

Trabajo Fin de Grado

DESARROLLO DE UNA APLICACIÓN EN ANDROID OS PARA LOS TRABAJOS DE ORGANIZACIÓN DEL TERRENO

Autor

Pedro Luis Tejero Gracia

Director/es

Director académico: Danilo Tardioli

Director militar: Daniel Sánchez López

Centro Universitario de la Defensa-Academia General Militar

202_



Agradecimientos

En agradecimiento al director académico D.Danilo Tardioli por su disposición para la realización de este proyecto así como a los integrantes del Batallón de Zapadores I por haber ofrecido sus conocimientos profesionales a lo largo de este proyecto y en especial al Capitán Daniel Sánchez López, director militar de este proyecto.



RESUMEN

En la actualidad, todos nosotros somos poseedores de un teléfono móvil, el cual llevamos a todas partes. Además, el sistema operativo Android es uno de los más populares dentro de los smartphones.

Existen infinidad de aplicaciones para gran cantidad de usos que van desde el ocio hasta el trabajo. Es por ello que muchas de estas aplicaciones nos pueden facilitar la vida en gran medida y todo ello al alcance de nuestro bolsillo.

El proyecto que se detalla en este documento pretende cubrir una de las necesidades dentro del Ejército de Tierra. El arma de Ingenieros, que es una de las que se compone el Ejército de Tierra, tiene como finalidad el apoyo de otras armas con la construcción de posiciones defensivas, acondicionando caminos y bases, entre otras muchas cosas.

Las Unidades de Ingenieros necesitan una manera rápida y cómoda que les ayude en la toma de decisiones a la hora de realizar trabajos de movimiento de tierra. Estas Unidades, trabajan a menudo en lugares apartados con limitaciones en la comunicación. A la hora de obtener datos como el tiempo de ejecución o el combustible necesario de un determinado trabajo se basan en la experiencia o en procesos lentos de búsqueda de información a través de las oficinas técnicas.

Después del estudio de las diferentes aplicaciones existentes, se llega a la conclusión de que no cumplen con esta necesidad y por lo tanto se busca desarrollar una aplicación específica para estas Unidades, que incluyan algunos de los trabajos fundamentales. Para determinar los diferentes requisitos se llevará a cabo una observación directa con una Unidad de Ingenieros además de un análisis documental.

El presente proyecto intenta ofrecer una aplicación que permita mostrar instantáneamente algunos de los datos más relevantes, que ayuden a tomar las mejores decisiones en los momentos necesarios. Esta aplicación tratará de calcular los tiempos de ejecución, coste y consumo de combustible, entre otros de los principales trabajos que se realizan en estas Unidades, como son excavaciones, explanaciones, compactaciones, transporte de tierras y losas. Para la obtención de estos datos se realizan diferentes cálculos, algunos de ellos con programas como CYPE 3D, con la ayuda de especificaciones de algunos fabricantes de maquinaria y manuales de construcción.

Para el desarrollo de la aplicación se contará con la ayuda de diferentes herramientas que permitan identificar los requisitos de la aplicación. Se realizará un análisis de riesgos previo que minimice el impacto de estos, así como diagramas de caso de uso y de actividad que nos aporten un esquema sencillo para facilitar su desarrollo. Para el desarrollo del código fuente de aplicación se utilizará el software Android Studio.

Palabras clave

Android
Aplicación
Trabajos
Ingenieros
Construcción



ABSTRACT

Nowadays, everyone owns a mobile phone, which we carry everywhere we go. Moreover, the Android operating system is one of the most popular among the smartphones.

There are countless applications for many uses ranging from leisure to work. That is why many of these applications can make our lives considerably easier and all within reach of our pocket.

The project detailed in this document aims to cover one of the needs of the Army. The Engineers branch, which is one of those that compose the Army, has the purpose of supporting other branches building defensive positions, conditioning roads and bases, and so on.

Engineer units need a quick and convenient way that help them to make decisions when they have to carry out earthmoving works. These units often work in remote locations with limited communication. When they have to obtaining data such as execution time or fuel required for a given job, they rely on experience or they get it from slow searching information processes through the technical offices.

After the study of the different existing applications, it is concluded that they do not meet this need and therefore it is sought to develop a specific application for these Units, which include some of the fundamental jobs. In order to determine the different requirements, a direct observation will be carried out with a Unit of Engineers and a documentary analysis will be performed.

The present project intends to offer an application that will allow to instantly display some of the most relevant data, which will help to make the best decisions at the necessary moments. This application will try to calculate the execution times, cost and fuel consumption, among others, of the main works that are carried out in these Units, such as excavations, levelling, compaction, earth transport and slabs. In order to obtain these data, different calculations are made, some of them with programs such as CYPE 3D, with the help of specifications from some machinery manufacturers and construction manuals.

For the development of the application, different tools will be used to identify the requirements of the application. A previous risk analysis will be performed to minimize the impact of these, as well as a use case and an activity diagram that provide us with a simple scheme to facilitate its development. For the development of the application source code, Android Studio software will be used.

KEYWORDS

Android
Application
Earthworks
Engineers
Construction



INDICE DE CONTENIDO

<i>RESUMEN</i>	<i>II</i>
<i>Palabras clave</i>	<i>II</i>
<i>ABSTRACT</i>	<i>III</i>
<i>KEYWORDS</i>	<i>III</i>
1. INTRODUCCIÓN	1
2. OBJETIVOS Y METODOLOGÍA	2
2.1. OBJETIVOS Y ALCANCE	2
2.2. METODOLOGÍA	2
3. ANTECEDENTES Y MARCO TEÓRICO	4
4. ANÁLISIS DE LOS REQUISITOS	7
4.1. Estudio de los trabajos de las Unidades de Ingenieros	7
4.2. Trabajos	8
4.2.1. Excavación	9
4.2.2. Explanación	10
4.2.3. Compactación	11
4.2.4. Transporte de tierra	11
4.2.5. Losa museo	11
5. Desarrollo de la aplicación	13
5.1. Análisis de riesgos	13
5.2. Modelo de casos de uso	15
5.3. Diagrama de actividad	16
5.4. Interfaz	18
5.4.1. Pantalla principal	18
5.4.2. Introducción de datos	19
5.4.3. Añadir maquinas y materiales	20
5.4.4. Losa museo	23
5.4.5. Muestra de los resultados	24
6. CONCLUSIONES	26
6.1. Trabajo Futuro	27
7. REFERENCIAS BIBLIOGRÁFICAS	28
<i>Bibliografía</i>	<i>28</i>



INDICE DE FIGURAS

Figura 1. Diagrama de Gantt 1 (Fuente: elaboración propia).....	3
Figura 2. Diagrama de Gantt 2 (Fuente: elaboración propia).....	3
Figura 3 Trabajos de Ingenieros (Fuentes: (Infodefensa, 2020) (FUTER, 2018) (DiarioCordoba, 2020) (Boletín de Tierra, 2021)).....	5
Figura 4. Losa museo (Fuente: (Wikipedia, s.f.))	12
Figura 5. Diagrama de casos de uso (Fuente: Elaboración propia).....	16
Figura 6. Diagrama de actividad (Fuente: elaboración propia)	17
Figura 7. Icono y nombre de la App (Fuente: elaboración propia).....	18
Figura 8. Pantalla principal (Fuente: elaboración propia).....	19
Figura 9. Pantalla trabajo excavación (Fuente: elaboración propia)	20
Figura 10. Pantallas para añadir maquinas (Fuente: elaboración propia)	21
Figura 11. Pantallas para añadir materiales (Fuente: elaboración propia)	22
Figura 12. Ventanas de información máquinas y materiales (Fuente: elaboración propia)	23
Figura 13. Pantalla trabajo losa museo (Fuente: elaboración propia).....	24
Figura 14. Pantalla principal con trabajos añadidos. (Fuente: elaboración propia)	25
Figura 15. Datos generales CYPE3D (Fuente: elaboración propia)	34
Figura 16. Datos grupos CYPE3D (Fuente: elaboración propia)	35
Figura 17. Tipo de vigas para losa (Fuente: elaboración propia).....	35
Figura 18. Datos paño para cálculos de losa (Fuente: elaboración propia).....	36
Figura 19. Cálculo armado losa Leopardo 2E (Fuente: elaboración propia).....	36
Figura 20. Cálculo armado losa BMR (Fuente: elaboración propia)	37
Figura 21. Cálculo armado losa Pizarro (Fuente: elaboración propia).....	37
Figura 22. Presupuestos Arquímedes para losa Leopardo 2E (Fuente: elaboración propia)	38
Figura 23. Presupuestos Arquímedes para losa BMR (Fuente: elaboración propia).....	38
Figura 24. Presupuestos Arquímedes para losa Leopardo 2E (Fuente: elaboración propia)	39
Figura 25. Diagrama de Gantt para losa Leopardo 2E (Fuente: elaboración propia)	39
Figura 26. Diagrama de Gantt para losa BMR (Fuente: elaboración propia).....	40
Figura 27. Diagrama de Gantt para losa Pizarro (Fuente: elaboración propia)	40
Figura 28. Clases aplicación <i>Android Studio</i> (Fuente: elaboración propia)	41
Figura 29. Layouts aplicación <i>Android Studio</i> (Fuente: elaboración propia).....	41



INDICE DE TABLAS

Tabla 1. Rendimientos según regímenes de trabajo (Fuente: (MADOC, 2005))	9
Tabla 2. Características vehículos	12
Tabla 3. Clasificación de riesgos (Fuente: elaboración propia)	13
Tabla 4. Análisis de riesgos (Fuente: elaboración propia)	15
Tabla 5. Excavaciones manuales rendimientos (Fuente: (Polpaico, 2011))	31
Tabla 6. Rendimientos compactación manual (Fuente: (Polpaico, 2011)).....	31
Tabla 7. Rendimientos de carga manual (Fuente: (Polpaico, 2011)).....	31
Tabla 8. Rendimientos de excavadoras (Fuente: (MaquinariasPesadas, 2015))	32
Tabla 9. Rendimiento bulldozer (Fuente: (MaquinariasPesadas, 2015)).....	32
Tabla 10. Rendimientos motoniveladoras (Fuente: (MaquinariasPesadas, 2015))	32
Tabla 11. Rendimientos compactadores (Fuente: (MaquinariasPesadas, 2015))	32
Tabla 12. Rendimientos cargadoras (Fuente: (MaquinariasPesadas, 2015)).....	32
Tabla 13. Rendimientos Volquetes (Fuente: (MaquinariasPesadas, 2015)).....	33
Tabla 14. Resultados armado losas (Fuente: elaboración propia).....	36
Tabla 15. Costes asociados a las losas de cada vehículo.....	37



ABREVIATURAS, SIGLAS Y ACRÓNIMOS

A- Acero

AE- Acciones en la edificación

App- Aplicación

BZAP- Batallón de Zapadores

CENAD- Centro de Adiestramiento

CTE- Código Técnico de la Edificación

DB- Documento Básico

EHE- Instrucción de Hormigón Estructural

FUTER- Fuerza Terrestre

HA- Hormigón Armado

IDE- Entorno Integrado de Desarrollo

MADOC – Mando de Adiestramiento y Doctrina

OS- Sistema Operativo

SE- Seguridad Estructural



1. INTRODUCCIÓN

El Ejército de Tierra está compuesto por diferentes Armas, las cuales realizan cometidos muy diferentes unas de otras, pero todos ellos igual de necesarios para la consecución de unos objetivos comunes. Las distintas Armas que lo componen son Infantería, Ingenieros, Transmisiones, Artillería, Caballería y Aviación.

Los Ingenieros del Ejército de Tierra son fundamentales para el apoyo a otras Unidades. Son capaces de llevar a cabo acciones de movilidad, contramovilidad y protección como parte de una operación. Disponen de medios únicos dentro del Ejército y cuentan con la capacidad y flexibilidad para llevar a cabo trabajos muy diferentes.

Algunos de los cometidos asignados a una Unidad de Ingenieros son proporcionar apoyo en la excavación de posiciones defensivas, modificar las condiciones del terreno, construcción o reparación de caminos, ayuda en catástrofes naturales o empleo de maquinaria para obras públicas.

En el Ejército de Tierra impera la necesidad de toma de decisiones rápida. Una decisión bien tomada puede ser motivo de éxito o por el contrario ser la desencadenante de una catástrofe. Es por ello que toda información es una buena ayuda para cualquier mando al que corresponda decidir sobre alguna situación.

En el caso de las Unidades de Ingenieros, especialmente de Zapadores (Ingenieros de combate) estas decisiones pueden ir desde montar una posición defensiva o tender un campo de minas, hasta realizar un hangar para sus vehículos. Además, para estas unidades que pueden realizar trabajos tan diversos y descentralizados resulta complicado organizar al personal y equipo para ser lo más eficiente posible. Las Unidades de Zapadores se suelen componer de varias Compañías de fusileros, una Compañía de Apoyo, que se encarga de la mayor parte de las obras y donde se centraliza la mayor parte de la maquinaria pesada, y una Compañía de Plana Mayor y Servicios.

Todas las Unidades de Ingenieros poseen una gran variedad de materiales. Por ello, y por falta de experiencia en trabajos con máquinas o desconocimiento de estas, resulta complicado saber el tiempo, personal y material necesario para realizar un trabajo.

Con este proyecto se pretende desarrollar una aplicación (App) para dispositivos móviles basados en el sistema operativo Android que sea fácilmente accesible y que tendrá como finalidad calcular y mostrar una serie de parámetros clave, como el tiempo de ejecución o el coste, de los trabajos que se vayan añadiendo para ayudar al jefe de una Unidad a tomar decisiones en base a lo que es posible o no es posible hacer con un presupuesto marcado o en un plazo establecido. Además, servirá como medio de organización, es decir, para que pueda llevarse un control de los trabajos que se han ido realizando y de cuántos trabajos se tienen activos y con qué medios.



2. OBJETIVOS Y METODOLOGÍA

2.1. OBJETIVOS Y ALCANCE

El objetivo general de este trabajo es desarrollar una aplicación en Android OS que sirva como ayuda para realizar trabajos de movimiento de tierras y otras obras que realicen las unidades de Zapadores.

Los objetivos específicos de la aplicación son los siguientes:

- Calcular de una manera rápida, en cualquier lugar, incluidos aquellos en los que no se disponga de un ordenador o una conexión a internet, los tiempos de ejecución, coste y combustible de un trabajo que se esté planteando ejecutar. Con el simple uso de esta aplicación será posible obtener los datos necesarios para tomar decisiones rápidamente.
- Permitir agregar tantos trabajos como sea necesario, favoreciendo la organización que se vaya a llevar a cabo, tanto de personal como de medios asignados a cada uno de ellos, ya sean máquinas, materiales, u otros.
- Poder diferenciar los trabajos asignando un nombre, identificador o similar para poder identificar claramente cada uno, evitando así posibles errores tanto logísticos como de organización.
- Poder asignar las máquinas que se vayan a utilizar para cada trabajo, ya sean cargadoras, retroexcavadoras, bulldozer, motoniveladoras, compactadoras o volquetes, para poder realizar los cálculos con mayor exactitud y precisión.
- Poder asignar materiales si fuera necesario para el trabajo, de manera que se incluyan en los costes del trabajo.

2.2. METODOLOGÍA

Después de realizar el estudio del estado del arte, en el que se han investigado algunas de las aplicaciones existentes, se ha llegado a la conclusión de que ninguna de ellas es válida para el uso en las Unidades de Zapadores del Ejército de Tierra, debido a que no cumplen con alguno de los requisitos definidos en el apartado anterior, aunque podrían utilizarse de manera complementaria a la aplicación que se va a desarrollar en el proyecto. Este estudio se detalla en el siguiente apartado.

Por otra parte, será necesario determinar qué tipo de parámetros se van a calcular y hasta qué punto la aplicación podrá servir de ayuda. Para obtener esta información se usarán métodos cualitativos en base a entrevistas, experiencias y otras observaciones.

Para conocer las principales obras que realiza una unidad de Zapadores se revisarán diferentes artículos en revistas de interés, como Ejército de Tierra, así como otras fuentes que incluyan diferentes prácticas y obras que se hayan realizado. También se realizarán entrevistas a los diferentes miembros del Batallón de Zapadores I además de realizar una observación directa de los trabajos que se realicen durante un tiempo estimado de 2 meses. De todos los trabajos se seleccionarán aquellos que se realicen de manera habitual y que puedan ser automatizados.

Posteriormente, se realizará un análisis de riesgos para determinar que acciones se deben llevar a cabo para minimizar el impacto de estos. Además, se realizará un modelo de requisitos con el objetivo de delimitar el sistema y obtener la funcionalidad que debería tener con el usuario. Se desarrollará un diagrama de casos de uso de la aplicación que describe las actividades que pueden llevarse a cabo, y un modelo de proceso que permita comprender las relaciones que se llevan a cabo dentro de la aplicación. Para todo ello se utilizará la herramienta "Visual Paradigm



Pedro Luis Tejero Gracia

Online". Por último, se detalla la interfaz que interactúa con el usuario y se explicaran las diferentes pantallas que contiene.

A continuación, se incluye un pequeño diagrama de Gantt del proyecto, con cada una de las tareas y su duración.

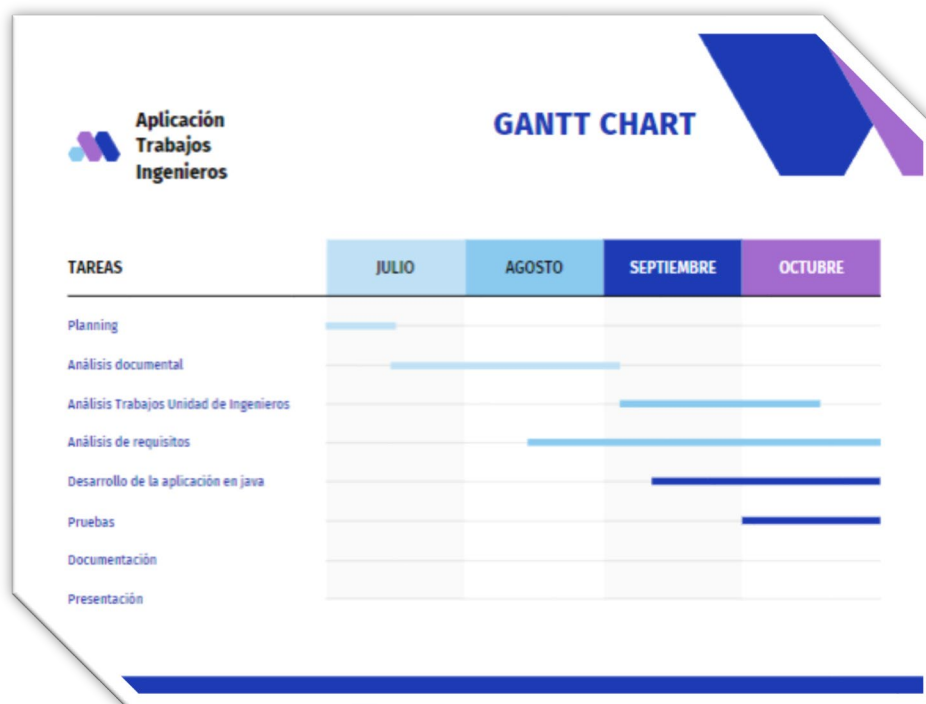


Figura 1. Diagrama de Gantt 1 (Fuente: elaboración propia)

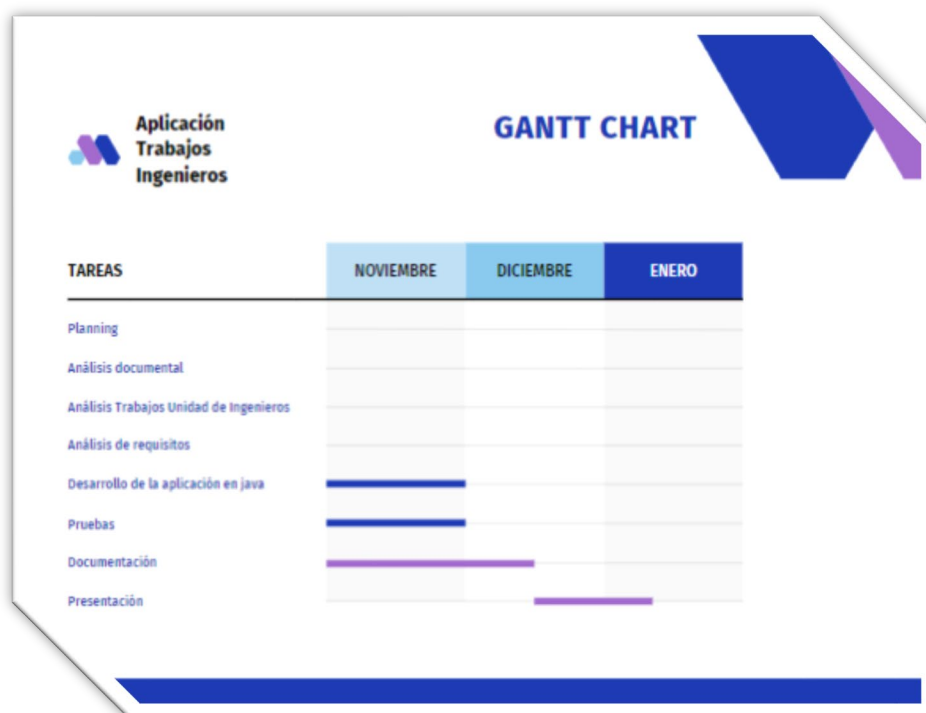


Figura 2. Diagrama de Gantt 2 (Fuente: elaboración propia)



3. ANTECEDENTES Y MARCO TEÓRICO

Según el manual de “Táctica y logística de Ingenieros” los Ingenieros del Ejército de Tierra son fundamentales para el apoyo a otras Unidades. Son capaces de llevar a cabo acciones de movilidad, contramovilidad y protección como parte de una operación. Disponen de medios únicos dentro del Ejército y cuentan con la capacidad y flexibilidad para llevar a cabo trabajos muy diferentes (MADOC, 2020).

Según este manual es posible realizar fosos contra carro como obstrucciones principales para preservar y proteger la capacidad propia de maniobrar y dar forma a la maniobra enemiga. Para la ejecución de estos fosos se recomienda el uso de máquinas bulldozers, cargadoras y retroexcavadoras/carros de zapadores. Además, indica que las posiciones defensivas deberán contar con pozos de tirador, asentamientos para armas colectivas y vehículos tanto de Sección como de Compañía y abrigos para el personal, todo ello enterrado y comunicado por ramales (MADOC, 2020).

En el manual “Organización del terreno para pequeñas Unidades” (MADOC, 2005) se marcan algunos trabajos de movimiento de tierras entre los que destacan los siguientes:

- Obstáculo de protección.
- Puestos de Mando y observatorios de las Pequeñas Unidades.
- Asentamientos de armas.
- Zanjas de protección de personal.
- Fosos de protección de vehículos.
- Abrigos de asentamiento.
- Abrigos de descanso.
- Puestos de Socorro de las Pequeñas Unidades.
- Depósitos de municiones de las Pequeñas Unidades.
- Ramales de comunicación.
- Enmascaramiento correspondiente.
- Despeje de campos de tiro

A continuación, se detallarán algunos de los trabajos de Ingenieros realizados en los últimos años (ver Figura 3).

En la imagen arriba a la izquierda se muestra la zona de combate urbano “El abejar” del Centro de Adiestramiento (CENAD) “San Gregorio” que es un conjunto de instalaciones utilizadas para el adiestramiento en combate en población, que cuentan con edificios sin tejado, zonas subterráneas, un foso de tiro con calles diferenciadas, obstáculos para apertura de puertas a la fuerza y una torre multiusos. La Brigada Aragón I encargó estas instalaciones a su Batallón de Zapadores para la utilización de las Unidades de Zaragoza. (Infodefensa, 2020)

En la imagen arriba a la derecha se muestra como la Unidad del Batallón de Zapadores VII realiza el “Camino Español” en la Base General Morillo (FUTER, 2018). Este tipo de obras de acondicionamiento y mejora de las instalaciones son muy comunes en las Compañías de apoyo de las Unidades de Zapadores, en las que realizan trabajos de movimiento de tierras para posteriormente realizar el hormigonado o asfaltado de un camino.

En la imagen abajo a la izquierda se pueden observar los trabajos de fortificación realizados por el Batallón de Zapadores X de la Brigada “Guzmán el Bueno” X, construyendo asentamientos, así como realizando trabajos de fortificación ligera. (DiarioCordoba, 2020)

En la imagen abajo a la derecha se muestra cómo durante unas maniobras del Batallón de



Pedro Luis Tejero Gracia

Zapadores de la Brigada Aragón I, el Pelotón de Máquinas realizó trabajos de preparación del terreno para canalización y hormigonado de una losa para aparcar vehículos en el Grupo Logístico de la Brigada. (Boletín de Tierra, 2021)



Figura 3 Trabajos de Ingenieros (Fuentes: (Infodefensa, 2020) (FUTER, 2018) (DiarioCordoba, 2020) (Boletín de Tierra, 2021))

Actualmente, la manera de trabajar en las Unidades de Zapadores es a través de la experiencia y preguntando a los operadores de las diferentes máquinas, de manera que cada vez que se quiere llevar a cabo un trabajo nuevo, resulta complicado saber cuál será el tiempo de ejecución, el combustible necesario, incluso el coste asociado a cada obra. Las compañías de apoyo, por suerte, disponen de una oficina técnica, que solicita datos concretos de cada uno de los trabajos, máquinas y materiales que se necesitan para efectuar el trabajo. Sin embargo, todo este proceso podría efectuarse de una manera mucho más sencilla, que permita reducir los plazos a través de una aplicación específica.

Existen en la Play Store de Android OS diferentes aplicaciones de ayuda a trabajos con maquinaria o de obras. La empresa Caterpillar, por ejemplo, pone a disposición de sus clientes diferentes aplicaciones especialmente para cálculo de productividades y rendimiento de sus máquinas:

- Cat Cycle Timer. Permite calcular productividades de las máquinas utilizadas cuando el usuario introduce los diferentes ciclos que se realizan, a través de un contador implementado en la App. Esta aplicación podría servir de manera complementaria a nuestra aplicación, sin embargo, no aporta ninguna información de los trabajos que se quieren realizar.
- Cat App: Fleet Management. Con esta aplicación es posible hacer un seguimiento de la flota de máquinas de las que se dispone. Es posible saber su ubicación, las horas de trabajo, combustible, posibles fallos y planes de mantenimiento. Sin embargo, no dice nada de sus productividades ni de los trabajos que quieran realizar con estas máquinas.

También pueden encontrarse otras de aplicaciones de productividad como:

- Suelo y Terraplén Calculadora. Aplicación con la que podemos realizar diferentes cálculos de las características del terreno como puede ser contenido de agua, coeficientes de presión, porcentajes de compactación, etc. Esta aplicación no nos da información de tiempos de ejecución, combustible necesario ni costes de los trabajos



Pedro Luis Tejero Gracia

que queramos realizar.

- The volume of excavation works. Calcula volúmenes de tierra en base a dimensiones especificadas por el usuario. Igual que la aplicación anterior, no nos aporta información de los trabajos que queramos realizar.
- PlanRadar. Con esta App es posible llevar el seguimiento de una obra de una edificación. Es posible introducir planos y documentación, administrar tareas y organizar diferentes proyectos. Sin embargo, no pueden realizarse actividades de movimiento de tierras.

También existen algunos softwares para ordenador relacionados con trabajos de movimiento de tierras entre los que se destaca "Fleet Production and Cost Analysis". Es un software que incluye algunas de las funciones que se buscan en un trabajo de movimiento de tierra. Este software ayuda a seleccionar qué máquinas son necesarias para el trabajo, además de ser de ayuda para la supervisión del trabajo. También permite hacer cálculos estimados de los diferentes costes de producción.

Debido a que no existe una aplicación como esta, para Android OS es necesario desarrollar una, que permita conocer la información básica, necesaria para efectuar los proyectos que se realizan en las Unidades de Zapadores, especialmente las Compañías de Apoyo. Para llegar al desarrollo de la aplicación se utilizará el software *Android Studio* que permite el desarrollo de aplicaciones en el lenguaje de programación java.



4. ANÁLISIS DE LOS REQUISITOS

En este apartado se refleja el proceso que se ha llevado a cabo para conseguir extraer los requisitos que necesita la aplicación, más concretamente, cuáles son los trabajos que deberán incluirse y qué es lo que se necesita que se calcule y se muestre en ella.

Se ha realizado un estudio de los trabajos que se han observado en el Batallón de Zapadores I, en Zaragoza, con la finalidad de obtener estos requisitos.

Posteriormente se explicará detalladamente cuáles son los trabajos que se incluirán en la aplicación con los respectivos cálculos en cada uno de ellos.

4.1. Estudio de los trabajos de las Unidades de Ingenieros

Como se puede observar, existe una gran diversidad de cometidos que llevan implícitos la realización de trabajos de movimiento de tierra u obras. Por lo tanto, se escogerán para implementar en la aplicación algunos de los trabajos fundamentales que se realizan frecuentemente y puedan ser útiles en la mayor parte de cometidos.

Durante el periodo comprendido entre el 6 de septiembre y el 15 de octubre, se ha podido observar de manera directa, en el Batallón de Zapadores I, algunos de los trabajos realizados por la Compañía de Apoyo, siendo estos los siguientes:

- Explanación de superficie de 400 m² en interior de dos edificaciones para realizar losas de cimentación que permitan utilizar las edificaciones como almacén
- Explanación de superficie de 225 m² para realización de una losa de cimentación
- Excavación de 36 m³ para sustitución del sistema de suministro de agua
- Explanación y acondicionamiento de superficie de 1300m² para posterior utilización como zona de aparcamiento de vehículos
- Construcción de losas para elaboración de museo de carros de combate

Del estudio realizado, tanto documental como por observación, se extrae como conclusión que para la realización de cualquier obra hacen falta trabajos de movimiento de tierras. Además, se ha podido observar que los trabajos más frecuentes son excavaciones, explanaciones, compactación del terreno, transporte de tierra y trabajos de hormigonado. Por lo tanto, se pretende que en la aplicación puedan calcularse estos trabajos pudiendo utilizar tanto especificaciones técnicas de las máquinas, como rendimientos del personal incluido en los trabajos. Seguidamente necesitaremos definir que queremos obtener de estos trabajos para que nuestra aplicación resulte útil en las Unidades.

Para especificar qué es lo que se espera realmente de la App, se han llevado a cabo diferentes entrevistas personales con el Capitán Daniel Sánchez López, jefe de la Compañía de Apoyo del Batallón de Zapadores I, de las que se han podido extraer las conclusiones que se exponen a continuación:

- Resulta complicado hacer estimaciones del tiempo que necesita una máquina para realizar un trabajo
- Es común que el combustible de aprovisionamiento sea mucho mayor del que realmente se utiliza
- El presupuesto para cada proyecto es limitado y en la mayor parte de ocasiones es el coste de los trabajos lo que determina si se va a llevar a cabo

En el apartado siguiente se detallarán más específicamente los trabajos que incluirá la aplicación.



4.2. Trabajos

Después de analizar las diferentes obras que se van a llevar a cabo en una unidad de Zapadores hará falta decidir qué trabajos se van a incluir en la versión inicial de la aplicación.

En base a las conclusiones obtenidas del apartado anterior, en la aplicación se incluyen cinco tipos de trabajos, excavaciones, explanaciones, compactación, transporte de tierra y losas museo. Los cuatro primeros por ser trabajos que se necesitan para cualquier proyecto de Ingenieros que se lleve a cabo y las losas museo por ser un tipo de trabajo que realiza habitualmente la Compañía de Apoyo del Batallón de Zapadores I. La construcción de losas de cimentación es un trabajo que no es posible implementar en la aplicación debido a la complejidad de los cálculos, ya que cada losa que se construye lleva definidas unas cargas que son las que determinan el armado que deberá llevar finalmente. Sin embargo, como suelen repetirse losas de exposición de vehículos de modelos iguales o muy similares, será posible calcularlos solamente una vez externamente a la aplicación y que estos cálculos puedan ser utilizados cada vez que se vaya a exponer un vehículo de ese modelo.

Será necesario calcular una estimación de tiempo, coste y combustible, en los trabajos de movimiento de tierras, para que la aplicación pueda realizarlos de una manera sistemática. Para llegar a ser lo más preciso posible, se estudiará incluir otro tipo de información como tipo de terreno, herramientas a utilizar, maquinaria, etc.

Para los cálculos se utilizarán diferentes regímenes de trabajo extraídos del manual “Organización del terreno para pequeñas unidades” (MADOC, 2005) y que serán los siguientes:

- Discontinuo de día
Corresponde al trabajo de un turno en jornada diurna de ocho horas, divididas en dos períodos por un descanso intermedio que se aprovecha para la comida. Excepcionalmente podrá aumentarse la duración de la jornada hasta diez horas, con merma en el rendimiento horario de un 5% aproximadamente. Este régimen está indicado cuando se dispone de personal y herramienta en número que permite el trabajo en un solo turno y la situación táctica no impone el trabajo nocturno.
- Discontinuo de noche
Corresponde al trabajo de un turno en jornada nocturna de ocho horas, dividida en dos períodos por un descanso intermedio que se aprovecha para la cena. El rendimiento horario obtenido es sensiblemente menor al del trabajo discontinuo de día. Excepcionalmente podrá incrementarse la duración de la jornada hasta diez horas, con merma en el rendimiento de un 5% aproximadamente. Este régimen sólo se empleará cuando la situación táctica lo imponga.
- Continuo de día
Corresponde al trabajo de dos turnos, cada uno de ellos de seis horas, que se relevan al mediodía. El rendimiento horario obtenido es igual al del trabajo discontinuo de día. Excepcionalmente, si el arco diurno lo permite, podrá aumentarse la jornada hasta catorce horas, trabajando cada turno siete horas como máximo, sin merma en el rendimiento horario. Este régimen está indicado cuando se dispone de personal en exceso para organizar solamente un turno y la situación táctica lo permita.
- Continuo de noche
Corresponde al trabajo de dos turnos, cada uno de ellos de seis horas, que se relevan a media noche. El rendimiento horario obtenido es igual al del trabajo discontinuo de noche. Excepcionalmente podrá incrementarse la jornada lo que permita el arco nocturno, unas siete horas como máximo por cada turno, sin merma sensible en el rendimiento horario. Este régimen está indicado cuando se dispone de personal en exceso para organizar un solo turno y la situación táctica impone el trabajo nocturno.



- Continuo de día y de noche

Corresponde al trabajo ininterrumpido de tres o cuatro turnos con jornadas de ocho o seis horas, respectivamente. El rendimiento horario obtenido es intermedio entre el del trabajo discontinuo de día y el discontinuo de noche. Este régimen de trabajo está indicado cuando se dispone de personal en exceso para organizar dos o tres turnos y la situación lo permita.

- A destajo

Este régimen de trabajo da buen rendimiento cuando la tarea asignada, individual o por equipos, está bien calculada: ni inferior a la que se pudiera hacer en jornada normal, ni tan excesiva que no pueda cumplirse. Consiste en fijar a cada hombre o equipo de trabajo una tarea, cumplida la cual el personal se retirará a descansar sin sujeción a horario. Puede obtener rendimientos superiores al normal, pero tiene el inconveniente de que desorganiza los horarios.

Los rendimientos relativos utilizados para los diferentes cálculos serán mostrados en la Tabla 1 extraídos del mismo manual (MADOC, 2005). Se entiende por rendimiento relativo el cociente entre la duración en jornadas de un trabajo realizado en régimen normal (discontinuo de día con jornada de ocho horas) y la duración del mismo realizado en un régimen cualquiera y con la misma mano de obra.

Régimen	Rendimiento relativo
Discontinuo de día (jornada de 8h)	1
Discontinuo de día (jornada de 10h)	1.15
Discontinuo de noche (jornada de 8h)	0.6
Discontinuo de noche (jornada de 10h)	0.70
Continuo de día (jornada de 6h)	0.75
Continuo de día (jornada de 7h)	0.85
Continuo de noche (jornada de 6h)	0.45
Continuo de noche (jornada de 7h)	0.5
Continuo de día y noche (jornada de 6h)	0.6
Continuo de día y noche (jornada de 8h)	0.8
A destajo	1.2

Tabla 1. Rendimientos según regímenes de trabajo (Fuente: (MADOC, 2005))

4.2.1. Excavación

Todos los trabajos de excavación se entenderán como excavaciones superficiales, es decir, la profundidad de la excavación será de entre 2 y 3 metros de profundidad como máximo.

“Las excavaciones superficiales se pueden realizar con herramientas manuales ya que no implica un proceso extenso. Por el contrario, pueden ser muy superficiales (0,60 mts) lo que hará innecesario el uso de maquinaria pesada.” (Foxtand, s.f.)

Se tendrá en cuenta el tipo de suelo para los cálculos de tiempo y maquinaria a utilizar en su caso. Se diferenciará entre excavaciones en roca, en terreno de tránsito y en tierra.

El cálculo de tiempo para este tipo de trabajos será el siguiente:

$$Tiempo = Tp + V/(Rmq + Rp * E)$$

Siendo:

- Tp = Tiempo de preparación. En los trabajos realizados en la Compañía de Apoyo del Batallón de Zapadores I, se observaron tiempos de preparación de 20, 35 y 35



minutos. Para la realización de los cálculos se hará la media de dichos datos, siendo 0.5 el valor de los trabajos de excavación.

- V = Volumen de tierra de excavación.
- Rmq = Rendimiento total de las máquinas utilizadas. Los rendimientos de las máquinas utilizadas en la aplicación han sido extraídos de la página web de su fabricante “Caterpillar”, recopilados, testeados y ajustados por diferentes operarios (MaquinariasPesadas, 2015). Se pueden consultar las tablas el Anexo I.
- Rp = Rendimiento total del personal que realiza excavaciones manuales. Los rendimientos de personal que se utilizan en la aplicación han sido extraídos del manual del constructor del grupo “Polpaico” (Polpaico, 2011). Se pueden consultar las tablas en el Anexo I.
- E = Eficiencia del personal atendiendo al régimen de trabajo seleccionado. Debido a los diferentes regímenes de trabajo que se puedan llevar a cabo, se incluirán factores de eficiencia para tratar de aproximarse lo máximo posible a tiempos reales de ejecución. No se tendrán en cuenta diferencias entre turnos diurnos o nocturnos, aunque generalmente la eficiencia es menor en turnos nocturnos. Para jornadas de trabajo de hasta 8 horas se considerará una eficiencia del 100%, de hasta 12 horas se considerará una eficiencia del 85% y para jornadas de trabajo de más de 12 horas del 70%, extraído y adaptado del manual de “Táctica y logística de Ingenieros” del Ejército de Tierra (MADOC, 2020).

El cálculo del coste de los trabajos de excavación se calculará de la siguiente manera:

$$Coste = C * T * P + M$$

Siendo:

- C = Consumo de combustible de las diferentes máquinas, que se calculará conforme vayamos añadiendo las máquinas. Los consumos de combustible han sido extraídos del mismo manual que los rendimientos (MaquinariasPesadas, 2015)
- T = Tiempo de ejecución de la obra calculado anteriormente
- P = Precio aproximado del combustible
- M = Costes de material, que serán calculados al añadir los materiales

4.2.2. Explanación

Las explanaciones son obras muy comunes que se llevan a cabo en la construcción de caminos, parking o simplemente como una base asentada para la construcción de una losa de hormigón u otros elementos.

En los trabajos de explanación se determinará la superficie en metros cuadrados. No será obligatorio añadir al menos una motoniveladora, pero sí recomendable, debido a que, aunque las explanaciones se puedan hacer manualmente, el tiempo de ejecución y el acabado final son muy diferentes. Además, sería de utilidad añadir alguna “bulldozer” u otra maquinaria que posea ripper para poder remover la tierra, lo que facilita mucho el trabajo y disminuye su tiempo de ejecución.

Habrà que tener en cuenta que las explanaciones tendrán como máximo 30 centímetros de espesor y que si los sobrepasa será necesario añadir un trabajo de compactación cada vez que sea necesario.

Los cálculos utilizados para calcular el tiempo en los trabajos de explanación son:

$$Tiempo = Tp + (Tp/2) * (S/1000) + S/(Rmq + Rp * E)$$

Siendo:

- Tp = Para trabajos de explanación, el tiempo de preparación se estima en 1 hora debido a que es necesario hacer el replanteo del terreno antes de realizar la explanación. Se incrementará la mitad del tiempo de preparación cada 1000 m² de superficie.
- S = Superficie del terreno en m²
- Rmq = Rendimiento total de las máquinas utilizadas.
- Rp = Rendimiento total del personal que realiza explanaciones manuales.
- E = Eficiencia del personal atendiendo al régimen de trabajo seleccionado.



Los costes de las explanaciones serán calculados de la misma forma que para los trabajos de excavación.

4.2.3. Compactación

Cuando se trata de desarrollar un trabajo de obra, siempre es aconsejable realizar un trabajo previo de compactación del terreno. La compactación del suelo es un proceso mediante el cual se consigue aglomerar con mayor insistencia las partículas que conforman la superficie, consiguiendo una reducción del vacío y, de esta forma, un terreno mucho más sólido y viable para el trabajo de obra que se vaya a desarrollar. (ConstruccionesVale, 2019)

El cálculo de tiempo será de la siguiente manera:

$$Tiempo = Tp + S/(Rmq + Rp * E)$$

- Tp = El tiempo de preparación se estima en 0.5 horas para los trabajos de compactación.
- S = Superficie del terreno en m^2
- Rmq = Rendimiento total de las máquinas utilizadas.
- Rp = Rendimiento total del personal que realiza compactaciones manuales.
- E = Eficiencia del personal atendiendo al régimen de trabajo seleccionado.

Los costes de los trabajos de compactación serán calculados de la misma forma que para los trabajos anteriores, cambiando únicamente que para estos trabajos no podrán añadirse materiales.

4.2.4. Transporte de tierra

En todos los trabajos de movimiento de tierras es casi indispensable el transporte de tierra de un lugar a otro.

El cálculo para los trabajos de transporte de tierra será de la misma forma que para los trabajos de excavación utilizando los rendimientos de las máquinas y de personal respectivos a estos trabajos. El rendimiento de las máquinas para transporte de tierra será:

$$Rmq = Cp * V$$

- Rmq = Rendimiento total de las máquinas utilizadas.
- Cp = Capacidad de carga.
- V = Velocidad máxima.

4.2.5. Losa museo

Por su uso habitual en gran parte de los acuartelamientos se va a seleccionar la losa museo (ver Figura 4) como uno de los trabajos que incluya la aplicación, aunque este pueda dividirse en otros trabajos más sencillos.

Se realizarán anteriormente cálculos de las diferentes losas a realizar para cada uno de los carros. Así se determinará el tipo de hormigón a utilizar, las dimensiones y el tipo de armadura a utilizar. Todos los cálculos se realizarán con el programa CYPECAD y se podrán observar con detalle en el Anexo II.

Este trabajo incluirá el cálculo de la losa de hormigón, tanto dimensiones como armado, así como los trabajos de encofrado y hormigonado. Quedarán excluidos de este trabajo, el trabajo de excavación previo y también el de explanación. En caso que fuera necesario deberán seleccionarse como un trabajo previo.

Se incluirán tres tipos de vehículos de combate: Leopardo 2E, BMR y Pizarro. Para ello se incluirán en CYPE3D como dos cargas lineales separadas la distancia entre cadenas/ruedas, equivalentes a las que soportaría la losa debido al peso de los vehículos en orden de combate. Los pesos y dimensiones que se utilizan han sido obtenidos de la web de defensa del Ejército de Tierra (Tierra, s.f.) y son los que se presentan en la Tabla 2.



Pedro Luis Tejero Gracia

Vehículo	Peso [Tm]	Dimensiones [m]
Leopardo 2E	62	9.67x3.75
BMR	15.4	6.15x2.5
Pizarro	26.3	6.83x3.15

Tabla 2. Características vehículos

La norma que se va a seguir para los cálculos de la losa será la norma del código técnico de la edificación EHE 08. Los cálculos se realizan sobre losas de 30 cm de espesor, y dimensionadas para que haya un metro de espacio por cada lado del vehículo. El hormigón utilizado en las losas será HA-30. Se considerarán unas cargas uniformes para una losa con categoría de uso para tráfico y aparcamiento de vehículos ligeros.

Para cálculos de presupuesto se obtendrán exportando el proyecto de CYPE3D al software “Arquímedes”, donde se tendrá en cuenta únicamente el coste de los materiales y no de la mano de obra.

Para la estimación de tiempo de ejecución se han obtenido los diagramas de Gantt de cada una de las losas. Estos se pueden consultar en el Anexo II.



Figura 4. Losa museo (Fuente: (Wikipedia, s.f.))



5. Desarrollo de la aplicación

En este apartado se describen todos los métodos utilizados para el correcto desarrollo de la aplicación, tanto los referidos al correcto funcionamiento y evitar posibles fallos, como los estéticos y visuales.

Para ello se va a llevar a cabo un análisis de algunos de los riesgos a evaluar con la finalidad de evitar potenciales errores que puedan repercutir en el uso de la aplicación.

Más adelante, se realizará un estudio de los diferentes procesos que se llevarán a cabo con la aplicación. Para ello se ha llevado a cabo un diagrama de casos de uso.

Posteriormente esto se complementará con un diagrama de actividad que servirá para entender cómo se comporta el sistema y cómo interactúa con el usuario. “El diagrama de actividad nos permite representar tanto las acciones secuenciales y concurrentes de un proceso, el flujo de datos producido/requerido por las acciones, como los participantes que interactúan en el proceso.” (Bernardi & Dranca, 2009)

Finalmente, se mostrará la interfaz de la aplicación, acompañada de una descripción de la aplicación y sus diferentes posibilidades de uso.

Tanto el código fuente de la aplicación, así como su estructura, está detallada en el Anexo III.

5.1. Análisis de riesgos

Antes de la realización de la aplicación se llevará a cabo un análisis de riesgos con objeto de evitar posibles fallos o imprevistos que puedan hacer que la aplicación funcione incorrectamente o no cumpla con lo previsto (ver Tabla 4). Se contemplarán tanto factores internos como puede ser fallos en el lenguaje de programación, como factores externos debidos al mal uso de la aplicación.

El criterio que se seguirá para la clasificación de los riesgos se detallará en la tabla de clasificación (ver Tabla 3) en base a los parámetros descritos a continuación.

Probabilidad:

- Baja: Es muy poco probable que ocurra. Seguramente el problema no va a suceder nunca. De 0% a 10% de probabilidad
- Media: El riesgo puede aparecer de vez en cuando. Probabilidad de 10% a 50% de que ocurra
- Alta: El riesgo va a aparecer seguramente en algún momento de su utilización. Probabilidad mayor del 50%

Impacto:

- Bajo: El riesgo no afecta prácticamente a nada, ni al uso de la aplicación, ni a los resultados obtenidos
- Medio: El riesgo afecta parcialmente al uso de la aplicación o a los resultados obtenidos
- Alto: El riesgo afecta gravemente al uso de la aplicación o a los resultados obtenidos

Clasificación de riesgos				
Probabilidad	Alta	Medio	Alto	Crítico
	Media	Bajo	Medio	Crítico
	Baja	Muy bajo	Bajo	Alto
		Bajo	Medio	Alto
		Impacto		

Tabla 3. Clasificación de riesgos (Fuente: elaboración propia)



Riesgo	Razón	Probabilidad	Impacto	Clase de riesgo	Acciones a realizar
La aplicación no funciona	Fallo en el código de programación	Baja	Alto	Alto	Realizar pruebas en un teléfono Android además de con otros simuladores desde el ordenador
La aplicación no sirve	La aplicación no da solución a los problemas establecidos inicialmente	Media	Medio	Medio	Hacer un seguimiento de lo que se espera de la aplicación, tanto por futuros usuarios, como por observación directa de una Unidad de Ingenieros
La aplicación no es precisa	Fallo en cálculos de la aplicación	Media	Medio	Medio	Obtener información de diferentes manuales y programas que dispongan de los datos necesarios para los cálculos requeridos
La aplicación es demasiado específica	Los medios utilizados son diferentes a los incluidos en la aplicación	Alta	Medio	Crítico	Añadir opciones para poder incluir datos por el usuario
La aplicación es difícil de utilizar	La aplicación es demasiado compleja. Es necesaria una formación específica para poder utilizarla	Bajo	Medio	Bajo	Utilizar una interfaz sencilla siguiendo el método "Poka-yoke" que evite que puedan cometerse



Pedro Luis Tejero Gracia

					errores
--	--	--	--	--	---------

Tabla 4. Análisis de riesgos (Fuente: elaboración propia)

Como se puede observar solo dos de los riesgos son de clase “crítico” o “alto”. Para el caso en el que la aplicación pueda ser demasiado específica, debido a que existen una gran variedad de maquinas y materiales, y no contenga medios con los que se quiera trabajar, se incluirá una opción en la que el usuario pueda incluir otros medios diferentes a los que ofrece la aplicación, de esta manera el impacto se reducirá y con el la clase riesgo será menor. De la misma manera, para evitar fallos en el lenguaje de programación y que la aplicación no funcione, se realizaran pruebas tanto en ordenador, como en dispositivos móviles, reduciendo así al mínimo la probabilidad de que suceda.

Para el resto de riesgos, aunque son de clases menores, se llevarán igualmente algunas acciones para reducir la probabilidad de que estos sucedan.

5.2. Modelo de casos de uso

El diagrama de casos de uso podemos ver los actores que interactúan con el sistema de información. En nuestro caso (ver Figura 5), nos muestra un único actor, que será el usuario que interactúa directamente con la App. La herramienta utilizada para su realización ha sido *Visual Paradigm*.

Los casos de uso que se pueden llevar a cabo, son representados dentro de los círculos, que pueden relacionarse entre ellos a través de relaciones tipo “include” o “extend”. Serán de tipo “include” cuando los procesos vayan a realizarse siempre, y de tipo “extend” cuando los procesos se realicen solo en algunas condiciones.

En la aplicación, el usuario tendrá los casos de uso principales “añadir trabajo” o “ver trabajos”. Al añadir los trabajos tendrá que ser posible los casos de “añadir maquina” y “añadir materiales” además de poder borrar un trabajo después de haber sido guardado. A continuación, se describen cada una de las interacciones:

- Añadir trabajo: El usuario podrá seleccionar de una lista cualquiera de los trabajos detallados anteriormente en el apartado “Trabajos”.
- Añadir máquina: El usuario podrá seleccionar de una lista cualquiera de las máquinas que necesite o introducirla manualmente.
- Añadir materiales: El usuario podrá seleccionar de una lista cualquiera de los materiales que necesite o introducirlos manualmente.
- Ver trabajos: La aplicación mostrará los trabajos que el usuario a introducido anteriormente.

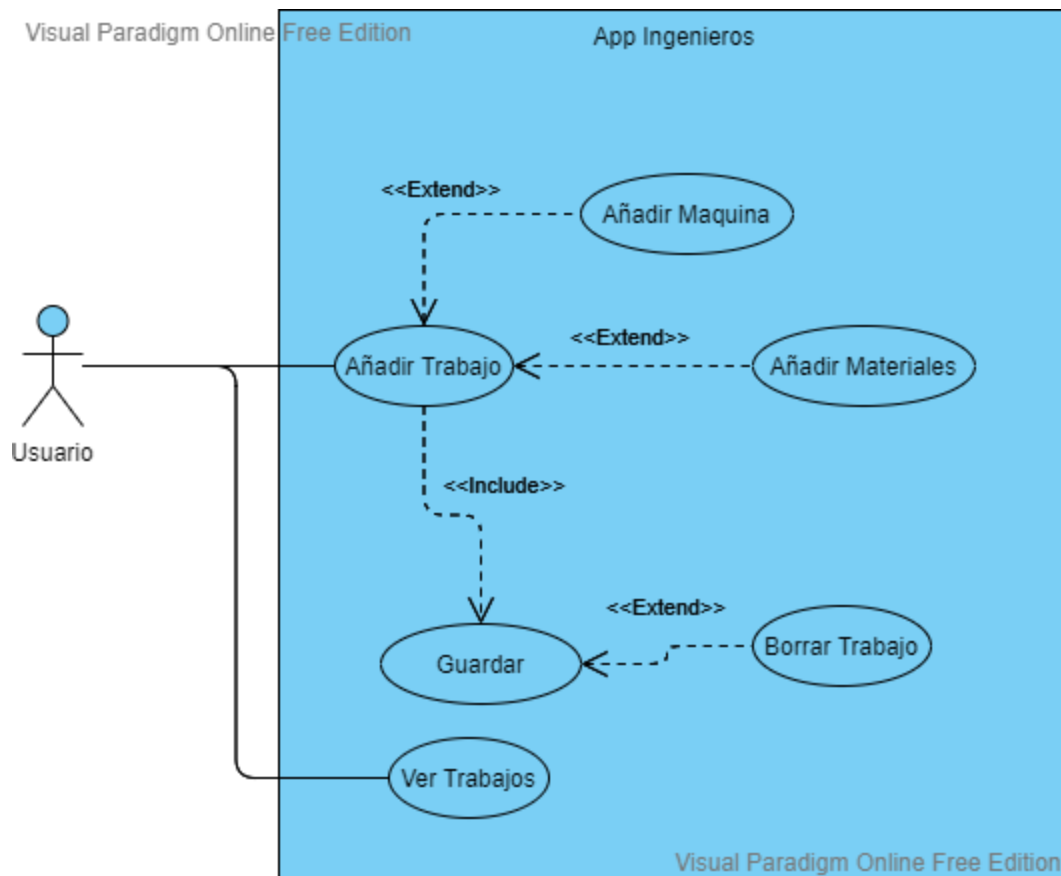


Figura 5. Diagrama de casos de uso (Fuente: Elaboración propia)

5.3. Diagrama de actividad

A continuación, detallaremos la secuencia lógica que debería seguir el usuario para poder hacer uso de la aplicación. Esto se mostrará gráficamente a través de un diagrama de actividad. El diagrama de actividad nos mostrará también los actores que interactúan, además de las actividades que realiza cada uno de ellos, y los flujos de información. Para la realización del diagrama de actividad se ha utilizado, igual que para el modelo de casos de uso, la herramienta Visual Paradigm.

Como podemos observar en el diagrama (ver Figura 6). El proceso comenzará cuando el usuario seleccione un tipo de trabajo. Posteriormente la App solicitará los datos relacionados con el trabajo. El usuario podrá decidir incluir en los trabajos diferentes máquinas o materiales mientras que la aplicación irá mostrando los datos de estos en pantalla. Finalmente, se guardará el trabajo y la aplicación calculará y mostrará la información de interés en la pantalla.

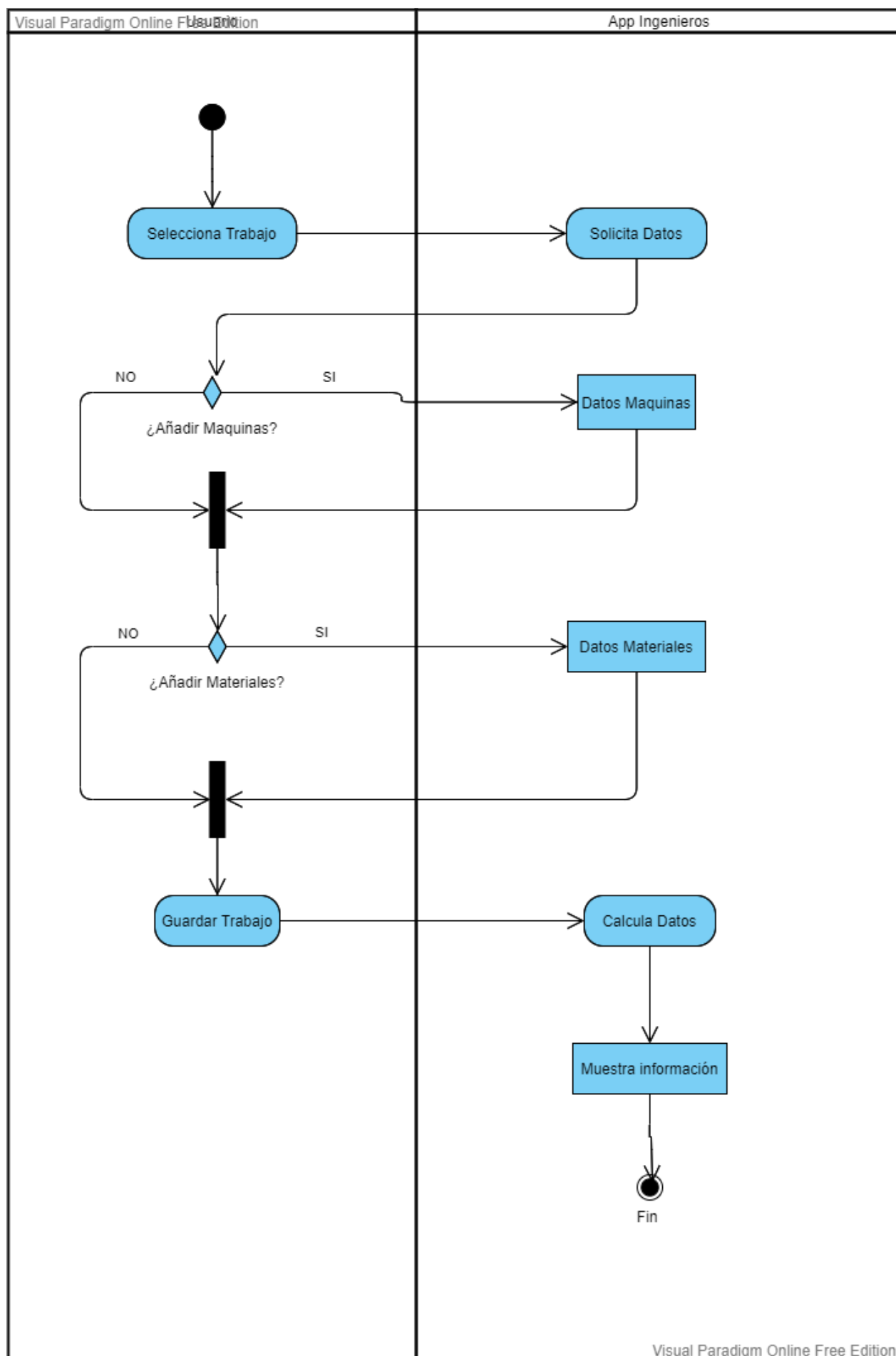


Figura 6. Diagrama de actividad (Fuente: elaboración propia)



5.4. Interfaz

En este apartado se mostrará el desarrollo de la aplicación para que esta sea útil y pueda ser utilizada por cualquiera.

La interfaz con el usuario deberá ser clara y sencilla, no debiendo ser necesario conocimientos previos para su manejo.

Para una identificación sencilla de la App se ha elegido la torre del emblema de Ingenieros sobre un fondo negro como icono. El nombre de la aplicación será “Trabajos de Ingenieros”. (Ver Figura 7)

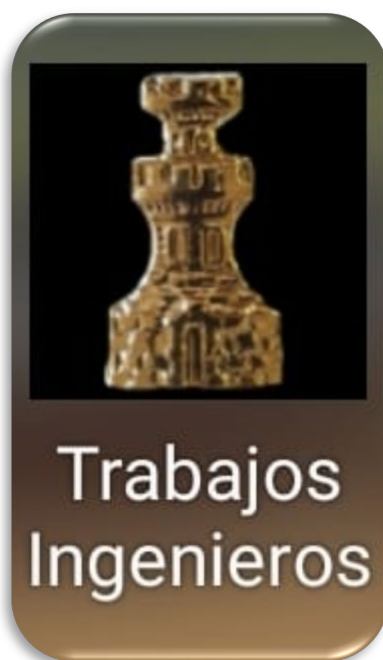


Figura 7. Icono y nombre de la App (Fuente: elaboración propia)

5.4.1. Pantalla principal

Incluirá una pantalla principal (ver Figura 8) donde se podrá ver reflejada la información más relevante de los trabajos que se vayan añadiendo. En el primer desplegable seleccionaremos el tipo de trabajo entre los tipos: excavación, explanación, compactación, transporte de tierra y losa museo. Cuando hayamos seleccionado esto podremos pulsar el botón de la parte inferior, “Añadir Trabajo”.



Figura 8. Pantalla principal (Fuente: elaboración propia)

La aplicación nos llevará a la siguiente pantalla, que será diferente dependiendo del trabajo que hayamos seleccionado, y requerirá que se introduzcan los datos necesarios. A continuación, se va a explicar que nos vamos a encontrar al seleccionar cada uno de los trabajos.

5.4.2. Introducción de datos

Al seleccionar el trabajo de “excavación” nos llevará a una pantalla donde deberemos introducir un identificador, el volumen a excavar, número de personal que trabaja en la obra, el régimen de trabajo y tipo de suelo (ver Figura 9). A continuación, se va a detallar cada uno de los apartados.

Identificador, es el nombre que le queramos dar al trabajo. Podrá ser cualquiera y servirá únicamente para identificar el trabajo cuando tengamos varios trabajos del mismo tipo. Es decir, si tenemos 2 trabajos de excavación, uno podrá ser un “pozo de tirador” y otro un “abrigo de pelotón”.

Volumen de tierra a excavar, será aproximado y sus unidades estarán en m^3 .

Número de personal, son las personas que van a trabajar en la obra sin utilizar máquinas, es decir, con herramientas de mano, ya sean picos, palas, azadas, carretillas u otras herramientas similares.



Pedro Luis Tejero Gracia

Régimen de trabajo, es el número de horas que el equipo va a estar trabajando al día. Podrán seleccionarse 3 tipos de regímenes de trabajo, de 8 horas, 12 horas y 24 horas. Dependiendo del régimen elegido, se aplicarán factores de eficiencia para aproximar al máximo el tiempo de ejecución como se ha explicado en el apartado 4.3.

Tipo de suelo, podrá ser, de terreno más blando a más duro: tierra suelta, arenoso, compacto, dureza media o suelo duro. Para cada uno de los tipos de suelo elegidos se tendrá en cuenta diferentes eficiencias.



Figura 9. Pantalla trabajo excavación (Fuente: elaboración propia)

Para los trabajos de “explanación”, “compactación” y “transporte de tierra” la pantalla de introducción de datos será similar, cambiando algunos de los datos a introducir. Para “explanación” en vez de volumen, el usuario deberá introducir la superficie. Para “transporte de tierra” deberá indicarse además la distancia.

El trabajo de “losa museo” se mostrará mas adelante.

5.4.3. Añadir maquinas y materiales

Seguidamente el usuario podrá añadir máquinas y añadir materiales al trabajo seleccionado. Se incluyen dos botones en la pantalla, cada uno con su respectiva función y dos ventanas en las que se irá mostrando la información de los elementos que se añadan.



Pedro Luis Tejero Gracia

Para añadir una máquina, deberemos hacer clic en el botón “Añadir Máquina” y nos aparecerá una nueva pantalla (ver Figura 10). Aquí, tendremos la opción de seleccionar del desplegable la máquina que vayamos a utilizar y la cantidad. Al pulsar el botón de aceptar se mostrará en la ventana la máquina añadida. Si queremos añadir más máquinas, podremos hacerlo de la misma manera pulsando de nuevo al botón “Añadir Máquina”. Si la máquina que queremos utilizar no aparece en el desplegable, tendremos la opción de añadirla nosotros mismos pulsando el botón de “Añadir otra máquina”. Aparecerá una nueva pantalla donde tendremos que indicar el modelo, así como su rendimiento, consumo y cantidad de máquinas que vamos a incluir, en caso de que vayan a ser más de una.

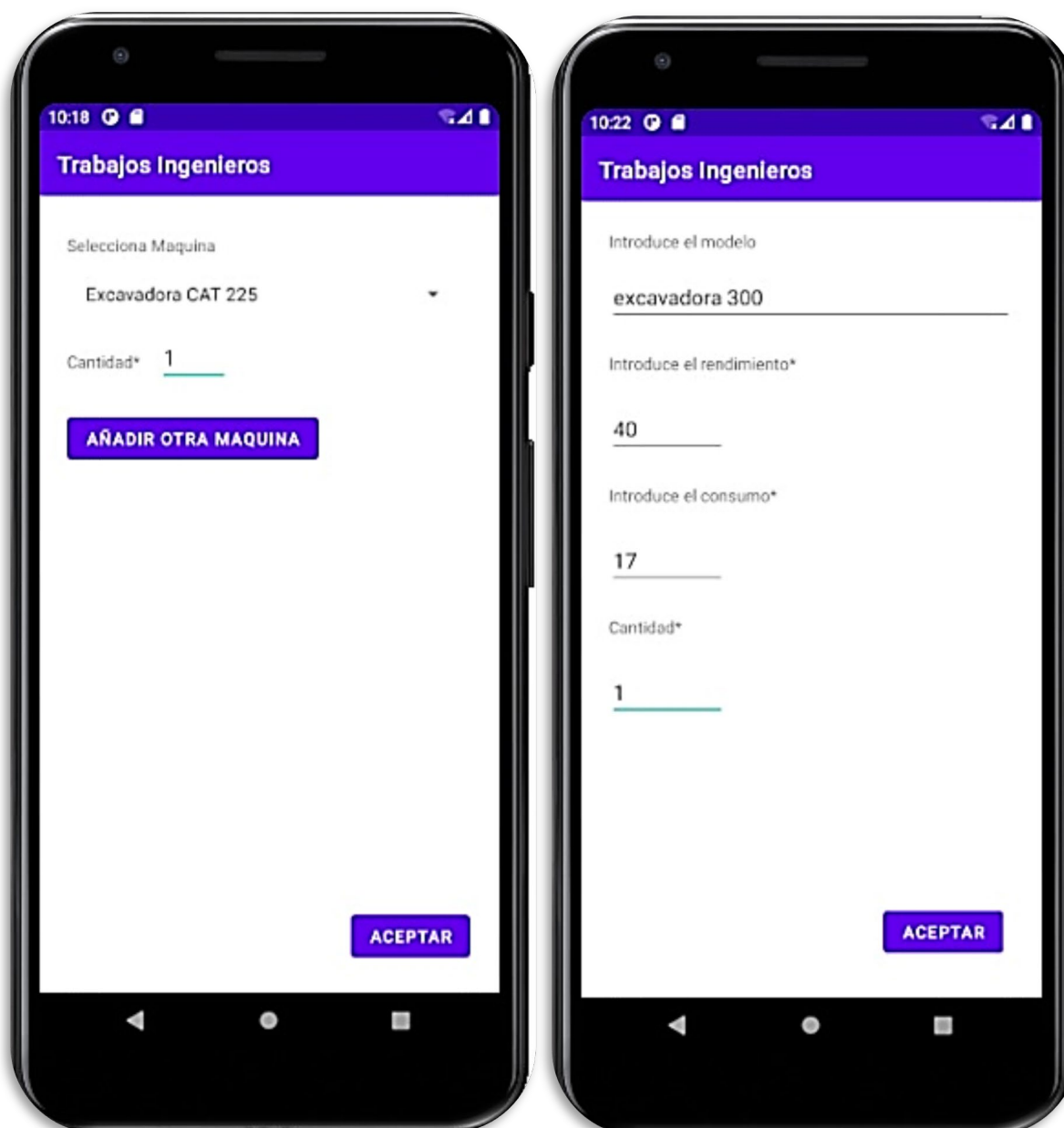


Figura 10. Pantallas para añadir maquinas (Fuente: elaboración propia)

De la misma manera podremos añadir materiales que queramos utilizar en la obra, ya sea madera para reforzar las excavaciones, áridos que nos sirvan como base para construir encima una losa de hormigón o cualquier otro que se vaya a necesitar y queramos incluir. Si queremos añadir cualquier otro material que no aparezca en el desplegable, se podrá hacer de manera similar a añadir una máquina nueva. Deberemos indicar el material a añadir, su coste y la cantidad. (ver Figura 11)

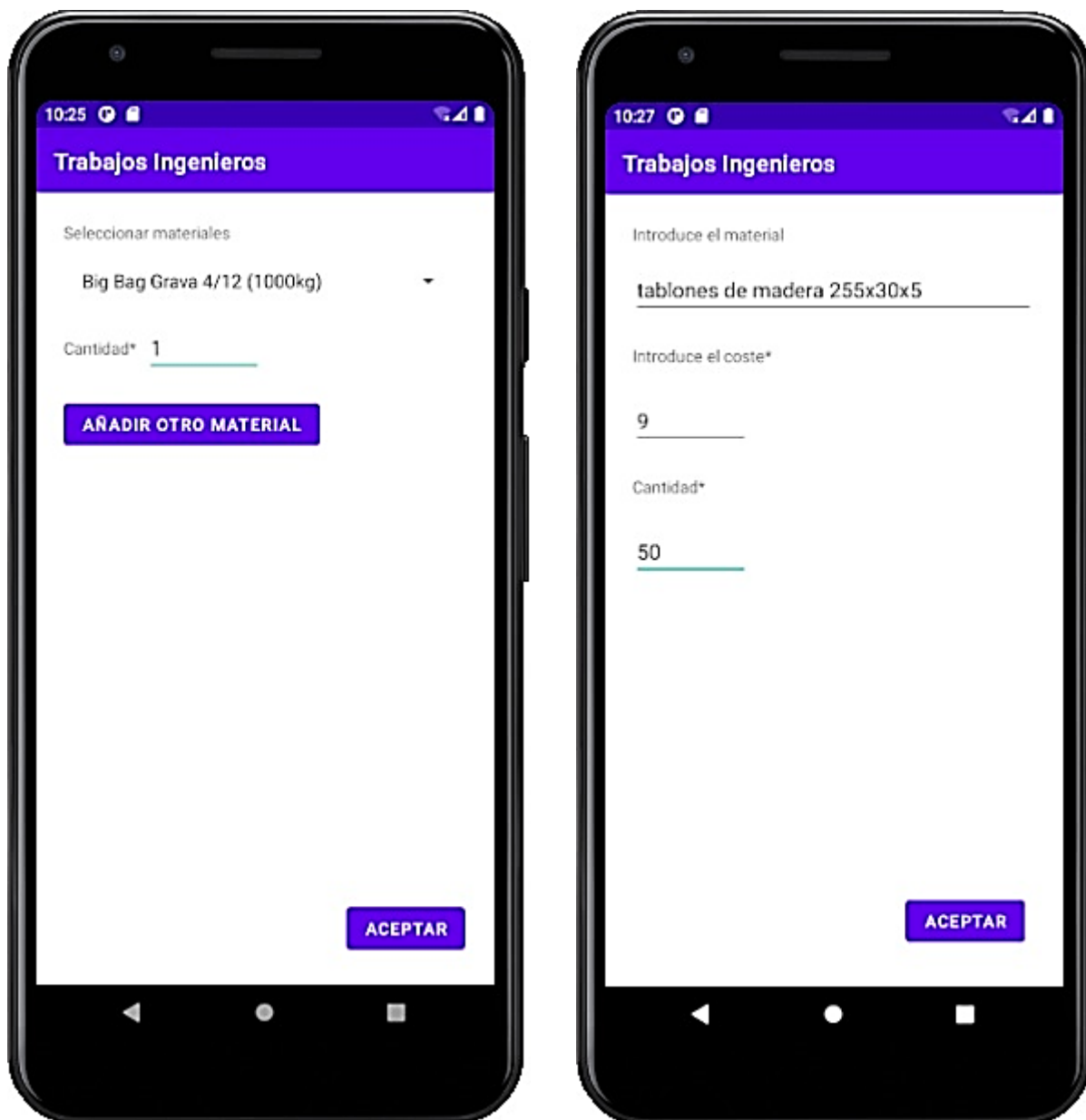


Figura 11. Pantallas para añadir materiales (Fuente: elaboración propia)

Una vez añadidas las máquinas y materiales podremos observar el gasto de combustible y el coste de los materiales que nos irán apareciendo en sus respectivas ventanas (ver Figura 12). En la parte inferior podremos ver el gasto total. Si en algún momento queremos eliminar alguno de estos, solamente tendremos que pulsar el botón “Borrar” que se encuentra a la derecha de la máquina o material que queramos borrar.

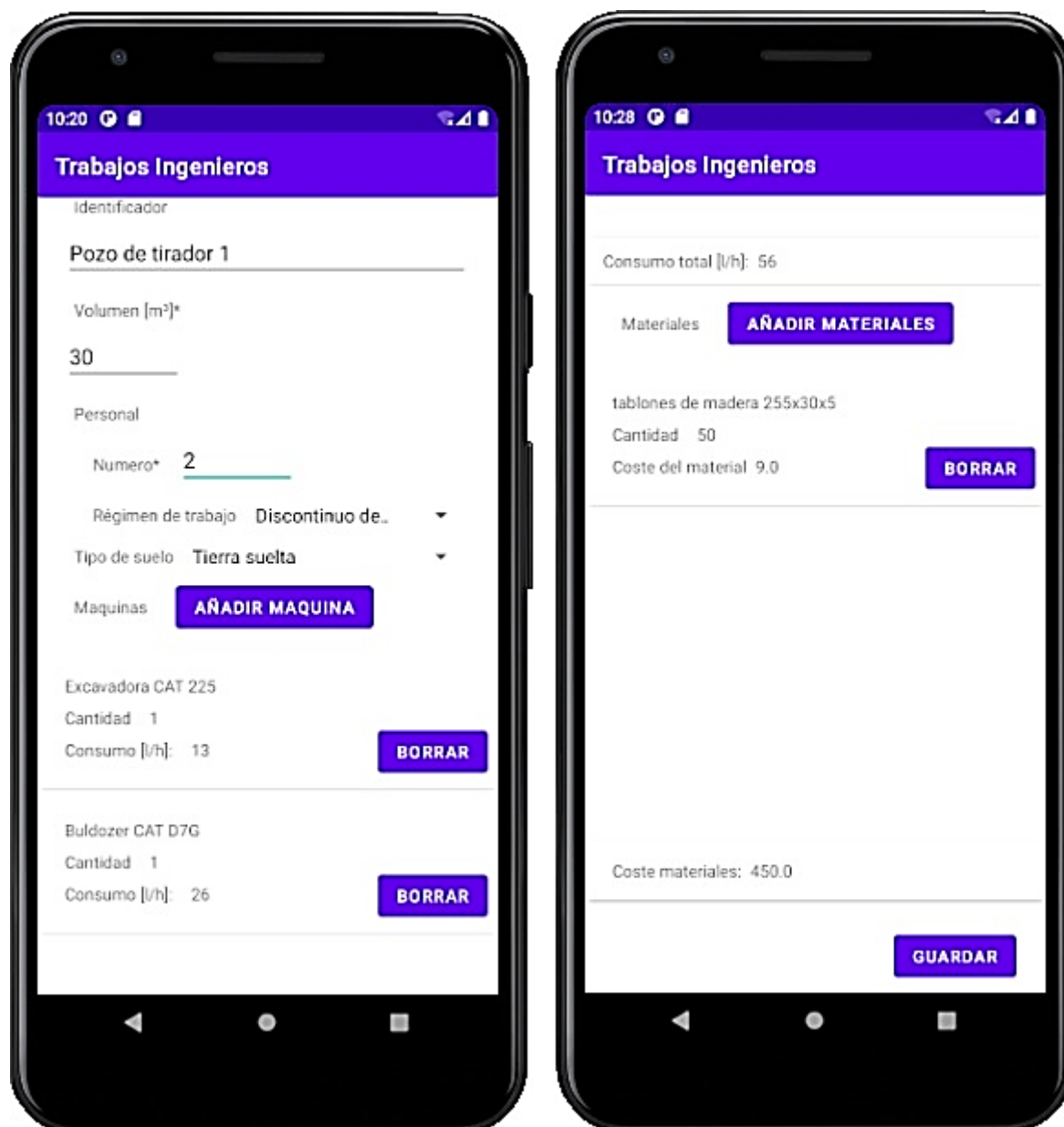


Figura 12. Ventanas de información máquinas y materiales (Fuente: elaboración propia)

Una vez rellenado todos los campos y seleccionado las máquinas y materiales que se vayan a utilizar podremos usar el botón de “Guardar” para volver a la pantalla principal, donde aparecerá la información en la ventana y podremos volver a seleccionar otro trabajo.

5.4.4. Losa museo

Si en la pantalla principal seleccionamos “loza museo” la pantalla que aparece es completamente diferente (ver Figura 13). Únicamente, deberemos seleccionar en el desplegable el tipo de vehículo que se va a exponer encima de la losa y, como en los casos anteriores, el identificador que se quiera asignar. Los cálculos asociados a estos vehículos han sido efectuados anteriormente con el software “CYPECAD” detallados más adelante. Una vez que se pulse el botón “Guardar” el trabajo quedará guardado y se mostrará la pantalla principal con la información de los trabajos en la ventana correspondiente.



Figura 13. Pantalla trabajo losa museo (Fuente: elaboración propia)

5.4.5. Muestra de los resultados

Como se puede observar en la imagen (ver Figura 14), en la ventana de la pantalla principal, se irá mostrando la información de interés de cada trabajo que podremos observar deslizando la pantalla arriba y abajo en caso de que no puedan mostrarse todos a la vez en la pantalla. En la parte inferior de esta ventana, se irá sumando los parámetros de tiempo, combustible y coste.

Para eliminar cualquiera de los trabajos que se han creado, bastará con pulsar el botón “Borrar” que se encuentra dentro de la tarjeta asociada al trabajo y los datos se actualizarán automáticamente.



Figura 14. Pantalla principal con trabajos añadidos. (Fuente: elaboración propia)



6. CONCLUSIONES

Este apartado se dedica a las conclusiones finales del proyecto, extraídas durante todo el proceso del mismo. En él, se detalla el objetivo de crear una aplicación en Android OS para la ayuda de los trabajos de organización del terreno en las Unidades de Ingenieros del Ejército de Tierra.

Las Unidades de Ingenieros del Ejército de Tierra encargadas del apoyo a otras armas, con la organización del terreno en una gran parte de ocasiones, mediante trabajos de movimiento de tierras, necesitan de un método rápido y eficaz que les permita, en cualquier momento y lugar saber algunos datos referentes a los trabajos que pretendan llevar a cabo.

Con la situación actual en la que cada uno de nosotros dispone de un teléfono smartphone, la mayor parte de ellos con sistema operativo Android y al no encontrarse ninguna aplicación ya desarrollada que sea de utilidad, se propone la realización de una aplicación capaz de mostrar instantáneamente algunos de estos datos.

Después de estudiar los tipos de trabajos llevados a cabo por Unidades de Ingenieros, ante la dificultad de poder seleccionar trabajos demasiado específicos por la infinitud de posibilidades, se han seleccionado algunos que se repiten normalmente. Los trabajos incluidos en la aplicación son: excavación, explanación, compactación, transporte de tierra y losa museo. Para cada uno de estos trabajos se calcula el tiempo de ejecución en base a rendimientos, extraídos de algunos fabricantes y manuales de construcción, y el coste asociado a cada uno. También determina consumos de combustible, lo que facilitará en gran medida la logística evitando transportar mayor combustible del necesario, lo que conlleva incrementos en los costes de ejecución. Permite, además, para el caso de las "losas museo", dar información específica de la construcción de esta para diferentes vehículos calculados previamente, facilitando y agilizando el proceso, estos datos son las dimensiones, el número de hierros y su diámetro.

Para el desarrollo de la aplicación se expone un análisis de riesgos, en el que se detallan algunas acciones llevadas a cabo para minimizar el impacto de los riesgos, encontrando como clase de riesgos mas altos los de "fallos de funcionamiento" por errores en la programación y "aplicación demasiado específica" por no incluir todos grandes cantidades de medios a utilizar ya sean maquinas o materiales a los que se aportan las acciones de realización de pruebas y añadir opciones para que el usuario pueda incluir medios.

Para poder desarrollar una aplicación sencilla se ha llevado a cabo un diagrama de casos de uso que permite ver las relaciones entre el usuario y la aplicación y otro de actividad en el que se detalla el proceso a seguir para añadir un trabajo.

En el resultado final es una aplicación que da solución a los problemas que se han planteado para las Unidades de Ingenieros. Permite añadir tantos trabajos como se desee, facilitando las tareas de organización para el usuario. Para diferenciar unos trabajos de otros se muestra un identificador para cada trabajo asignado previamente por el usuario. En la aplicación es posible incluir en cada uno de los trabajos las máquinas y los materiales que vayan a utilizarse, ya sean predeterminados por la aplicación o incluidos por el usuario a través de una de las funciones específicas. Después de seleccionar los trabajos a realizar e introducir los datos que se solicitan la aplicación calculará y mostrará en pantalla de manera clara la información que se ha descrito anteriormente.

Por lo tanto, es una herramienta de uso sencillo, que se maneja a través de una interfaz intuitiva y expone los datos más significativos de cada trabajo que vaya agregándose. A pesar de ello, el usuario deberá tener conocimientos en el ámbito de Ingenieros en el cual se vayan a desarrollar los trabajos. Por lo tanto, se recomienda su uso junto con manuales específicos del Ejército de Tierra, especialmente el manual de "Organización del terreno para pequeñas Unidades".

No obstante, aunque la aplicación ya da solución a las necesidades planteadas, es posible desarrollar futuras versiones que ofrezcan mejores resultados. Para ello en el apartado siguiente se proponen algunos aspectos que podrían desarrollarse.



6.1. Trabajo Futuro

En este apartado se va a detallar algunos de los trabajos que podrán llevarse a cabo más adelante con la finalidad de una mejora de la aplicación. Algunos de ellos están relacionados con la comodidad y facilidad de gestión por parte del usuario mientras que otros simplemente se basan en la realización de una base de datos para ampliar algunas posibilidades como el uso de más máquinas y materiales.

Uno de los aspectos en los que se puede avanzar en mejorar la aplicación será agregar otros trabajos, más allá de los de movimiento de tierras, que puedan ser llevados a cabo para la organización del terreno como, por ejemplo, la tala de árboles, la construcción de muros, o estructuras, para construcción de bunkers u otras finalidades. Para ello únicamente será necesario hacer un cálculo de los rendimientos e incluirlos en un nuevo trabajo siguiendo el mismo patrón que para el resto de trabajos. También será posible aumentar el número de máquinas y materiales de la misma manera.

El otro aspecto que podría ser desarrollado sería la creación de cuentas de usuario. De esta manera sería posible guardar los trabajos para ser visualizados desde otro lugar. Sin embargo, habría que tener en cuenta que la mayor parte de las veces que se vaya a utilizar la aplicación, será probable que no exista una conexión a internet, por lo que podría resultar de escasa utilidad.



7. REFERENCIAS BIBLIOGRÁFICAS

Bibliografía

AcademiaAndroid, 2014. *AcademiaAndroid*. [En línea]

Available at: <https://academiaandroid.com/ide-entornos-integrados-de-desarrollo-para-android/>
[Último acceso: 21 10 2021].

AcademiaAndroid, 2014. *AcademiaAndroid*. [En línea]

Available at: <http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
[Último acceso: 21 10 2021].

Bernardi, S. & Dranca, L., 2009. *Sistemas de información para la dirección*. Zaragoza: Centro Universitario de la Defensa.

Boletín de Tierra, 2021. *Ejército de Tierra*. [En línea]

Available at:

https://ejercito.defensa.gob.es/noticias/2021/02/8321_zapadores_y_artilleros_adiestramiento_operacion_libre_hidalgo.html

[Último acceso: 2021 10 2021].

ConstruccionesVale, 2019. *ConstruccionesVale*. [En línea]

Available at: https://www.construccionesvale.com/la-importancia-de-la-compactacion-del-terreno_fb24684.html

[Último acceso: 14 10 2021].

DiarioCordoba, 2020. *DiarioCordoba*. [En línea]

Available at: <https://www.diariocordoba.com/cordoba-ciudad/2020/07/13/batallon-zapadores-x-imparte-instruccion-35977673.html>

[Último acceso: 20 10 2021].

Foxtand, s.f. *demolicionesfe*. [En línea]

Available at: <https://demolicionesfe.cl/movimiento-de-tierra/clasificacion-excavaciones/>

[Último acceso: 13 09 2021].

FUTER, 2018. *Twitter*. [En línea]

Available at: https://twitter.com/futer_et/status/978197642365886464

[Último acceso: 20 10 2021].

Infodefensa, 2020. *Infodefensa*. [En línea]

Available at: <https://www.infodefensa.com/texto-diario/mostrar/3125820/ejercito-cuenta-nueva-zona-combate-urbano-san-gregorio>

[Último acceso: 20 10 2021].

MADOC, 2005. *Organización del terreno para pequeñas unidades*. s.l.:MADOC.

MADOC, 2020. *Táctica y logística de Ingenieros AGM-CM-013*. MADOC ed. s.l.:s.n.

MaquinariasPesadas, 2015. *MaquinariasPesadas*. [En línea]

Available at: <https://es.slideshare.net/CorporacionesAmcSac/material-tablasrendimientoconsumocombustiblecostomaquinariapesada>

[Último acceso: 14 06 2021].

Polpaico, 2011. *Manual del constructor*. s.l.:s.n.

Tierra, B. d., 2020. *Ejército de Tierra*. [En línea]

Available at: <https://ejercito.defensa.gob.es/noticias/2020/10/8142-zona-combate-urbanizada.html>



Pedro Luis Tejero Gracia

[Último acceso: 20 10 2021].

Tierra, E. d., s.f. *Ejército de Tierra*. [En línea]

Available at: <https://ejercito.defensa.gob.es>

[Último acceso: 06 08 2021].

Wikipedia, s.f. *Wikipedia*. [En línea]

Available at: [https://es.wikipedia.org/wiki/Centurion_\(carro_de_combate\)](https://es.wikipedia.org/wiki/Centurion_(carro_de_combate))

[Último acceso: 20 10 2021].

ANEXOS

Anexo I. Tablas de rendimientos

A continuación, se detallan las tablas de rendimientos utilizados en los cálculos.

EXCAVACIONES MANUALES					
TIPO	TIERRA SUELTA	TIERRA VEGETAL ARCILLOSA O ARENOSA	SUELO COMPACTO	DUREZA MEDIA	SUELO DURO
EN EXPLANACION	2.5	1.7	0.7	0.6	0.3
EN POZO O ZANJA	-	1.2	0.6	0.5	0.3

Tabla 5. Excavaciones manuales rendimientos (Fuente: (Polpaico, 2011))

Para terreno duro será necesaria la utilización de explosivo. Se añadirán a los costes 0.4 kg de explosivo (34€ por kg) por cada m³ y 0.4 m³ de cordón detonante (2€ por metro).

Rendimientos esparcimiento y compactación manualmente m ³ /h por persona					
	Transporte de tierra con carretilla	Esparcimiento	Apisonado a brazo medido después de consolidado		
			Tierra suelta	Escombros	Grava
Capas de 30 cm	1.9	2.6	2.5	1.4	1.2

Tabla 6. Rendimientos compactación manual (Fuente: (Polpaico, 2011))

Rendimiento de carga de tierra manualmente (Con pala)	
Clase de material	m ³ /h por persona
Terreno suelto	2.5
Dureza media	2.0
Suelo granular	1.7

Tabla 7. Rendimientos de carga manual (Fuente: (Polpaico, 2011))

EXCAVADORAS				
EQUIPO	POTENCIA (HP)	TIPO DE TRABAJO	RENDIMIENTO (m ³ /h)	CONSUMO PROMEDIO (l/h)
CAT 225	125	Tierra	130	13.13
		Mat. Heterogéneo	70	
		Roca con voladura	50	
CAT 235	195	Tierra	170	14
		Mat. Heterogéneo	100	
		Roca con voladura	70	

Tabla 8. Rendimientos de excavadoras (Fuente: (MaquinariasPesadas, 2015))

BULDOZER DE ORUGAS				
EQUIPO	POTENCIA (HP)	TIPO DE TRABAJO	RENDIMIENTO (m³/h)	CONSUMO PROMEDIO (l/h)
CAT D6D	140	Tierra	30	14.64
		Mat. Heterogéneo	30	
		Roca con voladura	20	
CAT D7G	200	Tierra	50	26.49
		Mat. Heterogéneo	40	
		Roca con voladura	30	

Tabla 9. Rendimiento bulldozer (Fuente: (MaquinariasPesadas, 2015))

MOTONIVELADORAS			
EQUIPO	POTENCIA (HP)	RENDIMIENTO* (m²/h)	CONSUMO PROMEDIO (l/h)
CAT 120 G	125	290	20.55
CAT 140 G	140	300	18.92

Tabla 10. Rendimientos motoniveladoras (Fuente: (MaquinariasPesadas, 2015))

*Para el rendimiento se escogerá el rendimiento menor de todos los tipos de trabajo

COMPACTADORES			
EQUIPO	POTENCIA	RENDIMIENTO (m²/h)	
COMPACTADOR CA-15 LISO	101	160	

Tabla 11. Rendimientos compactadores (Fuente: (MaquinariasPesadas, 2015))

CARGADORAS					
EQUIPO	POTENCIA (HP)	RENDIMIENTO* (m³/h)	CONSUMO PROMEDIO (l/h)	Capacidad del balde (m³)	Velocidad (km/h)
CAT 950	155	846	12	3.3	25
CAT 966	200	1053	17	4.2	39.5

Tabla 12. Rendimientos cargadoras (Fuente: (MaquinariasPesadas, 2015))

*Para el rendimiento se escogerá un promedio entre los rendimientos de los trabajos hasta 2300 metros

VOLQUETES				
EQUIPO	POTENCIA (HP)	CAPACIDAD (m³)	CONSUMO PROMEDIO (l/h)	VELOCIDAD MEDIA (km/h)

IVECO	294	10	30	80 (CAMINO ASFALTADO) 60 (CAMINO NO ASFALTADO)
-------	-----	----	----	--

Tabla 13. Rendimientos Volquetes (Fuente: (Maquinarias Pesadas, 2015))

Anexo II. Cálculos losa museo

Los cálculos para realizar las losas se realizarán con el programa “Cypecad”. Se seguirán las normas del Código Técnico de la Edificación (CTE), para hormigón la Instrucción de Hormigón Estructural (EHE-08) aunque esta vaya a ser derogada el 10 de noviembre de 2021 por el Código Estructural y para acero el CTE DB SE-A.

Se utilizará hormigón de la clase HA-30 con un coeficiente de cálculo de 1,5 y un acero de tipo B500 S para barras y B400 S para pernos. No se tendrán en cuenta acciones de viento ni sísmicas (ver Figura 15).

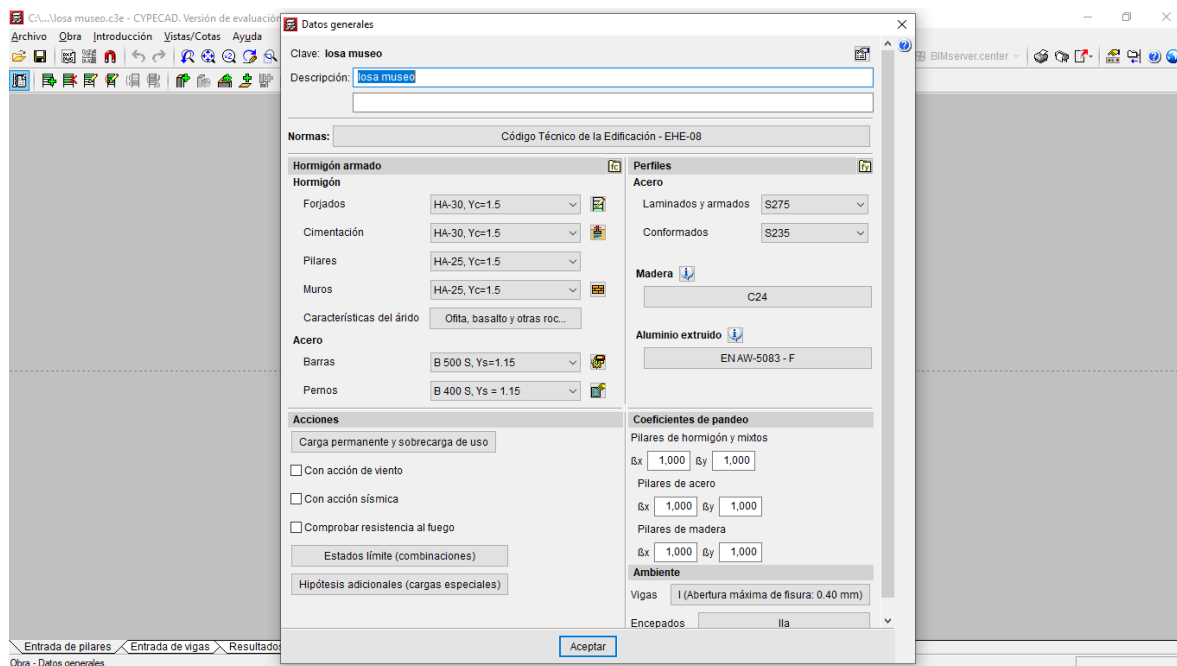


Figura 15. Datos generales CYPE3D (Fuente: elaboración propia)

Como hipótesis adicionales se tendrán en cuenta sobrecargas debidas a la categoría de uso E (Zonas de tráfico y de aparcamiento para vehículos ligeros). Se han incluido dos grupos (ver Figura 16), la cimentación donde se incluirá nuestra losa de hormigón y un forjado que únicamente se incluye para poder trabajar con el software, pero sobre el que no se tendrá en cuenta ningún tipo de carga ni se va a dibujar ningún forjado. Sobre la cimentación se incluirán, tal y como indica el “Documento Básico de la Seguridad Estructural Acciones en la Edificación” (DB SE-AE), una carga uniforme de 2 kN/m² y otra del mismo valor que sustituye a una carga concentrada de 20 kN para las losas.

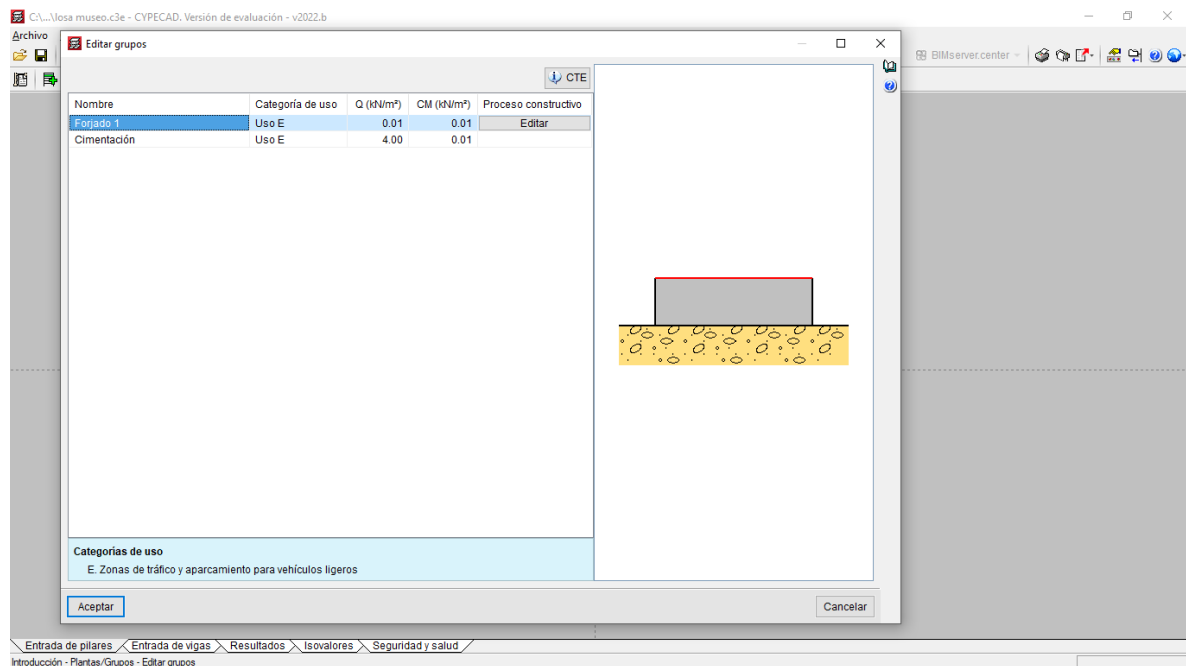


Figura 16. Datos grupos CYPE3D (Fuente: elaboración propia)

A continuación, se ha creado una losa con vigas no estructurales del tipo que se indica en la imagen (ver Figura 17) para posteriormente crear un paño, para losas apoyadas sobre el terreno de 30 cm de canto, donde poder dibujar las cargas de los vehículos que queremos colocar (ver Figura 18).

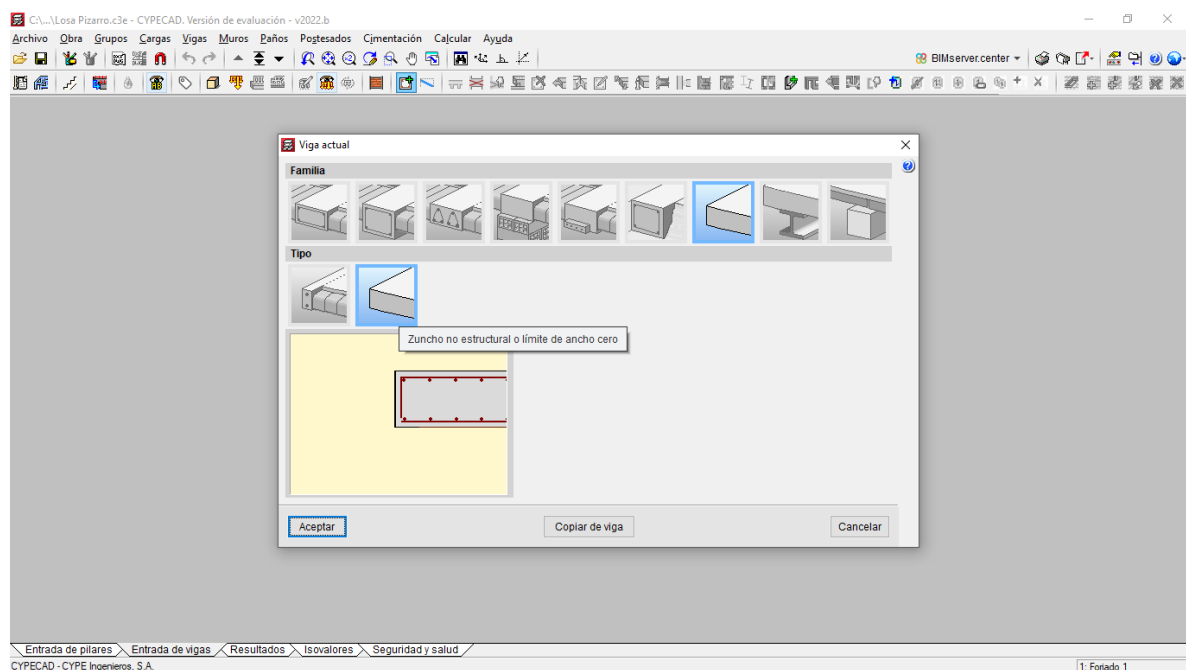


Figura 17. Tipo de vigas para losa (Fuente: elaboración propia)

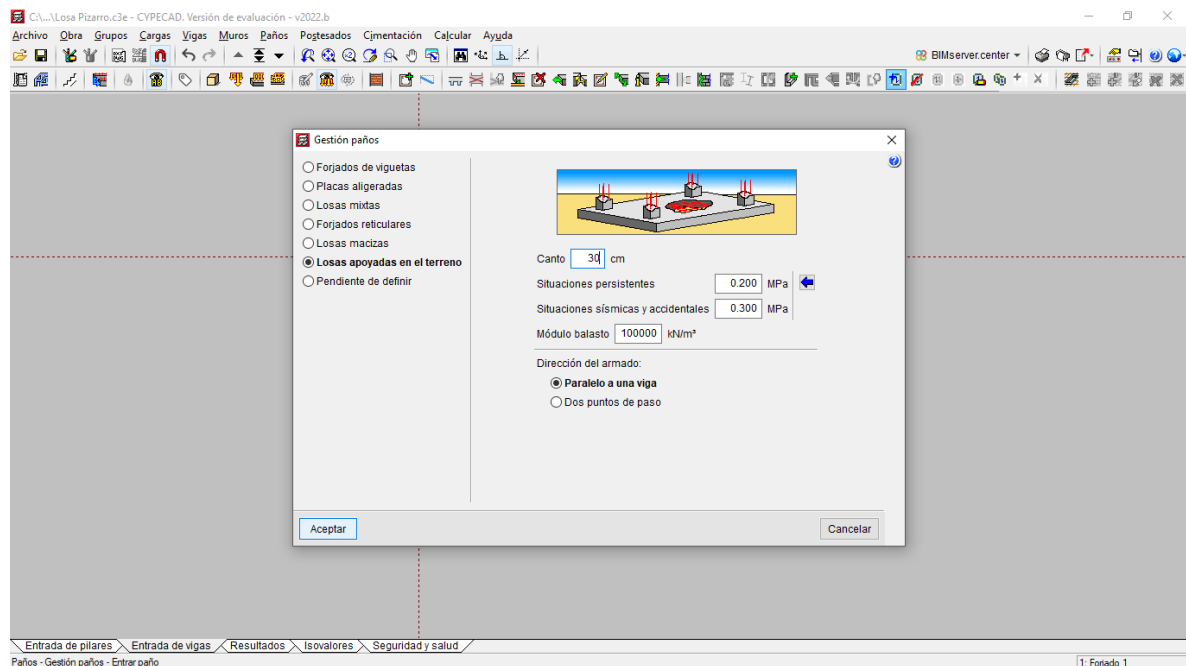


Figura 18. Datos paño para cálculos de losa (Fuente: elaboración propia)

A continuación, se detallarán para cada vehículo cuales son las cargas que han definido para cada uno de los vehículos y los resultados obtenidos tanto del armado del hormigón como del coste y duración tras haber exportado los trabajos al software “Arquímedes”.

Los resultados obtenidos del cálculo de la armadura para los tres vehículos fueron hierros de diámetro 12 con una separación de 30 cm (ver Figura 19, Figura 20, Figura 21) quedando de la siguiente forma:

Vehículo	Dimensiones losa[m]	Diámetro [mm]	Separación [cm]	Nº de hierros
Leopardo 2E	11.67x5.75	12	30	116
BMR	8.15x4.5	12	30	84
Pizarro	8.83x5.15	12	30	93

Tabla 14. Resultados armado losas (Fuente: elaboración propia)

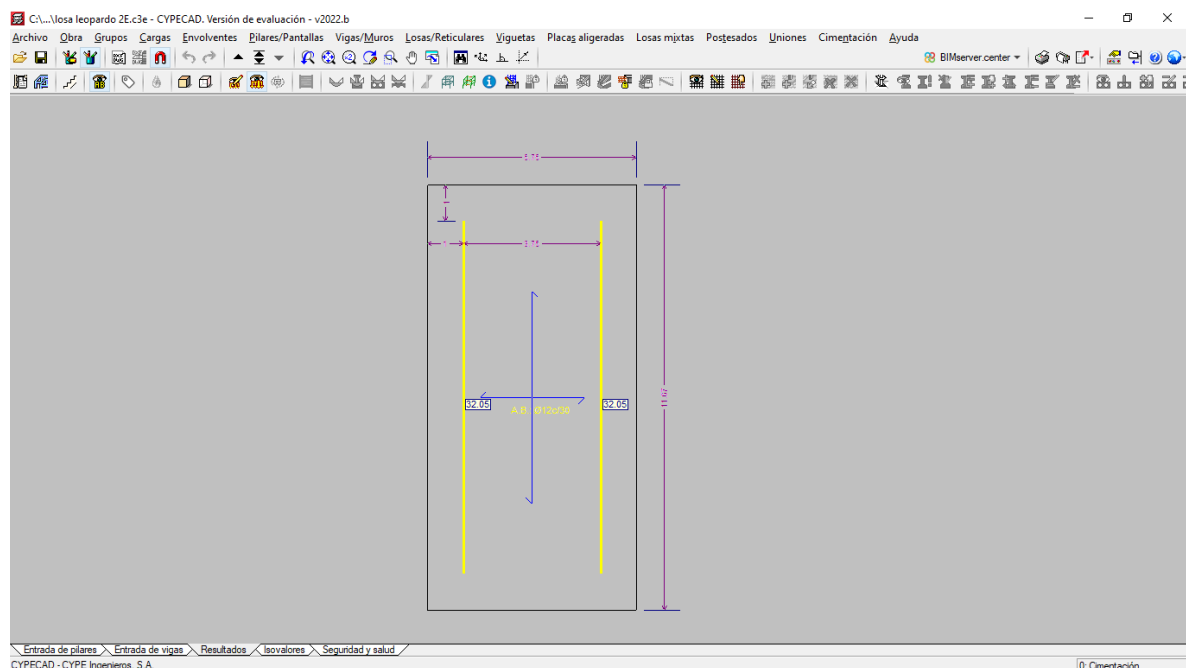


Figura 19. Cálculo armado losa Leopardo 2E (Fuente: elaboración propia)

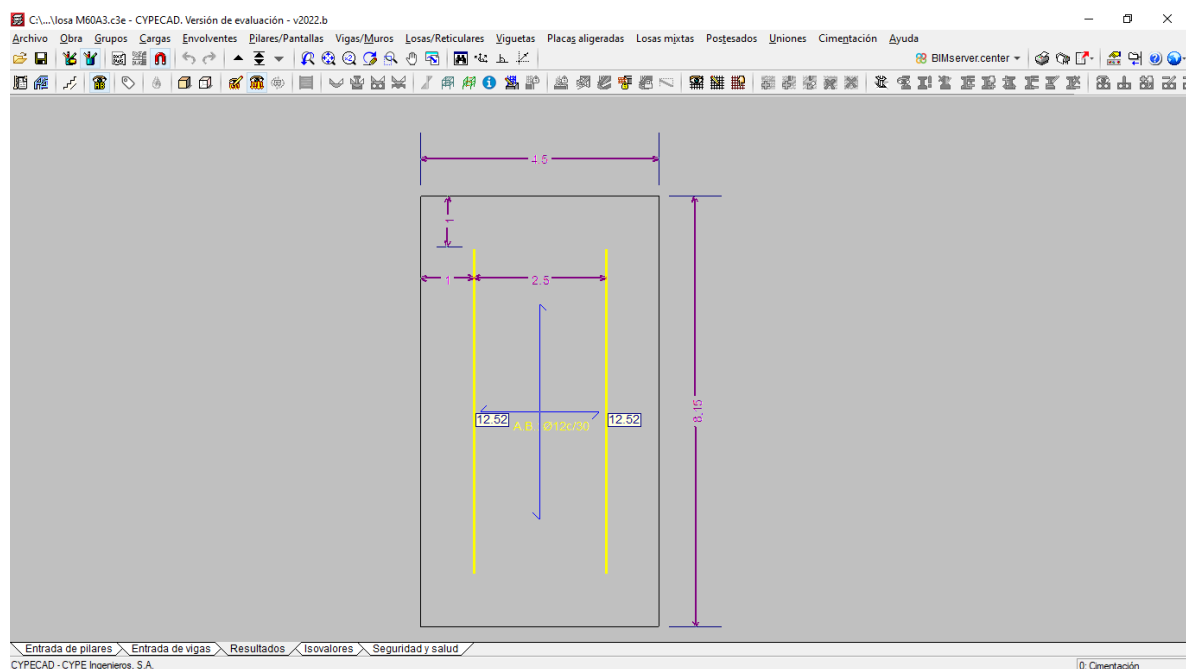


Figura 20. Cálculo armado losa BMR (Fuente: elaboración propia)

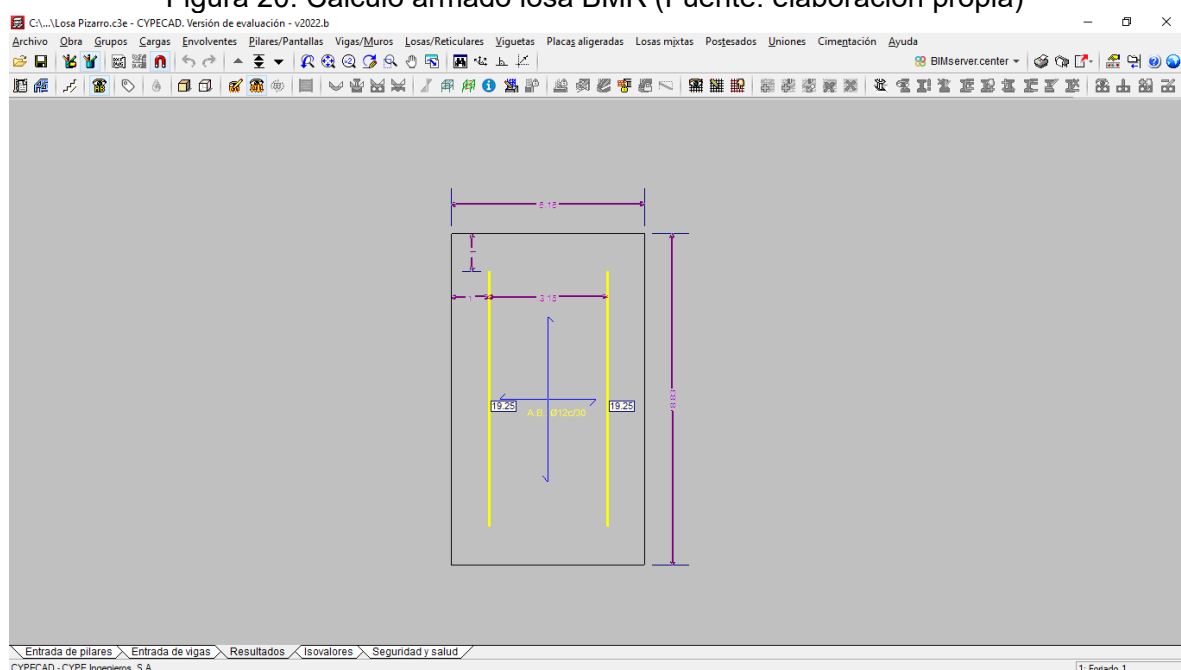


Figura 21. Cálculo armado losa Pizarro (Fuente: elaboración propia)

Como se ha expuesto anteriormente, para la elaboración de los costes se han exportado los trabajos a “Arquímedes” obteniendo los presupuestos para cada una de las losas (ver Figura 22, Figura 23, Figura 24). Para la implementación en la aplicación se han seleccionado únicamente los correspondientes a materiales y los derivados de la maquinaria debido a que para el Ejército los derivados de la mano de obra no son determinantes. En la Tabla 15 se puede observar cómo quedarán los costes para cada una de las losas.

Vehículo	Coste [€]
Leopardo 2E	2990
BMR	1648
Pizarro	2043

Tabla 15. Costes asociados a las losas de cada vehículo

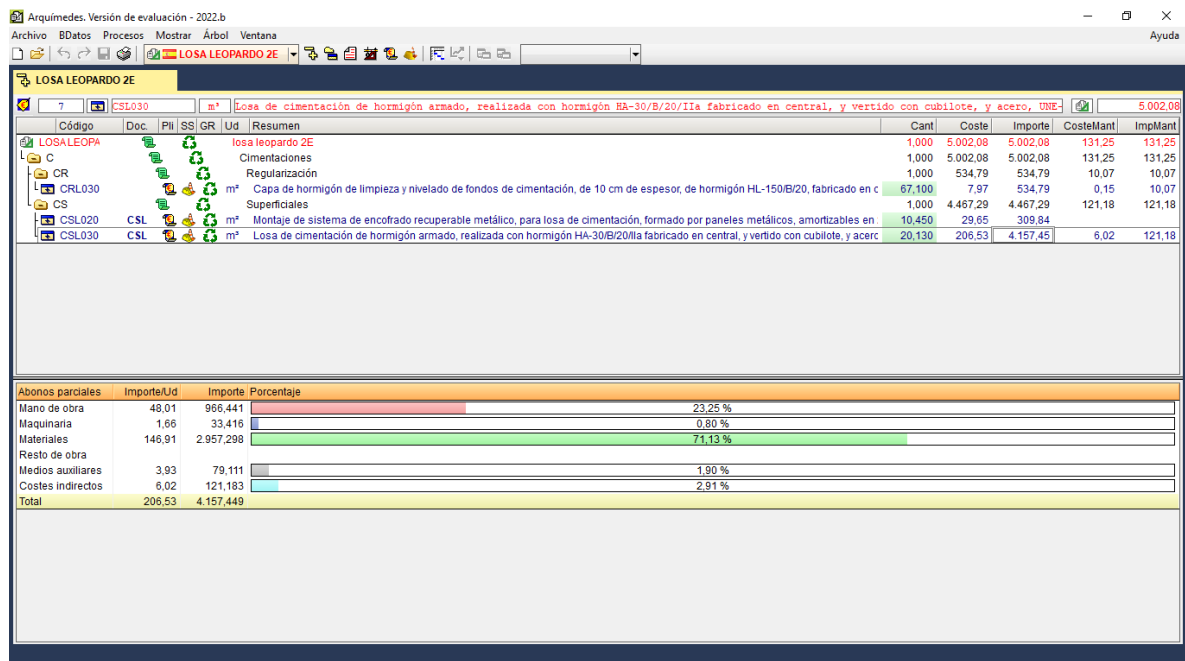


Figura 22. Presupuestos Arquimedes para losa Leopardo 2E (Fuente: elaboración propia)

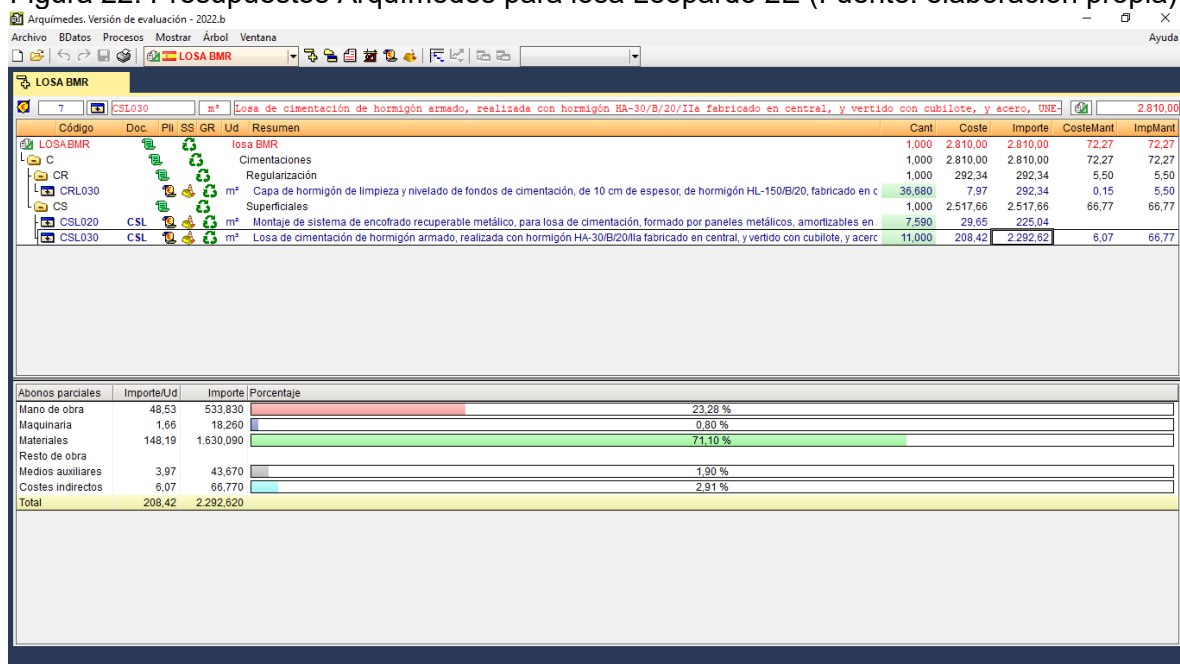


Figura 23. Presupuestos Arquimedes para losa BMR (Fuente: elaboración propia)

Archimedes. Versión de evaluación - 2022.b

ArchivoBDatosProcesosMostrarÁrbolVentanaAyuda

LOSA PIZARRO

17CSL030m²Losa de cimentación de hormigón armado, realizada con hormigón HA-30/B/20/I/a fabricado en central, y vertido con cubilote, y acero, UNE-3.454,01

Código	Doc.	Pli	SS	GR	Ud	Resumen	Cant.	Coste	Importe	CosteMant	ImpMant
CR						Regularización	1,000	362,40	362,40	6,82	6,82
CRL030					m²	Capa de hormigón de limpieza y nivelado de fondos de cimentación, de 10 cm de espesor, de hormigón HL-150/B/20, fabricado en central, y vertido con cubilote, y acero, UNE-45,470	7,97	362,40	0,15	6,82	
CS						Superficiales	1,000	3.091,61	3.091,61	82,79	82,79
CSL020	CSL				m²	Montaje de sistema de encofrado recuperable metálico, para losa de cimentación, formado por paneles metálicos, amortizables8,390	29,65	248,76			
CSL030	CSL				m²	Losa de cimentación de hormigón armado, realizada con hormigón HA-30/B/20/I/a fabricado en central, y vertido con cubilote, y acero, UNE-13,640	208,42	2.842,85	6,07	82,79	
mt07aco020					Ud	Separador homologado para cimentaciones.	5,000	0,15	0,75		
mt07aco010					kg	Acero en barras corrugadas, UNE-EN 10080 B 500 S, suministrado en obra en barras sin elaborar, de varios diámetros.	48,300	1,26	60,86		
mt08var050					kg	Alambre galvanizado para atar, de 1,30 mm de diámetro.	0,237	1,13	0,27		
mt10haf010r					m³	Hormigón HA-30/B/20/I/a, fabricado en central.	1,050	82,20	86,31		
mq06vib020					h	Regla vibrante de 3 m.	0,351	4,74	1,66		
mo043					h	Oficial 1ª ferrallista.	0,395	25,36	10,02		
mo090					h	Ayudante ferrallista.	0,593	23,64	14,02		
mo045					h	Oficial 1ª estructurista, en trabajos de puesta en obra del hormigón.	0,456	25,36	11,56		
mo092					h	Ayudante estructurista, en trabajos de puesta en obra del hormigón.	0,547	23,64	12,93		
%					%	Costes directos complementarios	2,000	198,38	3,97		

Abonos parciales	Importe/Ud	Importe	Porcentaje
Mano de obra	48,53	661,949	23,28 %
Maquinaria	1,66	22,642	0,80 %
Materiales	148,19	2.021,312	71,10 %
Resto de obra			
Medios auxiliares	3,97	54,151	1,90 %
Costes indirectos	6,07	82,795	2,91 %
Total	208,42	2.842,849	

Figura 24. Presupuestos Arquimedes para losa Leopardo 2E (Fuente: elaboración propia)

Se observan de cada uno de los diagramas de Gantt (ver Figura 25, Figura 26, Figura 27) para cada vehículo la duración en tiempo de cada una de las losas, que serán de 3 días habiendo seleccionado jornadas de 8 horas, por lo que el total de horas para cada losa será de 24 horas.

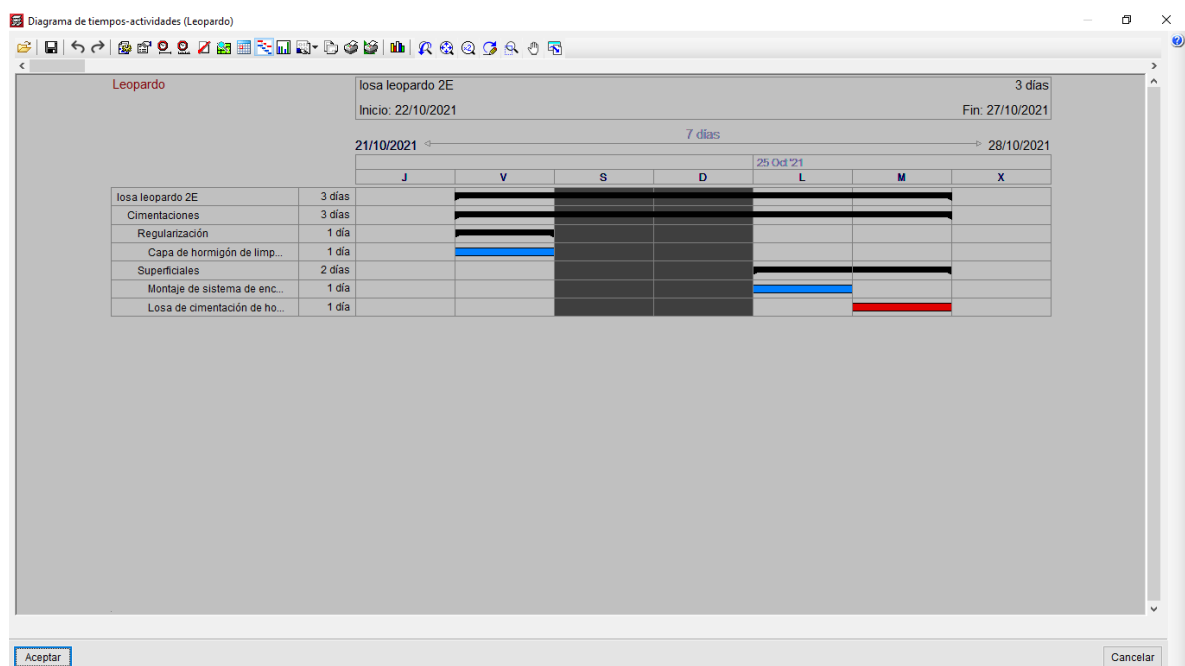


Figura 25. Diagrama de Gantt para losa Leopardo 2E (Fuente: elaboración propia)

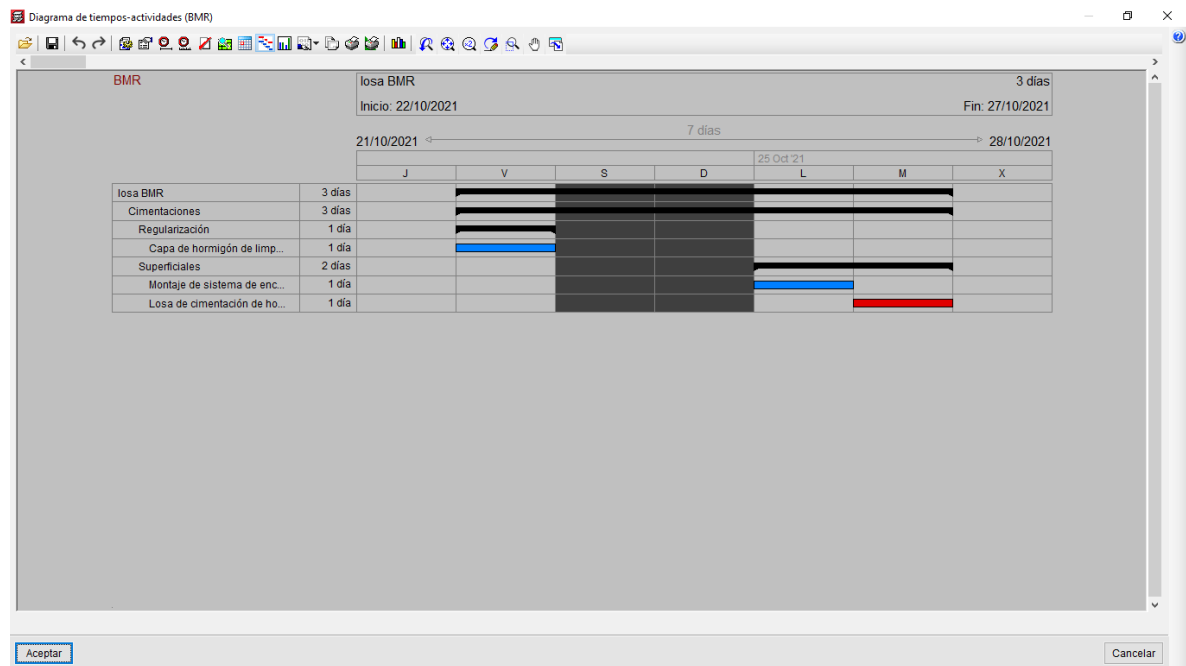


Figura 26. Diagrama de Gantt para losa BMR (Fuente: elaboración propia)

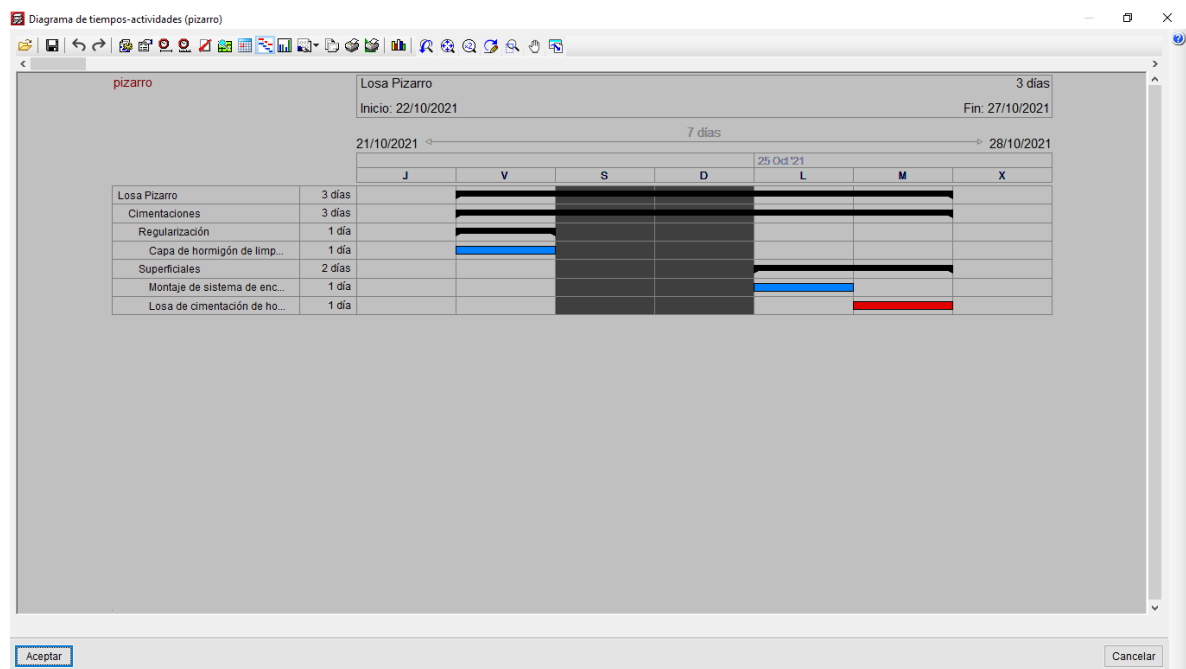


Figura 27. Diagrama de Gantt para losa Pizarro (Fuente: elaboración propia)

Anexo III. Código fuente

En este anexo se incluye el código fuente de la aplicación desarrollado en *Android Studio*. Las siguientes dos imágenes muestran como se ha organizado la aplicación, sus diferentes clases (ver Figura 28), así como sus layouts (ver Figura 29).

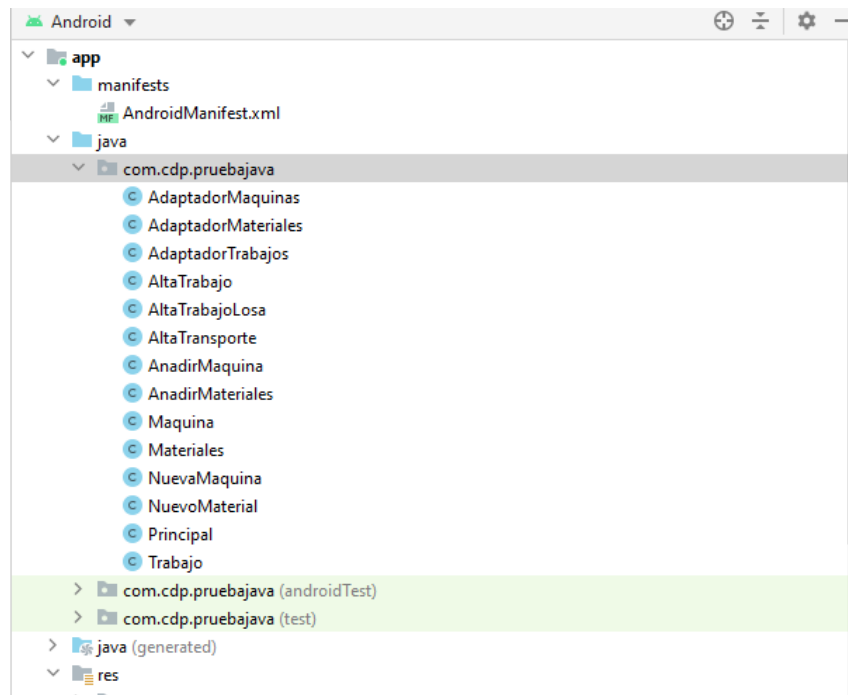


Figura 28. Clases aplicación *Android Studio* (Fuente: elaboración propia)

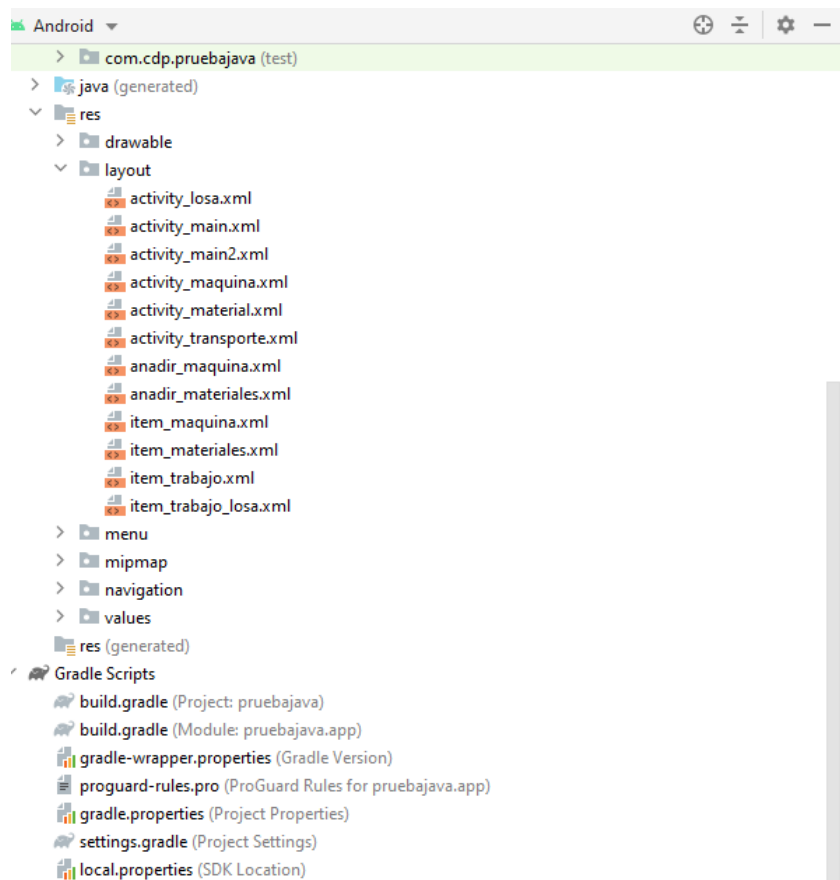


Figura 29. Layouts aplicación *Android Studio* (Fuente: elaboración propia)

Código “AndroidManifest.xml”. Es el código general donde se muestran las diferentes actividades, los iconos de la aplicación y nombre de la aplicación con los diferentes formatos de cada uno.

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher"
        android:supportRtl="true"
        android:theme="@style/Theme.Pruebajava">
        <activity android:name=".AnadirMaquina"></activity>
        <activity android:name=".AnadirMateriales"></activity>
        <activity android:name=".AltaTrabajo"></activity>
        <activity android:name=".AltaTrabajoLosa"></activity>
        <activity android:name=".AltaTransporte"></activity>
        <activity android:name=".NuevaMaquina"></activity>
        <activity android:name=".NuevoMaterial"></activity>
        <activity
            android:name=".Principal"
            android:label="@string/app_name"
            android:theme="@style/Theme.Pruebajava.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

A continuación, se mostrará el código de las diferentes actividades y clases.

1. Clase “AdaptadorMaquinas”.

```
package com.cdp.pruebajava;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;

import java.util.List;

public class AdaptadorMaquinas extends BaseAdapter {

    private List<Maquina> maquinas;
    private Context contexto;

    AdaptadorMaquinas(List<Maquina> maquinas, Context contexto) {
        this.maquinas = maquinas;
        this.contexto = contexto;
    }
}
```

```

@Override
public int getCount() {
    return maquinas.size();
}

@Override
public Object getItem(int position) {
    return maquinas.get(position);
}

@Override
public long getItemId(int position) {
    return 0;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder = null;

    if (holder == null) {
        LayoutInflater inflater = LayoutInflater.from(contexto);
        convertView = inflater.inflate(R.layout.item_maquina, null);

        holder = new ViewHolder();

        holder.titulo =
convertView.findViewById(R.id.tvItemIdentificador);
        holder.consumo = convertView.findViewById(R.id.tvItemConsumo);
        holder.cantidad = convertView.findViewById(R.id.tvItemCantidad);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }

    holder.titulo.setText(maquinas.get(position).getIdentificador());
    holder.consumo.setText(String.valueOf(maquinas.get(position).getConsumo()));
    holder.cantidad.setText(String.valueOf(maquinas.get(position).getCantidad()));

    holder.eliminar =
(Button)convertView.findViewById(R.id.btEliminarMaquina);
    holder.eliminar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            maquinas.remove(position);
            AdaptadorMaquinas.this.notifyDataSetChanged();

            int suma = 0;
            for (Maquina m : maquinas) {
                suma += m.getConsumo();
            }
            TextView tvConsumoTotal = (TextView)
((AltaTrabajo)contexto).findViewById(R.id.tvConsumoTotal);
            tvConsumoTotal.setText(String.valueOf(suma));

        }
    });

    return convertView;
}

```

```

private class ViewHolder {
    TextView titulo;
    TextView consumo;
    TextView cantidad;
    Button eliminar;
}
}

```

2. Clase "AdaptadorMateriales".

```

package com.cdp.pruebajava;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;

import java.util.List;

public class AdaptadorMateriales extends BaseAdapter {

    private List<Materiales> materiales;
    private Context contexto;

    AdaptadorMateriales(List<Materiales> materiales, Context contexto) {
        this.materiales = materiales;
        this.contexto = contexto;
    }

    @Override
    public int getCount() {
        return materiales.size();
    }

    @Override
    public Object getItem(int position) {
        return materiales.get(position);
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder holder = null;

        if (holder == null) {
            LayoutInflater inflater = LayoutInflater.from(contexto);
            convertView = inflater.inflate(R.layout.item_materiales, null);

            holder = new ViewHolder();

            holder.titulo =
convertView.findViewById(R.id.tvItemIdentificador);
            holder.coste = convertView.findViewById(R.id.tvCosteMaterial);
            holder.cantidad = convertView.findViewById(R.id.tvItemCantidad);
            convertView.setTag(holder);

```

```

        } else {
            holder = (ViewHolder) convertView.getTag();
        }

        holder.titulo.setText(materiales.get(position).getIdentificador());
holder.coste.setText(String.valueOf(materiales.get(position).getCoste()));
holder.cantidad.setText(String.valueOf(materiales.get(position).getCantidad()));
};

        holder.eliminar =
(Button)convertView.findViewById(R.id.btEliminarMateriales);
        holder.eliminar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                materiales.remove(position);
                AdaptadorMateriales.this.notifyDataSetChanged();

                int suma = 0;
                for (Materiales m : materiales) {
                    suma += m.getCoste()*m.getCantidad();
                }
                TextView tvCosteMateriales = (TextView)
((AltaTrabajo) contexto).findViewById(R.id.tvCosteMateriales);
                tvCosteMateriales.setText(String.valueOf(suma));

            }
        });

        return convertView;
    }

    private class ViewHolder {
        TextView titulo;
        TextView coste;
        TextView cantidad;
        Button eliminar;
    }
}

```

3. Clase "AdaptadorTrabajos".

```

package com.cdp.pruebajava;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;

import java.text.DecimalFormat;
import java.util.List;

public class AdaptadorTrabajos extends BaseAdapter {

    private List<Trabajo> trabajos;
    private Context contexto;
    private String tipoTrabajo;

```

```

AdaptadorTrabajos(List<Trabajo> trabajos, Context contexto) {
    this.trabajos = trabajos;
    this.contexto = contexto;
}

@Override
public int getCount() {
    return trabajos.size();
}

@Override
public Object getItem(int position) {
    return trabajos.get(position);
}

@Override
public long getItemId(int position) {
    return 0;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    tipoTrabajo= trabajos.get(position).getTipo();
    ViewHolder holder = null;
    DecimalFormat db = new DecimalFormat("0.00");

    if (tipoTrabajo.equals("Losa museo")){

        if (holder == null) {
            LayoutInflater inflater = LayoutInflater.from(contexto);
            convertView = inflater.inflate(R.layout.item_trabajo_losa,
null);

            holder = new ViewHolder();

            holder.tipo = convertView.findViewById(R.id.tvItemTipo);
            holder.vehiculo =
convertView.findViewById(R.id.tvItemVehiculo);
            holder.titulo =
convertView.findViewById(R.id.tvItemIdentificador);

            holder.ancho = convertView.findViewById(R.id.tvItemAncho);
            holder.largo = convertView.findViewById(R.id.tvItemLargo);
            holder.alto = convertView.findViewById(R.id.tvItemAlto);
            holder.hierros = convertView.findViewById(R.id.tvItemHierros);
            holder.diametro =
convertView.findViewById(R.id.tvItemDiametro);
            holder.separacion =
convertView.findViewById(R.id.tvItemSeparacion);

            holder.tiempo = convertView.findViewById(R.id.tvItemTiempo);
            holder.coste = convertView.findViewById(R.id.tvItemCoste);

            convertView.setTag(holder);

        } else {
            holder = (ViewHolder) convertView.getTag();
        }
    }
}

```

```

        holder.tipo.setText(trabajos.get(position).getTipo());
        holder.vehiculo.setText(trabajos.get(position).getVehiculo());
        holder.titulo.setText(trabajos.get(position).getIdentificador());

holder.ancho.setText(String.valueOf(trabajos.get(position).getAncho()));
holder.largo.setText(String.valueOf(trabajos.get(position).getLargo()));
holder.alto.setText(String.valueOf(trabajos.get(position).getAlto()));
holder.hierros.setText(String.valueOf(trabajos.get(position).getHierros()));
holder.diametro.setText(String.valueOf(trabajos.get(position).getDiametro()));
holder.separacion.setText(String.valueOf(trabajos.get(position).getSeparacion(
)));

holder.tiempo.setText(db.format(trabajos.get(position).getTiempo()));
holder.coste.setText(db.format(trabajos.get(position).getCoste()));

        holder.eliminar = (Button)
convertView.findViewById(R.id.btEliminar);
        holder.eliminar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                trabajos.remove(position);
                AdaptadorTrabajos.this.notifyDataSetChanged();

                double sumaTiempo = 0;
                double sumaCombustible = 0;
                double sumaCoste = 0;

                for (Trabajo t : trabajos) {
                    sumaTiempo += t.getTiempo();
                    sumaCombustible += t.getConsumoTotal() *
t.getTiempo();
                    sumaCoste += t.getCoste();
                }
                TextView tvSumaTiempo = (TextView) ((Principal)
contexto).findViewById(R.id.tvSumaTiempo);
                tvSumaTiempo.setText(db.format(sumaTiempo));
                TextView tvSumaCombustible = (TextView) ((Principal)
contexto).findViewById(R.id.tvSumaCombustible);
                tvSumaCombustible.setText(db.format(sumaCombustible));
                TextView tvSumaCoste = (TextView) ((Principal)
contexto).findViewById(R.id.tvSumaCoste);
                tvSumaCoste.setText(db.format(sumaCoste));

                TextView tvMostrar = (TextView) ((Principal)
contexto).findViewById(R.id.tvMostrar);
                tvMostrar.setText(trabajos.size() + " Trabajos");

            }
        });
    } else {

```

```

        if (holder == null) {
            LayoutInflater inflater = LayoutInflater.from(contexto);
            convertView = inflater.inflate(R.layout.item_trabajo, null);

            holder = new ViewHolder();

            holder.tipo = convertView.findViewById(R.id.tvItemTipo);
            holder.titulo =
convertView.findViewById(R.id.tvItemIdentificador);
            holder.subtitulo =
convertView.findViewById(R.id.tvItemSuperficie);
            holder.tiempo = convertView.findViewById(R.id.tvItemTiempo);
            holder.consumoTotal =
convertView.findViewById(R.id.tvItemConsumoTotal);
            holder.tvSuperficie =
convertView.findViewById(R.id.tvSuperficie);
            holder.coste = convertView.findViewById(R.id.tvItemCoste);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }

        holder.tipo.setText(trabajos.get(position).getTipo());
        holder.titulo.setText(trabajos.get(position).getIdentificador());

        holder.subtitulo.setText(String.valueOf(trabajos.get(position).getSuperficie()
));

        holder.tiempo.setText(db.format(trabajos.get(position).getTiempo()));

        holder.consumoTotal.setText(db.format(trabajos.get(position).getConsumoTotal()
));

        holder.coste.setText(db.format(trabajos.get(position).getCoste()));

        String tipo = trabajos.get(position).getTipo();
        if (tipo.equals("Excavación")) {
            holder.tvSuperficie.setText("Volumen [m³]:");
        } else if (tipo.equals("Explicación")) {
            holder.tvSuperficie.setText("Superficie [m²]:");
        } else {
            holder.tvSuperficie.setText("Cantidad:");
        }

        holder.eliminar = (Button)
convertView.findViewById(R.id.btEliminar);
        holder.eliminar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                trabajos.remove(position);
                AdaptadorTrabajos.this.notifyDataSetChanged();

                double sumaTiempo = 0;
                double sumaCombustible = 0;
                double sumaCoste = 0;

                for (Trabajo t : trabajos) {
                    sumaTiempo += t.getTiempo();

```



```

        sumaCombustible += t.getConsumoTotal() *
t.getTiempo();
        sumaCoste += t.getCoste();
    }
    TextView tvSumaTiempo = (TextView) ((Principal)
contexto).findViewById(R.id.tvSumaTiempo);
    tvSumaTiempo.setText(db.format(sumaTiempo));
    TextView tvSumaCombustible = (TextView) ((Principal)
contexto).findViewById(R.id.tvSumaCombustible);
    tvSumaCombustible.setText(db.format(sumaCombustible));
    TextView tvSumaCoste = (TextView) ((Principal)
contexto).findViewById(R.id.tvSumaCoste);
    tvSumaCoste.setText(db.format(sumaCoste));

    TextView tvMostrar = (TextView) ((Principal)
contexto).findViewById(R.id.tvMostrar);
    tvMostrar.setText(trabajos.size() + " Trabajos");

    }
    });
}
return convertView;
}

private class ViewHolder {
    TextView tipo;
    TextView titulo;
    TextView subtitulo;
    TextView tiempo;
    TextView tvSuperficie;
    TextView consumoTotal;
    TextView coste;
    TextView ancho, largo, alto, hierros, diametro, separacion, vehiculo;
    Button eliminar;
}
}

```

4. Clase "AltaTrabajo".

```

package com.cdp.pruebajava;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;

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

public class AltaTrabajo extends AppCompatActivity {

    private EditText etSuperficie, etIdentificador, etPersonal;
    private TextView tvConsumoTotal, tvDimension, tvCosteMateriales, tvError,

```

```

tvSuelo;
private Button btGuardar;
private Button btAnadir, btAnadirMateriales;
String tipo;
double tiempo;
double coste;

private List<Maquina> maquinas;
private List<Materiales> materialesLista;
private ListView lvMaquinas;
private ListView lvMateriales;
private AdaptadorMaquinas adaptadorMaquinas;
private AdaptadorMateriales adaptadorMateriales;

private Context contexto;
private int suma;
private double sumaMateriales;

private double precioCombustible;
private double rendimientoMaquinas;
private double rendimientoPersonal;
private double eficiencia;
private double superficie;
private double precioExplosivo, precioCordon;
private int indicador;
private Spinner spRegimen, spSuelo;
private List<String> tiposRegimen, tiposSuelo;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    inicializarComponentesUI();
    anadirEscuchadores();
}

private void anadirEscuchadores() {
    btGuardar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (etSuperficie.getText().toString().isEmpty() ||
etPersonal.getText().toString().isEmpty()) {
                tvError.setVisibility(TextView.VISIBLE);
            } else{
                rendimientoMaquinas = 0;
                for (Maquina m : maquinas) {
                    rendimientoMaquinas += m.getRendimiento();
                }

                if (spSuelo.getSelectedItemPosition() == 0) {
                    rendimientoPersonal = 2.5;
                } else if (spSuelo.getSelectedItemPosition() == 1) {
                    rendimientoPersonal = 1.7;
                } else if (spSuelo.getSelectedItemPosition() == 2) {
                    rendimientoPersonal = 0.7;
                } else if (spSuelo.getSelectedItemPosition() == 3) {
                    rendimientoPersonal = 0.6;
                } else {
                    rendimientoPersonal = 0.3;
                    indicador = 1;
                }

                if (spRegimen.getSelectedItemPosition() == 0) {
                    eficiencia = 1;
                }
            }
        }
    });
}

```

```

        } else if (spRegimen.getSelectedItemPosition() == 1) {
            eficiencia = 1.15;
        } else if (spRegimen.getSelectedItemPosition() == 2) {
            eficiencia = 0.6;
        } else if (spRegimen.getSelectedItemPosition() == 3) {
            eficiencia = 0.7;
        } else if (spRegimen.getSelectedItemPosition() == 4) {
            eficiencia = 0.75;
        } else if (spRegimen.getSelectedItemPosition() == 5) {
            eficiencia = 0.85;
        } else if (spRegimen.getSelectedItemPosition() == 6) {
            eficiencia = 0.45;
        } else if (spRegimen.getSelectedItemPosition() == 7) {
            eficiencia = 0.5;
        } else if (spRegimen.getSelectedItemPosition() == 8) {
            eficiencia = 0.6;
        } else if (spRegimen.getSelectedItemPosition() == 9) {
            eficiencia = 0.8;
        } else if (spRegimen.getSelectedItemPosition() == 10) {
            eficiencia = 1.2;
        }

        superficie =
Integer.parseInt(etSuperficie.getText().toString());
        Trabajo trabajo = new Trabajo();

trabajo.setSuperficie(Integer.parseInt(etSuperficie.getText().toString()));

trabajo.setIdentificador(etIdentificador.getText().toString());
        trabajo.setTipo(tipo);

trabajo.setConsumoTotal(Integer.parseInt(tvConsumoTotal.getText().toString()));
;

trabajo.setPersonal(Integer.parseInt(etPersonal.getText().toString()));
        if (tipo.equals("Excavación")){
            tiempo = 0.5 + (trabajo.getSuperficie()) /
(rendimientoMaquinas + trabajo.getPersonal() * rendimientoPersonal *
eficiencia);
        }else{
            tiempo = 1 + (1/2)* (trabajo.getSuperficie()/1000)+
(trabajo.getSuperficie()) / (rendimientoMaquinas + trabajo.getPersonal() *
rendimientoPersonal * eficiencia);
        }
        trabajo.setTiempo(tiempo);

        coste =
Double.parseDouble(tvConsumoTotal.getText().toString()) * tiempo *
precioCombustible +
            superficie * indicador * (0.4 * precioExplosivo +
0.4 * precioCordon) +
Double.parseDouble(tvCosteMateriales.getText().toString());
        trabajo.setCoste(coste);

        Intent intent = new Intent();
        intent.putExtra("trabajo", trabajo);
        setResult(1, intent);
        finish();
    }

}

});
btAnadir.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {

            Intent intent = new Intent(AltaTrabajo.this,
AnadirMaquina.class);
            intent.putExtra("tipo", tipo);
            startActivityForResult(intent, 2);

        }
    });
    btAnadirMateriales.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v){
            Intent intent = new Intent(AltaTrabajo.this,
AnadirMateriales.class);
            startActivityForResult(intent, 3);
        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 2 && resultCode == 2) {
        Maquina maquina = (Maquina) data.getSerializableExtra("maquina");
        //tvMostrar.setText(trabajo.getIdentificador() + " " +
trabajo.getSuperficie());

        maquinas.add(maquina);
        adaptadorMaquinas.notifyDataSetChanged();

        suma = 0;
        for (Maquina m : maquinas) {
            suma += m.getConsumo();
        }
        tvConsumoTotal.setText(String.valueOf(suma));
    }
    if (requestCode == 3 && resultCode == 3) {
        Materiales materiales = (Materiales)
data.getSerializableExtra("materiales");
        //tvMostrar.setText(trabajo.getIdentificador() + " " +
trabajo.getSuperficie());

        materialesLista.add(materiales);
        adaptadorMateriales.notifyDataSetChanged();

        sumaMateriales = 0;
        for (Materiales m : materialesLista) {
            sumaMateriales += (m.getCoste() * m.getCantidad());
        }
        tvCosteMateriales.setText(String.valueOf(sumaMateriales));
    }
}

private void inicializarComponentesUI() {
    btGuardar = findViewById(R.id.btGuardarExplanacion);
    btAnadir = findViewById(R.id.btAnadirMaquina);
    btAnadirMateriales = findViewById(R.id.btAnadirMateriales);
    tvConsumoTotal = findViewById(R.id.tvConsumoTotal);
    tvDimension = findViewById(R.id.tvDimension);
}

```

```

tvCosteMateriales = findViewById(R.id.tvCosteMateriales);
etSuperficie = findViewById(R.id.etSuperficie);
etIdentificador = findViewById(R.id.etIdentificador);
tipo= getIntent().getStringExtra("tipo");
precioCombustible= ((Principal) contexto).precioCombustible;
etPersonal= findViewById(R.id.etPersonal);
rendimientoPersonal = 2.6;
indicador=0;
precioExplosivo=34;
precioCordon=2;
tvError= findViewById(R.id.tvError);
suma=0;
tvSuelo = findViewById(R.id.tvSuelo);

spSuelo = findViewById(R.id.spSuelo);
tiposSuelo = new ArrayList<>();
tiposSuelo.add("Tierra suelta");
tiposSuelo.add("Arenoso");
tiposSuelo.add("Compacto");
tiposSuelo.add("Dureza media");
tiposSuelo.add("Suelo duro");
ArrayAdapter<String> arrayAdapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, tiposSuelo);
spSuelo.setAdapter(arrayAdapter2);

if (tipo.equals("Excavación")){
    tvDimension.setText("Volumen [m³]*");
}else if (tipo.equals("Explicación")){
    tvDimension.setText("Superficie [m²]*");
    tvSuelo.setVisibility(TextView.INVISIBLE);
    spSuelo.setVisibility(Spinner.INVISIBLE);
}else{
    tvDimension.setText("Cantidad*");
}

lvMaquinas = findViewById(R.id.lvMaquinas);
maquinas = new ArrayList<>();
adaptadorMaquinas = new AdaptadorMaquinas(maquinas, this);
lvMaquinas.setAdapter(adaptadorMaquinas);

lvMateriales = findViewById(R.id.lvMateriales);
materialesLista = new ArrayList<>();
adaptadorMateriales = new AdaptadorMateriales(materialesLista, this);
lvMateriales.setAdapter(adaptadorMateriales);

spRegimen = findViewById(R.id.spRegimen);
tiposRegimen = new ArrayList<>();
tiposRegimen.add("Discontinuo de día (8h)");
tiposRegimen.add("Discontinuo de día (10h)");
tiposRegimen.add("Discontinuo de noche (8h)");
tiposRegimen.add("Discontinuo de noche (10h)");
tiposRegimen.add("Continuo de día (6h)");
tiposRegimen.add("Continuo de día (7h)");
tiposRegimen.add("Continuo de noche (6h)");
tiposRegimen.add("Continuo de noche (7h)");
tiposRegimen.add("Continuo de día y noche (6h)");
tiposRegimen.add("Continuo de día y noche (8h)");
tiposRegimen.add("A destajo");
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, tiposRegimen);
spRegimen.setAdapter(arrayAdapter);

```

```

    }
}

```

5. Clase "AltaTrabajoLosa".

```

package com.cdp.pruebajava;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;

import androidx.appcompat.app.AppCompatActivity;

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

public class AltaTrabajoLosa extends AppCompatActivity {

    private Spinner spLosa;
    private List<String> tiposLosa;
    private Button btGuardar;
    private double coste, tiempo;
    String tipo;
    private EditText etIdentificador;

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

    private void inicializarComponentesUI() {
        spLosa = findViewById(R.id.spLosa);
        tiposLosa = new ArrayList<>();
        tiposLosa.add("Leopardo 2E");
        tiposLosa.add("BMR");
        tiposLosa.add("Pizarro");
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, tiposLosa);
        spLosa.setAdapter(arrayAdapter);

        btGuardar = findViewById(R.id.btGuardar);
        tipo= getIntent().getStringExtra("tipo");
        etIdentificador= findViewById(R.id.etIdentificadorLosa);
    }

    private void anadirEscuchadores() {
        btGuardar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

//superficie=Integer.parseInt(etSuperficie.getText().toString());

```

```

Trabajo trabajo = new Trabajo();
if (spLosa.getSelectedItemPosition() == 0) {
    trabajo.setAncho(5.75);
    trabajo.setLargo(11.67);
    trabajo.setAlto(30);
    trabajo.setHierros(116);
    trabajo.setDiametro(12);
    trabajo.setSeparacion(30);
    trabajo.setConsumoTotal(0);
    tiempo = 36;
    trabajo.setTiempo(tiempo);

    coste = 2990;
    trabajo.setCoste(coste);
} else if (spLosa.getSelectedItemPosition() == 1) {
    trabajo.setAncho(4.5);
    trabajo.setLargo(8.15);
    trabajo.setAlto(30);
    trabajo.setHierros(84);
    trabajo.setDiametro(12);
    trabajo.setSeparacion(30);
    trabajo.setConsumoTotal(0);
    tiempo = 36;
    trabajo.setTiempo(tiempo);

    coste = 1648;
    trabajo.setCoste(coste);
} else {
    trabajo.setAncho(5.15);
    trabajo.setLargo(8.83);
    trabajo.setAlto(30);
    trabajo.setHierros(93);
    trabajo.setDiametro(12);
    trabajo.setSeparacion(30);
    trabajo.setConsumoTotal(0);
    tiempo = 36;
    trabajo.setTiempo(tiempo);

    coste = 2043;
    trabajo.setCoste(coste);
}

trabajo.setIdentificador(etIdentificador.getText().toString());
trabajo.setVehiculo((String) spLosa.getSelectedItem());
//losa.setIdentificador(etIdentificador.getText().toString());
trabajo.setTipo(tipo);

//losa.setConsumoTotal(Integer.parseInt(tvConsumoTotal.getText().toString()));

//losa.setPersonal(Integer.parseInt(etPersonal.getText().toString()));

Intent intent = new Intent();
intent.putExtra("trabajo", trabajo);
setResult(1, intent);
finish();
    }
});
}
}

```

6. Clase "AltaTransporte".

```
package com.cdp.pruebajava;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;

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

public class AltaTransporte extends AppCompatActivity {

    private EditText etSuperficie, etIdentificador, etPersonal, etDistancia;
    private TextView tvConsumoTotal, tvError, tvDimension, tvDistancia;
    private Button btGuardar;
    private Button btAnadir;
    private String tipo;
    private double tiempo;
    private double coste;

    private List<Maquina> maquinas;
    private ListView lvMaquinas;
    private AdaptadorMaquinas adaptadorMaquinas;

    private Context contexto;
    private int suma;

    private double precioCombustible;
    private double rendimientoMaquinas;
    private double rendimientoPersonal;
    private double eficiencia;
    private Spinner spRegimen;
    private List<String> tiposRegimen;

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

    private void anadirEscuchadores() {
        btGuardar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (etSuperficie.getText().toString().isEmpty() ||
                    etPersonal.getText().toString().isEmpty()) {
                    tvError.setVisibility(TextView.VISIBLE);
                } else if (tipo.equals("Transporte de tierra") &&
                    etDistancia.getText().toString().isEmpty()) {
                    tvError.setVisibility(TextView.VISIBLE);
                }
            }
        });
    }
}
```



```

    } else {
        rendimientoMaquinas = 0;
        for (Maquina m : maquinas) {
            rendimientoMaquinas += m.getRendimiento();
        }

        if (spRegimen.getSelectedItemPosition() == 0) {
            eficiencia = 1;
        } else if (spRegimen.getSelectedItemPosition() == 1) {
            eficiencia = 1.15;
        } else if (spRegimen.getSelectedItemPosition() == 2) {
            eficiencia = 0.6;
        } else if (spRegimen.getSelectedItemPosition() == 3) {
            eficiencia = 0.7;
        } else if (spRegimen.getSelectedItemPosition() == 4) {
            eficiencia = 0.75;
        } else if (spRegimen.getSelectedItemPosition() == 5) {
            eficiencia = 0.85;
        } else if (spRegimen.getSelectedItemPosition() == 6) {
            eficiencia = 0.45;
        } else if (spRegimen.getSelectedItemPosition() == 7) {
            eficiencia = 0.5;
        } else if (spRegimen.getSelectedItemPosition() == 8) {
            eficiencia = 0.6;
        } else if (spRegimen.getSelectedItemPosition() == 9) {
            eficiencia = 0.8;
        } else if (spRegimen.getSelectedItemPosition() == 10) {
            eficiencia = 1.2;
        }

        Trabajo trabajo = new Trabajo();

        trabajo.setSuperficie(Integer.parseInt(etSuperficie.getText().toString()));
        trabajo.setIdentificador(etIdentificador.getText().toString());
        trabajo.setTipo(tipo);

        trabajo.setConsumoTotal(Integer.parseInt(tvConsumoTotal.getText().toString()));
        ;

        trabajo.setPersonal(Integer.parseInt(etPersonal.getText().toString()));
        if (tipo.equals("Compactación")){
            tiempo = 0.5 + (trabajo.getSuperficie()) /
            (rendimientoMaquinas + trabajo.getPersonal() * rendimientoPersonal *
            eficiencia);
        } else{
            tiempo = 0.5 + (trabajo.getSuperficie()) /
            (rendimientoMaquinas/Integer.parseInt(etDistancia.getText().toString()) +
            trabajo.getPersonal() * rendimientoPersonal * eficiencia);
        }
        trabajo.setTiempo(tiempo);

        coste =
        Double.parseDouble(tvConsumoTotal.getText().toString()) * tiempo *
        precioCombustible;
        trabajo.setCoste(coste);

        Intent intent = new Intent();
        intent.putExtra("trabajo", trabajo);
        setResult(1, intent);
        finish();
    }
}

});
btAnadir.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {

            Intent intent = new Intent(AltaTransporte.this,
AnadirMaquina.class);
            intent.putExtra("tipo", tipo);
            startActivityForResult(intent, 2);

        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 2 && resultCode == 2) {
        Maquina maquina = (Maquina) data.getSerializableExtra("maquina");
        //tvMostrar.setText(trabajo.getIdentificador() + " " +
trabajo.getSuperficie());

        maquinas.add(maquina);
        adaptadorMaquinas.notifyDataSetChanged();

        suma = 0;
        for (Maquina m : maquinas) {
            suma += m.getConsumo();
        }
        tvConsumoTotal.setText(String.valueOf(suma));
    }

}

private void inicializarComponentesUI() {
    btGuardar = findViewById(R.id.btGuardarExplanacion);
    btAnadir = findViewById(R.id.btAnadirMaquina);
    tvConsumoTotal = findViewById(R.id.tvConsumoTotal);
    etSuperficie = findViewById(R.id.etSuperficie);
    etIdentificador = findViewById(R.id.etIdentificador);
    tipo= getIntent().getStringExtra("tipo");
    precioCombustible= ((Principal) contexto).precioCombustible;
    etPersonal= findViewById(R.id.etPersonal);
    rendimientoPersonal = 2;
    tvError = findViewById(R.id.tvError);
    tvDimension = findViewById(R.id.tvDimension);
    etDistancia = findViewById(R.id.etDistancia);

    if (tipo.equals("Compactación")){
        tvDimension.setText("Superficie [m²]*");
        rendimientoPersonal = 2.5;
        tvDistancia.setVisibility(TextView.INVISIBLE);
        etDistancia.setVisibility(TextView.INVISIBLE);
    }

    suma=0;

    lvMaquinas = findViewById(R.id.lvMaquinas);
    maquinas = new ArrayList<>();
    adaptadorMaquinas = new AdaptadorMaquinas(maquinas, this);
    lvMaquinas.setAdapter(adaptadorMaquinas);
}

```

```

        spRegimen = findViewById(R.id.spRegimen);
        tiposRegimen = new ArrayList<>();
        tiposRegimen.add("Discontinuo de día (8h)");
        tiposRegimen.add("Discontinuo de día (10h)");
        tiposRegimen.add("Discontinuo de noche (8h)");
        tiposRegimen.add("Discontinuo de noche (10h)");
        tiposRegimen.add("Continuo de día (6h)");
        tiposRegimen.add("Continuo de día (7h)");
        tiposRegimen.add("Continuo de noche (6h)");
        tiposRegimen.add("Continuo de noche (7h)");
        tiposRegimen.add("Continuo de día y noche (6h)");
        tiposRegimen.add("Continuo de día y noche (8h)");
        tiposRegimen.add("A destajo");
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, tiposRegimen);
        spRegimen.setAdapter(arrayAdapter);

    }

}

```

7. Clase “AnadirMaquina”.

```

package com.cdp.pruebajava;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

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

public class AnadirMaquina extends AppCompatActivity {

    private Button btAceptar, btNuevaMaquina;
    private Spinner spMaquinas;
    private List<String> tiposMaquina;
    private int cantidad;
    private EditText etCantidad;
    private int consumo1, consumo2, consumo3, consumo4, consumo5, consumo6,
consumo7, consumo8, consumo9, consumo10;
    private int rendimiento1, rendimiento2, rendimiento3, rendimiento4,
rendimiento5, rendimiento6, rendimiento7, rendimiento8, rendimiento9,
rendimiento10;
    private String tipo;
    private TextView tvError;

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

    private void inicializarComponentesUI() {

```

```

    tipo = getIntent().getStringExtra("tipo");
    btAceptar = findViewById(R.id.btAceptar);
    etCantidad = findViewById(R.id.etCantidad);
    tvError = findViewById(R.id.tvError);
    btNuevaMaquina = findViewById(R.id.btNuevaMaquina);

    consumo1 = 13;
    consumo2 = 14;
    consumo3 = 14;
    consumo4 = 26;
    consumo5 = 20;
    consumo6 = 18;
    consumo7 = 15;
    consumo8 = 12;
    consumo9 = 17;
    consumo10 = 30;
    rendimiento1 = 70;
    rendimiento2 = 100;
    rendimiento3 = 30;
    rendimiento4 = 40;
    rendimiento5 = 290;
    rendimiento6 = 300;
    rendimiento7 = 160;
    rendimiento8 = 82;
    rendimiento9 = 165;
    rendimiento10 = 10*80;

    spMaquinas = findViewById(R.id.spMaquina);
    tiposMaquina = new ArrayList<>();

    if (tipo.equals("Excavación")) {

        tiposMaquina.add("Excavadora CAT 225");
        tiposMaquina.add("Excavadora CAT 235");
        tiposMaquina.add("Buldozer CAT D6D");
        tiposMaquina.add("Buldozer CAT D7G");

    } else if (tipo.equals("Explicación")) {
        tiposMaquina.add("Motoniveladora CAT 120G");
        tiposMaquina.add("Motoniveladora CAT 140G");
    } else if (tipo.equals("Transporte de tierra")) {
        tiposMaquina.add("Cargadora CAT 950");
        tiposMaquina.add("Cargadora CAT 966");
        tiposMaquina.add("Volquete Iveco");
    } else {
        tiposMaquina.add("Rodillo compactador CA-15 LISO");
    }

    ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
    android.R.layout.simple_spinner_dropdown_item, tiposMaquina);
    spMaquinas.setAdapter(arrayAdapter);

}

private void anadirEscuchadores() {
    btAceptar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (etCantidad.getText().toString().isEmpty()) {
                tvError.setVisibility(TextView.VISIBLE);
            } else {
                Maquina maquina = new Maquina();

```

```

maquina.setIdentificador(spMaquinas.getSelectedItem().toString());
        cantidad =
Integer.parseInt(etCantidad.getText().toString());
maquina.setCantidad(cantidad);

        switch (maquina.getIdentificador()) {
            case "Excavadora CAT 225":
                maquina.setConsumo(consumo1 * cantidad);
                maquina.setRendimiento(rendimiento1 * cantidad);
                break;
            case "Excavadora CAT 235":
                maquina.setConsumo(consumo2 * cantidad);
                maquina.setRendimiento(rendimiento2 * cantidad);
                break;
            case "Buldozer CAT D6D":
                maquina.setConsumo(consumo3 * cantidad);
                maquina.setRendimiento(rendimiento3 * cantidad);
                break;
            case "Buldozer CAT D7G":
                maquina.setConsumo(consumo4 * cantidad);
                maquina.setRendimiento(rendimiento4 * cantidad);
                break;
            case "Motoniveladora CAT 120G":
                maquina.setConsumo(consumo5 * cantidad);
                maquina.setRendimiento(rendimiento5 * cantidad);
                break;
            case "Motoniveladora CAT 140G":
                maquina.setConsumo(consumo6 * cantidad);
                maquina.setRendimiento(rendimiento6 * cantidad);
                break;
            case "Rodillo compactador CA-15 LISO":
                maquina.setConsumo(consumo7 * cantidad);
                maquina.setRendimiento(rendimiento7 * cantidad);
                break;
            case "Cargadora CAT 950":
                maquina.setConsumo(consumo8 * cantidad);
                maquina.setRendimiento(rendimiento8 * cantidad);
                break;
            case "Cargadora CAT 966":
                maquina.setConsumo(consumo9 * cantidad);
                maquina.setRendimiento(rendimiento9 * cantidad);
                break;
            case "Volquete Iveco":
                maquina.setConsumo(consumo10 * cantidad);
                maquina.setRendimiento(rendimiento10 * cantidad);
                break;
        }
        Intent intent = new Intent();
        intent.putExtra("maquina", maquina);
        setResult(2, intent);
        finish();
    }
}

});
btNuevaMaquina.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent intent = new Intent(AnadirMaquina.this,
NuevaMaquina.class);
        startActivityForResult(intent, 4);
    }
}

```

```

        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 4 && resultCode == 4) {
        Maquina maquina = (Maquina) data.getSerializableExtra("maquina");

        Intent intent = new Intent();
        intent.putExtra("maquina", maquina);
        setResult(2, intent);
        finish();
    }
}
}

```

8. Clase "AnadirMateriales".

```

package com.cdp.pruebajava;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

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

public class AnadirMateriales extends AppCompatActivity {

    private Button btAceptar, btNuevoMaterial;
    private Spinner spMateriales;
    private List<String> tiposMateriales;
    private int cantidad;
    private EditText etCantidad;
    private double coste1, coste2, coste3;
    private TextView tvError;

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

    private void inicializarComponentesUI() {

        btAceptar = findViewById(R.id.btAceptarMateriales);
        btNuevoMaterial = findViewById(R.id.btNuevoMaterial);
        etCantidad = findViewById(R.id.etCantidadMateriales);
        tvError = findViewById(R.id.tvError);
    }
}

```

```

        coste1 = 6.95;
        coste2 = 0.13;
        coste3 = 21.33;

        spMateriales = findViewById(R.id.spMateriales);
        tiposMateriales = new ArrayList<>();
        tiposMateriales.add("Tablon de madera 250x20x2.5");
        tiposMateriales.add("Ladrillo ceramica 24x11x10");
        tiposMateriales.add("Big Bag Grava 4/12 (1000kg)");
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, tiposMateriales);
        spMateriales.setAdapter(arrayAdapter);

    }

    private void anadirEscuchadores() {
        btAceptar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (etCantidad.getText().toString().isEmpty()) {
                    tvError.setVisibility(TextView.VISIBLE);
                } else {
                    Materiales materiales = new Materiales();
materiales.setIdentificador(spMateriales.getSelectedItem().toString());
                    cantidad =
Integer.parseInt(etCantidad.getText().toString());
                    materiales.setCantidad(cantidad);
                    switch (spMateriales.getSelectedItemPosition()) {
                        case 0:
                            materiales.setCoste(coste1);
                            break;
                        case 1:
                            materiales.setCoste(coste2);
                            break;
                        case 2:
                            materiales.setCoste(coste3);
                            break;
                    }
                    Intent intent = new Intent();
                    intent.putExtra("materiales", materiales);
                    setResult(3, intent);
                    finish();
                }
            }
        });
        btNuevoMaterial.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(AnadirMateriales.this,
NuevoMaterial.class);
                startActivityForResult(intent, 4);

            }
        });
    }

    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);

```

```

        if (requestCode == 4 && resultCode == 4) {
            Materiales materiales = (Materiales)
data.getSerializableExtra("materiales");

            Intent intent = new Intent();
            intent.putExtra("materiales", materiales);
            setResult(3, intent);
            finish();
        }
    }
}

```

9. Clase "Maquina".

```

package com.cdp.pruebajava;

import java.io.Serializable;

public class Maquina implements Serializable {
    private int consumo;
    private double rendimiento;
    private String identificador;
    private int cantidad;
    //private String tipo;
    // private double tiempo;

    public int getConsumo() {
        return consumo;
    }

    public void setConsumo(int consumo) {
        this.consumo = consumo;
    }

    public String getIdentificador() {
        return identificador;
    }

    public void setIdentificador(String identificador) {
        this.identificador = identificador;
    }

    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }

    public double getRendimiento() {
        return rendimiento;
    }

    public void setRendimiento(int rendimiento) {
        this.rendimiento = rendimiento;
    }

    // public String getTipo() { return tipo; }

    // public void setTipo(String tipo) { this.tipo = tipo; }

    // public double getTiempo() { return tiempo; }

```



```
// public void setTiempo(double tiempo) { this.tiempo = tiempo; }  
}
```

10. Clase “Materiales”.

```
package com.cdp.pruebajava;  
  
import java.io.Serializable;  
  
public class Materiales implements Serializable {  
    private double coste;  
    private String identificador;  
    private int cantidad;  
    //private String tipo;  
    // private double tiempo;  
  
    public String getIdentificador() {  
        return identificador;  
    }  
  
    public void setIdentificador(String identificador) {  
        this.identificador = identificador;  
    }  
  
    public int getCantidad() {  
        return cantidad;  
    }  
  
    public void setCantidad(int cantidad) {  
        this.cantidad = cantidad;  
    }  
  
    public double getCoste() {  
        return coste;  
    }  
  
    public void setCoste(double coste) {  
        this.coste = coste;  
    }  
}
```

11. Clase “NuevaMaquina”.

```
package com.cdp.pruebajava;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class NuevaMaquina extends AppCompatActivity {  
  
    private Button btAceptarMaquina;  
    private EditText etModelo, etCantidadMaquina, etRendimiento, etConsumo;  
    private TextView tvError;  
  
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maquina);
    inicializarComponentesUI();
    anadirEscuchadores();
}

private void inicializarComponentesUI(){

    btAceptarMaquina = findViewById(R.id.btAceptarMaquina);
    etModelo = findViewById(R.id.etModelo);
    etConsumo = findViewById(R.id.etConsumo);
    etCantidadMaquina = findViewById(R.id.etCantidadMaquina);
    etRendimiento = findViewById(R.id.etRendimiento);
    tvError = findViewById(R.id.tvError);

}

private void anadirEscuchadores() {
    btAceptarMaquina.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (etRendimiento.getText().toString().isEmpty() ||
etCantidadMaquina.getText().toString().isEmpty()
|| etConsumo.getText().toString().isEmpty()) {
                tvError.setVisibility(TextView.VISIBLE);
            } else{
                Maquina maquina = new Maquina();

maquina.setCantidad(Integer.parseInt(etCantidadMaquina.getText().toString()));
maquina.setConsumo(Integer.parseInt(etConsumo.getText().toString()));
maquina.setRendimiento(Integer.parseInt(etRendimiento.getText().toString()));
                maquina.setIdentificador(etModelo.getText().toString());

                Intent intent = new Intent();
                intent.putExtra("maquina", maquina);
                setResult(4, intent);
                finish();

            }

        }
    });
}
}

```

12. Clase "Nuevo material".

```

package com.cdp.pruebajava;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

```

```

public class NuevoMaterial extends AppCompatActivity {

    private Button btAceptarMaterial;
    private EditText etMaterial, etCantidadMaterial, etCoste;
    private TextView tvError;

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

    private void inicializarComponentesUI(){

        btAceptarMaterial = findViewById(R.id.btAceptarMaterial);
        etMaterial = findViewById(R.id.etMaterial);
        etCoste = findViewById(R.id.etCoste);
        etCantidadMaterial = findViewById(R.id.etCantidadMaterial);
        tvError = findViewById(R.id.tvError);

    }

    private void anadirEscuchadores() {
        btAceptarMaterial.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (etCoste.getText().toString().isEmpty() ||
etCantidadMaterial.getText().toString().isEmpty()) {
                    tvError.setVisibility(View.VISIBLE);
                } else{
                    Materiales materiales = new Materiales();

materiales.setCantidad(Integer.parseInt(etCantidadMaterial.getText().toString(
)));

materiales.setCoste(Integer.parseInt(etCoste.getText().toString()));

materiales.setIdentificador(etMaterial.getText().toString());

                    Intent intent = new Intent();
                    intent.putExtra("materiales", materiales);
                    setResult(4, intent);
                    finish();

                }

            }
        });
    }
}

```

13. Clase "Principal".

```

package com.cdp.pruebajava;

import android.content.Context;
import android.content.Intent;

```

```

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.List;

public class Principal extends AppCompatActivity {

    private Button btMostrar;
    private TextView tvMostrar;
    public TextView tvSumaTiempo, tvSumaCombustible, tvSumaCoste;
    private Context contexto;
    private AdaptadorTrabajos adaptadorTrabajos;
    private List<Trabajo> trabajos;
    private ListView lvTrabajos;
    private Spinner spTrabajos;
    private List<String> tiposTrabajo;
    DecimalFormat db = new DecimalFormat("0.00");

    public static double precioCombustible;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        contexto = this;
        inicializarComponentesUI();
        anadirEscuchadores();
    }

    private void anadirEscuchadores() {
        btMostrar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                tvMostrar.setText("Trabajos Ingenieros");

                if (spTrabajos.getSelectedItemPosition() == 0) {
                    Intent intent = new Intent(Principal.this,
AltaTrabajo.class);
                    intent.putExtra("tipo",
spTrabajos.getSelectedItem().toString());
                    startActivityForResult(intent, 1);
                } else if (spTrabajos.getSelectedItemPosition() == 1) {
                    Intent intent = new Intent(Principal.this,
AltaTrabajo.class);
                    intent.putExtra("tipo",
spTrabajos.getSelectedItem().toString());
                    startActivityForResult(intent, 1);
                    //Intent intent = new Intent(contexto,
MainActivity3.class);
                    //startActivityForResult(intent, 1);
                } else if (spTrabajos.getSelectedItemPosition() == 2) {
                    Intent intent = new Intent(Principal.this,
AltaTransporte.class);
                    intent.putExtra("tipo",

```

```

spTrabajos.getSelectedItem().toString());
        startActivityForResult(intent, 1);

        }else if (spTrabajos.getSelectedItemPosition() == 3) {
            Intent intent = new Intent(Principal.this,
AltaTransporte.class);
            intent.putExtra("tipo",
spTrabajos.getSelectedItem().toString());
            startActivityForResult(intent, 1);
        } else {
            Intent intent = new Intent(Principal.this,
AltaTrabajoLosa.class);
            intent.putExtra("tipo",
spTrabajos.getSelectedItem().toString());
            startActivityForResult(intent, 1);
        }
    }
});
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 1 && resultCode == 1) {
        Trabajo trabajo = (Trabajo) data.getSerializableExtra("trabajo");
        tvMostrar.setText(trabajos.size()+1+" Trabajos");

        trabajos.add(trabajo);
        adaptadorTrabajos.notifyDataSetChanged();

        double sumaTiempo = 0;
        double sumaCombustible = 0;
        double sumaCoste = 0;

        for (Trabajo t : trabajos) {
            sumaTiempo += t.getTiempo();
            sumaCombustible += t.getConsumoTotal()* t.getTiempo();

            sumaCoste += t.getCoste();
        }
        tvSumaTiempo.setText(db.format(sumaTiempo));
        tvSumaCombustible.setText(db.format(sumaCombustible));

        tvSumaCoste.setText(db.format(sumaCoste));
    }
}

private void inicializarComponentesUI() {
    btMostrar = findViewById(R.id.btAnadirTrabajo);
    tvMostrar = findViewById(R.id.tvMostrar);

    tvSumaTiempo = findViewById(R.id.tvSumaTiempo);
    tvSumaCombustible = findViewById(R.id.tvSumaCombustible);
    tvSumaCoste = findViewById(R.id.tvSumaCoste);

    lvTrabajos = findViewById(R.id.lvTrabajos);
    trabajos = new ArrayList<>();
    adaptadorTrabajos = new AdaptadorTrabajos(trabajos, contexto);
    lvTrabajos.setAdapter(adaptadorTrabajos);
}

```

```

        spTrabajos = findViewById(R.id.spTrabajo);
        tiposTrabajo = new ArrayList<>();
        tiposTrabajo.add("Excavación");
        tiposTrabajo.add("Explanación");
        tiposTrabajo.add("Compactación");
        tiposTrabajo.add("Transporte de tierra");
        tiposTrabajo.add("Losa museo");
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, tiposTrabajo);
        spTrabajos.setAdapter(arrayAdapter);

        precioCombustible=1.40;

    }

    // public void ActualizarCampos(List<Trabajo> Trabajos){
    //
    //     double sumaTiempo = 0;
    //     double sumaCombustible = 0;
    //
    //     for (Trabajo t : trabajos) {
    //         sumaTiempo += t.getTiempo();
    //         sumaCombustible += t.getConsumoTotal() * t.getTiempo();
    //     }
    //     TextView tvSumaTiempo = (TextView)
    ((Principal)contexto).findViewById(R.id.tvSumaTiempo);
    //     tvSumaTiempo.setText(String.valueOf(sumaTiempo));
    //     TextView tvSumaCombustible = (TextView)
    ((Principal)contexto).findViewById(R.id.tvSumaCombustible);
    //     tvSumaCombustible.setText(String.valueOf(sumaCombustible));
    // }
}

```

14. Clase "Trabajo".

```

package com.cdp.pruebajava;

import java.io.Serializable;

public class Trabajo implements Serializable {
    private int superficie;
    private String identificador;
    private String tipo, vehiculo;
    private double tiempo;
    private double coste;
    private int consumoTotal;
    private int personal;

    public int getSuperficie() {
        return superficie;
    }

    public void setSuperficie(int superficie) {
        this.superficie = superficie;
    }

    public int getConsumoTotal() {
        return consumoTotal;
    }

    public void setConsumoTotal(int consumoTotal) {
        this.consumoTotal = consumoTotal;
    }
}

```

```

public String getIdentificador() {
    return identificador;
}

public void setIdentificador(String identificador) {
    this.identificador = identificador;
}

public String getTipo() {
    return tipo;
}

public void setTipo(String tipo) { this.tipo = tipo;}

public double getTiempo() {
    return tiempo;
}

public void setTiempo(double tiempo) {
    this.tiempo = tiempo;
}

public double getCoste() {
    return coste;
}

public void setCoste(double coste) {
    this.coste = coste;
}

public int getPersonal() { return personal; }

public void setPersonal(int personal) { this.personal = personal; }

private double ancho, largo, alto;
private int hierros, diametro, separacion;

public double getAncho() {
    return ancho;
}

public void setAncho(double ancho) {
    this.ancho = ancho;
}

public double getLargo() {
    return largo;
}

public void setLargo(double largo) {
    this.largo = largo;
}

public double getAlto() {
    return alto;
}

public void setAlto(double alto) {
    this.alto = alto;
}

public int getHierros() {
    return hierros;
}

```

```

    }

    public void setHierros(int hierros) {
        this.hierros = hierros;
    }

    public int getDiametro() {
        return diametro;
    }

    public void setDiametro(int diametro) {
        this.diametro = diametro;
    }

    public int getSeparacion() {
        return separacion;
    }

    public void setSeparacion(int separacion) {
        this.separacion = separacion;
    }

    public String getVehiculo() {
        return vehiculo;
    }

    public void setVehiculo(String vehiculo) { this.vehiculo = vehiculo;}
}

```

Seguidamente se expondrán el código de los “layouts” tanto de las pantallas principales como de los contenedores que se muestran en las listas de presentación.

1. Activity_losa.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <TextView
        android:id="@+id/textView14"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="Selecciona vehículo"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Spinner
        android:id="@+id/spLosa"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView14" />

```



```

<Button
    android:id="@+id/btGuardar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="24dp"
    android:text="Guardar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/textView18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Identificador"
    app:layout_constraintStart_toStartOf="@+id/textView14"
    app:layout_constraintTop_toBottomOf="@+id/spLosa" />

<EditText
    android:id="@+id/etIdentificadorLosa"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="24dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Trabajo"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/textView18"
    app:layout_constraintTop_toBottomOf="@+id/textView18" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

2. Activity_main.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context=".Principal">

    <Button
        android:id="@+id/btAnadirTrabajo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="32dp"
        android:text="Añadir Trabajo"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <TextView
        android:id="@+id/tvMostrar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="Trabajos Ingenieros"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```

```

<ListView
    android:id="@+id/lvTrabajos"
    android:layout_width="409dp"
    android:layout_height="0dp"
    android:layout_marginTop="32dp"
    app:layout_constraintBottom_toTopOf="@+id/cardView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spTrabajo" />

<androidx.cardview.widget.CardView
    android:id="@+id/cardView"
    android:layout_width="0dp"
    android:layout_height="90dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintBottom_toTopOf="@+id/btAnadirTrabajo"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/lvTrabajos">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="32dp"
            android:layout_marginTop="8dp"
            android:text="Tiempo Total [h]:"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/tvSumaTiempo"
            android:layout_width="60dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginTop="8dp"
            app:layout_constraintStart_toEndOf="@+id/textView2"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/textView7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Combustible Total [l]:"
            app:layout_constraintStart_toStartOf="@+id/textView2"
            app:layout_constraintTop_toBottomOf="@+id/textView2" />

        <TextView
            android:id="@+id/tvSumaCombustible"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginTop="8dp"
            app:layout_constraintStart_toEndOf="@+id/textView7"
            app:layout_constraintTop_toBottomOf="@+id/tvSumaTiempo" />

```

```

        <TextView
            android:id="@+id/textView3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Coste Total [€]:"
            app:layout_constraintStart_toStartOf="@+id/textView7"
            app:layout_constraintTop_toBottomOf="@+id/textView7" />

        <TextView
            android:id="@+id/tvSumaCoste"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginTop="8dp"
            android:layout_marginEnd="32dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toEndOf="@+id/textView3"
            app:layout_constraintTop_toBottomOf="@+id/tvSumaCombustible"
        />

    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>

<Spinner
    android:id="@+id/spTrabajo"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    app:layout_constraintBottom_toTopOf="@+id/lvTrabajos"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvMostrar" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

3. Activity_main2.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context=".AltaTrabajo">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="-90dp">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <EditText
                android:id="@+id/etSuperficie"
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:layout_marginTop="8dp"

```

```

        android:ems="10"
        android:inputType="number"
        app:layout_constraintStart_toStartOf="@+id/etIdentificador"
        app:layout_constraintTop_toBottomOf="@+id/tvDimension" />

<EditText
    android:id="@+id/etIdentificador"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="24dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Trabajo"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4" />

<Button
    android:id="@+id/btGuardarExplanacion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="24dp"
    android:text="Guardar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/tvDimension"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:text="Superficie*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etIdentificador" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="Identificador"
    app:layout_constraintStart_toStartOf="@+id/tvDimension"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/btAnadirMaquina"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:text="Añadir Maquina"
    app:layout_constraintBottom_toBottomOf="@+id/tvMaquinas"
    app:layout_constraintStart_toEndOf="@+id/tvMaquinas"
    app:layout_constraintTop_toTopOf="@+id/tvMaquinas" />

<ListView
    android:id="@+id/lvMaquinas"
    android:layout_width="0dp"
    android:layout_height="400dp"
    android:layout_marginTop="24dp"
    app:layout_constraintEnd_toEndOf="parent"

```

```

        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvMaquinas" />

<androidx.cardview.widget.CardView
    android:id="@+id/cardView2"
    android:layout_width="409dp"
    android:layout_height="40dp"
    android:layout_marginTop="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/lvMaquinas">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="24dp"
            android:text="Consumo total [l/h]:"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/tvConsumoTotal"
            android:layout_width="30dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:text="0"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@+id/textView8"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>

<TextView
    android:id="@+id/tvPersonal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Personal"
    app:layout_constraintStart_toStartOf="@+id/tvDimension"
    app:layout_constraintTop_toBottomOf="@+id/etSuperficie" />

<EditText
    android:id="@+id/etPersonal"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintStart_toEndOf="@+id/tvNumero"
    app:layout_constraintTop_toBottomOf="@+id/tvPersonal" />

<TextView
    android:id="@+id/tvNumero"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"

```

```

        android:layout_marginTop="24dp"
        android:text="Numero*"
        app:layout_constraintStart_toStartOf="@+id/tvPersonal"
        app:layout_constraintTop_toBottomOf="@+id/tvPersonal" />

<TextView
    android:id="@+id/tvRegimen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="Régimen de trabajo"
    app:layout_constraintStart_toStartOf="@+id/tvNumero"
    app:layout_constraintTop_toBottomOf="@+id/tvNumero" />

<TextView
    android:id="@+id/tvMaquinas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="Maquinas"
    app:layout_constraintStart_toStartOf="@+id/tvSuelo"
    app:layout_constraintTop_toBottomOf="@+id/tvSuelo" />

<Spinner
    android:id="@+id/spRegimen"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="24dp"
    app:layout_constraintBottom_toBottomOf="@+id/tvRegimen"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/tvRegimen"
    app:layout_constraintTop_toTopOf="@+id/tvRegimen" />

<TextView
    android:id="@+id/tvSuelo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Tipo de suelo"
    app:layout_constraintStart_toStartOf="@+id/tvPersonal"
    app:layout_constraintTop_toBottomOf="@+id/tvRegimen" />

<Spinner
    android:id="@+id/spSuelo"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    app:layout_constraintBottom_toBottomOf="@+id/tvSuelo"
    app:layout_constraintEnd_toEndOf="@+id/spRegimen"
    app:layout_constraintStart_toEndOf="@+id/tvSuelo"
    app:layout_constraintTop_toTopOf="@+id/tvSuelo" />

<TextView
    android:id="@+id/tvMateriales"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="Materiales"
    app:layout_constraintStart_toStartOf="@+id/tvMaquinas"
    app:layout_constraintTop_toBottomOf="@+id/cardView2" />

<ListView
    android:id="@+id/lvMateriales"

```

```

        android:layout_width="0dp"
        android:layout_height="400dp"
        android:layout_marginTop="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvMateriales" />

<androidx.cardview.widget.CardView
    android:id="@+id/cardView3"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="24dp"
    app:layout_constraintBottom_toTopOf="@+id/btGuardarExplanacion"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/lvMateriales">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="24dp"
            android:text="Coste materiales:"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/tvCosteMateriales"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:text="0"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@+id/textView10"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

<Button
    android:id="@+id/btAnadirMateriales"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:text="AÑADIR MATERIALES"
    app:layout_constraintBottom_toBottomOf="@+id/tvMateriales"
    app:layout_constraintStart_toEndOf="@+id/tvMateriales"
    app:layout_constraintTop_toTopOf="@+id/tvMateriales" />

<TextView
    android:id="@+id/tvError"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="Los campos con asterisco (*) son obligatorios"
    android:visibility="invisible"

```



```

app:layout_constraintBottom_toBottomOf="@+id/btGuardarExplanacion"
    app:layout_constraintEnd_toStartOf="@+id/btGuardarExplanacion"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btGuardarExplanacion"
/>

</androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

4. Activity_maquina.

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.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">

<TextView
    android:id="@+id/tvError"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="Los campos con asterisco (*) son obligatorios"
    android:visibility="invisible"
    app:layout_constraintBottom_toBottomOf="@+id/btAceptarMaquina"
    app:layout_constraintEnd_toStartOf="@+id/btAceptarMaquina"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btAceptarMaquina" />

<TextView
    android:id="@+id/textView20"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="Introduce el modelo"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/etModelo"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="24dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView20" />

<TextView
    android:id="@+id/textView21"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="Introduce el rendimiento*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etModelo" />

```



```

<EditText
    android:id="@+id/etRendimiento"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView21" />

<TextView
    android:id="@+id/textView22"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="Cantidad*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etConsumo" />

<EditText
    android:id="@+id/etCantidadMaquina"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView22" />

<Button
    android:id="@+id/btAceptarMaquina"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="32dp"
    android:layout_marginBottom="32dp"
    android:text="Aceptar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/textView23"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="Introduce el consumo*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etRendimiento" />

<EditText
    android:id="@+id/etConsumo"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView23" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

5. Activity_material.

```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.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">

<TextView
    android:id="@+id/tvError"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="Los campos con asterisco (*) son obligatorios"
    android:visibility="invisible"
    app:layout_constraintBottom_toBottomOf="@+id/btAceptarMaterial"
    app:layout_constraintEnd_toStartOf="@+id/btAceptarMaterial"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btAceptarMaterial" />

<TextView
    android:id="@+id/textView20"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="Introduce el material"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/etMaterial"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="24dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView20" />

<TextView
    android:id="@+id/textView21"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="Introduce el coste*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etMaterial" />

<EditText
    android:id="@+id/etCoste"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView21" />
```

```

<TextView
    android:id="@+id/textView22"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:text="Cantidad*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etCoste" />

<EditText
    android:id="@+id/etCantidadMaterial"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView22" />

<Button
    android:id="@+id/btAceptarMaterial"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="32dp"
    android:layout_marginBottom="32dp"
    android:text="Aceptar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

6. Activity_transporte.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context=".AltaTrabajo">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="-16dp"
        tools:layout_editor_absoluteY="-66dp">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <EditText
                android:id="@+id/etSuperficie"
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:layout_marginTop="8dp"
                android:ems="10"
                android:inputType="number"
                app:layout_constraintStart_toStartOf="@+id/etIdentificador"

```

```

        app:layout_constraintTop_toBottomOf="@+id/tvDimension" />

<EditText
    android:id="@+id/etIdentificador"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="24dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Trabajo"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4" />

<Button
    android:id="@+id/btGuardarExplanacion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="24dp"
    android:text="Guardar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/cardView2" />

<TextView
    android:id="@+id/tvDimension"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:text="Volumen [m³]*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etIdentificador" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="Identificador"
    app:layout_constraintStart_toStartOf="@+id/tvDimension"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/btAnadirMaquina"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:text="Añadir Maquina"
    app:layout_constraintBottom_toBottomOf="@+id/tvMaquinas"
    app:layout_constraintStart_toEndOf="@+id/tvMaquinas"
    app:layout_constraintTop_toTopOf="@+id/tvMaquinas" />

<ListView
    android:id="@+id/lvMaquinas"
    android:layout_width="0dp"
    android:layout_height="400dp"
    android:layout_marginTop="24dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"

```

```

        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvMaquinas" />

<androidx.cardview.widget.CardView
    android:id="@+id/cardView2"
    android:layout_width="409dp"
    android:layout_height="40dp"
    android:layout_marginTop="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/lvMaquinas">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="24dp"
            android:text="Consumo total [l/h]:"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/tvConsumoTotal"
            android:layout_width="30dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:text="0"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@+id/textView8"
            app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>

<TextView
    android:id="@+id/tvPersonal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Personal"
    app:layout_constraintStart_toStartOf="@+id/tvDimension"
    app:layout_constraintTop_toBottomOf="@+id/etSuperficie" />

<EditText
    android:id="@+id/etPersonal"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintStart_toEndOf="@+id/tvNumero"
    app:layout_constraintTop_toBottomOf="@+id/tvPersonal" />

<TextView
    android:id="@+id/tvNumero"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="24dp"

```

```

        android:text="Numero*"
        app:layout_constraintStart_toStartOf="@+id/tvPersonal"
        app:layout_constraintTop_toBottomOf="@+id/tvPersonal" />

<TextView
    android:id="@+id/tvRegimen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="Régimen de trabajo"
    app:layout_constraintStart_toStartOf="@+id/tvNumero"
    app:layout_constraintTop_toBottomOf="@+id/tvNumero" />

<TextView
    android:id="@+id/tvMaquinas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="Maquinas"
    app:layout_constraintStart_toStartOf="@+id/tvPersonal"
    app:layout_constraintTop_toBottomOf="@+id/tvRegimen" />

<Spinner
    android:id="@+id/spRegimen"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="24dp"
    app:layout_constraintBottom_toBottomOf="@+id/tvRegimen"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/tvRegimen"
    app:layout_constraintTop_toTopOf="@+id/tvRegimen" />

<TextView
    android:id="@+id/tvError"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="Los campos con asterisco (*) son obligatorios"
    android:visibility="invisible"

app:layout_constraintBottom_toBottomOf="@+id/btGuardarExplanacion"
    app:layout_constraintEnd_toStartOf="@+id/btGuardarExplanacion"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btGuardarExplanacion"
/>

<TextView
    android:id="@+id/tvDistancia"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="100dp"
    android:text="Distancia [km] *"
    app:layout_constraintBottom_toBottomOf="@+id/tvDimension"
    app:layout_constraintStart_toEndOf="@+id/tvDimension"
    app:layout_constraintTop_toTopOf="@+id/tvDimension" />

<EditText
    android:id="@+id/etDistancia"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="-8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="number"

```

```

        app:layout_constraintStart_toStartOf="@+id/tvDistancia"
        app:layout_constraintTop_toBottomOf="@+id/tvDistancia" />

    </androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

7. Anadir_maquina.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

    <Spinner
        android:id="@+id/spMaquina"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvMaterial" />

    <TextView
        android:id="@+id/tvMaterial"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="32dp"
        android:text="Selecciona Maquina"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btAceptar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="24dp"
        android:layout_marginBottom="24dp"
        android:text="Aceptar"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <TextView
        android:id="@+id/tvCantidad"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="Cantidad*"
        app:layout_constraintStart_toStartOf="@+id/tvMaterial"
        app:layout_constraintTop_toBottomOf="@+id/spMaquina" />

    <EditText
        android:id="@+id/etCantidad"
        android:layout_width="60dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:ems="10"

```



```

        android:inputType="number"
        app:layout_constraintStart_toEndOf="@+id/tvCantidad"
        app:layout_constraintTop_toBottomOf="@+id/spMaquina" />

<TextView
    android:id="@+id/tvError"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="Los campos con asterisco (*) son obligatorios"
    android:visibility="invisible"
    app:layout_constraintBottom_toBottomOf="@+id/btAceptar"
    app:layout_constraintEnd_toStartOf="@+id/btAceptar"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btAceptar" />

<Button
    android:id="@+id/btNuevaMaquina"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="32dp"
    android:text="Añadir otra maquina"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvCantidad" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

8. Anadir_materiales.

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.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">

    <TextView
        android:id="@+id/textView11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:text="Seleccionar materiales"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Spinner
        android:id="@+id/spMateriales"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView11" />

    <TextView
        android:id="@+id/textView12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:text="Cantidad*"

```



```

        app:layout_constraintStart_toStartOf="@+id/textView11"
        app:layout_constraintTop_toBottomOf="@+id/spMateriales" />

<EditText
    android:id="@+id/etCantidadMateriales"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:ems="10"
    android:inputType="number"
    app:layout_constraintBottom_toBottomOf="@+id/textView12"
    app:layout_constraintStart_toEndOf="@+id/textView12"
    app:layout_constraintTop_toTopOf="@+id/textView12" />

<Button
    android:id="@+id/btAceptarMateriales"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="24dp"
    android:text="Aceptar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/tvError"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="Los campos con asterisco (*) son obligatorios"
    android:visibility="invisible"
    app:layout_constraintBottom_toBottomOf="@+id/btAceptarMateriales"
    app:layout_constraintEnd_toStartOf="@+id/btAceptarMateriales"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btAceptarMateriales" />

<Button
    android:id="@+id/btNuevoMaterial"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="32dp"
    android:text="Añadir otro material"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView12" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

9. Item_maquina

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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="wrap_content">

    <TextView
        android:id="@+id/tvItemIdentificador"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="24dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="24dp"
        android:text="TextView"
    >

```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="24dp"
    android:text="Consumo [l/h]:"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="@+id/textView5"
    app:layout_constraintTop_toBottomOf="@+id/textView5" />

<TextView
    android:id="@+id/tvItemConsumo"
    android:layout_width="60dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:text="TextView"
    app:layout_constraintStart_toEndOf="@+id/textView"
    app:layout_constraintTop_toBottomOf="@+id/tvItemCantidad" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Cantidad"
    app:layout_constraintStart_toStartOf="@+id/tvItemIdentificador"
    app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

<TextView
    android:id="@+id/tvItemCantidad"
    android:layout_width="60dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:text="TextView"
    app:layout_constraintStart_toEndOf="@+id/textView5"
    app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

<Button
    android:id="@+id/btEliminarMaquina"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="Borrar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

10. Item_materiales.

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.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="wrap_content">

<TextView
    android:id="@+id/tvItemIdentificador"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="24dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="24dp"
    android:text="Coste del material"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="@+id/textView5"
    app:layout_constraintTop_toBottomOf="@+id/textView5" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Cantidad"
    app:layout_constraintStart_toStartOf="@+id/tvItemIdentificador"
    app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

<TextView
    android:id="@+id/tvItemCantidad"
    android:layout_width="60dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:text="TextView"
    app:layout_constraintStart_toEndOf="@+id/textView5"
    app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

<TextView
    android:id="@+id/tvCosteMaterial"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="16dp"
    android:text="TextView"
    app:layout_constraintBottom_toBottomOf="@+id/textView"
    app:layout_constraintEnd_toStartOf="@+id/btEliminarMateriales"
    app:layout_constraintStart_toEndOf="@+id/textView"
    app:layout_constraintTop_toTopOf="@+id/textView" />

<Button
    android:id="@+id/btEliminarMateriales"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="Borrar"
    app:layout_constraintBottom_toBottomOf="parent"

```

```

        app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

11. Item_trabajo.

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.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="wrap_content">

    <TextView
        android:id="@+id/tvItemIdentificador"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="32dp"
        android:text="TextView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/tvIdentificador"
        app:layout_constraintTop_toBottomOf="@+id/tvItemTipo" />

    <TextView
        android:id="@+id/tvItemSuperficie"
        android:layout_width="50dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintStart_toEndOf="@+id/tvSuperficie"
        app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

    <TextView
        android:id="@+id/tvItemTipo"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="8dp"
        android:text="TextView"
        app:layout_constraintBottom_toTopOf="@+id/tvItemIdentificador"
        app:layout_constraintEnd_toEndOf="@+id/tvItemIdentificador"
        app:layout_constraintStart_toEndOf="@+id/tvTipo"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvTipo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:layout_marginTop="16dp"
        android:text="Tipo:"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvIdentificador"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Identificador:"

```

```

        app:layout_constraintEnd_toEndOf="@+id/tvTipo"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="@+id/tvTipo"
        app:layout_constraintTop_toBottomOf="@+id/tvTipo" />

<TextView
    android:id="@+id/tvSuperficie"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Volumen [m³]:"
    app:layout_constraintStart_toStartOf="@+id/tvIdentificador"
    app:layout_constraintTop_toBottomOf="@+id/tvIdentificador" />

<TextView
    android:id="@+id/tvTiempo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="8dp"
    android:text="Tiempo [h]:"
    app:layout_constraintStart_toEndOf="@+id/tvItemSuperficie"
    app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

<TextView
    android:id="@+id/tvItemTiempo"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintStart_toEndOf="@+id/tvTiempo"
    app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Consumo Total [l/h]:"
    app:layout_constraintStart_toStartOf="@+id/tvSuperficie"
    app:layout_constraintTop_toBottomOf="@+id/tvSuperficie" />

<TextView
    android:id="@+id/tvItemConsumoTotal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="TextView"
    app:layout_constraintStart_toEndOf="@+id/textView6"
    app:layout_constraintTop_toBottomOf="@+id/tvTiempo" />

<Button
    android:id="@+id/btEliminar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="Borrar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvItemTiempo" />

```

```

<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Coste [€]:"
    app:layout_constraintStart_toStartOf="@+id/textView6"
    app:layout_constraintTop_toBottomOf="@+id/textView6" />

<TextView
    android:id="@+id/tvItemCoste"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="TextView"
    app:layout_constraintStart_toEndOf="@+id/textView9"
    app:layout_constraintTop_toBottomOf="@+id/tvItemConsumoTotal" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

12. Item_trabajo_losa.

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.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="wrap_content">

    <TextView
        android:id="@+id/tvItemIdentificador"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="32dp"
        android:text="TextView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/tvIdentificador"
        app:layout_constraintTop_toBottomOf="@+id/tvItemTipo" />

    <TextView
        android:id="@+id/tvItemAncho"
        android:layout_width="50dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintStart_toEndOf="@+id/tvAncho"
        app:layout_constraintTop_toBottomOf="@+id/tvItemIdentificador" />

    <TextView
        android:id="@+id/tvItemTipo"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginBottom="8dp"
        android:text="TextView"
        app:layout_constraintBottom_toTopOf="@+id/tvItemIdentificador"
        app:layout_constraintStart_toEndOf="@+id/tvTipo"
        app:layout_constraintTop_toTopOf="parent" />

```

```

<TextView
    android:id="@+id/tvTipo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="16dp"
    android:text="Tipo:"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/tvIdentificador"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Identificador:"
    app:layout_constraintEnd_toEndOf="@+id/tvTipo"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/tvTipo"
    app:layout_constraintTop_toBottomOf="@+id/tvTipo" />

<TextView
    android:id="@+id/tvAncho"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Ancho"
    app:layout_constraintStart_toStartOf="@+id/tvIdentificador"
    app:layout_constraintTop_toBottomOf="@+id/tvIdentificador" />

<TextView
    android:id="@+id/tvTiempo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Tiempo [h]:"
    app:layout_constraintStart_toStartOf="@+id/tvAncho"
    app:layout_constraintTop_toBottomOf="@+id/textView6" />

<TextView
    android:id="@+id/tvItemTiempo"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    app:layout_constraintBottom_toBottomOf="@+id/tvTiempo"
    app:layout_constraintStart_toEndOf="@+id/tvTiempo"
    app:layout_constraintTop_toTopOf="@+id/tvTiempo" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="Nº Hierros"
    app:layout_constraintStart_toStartOf="@+id/tvAncho"
    app:layout_constraintTop_toBottomOf="@+id/tvAncho" />

<TextView
    android:id="@+id/tvItemHierros"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    app:layout_constraintBottom_toBottomOf="@+id/textView6"
    app:layout_constraintStart_toEndOf="@+id/textView6"

```



```

        app:layout_constraintTop_toTopOf="@+id/textView6" />

<Button
    android:id="@+id/btEliminar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="Borrar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp"
    android:text="Coste [€]:"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="@+id/textView6"
    app:layout_constraintTop_toBottomOf="@+id/tvTiempo" />

<TextView
    android:id="@+id/tvItemCoste"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="TextView"
    app:layout_constraintBottom_toBottomOf="@+id/textView9"
    app:layout_constraintStart_toEndOf="@+id/textView9"
    app:layout_constraintTop_toTopOf="@+id/textView9" />

<TextView
    android:id="@+id/textView13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="Largo"
    app:layout_constraintBottom_toBottomOf="@+id/tvItemAncho"
    app:layout_constraintStart_toEndOf="@+id/tvItemAncho"
    app:layout_constraintTop_toTopOf="@+id/tvItemAncho" />

<TextView
    android:id="@+id/tvItemLargo"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    app:layout_constraintBottom_toBottomOf="@+id/textView13"
    app:layout_constraintStart_toEndOf="@+id/textView13"
    app:layout_constraintTop_toTopOf="@+id/textView13" />

<TextView
    android:id="@+id/textView15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="Alto"
    app:layout_constraintBottom_toBottomOf="@+id/textView13"
    app:layout_constraintStart_toEndOf="@+id/tvItemLargo"
    app:layout_constraintTop_toTopOf="@+id/textView13" />

<TextView
    android:id="@+id/tvItemAlto"

```



```

        android:layout_width="50dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        app:layout_constraintBottom_toBottomOf="@+id/textView15"
        app:layout_constraintStart_toEndOf="@+id/textView15"
        app:layout_constraintTop_toTopOf="@+id/textView15" />

<TextView
    android:id="@+id/textView17"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="Diámetro"
    app:layout_constraintBottom_toBottomOf="@+id/tvItemHierros"
    app:layout_constraintStart_toEndOf="@+id/tvItemHierros"
    app:layout_constraintTop_toTopOf="@+id/tvItemHierros" />

<TextView
    android:id="@+id/tvItemDiametro"
    android:layout_width="30dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    app:layout_constraintBottom_toBottomOf="@+id/textView17"
    app:layout_constraintStart_toEndOf="@+id/textView17"
    app:layout_constraintTop_toTopOf="@+id/textView17" />

<TextView
    android:id="@+id/textView19"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="Separación [cm]"
    app:layout_constraintBottom_toBottomOf="@+id/tvItemDiametro"
    app:layout_constraintStart_toEndOf="@+id/tvItemDiametro"
    app:layout_constraintTop_toTopOf="@+id/tvItemDiametro" />

<TextView
    android:id="@+id/tvItemSeparacion"
    android:layout_width="30dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    app:layout_constraintBottom_toBottomOf="@+id/textView19"
    app:layout_constraintStart_toEndOf="@+id/textView19"
    app:layout_constraintTop_toTopOf="@+id/textView19" />

<TextView
    android:id="@+id/textView16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Vehículo"
    app:layout_constraintBottom_toBottomOf="@+id/tvItemTipo"
    app:layout_constraintStart_toEndOf="@+id/tvItemTipo"
    app:layout_constraintTop_toTopOf="@+id/tvItemTipo" />

<TextView
    android:id="@+id/tvItemVehiculo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="TextView"
    app:layout_constraintBottom_toBottomOf="@+id/textView16"
    app:layout_constraintStart_toEndOf="@+id/textView16"
    app:layout_constraintTop_toTopOf="@+id/textView16" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```