias a una traza de referencia con sistema u-Blox+Agrisoft y receptor Leica CS20

Por lo tanto, se puede deducir, que en la condiciones de la prueba anteriormente resumida, la aplicación Agrisoft más el receptor GPS u-Blox permite obtener una precisión de +/-0,50 metros en su función de guiado de maquinaria agrícola, dado que se trabaja en coordenadas relativas, pero las coordenadas absolutas conseguidas no ofrecen una precisión suficiente para otros trabajos, como puede ser un posicionamiento absoluto para el replanteo de los linderos de una finca.

6.2. Prueba en campo de la aplicación Agrisoft en tratamiento de pulverización

La prueba de campo se ejecutó el día 2 de septiembre en condiciones óptimas para una buena recepción de señal GPS por parte del receptor u-Blox y para una buena recepción de correcciones diferenciales vía internet dado que el campo de trabajo era abierto y sin obstáculos.

La prueba consistió en simular una operación de tratamiento de pulverización con un pulverizador arrastrado de 17 metros de anchura y un tractor que le arrastra (ver imagen 37), haciendo uso de la función guiado que ofrece la aplicación Agrisoft y generando el fichero xml con los datos de la operación que se va a realizar.

Imagen 37.- Pulverizador y tractor empleados en la prueba

La prueba comienza con la instalación del receptor GPS en la cabina del tractor. Con este propósito y para proteger al receptor de polvo o golpes, se le ha preparado una caja de plástico en la cual va alojado y fijado con 4 tornillos aprovechando los cuatro orificios de sujeción que tiene la estructura del receptor (ver imagen 38). La caja dispone de dos orificios, uno para la entrada de corriente eléctrica y otro para la salida de la antena magnética la cual va situada en la cabina del tractor, en su parte alta y centrada en ella (ver imagen 39). A su vez se ha realizado una ventana en la parte superior de la carcasa y se ha protegido el hueco dejado con plástico transparente para ver el estado de las luces informativas que tiene el receptor.

La corriente eléctrica al receptor le llega de la toma de corriente eléctrica que dispone el tractor, tipo mechero. A esta toma de corriente se le adapta un enchufe con salida USB y desde este adaptador, mediante un cable de dos hilos le llega la corriente al receptor u-Blox. Una vez conectado el receptor a la corriente, se ilumina una luz azul de forma continua. Cuando dicha luz se transforma en parpadeante significa que ya está recibiendo señal de los satélites GPS y por lo tanto ya está en condiciones de enviar datos a la Tablet, vía bluetooth, de las coordenadas geográficas que obtiene para la posición de la antena.

Imagen 38.- Receptor GPS y Bluetooth dentro de caja protectora en cabina tractor

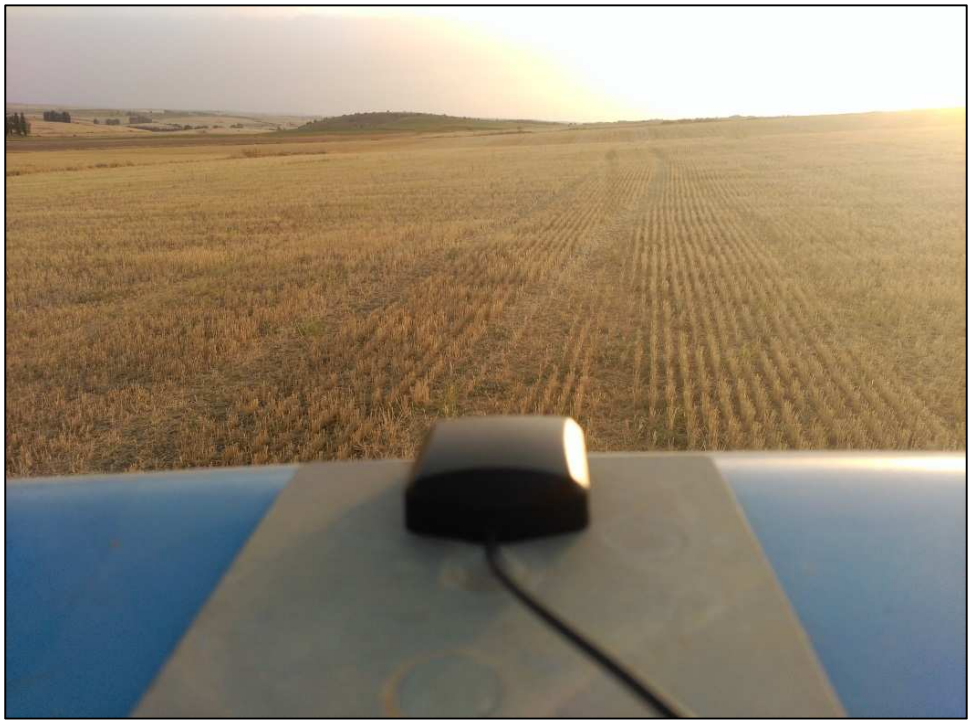


Imagen 39.- Antena magnética del receptor u-Blox sobre la cabina del tractor

A continuación se enciende la Tablet. Esta va situada en la luna delantera del tractor fijada a ella por una ventosa y contenida en un soporte apto para ella.

Encendida la Tablet, se arranca la aplicación Lefebure para el envío de correcciones diferenciales vía Bluetooth al receptor GPS como ya se ha explicado en el punto 6.1. En este caso el punto de montaje de la Red GNSS del ItaCyL de la que se reciben dichas correcciones está situado en la localidad de Riaza (RIAZ3), en la provincia de Segovia, a 20 kilómetros de la zona donde se realiza esta prueba de campo. Las coordenadas geográficas de dicho punto de montaje son las siguientes:

- Latitud: 41º 16´43,446680N
- Longitud: 3º 28´49,117651 W
- Altura elipsoidal: 1263,775

Comprobado que el receptor GPS u-Blox recibe señal de satélites GPS, recibe correcciones diferenciales vía Bluetooth desde la aplicación Lefebure, y que envía coordenadas de posición vía Bluetooh, se arranca la aplicación Agrisoft.

Desde el menú inicio se accede al layout relativo al registro de las operaciones a realizar con el botón *BASE DATOS.* En el layout relativo a las operaciones se pulsa el botón *TRATAMIENTO* para acceder a este tipo de operación. Dentro del layout relativo a la operación tratamiento se rellenar los campos considerados oportunos. Para la prueba realizada se rellena el campo *Lote*, con el nombre de la parcela que es *cruz*, se rellena el campo *Tratamiento 1,* con el texto *agua* (se simula una operación de tratamiento de pulverización con un producto químico, pero como realmente el pulverizador no lleva ningún producto, por eso se ha rellenado con el texto agua) y el campo *Dosis 1*, se rellena con el valor de 200 (se simula que la operación de tratamiento distribuye 200 litros de agua por hectárea). Con el botón *GUARDAR*, se guardan las entradas de teclado introducidas y se genera el fichero. Los textos introducidos se borran al pulsar el botón *GUARDAR*. El archivo Tratamiento.xml generado es el siguiente:

<?xml version="1.0"?>

-<TRATAMIENTO>

<FECHA>2018/9/2</FECHA>

<LOTE>cruz</LOTE>

<TRATAMIENTO1>agua</TRATAMIENTO1>

<DOSIS1>200</DOSIS1>

<TRATAMIENTO2/>

<DOSIS2/>

<TRATAMIENTO3/>

<DOSIS3/>

<TRATAMIENTO4/>

<DOSIS4/>

<DURACION/>

<OBSERVACIONES/>

</TRATAMIENTO>

Con el botón *VOLVER A BASE DATOS,* vuelvo al layout relativo a la selección del tipo de operación a realizar y una vez dentro de este layout, al pulsar el botón *VOLVER A MENU INICIO* se regresa al menú inicio.

Ahora activo la función guiado pulsando el botón *GUIADO.* Dentro del layout de esta función introduzco el ancho de trabajo, que como ya he comentado es de 17 metros. Cuando la precisión en la obtención de coordenadas baja de 10, se activa el campo *PRECISION HZ.* Una vez activado este campo, se muestran en pantalla las coordenadas cartesianas *X* e *Y,* en el sistema de coordenadas ETRS89, proyección UTM, de los puntos que calcula el receptor GPS en coordenadas geográficas en el Sistema de Coordenadas WGS84, envía a la aplicación Agrisoft y esta los transforma el sistema ETRS89. También se activa la velocidad que lleva la máquina gracias y que aparece a la derecha del campo *VELOCIDAD (km/h*). Esta valor de la velocidad también se extrae del protocolo NMEA con el método de Android *getSpeed().* Esta velocidad varía respecto de la que marca el velocímetro digital del tracto como máximo en +/-0,2km/h. El método *getSpeed()* proporciona el valor de la velocidad en metros por segundo, y por lo tanto ha sido necesario crear un código en la aplicación Agrisoft para que se muestre en kilómetros hora.

Dado que la frecuencia de actualización de las coordenadas obtenidas de un receptor GPS por medio de los métodos getLongitude() y *getLatitude()* de Android, es como máximo de 1 actualización por segundo, y que la velocidad media de trabajo de un tractor puede ser del orden de 7 km/h, esto significa que durante ese segundo el tractor avanza 1,94 metros lineales, que como norma general no afectaría a los trabajos de guiado, pero para trabajos a velocidades superiores (o que requieran más precisión, como trabajos de cosecha), por ejemplo a 14 km/h, sí podría significar un retardo importante en la actualización de la posición, que podría indicar que durante 4 metros de avance al maquinista no tiene referencia.

Situado en la parcela o lote en la posición por donde quiero empezar a trabajar, y una vez que el valor de *PRECISION HZ*, tiene un valor adecuado, (se ha seleccionado el punto con precisión 0,50) selecciono el punto que será el inicio de la traza de referencia con el botón *INICIO TRAZA.* A la izquierda de este botón aparecen las coordenadas seleccionadas. Durante la prueba el valor de *PRECISION HZ* ha variado únicamente entre los valores 0,02 y 0,50, sin ningún tipo de valor intermedio.

Recorridos unos metros por la parcela, y cuando la *PRECISION HZ* lo permita, que no sea excesiva, aunque ese caso no se ha dado como he comentado anteriormente, selecciono el final de la traza de referencia con el botón *FINAL TRAZA*. Las coordenadas de este punto aparecen a la derecha del botón pulsado.

Por lo tanto ya tenemos la traza de referencia seleccionada y se activa el campo que muestra la distancia *Da* a la traza de referencia. A continuación giro el tractor a un lado y cuando el valor de *Da* supera la mitad del ancho de trabajo, es decir 8,50 metros, la aplicación considera que me desplazo hacia una nueva traza paralela a la referencia, situada a 17 metros y se activan los campos con fondo en color naranja, *despla* y el campo con símbolos *>>>* y *<<<.*

Ambos campos con fondo en color naranja me orientan en el valor (campo *despla*) y en el sentido (campo con valores *>>>* y *<<<*) que he de desplazar el tractor para mantenerlo en una traza paralela a la de referencia 17 metros.

Imagen 40.- Aplicación Agrisoft durante la prueba de campo en operación de tratamiento y en función guiado

Capítulo 7. Conclusiones

La realización de este PFM ha supuesto la adquisición de nuevos conocimientos en materias nuevas:

1.- Programación en lenguaje Java empleando el entorno de desarrollo Android Studio.

2.- Protocolos de transferencia de información GPS.

3.- Receptores de satélites GPS de bajo coste.

La programación en lenguaje Java en el entorno de desarrollo Android Studio, no ha estado exenta de dificultades. Por ejemplo, los cambios de versiones del entorno de desarrollo suponían la aparición de errores en el código que hasta la aparición de la versión nueva no implicaban ninguna distorsión. La eliminación de dichos errores suponía la consulta en enlaces de internet especializados en programación en Java. A veces la solución era drástica y consistía en reinstalar Android Studio. Los errores de código eran solucionados también mediante consulta en enlaces de internet y visualizando canales de youtube donde expertos en Android Studio desarrollaban aplicaciones de las cuales podía extraer algún detalle para la aplicación Agrisoft.

Han sido muchas las horas visualizando videos de programación en Android Studio y de consultar páginas de internet, de avanzar con el "*método de aprendizaje prueba-error*" y al final del proyecto me doy cuenta que de todas esas horas (una cuenta de brocha gorda podría estimarlas en 4 meses de trabajo, 16 semanas y 7 horas a la semana, en total 112 horas), las verdaderamente válidas han sido muy pocas. Pero claro, a esa conclusión se llega cuando ya se tiene unos conocimientos básicos en la materia, saber dónde buscar y qué consultar. Ese es el peaje de partir de cero en un PFM como este empleando el lenguaje de programación Java.

El lenguaje de programación Java en Android Studio, personalmente, me sigue pareciendo más abstracto que los otros lenguajes de programación que he manejado, ya obsoletos como C y Basic en M-DOS, lento y es necesario estar atento a los posibles cambios que puede sufrir Android Studio y que conlleven actualizaciones.

A día de hoy, me quedo con la gran fuente de información que proporciona internet en el entorno de desarrollo Android Studio y lenguaje de programación Java. De hecho, elegí el sistema operativo Android por recomendación de uno de los tutores de este PFM cuando en el mes de julio del año 2017 me puse en contacto con él para plantearle mí idea de PFM. Con esa fuente de información veo grandes posibilidades de mejora en la aplicación Agrisoft, de incorporarla nueva opciones, como las que se reseñan en el capítulo 8.

Las técnicas GPS, desde hace 15 años han sido mi herramienta de trabajo, no siempre de manera continua, pero principalmente como usuario de una aplicación de una casa comercial (casi siempre Leica) de tal manera que dejaba de lado los conocimientos relativos a protocolos de transferencia de datos, ya que el equipo se te entrega listo para trabajar y solo en contadas ocasiones ante la aparición de algún problema te ves en la necesidad de "*trastear*" con el equipo GPS. En realidad, al final si no muestras interés te conviertes en un mero "*pulsador*" de botones, con conocimientos sobre el programa que la casa fabricante del receptor GPS ha realizado para tomar taquimétricos, replantear puntos, etc.

Con la realización de este PFM, me acerco a esa parte de la tecnología GPS, que hasta ahora para mí era "*simple teoría*", y que ahora he encontrado su función práctica. El protocolo NMEA me ha servido para conocer esa teoría, comprenderla y en un futuro desarrollar otras aplicaciones para Android que me permitan realizar trabajos topográficos (principalmente replanteos y taquimétricos).

Los receptores de bajo coste, como el empleado en este PFM, suponen otra fuente de conocimiento nueva y de desarrollo para un futuro. De las diversas mejoras que planteo realizar en la aplicación Agrisoft en un futuro cercano, algunas son relativas al receptor GPS. Mejorar la precisión de la función *guiado* o introducir opciones de posicionamiento absoluto con garantía de precisión y fiabilidad, implican buscar algún otro receptor que con las características reseñadas en el capítulo 8 me proporcione mayor precisión.

De igual manera, la ventana abierta anteriormente al desarrollo de una aplicación para poder realizar trabajos topográficos, necesariamente precisa mejorar el receptor GPS y la antena.

En definitiva, como principal conclusión a la que llego una vez acabado el PFM y por lo anteriormente expuesto, el PFM aquí mostrado no es un punto y final, sino un punto y seguido en el desarrollo de aplicaciones en lenguaje de programación Java bajo el entorno de desarrollo Android Studio.

Esa es para mí la principal enseñanza de este Master, que lo visto, aprendido y probado me sirva en el futuro.

Capítulo 8. Desarrollo futuro de la aplicación Agrisoft

El desarrollo futuro de la aplicación Agrisoft es muy amplio. La actividad agraria hoy en día demanda más tecnología y más información sobre todos los elementos que intervienen en ella. Ambas demandas pueden unirse en una aplicación como la presentada en este PFM.

Todos los recursos empleados en este PFM tienen además posibilidades de avance para mejorar y ampliar las opciones que la aplicación Agrisoft ofrece en el momento de presentar este PFM. Las opciones de mejora pueden resumirse en tres puntos:

1.- La tecnología GPS, con receptores cada vez más asequibles, constelaciones de satélites cada vez más amplias, instituciones como el ItaCyL que proporcionan correcciones diferenciales, aumentando y mejorando su red de puntos de montaje o emisores de correcciones. Todo ello tiene como consecuencia una mejora en la precisión de la posición obtenida por el receptor GPS y una mejora en la estabilidad y fiabilidad de dicha posición.

2.- Recursos informáticos con cada vez más potentes procesadores.

3.- La programación Java empleando el entorno de desarrollo Android Studio.

A continuación enumero alguno de los posibles desarrollos futuros que puede tener la aplicación Agrisoft:

1.- Mejoras concernientes a la tecnología GPS. Las aplicaciones agrícolas requieren diversas precisiones: precisiones de +/-30cm para tratamientos de fitosanitarios, de +/-50 cm para operaciones de abonado (ambas operaciones requieren cierto solape entre pasadas consecutivas para una buena aplicación); precisiones de +/- 10cm para trabajos de siembra, labranza y cosecha de cereal. Para mejorar la precisión conseguida en las pruebas de campo de esta aplicación reflejada en el capítulo 6 de este PFM, achacable a los recursos GPS empleados, se pueden adoptar las siguientes soluciones:

   1.1.- Emplear receptores más precisos. Por ejemplo, sería suficiente emplear receptores con frecuencia L1 con observables de código C/A y fase que pueden proporcionar precisiones de hasta 1cm+2ppm. El aspecto negativo de esta opción es que a mayor precisión en la posición dada por un receptor GPS más elevado es su precio y por lo tanto es otra variable a considerar en el momento de compra de un equipo GPS para trabajos agrícolas. Según se desprende de la prueba de campo descrita en el capítulo 6.1, la precisión para trabajos de guiado del receptor u-Blox es +/-0,40 metros. Para trabajos de abonado o pulverización podría ser suficiente, aunque sería recomendable mejorarla pero para trabajos de arada o cosecha esa precisión es insuficiente.

   1.2.- Emplear una antena de recepción de señal GPS de mayor precisión. Las antenas receptoras de la señal GPS tienen gran influencia en la precisión final conseguida. Las antenas de baja calidad no son capaces de eliminar el efecto multipath de la señal. El empleo de antenas comerciales con el receptor u-Blox podría mejorar los resultados de precisión obtenidos.

2.- Mejoras relativas a programación y desarrollo informático.

2.1.- Desarrollar un algoritmo en la aplicación que permita realizar guiado respecto de líneas poligonales, no solo respecto de líneas rectas. El agricultor tendría que elegir entre trazas de referencia rectas o poligonales. Habría que implementar un layout intermedio entre el layout activity_menuinicio.xml y el layout activity_guiado.xml en el cual seleccionar la tipología de la traza de referencia: recta o poligonal.

2.2.- Incorporar al layout activity_guiado.xml, ya sea con guiado respecto a una traza de referencia recta o respecto a una traza de referencia poligonal, un mapa con la cartografía catastral de la zona de trabajo. De esta manera el agricultor, ante linderos de parcelas dudosos, podría saber en qué momento ha llegado al final de la parcela en la que se encuentra trabajando y de esa manera no sobrepasar los límites de esta.

2.3.- Incorporar una función a la aplicación para obtener fotografías sin necesidad de salir de ella. Consistiría en incorporar un botón en todos los layout que activase la cámara fotográfica de la Tablet y una vez activada esta usar la aplicación cámara de la Tablet para obtenerla y registrarla.

2.4.- Incorporar un layout, accesible desde el inicial de la aplicación Agrisoft con un nuevo botón (botón ENLACES), que nos permitiera acceder a enlaces de internet desde la propia aplicación, de importancia para el sector agrario (páginas web con información meteorológica, con información sobre insumos, páginas web de las Administraciones Públicas con información agrícola, información sobre las lonjas y mercados agrarios, etc.).

2.5.- Aprovechando las imágenes procedentes de los satélites de la constelación Sentinel, satélites Sentinel 2A y Sentinel 2B, se podrían incorporar dicha imágenes, convenientemente procesadas por otros software (como puede ser el Sistema de Información Geográfica gratuito QGIS), en las operaciones de tratamiento aplicando abonos químicos. Las imágenes procesadas pueden mostrar índices de vegetación, como el Índice Normalizado de Diferencia de Vegetación (NDVI). Cuanto más cercano a 1 es este índice la planta goza de mejor salud y tiene un mayor vigor vegetativo y por lo tanto puede requerir una menor dosis de abono.

Estas imágenes procesadas que muestran el índice NDVI junto al posicionamiento que obtenemos de la aplicación Agrisoft, podría ofrecer información al agricultor sobre las necesidades de abono químico que tiene su cultivo y por lo tanto realizar una distribución variable del mismo en función de la ubicación que tiene dentro de la parcela y de la información que el ofrece la imagen del satélite en la posición en la que se encuentra. Como complemento a esta información que ofrecería la aplicación Agrisoft y desde un punto de vista de mejorar también la eficiencia del tiempo de trabajo, el agricultor debería tener una máquina distribuidora de abono (un pulverizador si es abono líquido o una abonadora si es abono sólido) que le permitiera variar la dosis por unidad de superficie desde la propia cabina del tractor, sin necesidad de hacer ajustes bajando de esta.

La constelación de satélites Sentinel permite disponer de imágenes de 10 metros de resolución en el espectro visible e infrarrojo cercano con una frecuencia al menos cada 5 días y en algunas zonas cada 2-3 días. La Agencia Espacial Europea

(ESA) ofrece dichas imágenes de manera gratuita y las publica de forma regular en tiempo casi real. Por lo tanto el agricultor tendrá información muy precisa del estado de sus cultivos en el momento de realizar las operaciones de abonado. Este insumo, el abono químico, es el insumo más caro de los empleados por el agricultor, por lo que tener una información precisa y en el momento adecuado del estado de sus cultivos puede ayudarle a disminuir dicho coste.

2.6.- Incorporar algoritmos en la aplicación que permitan al agricultor tener constancia de la superficie que ha trabajado y lo que le queda por trabajar dentro de una parcela. Dicha información se añadiría al layout activity_guiado.xml y consistiría en añadir dos textviews que mostrasen dicha información.

2.7.- Incorporar la posibilidad de registrar en un fichero .xml la superficie tratada, guardando en dicho fichero no solo la posición planimétrica sino también la altitud respecto del nivel del mar. Con dicho "taquimétrico", empleando terminología topográfica, el agricultor tendría un levantamiento 3D de sus parcelas, de interés para estudios de irrigación o para cultivar siguiendo las curvas de nivel del terreno y así reducir los efectos de la erosión hídrica.

2.8.- Implementar el autoguiado. El autoguiado permite al agricultor, mientras este no lo quiera, no tener el control de los movimientos sobre el volante de su máquina de tal manera que el desplazamiento a realizar a la máquina lo realizan unos servomotores que reciben de la aplicación el sentido de giro que han de aplicar al volante. Este sistema ha de tener un dispositivo para activarse solo cuando la máquina ya esté en la traza paralela a la de referencia, el agricultor en ese momento pueda perder el control del volante y mantener su atención en la buena ejecución de la tarea agrícola y recuperar el control del volante cuando llegue al final de la parcela para realizar las operaciones de maniobra y continuar con otra traza paralela a la de referencia.

Bibliografía

1.- Tecnología en la agricultura

- https://smartbiosystem.com/agrotecnologia/
- https://www.iagua.es/blogs/rodrigo-munoz-herraiz/nuevas-tecnologias-agricultura-optimizar-cultivos
- http://www.sofoscorp.com/impacto-tecnologia-aplicada-agricultura/
- http://agriculturers.com/nuevas-tendencias-en-tecnologia-agricola/
- https://www.agroptima.com/blog/sentinel-agricultura-precision/
- https://hemav.com/introduccion-a-la-teledeteccion-agricola/
- https://grupomsc.com/blog/medio-ambiente/agricultura-tradicional-agricultura-moderna

2.- Programación en Android Studio

2.1.- Enlaces de internet. Los más destacados dentro de los muchos consultados

- http://www.sgoliver.net/blog/entorno-de-desarrollo-android-android-studio/
- *https://andreaardions.com/android-studio-faqs*
- https://stackoverflow.com/questions/tagged/android-studio
- https://developer.android.com/studio/
- https://www.dosmweb.com/blog/index.php?post/2016/03/28/OpenStreetMap-en-Android-Studio
- http://www.aprendeandroid.com/
- https://www.lawebdelprogramador.com/cursos/Android/8784-Programacion-en-Android-Studio.html
- *https://www.youtube.com/channel/UC_tBJwfiDqQCGsHYUG03XmA, canal en youtube en programación Android de Fran García*
- *https://www.youtube.com/channel/UCmfkhTdPGucKQeQnadOizCQ, canal en youtube en programación del profesor de la UPV (Universidad Politécnica de Valencia) Jesús Tomas*

2.2.- Libros

- Tomás Girones, Jesús. El gran libro de Android. Ediciones Marcombo. 2017

3.- Tecnología GPS

3.1.- Enlaces de internet

- ⇨ http://gnss.itacyl.es/
- ⇨ https://www.u-blox.com/
- ⇨ http://cartodruid.es/index.php/2018/01/31/asociar-un-gps-externo-a-un-dispositivo-android-para-el-uso-de-la-posicion-gps-mejorada-con-cartodruid/
- ⇨ http://lefebure.com/software/ntripclient/
- ⇨ http://www.tallysman.com/
- ⇨ https://www.novatel.com/
- ⇨ https://www.gpsinformation.org/dale/nmea.htm
- ⇨ https://www.use-snip.com/kb/knowledge-base/where-do-i-get-a-free-ntrip-client/
- ⇨ http://www.effigis.com/es/lograr-precision-centimetrica-con-receptores-l1-de-precio-asequible/
- ⇨ https://www.agps.es/localizador-para-vehiculos-modelo-agps-0100-pro/

3.2.- Otro material

- ⇨ Apuntes de la asignatura del Master Geotecnologías Cartográficas en Ingeniería y Arquitectura, Posicionamiento y Navegación.
- ⇨ Apuntes de la asignatura del Master Geotecnologías Cartográficas en Ingeniería y Arquitectura, Procesamiento y Gestión de datos de posicionamiento y navegación.
- ⇨ Apuntes de la asignatura del Master Geotecnologías Cartográficas en Ingeniería y Arquitectura, Procesamiento y Gestión de datos de Sensores Híbridos.

4.- Proyectos Fin de Master consultados

- ➢ Mauricio Freire Bastidas. Sistemas de georreferenciación para trazabilidad en la entrega de paquetes. Universidad de Vigo, Master en Geoinformática. Julio 2016
- ➢ Hector Sancho Chilet. Desarrollo de un sistema de localización y aplicación móvil para vehículos en aparcamientos. Universidad Politécnica de Valencia, Grado en Ingeniería en Tecnologías Industriales. Curso 2016-2017
- ➢ Iñigo de la Fuente Patiño. Estudio de la Mejora en la Precisión en los Receptores GPS. Universidad del País Vasco, Ingeniería Técnica en Informática de Sistemas. Julio 2012
- ➢ Oscar Cuadrado Méndez. Estudio y análisis de diferentes sistemas de posicionamiento mediante instrumentación de bajo coste y herramientas de procesado de software libre. Universidad de Salamanca, Master en Geotecnologías Cartográficas en Ingeniería y Arquitectura. Julio 2011

- David Peris Martínez. Desarrollo de navegador para coches basado en Android y OpenStreetMap. Universidad Politécnica de Valencia, Grado en Ingeniería Informática. Curso 2013-2014

- David Frías Garrido. Desarrollo de una aplicación móvil en Android para cartografiar y procesar rutas en formato GPX. Universidad Politécnica de Valencia, Trabajo Fin de Carrera Ingeniería Superior Geodésica, Cartográfica y Topográfica.

Anexo 1.- Código de la aplicación Agrisoft

1.- Fichero AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.josemanuel.agrisoftv1">

  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
  <uses-feature android:name="android.hardware.camera" android:required="true" />

  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.NoActionBar">
    <activity android:name=".Base_DatosActivity" />
    <activity android:name=".Visita" />
    <activity android:name=".tratamiento" />
    <activity android:name=".Siembra" />
    <activity android:name=".Cosecha" />
    <activity android:name=".Guiado" />
    <activity android:name=".Cultivar" />

    <provider
      android:name="android.support.v4.content.FileProvider"
      android:authorities="${applicationId}.share"
      android:exported="false"
      android:grantUriPermissions="true">
      <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/provider_path"/>
    </provider>

    <activity android:name=".Menu_inicio_Activity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".activity_acercade"></activity>
  </application>
</manifest>
```

2.- Fichero Menu_inicio_Activity.java

```
package com.example.josemanuel.agrisoftv1;

import android.app.Activity;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class Menu_inicio_Activity extends AppCompatActivity {

   //Declaración de variables botones

   private Button bd, salir, gui, acerca;


   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_menu_inicio_);


      // La variable button bd es igual al botón btn_bd
      bd = (Button) findViewById(R.id.btn_bd);
      bd.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            //bd.setText(String.format("Botón presionado: "));
            Intent ir = new Intent(Menu_inicio_Activity.this, Base_DatosActivity.class); //Ir a Base
de datos si pulso este botón
            startActivity(ir);
         }
      });


      // Variable del botón SALIR de la Base de datos
      salir = (Button) findViewById(R.id.btn_s);
      salir.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Intent intent = new Intent(Intent.ACTION_MAIN);
            intent.addCategory(Intent.CATEGORY_HOME);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
         }
      });
```

```
        // La variable button gui es igual al botón btn_g
        gui = (Button) findViewById(R.id.btn_g);
        gui.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //gui.setText(String.format("Botón presionado:"));
                Intent ir = new Intent(Menu_inicio_Activity.this, Guiado.class); //Ir a Guiado
                startActivity(ir);
            }
        });

        // La variable button acerca es igual al botón btn_acerca
        acerca = (Button) findViewById(R.id.btn_acerca);
        acerca.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //bd.setText(String.format("Botón presionado: "));
                Intent ir = new Intent(Menu_inicio_Activity.this,activity_acercade.class); //Ir a Base
de datos si pulso este botón
                startActivity(ir);
            }
        });

    } //Fin onCreate

} //Fin Main
```

3.- Fichero Base_DatosActivity.java

```
package com.example.josemanuel.agrisoftv1;

import android.content.Context;
import android.content.Intent;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.format.Time;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.util.Date;


public class Base_DatosActivity extends AppCompatActivity {
```

```
  //Declaración de las variables botones
  private Button visita, tratamiento, siembra, cultivar, cosecha;

  //Declaración de las variables EditText
   public TextView tv_fecha;
   public String fecha;


  @Override
  public void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_base__datos);

     tv_fecha=(TextView) findViewById(R.id.tv_fecha);

     Time today=new Time(Time.getCurrentTimezone());
     today.setToNow();
     int anio= today.year;
     int mes= today.month;
     int dia= today.monthDay;
     mes=mes+1;
     tv_fecha.setText(anio+"/"+mes+"/"+dia);

     // Variable del botón VISITA

     visita = (Button) findViewById(R.id.btn_visita);
     visita.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
           //visita.setText(String.format("Botón presionado: "));
           Intent intent = new Intent(Base_DatosActivity.this, Visita.class); //Ir a layout Visita si
pulso este botón
           intent.putExtra("tv_fecha",1);
           startActivity(intent);
        }
     });

     // Variable del botón TRATAMIENTO
     tratamiento = (Button) findViewById(R.id.btn_tratamiento);
     tratamiento.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

           Intent intent = new Intent(Base_DatosActivity.this, tratamiento.class); //Ir a layout
Tratamiento si pulso este botón
           intent.putExtra("tv_fecha",1);
           startActivity(intent);
        }
     });
```

```
        // Variable del botón SIEMBRA
        siembra = (Button) findViewById(R.id.btn_siembra);
        siembra.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //siembra.setText(String.format("Botón presionado: "));
                Intent intent = new Intent(Base_DatosActivity.this, Siembra.class); //Ir a layout
Siembra si pulso este botón
                intent.putExtra("tv_fecha",1);
                startActivity(intent);
            }
        });

        // Variable del botón CULTIVAR
        cultivar = (Button) findViewById(R.id.btn_cultivar);
        cultivar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //cultivar.setText(String.format("Botón presionado: "));
                Intent intent = new Intent(Base_DatosActivity.this, Cultivar.class); //Ir a layout
Cultivar si pulso este botón
                intent.putExtra("tv_fecha",1);
                startActivity(intent);
            }
        });

        // Variable del botón COSECHA
        cosecha = (Button) findViewById(R.id.btn_cosecha);
        cosecha.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Base_DatosActivity.this, Cosecha.class); //Ir a layout
Cosecha si pulso este botón
                intent.putExtra("tv_fecha",1);
                startActivity(intent);
            }
        });


                    } //Fin onCreate

    //Código para volver al menú inicio
    public void onClickMenuInicio(View view){
        Intent i= new Intent(this, Menu_inicio_Activity.class);
         startActivity(i);
    }


        } //Fin Base_DatosActivity
```

4.- Fichero activity_acercade.java

```
package com.example.josemanuel.agrisoftv1;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class activity_acercade extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_acercade);
    }

    //Código para volver al menú inicio
    public void onClickMenuInicio(View view){
        Intent i= new Intent(this, Menu_inicio_Activity.class);
        startActivity(i);
    }
}
```

5.- Fichero Guiado.java

```
package com.example.josemanuel.agrisoftv1;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.GpsStatus;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.res.ResourcesCompat;
import android.support.v7.app.AppCompatActivity;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import static android.location.LocationManager.GPS_PROVIDER;
```

```
public class Guiado extends AppCompatActivity implements View.OnTouchListener{

    ArrayList <Double> distanciaorigen=new ArrayList<Double>();
    ArrayList <Integer> paralelas= new ArrayList<Integer>();
    ArrayList <Integer> paralelasnegativas= new ArrayList<Integer>();
    private Button btn_final, btn_inicio;
    public EditText et_ancho;

    //Variables para transformar coordenadas geográficas del GPS a UTM
    public String elipsoide = "WGS84";
    public double grados_longitud = 0;
    public double minutos_longitud = 0;
    public double segundos_longitud = 0;
    public String E_W = "W";
    public double grados_latitud = 0;
    public double minutos_latitud = 0;
    public double segundos_latitud = 0;
    public String N_S = "N";
    public double x_UTM = 0;
    public double y_UTM = 0;
    public double huso;
    public double a_semieje_mayor = 6378137;
    public double b_semieje_menor = 6356752.314245;
    public double excentricidad = Math.sqrt(a_semieje_mayor * a_semieje_mayor -
b_semieje_menor * b_semieje_menor) / a_semieje_mayor;
    public double excentricidad_2 = Math.sqrt(a_semieje_mayor * a_semieje_mayor -
b_semieje_menor * b_semieje_menor) / b_semieje_menor;
    public double cuadrado_2_excen = excentricidad_2 * excentricidad_2;
    public double c_radio_polar_curvatura = (a_semieje_mayor * a_semieje_mayor) /
b_semieje_menor;
    public double radianes_longitud = 0;
    public double radianes_latitud = 0;
    public double meridiano_huso;
    public double delta;
    public double A;
    public double Xi;
    public double Eta;
    public double Ni;
    public double Zeta;
    public double A1;
    public double A2;
    public double J2;
    public double J4;
    public double J6;
    public double alfa, beta, gamma;
    public double B_latitud;


    //public boolean estaPresionado = false;
    public double x_inicio, y_inicio, x_final, y_final;
```

```
   LocationManager locationManager;
   public TextView tv_xi, tv_yi, tv_xf, tv_yf, tv_despla, tv_movi, tv_xutm, tv_yutm, tv_Da,
tv_velocidad;
   public TextView tv_precision;
   Location location;
   LocationListener locationListener;

   //Variables del cálculo de la distancia a la traza
    public double ix=0;
    public double iy=0;
    public double Dif=0;
    public double azal=0;
    public double Dip=0;
    public double iix=0;
    public double iiy=0;
    public double azap=0;
    public double ALF=0;
    public double Da=0;
    public double despla=0;
    public float ancho=0;
    public double Do=0;
    public int i=1;
    public int m=0;
    public int n=0;
    public float para;
    public float paranegativas;
    public double iDo;
    public double v;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_guiado);

      btn_inicio=(Button)findViewById(R.id.btn_inicio);
      btn_final=(Button)findViewById(R.id.btn_final);
      tv_xi = (TextView) findViewById(R.id.tv_xi);
      tv_movi=(TextView) findViewById(R.id.tv_movi);
      tv_yi = (TextView) findViewById(R.id.tv_yi);
      tv_despla = (TextView) findViewById(R.id.tv_despla);
      tv_xf= (TextView) findViewById(R.id.tv_xf);
      tv_yf= (TextView) findViewById(R.id.tv_yf);
      tv_xutm= (TextView) findViewById(R.id.tv_xutm);
      tv_yutm= (TextView) findViewById(R.id.tv_yutm);
      tv_precision = (TextView) findViewById(R.id.tv_precision);
      et_ancho = (EditText) findViewById(R.id.et_ancho);
      tv_Da = (TextView) findViewById(R.id.tv_Da);
      tv_velocidad=(TextView)findViewById(R.id.tv_velocidad);
      locationManager= (LocationManager) getSystemService(Context.LOCATION_SERVICE);

      btn_inicio.setOnTouchListener(this);
```

```
    btn_final.setOnTouchListener(this);

    if (tengoPermisoEscritura()) {
      // Detectar cambios de ubicación

       if (tengoPermisoUbicacion()) {
        Location ultimaPosicionConocida = null;
        for (String provider : locationManager.getProviders(true)) {
          if (Build.VERSION.SDK_INT >= 23 &&
checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED)
            ultimaPosicionConocida =
locationManager.getLastKnownLocation(GPS_PROVIDER);
          if (ultimaPosicionConocida != null) {
            actualizaPosicionActual(ultimaPosicionConocida);
          }
          break;
        }
      } else {
        // No tengo permiso de ubicación
      }
    }


    locationListener=new LocationListener() {
      @Override
      public void onLocationChanged(Location location) {
        actualizaPosicionActual(location);
      }

      @Override
      public void onStatusChanged(String provider, int status, Bundle bundle) {

      }

      @Override
      public void onProviderEnabled(String s) {

      }

      @Override
      public void onProviderDisabled(String s) {

      }
    };

    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
locationListener);
```

```
    } //Fin onCreate


    public void actualizaPosicionActual(Location location) {

        //Para transformar la anchura del edittext en número
        double ancho = Double.parseDouble(et_ancho.getText().toString().trim());
        //Solo si ancho es mayor de cero empieza a funcionar
        if (ancho != 0) {

            //Precisión horizontal del GPS en metros
            tv_precision.setText(String.valueOf(location.getAccuracy()));
            //Solo si la precisión horizontal es menor de 10 se muestran datos de posición en
pantalla
            if (location.getAccuracy() < 10) {

                //Transformación de coordenadas geográficas a UTM en el huso 30
                //Convierto a radianes, también podría multiplicar por PI y dividir por 180
                radianes_longitud = (Math.PI * (location.getLongitude())) / 180;
                radianes_latitud = (Math.PI * (location.getLatitude())) / 180;
                //Cálculo del huso con floor que calcula el entero inferior al calculado
                huso = Math.floor((location.getLongitude() / 6.0) + 31);

                meridiano_huso = 6 * huso - 183;
                delta = radianes_longitud - ((meridiano_huso * Math.PI) / 180.0);
                A = Math.cos(radianes_latitud) * Math.sin(delta);
                Xi = (1.0 / 2.0) * Math.log((1 + A) / (1 - A));

                Eta = Math.atan((Math.tan(radianes_latitud)) / Math.cos(delta)) - radianes_latitud;
                Ni = (c_radio_polar_curvatura / Math.pow(1.0 + cuadrado_2_excen *
Math.pow(Math.acos(radianes_latitud), 2), (1.0 / 2.0))) * 0.9996;
                Zeta = (cuadrado_2_excen / 2.0) * Math.pow(Xi, 2) *
Math.pow((Math.cos(radianes_latitud)), 2);

                //Cálculo de la coordenada Xutm
                x_UTM = Xi * Ni * (1 + Zeta / 3.0) + 500000;
                //Cálculo de la coordenada Yutm
                A1 = Math.sin(2 * radianes_latitud);
                A2 = A1 * Math.pow(Math.cos(radianes_latitud), 2);
                J2 = radianes_latitud + (A1 / 2.0);
                J4 = ((3 * J2) + A2) / 4.0;
                J6 = (5 * J4 + A2 * Math.pow(Math.cos(radianes_latitud), 2)) / 3.0;
                alfa = (3.0 / 4.0) * cuadrado_2_excen;
                beta = (5.0 / 3.0) * Math.pow(alfa, 2);
                gamma = (35.0 / 27.0) * Math.pow(alfa, 3);
                B_latitud = 0.9996 * c_radio_polar_curvatura * (radianes_latitud - (alfa * J2) + (beta
* J4) - (gamma * J6));
                y_UTM = Eta * Ni * (1 + Zeta) + B_latitud;
                if (N_S == "S") {
                    y_UTM = y_UTM + 10000000;
                }
```

```
//Muestro la velocidad de desplazamiento de la máquina
v=(3600*location.getSpeed())/1000;
tv_velocidad.setText(String.valueOf(v));

// Captura del punto inicio de la traza de referencia
btn_inicio.setOnTouchListener(new View.OnTouchListener() {
   @Override
   public boolean onTouch(View v, MotionEvent event) {
      if (v.getId() == R.id.btn_inicio)
         //btn_inicio.setText(String.format("Punto seleccionado"));
         switch (event.getAction()) {
            case (MotionEvent.ACTION_DOWN):
               v.performClick();
               x_inicio = x_UTM;
               y_inicio = y_UTM;
         }
      return true;
   }
});

//Captura del punto final de la traza de referencia
btn_final.setOnTouchListener(new View.OnTouchListener() {
   @Override
   public boolean onTouch(View v, MotionEvent event) {
      if (v.getId() == R.id.btn_final)
         //btn_final.setText(String.format("Punto seleccionado"));
         switch (event.getAction()) {
            case (MotionEvent.ACTION_DOWN):
               v.performClick();
               x_final = x_UTM;
               y_final = y_UTM;
         }
      return true;
   }
});

 //Para mostrar en pantalla las coordenadas de la traza de referencia y de la posición
 actual de la máquina
tv_xi.setText(String.valueOf(x_inicio));
tv_yi.setText(String.valueOf(y_inicio));
tv_xf.setText(String.valueOf(x_final));
tv_yf.setText(String.valueOf(y_final));
tv_xutm.setText(String.valueOf(x_UTM));
tv_yutm.setText(String.valueOf(y_UTM));

//Si la posición inicial y final no es nula
if (x_inicio != 0 && x_final != 0) {

   //Código para cálculo distancia a la traza de referencia
   //Cálculo datos traza de referencia
   ix = x_final - x_inicio;
```

```
iy = y_final - y_inicio;
//Cálculo azimut de la traza de referencia
if (ix > 0 && iy > 0)
   azal = Math.atan(ix / iy);
if (ix > 0 && iy <= 0)
   azal = (Math.PI / 2) + Math.atan(iy / ix);
if (ix < 0 && iy < 0)
   azal = Math.PI + Math.atan(ix / iy);
if (ix < 0 && iy > 0)
   azal = (Math.PI * 1.5) + Math.atan(iy / ix);
Dif = Math.sqrt(Math.pow(ix, 2) + Math.pow(iy, 2));

//Cálculo datos entre traza y el punto a calcular
//while (i <= 200) {
iix = x_UTM - x_inicio;
iiy = y_UTM - y_inicio;
//Cálculo azimut desde inicio traza al punto
if (iix > 0 && iiy > 0)
   azap = Math.atan(iix / iiy);
if (iix > 0 && iiy <= 0)
   azap = (Math.PI / 2) + Math.atan(iiy / iix);
if (iix < 0 && iiy < 0)
   azap = Math.PI + Math.atan(iix / iiy);
if (iix < 0 && iiy > 0)
   azap = (Math.PI * 1.5) + Math.atan(iiy / iix);
Dip = Math.sqrt(Math.pow(iix, 2) + Math.pow(iiy, 2));

//Cálculo distancia a la traza y al punto inicio del punto a calcular
ALF = -azap+azal;
Da = Math.sin(ALF) * Dip;
Do = Math.cos(ALF) * Dip;
//Relleno el ArrayList con las distancias al origen
distanciaorigen.add(Math.abs(Do));

//Relleno el ArrayList con las traza paralelas a la de referencia
for(n=0;n<300;n++){
   para= (float) (n*ancho);
   paranegativas=(float)(-1*n*ancho);
   paralelasnegativas.add((int) paranegativas);
   paralelas.add((int) para);
} //Fin for


//Saco por pantalla en un textview el valor de la distancia a la traza de referencia
tv_Da.setText(String.valueOf(Da));

   //Cálculo de los incrementos de distancia al inicio de la traza origen
   for (int m = 1; m < distanciaorigen.size(); m++) {

      iDo = distanciaorigen.get(m) - distanciaorigen.get(m - 1);
      //Aumento el valor de n para pasar a otra traza paralela a la inicial
```

```
n++;
//En función del lado de la traza de referencia en el que estoy
//Si la traza paralela está a la derecha Da>0 de la traza origen
if(Da>=0) {


  //Cálculo de la traza paralela más cercana a la que está la máquina
  //según el valor de Da

  if(Da - (ancho*0.5)< paralelas.get(n)&&paralelas.get(n)<Da+(ancho*0.5)) {


    //Cálculo del desplazamiento a aplicar a la máquina
    despla = paralelas.get(n) - Da;
    //Salida por pantalla del desplazamiento a aplicar a la máquina en metros
    tv_despla.setText(String.valueOf(Math.abs(despla)));
     //Salida por pantalla de las flechas que indican el sentido del
     desplazamiento a relizar
                        if (iDo >= 0) {
                           if (despla >= 0) {
                              tv_movi.setText(String.valueOf("<<<"));
                           } else {
                              tv_movi.setText(String.valueOf(">>>"));
                           } //Fin if relativo al valor de despla si iDo>0
                        } else {

      //Salida por pantalla del desplazamiento a aplicar a la máquina con
    flechas
                           if (despla <= 0) {
                              tv_movi.setText(String.valueOf(">>>"));
                           } else {
                              tv_movi.setText(String.valueOf("<<<"));
                           } //Fin if relativo al valor de despla

                        } //Fin if relativo al valor de iDo>0

                              } //Fin if para el salto a otra traza

                     }else{
                        //Cálculo de la traza paralela más cercana a la que está
    la máquina
                        //según el valor de Da

                        if (Da + (ancho * 0.5) < (paralelasnegativas.get(n)) &&
    (paralelasnegativas.get(n)) < Da - (ancho * 0.5)) {

                           despla = (paralelasnegativas.get(n)) - Da;

                           //Salida por pantalla del desplazamiento a aplicar a la
    máquina en metros
                           tv_despla.setText(String.valueOf(Math.abs(despla)));
```

```
                        //Salida por pantalla del desplazamiento a aplicar a la
máquina con flechas
                        if (iDo >= 0) {
                            if (despla <= 0) {
                                tv_movi.setText(String.valueOf("<<<"));
                            } else {
                                tv_movi.setText(String.valueOf(">>>"));
                            } //Fin if relativo al valor de despla si iDo>0
                        } else {

                            //Salida por pantalla del desplazamiento a aplicar a
la máquina con flechas
                            if (despla >= 0) {
                                tv_movi.setText(String.valueOf(">>>"));
                            } else {
                                tv_movi.setText(String.valueOf("<<<"));
                            } //Fin if relativo al valor de despla

                        } //Fin if relativo al valor de iDo<0

                    } //Fin if para el salto de traza

                } //Fin if para Da<0

                                        } //Fin for
                    } else {
                        Toast toast1 = Toast.makeText(getApplicationContext(),
"SELECCIONA INICIO Y FINAL DE LA TRAZA", Toast.LENGTH_SHORT);
                        toast1.show();

                    }//Fin if dependiente de los botones de selección de los puntos
que definen la traza

                } else {
                    Toast toast1 = Toast.makeText(getApplicationContext(), "SIN
SUFICIENTE PRECISIÓN", Toast.LENGTH_SHORT);
                    toast1.show();
                } //Fin if dependiente de la precisión

            } else {

                Toast toast1 = Toast.makeText(getApplicationContext(), "LA
ANCHURA DE TRABAJO DEBE SER MAYOR DE 0", Toast.LENGTH_SHORT);
                toast1.show();

            }//Fin if si el ancho es mayor que 0

        }//Fin actualizaPosicionActual

        @Override
        protected void onPause() {
```

```
        super.onPause();
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if
(checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED){
                ;
            return;
        } else {
            locationManager.removeUpdates(locationListener);
        }
    }else{
        locationManager.removeUpdates(locationListener);
    }

    } //Fin onPause

    @Override
    public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Intent intent = new Intent();
            intent.setClass(this, this.getClass());
            startActivity(intent);
            finish();
        } else {
            // El usuario no ha dado permiso
        }
    } //Fin onRequestPermissionsResult

        public boolean tengoPermisoEscritura() {
            if (Build.VERSION.SDK_INT >= 23) {
                if
(checkSelfPermission(android.Manifest.permission.WRITE_EXTERNAL_ST
ORAGE) == PackageManager.PERMISSION_GRANTED) {
                    return true;
                } else {
                    ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1);
                    return false;
                }
            } else {
                return true;
            }
        } //Fin tengoPermisoEscritura


        public boolean tengoPermisoUbicacion() {
            if (Build.VERSION.SDK_INT >= 23) {
                if
```

```
(checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
                return true;
            } else {
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 2);
                return false;
            }
        } else {
            return true;
        }
    } //Fin tengoPermisoUbicacion

    @Override
    public boolean onTouch(View v, MotionEvent event) {

        return false;
    } //Fin onTouch

    //Código para volver al menú inicio
    public void onClickMenuInicio(View view){
        Intent i= new Intent(this,Menu_inicio_Activity.class);
        distanciaorigen.clear();
        paralelas.clear();
        paralelasnegativas.clear();
        startActivity(i);
    } //Fin onClickMenuInicio

} //Fin Guiado
```

6.- Fichero Cosecha.java

```
package com.example.josemanuel.agrisoftv1;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Environment;
import android.os.StrictMode;
import android.provider.MediaStore;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v4.content.FileProvider;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.format.Time;
import android.util.Log;
```

```
import android.util.Xml;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.xmlpull.v1.XmlSerializer;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.net.URI;

public class Cosecha extends AppCompatActivity {

   //Declaración de variables globales
   public EditText bd_co_lote, bd_co_producto, bd_co_cantidad, bd_co_duracion,
bd_co_observaciones;
   //Variable para saber si la tarjeta SD esta disponible
   boolean sdDisponible = false;
   //Variable para acceder a la escritura en la tarjeta SD
   boolean sdAccesoEscritura = false;
   private Button btn_co_guardar, btn_co_volver;
   //Variables de la fecha
   public int anio, mes, dia;
   public String fecha;
   public TextView tv_fecha;
  //Para no sobreescribir el fichero creado
   BufferedWriter bw=null;
   FileWriter fw=null;


   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_cosecha);

      bd_co_lote = (EditText) findViewById(R.id.bd_co_lote);
      bd_co_producto = (EditText) findViewById(R.id.bd_co_producto);
      bd_co_cantidad = (EditText) findViewById(R.id.bd_co_cantidad);
      bd_co_duracion = (EditText) findViewById(R.id.bd_co_duracion);
      bd_co_observaciones = (EditText) findViewById(R.id.bd_co_observaciones);
      tv_fecha = (TextView) findViewById(R.id.tv_fecha);

      btn_co_guardar = (Button) findViewById(R.id.btn_co_guardar);
      btn_co_volver = (Button) findViewById(R.id.btn_co_volver);
```

```
//Para tomar la fecha del día
Time today=new Time(Time.getCurrentTimezone());
today.setToNow();
int anio= today.year;
int mes= today.month;
int dia= today.monthDay;
mes=mes+1;
tv_fecha.setText(anio+"/"+mes+"/"+dia);

//Código que comprueba si existe tarjeta SD y si puedo escribir o no
String estado = Environment.getExternalStorageState();
if (estado.equals(Environment.MEDIA_MOUNTED)) {
   sdDisponible = true;
   sdAccesoEscritura = true;
} else if (estado.equals(Environment.MEDIA_MOUNTED_READ_ONLY)) {
   sdDisponible = true;
   sdAccesoEscritura = false;
} else {
   sdDisponible = false;
   sdAccesoEscritura = false;
}

//Botón guardar datos en un archivo xml
btn_co_guardar.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      //Para pasar a cadena de testo los datos introducidos por teclado
      String lote=bd_co_lote.getText().toString();
      String co1 = bd_co_producto.getText().toString();
      String co2 = bd_co_cantidad.getText().toString();
      String co3 = bd_co_duracion.getText().toString();
      String co4 = bd_co_observaciones.getText().toString();
      String fecha=tv_fecha.getText().toString();

      if (sdAccesoEscritura && sdDisponible) {

         try {
            XmlSerializer ser= Xml.newSerializer();
            File tarjeta = Environment.getExternalStorageDirectory();
            File fiche = new File(tarjeta.getAbsoluteFile(), "Cosecha.xml");
            OutputStreamWriter fout = new OutputStreamWriter(new
FileOutputStream(fiche));
            //Se crea la estructura del archivo xml
            ser.setOutput(fout);
            ser.startTag("", "COSECHA");
            fout.append("\r\n");
            ser.startTag("", "FECHA");
            fout.append("\r\n");
            ser.text(fecha);
            fout.append("\r\n");
            ser.endTag("", "FECHA");
```

```
                fout.append("\r\n");
                ser.startTag("", "LOTE");
                fout.append("\r\n");
                ser.text(lote);
                fout.append("\r\n");
                ser.endTag("", "LOTE");
                fout.append("\r\n");
                ser.startTag("", "PRODUCTO");
                fout.append("\r\n");
                ser.text(co1);
                fout.append("\r\n");
                ser.endTag("", "PRODUCTO");
                fout.append("\r\n");
                ser.startTag("", "CANTIDAD");
                fout.append("\r\n");
                ser.text(co2);
                fout.append("\r\n");
                ser.endTag("", "CANTIDAD");
                fout.append("\r\n");
                ser.startTag("", "DURACION");
                fout.append("\r\n");
                ser.text(co3);
                fout.append("\r\n");
                ser.endTag("", "DURACION");
                fout.append("\r\n");
                ser.startTag("", "OBSERVACIONES");
                fout.append("\r\n");
                ser.text(co4);
                fout.append("\r\n");
                ser.endTag("", "OBSERVACIONES");
                fout.append("\r\n");
                ser.endTag("", "COSECHA");
                fout.append("\r\n");
                ser.endDocument();
                fout.close();

                Log.e("Ficheros", "Los datos fueron grabados correctamente");
                //Se borrar los dartos introducidos en los campos
                bd_co_lote.setText("");
                bd_co_producto.setText("");
                bd_co_cantidad.setText("");
                bd_co_duracion.setText("");
                bd_co_observaciones.setText("");


            }catch (IOException ioe) {
                Log.e("Ficheros", "Error al escribir fichero a tarjeta SD");
            }
        } //Fin if
    }
}); //Fin botón guardar
```

```
    } //Fin onCreate


    //Código para volver al menú inicio
    public void onClickBaseDatos(View view){
        Intent i= new Intent(this, Base_DatosActivity.class);
        startActivity(i);
    }

} //Fin cosecha
```

7.- Cultivar.java

```
package com.example.josemanuel.agrisoftv1;

import android.content.Intent;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.format.Time;
import android.util.Log;
import android.util.Xml;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.xmlpull.v1.XmlSerializer;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;

public class Cultivar extends AppCompatActivity {

    public EditText bd_cu_labor, bd_cu_duracion, bd_cu_observaciones, bd_cu_lote;
    boolean sdDisponible = false;
    //Variable para acceder a la escritura en la tarjeta SD
    boolean sdAccesoEscritura = false;
    public EditText trata_lote;
    public String fichero;
    public Button btn_cu_volver, btn_cu_guardar;
    //Variables de la fecha
    public int anio, mes, dia;
    public String fecha;
```

```
public TextView tv_fecha;
//Para no sobreescribir el fichero creado
BufferedWriter bw=null;
FileWriter fw=null;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_cultivar);

    bd_cu_lote = (EditText) findViewById(R.id.bd_cu_lote);
    bd_cu_labor = (EditText) findViewById(R.id.bd_cu_labor);
    bd_cu_duracion = (EditText) findViewById(R.id.bd_cu_duracion);
    bd_cu_observaciones = (EditText) findViewById(R.id.bd_cu_observaciones);
    tv_fecha = (TextView) findViewById(R.id.tv_fecha);

    btn_cu_guardar = (Button) findViewById(R.id.btn_cu_guardar);
    btn_cu_volver = (Button) findViewById(R.id.btn_cu_volver);

    //Para tomar la fecha del día
    Time today=new Time(Time.getCurrentTimezone());
    today.setToNow();
    int anio= today.year;
    int mes= today.month;
    int dia= today.monthDay;
    mes=mes+1;
    tv_fecha.setText(anio+"/"+mes+"/"+dia);

    //Código que comprueba si existe tarjeta SD y si puedo escribir o no
    String estado = Environment.getExternalStorageState();
    if (estado.equals(Environment.MEDIA_MOUNTED)) {
        sdDisponible = true;
        sdAccesoEscritura = true;
    } else if (estado.equals(Environment.MEDIA_MOUNTED_READ_ONLY)) {
        sdDisponible = true;
        sdAccesoEscritura = false;
    } else {
        sdDisponible = false;
        sdAccesoEscritura = false;
    }
    //Botón guardar datos en un archivo xml
    btn_cu_guardar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //Para pasar a cadena de testo los datos introducidos por teclado
            String lote = bd_cu_lote.getText().toString();
            String cu1 = bd_cu_labor.getText().toString();
            String cu2 = bd_cu_duracion.getText().toString();
            String cu3 = bd_cu_observaciones.getText().toString();
            String fecha=tv_fecha.getText().toString();
```

```
        //Si hay acceso de escritura y esta disponible la tarjeta
        if (sdAccesoEscritura && sdDisponible) {

            try {
                XmlSerializer ser= Xml.newSerializer();
                File tarjeta = Environment.getExternalStorageDirectory();
                File fiche = new File(tarjeta.getAbsoluteFile(), "Cultivar.xml");
                OutputStreamWriter fout = new OutputStreamWriter(new
FileOutputStream(fiche));
                //Se crea la estructura del archivo xml
                ser.setOutput(fout);
                ser.startTag("", "CULTIVAR");
                fout.append("\r\n");
                ser.startTag("", "FECHA");
                fout.append("\r\n");
                ser.text(fecha);
                fout.append("\r\n");
                ser.endTag("", "FECHA");
                fout.append("\r\n");
                ser.startTag("", "LOTE");
                fout.append("\r\n");
                ser.text(lote);
                fout.append("\r\n");
                ser.endTag("", "LOTE");
                fout.append("\r\n");
                ser.startTag("", "LABOR");
                fout.append("\r\n");
                ser.text(cu1);
                fout.append("\r\n");
                ser.endTag("", "LABOR");
                fout.append("\r\n");
                ser.startTag("", "DURACIÓN");
                fout.append("\r\n");
                ser.text(cu2);
                fout.append("\r\n");
                ser.endTag("", "DURACIÓN");
                fout.append("\r\n");
                ser.startTag("", "OBSERVACIONES");
                fout.append("\r\n");
                ser.text(cu3);
                fout.append("\r\n");
                ser.endTag("", "OBSERVACIONES");
                fout.append("\r\n");
                ser.endTag("", "CULTIVAR");
                fout.append("\r\n");
                ser.endDocument();
                fout.close();

                Log.e("Ficheros", "Los datos fueron grabados correctamente");

                //Se borrar los dartos introducidos en los campos
```

```
                bd_cu_lote.setText("");
                bd_cu_labor.setText("");
                bd_cu_duracion.setText("");
                bd_cu_observaciones.setText("");


            } catch (IOException ioe) {
                Log.e("Ficheros", "Error al escribir fichero a tarjeta SD");
            }
          } //Fin if


        }
    }); //Fin botón guardar


} //Fin onCreate


//Código para volver al menú BASE DATOS
public void onClickBaseDatos(View view) {
    Intent i = new Intent(this, Base_DatosActivity.class);
    startActivity(i);
}



} //Fin cultivar
```

8.- Siembra.java

```
package com.example.josemanuel.agrisoftv1;

import android.content.Intent;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.format.Time;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
```

```
import java.io.OutputStreamWriter;
import org.xmlpull.v1.XmlSerializer;
import android.util.Xml;


public class Siembra extends AppCompatActivity {

   public EditText bd_si_lote, bd_si_cultivo1, bd_si_cultivo2, bd_si_observaciones,
bd_si_duracion, bd_si_dosis1, bd_si_dosis2;
   boolean sdDisponible = false;
   //Variable para acceder a la escritura en la tarjeta SD
   boolean sdAccesoEscritura = false;
   public EditText trata_lote;
   public String fichero;
   public Button btn_si_volver, btn_si_guardar;
   //Variables de la fecha
   public int anio, mes, dia;
   public String fecha;
   public TextView tv_fecha;
   //Para no sobreescribir el fichero creado
   BufferedWriter bw=null;
   FileWriter fw=null;



   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_siembra);

      bd_si_cultivo1 = (EditText) findViewById(R.id.bd_si_cultivo1);
      bd_si_cultivo2 = (EditText) findViewById(R.id.bd_si_cultivo2);
      bd_si_dosis1 = (EditText) findViewById(R.id.bd_si_dosis1);
      bd_si_dosis2 = (EditText) findViewById(R.id.bd_si_dosis2);
      bd_si_duracion = (EditText) findViewById(R.id.bd_si_duracion);
      bd_si_observaciones = (EditText) findViewById(R.id.bd_si_observaciones);
      bd_si_lote= (EditText) findViewById(R.id.bd_si_lote);
      tv_fecha = (TextView) findViewById(R.id.tv_fecha);

      btn_si_guardar = (Button) findViewById(R.id.btn_si_guardar);
      btn_si_volver = (Button) findViewById(R.id.btn_si_volver);

      //Para tomar la fecha del día
      Time today=new Time(Time.getCurrentTimezone());
      today.setToNow();
      int anio= today.year;
      int mes= today.month;
      int dia= today.monthDay;
      mes=mes+1;
      tv_fecha.setText(anio+"/"+mes+"/"+dia);
```

```
//Código que comprueba si existe tarjeta SD y si puedo escribir o no
String estado = Environment.getExternalStorageState();
if (estado.equals(Environment.MEDIA_MOUNTED)) {
   sdDisponible = true;
   sdAccesoEscritura = true;
} else if (estado.equals(Environment.MEDIA_MOUNTED_READ_ONLY)) {
   sdDisponible = true;
   sdAccesoEscritura = false;
} else {
   sdDisponible = false;
   sdAccesoEscritura = false;
}

//Botón guardar
btn_si_guardar.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      String lote = bd_si_lote.getText().toString();
      String si1 = bd_si_cultivo1.getText().toString();
      String si2 = bd_si_cultivo2.getText().toString();
      String si3 = bd_si_dosis1.getText().toString();
      String si4 = bd_si_dosis2.getText().toString();
      String si5 = bd_si_duracion.getText().toString();
      String si6 = bd_si_observaciones.getText().toString();
      String fecha=tv_fecha.getText().toString();

      //Si está disponible la escritura en la tarjeta
      if (sdAccesoEscritura && sdDisponible) {


         try {
            XmlSerializer ser= Xml.newSerializer();
            File tarjeta = Environment.getExternalStorageDirectory();
            File fiche = new File(tarjeta.getAbsoluteFile(), "Siembra.xml");
            OutputStreamWriter fout = new OutputStreamWriter(new
FileOutputStream(fiche));
            ser.setOutput(fout);
            ser.startTag("", "SIEMBRA");
            fout.append("\r\n");
            ser.startTag("", "FECHA");
            fout.append("\r\n");
            ser.text(fecha);
            fout.append("\r\n");
            ser.endTag("", "FECHA");
            fout.append("\r\n");
            ser.startTag("", "LOTE");
            fout.append("\r\n");
            ser.text(lote);
            fout.append("\r\n");
            ser.endTag("", "LOTE");
```

```
            fout.append("\r\n");
            ser.startTag("", "CULTIVO1");
            fout.append("\r\n");
            ser.text(si1);
            fout.append("\r\n");
            ser.endTag("", "CULTIVO1");
            fout.append("\r\n");
            ser.startTag("", "DOSIS1");
            fout.append("\r\n");
            ser.text(si3);
            fout.append("\r\n");
            ser.endTag("", "DOSIS1");
            fout.append("\r\n");
            ser.startTag("", "CULTIVO2");
            fout.append("\r\n");
            ser.text(si2);
            fout.append("\r\n");
            ser.endTag("", "CULTIVO2");
            fout.append("\r\n");
            ser.startTag("", "DOSIS2");
            fout.append("\r\n");
            ser.text(si4);
            fout.append("\r\n");
            ser.endTag("", "DOSIS2");
            fout.append("\r\n");
            ser.startTag("", "OBSERVACIONES");
            fout.append("\r\n");
            ser.text(si6);
            fout.append("\r\n");
            ser.endTag("", "OBSERVACIONES");
            fout.append("\r\n");
            ser.endTag("", "SIEMBRA");
            fout.append("\r\n");
            ser.endDocument();
            fout.close();

            Log.e("Ficheros", "Los datos fueron grabados correctamente");

            //Se borrar los dartos introducidos en los campos
            bd_si_lote.setText("");
            bd_si_cultivo1.setText("");
            bd_si_cultivo2.setText("");
            bd_si_dosis1.setText("");
            bd_si_dosis2.setText("");
            bd_si_duracion.setText("");
            bd_si_observaciones.setText("");

            } catch (IOException ioe) {
            Log.e("Ficheros", "Error al escribir fichero a tarjeta SD");
        }
    } //Fin if
```

```
        }
    });

} //Fin onCreate

//Código para volver al menú inicio
public void onClickBaseDatos(View view){
    Intent i= new Intent(this, Base_DatosActivity.class);
    startActivity(i);
}


} //Fin Siembra
```

9.- Tratamiento.java

```
package com.example.josemanuel.agrisoftv1;

import android.content.Intent;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.format.Time;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import android.util.Xml;
import org.xmlpull.v1.XmlSerializer;

import static android.os.Environment.getExternalStorageDirectory;


public class tratamiento extends AppCompatActivity {

    //Declaración de las variables públicas
    public EditText bd_trata1, bd_dosis1, bd_trata2, bd_dosis2, bd_trata3, bd_dosis3,
```

```
bd_trata4, bd_dosis4, bd_trata_duracion, bd_trata_observaciones;
    //Variable para saber si la tarjeta SD esta disponible
    boolean sdDisponible = false;
    //Variable para acceder a la escritura en la tarjeta SD
    boolean sdAccesoEscritura = false;
    public EditText trata_lote;
    //Declaración de las variables botones
    private Button btn_guardar, btn_volver;
    //Variables de la fecha
    public int anio, mes, dia;
    public String fecha;
    public TextView tv_fecha;
    //Para no sobreescribir el fichero creado
    BufferedWriter bw=null;
    FileWriter fw=null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tratamiento);

        trata_lote=(EditText) findViewById(R.id.trata_lote);
        bd_trata1 = (EditText) findViewById(R.id.bd_trata1);
        bd_trata2 = (EditText) findViewById(R.id.bd_trata2);
        bd_trata3 = (EditText) findViewById(R.id.bd_trata3);
        bd_trata4 = (EditText) findViewById(R.id.bd_trata4);
        bd_dosis1 = (EditText) findViewById(R.id.bd_dosis1);
        bd_dosis2 = (EditText) findViewById(R.id.bd_dosis2);
        bd_dosis3 = (EditText) findViewById(R.id.bd_dosis3);
        bd_dosis4 = (EditText) findViewById(R.id.bd_dosis4);
        bd_trata_duracion = (EditText) findViewById(R.id.bd_trata_duracion);
        bd_trata_observaciones = (EditText) findViewById(R.id.bd_tra_observaciones);
        tv_fecha = (TextView) findViewById(R.id.tv_fecha);

        btn_guardar = (Button) findViewById(R.id.btn_guardar);
        btn_volver = (Button) findViewById(R.id.btn_volver);

        //Para tomar la fecha del día
        Time today=new Time(Time.getCurrentTimezone());
        today.setToNow();
        int anio= today.year;
        int mes= today.month;
        int dia= today.monthDay;
        mes=mes+1;
        tv_fecha.setText(anio+"/"+mes+"/"+dia);

        //Código que comprueba si existe tarjeta SD y si puedo escribir o no
        String estado = Environment.getExternalStorageState();
        if (estado.equals(Environment.MEDIA_MOUNTED)) {
            sdDisponible = true;
            sdAccesoEscritura = true;
```

```
    } else if (estado.equals(Environment.MEDIA_MOUNTED_READ_ONLY)) {
      sdDisponible = true;
      sdAccesoEscritura = false;
    } else {
      sdDisponible = false;
      sdAccesoEscritura = false;
    }

    //Para crear fichero de datos y guardar
    btn_guardar.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {

        //Para pasar a cadena de texto los datos introducidos por teclado
        String lote = trata_lote.getText().toString();
        String trata1 = bd_trata1.getText().toString();
        String trata2 = bd_trata2.getText().toString();
        String trata3 = bd_trata3.getText().toString();
        String trata4 = bd_trata4.getText().toString();
        String dosis1 = bd_dosis1.getText().toString();
        String dosis2 = bd_dosis2.getText().toString();
        String dosis3 = bd_dosis3.getText().toString();
        String dosis4 = bd_dosis4.getText().toString();
        String trata_duracion = bd_trata_duracion.getText().toString();
        String trata_observaciones = bd_trata_observaciones.getText().toString();
        String fecha=tv_fecha.getText().toString();

        //Si hay permiso para escribir en tarjeta
        if (sdAccesoEscritura && sdDisponible) {


          try {
            XmlSerializer ser= Xml.newSerializer();
            File tarjeta = Environment.getExternalStorageDirectory();
            File fiche = new File(tarjeta.getAbsoluteFile(), "Tratamiento.xml");
            OutputStreamWriter fout = new OutputStreamWriter(new
FileOutputStream(fiche));
            //Se crea la estructura del archivo xml
            ser.setOutput(fout);
            ser.startTag("", "TRATAMIENTO");
            fout.append("\r\n");
            ser.startTag("", "FECHA");
            fout.append("\r\n");
            ser.text(fecha);
            fout.append("\r\n");
            ser.endTag("", "FECHA");
            fout.append("\r\n");
            ser.startTag("", "LOTE");
            fout.append("\r\n");
            ser.text(lote);
            fout.append("\r\n");
```

```
ser.endTag("", "LOTE");
fout.append("\r\n");
ser.startTag("", "TRATAMIENTO1");
fout.append("\r\n");
ser.text(trata1);
fout.append("\r\n");
ser.endTag("", "TRATAMIENTO1");
fout.append("\r\n");
ser.startTag("", "DOSIS1");
fout.append("\r\n");
ser.text(dosis1);
fout.append("\r\n");
ser.endTag("", "DOSIS1");
fout.append("\r\n");
ser.startTag("", "TRATAMIENTO2");
fout.append("\r\n");
ser.text(trata2);
fout.append("\r\n");
ser.endTag("", "TRATAMIENTO2");
fout.append("\r\n");
ser.startTag("", "DOSIS2");
fout.append("\r\n");
ser.text(dosis2);
fout.append("\r\n");
ser.endTag("", "DOSIS2");
fout.append("\r\n");
ser.startTag("", "TRATAMIENTO3");
fout.append("\r\n");
ser.text(trata3);
fout.append("\r\n");
ser.endTag("", "TRATAMIENTO3");
fout.append("\r\n");
ser.startTag("", "DOSIS3");
fout.append("\r\n");
ser.text(dosis3);
fout.append("\r\n");
ser.endTag("", "DOSIS3");
fout.append("\r\n");
ser.startTag("", "TRATAMIENTO4");
fout.append("\r\n");
ser.text(trata4);
fout.append("\r\n");
ser.endTag("", "TRATAMIENTO4");
fout.append("\r\n");
ser.startTag("", "DOSIS4");
fout.append("\r\n");
ser.text(dosis4);
fout.append("\r\n");
ser.endTag("", "DOSIS4");
fout.append("\r\n");
ser.startTag("", "DURACION");
```

```
                fout.append("\r\n");
                ser.text(trata_duracion);
                fout.append("\r\n");
                ser.endTag("", "DURACION");
                fout.append("\r\n");
                ser.startTag("", "OBSERVACIONES");
                fout.append("\r\n");
                ser.text(trata_observaciones);
                fout.append("\r\n");
                ser.endTag("", "OBSERVACIONES");
                fout.append("\r\n");
                ser.endTag("", "TRATAMIENTO");
                fout.append("\r\n");
                ser.endDocument();
                fout.close();


                Log.e("Ficheros", "Los datos fueron grabados correctamente");

                //Se borrar los dartos introducidos en los campos
                trata_lote.setText("");
                bd_trata1.setText("");
                bd_trata2.setText("");
                bd_trata3.setText("");
                bd_trata4.setText("");
                bd_dosis1.setText("");
                bd_dosis2.setText("");
                bd_dosis3.setText("");
                bd_dosis4.setText("");
                bd_trata_duracion.setText("");
                bd_trata_observaciones.setText("");

            } catch (IOException ioe) {
                Log.e("Ficheros", "Error al escribir fichero a tarjeta SD");
            }
          } //Fin if


      }
    }); //Fin botón guardar datos


} //Fin onCreate


//Volver al menú Base de Datos
public void onClickVolver(View view) {
    Intent i = new Intent(this, Base_DatosActivity.class);
    startActivity(i);
}
```

```
} //fin tratamiento
```

10.- Visita.java

```
package com.example.josemanuel.agrisoftv1;

import android.content.Intent;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.format.Time;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import android.util.Xml;
import org.xmlpull.v1.XmlSerializer;

public class Visita extends AppCompatActivity {

    public EditText bd_vi_observaciones, bd_vi_lote;
    boolean sdDisponible = false;
    //Variable para acceder a la escritura en la tarjeta SD
    boolean sdAccesoEscritura = false;
    public EditText trata_lote;
    public String fichero;
    public Button btn_vi_guardar, btn_vi_volver;
    //Variables de la fecha
    public int anio, mes, dia;
    public String fecha;
    public TextView tv_fecha;
    //Para no sobreescribir el fichero creado
    BufferedWriter bw=null;
    FileWriter fw=null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_visita);

        bd_vi_observaciones = (EditText) findViewById(R.id.bd_vi_observaciones);
```

```
        bd_vi_lote = (EditText) findViewById(R.id.bd_vi_lote);
        tv_fecha = (TextView) findViewById(R.id.tv_fecha);

        btn_vi_guardar = (Button) findViewById(R.id.btn_vi_guardar);
        btn_vi_volver = (Button) findViewById(R.id.btn_vi_volver);

        //Para tomar la fecha del día
        Time today=new Time(Time.getCurrentTimezone());
        today.setToNow();
        int anio= today.year;
        int mes= today.month;
        int dia= today.monthDay;
        mes=mes+1;
        tv_fecha.setText(anio+"/"+mes+"/"+dia);

        //Código que comprueba si existe tarjeta SD y si puedo escribir o no
        String estado = Environment.getExternalStorageState();
        if (estado.equals(Environment.MEDIA_MOUNTED)) {
            sdDisponible = true;
            sdAccesoEscritura = true;
        } else if (estado.equals(Environment.MEDIA_MOUNTED_READ_ONLY)) {
            sdDisponible = true;
            sdAccesoEscritura = false;
        } else {
            sdDisponible = false;
            sdAccesoEscritura = false;
        }

        //Abrir botón guardar
        btn_vi_guardar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                //Para pasar a cadena de texto los datos introducidos por teclado
                String lote = bd_vi_lote.getText().toString();
                String vi1 = bd_vi_observaciones.getText().toString();
                String fecha=tv_fecha.getText().toString();

                //Si hay permiso para escribir en tarjeta
                if (sdAccesoEscritura && sdDisponible) {

                    try {
                        XmlSerializer ser= Xml.newSerializer();
                        File tarjeta = Environment.getExternalStorageDirectory();
                        File fiche = new File(tarjeta.getAbsoluteFile(), "Visita.xml");
                        OutputStreamWriter fout = new OutputStreamWriter(new
FileOutputStream(fiche));

                        //Se crea la estructura del archivo xml
                        ser.setOutput(fout);
                        ser.startTag("", "VISITA");
```

```
                fout.append("\r\n");
                ser.startTag("", "FECHA");
                fout.append("\r\n");
                ser.text(fecha);
                fout.append("\r\n");
                ser.endTag("", "FECHA");
                ser.startTag("", "LOTE");
                fout.append("\r\n");
                ser.text(lote);
                fout.append("\r\n");
                ser.endTag("", "LOTE");
                fout.append("\r\n");
                ser.startTag("", "COMENTARIO");
                fout.append("\r\n");
                ser.text(vi1);
                fout.append("\r\n");
                ser.endTag("", "COMENTARIO");
                fout.append("\r\n");
                ser.endTag("", "VISITA");
                fout.append("\r\n");
                ser.endDocument();
                fout.close();

                Log.e("Ficheros", "Los datos fueron grabados correctamente");

                 //Se borrar los dartos introducidos en los campos
                bd_vi_lote.setText("");
                bd_vi_observaciones.setText("");

              } catch (IOException ioe) {
                Log.e("Ficheros", "Error al escribir fichero a tarjeta SD");
              }
            } //Fin if

          }
        }); //Fiun botón guardar

    } //Fin onCreate

    //Código para volver al menú inicio
    public void onClickBaseDatos(View view){
      Intent i= new Intent(this, Base_DatosActivity.class);
      startActivity(i);
    }

} //Fin Visita
```

11.- Fichero activity_acercade.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@android:color/holo_green_light"
   tools:context=".activity_acercade">

   <TextView
      android:id="@+id/textView27"
      android:layout_width="1073dp"
      android:layout_height="27dp"
      android:layout_marginBottom="8dp"
      android:layout_marginEnd="8dp"
      android:layout_marginLeft="8dp"
      android:layout_marginRight="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:text="- Agrisoftv1 es una aplicación en Android que acerca la nuevas
geotecnologías al sector agrario "
      android:textColor="@android:color/black"
      app:layout_constraintBottom_toTopOf="@+id/guideline27"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.89"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="1.0" />

   <android.support.constraint.Guideline
      android:id="@+id/guideline27"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      app:layout_constraintGuide_begin="67dp" />

   <TextView
      android:id="@+id/textView29"
      android:layout_width="806dp"
      android:layout_height="40dp"
      android:layout_marginBottom="8dp"
      android:layout_marginEnd="8dp"
      android:layout_marginLeft="8dp"
      android:layout_marginRight="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:text="- Permite generar un fichero donde se guardan todos los datos sobre las
actividades en el campo"
      android:textColor="@android:color/black"
      app:layout_constraintBottom_toTopOf="@+id/guideline29"
      app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintHorizontal_bias="0.209"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline27"
    app:layout_constraintVertical_bias="0.277" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline29"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="141dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline30"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="200dp" />

  <TextView
    android:id="@+id/textView30"
    android:layout_width="818dp"
    android:layout_height="27dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="- Permite el guiado de maquinaria en base a la tecnología GPS"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toTopOf="@+id/guideline30"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.215"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline29"
    app:layout_constraintVertical_bias="0.75" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline32"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="264dp" />

  <TextView
    android:id="@+id/textView31"
    android:layout_width="wrap_content"
    android:layout_height="39dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
```

```
      android:layout_marginLeft="8dp"
      android:layout_marginRight="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:text="- Trabajo Fin de Master, José Manuel Casla Francisco, Septiembre de
2018"
      android:textColor="@android:color/black"
      app:layout_constraintBottom_toTopOf="@+id/guideline32"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.12"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="@+id/guideline30"
      app:layout_constraintVertical_bias="1.0" />

   <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginBottom="8dp"
      android:layout_marginEnd="8dp"
      android:layout_marginLeft="8dp"
      android:layout_marginRight="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:background="@android:color/holo_blue_light"
      android:onClick="onClickMenuInicio"
      android:text="VOLVER AL MENU INICIO"
      android:textColor="@android:color/black"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="@+id/guideline32" />

</android.support.constraint.ConstraintLayout>
```

12.- Fichero activity_base_datos.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@android:color/holo_green_light"
   tools:context="com.example.josemanuel.agrisoftv1.Base_DatosActivity"
   tools:layout_editor_absoluteY="89dp">

   <android.support.constraint.Guideline
```

```
    android:id="@+id/guideline4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:visibility="visible"
    app:layout_constraintGuide_begin="348dp" />

<android.support.constraint.Guideline
    android:id="@+id/guideline5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:visibility="visible"
    app:layout_constraintGuide_begin="163dp" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="229dp"
    android:layout_height="38dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="1.5"
    android:scaleY="1.5"
    android:text="OPERACIÓN"
    android:textAppearance="@style/TextAppearance.AppCompat.Button"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline3"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.55"
    app:layout_constraintStart_toStartOf="parent" />

<android.support.constraint.Guideline
    android:id="@+id/guideline3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:visibility="visible"
    app:layout_constraintGuide_begin="231dp" />

<Button
    android:id="@+id/btn_visita"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
```

```
    android:background="@android:color/holo_blue_light"
    android:text="VISITA"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toTopOf="@+id/guideline4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.057"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintVertical_bias="0.566" />

  <Button
    android:id="@+id/btn_tratamiento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:text="TRATAMIENTO"
    android:textColor="@android:color/black"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.261"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintVertical_bias="0.566" />

  <Button
    android:id="@+id/btn_siembra"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:text="SIEMBRA"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toTopOf="@+id/guideline4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.478"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintVertical_bias="0.566" />

  <Button
    android:id="@+id/btn_cultivar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:text="CULTIVAR"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toTopOf="@+id/guideline4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.691"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintVertical_bias="0.566" />

  <Button
    android:id="@+id/btn_cosecha"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="36dp"
    android:background="@android:color/holo_blue_light"
    android:text="COSECHA"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toTopOf="@+id/guideline4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.927"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintVertical_bias="0.0" />

  <Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickMenuInicio"
    android:text="Volver a Menu Inicio"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.476"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline4"
    app:layout_constraintVertical_bias="0.431" />
```

```
    <TextView
        android:id="@+id/tv_fecha"
        android:layout_width="216dp"
        android:layout_height="40dp"
        android:layout_marginBottom="92dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"
        android:textColor="@android:color/black"
        android:visibility="visible"
        app:layout_constraintBottom_toTopOf="@+id/guideline5"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.047"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

13.- Fichero activity_cosecha.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_green_light"
    tools:context="com.example.josemanuel.agrisoftv1.Cosecha">

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/holo_green_light"
        android:scaleX="0.5"
        android:scaleY="0.5"
        android:text="PRODUCTO:"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="@android:color/black"
        android:textStyle="bold"
        android:visibility="visible"
        app:layout_constraintBottom_toTopOf="@+id/guideline10"
        app:layout_constraintEnd_toStartOf="@+id/guideline8"
```

```
    app:layout_constraintHorizontal_bias="0.6"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.736" />


  <TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="CANTIDAD (kg/ha):"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline9"
    app:layout_constraintEnd_toStartOf="@+id/guideline8"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline10" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:visibility="visible"
    app:layout_constraintGuide_begin="400dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:visibility="visible"
    app:layout_constraintGuide_begin="188dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:visibility="visible"
    app:layout_constraintGuide_begin="292dp" />
```

```
<android.support.constraint.Guideline
    android:id="@+id/guideline10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintGuide_begin="195dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:layout_constraintHorizontal_bias="0.024"
    tools:layout_constraintVertical_bias="0.019" />

<EditText
    android:id="@+id/bd_co_producto"
    android:layout_width="546dp"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:selectAllOnFocus="false"
    android:singleLine="false"
    android:textColor="@android:color/black"
    android:textColorHighlight="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textStyle="italic"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline10"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.186"
    app:layout_constraintStart_toStartOf="@+id/guideline8"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.784" />

<EditText
    android:id="@+id/bd_co_cantidad"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textStyle="italic"
```

```
    app:layout_constraintBottom_toTopOf="@+id/guideline9"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.115"
    app:layout_constraintStart_toStartOf="@+id/guideline8"
    app:layout_constraintTop_toTopOf="@+id/guideline10"
    app:layout_constraintVertical_bias="1.0" />

  <EditText
    android:id="@+id/bd_co_observaciones"
    android:layout_width="747dp"
    android:layout_height="112dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.483"
    app:layout_constraintStart_toStartOf="@+id/guideline8"
    app:layout_constraintTop_toTopOf="@+id/guideline7"
    app:layout_constraintVertical_bias="0.175" />

  <TextView
    android:id="@+id/textView19"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="DURACIÓN (h.min):"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline7"
    app:layout_constraintEnd_toStartOf="@+id/guideline8"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline9" />

  <EditText
    android:id="@+id/bd_co_duracion"
    android:layout_width="239dp"
```

```
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline7"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.118"
    app:layout_constraintStart_toStartOf="@+id/guideline8"
    app:layout_constraintTop_toTopOf="@+id/guideline9"
    app:layout_constraintVertical_bias="0.394" />

  <Button
    android:id="@+id/btn_co_volver"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickBaseDatos"
    android:text="VOLVER A BASE DATOS"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/guideline8"
    app:layout_constraintTop_toTopOf="@+id/guideline7"
    app:layout_constraintVertical_bias="0.924" />

  <Button
    android:id="@+id/btn_co_guardar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickGuardar"
    android:text="GUARDAR"
    android:textColor="@android:color/black"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.477"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline7"
    app:layout_constraintVertical_bias="0.923" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline31"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="80dp" />

  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="LOTE:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textSize="36sp"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline31"
    app:layout_constraintEnd_toStartOf="@+id/guideline8"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <EditText
    android:id="@+id/bd_co_lote"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
```

```
    android:textSize="18sp"
    android:textStyle="normal|italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline31"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.154"
    app:layout_constraintStart_toStartOf="@+id/guideline8"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.444" />

  <TextView
    android:id="@+id/tv_fecha"
    android:layout_width="wrap_content"
    android:layout_height="14dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:textColor="@android:color/holo_green_light"
    app:layout_constraintBottom_toTopOf="@+id/guideline31"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.941"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.62" />

  <TextView
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="OBSERVACIONES:"
    android:textColor="@android:color/black"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline8"
    app:layout_constraintHorizontal_bias="0.403"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline7"
    app:layout_constraintVertical_bias="0.085" />

</android.support.constraint.ConstraintLayout>
```

14.- Fichero activity_cultivar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@android:color/holo_green_light"
   tools:context="com.example.josemanuel.agrisoftv1.Cultivar">

   <TextView
      android:id="@+id/textView10"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginBottom="8dp"
      android:layout_marginEnd="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:background="@android:color/holo_green_light"
      android:scaleX="0.5"
      android:scaleY="0.5"
      android:text="TIPO LABOR:"
      android:textAppearance="@style/TextAppearance.AppCompat.Display1"
      android:textColor="@android:color/black"
      android:textStyle="bold"
      android:visibility="visible"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toStartOf="@+id/guideline15"
      app:layout_constraintHorizontal_bias="0.351"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.219" />

   <TextView
      android:id="@+id/textView11"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginBottom="8dp"
      android:layout_marginEnd="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:background="@android:color/holo_green_light"
      android:scaleX="0.5"
      android:scaleY="0.5"
      android:scrollX="0dp"
      android:scrollY="0dp"
      android:text="DURACIÓN(hrs.min):"
```

```
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline15"
    app:layout_constraintHorizontal_bias="0.545"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.345" />

  <TextView
    android:id="@+id/textView12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="OBSERVACIONES:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline15"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_begin="292dp" />

  <EditText
    android:id="@+id/bd_cu_labor"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="40dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textStyle="italic"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.044"
    app:layout_constraintStart_toStartOf="@+id/guideline15"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.18" />

  <EditText
    android:id="@+id/bd_cu_duracion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.044"
    app:layout_constraintStart_toStartOf="@+id/guideline15"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.345" />

  <EditText
    android:id="@+id/bd_cu_observaciones"
    android:layout_width="639dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textColorLink="@android:color/black"
    android:textStyle="bold|italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.285"
    app:layout_constraintStart_toStartOf="@+id/guideline15"
    app:layout_constraintTop_toTopOf="parent" />

  <Button
    android:id="@+id/btn_cu_volver"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
```

```
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickBaseDatos"
    android:text="VOLVER A BASE DATOS"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.041"
    app:layout_constraintStart_toStartOf="@+id/guideline15"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.745" />

  <Button
    android:id="@+id/btn_cu_guardar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:text="GUARDAR"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.528"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.745" />

  <TextView
    android:id="@+id/textView32"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="LOTE:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textSize="18sp"
    android:textStyle="bold"
    android:visibility="visible"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline15"
    app:layout_constraintHorizontal_bias="0.476"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.113" />

  <EditText
    android:id="@+id/bd_cu_lote"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="36dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="36dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textSize="18sp"
    android:textStyle="normal|italic"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.037"
    app:layout_constraintStart_toStartOf="@+id/guideline15"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.099" />

  <TextView
    android:id="@+id/tv_fecha"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:textColor="@android:color/holo_green_light"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.894"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.037" />
</android.support.constraint.ConstraintLayout>
```

15.- Fichero activity_guiado.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@android:color/holo_green_light"
   android:visibility="visible"
   tools:context="com.example.josemanuel.agrisoftv1.Guiado">


   <TextView
      android:id="@+id/textView21"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginBottom="8dp"
      android:layout_marginEnd="8dp"
      android:layout_marginLeft="8dp"
      android:layout_marginRight="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:background="@android:color/holo_green_light"
      android:scaleX="2"
      android:scaleY="2"
      android:text="@string/ancho_trabajo"
      android:textAppearance="@style/TextAppearance.AppCompat.Body2"
      android:textColor="@android:color/black"
      android:textColorLink="@android:color/black"
      android:textStyle="bold"
      android:visibility="visible"
      app:layout_constraintBottom_toTopOf="@+id/guideline6"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.403"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.479" />

   <android.support.constraint.Guideline
      android:id="@+id/guideline6"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      app:layout_constraintGuide_begin="83dp" />

   <android.support.constraint.Guideline
      android:id="@+id/guideline26"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```

```
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="166dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline28"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="320dp" />

  <EditText
    android:id="@+id/et_ancho"
    android:layout_width="94dp"
    android:layout_height="30dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:ems="10"
    android:inputType="numberDecimal"
    android:scaleX="1"
    android:scaleY="1"
    android:selectAllOnFocus="false"
    android:text="0.0"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="italic"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline6"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.675"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.371" />

  <TextView
    android:id="@+id/textView28"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="2"
    android:scaleY="2"
```

```
        android:text="PRECISION HZ"
        android:textAppearance="@style/TextAppearance.AppCompat.Body2"
        android:textColor="@android:color/black"
        android:textStyle="bold"
        android:visibility="visible"
        app:layout_constraintBottom_toTopOf="@+id/guideline26"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.402"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline6"
        app:layout_constraintVertical_bias="0.52" />

    <Button
        android:id="@+id/btn_inicio"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="52dp"
        android:background="@android:color/holo_blue_dark"
        android:scaleX="1.5"
        android:scaleY="1.5"
        android:text="@string/inicio_traza"

android:textAppearance="@android:style/TextAppearance.Material.Widget.Button.Colored
"
        android:visibility="visible"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.333"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline26"
        tools:targetApi="n" />

    <Button
        android:id="@+id/btn_final"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/holo_blue_dark"
        android:scaleX="1.5"
        android:scaleY="1.5"
        android:text="@string/final_traza"
```

```
android:textAppearance="@android:style/TextAppearance.Material.Widget.Button.Colored
"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline28"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.637"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline26"
    tools:targetApi="n" />

  <TextView
    android:id="@+id/tv_xutm"
    android:layout_width="169dp"
    android:layout_height="35dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="436dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="436dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="1"
    android:scaleY="1"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.841"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.05" />

  <TextView
    android:id="@+id/tv_precision"
    android:layout_width="94dp"
    android:layout_height="30dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline26"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.675"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline6"
    app:layout_constraintVertical_bias="0.675" />


 <TextView
    android:id="@+id/tv_despla"
    android:layout_width="47dp"
    android:layout_height="21dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_orange_light"
    android:scaleX="5"
    android:scaleY="7"
    android:text="      "
    android:textColor="@android:color/black"
    android:textSize="14sp"
    android:textStyle="italic"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.628"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.283" />

  <TextView
    android:id="@+id/tv_movi"
    android:layout_width="47dp"
    android:layout_height="21dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_orange_light"
    android:scaleX="5"
    android:scaleY="7"
    android:textColor="@android:color/black"
    android:textSize="14sp"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.341"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.283" />

  <Button
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickMenuInicio"
    android:text="VOLVER AL MENU INICIO"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.772" />

  <TextView
    android:id="@+id/tv_xf"
    android:layout_width="169dp"
    android:layout_height="47dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.903"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline26"
    app:layout_constraintVertical_bias="0.0" />

  <TextView
    android:id="@+id/tv_yf"
    android:layout_width="169dp"
    android:layout_height="47dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
```

```
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.903"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline26"
    app:layout_constraintVertical_bias="0.13" />

<TextView
    android:id="@+id/tv_yutm"
    android:layout_width="169dp"
    android:layout_height="35dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.903"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.196" />

<TextView
    android:id="@+id/tv_Da"
    android:layout_width="100dp"
    android:layout_height="25dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.473"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.0" />
```

```
<TextView
    android:id="@+id/tv_velocidad"
    android:layout_width="169dp"
    android:layout_height="35dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.587"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.607" />

<TextView
    android:id="@+id/textView35"
    android:layout_width="wrap_content"
    android:layout_height="47dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="VELOCIDAD (km/h):"
    android:textColor="@android:color/black"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.335"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline28"
    app:layout_constraintVertical_bias="0.607" />

<TextView
    android:id="@+id/tv_xi"
    android:layout_width="169dp"
    android:layout_height="47dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
```

```
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:background="@android:color/holo_green_light"
      android:textColor="@android:color/black"
      app:layout_constraintBottom_toTopOf="@+id/guideline28"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.043"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="@+id/guideline26"
      app:layout_constraintVertical_bias="0.0" />

   <TextView
      android:id="@+id/tv_yi"
      android:layout_width="169dp"
      android:layout_height="47dp"
      android:layout_marginBottom="8dp"
      android:layout_marginEnd="8dp"
      android:layout_marginLeft="8dp"
      android:layout_marginRight="8dp"
      android:layout_marginStart="8dp"
      android:layout_marginTop="8dp"
      android:background="@android:color/holo_green_light"
      android:textColor="@android:color/black"
      android:textStyle="italic"
      app:layout_constraintBottom_toTopOf="@+id/guideline28"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.042"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.898" />

</android.support.constraint.ConstraintLayout>
```

16.- Fichero activity_menu_inicio_.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@drawable/fondo1"
   tools:context="com.example.josemanuel.agrisoftv1.Menu_inicio_Activity">

   <android.support.constraint.Guideline
      android:id="@+id/guideline"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
```

```
    app:layout_constraintGuide_begin="20dp" />

<android.support.constraint.Guideline
    android:id="@+id/guideline2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="298dp" />

<Button
    android:id="@+id/btn_g"
    style="@style/Widget.AppCompat.Button.Colored"
    android:layout_width="113dp"
    android:layout_height="73dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_bright"
    android:text="GUIADO"
    android:textColorLink="@android:color/holo_green_light"
    android:textSize="24sp"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.524"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline2"
    app:layout_constraintVertical_bias="0.505" />

<Button
    android:id="@+id/btn_bd"
    style="@style/Widget.AppCompat.Button.Colored"
    android:layout_width="173dp"
    android:layout_height="73dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_bright"
    android:text="BASE DATOS"
    android:textSize="24sp"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.192"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline2"
    app:layout_constraintVertical_bias="0.504" />
```

```
<Button
    android:id="@+id/btn_s"
    style="@style/Widget.AppCompat.Button.Colored"
    android:layout_width="173dp"
    android:layout_height="73dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_bright"
    android:text="SALIR"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.853"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline2"
    app:layout_constraintVertical_bias="0.505" />

<TextView
    android:id="@+id/textView2"
    style="@android:style/Widget.DeviceDefault.TextView"
    android:layout_width="785dp"
    android:layout_height="113dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:rotationX="10"
    android:scaleX="4"
    android:scaleY="3"
    android:text="AGRISOFT"
    android:textAlignment="center"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/holo_red_light"
    android:textColorHighlight="@color/colorPrimary"
    android:textColorLink="@color/colorPrimary"
    android:textDirection="locale"
    android:textScaleX="1"
    android:textStyle="bold"
    android:verticalScrollbarPosition="right"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.537"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline"
    app:layout_constraintVertical_bias="1.0" />

<Button
    android:id="@+id/btn_acerca"
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/holo_blue_bright"
        android:text="Acerca de.."
        android:textColor="@android:color/black"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.974"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.031" />

</android.support.constraint.ConstraintLayout>
```

17.- Fichero activity_siembra.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_green_light"
    android:visibility="visible"
    tools:context="com.example.josemanuel.agrisoftv1.Siembra">

    <android.support.constraint.Guideline
        android:id="@+id/guideline20"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_begin="348dp" />

    <android.support.constraint.Guideline
        android:id="@+id/guideline21"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="76dp" />

    <android.support.constraint.Guideline
        android:id="@+id/guideline22"
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="148dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline23"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="231dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline24"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_begin="888dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline25"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="314dp" />

  <EditText
    android:id="@+id/bd_si_cultivo1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline22"
    app:layout_constraintEnd_toStartOf="@+id/guideline24"
    app:layout_constraintStart_toStartOf="@+id/guideline20"
    app:layout_constraintTop_toTopOf="@+id/guideline21" />

  <EditText
    android:id="@+id/bd_si_dosis1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
```

```
    android:singleLine="true"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline22"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline24"
    app:layout_constraintTop_toTopOf="@+id/guideline21" />

  <EditText
    android:id="@+id/bd_si_cultivo2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline23"
    app:layout_constraintEnd_toStartOf="@+id/guideline24"
    app:layout_constraintStart_toStartOf="@+id/guideline20"
    app:layout_constraintTop_toTopOf="@+id/guideline22" />

  <EditText
    android:id="@+id/bd_si_dosis2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline23"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline24"
    app:layout_constraintTop_toTopOf="@+id/guideline22" />

  <EditText
    android:id="@+id/bd_si_duracion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
```

```
    android:inputType="number"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline25"
    app:layout_constraintEnd_toStartOf="@+id/guideline24"
    app:layout_constraintStart_toStartOf="@+id/guideline20"
    app:layout_constraintTop_toTopOf="@+id/guideline23" />

  <EditText
    android:id="@+id/bd_si_observaciones"
    android:layout_width="362dp"
    android:layout_height="331dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline24"
    app:layout_constraintStart_toStartOf="@+id/guideline20"
    app:layout_constraintTop_toTopOf="@+id/guideline25" />

  <TextView
    android:id="@+id/textView22"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="DOSIS(kg.semillas/ha):"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline21"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline24"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/textView23"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
```

```
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="CULTIVO 1:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline22"
    app:layout_constraintEnd_toStartOf="@+id/guideline20"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline21" />

  <TextView
    android:id="@+id/textView24"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="CULTIVO 2:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline23"
    app:layout_constraintEnd_toStartOf="@+id/guideline20"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline22" />

  <TextView
    android:id="@+id/textView25"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="DURACIÓN(hrs.min):"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
```

```
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline25"
    app:layout_constraintEnd_toStartOf="@+id/guideline20"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline23" />

  <TextView
    android:id="@+id/textView26"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="OBSERVACIONES:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline20"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline25" />

  <Button
    android:id="@+id/btn_si_volver"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickBaseDatos"
    android:text="VOLVER A BASE DE DATOS"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.489"
    app:layout_constraintStart_toStartOf="@+id/guideline24"
    app:layout_constraintTop_toTopOf="@+id/guideline25"
    app:layout_constraintVertical_bias="0.455" />

  <Button
    android:id="@+id/btn_si_guardar"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:text="GUARDAR"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.486"
    app:layout_constraintStart_toStartOf="@+id/guideline24"
    app:layout_constraintTop_toTopOf="@+id/guideline25"
    app:layout_constraintVertical_bias="0.713" />

  <TextView
    android:id="@+id/textView33"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="LOTE:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textSize="18sp"
    android:textStyle="normal|bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline21"
    app:layout_constraintEnd_toStartOf="@+id/guideline20"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <EditText
    android:id="@+id/bd_si_lote"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
```

```
        android:textColor="@android:color/black"
        android:textSize="18sp"
        android:textStyle="italic"
        app:layout_constraintBottom_toTopOf="@+id/guideline21"
        app:layout_constraintEnd_toStartOf="@+id/guideline24"
        app:layout_constraintStart_toStartOf="@+id/guideline20"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tv_fecha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:textColor="@android:color/holo_green_light"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.93"
        app:layout_constraintStart_toStartOf="@+id/guideline24"
        app:layout_constraintTop_toTopOf="@+id/guideline25"
        app:layout_constraintVertical_bias="0.084" />

</android.support.constraint.ConstraintLayout>
```

18.- Fichero activity_tratamiento.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_green_light"
    tools:context="com.example.josemanuel.agrisoftv1.tratamiento">


    <EditText
        android:id="@+id/bd_trata1"
        android:layout_width="wrap_content"
        android:layout_height="54dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:ems="10"
```

```
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline13"
    app:layout_constraintEnd_toStartOf="@+id/guideline12"
    app:layout_constraintStart_toStartOf="@+id/guideline11" />


<EditText
    android:id="@+id/bd_dosis1"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline13"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline12" />


<android.support.constraint.Guideline
    android:id="@+id/guideline11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:visibility="visible"
    app:layout_constraintGuide_begin="310dp" />

<android.support.constraint.Guideline
    android:id="@+id/guideline12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_begin="715dp" />

<android.support.constraint.Guideline
    android:id="@+id/guideline13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="130dp" />

<android.support.constraint.Guideline
    android:id="@+id/guideline16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
```

```
    android:visibility="visible"
    app:layout_constraintGuide_begin="281dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline17"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="205dp" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="351dp" />

  <TextView
    android:id="@+id/textView13"
    android:layout_width="wrap_content"
    android:layout_height="44dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="PRODUCTO"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"

    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline14"
    app:layout_constraintEnd_toStartOf="@+id/guideline12"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/textView14"
    android:layout_width="346dp"
    android:layout_height="46dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="DOSIS(lts.grs/ha)"
```

```
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"

    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline14"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/textView15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="TRATAMIENTO 2:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline17"
    app:layout_constraintEnd_toStartOf="@+id/guideline11"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline13" />

  <TextView
    android:id="@+id/textView16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="TRATAMIENTO 3:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline16"
    app:layout_constraintEnd_toStartOf="@+id/guideline11"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline17" />
```

```
<TextView
    android:id="@+id/textView17"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="TRATAMIENTO 4:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline18"
    app:layout_constraintEnd_toStartOf="@+id/guideline11"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline16" />

<TextView
    android:id="@+id/textView18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="OBSERVACIONES:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline11"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline19"
    app:layout_constraintVertical_bias="0.088" />

<EditText
    android:id="@+id/bd_tra_observaciones"
    android:layout_width="300dp"
    android:layout_height="262dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
```

```
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guideline12"
    app:layout_constraintHorizontal_bias="0.363"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    app:layout_constraintTop_toTopOf="@+id/guideline19"
    app:layout_constraintVertical_bias="0.0" />

<EditText
    android:id="@+id/bd_trata2"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline17"
    app:layout_constraintEnd_toStartOf="@+id/guideline12"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    app:layout_constraintTop_toTopOf="@+id/guideline13" />

<EditText
    android:id="@+id/bd_dosis2"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline17"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    app:layout_constraintTop_toTopOf="@+id/guideline13" />

<EditText
    android:id="@+id/bd_trata3"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
```

```
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline16"
    app:layout_constraintEnd_toStartOf="@+id/guideline12"
    app:layout_constraintStart_toStartOf="@+id/guideline11"
    app:layout_constraintTop_toTopOf="@+id/guideline17" />

  <EditText
    android:id="@+id/bd_dosis3"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline16"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    app:layout_constraintTop_toTopOf="@+id/guideline17" />

  <EditText
    android:id="@+id/bd_trata4"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:ems="10"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline18"
    app:layout_constraintEnd_toStartOf="@+id/guideline12"
    app:layout_constraintStart_toStartOf="@+id/guideline11" />

  <EditText
    android:id="@+id/bd_dosis4"
    android:layout_width="wrap_content"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
```

```
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline18"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    app:layout_constraintTop_toTopOf="@+id/guideline16" />

<android.support.constraint.Guideline
    android:id="@+id/guideline19"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="429dp" />

<TextView
    android:id="@+id/textView20"
    android:layout_width="wrap_content"
    android:layout_height="42dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="4dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="DURACIÓN(hrs.min):"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textSize="30sp"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline19"
    app:layout_constraintEnd_toStartOf="@+id/guideline11"
    app:layout_constraintHorizontal_bias="0.41"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline18"
    app:layout_constraintVertical_bias="0.458" />

<EditText
    android:id="@+id/bd_trata_duracion"
    android:layout_width="234dp"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
```

```
    android:inputType="number"
    android:textColor="@android:color/black"
    android:textStyle="italic"
    app:layout_constraintBottom_toTopOf="@+id/guideline19"
    app:layout_constraintEnd_toStartOf="@+id/guideline12"
    app:layout_constraintHorizontal_bias="0.563"
    app:layout_constraintStart_toEndOf="@+id/textView20"
    app:layout_constraintTop_toTopOf="@+id/guideline18"
    app:layout_constraintVertical_bias="0.888" />

  <Button
    android:id="@+id/btn_volver"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickVolver"
    android:text="VOLVER BASE DATOS"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toTopOf="@+id/guideline19"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.753"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    app:layout_constraintTop_toTopOf="@+id/guideline18" />

  <Button
    android:id="@+id/btn_guardar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:text="Guardar"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toTopOf="@+id/guideline19"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.073"
    app:layout_constraintStart_toStartOf="@+id/guideline12"
    app:layout_constraintTop_toTopOf="@+id/guideline18" />

  <android.support.constraint.Guideline
    android:id="@+id/guideline14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="62dp" />
```

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="TRATAMIENTO 1:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline13"
    app:layout_constraintEnd_toStartOf="@+id/guideline11"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline14" />

<EditText
    android:id="@+id/trata_lote"
    style="@style/Widget.AppCompat.EditText"
    android:layout_width="166dp"
    android:layout_height="54dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:textColor="@android:color/black"
    android:textSize="30sp"
    android:textStyle="italic"
    android:visibility="visible"
    app:layout_constraintBottom_toTopOf="@+id/guideline14"
    app:layout_constraintEnd_toStartOf="@+id/guideline11"
    app:layout_constraintHorizontal_bias="0.873"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="87dp"
    android:layout_height="31dp"
    android:layout_marginBottom="8dp"
```

```
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/holo_green_light"
        android:scaleX="0.5"
        android:scaleY="0.5"
        android:text="LOTE"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="@android:color/black"
        android:textSize="24sp"
        android:textStyle="bold"
        android:visibility="visible"
        app:layout_constraintBottom_toTopOf="@+id/guideline14"
        app:layout_constraintEnd_toStartOf="@+id/guideline11"
        app:layout_constraintHorizontal_bias="0.042"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0" />

    <TextView
        android:id="@+id/tv_fecha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:textColor="@android:color/holo_green_light"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.947"
        app:layout_constraintStart_toStartOf="@+id/guideline12"
        app:layout_constraintTop_toTopOf="@+id/guideline19"
        app:layout_constraintVertical_bias="0.114" />

</android.support.constraint.ConstraintLayout>
```

19.- Fichero activity_visita.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_green_light"
    tools:context="com.example.josemanuel.agrisoftv1.Visita">
```

```
<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_green_light"
    android:scaleX="0.5"
    android:scaleY="0.5"
    android:text="OBSERVACIONES:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textStyle="bold"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.039"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.499" />

<EditText
    android:id="@+id/bd_vi_observaciones"
    android:layout_width="668dp"
    android:layout_height="412dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="text"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textSize="18sp"
    android:textStyle="italic"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.764"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.498" />

<Button
    android:id="@+id/btn_vi_volver"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
```

```
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:onClick="onClickBaseDatos"
    android:text="VOLVER BASE DATOS"
    android:textColor="@android:color/black"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.27"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.927" />

  <Button
    android:id="@+id/btn_vi_guardar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_blue_light"
    android:text="GUARDAR"
    android:textColor="@android:color/black"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.468"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.927" />

  <TextView
    android:id="@+id/EE"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="LOTE:"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="@android:color/black"
    android:textSize="18sp"
    android:textStyle="bold"
```

```
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.126"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.119" />

    <EditText
        android:id="@+id/bd_vi_lote"
        android:layout_width="224dp"
        android:layout_height="54dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="@android:color/black"
        android:textSize="18sp"
        android:textStyle="italic"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.351"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.097" />

    <TextView
        android:id="@+id/tv_fecha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:textColor="@android:color/holo_green_light"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.941"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.037" />
</android.support.constraint.ConstraintLayout>
```