Universidad Zaragoza
1542

Trabajo Fin de Grado

Desarrollo de software para la determinación de los planes de carga de PTF-MAN

Autor

CAC. D. Ahmad Mohammad Alqadi

Directores

Director académico: Prof. D. Joaquín Sancho Val
Director militar: Tte. D. David Perea López

Centro Universitario de la Defensa-Academia General Militar

Año 2018

En primer lugar, quiero agradecer a mi patria por haberme permitido tener una gran experiencia en mi vida, aprender una cultura e idioma nuevo y tener recuerdos que siempre se quedaran en mi mente. Agradezco a mis padres: Mohammad Alqadi y Majedah Alhazaima que son los mejores de mi vida ahora y para siempre, gracias por todo su apoyo y gran esfuerzo, por aguantar mis pesadillas y aguantar la gran distancia que nos separó durante muchos años. Agradezco a mi tutor académico D. Joaquín Sancho Val por la ayuda que me proporcionó para darle vida a este Trabajo Fin de Grado, y por su gran apoyo y esfuerzo continuo. También quiero agradecer a mi tutor militar Tte. D. David Perea López por la gran ayuda que siempre me proporcionaba durante las prácticas externas en el Regimiento de pontoneros y Especialidad de ingenieros número 12. Quiero darle las gracias a mi gran profesora Dª. Carmen Dubois por toda la ayuda que siempre me daba desde el primer día en España en 2014 hasta el actual momento, quiero que sepa que es un ejemplo para seguir y siempre se quedará en mi mente como una gran madre. No me olvido de darle las gracias al Tte. D. Emmanuel Pérez Jiménez por su gran apoyo, es el ejemplo claro del buen teniente que se tiene que seguir siempre, y es un gran compañero que me regaló el Regimiento durante mis prácticas externas y será un compañero para toda la vida. Agradezco a mi novia Leticia Cebrián Reviejo por toda la ayuda y el ánimo que siempre me da en los buenos y malos momentos, también gracias por el gran esfuerzo que siempre da para aguantarme y aguantar mis humores. Quiero agradecer a todos los profesores del CUD que me ayudaron para llegar a esta fase de mi vida, también a todos mis amigos y compañeros de la academia y de fuera que siempre me animaban y me ayudaban para seguir adelante con mi carrera.

RESUMEN

La Compañía de Puentes Flotantes MAN, que se encuentra en el RPEI nº12 en Zaragoza, se encarga del mantenimiento de las piezas de los puentes flotantes, del montaje de todo tipo y clase de dicho puente, así como del transporte de las piezas y herramientas necesarias del puente para montarlo en distintas zonas de operaciones, o ciudades si es para uso civil. Con el fin de simplificar y automatizar esta tarea, el presente Trabajo Fin de Grado tiene como objetivo la creación de una aplicación informática AutoPuente, creada en base a los lenguajes de programación JAVA y XML, y aplicada al programa Android Studio para facilitar su uso.

Para ello se han creado una serie de códigos que permiten calcular las herramientas necesarias partiendo de los datos precisos del puente que se quiere montar. Al mismo tiempo, también se han creado una serie de códigos para calcular el número de viajes con los medios disponibles y, finalmente, indicar el consumo aproximado de combustible que se gasta en los viajes de transporte, de manera que se obtenga la mayor rentabilidad posible, ya que los códigos indican si se tiene que usar todos los vehículos en los últimos viajes o no, para no añadir un coste de combustible innecesario.

Estas funcionalidades se han integrado dentro de la aplicación informática de manera que las pueda utilizar cualquier usuario sin problemas, ya que es una aplicación que se puede usar en el dispositivo móvil.

ABSTRACT

The Company of MAN Floating Bridge, which is in the RPEI nº12 in Zaragoza, is the responsible of the maintenance of the floating bridge's pieces, the montage of all types and all classes of the floating bridge, and the transport of all pieces and tools which are necessary for the montage in different operating areas and cities for the civilian use. With the purpose of helping them in this task, this final degree project has the aim of creation an application *AutoPuente*, which uses JAVA and XML languages, and which is applied on Android Studio program to facilitate its use.

For this they have been have been created a series of cods to calculate the necessary tools with the base which we have the necessary data of the floating bridge which we need. At the same time, they have been created a series of cods to calculate how many travels with our available means, and finally to calculate the approximately consume of fuel that we need for our travels in a profitable way because of the cods which the program uses and which indicates us if we need all vehicles for the last travel or no, to not add unnecessary cost.

These functionalities have been integrated into the application in a way which facilitates the uses of it by any user without having any problems because we can use it on our phones and it has different screens according to what the user needs.

LISTA DE ABREVIATURAS

ET: Ejercito de Tierra.

PTF-MAN: Puente Flotante MAN.

CIA: Compañía.

MINISDEF: Ministerio de Defensa.

PLMM: Plana Mayor de Mando.

RPEI nº12: Regimiento de Pontoneros y Especialidades de Ingenieros número 12.

SC: Sección.

TN: Territorio Nacional.

WORA: Write once, run anywhere.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE

MEMORIA

1. INTRODUCCIÓN DEL TFG

La siguiente memoria expone los resultados y procedimientos utilizados en el Trabajo Fin de Grado con título "*Desarrollo de software para la determinación de los planes de carga PTF-MAN (Figura 1)"* que corresponde al grado de Ingeniería de Organización Industrial impartido por el Centro Universitario de la Defensa en la Academia General Militar de Zaragoza.

Figura 1: Puente flotante MAN. Fuente: CIA de puentes flotantes.

El fin último del trabajo ha sido la creación de una nueva aplicación *AutoPuente* para ayudar a la Compañía de Puentes Flotantes en el Regimiento de Pontoneros y Especialidad de Ingenieros número 12 de Zaragoza. Inicialmente la aplicación calculará las piezas y herramientas necesarias para la instalación de cualquier puente flotante. Para tener una trabajo completo, la aplicación además planteará el plan de carga de cualquier puente flotante desde el RPEI nº12 hasta cualquier ciudad dentro del TN, proporcionando el número de viajes en total y el consumo de combustible en base a la distancia y los vehículos disponibles para el transporte de dicho material.

1.1. Motivación y ámbito de aplicación

El planteamiento de este Trabajo Fin de Grado nace en la CIA de Puentes Flotantes del RPEI nº12. El trabajo de dicha CIA se basa en el montaje de los diferentes puentes flotantes con todas las clases disponibles, y es la CIA que tiene a su cargo los puentes flotantes en toda España. Por lo tanto, si se necesita montar algún puente flotante en otra ciudad dentro del TN, se precisará el cálculo de las necesidades para el puente requerido, también se necesitará saber el número de viajes que realizarán los vehículos y el consumo total de combustible. Por lo que también facilitaría la planificación del montaje de puentes en zonas de instrucción cercanas.

Todo lo mencionado anteriormente se hace actualmente de forma manual en la CIA de Puentes Flotantes, necesitándose mucho tiempo para dicho cálculo, pudiendo llevar a error y a tener que realizar viajes innecesarios con herramientas inútiles, lo que conlleva un coste adicional que se puede evitar con la creación de una aplicación que permita calcular todo sin ningún error.

Para facilitar esta tarea, se ha procedido a crear una aplicación con una serie de códigos que permiten un proceso rápido para calcular las herramientas de cualquier puente, los viajes que tienen que realizar y el consumo de combustible con respecto al número de vehículos

disponibles. Este proceso se basa en el tipo de puente y su longitud, también, en el número de los vehículos disponibles y la distancia que realizan en cada viaje.

Con el fin de comparar la diferencia entre el método nuevo y el antiguo, se realizarán distintas pruebas a la aplicación informática *AutoPuente* que se mostrará posteriormente en la memoria.

1.2. Objetivo y Alcance

El proyecto se ha realizado con la intención de la creación de una aplicación informática *AutoPuente* creando tres funcionalidades principales enfocadas en el tipo, la clase y la luz como una primera funcionalidad, obtener las tablas de todas las piezas y herramientas necesarias como segunda funcionalidad, y la última meter los datos de los vehículos disponibles y la distancia del viaje para obtener el número de viajes y el gasto de combustible total. Con el propósito de alcanzar la finalidad de este trabajo, es necesario definir una serie de objetivos.

Como objetivos generales del proyecto se han considerado los siguientes:

- Conseguir una aplicación con los lenguajes de JAVA y XLM aplicados al programa *Android Studio* de manera que cualquiera puede descargarla en su dispositivo debido a que el sistema de *Android* es muy utilizado.
- Plantear nuevas fórmulas en el lenguaje de JAVA aplicado al programa *Android Studio* para cada tipo de puente, de manera que se pueda obtener el número total de todas las herramientas necesarias para el puente a construir.
- Crear una base de datos y unas fórmulas en el lenguaje de JAVA aplicado al programa *Android Studio* para calcular el número de viajes y combustible total de la manera más eficiente posible.
- Crear un diseño en el lenguaje XML aplicado al programa *Android Studio* que permita al usuario utilizar la aplicación mediante listas desplegables, y así evitar la introducción manual de los datos lo máximo posible.
- Crear un manual de usuario de la aplicación con todas las funcionalidades, para así resolver cualquier tipo de duda que pueda surgir durante su uso.

El funcionamiento de la aplicación se basa en la creación de una posible composición, teniendo en cuenta el tipo, la clase y la longitud (luz) del puente por un lado. Y el número de los vehículos y la distancia del viaje por otro lado.

Una vez abierta la aplicación se mostrarán cuatro pantallas *Android* que se explicarán a continuación:

1. Pantalla Bienvenida: contiene un párrafo de bienvenida a la aplicación, el objetivo de dicha aplicación y, al final de la pantalla, un botón de “Continuar” para poder acceder a la segunda pantalla.
2. Pantalla Auto: en esta pantalla, el usuario introduce el tipo, la clase y la luz del puente. Esta pantalla avisa al usuario de los errores que comete, de manera que no se puede continuar hasta que el usuario introduzca todos los datos de manera correcta. Y al final de la pantalla, hay un botón de “Continuar” para pasar a la siguiente pantalla.
3. Pantalla Tablas: esta pantalla muestra diferentes tablas de todas las herramientas que se necesitan para el montaje de nuestro puente.
4. Pantalla Plan de Carga: aquí el usuario añade los datos de los vehículos disponibles y la distancia que van a realizar. Esa misma pantalla, ofrece el número de idas y venidas que tienen que hacer los vehículos y el consumo de combustible total. Además, muestra avisos en el caso de que sobren vehículos para el último viaje.

1.3. Metodología de trabajo

Este Trabajo Fin de Grado se basa en la creación de una aplicación informática, por tanto cabe destacar que se ha procedido desde el punto de vista práctico con el objetivo de ayudar a la CIA de puentes flotantes de Zaragoza a disminuir los tiempos de cálculo de herramientas y piezas necesarias y, del mismo modo, a economizar el tiempo de preparación de carga necesario para el transporte del puente a cualquier ciudad dentro del TN. La aplicación se creó mediante la combinación de JAVA y XML con *Android Studio*, siendo uno de los objetivos con los que se creó la aplicación la constante mejora de esta. Las herramientas utilizadas son las siguientes (Figura 2):

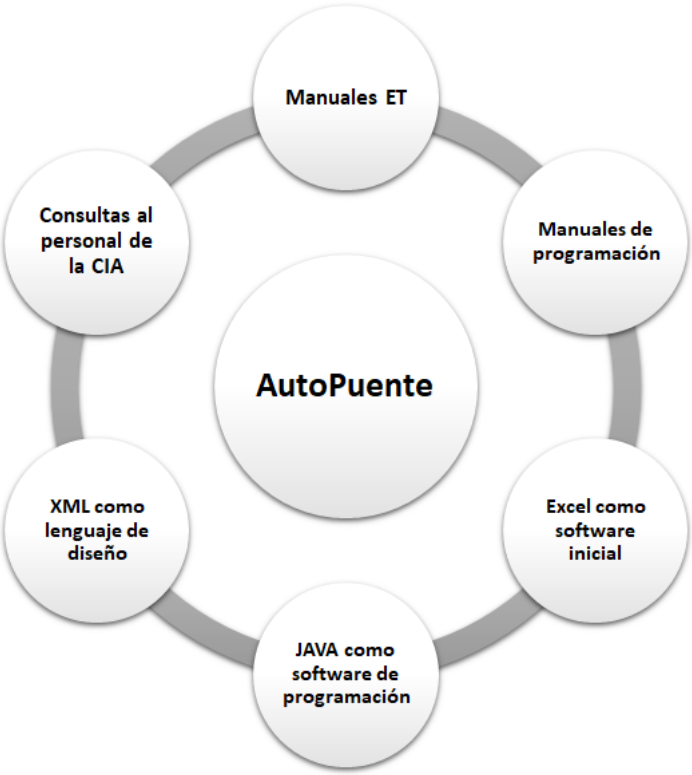


Figura 2: Gráfico de la metodología del trabajo. Elaboración propia.

- Manuales técnicos de puentes flotantes en el Ejército de Tierra.
- Manuales de conocimientos informáticos y programación.
- Excel como software inicial e interfaz de usuario para la aplicación.
- Lenguaje JAVA como software de programación.
- Lenguaje XML como software para la interfaz de usuario.
- Consultas al personal de la CIA de puentes flotantes y al personal de la CIA de puentes fijos.

La aplicación informática *AutoPuente* fue creada en Septiembre del año 2018 con el objetivo de facilitar la tarea de hacer tablas de las piezas y herramientas necesarias y también de hacer el plan de carga para el transporte de cualquier puente, teniendo en cuenta que, al hacerlo manualmente, se necesitaría a dos sargentos trabajando de dos a tres horas para determinar dicho plan. También al finalizar la aplicación *AutoPuente* y ponerla bajo prueba, se detectaron errores que se producían al hacer estos cálculos manualmente[1]. Además, la aplicación está

[1] Uno de los errores que se detectaron con la aplicación fue en el manual "Tareas individuales del pontonero", publicado en 2013, página 86, tabla de puentes. En la segunda columna ponía que puente

hecha de manera que permite calcular el número exacto de los viajes con los vehículos que se requieren, y así no se pagarán más gastos de los necesarios[2]. *AutoPuente* al principio se hizo de manera que estaba integrada en el entorno gráfico de *Microsoft Excel* utilizando el lenguaje de programación *Visual Basic,* pero luego se procedió a modificar los cálculos y estudiar los lenguajes JAVA y XML para posibilitar la integración con el entorno gráfico de *Android Studio*.

A continuación se presenta la planificación en tiempo del trabajo (Tabla 1). El proyecto ha contado con las fases que se muestran en la tabla:

Etapa	Proceso	Fecha Inicio	Duración	Fecha Fin
Fase Informática	Manuales ET	28/06/2018	8	06/07/2018
	Manuales programación en Excel	06/07/2018	3	09/07/2018
	Manuales programación Android Studio	09/07/2018	13	22/07/2018
Fase consultiva	Consultas personal RPEI Nº12	22/07/2018	2	24/07/2018
	Consultas personal AGM	24/07/2018	2	26/07/2018
Fase de herramientas	Cálculo de las piezas generales	26/07/2018	3	29/07/2018
	Cáculo de material de rampa	29/07/2018	2	31/07/2018
	Mejora de los cálculos para tener menos coste	31/07/2018	3	03/08/2018
	Cálculo de herramientas neesarias para el montaje	03/08/2018	1	04/08/2018
	Implementación de piezas generales a la aplicación	04/08/2018	4	08/08/2018
	Implementación material de rampa a la aplicación	08/08/2018	1	09/08/2018
	Implementación de herramientas necesarias para el montaje a la aplicación	09/08/2018	3	12/08/2018
Fase del plan de carga	Cálculo del consumo de combustible en el transporte por carreteras	12/08/2018	2	14/08/2018
	Cálculo de número de viajes totales	14/08/2018	1	15/08/2018
	Implementación del consumo de combustible a la aplicación	15/08/2018	4	19/08/2018
	Implementacoón del número de viajes a la aplicación	19/08/2018	1	20/08/2018
	Mejora de los códigos y de la interfaz de la aplicación	20/08/2018	20	09/09/2018
Fase de control	Verificación de la aplicación	09/09/2018	5	14/09/2018
	Validación de la aplicación	14/09/2018	6	20/09/2018

Tabla 1: Tabla con las fases del proyecto. Elaboración propia.

Con estas fases se realizó un diagrama de Gantt (Figura 3) que permite mostrar de manera gráfica las distintas fases del proyecto y su duración.

tipo puente y clase MLC50 con luz = 94.5m necesitaría 60 pontones extremos y 40 pontones centrales. Pero según los cálculos debería tener 54 pontones extremos y 36 pontones centrales.

[2] Tras una reunión el día 18 de septiembre con el capitán Gordo, jefe de la CIA de puentes flotantes. El capitán en dicha reunión comentó que debido a fallos que se cometieron durante los cálculos, se tuvo que volver a la unidad por falta de material, y así añadiendo unos costes innecesarios e inesperados.

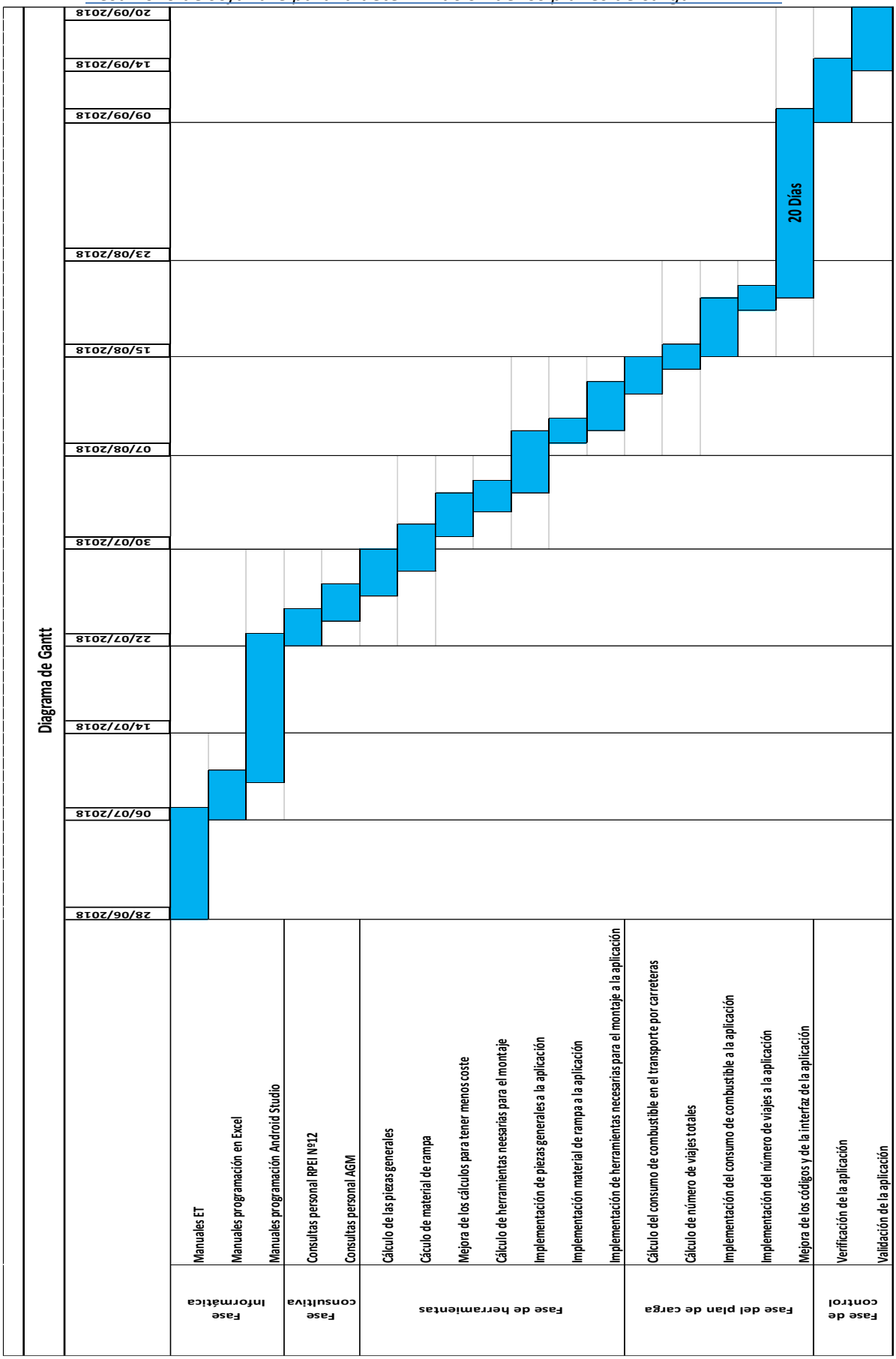


Figura 3: Diagrama de Gantt del trabajo. Elaboración propia.

2. LOS PUENTES FLOTANTES EN EL EJÉRCITO ESPAÑOL

2.1. Historia

El RPEI nº12 se creó en Zaragoza por la Real Orden de 25 de julio de 1815[3]. Está encuadrado en el Mando de Ingenieros de la fuerza terrestre. Este regimiento se creó para dar paso a las tropas mediante la construcción de puentes. El regimiento empezó a participar en las misiones en el extranjero desde el año 1992. El nombre de este regimiento viene del primer batallón llamado "Batallón de Pontoneros I/12" que se encuentra dentro de este.

Los pontoneros fueron así denominados por los puentes que son capaces de construir. Uno de estos puentes es el PTF-MAN, y este regimiento es el único capaz de construirlo ya que no existen otras unidades en el ejército español que tengan el mismo puente.

La antigüedad del uso de los puentes flotantes pertenece al año 1504[4], cuando en la Batalla del Garellano se usó por primera vez una pasarela flotante según instrucciones del ingeniero y capitán Pedro Navarro, bajo las órdenes del Gran Capitán, como parte de una acción envolvente contra el ejército francés. Además, en el año 1659[5] con motivo de la Guerra de Restauración de Portugal, un puente de barcas se tendió sobre el Miño en Salvatierra.
El puente de barcas resultaba especialmente útil en las operaciones militares. Diversos son los tratados que de ello se ocupan. En el Tratado de Artillería de 1753[6] se describe la construcción de puentes militares o plataformas flotantes sobre barcas, pontones y lanchas.

2.2. La Compañía de puentes flotantes

La CIA de puentes flotantes se encuadra dentro del Batallón de Pontoneros I/12. Esta unidad se divide en tres secciones principales: dos de maniobra y una de apoyo. Además, como todas las unidades de España, tiene una oficina de mando y plana mayor para facilitar los documentos, informes y trabajos necesarios para todas las secciones mencionadas.

Figura 4: Escudo de RPEI nº12. Fuente: Internet.

[3] Cfr. Ejército de Tierra, "Regimiento de Pontoneros y Especialidades de Ingenieros nº 12", disponible en www.ejercito.mde.es/unidades/Zaragoza/rpei12/Historial/index.html, consultado el día 25 de septiembre de 2018.

[4] Cfr. Una Pica en Flandes, "¿Existió el Gran Capitán?", disponible en http://www.unapicaenflandes.es/Gran_Capitan.html, consultado el día 12 de septiembre de 2018.

[5] Cfr. Historia desde Benavente, "UN PASO FIJO Y FLOTANTE", disponible en http://historiadesdebenavente.blogspot.com/2016/11/los-puentes-de-barcas-en-espana-1.html, consultado el día 12 de septiembre de 2018.

[6] Cfr. Historia desde Benavente, "UN PASO FIJO Y FLOTANTE", disponible en http://historiadesdebenavente.blogspot.com/2016/11/los-puentes-de-barcas-en-espana-1.html, consultado el día 12 de septiembre de 2018.

A continuación, en la Figura 5 se encuentra el organigrama de la CIA de puentes flotantes en el que se muestran las unidades que tiene a su cargo:

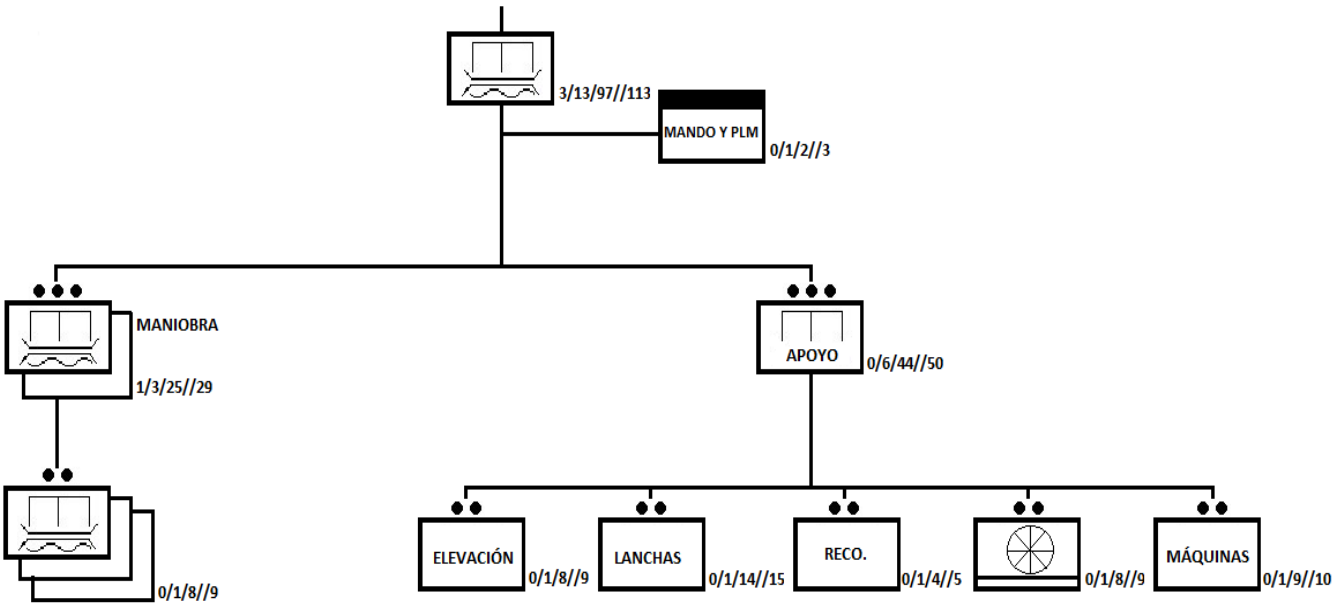


Figura 5: Organigrama de la CIA de puentes flotantes. Fuente: Mando de Ingenieros. Ficha de capacidad CIA puentes flotantes[7].

Dicha CIA tiene varios cometidos que se van a explicar a continuación:

- Montaje y desmontaje de todo tipo y clase de puente.
- Mantenimiento de las piezas.
- Respuesta a las necesidades humanitarias en el caso de necesitar un puente flotante.
- Preparación y transporte de todo el material en el caso atender a las ayudas necesarias.
- Preparación de todos los informes y cálculo del presupuesto total en caso del montaje fuera de Zaragoza.

2.3. Procedimientos para preparación y transporte de PTF-MAN

La CIA de puentes flotantes en Zaragoza es la que se encarga de montar dicho puente en cualquier zona dentro del TN. Se llevan a cabo una serie de trabajos previos antes del montaje del puente que se explicarán a continuación[8] (Figura 6):

[7] En el organigrama aparecen varios elementos que se explicarán a continuación:
- Mando y PLM: La Plana de Mando es un órgano dentro de la CIA que ayuda al mando en labores de personal y logística.
- Los números que se encuentran junto a cada cuadro determinan el personal que hay en esa unidad. Por ejemplo 1/3/25//29: 1 oficial, 3 suboficiales, 25 soldados sumando en total 29.
- Los tres puntos que se encuentra encima de las tres figuras determinan que esa unidad es una unidad de entidad SC.
- Los dos puntos que se encuentra encima de las tres figuras determinan que esa unidad es una unidad de entidad pelotón.
- Los pelotones que se muestran en la figura son: pelotón de elevación, pelotón de Lanchas, pelotón de reconocimiento, pelotón de parque y transporte, y pelotón de estabilización del terreno.

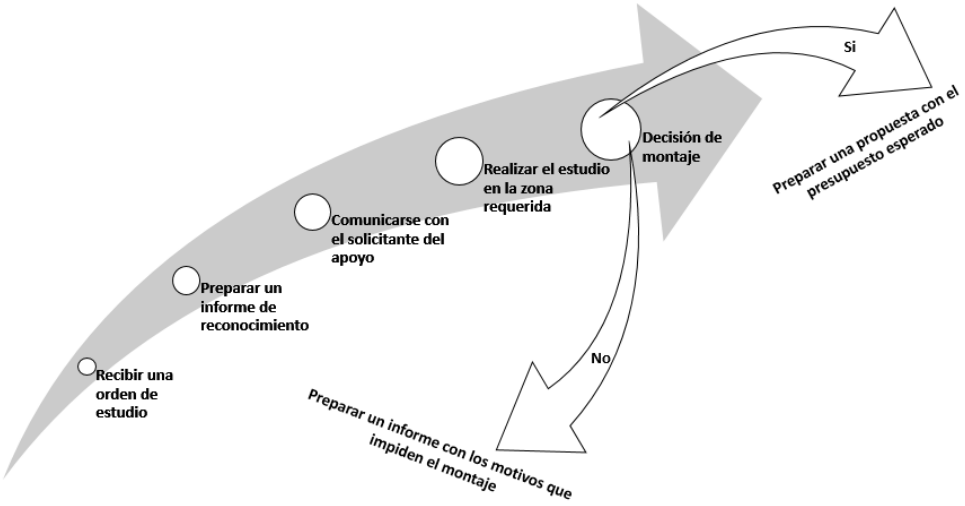


Figura 6: Gráfico de las fases de preparación el montaje fuera de Zaragoza. Elaboración propia.

1. Se recibe la orden de estudiar la posibilidad de tender un puente sobre un río con las exigencias del uso requeridas.
2. La CIA prepara un informe para la preparación de un reconocimiento a la zona mencionada en la solicitud del estudio. El reconocimiento será realizado por el un equipo compuesto de un oficial jefe de la Sección de montaje, un operador de maquinaría y un suboficial.
3. Se comunica con el solicitante del apoyo, que en la mayoría de los casos es el Alcalde de la zona si es de uso civil, y es el jefe de la unidad si es de uso militar. Dicha comunicación estará dirigida a coordinar una reunión del equipo de reconocimiento con el solicitante.
4. Una vez acabada la reunión con el solicitante, el equipo de reconocimiento empieza a realizar el estudio en la ubicación requerida. Se tendrá en cuenta el caudal del río, la diferencia de altura entre las dos orillas, la velocidad de la corriente, la anchura del río y gálibos de acceso a las zonas de trabajo.
5. Tras el estudio realizado, se decide montar el puente o no. En el caso afirmativo, se estudia la posibilidad de montar el puente en otros puntos que resulten más fáciles de acceder por parte de la unidad, teniendo en cuenta las carreteras y viales de ferrocarril (Figura 7).

Figura 7: Puntos posibles para realizar el montaje tras realizar el estudio de la zona. Fuente: Informe de preparación de montaje fuera de Zaragoza (Anexo 1).

[8] Tras una reunión, el día 18 de septiembre, con el capitán Gordo, se pudo obtener la información de cómo se preparaba el montaje de los puentes flotantes fuera de Zaragoza. Dicho capitán facilitó un informe de un ejemplo antiguo cuando se pidió el apoyo de la CIA.

6. En el caso negativo, se prepara un informe de la situación y los motivos que impiden la posibilidad del montaje.
7. Se decide el punto final del montaje requerido con el tipo, la clase y la longitud del puente que se va a montar.
8. Se prepara una propuesta de trabajo con los datos necesarios para realizar el montaje del puente. El jefe de SC. de montaje se encarga de repartir el trabajo entre los suboficiales para calcular todas las herramientas y piezas necesarias para el puente elegido, y también preparar el plan de carga para transportarlo, el número de viajes totales y el consumo de combustible aproximado.

En un futuro próximo, la CIA empezará a utilizar la aplicación *AutoPuente* para hacer la última fase del trabajo con mayor exactitud, y así evitar posibles errores y ahorrar el tiempo perdido en dicha fase.

3. DESARROLLO DEL SOFTWARE

En este apartado se va a explicar primero el material que estaba reflejado en la aplicación *AutoPuente*. Luego se mencionará la manera que se siguió para elegir el lenguaje adecuado de programación. También se va a proceder a explicar la interfaz con los requisitos esperados y los procesos que hace la aplicación. Y por último se va a poner la aplicación bajo prueba para verificar la eficiencia de su funcionamiento.

3.1. Punto de partida: Material y cálculos

3.1.1. Puente flotante MAN

El material del puente MAN es un material pesado ideado para la construcción de puentes flotantes y compuertas, pudiendo ser empleadas también para construir pasarelas flotantes y plataformas de trabajo. Toma su nombre de los pontones centrales y extremos que están acoplados de modo rígido a la flexión en los sentidos longitudinal y transversal. El producto es un tablero flotante para el paso de vehículos. La rampa, que aparece al principio y final, se compone de vigas de metal ligero que permiten el montaje manual.

A continuación se explicarán los diferentes tipos de puentes con sus clases correspondientes, también las diferentes piezas que forman los puentes flotantes con una pequeña explicación de cada una de ellas:

3.1.1.1. Tipos de puentes

Hay diferentes tipos de puentes flotantes dependiendo del uso requerido por parte de la unidad que considera la necesidad de montarlo o no. Los puentes montados pueden ser para el uso civil (Figura 8) o militar según la situación. El objetivo principal de dichos puentes es permitir el acceso de una orilla de un río a la otra, y ese acceso puede ser peatonal o peatonal y de vehículos a la vez. Todo eso requiere un estudio de la situación para determinar el tipo y la clase, algo fundamental y necesario.

3.1.1.1.1. Pasarelas

Las pasarelas son medios de paso peatonales o para vehículos ligeros. Las pasarelas se pueden construir del material que se dispone debido a su simple estructura. Las pasarelas se amarran a tierra de manera que pueden resistir la fuerza del viento. Hay cuatro clases de pasarelas, que son Tipo 1, Tipo 2, Tipo 3 y Tipo 4. La única diferencia entre estas clases es la estructura requerida (Figura 9).

Figura 8: Una pasarela montada con la ayuda de la CIA de puentes flotantes para realizar una maratón en Zaragoza. Fuente: Tareas individuales del pontonero.

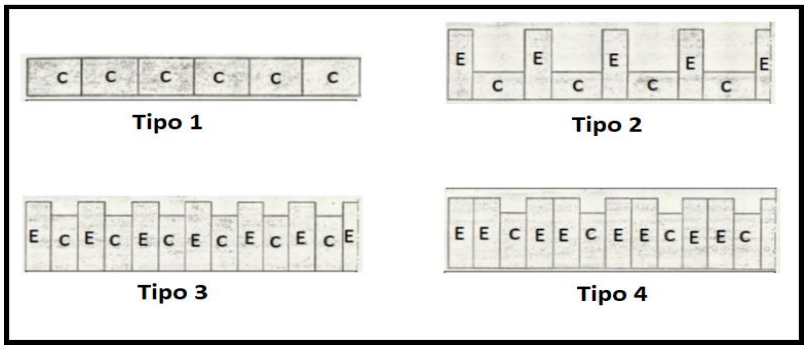


Figura 9: Clases de pasarelas. Fuente: Tareas individuales del pontonero.

3.1.1.1.2. Compuertas

El objetivo de usar las compuertas es el transporte de personal o vehículos. Las compuertas pueden navegar con una velocidad de corriente de hasta 3,6 m/s. Hay diferentes clases de compuertas que son MLC20, MLC50T1, MLC50T2 y MLC80. Los números que aparecen después de MLC[9] representan, aproximadamente, las toneladas que puede soportar el puente, por lo cual un vehículo que exceda la clase marcada no podría pasar el puente porque podría ser un riesgo y causar graves problemas. La clase MLC50 tiene dos tipos diferentes que son el Tipo 1 y el Tipo 2, la única diferencia entre dichos tipos es la estructura del mismo (Figura 10).

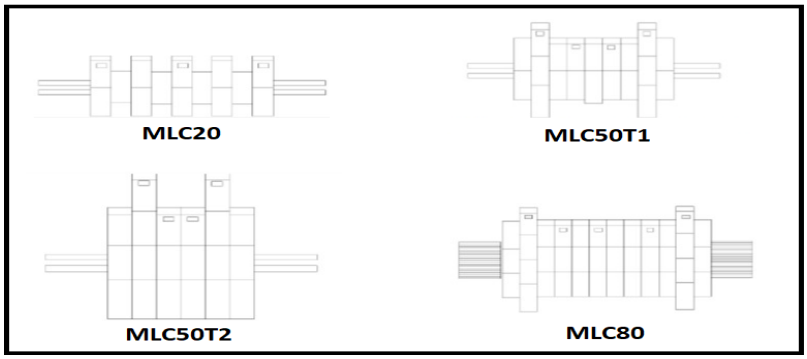


Figura 10: Clases de compuertas. Fuente: Tareas individuales del pontonero.

[9] MLC es la clase militar que aguantaría cada puente. Los vehículos militares son de diferentes clases ya que sus pesos son diferentes. Cada vehículo tiene una pegatina de su clase militar, y así se podrá saber si puede pasar el puente o no.

3.1.1.1.3. Puentes

Las diferencias entre las compuertas y los puentes son las distintas clases que tienen y que los puentes comunican las dos orillas continuamente. Las diferentes clases de puentes son MLC30, MLC50, MLC80A y MLC80B. El puente MLC80 tiene dos tipos: - Tipo A y Tipo B y se diferencian en su estructura (Figura 11).

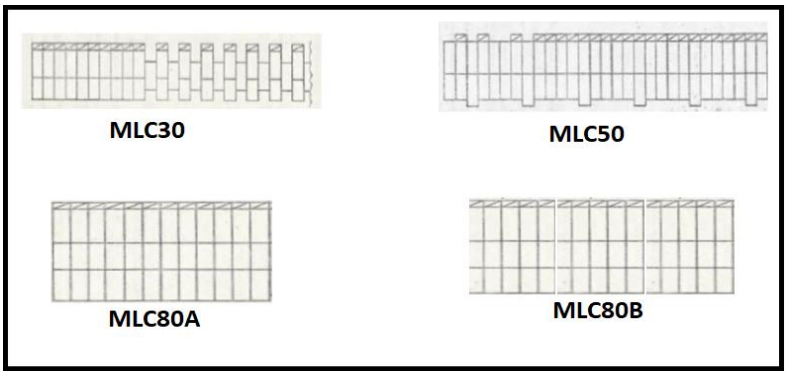


Figura 11: Clases de puentes. Fuente: Tareas individuales del pontonero.

3.1.1.2. Material

El material necesario para el montaje de los puentes flotantes se puede dividir en dos grupos que son el material general de los elementos centrales del puente y el material de las rampas. A continuación se muestra el desglose de los grupos explicándolos de manera detallada:

3.1.1.2.1. Material general

Es el material usado en el montaje de los elementos centrales del puente. A continuación se explicarán los elementos que se tuvieron en cuenta para la aplicación informática *AutoPuente,* así como algunos cálculos utilizados en hallar dichos elementos. Cada tipo y clase tiene unos cálculos (Anexo 4) diferentes a los otros en algunas herramientas:

3.1.1.2.1.1. Pontones

Es un cuerpo en forma de cajón soldado, hermético y flotante. Hay dos tipos diferentes de pontones: los pontones centrales y los pontones extremos. El pontón central es más pequeño que el extremo, tiene una flotabilidad de 5550Kg y es 4,2X2,1X0,73 metros cúbicos. Mientras que la flotabilidad del pontón extremo es de 5750Kg y es 4,73X2,1X0,73 metros cúbicos (Figura 12).

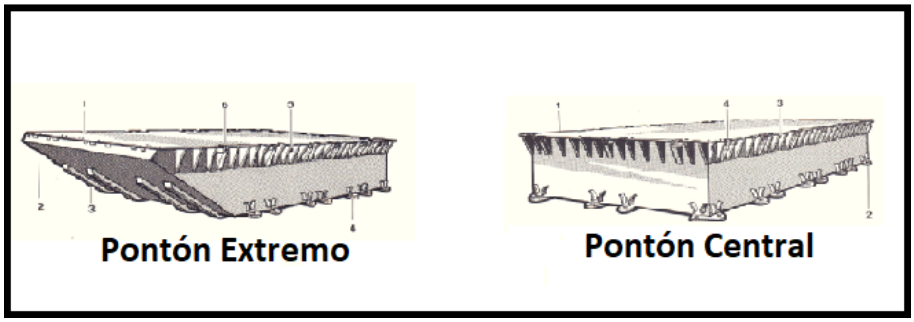


Figura 12: Tipos de pontones. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

El cálculo de los pontones necesarios depende de cada tipo y clase de puente. Por ejemplo, si es de tipo pasarela Tipo 1, se necesitan unos pontones centrales igual a la longitud/4.2m.

3.1.1.2.1.2. Estribos

Son unas piezas usadas para unir los pontones. Con respecto al número de estribos necesarios, son 10 estribos por cada pontón central o extremo. Los estribos se ponen en cajas para facilitar su transporte de manera que 100 estribos caben en una caja pequeña (Figura 13).

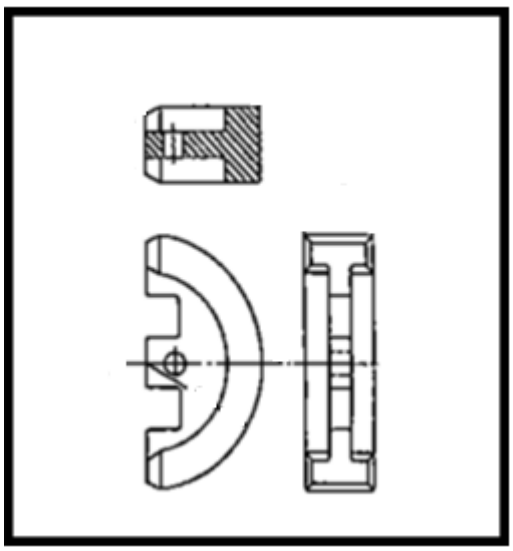


Figura 13: Estribo. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

3.1.1.2.1.3. Horquillas

Son piezas que se usan para acoplar los pontones tanto longitudinal como transversalmente. Se necesitan unas 11 horquillas por cada pontón central o extremo. Las horquillas se meten en cajas para el transporte, de manera que 68 horquillas caben en una caja. Si las horquillas son menos de 11, se llevan sueltas sin cajas en el camión que las transporta (Figura 14).

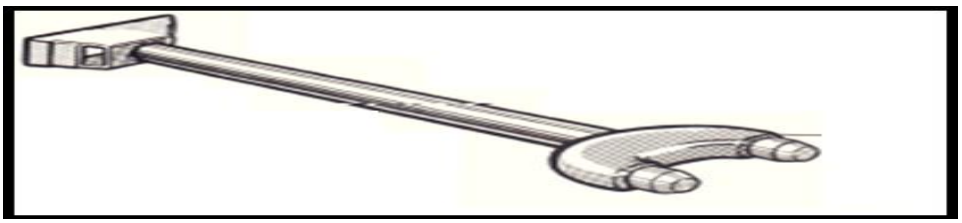


Figura 14: Horquilla. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

3.1.1.2.1.4. Bordillos

Sirven para limitar la vía de circulación. Siempre se lleva una caja de bordillos cuando se decide montar un puente (Figura 15).

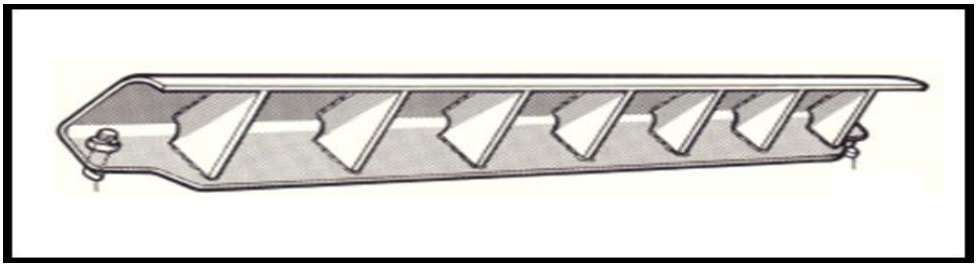


Figura 15: Bordillo. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

3.1.1.2.1.5. Bolardos

Realizan los anclajes o amarres de pontones. También se usan para el montaje sobre la cubierta. El cálculo de los bolardos necesarios depende del tipo y la clase del puente (Figura 16).

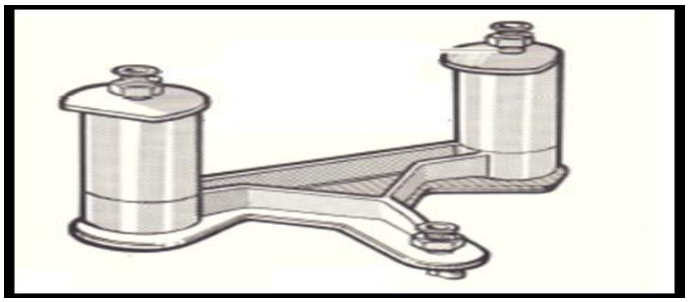


Figura 16: Bolardo. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

3.1.1.2.1.6. Placa deflectora

Chapa para prolongar la proa del pontón extremo para aguantar velocidades de la corriente mayor (Figura 17).

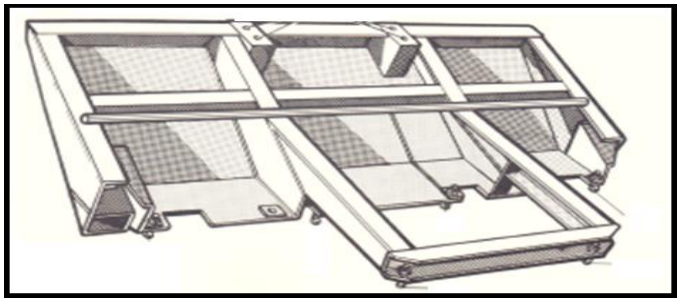


Figura 17: Placa deflectora. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

Cada 7 placas deflectoras se llevan en una jaula grande para el transporte. Si son menos de 4 placas, se llevan sueltas en el camión de transporte.

3.1.1.2.1.7. Cabestrantes

Son piezas hechas para evitar el movimiento lateral del puente por la fuerza del viento o la corriente de agua. Tienen un cable de grosor de 10mm y 120m de longitud que se ancla en la tierra. El peso total de cada cabestrante es de 160Kg. Se necesitan unos cabestrantes iguales a (longitud/10,5)X2,3 (Figura 18).

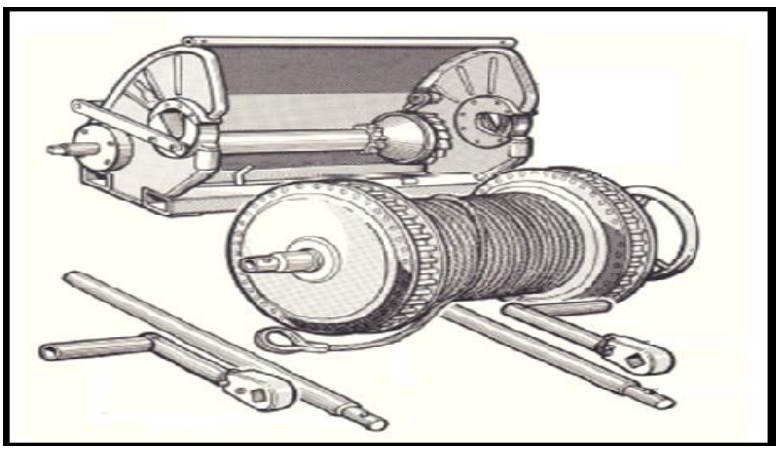


Figura 18: Cabestrante. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

Los cabestrantes se meten en jaulas grandes para el transporte y en cada jaula caben unos 8 cabestrantes. Si los cabestrantes son menos de 5, se llevan sueltos en el camión de transporte.

3.1.1.2.1.8. Soporte de cabestrante

Es una pieza que se coloca encima del pontón, y luego se coloca el cabestrante por dentro de dicho soporte. Se necesitan unos soportes iguales al número de cabestrantes necesarios.

3.1.1.2.1.9. Traveseros

Perfiles huecos que se fijan al pontón mediante tornillos de talón. Sirven como piezas de unión de la placa deflectora o del bastidor del torno. Se necesitan unos traveseros iguales al número de cabestrantes necesarios.

3.1.1.2.1.10. Rodillo

Se fija sobre la placa deflectora o sobre el soporte de cabestrante para guiar el cable del ancla y para guiarlo en caso de desviación oblicua. Se necesitan unos rodillos iguales al número de cabestrantes necesarios (Figura 19).

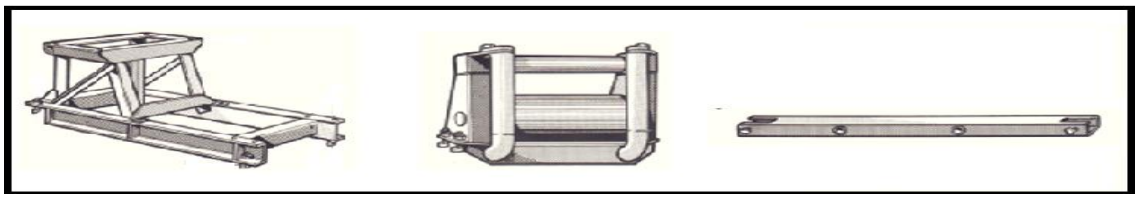


Figura 19: Soporte de cabestrante, travesero y rodillo. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

Cada 13 rodillos se llevan en una jaula grande de rodillos para su transporte. Si los rodillos son menos de 7 se llevan sueltos en el camión de transporte.

3.1.1.2.2. Material rampa

Son los elementos necesarios para el montaje de la rampa donde será el punto de acceso al puente. En las pasarelas no es necesario montarla ya que el puente está amarrado en la tierra. Con respecto al transporte de ese material, si son 10 vigas de rampa, se pone todo el material en una jaula grande, y si son 24 vigas, se pone todo el material en dos jaulas grandes.

A continuación se van a explicar los elementos necesarios para su montaje y los cálculos que se tuvieron en cuenta para la aplicación:

3.1.1.2.2.1. Viga de rampa

Forma la vía de acceso al puente. Las vigas se unen mediante estribos y se apoyan a la tierra (Figura 20).

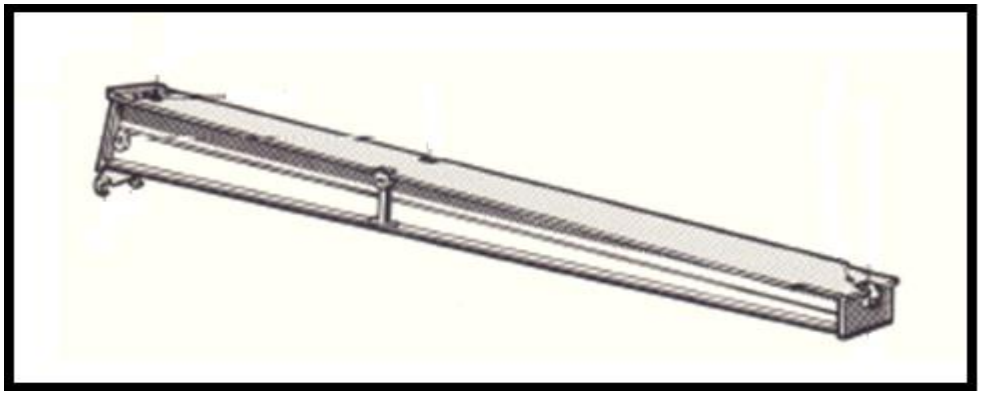


Figura 20: Viga de rampa. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

Las vigas necesarias dependen del tipo y la clase del puente.

3.1.1.2.2. Cumbrera

Es la pieza que permite la conexión entre la compuerta y las vigas de la rampa (Figura 21).

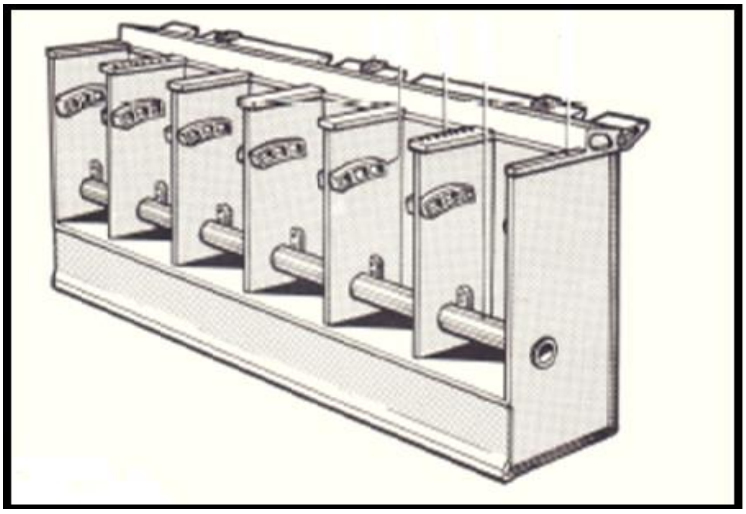


Figura 21: Cumbrera. Fuente: Manual técnico de puentes alemán Hohlplattenbrückengerät.

Si son 10 vigas, se necesitan 2 cumbreras, y si son 24 vigas, se necesitan 4 cumbreras.

3.1.2. Vehículos y barcas

Tanto para el montaje como el transporte del puente se necesitan unos vehículos determinados.

A continuación se va a explicar cada uno de estos vehículos que el RPEI nº12 tiene disponibles:

3.1.2.1. Vehículos y barcas de montaje

Hay dos vehículos que son necesarios en la zona de trabajo donde se monta el puente flotante: el manipulador telescópico[10] y la grúa móvil[11]. El manipulador telescópico tiene que ser transportado en un remolque de un camión tipo M250 o Vempar debido a la dificultad que conlleva realizar unos viajes largos a su velocidad, que es muy baja. En cambio la grúa luna hace el viaje por las carreteras sin necesitar a otro medio de transporte (Figura 22).

Otro material necesario para el montaje es la barca SCHOTTEL, que se usa para mover los pontones una vez están encima del agua para facilitar el montaje, también se utilizan para realizar la navegación con las compuertas. Las SCHOTTELS se transportan en los remolques de los vehículos tipo Vempar (Figura 23).

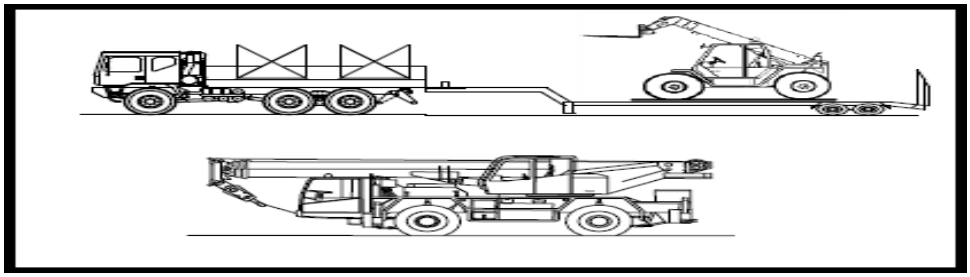


Figura 22: Merlo en un gontrailer y Grúa Luna en una carretera. Fuente: Manual técnico de puentes MT6-401.

[10] El manipulador telescópico usado en el ejército de tierra es de tipo Merlo.

[11] La grúa móvil usada en el ejército de tierra es de tipo Luna.

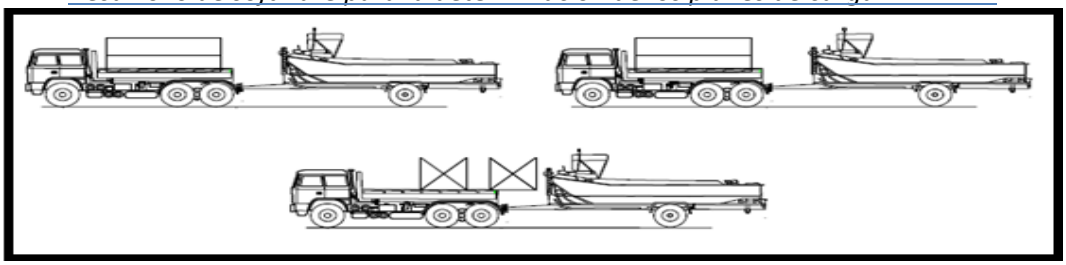


Figura 23: Schottel transportado en un gontrailer tipo Vempar. Fuente: Manual técnico de puentes MT6-401.

3.1.2.2. Vehículos de transporte

El RPEI nº12 dispone de diferentes vehículos para transportar el material necesario para el montaje del puente. Los vehículos que se usan para transportar los pontones y las jaulas grandes son los camiones de tipo M250 y Vempar con sus correspondientes remolques. Mientras que el camión 7217 se usa para el transporte de las jaulas grandes y no suele cargar remolque ya que su potencia es demasiado pequeña (Figura 24).

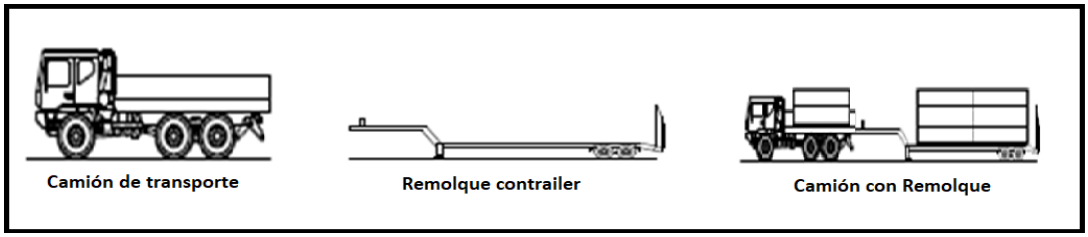


Figura 24: Camión M250 con su correspondiente gontrailer y la capacidad de la caja y del gontrailer. Fuente: Manual técnico de puentes MT6-401.

Con respecto al consumo de combustible, a continuación se presenta una tabla facilitada por el personal de la CIA de puentes flotantes que muestra el consumo según el vehículo usado (Tabla 2).

Modelo	Consumo (L) a los 100 KM
Vehículo Ligero Logístico(tipo furgoneta/representación):	13
Vehiculo Ligero TT. (tipo NISSAN PATROL O SIMILAR):	15
Vehiculo Ligero TT. (tipo AMBUANCIA IVECO O SIMILAR):	16
Vehiculo Ligero TT. (tipo NISSAN PATROL BLINDADO):	20
Vehiculo Ligero TT. (tipo URO VAMTAC):	20
Vehiculo Ligero TT. (tipo URO 3TM):	30
Vehiculo Medio Logístico(tipo CN NISSAN M-110):	23
Vehiculo Medio TT3 Tm(tipo CNTT PEGASO 7217; 7226 O SIMILAR):	35
Vehiculo Pesado TT5 Tm(tipo CNTT PEGASO 7323 O SIMILAR):	46
Vehiculo Pesado TT10 Tm(tipo CNTT IVECO M-250):	55
Vehiculo Pesado Logístico(tipo CN VEMPAR 10 Tm):	55
Vehiculo Pesado Logístico(tipo CN VEMPAR 15 Tm):	60
Vehiculo Pesado Logístico(tipo CN VEMPAR 32 Tm):	60
Cabezas Tractoras:	66
Microbús 15-30 plazas:	25
Microbús 30-60 plazas:	35
Grúa Luna AT 2020; AT 35/32 O SIMILAR:	60

Tabla 2: Tabla de los consumos de combustible (en litros) de los vehículos militares a los 100KM. Fuente: Mando de Ingenieros.

3.2. Análisis y elección del lenguaje de programación

En este apartado se van a explicar las ventajas y desventajas de diversos lenguajes de programación para nuestra aplicación. Posteriormente se elegirá el lenguaje más adecuado para cumplir requisitos mencionados anteriormente y que faciliten el manejo de la aplicación por cualquier persona.

Debido a la gran variedad de los lenguajes de programación, y tras consultar a diferentes programadores, se van a estudiar solo cuatro: C#, PHP, Java y XML aplicados al entorno de *Android Studio*, y por último Visual Basic.

3.2.1. Lenguaje C#

Es un lenguaje base para el desarrollo de la plataforma de Microsoft. Se construyó con la base de los lenguajes C y C++[12]. Hoy en día, muchas páginas web utilizan dicho lenguaje. C# soporta toda la plataforma de ASP.NET, permitiendo a los programadores construir sitios de web dinámicas.

ASP.NET facilita la programación de aplicaciones en múltiples capas lo que, en definitiva, se traduce en la total separación entre lo que el usuario ve y lo que la base de datos tiene almacenado.

Sus ventajas son:

- Un fácil mantenimiento de grandes aplicaciones.
- Un incremento de la velocidad de respuesta de los servidores.
- Una mayor seguridad y, sobretodo, el poder contar con todo el respaldo de Microsoft.

Sus desventajas son:

- La utilización de muchos recursos puede repercutir en un hospedaje bastante costoso.
- El requisito de pagar una licencia para sus herramientas de desarrollo.

3.2.2. Lenguaje PHP

Es un lenguaje de uso general de código por parte del servidor[13]. Originalmente estaba diseñado para el desarrollo web de contenido dinámico. Es uno de los primeros lenguajes que pueden incorporarse directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.

PHP es una plataforma web que mantiene operando a más de 20000 millones de sitios WEB incluyendo algunos tan populares como Facebook y Wikipedia. Cada blog construido sobre Wordpress o cada web diseñada sobre Grupal usa este lenguaje de programación Open Source.

No necesita ser complicado para ejecutarse y, para su funcionamiento, sólo hay que tener instalado “Apache” con las librerías de PHP.

Sus ventajas son:

- Es un lenguaje multiplataforma Open Source.
- Cuenta con la una capacidad de conexión con la mayoría de los sistemas de gestión de base de datos como MySQL, Postgres y SQL Server.
- No requiere una definición de tipos de variables ni manejo detallado.

[12] Cfr. Introducción a C#, Miguel Muñoz Serafín, consultado el día 27 de septiembre de 2018.

[13] Cfr. Programación en PHP a través de ejemplos, Manuel Palomo Duarte, Universidad de Cádiz, Ildefonso Montero Pérez, universidad de Sevilla, disponible en http://servicio.uca.es, consultado el día 27 de septiembre de 2018.

Sus desventajas son:

- Todo el trabajo lo realiza el servidor en lugar del cliente. Por tanto, puede ser más ineficiente a medida que las solicitudes aumentan de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- Dificulta la organización por capas de la aplicación.

3.2.3. Android Studio

Utiliza los lenguajes JAVA y XML para la programación de aplicaciones[14]. JAVA es un lenguaje de programación y una plataforma informática con actividad comercial desde el año 1995. Es fiable, rápido y seguro. Tiene un propósito general, concurrente, orientado a los objetos y fue diseñado específicamente para tener tan pocas dependencias de implementaciones como fuera posible. JAVA fue creado por WORA, se ejecuta en una máquina virtual sin importar la arquitectura del dispositivo, y se utiliza en *Android Studio* para manejar los códigos de los cálculos necesarios. En cambio, el lenguaje XML se utiliza para el diseño de las diferentes aplicaciones.

Actualmente estos dos lenguajes son los más utilizados en instituciones financieras y banca, en línea y no en línea. Los portales como Linkedin, Google+, Amazon están basados en esta plataforma, y también Facebook lo utiliza en algunos de sus servicios.

Sus ventajas son:

- Es un lenguaje multiplataforma y es Open Source.
- Está soportado por toda la gama de dispositivos basados en *Android*.
- Los móviles que no soportan *Android* pueden utilizar las aplicaciones de *Android Studio* en línea.

Sus desventajas son:

- Requiere un intérprete, por lo que en algunas ocasiones tiende a ser lento, aunque correctamente eficiente comparado con C#.
- Los móviles que no soportan *Android* tienen que utilizar las aplicaciones de *Android Studio* en línea, lo que puede causar problemas si el usuario está en una situación sin internet.

3.2.4. Lenguaje Visual Basic

Es un lenguaje orientado a objetos desarrollado por Microsoft[15]. Se utiliza para diseñar aplicaciones y programas informáticos de bajo nivel. También se usa para la programación de interfaces gráficas de usuarios para los sistemas, utilizando infraestructuras de Apis como Motiv y Xveiw. Son realizados usualmente en lenguajes, pero organizando el código de manera que parezcan objetos.

Sus ventajas son:

- Fácil de usar y programar ya que la gran mayoría de las funcionalidades no hace falta escribirlas en forma de código porque se pueden elegir entre las operaciones gráficas que facilita el programa.
- Facilidad de desarrollar aplicaciones en poco tiempo.

[14] Cfr. Desarrollo de Aplicaciones para Android, Universidad de Alicante, disponible en http://www.jtech.ua.es, consultado el día 9 de julio de 2018.

[15] Cfr. Manual práctico de programación en Visual Basic.NET 2016, Carlos Capellán, Centro de formación y desarrollo integral Padre Fantino, disponible en http://www.liccarloscapellan.com, consultado el día 6 de julio de 2018.

Sus desventajas son:

- No se pueden desarrollar aplicaciones de alto nivel como las aplicaciones basadas en el lenguaje JAVA.
- Menor velocidad o eficiencia en las aplicaciones en el proceso de cálculo con objetos.
- No permite el uso de las aplicaciones con los móviles, únicamente con los ordenadores.

3.2.5. Elección del lenguaje de programación

C# se descarta debido a que las funcionalidades que proporciona son demasiado caras. Además, se requiere licencia para el uso de sus herramientas, por lo cual resulta más difícil elegirlo como lenguaje de programación.
PHP se descarta también por las diferentes desventajas que tiene, sobretodo porque requiere insertar paquetes específicos que pueden dificultar el proceso de programación. Otro motivo para descartarlo es que requiere mucha experiencia para que el código no se vea afectado al mezclar HTML y PHP.
Se ha elegido *Android Studio* como el lenguaje de programación (Anexo 2) por ser el más adecuado para la consecución de los objetivos mencionados anteriormente de la manera más sencilla. De esta forma, se podrán aprovechar las ventajas que ofrece *Android Studio* para realizar las siguientes tareas:

- Desarrollar la aplicación en un entorno grafico ofrecido por Android para poder usarla en los móviles.
- Se pueden hacer todos los cálculos necesarios usando el lenguaje de JAVA.
- XML ofrece la capacidad de diseñar una interfaz sencilla y fácil de usar por parte del usuario.

Visual Basic no sería el lenguaje adecuado por no ofrecer el uso de la aplicación a través de nuestros móviles. Además, el nivel de dicho lenguaje es bajo y no permite cumplir todas las necesidades requeridas.

3.3. Diseño de la interfaz y sintaxis

En este apartado, se va a explicar la interfaz con diferentes pantallas de la aplicación y todas las funcionalidades que facilita *AutoPuente*. Para poner en contexto su funcionamiento, también se va a proceder a explicar los requisitos que se definieron en la primera fase del proyecto, y cómo se procedió para tener una aplicación completa para el personal de la CIA de puentes flotantes en la última fase de su trabajo.

3.3.1. Requisitos

Una vez se conoció el tipo de lenguaje que se iba a utilizar, se procedió a recoger una serie de requisitos por medio de unas consultas y entrevistas del personal de la CIA de puentes flotantes. Con el propósito de alcanzar el éxito de este trabajo, hay que definir una serie de objetivos que se pueden definir en el sistema **SMART**:

-**S** (Specific): Debe ser específico, lo más concreto posible.

-**M** (Measurable): Debe ser medible, por lo que ha de ser una meta cuantificable.

-**A** (Achievable): Debe ser alcanzable.

-**R** (Reasonable): Debe ser realista, dentro de nuestras posibilidades.

-**T** (Time bound): Debe estar definido en el tiempo, dentro de una línea temporal.

A continuación se muestran los requisitos que se tendrán en cuenta:

- Ser una aplicación que se pueda usar a través de los móviles.
- Que sea de fácil uso e intuitiva para el personal de la CIA.
- Tener una interfaz atractiva.

- Usar listas desplegables para evitar la introducción manual de los datos.
- Calcular todas las herramientas necesarias para el montaje, y calcular las jaulas que se van a necesitar para el transporte del material.
- Realizar los cálculos de forma rápida y eficiente.
- Tener una base de datos sobre los vehículos de los que dispone la CIA para el uso de transporte.
- Calcular las idas y venidas que tienen que realizar los vehículos tras la elección de los vehículos disponibles.
- Calcular el consumo total de combustible en litros tras introducir la distancia del viaje en kilómetros.
- Tener una ventanilla de avisos importantes para el usuario, como por ejemplo si sobran vehículos o si no se puede transportar un material específico debido a la falta de un tipo de vehículo de transporte.
- Poder modificar los datos de forma manual y sencilla.

3.3.2. Casos de uso

Un caso de uso describe la interacción entre el usuario y el programa. Los casos de uso que se ven en el siguiente gráfico describen dicha interacción entre nuestro principal usuario que es el RPEI nº12 y la aplicación *AutoPuente.* Estos casos de uso son las principales funcionalidades que aparecen en la aplicación. Así, en la Figura 25, se muestra este diagrama creado con el programa informático *Visual Paradigm.*

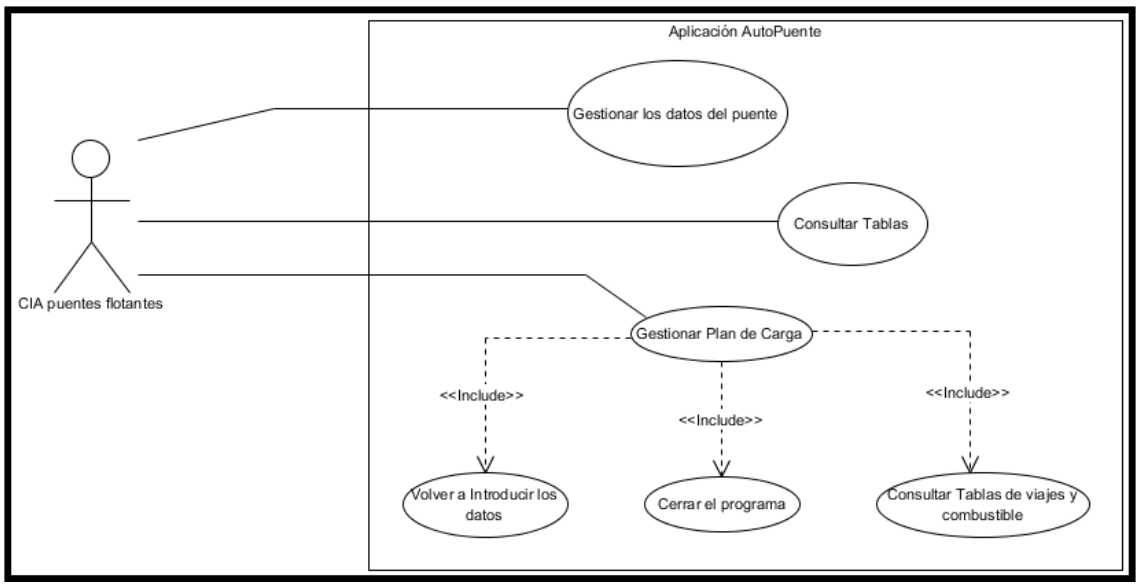


Figura 25: Diagrama con casos de uso de AutoPuente. Elaboración propia.

Como se observa en la figura anterior existen tres casos de usos principales en los que se incluyen otros de menor importancia pero necesarios para cumplir con todos los requisitos para que la aplicación funcione sin ningún problema.

El primero de los casos de usos se denomina “Gestionar los datos del puente”. El usuario desea consultar las tablas de todas las herramientas necesarias para el montaje de su puente flotante. El usuario introduce los datos necesarios, que serán guardados y validados.

El segundo caso de uso se denomina “Consultar Tablas” y empezaría a funcionar una vez se hayan introducidos los datos necesarios. Las tablas serán protegidas de manera que el usuario

no puede cambiar ni introducir datos en las mismas. Esas serán las tablas del material general, el material de la rampa y las herramientas necesarias para el montaje.

El último caso de uso se denomina "Gestionar Plan de Carga" que incluye otras tres funcionalidades que se explican a continuación. El usuario introduce el número disponible de cada tipo de vehículo, además introduce los datos del viaje (distancia, origen y destino). Los datos se guardan y se validan y, a continuación, el caso de uso "Consultar Tablas de viajes y combustible" empieza a funcionar ofreciéndole al usuario el número exacto de viajes y el consumo de combustible de todos los vehículos.

El segundo caso de uso mencionado se denomina "Volver a introducir los datos", se usaría si el usuario desea cambiar el tipo o cualquier otro dato del puente. Y el último caso de uso, que incluye el caso de uso Gestionar Plan de carga, se denomina "Cerrar el programa", se utilizaría si el usuario desea salir de la aplicación.

El funcionamiento del sistema de casos de uso consta de los siguientes pasos:

1. El usuario introduce los datos necesarios para el montaje.
2. El sistema comprueba los datos introducidos.
    A. Los datos están correctos.
        - El usuario puede continuar y consultar las tablas necesarias.
    B. Los datos no están correctos.
        - El sistema avisa con dicha novedad.
        - El usuario añade los datos que faltan o corrige los datos introducidos.
        - El usuario puede continuar y consultar las tablas necesarias.
3. El sistema usa su algoritmo para calcular todas las herramientas y mostrar las tablas.
4. El usuario desea continuar con el plan de carga para el transporte del puente.
5. El usuario introduce los datos necesarios, así como la distancia del viaje.
6. El sistema comprueba los datos introducidos.
    A. Los datos están correctos.
        - El sistema usa su algoritmo para mostrar el número de viajes, el consumo de combustible y avisos varios para el usuario.
    B. Los datos no están correctos.
        - El sistema avisa con dicha novedad.
        - El usuario corrige los datos introducidos.
        - El sistema usa su algoritmo y muestra todos los datos mencionados anteriormente.
7. El usuario comprueba los datos obtenidos.
    A. Los datos son óptimos.
        - El usuario guarda los datos y sale de la aplicación.
    B. Los datos no son óptimos.
        - El usuario vuelve a la primera página para introducir otros datos.

3.3.3. Interfaz gráfica

La aplicación denominada *AutoPuente* ha sido diseñada usando el entorno gráfico de *Android Studio*, con los lenguajes JAVA y XML. Dichos lenguajes facilitaron el trabajo y los cálculos a alto

nivel, además el lenguaje XML facilitó la tarea de tener un diseño atractivo y fácil de usar e interpretar por parte del usuario. Se ha redactado un manual de usuario para facilitar el uso de la aplicación (Anexo 3).

Como se mencionó anteriormente en el apartado de "Metodología de trabajo", la aplicación consta de cuatro pantallas. A continuación se va a explicar cada una de ellas:

Pantalla Bienvenida: En la pantalla de bienvenida (Figura 26), solo se menciona el objetivo de dicha aplicación con el nombre del autor y el escudo del RPEI nº12.

Figura 26: Pantalla bienvenida. Elaboración propia.

Pantalla Auto: En esta pantalla (Figura 27), el personal de la CIA introduce todos los datos necesarios para el puente requerido. En el caso de introducir algún dato incorrecto, la aplicación avisa al personal y no le permite continuar.

Figura 27: Pantalla auto. Elaboración propia.

Pantalla Tablas: Aquí (Figura 28) se muestran todas las herramientas que son necesarias para el montaje. Estos datos salen automáticamente y de forma rápida tras introducir los datos en la pantalla Auto.

Figura 28: Pantalla tablas. Elaboración propia.

Pantalla Plan de Carga: Finalmente, cuando el usuario desea hacer el plan de carga para el transporte del puente, elige el número disponible de cada vehículo (Figura 29) en la CIA. Después introduce los datos necesarios del viaje (la distancia del viaje en especial). Así, el programa devuelve de forma automática el número de viajes, el consumo total de combustible y unos avisos para el usuario.

Figura 29: Pantalla plan de Carga. Elaboración propia.

Finalmente, cuando el usuario desea hacer el plan de carga para el transporte del puente, elige el número disponible de cada vehículo en la CIA. Después introduce los datos necesarios del

viaje (la distancia del viaje en especial). Así, el programa devuelve de forma automática el número de viajes, el consumo total de combustible y unos avisos para el usuario.

Para un funcionamiento más detallado con los mensajes de avisos, puede consultar el manual de usuario en el Anexo 3.

3.3.4. Funcionamiento

Los códigos que usa la aplicación se determinaron de forma manual usando manuales del Ejercito de Tierra y ejemplos varios de la CIA de puentes flotantes. Para consultar todos los cálculos de forma más detallada, puede mirar el Anexo 4.

Como se mencionó anteriormente, la aplicación tiene tres procesos principales que dependen directamente del tipo, clase y luz del puente, así como de los vehículos disponibles para realizar el transporte.

3.3.4.1. Proceso de cálculo de herramientas necesarias

Las herramientas de los puentes flotantes son varias. Y cada tipo y clase de puente tiene unos cálculos distintos a los otros (Anexo 4). Las piezas que no son pontones centrales ni extremos se empaquetan en jaulas para facilitar su transporte. La aplicación en este proceso depende del tipo, la clase y la luz del puente. Al introducir estos datos y pulsar el botón Continuar, la aplicación usa los cálculos mencionados en el Anexo 4 y devuelve todas las tablas necesarias.

3.3.4.2. Proceso de cálculo de viajes necesarios

El número de viajes que se realizan depende del número total de los pontones, las jaulas y los vehículos de transporte. El programa empieza a realizar ciclos de viajes invisibles para el usuario según la capacidad de cada vehículo y lo que puede transportar en cada viaje (Figura 29 en el apartado 3.3.3.). Finalmente, la aplicación muestra el número total de viajes que hay que realizar (idas y venidas).

3.3.4.3. Proceso de cálculo de combustible

El consumo de combustible depende de tres cosas principales: primero, el número de viajes que se van a realizar; segundo, el tipo de vehículo (Tabla 2 apartado 3.1.2.3.); y por último, los avisos para el usuario. Si la aplicación avisa al usuario de que en el último viaje sobran vehículos, ésta descontará el consumo de los vehículos sobrantes de manera automática.

3.4. Verificación y validación

En este apartado se ha procedido a evaluar todas las funcionalidades creadas en la aplicación *AutoPuente* sobre el transporte de los distintos puentes flotantes. Para llevarlo a cabo, se utilizaron tres plantillas Excel utilizadas anteriormente en el RPEI nº12 para determinar todas las herramientas de un ejemplo de 2013, el número de viajes necesarios para el transporte y el consumo de combustible asociado. Se utilizará la aplicación con el mismo ejemplo que sale en las plantillas para evaluar su funcionamiento.

A su vez, con el objetivo de comprobar la satisfacción de los miembros del RPEI nº12 se realizó un cuestionario (Anexo 5) que evaluaba una serie de aspectos relativos al funcionamiento, uso y diseño de las funciones de dicha aplicación.

3.4.1. Verificación

El objetivo principal de este Trabajo Fin de Grado es ayudar al personal de la CIA de puentes flotantes para reducir el tiempo implicado en calcular las herramientas y el plan de carga de cualquier puente flotante. Por lo tanto, se procedió a la verificación de *AutoPuente* con un ejemplo de montaje de puente flotante MAN tipo puente, clase MLC50 y 115.5 metros de

longitud. Dicho ejemplo se llevó a cabo el año 2013 en la orilla del río cerca de la plaza del Pilar en Zaragoza por un oficial de la CIA[16].

A continuación se presentarán unas plantillas de Excel que se utilizaron en 2013, y unas capturas de pantalla de la aplicación *AutoPuente* para comprobar su buen funcionamiento.

Figura 30: Datos necesarios para el montaje del puente en 2013. Fuente: CIA puentes flotantes.

En esta figura, se pueden ver los datos principales que se necesitarían para calcular todas las herramientas del puente.

[16] Capitán. D. Joaquín Peralta Español.

MATERIAL	CANTIDAD POR ELEMENTO	TOTAL	JAULAS / TRANPORTE
P. extremo	6	66	
P. central	4	44	
estribos	100	1100	11 cajas
horquillas	110	1210	18 cajas
cabestrantes	///////	26	3 jaulas pequeñas mas 2 sueltos
placas deflectoras	5	55	8 jaulas pequeñas
traveseros	5	55	2 Jaulas pequeñas rodillos
soportes cabestrante	///////	26	
rodillos	//////	26	
bolardos		10	
duques de alba		4	
bordillos			1 jaula bordillos

Material rampas			
cumbreras		4	2 jaulas grandes
vigas de rampa		20	
finales de rampa			
Postecillos		4	
Gatos		4	
Cable		4	

CAJA HERRAMIENTAS CONTENIENDO:
2 martillos
6 martillos fibra
8 barras de pincho
4 barras de acoplamiento
2 barras de nivelación
6 llaves del 32
4 escalerillas
Amarras
4 cáncamos de cabestrante
4 cadenas
20 piquetes

Figura 31: Tablas de las herramientas. Fuente: CIA puentes flotantes.

En la Figura 31 se puede comprobar que todos los datos que calcula *AutoPuente* son iguales al ejemplo de 2013.

VEHICULOS PARA LA CARGA : 2 M 250 CON GONTRAILER + 1 7217
VEHICULOS PARA EL MONTAJE : 1 MERLO + 2 GRÚA LUNA

DISTANCIA:	13km
VIAJES:	15
CIMBUSTIBLE:	290L

CICLO 1	M250 + GONT	2 centrales merlo	CICLO 2	M250 + GONT	6 centrales 1 jaula placas 1 jaula rodillos	CICLO 3	M250 + GONT	6 centrales 2 extremos	CICLO 4	M250 + GONT	2 centrales 6 extremos
	M250 + GONT	6 extremos 2 jaulas cabestrante		M250 + GONT	6 extremos 2 jaulas rampa		M250 + GONT	6 centrales 2 extremos		M250 + GONT	2 extremos 6 centrales
	7217	caja herramienta 10 jaulas horquillas 11 cajas estribos		7217	8 jaulas horquillas 1 jaula rodillos		7217	1 jaula placas 2 cabestrantes sueltos		7217	1 jaula placas
CICLO 5	M250 + GONT	2 jaulas placas 6 extremos	CICLO 6	M250 + GONT	2 centrales 6 extremos	CICLO 7	M250 + GONT	2 centrales 6 extremos	CICLO 8	M250 + GONT	2 centrales 6 extremos
	M250 + GONT	8 extremos		M250 + GONT	2 centrales 6 extremos		M250 + GONT	2 extremos 6 centrales		M250 + GONT	2 extremos 2 centrales 1 jaula bordillos
	7217	1 jaula placas		7217	1 jaula cabestrante		7217	1 jaula placas		7217	-

*** Aprovechar los huecos que quedaran en el 7217 (ya que solo se transportará una jaula en cada ciclo) para herramienta varia.**

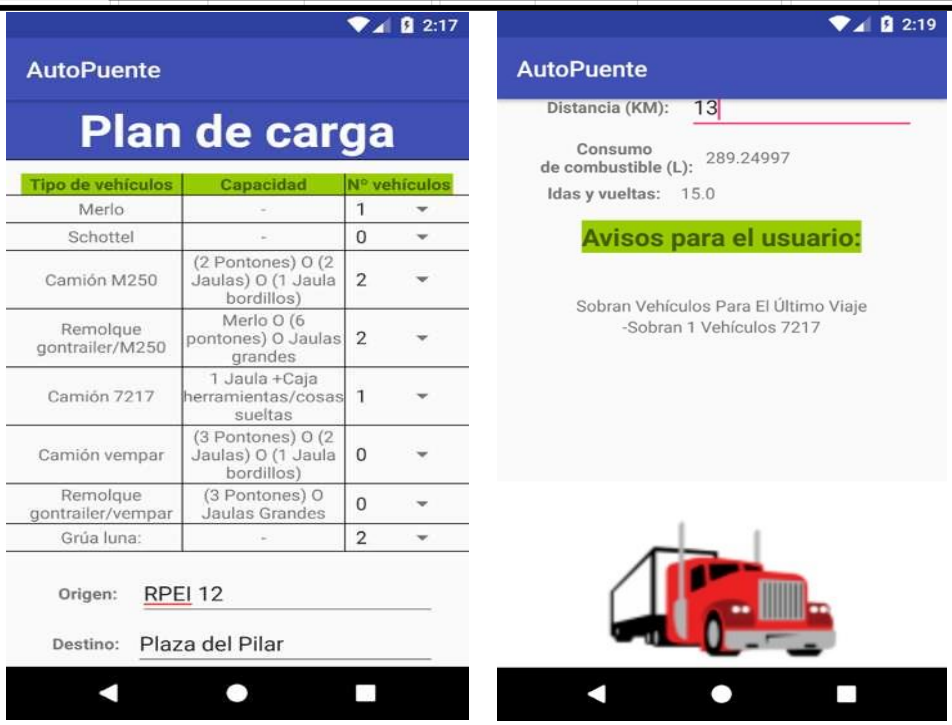


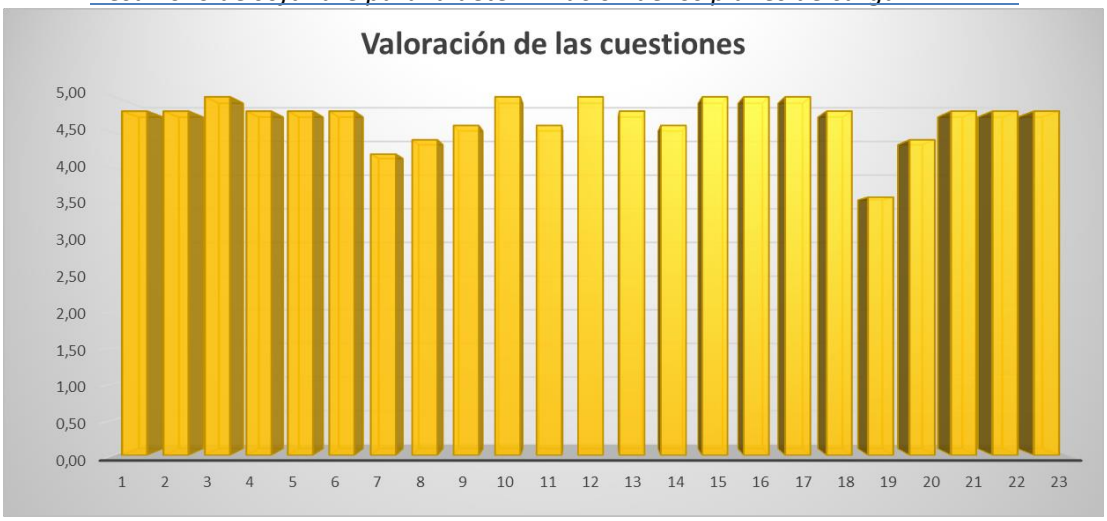
Figura 32: Tablas de los viajes y el consumo de combustible. Fuente: CIA puentes flotantes.

En la anterior figura, se puede observar que al introducir los mismos datos de los vehículos disponibles para el transporte del puente, se obtendría los mismos datos del número de viajes y el consumo de combustible. *AutoPuente* realiza dichos cálculos sin la necesidad de mostrar los ciclos de viajes mostrados en el ejemplo de 2013 ya que son invisibles para el usuario.
Con todo lo mencionado anteriormente se puede concluir que la aplicación ha hecho un cálculo óptimo ya que todos los datos obtenidos eran iguales a los del ejemplo de 2013.

3.4.2. Validación

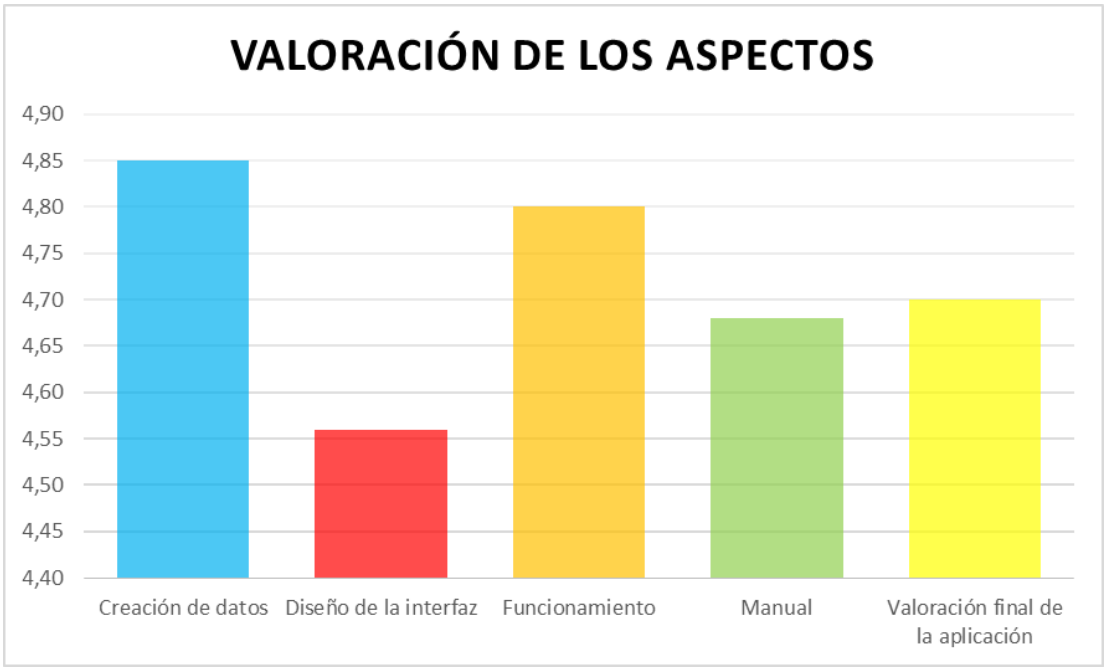
Se ha realizado un cuestionario (Anexo 5) para medir el nivel de satisfacción del personal de la CIA de puentes flotantes que van a usar la aplicación en el futuro. Dicho cuestionario fue realizado por 5 personas (el capitán Jefe de la CIA, el teniente Jefe de la SC. de puentes flotantes y tres sargentos de la SC.) de manera anónima.

El cuestionario consta de 19 preguntas enfocadas a la creación de los datos, el diseño de la interfaz, el funcionamiento de la aplicación y el manual de usuario. También otras 4 cuestiones sobre la valoración final de la aplicación. Dicho cuestionario tiene una nota de 1 a 5, siendo el 1 muy deficiente y el 5 excelente. Se han estudiado todas las cuestiones respondidas por los 5 de manera individual (Figura 33), también se han estudiado diferentes aspectos de todos los apartados (Figura 34).



Persona 1	Persona 2	Persona 3	Persona 4	Persona 5	Nota media
4,48	4,65	4,65	4,96	4,83	4,72

Figura 33: Gráfica de valoración de las cuestiones. Elaboración propia.



Creación de datos	Diseño de la interfaz	Funcionamiento	Manual	Valoración final de la aplicación	Nota Media
4,85	4,56	4,80	4,68	4,70	4,72

Figura 34: Gráfica de valoración de las cuestiones. Elaboración propia.

Con los resultados obtenidos anteriormente se pudo llegar a las siguientes conclusiones:

- La nota media de la aplicación según las 5 personas que realizaron el cuestionario era 4,72. Teniendo en cuenta que la valoración final de la aplicación tuvo una nota precisamente bastante parecida (4,70).
- La nota media de cada persona varía entre el 4,48 y el 4,96, teniendo en cuenta que la puntuación de cada uno de ellos era casi parecida.
- Había muchos apartados que tenían una puntuación de 5, por lo cual no hay un apartado en especial con la mejor valoración.
- El apartado peor valorado era el apartado 19 (Forma de descargar la aplicación/Manual). Según los comentarios, la forma de descargarla les parecía difícil ya que la aplicación no

aparece en Play Store del sistema *Android*, donde se descargan las aplicaciones normalmente. Esto se debe a que es necesario pagar 25€ al mes para mantener *AutoPuente* en Play Store.

- Los aspectos (creación de datos y funcionamiento) han sido los aspectos mejor valorados.
- Los aspectos (diseño de la interfaz y manual) han sido los aspectos peor valorados.

4. CONCLUSIONES

Con los resultados obtenidos en el apartado anterior, se puede concluir que *AutoPuente* cumple con todos los objetivos y requisitos mencionados anteriormente en la memoria. Ya que permite reducir los tiempos de hacer los cálculos y también tener un cálculo óptimo.
Después de la comprobación de los datos, se puede llegar a las siguientes conclusiones:

- La introducción de los datos del puente se ha llevado unos 20 segundos aproximadamente y el tiempo de calcular las herramientas ha sido inferior a un segundo. Por lo cual se ahorraría unas 2-3 horas haciendo los cálculos con la misma exactitud y sin ningún error.
- El tiempo de la introducción de los datos de los vehículos disponibles y la distancia del viaje ha sido un minuto aproximadamente y el tiempo de calcular los viajes, el consumo de combustible y los avisos del usuario ha sido inferior a un segundo. Ese ayudaría a ahorrar unas 4-6 horas calculando el plan de carga adecuado con el consumo exacto de combustible.
- Todos los resultados obtenidos han sido iguales a los que aparecen en el ejemplo de 2013 y con la misma exactitud.
- Todas las pruebas con los ejemplos anteriores que se disponen en la CIA resultaron satisfactorias. Y así, lo que llevaba horas para calcular y planificar, se hace en cuestión de segundos y sin error alguno, gracias a las fórmulas escritas en el lenguaje de JAVA y aplicados al programa *Android Studio*. Todo este tiempo ahorrado se podrá aprovechar en otras tareas como la preparación de la maniobra en sí.
- La realización del manual de usuario ha sido muy acertada, ya que en la CIA están teniendo cambios en los cuadros de mando, y así se podría solucionar muchas dudas al personal que se enfrenta a un trabajo de puentes flotantes sin una previa experiencia.
- La aplicación cumple con el alcance propuesto al principio de esta memoria y puede ser ampliada y mejorada en unos posibles trabajos en un futuro.

5. LÍNEAS FUTURAS

Como se mencionó anteriormente, la aplicación se desarrolló con la visión de ser ampliada y mejorada. *AutoPuente* es la primera versión, por lo cual se podrá tener nuevas versiones que estarán centradas en hacer unas mejoras que se pueden concluir en los siguientes trabajos:

- Tener una nueva funcionalidad que permite tener todos los ciclos de viajes de manera visible al usuario permitiéndole la opción de hacer cambios según el usuario considere.
- Mostrar un gráfico del puente con su longitud total, y así el usuario se hará idea de cómo se quedaría después de realizar el montaje.
- Añadir una nueva funcionalidad que permite el cálculo en medios auxiliares y personal para ahorrar el tiempo gastado en la preparación de esta tarea.
- Añadir un nuevo software que permite realizar los cálculos para el montaje de puentes fijos con sus dos tipos “Bailey y Mabey”.

6. BIBLIOGRAFÍA

MT6-401. Manual técnico puentes. Características técnicas. Estado Mayor del Ejército, diciembre 1994.

TDv 5420/003-13. Hohlplattenbrückengerät. Manual técnico de puentes Alemán. Alemania, mayo 1996.

M.A.N KRUPP. Descripción del equipo. Beschreibung der Hohlplattenbrücke. Regimiento de Pontoneros y Especialidad de Ingenieros número 12, diciembre 1996.

Tareas individuales del pontonero. Capitán. D. Joaquín Peralta Español, noviembre 2016.

Desarrolla de aplicaciones para Android. Universidad de Alicante. Alicante, 2014.

Manual práctico de programación en Visual Basic.NET 2016. Carlos Capellán. Centro de formación y desarrollo integral Padre Fantino. República Dominicana, 2015.

Introducción a C#. Manual del estudiante. Miguel Muñoz Serafín. Agosto 2017.

Programación en PHP a través de ejemplos. Manuel Palomo Durate. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Cádiz. Ildefonso Montero Pérez. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Cádiz.

Manual de Desarrollador Android Studio. La magia de las palabras. Juan Pablo Rodríguez Montoya. Gabriel Hernando Fuentes Amorocho.

Manual Básico Android Studio. Página WEB www.tutorialesenpdf.com.

Página oficial del Ejército de tierra. Página WEB www.ejercito.mde.

Anexo 1: Informe de preparación de montaje.

MINISTERIO
DE DEFENSA

USO OFICIAL

EJÉRCITO DE TIERRA

FUERZA TERRESTRE
MANDO DE INGENIEROS
RPEI 12

INFORME QUE FORMULA EL CAPITÁN DE INGENIEROS DON SAÚL GORDO PEREO, ESCALA DE OFICIALES CON T.I.M. 51225479119X, JEFE DE LA COMPAÑÍA DE PUENTES FLOTANTES DEL BATALLON DE PONTONEROS I/12, PERTENECIENTE AL REGIMIENTO DE PONTONEROS Y ESPECIALIDADES DE INGENIEROS Nº 12, RELATIVO AL RECONOCIMIENTO DEL RÍO DEVA EFECTUADO EN LA LOCALIDAD DEL MISMO NOMBRE.

REFERENCIAS:

A. OR5-406: INTELIGENCIA Y RECONOCIMIENTO DE INGENIEROS
B. MI4-XXX: LA SECCIÓN DE PONTONEROS
C. MANUAL TÉCNICO PUENTE MAN-KRUPP
D. MT6-4XX: PUENTE LOGÍSTICO COMPACT 200
E. M-4-5-5: PUENTE BAILEY ANCHO
F. MI6-407: PUENTE DORNIER
G. NOP,s COMPAÑÍA DE PUENTES

ANTECEDENTES:

El pasado día 2 de agosto, el Batallón de Pontoneros recibe la orden de realizar un estudio sobre la posibilidad de tender sobre el río Deva, a su paso por la localidad del mismo nombre, alguno de los Puentes reglamentarios en dotación en la Unidad. La finalidad es apoyar al Ayuntamiento de la localidad, debido al colapso de una de las pilas del Puente peatonal que une las localidades de Deva y Mutriku.

1. OBJETO DEL INFORME.

El objeto del presente informe es plasmar los datos obtenidos del reconocimiento in situ realizado el pasado día 7 de agosto de 2018. El reconocimiento fue realizado por el Jefe de la Compañía de Puentes Flotantes y un gruista de la misma, además del Jefe de la Sección de Montaje y un Suboficial pertenecientes a la Compañía de Puentes de Apoyos Fijos. Un total de cuatro (4) personas.

2. RECONOCIMIENTO IN SITU.

Tras realizar los contactos oportunos con el Alcalde de la localidad, D. Pedro Bengoetxea Loiola (solicitante del apoyo), el personal del Equipo de Reconocimiento se desplaza al lugar para realizar una reunión previa de coordinación en el Ayuntamiento. En dicha reunión participan:

1. Subdelegado de Gobierno en Guipúzcoa
2. Alcalde de la localidad de Deva
3. Miembros del BPON I/12

3. UBICACIÓN ACTUAL DEL PUENTE.

El puente de arco actual se encuentra en coordenadas 30T 552162 4793528 (WGS84). Junto al mismo se encuentra un embarcadero aguas abajo en el margen derecho (orilla este), además de otros embarcaderos de madera en el margen izquierdo, pero aguas arriba (orilla oeste), para las embarcaciones de pesca y recreo que operan en la zona.

4. ESTUDIO INICIAL DE PUNTOS DE TENDIDO.

Los datos iniciales, así como los requerimientos que se poseían para realizar el proyecto de tendido eran escasos. Las posibles opciones se valoraron en base a fotografía aérea, sin poder tener en cuenta factores determinantes como la longitud total de la brecha a salvar, el caudal y velocidad de la corriente, diferencia de altura entre orillas, gálibos de acceso a las zonas de trabajo, etc.

5. RESTRICCIONES DETERMINADAS POR EL TIPO DE MATERIAL.

5.1. PUENTE FLOTANTE MAN:

Este tipo de puente permite el tendido de una pasarela para personal, ajustándose al apoyo inicial solicitado. Permitiría realizar el tendido con una cantidad de material relativamente baja respecto al resto de los puentes reglamentarios.

Presenta sin embargo varias restricciones. La primera es que al ser un puente flotante, se ve afectado por los efectos de las mareas (proximidad del mar), además de los cambios súbitos de caudal del propio río Deva. Por estos motivos sería necesario tener un Equipo de Mantenimiento permanentemente en el lugar de tendido, para realizar los ajustes de los tornos y revisiones necesarias diariamente. Precisa una profundidad mínima de 0'75 metros de agua para que el material trabaje de manera adecuada.

La segunda restricción viene marcada por la existencia de embarcaderos próximos al puente actual, que hace inviable el tendido del puente aguas abajo del mismo, ya que las embarcaciones civiles no tendrían acceso al mar. Teniendo en cuenta que la reparación del actual puente de fábrica podría alargarse varios meses, causaría serias molestias a la población de la zona.

El tendido aguas arriba podría llegar a ser viable, aunque habría que considerar que los ríos arrastran consigo materiales como troncos y otros elementos que podrían afectar al normal funcionamiento del mismo, debido a crecidas súbitas del caudal del mismo en época de lluvias.

5.2. PUENTE FIJO TIPO MABEY/BAILEY:

Se tratan ambos puentes en un mismo supuesto ya que los requerimientos para el tendido son bastante similares.

Ambos puentes están concebidos y diseñados para el paso de vehículos de diferentes clases según la configuración empleada. Debemos tener en cuenta que aunque el requerimiento es para el paso de personal a pie, las configuraciones de ambos requieren una estructura que conlleva el empleo de gran cantidad de material. En el caso de estos puentes, cuando supera la luz de sesenta (60) metros, se debe realizar el tendido con pila intermedia. Las luces de los posibles puntos de tendido varían entre los setenta (70) y los ochenta (80) metros de luz.

Junto al margen derecho se encuentra el trazado de la línea de ferrocarril, que limita el acceso, carga y descarga de cualquier vehículo o material.

Junto al margen izquierdo encontramos la GI-638, que reduce tanto las posibilidades de emplearla como playa de lanzamiento, como de segunda orilla, pues se vería afectada gravemente la circulación en esta vía, poniendo también en riesgo al personal usuario de la estructura.

Ambos puentes admiten un desnivel máximo entre orillas de cuatro (4) metros aproximadamente.

Para asegurar el correcto mantenimiento y funcionamiento del material sería necesario que un Equipo se desplazara hasta la zona semanalmente, para verificar todos los elementos constructivos del mismo.

5.3. PUENTE FIJO TIPO DORNIER:

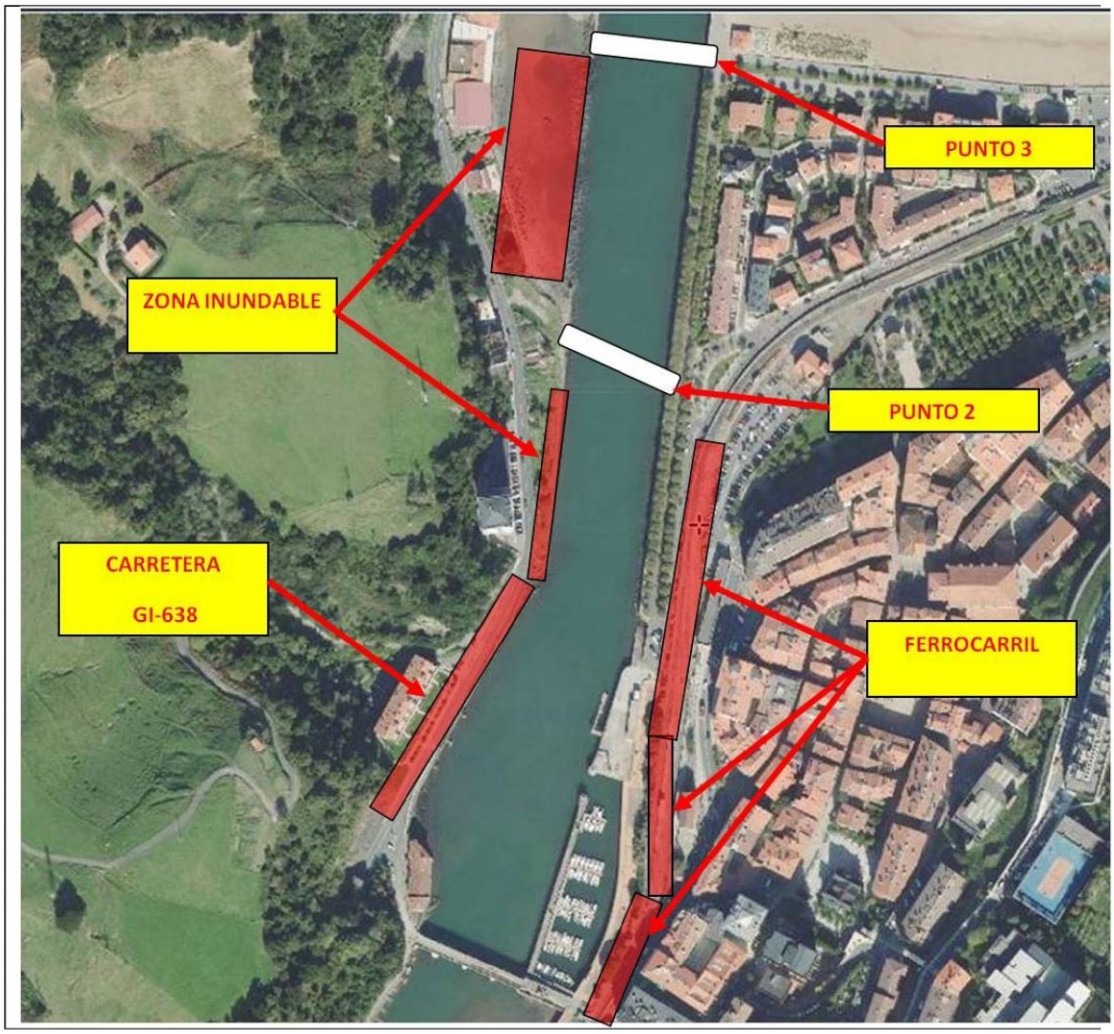
No se contempla el tendido de este tipo de puente debido a que su luz máxima es de cuarenta (40) metros.

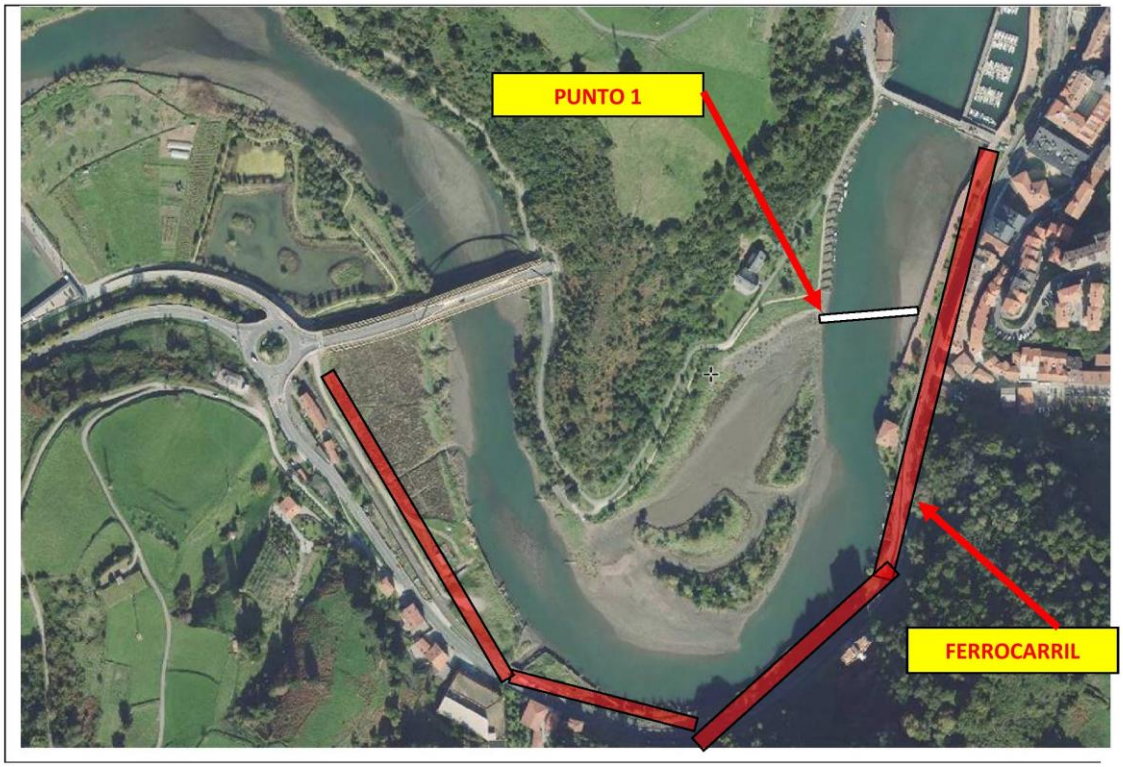
6. POSIBLES PUNTOS DE TENDIDO.

Una vez reconocido el terreno, y recabados los datos necesarios, se determinan tres (3) posibles puntos de tendido. El primero de ellos para una pasarela de personal de Puente Flotante MAN, y los otros dos (2) puntos para realizar el tendido de un puente MABEY.

En la fotografía siguiente se marcan las zonas de montaje de los puentes, y en rojo las zonas restringidas para el tendido, ya sea por la línea de ferrocarril, por la carretera que discurre por el margen izquierdo del río o por otras circunstancias.

Vista aguas abajo (zona norte) y aguas arriba (zona sur):





7. DATOS RECABADOS DEL RECONOCIMIENTO.

Una vez obtenidos los datos de cada uno de los puntos, se desechan los puntos dos (2) y tres (3) para el tendido, por los siguientes motivos:

- Punto 2: la brecha a salvar es de setenta (70) metros. En este caso, el puente requiere para el tendido una pila intermedia. El Alcalde rechaza la opción de realizar una losa de apoyo de la pila, o el aporte de material para reducir la longitud de la brecha, todo ello por motivos medioambientales.

- Punto 3: la brecha a salvar es de casi ochenta (80) metros. Al igual que el anterior, requiere una pila intermedia. Además, este punto se encuentra demasiado alejado del punto de paso original que utiliza la población, generando situaciones de riesgo, ya que deben circular varios cientos de metros por el arcén de una carretera.

8. PUNTO DE TENDIDO.

Tras no ser factibles los puntos anteriores, se centran los esfuerzos en realizar el estudio de los factores que afectan a la viabilidad del tendido de la pasarela flotante en el punto uno (1).

Los datos extraídos del reconocimiento son los siguientes:

- Brecha: 78 metros
- Velocidad de la corriente: <1 m/seg en el momento del reconocimiento
- Playa de lanzamiento: la localizada en el margen izquierdo (zona oeste) permitiría la botadura del material mediante maquinaria, en una zona localizada aguas arriba del punto de montaje. Se transportaría el material posteriormente por el cauce del río hasta su emplazamiento final.
- Empleo del medio de paso: exclusivamente para personal desde el momento del tendido hasta la reparación del puente de fábrica.
- Empleo como primera orilla: orilla oeste (margen izquierdo).
- Otros aspectos: el aspecto fundamental a tener en cuenta es la acción de las mareas, que pueden generar, comprobado en el momento del reconocimiento, una variación en el nivel de las aguas de hasta tres (3) metros. Estas variaciones provocarían que parte del material quedara varado sobre una zona sedimentaria que queda al descubierto en bajamar, en el margen derecho (orilla este).

9. PROPUESTA.

Con estos datos se llega a la conclusión de que la única opción viable es construir una pasarela compuesta por dos (2) embarcaderos unidos por una rampa que actuaría como elemento articulado, adecuándose al nivel del agua en cada momento. Hay que tener en cuenta que parte del material quedaría apoyado sobre el fondo del lecho en momentos de bajamar (dos veces al día). Por este motivo sería necesario mantener sobre el terreno un Equipo encargado de ajustar los tornos y demás elementos de sujeción de manera permanente, para evitar deterioros del material así como movimientos no deseados de la pasarela, que impidieran el acceso y salida del personal, por quedar demasiado separada en sus orillas.

10. ACCIONES EXTERNAS PREVIAS.

Para poder ejecutar el tendido de la pasarela, se requieren las siguientes acciones por parte del Ayuntamiento de Deva:

- Retirada / acondicionamiento del material sedimentario próximo a la orilla este.
- Adecuación / ampliación del punto de acceso a la pasarela en la orilla oeste.
- Instalar puntos de anclaje (argollas) en el muro de contención de la orilla este.
- Regulación del tráfico por parte de la policía municipal durante la descarga en primera orilla (oeste) del material reglamentario.
- Teniendo en cuenta que la reparación del puente de fábrica puede dilatarse en el tiempo, iluminación del punto de paso en momentos de baja visibilidad (tardes de otoño-invierno). Se ha acordado el cierre de la pasarela en arco nocturno con el Alcalde de la localidad.

11. MEDIOS Y MATERIALES NECESARIOS PARA EL MONTAJE.

PASARELA PEATONAL	
Pontones centrales	22
Tornos cabestrantes	8
Soporte de tornos cabestrantes	8
Rodillos guía	8
Cumbreras	2
Vigas de rampa	12
Final de rampa corto	2
Gatos de elevación	4
Postecillos	4
Bitas o Bolardo Anular	12

12. TRANSPORTE DE MATERIAL.

Para el traslado del material al lugar de tendido no se requieren medios externos al REPEI 12. Para ello serían necesarios los siguientes vehículos:

MEDIOS PARA TRANSPORTE		Observaciones
M-250	2	
IVECO 7217	1	
VEMPAR + Remolque	4	
CNLTT 1 Tn	2	
Ambulancia	1	SEGÚN PROPUESTA EMPLEO SANITARIA

13. PRESUPUESTO.

13.1. DIETAS:

EMPLEO	NÚMERO	DÍAS	CUANTIA	TOTAL	OBSERVACIONES
OFICIALES / SUBOFICIALES	3	5	103,37 €	1.551 €	VIAJE IDA / MONTAJE / DESMONTAJE / VIAJE VUELTA
MPTM	16	5	77,13 €	6.170 €	
OFICIALES / SUBOFICIALES	*1*	*7*	*103,37 €*	*724 €*	*TANTAS COMO SEMANAS DE TENDIDO (mantenimiento puente)*
MPTM	*7*	*7*	*77,13 €*	*3.779 €*	
SUMA TOTAL DIETAS				**7.721 € + (Nº SEMANAS x 4.503 €)**	

13.2. MANUTENCIÓN (en el caso de que se proporcione alojamiento sin coste):

EMPLEO	NÚMERO	DÍAS	CUANTIA	TOTAL	OBSERVACIONES
OFICIALES / SUBOFICIALES	3	5	37,40 €	561 €	VIAJE IDA / MONTAJE / DESMONTAJE / VIAJE VUELTA
MPTM	16	5	28,21 €	2.257 €	
OFICIALES / SUBOFICIALES	*1*	*7*	*37,40 €*	*262 €*	*TANTAS COMO SEMANAS DE TENDIDO (mantenimiento puente)*
MPTM	*7*	*7*	*28,21 €*	*1.382 €*	
SUMA TOTAL MANUTENCIÓN				**2.818 € + (Nº SEMANAS x 1.644 €)**	

13.3. CARBURANTE:

VEHÍCULOS		TRASLADO IDA Y REGRESO			
MODELO	**CANTIDAD**	**LITROS / KM**	**Kms.**	**PRECIO / LITRO**	**SUBTOTAL**
CLTT1T	2	0,15	700	1,3	273 €
CNTT 12T	3	0,65	700	1,3	1.774,50 €
Vempar + RQ	4	0,65	700	1,3	2.366 €
Ambulancia	1	0,60	700	1,3	546 €
				TOTAL CARBURANTE	**4.959,50 €**

En caso de que el lugar de alojamiento para el personal participante en la operación, fuera en localidad distinta a Deva, habría que considerar el consumo de carburante como consecuencia de los desplazamientos diarios con motivo del mantenimiento, durante el tiempo que se encontrara instalada la pasarela.

14. VIDA Y FUNCIONAMIENTO.

En el momento del reconocimiento no se han establecido las condiciones de vida del personal participante en la operación, especialmente del Equipo a cargo del mantenimiento de la pasarela.

15. OTRAS CONSIDERACIONES.

El Ayuntamiento de Deva será el encargado de realizar las gestiones necesarias a efectos de permisos, licencias y seguros de responsabilidad civil con todas las partes implicadas en la operación. Se proporcionará al Ayuntamiento los datos necesarios del material, para la transmisión de la información a la Cuenca Hidrográfica correspondiente.

Monzalbarba, a 8 de agosto de 2018

EL CAPITÁN

FIRMADO EL ORIGINAL

Anexo 2: Lenguaje de programación de AutoPuente.

Programación de aplicación informática para planes de carga de PTF-MAN

Códigos de la aplicación AutoPuente

Descripción breve

La aplicación AutoPuente esta basada en dos lenguajes que son Java y XML. Dichos lenguajes están aplicados en el sistema Android Studio. El presente documento contiene todos los codigos usados en la aplicación.

Ahmad Alqadi
ahmadalqadi70@gmail.com

Contenido

1

1. Lenguaje JAVA:

1.1. MainActivity.

```
package com.ahmadalqadi.app.letipuente;

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

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button continue_button =
(Button)findViewById(R.id.continue_button);

        continue_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this,
page2.class);
                startActivity(i);
            }
        });
    }
}
```

1.2. Page 2.

```
package com.ahmadalqadi.app.letipuente;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
public class page2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_page2);

        final Spinner types_Spinner = (Spinner)
findViewById(R.id.types_spinner);
        final Spinner classes_Spinner = (Spinner)
findViewById(R.id.classes_spinner);
        final EditText distance_EditText =
```

```
(EditText)findViewById(R.id.distance_editText);
        final TextView warning_TextView =
(TextView)findViewById(R.id.warning_text_view);
        final Button continue_button =
(Button)findViewById(R.id.continue_button_page2);
        final TextView type_Validation = (TextView)
findViewById(R.id.type_validation);
        final TextView class_Validation =
(TextView)findViewById(R.id.class_validation);
        final TextView distance_Validation =
(TextView)findViewById(R.id.distance_validation);
        Bundle data_Bundle = getIntent().getExtras();




        ArrayAdapter<CharSequence> types_Adapter =
ArrayAdapter.createFromResource(this, R.array.types_array,
android.R.layout.simple_spinner_item);

types_Adapter.setDropDownViewResource(android.R.layout.simple_sp
inner_dropdown_item);
        types_Spinner.setAdapter(types_Adapter);
        types_Spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent,
View view, int position, long id) {
                switch(types_Spinner.getSelectedItemPosition()){
                    case 0:{
                        classes_Spinner.setEnabled(false);
                        break;
                    }
                    case 1 :{
                        ArrayAdapter<CharSequence>
classes_Adapter =
ArrayAdapter.createFromResource(page2.this,R.array.Pasarela_arra
y, android.R.layout.simple_spinner_item);

classes_Adapter.setDropDownViewResource(android.R.layout.simple_
spinner_dropdown_item);

classes_Spinner.setAdapter(classes_Adapter);
                        classes_Spinner.setEnabled(true);
                        break;
                    }
                    case 2:{
                        ArrayAdapter<CharSequence>
classes_Adapter =
ArrayAdapter.createFromResource(page2.this,R.array.Puente_array,
android.R.layout.simple_spinner_item);

classes_Adapter.setDropDownViewResource(android.R.layout.simple_
spinner_dropdown_item);

classes_Spinner.setAdapter(classes_Adapter);
                        classes_Spinner.setEnabled(true);
                        break;
                    }
                    case 3: {
```

```
                        ArrayAdapter<CharSequence>
classes_Adapter = ArrayAdapter.createFromResource(page2.this,
R.array.Compuerta_array, android.R.layout.simple_spinner_item);

classes_Adapter.setDropDownViewResource(android.R.layout.simple_
spinner_dropdown_item);

classes_Spinner.setAdapter(classes_Adapter);
                        classes_Spinner.setEnabled(true);
                        break;
                    }
                }
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent)
{

            }
        });


        distance_EditText.addTextChangedListener(new
TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int
start, int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start,
int before, int count) {

                try{
                    int distance =
Integer.parseInt(distance_EditText.getText().toString());
                    if (distance>100){

warning_TextView.setVisibility(View.VISIBLE);

distance_Validation.setVisibility(View.INVISIBLE);
                    }

                    else{

warning_TextView.setVisibility(View.INVISIBLE);

distance_Validation.setVisibility(View.INVISIBLE);
                    }

                }
                catch(Exception e){
                }

            }

            @Override
            public void afterTextChanged(Editable s) {

            }
```

4

```
        });

        continue_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int type_Index;
                int class_Index;
                float distance;
                Intent i = new Intent(page2.this, page3.class);
                if(classes_Spinner.getSelectedItemPosition()== 0
|| !classes_Spinner.isEnabled())

class_Validation.setVisibility(View.VISIBLE);
                else

class_Validation.setVisibility(View.INVISIBLE);
                if(types_Spinner.getSelectedItemPosition()== 0)
                    type_Validation.setVisibility(View.VISIBLE);
                else

type_Validation.setVisibility(View.INVISIBLE);
                try {
                    if
(distance_EditText.getText().toString().trim().length() < 1 ||
Integer.parseInt(distance_EditText.getText().toString()) < 1)

distance_Validation.setVisibility(View.VISIBLE);
                    else

distance_Validation.setVisibility(View.INVISIBLE);
                }
                catch(Exception e){
                }

if(type_Validation.getVisibility()==View.INVISIBLE &&
class_Validation.getVisibility()==View.INVISIBLE &&
distance_Validation.getVisibility()== View.INVISIBLE){
                    {
                        type_Index =
types_Spinner.getSelectedItemPosition();
                        class_Index =
classes_Spinner.getSelectedItemPosition();
                        distance =
Float.parseFloat(distance_EditText.getText().toString());
                        Bundle data_Bundle = new Bundle();
                        data_Bundle.putInt("type_Index",
type_Index);
                        data_Bundle.putInt("class_Index",
class_Index);
                        data_Bundle.putFloat("distance",
distance);
                        i.putExtras(data_Bundle);
                        startActivity(i);
                    }


                }

        }
```

```
        });

        }
```

1.3. Page 3.

```
package com.ahmadalqadi.app.letipuente;

import android.content.Intent;
import android.renderscript.Sampler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class page3 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_page3);

        Bundle data_Bundle = getIntent().getExtras();
        final int type_Index = data_Bundle.getInt("type_Index");
        final int class_Index =
data_Bundle.getInt("class_Index");
        final float distance = data_Bundle.getFloat("distance");
        final String selectedType =
getTypeFromIndex(type_Index);
        final String selectedClass =
getClassFromIndex(class_Index, selectedType);

        TextView pExtremoTotalTextView = (TextView)
findViewById(R.id.P_Extremo_total);



        TextView pCentralTotalTextView = (TextView)
findViewById(R.id.P_Central_total);


        TextView estribosTotalTextView = (TextView)
findViewById(R.id.Estribos_total);
        TextView estribosJaulasTextView = (TextView)
findViewById(R.id.Estribos_Jaulas);

        TextView horquillasTotalTextView = (TextView)
findViewById(R.id.Horquillas_total);
        TextView horquillasJaulasTextView = (TextView)
findViewById(R.id.Horquillas_Jaulas);
        TextView horquillasSueltosTextView = (TextView)
findViewById(R.id.Horquillas_Sueltos);

        TextView cabestrantesTotalTextView = (TextView)
findViewById(R.id.Cabestrantes_total);
        TextView cabestrantesJaulasTextView = (TextView)
findViewById(R.id.Cabestrantes_Jaulas);
        TextView cabestrantesSueltosTextView = (TextView)
findViewById(R.id.Cabestrantes_Sueltos);
```

```
        TextView placasDeflectorasTextView = (TextView)
findViewById(R.id.Placas_Deflectoras_total);
        TextView placasDeflectorasJaulasTextView = (TextView)
findViewById(R.id.Placas_Deflectoras_Jaulas);
        TextView placasDeflectorasSueltosTextView = (TextView)
findViewById(R.id.Placas_Deflectoras_Sueltos);

        TextView traveserosTotalTextView = (TextView)
findViewById(R.id.Traveseros_total);


        TextView soportesCabestranteTotalTextView = (TextView)
findViewById(R.id.Sportes_Cabestrante_total);

        TextView rodillosTotalTextView = (TextView)
findViewById(R.id.Rodillos_total);
        TextView rodillosJaulasTextView = (TextView)
findViewById(R.id.Rodillos_Jaulas);
        TextView rodillosSueltosTextView = (TextView)
findViewById(R.id.Rodillos_Sueltos);

        TextView bolardosTotalTextView = (TextView)
findViewById(R.id.Bolardos_total);

        TextView buquesDeAlbaTotalTextView = (TextView)
findViewById(R.id.Buques_de_Alba_total);


        TextView bordillosTextView = (TextView)
findViewById(R.id.Bordillos_total);
        TextView bordillosJaulasTextView = (TextView)
findViewById(R.id.Bordillos_Jaulas);


        TextView anchuraTextView = (TextView)
findViewById(R.id.anchura_amount);

        TextView cumbrerasTotalTextView =
(TextView)findViewById(R.id.cumbreras_Total);
        TextView vigasDeRampaTotalTextView = (TextView)
findViewById(R.id.vigas_de_rampa_Total);
        TextView finalesDeRampaTotalTextView = (TextView)
findViewById(R.id.finales_de_rampa_Total);
        TextView postecillosTotalTextView = (TextView)
findViewById(R.id.postecillos_Total);
        TextView gatosTotalTextView = (TextView)
findViewById(R.id.gatos_Total);
        TextView cableTotalTextView = (TextView)
findViewById(R.id.cable_Total);
        TextView rampaJaulasTextView = (TextView)
findViewById(R.id.rampa_Jaulas);

        final float pExtremoTotal =
getPExtremoTotal(selectedType, selectedClass, distance);
        final float pCentralTotal =
getPCentralTotal(selectedType, selectedClass, distance,
pExtremoTotal);

        final float estribosTotal =
getEstribosTotal(pExtremoTotal, pCentralTotal);
        final float estribosJaulas =
```

```
getEstribosJaulas(estribosTotal);

        final float horquillasTotal =
getHorquillasTotal(pExtremoTotal, pCentralTotal);
        final float horquillasJaulas =
getHorquillasJaulas(horquillasTotal);

        final float cabestrantesTotal =
getCabestrantesTotal(distance);
        final float cabestrantesJaulas =
getCabestrantesJaulas(cabestrantesTotal);

        final float placasDeflectorasTotal =
getPlacasDeflectorasTotal(selectedType, selectedClass, distance,
pExtremoTotal, pCentralTotal);
        final float placasDeflectorasJaulas =
getPlacasDeflectorasJaulas(placasDeflectorasTotal);

        final float traveserosTotal =
getTraveserosTotal(placasDeflectorasTotal);
        final float soportesCabestranteTotal =
getSoportesCabestranteTotal(cabestrantesTotal);

        final float rodillosTotal =
getRodillosTotal(cabestrantesTotal);
        final float rodillosJaulas =
getRodillosJaulas(rodillosTotal);

        final String bolardosTotal =
getBolardosTotal(selectedType, selectedClass, distance);
        final float buquesDeAlbaTotal =
getBuquesDeAlbaTotal(selectedType, selectedClass, distance);


        String anchura = getAnchura(selectedType,
selectedClass);

        int cumbrerasTotal = getCumbrerasTotal(selectedType,
selectedClass, distance);
        float vigasDeRampaTotal =
getVigasDeRampaTotal(selectedType, selectedClass, distance);
        final float rampaJaulas =
getRampaJaulas(selectedType,selectedClass,distance);

        Button continueButton =
(Button)findViewById(R.id.continue_button_page3);
        continueButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(page3.this, page4.class);
                Bundle data_Bundle = new Bundle();
                data_Bundle.putString("selected_Type",
selectedType);
                data_Bundle.putString("selected_Class",
selectedClass);
                data_Bundle.putFloat("distance", distance);
                data_Bundle.putFloat("cabestrantesJaulas",
cabestrantesJaulas);
                data_Bundle.putFloat("placasDeflectorasJaulas",
placasDeflectorasJaulas);
```

8

```
data_Bundle.putFloat("rodillosJaulas",rodillosJaulas);
                data_Bundle.putFloat("bordillosJaulas", 1);
                data_Bundle.putFloat("rampasJaulas",
rampaJaulas);
                data_Bundle.putFloat("horquillasJaulas",
horquillasJaulas);
                data_Bundle.putFloat("estribosJaulas",
estribosJaulas);
                data_Bundle.putFloat("pExtremoTotal",
pExtremoTotal);
                data_Bundle.putFloat("pCentralTotal",
pCentralTotal);
                data_Bundle.putString("bolardos",
bolardosTotal);
                i.putExtras(data_Bundle);
                startActivity(i);
            }
        });


        pExtremoTotalTextView.setText("" +
(int)(pExtremoTotal));



        pCentralTotalTextView.setText("" + (int) pCentralTotal);


        estribosTotalTextView.setText("" + (int) estribosTotal);
        estribosJaulasTextView.setText( "" + (int)
estribosJaulas);

        horquillasTotalTextView.setText("" + (int)
horquillasTotal);
        horquillasJaulasTextView.setText("" + (int)
horquillasJaulas);
        horquillasSueltosTextView.setText("" + (int)
getHorquillasSueltos(horquillasTotal));

        cabestrantesTotalTextView.setText("" + (int)
cabestrantesTotal);
        cabestrantesJaulasTextView.setText("" + (int)
cabestrantesJaulas);
        cabestrantesSueltosTextView.setText("" + (int)
getCabestrantesSueltos(cabestrantesTotal));

        placasDeflectorasTextView.setText("" + (int)
placasDeflectorasTotal);
        placasDeflectorasJaulasTextView.setText("" + (int)
placasDeflectorasJaulas);
        placasDeflectorasSueltosTextView.setText("" + (int)
getPlacasDeflectorasSueltos(placasDeflectorasTotal));

        traveserosTotalTextView.setText("" + (int)
traveserosTotal);

        soportesCabestranteTotalTextView.setText("" + (int)
soportesCabestranteTotal);
```

9

```
        rodillosTotalTextView.setText("" + (int) rodillosTotal);
        rodillosJaulasTextView.setText("" + (int)
rodillosJaulas);
        rodillosSueltosTextView.setText("" + (int)
getRodillosSueltos(rodillosTotal));

        bolardosTotalTextView.setText("" +  bolardosTotal);

        buquesDeAlbaTotalTextView.setText("" + (int)
buquesDeAlbaTotal);

        bordillosTextView.setText("-");
        bordillosJaulasTextView.setText("1");


        anchuraTextView.setText(anchura);


        cumbrerasTotalTextView.setText("" + cumbrerasTotal);
        vigasDeRampaTotalTextView.setText("" +
(int)vigasDeRampaTotal);
        finalesDeRampaTotalTextView.setText("-");
        postecillosTotalTextView.setText("" + cumbrerasTotal);
        gatosTotalTextView.setText("" + cumbrerasTotal);
        cableTotalTextView.setText("" + cumbrerasTotal);
        rampaJaulasTextView.setText("" + (int)rampaJaulas);
    }

    public float getPExtremoTotal(String selected_Type, String
selected_Class, float distance){
        float value;
        if(selected_Type == "Pasarela" && selected_Class ==
"Tipo 1")
            return 0;
        else if(selected_Type == "Pasarela" && selected_Class ==
"Tipo 2" && ((distance/6.3)-Math.floor(distance/6.3)>0)){
            value = (float) Math.floor(distance/6.3+1);
            return value;
        }
        else if(selected_Type == "Pasarela" && selected_Class ==
"Tipo 2" && ((distance/6.3) - Math.floor(distance/6.3)) == 0)
        {
            value = (float)Math.floor(distance/6.3);
            return value;
        }
        else if(selected_Type == "Pasarela" && selected_Class ==
"Tipo 3" && ((distance/4.2)-Math.floor(distance/4.2)>0))
        {
            value = (float) Math.floor(distance/4.2+1);
            return value;
        }
        else if(selected_Type == "Pasarela" && selected_Class ==
"Tipo 3" && ((distance/4.2)-Math.floor(distance/4.2) == 0))
        {
            value = (float) Math.floor(distance/4.2);
            return value ;
        }
        else if(selected_Type == "Pasarela" && selected_Class ==
"Tipo 4" && ((distance/6.3)-Math.floor(distance/6.3)>0))
        {
            value = (float) Math.floor(distance/6.3+1);
```

```
            return value ;
        }
        else if(selected_Type == "Pasarela" && selected_Class ==
"Tipo 4" && ((distance/6.3)-Math.floor(distance/6.3) == 0))
        {
            value = (float) Math.floor(distance/6.3);
            return value;
        }
        else if(selected_Type == "Compuerta" && selected_Class
== "MLC20" && distance>0)
        {
            value = (float) Math.ceil(2+(distance-4.2)/2.1);
            return value;
        }

        else if(selected_Type == "Compuerta" && selected_Class
== "MLC50T1" && distance>0)
        {
            value = (float) Math.ceil((6+((distance-
8.4)/2.1)/5+((distance-8.4)/2.1)));
            return value;
        }
        else if(selected_Type == "Compuerta" && selected_Class
== "MLC50T2" && distance>0)
        {
            value = (float) Math.ceil((distance/2.1)*2);
            return value;
        }
        else if(selected_Type == "Compuerta" && selected_Class
== "MLC80" && distance>0)
        {
            value = (float) Math.ceil((8+((distance-
8.4)/2.1)*2));
            return value;
        }
        else if(selected_Type == "Puente" && selected_Class ==
"MLC30" && distance>0)
        {
            value = (float) Math.ceil(distance/2.1);
            return value;
        }
        else if(selected_Type == "Puente" && selected_Class ==
"MLC50" && distance>0)
        {
            value =
(float)Math.ceil((distance/2.1)+(distance/2.1)/5);
            return value;
        }
        else if(selected_Type == "Puente" && selected_Class ==
"MLC80A" && distance>0)
        {
            value = (float) Math.ceil((distance/2.1)*2);
            return value;
        }
        else if(selected_Type == "Puente" && selected_Class ==
"MLC80B" && distance>0)
        {
            value = (float) Math.ceil(distance/2.1);
            return value;
        }
        else return 0;
```

11

```
    }

    public String getTypeFromIndex(int type_Index){
        String selected_Type = "";
        switch(type_Index){
            case 1 : selected_Type = "Pasarela";
                break;
            case 2 : selected_Type = "Puente";
                break;
            case 3 : selected_Type = "Compuerta";
                break;
        }
        return selected_Type;
    }

    public String getClassFromIndex(int class_Index, String
type){
        String selected_Class = "";
        switch(class_Index){
            case 1: {
                if (type=="Pasarela")
                    selected_Class="Tipo 1";
                else if(type=="Puente")
                    selected_Class="MLC30";
                else if(type=="Compuerta")
                    selected_Class="MLC20";
            }
            break;
            case 2: {
                if (type=="Pasarela")
                    selected_Class="Tipo 2";
                else if(type=="Puente")
                    selected_Class="MLC50";
                else if(type=="Compuerta")
                    selected_Class="MLC50T1";
            }
            break;
            case 3: {
                if (type=="Pasarela")
                    selected_Class="Tipo 3";
                else if(type=="Puente")
                    selected_Class="MLC80A";
                else if(type=="Compuerta")
                    selected_Class="MLC50T2";
            }
            break;
            case 4: {
                if (type=="Pasarela")
                    selected_Class="Tipo 4";
                else if(type=="Puente")
                    selected_Class="MLC80B";
                else if(type=="Compuerta")
                    selected_Class="MLC80";
            }
        }
        return selected_Class;
    }

    public float getPCentralTotal(String selected_Type, String
selected_Class, float distance, float pExtremoTotal){
        float value;
```

```
        if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 1") && distance > 0 &&
((distance/4.2) - Math.floor(distance/4.2)) > 0){
            value = (float) Math.floor(distance/4.2)+1;
            return value;
        }
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 1") && distance > 0 &&
((distance/4.2) - Math.floor(distance/4.2)) == 0){
            value = (float)Math.floor(distance/4.2);
            return value;
        }
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 2")){
            return pExtremoTotal;
        }
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 3") && distance > 0){
            return pExtremoTotal - 1;
        }
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 4") && ((pExtremoTotal/2) -
Math.floor(pExtremoTotal/2)) > 0 && distance > 0){
            value = (float) Math.floor(pExtremoTotal/2);
            return value;
        }
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 4") && ((pExtremoTotal/2) -
Math.floor(pExtremoTotal/2)) == 0 && distance > 0){
            value = (float) Math.floor(pExtremoTotal/2-1);
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC20") && distance > 0){
            value = (float) Math.ceil(2+((distance - 4.2) / 2.1
- 1) / 2);
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1") && distance > 0){
            value = (float) Math.ceil(4+((distance - 8.4) / 2.1
- ((distance-8.4) / 2.1) / 5));
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2") && distance > 0){
            value = (float) Math.ceil((distance / 2.1) +
(distance / 2.1) / 4);
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80") && distance > 0){
            value = (float) Math.ceil(6 + (distance - 8.4) /
2.1);
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC30") && distance > 0){
            value = (float) Math.ceil(pExtremoTotal / 2);
            return value;
        }
```

```
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC50") && distance > 0){
            value = (float) Math.ceil(distance / 2.1 - (distance
/ 2.1) / 5);
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A") && distance > 0){
            value = (float) Math.ceil(distance / 2.1);
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B") && distance > 0){
            value = (float) Math.ceil((distance / 2.1) * 2);
            return value;
        }
        else return 0;
    }

    float getEstribosTotal(float pExtremoTotal, float
pCentralTotal){
        return 10 * (pExtremoTotal + pCentralTotal);
    }

    float getHorquillasTotal(float pExtremoTotal, float
pCentralTotal){
        return 11 * (pExtremoTotal + pCentralTotal);
    }

    float getTraveserosTotal(float placas_Deflectoras){
        return placas_Deflectoras;
    }

    float getSoportesCabestranteTotal(float cabestrantes){
        return cabestrantes;
    }

    float getRodillosTotal(float cabestrantes){
        return cabestrantes;
    }

    String getBolardosTotal(String selected_Type, String
selected_Class, float distance){
        String bolardos;

        if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC30")){
            bolardos = "" + ((int) Math.ceil(distance / 21 -
1));
            return bolardos;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC50")){
            bolardos = "" + (int) Math.ceil(distance / 10.5 -
1);
            return bolardos;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A")){
            bolardos = "" + (int) Math.ceil(distance / 12.6 -
1);
```

```
            return bolardos;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B")){
            bolardos = "" + (int) Math.ceil(distance / 12.6 -
1);
            return bolardos;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC20")){
            bolardos = "" + (int) Math.ceil((distance - 4.2) /
14.7 - 1);
            return bolardos;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1")){
            bolardos = "" + (int) Math.ceil((distance - 8.4) /
10.5 - 1);
            return bolardos;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2")){
            bolardos = "" + (int) Math.ceil(distance / 12.6 -
1);
            return bolardos;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80")){
            bolardos = "" + (int) Math.ceil((distance - 8.4) /
16.8 - 1);
            return bolardos;
        }
        else return "-";
    }

    public float getPlacasDeflectorasTotal(String selected_Type,
String selected_Class, float distance, float pExtremoTotal,
float pCentralTotal){
        float value;
        if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC20")){
            return pExtremoTotal;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1")){
            value = (float)  Math.ceil(distance/2.1);
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2")){
            value = (float) Math.ceil(distance/2.1);
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80")){
            value = (float) Math.ceil(distance/2.1);
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC30")){
            value = (float) Math.ceil(distance/2.1);
```

```
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC50")){
            value = (float) Math.ceil(distance/2.1);
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A")){
            value = (float) Math.ceil(distance/2.1);
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B")){
            value = (float) Math.ceil(distance/2.1);
            return value;
        }
        else return pExtremoTotal+pCentralTotal;
    }

    public float getCabestrantesTotal(float distance){
        float value;
        value=(float) Math.ceil((distance/10.5)*2.3);
                return value;
    }

    public float getBuquesDeAlbaTotal(String selected_Type,
String selected_Class, float distance){
        if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 1") && distance > 0)
            return 2;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 2") && distance > 0)
            return 2;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 3") && distance > 0)
            return 2;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 4") && distance > 0)
            return 2;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC20") && distance > 0)
            return 4;
        else if(selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1") && distance > 0)
            return 4;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2") && distance > 0)
            return 4;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80") && distance > 0)
            return 4;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC30") && distance > 0)
            return 4;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC50") && distance > 0)
            return 4;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A") && distance > 0)
            return 4;
```

```
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B") && distance > 0)
            return 4;
        else return 0;
    }

    public float getEstribosJaulas(float estribos_Total){
        float value;
        if ((estribos_Total / 100) - Math.floor(estribos_Total /
100) == 0) {
            value = (float) Math.floor(estribos_Total / 100);
            return value;
        }
        else {
            value = (float) Math.floor(estribos_Total / 100) +
1;
            return value;
        }
    }

    public float getHorquillasJaulas(float horquillas_Total){
        float value;
        if ((horquillas_Total/68) -
Math.floor(horquillas_Total/68) == 0){
            value = (float) Math.ceil(horquillas_Total/68) + 1;
            return value;
        }
        else if (((horquillas_Total/68) -
Math.floor(horquillas_Total/68)) > 0 && ((horquillas_Total/68) -
Math.floor(horquillas_Total/68)) < 0.15){
            value = (float) Math.floor(horquillas_Total/68);
            return value;
        }
        else{
            value = (float) Math.floor((horquillas_Total/68)+1);
            return value;
        }
    }

    public float getHorquillasSueltos(float horquillas_Total){
        float value;
        if (((horquillas_Total / 68) -
Math.floor(horquillas_Total / 68)) > 0 && ((horquillas_Total /
68)-Math.floor(horquillas_Total / 68)) < 0.15){
            value = (float) ((horquillas_Total / 68) -
Math.floor(horquillas_Total / 68)) * 68;
            if (value < 3.0f)
                value =3.0f;
            return value;
        }
        else
            return 0;
    }

    public float getCabestrantesJaulas(float
cabestrantes_Total){
        float value;
        if ((cabestrantes_Total / 8) -
Math.ceil(cabestrantes_Total / 8) == 0){
            value = (cabestrantes_Total / 8);
            return value;
```

```
        }
        else if (Math.ceil(cabestrantes_Total / 8) -
(cabestrantes_Total / 8) > 0 && (cabestrantes_Total / 8) -
Math.ceil(cabestrantes_Total / 8) < 0.6){
            value = (cabestrantes_Total / 8);
            return value;
        }
        else {
            value = (float) Math.ceil(cabestrantes_Total / 8);
            return value;
        }
    }

    public float getCabestrantesSueltos(float
cabestrantes_Total){
        float value;
        if (((cabestrantes_Total / 8) -
(Math.floor(cabestrantes_Total / 8))) > 0 &&
((cabestrantes_Total / 8) - Math.floor(cabestrantes_Total / 8))
< 0.6){
            value = (float) ((cabestrantes_Total / 8) -
Math.floor(cabestrantes_Total / 8)) * 8;
            return value;
        }
        else
            return 0;
    }

    public float getPlacasDeflectorasJaulas(float
placas_Deflectoras_Total){
        float value;
        if ((placas_Deflectoras_Total / 7) -
Math.floor(placas_Deflectoras_Total / 7) == 0){
            value = placas_Deflectoras_Total / 7;
            return value;
        }
        else if (((placas_Deflectoras_Total / 7) -
Math.floor(placas_Deflectoras_Total / 7)) > 0 &&
((placas_Deflectoras_Total/7) -
Math.floor(placas_Deflectoras_Total / 7)) < 0.5){
            value = (float) Math.floor(placas_Deflectoras_Total
/ 7);
            return value;
        }
        else{
            value = (float) Math.floor((placas_Deflectoras_Total
/ 7) + 1);
            return value;
        }
    }

    public float getPlacasDeflectorasSueltos(float
placas_Deflectoras_Total){
        float value;
        if (((placas_Deflectoras_Total / 7) -
Math.floor(placas_Deflectoras_Total / 7)) > 0 &&
((placas_Deflectoras_Total / 7) -
Math.floor(placas_Deflectoras_Total / 7)) < 0.5){
            value = (float) ((placas_Deflectoras_Total / 7) -
Math.floor(placas_Deflectoras_Total / 7)) * 7;
            return value;
```

```
        }
        else
            return 0;
    }

    public float getRodillosJaulas(float rodillos_Total){
        float value;
        if (((rodillos_Total / 13) - Math.floor(rodillos_Total /
13)) == 0){
            value = rodillos_Total / 13;
            return value;
        }
        else if (((rodillos_Total / 13) -
Math.floor(rodillos_Total / 13)) > 0.5){
            value = (float) Math.floor((rodillos_Total / 13)+1);
            return value;
        }
        else {
            value = (float) Math.floor(rodillos_Total / 13);
            return value;
        }
    }

    public float getRodillosSueltos(float rodillos_Total){
        float value;
        if (((rodillos_Total / 13) - Math.floor(rodillos_Total /
13)) > 0 && ((rodillos_Total / 13) - Math.floor(rodillos_Total /
13)) <= 0.5){
            value = (float) ((rodillos_Total/13)-
Math.floor(rodillos_Total/13))*13;
            return value;
        }
        else
            return 0;
    }


    public String getAnchura(String selected_Type, String
selected_Class){
        String value;
        if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC20")){
            value = "" + 3.5;
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1")){
            value = "" + 6.3;
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2")){
            value = "" + 8.1;
            return value;
        }
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80")){
            value = "" + 8.1;
            return value;
        }
        else if (selected_Type.equals("Puente") &&
```

```
selected_Class.equals("MLC30")){
            value = "" + 3.5;
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC50")){
            value = "" + 6.3;
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A")){
            value = "" + 8.1;
            return value;
        }
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B")){
            value = "" + 8.1;
            return value;
        }
        else
            return "-";
    }

    public int getCumbrerasTotal(String selected_Type, String
selected_Class, float distance){
        if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 1") && distance > 0.0)
            return 2;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 2") && distance > 0.0)
            return 2;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 3") && distance > 0.0)
            return 2;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 4") && distance > 0.0)
            return 2;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC20") && distance > 0.0)
            return 4;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1") && distance > 0.0)
            return 4;
        else if(selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2") && distance > 0.0)
            return 4;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80") && distance > 0.0)
            return 4;
        else if(selected_Type.equals("Puente") &&
selected_Class.equals("MLC30") && distance > 0.0)
            return 4;
        else if(selected_Type.equals("Puente") &&
selected_Class.equals("MLC50") && distance > 0.0)
            return 4;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A") && distance > 0.0)
            return 4;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B") && distance > 0.0)
            return 4;
```

```
        else
            return 0;
    }

    public float getVigasDeRampaTotal(String selected_Type,
String selected_Class, float distance){
        if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 1") && distance>0.0f)
            return 10;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 2") && distance>0.0f)
            return 10;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 3") && distance>0.0f)
            return 10;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 4") && distance>0.0f)
            return 10;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC20") && distance>0.0f)
            return 24;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1") && distance>0.0f)
            return 24;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2") && distance>0.0f)
            return 24;
        else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80") && distance>0.0f)
            return 24;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC30") && distance>0.0f)
            return 24;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC50") && distance>0.0f)
            return 24;
        else if(selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A") && distance>0.0f)
            return 24;
        else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B") && distance>0.0f)
            return 24;
        else
            return 0;
    }

    public float getRampaJaulas(String selected_Type, String
selected_Class, float distance ){
        if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 1") && distance > 0.0f)
            return 1;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 2") && distance > 0.0f)
            return 1;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 3") && distance >0.0f)
            return 1;
        else if (selected_Type.equals("Pasarela") &&
selected_Class.equals("Tipo 4") && distance >0.0f)
            return 1;
        else if (selected_Type.equals("Compuerta") &&
```

```
selected_Class.equals("MLC20") && distance >0.0f)
                return 2;
            else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T1") && distance >0.0f)
                return 2;
            else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC50T2") && distance >0.0f)
                return 2;
            else if (selected_Type.equals("Compuerta") &&
selected_Class.equals("MLC80") && distance >0.0f)
                return 2;
            else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC30") && distance >0.0f)
                return 2;
            else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC50") && distance >0.0f)
                return 2;
            else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80A") && distance >0.0f)
                return 2;
            else if (selected_Type.equals("Puente") &&
selected_Class.equals("MLC80B") && distance >0.0f)
                return 2;
            else return 0;
        }
    }
```

1.4. Page 4.

```
package com.ahmadalqadi.app.letipuente;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

public class page4 extends AppCompatActivity {

    Spinner merloSpinner;
    Spinner schottelSpinner;
    Spinner camionM250Spinner;
    Spinner remolquesSpinner;
    Spinner camion7217Spinner;
    Spinner camionVemperSpinner;
    Spinner remolquesVemperSpinner;
    EditText DistanciaEnKM_EditText;
    TextView ConsumoDeCombustibleTextView;
    TextView CuantasIdasYVueltasTextView;
    Spinner gruaLunaSpinner;
    TextView avisosLegalesTextView1;
    TextView avisosLegalesTextView2;
    TextView avisosLegalesTextView3;
    TextView avisosLegalesTextView4;
```

```
    TextView avisosLegalesTextView5;
    Button exitButton;
    Button restartButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_page4);

        merloSpinner =
findViewById(R.id.MERLO_Number_Of_Vehicles);
        schottelSpinner =
findViewById(R.id.SCHOTTEL_Number_Of_Vehicles);
        camionM250Spinner =
findViewById(R.id.CAMIÓN_M250_Number_Of_Vehicles);
        remolquesSpinner =
findViewById(R.id.REMOLQUES_GONTRAILER_M250_Number_Of_Vehicles);
        camion7217Spinner =
findViewById(R.id.CAMIÓN_7217_Number_Of_Vehicles);
        camionVemperSpinner =
findViewById(R.id.CAMIÓN_VEMPER_Number_Of_Vehicles);
        remolquesVemperSpinner =
findViewById(R.id.REMOLQUES_GONTRAILER_VEMPER_Number_Of_Vehicles
);
        DistanciaEnKM_EditText =
findViewById(R.id.DistanciaEnKM_EditText);
        ConsumoDeCombustibleTextView =
findViewById(R.id.Consumo_de_combustible_TextView);
        CuantasIdasYVueltasTextView =
findViewById(R.id.Cuantas_idas_y_vueltas_TextView);
        gruaLunaSpinner =
findViewById(R.id.Grua_Luna_Number_Of_Vehicles);
        avisosLegalesTextView1 =
findViewById(R.id.Avisos_Legales_TextView1);
        avisosLegalesTextView2 =
findViewById(R.id.Avisos_Legales_TextView2);
        avisosLegalesTextView3 =
findViewById(R.id.Avisos_Legales_TextView3);
        avisosLegalesTextView4 =
findViewById(R.id.Avisos_Legales_TextView4);
        avisosLegalesTextView5 =
findViewById(R.id.Avisos_Legales_TextView5);
        exitButton = findViewById(R.id.btn_exit);
        restartButton = findViewById(R.id.btn_restart);

        exitButton.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                finishAffinity();
                System.exit(0);
            }
        });

        restartButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(page4.this, page2.class);
                startActivity(i);
            }
```

```
        });

        ArrayAdapter<CharSequence> items_3_Adapter =
ArrayAdapter.createFromResource(this, R.array.spinner_items_3,
android.R.layout.simple_spinner_item);

items_3_Adapter.setDropDownViewResource(android.R.layout.simple_
spinner_dropdown_item);
        merloSpinner.setAdapter(items_3_Adapter);
        schottelSpinner.setAdapter(items_3_Adapter);
        gruaLunaSpinner.setAdapter(items_3_Adapter);

        ArrayAdapter<CharSequence> items_10_Adapter =
ArrayAdapter.createFromResource(this, R.array.spinner_items_10,
android.R.layout.simple_spinner_item);

items_10_Adapter.setDropDownViewResource(android.R.layout.simple
_spinner_dropdown_item);
        camionM250Spinner.setAdapter(items_10_Adapter);
        remolquesSpinner.setAdapter(items_10_Adapter);
        camion7217Spinner.setAdapter(items_10_Adapter);
        camionVemperSpinner.setAdapter(items_10_Adapter);
        remolquesVemperSpinner.setAdapter(items_10_Adapter);

        DistanciaEnKM_EditText.addTextChangedListener(new
TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int
start, int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start,
int before, int count) {

            }

            @Override
            public void afterTextChanged(Editable s) {
                try {
                    updateValues();
                }
                catch(Exception e){

                }
            }
        });
    }


    public double getConsumo_De_Combustible(int
camionM250Vehicles, int camion7217Vehicles, int
camionVemperVehicles, float distanceInKilometers,float
cuantasIdasYVueltas , double ahorro7217, float
ahorroM250_Vemper, int GruaLuna){
        Log.d("consumo", "camionM250Vehicles: " +
camionM250Vehicles);
        Log.d("consumo", "camion7217Vehicles: " +
camion7217Vehicles);
        Log.d("consumo", "camionVemperVehicles: " +
```

```
camionVemperVehicles);
        Log.d("consumo", "distanceInKilometers: " +
distanceInKilometers);
        Log.d("consumo", "ahorro7217: " + ahorro7217);
        Log.d("consumo", "ahorroM250_Vemper: " +
ahorroM250_Vemper);
        Log.d("consumo", "GruaLuna: " + GruaLuna);

        return
(camionM250Vehicles*55+camion7217Vehicles*35+camionVemperVehicle
s*60)*(distanceInKilometers/100)*cuantasIdasYVueltas-ahorro7217-
ahorroM250_Vemper+GruaLuna*60*(distanceInKilometers/100);
    }

    public float getcuantasIdasYVueltas(float viajes){
        return viajes*2-1;
    }

    public double getAhorro7217(float sobrantes7217, double
distanceInKM, float cuantasIdasYVueltas){
        if (cuantasIdasYVueltas > 1)
            if (sobrantes7217>0)
                return (sobrantes7217 * 35 * (distanceInKM/100))
* 2;
            else return 0;
        else  if (sobrantes7217 > 0)
            return (sobrantes7217 * 35 * (distanceInKM/100));
        else return 0;
    }

    public float getAhorroM250_Vemper(float cuantasIdasYVueltas,
float N22){
        if (cuantasIdasYVueltas > 1)
            return N22;
        else return N22 / 2;
    }

    public float getViajes(float J12, float J13, float K2, float
camion7217Vehicles, float controlPontonespontonesSobrantes,
float J7){
        if (J12 > J13)
            return (float) Math.ceil((K2 - J13 *
camion7217Vehicles - controlPontonespontonesSobrantes) /
(camion7217Vehicles + J7) + J13);
        else return J13;
    }

    private float getJ12(float camion7271Vehicles, float K2){
        if (camion7271Vehicles > 0)
            return (float) Math.ceil(K2 / camion7271Vehicles);
        else return 0;
    }

    private float getJ13(float camion7271Vehicles, float J2, int
J8, float K2, int J7){
        if (camion7271Vehicles>0)
            return (float) Math.ceil(J2/J8);
        else return (float) Math.ceil(J2/J8+K2/J7);
    }

    private float getJ2(float PExtremoTotal, float
```

```
PCentralTotal, int MerloVehicles, int schottelBarca){
        return PExtremoTotal + PCentralTotal + 6 * MerloVehicles
+ 3 * schottelBarca;
    }

    private int getJ8(int camionM250Vehicles, int
remolquesVehicles, int camionVemperVehicles, int
remolquesVemperVehicles){
        return camionM250Vehicles * 2 + remolquesVehicles * 6 +
camionVemperVehicles * 3 + remolquesVemperVehicles * 3;
    }

    private float getK2(float cabestrantesJaulas, float
placasDeflectorasJaulas, float rodillosJaulas, float
bordillosJaulas, float rampasJaulas, float horquillasJaulas,
float estribosJaulas){
        double L2 = horquillasJaulas + Math.ceil(estribosJaulas
/ 8);
        int L3 = (int)(L2 / 12);
        return cabestrantesJaulas + placasDeflectorasJaulas +
rodillosJaulas + bordillosJaulas * 2 + L3 + rampasJaulas;
    }

    private int getJ7(int camionM250Vehicles, int
remolquesVehicles, int camionVemperVehicles, int
remolquesVemperVehicles){
        return camionM250Vehicles * 2 + camionVemperVehicles * 2
+ remolquesVehicles * 6 + remolquesVemperVehicles * 3;
    }

    public float getN22(int camionVemperVehicles,float
M250VemperSobrantes, float distanceInKilometers ){
        if (M250VemperSobrantes > 0){
            if (camionVemperVehicles > 0 && camionVemperVehicles
- M250VemperSobrantes > 0)
                return 2 * (M250VemperSobrantes * 60 *
(distanceInKilometers / 100));
            else return 2*((M250VemperSobrantes -
camionVemperVehicles) * 55 * (distanceInKilometers / 100) +
camionVemperVehicles * 60 * (distanceInKilometers / 100));
        }
        else return 0;
    }

    public float getControlJaulasVehiculosSobrantes(float
controlJaulasJaulasTransporte, float K2){
        if (controlJaulasJaulasTransporte > K2)
            return (float)
Math.ceil((controlJaulasJaulasTransporte - K2) / 2);
        else return 0;
    }

    public float getControlPontonesPontonesTransporte(int J8,
float J2){
        return (float) Math.ceil(J2/J8)*J8;
    }

    public int getM6(int remolquesVehicles, int
remolquesVemperVehicles){
        return remolquesVehicles * 6 + remolquesVemperVehicles *
3;
```

```
    }

    public int getSobrantes7217(float
controlJaulasVehiculosSobrantes,int camion7217Vehicles){
        if (controlJaulasVehiculosSobrantes >=
camion7217Vehicles)
            return camion7217Vehicles;
        else return 0;
    }

    public float getM250VemperSobrantes(int camion7217Vehicles,
float controlJaulasVehiculosSobrantes, float sobrantes7217,float
viajes ,float J13, float controlPontonesVehiculosSobrantes){
        if (viajes == J13 && controlJaulasVehiculosSobrantes >
camion7217Vehicles)
            return controlPontonesVehiculosSobrantes -
sobrantes7217;
        else return controlPontonesVehiculosSobrantes;
    }

    public float getJaulasTransporte(float camion7217Vehicles,
float J13, float viajes, float J7, float
controlPontonesPontonesSobrantes ){
        return J13 * camion7217Vehicles + (viajes - J13) * (J7 +
camion7217Vehicles) + controlPontonesPontonesSobrantes;
    }

    public void showWarnings(int MerloVehicles, int
schottelBarca, int remolquesVehicles, int
remolquesVemperVehicles, float controlJaulasVehiculosSobrantes,
float sobrantes7217, float M250VemperSobrantes){
        if ((MerloVehicles>0 && remolquesVemperVehicles==0))
            avisosLegalesTextView1.setVisibility(View.VISIBLE);
        else
avisosLegalesTextView1.setVisibility(View.INVISIBLE);

        if ((schottelBarca>0 && remolquesVehicles==0)){
            avisosLegalesTextView5.setVisibility(View.VISIBLE);
        }
        else
avisosLegalesTextView5.setVisibility(View.INVISIBLE);

        if (controlJaulasVehiculosSobrantes > 0){
            avisosLegalesTextView2.setText("Sobran Vehículos
Para El Último Viaje");
            avisosLegalesTextView2.setVisibility(View.VISIBLE);
        }
        else
avisosLegalesTextView2.setVisibility(View.INVISIBLE);

        if (sobrantes7217 > 0){
            avisosLegalesTextView3.setText("-Sobran " +
(int)sobrantes7217 + " Vehículos 7217");
            avisosLegalesTextView3.setVisibility(View.VISIBLE);
        }
        else
avisosLegalesTextView3.setVisibility(View.INVISIBLE);

        if (M250VemperSobrantes > 0){
            avisosLegalesTextView4.setText("-Sobran " +
(int)M250VemperSobrantes + " Vehículos M250/Vemper");
```

```
            avisosLegalesTextView4.setVisibility(View.VISIBLE);
        }
        else
avisosLegalesTextView4.setVisibility(View.INVISIBLE);

    }
    public float getControlPontonesVehiculosSobrantes(float M6,
float M11, float controlPontonesPontonesSobrantes, float
camionM250Vehicles, float camionVemperVehicle ){
        double value;
        if (M6 > 0)
            value = Math.ceil(M11 / Math.ceil(M6 / 6));
        else value = Math.ceil(M11 / 2);

        if (M11 < 0){
            if (controlPontonesPontonesSobrantes - M6 > M11)
                return (float)
Math.ceil((controlPontonesPontonesSobrantes - M6) / 2);
            else return (float) Math.ceil((M11 * -1) / 2);
        }
        else if (M11 > 0)
            return (float) (camionM250Vehicles +
camionVemperVehicle - value);
        else return 0;

    }

    public float getcontrolPontonesPontonesSobrantes(float
pontonesTransporte, float J2){
        if (pontonesTransporte > J2)
            return pontonesTransporte - J2;
        else return 0;
    }


    public float getM11(float K2, float J13, float
camion7217Vehicles, float controlPontonesPontonesSobrantes){
        return K2 - J13 * camion7217Vehicles -
controlPontonesPontonesSobrantes;
    }

    public void updateValues(){
        Bundle data_Bundle = getIntent().getExtras();
        String selectedType =
data_Bundle.getString("selected_Type");
        String selectedClass =
data_Bundle.getString("selected_Class");
        float distance = data_Bundle.getFloat("distance");
        float cabestrantesJaulas =
data_Bundle.getFloat("cabestrantesJaulas");
        float placasDeflectorasJaulas =
data_Bundle.getFloat("placasDeflectorasJaulas");
        float rodillosJaulas =
data_Bundle.getFloat("rodillosJaulas");
        float bordillosJaulas =
data_Bundle.getFloat("bordillosJaulas");
        float rampasJaulas =
data_Bundle.getFloat("rampasJaulas");
        float horquillasJaulas =
data_Bundle.getFloat("horquillasJaulas");
        float estribosJaulas =
```

```
data_Bundle.getFloat("estribosJaulas");
        float pExtremoTotal =
data_Bundle.getFloat("pExtremoTotal");
        float pCentralTotal =
data_Bundle.getFloat("pCentralTotal");
        float distanceInKilometers =
Float.parseFloat(DistanciaEnKM_EditText.getText().toString());
        int merloVehicles =
merloSpinner.getSelectedItemPosition();
        int schottelBarca =
schottelSpinner.getSelectedItemPosition();



        int camionM250Vehicles
=camionM250Spinner.getSelectedItemPosition();
        int camion7217Vehicles =
camion7217Spinner.getSelectedItemPosition();
        int camionVemperVehicles =
camionVemperSpinner.getSelectedItemPosition();
        int remolquesVehicles =
remolquesSpinner.getSelectedItemPosition();
        int remolquesVemperVehicles =
remolquesVemperSpinner.getSelectedItemPosition();


        int J8 = getJ8(camionM250Vehicles, remolquesVehicles,
camionVemperVehicles,remolquesVemperVehicles);
        float J2 =
getJ2(pExtremoTotal,pCentralTotal,merloVehicles, schottelBarca);
        int J7 = getJ7(camionM250Vehicles,
remolquesVehicles,camionVemperVehicles,remolquesVemperVehicles);
        float K2 = getK2(cabestrantesJaulas,
placasDeflectorasJaulas,rodillosJaulas,bordillosJaulas,rampasJau
las,horquillasJaulas,estribosJaulas);
        float J13 = getJ13(camion7217Vehicles,J2,J8,K2,J7);
        float J12 = getJ12(camion7217Vehicles,K2);
        float pontonesTransporte =
getControlPontonesPontonesTransporte(J8,J2);
        float controlPontonesPontonesSobrantes =
getcontrolPontonesPontonesSobrantes(pontonesTransporte, J2);
        float viajes =
getViajes(J12,J13,K2,camion7217Vehicles,controlPontonesPontonesS
obrantes,J7);
        float controlJaulasJaulasTransporte =
getJaulasTransporte(camion7217Vehicles,J13,viajes,J7,controlPont
onesPontonesSobrantes);
        float controlJaulasVehiculosSobrantes =
getControlJaulasVehiculosSobrantes(controlJaulasJaulasTransporte
, K2);
        float sobrantes7217 =
getSobrantes7217(controlJaulasVehiculosSobrantes,camion7217Vehic
les);
        float cuantasIdasYVueltas =
getcuantasIdasYVueltas(viajes);
        float M11 =
getM11(K2,J13,camion7217Vehicles,controlPontonesPontonesSobrante
s);
        float M6 =
getM6(remolquesVehicles,remolquesVemperVehicles);
        float controlPontonesVehiculosSobrantes =
```

```
getControlPontonesVehiculosSobrantes(M6,M11,controlPontonesPonto
nesSobrantes,camionM250Vehicles,camionVemperVehicles);
        float M250VemperSobrantes =
getM250VemperSobrantes(camion7217Vehicles,controlJaulasVehiculos
Sobrantes,sobrantes7217,viajes,J13,controlPontonesVehiculosSobra
ntes);
        float N22 =
getN22(camionVemperVehicles,M250VemperSobrantes,distanceInKilome
ters);
        float ahorroM250_Vemper =
getAhorroM250_Vemper(cuantasIdasYVueltas,N22);
        double ahorro7217 =
getAhorro7217(sobrantes7217,distanceInKilometers,cuantasIdasYVue
ltas);
        int grauLuna =
gruaLunaSpinner.getSelectedItemPosition();
        double consumoDeCombustible =
getConsumo_De_Combustible(camionM250Vehicles,camion7217Vehicles,
camionVemperVehicles,distanceInKilometers,cuantasIdasYVueltas,ah
orro7217,ahorroM250_Vemper,grauLuna);
        ConsumoDeCombustibleTextView.setText("" +
(float)consumoDeCombustible);
        CuantasIdasYVueltasTextView.setText("" +
cuantasIdasYVueltas);
        showWarnings(merloVehicles, schottelBarca,
remolquesVemperVehicles,remolquesVehicles,controlJaulasVehiculos
Sobrantes,sobrantes7217,M250VemperSobrantes);

    }
}
```

2. Lenguaje XML:

2.1. MainActivity.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="70"
            android:layout_marginBottom="50dp"
            android:layout_marginTop="50dp"
            app:srcCompat="@drawable/rpei12" />

        <me.biubiubiu.justifytext.library.JustifyTextView
```

```
            android:id="@+id/welcome_screen_text_view1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:layout_marginLeft="8dp"
            android:layout_marginRight="8dp"
            android:layout_marginTop="16dp"
            android:text="@string/welcome_screen_text1"
            android:layout_weight="1"/>

        <me.biubiubiu.justifytext.library.JustifyTextView
            android:id="@+id/welcome_screen_text_view2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="100dp"
            android:layout_marginLeft="8dp"
            android:layout_marginRight="8dp"
            android:layout_marginTop="16dp"
            android:text="@string/welcome_screen_text2"
            android:layout_weight="1"/>

        <Button
            android:id="@+id/continue_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|center_horizontal"
            android:text="@string/continue_button"
            android:layout_weight="1"/>
    </LinearLayout>
```

2.2. Page 2.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".page2">



    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">



            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:textSize="25sp"
                android:textStyle="bold"

android:background="@color/design_default_color_primary"
                android:textColor="@android:color/white"
                android:text="Datos del puente"/>
        <View
```

```
        android:layout_width="match_parent"
        android:layout_height="2dp"
        android:background="@android:color/black" />

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/type_text_view"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="20dp"/>


        <Spinner
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/types_spinner"
            android:layout_marginLeft="20dp" />
        <TextView
            android:id="@+id/type_validation"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/validation_error_message"
            android:visibility="invisible"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@android:color/black" />

        <TextView
            android:id="@+id/class_text_view"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/class_text_view"
            android:layout_marginLeft="10dp"/>
        <Spinner
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/classes_spinner"
            android:layout_marginLeft="20dp"/>

        <TextView
            android:id="@+id/class_validation"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/validation_error_message"
            android:visibility="invisible"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@android:color/black" />

        <TextView
            android:id="@+id/distance_text_view"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/distance_text_view"
            android:layout_marginLeft="10dp"/>
```

```
            <EditText
                android:id="@+id/distance_editText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginLeft="20dp"
                android:ems="10"
                android:inputType="numberDecimal"
                android:text="0"/>

            <TextView
                android:id="@+id/distance_validation"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

android:text="@string/amount_validation_error_message"
                android:visibility="invisible"/>

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@android:color/black" />


        <Button
            android:id="@+id/continue_button_page2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:text="@string/continue_button"/>



        <TextView
            android:id="@+id/warning_text_view"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/warning_label_text_view"
            android:layout_marginLeft="20dp"
            android:visibility="invisible"/>



        <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/autopuenteal"
            android:scaleType="fitXY"/>

    </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

2.3. Page 3.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    tools:context=".page3">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">
                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:gravity="center_horizontal"
                    android:textSize="40sp"
                    android:textStyle="bold"

android:background="@color/design_default_color_primary"
                    android:textColor="@android:color/white"
                    android:text="@string/page_5_title" />
                <View
                    android:layout_width="match_parent"
                    android:layout_height="1dp"
                    android:background="@android:color/black" />

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginLeft="5dp"
                    android:textSize="16sp"
                    android:textStyle="bold"
                    android:text="Pontón extremo" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginBottom="15sp"
                android:orientation="horizontal">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:gravity="left"
                        android:text="Total: " />

                    <TextView
                        android:id="@+id/P_Extremo_total"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="2"
                        android:text="0" />

            </LinearLayout>
                <View
                    android:layout_width="match_parent"
                    android:layout_height="1dp"
```

```
    android:background="@android:color/black" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:textSize="16sp"
    android:textStyle="bold"
    android:text="Pontón central" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="15sp"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:text="Total: " />

    <TextView
        android:id="@+id/P_Central_total"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="0" />


</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@android:color/black" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:textSize="16sp"
    android:textStyle="bold"
    android:text="Estribos" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="15sp"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:text="Total: " />

    <TextView
        android:id="@+id/Estribos_total"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="0" />
```

35

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Jaulas: " />

        <TextView
            android:id="@+id/Estribos_Jaulas"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:text="0 Cajas" />


    </LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@android:color/black" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:textSize="16sp"
        android:textStyle="bold"
        android:text="Horquillas" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="15sp"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="left"
            android:text="Total: " />

        <TextView
            android:id="@+id/Horquillas_total"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:text="0" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Jaulas: " />

        <TextView
            android:id="@+id/Horquillas_Jaulas"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1.8"
            android:text="0 Cajas" />

        <TextView
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="right"
            android:text="Sueltos: " />

        <TextView
            android:id="@+id/Horquillas_Sueltos"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:text="0" />
    </LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@android:color/black" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:textSize="16sp"
        android:textStyle="bold"
        android:text="Cabestrantes" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="15sp"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="left"
            android:text="Total: " />

        <TextView
            android:id="@+id/Cabestrantes_total"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:text="0" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Jaulas: " />

        <TextView
            android:id="@+id/Cabestrantes_Jaulas"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:text="0 Cajas" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="right"
```

37

```
                android:text="Sueltos: " />

            <TextView
                android:id="@+id/Cabestrantes_Sueltos"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:text="0" />
        </LinearLayout>

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@android:color/black" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:textSize="16sp"
            android:textStyle="bold"
            android:text="Placas deflectoras" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginBottom="15sp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="left"
                android:text="Total: " />

            <TextView

android:id="@+id/Placas_Deflectoras_total"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:text="0" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="Jaulas: " />

            <TextView

android:id="@+id/Placas_Deflectoras_Jaulas"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:text="0 Cajas" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="right"
                android:text="Sueltos: " />
```

38

```
                    <TextView

android:id="@+id/Placas_Deflectoras_Sueltos"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="2"
                        android:text="0" />
                </LinearLayout>

                <View
                    android:layout_width="match_parent"
                    android:layout_height="1dp"
                    android:background="@android:color/black" />
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginLeft="5dp"
                    android:textSize="16sp"
                    android:textStyle="bold"
                    android:text="Traveseros" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginBottom="15sp"
                    android:orientation="horizontal">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:gravity="left"
                        android:text="Total: " />

                    <TextView
                        android:id="@+id/Traveseros_total"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="2"
                        android:text="0" />


                </LinearLayout>

                <View
                    android:layout_width="match_parent"
                    android:layout_height="1dp"
                    android:background="@android:color/black" />
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginLeft="5dp"
                    android:textSize="16sp"
                    android:textStyle="bold"
                    android:text="Soportes cabestrante" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginBottom="15sp"
                    android:orientation="horizontal">
```

```
                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:gravity="left"
                        android:text="Total: " />

                    <TextView

android:id="@+id/Sportes_Cabestrante_total"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="2"
                        android:text="0" />


                </LinearLayout>

                <View
                    android:layout_width="match_parent"
                    android:layout_height="1dp"
                    android:background="@android:color/black" />
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginLeft="5dp"
                    android:textSize="16sp"
                    android:textStyle="bold"
                    android:text="Rodillos" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_marginBottom="15sp"
                    android:orientation="horizontal">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:gravity="left"
                        android:text="Total: " />

                    <TextView
                        android:id="@+id/Rodillos_total"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="2"
                        android:text="0" />

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:gravity="center"
                        android:text="Jaulas: " />

                    <TextView
                        android:id="@+id/Rodillos_Jaulas"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_weight="2"
                        android:text="0 Cajas" />
```

40

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="right"
        android:text="Sueltos: " />

    <TextView
        android:id="@+id/Rodillos_Sueltos"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="0" />
</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@android:color/black" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:textSize="16sp"
    android:textStyle="bold"
    android:text="Bolardos" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="15sp"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:text="Total: " />

    <TextView
        android:id="@+id/Bolardos_total"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="0" />


</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@android:color/black" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:textSize="16sp"
    android:textStyle="bold"
    android:text="Buques de alba" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="15sp"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:text="Total: " />

    <TextView
        android:id="@+id/Buques_de_Alba_total"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="0" />


</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@android:color/black" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:textSize="16sp"
    android:textStyle="bold"
    android:text="Bordillos" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="15sp"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:text="Total: " />

    <TextView
        android:id="@+id/Bordillos_total"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="0" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Jaulas: " />

    <TextView
        android:id="@+id/Bordillos_Jaulas"
```

```
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:text="0 Cajas" />


        </LinearLayout>

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@android:color/black" />
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="16sp"
                android:textStyle="bold"
                android:text="@string/anchura" />

            <TextView
                android:id="@+id/anchura_amount"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="0" />
        </LinearLayout>

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@android:color/black" />
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"
                android:layout_marginLeft="15dp"
                android:orientation="vertical">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="15dp"
                    android:textSize="17sp"

android:background="@android:color/holo_green_light"
                    android:textStyle="bold"
                    android:text="Material rampas " />

                <LinearLayout
                    android:layout_width="wrap_content"

android:layout_height="wrap_content">
```

```
                            <TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
                                android:layout_marginLeft="15dp"
                                android:text="Cumbreras : " />
                            <TextView
android:id="@+id/cumbreras_Total"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
                                android:text="0" />
                        </LinearLayout>
                        <LinearLayout
                            android:layout_width="wrap_content"
android:layout_height="wrap_content">
                            <TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
                                android:layout_marginLeft="15dp"
                                android:text="Vigas de rampa : "
/>
                            <TextView
android:id="@+id/vigas_de_rampa_Total"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
                                android:text="0" />
                        </LinearLayout>
                        <LinearLayout
                            android:layout_width="wrap_content"
android:layout_height="wrap_content">
                            <TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
                                android:layout_marginLeft="15dp"
                                android:text="Finales de rampa :
" />
                            <TextView
android:id="@+id/finales_de_rampa_Total"
```

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:text="0" />
                        </LinearLayout>

                        <LinearLayout
                            android:layout_width="wrap_content"

android:layout_height="wrap_content">

                                <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:layout_marginLeft="15dp"
                                android:text="Postecillos : " />

                                <TextView

android:id="@+id/postecillos_Total"

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:text="0" />
                        </LinearLayout>

                        <LinearLayout
                            android:layout_width="wrap_content"

android:layout_height="wrap_content">

                                <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:layout_marginLeft="15dp"
                                android:text="Gatos : " />

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

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:text="0" />
                        </LinearLayout>

                        <LinearLayout
                            android:layout_width="wrap_content"

android:layout_height="wrap_content">

                                <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"
```

```
                                android:layout_marginLeft="15dp"
                                android:text="Cable : " />

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

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:text="0" />
                        </LinearLayout>

                        <LinearLayout
                            android:layout_width="wrap_content"

android:layout_height="wrap_content">

                            <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:layout_marginLeft="15dp"
                                android:text="Jaulas/Transporte
: " />

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

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                android:text="0" />
                        </LinearLayout>

                    </LinearLayout>

                    <LinearLayout
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="30dp"
                        android:orientation="vertical">

                        <TextView
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:layout_marginTop="15dp"
                            android:textSize="17sp"

android:background="@android:color/holo_green_light"
                            android:textStyle="bold"
                            android:text="Caja de herramientas"
/>

                        <TextView
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="2 Martillos" />

                        <TextView
                            android:layout_width="wrap_content"
```

```
                android:layout_height="wrap_content"
                android:text="6 Martillos fibra" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="8 Barras de pincho" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="4 Barras de
acoplamiento" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="2 Barras de
nivelación" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="6 Llaves del 32" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="4 Escalerillas" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Ammarras" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="4 Cáncamos de
cabestrante" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="4 Cadenas" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="20 Piquetes" />


        </LinearLayout>


    </LinearLayout>

    <Button
        android:id="@+id/continue_button_page3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

47

```
                android:text="@string/continue_button"
                android:layout_gravity="center_horizontal"/>
        </LinearLayout>


    </ScrollView>

</LinearLayout>
```

2.4. Page 4.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".page4">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_horizontal"
        android:orientation="vertical">

        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:gravity="center_horizontal"
                    android:textSize="40sp"
                    android:textStyle="bold"

android:background="@color/design_default_color_primary"
                    android:textColor="@android:color/white"
                    android:text="@string/page_4_title" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical"
                    android:layout_gravity="center_vertical">
                    <View
                        android:layout_width="match_parent"
                        android:layout_height="1dp"

android:background="@android:color/black" />
                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="12dp"
                        android:layout_marginRight="12dp"
```

```
                android:layout_marginTop="12dp"
                android:orientation="horizontal">


                <TextView
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_weight="3"
                    android:gravity="center_horizontal"
                    android:textStyle="bold"

android:background="@android:color/holo_green_light"
                    android:text="Tipo de vehículos" />

                <View
                    android:layout_width="1dp"
                    android:layout_height="match_parent"

android:background="@android:color/black" />
                <TextView
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_weight="3"
                    android:gravity="center_horizontal"
                    android:textStyle="bold"

android:background="@android:color/holo_green_light"
                    android:text="Capacidad" />

                <View
                    android:layout_width="1dp"
                    android:layout_height="match_parent"

android:background="@android:color/black" />
                <TextView
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_weight="2"
                    android:gravity="center_horizontal"
                    android:textStyle="bold"

android:background="@android:color/holo_green_light"
                    android:text="N° vehículos" />

            </LinearLayout>
            <View
                android:layout_width="match_parent"
                android:layout_height="1dp"

android:background="@android:color/black" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginLeft="12dp"
                android:layout_marginRight="12dp"
                android:orientation="horizontal">


                <TextView
                    android:layout_width="0dp"
```

```
                android:layout_height="wrap_content"
android:layout_gravity="center_vertical"
                android:layout_weight="3"
                android:gravity="center_horizontal"
                android:text="Merlo" />
            <View
                android:layout_width="1dp"
                android:layout_height="match_parent"
android:background="@android:color/black" />

            <TextView
                android:layout_width="0dp"
                android:layout_height="wrap_content"
android:layout_gravity="center_vertical"
                android:layout_weight="3"
                android:gravity="center_horizontal"
                android:text="-" />
            <View
                android:layout_width="1dp"
                android:layout_height="match_parent"
android:background="@android:color/black" />

            <Spinner
android:id="@+id/MERLO_Number_Of_Vehicles"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
android:layout_gravity="center_vertical"
                android:layout_weight="2" />

        </LinearLayout>
        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
android:background="@android:color/black" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="12dp"
            android:layout_marginRight="12dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="0dp"
                android:layout_height="wrap_content"
android:layout_gravity="center_vertical"
                android:layout_weight="3"
                android:gravity="center_horizontal"
                android:text="Schottel" />

            <View
                android:layout_width="1dp"
                android:layout_height="match_parent"
```

```
android:background="@android:color/black" />
                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="-" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <Spinner

android:id="@+id/SCHOTTEL_Number_Of_Vehicles"
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="2" />

                    </LinearLayout>
                    <View
                        android:layout_width="match_parent"
                        android:layout_height="1dp"

android:background="@android:color/black" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="12dp"
                        android:layout_marginRight="12dp"
                        android:orientation="horizontal">

                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="Camión M250" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="(2 Pontones) O (2
```

```
Jaulas) O (1 Jaula bordillos)" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <Spinner

android:id="@+id/CAMIÓN_M250_Number_Of_Vehicles"
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="2" />

                    </LinearLayout>
                    <View
                        android:layout_width="match_parent"
                        android:layout_height="1dp"

android:background="@android:color/black" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="12dp"
                        android:layout_marginRight="12dp"
                        android:orientation="horizontal">

                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="Remolque
gontrailer/M250" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="Merlo O (6 pontones) O
Jaulas grandes" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <Spinner
```

```
android:id="@+id/REMOLQUES_GONTRAILER_M250_Number_Of_Vehicles"
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="2" />

                    </LinearLayout>
                    <View
                        android:layout_width="match_parent"
                        android:layout_height="1dp"

android:background="@android:color/black" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="12dp"
                        android:layout_marginRight="12dp"
                        android:orientation="horizontal">

                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="Camión 7217" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="1 Jaula +Caja
herramientas/cosas sueltas" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <Spinner

android:id="@+id/CAMIÓN_7217_Number_Of_Vehicles"
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="2" />

                    </LinearLayout>
```

```
                    <View
                        android:layout_width="match_parent"
                        android:layout_height="1dp"

android:background="@android:color/black" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="12dp"
                        android:layout_marginRight="12dp"
                        android:orientation="horizontal">

                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="Camión vempar" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="(3 Pontones) O (2
Jaulas) O (1 Jaula bordillos)" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <Spinner

android:id="@+id/CAMIÓN_VEMPER_Number_Of_Vehicles"
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="2" />

                    </LinearLayout>
                    <View
                        android:layout_width="match_parent"
                        android:layout_height="1dp"

android:background="@android:color/black" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
```

```
                        android:layout_marginLeft="12dp"
                        android:layout_marginRight="12dp"
                        android:orientation="horizontal">

                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="Remolque
gontrailer/vempar" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="(3 Pontones) O Jaulas
Grandes" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <Spinner

android:id="@+id/REMOLQUES_GONTRAILER_VEMPER_Number_Of_Vehicles"
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="2" />

                    </LinearLayout>
                    <View
                        android:layout_width="match_parent"
                        android:layout_height="1dp"

android:background="@android:color/black" />

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_marginLeft="12dp"
                        android:layout_marginRight="12dp"
                        android:orientation="horizontal">

                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"
```

```
android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="Grúa luna:" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <TextView
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="3"
                            android:gravity="center_horizontal"
                            android:text="-" />

                        <View
                            android:layout_width="1dp"
                            android:layout_height="match_parent"

android:background="@android:color/black" />
                        <Spinner

android:id="@+id/Grua_Luna_Number_Of_Vehicles"
                            android:layout_width="0dp"
                            android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                            android:layout_weight="2" />

                    </LinearLayout>

                </LinearLayout>
                <View
                    android:layout_width="match_parent"
                    android:layout_height="1dp"
                    android:background="@android:color/black" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical">

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:orientation="vertical">
                        <LinearLayout
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:orientation="vertical"
                            android:layout_marginTop="15dp"
                            android:layout_marginRight="25dp"
                            android:layout_marginLeft="25dp"
                            android:gravity="center_horizontal"

android:layout_gravity="center_horizontal">
                            <LinearLayout
```

```
android:layout_width="match_parent"

android:layout_height="wrap_content"

android:gravity="center_horizontal"

android:orientation="horizontal">

                                <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="1"

android:gravity="center_horizontal"
                                    android:textStyle="bold"
                                    android:text="Origen:" />

                                <EditText

android:id="@+id/origen_EditText"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="5" />
                            </LinearLayout>

                            <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:gravity="center_horizontal"

android:orientation="horizontal">

                                <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="1"

android:gravity="center_horizontal"
                                    android:textStyle="bold"
                                    android:text="Destino:" />

                                <EditText

android:id="@+id/destino_EditText"
```

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="5" />
                            </LinearLayout>

                            <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:gravity="center_horizontal"

android:orientation="horizontal">

                                    <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="1"

android:gravity="center_horizontal"
                                    android:textStyle="bold"
                                    android:text="Distancia
(KM):" />

                                    <EditText

android:id="@+id/DistanciaEnKM_EditText"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="5"
                                    android:text="0" />
                            </LinearLayout>

                            <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:gravity="center_horizontal"
                                    android:orientation="horizontal"
                                    android:layout_marginTop="5dp">

                                    <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"
```

```
android:layout_gravity="center_vertical"
                                    android:layout_weight="1"

android:gravity="center_horizontal"
                                    android:textStyle="bold"
                                    android:text="Consumo \nde
combustible (L):" />

                                <TextView

android:id="@+id/Consumo_de_combustible_TextView"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="5" />
                            </LinearLayout>

                            <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:gravity="center_horizontal"
                                android:orientation="horizontal"
                                android:layout_marginTop="5dp">

                                <TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                                    android:layout_weight="1"

android:gravity="center_horizontal"
                                    android:textStyle="bold"
                                    android:text="Idas y
vueltas:" />

                                <TextView

android:id="@+id/Cuantas_idas_y_vueltas_TextView"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="center_vertical"
                                    android:layout_weight="5" />
                            </LinearLayout>
                        </LinearLayout>
                    </LinearLayout>

                </LinearLayout>

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
```

```
                    android:orientation="vertical">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="center"
                        android:textStyle="bold"
                        android:textSize="22sp"

android:background="@android:color/holo_green_light"
                        android:text="Avisos para el usuario:"
                        android:layout_marginTop="15dp"/>

                    <TextView

android:id="@+id/Avisos_Legales_TextView1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="center"
                        android:text="-No se puede transportar
la Merlo"
                        android:visibility="invisible" />

                    <TextView

android:id="@+id/Avisos_Legales_TextView5"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="center"
                        android:text="-No se puede transportar
el SCHOTTEL"
                        android:visibility="invisible" />


                    <TextView

android:id="@+id/Avisos_Legales_TextView2"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="center"
                        android:text="Sobran vehículos para el
último viaje:"
                        android:visibility="invisible" />

                    <TextView

android:id="@+id/Avisos_Legales_TextView3"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:visibility="invisible"
                        android:layout_gravity="center"/>

                    <TextView

android:id="@+id/Avisos_Legales_TextView4"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:visibility="invisible"
                        android:layout_gravity="center"/>
                </LinearLayout>
                <ImageView
```

```
                    android:id="@+id/imageView"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="50dp"
                    android:layout_marginTop="50dp"
                    app:srcCompat="@drawable/autopuentealqadi"
/>


                <Button
                    android:id="@+id/btn_exit"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Cerrar el programa"/>

                <Button
                    android:id="@+id/btn_restart"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Volver a la primera pagina"/>

            </LinearLayout>
        </ScrollView>

    </LinearLayout>
```

Anexo 3: Manual de usuario de AutoPuente.

2018

AutoPuente

Manual de usuario de la aplicación AutoPuente

Realizado por:

Ahmad Mohammad Alqadi

INDICE

1. Introducción.

AutoPuente en una aplicación integrada en el entorno gráfico de Android Studio que utiliza el lenguaje de Java y el lenguaje XML. Con esta aplicación podrá calcular las herramientas necesarias para cualquier tipo y clase de puente flotante MAN, según los datos que los introduce el usuario, también nos permite calcular el número de viajes que deben realizar los vehículos, y por fin el consumo de combustible en litros. Se intentó evitar lo máximo posible introducir los datos manualmente para que no se cometan errores, sino se introducirán pulsando botones mediante listas desplegables. Utilizar Android Studio nos facilita utilizar nuestra aplicación con nuestros móviles que usan ese tipo de lenguaje, también se podrá utilizar mediante nuestros ordenadores y tabletas.

2. Inicio.

2.1. Pantalla de Bienvenida.

Nada más entrar al programa, saldrá una pantalla de bienvenida con el siguiente mensaje:

"Bienvenido a AutoPuente. Es un Programa utilizado en la Compañía de puentes flotantes del Regimiento de Pontoneros Numero 12 en Zaragoza. El programa le ayudará a calcular todas las herramientas necesarias para montar su puente flotante, también le ayudará a montar su plan de carga para transportar el puente.

El programa está hecho por el Alférez ALQADI del Ejército de Tierra de Jordania."

2.2. Introducir los datos.

Al pulsar al botón de continuar que sale en la primera pantalla, el programa te pedirá los datos necesarios para el cálculo de las herramientas necesarias (Pontones, Horquillas, Estribos, Jaulas,…etc.) en la segunda pantalla.

Los datos necesarios son Tipo, Clase y la Luz del puente.

El tipo consta de una lista desplegable donde salen los **tres tipos principales, Pasarela, Puente, y Compuerta.**

Al elegir el tipo, se cambiara el nombre del segundo dato necesario, si por ejemplo ponemos Tipo: Pasarela, el segundo dato será Tipo de pasarela. Y habrá otra lista desplegable de cada tipo con los tipos o clases utilizados para su cálculo eficiente.

Hay 4 tipos de pasarela, Tipo1, Tipo2, Tipo3, y Tipo4.

Son 4 clases de Compuertas. MLC20, MLC50T1, MLC50T2, y MLC80.

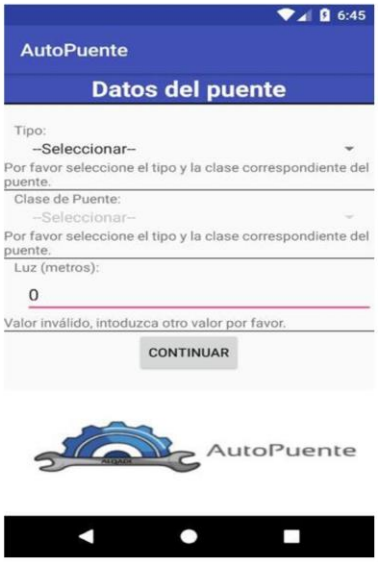
4

Son 4 clases de Puentes. MLC30, MLC50, MLC80A, y MLC80B. Debido a que las medidas de las compuertas son fijas, al seleccionar la clase de la compuerta, sale a su derecha la medida que el usuario debe meter en la luz del puente. El usuario debe meter el mismo número que aparece al lado de cada clase de las compuertas.

Después de introducir los datos, nos falta escribir a mano el tercer y el último dato para que el programa empiece a hacer los cálculos. Ese dato es la luz del puente necesario. Teniendo en cuenta que, **si el usuario introduce una luz mayor que 100 metros, saldrá un aviso de que es necesario utilizar anclas para nuestro puente.**

5

Si el usuario no mete el tipo o la clase o la luz del puente, el programa no le permite continuar, y le sale un mensaje de aviso de los datos que les falta por meter.



3. Las tablas que calcula el programa.

Al pulsar al botón de continuar el que sale en la segunda pantalla, nos saldrá tres tablas con el material necesario para montar el puente. La primera tabla es la tabla del material general necesario**, con la anchura en el caso de que si tenemos compuerta o puente.** La segunda tabla es la tabla del material de montar la Rampa. Y la última tabla es la tabla de las herramientas necesarias para montar el puente.

Esta tabla cambia de códigos al cambiar los datos de la primera pantalla **"Pantalla de Bienvenida"**.

A continuación, se explicara cada una de las tres tablas con los cálculos que hará el programa.

3.1. Tabla de material general.

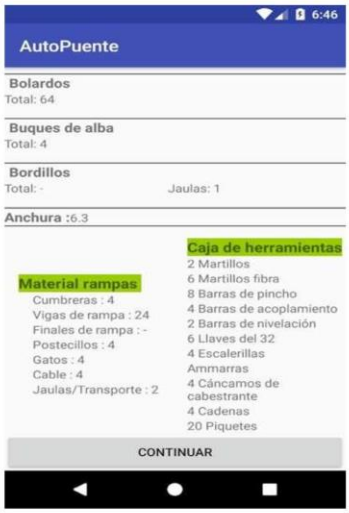
En esta tabla, se pondrá el número necesario de los pontones extremos y centrales, también las otras herramientas empaquetándolas en jaulas para facilitar el transporte. Se tuvo en cuenta en el caso de que tenemos material suelto, y eso para no empaquetar el material suelo en jaulas para que no ocupen más de lo que deberían ocupar. Todos estos cálculos dependen de los datos introducidos en la Pantalla de Bienvenida.

3.2. Tabla del material de la Rampa.

Esta tabla se ve en la parte inferior izquierda que sale en la misma pantalla de la tabla anterior "pantalla número 3". Tiene las herramientas necesarias para montar la Rampa de entrada y salida al puente. El número de las herramientas depende de los datos introducidos en la segunda pantalla. Se empaquetan las herramientas en jaulas para facilitar su transporte.

3.3. Caja de herramientas necesarias para montar el puente.

Esta tabla se ve en la parte inferior derecha y tiene el mismo material siempre. No depende de ningún dato introducido. Esta tabla se sitúa al final de la tercera pantalla.



4. Plan de carga.

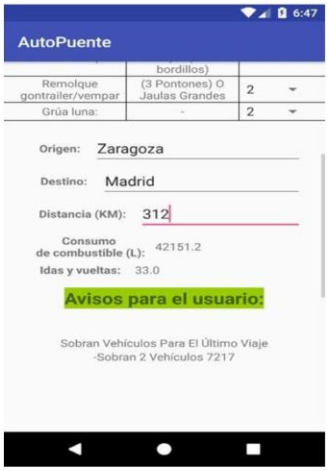
Al pulsar al botón de continuar que sale en la tercera pantalla, se cambia de pantalla y nos sale una tabla de los vehículos disponibles para hacer el transporte a la ciudad de destino, y otra tabla de la ciudad de destino.

4.1. Tabla de vehículos disponibles.

En esta tabla, el usuario elegirá cuantos vehículos tiene disponibles para hacer el transporte. Cada vehiculo tiene a su lado una lista desplegable para que el usuario elija de cuantos vehículos dispone. Esa lista será de 0-10 vehículos si es vehiculo de transporte, 0-3 vehículos si es Merlo, Grúa Luna o Schottel.

4.2. Tabla de kilometraje y combustible consumido.

Esta tabla sale a continuación de la tabla anterior y en la misma pantalla. En esta tabla, **el usuario elegirá la ciudad de destino y el kilometraje que van a realizar los vehículos en cada viaje**. Nada más introducir los datos, la tabla se actualiza automáticamente y nos sale cuantas idas y vueltas van a realizar los vehículos y el consumo de combustible en litros.



4.3. Avisos legales.

Al final de la cuarta pantalla, nos salen unos avisos para indicar al usuario si sobran vehículos para el último viaje, y así para ahorrar el consumo de combustible.

Al final de la cuarta pantalla, habrá un botón **"VOLVER A LA PRIMERA PAGINA"**, que al pulsarlo, nos permite volver a la primera pantalla para introducir los nuevos datos y calcular todas las herramientas y el plan de carga. Y también hay otro botón **"CERRAR EL PROGRAMA"**, que al pulsarlo, nos permite salir del programa.

5. Descargar AutoPuente.

La aplicación AutoPuente se puede descargar en los móviles que tienen el sistema Android. Para los que no lo tienen, puedes usar la aplicación AutoPuente a través de la línea. A continuación se va a explicar cada uno de ellos.

5.1. AutoPuente para el sistema Android.

Las aplicaciones de Andorid se pueden descargar a través de PlayStore si el desarrollador paga el mantenimiento para dicho sitio. AutoPuente al ser una aplicación con el fin de estar usada solo en el regimiento de pontoneros número 12, no se pagó el mantenimiento de la aplicación en PlayStore para que nos salga más rentable. Para descargar la aplicación por favor sigue los pasos siguientes:

1. Desde ajustes, entre a seguridad. Luego active la opción de descargar **aplicaciones de fuentes no fiables (Orígenes desconocidos)**.

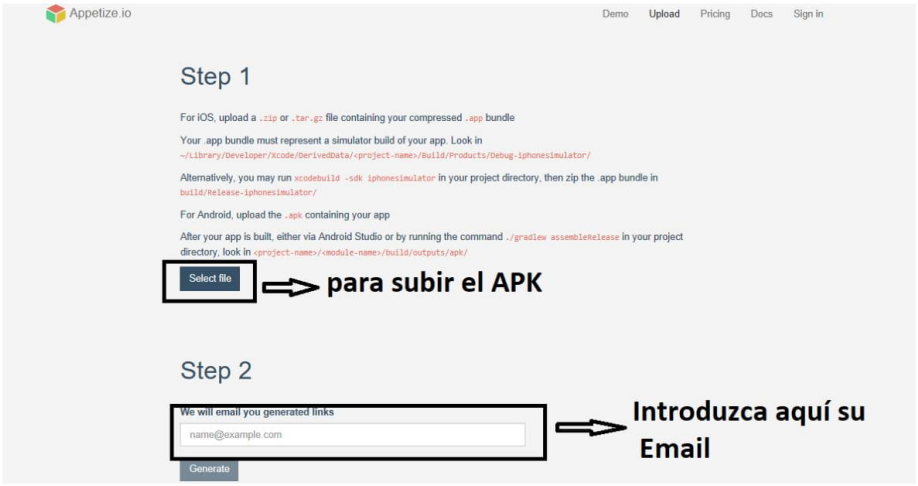
2. Desde ajustes, entre a los datos del móvil (Sobre el móvil/ A cerca del móvil). Luego pulse la opción **Número de compilación (Build number)** de la información del software unos 5-6 veces hasta que le salga un mensaje diciendo **(Ya eres el desarrollador de la aplicación)**.

3. Abra el **APK** desde su email y descarga la aplicación.
4. Ya puede disfrutar de usar la aplicación.

5.2. AutoPuente a través de la línea.

Los sistemas de IOS, Windows y otros no tienen la posibilidad de descargar aplicaciones de Android, por lo cual solo pueden disfrutar de usar AutoPuente a través de la line. Para dicho objetivo, por favor sigue los pasos siguientes:

1. Abra la página ***Appetize*** a través del siguiente link ***https://appetize.io/upload.***
2. Suba el archivo **APK** que tiene guardado en su PC.
3. Introduzca su email, y en seguida recibirá usted un correo de la página web con el link donde puede usar AutoPuente.



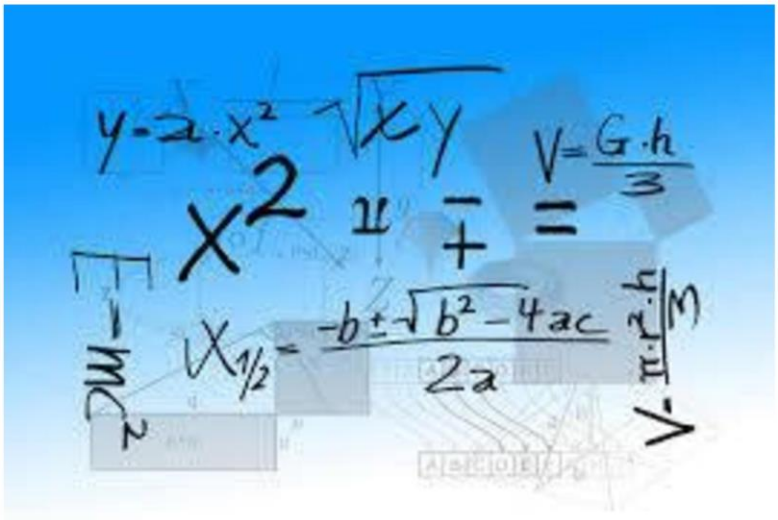
4. En este caso, solo puede disfrutar de usar AutoPuente 100 minutos de cada email que usas. Por lo tanto, cuando se le caduquen los 100 minutos, vuelva a hacer los mismos pasos pero introduzca otro email.

Para futuros trabajos de mejoras a la aplicación AutoPuente, pueden conectarse con el autor a través del email ahmadalqadi70@gmail.com .

Anexo 4: Cálculos utilizados en AutoPuente.

CÁLCULOS UTILIZADOS EN AUTOPUENTE

Este documento representa todos los cálculos que fueron necesarios para la programación de la aplicación informática AutoPuente



2 DE OCTUBRE DE 2018

AHMAD MOHAMMAD ALQADI
ahmadalqadi70@gmail.com

Índice

En este documento se van a explicar los cálculos que se tuvieron en cuenta para la creación de la aplicación AutoPuente. Los cálculos se hicieron por parte del autor usando manuales militares y ejemplos de montaje de puentes que se hicieron por parte del personal de compañía de puentes flotantes.

A continuación se procederá primero a explicar los cálculos que tienen en cuenta el tipo y la clase de puente usado, luego se explicará los otros cálculos:

1. Cálculos de material.

1.1. Cálculos dependientes del tipo y la clase de puente.

1.1.1. Pasarelas.

1.1.1.1. Tipo 1.

Pontones extremos = 0

Pontones centrales = Redondear (Luz / 4.2)

Placas deflectoras = pontones centrales

Buques de alba = 2

Cumbreras = 2

Vigas de rampa = 10

Postecillos = 2

Gatos = 2

Cable = 2

Bolardos = 0

1.1.1.2. Tipo 2.

Pontones extremos = Redondear (Luz / 6.3)

Pontones centrales = Pontones extremos

Placas deflectoras = pontones extremos + Pontones centrales

Buques de alba = 2

Cumbreras = 2

Vigas de rampa = 10

Postecillos = 2

Gatos = 2

Cable = 2

Bolardos = 0

1.1.1.3. Tipo 3.

Pontones extremos = Redondear (Luz / 4.2)

Pontones centrales = pontones extremos – 1

Placas deflectoras = pontones extremos + Pontones centrales

Buques de alba = 2

Cumbreras = 2

Vigas de rampa = 10

Postecillos = 2

Gatos = 2

Cable = 2

Bolardos = 0

1.1.1.4. Tipo 4.

Pontones extremos = Redondear ((Luz / 6.3) * 2)

Pontones centrales = Redondear ((Pontones extremos / 2) – 1)

Placas deflectoras = pontones extremos + Pontones centrales

Buques de alba = 2

Cumbreras = 2

Vigas de rampa = 10

Postecillos = 2

Gatos = 2

Cable = 2

Bolardos = 0

1.1.2. Compuerta.

1.1.2.1. Clase MLC20.

Pontones extremos = Redondear (2 + ((Luz – 4.2) / 2.1))

Pontones centrales = Redondear (2 + ((((Luz – 4.2) / (2.1)) – 1) / 2))

Placas deflectoras = pontones extremos

Buques de alba = 4

Cumbreras = 2

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear (((Luz – 4.2) / 14.7) – 1)

Anchura = 3.5m

1.1.2.2. Clase MLC50T1.

Pontones extremos = Redondear (6 + ((((Luz – 8.4) / 2.1)/5) + ((Luz -8.4) / 2.1)))

Pontones centrales = Redondear (4 + (((Luz – 8.4) / 2.1) – (((Luz – 8.4) / 2.1) / 5)))

Placas deflectoras = Redondear (Luz / 2.1)

Buques de alba = 4

Cumbreras = 4

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear (((Luz – 8.4) / 10.5) – 1)

Anchura = 6.3m

1.1.2.3. Clase MLC50T2.

Pontones extremos = Redondear ((Luz / 2.1) * 2)

Pontones centrales = Redondear ((Luz / 2.1) + ((Luz / 2.1) / 4))

Placas deflectoras = Redondear (Luz / 2.1)

Buques de alba = 4

Cumbreras = 4

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear ((Luz / 12.6) – 1)

Anchura = 8.1m

1.1.2.4. Clase MLC80.

Pontones extremos = Redondear (8 + (((Luz – 8.4) / 2.1) * 2))

Pontones centrales = Redondear (6 + ((Luz – 8.4) / 2.1))

Placas deflectoras = Redondear (Luz / 2.1)

4

Buques de alba = 4

Cumbreras = 4

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear (((Luz – 8.4) / 16.8) – 1)

Anchura = 8.1m

1.1.3. Puentes.

1.1.3.1. Clase MLC30.

Pontones extremos = Redondear (Luz / 2.1)

Pontones centrales = Redondear (Pontones extremos / 2)

Placas deflectoras = Redondear (Luz / 2.1)

Buques de alba = 4

Cumbreras = 4

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear ((Luz / 21) – 1)

Anchura = 3.5m

1.1.3.2. Clase MLC50.

Pontones extremos = Redondear ((Luz / 2.1) + ((Luz / 2.1) / 5))

Pontones centrales = Redondear ((Luz / 2.1) - ((Luz / 2.1) / 5))

Placas deflectoras = Redondear (Luz / 2.1)

Buques de alba = 4

Cumbreras = 4

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear ((Luz / 10.5) – 1)

Anchura = 6.3m

1.1.3.3. Clase MLC80A.

Pontones extremos = Redondear ((Luz / 2.1) * 2)

Pontones centrales = Redondear (Luz / 2.1)

Placas deflectoras = Redondear (Luz / 2.1)

Buques de alba = 4

Cumbreras = 4

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear ((Luz / 12.6) – 1)

Anchura = 8.1m

1.1.3.4. Clase MLC80B.

Pontones extremos = Redondear (Luz / 2.1)

Pontones centrales = Redondear ((Luz / 2.1) * 2)

Placas deflectoras = Redondear (Luz / 2.1)

Buques de alba = 4

Cumbreras = 4

Vigas de rampa = 24

Postecillos = 4

Gatos = 4

Cable = 4

Bolardos = Redondear ((Luz / 12.6) – 1)

Anchura = 8.1m

1.2. Cálculos independientes del tipo y la clase de puente.

Estribos = 10 * (Pontones extremos + Pontones centrales)

Horquillas = 11 * (Pontones extremos + Pontones centrales)

Cabestrantes = Redondear ((Luz / 10.5) * 2.3)

Traveseros = Placas deflectoras

Soportes cabestrantes = Cabestrantes

Rodillos = Cabestrantes

1.3. Empaquetar las herramientas en jaulas.

Jaulas estribos = Estribos / 100

Jaulas horquillas = Horquillas / 110

Jaulas Cabestrantes = Cabestrantes / 8

Jaulas Placas deflectoras = Placas deflectoras / 7

Jaulas Rodillos = Rodillos / 13

Si la jaula de Horquillas tiene 34 Horquillas o menos, se llevan sueltas sin jaula

Si la jaula de cabestrantes tiene 4 cabestrantes o menos, se llevan sueltas sin jaula

Si la jaula de rodillos tiene 6 rodillos o menos, se llevan sueltas sin jaula

Si las vigas = 10, se lleva una jaula grande para la rampa

Si las vigas = 24, se llevan dos jaulas grandes para la viga

Siempre se lleva una jaula de Bordillos.

Siempre se lleva una caja de herramientas que tendrá:

CAJA HERRAMIENTAS
2 martillos
6 martillos fibra
8 barras de pincho
4 barras de acoplamiento
2 barras de nivelación
6 llaves del 32
4 escalerillas
Amarras
4 cáncamos de cabestrante
4 cadenas
20 piquetes

2. Plan de carga.

Con las consultas al personal de la compañía de puentes flotantes, se pudo llegar a tener todos los tipos de vehículos que usan para el plan de carga con las capacidades de cada uno de ellos.

A continuación se va a mostrar una tabla de los tipos de vehículos con sus correspondientes capacidades:

VEHÍCULOS EMBARCACIÓN	CAPACIDAD EN CADA VIAJE
MERLO	-
SCHOTTEL	-
CAMIÓN M250	(2 Pontones) **O** (2 Jaulas) **O** (1 Jaula Bordillos)
REMOLQUES GONTRAILER/M250	Merlo **O** (6 pontones) O Jaulas grandes
CAMIÓN 7217	1 Jaula +Caja herramientas/cosas sueltas
CAMIÓN VEMPAR	(3 Pontones) **O** (2 Jaulas) **O** (1 Jaula Bordillos)
REMOLQUES GONTRAILER/VEMPAR	3 Pontones O Jaulas grandes
Grúa Luna	-

El usuario al elegir el número de cada vehículo que quiere usar para el transporte del material, el programa se accede a una tabla para saber el consumo de cada vehículo durante el viaje, y al meter la distancia del viaje, el programa calcula el consumo total de combustible teniendo en cuenta los vehículos que sobran en el último viaje. A continuación se va a mostrar la tabla de los consumos de los vehículos cada 100KM:

Modelo	Consumo a los 100 KM
Vehículo Ligero Logístico(tipo furgoneta/representación):	13
Vehículo Ligero TT. (tipo NISSAN PATROL O SIMILAR):	15
Vehículo Ligero TT. (tipo AMBUANCIA IVECO O SIMILAR):	16
Vehículo Ligero TT. (tipo NISSAN PATROL BLINDADO):	20
Vehículo Ligero TT. (tipo URO VAMTAC):	20
Vehículo Ligero TT. (tipo URO 3TM):	30
Vehículo Medio Logístico(tipo CN NISSAN M-110):	23
Vehículo Medio TT3 Tm(tipo CNTT PEGASO 7217; 7226 O SIMILAR):	35
Vehículo Pesado TT5 Tm(tipo CNTT PEGASO 7323 O SIMILAR):	46
Vehículo Pesado TT10 Tm(tipo CNTT IVECO M-250):	55
Vehículo Pesado Logístico(tipo CN VEMPAR 10 Tm):	55
Vehículo Pesado Logístico(tipo CN VEMPAR 15 Tm):	60
Vehículo Pesado Logístico(tipo CN VEMPAR 32 Tm):	60
Cabezas Tractoras:	66
Microbús 15-30 plazas:	25
Microbús 30-60 plazas:	35
Grúa Luna AT 2020; AT 35/32 O SIMILAR:	60

Por lo cual, al elegir el tipo de vehículo, la aplicación hace unos ciclos de viajes invisibles para el usuario para saber el número total de viajes (Idas y Vueltas) y el número de vehículos que sobran en el último viaje con sus correspondientes tipos.

El consumo total = (((vehículo 1 * consumo de vehículo 1) + (vehículo 2 * consumo de vehículo 2) + (vehículo X * consumo de vehículo X)) * número de viajes en total * (Distancia del viaje / 100)) – (((vehículo sobrante 1 * consumo de vehículo sobrante 1) + (vehículo sobrante X *

consumo de vehículo sobrante X)) * 2 * (Distancia del viaje / 100)) + (Grúa Luna * 60 * (Distancia del viaje / 100)

Ejemplo:

Se va a transportar un puente tipo puente, clase MLC50 y luz = 115,5m. Los vehículos disponibles son 2 M250 con sus gontrailers y un vehículo 7217. También se sabe que se necesitaría una Merlo y dos Grúas Luna para la construcción. La distancia del viaje es de 13KM.

El programa al hacer los ciclos invisibles, se pudo llegar a que el número de viajes en total son 15 viajes, y en el último viaje sobra el vehículo 7217. ¿Calcula el consumo total de combustible?

El consumo total = (((2 * 55) + (1 * 35)) * 15 * (13 / 100)) – ((1 * 35) * 2 * (13 / 100)) + (2 Grúas luna * 60 * (13/100)) = 289,25L.

Anexo 5: Cuestionario de satisfacción.

Centro Universitario de la Defensa Zaragoza

Cuestionario AutoPuente
2018/2019

CUESTIONARIO

En este cuestionario se va a proceder a evaluar la nueva aplicación que se ha creado para la compañía de puentes flotantes en el regimiento de pontoneros y especialidad de ingenieros número 12 en Zaragoza. **Este cuestionario es totalmente anónimo** por ello puede responder las preguntas con total libertad. Así se van a evaluar X aspectos de la aplicación en una escala de 1 al 5, siendo **1 muy deficiente y 5 excelente**.

Las respuestas están orientadas a la creación de un estudio de calidad para el trabajo fin de grado con título Desarrollo de Software para la determinación de los planes de carga PTF-MAN.

Marque con una X la puntuación que considere más acorde con el aspecto a evaluar.

Creación de datos					
	1	2	3	4	5
Tipos y clases de puente					
Los mensajes de error					
La manera de meter los datos					
Los vehículos					
Comentarios					

Diseño de la interfaz					
	1	2	3	4	5
Tabla del material general					
Tablas de la rampa y de las herramientas					
Tabla de los vehículos disponibles en la unidad					
Los botones de continuar y cerrar el programa					
Los avisos para el usuario					
Comentarios					

Funcionamiento						
¿Es/son correcto/s…?	1	2	3	4	5
Los datos de las tablas del material					
El tipo de vehículos disponibles					
El número de viajes en total					
El consumo de combustible total					
Los avisos para el usuario					
Comentarios					

Manual					
	1	2	3	4	5
Comprensión					
Vocabulario					
Imágenes aclaratorias					
Formato					
Forma de descargar la aplicación					
Comentarios					

Valoración final de la aplicación					
¿La aplicación…?	1	2	3	4	5
Ayuda en las tareas de la CIA					
Devuelve resultados que concuerdan con la realidad					
Reduce el tiempo de la fase de cálculo de piezas					
Es útil					
Comentarios					