

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Carrera

**DESARROLLO DE LA APLICACIÓN ZIRA
PARA LA RED FERROVIARIA DE ANCHO
MÉTRICO ESPAÑOLA**

**(Development of the ZIRA application for the
Spanish Metric Railway Network)**

Para acceder al Título de

INGENIERO TÉCNICO DE TELECOMUNICACIÓN

Autor: Juan Ramón Cano Lavín

Octubre - 2016



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN ESPECIALIDAD SISTEMAS ELECTRONICOS

CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

Realizado por: Juan Ramón Cano Lavín

Director del PFC: Tomás Fernández Ibáñez

Título: “Desarrollo de la aplicación ZIRA para la red ferroviaria de ancho métrico española”

Title: “Development of the ZIRA appication for the spanish metric railway network“

Presentado a examen el día:

para acceder al Título de

INGENIERO TÉCNICO DE TELECOMUNICACIÓN ESPECIALIDAD SISTEMAS ELECTRONICOS

Composición del Tribunal:

Presidente (Apellidos, Nombre): Alicia Casanueva López

Secretario (Apellidos, Nombre): Tomás Fernández Ibáñez

Vocal (Apellidos, Nombre): José Angel García García

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del PFC
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Proyecto Fin de Carrera N°
(a asignar por Secretaría)

*Dedicado a la memoria de mi padre,
que me enseñó que todo se consigue
a base de esfuerzo y sacrificio.
Y también a mi madre y a mi hermano,
por ser los dos pilares de mi vida,
y a mis tres sobrinos, IKER, DAVID y AINARA,
por la alegría y cariño que nos dan.*

INDICE:

1. INTRODUCCION Y OBJETIVOS	1
2. CONCEPTOS BASICOS FERROVIARIOS.....	4
2.1 LA HOJA DE MARCHA	4
2.2 EL BOLETIN DE FRENADO	5
3. PLANTEAMIENTO GENERAL.....	9
4. DESARROLLO DEL CODIGO	15
4.1 LA CLASE ESTACION.....	16
4.2 LA CLASE LOCOMOTORA	17
4.3 LA CLASE VAGON	19
4.4 LA CLASE CANTON.....	21
4.5 LA CLASE MARCHA.....	23
4.6 LA CLASE ZIRA	24
4.7 LA CLASE HOJMARCHA	33
4.8 LA CLASE BOLETINFRENADO	36
4.9 LA CLASE UTILPDF	41
4.10 LA CLASE CREACIONPDF.....	42
4.11 LA CLASE TABLAMODELO	48
4.12 LA CLASE VENTANA	49
5. CONCLUSIONES Y CAMBIOS FUTUROS	68
ANEXO I: CONSIGNA SERIE C N° 13/2016	69
ANEXO II: CONSIGNA SERIE C N° 6/2011.....	73
ANEXO III: CACTERÍSTICAS DE FRENADO DEL MATERIAL MOTOR Y REMOLCADO	78
ANEXO IV: PORCENTAJES DE FRENADO AUTOMÁTICO Y DE ESTACIONAMIENTO.....	79
ANEXO V: MARCHAS DE ASIMILACIÓN	82
ANEXO VI: ORDENES DEL DIA DE LOS TRENES GRAFIADOS.....	134
ANEXO VII: VELOCIDADES, NUMERO DE EJES Y CARGAS	142

1. INTRODUCCIÓN Y OBJETIVOS

Los orígenes de FEVE están en el organismo Explotación de Ferrocarriles por el Estado (EFE), que había sido creado en 1926 para hacerse cargo de la gestión de las líneas ferroviarias de construcción estatal, o de aquellas que hubieran pasado a gestión estatal. Tras la creación de Red Nacional de los Ferrocarriles Españoles (RENFE) en 1941 el ámbito de actuación de EFE se redujo enormemente, ya que se vio limitado a los ferrocarriles de vía estrecha. La situación cambió radicalmente a partir de 1960, cuando un gran número de pequeños ferrocarriles mineros quebraron y el Estado salió al rescate de los mismos. Debido a la gran cantidad de líneas que el estado se vio obligado a gestionar en tan poco tiempo, para agilizar dicha administración se formalizó la creación el 23 de septiembre de 1965 de un nuevo organismo autónomo; ese mismo día EFE cambió su nombre, adoptando el de Ferrocarriles Españoles de Vía Estrecha (FEVE).

La red de FEVE estaba formada por las líneas de antiguas compañías como la Sociedad General de Ferrocarriles Vasco Asturiana, Ferrocarriles Económicos de Asturias, Ferrocarril de Langreo, Ferrocarril de Carreño, Ferrocarril Cantábrico, Ferrocarril Santander-Bilbao, Ferrocarril de La Robla, Ferrocarriles de Mallorca, Ferrocarril Cartagena-Los Blancos, además del Ferrocarril Ferrol - Gijón, construido íntegramente por el Estado. Sin embargo, muchas de estos ferrocarriles se encontraban en una situación muy crítica, con el material ferroviario y/o las infraestructuras muy necesitadas de mejoras. Por esta razón hubo otras líneas de vía estrecha, como las heredadas a través de la Compañía de Ferrocarriles Secundarios de Castilla, que fueron clausuradas al poco tiempo al ser consideradas deficitarias.

La línea de mayor longitud que poseía la compañía tiene 330 kilómetros, es el conocido como Ferrocarril de La Robla. Originalmente iba desde La Robla (León) hasta Valmaseda y se construyó para hacer llegar los carbones de la minería leonesa y palentina hasta la siderurgia vizcaína. A principios del siglo XX se construyeron los tramos Valmaseda-Luchana y el acceso a Bilbao (aprovechando el trazado del Ferrocarril Santander-Bilbao) y Matallana-León, con lo que de esta manera quedaban comunicadas las capitales de León y Bilbao. Actualmente los tramos entre Matallana y La Robla e Iráuregui-Luchana no prestan servicio de viajeros. Es la más larga de Europa de vía métrica, recorriendo la montaña oriental leonesa, el norte de la provincia de Palencia, el sur de Cantabria, el norte de la provincia de Burgos y Vizcaya.

Tras el final de la Dictadura franquista, a partir de 1979 una parte significativa de la red de vía estrecha fue transferida por el gobierno central a las nuevas comunidades autónomas, las cuales la pasaron a gestionar a través de organismos públicos:

- * EuskoTren: creado por el Gobierno vasco en 1982, pasó a operar varias líneas en las provincias de Guipúzcoa y Vizcaya.

- * Ferrocarriles de la Generalidad de Cataluña (FGC): creada en 1979, pasó a operar varias líneas regionales y/o cercanías en la provincia de Barcelona.

* Ferrocarriles de la Generalidad Valenciana (FGV): creada en 1986 pasó a operar varias líneas regionales y/o cercanías de la Comunidad Valenciana.

* Servicios Ferroviarios de Mallorca (SFM): creados en 1994, operan varias líneas de la isla de Mallorca.

FEVE entró en una fase de especialización de los servicios ofrecidos al público, pasándose a especializar en transporte de mercancías y regional y de cercanías de pasajeros. Por su parte, el tren turístico de lujo Transcantábrico, que comenzó sus servicios en 1982, se convirtió en el servicio estrella de la compañía. Realiza el trayecto Santiago de Compostela-Ferrol-Bilbao-León (el tramo Santiago-Ferrol se realiza por autobús).

Además, el Ministerio de fomento realizó numerosas inversiones para mejorar tanto las infraestructuras como el material rodante. Tras las cesiones realizadas a las autonomías, la red ferroviaria de FEVE quedó centrada principalmente en la zona cantábrica. La principal línea de FEVE en la Cornisa Cantábrica era la que conectaba Ferrol con Bilbao a través de las provincias de Vizcaya, Cantabria, Asturias y Provincia de Lugo. Así mismo enlazan con esta línea general diversas líneas menores y ramales, donde prestan servicio como norma general trenes de cercanías. Estas líneas y ramales eran las siguientes:

- Tramo **FERROL - PRAVIA - GIJON**, que recorre las provincias de Coruña, Lugo y Asturias.
- Tramo **SAN ESTEBAN DE PRAVIA - PRAVIA - TRUBIA - OVIEDO**, en Asturias.
- Tramo **OVIEDO - TRUBIA - COLLANZO**, en Asturias.
- Tramo **GIJON - POLA DE LAVIANA**, en Asturias.
- Tramo **OVIEDO - SANTANDER**, por Asturias y Cantabria.
- Tramo **SANTANDER - ARANGUREN - BILBAO y sus ramales**, por Cantabria y Vizcaya.
- Tramo **BILBAO - ARANGUREN - LEON y sus ramales**, que recorre las provincias de Vizcaya, Burgos, Cantabria, Palencia y León.
- Tramo de **CARTAGENA - LOS NIETOS**, en tierras de la provincia de Murcia.

En 2005 FEVE, ante el nuevo escenario ferroviario español protagonizado por la liberalización y la apertura a la competencia (de la que FEVE queda exenta inicialmente), el ente público emprende un plan estratégico que marca las pautas para la empresa a medio plazo para iniciar el proceso de liberación ferroviario que obliga la Unión Europea. El plan hace hincapié en la optimización de los recursos humanos y en la especialización por sectores, por lo que se crean las direcciones-gerencias (similares a las Unidades de Negocio de Renfe Operadora), que se especializarán en sus correspondientes mercados. El

plan estratégico tiene como fin aumentar el índice de cobertura de la empresa 12 puntos hasta situarlo en el 45%, destacando la búsqueda de ingresos mediante la captación de tráficos de tipo turístico y de nuevos clientes en mercancías, así como su consolidación en viajeros de trenes de cercanías y regionales.

El 20 de julio de 2012, se aprobó el Real Decreto-ley 22/2012, de 20 de julio, por el cual FEVE quedó extinguida el día 31 de diciembre de 2012 y se subrogó a Administrador de Infraestructuras Ferroviarias Adif la infraestructura y a Renfe Operadora la explotación de los trenes. De esta forma se unifican las redes nacionales de vía ancha y vía estrecha. En la práctica Renfe Operadora sigue usando la marca "Feve" para referirse a todos los servicios ferroviarios que heredó de la anterior compañía.

Como resultado de esta subrogación de FEVE en Adif y Renfe, empieza un proceso de integración en ambas compañías que implican numerosos cambios para adaptarse a la liberalización del sector ferroviario, una de ellas es que hasta ahora eran los jefes de circulación los responsables de elaborar la hoja de marcha y de calcular el boletín de frenado de los trenes de mercancías y de los trenes de viajeros compuestos por locomotoras, coches y furgones, sin embargo, los factores de circulación, integrados todos en Adif, dejarán de realizar ambos documentos, siendo las distintas operadoras quienes tengan que elaborar dicha documentación, hecho que motiva la realización de este proyecto fin de carrera de desarrollar una aplicación que genere ambos documentos sin la necesidad de tener los conocimientos necesarios del reglamento de circulación de trenes y las consignas de la extinta FEVE que regulan las composiciones de los trenes según el trayecto que realizan.

Ambos documentos se van a implementar siguiendo el Reglamento de Circulación de Trenes de FEVE todavía vigente a la espera de la futura implantación del nuevo reglamento conjunto con ADIF que previsiblemente tendrá lugar el año que viene.

2. CONCEPTOS BÁSICOS FERROVIARIOS

2.1 LA HOJA DE MARCHA:

Para elaborar la hoja de marcha de cualquier tren de mercancías o de viajeros que está compuesto por locomotoras, vagones y furgones vamos a describir las dos partes que va a elaborar la aplicación.

feve		Hoja de Marcha		Naturaleza <u>MECANICA</u> Tren Nº <u>22802</u> Tipo <u>50</u>		Origen: <u>ANIZ</u> <u>5713</u> Fecha <u>30/02/36</u>									
		Asimil. a: <u>73764</u> Entre <u>ANIZ</u> y <u>ARAGONEA</u>		Destino: <u>ARAGONEA</u> <u>5473</u> Fecha <u>30/02/36</u>											
Marcha efectiva				Composición inicial				Variación de la composición							
Código Estación	Horas Llegada	Par	Salida	Motivo	Justificación	Vehículos tractores Nº 3652 Nº 3655		Código Estación	Vehículo Serie	Nº	Motivo variación	Carga remolcada			
						Mat. remolcada	Carga remolcada					Mercancía	Peso total	Acumulado	
5713	x		21:50			2º 389	ARAGONEA								
5473						" 354	"								
						" 246	"								
						" 272	"								
						2º 368	"								
						5º 9088	BOBIZAS								
						" 9337	"								
						" 9040	"								
						" 2030	"								
						5º 2032	BOBIZAS								
						Total 515									
						Agregado (1) Diferido (2) Diferido avería (3)									
						Observaciones:		No señalado a la vía							
Personal del tren						Jefe de circulación de salida				Jefe de circulación de llegada					
Nº de C.F.		Nombre y Apellidos		de		a									
9750		Isidro Beredo Llaguno		Ariz		Aranguren		 2997621							

Hoja de Marcha

En primer lugar tenemos el encabezamiento que hace referencia al maquinista del trayecto que debe realizar y la velocidad a la que puede circular. Se compone de los siguientes elementos:

- Naturaleza del tren: hace referencia al tipo de tren y según un artículo del reglamento de circulación de trenes de FEVE recientemente modificado se clasificaban en:
 1. Cercanías
 2. Regionales
 3. Transcantábrico
 4. Turísticos
 5. Servicio y locomotoras aisladas
 6. No clasificados (Adicionales, facultativos, etc)

7. Mantenimiento e infraestructura
8. Socorro y Talleres
9. Mercancías

- Nombre del tren: hace referencia a la denominación del tren, puede ser un tren grafiado cuya marcha viene determinada por el libro de itinerarios, o bien, puede ser un tren especial sin marcha determinada. Para la designación de los trenes sin marcha determinada queda determinada por cinco caracteres:
 1. El primer y segundo caracteres corresponden a las dos letras asignadas a la estación de origen en la consigna serie C N°13/2016 que figuran en el anexo I.
 2. El tercer carácter, que identifica a los trenes de la red de ancho métrico, será un 8.
 3. El cuarto y quinto caracteres corresponden conjuntamente al número de orden y paridad del tren.
- Asimilado a: para los trenes especiales sin marcha determinada sirve para asignarle una marcha, ya sea una determinada de un tren grafiado o bien una marcha de asimilación tipo.
- Origen: hace referencia a la estación de origen del trayecto que el maquinista debe recorrer con el tren, con su código y su denominación.
- Destino: hace referencia a la estación de destino del trayecto que el maquinista debe recorrer con el tren, con su código y su denominación.

A continuación haremos referencia a la composición del tren en el que el usuario deberá introducir las locomotoras y los vagones que van a formar parte del tren del que se va a calcular posteriormente el boletín de frenado. Las locomotoras quedaran descritas con su número corto y los vagones quedaran descritos por su serie, su número, la mercancía que lleva y la carga total del vagón. El peso total de cada vehículo se obtendrá sumando la tara más la carga real. La tara y la carga se redondearan por separado, despreciando las fracciones inferiores a 500 kg y estimando como UNA tonelada las que igualen o superen esa cifra.

2.2 EL BOLETIN DE FRENADO:

El boletín de frenado se calcula a partir de la composición del tren que hemos descrito en la hoja de marcha y debe verificar que el peso total de la composición no supera la carga máxima según la Consigna Serie C N° 6/2011, que figura en el Anexo I, y los porcentajes de frenado obtenidos cumplan con los especificados en el anexo de frenado del Reglamento de Circulación de Trenes para el trayecto que va a recorrer el tren.

Para todos los cálculos que se van a realizar en el boletín de frenado no habrá de tenerse en cuenta a la locomotora o locomotoras que formen parte de la composición como unidades tractoras; se incluirá en el cálculo a las locomotoras cuando formen parte de la composición como unidades remolcadas.

FEVE		Boletín de Frenado												Tren n.º: <u>A2802</u>			
														Fecha: <u>30 02 16</u>			
Estaciones	N.º vehículos	Clase freno	Peso total remolcado	Freno automático					Freno estacionamiento					N.º ejes de la composición			
				Peso freno			Porcentaje		Peso freno			Porcentaje		Sin freno	Régimen G	Régimen P	Total
				Vehículos		Total	Disponible	Necesario	Vehículos		Total	Disponible	Necesario				
				Motora	Remolcado				Motora	Remolcado							
05713 ARIZ	30	A	515	-	440	440	76	50	-	375	375	30	32	-	-	40	40
Clase de prueba	Agente de cola		Agente de cabeza		Maquinista de cabeza		Jefe de circulación										
COMPLETA	9534 <i>[Firma]</i>						2997021										
Observaciones:																	

Boletín de Frenado

EL NÚMERO DE VEHÍCULOS de un tren es el número de vehículos remolcados que componen dicho tren, sin contabilizar las locomotoras siempre y cuando circulen traccionando.

La CLASE DE FRENO siempre va a ser “A” haciendo referencia a que es por aire comprimido ya que en la actualidad no circula ningún vehículo con freno de vacío.

EL PESO TOTAL del tren se obtendrá sumando los pesos totales de todos los vehículos remolcados. Este peso deberá ser inferior a la carga máxima que la o las locomotoras pueden remolcar para el trayecto que debe recorrer el tren y que vienen determinadas por la consigna serie C N°6/2011 que se recoge en el anexo II.

EL PORCENTAJE DE FRENADO AUTOMÁTICO de un vehículo o de un tren es una magnitud de capital importancia para la explotación ferroviaria, ya que existe una relación estrecha entre la velocidad de circulación de un tren, su porcentaje de frenado automático y su distancia de parada. Esta relación estrecha se manifiesta, para cada tramo concreto de la red, en lo siguiente:

- un tren, al que se autorice una determinada velocidad máxima, precisará (como mínimo) un PORCENTAJE DE FRENADO AUTOMÁTICO también determinado, para detener su marcha sin rebasar la distancia establecida.
- un tren, que tenga un determinado PORCENTAJE DE FRENADO AUTOMÁTICO, no podrá rebasar una velocidad máxima también determinada, para tener garantías de detener su marcha sin rebasar la distancia establecida.

A continuación vamos a describir como se calculan las distintas casillas correspondientes al freno automático:

- **Motores**: es la suma de los pesos automáticos de las locomotoras que van remolcadas en la composición. Si no hubiera ponemos cero.
- **Remolcado**: se obtiene sumando los PESOS-FRENO automático de todos y cada uno de los vehículos remolcados que lleven el freno en servicio, de acuerdo con el estado de sus cambiadores de potencia y con los valores del freno inscritos en sus marcas.
- **Total**: es la suma de los dos valores anteriores, motores y remolcado.
- **Disponible**: es la relación (expresada en tanto por ciento) del cociente entre el freno automático total y el peso total del tren.
- **Necesario**: es el porcentaje de freno automático que requiere el tren según el tipo y el trayecto que va a recorrer.

El FRENO DE ESTACIONAMIENTO de un vehículo o de un tren se utiliza como freno de reserva para mantener detenido el material, aún en el caso de que se llegue a perder el frenado automático. A continuación vamos a describir como se calculan las distintas casillas correspondientes al freno automático:

- **Motores**: es la suma de los pesos de estacionamiento de las locomotoras que van remolcadas en la composición. Si no hubiera ponemos cero.
- **Remolcado**: se obtiene sumando los PESOS-FRENO de estacionamiento de todos y cada uno de los vehículos remolcados que lleven el freno de estacionamiento en servicio y se obtendrá multiplicando por 0,5 el valor del “Freno mano máx.” inscrito en las marcas, salvo que por consigna se indique lo contrario como sucede para los vagones de las series Sghmmns y Sgh.
- **Total**: es la suma de los dos valores anteriores, motores y remolcado.
- **Disponible**: es la relación (expresada en tanto por ciento) del cociente entre el freno de estacionamiento total y el peso total del tren.
- **Necesario**: es el porcentaje de freno de estacionamiento que requiere el tren según el tipo y el trayecto que va a recorrer.

El NÚMERO DE EJES DE LA COMPOSICIÓN desglosa como circulan los vagones de los que se compone el tren, sabiendo que cada vagón tiene 4 ejes, se calculará cada apartado como se indica a continuación:

- **Sin freno**: indica el número de ejes que circulan aislados de freno, su valor se obtiene multiplicando por 4 el número de vagones aislados.
- **Régimen G**: indica el número de ejes que circulan en régimen de mercancías. Dado que está prohibido salvo autorización especial que los vagones circulen en distintos regímenes de frenado, este valor va a ser siempre cero.
- **Régimen P**: indica el número de ejes que circulan en régimen de viajeros, su valor se obtiene multiplicando por 4 el número de vagones con freno.
- **Total**: indica el número total de ejes de la composición del tren, su valor se obtiene multiplicando por 4 el número de vagones totales, y deberá ser igual a la suma de ejes en régimen P y los ejes sin freno.

En el Anexo III se expresan los valores de peso freno automático y del freno de estacionamiento de las unidades y vehículos que componen el parque habitualmente utilizado por FEVE.

En el anexo IV figuran los PORCENTAJES DE FRENADO, que requieren los diferentes tipos de tren (clasificación según artículo 4.01.00 del Reglamento de Circulación de Trenes) para poder circular por cada uno de los trayectos de la red. Los indicados PORCENTAJES DE FRENADO son de obligatorio e inexcusable cumplimiento. Si por algún motivo no se cumplieran dichos porcentajes habría que modificar la composición del tren eliminando vagones o reduciendo la velocidad del tren.

En el anexo VII figuran las velociades, el número de ejes según el tipo de tren y la carga máxima que puede transportar cada vagón. Todos estos requisitos son de obligado cumplimiento.

3.0 PLANTEAMIENTO GENERAL

Para el desarrollo de la aplicación ZIRA se ha de partir de un planteamiento basado en los objetivos a cumplir. La aplicación debe recoger unos datos de entrada relacionados con el trayecto del tren y su composición y elaborar la hoja de marcha que describan dichos datos de entrada y calcular el boletín de frenado de la composición y verificar que cumple dos requisitos fundamentales: que no supera la carga máxima remolcada del trayecto y que los porcentajes de frenado son superiores a los mínimos requeridos según la velocidad y el trayecto del tren. Para ello debemos de partir de una serie de datos obtenidos a partir del Reglamento de Circulación de Trenes y de las distintas consignas que regulan los distintos aspectos relacionados con el frenado de los trenes.

En primer lugar vamos a partir de la Consigna serie C N° 13/2016 (anexo I) para crear el archivo **ESTACIONES.csv** en el que se ha creado una base de datos que contiene todas las estaciones clasificadas como tal con su correspondiente abreviatura y su código comercial.

Campo [0] = código de la estación

Campo [1] = nombre de la estación

Campo [2] = abreviatura de la estación

A continuación a partir de la Consigna serie C N°6/2011 (anexo II) que regula las cargas máximas que pueden remolcar las locomotoras en las distintas líneas y sus ramales se ha creado el archivo **CANTONES.csv** en las que cada tramo se ha codificado según la línea y su paridad correlativamente. Para la línea 1 que une las estaciones de Ferrol y Oviedo se ha implementado la siguiente tabla que describe los distintos cantones:

1.1	Ferrol-Cerdido	1.2	Oviedo-Trubia
1.3	Cerdido-Ortigueira	1.4	Trubia-Pravia
1.5	Ortigueira-Xove	1.6	Pravia-Ortigueira
1.7	Xove-Ribadeo	1.8	Ortigueira-Ferrol
1.9	Ribadeo-Pravia		
1.11	Pravia-Trubia		
1.13	Trubia-Oviedo		

Para la línea 2 que une las estaciones de San Esteban – Pravia - Gijón se ha implementado la siguiente tabla que describe los distintos cantones:

2.1	Pravia-San Esteban	2.2	Sotiello-Veriña
2.3	Pravia-La Maruca	2.4	Gijón-Veriña
2.5	La Maruca-Trasona	2.6	Veriña-Trasona
2.7	Trasona-Veriña	2.8	Trasona-La Maruca
2.9	Veriña-Gijón	2.10	La Maruca-Pravia
2.11	Veriña-Sotiello	2.12	San Esteban-Pravia

Para la línea 3 que une las estaciones de Collanzo y Trubia se ha implementado la siguiente tabla que describe los distintos cantones:

3.1	Collanzo-Fuso	3.2	Trubia-Fuso
3.3	Fuso-Trubia	3.4	Fuso-Ujo
		3.6	Ujo-Collanzo

Para la línea 4 que une las estaciones de Gijón y Laviana y su ramal Sotiello – Aboño se ha implementado la siguiente tabla que describe los distintos cantones:

4.1	Gijón-Sotiello	4.2	Laviana- Carbayín
4.3	Aboño-Sotiello	4.4	Carbayín-El Berrón
4.5	Sotiello-El Berrón	4.6	El Berrón-Sotiello
4.7	El Berrón-Carbayín	4.8	Sotiello-Gijón
4.9	Carbayín-Laviana	4.10	Sotiello-Aboño

Para la línea 5 que une las estaciones de Oviedo y Santander y su ramal Llovio – Ribadesella Puerto se ha implementado la siguiente tabla que describe los distintos cantones:

5.1	Oviedo-Colloto	5.2	Ribadesella Puerto-Llovio
5.3	Colloto-El Berrón	5.4	Santander-Requejada
5.5	El Berrón-Lieres	5.6	Requejada-Cabezón
5.7	Lieres-Nava	5.8	Cabezón-Llanes
5.9	Nava-Llovio	5.10	Llanes-Llovio
5.11	Llovio-Ribadesella	5.12	Llovio-Infiesto
5.13	Ribadesella-Nueva	5.14	Infiesto-Rianes
5.15	Nueva-Posada	5.16	Rianes-Pola de Siero
5.17	Posada-Llanes	5.18	Pola de Siero-El Berrón
5.19	Llanes-Unquera	5.20	El Berrón-Colloto
5.21	Unquera-Cabezón	5.22	Colloto-Oviedo
5.23	Cabezón-Requejada		
5.25	Requejada-Santander		
5.27	Llovio-Ribadesella Puerto		

Para la línea 6 que une las estaciones de Santander y Bilbao y sus ramales Orejo – Liérganes, Irauregui – Lutzana y Basurto - Ariz se ha implementado la siguiente tabla que describe los distintos cantones:

6.1	Santander-Orejo	6.2	Ariz-Basurto Hospital
6.3	Orejo-Gama	6.4	Lutzana-Irauregui
6.5	Gama-Gibaja	6.6	Liérganes -Orejo
6.7	Gibaja-Karrantza	6.8	Bilbao-Basurto Hospital
6.9	Karrantza- Traslaviña	6.10	Basurto Hospital-Irauregi
6.11	Traslaviña-Aranguren	6.12	Irauregi-Aranguren
6.13	Aranguren-Irauregi	6.14	Aranguren-Karrantza
6.15	Irauregi-Basurto Hospital	6.16	Karrantza-Gama
6.17	Basurto Hospital-Bilbao	6.18	Gama-Orejo
6.19	Orejo-Liérganes	6.20	Orejo-Maliaño

6.21	Irauregi-Lutxana	6.22	Maliaño-Santander
6.23	Basurto Hospital-Ariz		

Para la línea 7 que une las estaciones de Aranguren y Asunción Universidad y su ramal Matallana – La Robla se ha implementado la siguiente tabla que describe los distintos cantones:

7.1	La Robla-Matallana	7.2	Aranguren-Balmaseda
7.3	Asunción Universidad-Matallana	7.4	Balmaseda-Bercedo Montija
7.5	Matallana-Guardo	7.6	Bercedo Montija-Guardo
7.7	Guardo- Los Carabeos	7.8	Guardo-Matallana
7.9	Los Carabeos-Arija	7.10	Matallana-La Robla
7.11	Arija-Soncillo	7.12	Matallana-Asunción Universidad
7.13	Soncillo-Pedrosa		
7.15	Pedrosa-Sotoscueva		
7.17	Sotoscueva-Bercedo Montija		
7.19	Bercedo Montija-Balmaseda		
7.21	Balmaseda-Aranguren		

Para la línea 8 que une las estaciones de Cartagena y Los Nietos no se ha implementado ningún cantón ya que no hay datos referentes a frenado por ausencia de material tanto motor como remolcado que hagan necesaria dicha implementación, por lo tanto en el presente proyecto se ha omitido dicha línea.

Partiendo de la anterior codificación de cantones hemos añadido los porcentajes de freno automático y de estacionamiento requeridos en cada tramo que figuran en el anexo IV, las cargas máximas que pueden remolcar las locomotoras en cada tramo de las distintas líneas así como el código de las estaciones que pertenecen a cada tramo, por lo que la estructura de cada línea del archivo CANTONES.csv es la siguiente:

Campo [0] = número del canton descrito en las anteriores tablas
 Campo [1] = freno automático del cantón para tipo 70
 Campo [2] = freno automático del cantón para tipo 60
 Campo [3] = freno automático del cantón para tipo 50
 Campo [4] = freno automático del cantón para tipo 40
 Campo [5] = freno automático del cantón para tipo 30
 Campo [6] = freno de estacionamiento del cantón
 Campo [7] = carga máxima del cantón para locomotoras de la serie 1500
 Campo [8] = carga máxima del cantón para locomotoras de la serie 1600
 Campo [9] = carga máxima del cantón para locomotoras de la serie 1900
 Campo [10-fin de línea] = código de las estaciones que pertenecen al cantón

Una vez que tenemos codificados los distintos tramos se ha creado el archivo **MARCHAS.csv** en la que a partir de las marchas de asimilación (anexo V) y de las distintas Órdenes del día que implementan los trenes grafiados (anexo VI) se ha implementado los distintos tramos que recorre cada tren o marcha de asimilación.

Campo[0] = marchas de asimilación o número del tren grafiado
 Campo[1-fin de línea]= cantones que recorre el tren o su marcha de asimilación

Dado que para línea existen 12 marchas de asimilación, seis para sentido que determinan el tipo de cada marcha según la tabla que se muestra a continuación:

Marchas	*01	*02	*03	*04	*05	*06	*07	*08	*09	*10	*11	*12
Tipo	80	80	70	70	60	60	50	50	40	40	30	30

Teniendo en cuenta que los trenes de mercancías y los compuestos de locomotora y vagones remolcados no pueden circular a tipo 80 se han agrupado las marchas de cada línea al tipo 80, para las líneas que tienen algún ramal se han implementado tantas marchas como ramales tiene la línea.

Como ejemplo la línea 1 que une las estaciones de Ferrol y Oviedo se han implementado las marchas 101 y 100, sin embargo para la línea 6 que une las estaciones de Santander y Bilbao y sus tres ramales se han implementado cuatro marchas 601 y cuatro marchas 600 para describir tanto la línea general como los posibles trayectos a recorrer por sus ramales. En la siguiente tabla mostramos la implementación de los cantones de las distintas marchas:

Marchas	Trayecto	Cantones										
101	Ferrol-Oviedo	1,1	1,3	1,5	1,7	1,9	1,11	1,13				
201	San Esteban-Gijón	2,1	2,3	2,5	2,7	2,9	2,11					
301	Collanzo-Trubia	3,1	3,3									
401	Gijón-Laviana	4,1	4,5	4,7	4,9							
401	Aboño-Laviana	4,3	4,5	4,7	4,9							
501	Oviedo-Santander	5,1	5,3	5,5	5,7	5,9	5,11	5,13	5,15	5,17	5,19	
		5,21	5,23	5,25								
501	Oviedo-Rib. Puerto	5,1	5,3	5,5	5,7	5,9	5,27					
601	Santander-Lierganes	6,1	6,2									
601	Santander-Bilbao	6,1	6,3	6,5	6,7	6,9	6,11	6,13	6,15	6,17		
601	Santander-Lutxana	6,1	6,3	6,5	6,7	6,9	6,11	6,13	6,15	6,21		
601	Santander-Ariz	6,1	6,3	6,5	6,7	6,9	6,11	6,13	6,15	6,23		
701	La Robla-Aranguren	7,1	7,5	7,7	7,9	7,11	7,13	7,15	7,17	7,19		
		7,21										
701	Universidad-Aranguren	7,3	7,5	7,7	7,9	7,11	7,13	7,15	7,17	7,19		
		7,21										
100	Oviedo-Ferrol	1,2	1,4	1,6	1,8							
200	Gijón-San Esteban	2,2	2,4	2,6	2,8	2,10	2,12					
300	Trubia-Collanzo	3,2	3,4	3,6								
400	Laviana-Gijón	4,2	4,4	4,6	4,8							
400	Laviana-Aboño	4,2	4,4	4,6	4,10							
500	Santander-Oviedo	5,4	5,6	5,8	5,10	5,12	5,14	5,16	5,18	5,20	5,22	
500	Rib. Puerto -Oviedo	5,2	5,12	5,14	5,16	5,18	5,20	5,22				
600	Bilbao-Santander	6,8	6,10	6,12	6,14	6,16	6,18	6,20	6,22			
600	Ariz-Santander	6,2	6,10	6,12	6,14	6,16	6,18	6,20	6,22			
600	Lutxana-Santander	6,4	6,12	6,14	6,16	6,18	6,20	6,22				
600	Liérganes-Santander	6,6	6,20	6,22								
700	Aranguren-La Robla	7,2	7,4	7,6	7,8	7,10						
700	Aranguren-Universidad	7,2	7,4	7,6	7,8	7,12						

Ahora partiendo de las distintas Ordenes del día que regulan los trenes grafiados (anexo VI) y el trayecto que recorren se han implementado en el archivo **MARCHAS.csv** los números de trenes de mercancías y los cantones que recorren a partir de la siguiente tabla:

Nº Tren	Trayecto	Cantones									
73201	Pravia-Trasona	2,3	2,5								
73203	Pravia-Trasona	2,3	2,5								
73211	Trasona-El Berrón	2,7	2,11	4,5							
73213	Trasona-El Berrón	2,7	2,11	4,5							
73215	Trasona-El Berrón	2,7	2,11	4,5							
73231	Pravia-El Berrón	1,1	1,1	5,1	5,3						
73245	Aboño-El Berrón	4,3	4,5								
73247	Soto Ribera-El Berrón	3,1	3,3	1,13	5,1	5,3					
73401	Santander-Maliaño	6,1									
73403	Barreda-Maliaño	5,23	5,25	6,1							
73405	Barreda-Maliaño	5,23	5,25	6,1							
73407	Barreda-Maliaño	5,23	5,25	6,1							
73409	Barreda-Santander	5,23	5,25								
73351	El Berrón-Santander	5,5 5,25	5,7	5,9	5,11	5,13	5,15	5,17	5,19	5,21	5,23
73353	El Berrón-Aranguren	5,5 5,25	5,7 6,1	5,9 6,3	5,11 6,5	5,13 6,7	5,15 6,9	5,17 6,11	5,19	5,21	5,23
73561	Santander-Ariz	6,1	6,3	6,5	6,7	6,9	6,11	6,13	6,15	6,23	
73563	Santander-Ariz	6,1	6,3	6,5	6,7	6,9	6,11	6,13	6,15	6,23	
73565	Santander-Aranguren	6,1	6,3	6,5	6,7	6,9	6,11				
73603	Aranguren-Lutxana	6,13	6,19								
73605	Aranguren-Lutxana	6,13	6,19								
73607	Balmaseda-Ariz	7,21	6,13	6,15	6,23						
73701	Arija-Balmaseda	7,11	7,13	7,15	7,17	7,19					
73700	Balmaseda-Arija	7,4	7,6								
73602	Lutxana-Aranguren	6,4	6,12								
73604	Ariz-Aranguren	6,2	6,1	6,12							
73606	Ariz-Balmaseda	6,2	6,1	6,12	7,2						
73760	Aranguren-Santander	6,14	6,16	6,18	6,20	6,22					
73762	Aranguren-Santander	6,14	6,16	6,18	6,20	6,22					
73764	Ariz-Santander	6,2	6,10	6,12							
73766	Aranguren-Santander	6,14	6,16	6,18	6,20	6,22					
73500	Santander-Pravia	5,4 1,2	5,6 1,4	5,8	5,10	5,12	5,14	5,16	5,18	5,2	5,22
73502	Santander-Pravia	5,4 1,2	5,6 1,4	5,8	5,10	5,12	5,14	5,16	5,18	5,2	5,22
73504	Santander-El Berrón	5,4	5,6	5,8	5,10	5,12	5,14	5,16	5,18		
73400	Santander-Barreda	5,4	5,6								
73402	Maliaño-Barreda	6,22	5,4	5,6							
73404	Maliaño-Santander	6,22									
73406	Santander-Barreda	5,4									
73408	Maliaño-Barreda	6,22	5,4	5,6							
73410	Maliaño-Santander	6,22									
73412	Maliaño-Santander	6,22									
73300	Pravia-Xove	1,6									
73210	El Berrón-Trasona	4,6	2,2	2,6							
73212	El Berrón-Trasona	4,6	2,2	2,6							
73244	El Berrón-Aboño	4,6	2,2								
73246	El Berrón-Soto Ribera	4,6	4,10								

A continuación partiendo de las características del material motor que circula por la red de ancho métrico Española (anexo III) se ha creado el archivo **LOCOMOTORAS.csv** en la que cada línea queda descrita de la siguiente forma:

Campo[0]=serie de la locomotora
Campo[1]=número de la locomotora
Campo[2]=uic de la locomotora
Campo[3]=peso de la locomotora
Campo[4]=freno automático de la locomotora
Campo[5]=freno de estacionamiento de la locomotora

A continuación partiendo de las características del material remolcado que circula por la red de ancho métrico Española (anexo III) se ha creado el archivo **VAGONES.csv** en la que cada línea queda descrita de la siguiente forma:

Campo[0]=serie del vagón
Campo[1]=número del vagón
Campo[2]=uic del vagón
Campo[3]=tara del vagón
Campo[4]=freno automático en vacío del vagón
Campo[5]=freno automático en cargado del vagón
Campo[6]=freno de estacionamiento de la locomotora

4.0 DESARROLLO DEL CÓDIGO

El proyecto tiene como objetivo principal elaborar los dos documentos que son necesarios para que los trenes de mercancías puedan circular por la red ferroviaria de ancho métrico española. Los documentos son la HOJA DE MARCHA, en la que figuran los datos correspondientes al trayecto a realizar por el tren y su composición de vehículos tractores y vagones remolcados. A partir de esta hoja de marcha se calcula el BOLETIN DE FRENADO, que es el otro documento que se pretende elaborar con la aplicación ZIRA. En dicho boletín de frenado vamos a calcular básicamente tres valores y comprobaremos que cumplen los requisitos establecidos para ese trayecto por el Reglamento de Circulación de Trenes y las Consignas que regulan el frenado de los trenes compuestos por locomotoras y vagones remolcados.

El primero de ellos va a ser la carga máxima, según el tipo de locomotoras y el trayecto a realizar la composición del tren no podrá superar dicho valor que viene regulado por consigna.

El segundo valor a va ser el porcentaje de freno automático, que viene determinado por la composición de los vagones remolcados y que debe ser superior al que determina el Reglamento de Circulación de Trenes según el trayecto.

El tercer valor a va ser el porcentaje de freno de estacionamiento, que viene determinado también por la composición de los vagones remolcados y que debe ser superior al que determina el Reglamento de Circulación de Trenes según el trayecto que deba realizar el tren.

Una vez que se ha comprobado que la composición cumple los requisitos mencionados anteriormente, elaborará dos archivos conforme establece el actual Reglamento de Circulación de Trenes de FEVE.

Para conseguir estos objetivos en este capítulo vamos a describir las distintas clases que componen nuestra aplicación desarrollada en lenguaje JAVA a la que se ha denominado ZIRA.

4.1 Clase Estacion:

Para la implementación de la clase locomotora se ha partido del archivo ESTACIONES.csv en el ya hemos visto que cada estación queda caracterizada por su código comercial, su nombre completo y su abreviatura para denominar a los trenes especiales que tienen como origen dicha estación. Para ello hemos definido los siguientes atributos de la clase Estacion:

```
package zira;

public class Estacion {

    private final String codigo;
    private final String nombre, abrev;
```

A continuación hemos implementado el constructor vacío de esos atributos:

```
public Estacion() {
    codigo = "";
    nombre = "";
    abrev = "";
}
```

A continuación hemos implementado el constructor con sus respectivos parámetros:

```
public Estacion(String codigo, String nombre, String abrev) {
    this.codigo = codigo;
    this.nombre = nombre;
    this.abrev = abrev;
}
```

Y por último hemos implementado los métodos getter y setter de los parámetros que necesitamos leer o modificar en otras clases de la aplicación:

```
public String getcodigo() {
    return codigo;
}

public String getnombre() {
    return nombre;
}

public String getabrev() {
    return abrev;
}
}
```

4.2 Clase Locomotora:

Para la implementación de la clase locomotora se ha partido del archivo LOCOMOTORAS.csv en el ya hemos visto que cada locomotora queda caracterizada por su serie, número, u.i.c., tara, peso freno automático, peso freno de estacionamiento y además vamos a añadir un nuevo atributo que es el concepto de locomotora remolcada para cuando en una composición haya locomotoras que no van traccionando sino remolcadas como si de un vagón más se tratara. Para ello hemos definido los siguientes atributos de la clase Locomotora:

```
package zira;

public class Locomotora {

    private int serie_;
    private int num_;
    private String uic_;
    private int peso;
    private int freno;
    private int est_;
    private boolean remolcada;
```

A continuación hemos implementado el constructor vacío de esos atributos:

```
public Locomotora() {
    serie_ = 0;
    num_ = 0;
    uic_ = "";
    peso = 0;
    freno = 0;
    est_ = 0;
    remolcada = false;
}
```

A continuación hemos implementado el constructor con sus respectivos parámetros:

```
public Locomotora(int serie_, int num_, String uic_, int peso, int freno, int est_) {
    this.serie_ = serie_;
    this.num_ = num_;
    this.uic_ = uic_;
    this.peso = peso;
    this.freno = freno;
    this.est_ = est_;
    remolcada = false;
}
```

Y por último hemos implementado los métodos getter y setter de los parámetros que necesitamos leer o modificar en otras clases de la aplicación:

```
public int getSerie_() {
    return serie_;
}

public int getNum_() {
    return num_;
}

public String getUic_() {
    return uic_;
}
```

```

    public int getPeso() {
        return peso;
    }

    public int getFreno() {
        return freno;
    }

    public int getEst_l() {
        return est_l;
    }

    public boolean isRemolcada() {
        return remolcada;
    }

    public void setSerie_l(int serie_l) {
        this.serie_l = serie_l;
    }

    public void setNum_l(int num_l) {
        this.num_l = num_l;
    }

    public void setUic_l(String uic_l) {
        this.uic_l = uic_l;
    }

    public void setPeso(int peso) {
        this.peso = peso;
    }

    public void setFreno(int freno) {
        this.freno = freno;
    }

    public void setEst_l(int est_l) {
        this.est_l = est_l;
    }

    public void setRemolcada(boolean remolcada) {
        this.remolcada = remolcada;
    }
}

```

4.3 Clase Vagon:

Para la implementación de la clase Vagon se ha partido del archivo VAGONES.csv en el ya hemos visto que cada vagón queda caracterizado por su serie, número, u.i.c., tara, peso freno automático en vacío, peso freno automático en cargado, peso freno de estacionamiento y además vamos a añadir tres nuevos atributos que son la mercancía que va a ser transportada en ese vagón, aislado que va reflejar la circunstancia de que no funcione correctamente el freno automático y por último el atributo de agarrotado que hace referencia a que no funciona el freno de estacionamiento de dicho vagón. Para ello hemos definido los siguientes atributos de la clase Vagon:

```
package zira;

public class Vagon {

    private String uic;
    private String serie;
    private int numero;
    private int tara;
    private int freno;
    private int est_v;
    private String mercancia;
    private boolean aislado;
    private boolean agarrotado;
    private int freno_cargado;
    private int freno_vacio;
```

A continuación hemos implementado el constructor vacío de esos atributos:

```
public Vagon() {

    serie = "";
    numero = -1;
    uic = "";
    tara = -1;
    freno_vacio = -1;
    freno_cargado = -1;
    est_v = -1;
    mercancia = "";
    aislado = false;
    agarrotado = false;

}
```

A continuación hemos implementado el constructor con sus respectivos parámetros:

```
public Vagon(String uic, String serie, int numero, int tara, int freno_vacio, int freno_cargado, int est_v,
              String mercancia) {

    this.serie = serie;
    this.numero = numero;
    this.uic = uic;
    this.tara = tara;
    this.freno_vacio = freno_vacio;
    this.freno_cargado = freno_cargado;
    this.est_v = est_v;
    aislado = false;
    agarrotado = false;

}
```

Y por último hemos implementado los métodos getter y setter de los parámetros que necesitamos leer o modificar en otras clases de la aplicación:

```
public String getUic() {
    return uic;
}

public String getSerie() {
    return serie;
}

public int getNumero() {
    return numero;
}

public int getTara() {
    return tara;
}

public int getFreno() {
    return freno;
}

public int getEst_v() {
    return est_v;
}

public String getMercancia() {
    return mercancia;
}

public boolean isAislado() {
    return aislado;
}

public boolean isAgarrotado() {
    return agarrotado;
}

public int getFreno_cargado() {
    return freno_cargado;
}

public int getFreno_vacio() {
    return freno_vacio;
}
}
```

4.4 Clase Canton:

Para la implementación de la clase Canton se ha partido del archivo CANTONES.csv en el ya hemos visto que cada cantón queda caracterizado por su número, un array de 5 elementos que determinan el peso freno automático requerido según el tipo del tren, el freno de estacionamiento requerido, un array de 3 elementos que determinan la carga máxima según la serie de la locomotora y un array de longitud variable con la lista de los códigos de las estaciones que componen dicho cantón. Para ello hemos definido los siguientes atributos de la clase Canton:

```
package zira;

import java.util.ArrayList;

public class Canton {
    String numero;
    int[] marcha_asimil = new int[5];
    /*
     * Marcha de asimiliacion por tramo 0(02/03) 1(04/05) 2(06/07) 3(08/09)
     * 4(10/11)
     */
    int est;
    int[] carga_max = new int[3];
    ArrayList<String> canton_origen = new ArrayList<String>();
}
```

A continuación hemos implementado el constructor vacío de esos atributos:

```
public Canton() {
    numero = "";
    for (int i = 0; i < 5; i++) {
        marcha_asimil[i] = 0;
    }
    est = 0;
    for (int i = 0; i < 3; i++) {
        carga_max[i] = 0;
    }
}
```

A continuación hemos implementado el constructor con sus respectivos parámetros:

```
public Canton(String numero, int[] m_asimil, int est, int[] c_max, ArrayList<String> cantones) {

    this.numero = numero;
    for (int i = 0; i < 5; i++) {
        marcha_asimil[i] = m_asimil[i];
    }
    this.est = est;
    for (int i = 0; i < 3; i++) {
        carga_max[i] = c_max[i];
    }
    this.canton_origen = cantones;
}
```

Y por último hemos implementado los métodos getter y setter de los parámetros que necesitamos leer o modificar en otras clases de la aplicación:

```
public String getnumero() {
    return numero;
}
```

```

public void setnumero(String n) {
    numero = n;
}

public ArrayList<String> getlista_estaciones() {
    return canton_origen;
}

// Devuelve el freno automático segun el tipo del tren
public int getmarcha_asimil(int pos) {
    if (pos >= 0 && pos < 5)
        return marcha_asimil[pos];
    else
        return -1;
}

public int gettest() {
    return est;
}

// Devuelve la carga maxima segun la serie de la locomotora
int getcarga_max(int posi) {
    if (posi >= 0 && posi < 3)
        return carga_max[posi];
    else
        return -1;
}
}

```

4.5 Clase Marcha:

Para la implementación de la clase Marcha se ha partido del archivo MARCHAS.csv en el ya hemos visto que cada marcha queda caracterizado por su número de tren o bien la marcha de asimilación tipo 80 según la línea y su paridad y por un array de longitud variable que contiene los códigos de los cantones que describen el trayecto de un tren o marcha de asimilación. Para ello hemos definido los siguientes atributos de la clase Marcha:

```
package zira;

import java.util.ArrayList;

public class Marcha {
    private int marcha_asimil;
    private ArrayList<String> cantones;
```

A continuación hemos implementado el constructor vacío de esos atributos:

```
Marcha() {
    marcha_asimil = 0;
    cantones = new ArrayList<String>();
}
```

A continuación hemos implementado el constructor con sus respectivos parámetros:

```
Marcha(int marcha_asimil, ArrayList<String> c) {
    this.marcha_asimil = marcha_asimil;
    cantones = new ArrayList<String>();
    this.cantones = c;
}
```

Y por último hemos implementado los métodos getter de los parámetros que necesitamos leer en otras clases de la aplicación:

```
public int getmarcha_asimil() {
    return marcha_asimil;
}

public ArrayList<String> getcantones() {
    return cantones;
}
}
```

4.6 Clase Zira:

Para la implementación de la clase Marcha se ha partido de las cinco clases anteriormente descritas para encontrar en el archivo CANTONES.csv los tres valores que van a ser de obligado cumplimiento para que pueda circular la composición del tren por el trayecto a recorrer. Dichos valores son el porcentaje de freno automático, el porcentaje de freno de estacionamiento y la carga máxima del trayecto que no debe superar la composición. Para ello vamos a ir describiendo las distintas funciones implementadas para hallar dichos valores, vamos a empezar definiendo unas variables:

```
package zira;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;

public class zira {

    /**
     *
     * VARIABLES
     */
    private static ArrayList<Canton> cantones = new ArrayList<Canton>();
    private static ArrayList<Estacion> estaciones = new ArrayList<Estacion>();
    private static ArrayList<Marcha> marchas = new ArrayList<Marcha>();
    private static ArrayList<Locomotoras> locomotoras = new ArrayList<Locomotoras>();
    private static ArrayList<Vagon> vagones = new ArrayList<Vagon>();

    private static int[] tabla_tipos = { 70, 70, 60, 60, 50, 50, 40, 40, 30, 30 };
    private static int[] tabla_peso_frenos = { 70, 60, 50, 40, 30 };
    private static int[] tabla_cargas = { 1900, 1600, 1500 };

    private static int peso_freno;
    private static int estacionamiento;
    private static int serie_;
    private static int carga;

    private HashMap<String, Integer> c_vagones_tipo;
    private HashMap<String, Integer> vagones_cm;
```

A continuación hemos implementado el constructor que inicia la carga de los archivos en memoria y otras variables necesarias:

```
public zira() throws FileNotFoundException, IOException {
    cargarEstaciones("ESTACIONES.csv");
    cargarMarchas("MARCHAS.csv");
    cargarCantones("CANTONES.csv");
    cargarLocomotoras("LOCOMOTORAS.csv");
    cargarVagones("VAGONES.csv");
    peso_freno = 0;
    estacionamiento = 0;
    cargarCantVagones();
    cargarSerie();
}
```

A continuación hemos implementado la función que carga la información que contiene el archivo ESTACIONES.csv:

```
public static void cargarEstaciones(String NombreFichero) throws FileNotFoundException, IOException {
    String cod;
    String nom, abre;

    BufferedReader br = null;
    br = new BufferedReader(new FileReader(NombreFichero));
    String linea = br.readLine();
    while (null != linea) {
        String[] campos = linea.split(",");
        cod = campos[0];
        nom = campos[1];
        abre = campos[2];
        Estacion e = new Estacion(cod, nom, abre);
        estaciones.add(e);
        linea = br.readLine();
    }
    br.close();
}
```

A continuación hemos implementado la función que carga la información que contiene el archivo CANTONES.csv:

```
public static void cargarCantones(String NombreFichero) throws FileNotFoundException, IOException {
    String numero;
    int[] m_asimil = new int[5];
    int[] c_max = new int[3];
    int est;
    BufferedReader br = null;
    br = new BufferedReader(new FileReader(NombreFichero));
    String linea = br.readLine();
    while (null != linea) {
        String[] campos = linea.split(",");
        numero = campos[0];
        ArrayList<String> canton_estaciones = new ArrayList<String>();
        // dividir en linea y numeros de canton
        for (int i = 0; i < 5; i++) {
            m_asimil[i] = Integer.parseInt(campos[1 + i]);
        }
        est = Integer.parseInt(campos[6]);
        for (int i = 0; i < 3; i++) {
            c_max[i] = Integer.parseInt(campos[7 + i]);
        }
        for (int i = 10; i < campos.length; i++) {
            canton_estaciones.add(campos[i]);
        }

        Canton nuevo_canton = new Canton(numero, m_asimil, est, c_max,
canton_estaciones);
        cantones.add(nuevo_canton);

        linea = br.readLine();
    }
    br.close();
}
```

A continuación hemos implementado la función que carga la información que contiene el archivo MARCHAS.csv:

```
public static void cargarMarchas(String NombreFichero) throws FileNotFoundException, IOException {
    int m_a;
    BufferedReader br = null;
    br = new BufferedReader(new FileReader(NombreFichero));
    String linea = br.readLine();
    while (null != linea) {
        String[] campos = linea.split(",");
        m_a = Integer.parseInt(campos[0]);
        ArrayList<String> c = new ArrayList<String>();
        for (int i = 0; i < campos.length - 1; i++) {
            c.add(campos[1 + i]);
        }
        Marcha m = new Marcha(m_a, c);
        marchas.add(m);
        linea = br.readLine();
    }
    br.close();
}
```

A continuación hemos implementado la función que carga la información que contiene el archivo LOCOMOTORAS.csv:

```
public static void cargarLocomotoras(String NombreFichero) throws FileNotFoundException, IOException {
    int serie = 0;
    String uic_l = "";
    int num_l = 0;
    int peso = 0;
    int freno = 0;
    int est_l = 0;

    BufferedReader br = null;
    br = new BufferedReader(new FileReader(NombreFichero));
    String linea = br.readLine();
    while (null != linea) {
        String[] campos = linea.split(",");
        serie = Integer.parseInt(campos[0]);
        num_l = Integer.parseInt(campos[1]);
        uic_l = campos[2];
        peso = Integer.parseInt(campos[3]);
        freno = Integer.parseInt(campos[4]);
        est_l = Integer.parseInt(campos[5]);
        Locomotora l = new Locomotora(serie, num_l, uic_l, peso, freno, est_l);
        locomotoras.add(l);
        linea = br.readLine();
    }
    br.close();
}
```

A continuación hemos implementado la función que carga la información que contiene el archivo VAGONES.csv:

```
public static void cargarVagones(String NombreFichero) throws FileNotFoundException, IOException {
    String uic = "";
    String serie = "";
    int numero = -1;
    int tara = -1;
    int freno_vacio = -1;
    int freno_cargado = -1;
    int est_v = -1;
    String mercancia = "";

    BufferedReader br = null;
    br = new BufferedReader(new FileReader(NombreFichero));
    String linea = br.readLine();
}
```

```

while (null != linea) {
    String[] campos = linea.split(";");
    serie = campos[0];
    numero = Integer.parseInt(campos[1]);
    uic = campos[2];
    tara = Integer.parseInt(campos[3]);
    freno_vacio = Integer.parseInt(campos[4]);
    freno_cargado = Integer.parseInt(campos[5]);
    est_v = Integer.parseInt(campos[6]);
    Vagon v = new Vagon(uic, serie, numero, tara, freno_vacio, freno_cargado, est_v,
mercancia);

    vagones.add(v);
    linea = br.readLine();
}
br.close();
}

```

A continuación hemos implementado la función que obtiene el nombre de una estación si se le proporciona el código de la misma:

```

public String Estacionporcodigo(String cod) {
    String nom = "";
    boolean encontrado = false;
    int i = 0;
    while ((i < estaciones.size()) && (!encontrado)) {
        if (estaciones.get(i).getcodigo().equalsIgnoreCase(cod)) {
            encontrado = true;
            nom = estaciones.get(i).getnombre();
        } else
            i++;
    }
    return nom;
}

```

A continuación hemos implementado la función que obtiene el código de una estación si se le proporciona el nombre de la misma:

```

public String Estacionpornombre(String nom) {
    String cod = "";
    boolean encontrado = false;
    int i = 0;
    while ((i < estaciones.size()) && (!encontrado)) {
        if (estaciones.get(i).getnombre().equalsIgnoreCase(nom)) {
            encontrado = true;
            cod = estaciones.get(i).getcodigo();
        } else
            i++;
    }

    return cod;
}

public ArrayList<String> getcantones() {
    return cantones;
}
}

```

A continuación hemos implementado la función que obtiene la lista de estaciones de un determinado cantón a partir de la lista de cantones:

```

public Canton buscarcanton(String c, ArrayList<Canton> l_c) {
    Canton can = new Canton();
    can.setnumero("-1");
    boolean encontrado = false;
    int i = 0;
    while ((i < l_c.size()) && (!encontrado)) {
        if (l_c.get(i).getnumero().equalsIgnoreCase(c)) {
            can = l_c.get(i);
            encontrado = true;
        } else

```

```

        i++;
    }
    return can;
}

```

A continuación hemos implementado la función que verifica que el número de locomotora que el usuario introduce está en el archivo LOCOMOTORAS.csv:

```

public Locomotora buscarlocomotora(int numero) {
    boolean encontrado = false;
    Locomotora loc = new Locomotora();
    loc.setNum_l(-1);
    int i = 0;
    while ((i < locomotoras.size()) && (!encontrado)) {
        if (locomotoras.get(i).getNum_l() == numero) {
            loc = locomotoras.get(i);
            encontrado = true;
        } else
            i++;
    }
    return loc;
}

```

A continuación hemos implementado la función que verifica que la seire y el número de vagón que el usuario introduce está en el archivo VAGONES.csv:

```

public Vagon buscarvagon(String serie, int numero) {
    boolean encontrado = false;
    Vagon vag = new Vagon();
    int i = 0;
    while ((i < vagones.size()) && (!encontrado)) {
        if ((vagones.get(i).getNumero() == numero) &&
(vagones.get(i).getSerie().equalsIgnoreCase(serie))) {
            vag = vagones.get(i);
            encontrado = true;
        } else
            i++;
    }
    return vag;
}

```

A continuación hemos implementado la función que partiendo del código de una estación, la lista de cantones y si la estación es la de origen o la de destino localiza el canton al que pertenece dicha estación. Se ha implementado así para solucionar las estaciones frontera entre cantones, si es la estación de origen buscamos en la lista de estaciones del cantón menos en la última posición, y si es la estación de destino buscamos el código de estación menos en la primera posición:

```

public String buscar_canton_de_estacion(String estacion, ArrayList<String> l_cantones, String Modo) {
    String canton_devuelto = "";
    int posicion;
    ArrayList<String> l_estacion = new ArrayList<String>();
    Canton c;
    boolean encontrado = false;
    int i = 0;
    while ((i < l_cantones.size()) && (!encontrado)) {

        c = buscarcanton(l_cantones.get(i), cantones);
        if (c.getnumero().equalsIgnoreCase("-1")) {

            i++;
        } else {

            l_estacion = c.getlista_estaciones();

            if (l_estacion.contains(estacion)) {

```

```

        posicion = l_estacion.indexOf(estacion);

        if (Modo.equalsIgnoreCase("origen") && posicion !=
l_estacion.size() - 1) {
            encontrado = true;
            canton_devuelto = l_cantones.get(i);

        } else if (Modo.equalsIgnoreCase("destino") && posicion != 0) {
            encontrado = true;
            canton_devuelto = l_cantones.get(i);
        } else
            i++;
        } else {
            i++;
        }
    }
}
return canton_devuelto;
}
}

```

A continuación hemos implementado la función que obtiene la lista de cantones del trayecto definido por el cantón de la estación de origen y el cantón de la estación de destino:

```

public ArrayList<String> creartrayecto(String o, String d, ArrayList<String> l_c) {
    ArrayList<String> trayecto = new ArrayList<String>();
    int p_origen = l_c.indexOf(o);
    int p_destino = l_c.indexOf(d);
    for (int i = p_origen; i <= p_destino; i++) {
        trayecto.add(l_c.get(i));
    }
    return trayecto;
}

```

A continuación hemos implementado la función que verifica que el número de tren introducido por el usuario está en el archivo MARCHAS.csv:

```

public boolean estaMarcha(int m_a) {
    boolean encontrado = false;
    int i = 0;
    while (i < marchas.size() && !encontrado) {
        if (marchas.get(i).getmarcha_asimil() == m_a) {
            encontrado = true;
        } else
            i++;
    }
    return encontrado;
}

```

A continuación hemos implementado la función que partiendo de la marcha de asimilación introducida por el usuario calcula el código general implementado según la línea y la paridad en el archivo MARCHAS.csv:

```

public int calcularmarcha_asimil(int m_a) {
    int pos;
    int asimil = -1;
    if (m_a < 999) {
        pos = m_a % 100;

        if ((pos % 2 == 0) && (pos < 10) && (pos > 1)) {
            asimil = m_a - pos;
        }
    }
}

```

```

        else if ((pos < 10) && (pos > 1))
            asimil = (m_a - pos) + 1;
    } else {
        asimil = m_a;
    }

    return asimil;
}

```

A continuación hemos implementado la función que obtiene el tipo del tren partiendo del número de tren o marcha de asimilación introducida por el usuario:

```

public int calculartipo(int m_a) {
    int tipo = -1;
    int pos;
    if (m_a > 999) {
        if (estaMarcha(m_a))
            tipo = 60;
        else
            tipo = -4;
    } else {
        pos = (m_a % 100) - 2;

        if ((pos >= 0) && (pos < 12)) {

            tipo = tabla_tipos[pos];
        }
    }
    return tipo;
}

```

A continuación hemos implementado la función que obtiene la posición a buscar dependiendo del tipo del tren para calcular el freno automático necesario:

```

public int calcularposicionfreno(int tipo) {
    int posicion = -1;
    boolean encontrado = false;
    int i = 0;
    while ((i < tabla_peso_frenos.length) && (!encontrado)) {
        if (tipo == tabla_peso_frenos[i]) {
            encontrado = true;
            posicion = i;
        } else
            i++;
    }
    return posicion;
}

```

A continuación hemos implementado la función que obtiene la serie de la locomotora a partir del número de locomotora introducida por el usuario:

```

public void calcularserie_locomotora(int numero_l) {
    boolean encontrado = false;
    int i = 0;
    while ((i < locomotoras.size()) && (!encontrado)) {
        if (numero_l == locomotoras.get(i).getNum_l()) {
            encontrado = true;
            serie_l = locomotoras.get(i).getSerie_l();
        } else
            i++;
    }
}

```

A continuación hemos implementado la función que calcula la posición de la carga máxima que se puede remolcar según la serie de la locomotora introducida por el usuario:

```
public int calcularposicioncarga(int serie_l) {
    int posicion = -1;
    boolean encontrado = false;
    int i = 0;
    System.out.println("Buscamos la carga " + serie_l);
    while ((i < tabla_cargas.length) && (!encontrado)) {
        if (serie_l == tabla_cargas[i]) {
            encontrado = true;
            posicion = i;
        } else
            i++;
    }
    return posicion;
}
```

A continuación hemos implementado la función que va a obtener los valores del porcentaje de freno automático y de freno de estacionamiento necesarios y la carga máxima. Para ello vamos a recorrer la lista de los cantones y vamos a buscar en el archivo CANTONES.csv para los porcentajes de frenado el valor máximo y sin embargo para buscar la carga máxima vamos a quedarnos con el valor mínimo. Devolverá un valor no válido en la variable tipo del tren si no encuentra la estación de origen y/o destino dentro del número de tren o marcha de asimilación introducida por el usuario:

```
public int revisor(int tipo, int asimil, String origen, String destino, int numero_l) {
    int i = 0;
    String canton_origen;
    String canton_destino;

    Canton c;
    int posicion_freno;
    int posicion_carga;
    ArrayList<ArrayList<String>> lista_cantones = new ArrayList<ArrayList<String>>();

    ArrayList<String> marcha_cantones;
    ArrayList<String> trayecto = new ArrayList<String>();

    for (i = 0; i < marchas.size(); i++) {
        if (asimil == marchas.get(i).getmarcha_asimil()) {
            marcha_cantones = new ArrayList<>();
            marcha_cantones = marchas.get(i).getcantones();
            lista_cantones.add(marcha_cantones);
        }
    }
    i = 0;
    boolean encontrado = false;
    while (i < lista_cantones.size() && !encontrado) {
        marcha_cantones = lista_cantones.get(i);
        canton_origen = buscar_canton_de_estacion(origen, marcha_cantones, "origen");
        i++;
        tipo = -3;
    } else {
        canton_destino = buscar_canton_de_estacion(destino, marcha_cantones,
"destino");

        if (!canton_origen.isEmpty() && !canton_destino.isEmpty()) {
            encontrado = true;
            trayecto = creartrayecto(canton_origen, canton_destino, marcha_cantones);
            posicion_freno = calcularposicionfreno(tipo);
            calcularserie_locomotora(numero_l);
            posicion_carga = calcularposicioncarga(serie_l);
            peso_freno = -1;
        }
    }
}
```

```

        estacionamiento = -1;
        carga = 3000;
        for (i = 0; i < trayecto.size(); i++) {
            c = buscarcanton(trayecto.get(i), cantones);
            if (peso_freno < c.getmarcha_asimil(posicion_freno)) {
                peso_freno = c.getmarcha_asimil(posicion_freno);
            }
            if (estacionamiento < c.gettest()) {
                estacionamiento = c.gettest();
            }
            if (carga > c.getcarga_max(posicion_carga)) {
                carga = c.getcarga_max(posicion_carga);
            }
        }
    } else {
        i++;
        tipo = -3;
    }
}

return tipo;
}

```

A continuación hemos inicializado la lista de la cantidad máxima de vagones según el tipo del tren:

```

private void cargarCantVagones() {
    c_vagones_tipo = new HashMap<String, Integer>();
    c_vagones_tipo.put("70", 20);
    c_vagones_tipo.put("60", 20);
    c_vagones_tipo.put("50", 25);
    c_vagones_tipo.put("40", 25);
    c_vagones_tipo.put("30", 25);
}

```

A continuación hemos implementado la función que obtiene la cantida máxima de vagones según el tipo del tren:

```

public int cantMaxVagones(String tipo) {
    if (vagones_cm.containsKey(tipo)) {
        return vagones_cm.get(tipo);
    } else {
        return 0;
    }
}

```

A continuación hemos implementado las funciones getter de los tres valores obtenidos en la función revisor y que van a determinar si la composición del tren puede circular por el trayecto:

```

public int getpeso_freno() {
    return peso_freno;
}

public int getestacionamiento() {
    return estacionamiento;
}

public int getcarga() {
    return carga;
}
}

```

4.7 Clase HojaMarcha:

Para la implementación de la clase HojaMarcha se ha partido de los datos que hay rellena de la plantilla para elaborar dicho documento que son:

1. Copiar la naturaleza que introduce el usuario
2. Copiar el número de tren que introduce el usuario
3. Poner el valor de la variable tipo
4. Copiar la asimilación que introduce el usuario
5. Nombre de la estación de origen
6. Código de la estación de origen
7. Nombre de la estación de destino
8. Código de la estación de destino
9. Copiar la fecha que introduce el usuario
10. Copiar la hora de salida que introduce el usuario
11. Copiar el número de locomotora que mete el usuario
12. Copiar el número de locomotora si se introduce otra locomotora
13. Serie del primer vagón que introduce el usuario
14. Número de vagón que introduce el usuario
15. Mercancía del vagón que introduce el usuario
16. Peso del vagón que introduce el usuario
17. Peso total de la composición del tren.

Para ello hemos definido los siguientes atributos de la clase HojaMarcha:

```
package zira;

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

public class HojaMarcha {

    private String valor_1;           // NATURALEZA
    private String valor_2;           // Num TREN
    private String valor_3;           // TIPO
    private String valor_4;           // ASIMILACION
    private String valor_5;           // NOMBRE ESTACION ORIGEN
    private String valor_6;           // CODIGO ESTACION ORIGEN
    private String valor_7;           // NOMBRE ESTACION DESTINO
    private String valor_8;           // CODIGO ESTACION DESTINO
    private String valor_9;           // FECHA
    private String valor_10;          // HORA
    private String valor_11;          // NUMERO LOCOMOTORA 1
    private String valor_12;          // NUMERO LOCOMOTORA 2
    private String valor_17;          // TOTAL PESO

    private List<HojaMarchaVagon> vagones; private ArrayList<String> cantones;
```

A continuación hemos implementado el constructor de esos atributos vacíos:

```
public HojaMarcha() {
    super();
    this.valor_1 = "";
    this.valor_2 = "";
    this.valor_3 = "";
    this.valor_4 = "";
    this.valor_5 = "";
    this.valor_6 = "";
    this.valor_7 = "";
```

```

        this.valor_8 = "";
        this.valor_9 = "";
        this.valor_10 = "";
        this.valor_11 = "";
        this.valor_12 = "";
        this.vagones = new ArrayList<HojaMarchaVagon>();
    }

```

A continuación hemos implementado el constructor para cada vagón con sus atributos vacíos y los métodos getter y setter:

```

class HojaMarchaVagon {

    private String valor_13;        // SERIE VAGON
    private String valor_14;        // Num VAGON
    private String valor_15;        // MERCANCIA
    private String valor_16;        // PESO

    public HojaMarchaVagon() {
        super();
        this.valor_13 = "";
        this.valor_14 = "";
        this.valor_15 = "";
        this.valor_16 = "";
    } public String getValor_13() {
        return valor_13;
    }

    public void setValor_13(String valor_13) {
        this.valor_13 = valor_13;
    }

    public String getValor_14() {
        return valor_14;
    }

    public void setValor_14(String valor_14) {
        this.valor_14 = valor_14;
    }

    public String getValor_15() {
        return valor_15;
    }

    public void setValor_15(String valor_15) {
        this.valor_15 = valor_15;
    }

    public String getValor_16() {
        return valor_16;
    }

    public void setValor_16(String valor_16) {
        this.valor_16 = valor_16;
    }
}

```

Y por último hemos implementado los métodos getter de los parámetros que necesitamos leer en otras clases de la aplicación:

```
public String getValor_1() {  
    return valor_1;  
}  
  
public void setValor_1(String valor_1) {  
    this.valor_1 = valor_1;  
}  
  
public String getValor_2() {  
    return valor_2;  
}  
  
public void setValor_2(String valor_2) {  
    this.valor_2 = valor_2;  
}  
  
public String getValor_3() {  
    return valor_3;  
}  
}
```

4.8 Clase BoletinFrenado:

Para la implementación de la clase HojaMarcha se ha partido de los datos que hay rellena de la plantilla para elaborar dicho documento que son:

1. Tren N°: copiar el N° de tren que mete el usuario
2. Fecha: copiar la fecha que mete el usuario
3. N° vehículos: es el número total de elementos de la tabla vagones
4. Clase freno: Siempre va a ser la letra "A".
5. Peso total remolcado: valor de la variable peso_total.
6. Motores: la suma de los pesos automáticos de las locomotoras que van remolcadas en la tabla vagones. Si no hubiera ponemos cero.
7. Remolcado: la suma de los pesos automáticos de los vagones que ha introducido el usuario.
8. Disponible: se obtiene multiplicando por cien en valor anterior (Total) y se divide por el peso total remolcado.
9. Total: es la suma de los dos valores anteriores, motores y remolcado.
10. Necesario: es el valor de la variable "peso_freno" de la clase zira.
11. Motores: la suma de los pesos de estacionamiento de las locomotoras que van remolcadas en la tabla vagones. Si no hubiera ponemos cero.
12. Remolcado: la suma de los pesos de estacionamiento de los vagones que ha introducido el usuario.
13. Total: es la suma de los dos valores anteriores, motores y remolcado.
14. Disponible: se obtiene multiplicando por cien en valor anterior (Total) y se divide por el peso total remolcado.
15. Necesario: es el valor de la variable "estacionamiento" de la clase zira.
16. Sin freno: se obtiene multiplicando por cuatro el número de vagones que circulan aislados.
17. Régimen G: siempre es cero.
18. Régimen P: se obtiene multiplicando por cuatro los vagones que circulan con freno, es decir, multiplicando por cuatro el número de vagones que no circulan aislados.
19. Total: se obtiene multiplicando por cuatro el N° de vehículos.

Para ello hemos definido los siguientes atributos de la clase HojaMarcha:

```
package zira;

public class BoletinFrenado {

    private String valor_1;           // Num TREN
    private String valor_2;           // FECHA
    private String valor_3;           // Num VEHICULOS
    private String valor_4;           // CLASE FRENO
    private String valor_5;           // PESO TOTAL REMOLCADO
    private String valor_6;           // MOTORES
    private String valor_7;           // REMOLCADO
    private String valor_8;           // TOTAL
    private String valor_9;           // DISPONILBE
    private String valor_10;          // NECESARIO
    private String valor_11;          // MOTORES
    private String valor_12;          // REMOLCADO
    private String valor_13;          // TOTAL
    private String valor_14;          // DISPONILBE
    private String valor_15;          // NECESARIO
```

```

private String valor_16;      // SIN FRENO
private String valor_17;      // REGIMEN G
private String valor_18;      // REGIMEN P
private String valor_19;      // TOTAL

```

A continuación hemos implementado el constructor de esos atributos vacíos:

```

public BoletinFrenado() {
    super();
    this.valor_1 = "";
    this.valor_2 = "";
    this.valor_3 = "";
    this.valor_4 = "A";
    this.valor_5 = "";
    this.valor_6 = "0";
    this.valor_7 = "";
    this.valor_8 = "";
    this.valor_9 = "";
    this.valor_10 = "";
    this.valor_11 = "0";
    this.valor_12 = "";
    this.valor_13 = "";
    this.valor_14 = "";
    this.valor_15 = "";
    this.valor_16 = "";
    this.valor_17 = "0";
    this.valor_18 = "";
    this.valor_19 = "";
}

```

Y por último hemos implementado los métodos getter y setter de los parámetros que necesitamos leer en otras clases de la aplicación:

```

public String getValor_1() {
    return valor_1;
}

public void setValor_1(String valor_1) {
    this.valor_1 = valor_1;
}

public String getValor_2() {
    return valor_2;
}

public void setValor_2(String valor_2) {
    this.valor_2 = valor_2;
}

public String getValor_3() {
    return valor_3;
}

public void setValor_3(String valor_3) {
    this.valor_3 = valor_3;
}

public String getValor_4() {
    return valor_4;
}

public void setValor_4(String valor_4) {
    this.valor_4 = valor_4;
}

public String getValor_5() {
    return valor_5;
}

```

```

public void setValor_5(String valor_5) {
    this.valor_5 = valor_5;
}

public String getValor_6() {
    return valor_6;
}

public void setValor_6(String valor_6) {
    this.valor_6 = valor_6;
}

public String getValor_7() {
    return valor_7;
}

public void setValor_7(String valor_7) {
    this.valor_7 = valor_7;
}

public String getValor_8() {
    return valor_8;
}

public void setValor_8(String valor_8) {
    this.valor_8 = valor_8;
}

public String getValor_9() {
    return valor_9;
}

public void setValor_9(String valor_9) {
    this.valor_9 = valor_9;
}

public String getValor_10() {
    return valor_10;
}

public void setValor_10(String valor_10) {
    this.valor_10 = valor_10;
}

public String getValor_11() {
    return valor_11;
}

public void setValor_11(String valor_11) {
    this.valor_11 = valor_11;
}

public String getValor_12() {
    return valor_12;
}

public void setValor_12(String valor_12) {
    this.valor_12 = valor_12;
}

public String getValor_13() {
    return valor_13;
}

public void setValor_13(String valor_13) {
    this.valor_13 = valor_13;
}

public String getValor_14() {
    return valor_14;
}

```

```
}

public void setValor_14(String valor_14) {
    this.valor_14 = valor_14;
}

public String getValor_15() {
    return valor_15;
}

public void setValor_15(String valor_15) {
    this.valor_15 = valor_15;
}

public String getValor_16() {
    return valor_16;
}

public void setValor_16(String valor_16) {
    this.valor_16 = valor_16;
}

public String getValor_17() {
    return valor_17;
}

public void setValor_17(String valor_17) {
    this.valor_17 = valor_17;
}

public String getValor_18() {
    return valor_18;
}

public void setValor_18(String valor_18) {
    this.valor_18 = valor_18;
}

public String getValor_19() {
    return valor_19;
}

public void setValor_19(String valor_19) {
    this.valor_19 = valor_19;
}

}
```

4.9 Clase UtilPdf:

En la implementación de la clase UtilPdf se han desarrollado distintos métodos que nos van a ayudar a la hora de la creación de los archivos Hoja de Marcha y Boletín de Frenado en formato PDF. A continuación se muestran dichos métodos y una breve explicación:

```
package zira;

import java.awt.Desktop;
import java.io.File;

import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Font.FontFamily;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.Rectangle;
import com.itextpdf.text.pdf.PdfPCell;

public class UtilPdf {

    /**
     * ESTILOS PARRAFOS Y TITULOS
     */
    protected static Font ftitulo = new Font(FontFamily.TIMES_ROMAN, 28, Font.BOLD, BaseColor.BLACK);
    protected static Font ftexth = new Font(FontFamily.TIMES_ROMAN, 10, Font.BOLD, BaseColor.BLACK);
    protected static Font ftext = new Font(FontFamily.TIMES_ROMAN, 10, Font.NORMAL, BaseColor.BLACK);

    /**
     * ESTILO DEL DOCUMENTO TAMAÑO DE LA HOJA Y MARGEN
     */
    protected static Document getDocument(int margintop) {
        Rectangle rectangle = PageSize.LEGAL.rotate();
        return new Document(rectangle,10,10,margintop,15);
    }
}
```

A continuación hemos implementado el método “createCellt” que construye una celda de una tabla del título dada una frase, la fila, la columna y la rotación:

```
protected static PdfPCell createCellt(String frase, int row, int col, int rot){
    PdfPCell cell = new PdfPCell(new Phrase(frase,ftitulo));
    if(row > 0)cell.setRowspan(row);
    if(col > 0)cell.setColspan(col);
    if(rot > 0)cell.setRotation(rot);
    cell.setHorizontalAlignment(Element.ALIGN_CENTER);
    cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
    return cell;
}
```

A continuación hemos implementado el método “createCellh” que construye una celda de una tabla del tipo párrafo centrado en negrita dada una frase, la fila, la columna y la rotación:

```
/**
 *
 * CONSTRUYE UNA CELDA DE UNA TABLA DEL TIPO PARRAFO CENTRADO EN NEGRILLA
 * DADO LA FRASE, FILAS, COLUMNAS Y ROTACION
 */
protected static PdfPCell createCellh(String frase, int row, int col, int rot){
    PdfPCell cell = new PdfPCell(new Phrase(frase,ftexth));
    if(row > 0)cell.setRowspan(row);
    if(col > 0)cell.setColspan(col);
}
```

```

        if(rot > 0)cell.setRotation(rot);
        cell.setHorizontalAlignment(Element.ALIGN_CENTER);
        cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
        return cell;
    }

```

A continuación hemos implementado el método “createCellhl” que construye una celda de una tabla del tipo párrafo en negrita dada una frase, la fila, la columna y la rotación:

```

protected static PdfPCell createCellhl(String frase, int row, int col, int rot){
    PdfPCell cell = new PdfPCell(new Phrase(frase,ftext));
    if(row > 0)cell.setRowspan(row);
    if(col > 0)cell.setColspan(col);
    if(rot > 0)cell.setRotation(rot);
    cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
    cell.setHorizontalAlignment(Element.ALIGN_LEFT);
    return cell;
}

```

A continuación hemos implementado el método “createCellhr” que construye una celda de una tabla del tipo párrafo en negrita alineado a la derecha dada una frase, la fila, la columna y la rotación:

```

protected static PdfPCell createCellhr(String frase, int row, int col, int rot){
    PdfPCell cell = new PdfPCell(new Phrase(frase,ftext));
    if(row > 0)cell.setRowspan(row);
    if(col > 0)cell.setColspan(col);
    if(rot > 0)cell.setRotation(rot);
    cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
    cell.setHorizontalAlignment(Element.ALIGN_RIGHT);
    return cell;
}

```

A continuación hemos implementado el método “createCelld” que construye una celda de una tabla del tipo en negrita dada una frase, la fila, la columna y la rotación:

```

protected static PdfPCell createCelld(String frase, int row, int col, int rot){
    PdfPCell cell = new PdfPCell(new Phrase(frase,ftext));
    cell.setMinimumHeight(12);
    if(row > 0)cell.setRowspan(row);
    if(col > 0)cell.setColspan(col);
    if(rot > 0)cell.setRotation(rot);
    cell.setVerticalAlignment(Element.ALIGN_MIDDLE);
    return cell;
}

```

A continuación hemos implementado el método “openPdf” que abre un fichero en formato PDF si existe:

```

protected static void openPDF(String path) {
    try {
        File file = new File(path);
        if (file.exists()) {
            Desktop desktop = Desktop.getDesktop();
            desktop.open(file);
        } else {
            System.out.println("no existe");
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
};

```



```

table.addCell(createCellh("Peso freno", 0, 3, 0)); // PESO FRENO
table.addCell(createCellh("Porcentaje", 0, 2, 0)); // PORCENTAJE
table.addCell(createCellh("Peso freno", 0, 3, 0)); // PESO FRENO
table.addCell(createCellh("Porcentaje", 0, 2, 0)); // PORCENTAJE
table.addCell(createCellh("Vehiculos", 0, 2, 0)); // VEHICULOS
table.addCell(createCellh("Total", 2, 0, 0)); // TOTAL
table.addCell(createCellh("Disponible", 2, 0, 0)); // DISPONIBLE
table.addCell(createCellh("Necesario", 2, 0, 0)); // NECESARIO
table.addCell(createCellh("Vehiculos", 0, 2, 0)); // VEHICULOS
table.addCell(createCellh("Total", 2, 0, 0)); // TOTAL
table.addCell(createCellh("Disponible", 2, 0, 0)); // DISPONIBLE
table.addCell(createCellh("Necesario", 2, 0, 0)); // NECESARIO
table.addCell(createCellh("Sin freno", 2, 0, 0)); // SIN FRENO
table.addCell(createCellh("Regimen\nG", 2, 0, 0)); // REGIMEN G
table.addCell(createCellh("Regimen\nP", 2, 0, 0)); // REGIMEN P
table.addCell(createCellh("Total", 2, 0, 0)); // TOTAL
table.addCell(createCellh("Motores", 0, 0, 0)); // MOTORES
table.addCell(createCellh("Remolcado", 0, 0, 0)); // REMOLCADO
table.addCell(createCellh("Motores", 0, 0, 0)); // MOTORES
table.addCell(createCellh("Remolcado", 0, 0, 0)); // REMOLCADO

table.addCell(createCellId("", 0, 0, 0)); // DATOS
table.addCell(createCellId(bf.getValor_3(), 0, 0, 0)); // DATOS 3
table.addCell(createCellId(bf.getValor_4(), 0, 0, 0)); // DATOS 4
table.addCell(createCellId(bf.getValor_5(), 0, 0, 0)); // DATOS 5
table.addCell(createCellId(bf.getValor_6(), 0, 0, 0)); // DATOS 6
table.addCell(createCellId(bf.getValor_7(), 0, 0, 0)); // DATOS 7
table.addCell(createCellId(bf.getValor_8(), 0, 0, 0)); // DATOS 8
table.addCell(createCellId(bf.getValor_9(), 0, 0, 0)); // DATOS 9
table.addCell(createCellId(bf.getValor_10(), 0, 0, 0)); // DATOS 10
table.addCell(createCellId(bf.getValor_11(), 0, 0, 0)); // DATOS 11
table.addCell(createCellId(bf.getValor_12(), 0, 0, 0)); // DATOS 12
table.addCell(createCellId(bf.getValor_13(), 0, 0, 0)); // DATOS 13
table.addCell(createCellId(bf.getValor_14(), 0, 0, 0)); // DATOS 14
table.addCell(createCellId(bf.getValor_15(), 0, 0, 0)); // DATOS 15
table.addCell(createCellId(bf.getValor_16(), 0, 0, 0)); // DATOS 16
table.addCell(createCellId(bf.getValor_17(), 0, 0, 0)); // DATOS 17
table.addCell(createCellId(bf.getValor_18(), 0, 0, 0)); // DATOS 18
table.addCell(createCellId(bf.getValor_19(), 0, 0, 0)); // DATOS 19

for(int i = 0; i < 5; i++){
    for(int j = 0; j < 18; j++){
        table.addCell(createCellId("", 0, 0, 0)); // VACIO
    }
}

table.addCell(createCellh("Clase de prueba", 0, 0, 0)); // CLASE DE PRUEBA
table.addCell(createCellh("Agente de cola", 0, 5, 0)); // AGENTE DE COLA
table.addCell(createCellh("Agente de Cabeza", 0, 4, 0)); // AGENTE DE CABEZA
table.addCell(createCellh("Maquinista de cabeza", 0, 4, 0)); // MAQUINISTA DE CABEZA
table.addCell(createCellh("Jefe de circulacion", 0, 4, 0)); // JEFE DE CIRCULACION

for(int i = 0; i < 5; i++){
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 5, 0)); // VACIO
    table.addCell(createCellId("", 0, 4, 0)); // VACIO
    table.addCell(createCellId("", 0, 4, 0)); // VACIO
    table.addCell(createCellId("", 0, 4, 0)); // VACIO
}
//-----
PdfPCell cell = new PdfPCell(new Phrase("Observaciones:",ftexth)); // OBSERVACIONES
cell.setRowspan(5);
cell.setColspan(18);
cell.setMinimumHeight(100);
table.addCell(cell);

document.add(table);

document.close();
}catch(Exception e){
    e.printStackTrace();
}

```

```

        throw e;
    }
}

```

A continuación hemos generado el documento de la hoja de marcha recopilando todos los datos descritos en la clase HojaMarcha en el método “generarHojaMarcha”:

```

public static void generarHojaMarcha(HojaMarcha hm) throws Exception{
    try{
        Document document = getDocument(15);
        PdfWriter.getInstance(document, new
        FileOutputStream(FILE_HOJA_MARCHA)).setAtLeastPdfVersion(PdfWriter.VERSION_1_7);

        document.open();
        float[] columnWidths = {12, 12, 12, 12, 12, 16, 12, 12, 12, 12, 12, 12, 12, 12, 12,
12, 12, 14};

        PdfPTable table = new PdfPTable(columnWidths);
        table.getDefaultCell().setHorizontalAlignment(Element.ALIGN_CENTER);
        table.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
        table.setWidthPercentage(100);

        table.addCell(createCellt("FEVE", 6, 3, 0));
        table.addCell(createCellh("Hoja de marcha", 6, 2, 0));

        table.addCell(createCellh("Naturaleza:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_1(), 3, 2, 0));
        table.addCell(createCellh("Tren N°:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_2(), 3, 2, 0));
        table.addCell(createCellh("Tipo:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_3(), 3, 1, 0));

        table.addCell(createCellh("Origen:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_5(), 3, 3, 0));
        table.addCell(createCellh(hm.getValor_6(), 3, 1, 0));
        table.addCell(createCellh("Fecha:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_9(), 3, 1, 0));

        table.addCell(createCellh("Asimilado a:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_4(), 3, 7, 0));

        table.addCell(createCellh("Destino:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_7(), 3, 3, 0));
        table.addCell(createCellh(hm.getValor_8(), 3, 1, 0));
        table.addCell(createCellh("Fecha:", 3, 1, 0));
        table.addCell(createCellh(hm.getValor_9(), 3, 1, 0));

        table.addCell(createCellh("Marcha efectiva", 0, 7, 0));
        table.addCell(createCellh("Composicion inicial", 0, 5, 0));
        table.addCell(createCellh("Variacion de la composicion", 0, 8, 0));

        table.addCell(createCellh("Codigo\nEstacion", 2, 0, 0));
        table.addCell(createCellh("Horas", 0, 3, 0));
        table.addCell(createCellh("Motivo", 2, 0, 90));
        table.addCell(createCellh("Justificacion", 2, 2, 0));
        table.addCell(createCellh("Vehiculos tractores", 2, 3, 0));
        table.addCell(createCellh("N°", 0, 0, 0));
        table.addCell(createCellh(hm.getValor_11(), 0, 0, 0)); // LOCOMOTORA1
        table.addCell(createCellh("Codigo\nEstacion", 2, 0, 0));
        table.addCell(createCellh("Vehiculo", 0, 2, 0));
        table.addCell(createCellh("Motivo\nVariacion", 2, 0, 0));
        table.addCell(createCellh("Carga Remolcada", 0, 4, 0));
        table.addCell(createCellh("Llegada", 0, 0, 0));
        table.addCell(createCellh("Par", 0, 0, 0));
        table.addCell(createCellh("Salida", 0, 0, 0));
        table.addCell(createCellh("N°", 0, 0, 0));
        table.addCell(createCellh(hm.getValor_12(), 0, 0, 0)); // LOCOMOTORA2
        table.addCell(createCellh("Serie", 0, 0, 0));
        table.addCell(createCellh("Num", 0, 0, 0));
        table.addCell(createCellh("Mercancia", 0, 2, 0));
    }
}

```

```

table.addCell(createCellh("Peso total", 0, 0, 0));
table.addCell(createCellh("Acumulado", 0, 0, 0));

// linea 1 de datos
table.addCell(createCellId(hm.getValor_6(), 0, 0, 0));
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId(hm.getValor_10(), 0, 0, 0));
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
table.addCell(createCellh("Mat. Remolcado", 0, 2, 0));
table.addCell(createCellh("Carga remolcada", 0, 3, 0));
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO

// linea 2 de datos
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
table.addCell(createCellh("Serie", 0, 0, 0));
table.addCell(createCellh("Num", 0, 0, 0));
table.addCell(createCellh("Mercancia", 0, 2, 0));
table.addCell(createCellh("Peso", 0, 0, 0));
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO

// linea 3 de datos
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
boolean flag = false;
List<HojaMarchaVagon> lhm = hm.getVagones();
int vagones = lhm.size();
int limite = 0;
if(lhm.isEmpty()){
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
}else{
    HojaMarchaVagon hmv = lhm.get(limite++);
    table.addCell(createCellId(hmv.getValor_13(), 0, 0, 0));
    table.addCell(createCellId(hmv.getValor_14(), 0, 0, 0));
    table.addCell(createCellId(hmv.getValor_15(), 0, 2, 0));
    table.addCell(createCellId(hmv.getValor_16(), 0, 0, 0));
    flag = true;
}
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO

```

```

for(int i = 0; i < 18; i++){
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    if(flag && vagones > limite){
        HojaMarchaVagon hmv = lhm.get(limite++);
        table.addCell(createCellId(hmv.getValor_13(), 0, 0, 0));
        table.addCell(createCellId(hmv.getValor_14(), 0, 0, 0));
        table.addCell(createCellId(hmv.getValor_15(), 0, 2, 0));
        table.addCell(createCellId(hmv.getValor_16(), 0, 0, 0));
    }else{
        table.addCell(createCellId("", 0, 0, 0)); // VACIO
        table.addCell(createCellId("", 0, 0, 0)); // VACIO
        table.addCell(createCellId("", 0, 2, 0)); // VACIO
        table.addCell(createCellId("", 0, 0, 0)); // VACIO
    }

    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
}

table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
if(flag && vagones > limite){
    HojaMarchaVagon hmv = lhm.get(limite++);
    table.addCell(createCellId(hmv.getValor_13(), 0, 0, 0));
    table.addCell(createCellId(hmv.getValor_14(), 0, 0, 0));
    table.addCell(createCellId(hmv.getValor_15(), 0, 2, 0));
    table.addCell(createCellId(hmv.getValor_16(), 0, 0, 0));
}else{
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
}

table.addCell(createCellId("Agregado (1)    Diferido (2)    Diferido averia (3)", 0, 8,
0));

table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
if(flag && vagones > limite){
    HojaMarchaVagon hmv = lhm.get(limite++);
    table.addCell(createCellId(hmv.getValor_13(), 0, 0, 0));
    table.addCell(createCellId(hmv.getValor_14(), 0, 0, 0));
    table.addCell(createCellId(hmv.getValor_15(), 0, 2, 0));
    table.addCell(createCellId(hmv.getValor_16(), 0, 0, 0));
}else{
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
}
}

```

```

table.addCell(createCellh("Observaciones:", 6, 8, 0));

for(int i = 0; i < 4; i++){
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 0, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    if(flag && vagones > limite){
        HojaMarchaVagon hmv = lhm.get(limite++);
        table.addCell(createCellId(hmv.getValor_13(), 0, 0, 0));
        table.addCell(createCellId(hmv.getValor_14(), 0, 0, 0));
        table.addCell(createCellId(hmv.getValor_15(), 0, 2, 0));
        table.addCell(createCellId(hmv.getValor_16(), 0, 0, 0));
    }else{
        table.addCell(createCellId("", 0, 0, 0)); // VACIO
        table.addCell(createCellId("", 0, 0, 0)); // VACIO
        table.addCell(createCellId("", 0, 2, 0)); // VACIO
        table.addCell(createCellId("", 0, 0, 0)); // VACIO
    }
}

table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 0, 0)); // VACIO
table.addCell(createCellId("", 0, 2, 0)); // VACIO
table.addCell(createCellh("Total ...", 0, 4, 0));
table.addCell(createCellId(hm.getValor_17(), 0, 0, 0)); // TOTAL_PESO

table.addCell(createCellh("Personal del tren", 0, 12, 0));
table.addCell(createCellh("Jefe de circulacion\nde salida", 2, 4, 0));
table.addCell(createCellh("Jefe de circulacion\nde llegada", 2, 4, 0));
table.addCell(createCellh("Nº de C.F.", 0, 2, 0));
table.addCell(createCellh("Nombre y Apellidos", 0, 6, 0));
table.addCell(createCellh("de", 0, 2, 0));
table.addCell(createCellh("a", 0, 2, 0));

for(int i = 0; i < 6; i++){
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    table.addCell(createCellId("", 0, 6, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    table.addCell(createCellId("", 0, 2, 0)); // VACIO
    table.addCell(createCellId("", 0, 4, 0)); // VACIO
    table.addCell(createCellId("", 0, 4, 0)); // VACIO
}

document.add(table);

document.close();
}catch(Exception e){
    e.printStackTrace();
    throw e;
}
}

```

4.11 Clase TablaModelo:

En la implementación de la clase TablaModelo se han desarrollado dos métodos que nos van a ayudar a la hora de la edición de las tablas de las locomotoras y de los vagones que se van a introducir en la clase Ventana. A continuación se muestran dichos métodos y una breve explicación:

```
package zira;

import javax.swing.table.DefaultTableModel;

@SuppressWarnings({"rawtypes","unchecked"})
public class TablaModelo extends DefaultTableModel {

    private static final long serialVersionUID = 4331703533271487254L;
```

En primer lugar hemos implementado el método “isCellEditable” que permite la edición de un determinado registro a partir de sus índices de fila y columna:

```
public boolean isCellEditable(int row, int column) {
    if (column == 0)
        return true;

    return false;
}
```

A continuación hemos implementado el método “getColumnClass” que cambia el tipo de columna automáticamente si se ha seleccionado:

```
public Class getColumnClass(int columna) {
    if (columna == 0){
        return Boolean.class;
    }
    if (columna == 6){
        return Boolean.class;
    }
    if (columna == 8){
        return Boolean.class;
    }
    return Object.class;
}
}
```

4.12 Clase Ventana:

En la implementación de la clase Ventana se han desarrollado el main, en el que se pretende generar una ventana para recoger todos los datos que el usuario debe introducir para complementar la hoja de marcha referentes al trayecto y la composición del tren, y a partir de ellos calcular los valores del boletín de frenado y una vez que hemos comprobado que la composición cumple los requisitos exigidos por los porcentajes de freno automático y de estacionamiento y que no supera la carga máxima del trayecto, elaborar los dos documentos en formato PDF. A continuación voy a describir el código desarrollado para implementar la siguiente ventana:

VENTANA ESTACIONES

Naturaleza **Elegir Naturaleza**

Tren N° Marcha asimilación Tipo

Origen Código Nombre **VALIDAR**

Destino Código Nombre Boletin Frenado

Fecha / / Hora : Hoja Marcha

LOCOMOTORAS

Numero Veh_tractores

☐ Remolcada

Añadir

VAGONES

Serie Peso

Numero F. Aut. ☐ Aislado

Carga F. Est. ☐ Sin Estacionamiento

Tara Mercancia **Añadir Vagon**

Serie	Numero	Mercancia	Peso	F. Aut.	F. Est.

Total Peso 0

Para la implementación de la ventana anterior y el tratamiento de los datos introducidos por el usuario se ha implementado el siguiente código:

```
package zira;

import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.IOException;
import java.text.ParseException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.BorderFactory;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFormattedTextField;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.ScrollPaneConstants;
import javax.swing.SwingConstants;
import javax.swing.WindowConstants;
import javax.swing.text.MaskFormatter;

import zira.HojaMarcha.HojaMarchaVagon;
```

En primer lugar hemos declarado las variables para swing y las clases manejadoras de información:

```
public class Ventana extends JFrame {

    private static final long serialVersionUID = -580937980325160655L;
    /**
     * DECLARACION DE VARIABLES PARA SWING Y CLASES MANEJADORAS DE INFORMACION
     */
    private zira z;

    private TablaModelo modelo, modelo_vagon;

    private JButton jButton1;
    private JButton jButton2;
    private JButton aniadir_vag;
    private JButton btnModificarLoc;
    private JButton btnEliminarLoc;
    private JButton btnModificarVag;
    private JButton btnEliminarVag;
    private JButton impHMBButton;
    private JButton impBFBButton;

    private JTextField cod_destino;
    private JTextField cod_origen;
    private JTextField est_destino;
    private JTextField est_origen;
```

```

private JTextField in_m_asimil;
private JTextField in_carga;
private JTextField in_num_vag;
private JTextField in_numero;
private JTextField in_peso;
private JTextField in_serie_vag;
private JTextField in_tipo;
private JTextField in_total_peso;
private JTextField in_tren_num;
private JTextField in_tara;
private JTextField in_faut;
private JTextField in_fest;
private JTextField in_mercancia;

private JFormattedTextField fecha;
private JFormattedTextField hora;

private JCheckBox jCheckBox1;
private JCheckBox chckbxFreno;
private JCheckBox chckbxFrenoEstacinamiento;

private JComboBox<String> jComboBox1;

private JTable jTable1;
private JTable jTable2;

private JLabel jLabel1;
private JLabel jLabel10;
private JLabel jLabel11;
private JLabel jLabel13;
private JLabel jLabel14;
private JLabel jLabel2;
private JLabel jLabel3;
private JLabel jLabel4;
private JLabel jLabel5;
private JLabel jLabel6;
private JLabel jLabel7;
private JLabel jLabel8;
private JLabel jLabel9;
private JLabel Naturaleza;
private JLabel lblCarga;
private JLabel marcha_asimil;
private JLabel numero;
private JLabel tipo;
private JLabel tren_num;
private JLabel lblFecha;
private JLabel lblHora;
private JLabel lblTara;
private JLabel lblFest;
private JLabel lblFAut;
private JLabel lblMercancia;

private JPanel jPanel1;

private JScrollPane jScrollPane2;
private JScrollPane jScrollPane3;
private JScrollPane scroll2;

private Container contentPane;
private int BF;

```

A continuación hemos inicializado todos los componentes de la ventana, ubicándolo en una determinada posición y etiquetándolo:

```

@SuppressWarnings({ "unchecked", "rawtypes"})
private void initComponents() {

    /**
     * SE DECLARA CADA COMPONENTE DE LA VENTANA

```

```

*/
contentPane = getContentPane();
jPanel1 = new JPanel();

jLabel1 = new JLabel();
jLabel2 = new JLabel();
jLabel3 = new JLabel();
jLabel4 = new JLabel();
jLabel5 = new JLabel();
jLabel6 = new JLabel();
jLabel7 = new JLabel();
jLabel8 = new JLabel();
jLabel9 = new JLabel();
jLabel10 = new JLabel();
jLabel11 = new JLabel();
jLabel13 = new JLabel();
jLabel14 = new JLabel();
tipo = new JLabel();
marcha_asimil = new JLabel();
Naturaleza = new JLabel();
tren_num = new JLabel();
numero = new JLabel();
lblCarga = new JLabel();
lblFecha = new JLabel();
lblHora = new JLabel();
lblTara = new JLabel();
lblFAut = new JLabel();
lblFEst = new JLabel();
lblMercancia = new JLabel();

chckbxFreno = new JCheckBox();
chckbxFrenoEstacinamiento = new JCheckBox();

jTable2 = new JTable();

jButton1 = new JButton();
jButton2 = new JButton();
aniadir_vag = new JButton();
btnModificarLoc = new JButton();
btnEliminarLoc = new JButton();
btnEliminarVag = new JButton();
btnModificarVag = new JButton();
impHMBButton = new JButton("Hoja Marcha");
impBFButton = new JButton("Boletin Frenado");

jComboBox1 = new JComboBox();

jCheckBox1 = new JCheckBox();

cod_origen = new JTextField();
est_origen = new JTextField();
cod_destino = new JTextField();
est_destino = new JTextField();
in_m_asimil = new JTextField();
in_tipo = new JTextField();
in_tren_num = new JTextField();
in_numero = new JTextField();
in_serie_vag = new JTextField();
in_num_vag = new JTextField();
in_carga = new JTextField();
in_peso = new JTextField();
in_total_peso = new JTextField();
in_tara = new JTextField();
in_faut = new JTextField();
in_fest = new JTextField();
in_mercancia = new JTextField();

fecha = new JFormattedTextField();
hora = new JFormattedTextField();

scroll2 = new JScrollPane(jPanel1);

```

```

jScrollPane2 = new JScrollPane();
jScrollPane3 = new JScrollPane();
modelo = new TablaModelo();
modelo.setDataVector(new Object[][] {}, new String[] { "", "Veh_tractores" });
jTable1 = new JTable(modelo);

modelo_vagon = new TablaModelo();
modelo_vagon.setDataVector(new Object[][] {},
new String[] { "", "Serie", "Numero", "Mercancia", "Peso", "F. Aut.", "", "F. Est.", "" });
jTable2 = new JTable(modelo_vagon);

/**
 * SE UBICAN LAS COORDENADAS DE CADA COMPONENTE
 */
jLabel1.setBounds(89, 152, 106, 14);
jLabel2.setBounds(92, 197, 103, 14);
jLabel3.setBounds(9, 169, 70, 14);
jLabel4.setBounds(10, 214, 78, 14);
jLabel5.setBounds(214, 11, 170, 38);
jLabel6.setBounds(225, 152, 112, 14);
jLabel7.setBounds(225, 197, 112, 14);
jLabel9.setBounds(16, 453, 63, 35);
cod_origen.setBounds(89, 166, 106, 20);
est_origen.setBounds(225, 166, 179, 20);
cod_destino.setBounds(89, 211, 106, 20);
est_destino.setBounds(225, 211, 179, 20);
in_m_asimil.setBounds(225, 121, 179, 20);
in_tipo.setBounds(433, 121, 123, 20);
tipo.setBounds(433, 104, 112, 14);
jButton1.setBounds(414, 156, 142, 41);
marcha_asimil.setBounds(225, 104, 179, 14);
Naturaleza.setBounds(172, 76, 63, 14);
jComboBox1.setBounds(245, 73, 213, 20);
tren_num.setBounds(89, 104, 103, 14);
in_tren_num.setBounds(89, 121, 103, 20);
numero.setBounds(21, 337, 79, 14);
in_numero.setBounds(89, 334, 106, 20);
jCheckBox1.setBounds(72, 358, 123, 23);
jScrollPane2.setBounds(225, 336, 179, 91);
jButton2.setBounds(59, 388, 98, 36);
jLabel10.setBounds(225, 291, 95, 35);
jScrollPane3.setBounds(21, 651, 535, 228);
jLabel8.setBounds(9, 499, 70, 14);
in_serie_vag.setBounds(89, 496, 100, 20);
jLabel11.setBounds(9, 524, 70, 14);
in_num_vag.setBounds(89, 524, 100, 20);
lblCarga.setBounds(9, 555, 70, 14);
in_carga.setBounds(89, 555, 100, 20);
jLabel13.setBounds(214, 499, 70, 14);
in_peso.setBounds(294, 496, 100, 20);
aniadir_vag.setBounds(433, 591, 123, 38);
jLabel14.setBounds(374, 901, 84, 14);
in_total_peso.setBounds(464, 898, 92, 20);
fecha.setBounds(89, 260, 106, 20);
hora.setBounds(298, 260, 106, 20);
lblFecha.setBounds(9, 263, 46, 14);
lblHora.setBounds(225, 263, 46, 14);
btnModificarLoc.setBounds(447, 351, 109, 36);
btnEliminarLoc.setBounds(447, 391, 109, 36);
btnModificarVag.setBounds(21, 890, 89, 36);
btnEliminarVag.setBounds(120, 890, 89, 36);
lblTara.setBounds(9, 583, 46, 14);
in_tara.setBounds(89, 583, 100, 20);
chckbxFreno.setBounds(407, 520, 138, 23);
chckbxFrenoEstacionamiento.setBounds(407, 551, 149, 23);
lblFAut.setBounds(214, 524, 70, 14);
lblFEst.setBounds(214, 555, 70, 14);
in_faut.setBounds(294, 524, 100, 20);
in_fest.setBounds(294, 555, 100, 20);
lblMercancia.setBounds(214, 583, 70, 14);
in_mercancia.setBounds(294, 583, 100, 20);

```

```

impHMBButton.setBounds(414, 242, 142, 41);
impBFButton.setBounds(414, 200, 142, 41);

/**
 * SE ETIQUETAN LOS NOMBRES A LOS COMPONENTES QUE SE MOSTRARAN EN PANTALLA
 */
jButton2.setText("A\u00F1adir");
jLabel3.setText("Origen");
jLabel4.setText("Destino");
jLabel5.setText("VENTANA ESTACIONES");
jButton1.setText("VALIDAR");
tipo.setText("Tipo");
marcha_asimil.setText("Marcha asimilaci\u00F3n");
Naturaleza.setText("Naturaleza");
tren_num.setText("Tren N\u00BA");
jLabel1.setText("C\u00F3digo");
jLabel2.setText("C\u00F3digo");
jLabel6.setText("Nombre");
jLabel7.setText("Nombre");
jLabel9.setText("VAGONES");
numero.setText("Numero");
jCheckBox1.setText("Remolcada");
jLabel10.setText("LOCOMOTORAS");
jLabel8.setText("Serie");
jLabel11.setText("Numero");
lblCarga.setText("Carga");
jLabel13.setText("Peso");
aniadir_vag.setText("A\u00F1adir Vagon");
jLabel14.setText("Total Peso");
in_total_peso.setText("0");
lblFecha.setText("Fecha");
lblHora.setText("Hora");
lblTara.setText("Tara");
btnModificarLoc.setText("Modificar");
btnEliminarLoc.setText("Eliminar");
btnModificarVag.setText("Modificar");
btnEliminarVag.setText("Eliminar");
lblFAut.setText("F. Aut.");
lblFest.setText("F. Est.");
chckbxFreno.setText("Aislado");
chckbxFrenoEstacionamiento.setText("Sin Estacionamiento");
lblMercancia.setText("Mercanc\u00EDa");

/**
 * SE SETEAN OTROS VALORES A LOS COMPONENTES NECESARIOS ESTILO , EDICION ,
IMPLEMENTACION , ETC
 */

in_num_vag.setHorizontalAlignment(SwingConstants.RIGHT);
in_carga.setHorizontalAlignment(SwingConstants.RIGHT);
in_peso.setHorizontalAlignment(SwingConstants.RIGHT);
in_total_peso.setHorizontalAlignment(SwingConstants.RIGHT);
in_tara.setHorizontalAlignment(SwingConstants.RIGHT);
in_faut.setHorizontalAlignment(SwingConstants.RIGHT);
in_fest.setHorizontalAlignment(SwingConstants.RIGHT);

jComboBox1.setModel(new DefaultComboBoxModel(new String[] { "Elegir Naturaleza",
"Mercancias", "Cercanias",
"Regionales", "Trascantabrico", "Turisticos", "Servicios y Locomotoras
Aisladas",
"Mantenimiento de Infraestructura", "No clasificados", "Socorro y
talleres"}));
jComboBox1.setToolTipText("");

in_peso.setEditable(false);

jTable1.getColumnModel().getColumn(0).setPreferredWidth(2);
jTable1.getColumnModel().getColumn(0).setWidth(2);
jScrollPane2.setViewportViewView(jTable1);

jLabel10.setFont(new Font("Tahoma", Font.BOLD, 12));

```

```

jTable2.getColumnModel().getColumn(0).setPreferredWidth(2);
jTable2.getColumnModel().getColumn(0).setWidth(2);
jTable2.getColumnModel().getColumn(6).setPreferredWidth(2);
jTable2.getColumnModel().getColumn(6).setWidth(2);
jTable2.getColumnModel().getColumn(8).setPreferredWidth(2);
jTable2.getColumnModel().getColumn(8).setWidth(2);
jScrollPane3.setViewportViewView(jTable2);

btnModificarVag.setVisible(false);
btnEliminarVag.setVisible(false);
btnModificarLoc.setVisible(false);
btnEliminarLoc.setVisible(false);

try {
    MaskFormatter dateMask = new MaskFormatter("##/##/####");
    dateMask.install(fecha);
} catch (ParseException ex) {
}

try {
    MaskFormatter dateMask = new MaskFormatter("##:##");
    dateMask.install(hora);
} catch (ParseException ex) {
}

fecha.setColumns(10);
hora.setColumns(5);

jPanel1.setBorder(BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
in_tara.setEditable(false);
impBFButton.setEnabled(false);
impHMBButton.setEnabled(false);

jPanel1.setPreferredSize(new Dimension(550,1000));
scroll2.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
scroll2.setPreferredSize(new Dimension(600, 1000));
scroll2.setViewportViewView(jPanel1);

```

A continuación hemos implementado los eventos según las necesidades y los requerimientos establecidos:

```

in_tren_num.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent event) {
        preActionPerformed(event);
    }
});
in_m_asimil.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent event) {
        preActionPerformed(event);
    }
});
cod_origen.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent event) {
        preActionPerformed(event);
    }
});
cod_destino.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent event) {
        preActionPerformed(event);
    }
});

jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

```

```

    }
});
in_serie_vag.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent e) {
        in_serie_vagActionPerformed(e);
    }
});
in_num_vag.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent e) {
        in_serie_vagActionPerformed(e);
    }
});
in_carga.addKeyListener(new KeyAdapter() {
    @Override
    public void keyReleased(KeyEvent e) {
        in_carga_vagActionPerformed();
    }
});

jTable1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        int row = jTable1.getSelectedRow();
        String valor = "";
        if(jTable1.getValueAt(row, 0).toString().equals("true")){
            valor = jTable1.getValueAt(row, 1).toString();
            btnModificarLoc.setVisible(true);
            btnEliminarLoc.setVisible(true);
        }else{
            btnModificarLoc.setVisible(false);
            btnEliminarLoc.setVisible(false);
        }
        in_numero.setText(valor);
    }
});

jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jTable2.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        int row = jTable2.getSelectedRow();
        String serie_vagon = "";
        String numero_vagon = "";
        String mercancia = "";
        String peso = "";
        String faut = "";
        String faut_b = "";
        String fest = "";
        String fest_b = "";
        if(jTable2.getValueAt(row, 0).toString().equals("true")){
            serie_vagon = jTable2.getValueAt(row, 1).toString();
            numero_vagon = jTable2.getValueAt(row, 2).toString();
            mercancia = jTable2.getValueAt(row, 3).toString();
            peso = jTable2.getValueAt(row, 4).toString();
            faut = jTable2.getValueAt(row, 5).toString();
            faut_b = jTable2.getValueAt(row, 6).toString();
            fest = jTable2.getValueAt(row, 7).toString();
            fest_b = jTable2.getValueAt(row, 8).toString();
            btnModificarVag.setVisible(true);
            btnEliminarVag.setVisible(true);
        }else{
            btnModificarVag.setVisible(false);
            btnEliminarVag.setVisible(false);
        }
    }
});

```

```

        }
        in_serie_vag.setText(serie_vagon);
        in_num_vag.setText(numero_vagon);
        in_mercancia.setText(mercancia);
        in_peso.setText(peso);
        in_faut.setText(faut);
        in_fest.setText(fest);
        chckbxFreno.setSelected((faut_b.equals("true")));
        chckbxFrenoEstacinamiento.setSelected((fest_b.equals("true")));
    }
});

aniadir_vag.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        aniadir_vagActionPerformed(evt);
    }
});

btnModificarLoc.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        editar_Locomotora(evt);
    }
});

btnModificarVag.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        editar_Vagon(evt);
    }
});

btnEliminarLoc.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        eliminar_Locomotora(evt);
    }
});

btnEliminarVag.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        eliminar_Vagon(evt);
    }
});

chckbxFreno.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        in_carga_vagActionPerformed();
    }
});

chckbxFrenoEstacinamiento.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        in_carga_vagActionPerformed();
    }
});

impHMButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        imp_HojaMarchaActionPerformed();
    }
});

impBFButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        imp_BoletinFrenadoActionPerformed();
    }
});

```

A continuación hemos añadido todos los componentes a la ventana:

```
jPanel1.setLayout(null);
jPanel1.add(jLabel3);
jPanel1.add(cod_origen);
jPanel1.add(jLabel1);
jPanel1.add(jLabel6);
jPanel1.add(est_origen);
jPanel1.add(jLabel4);
jPanel1.add(cod_destino);
jPanel1.add(jLabel2);
jPanel1.add(est_destino);
jPanel1.add(jLabel7);
jPanel1.add(jButton1);
jPanel1.add(Naturaleza);
jPanel1.add(jComboBox1);
jPanel1.add(marcha_asimil);
jPanel1.add(in_m_asimil);
jPanel1.add(tren_num);
jPanel1.add(tipo);
jPanel1.add(in_tipo);
jPanel1.add(in_tren_num);
jPanel1.add(jLabel5);
jPanel1.add(jScrollPane2);
jPanel1.add(jLabel8);
jPanel1.add(in_serie_vag);
jPanel1.add(jLabel11);
jPanel1.add(numero);
jPanel1.add(in_numero);
jPanel1.add(jCheckBox1);
jPanel1.add(jButton2);
jPanel1.add(jScrollPane3);
jPanel1.add(in_num_vag);
jPanel1.add(lblCarga);
jPanel1.add(in_carga);
jPanel1.add(jLabel13);
jPanel1.add(aniadir_vag);
jPanel1.add(in_peso);
jPanel1.add(jLabel14);
jPanel1.add(in_total_peso);
jPanel1.add(jLabel10);
jPanel1.add(jLabel9);
jPanel1.add(fecha);
jPanel1.add(hora);
jPanel1.add(lblFecha);
jPanel1.add(lblHora);
jPanel1.add(btnModificarLoc);
jPanel1.add(btnEliminarLoc);
jPanel1.add(btnModificarVag);
jPanel1.add(btnEliminarVag);
jPanel1.add(lblTara);
jPanel1.add(lblFAut);
jPanel1.add(lblFEst);
jPanel1.add(in_tara);
jPanel1.add(in_faut);
jPanel1.add(in_fest);
jPanel1.add(chckbxFreno);
jPanel1.add(chckbxFrenoEstacinamiento);
jPanel1.add(lblMercancia);
jPanel1.add(in_mercancia);
jPanel1.add(impHMBButton);
jPanel1.add(impBFBButton);

contentPane.add(scroll2, BorderLayout.CENTER);
pack();
}
```

A continuación hemos implementado una función que calcula directamente el tipo según el número de tren o marcha de asimilación:

```
private void preActionPerformed(KeyEvent event) {
    String c_origen;
    String c_destino;
    String e_origen, e_destino;
    String m_asimil;
    int tipo = 0;
    String tren_n;
    c_origen = cod_origen.getText();
    c_destino = cod_destino.getText();
    e_origen = z.Estacionporcodigo(c_origen);
    est_origen.setText(e_origen);
    e_destino = z.Estacionporcodigo(c_destino);
    est_destino.setText(e_destino);

    m_asimil = in_m_asimil.getText();
    tren_n = in_tren_num.getText();

    if ((m_asimil.isEmpty()) && (tren_n.charAt(0) == '7')) {
        m_asimil = tren_n;
    }
    if (!m_asimil.isEmpty()) {
        tipo = z.calculartipo(Integer.parseInt(m_asimil));
        if (tipo < 0) {
            in_tipo.setText("");
        } else {
            in_tipo.setText(String.valueOf(tipo));
        }
    }
}
```

A continuación hemos implementado una función para añadir la locomotora a la tabla de locomotoras o la tabla de vagones si circula remolcada:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
String num_locomotora = in_numero.getText();
int peso_total = 0;
Locomotora loc = z.buscarlocomotora(Integer.parseInt(num_locomotora));
if ((loc.getNum_l() != -1) && (!jCheckBox1.isSelected()) && (modelo.getRowCount() < 2)) {
    // Agregar nueva fila
    Object[] newRow = {false, String.valueOf(loc.getNum_l())};
    modelo.addRow(newRow);
}
if ((loc.getNum_l() != -1) && (jCheckBox1.isSelected())) {
    // Agregar nueva fila pero a los vagones
    Object[] newRow = { false, loc.getSerie_l(), loc.getNum_l(), "LOCOMOTORA",
loc.getPeso(), loc.getFreno(), false, loc.getEst_l(), false};
    modelo_vagon.addRow(newRow);
    // Actualizar el peso total de los vagones
    peso_total = Integer.parseInt(in_total_peso.getText());
    peso_total += loc.getPeso();
    in_total_peso.setText(String.valueOf(peso_total));
}
}
```

A continuación se ha implementado la función que permite modificar el número de la locomotora introducida por el usuario:

```
private void editar_Locomotora(java.awt.event.ActionEvent evt){
    int rows = jTable1.getRowCount();
    for (int i = 0; i < rows; i++) {
        if(jTable1.getValueAt(i,0).toString().equals("true")){
            String valor = in_numero.getText();
            Locomotora loc = z.buscarlocomotora(Integer.parseInt(valor));
            if ((loc.getNum_l() != -1) && (!jCheckBox1.isSelected()) &&
```

```

(modelo.getRowCount() < 2)) {
    jTable1.setValueAt(valor, i, 1);
    jTable1.setValueAt(false, i, 0);
}
if ((loc.getNum_l() != -1) && (jCheckBox1.isSelected())) {
    // Agregar nueva fila pero a los vagones
    Object[] newRow = { false, loc.getSerie_l(), loc.getNum_l(),
"LOCOMOTORA", loc.getPeso(), loc.getFreno(), false, loc.getEst_l(), false};
    modelo_vagon.addRow(newRow);
    // Actualizar el peso total de los vagones
    int peso_total = Integer.parseInt(in_total_peso.getText());
    peso_total += loc.getPeso();
    in_total_peso.setText(String.valueOf(peso_total));
    modelo.removeRow(i);
}
}
}
btnModificarLoc.setVisible(false);
btnEliminarLoc.setVisible(false);
}
}

```

A continuación hemos implementado la función que permite eliminar una locomotora introducida por el usuario:

```

private void eliminar_Locomotora(java.awt.event.ActionEvent evt){
    int rows = jTable1.getRowCount();
    for (int i = rows-1; i >= 0; i--) {
        if(jTable1.getValueAt(i,0).toString().equals("true")){
            modelo.removeRow(i);
        }
    }
    btnModificarLoc.setVisible(false);
    btnEliminarLoc.setVisible(false);
}
}

```

A continuación se ha implementado la función que calcula los valores del peso freno automático y del peso freno de estacionamiento de cada vagón según las características de su serie y la carga que lleva remolcada:

```

private void in_carga_vagActionPerformed() {
    try {
        String carga = in_carga.getText();
        if(carga != null && (!carga.equals(""))){
            int numero_vagon = Integer.parseInt(in_num_vag.getText());
            String serie_vagon = in_serie_vag.getText();
            Vagon v = new Vagon();
            v = z.buscarvagon(serie_vagon, numero_vagon);
            if(v.getNumero() != -1){
                int mer = Integer.parseInt(carga);
                int tara = v.getTara();
                int max = z.cantMaxVagones(serie_vagon);
                int faut = mer + tara;
                int fest = v.getEst_v();
                if(faut > 62){
                    JOptionPane.showMessageDialog(this, "Error. CARGA
MAXIMA SUPERADA");
                    in_faut.setText("");
                    in_fest.setText("");
                    in_carga.setText("");
                }else{
                    switch (max) {
                        case 62:
                            if(((max-tara)/2) > mer){
                                faut =
                                =
                            }
                            if(((max-tara)/2) <= mer){
                                faut =
                                =
                            }
                        }
                    }
                }
            }
            v.getFreno_vacio();
        }
    }
}

```

```

v.getFreno_cargado());
        }
        break;
    case 52:
        if(faut > 52) faut = 52;
        break;
    default:
        faut = 0;
        break;
    }
    if(chckbxFreno.isSelected()) faut = 0;
    if(chckbxFrenoEstacionamiento.isSelected()) fest = 0;
    in_faut.setText("" + faut);
    in_fest.setText("" + fest);
    in_peso.setText("" + (mer + tara));
    }
    }else{
        in_faut.setText("");
        in_fest.setText("");
    }
    }
} catch (NumberFormatException e) {
    in_faut.setText("");
    in_fest.setText("");
    in_carga.setText("");
}
}
}

```

A continuación hemos implementado una función para añadir los valores de los pesos freno de cada vagón:

```

private void in_serie_vagActionPerformed(KeyEvent event) {
    try {
        int numero_vagon = Integer.parseInt(in_num_vag.getText());
        String serie_vagon = in_serie_vag.getText();
        Vagon v = new Vagon();
        v = z.buscarvagon(serie_vagon, numero_vagon);
        if(v.getNumero() != -1){
            in_tara.setText("" + v.getTara());
            in_fest.setText("" + v.getEst_v());
        }else{
            in_tara.setText("");
            in_fest.setText("");
        }
    }
    } catch (NumberFormatException e) {
        in_tara.setText("");
        in_fest.setText("");
    }
}
}

```

A continuación hemos implementado una función para añadir los vagones con sus respectivos cálculos a la tabla:

```

private void aniadir_vagActionPerformed(java.awt.event.ActionEvent evt)
    try {
        int numero_vagon = Integer.parseInt(in_num_vag.getText());
        String serie_vagon = in_serie_vag.getText();
        String mercancia = in_mercancia.getText();
        int peso = Integer.parseInt(in_peso.getText());
        int peso_total = 0;
        Vagon v = new Vagon();
        v = z.buscarvagon(serie_vagon, numero_vagon);
        String tipo = in_tipo.getText();
        if(tipo.trim().equals("")){
            JOptionPane.showMessageDialog(this, "Error. ES NECESARIO INGRESAR
NRO DE TREN Y/O MARCHA DE ASIMILACION PARA PODER AGREGAR VAGONES");
        }
        if (v.getNumero() != -1 && (modelo_vagon.getRowCount() < z.cantidadVagones(tipo)))
    {

```

```

        try {
            boolean fest_B = chkcbxFrenoEstacinamiento.isSelected();
            boolean faut_B = chkcbxFreno.isSelected();
            int faut = Integer.parseInt(in_faut.getText());
            int fest = Integer.parseInt(in_fest.getText());
            Object[] newRow = { false, serie_vagon, numero_vagon,
mercancia, peso, faut, faut_B, fest, fest_B };

            modelo_vagon.addRow(newRow);
            peso_total = Integer.parseInt(in_total_peso.getText());
            peso_total += peso;
            in_total_peso.setText(String.valueOf(peso_total));
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(this, "Error. CALCULO DE
FRENOS");
        }
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, "PROBLEMA AGREGANDO VAGON");
}
}
}

```

A continuación se ha implementado la función que permite editar la serie, el número, la carga o la mercancía que transporta un vagón:

```

private void editar_Vagon(java.awt.event.ActionEvent evt){
    int rows = jTable2.getRowCount();
    for (int i = 0; i < rows; i++) {
        if(jTable2.getValueAt(i,0).toString().equals("true")){
            int numero_vagon = Integer.parseInt(in_num_vag.getText());
            String serie_vagon = in_serie_vag.getText();
            String mercancia = in_mercancia.getText();
            int peso = Integer.parseInt(in_peso.getText());
            int peso_total = 0;
            Vagon v = new Vagon();
            v = z.buscarvagon(serie_vagon, numero_vagon);
            if (v.getNumero() != -1) {
                peso_total = Integer.parseInt(in_total_peso.getText());
                int peso_resto = Integer.parseInt(jTable2.getValueAt(i,
4).toString());

                peso_total -= peso_resto;

                boolean fest_B = chkcbxFrenoEstacinamiento.isSelected();
                boolean faut_B = chkcbxFreno.isSelected();
                int faut = Integer.parseInt(in_faut.getText());
                int fest = Integer.parseInt(in_fest.getText());

                jTable2.setValueAt(serie_vagon, i, 1);
                jTable2.setValueAt(numero_vagon, i, 2);
                jTable2.setValueAt(mercancia, i, 3);
                jTable2.setValueAt(peso, i, 4);
                jTable2.setValueAt(faut, i, 5);
                jTable2.setValueAt(faut_B, i, 6);
                jTable2.setValueAt(fest, i, 7);
                jTable2.setValueAt(fest_B, i, 8);
                jTable2.setValueAt(false, i, 0);
                peso_total += peso;
                in_total_peso.setText(String.valueOf(peso_total));
            }
        }
    }
    btnModificarVag.setVisible(false);
    btnEliminarVag.setVisible(false);
}
}

```

A continuación hemos implementado la función que permite eliminar un vagón introducido por el usuario:

```
private void eliminar_Vagon(java.awt.event.ActionEvent evt){
    int rows = jTable2.getRowCount();
    for (int i = rows-1; i >= 0; i--) {
        if(jTable2.getValueAt(i,0).toString().equals("true")){
            int peso_total = Integer.parseInt(in_total_peso.getText());
            int peso_resto = Integer.parseInt(jTable2.getValueAt(i, 4).toString());
            peso_total -= peso_resto;
            in_total_peso.setText(String.valueOf(peso_total));
            modelo_vagon.removeRow(i);
        }
    }
    btnModificarVag.setVisible(false);
    btnEliminarVag.setVisible(false);
}
```

A continuación hemos implementado la función que va a cumplimentar la plantilla que teníamos del documento Hoja de Marcha con los datos introducidos por el usuario:

```
private void generarReportesHM() throws Exception{
    HojaMarcha hm = new HojaMarcha();

    try{

        hm.setValor_1(jComboBox1.getSelectedItem().toString());
        hm.setValor_2(in_tren_num.getText());
        hm.setValor_3(in_tipo.getText());
        hm.setValor_4(in_m_asimil.getText());
        hm.setValor_5(est_origen.getText());
        hm.setValor_6(cod_origen.getText());
        hm.setValor_7(est_destino.getText());
        hm.setValor_8(cod_destino.getText());
        hm.setValor_9(fecha.getText());
        hm.setValor_10(hora.getText());
        int loc = modelo.getRowCount();
        if(loc >= 1){
            hm.setValor_11(modelo.getValueAt(0, 1).toString());
        }
        if(loc > 1){
            hm.setValor_12(modelo.getValueAt(1, 1).toString());
        }
        int vagones = modelo_vagon.getRowCount();
        int total_peso = 0;
        List<HojaMarchaVagon> lhmv = new ArrayList<HojaMarcha.HojaMarchaVagon>();
        for (int i = 0; i < vagones; i++) {
            HojaMarchaVagon hmv = hm.new HojaMarchaVagon();
            hmv.setValor_13(modelo_vagon.getValueAt(i, 1).toString());
            hmv.setValor_14(modelo_vagon.getValueAt(i, 2).toString());
            hmv.setValor_15(modelo_vagon.getValueAt(i, 3).toString());
            hmv.setValor_16(modelo_vagon.getValueAt(i, 4).toString());
            try{
                total_peso += Integer.parseInt(hmv.getValor_16());
            }catch (Exception e) {
            }
            lhmv.add(hmv);
        }
        hm.setValor_17(" " + total_peso);
        hm.setVagones(lhmv);
        creacionPDF.generarHojaMarcha(hm);
    }catch(Exception e){
        JOptionPane.showMessageDialog(this, "Error. PROBLEMAS AL GENERAR HOJA DE
        MARCHA " + e.getMessage());
        BF=-5;
    }
    finally{
        if (BF==0){
```

```

        System.out.println("Hemos entrado en BF 0");
        creacionPDF.generarHojaMarcha(hm);
    }
}
}

```

A continuación hemos implementado la función que va a cumplimentar la plantilla que teníamos del documento Boletín de Frenado con los datos introducidos por el usuario si los cálculos realizados cumplen los requisitos ya mencionados de los porcentajes de frenado y la carga máxima:

```

private int generarReportesBF() throws Exception{
    BoletinFrenado bf = new BoletinFrenado();
    try{

        bf.setValor_1(in_tren_num.getText());
        bf.setValor_2(fecha.getText());
        int vagones = modelo_vagon.getRowCount();
        bf.setValor_3("" + vagones);
        bf.setValor_5(in_total_peso.getText());
        Integer m1 = 0, r1 = 0, t1 = 0;
        double d1 = 0.0;
        Integer m2 = 0, r2 = 0, t2 = 0;
        double d2 = 0.0;
        Integer va = 0, vna = 0;
        for (int i = 0; i < vagones; i++) {
            String temp = modelo_vagon.getValueAt(i, 3).toString();
            if(temp.equalsIgnoreCase("LOCOMOTORA")){
                m1 += (Integer)modelo_vagon.getValueAt(i, 5);
                m2 += (Integer)modelo_vagon.getValueAt(i, 7);
            }else{
                r1 += (Integer)modelo_vagon.getValueAt(i, 5);
                r2 += (Integer)modelo_vagon.getValueAt(i, 7);
            }
        }

        if(((Integer)modelo_vagon.getValueAt(i, 5))==0){
            va++;
        }else{
            vna++;
        }
    }
    bf.setValor_6(m1.toString());
    bf.setValor_7(r1.toString());
    t1 = m1 + r1;
    bf.setValor_8(t1.toString());
    if(r1 > 0)d1 = (t1*100/Integer.parseInt((bf.getValor_5())));
    bf.setValor_9("" + d1);
    bf.setValor_10("" + z.getpeso_freno());

    bf.setValor_11(m2.toString());
    bf.setValor_12(r2.toString());
    t2 = m2 + r2;
    bf.setValor_13(t2.toString());
    if(r2 > 0)d2 = (t2*100/Integer.parseInt((bf.getValor_5())));
    bf.setValor_14("" + d2);
    bf.setValor_15("" + z.getestacionamiento());
    va *= 4;
    bf.setValor_16("" + va);
    vna *= 4;
    bf.setValor_18("" + vna);
    vagones *= 4;
    bf.setValor_19("" + vagones);
    /**/
    int v9 = Integer.parseInt(bf.getValor_9());
    int v10 = Integer.parseInt(bf.getValor_10());
    System.out.println("Valor de v9 "+v9+" valor v10"+v10);
    int v14 = Integer.parseInt(bf.getValor_14());
    int v15 = Integer.parseInt(bf.getValor_15());
    System.out.println("Valor de v14 "+v14+" valor v15"+v15);
}

```

```

int v5 = Integer.parseInt(bf.getValor_5());

if((v9 > v10) && (v14 > v15) && Integer.parseInt(bf.getValor_5()) < z.carga()){

    creacionPDF.generarBoletinFrenado(bf);
    BF = 0;
}else if(v9 < v10){
    BF = -1;
}
else if(v14 < v15){
    BF = -2;
    }else if(v5 > z.getcarga()){
        BF = -3;
    }
}

```

A continuación se ha implementado la función que visualiza el mensaje correspondiente dependiendo si los valores son válidos o por el contrario no cumple alguno de las condiciones:

```

private void generarReportes(){
    try{
        BF=generarReportesBF();
        if(BF == 0){
            generarReportesHM();
            JOptionPane.showMessageDialog(this, "REPORTES GENERADOS CON
EXITO!");
        }
        else if(BF == -1){
            JOptionPane.showMessageDialog(this, "La composicion no cumple con el
porcentaje de freno automatico");
        }else if(BF == -2){
            JOptionPane.showMessageDialog(this, "La composicion no cumple con el
porcentaje de freno de estacionamiento");
        }else if(BF == -3){
            JOptionPane.showMessageDialog(this, "La composicion supera la carga
maxima del trayecto");
        }
    }catch(Exception e){
        JOptionPane.showMessageDialog(this, "PROBLEMAS AL GENERAR REPORTES");
    } }

```

A continuación se ha implementado la función que crea el archivo “BOLETIN_FRENADO” y lo abre:

```

private void imp_BoletinFrenadoActionPerformed() {
    creacionPDF.openPDF(creacionPDF.FILE_BOLETIN_FRENADO);
}

```

A continuación se ha implementado la función que crea el archivo “HOJA_MARCHA” y lo abre:

```

private void imp_HojaMarchaActionPerformed() {
    creacionPDF.openPDF(creacionPDF.FILE_HOJA_MARCHA);
}

```

A continuación hemos creado una función que borra los datos del encabezamiento de la hoja de marcha:

```

public void limpiar() {
    cod_origen.setText("");
    est_origen.setText("");
    cod_destino.setText("");
    est_destino.setText("");
    in_m_asimil.setText("");
    in_tipo.setText("");
    in_tren_num.setText("");
}

```

La última función que se ha implementado realiza las comprobaciones y el tratamiento de errores de los datos introducidos en el encabezamiento de la hoja de marcha para poder crear los dos archivos en formato pdf y permitiendo su visualización:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    String c_origen;
    String c_destino;
    String e_origen, e_destino;
    String m_asimil;
    int tipo = 0;
    int asimil;
    int numero_l;
    String tren_n;
    c_origen = cod_origen.getText();
    c_destino = cod_destino.getText();
    if ((c_origen.isEmpty()) || (c_destino.isEmpty())) {
        e_origen = est_origen.getText();
        e_destino = est_destino.getText();
        if ((e_origen.isEmpty()) || (e_destino.isEmpty())) {
            tipo = -2;
        } else {
            c_origen = z.Estacionpornombre(e_origen);
            c_destino = z.Estacionpornombre(e_destino);
            cod_origen.setText(c_origen);
            cod_destino.setText(c_destino);
        }
    } else {
        e_origen = z.Estacionporcodigo(c_origen);
        e_destino = z.Estacionporcodigo(c_destino);
        est_origen.setText(e_origen);
        est_destino.setText(e_destino);
    }
    if (tipo == 0) { // Puedo asegurar que las estaciones estan bien
        m_asimil = in_m_asimil.getText();
        tren_n = in_tren_num.getText();
        /* Marcha o numero de tren */

        if ((m_asimil.isEmpty()) && (tren_n.charAt(0) == '7')) {
            m_asimil = tren_n;
            in_m_asimil.setText("");
        }
        if (!m_asimil.isEmpty()) {
            tipo = z.calculartipo(Integer.parseInt(m_asimil));
            asimil = z.calcularmarcha_asimil(Integer.parseInt(m_asimil));

            in_tipo.setText(String.valueOf(tipo));
            in_m_asimil.setText(String.valueOf(m_asimil));
            if (tipo >= 0) {
                numero_l = Integer.parseInt(in_numero.getText());
                tipo = z.revisor(tipo, asimil, c_origen, c_destino, numero_l);
            } else {
                tipo = -1; // Quiere decir que los parametros estan mal
            }
        }
    }

    if (tipo == -1) {
        JOptionPane.showMessageDialog(this, "Error. PARAMETROS INCORRECTOS");
        limpiar();
    } else if (tipo == -2) {
        JOptionPane.showMessageDialog(this, "Error. INTRODUZCA ESTACIONES ORIGEN Y DESTINO");
        limpiar();
    } else if (tipo == -3) {
        JOptionPane.showMessageDialog(this,
            "Error. ESTACIONES INCORRECTA. NO PERTENECEN AL TREN O MARCHA DE ASIMILACION");
    }
}
```

```

        limpiar();
    } else if (tipo == -4) {
        JOptionPane.showMessageDialog(this, "Error. NO SE ENCUENTRA EL NUMERO DE
TREN");
        limpiar();
    } else {
        this.generarReportes();
        if(BF == 0){
            impBFButton.setEnabled(true);
            impHMButton.setEnabled(true);
        }
    }
}
}

```

Y la función principal lo único que va a realizar es visualizar la función ventana con todos sus componentes para que el usuario pueda introducir los datos:

```

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                new Ventana().setVisible(true);
            } catch (IOException ex) {
                Logger.getLogger(Ventana.class.getName()).log(Level.SEVERE,
null, ex);
            }
        }
    });
}
}

```

5.0 CONCLUSIONES Y FUTUROS CAMBIOS

Vamos a recordar que el proyecto tenía como objetivo principal elaborar los dos documentos que son necesarios para que los trenes de mercancías puedan circular por la red ferroviaria de ancho métrico española. Los documentos son la HOJA DE MARCHA, en la que figuran los datos correspondientes al trayecto a realizar por el tren y su composición de vehículos tractores y vagones remolcados. A partir de esta hoja de marcha se calcula el BOLETIN DE FRENADO, que es el otro documento en el que calculamos tres valores y comprobaremos que cumplen los requisitos establecidos para ese trayecto por el Reglamento de Circulación de Trenes y las Consignas que regulan el frenado de los trenes compuestos por locomotoras y vagones remolcados.

El primero de ellos va a ser la carga máxima, según el tipo de locomotoras y el trayecto a realizar la composición del tren no podrá superar dicho valor que viene regulado por consigna.

El segundo valor a va ser el porcentaje de freno automático, que viene determinado por la composición de los vagones remolcados y que debe ser superior al que determina el Reglamento de Circulación de Trenes según el trayecto.

El tercer valor a va ser el porcentaje de freno de estacionamiento, que viene determinado también por la composición de los vagones remolcados y que debe ser superior al que determina el Reglamento de Circulación de Trenes según el trayecto que deba realizar el tren.

Queda pendiente la depuración de errores de este proyecto para los trenes especiales con marcha de asimilación ya que por tiempo no ha sido posible lograr dicho objetivo, así como otr tipo de errores que pudieran surgir de las pruebas realizadas con el mismo.

Por otro lado hay que recordar que el presente proyecto se ha realizado bajo el Reglamento de Circulación de Trenes de FEVE, que es el que actualmente está en vigor, pero está previsto que el año que viene se unifique con el Reglamento de Circulación del Administrador de Infraestructuras Ferroviario (ADIF) por lo que previsiblemente alguno de los requisitos y/o criterios pueden cambiar sensiblemente.

ANEXO I: CONSIGNA SERIE C Nº 13/2016

ABREVIATURA DE LAS ESTACIONES DE ADIF RAM A EFECTOS DE LA DESIGNACIÓN DE LOS TRENES SIN MARCHA DETERMINADA

1.- OBJETO.

- Actualizar las abreviaturas de las estaciones según dispone el artículo 1|07|00| del R.C.T. vigente por:
 - Reclasificar a apeaderos las estaciones de Foz, Moeche y Ponte Mera.

2.- LISTADO DE LAS ABREVIATURAS DE LAS ESTACIONES.

Estación	Abreviatura de las estaciones de					
	Asturias	Bizkaia	Cantabria	C. y León	Galicia	Murcia
Ablaña	AA					
Aboño Mercancías	AB					
Abrevadero						AO
Adarzo			AD			
Aguda	GD					
Aranguren		AG				
Arija				AJ		
Áriz		AZ				
Arriondas	AR					
Astillero			AS			
Avilés	IM					
Balmaseda		BM				
Barreda			BA			
Basurto Hospital		BS				
Bendición	BE					
Beranga			BG			
Bercedo Montija				BC		
Bezana			BZ			
Bilbao La Concordia		BI				
Boñar				BO		
Burela					BU	
Cabañaquinta	CB					
Cabezón de la Sal			CS			
Candás	CD					
Carancos	CC					
Carbayín	CA					
Cartagena Plaza Bastarreche						CN
Casar de Periedo			CP			
Cerdido					CE	
Cistierna				CI		
Collanzo	CO					

Estación	Abreviatura de las estaciones de					
	Asturias	Bizkaia	Cantabria	C. y León	Galicia	Murcia
Colloto	CT					
Cudillero	CU					
El Berrón	EB					
El Entrego La Oscura	EE					
El Estrecho						ET
Espinosa de los Monteros				EM		
Ferrol					FR	
Figaredo	FI					
Fuso de la Reina	FU					
Gama			GA			
Gibaja			GJ			
Gijón Sanz Crespo	GI					
Grado	GR					
Guardo				GU		
Heras			HE			
Hoz de Anero			HA			
Infiesto	IN					
Irauregi		IR				
Karrantza		CZ				
La Asunción Universidad				AU		
La Florida	LF					
La Robla				RL		
La Unión						UN
La Vecilla				VL		
Laviana	LA					
Lieres	LI					
Liérganes			LG			
Llanes	LL					
Llovio	LO					
Los Carabeos				CR		
Los Nietos						LN
Luarca	LR					
Lutxana Barakaldo		LC				
Maliaño La Vidriera			MA			
Marrón			MR			
Matallana				MT		
Mataporquera			MP			
Mercadillo Villasana				MC		
Meres	ME					
Mieres Vasco	MI					
Mogro			MO			
Moreda de Aller	MD					
Muros de Nalón	MN					
Nava	NV					

Estación	Abreviatura de las estaciones de					
	Asturias	Bizkaia	Cantabria	C. y León	Galicia	Murcia
Navia	NA					
Nueva	NU					
Nueva Montaña			NM			
Orejo			OR			
Ortigueira					OT	
Oviedo	OV					
Pedrosa				PD		
Perlora	PE					
Piedras Blancas	PB					
Pinzales	PI					
Piñeres	PN					
Pola de Siero	PS					
Posada	PO					
Pravia	PR					
Puente Almuhey				PA		
Puente San Miguel			SM			
Regueral	RE					
Requejada			RJ			
Rianes	RI					
Ribadeo					RD	
Ribadesella	RS					
Ribadesella Puerto	RP					
San Claudio	SC					
San Esteban	EP					
San Feliz				SF		
San Pedro de Rudagüera			SP			
San Román	RO					
San Sadurniño					SS	
San Vicente de la Barquera			VB			
Santa María de Grado	MG					
Santander			SR			
Santiago del Monte	SG					
Santibáñez de la Peña				SZ		
Sebares	SE					
Sodupe		SD				
Solares			SL			
Soncillo				SN		
Sotiello	SO					
Soto de Dueñas	DU					
Soto de Luiña	LU					
Soto de Ribera	RB					
Soto del Barco	SB					
Sotoscueva				SV		

Estación	Abreviatura de las estaciones de					
	Asturias	Bizkaia	Cantabria	C. y León	Galicia	Murcia
Sotrondio	ST					
Sueros	SU					
Toraño	TO					
Torrelavega Centro			TV			
Traslaviña		TS				
Trasona	TR					
Treceño			TC			
Treto			TT			
Trubia	TB					
Tuilla	TU					
Ujo Taruelo	UT					
Unquera			UQ			
Vado Cervera				VC		
Vega de Anzo	VG					
Veriña	VE					
Villamayor	VM					
Virgen de la Peña			VP			
Viveiro					VV	
Xove					XO	
Xuvia					XU	
Zalla		ZA				
Zaramillo		ZR				
Zorrotza Zorrozoiti		ZO				

3.- DOCUMENTOS ANULADOS.

- Consigna Serie C nº 26/2011 DGCC-SC.
- Consigna Serie C nº 28/2011 DGCC-SC.
- Consigna Serie C nº 92/2015 GATR.
- Consigna Serie C nº 93/2015 GATR.

4.- ENTRADA EN VIGOR.

- Viernes, día 4 de marzo de 2016.

Oviedo, 26 de
febrero de 2016 El



Gerente de Área de
Tráfico RAM

Fdo.: Enrique González Rodríguez

ANEXO II: CONSIGNA SERIE C Nº 6/2011

CARGAS MÁXIMAS QUE PUEDEN REMOLCAR LAS LOCOMOTORAS

1.- OBJETO.

- **Actualizar** las Cargas máximas que pueden remolcar las locomotoras en los distintos trayectos de la red.

2.- CARGAS MÁXIMAS.

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
Ferrol-Aranguren: Línea General			
Ferrol-Cerdido	485	485	305
Cerdido-Ortigueira	1325	1325	965
Ortigueira-Xove	605	605	365
Xove-Ribadeo	725	725	425
Ribadeo-Pravia	665	665	425
Pravia-Trubia	965	965	605
Trubia-Oviedo	485	485	245
Oviedo-Colloto	1325	1325	965
Colloto-El Berrón	905	905	545
El Berrón-Lieres	905	905	545
Lieres-Nava	725	725	425
Nava-Llovio	1385	1385	965
Llovio-Ribadesella	665	665	365
Ribadesella-Nueva	845	845	485
Nueva-Posada	725	725	425
Posada-Llanes	965	965	565
Llanes-Unquera	725	725	425
Unquera-Cabezón	485	545	305
Cabezón-Requejada	1385	1385	965
Requejada-Santander	665	665	365
Santander-Orejo	1025	1025	585
Orejo-Gama	725	725	425
Gama-Gibaja	1025	1025	605
Gibaja-Karrantza	725	725	425
Karrantza-Villaverde de Trucíos	485	545	305
Villaverde de Trucíos-Traslaviña	725	725	425
Traslaviña-Aranguren	1385	1385	965

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
Ferrol-Aranguren: Ramales			
Pravia-San Esteban	1325	1325	965
Pravia-La Maruca	605	605	365
La Maruca-San Juan	545	545	305
La Maruca-Trasona	605	605	245
Trasona-Veriña	425	425	245
Veriña-Sotiello	425	425	245
Veriña-Gijón	425	425	245
Gijón-Sotiello	905	905	565
Musel-Aboño	965	965	585
Aboño-Sotiello	905	905	565
Sotiello-El Berrón	545	545	365
El Berrón-Carbayín	545	545	365
Carbayín-La Felguera	665	665	425
La Felguera-Sama	965	965	605
Sama-Laviana	665	665	425
Trubia-Fuso	965	965	605
Fuso-Ujo	905	905	565
Ujo-Collanzo	605	605	365
Llovio-Ribadesella Puerto	1385	1385	965
Maliaño-Raos	1500	1500	1080
Orejo-Liérganes	845	845	545

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
Aranguren-Ferrol: Línea General			
Aranguren-Karrantza	485	485	305
Karrantza-Gama	965	965	605
Gama-Orejo	665	665	425
Orejo-Maliaño	1325	1325	965
Maliaño-Santander	965	965	605
Santander-Requejada	605	605	365
Requejada-Cabezón	785	785	485
Cabezón-Llanes	545	545	365
Llanes-Llovio	785	785	485
Llovio-Infiesto	785	785	485
Infiesto-Rianes	665	665	425
Rianes-Pola de Siero	1325	1325	965
Pola de Siero-El Berrón	665	665	425
El Berrón-Colloto	1325	1325	965
Colloto-Oviedo	485	485	245
Oviedo-Trubia	1325	1325	965
Trubia-Pravia	1325	1325	965
Pravia-Ortigueira	605	605	365
Ortigueira-Moeche	545	545	365
Moeche-Ferrol	725	725	425

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
Aranguren-Ferrol: Ramales			
Liérganes-Orejo	965	965	725
Raos-Maliaño	1500	1500	1080
Ribadesella Puerto-Llovio	785	785	485
Collanzo-Trubia	1325	1325	965
Laviana-La Felguera	1325	1325	965
La Felguera-Carbayín	545	545	365
Carbayín-El Berrón	1325	1325	965
El Berrón-Sotiello	1325	1325	965
Sotiello-Aboño	1325	1325	965
Aboño-Musel	1325	1325	965
Sotiello-Gijón	905	905	565
Gijón-Veriña	485	485	305
Sotiello-Veriña	485	485	305
Veriña-Trasona	485	485	305
Trasona-La Maruca	605	605	305
San Juan-La Maruca	785	785	485
La Maruca-Pravia	605	605	365
San Esteban-Pravia	1325	1325	965

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
La Robla-Bilbao: Línea General			
La Robla-Matallana	545	545	305
Matallana-Guardo	485	485	305
Guardo-Villaverde Tarilonte	485	485	305
Villaverde Tarilonte-Cillamayor	785	785	485
Cillamayor-Los Carabeos	485	485	305
Los Carabeos-Arija	965	965	605
Arija-Soncillo	665	665	450
Soncillo-Pedrosa	965	965	585
Pedrosa-Sotoscueva	485	485	340
Sotoscueva-Balmaseda	965	965	585
Balmaseda-Aranguren	1325	1325	965
Aranguren-Irauregi	1325	1325	965
Irauregi-Basurto Hospital	625	625	485
Basurto Hospital-Bilbao	625	625	385

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
La Robla-Bilbao: Ramales			
Matallana-Asunción Universidad	1325	1325	965
Guardo-Térmica de Velilla	485	485	305
Irauregi-Lutxana	1325	1325	965
Basurto Hospital-Ariz	625	625	485

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
Bilbao-La Robla: Línea General			
Bilbao-Basurto Hospital	545	545	365
Basurto Hospital-Irauregi	905	905	565
Irauregi-Aranguren	905	905	565
Aranguren-Balmaseda	905	905	565
Balmaseda-Bercedo Montija	425	425	300
Bercedo Montija-Guardo	485	485	305
Guardo-Matallana	485	485	305
Matallana-La Robla	550	550	365

TRAYECTO	Dual 1900	Alsthon 1600	Geco 1500
Bilbao-La Robla: Ramales			
Ariz-Basurto Hospital	905	905	565
Lutxana-Irauregi	725	725	485
Térmica de Velilla-Guardo	485	485	305
Asunción Universidad-Matallana	665	665	425

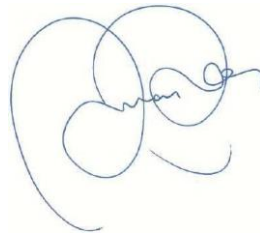
3.- ENTRADA EN VIGOR.

- Martes, día 22 de noviembre de 2011.

4.- DOCUMENTO ANULADO.

- Consigna Serie C nº 4-DG-2001.

Vº Bº
El Director General y de Infraestructuras



Fdo. Juan Díez López

Oviedo, 15 de noviembre de 2011

El Director Gerente de Gestión de Capacidad y Seguridad en la Circulación



Fdo.: Enrique González Rodríguez

ANEXO III: CACTERÍSTICAS DE FRENADO DEL MATERIAL MOTOR Y REMOLCADO

1. LOCOMOTORAS:

<u>FAMILIA</u>	<u>PESO / SERV(TM)</u>	<u>SIN CAMB.</u>	<u>VIAJERO S</u>	<u>MERCANCIA S</u>	<u>FRENO ESTAC</u>
Geco 1500	56	40	---	---	9
Alsthom 1600	58	---	58	46	23
Dual 1900	60	---	60	47	23

2. VAGONES:

FAMILIA	FRENO DE AIRE			FRENO ESTC.
	Tara (tons)	En vacío	En carga	
2SSag	14	16	36	30
2SSag (601 a 650)	14	16	36	18
2TTag (40m ³)	20	19	42	18
2TTag (42m ³)	21	19	42	18
2TTag (Balastera)	19	19	42	25
2JJag	18	21	37	31
Sghmmns	17	17	52max	20
Sgh	17	17	52max	20

ANEXO IV: PORCENTAJES DE FRENADO AUTOMÁTICO Y DE ESTACIONAMIENTO

TRAYECTOS DE LA RED	T.70	T.60	T.50	T.40	T.30	EST.
<u>Ferrol a Gijón</u>						
<i>Ferrol-Pravia</i>	75	55	40	30	25	12
<i>Pravia-Avilés</i>	75	55	40	30	25	12
<i>Avilés-Gijón</i>	100	70	55	45	35	15
<u>Gijón-Ferrol</u>						
<i>Gijón-Avilés</i>	100	75	60	45	35	15
<i>Avilés-Pravia</i>	75	55	40	30	25	12
<i>Pravia-Ferrol</i>	75	55	40	30	25	12
<u>Oviedo-S.Esteban y Trubia-Collanzo</u>						
<i>Oviedo-Trubia</i>	100	70	55	45	35	15
<i>Trubia-Pravia</i>	90	65	50	35	25	12
<i>Pravia-S.Esteban</i>	75	55	35	25	25	8
<i>Trubia-Fuso</i>	70	50	30	25	25	8
<i>Fuso-Collanzo</i>	86	65	50	35	25	12
<u>S. Esteban-Oviedo y Collanzo-Trubia</u>						
<i>S. Esteban-Pravia</i>	70	50	30	25	25	8
<i>Pravia-Trubia</i>	70	50	30	25	25	8
<i>Trubia-Oviedo</i>	100	70	55	45	35	15
<i>Collanzo-Fuso</i>	85	65	50	35	25	12
<i>Fuso-Trubia</i>	90	65	50	35	25	12

TRAYECTOS DE LA RED	T.70	T.60	T.50	T.40	T.30	EST.
<u>Gijón-Laviana</u>						
Gijón-Sotiello	80	60	45	30	25	15
Musel-Aboño	90	65	50	35	25	12
Aboño-Sotiello	85	65	50	35	25	15
Sotiello-Laviana	85	65	50	35	25	15
Sama-Modesta	80	60	40	25	25	8
<u>Laviana-Gijón</u>						
Laviana-Sotiello	90	65	50	35	25	15
Modesta-Sama	80	60	40	25	25	8
Sotiello-Aboño	80	60	40	25	25	9
Aboño-Musel	90	65	50	35	25	12
Sotiello-Gijón	80	60	40	25	25	9
<u>Oviedo-Santander</u>						
Oviedo-Colloto	100	70	55	45	35	15
Colloto-Santander	90	65	50	35	25	12
<u>Santander-Oviedo</u>						
Santander-Colloto	90	65	50	35	25	15
Colloto-Oviedo	100	70	55	45	35	15

TRAYECTOS DE LA RED	T.70	T.60	T.50	T.40	T.30	EST.
<u>Santander-Aranguren y Orejo-Lierganes</u>						
Santander-Orejo	70	50	30	25	25	8
Orejo-Lierganes	70	50	30	25	25	9
Orejo-Traslaviña	90	65	50	35	25	12
Traslaviña-Aranguren	95	75	55	40	25	15
<u>Aranguren Santander y Lierganes-Orejo</u>						
Aranguren-Maliaño	90	65	50	35	25	15
Maliaño-Santander	70	50	30	25	25	9
Lierganes-Orejo	85	65	50	35	25	9
<u>Balmaseda-Bilbao Irauregui-Luchana y Basurto-Ariz</u>						
Balmaseda-Aranguren	80	60	45	30	25	9
Aranguren-Bilbao	90	65	50	35	25	12
Irauregui-Luchana	90	65	50	35	25	12
Basurto-Ariz	90	65	50	35	25	12
<u>Bilbao-Balmaseda Ariz-Basurto y Luchana-Irauregui</u>						
Bilbao-Irauregui	90	65	50	35	25	12
Ariz-Basurto	90	65	50	35	25	12
Luchana-Irauregui	70	50	30	25	25	9
Irauregui-Aranguren	70	50	30	25	25	9
Aranguren-Balmaseda	70	50	30	25	25	9
<u>La Robla-Balmaseda y León-Matallana</u>						
La Robla-Matallana	95	75	55	40	30	12
Matallana-Bercedo	95	75	55	40	30	15
Bercedo-Balmaseda	100	75	60	45	35	15
León-Matallana	95	75	55	40	30	15
<u>Balmaseda-La Robla y Matallana-León</u>						
Balmaseda-Matallana	95	75	55	40	30	15
Matallana-La Robla	95	75	55	40	30	15
Matallana-León	95	75	55	40	30	15

ANEXO V: MARCHAS DE ASIMILACIÓN

1.1 Marchas de Asimilación del Trayecto Ferrol – Oviedo:

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-60	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
FERROL 	0,000	-----						
		70						
Virxe do Mar	2,400							
Santa Ixia	3,002							
	3,640	-----						
	4,900	60	8	8	10	11	13	17
		70						
Piñeiros	5,472							
O Ponto	6,200							
XUBIA	7,660	-----						
	7,860	60						
	7,990	-----						
	8,470	70						
	8,470	60						
	8,670	-----						
		70						
Ferrerías	9,307							
	10,830	-----	10	10	18	19	20	22
		60						
Sedes	11,287	-----						
	11,500	70						
Pedroso	14,035	-----						
	14,530	60						
	15,000	-----						
		70						
	16,630	-----						
		60						
	16,850	-----						
SAN SADURNIÑO	17,041	70						
	18,090	-----						
		50						
	18,660	-----						
		60						
	19,300	-----						
		70						
Lamas	21,076							
	21,130	-----						
		50						
	21,386	-----						
		70						
	21,600	-----						
		60	18	18	31	32	34	38
	22,932	-----						
		70						
	22,950	-----						
		50						
	23,190	-----						
Apallia	24,603	70						
	24,930	-----						
		60						
	25,180	-----						
		70						
	26,355	-----						
Moeche	26,638	60						
	27,040	-----						
		70						
	27,840	-----						

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Labacengos	• 28,060	60						
	• 28,698	70						
	• 28,910	60						
	• 29,090	70						
	• 29,450	60						
	• 29,710	70						
Entrambarrias	• 31,401	60						
	• 31,960	70						
	• 32,190	60						
CERDIDO	• 33,623	70						
	• 37,110	60	19	19	23	25	31	40
	• 37,380	50						
A Cuqueira	• 38,081	60						
	• 38,360	50						
	• 38,750	60						
	• 41,570	50						
	• 42,500	60						
	• 42,687	70						
Sta. Mª de Mera	• 43,190	50						
	• 43,370	60						
	• 45,929	70						
Ponte Mera ▲	• 46,810	60						
	• 47,140	70						
	• 48,626	60						
Senra	• 50,402	70						
	• 51,280	60						
	• 51,475	70						
ORTIGUEIRA	• 52,592	60						
	• 57,565	70	17	17	26	28	31	37
	• 63,915	60						
Loiba O Barqueiro ▲	• 69,118	70						
	• 69,515	60	19	19	27	29	32	37
	• 70,425	70						
O Vicedo	• 72,362	60						
	• 73,205	70						
	• 74,579	60						
	• 75,515	70						

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Mosende	• 76,284	60						
	• 76,740	70						
	• 77,278	60						
	• 78,050	70						
	• 78,365	60						
Folgueiro	• 79,429	70						
	• 79,565	70						
	• 81,420	50						
	• 81,780	70						
	• 82,340	50						
Covas	• 83,034	70						
	• 83,585	70						
	• 83,945	50						
	• 84,221	70						
	• 85,980	60						
VIVEIRO	• 87,010	50	15	15	24	25	27	30
Xuances	• 87,250	60						
	• 93,780	70						
	• 97,480	60						
	• 97,670	70						
	• 97,745	70						
Xove Apdo.	• 98,760	60						
XOVE	• 99,020	60						
Lago	• 99,800	70						
	• 100,440	60						
	• 100,780	70						
	• 101,600	70						
	• 102,162	80						
Biduelros	• 104,510	80						
San Cibrao	• 107,155	60						
Madeiro	• 107,680	60	12	12	19	22	25	30
	• 108,100	80						
	• 108,760	60						
	• 108,990	80						
	• 111,380	60						
BURELA	• 111,630	80						
	• 112,300	50						
	• 112,465	80						
	• 112,715	80						
	• 113,057							

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
		60						
	113,439	80						
	113,552	70						
	114,317	80						
	115,280	60						
	116,369	80						
Cangas de Foz	117,335		13	14	20	22	25	31
Nois	119,770							
	120,970	50						
	121,129	80						
Fazouro	121,240	60						
	122,149	80						
	122,440	60						
	125,170	80						
	125,170	60						
Marzán	125,245							
	125,482	80						
	126,288	60						
	126,780	80						
Foz	127,320	70						
	129,410	80						
	129,764	60						
	129,863	80						
	130,060	60						
Barreiros	131,945							
Reinante	136,090		13	15	27	29	33	41
Esteiro	138,385	80						
Os Castros	139,600							
Rinlo	141,430							
RIBADEO	143,310	60						
Vegadeo	153,030							
	153,990	50						
Vegadeo Pueblo	154,180	70	25	25	32	35	41	50
	154,250	50						
	155,060	70						
	156,950	50						
Vilavedelle	158,470	70						
	158,720	50						
	159,300	70						
	159,860	50						

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Castropol	160,070	70						
	161,930	60						
	162,850	70						
Les Campas	163,520							
Tol	165,706							
Tapia de C.	169,711							
	171,770	50						
	171,980	70						
	173,310	60						
	173,430	70						
	177,100	50						
	177,520	70						
	178,020							
	178,130	50						
	178,240	70						
Cartavio	181,455		21	21	27	30	35	43
	182,240	50						
	182,520	70						
Loza	183,065							
Jarrio	184,766							
	185,130	50						
	187,480	70						
	184,420	50						
	188,650	70						
	189,180							
NAVIA	189,665	50						
	192,600	20						
	193,080	70						
	193,470	60						
Piñera	193,960							
	195,884	70	23	23	32	35	39	47
	197,030	60						
Villapedre	198,280	70						
	199,760							
	201,600	50						
	202,040	70						
	203,860	50						
	204,220							

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Otur	• 205,491	70						
	• 208,110	50						
	• 208,430	60						
	• 208,890	70						
	• 210,010	50						
LUARCA	• 211,440	40						
	• 212,162	50						
	• 212,822	70						
	• 213,590	50						
Barcia	• 214,300	70						
	• 215,520	50						
	• 216,580	70						
	• 217,530	60						
	• 217,760	70						
Canero	• 219,700	40	16	16	25	26	28	32
	• 220,264	70						
	• 221,140	60						
	• 221,720	50						
	• 222,190	70						
San Cristobal	• 222,450	50						
	• 223,362	70						
Cadavedo	• 225,390	60						
	• 225,580	40						
	• 225,760	60						
	• 228,023	50						
	• 228,600	60						
Tablizo	• 229,900	40						
	• 230,982	60						
	• 231,467	60						
Bailota	• 233,130	20	20	20	23	26	30	37
	• 234,612	50						
Novellana	• 236,088	60						
	• 238,070	40						
Valdredo	• 238,370	60						
	• 239,040	40						
	• 241,670	60						
SOTO DE LUIÑA	• 242,070	60						
	• 242,400							

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
San Martín	247,122	-----						
	247,140	50						
La Magdalena	247,232	-----	16	16	21	23	25	30
	251,600	-----						
Villademar	255,126	60						
CUDILLERO	255,938	-----						
El Pitto-Piñera	256,980	-----	4	4	9	9	10	11
	257,850	-----						
	50	-----						
	259,350	-----						
	60	-----						
MUROS DE NALÓN	260,650	-----						
	70	-----						
	261,410	-----						
	50	-----						
	262,800	-----						
	70	-----						
Los Cabos	264,400	-----	10	10	11	12	14	19
Santianes	265,241	-----						
	266,567	50						
	267,520	-----						
	70	-----						
	268,100	-----						
	60	-----						
	268,300	-----						
	70	-----						
PRAVIA	269,276	-----						
Beñar	271,990	-----						
	272,290	-----						
	50	-----						
	272,580	-----	6	6	9	10	11	14
	70	-----						
	274,470	-----						
	50	-----						
	274,650	-----						
	70	-----						
SAN ROMAN	275,806	-----						
	276,940	-----						
	50	-----						
	277,430	-----						
	70	-----						
Aces	277,545	-----						
	279,990	-----						
	50	-----						
	280,580	-----	8	8	13	13	15	19
	70	-----						
Sandiche	280,708	-----						
	283,100	-----						
	60	-----						
	284,300	-----						
	70	-----						
GRADO	284,468	-----						
	288,620	-----						
	60	-----						
	289,450	-----						
	70	-----						
Vega	290,190	-----						
	290,580	-----						
	60	-----	8	8	13	13	15	18
	291,400	-----						
	70	-----						
	292,200	-----						
	50	-----						

ESTACIONES	P.K.	V.M.	0101	0103	0105	0107	0109	0111
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
S. M ^a DE GRADO 	292,500	70						
	292,761							
	294,190	50						
	294,750	70						
	295,410	60	5	5	8	8	9	11
	295,800	70						
	296,150	50						
	296,850							
	296,884							
	296,970	70						
TRUBIA	297,874							
Soto Udrón	296,970							
Aguja de enlace	296,884							
San Pedro de Nora	11,146							
	9,589	80	6	7	12	14	18	23
	9,365	70						
	8,540							
SAN CLAUDIO	5,472							
Las Mazas	3,853	80						
Las Campas	2,600	60						
Argañosa - Lavapiés	2,524		6	7	10	12	15	17
	2,300							
OVIEDO	1,383	80						
	0,000							

1.2 Marchas de Asimilación del Trayecto Oviedo- Ferrol:

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
OVIEDO	0,000							
		80						
Argañosa - Lavapies	1,383		5	6	7	8	9	11
	2,300	---						
Las Campas	2,524	80						
	2,600	---						
Las Mazas	3,853							
SAN CLAUDIO	5,472	80						
	8,540	---						
	9,365	70	10	11	13	15	19	25
San Pedro de Nora	9,589	80						
	11,146	---						
Aguja de enlace	296,884							
Soto Udrón	296,970							
TRUBIA	297,874	70						
Soto Udrón	298,970							
Aguja de enlace	296,884							
	296,850	---						
	296,150	50						
	285,800	70						
	295,410	60	5	5	6	7	8	11
	294,750	70						
	294,190	50						
	292,761	70						
STA. Mª DE GRADO	292,500	---						
	292,200	50						
	291,400	70	3	3	4	5	6	7
	290,580	60						
	290,190	70						
Vega	289,450	---						
	286,820	60	5	5	6	6	8	11
	284,488	70						
GRADO	284,300							
	283,100	60						
	280,708	70	8	8	10	12	14	19
Sandiche	280,590	---						
	279,990	50						
	277,545	70						
Aces								

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
SAN ROMAN ▲	277,430	50						
	276,940	70						
	275,808	50						
	274,850	70						
	272,580	50	6	6	8	9	11	14
	272,290	70						
Beifar PRAVIA	271,990							
	269,276	60						
	268,300	70						
	267,520							
Santianes	266,567	50	10	10	20	20	20	21
Los Cabos	265,241							
	264,400	70						
	262,800	50						
	261,410	70						
MUROS DE NALON ▲	260,650	60						
	259,350	50						
	257,850		6	6	6	7	8	10
	256,980							
El Pito-Piñera	255,938	60						
CUDILLERO ▲	255,126							
Villademar	251,600							
La Magdalena	247,232	50	15	15	18	20	23	29
San Martín	247,140	60						
San Cosme	245,480							
SOTO DE LUIÑA	242,400	40						
	242,070	60						
	241,670							
	239,040	50	20	20	25	27	31	37
Valdredo	238,370							
Novellana	238,070							
Sla. Marina	236,088	60						
Ballota	234,612							
	233,130							
	231,467	40						
	230,982							

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Tablizo	• 229,900	60						
	• 228,600	50						
	• 228,023	60						
	• 225,760	40						
	• 225,580	60						
	• 225,380	70						
Cadavado	• 223,362	50						
San Cristobal	• 222,450	60						
	• 222,190	70						
	• 221,720	70						
	• 221,140	40						
Canero	• 220,264	70	16	16	21	22	25	30
	• 219,700	60						
	• 217,760	70						
	• 217,530	50						
	• 216,680	50						
	• 215,520	70						
Barcia	• 214,300	50						
	• 213,590	40						
	• 212,822	50						
	• 212,182	50						
LUARCA	• 211,440	70						
	• 210,010	60						
	• 208,830	50						
	• 208,430	70						
Otur	• 208,110	50						
	• 205,491	70						
	• 204,220	50						
	• 203,860	70						
Villapedre	• 202,040	50	23	23	30	32	37	46
	• 201,500	70						
	• 199,760	60						
	• 198,260	70						
Piñera-Villaoril	• 197,030	70						
	• 195,664	60						
	• 193,960	70						
	• 193,470	20						
NAVIA	• 193,080	50						
	• 192,600	70						
	• 189,665	70						

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Jarrio	188,650	----						
	188,420	50						
	187,480	70						
	186,130	50						
	184,766	70						
Loza	183,085	----						
	182,520	50						
	182,240	70						
Cartavio	181,455	----	21	21	32	34	38	44
	178,240	50						
	178,130	70						
La Caridad	178,020	----						
	177,520	50						
	177,100	70						
	173,430	60						
	173,310	70						
Tapia de C.	171,980	50						
	171,770	----						
	169,711	70						
	165,706	60						
	161,930	70						
Las Campas	160,070	----						
	159,880	50						
	159,300	70						
	158,720	50						
	158,470	70						
Castropol	156,950	----	26	26	30	33	39	49
	155,060	70						
	154,250	50						
Vilavedelle	154,180	----						
	153,990	60						
	153,030	70						
Vegadeo Pueblo	146,310	----						
	141,430	80						
	139,600	----						
RIBADEO								
Rinio								
Os Castros								

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Esteiro	•	138,385						
Reinante	•	136,090	13	15	23	26	31	39
Barreiros	•	131,945						
	•	130,060						
	•	129,863						
	•	129,784						
	•	129,410						
Foz	▲	127,320						
	•	126,780						
	•	126,268						
	•	125,482						
Marzán	•	125,245						
	•	125,170						
	•	122,440						
	•	122,149						
Fazouro	•	121,240						
	•	121,129						
	•	120,970						
Nois	•	119,770	13	14	21	23	26	31
Cangas de Foz	•	117,335						
	•	116,369						
	•	115,280						
	•	114,317						
	•	113,552						
	•	113,439						
	•	113,057						
BURELA	•	112,715						
	•	112,465						
	•	112,300						
	•	111,630						
	•	111,380						
	•	108,990						
	•	108,780						
	•	108,100						
	•	107,680						
Madeiro	•	107,155						
San Cibrao	•	104,510						
Bidueiros	•	102,162						
	•	101,900						
	•	100,780						
Lago	•	100,440						
XOVE	▲	99,800						
	•	99,020						

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Xove Apdo.	98,760	----						
	97,745	70						
	97,670	60						
	97,480	70	15	15	21	22	25	29
Xuances	93,780	----						
	87,250	60						
	87,010	50						
	85,980	60						
VIVEIRO	84,221	70						
	83,945	50						
	83,585	70						
	83,034	50						
Covas	82,340	70						
	81,780	50						
	81,420	70						
	79,565	70	19	19	27	29	32	37
Folgueiro	79,429	60						
	78,365	70						
	78,050	70						
	77,278	60						
Mosende	76,740	70						
	76,284	60						
	75,515	70						
	74,579	60						
O Vicedo	73,205	70						
	72,362	70						
	70,425	60						
	69,515	----						
O Barqueiro	69,118	70	17	17	25	28	31	37
Loiba	63,915	70						
Espasante	57,565	70						
ORTIGUEIRA	52,592	60						
	51,475	60						
	51,280	70						
	50,402	70						
Senra	48,626	60						
San Clodio	47,140	70						
	46,810	60						
	45,929	70						
	43,370	60						
Ponte Mera	43,190	50	22	22	37	38	41	44

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-90	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Sta. M ^a de Mera	42,687	60						
	42,500	50						
	41,570	60						
	38,750	50						
A Cuqueira	38,360	60						
	38,081	50						
	37,380	60						
	37,110	70						
CERDIDO	33,623	60						
	32,190	70						
	31,960	60						
	31,401	70						
Entrambarrias	29,710	60						
	29,450	70						
	29,090	60						
	28,910	70						
Labacengos	28,688	60						
	28,060	70						
	27,840	60						
	27,040	70						
Moeche	26,638	60	16	16	22	25	29	36
	26,355	70						
	25,180	60						
	24,930	70						
Apalla	24,603	50						
	23,190	70						
	22,950	60						
	22,932	70						
Lamas	21,600	50						
	21,386	70						
	21,130	60						
	21,076	70						
SAN SADURNIÑO	19,300	60						
	18,660	50						
	18,090	70						
	17,041	60						
Pedroso	16,890	70						
	16,630	60						
	15,000	70						
	14,530	60						
	14,035	70						
	11,500							

ESTACIONES	P.K.	V.M.	0100	0102	0104	0106	0108	0110
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Sedes	11,287	60	10	10	11	13	16	20
Ferrerías	10,830	70						
	9,307	80						
	8,670	60						
	8,470	70						
	7,990	60						
XUVIA	7,860							
	7,660							
O Ponto	6,200	70						
Piñeiros	5,472							
	4,900	60	8	8	13	14	16	18
Santa Iclia	3,640							
	3,002							
Virxe do Mar	2,400	70						
FERROL	0,000							

2.1 Marchas de Asimilación del Trayecto San Esteban – Pravia - Gijón:

ESTACIONES	P.K.	V.M.	0201	0203	0205	0207	0209	0211
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
SAN ESTEBAN 	10,212							
San Ramón	4,972	60	11	11	12	13	16	21
PRAVIA	0,000	80						
	0,355	60						
	1,310	80						
Peñaulán	1,585	80						
	2,454	80	8	8	10	11	13	16
	2,810	80						
Riberas	3,900	80						
	6,517							
SOTO DEL BARCO	6,967	70						
	8,093	60						
Santiago del Monte	11,501							
	11,855	80						
Vegarrozadas	14,673		12	12	17	18	19	22
	14,845	70						
	15,013	80						
	15,662	60						
PIEDRAS BLANCAS	16,484	80						
	16,598							
	17,778	60						
Salinas	18,779	80	7	7	7	8	9	11
	19,219							
	20,322							
Raíces	20,566							
Cristalería	21,197							
LA MARUCA	22,164	60	1	1	1	2	2	2
AVILES	23,200		1	1	1	1	1	2
AVILES Apdo. 	23,830							
	24,090							
Llaranes	25,554		4	4	5	6	7	9
Oficinas	26,767							
TRASONA 	27,646	80						
Laminación	28,454							
Zanzabomin	32,779		9	9	13	13	14	16
REGUERAL	35,144							
	37,117		2	2	3	4	4	5
CANDAS 	37,738	70						
Candas Apdo.	38,182							
	38,404	40						
	38,552	80	2	2	2	3	3	4
	38,996							

ESTACIONES	P.K.	V.M.	0201	0203	0205	0207	0209	0211
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
PERLÓRA	39,371	80						
	39,755	80						
	40,737	70						
	42,282	80						
Xivares	42,566	80						
	42,963	60						
	43,712	40	8	8	9	9	10	13
	44,147	80						
Aboño	44,208	80						
	44,946							
	46,068							
VERIÑA								
Centro de Transportes	47,850	60						
Tremañes	48,988	80						
	49,000	6	6	7	8	9	11	
	50,231	70						
La Braña	50,300							
	50,407							
GUJÓN	51,082							

2.2 Marchas de Asimilación del Trayecto Gijón – Pravia – San Esteban:

ESTACIONES	P.K.	V.M.	0200	0202	0204	0206	0208	0210
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
GIJÓN	51,082							
La Braña	50,407	----						
	50,300	70						
	50,231	----	6	6	7	8	9	11
		80						
Tremañes Centro de Transportes VERIÑA	49,000	----						
	48,988							
	47,850	60						
	46,068	----						
Aboño	44,946	----						
	44,208	80						
	44,147	----						
		40						
	43,712	----	8	8	9	9	11	14
		60						
	42,963	----						
	42,566	80						
Xivares	42,262	----						
		70						
	40,737	----						
	39,755	80						
PERLORA	39,371	----						
		60						
	38,996	----						
		80						
	38,552	----	2	2	2	3	3	4
		40						
	38,404	----						
	38,182	70						
Candás Apdo.								
CANDAS	37,738	----						
	37,117	----	2	2	3	4	4	5
REGUERAL	35,144	----						
Zanzabornin	32,779							
			9	9	11	12	13	15
Laminación	28,454							
TRASONA	27,646	80						
Oficinas	26,767							
			4	4	5	6	7	9
Llaranes	25,554							
	24,090	----						
AVILES Apdo.	23,830							
			1	1	1	1	1	2
AVILES	23,200							
			1	1	1	2	2	2
LA MARUCA	22,164	60						
Cristalería	21,197							
Raíces	20,566		7	7	9	9	10	12
	20,322	----						
		80						
	19,219	----						
Salinas	18,779	60						
	17,778	----						
		80						
PIEDRAS BLANCAS	16,598	----						
	16,484	60						
	15,662	----						
		80						
	15,013	----						
		70	12	12	17	18	20	22
	14,845	----						

ESTACIONES	P.K.	V.M.	0200	0202	0204	0206	0208	0210
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Vegarrozadas	14,673	80						
	11,855	-----						
Santiago del Monte	11,501	60						
	8,093	-----						
SOTO DEL BARCO	6,987	70						
	6,517	-----						
Riberas	3,900	80						
	2,810	-----	8	8	9	10	12	15
		60						
	2,454	-----						
Peñauflán	1,586	80						
	1,310	-----						
		60						
	0,355	-----						
		80						
PRAVIA	0,000							
San Ramón	4,972							
SAN ESTEBAN 	10,212	60	11	11	12	13	16	21

3.1 Marchas de Asimilación del Trayecto Collanzo - Trubia:

ESTACIONES	P.K.	V.M.	0301	0303	0305	0307	0309	0311
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
COLLANZO 	47.570	70						
	46.100	50						
	45.650	70	6	6	7	9	11	14
Levinco	43.217	70						
	42.850	60						
	42.560	70						
CABAÑAQUINTA 	40.722	70						
	38.959	60						
	38.870	70						
Santa Ana	37.240	70	5	5	5	6	8	11
	36.300	70						
	35.662	60						
Piñeres 	35.100	70						
	33.770	70						
	32.342	70	6	6	6	7	8	11
Oyanco	31.960	50						
	30.410	70						
	30.195	70						
MOREDA 	28.875	60	5	5	6	7	9	12
	25.800	70						
	24.538	70						
Caborana	24.450	50	2	2	2	2	3	3
	24.330	70						
	23.143	70	4	4	5	5	7	9
FIGAREDO	19.020	40						
	18.550	70						
	18.390	70						
Aguja Sueros	17.405	3	3	4	5	5	7	
	17.090	40						
	16.720	70						
ABLAÑA 	15.604	70						
	14.630	50	2	2	2	3	3	4
	14.465	70						
La Pareda	14.020	70						
	13.127	60						
	12.420	70	2	2	4	5	6	8
Baña	12.200	70						
	11.275	40						
	10.975	70						
Peñamiel 	10.065	70						
	9.960							

ESTACIONES	P.K.	V.M.	0301	0303	0305	0307	0309	0311
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Parlayar	8.462	50						
	8.420	70	5	5	6	6	8	10
	8.080	50						
	7.920	70						
	7.867	40						
	7.867	40						
SOTO DE RIBERA	5.222	70						
Palomar	1.664							
	1.370	60	5	5	6	7	8	11
	1.220							
FUSO	0.000							
Caces	304.491	70						
	303.264							
	303.180	50						
	302.100	70	7	7	9	10	12	16
	298.700	30						
TRUBIA	298.330							
	297.874	70						

3.2 Marchas de Asimilación del Trayecto Trubia - Collanzo:

ESTACIONES	P.K.	V.M.	0300	0302	0304	0306	0308	0310
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
TRUBIA	297.874	70						
	298.330	30						
	298.700	70						
	302.100	50	7	7	13	13	14	16
	303.180	70						
Caces	303.264							
FUSO	304.491							
	0.000							
	1.220	60						
	1.370		5	5	6	7	8	11
Palomar	1.664	70						
SOTO DE RIBERA	5.222							
	7.867	40						
	7.867	70						
	7.920	50						
	8.080	70						
	8.420		5	5	6	6	8	10
Parteayer	8.462	50						
	9.960							
Peñarniel	10.065	70						
	10.975	40						
	11.275	70						
	12.200	60						
	12.420		5	5	5	5	6	8
Baia	13.127	70						
	14.020							
		50						
La Pereda	14.465							
	14.630							
ABLAÑA	15.804	70						
	16.720	40						
	17.090							
Aguja Sueros	17.405		3	3	5	5	5	7
Aguja Sueros	18.390	70						
	18.550	40						
	18.550							
MIERES	19.020	70	4	4	5	5	7	9
FIGAREDO	23.143							
	24.330	50						
	24.460	70	2	2	3	3	3	3
UJO - TARUELO	24.638							
Santa Cruz	25.800	60	6	6	9	9	10	12
Caborana	28.876							
MOREDA	30.196							

ESTACIONES	P.K.	V.M.	0300	0302	0304	0306	0308	0310
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Oyanco	30.410	70						
	31.960	50						
	32.342	70	6	6	10	10	10	11
	33.770	60						
	35.100	70						
Piñeres	35.662							
Corigos	36.300	70						
	37.240	60	6	6	10	10	10	11
	38.870							
Santa Ana	38.859							
CABANAQUINTA	40.722	70						
	42.560	60						
	42.850	70						
Levinco	43.217	70						
	45.650	50	9	9	16	16	16	16
	46.100	70						
COLLANZO	47.570							

4.1 Marchas de Asimilación del trayecto Gijón – Laviana y ramal:

ESTACIONES	P.K.	V.M.	0401	0403	0405	0407	0409	0411
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
GIJÓN	0,000							
La Braña	0,700		5	6	8	9	10	13
Tremañes	2,490							
SOTIELLO	6,462		2	3	3	4	4	5
PINZALES	8,895							
La Aguda	11,575		5	6	12	12	13	14
FLORIDA	15,309	80						
Norraña	19,884		5	6	12	12	13	14
EL BERRON C.T.C	21,143							
Xixón	22,637		3	3	4	4	5	6
BENDICIÓN	24,029							
	25,500	80						
	25,800	80	4	4	9	9	9	9
Valdesoto	25,927							
	25,980	60						
	26,180							
CARBAYIN	28,650	80						
Boca S.-Curuxona	29,769		3	3	3	4	5	6
TUILLA	31,396		3	4	6	6	7	11
LA FELGUERA	35,615		2	2	2	2	3	4
SAMA	37,270							
Ciaño	38,811		4	4	5	6	6	8
San Vicente	40,092							
EL ENTREGO	41,127							
Carrocera	42,359	70	4	4	5	5	6	8
San Martín	43,752							
SOTRONDIO	44,804							
Blimea	46,444							
Argayón	47,100		5	5	7	8	8	10
Barredos	48,366							
LAVIANA	49,748							
ABOÑO	6,300	60	5	6	10	10	10	13
SOTIELLO	0,000							

4.1 Marchas de Asimilación del trayecto Gijón – Laviana y ramal :

ESTACIONES	P.K.	V.M.	0400	0402	0404	0406	0408	0410
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
LAVIANA	49,748	70						
Barredos	48,386							
Argayón	47,100		5	5	5	7	7	10
Blimea	46,444							
SOTRONDIO	44,804							
San Martín	43,752							
Carrocera	42,359		4	4	4	5	6	8
EL ENTREGO	41,127							
San Vicente	40,092		4	4	4	5	6	8
Ciaño	38,811							
SAMA	37,260	80	2	2	2	3	3	4
LA FELGUERA	35,615		3	4	10	10	10	11
TUILLA	31,396							
Boca S.-Curuxona	29,769		3	3	5	5	5	6
CARBAYIN	28,650							
	26,180							
	25,980		4	4	5	6	7	9
Valdesoto	25,927							
	25,800							
	25,500							
BENDICIÓN	24,090	80	3	3	4	4	5	6
Xixón	22,637							
EL BERRON C.T.C.	21,143							
Noreña	19,884		5	6	7	8	9	12
FLORIDA	15,309							
La Aguda	11,575		5	6	7	8	10	13
PINZALES	8,895		2	3	3	3	4	5
SOTIELLO	6,462							
Tremañes	2,490		5	6	9	10	11	13
La Braña	0,700							
GIJON	0,000							
SOTIELLO	0,000	60	5	6	7	8	10	13
ABOÑO	6,300							

5.1 Marchas de Asimilación del trayecto Oviedo – Santander y ramal:

ESTACIONES	P.K.	V.M.	0501	0503	0505	0507	0509	0511
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
OVIEDO	313,800	80						
	320,860	40	7	7	8	9	11	15
	321,020							
COLLOTO	321,161	80						
	321,800	70						
	322,300	80	4	4	6	6	7	9
	324,700	70						
	325,400							
MERES	325,431	80	3	3	6	6	7	8
EL BERRON C.T.C.	329,146							
	330,100	60						
	330,900	80	4	8	8	8	8	9
	331,420							
La Carrera	331,568	40						
	332,620							
POLA DE SIERO	333,042	80						
	333,250	60						
	333,450							
Los Corros	339,210	80	7	8	13	13	14	17
	340,570	40						
	340,790							
LIERES	340,956	80	2	2	3	3	4	4
	341,640	50						
	342,120							
Rianes	342,656	80	5	5	9	9	10	12
El Remedio	344,090							
Llames	345,900							
NAVA	347,780	40						
	347,930	50	7	8	9	10	12	16
	348,480							
	350,170							
Fuente Santa	350,954	80						
Caceda	353,734							
CARANCOS	355,407	70						
	356,300	80						
	356,500							
	357,060							

ESTACIONES	P.K.	V.M.	0501	0503	0505	0507	0509	0511
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Pintueles	357,170	50						
	359,000	80						
	359,500	60	6	6	7	8	10	13
	359,720	40						
	359,780	50						
	361,430	40						
INFIESTO	361,460	80						
	361,700	80						
	361,820	80						
Infiesto Apdo.	362,900	60	6	6	7	9	10	14
	363,430	80						
Villamayor	368,361	80						
Sebares	372,423	60	3	4	4	5	6	8
	374,330	50	4	4	4	5	6	7
	375,780	60						
SOTO DE DUEÑAS	375,910	40						
	377,950	60						
	378,250	60	6	6	7	9	11	14
Ozanes Policlinico Ariondas ARRIONDAS	380,515	40						
	381,900	60						
	382,938	40						
	384,850	60						
Fuentes Toraño Cuevas	385,290	60	6	7	8	9	11	15
	387,629	60						
	390,232	40						
	395,259	60	6	7	8	9	11	14
Llovio	396,020	50						
	396,220	60						
	397,588	50						
RIBADESELLA Camango Belmonte	397,710	40						
	400,180	60	4	4	4	4	5	6
	400,312	70						
	403,863	60						
NUEVA	407,560	60						
	409,640	60	11	11	17	18	20	25
	411,830	70						
	412,450	60						
Villahormes	412,560	70						
	412,960	60						
	415,293	60						
	417,750	60	8	8	10	12	14	17
POSADA Balmori	420,700	70						
	420,940	60						
	423,415	60						
	424,590	60						

ESTACIONES	P.K.	V.M.	0501	0503	0505	0507	0509	0511
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Celorio	424,930	----						
	425,550	70						
	428,000	----	7	8	11	12	15	19
Poo	428,138	60						
	428,215	----						
		70						
LLANES	430,127	----						
		80						
	432,080	----						
San Roque		60						
	432,380	----						
	435,418	80						
	435,560	----						
		60						
	435,750	----						
		80	14	16	17	20	24	30
	438,070	----						
		70						
Vidiago	438,760	----						
	440,255	80						
	440,300	----						
		60						
	441,000	----						
		80						
	442,320	----						
		60						
	442,870	----						
PENDUELES	444,116	80						
	446,240	----						
		70						
	446,900	----						
		80						
	447,500	----						
		60						
	447,850	----	12	13	15	16	19	24
	450,742	80						
Colombres	450,860	----						
		60						
	451,900	----						
		30						
	451,900	----						
		60						
	452,678	----						
		20						
	452,900	----						
		60						
	454,490	----						
	454,579	80						
UNQUERA	457,280	----						
		40						
	457,550	----						
Pesués	457,682	80						
	458,030	----						
		60						
	458,460	----						
		80						
	460,030	----						
		60						
	460,340	----	25	30	32	35	40	47
		80						
	461,020	----						
		60						
	461,210	----						
San Vicente		80						
	464,019	----						
		70						

ESTACIONES	P.K.	V.M.	0501	0503	0505	0507	0509	0511
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
El Barcenal	465,860	50						
	466,840	70						
	466,880	40						
	467,520	70						
	468,069	50						
	468,260	70						
	468,610	70						
	470,200	50						
	471,906	30						
	472,612	50						
Roiz	472,845	70						
	472,940	60						
	475,516	50						
	477,010	60						
	477,890	60						
TRECENO	480,860	50						
	480,917	60						
	481,030	50						
	483,290	50						
	484,490	60						
Turujal	485,103	80						
	486,635	60						
	486,676	40						
	486,770	60						
	487,050	60						
CABEZON DE LA SAL	488,655	10						
	488,655	80						
	488,741	80						
	491,245	60						
	491,400	80						
Ontoria	492,913	60						
	495,340	80						
	495,417	50						
	495,930	80						
	496,860	80						
VIRGEN DE LA PEÑA	498,840	50						
	499,086	80						
	499,200	70						
	499,920	70						
	500,110	70						
CASAR DE PERIEDO	501,107							
	501,107							
Golbardo	501,934							
SAN PEDRO-RUDAG.								
Santa Isabel								
PUENTE SAN MIGUEL								
Firestone								

ESTACIONES	P.K.	V.M.	0501	0503	0505	0507	0509	0511
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Ganzo	502,500	80	4	4	5	5	6	7
Altamira	503,715							
TORRELAVERGA	504,252		3	4	4	5	5	6
BARREDA	506,960							
	508,350	50	3	4	4	4	5	6
	508,500							
REQUEJADA	509,232							
	510,884	80						
Mar								
Gornazo	513,359		8	10	12	14	15	17
	514,090	70						
	514,590							
MOGRO	516,702	80						
Boo	518,166							
	518,520	70	6	8	9	10	11	13
	519,610							
Mortera	520,558	80						
BEZANA	522,207		4	5	6	7	8	9
ADARZO	526,116							
	527,210							
Cazoña	527,247	50	5	6	8	9	9	9
	527,520	60						
	528,600							
Valdecilla	529,223	80						
SANTANDER M. C.T.C	530,004							
SANTANDER V.	530,749		1	1	2	2	2	2

LLOVIO 	0,000	50	5	6	6	6	6	6
REBADESELLA Puerto	2,644							

5.2 Marchas de Asimilación del trayecto Santander – Oviedo y ramal:

ESTACIONES	P.K.	V.M.	0500	0502	0504	0506	0508	0510
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
SANTANDER V.	530,749	----	1	1	2	2	2	2
SANTANDER M. C.T.C	530,004	80						
Valdecilla	529,223							
	528,600	60						
	527,520	----						
Cazoña	527,487	50	5	6	8	9	9	10
	527,210	----						
ADARZO	526,116							
			4	5	6	7	8	9
BEZANA	522,207	80						
Mortera	520,558							
	519,610	70	6	7	8	9	10	12
	518,520	----						
Boo	518,166	80						
MOGRO	516,702							
	514,590	70						
	514,090	----						
Gornazo	513,359	80	8	10	11	13	14	16
Mar	510,884							
REQUEJADA	509,232							
	508,500	50	3	4	4	4	5	6
	508,350	----						
BARREDA	506,960							
			3	4	4	5	5	6
TORRELAVEGA	504,252	80						
Altamira	503,715							
Ganzo	502,500							
Firestone	501,934		4	4	5	5	6	7
PUENTE SAN MIGUEL	501,107							
	500,110	70						
	499,920	80						
	499,200	----	6	8	9	9	10	13
Santa Isabel	499,086	50						
	498,840	80						
	496,860	50						
	495,930	----						
SAN PEDRO-RUDAG.	495,417	80						
	495,340	----						
Golbardo	492,913	60	5	6	7	7	8	10
	491,400	----						
CASAR DE PERIEDO	491,245	80	3	4	4	5	5	6
VIRGEN DE LA PEÑA	488,741							
	488,655	10						
	(vías 3 y 5)	----						
	488,655	60						
	487,050	----						

Ribadesella Puerto - Llovia								
ESTACIONES	P.K.	V.M.	0500	0502	0504	0506	0508	0510
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Ontoria	486,770	40	5	5	6	6	7	9
	486,676	60						
	486,635	80						
CABEZON DE LA SAL	485,103	60						
	484,490	50						
	483,290	60						
Turujal	481,030	50	12	14	16	18	20	22
	480,917	60						
	480,860	50						
TRECEÑO	477,890	60						
	477,010	70						
	475,516	50						
Roiz	472,940	30						
	472,845	50						
	472,612	60						
El Barcenal	471,906	70	13	14	15	16	19	25
	470,200	50						
	468,610	70						
San Vicente	468,260	80						
	468,069	60						
	467,520	70						
Pesués	466,880	50	10	12	14	15	17	20
	466,840	60						
	465,660	80						
UNQUERA	464,019	60						
	461,210	80						
	461,020	60						
	460,340	80						
	460,030	60						
	458,460	80						
	458,030	60						
	457,682	80						
	457,550	40						
	457,280	60						
	454,579	80						
	454,490	20						
	452,900	60						
	452,678	80						
	451,900	60						

ESTACIONES	P.K.	V.M.	0500	0502	0504	0506	0508	0510
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Colombres	451,900	30						
	450,860	60						
	450,742	80						
	447,850	60	12	14	17	18	21	24
	447,500	80						
	446,900	70						
PENDUELES	446,240	80						
	444,116	60						
	442,870	80						
	442,230	60						
	441,000	80						
	440,300	60						
Vidiago	440,255	80						
	438,760	70	14	16	17	20	24	30
	438,070	80						
	435,750	60						
	435,560	80						
	435,418	60						
San Roque	432,380	80						
	432,080	70						
	430,127	60						
	428,215	80						
	428,138	70						
	428,000	60						
Celorio	425,550	70	8	8	14	15	16	19
	424,930	60						
	424,590	70						
	423,415	60						
	420,940	70						
	420,700	80						
Villahormes	417,750	60						
	415,293	70	8	9	13	14	16	18
	412,960	60						
	412,560	70						
	412,450	60						
	411,830	70						
NUEVA	409,640	60						
	407,560	70	10	11	14	16	19	24
	407,560	60						
	403,863	70						
	400,312	50						
	400,180	60						
RIBADESELLA	397,710	3	3	3	3	3	4	6
		60						

Rivadeseña Puerto - Llovio									
ESTACIONES	P.K.	V.M.	0500	0502	0504	0506	0508	0510	
			T-80	T-70	T-60	T-50	T-40	T-30	
			TIEMPO						
			CONC	CONC	CONC	CONC	CONC	CONC	
Llovio	▲	397,588							
	●	396,220	40	6	7	9	10	11	15
	●	396,020							
	●	395,259							
Cuevas	●								
Toraño	▲	390,232	60						
	●								
Fuentes	●	287,629		6	7	11	11	13	15
	●	385,290	40						
	●	384,850							
ARRIONDAS	▲	382,938							
Policlinico Arriondas	●	381,900	60						
Ozanes	●	380,515							
	●	378,250	40	6	6	7	9	11	14
	●	377,950	60						
SOTO DE DUEÑAS	▲	375,910							
	●	375,780	50	4	4	6	6	7	8
	●	374,330	60						
Sebares	▲	372,423	80	3	4	7	8	8	9
	●								
Villamayor	▲	368,361							
	●	363,430							
Infiesto Apdo.	●	362,900	60	6	7	12	12	13	14
	●	361,820							
INFIESTO	▲	361,700	80						
	●	361,460	40						
	●	361,430							
Pintueles	●	359,780	50						
	●	359,720	40						
	●	359,500	60						
	●	359,000	80						
	●	357,170	50	6	7	11	11	12	13
	●	357,060	80						
	●	356,500	70						
	●	356,300							
CARANCOS	▲	355,407	80						
	●								
Ceceda	●	353,740							
	●								
Fuente Santa	●	350,954		8	9	15	16	16	17
	●	350,170	50						
	●	348,480	40						
	●	347,930							
NAVA	▲	347,780							
Llames	●	345,900							
	●								
El Remedio	●	344,080	80	5	6	8	9	10	12

ESTACIONES	P.K.	V.M.	0500	0502	0504	0506	0508	0510
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Rianes	342,656							
	342,120	50	2	2	2	2	3	4
LIERES	341,640	80						
	340,956	40						
	340,790							
	340,570	80	7	8	9	10	12	16
Los Corros	339,210	60						
	333,450							
	333,250	80						
POLA DE SIERO	333,042							
	332,620	40						
La Carrera	331,568	80						
	331,420	60						
	330,900		4	5	6	6	7	9
	330,100							
EL BERRON	329,146		3	3	4	4	5	7
MERES	325,431	80						
	325,400	70						
	324,700	80	4	4	5	6	7	9
	322,300	70						
	321,800							
COLLOTO	321,161	80						
	321,020	40						
	320,860		7	8	14	17	19	22
		80						
OVIEDO	313,800							
RIBADESELLA Puerto	2,644	50	6	6	6	6	6	6
LLOVIO	0,000							

6.1 Marchas de Asimilación del trayecto Santander – Bilbao y ramales:

ESTACIONES	P.K.	V.M.	0601	0603	0605	0607	0609	0611
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
SANTANDER V.	530.749	----	1	1	2	2	2	2
SANTANDER M. C.T.C	531.470	80	-----	-----	-----	-----	-----	-----
Valdecilla	532.291	-----	5	6	7	8	9	9
	532.669	60	-----	-----	-----	-----	-----	-----
	533.770	-----	-----	-----	-----	-----	-----	-----
NUEVA MONTAÑA	534.851	-----	4	5	5	6	7	8
Valle Real	536.250	-----	3	4	4	5	6	7
MALIAÑO	537.900	-----	6	7	8	9	10	12
ASTILLERO	540.075	80	-----	-----	-----	-----	-----	-----
La Cantábrica	540.983	-----	3	4	4	5	6	7
San Salvador	542.115	-----	6	7	8	9	10	12
HERAS	545.290	-----	3	4	5	5	6	7
OREJO	547.836	-----	-----	-----	-----	-----	-----	-----
Puente Agüero	549.700	70	-----	-----	-----	-----	-----	-----
Villaverde de Pontones	551.145	-----	8	10	11	12	12	14
	551.940	60	-----	-----	-----	-----	-----	-----
	552.900	-----	-----	-----	-----	-----	-----	-----
HOZ DE ANERO 	554.396	70	-----	-----	-----	-----	-----	-----
	558.320	60	-----	-----	-----	-----	-----	-----
	558.610	70	10	12	13	14	15	18
	559.890	60	-----	-----	-----	-----	-----	-----
	560.550	-----	-----	-----	-----	-----	-----	-----
BERANGA 	562.301	70	-----	-----	-----	-----	-----	-----
	563.600	50	-----	-----	-----	-----	-----	-----
	564.590	70	-----	-----	-----	-----	-----	-----
	565.160	50	5	6	7	8	9	11
	565.650	70	-----	-----	-----	-----	-----	-----
GAMA 	566.995	80	-----	-----	-----	-----	-----	-----
	569.200	-----	-----	-----	-----	-----	-----	-----
Cicero	569.913	60	7	8	9	10	11	14
	570.900	80	-----	-----	-----	-----	-----	-----
TRETO 	573.389	70	-----	-----	-----	-----	-----	-----
	573.700	50	-----	-----	-----	-----	-----	-----
	574.360	70	-----	-----	-----	-----	-----	-----
	575.010	60	-----	-----	-----	-----	-----	-----
	575.360	70	-----	-----	-----	-----	-----	-----
	576.210	-----	-----	-----	-----	-----	-----	-----

ESTACIONES	P.K.	V.M.	0601	0603	0605	0607	0609	0611
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Limpías	576.672	50						
	578.648	70	10	11	12	13	16	19
	579.660	50						
MARRÓN	579.910	50						
	581.333	70						
	581.890	50						
	582.030	70						
	582.960	50						
	584.550	70						
Udalla	586.251	50	10	12	14	15	16	21
	586.720	70						
	587.500	50						
	589.250	70						
	589.650	60						
	590.728	70						
GIBAJA	594.080	50						
	594.960	70						
	596.330	50						
	596.900	70	11	13	15	15	16	20
	597.570	50						
	597.750	70						
CARRANZA	599.609	60						
	603.530	50						
	604.580	60						
	608.220	50						
	609.070	60	24	28	34	36	37	41
	610.810	50						
Vde. de Trucios El Peso	612.015	60						
	613.830	40						
	614.230	60						
Arcentales	615.193	40						
	615.530	60						
	615.730	40						
	616.140	60						
	616.650	40						
	616.851	60						
TRASLAVIÑA Ocharan ARANGUREN	619.562	50	12	13	13	13	15	19
	625.710	70						

ESTACIONES	P.K.	V.M.	0601	0603	0605	0607	0609	0611
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
	626.570	---						
	60	---						
	626.894	---						
Güeñes	626.942	---						

Lámbarni	628.149	70	7	9	10	11	12	13

Archube	629.283	---						
	630.005	---						
	60	---						
	630.369	---						
SODUPE	631.489	---						

	70	---						
La Cuadra	634.803	---	6	8	9	10	11	12
	635.167	---						
ZARAMILLO	636.830	50						
	636.997	---	4	5	6	7	7	8
IRAUREGUI	639.793	70						
	640.070	---						
	50	---						
	640.377	---						
	70	---						
	641.432	---	6	9	10	11	12	12
Castrejana	641.754	60						
Santa Águeda	642.598	---						
	643.242	---						
ZORROZA	644.100	---						
		---	2	3	3	4	4	6
BASURTO	646.864	70						
Bilbao Amézola	647.707	---						
	648.166	---	3	5	5	6	6	7
	60	---						
	649.152	---						
BILBAO CONC. C.T.C	649.235	70						

OREJO	0.149	---						

SOLARES	1.734	70	3	3	4	4	5	5
	1.796	---						
	10	---						
	1.796	---						
	70	---						
	2.000	---						
	40	---						
	2.660	---						
Geceñas	3.544	70	10	12	15	16	17	18
	6.040	---						
La Cavada	6.807	50						
	7.020	---						
	70	---						
LIÉRGANES	9.723	---						

IRAUREGUI	0.000	---						

	50	---	8	9	10	10	11	13
LUCHANA	5.719	---						

BASURTO	0.000	---						

	50	---	10	12	15	15	16	17
ARIZ	7.906	---						

6.2 Marchas de Asimilación del trayecto Bilbao - Santander y ramales:

ESTACIONES	P.K.	V.M.	0600	0602	0604	0606	0608	0610
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
BILBAO CONC. C.T.C	649.235	70						
	649.152	-----						
	60		3	3	4	4	4	5
Bilbao Amézola	648.166	-----						
BASURTO	647.707							
	646.864	70						
			2	2	2	3	3	3
ZORROZA	644.100							
	643.242	-----						
Santa Águeda	642.598							
Castrejana	641.754	60						
	641.432	-----	10	11	12	13	13	15
	60							
	640.377	-----						
	50							
	640.070	-----						
IRÁUREGUI	639.793	70						
	638.997	-----	3	5	6	7	8	9
ZARAMILLO	636.630	50						
	636.167	-----						
La Cuadra	634.803		6	8	9	10	11	12
	60							
SODUPE	631.489							
	630.369	-----						
	60							
	630.005	-----						
Archube	629.283							
	60							
Lámbarri	628.149	70	7	9	11	12	13	14
	60							
Güeñes	626.942							
	626.894	-----						
	60							
	626.570	-----						
	70							
ARANGUREN	625.710							
Ocharan	619.562	50	14	16	21	22	22	22
TRASLAVIÑA	616.851	-----						
	60							
	616.650	-----						
	40							
	616.140	-----						
	60							
	615.730	-----						
	40							
	615.530	-----						
Arcentales	615.193	60						
	614.230	-----						
	50							
	613.830	-----						
El Peso	612.015	60	24	28	31	33	34	37
Vde. de Trucios	610.810							
	609.070	-----						
	50							
	608.220	-----						
	60							
	604.580	-----						
	50							
	603.530	-----						
	60							
CARRANZA	599.608							

ESTACIONES	P.K.	V.M.	0600 T-80	0602 T-70	0604 T-60	0606 T-50	0608 T-40	0610 T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
GIBAJA 		70						
	597.750	50						
	597.570	70						
	596.900	50	11	12	13	14	15	20
	596.330	70						
	594.960	50						
	594.080	70						
	590.728	50						
	589.650	70						
	589.250	60						
	587.500	70						
		50						
Udalla	586.720	70	11	12	13	14	16	21
	584.550	50						
	582.960	70						
	582.030	50						
	581.890	70						
MARRÓN 	581.333	50						
	579.910	70						
Limpías	579.660	70						
	578.648	50						
	576.672	70						
	576.210	60						
	575.360	70	10	11	12	13	16	18
TRETO 	575.010	50						
	574.360	70						
	573.700	80						
	573.389	60						
	570.900	70						
Cicero	569.913	80	7	8	9	10	11	14
	569.200	70						
GAMA 	566.995	50						
	565.650	70						
	565.160	50						
BERANGA 	564.590	70	6	7	9	10	10	11
	563.600	50						
	562.301	70						

ESTACIONES	P.K.	V.M.	0600	0602	0604	0606	0608	0610
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
HOZ DE ANERO 	560.550	----						
	559.890	70	9	11	13	14	15	17
	558.610	60						
	558.320	70						
	554.396	70						
	552.900	60						
	551.940	----						
	551.145	70	7	9	9	10	12	14
	549.700							
	547.836	----	3	4	4	5	6	7
HERAS	545.290							
San Salvador	542.115	80	6	7	8	9	10	11
La Cantábrica	540.983							
ASTILLERO	540.075		3	4	4	5	6	6
MALIAÑO	537.900		4	5	5	6	7	8
Valle Real	536.250							
NUEVA MONTAÑA	534.851	60						
	533.770	----	5	6	7	8	9	9
Valdecilla	532.669							
	532.291							
SANTANDER M. C.T.C	531.470	80	1	1	2	2	2	2
SANTANDER V.	530.749							
LIERGANÉS	9.723	70						
La Cavada	6.807	50	10	12	12	13	15	18
Ceceñas	6.040	70						
	3.544	40						
	2.660	70						
	2.000	10						
	1.796	70						
	1.796	10						
SOLARES	1.734	70	3	3	4	4	5	5
OREJO	0.149	----						
LUCHANA 	5.719	50	8	10	12	12	13	14
IRAUREGUI	0.000	----						
ARIZ 	7.906	50	10	12	14	14	15	17
BASURTO	0.000	----						

7.1 Marchas de Asimilación del trayecto León – Aranguren y ramal:

ESTACIONES	P.K.	V.M.	0700	0702	0704	0706	0708	0710
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
ARANGUREN	0.000	70						
Aranguren Apd.	0.367							
	1.500	60	2	2	3	3	4	5
	1.800							
ZALLA	2.058							
Colegio	2.549	70						
Ibarra	3.485							
	3.900							
Boiumburu	4.247	50						
	5.300		8	8	8	9	11	14
La Herrera	6.705	60						
	6.750	40						
	7.100							
	8.831	70						
BALMASEDA V.	9.173		1	1	2	2	2	2
BALMASEDA M.								
Arla - Berrón	283.611							
	278.964		12	15	18	18	18	21
Ungo - Nava	273.815							
Menamayor	270.173		8	10	13	13	13	13
Mercadillo - Villasana	268.223							
Anzo	265.104	50						
	261.068		15	18	24	24	24	24
Vigo - Siones								
Cadagua	258.140							
Cantonad	255.113							
La Silla	251.998		19	23	33	33	33	33
BERCEDO - MONTIJA	245.233							
	241.790	60						
	241.280	60						
Quintana de los Prados	240.520		8	9	10	11	12	15
	238.448	60						
Espinosa de los M.	236.340	50						
	235.620	40						
Redondo	231.276		19	20	22	22	25	29
	225.225	50						
SOTOSCUEVA	223.200							
	223.200	30	9	10	12	12	14	15
PEDROSA	218.754							
Dosante - Ciudad	215.674	50						
Robredo - Ahedo	211.030		18	21	28	28	30	30
SONCILLO	205.445							

ESTACIONES	P.K.	V.M.	0700	0702	0704	0706	0708	0710
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Cabañas	203.604	-----						
		70						
	203.490	-----	12	15	17	18	20	23
		50						
	203.300	-----						
		70						
ARIJA	197.610	-----						
		60						
	196.362	-----						
		70						
	194.790	-----						
		70						
Llano	193.710	-----						
		60						
	192.743	-----						
		70						
	192.260	-----						
		70						
	191.380	-----						
		60						
	190.670	-----	28	28	31	39	46	55
		70						
Las Rozas	188.281	-----						
		50						
	185.560	-----						
		40						
	185.150	-----						
Montes Claros		50						
	181.025	-----						
		40						
	180.010	-----						
		50						
Los Carabeos	176.980	-----						
		50						
	173.729	-----						
		60						
	173.450	-----						
		50						
	172.200	-----						
		60	11	11	13	16	19	23
	166.130	-----						
		50						
	165.550	-----						
		60						
Mataporquera	164.393	-----						
Cordovilla		50						
	155.310	-----						
		20	17	17	18	23	27	32
	152.630	-----						
		50						
Cillamayor	152.430	-----						
		70						
	151.165	-----						
		60						
	148.750	-----						
		70						
	147.830	-----						
		60	11	11	13	16	19	23
	143.780	-----						
		70						
	143.300	-----						
		60						
Salinas de P.	142.060	-----						
		70						
	141.830	-----						
		60						
	141.580	-----						
		70						
	139.970	-----						
		60						
	139.800	-----						
		70						
	139.050	-----						
		40						

matallana - La Robla								
ESTACIONES	P.K.	V.M.	0700	0702	0704	0706	0708	0710
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Vado Cervera ▲	138.910	-----	13	13	15	19	22	27
		70						
	138.310	-----						
		40						
	138.000	-----						
		70						
	134.580	-----						
		40						
	133.500	-----						
	130.936	70						
Castrejón de la P.	130.050	-----						
		50						
	127.070	-----						
	120.264	70						
	117.960	-----						
		40						
	117.680	-----	27	27	32	41	48	57
		70						
	115.360	-----						
		50						
Vda. Tarilonte	115.250	-----						
	113.120	70						
	113.080	-----						
		50						
	111.900	-----						
		70						
	109.800	-----						
		50						
	108.320	-----						
	107.299	70						
Santibañez de la P. ▲	105.250	-----						
		50						
	104.420	-----						
		70						
	99.530	-----	12	12	14	18	21	25
		50						
	98.750	-----						
	98.324	70						
	97.016	-----						
		60						
Guardo Apdo.	94.730	-----						
		50						
	92.980	-----						
		60						
	91.630	-----			1			
		50						
	90.650	-----						
		60						
	88.760	-----	28	32	38	40	42	50
		50						
La Espina ▲	88.010	-----						
	86.547	-----						
		50						
	83.510	-----						
Valcuende	79.220	-----						
		50						
	77.990	-----						
	74.808	-----						
PUNTE ALMUHEY ▲		60						
	69.565	-----						
	68.120	-----						
		50						
Prado Guzpeña	65.880	-----						
	63.203	60						
	62.210	-----						

ESTACIONES	P.K.	V.M.	0700	0702	0704	0706	0708	0710
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
	61.450	50	23	30	34	37	39	45
	60.820	60						
	60.290	40						
	58.510	50						
	57.630	60						
	57.320	50						
CISTERNA	54.272	60						
	54.040							
Yugueros	46.235	50						
	46.000							
La Ercina	43.577		31	39	46	50	52	55
Barrillos	37.623	60						
La Losilla	33.250							
BOÑAR	30.185							
La Mata de la Riba	28.670							
Otero	23.820		13	16	19	19	21	21
Valdepiélagos	22.441	50						
LA VECILLA	20.925							
Campohermoso	19.418							
Aviados	18.237		14	16	19	19	21	22
La Valcueva	14.376							
	12.590	40						
	12.480							
Robles	12.176							
	10.867	50						
MATALLANA	28.376	70						
	28.214							
	27.990	50						
	27.650	70						
Naredo	27.543	50						
	27.490							
Pardavé	23.924	70	17	20	23	25	30	39
Pedrún	22.188							
	20.050							
Matueca	20.011	50						
	19.930							
Manzaneda	18.077							
Garrate	16.736	70						
Palazuelo	12.586							
SAN FELIZ	9.747							
	9.563	50						

ESTACIONES	P.K.	V.M.	0700	0702	0704	0706	0708	0710
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Villasinta	9.400	----						
	6.820							
Villaquilambre	5.589	70	12	13	14	15	17	21
La Raya	4.544							
Hospitales	1.470							
	0.500	----						
		40						
	0.350	----						
LEÓN	0.000	70						
MATALLANA	10.867	50						
	10.760	----						
		40	15	18	20	23	25	30
	10.460	----						
LA ROBLA	0.000	50						

7.2 Marchas de Asimilación del trayecto Aranguren - León y ramal:

ESTACIONES	P.K.	V.M.	0701	0703	0705	0707	0709	0711
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
LEÓN 	0,000	70						
	0,350	-----						
		40						
	0,500	-----						
	1,470							
Hospitales								
La Raya	4,544							
Villaquilambre	5,589	70	12	13	14	15	17	21
Villasinta	6,820							
	9,400	-----						
		50						
SAN FELIZ 	9,563	-----						
	9,747							
Palazuelo	12,586							
Garrafe	15,736	70						
Manzaneda	18,077							
	19,930	-----						
Matueca	20,011	50						
	20,050	-----						
Pedrún	22,188		17	21	24	27	32	39
		70						
Pardavé	23,924							
	27,490	-----						
Naredo	27,543	50						
	27,650	-----						
		70						
	27,990	-----						
		50						
	28,214	-----						
	28,376	70						
MATALLANA 		-----						
	10,867	50						
Robles	12,176							
	12,480	-----						
		40						
	12,590	-----						
La Valcueva	14,376		15	17	20	20	22	22
Aviados	18,237							
Campohermoso	19,418							
LA VECILLA 	20,925	50						
Valdepiélagos	22,441							
Otero	23,820		12	14	16	16	18	20
La Mata de la Riba	28,670							
BOÑAR 	30,185	-----						
La Losilla	33,250							
		60						
Barrillos	37,623							
La Ercina 	43,577		31	39	47	49	51	55
	46,000	-----						
Yugueros	46,235	50						
	54,040	-----						

ESTACIONES	P.K.	V.M.	0701	0703	0705	0707	0709	0711
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
CISTERNA 	54,272	60						
	67,320	50						
	57,630	60						
	58,510	50						
	60,290	40						
	60,820	60	24	31	40	41	43	45
	61,450	50						
	62,210							
	63,203	60						
	65,880	50						
Valle de las Casas	68,120							
Prado Guzpeña	69,565	60						
PUNTE ALMUHEY 	74,808							
	77,990	50						
	79,220							
Valcuende	83,510	60						
La Espina 	86,547							
	88,010	50						
	88,760	60	30	37	44	46	49	51
	90,650	50						
	91,630	60						
	92,980	50						
	94,730	60						
GUARDO 	97,016							
Guardo Apdo.	98,324	70						
	98,750	50						
	99,530	70	12	12	14	18	21	25
	104,420	50						
	105,250							
Santibañez de la P. 	107,299	70						
	108,320	50						
	109,800	70						
	111,900	50						
	113,080							
Vde. Tarilonte	113,120	70						
	115,250	50						
	115,360	70						
	117,680	40	27	27	32	41	48	57
	117,960							
Castrejón de la P.	120,264	70						

ESTACIONES	P.K.	V.M.	0701	0703	0705	0707	0709	0711
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Vado Cervera 	127.070	-----						
		50						
	130.050	-----						
	130.936	70						
	133.500	-----						
		40						
	134.580	-----						
		70						
	138.000	-----						
		40						
	138.310	-----						
		70						
Salinas de P. 	138.910	-----	13	13	15	19	22	27
		40						
	139.050	-----						
		70						
	139.800	-----						
		60						
	139.970	-----						
		70						
	141.580	-----						
	141.830	60						
	142.060	-----						
		70						
Cillamayor 	143.300	-----						
		60						
	143.780	-----	11	11	13	16	19	23
		70						
	147.830	-----						
		60						
	148.750	-----						
		70						
	151.165	-----						
		50						
	152.430	-----						
		20						
Cordovilla	152.630	-----	17	17	18	23	27	32
	155.310	50						
Mataporquera 	164.393	-----						
		60						
	165.550	-----						
		50						
	166.130	-----	11	11	13	16	19	23
		60						
Los Carabeos 	172.200	-----						
		50						
	173.450	-----						
		60						
	173.729	-----						
		50						
Montes Claros	176.980	-----						
		40						
	180.010	-----						
	181.025	50						
	185.150	-----						
		40						
Las Rozas	185.560	-----						
		50						
	188.281	-----						
		70						
	190.670	-----	28	28	31	39	46	55
		60						
	191.380	-----						
		70						
	192.280	-----						

La Robla - Matallana								
ESTACIONES	P.K.	V.M.	0701	0703	0705	0707	0709	0711
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Llano	192.743	60						
	193.710	70						
ARIJA	194.790							
	196.362	60						
	197.610	70						
	203.300	50	12	15	17	18	20	22
	203.490	70						
Cabañas	203.604							
SONCILLO	205.445							
Robredo - Ahedo	211.030							
Dosante - Ciudad	215.674	50	18	19	20	20	22	29
PEDROSA	218.754							
	223.200	30						
	223.200		9	11	14	14	16	16
SOTOSCUEVA	225.225	50						
Redondo	231.276							
	235.620	40	19	20	22	22	25	29
	236.340	50						
Espinosa de los M.	238.448	60						
Quintana de los Prados	240.520							
	241.280	50	8	9	10	11	12	15
	241.790	60						
BERGEDO - MONTIJA	245.233							
La Silla	251.996							
Cantonad	255.113		17	20	22	22	24	28
Cadagua	258.140							
Vigo - Siones	261.068							
Anzo	265.104	50	14	15	15	15	17	22
Mercadillo - Villasana	268.223							
Menamayor	270.173		8	8	8	8	10	12
Ungo - Nava	273.815							
Arla - Berrón	278.964		12	13	13	13	15	21
	283.611							
BALMASEDA M.	9.173		1	1	2	2	2	2
BALMASEDA V.	8.831	70						
	7.100	40						
	6.750		8	8	8	9	11	14
La Herrera	6.705	60						
	5.300							

ESTACIONES	P.K.	V.M.	0701	0703	0705	0707	0709	0711
			T-80	T-70	T-60	T-50	T-40	T-30
			TIEMPO					
			CONC	CONC	CONC	CONC	CONC	CONC
Bolumburu	4.247	50						
	3.900	-----						
Ibarra	3.485							
Colegio	2.549	70						
ZALLA	2.058							
	1.800	-----						
	1.500	60						
Aranguren Apd.	0.367	-----	2	2	3	3	4	5
ARANGUREN	0.000	70						

LA ROBLA	0,000	50						
	10,460	-----						
		40	15	18	20	23	25	30
	10,760	-----						
MATALLANA	10,867	50						

ANEXO VI: ORDENES DEL DÍA DE LOS TRENES **GRAFIADOS**

1. ORDEN DEL DÍA N° 105 / 2014

LIBRO ITINERARIO DE ASTURIAS

1.- OBJETO.

Supresión, Creación y Anuncio de trenes del Libro Itinerario de Asturias.

2.- SUPRESIÓN.

A partir del **domingo, día 14 de diciembre de 2014**, se suprimen del Libro Itinerario de Asturias los siguientes trenes:

• 2001, 2401, 2403, 2405, 2407, 2409, 2411, 2413, 2415, 2417, 2419, 2421, 2423, 2425, 2427, 2429, 2431, 2433, 2435, 2601, 2603, 2605, 2801, 2803, 2805, 2807, 2809, 2811, 2813, 2815, 2817, 2819, 2821, 2823, 2859, 2861, 2863, 3301, 3303, 3305, 3307, 3309, 3311, 3313, 3315, 3317, 3319, 3321, 3803, 3811, 3813, 3817, 3819, 3821, 3825, 3827, 3829, 3833, 3835, 3837, 3839, 3843, 3845, 3847, 3891, 3895, 3901, 3903, 3951, 3953, 3955, 3957, 3959, 3961, 3963, 3965, 3967, 3969, 3971, 3973, 3975, 3977, 3979, 3981, 4501, 4503, 4505, 4507, 4509, 4511, 4513, 4515, 4517, 4519, 4521, 4523, 4901, 4903, 4905, 4907, 4909, 4911, 4913, 4915, 4917, 4919, 4921, 4923, 4925, 4927, 4929, 4931, 4933, 5105, 5107, 5109, 5111, 5301, 5303, 5305, 5307, 5309, 5311, 5313, 5315, 5317, 5319, 5401, 5403, 5405, 5407, 5409, 5411, 5413, 5415, 5417, 5419, 5421, 5423, 5491, 5493, 5495, 5497, 5501, 5503, 5505, 5507, 5509, 5511, 5513, 5515, 5517, 5701, 5703, 5705, 5707, 5709, 5711, 5713, 5715, 5717, 5719, 5721, 5723, 5725, 5727, 5729, 5731, 5733, 5901, 5903, 9201, 9203, 9241, 9243, 9245, 9301, 9353, 9431, 9433, 9435, 9437 y 9439.

• 2000, 2040, 2400, 2402, 2404, 2406, 2408, 2410, 2412, 2414, 2416, 2418, 2420, 2422, 2424, 2426, 2428, 2430, 2432, 2450, 2600, 2602, 2800, 2802, 2804, 2806, 2808, 2810, 2812, 2814, 2816, 2818, 2820, 2822, 2824, 2860, 2862, 3200, 3300, 3302, 3304, 3306, 3308, 3310, 3312, 3314, 3316, 3318, 3320, 3802, 3804, 3810, 3814, 3816, 3818, 3820, 3822, 3824, 3828, 3830, 3832, 3834, 3838, 3840, 3842, 3846, 3900, 3950, 3952, 3954, 3956, 3958, 3960, 3962, 3964, 3966, 3968, 3970, 3972, 3974, 3976, 3978, 4502, 4504, 4506, 4508, 4510, 4512, 4514, 4516, 4518, 4520, 4522, 4524, 4550, 4552, 4554, 4556, 4900, 4902, 4904, 4906, 4908, 4910, 4912, 4914, 4916, 4918, 4920, 4922, 4924, 4926, 4928, 4930, 4932, 4950, 5100, 5102, 5104, 5106, 5108, 5300, 5302, 5304, 5306, 5308, 5310, 5312, 5314, 5316, 5318, 5320, 5402, 5404, 5406, 5408, 5410, 5412, 5414, 5416, 5418, 5420, 5422, 5424, 5500, 5502, 5504, 5506, 5508, 5510, 5512, 5514, 5516, 5700, 5702, 5704, 5706, 5708, 5710, 5712, 5714, 5716, 5718, 5720, 5722, 5724, 5726, 5728, 5730, 5732, 5900, 5902, 9200, 9202, 9300, 9420, 9422, 9430, 9432, 9434, 9436, 9438, 9530 y 9532.

3.- CREACIÓN.

Relación de trenes que se crean en la presente Orden del Día:

Tren nº		Origen - Destino	Periodicidad
73201	Mercancías	Pravia-Trasona	Diario
73203	Mercancías	Pravia-Trasona	Laborable de lunes a viernes
73211	Mercancías	Trasona-El Berrón	Diario
73213	Mercancías	Trasona-El Berrón	Sábados, domingos y festivos
73215	Mercancías	Trasona-El Berrón	Diario
73221	Mercancías	Sueros-Soto de Ribera	Laborable de lunes a viernes
73231	Mercancías	Pravia-El Berrón	Laborable de lunes a viernes
73241	Mercancías	Aboño Mercancías-El Berrón	Diario
73243	Mercancías	Soto de Ribera-El Berrón	Diario
73245	Mercancías	Aboño Mercancías-El Berrón	Diario
73247	Mercancías	Soto de Ribera-El Berrón	Diario
73249	Mercancías	Aboño Mercancías-El Berrón	Diario

Tren nº		Origen - Destino	Periodicidad
73200	Mercancías	Trasona-Pravia	Laborable de lunes a viernes
73202	Mercancías	Trasona-Pravia	Laborable de lunes a viernes
73210	Mercancías	El Berrón-Trasona	Diario
73212	Mercancías	El Berrón-Trasona	Diario
73220	Mercancías	Soto de Ribera-Sueros	Laborable de lunes a viernes
73240	Mercancías	El Berrón-Aboño Mercancías	Diario
73242	Mercancías	El Berrón-Soto de Ribera	Diario
73244	Mercancías	El Berrón-Aboño Mercancías	Diario
73246	Mercancías	El Berrón-Soto de Ribera	Diario
73248	Mercancías	El Berrón-Aboño Mercancías	Diario

2. ORDEN DEL DÍA Nº 28 / 2015

LIBRO ITINERARIO DE CANTABRIA

1.- OBJETO.

Supresión, creación y anuncio de trenes del Libro Itinerario de Cantabria.

2.- SUPRESIÓN.

A partir del **domingo, día 14 de junio de 2015**, se suprimen del Libro Itinerario de Cantabria todos los trenes creados por la **Orden del Día nº 108/2014 GATR.**

3.- CREACIÓN.

Relación de trenes que se crean en la presente Orden del Día:

Tren nº		Origen - Destino	Periodicidad
73401	Mercancías	Santander M. – Maliaño La Vidriera	Laborable de lunes a viernes
73403	Mercancías	Barreda – Maliaño La Vidriera	Laborable de lunes a viernes
73405	Mercancías	Barreda – Maliaño La Vidriera	Laborable de lunes a viernes
73407	Mercancías	Barreda – Maliaño La Vidriera	Laborable de lunes a viernes
73409	Mercancías	Barreda – Santander M.	Laborable de lunes a viernes

Tren nº		Origen - Destino	Periodicidad
73400	Mercancías	Santander M. – Barreda	Laborable de lunes a viernes
73402	Mercancías	Maliaño La Vidriera – Barreda	Laborable de lunes a viernes
73404	Mercancías	Maliaño La Vidriera – Santander M.	Laborable de lunes a viernes
73406	Mercancías	Santander M. – Barreda	Laborable de lunes a viernes
73408	Mercancías	Maliaño La Vidriera – Barreda	Laborable de lunes a viernes
73410	Mercancías	Maliaño La Vidriera – Santander M.	Laborable de lunes a viernes
73412	Mercancías	Maliaño La Vidriera – Santander M.	Laborable de lunes a viernes

3. ORDEN DEL DÍA Nº 62 / 2015

LIBRO ITINERARIO DE TRENES COMUNES: GALICIA Y ASTURIAS

1.- OBJETO.

Supresión, Creación y Anuncio de trenes del Libro Itinerario de Trenes Comunes de Galicia y Asturias, con motivo de la reclasificación de Foz de apeadero a estación

2.- SUPRESIÓN.

A partir del **jueves, día 15 de octubre de 2015**, se suprimen del Libro Itinerario de Trenes Comunes de Galicia y Asturias los siguientes trenes:

- 73300.

3.- CREACIÓN.

Relación de trenes que se crean en la presente Orden del Día:

Tren		Origen - Destino	Periodicidad
73300	Mercancías	Pravia – Xove	Domingo a viernes

4. ORDEN DEL DÍA Nº 1 / 2016

LIBRO ITINERARIO DE TRENES COMUNES: GALICIA, ASTURIAS Y CANTABRIA

1.- OBJETO.

Supresión, creación y anuncio de trenes del Libro Itinerario de Trenes Comunes de Galicia, Asturias y Cantabria.

2.- SUPRESIÓN.

A partir del **domingo, día 7 de febrero de 2016**, se suprime del Libro Itinerario de Trenes Comunes de Galicia, Asturias y Cantabria el siguiente tren:

- 73151

3.- CREACIÓN.

Se crea el siguiente tren regular en la presente Orden del Día:

Tren		Origen - Destino	Periodicidad
73151	Mercancías	Xove – Santander M.	Lunes a sábado
		El Berrón – Santander M.	Domingos

5. ORDEN DEL DÍA Nº 3 / 2016

LIBRO ITINERARIO DE ASTURIAS

1.- OBJETO.

Supresión, Creación y Anuncio de trenes del Libro Itinerario de Asturias.

2.- SUPRESIÓN.

A partir del **domingo, día 7 de febrero de 2016**, se suprimen del Libro Itinerario de Asturias los siguientes trenes:

- 73211, 73213, 73215, 73221, 73231, 73241, 73243, 73245, 73247, 73249.
- 73210, 73212, 73220, 73240, 73242, 73244, 73246, 73248.

3.- CREACIÓN.

Relación de trenes que se crean en la presente Orden del Día:

Tren nº		Origen - Destino	Periodicidad
73211	Mercancías	Trasona - El Berrón	Diario
73213	Mercancías	Trasona - El Berrón	Sábados, domingos y festivos
73215	Mercancías	Trasona - El Berrón	Diario
73231	Mercancías	Pravia - El Berrón	Laborable de lunes a viernes
73245	Mercancías	Aboño Mercancías - El Berrón	Diario
73247	Mercancías	Soto de Ribera - El Berrón	Diario

Tren nº		Origen - Destino	Periodicidad
73210	Mercancías	El Berrón - Trasona	Diario
73212	Mercancías	El Berrón - Trasona	Diario
73244	Mercancías	El Berrón - Aboño Mercancías	Diario
73246	Mercancías	El Berrón - Soto de Ribera	Diario

6. ORDEN DEL DÍA Nº 4 / 2016

LIBRO ITINERARIO DE TRENES COMUNES: ASTURIAS Y CANTABRIA

1.- OBJETO.

Supresión, creación y anuncio de trenes del Libro Itinerario de Trenes Comunes de Asturias y Cantabria.

2.- SUPRESIÓN.

A partir del **domingo, día 7 de febrero de 2016**, se suprimen del Libro Itinerario de Trenes Comunes de Asturias y Cantabria los siguientes trenes:

- 73351.
- 73500, 73502 y 73504.

3.- CREACIÓN.

Relación de trenes regulares que se crean en la presente Orden del Día:

Tren		Origen – Destino	Periodicidad
73351	Mercancías	El Berrón – Santander M.	Diario
73500	Mercancías	Santander M. – Pravia	Diario
73502	Mercancías	Santander M. – Pravia	Diario
73504	Mercancías	Santander M. – El Berrón	Diario

7. ORDEN DEL DÍA Nº 40 / 2016

LIBRO ITINERARIO DE TRENES COMUNES: ASTURIAS, CANTABRIA Y BIZKAIA

1.- OBJETO.

Supresión, creación y anuncio de trenes del Libro Itinerario de Trenes Comunes de Asturias, Cantabria y Bizkaia.

2.- SUPRESIÓN.

A partir del **domingo, día 4 de septiembre de 2016**, se suprime del Libro Itinerario de Trenes Comunes de Asturias, Cantabria y Bizkaia el siguiente tren:

- 73353

3.- CREACIÓN.

Se crea el siguiente tren regular en la presente Orden del Día:

Tren		Origen - Destino	Periodicidad
73353	Mercancías	El Berrón – Aranguren	Diario

8. ORDEN DEL DÍA Nº 41 / 2016

LIBRO ITINERARIO DE TRENES COMUNES: CANTABRIA Y BIZKAIA

1.- OBJETO.

Supresión, creación y anuncio de trenes del Libro Itinerario de Trenes Comunes de Cantabria y Bizkaia.

2.- SUPRESIÓN.

A partir del **domingo, día 4 de septiembre de 2016**, se suprimen del Libro Itinerario de trenes comunes Cantabria y Bizkaia, los siguientes trenes:

- 71851, 71853, 71855, 73561, 73563 y 73565.
- 71870, 71872, 71874, 73760, 73762, 73764 y 73766.

3.- CREACIÓN.

Relación de trenes regulares que se crean en la presente Orden del Día:

Tren nº		Origen - Destino	Periodicidad
73561	Mercancías	Santander M - Áriz	Sábados, domingos y festivos
73563	Mercancías	Santander M - Áriz	Laborable de lunes a viernes
73565	Mercancías	Santander M - Aranguren	Diario
73760	Mercancías	Aranguren - Santander M	Diario
73762	Mercancías	Aranguren - Santander M	Sábados, domingos y festivos
73764	Mercancías	Áriz - Santander M	Laborable de lunes a viernes
73766	Mercancías	Aranguren - Santander M	Diario

9. ORDEN DEL DÍA Nº 44 / 2016

LIBRO ITINERARIO DE BIZKAIA

1.- OBJETO.

Supresión, Creación y Anuncio de trenes del Libro Itinerario de Bizkaia.

2.- SUPRESIÓN.

A partir del **domingo, día 04 de septiembre de 2016**, se suprimen del Libro Itinerario de Bizkaia, los siguientes trenes:

- 73601, 73603, 73605, 73611, 73701 y 73703.
- 73600, 73602, 73612, 73700 y 73702.

3.- CREACIÓN.

Relación de trenes que se crean en la presente Orden del Día:

Tren nº		Origen - Destino	Periodicidad
73603	Mercancías	Aranguren – Lutzana Barakaldo	Sábados
73605	Mercancías	Aranguren – Lutzana Barakaldo	Lunes a viernes
73607	Mercancías	Balmaseda M – Ariz	Laborable de lunes a viernes
73701	Mercancías	Arija – Balmaseda M	Laborable de lunes a viernes

Tren nº		Origen - Destino	Periodicidad
73602	Mercancías	Lutzana Barakaldo– Aranguren	Lunes a sábado
73604	Mercancías	Ariz – Aranguren	Sábados, domingos y festivos
73606	Mercancías	Ariz – Balmaseda M	Laborable de lunes a viernes
73700	Mercancías	Balmaseda M – Arija	Laborable de lunes a viernes

ANEXO VII: VELOCIDADES, NUMERO DE EJES Y CARGA DE LOS TRENES

La velocidad máxima que pueden alcanzar los trenes, en las condiciones más favorables de trazado y de clase de vía (artículo 4.01.00 del Reglamento de Circulación de Trenes), está condicionada por la clase de material y por su sistema de frenado conforme a la siguiente clasificación:

<i>Tipo de</i>	<i>Condiciones del material y del</i>
T.3	Sin condiciones
T.4	Freno automático de vacío o freno automático de aire
T.50	Freno automático de aire Freno automático de aire comprimido Freno automático de aire
T.80	Unidades automotoras de viajeros
T.90	Unidades automotoras de viajeros, serie UTDH 2700, UTDH 2900 y UTE 3600. Unidades automotoras de viajeros, serie UTDH 2700, UTDH 2900 y UTE 3600

El número máximo de ejes que se puede alcanzar en la composición de unidades remolcadas, para trenes de mercancías, es el que figura seguidamente, de acuerdo con las características y régimen del sistema de frenado y el tipo de tren de que se trate.

SISTEMA DE FRENADO					
TIPOS	A mano (total, parcial)	Por vacío		Por aire comprimido	
		Sólo Material vacío	Otras composiciones	Régimen de mercancías	Régimen de viajeros
T.30	40	60	40	140	100
T.40	(1)	60	40	(3)	100
T.50	(1)	(2)	(2)	(3)	100
T.60	(1)	(2)	(2)	(3)	80
T.70	(1)	(2)	(2)	(3)	80

Dado que en la práctica todos los vagones circulan en régimen de viajeros, en el presente proyecto se va obviar el hecho de que los vagones puedan circular en régimen de mercancías.

La carga máxima de los vagones es la que, de acuerdo con las características establecidas por el constructor, figura en la placa, inscripción o marca correspondiente en el propio vagón. A los efectos del servicio, la carga máxima admisible en un vagón es la indicada anteriormente mas un incremento del 5% por razones de tolerancia.

La carga máxima admisible por cada eje de rodadura es de 15 toneladas, en toda la red. A los efectos del servicio, la carga máxima admisible por cada eje de rodadura es la indicada anteriormente, más un incremento de 500Kgrs. por razones de tolerancia. Teniendo en cuenta que cada vagón tiene 4 ejes, la carga máxima admitida resulta ser de 62 toneladas.