vetu
optimizando en Internet

Sistema SaaS para la resolución del problema de asignación de flotas

Manual de usuario

Versión 2.1

infozara.es

SOBRE INFOZARA

Infozara es una empresa que se constituyó en 2006 como spin off de la Universidad de Zaragoza a través del Grupo Nóesis, Grupo Consolidado de Investigación Aplicada del Gobierno de Aragón (España) dirigido por el Profesor Eladio Domínguez.

Infozara tiene una amplia experiencia en la realización de proyectos de I+D+I y en la prestación de servicios a clientes, que cuentan con un alto nivel de valor añadido derivado de las investigaciones industriales realizadas en dichos proyectos.

Todos los productos y desarrollos se han realizado para ser explotados a través de la Web, bajo la forma de lo que actualmente se llama servicios SaaS (Software as a Service).

PROPIEDAD Y CONFIDENCIALIDAD

CONTROL DE VERSIONES

Este manual se refiere a la versión 2.1 del servicio SaaS que publica VETU, para la resolución del problema de asignación de flotas. Esta no es la primera versión del servicio. A continuación se muestra un resumen de las aportaciones de cada versión.

Versión 1.0

Versión inicial del servicio SaaS.

Versión 2.0

Productos en vacío, como soporte a la toma de decisiones de planificación de la flota, y como opción de reducción de costes operativos.

Version 2.1

Limite de tipos de vehículo. Se puede fijar la cantidad minima, exacta o máxima de vehículos de cada tipo.

Próximas versiones

Consultar el capítulo 5 de este manual.

ÍNDICE

1 Introducción

VETU es un sistema que permite la resolución de problemas complejos de asignación de recursos mediante la modalidad SaaS. Uno de los problemas cuya resolución ofrece VETU, es la asignación de flotas, que permite definir las rutas de menor coste para los vehículos de una empresa de transporte, definidos los trayectos a realizar.

Este documento es el manual de usuario del sistema SaaS de asignación de flotas de VETU. Ofrece una guía completa sobre el problema de asignación de flotas y su acceso mediante el sistema SaaS.

Es un documento dirigido tanto al gestor de recursos de la empresa, que se encarga de confeccionar las rutas de los vehículos, como a los técnicos informáticos que realizarán la integración entre los sistemas empresariales con el sistema SaaS de VETU.

1.1 En este manual

Este manual está estructurado es los siguientes capítulos:

- Introducción

  En el presente capítulo se describen de forma resumida los principales conceptos del dominio de trabajo del sistema que se presenta.

- Problema de asignación de flotas

  Se ofrece una descripción detallada del problema de optimización que resuelve el sistema descrito en el presente manual de usuario.

- Acceso a los servicios

  Se muestra la forma de acceso al sistema SaaS de asignación de flotas, mediante los servicios web que publica VETU.

- Lista de mensajes

  Muestra una lista con los mensajes que se pueden producir durante la ejecución de un problema de asignación de flotas en el sistema SaaS de VETU.

- Próxima versión

  Se describen las nuevas funcionalidades previstas para la próxima versión del sistema de asignación de flotas que se publicará en VETU.

1.2 SaaS

Los sistemas ofrecidos en forma de software como servicio (SaaS - Software as a Service), posibilitan el acceso a través de Internet a servicios de tecnologías de la información (IT – Information Technologies) de diversa naturaleza, evitando costosas adquisiciones de licencias de software o el pago de elevadas cuotas anuales.

Las ventajas de los sistemas ofrecidos en forma de SaaS son las siguientes:

- Evitan la necesidad de adquirir infraestructura; la infraestructura se comparte entre los usuarios del servicio.

- Se agiliza la entrega de actualizaciones del software evitando costosas actualizaciones; el servicio se actualiza de forma transparente a sus usuarios.
- Se paga por uso.
- Facilitan la integración con sistemas empresariales (mediante estándares -WS-).

VETU posibilita la resolución de problemas de asignación de recursos ofreciendo como SaaS el acceso a un algoritmo que obtiene la solución óptima con un coste mínimo.

El acceso al servicio SaaS de VETU, se proporciona mediante servicios web tal y como se describe más adelante en este manual de usuario.

1.3 VETU

VETU es una plataforma que resuelve diversos problemas de optimización de recursos mediante la modalidad SaaS. Uno de los problemas que publica es el de asignación de flotas, pero publica más, como la planificación táctica de personal. Para ver todos los problemas que publica VETU, consulte la web www.vetu.es.

VETU proporciona una forma homogénea para el uso de todos sus problemas de optimización, tanto desde un punto de vista técnico (acceso mediante servicios Web) como desde un punto de vista contractual con los clientes. Toda la información sobre contratación de servicios e resolución de problemas de optimización de recursos se puede consultar en la web de VETU: www.vetu.es.

La plataforma VETU es proporcionada por Infozara Consultoría Informática S. L., entidad inscrita en el Registro Mercantil de Zaragoza, tomo 3368, libro O, folio 52, hoja Z-40867, cuyo domicilio social está ubicado en C/ Marina Española n.º 12, pral. C, Edificio Azabache, Zaragoza, España; NIF: B99104721.

1.4 ¿Quiénes somos?

Infozara es una empresa que se constituyó en 2006 como spin off de la Universidad de Zaragoza a través del Grupo Nóesis, Grupo Consolidado de Investigación Aplicada del Gobierno de Aragón (España) dirigido por el Profesor Eladio Domínguez.

Infozara tiene una amplia experiencia en la realización de proyectos de I+D+I y servicios con un alto nivel de valor añadido, derivado de las investigaciones industriales realizadas en dichos proyectos.

Una característica común a todos los proyectos ha sido la construcción, en cada uno de ellos, de productos en estado precompetitivo y el desarrollo de una metodología de construcción industrial del software como parte integral de un servicio.

Desde su fundación, Infozara:

- Ha participado en diversos proyectos de Desarrollo e Investigación Industrial destacando los proyectos SPOCS (www.spocs.es), LISBB (www.lisbb.es), QRP (qrp.infozara.es), AMBÚ (ambu.infozara.es) y SMOTY (www.smoty.es) del Plan nacional de I+D+i.
- Ha desarrollado productos y componentes industriales en estado precompetitivo en el marco de los proyectos anteriores o como desarrollo posterior ante demanda del mercado.
- Ha construido y está construyendo servicios en la Cloud y en el ámbito del Internet de las Cosas (IoT).

- Tiene una cartera de clientes a los que se les está ofreciendo servicios de valor añadido.

Todos los productos y desarrollos se han realizado para ser explotados a través de la Web, bajo la forma de lo que actualmente se llama servicios SaaS (Software as a Service).

1.5 Asignación de flotas

En este manual de usuario definimos el problema de asignación de flotas (Fleet Assignment *Problem*, FAP) como el problema de planificar los trayectos que realizarán los vehículos de una empresa de transporte de viajeros, al menor coste.

Las entradas del problema son:

- Flota de vehículos

  Diferentes tipos de vehículos que pueden realizar los trayectos programados.

- Trayectos

  Trayectos que debe realizar la flota de vehículos.

El modelo de problema que se presenta en este manual es un modelo integrado, que resuelve tres problemas a la vez:

- Planificación de horarios: fija los horarios de los trayectos.
- Planificación de flotas: asigna cada trayecto a un tipo de vehículo.
- Rotación de vehículos: define la secuencia de trayectos que realiza cada vehículo.

El objetivo es encontrar la cantidad de vehículos que son necesarios, y sus rutas, de forma que se asignen todos los trayectos con un horario fijo, al menor coste posible.

Este problema se define en detalle en el capítulo 2, para ayudar a comprender el enfoque, dimensión y versatilidad del FAP que publica VETU.

Las mayores aplicaciones del FAP corresponden al transporte de viajeros (aéreo, marítimo o terrestre), pero también tiene aplicación en problemas de rutas de mercancías en ferrocarril, camión, barco o avión. Siempre que se conozcan los trayectos que se deben realizar, y los tipos de vehículo disponibles, se puede plantear y resolver un problema de asignación de flotas.

Si el problema de asignación de flotas de su empresa difiere del aquí presentado, o requiere alguna característica no recogida en el FAP de VETU, puede ponerse en contacto con Infozara por cualquiera de los medios indicados en el apartado 1.9, para plantear una solución satisfactoria para usted.

1.6 Nomenclatura

A lo largo del presente documento se hará referencia a los siguientes términos:

SaaS	Software as a Service, Software como Servicio, definido en el apartado 1.2.
FAP	Problema de asignación de flotas (*fleet assignment problem*), introducido en apartado 1.5 y descrito en mayor detalle en el capítulo 2.
UM	Unidades monetarias.

Los servicios web de VETU y sus funciones se muestran del color azul, como solve(). Los objetos de definición de los problemas a resolver y las soluciones de los problemas de

optimización se muestran en color rojo, como EntryFAP.

Las direcciones web o direcciones de correo electrónico que se referencian en este documento, se muestran en color azul, como www.vetu.es.

1.7 Soporte

Si es usted cliente de la plataforma VETU de Infozara, puede obtener soporte técnico sobre el FAP según los mecanismos que le haya especificado Infozara.

Si no es usted cliente de la plataforma VETU de Infozara, y tiene cualquier duda o sugerencia, puede ponerse en contacto con el equipo de Infozara por los mecanismos descritos en el apartado 1.9.

1.8 Actualización de versiones

Todos los problemas que ofrece la plataforma VETU, y en particular el FAP, siguen un ciclo de vida que va incorporando nuevas funciones que permiten mejorar el uso que obtienen los clientes de los problemas de optimización. El capítulo 5 muestra una descripción de las nuevas funciones que incorporará la siguiente versión del FAP de VETU.

Si es usted cliente de la plataforma VETU de Infozara, podrá consultar la política de cambio de versiones en su contrato.

1.9 Más información

Si desea más información sobre cualquier aspecto del sistema SaaS de asignación de flotas de VETU, puede ponerse en contacto con Infozara, a través de los siguientes medios:

Teléfono:	+34 976 25 43 76
Correo electrónico:	info@vetu.es
	fap@vetu.es

También puede consultar las siguientes direcciones web:

www.infozara.es

www.vetu.es

www.vetu.es/webvetu/saas.do

www.vetu.es/webvetu/asignacionFlotas.do

2 Problema de asignación de flotas

2.1 Modelo conceptual del problema

La Figura 2-1 muestra el modelo conceptual del FAP. El diagrama se lee de la siguiente forma:

> 'El problema de asignación de flotas (FAP) viene definido por una serie de tipos de vehículos y una definición de productos comerciales. El tipo de vehículo define características de coste de un vehículo y de su operativa, y se puede limitar la cantidad de vehículos de un tipo. Un producto comercial es la información del trayecto que se va a planificar. La definición de un producto viene dada por toda la información que identifica al trayecto, e indica los rangos de valores que pueden tomar ciertas características de ese trayecto. También se definen trayectos en vacío, que sirven como sugerencia de productos comerciales, y pueden reducir los costes de la operativa. La solución del problema es el conjunto de rutas que se asignan a cada tipo de vehículo, para lograr el menor coste. Cada ruta es una lista ordenada en el tiempo de trayectos, que tienen fijadas la hora de la salida y la duración del trayecto'.

En el siguiente apartado se explican con detalle cada uno de los conceptos que definen el FAP.

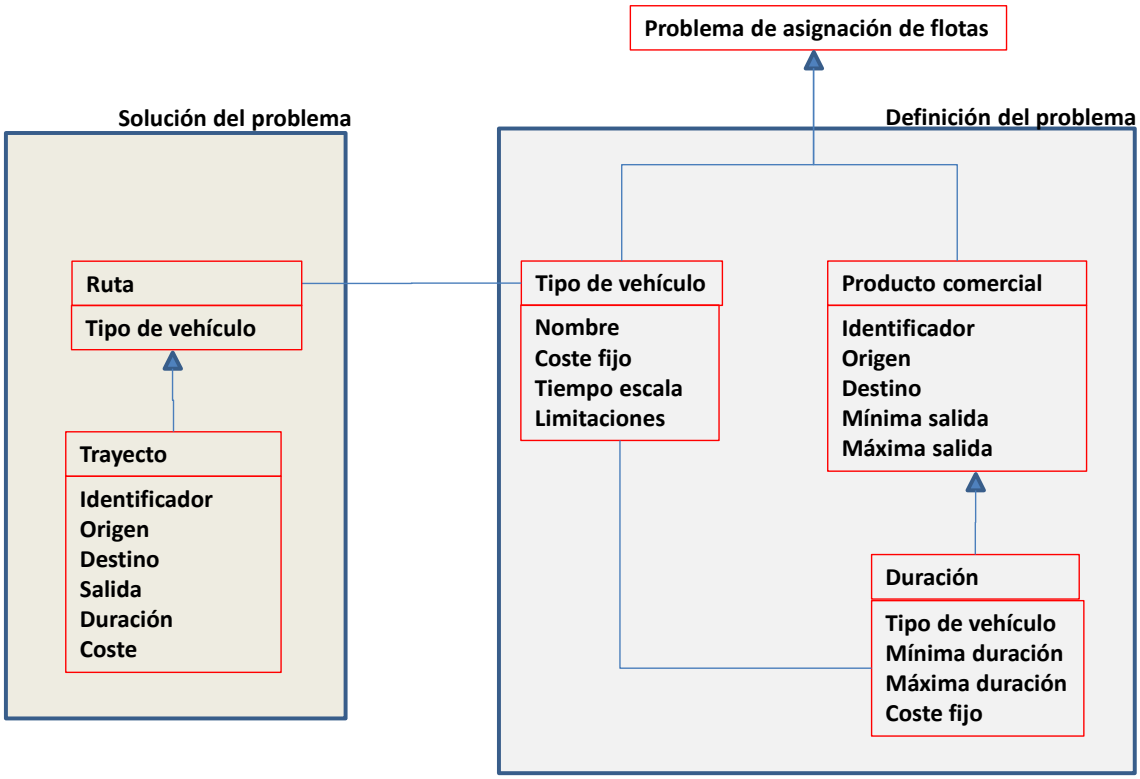

Figura 2-1. Modelo conceptual del FAP

2.2 El problema al detalle

El objetivo que busca el FAP es crear las rutas de menor coste para una flota de vehículos, de forma que se asignen todos los trayectos definidos. El término trayecto tienen dos acepciones dentro del FAP, por una parte está la definición de un trayecto, que llamamos producto comercial, que deja abiertas ciertas características para que las fije el algoritmo que resuelve el problema, y el trayecto asignado, que llamamos simplemente trayecto, que tiene fijas todas las características, y está dentro de la ruta de un tipo de vehículo.

Esas características que deja abierta el producto, son las opciones de optimización que tiene el algoritmo que resuelve el problema, para asignar todos los trayectos a alguna ruta y fijar las características de todos los trayectos, con el menor coste global de la operativa. Estas características son el tipo de vehículo, la hora de salida del trayecto y la duración del trayecto.

En la definición de un producto coercial, se puede especificar que ese trayecto lo pueden realizar varios tipos de vehículo diferentes, cada uno con un coste diferente y con una duración diferente. El algoritmo de optimización que resuelve el problema, deberá seleccionar el tipo de vehículo que realizará ese trayecto de forma que el global de las rutas tenga el menor coste. La duración de un trayecto puede venir definida como una ventana de tiempo, y el algoritmo deberá fijar un valor dentro de esa ventana de tiempo.

La otra característica que ofrece opciones de optimización es la salida de un trayecto. En la definición de un producto comercial, se puede especificar la hora de salida del trayecto como una ventana de tiempo, y el algoritmo que resuelve el problema debe fijar la hora de salida del trayecto dentro de esa ventana de tiempo, de forma que se obtengan las rutas de menor coste.

El criterio de optimización es el coste, es decir, la mejor solución de un problema de asignación de flotas es la que asigna todos los trayectos dericado de los productos comerciales, con el menor coste posible. Las fuentes de coste son dos: el coste fijo de utilizar un vehículo, y el coste de asignar cada trayecto a un tipo de vehículo.

La Tabla 2-1 muestra la definición de un tipo de vehículo con el significado de cada uno de los parámetros que lo definen, y la Tabla 2-2 muestra dos ejemplo de definición de tipos de vehículo. Si una planificación tiene dos rutas para vehículos MD83 y tres rutas para vehículos B767, el coste fijo de esa planificación es 2·33.000+3·45.000=201.000 euros. No se fija ninguna limitación sobre la cantidad de vehículos MD83, pero se fija una cantidad máxima de 20 vehículos de tipo B767. Esto significa que la planificación óptima solo podrá utilizar un máximo de 20 vehículos de tipo B767. Se pude fijar una limitación de tipo máxima, igual a o mínima, o se pueden fijar dos limitaciones a la vez de los tipos mínima y máxima. Las limitaciones pueden llevar a problema sin solución, por lo que deben fijarse con sumo cuidado. Por ejemplo, si la planificación óptima de una serie de productos comerciales necesitara de 20 vehículos y se limitan los vehículos a 19 como máximo, el problema no tendría solución.

Si un problema solo define un tipo de vehículo, diremos que es un problema de asignación de flotas homogéneo, y si un problema define varios tipos de vehículo, diremos que es un problema de asignación de flotas heterogéneo.

Parámetro	Descripción	Tipo
Nombre	Nombre del contrato.	Texto
Coste fijo	Es el coste que tiene crear una rotación para este tipo de vehículo.	Double
Tiempo escala	Es el tiempo mínimo que debe permanecer el vehículo desde que llega a una estación hasta que sale de la estación. Es el tiempo mínimo requerido ara la baja	Entero

	de pasajeros, labores de mantenimiento, y subida de los nuevos pasajeros.	
Limitaciones	Limitaciones sobre la cantidad de vehículos de un tipo que se pueden utilizar en la planificación	Lista
Limitación	Limitación sobre la cantidad de vehículos de un tipo que se pueden utilizar en la planificación	Objeto
Criterio	Establece el criterio que limita el número de vehículos de un tipo. Puede tomar los valores: • Minimum: Cantidad mínima de vehículos • Equal: Cantidad exacta de vehículos • Maximum: Cantidad máxima de vehículos	String
Cantidad	Cantidad de vehículos que se utilizan en la planificación, según el criterio definido.	Entero

Tabla 2-1. Definición de los parámetros de un tipo de vehículo

Parámetro	Vehículo 1	Vehículo 2
Nombre	MD83	B767
Coste fijo	33.000 euros	45.000 euros
Tiempo escala	60 minutos	90 minutos
Limitaciones		
Limitación		
Criterio		Maximun
Cantidad		20

Tabla 2-2. Ejemplos de tipos de vehículo

La Tabla 2-3 muestra la definición de un producto comercial, con el significado de cada uno de los parámetros que lo definen. A continuación se van mostrar diversos ejemplos de definición de productos comerciales, que ayudan a comprender el significado de cada parámetro, y la relación que existe entre productos y trayectos.

La Tabla 2-4 muestra un ejemplo de producto JG0701, entre MAD (Madrid) y BCN (Barcelona), que se debe asignar a un MD83, debe salir de MAD el lunes entre las 10:00 y las 12:00, y puede durar entre 55 minutos y una hora y cinco minutos. El producto JG0701 tiene dos opciones de optimización: la hora de salida y la duración. El proceso de optimización debe fijar el valor de esas dos opciones, y como ejemplo, puede crear el trayecto que muestra la Tabla 2-5, donde la salida se ha fijado para el lunes a las 11:00 (que es un valor que está dentro de la ventana de tiempo definida para la salida del producto) y la duración se ha fijado en una hora (que es un valor que está dentro de la ventana definida en el producto para la duración del trayecto).

Parámetro	Descripción	Tipo
Identificador	Nombre único que identifica al producto	Texto
Origen	Estación de origen del trayecto.	Texto
Destino	Estación de destino del trayecto.	Texto
Mínima salida	Mínima hora de salida del trayecto, expresada en minutos.	Entero
Máxima salida	Máxima hora de salida del trayecto, expresada en minutos.	Entero
Duración	Información del trayecto cuando se asigna a un tipo de	

	vehículo. Un producto comercial debe tener definida al menos una Duración.	
Tipo de vehículo	Tipo de vehículo al que se refieren los datos de Duración.	Texto
Mínima duración	Mínima duración que puede tener el trayecto cuanto los realiza este tipo de vehículo, expresada en minutos.	Entero
Máxima duración	Máxima duración que puede tener el trayecto cuanto los realiza este tipo de vehículo, expresada en minutos.	Entero
Coste fijo	Coste del trayecto cuando lo realiza este tipo de vehículo.	Double

Tabla 2-3. Definición de los parámetros de un producto

Parámetro	Descripción del producto
Identificador	JG0701
Origen	MAD
Destino	BCN
Mínima salida	600 minutos (Lunes a las 10:00)
Máxima salida	720 minutos (Lunes a las 12:00)
Duración	
Tipo de vehículo	MD83
Mínima duración	55 minutos
Máxima duración	65 minutos (01:05 horas)
Coste fijo	8.000 euros

Tabla 2-4. Definición de productocomercial para un tipo de vehículo

Parámetro	Descripción del producto
Identificador	JG0701
Origen	MAD
Destino	BCN
Salida	660 minutos (Lunes a las 11:00)
Duración	60 minutos (01:00 horas)
Tipo de vehículo	MD83
Coste	8.000 euros

Tabla 2-5. Posible trayecto para el producto definido en la Tabla 2-4

Parámetro	Descripción del producto
Identificador	JG0701
Origen	MAD
Destino	BCN
Salida	600 minutos (Lunes a las 10:00)
Duración	
Tipo de vehículo	MD83
Coste	8.000 euros

Tabla 2-6. Otro posible trayecto para el producto definido en la Tabla 2-4

La Tabla 2-7 muestra la definición del producto comercial JG0702, que se puede asignar a dos tipos de vehículos, un MD83, que tardaría entre 55 y 65 minutos y tendría un coste de 8.000 euros, o un MD87, que tardaría entre 60 y 70 minutos, con un coste de 9.000 euros. El algoritmo que resuelve el problema debe decidir si el trayecto JG0702 lo realiza un MD83 o un MD87, y además debe fijar la duración en función del tipo de vehículo asignado, para obtener el menor coste de asignación de todos los productos. La Tabla 2-8 muestra un posible trayecto para el producto JG0702, donde se ha asignado el tipo de vehículo MD83, con una duración de una hora, y la Tabla 2-9 muestra otro posible trayecto para el mismo producto, pero esta vez se le ha asignado un MD87, con una duración de una hora y diez minutos.

Parámetro	Descripción del producto
Identificador	JG0702
Origen	MAD
Destino	BCN
Mínima salida	600 minutos (Lunes a las 10:00)
Máxima salida	720 minutos (Lunes a las 12:00)
Duración	
Tipo de vehículo	MD83
Mínima duración	55 minutos
Máxima duración	65 minutos (01:05 horas)
Coste fijo	8.000 euros
Duración	
Tipo de vehículo	MD87
Mínima duración	60 minutos (01:00 horas)
Máxima duración	70 minutos (01:10 horas)
Coste fijo	9.000 euros

Tabla 2-7. Definición de producto con dos tipos de vehículo

Parámetro	Descripción del producto
Identificador	JG0702
Origen	MAD
Destino	BCN
Salida	630 minutos (Lunes a las 10:30)
Duración	60 minutos (01:00 horas)
Tipo de vehículo	MD83
Coste	8.000 euros

Tabla 2-8. Posible trayecto para el producto definido en la Tabla 2-7

Parámetro	Descripción del producto
Identificador	JG0702
Origen	MAD
Destino	BCN
Salida	645 minutos (Lunes a las 10:45)
Duración	70 minutos (01:10 horas)

Tipo de vehículo	MD87
Coste	9.000 euros

Tabla 2-9. Otro posible trayecto para el producto definido en la Tabla 2-7

Un producto en vacío se define de forma parecida a un producto comercial, excepto que no tiene identificador, ni define una ventana de tiempo en la salida. La Tabla 2-10 muestra la definición de un producto vacío entre MAD y BCN, que pueden realizar los tipos de vehículo MD83 o MD87.

Parámetro	**Descripción del producto**
Origen	MAD
Destino	BCN
Duración	
Tipo de vehículo	MD83
Mínima duración	55 minutos
Máxima duración	65 minutos (01:05 horas)
Coste fijo	8.000 euros
Duración	
Tipo de vehículo	MD87
Mínima duración	60 minutos (01:00 horas)
Máxima duración	70 minutos (01:10 horas)
Coste fijo	9.000 euros

Tabla 2-10. Definición de un producto vacío

El proceso de optimización que resuelve el proceso de planificación de flotas, debe asignar todos los productos comerciales, es decir, debe crear un trayecto por cada producto comercial. Además, es proceso tiene que decidir si debe crear trayectos en vacío, asociados a los productos en vacío. Solo se crearán trayectos en vacio si se consiguen operativas de transporte de menos coste. Los trayectos en vacío ayudan al planificador durante le proceso de definir los horarios de los trayectos, y son una opción de optimización.

Las opciones de optimización que presenta el modelo de problema de asignación de flotas que publica VETU, son las ventanas de tiempo en la salida, los tipos de vehículo que se pueden asignar, la duración del trayecto en función del tipo de vehículo que se asigna, y los trayectos en vacío. Pero esto no obliga a que se deba fijar siempre una ventana de tiempo en la salida y una duración variable para cada tipo de vehículo estos valores pueden ser fijos si así se requiere. Como ejemplo de este caso, la Tabla 2-11 muestra la definición de un producto comercial con la salida fijada en el lunes a las 10:00, que se debe asignar a un MD83 con una duración de una hora. En este caso el algoritmo no puede cambiar la hora de salida, ni el tipo de vehículo ni la duración, con lo que el trayecto asignado a este producto será el que muestra la Tabla 2-12. En este caso, el algoritmo que resuelve el problema de asignación de flotas deberá decidir únicamente en qué rotación coloca a este trayecto.

Parámetro	**Descripción del producto**
Identificador	JG0703
Origen	MAD
Destino	BCN
Mínima salida	600 minutos (Lunes a las 10:00)

Máxima salida	600 minutos (Lunes a las 10:00)
Duración	
Tipo de vehículo	MD83
Mínima duración	60 minutos (01:00 horas)
Máxima duración	60 minutos (01:00 horas)
Coste fijo	8.000 euros

Tabla 2-11. Producto sin ventana de tiempo en la salida y duración fija

Parámetro	Descripción del producto
Identificador	JG0703
Origen	MAD
Destino	BCN
Salida	600 minutos (Lunes a las 10:00)
Duración	60 minutos (01:00 horas)
Tipo de vehículo	MD83
Coste	8.000 euros

Tabla 2-12. Trayecto asociado al producto definido en la Tabla 2-10

2.3 Naturaleza del problema

El FAP es un problema de optimización combinatoria, lo que significa que existen muchas soluciones diferentes, es decir, existen muchas formas de construir las rutas de los vehículos y fijar los valores de las características definidas en un producto comercial. Cada una de estas formas de definir las rutas tiene un coste diferente, con lo que el problema consiste en encontrar las rutas de coste mínimo que asigne todos los trayectos.

Desde un punto de vista matemático, existe un número finito de rutas diferentes, pero es un número tan elevado que no permite generar todas las rutas para evaluarlas y quedarse con la mejor. Infozara ha desarrollado un algoritmo basado en técnicas de Investigación Operativa, que permite encontrar la solución óptima en varios minutos de cálculo intensivo. Este algoritmo es el que se encapsula detrás de los servicios Web, y está disponible para sus clientes, sin que estos tengan la necesidad de conocer técnicas matemáticas, Investigación Operativa o programación informática: el cliente tiene un problema de asignación de flotas y quiere que se resuelva de la mejor forma posible, sin preocuparse de la infraestructura que tiene implantada Infozara para resolver problemas de este tipo.

Un componente de optimización es un cierto concepto que puede tomar un rango de valores. El FAP que se presenta en este manual, presenta los siguientes conceptos de optimización:

- Hora de salida del trayecto: El producto especifica la ventana de tiempo de salida del trayecto, que es un intervalo de tiempo permitido para la salida del vehículo. El algoritmo que resuelve el problema debe fijar la hora de salida del trayecto, como un valor dentro de ese intervalo.
- Asignación de tipo de vehículo: El producto especifica varios tipos de vehículo que pueden realizar el trayecto, cada uno con un coste diferente. El algoritmo que resuelve el problema debe asignar un tipo de vehículo al trayecto, de entre los tipos de vehículo que se definen en el producto comercial.
- Rotación de un vehículo: Se crea una lista de trayectos ordenados en el tiempo, que puede realizar un mismo vehículo.

- Productos en vacío.

A continuación se va a mostrar un ejemplo de cómo las ventanas de tiempo en la salida pueden llevar a planificaciones de menor coste. Sean los productos que se definen en el plan comercial de la Tabla 2-12. Ahora se plantea un problema de asignación de flotas que calcule las rotaciones de los vehículos al menor coste. Como solo hay un tipo de vehículo, el coste de la planificación es proporcional al número de vehículos, así que el objetivo es minimizar el número de vehículos. La solución al problema es la que muestra la Tabla 2-13, donde son necesarios tres vehículos para realizar todos los trayectos. Ahora se va a cambiar el plan comercial, permitiendo un intervalo de una hora de longitud en la salida de algunos trayectos, como muestra la Tabla 2-14. Si ahora se resuelve el problema de asignación de flotas, se obtiene la solución que muestra la Tabla 2-15, donde son necesarios 2 vehículos, han cambiado las asignaciones de trayectos a los vehículos, han cambiado las rotaciones de los vehículos, y se ha fijado una hora de salida diferente a los trayectos JG000 y JG002. Todos estos cambios que llevan a una mejor planificación, se han producido porque se ha dado una cierta libertad a la hora de salida de algunos trayectos. Este sencillo ejercicio muestra que las ventanas de tiempo en la salida son una opción de optimización muy clara en los problemas de asignación de flotas.

Identificador	Vehículo	Origen	Destino	Salida	Llegada
JG000	Tipo1	CCC	BBB	10:00	13:00
JG001	Tipo1	AAA	BBB	10:00	12:00
JG002	Tipo1	BBB	CCC	12:00	14:00
JG003	Tipo1	BBB	AAA	13:00	15:00
JG004	Tipo1	AAA	CCC	16:00	20:00
JG005	Tipo1	CCC	AAA	17:00	21:00

Tabla 2-13. Plan comercial sin ventanas de tiempo en la salida

Vehículo	Trayectos		
Vehículo 1	JG001 AAA 10:00 – BBB 12:00	JG003 BBB 13:00 – AAA 15:00	JG004 AAA 16:00 – 20:00
Vehículo 2	JG000 CCC 10:00 – BBB 13:00		
Vehículo 3	JG002 BBB 12:00 – CCC 14:00	JG005 CCC 17:00 – AAA 21:00	

Tabla 2-14. Planificación para el plan comercial sin ventanas de tiempo en la salida

Identificador	Vehículo	Origen	Destino	Salida	Duración
JG000	Tipo1	CCC	BBB	Entre 09:00 y 10:00	03:00
JG001	Tipo1	AAA	BBB	10:00	02:00
JG002	Tipo1	BBB	CCC	Entre 12:00 y 13:00	02:00
JG003	Tipo1	BBB	AAA	13:00	02:00
JG004	Tipo1	AAA	CCC	16:00	04:00
JG005	Tipo1	CCC	AAA	17:00	04:00

Tabla 2-15. Plan comercial con ventanas de tiempo en la salida

Vehículo	Trayectos		
Vehículo 1	JG001 AAA 10:00 – BBB 12:00	JG002 BBB 13:00 – CCC 16:00	JG005 CCC 17:00 – AAA 21:00
Vehículo 2	JG000 CCC 09:00 – BBB 12:00	JG003 BBB 13:00 – AAA 15:00	JG004 AAA 16:00 – CCC 20:00

Tabla 2-16. Planificación para el plan comercial con ventanas de tiempo en la salida

3 Acceso a los servicios de VETU

En este capítulo se indica cómo acceder a los servicios Web que publica el sistema SaaS de VETU, para resolver el problema de asignación de flotas.

3.1 Arquitectura general de todo el sistema

La arquitectura del sistema SaaS de VETU, que publica Infozara para la resolución de problemas de optimización, es la que muestra la Figura 3-1. En un determinado momento de ejecución de los procesos empresariales del cliente, es necesario dimensionar una plantilla de personal para dar un servicio. En ese momento, se lanza un proceso de llamada que recoge la información de los sistemas empresariales del cliente, y llama a los servicios web del sistema SaaS de VETU. El problema se resuelve en la infraestructura de cálculo de Infozara y la planificación obtenida la recoge el proceso de llamada, que la almacena en los sistemas empresariales del cliente, poniéndolos a disposición del usuario.

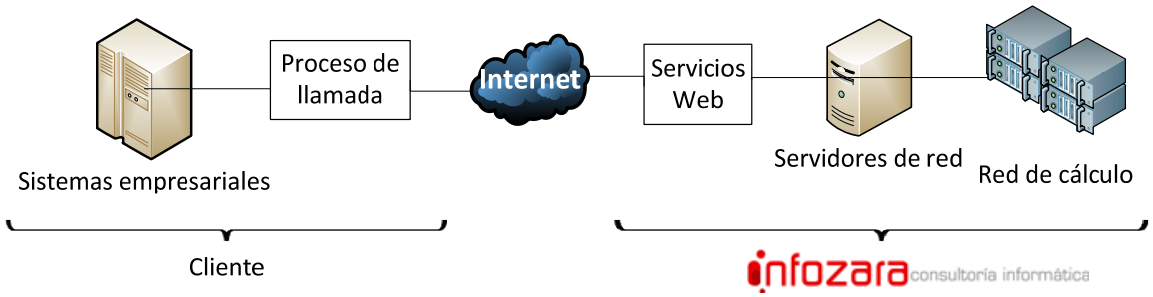

Figura 3-1. Arquitectura del sistema SaaS

Cabe destacar que la infraestructura de red y de cálculo de Infozara no es conocida por el cliente, que lo que quiere es que se resuelva su problema de planificación táctica de personal, sin necesidad de conocer los procesos y máquinas de Infozara.

3.2 Servicios Web

El acceso a la resolución de problemas de optimización mediante SaaS, se hace mediante dos servicios Web: un servicio Web para enviar los datos del problema y obtener la solución, y otro servicio Web para comprobar el estado de ejecución de la resolución del problema.

El servicio Web de comprobación del estado de ejecución es necesario, porque el FAP no se resuelve de forma instantánea, sino que puedes requerir de varios minutos, dado que es un problema complejo de optimización combinatoria (véase apartado 2.3). En función del tipo de problema y de su configuración, el tiempo de resolución puede ir desde varios segundos, hasta varios minutos.

El flujo general de uso del sistema SaaS de VETU es el siguiente:

1. Se envía el problema
2. Se comprueba reiteradamente el estado de resolución del problema hasta que esté terminado

3. Se accede a la solución

El servicio Web de resolución de problemas devuelve una clase Request, con un ticket cada vez que se envía un problema. Este ticket se utiliza para acceder al servicio Web de comprobación del estado de la resolución del problema. Si se producen errores, el ticket tendrá el valor -1, y se mostrarán los errores detectados.

Dado que los servicios Web están basados en estándares, es posible acceder a ellos mediante cualquier plataforma de desarrollo. En este manual se presenta el acceso a los servicios Web desde la Plataforma .NET, NetBeans, y Eclipse.

El servicio web de comprobación de estado se llama wsProgress y expone la función checkStatus(). El servicio web de resolución del FAP se llama wsFAP y expone las funciones solve() y getSolution().

En los siguientes apartados se describe cada uno de los servicios web.

3.2.1 Servicio web de comprobación de estado de problemas

El servicio web que permite obtener el estado de la resolución de una instancia de problema enviada a VETU se llama wsProgress, y tiene la función:

String getStatus(String username, String password, String nTicket)

Este servicio web es independiente del tipo de problema a resolver, y expone una función para obtener el estado de la ejecución de un problema a partir de su ticket de identificación. Esta función devuelve uno de los estados que muestra la Tabla 3-1.

Estado	Significado
UNKNOWN	El ticket no corresponde a una solicitud de resolución de ningún problema registrado en el sistema.
ALLOCATED	El problema se ha recibido correctamente, y está listo para su ejecución.
IN_PROCESS	El problema se está resolviendo en este instante.
PROCESSED	El problema ya está resuelto, y se puede acceder a la solución
ERROR	Se ha producido un error en la llamada a la función. Contactar con el administrador.

Tabla 3-1. Estados de resolución de un problema

Para obtener la definición del servicio web, puede escribir la siguiente dirección en cualquier navegador:

http://www.vetu.es/wsProgress/wsProgress?wsdl

3.2.2 Servicio web de resolución del problema

El servicio web que permite la resolución del FAP se llama wsFAP, y tiene dos funciones

Request solve(String username, Strign password, Fap oProblem)

FapSolution getSolution(String nTicket)

Para obtener la definición del servicio web, puede escribir la siguiente dirección en cualquier navegador:

http://www.vetu.es/wsFAP/wsFAP?wsdl

La función solve() recibe la clase Fap, que encapsula la instancia del FAP a resolver, y devuelve un objeto de la clase Request, que indica si la instancia enviada es correcta. La Figura 3-2 muestra la versión XML de una clase Request que corresponde a un problema válido: el campo status vale Ok, hay un ticket válido, y el mensaje es 'The problem has been successfully solved'. La Figura 3-3 muestra la versión XML de una clase Request que corresponde a un problema con errores: el campo status tiene el valor Failed, el ticket vale -1, y hay un mensaje de error. Se puede ver la lista de mensajes que puede contener un resultado Request en el apartado 4.1.

```
<?xml version="1.0" encoding="utf-8"?>
<Request>
   <Status>Ok</Status>
   <Ticket>457223874</Ticket>
   <Message Code="SRV0">The problem has been received</Message>
</Request>
```

Figura 3-2. Resultado de un problema sin errores

```
<?xml version="1.0" encoding="utf-8"?>
<Request>
   <Status>Failed</Status>
   <Ticket>-1</Ticket>
   <Message Code="SRV4">The contract has expired</Message>
</Request>
```

Figura 3-3. Resultado de un problema con errores

El objeto Fap encapsula la información del problema que se quiere resolver. La Figura 3-4 muestra la versión XML de la definición de un FAP. El elemento raíz FAP_Problem se compone de tres objetos: Fleet, que define los tipos de vehículo disponibles, y Products, que es una lista de objetos Product, que definen un producto comercial, y EmptyProducts, que es una lista de objetos EmptyProduct, que definen un producto vacío.

```
<?xml version="1.0"?>
<FAP_Problem>
  <Fleet>
    <VehicleType>
      <Name>Small</Name>
      <FixedCost>6000</FixedCost>
      <TimeSpam>45</TimeSpam>
      <Limitations/>
    </VehicleType>
    <VehicleType>
      <Name>Large</Name>
      <FixedCost>10000</FixedCost>
      <TimeSpam>60</TimeSpam>
      <Limitations>
        <Limitation>
          <Criterion>Minimum</Criterion>
          <Amount>10</Amount>
        </Limitation>
        <Limitation>
          <Criterion>Maximum</Criterion>
          <Amount>20</Amount>
        </Limitation>
      </Limitations>
    </VehicleType>
  </Fleet>
```

```
<Products>
  <Product>
    <Identifier>JG001</Identifier>
    <From>MAD</From>
    <To>BCN</To>
    <MinimunDeparture>100</MinimunDeparture>
    <MaximunDeparture>200</MaximunDeparture>
    <Durations>
      <Duration>
        <Vehicle>Small</Vehicle>
        <MinimunDuration>50</MinimunDuration>
        <MaximunDuration>50</MaximunDuration>
        <FixedCost>100</FixedCost>
      </Duration>
      <Duration>
        <Vehicle>Large</Vehicle>
        <MinimunDuration>60</MinimunDuration>
        <MaximunDuration>60</MaximunDuration>
        <FixedCost>120</FixedCost>
      </Duration>
    </Durations>
  </Product>
  <Product>
    <Identifier>JG002</Identifier>
    <From>MAD</From>
    <To>BCN</To>
    <MinimunDeparture>120</MinimunDeparture>
    <MaximunDeparture>120</MaximunDeparture>
    <Durations>
      <Duration>
        <Vehicle>Small</Vehicle>
        <MinimunDuration>50</MinimunDuration>
        <MaximunDuration>60</MaximunDuration>
        <FixedCost>100</FixedCost>
      </Duration>
    </Durations>
  </Product>
  <Product>
    <Identifier>JG003</Identifier>
    <From>BCN</From>
    <To>BIO</To>
    <MinimunDeparture>500</MinimunDeparture>
    <MaximunDeparture>500</MaximunDeparture>
    <Durations>
      <Duration>
        <Vehicle>Small</Vehicle>
        <MinimunDuration>100</MinimunDuration>
        <MaximunDuration>100</MaximunDuration>
        <FixedCost>100</FixedCost>
      </Duration>
    </Durations>
  </Product>
  <Product>
    <Identifier>JG004</Identifier>
    <From>BCN</From>
    <To>BIO</To>
    <MinimunDeparture>400</MinimunDeparture>
    <MaximunDeparture>450</MaximunDeparture>
    <Durations>
      <Duration>
```

```
          <Vehicle>Small</Vehicle>
          <MinimunDuration>100</MinimunDuration>
          <MaximunDuration>120</MaximunDuration>
          <FixedCost>100</FixedCost>
        </Duration>
        <Duration>
          <Vehicle>Large</Vehicle>
          <MinimunDuration>110</MinimunDuration>
          <MaximunDuration>130</MaximunDuration>
          <FixedCost>140</FixedCost>
        </Duration>
      </Durations>
    </Product>
  </Products>
  <EmptyProducts>
    <EmptyProduct>
      <From>MAD</From>
      <To>BIO</To>
      <EmptyAssignments>
        <EmptyAssignment>
          <Vehicle>Small</Vehicle>
          <MinimunDuration>45</MinimunDuration>
          <MaximunDuration>45</MaximunDuration>
          <FixedCost>85</FixedCost>
        </EmptyAssignment>
        <EmptyAssignment>
          <Vehicle>Large</Vehicle>
          <MinimunDuration>45</MinimunDuration>
          <MaximunDuration>45</MaximunDuration>
          <FixedCost>90</FixedCost>
        </EmptyAssignment>
      </EmptyAssignments>
    </EmptyProduct>
  </EmptyProducts>
</FAP_Problem>
```

Figura 3-4. Ejemplo de instancia XML de un FAP

El objeto Fleet contiene la definición de la flota de vehículos disponible, y es una lista de objetos VehicleType. Cada uno de estos objetos VehicleType indica un tipo de vehículo, que viene definido por estos campos:

- Name: Nombre del tipo de vehículo.
- FixedCost: Coste fijo en que incurre por el uso de un vehículo de este tipo.
- TimeSpam: Tiempo mínimo de escala del tipo de vehículo en una estación.
- Limitations: Limitaciones en la cantidad de vehículos.

Los tipos de vehículo del ejemplo de la Figura 3-4 son los del tipo 'Small' y 'Large', que tienen un coste fijo de 6.000 UM y 1.000 UM, respectivamente, y unos tiempos mínimos de escala de 45 y 60 minutos respectivamente. No se define ninguna limitación sobre el tipo 'Small', y establece un límite entre 10 vehículos del tipo 'Large'.

El objeto Products contiene los productos comerciales, que se deben asignar a un tipo de vehículo. El objeto Products es una lista de objetos Product, que contiene la definición de un producto comercial. Un producto comercial viene definido por los siguientes campos:

- Identifier: Identificador del producto comercial. Es un texto que identifica de forma única a cada producto.
- From: Estación de la que parte el trayecto asociado al producto comercial. Es un texto que identifica la estación de salida.
- To: Estación de llegada del trayecto. Es un texto que identifica la estación de

llegada.

- Ventana de tiempo en la salida: Define el intervalo de comienzo permitido para el trayecto. Viene definido por los objetos MinimunDeparture, que indica el mínimo comienzo permitido para el trayecto, y MaximunDeparture, que indica el máximo comienzo permitido para el trayecto. Cuando se resuelva el problema, se habrá fijado la hora de salida del trayecto entre esos dos valores. Los dos campos se expresan en minutos. Si el trayecto tiene definida una hora de salida fija, el valor de ambos campos será el mismo. La Tabla 3-2 muestra un ejemplo de producto definido con ventana de tiempo en la salida (la salida del trayecto puede estar entre los minutos 100 y 200), y la Tabla 3-3 muestra un ejemplo de trayecto con la hora de salida fija (el minuto 500).
- Duraciones: Duración del trayecto en función del tipo de vehículo que se asigne a este producto. Se pueden definir varias duraciones, para diferentes tipos de trayecto. Si varios tipos de vehículo pueden realizar este trayecto, se deberá especificar un objeto Duration para cada uno de ellos. El campo Durations es una lista de objetos Duration.

```
<Product>
   <Identifier>JG001</Identifier>
   <From>MAD</From>
   <To>BCN</To>
   <MinimunDeparture>100</MinimunDeparture>
   <MaximunDeparture>200</MaximunDeparture>
```

Tabla 3-2. Producto definido con ventana de tiempo en la salida

```
<Product>
  <Identifier>JG003</Identifier>
  <From>BCN</From>
  <To>BIO</To>
  <MinimunDeparture>500</MinimunDeparture>
  <MaximunDeparture>500</MaximunDeparture>
```

Tabla 3-3. Producto definido con hora fija de salida

Un producto comercial deb defienir al menos por un objeto Duration. El objeto Duration especifica la duración que puede tener el trayecto si se le asigna un tipo de vehículo, y el coste que tendrá el trayecto si se asigna ese tipo de vehículo. La Tabla 3-4 muestra un ejemplo de trayecto que solo puede realizar un tipo de vehículo (el tipo de vehículo 'Small'), y la Tabla 3-5 muestra un ejemplo de trayecto que se puede asignar a varios tipos de vehículo (los tipos 'Small' y 'Large'). El algoritmo que resuelve el problema deberá seleccionar uno de los dos tipos de vehículo, de forma que se minimicen los costes globales de toda la operativa.

```
<Product>
   <Identifier>JG002</Identifier>
   <From>MAD</From>
   <To>BCN</To>
   <MinimunDeparture>120</MinimunDeparture>
   <MaximunDeparture>120</MaximunDeparture>
   <Durations>
     <Duration>
       <Vehicle>Small</Vehicle>
```

```
        <MinimunDuration>50</MinimunDuration>
        <MaximunDuration>60</MaximunDuration>
        <FixedCost>100</FixedCost>
      </Duration>
    </Durations>
  </Product>
```

Tabla 3-4. Trayecto definido para un solo tipo de vehículo

```
  <Product>
    <Identifier>JG001</Identifier>
    <From>MAD</From>
    <To>BCN</To>
    <MinimunDeparture>100</MinimunDeparture>
    <MaximunDeparture>200</MaximunDeparture>
    <Durations>
      <Duration>
        <Vehicle>Small</Vehicle>
        <MinimunDuration>50</MinimunDuration>
        <MaximunDuration>50</MaximunDuration>
        <FixedCost>100</FixedCost>
      </Duration>
      <Duration>
        <Vehicle>Large</Vehicle>
        <MinimunDuration>60</MinimunDuration>
        <MaximunDuration>60</MaximunDuration>
        <FixedCost>120</FixedCost>
      </Duration>
    </Durations>
  </Product>
```

Tabla 3-5. Trayecto definido para varios tipos de vehículo

La posible asignación de un tipo de vehículo a un trayecto se hace mediante el campo Duration, que muestra los siguientes campos:

- Vehicle: Nombre del tipo de vehículo que puede realizar el trayecto. Es un texto. Si el nombre del tipo de vehículo no ha sido definido previamente en el objeto Fleet, se lanza el error tipo FAP10 (ver capítulo 4.2).
- Ventana de tiempo en la duración: La duración del trayecto si se asigna este tipo de vehículo se define como un intervalo, y se especifica con los campos MinimunDuration, que es la mínima duración que puede tener el trayecto si se asigna a este tipo de vehículo, y MaximunDuration, que es la máxima duración que puede tener el trayecto si se asigna a este tipo de vehículo. El algoritmo que resuelve el problema, deberá fijar la duración del trayecto dentro del intervalo definido por los valores de MinimunDuration y MaximunDuration. La Tabla 3-5 muestra dos ejemplos de duración fija para los tipos de vehículo 'Small' y 'Large', y la Tabla 3-4 muestra un ejemplo de duración variable, entre 50 y 60 minutos.
- FixedCost: Coste fijo en que se incurre si se asigna este tipo de vehículo a este trayecto.

Los productos en vacío se representan en el objeto EmptyProducts. Un producto vacío se representa por el objeto EmptyProduct, que se define por la estación de origen (objeto From), la estación de destino (objeto To) y objetos de asignación de un tipo de vehículo (objeto EmptyAssignments). Los objetos de asignación vienen definidos por el objeto EmptyAssignment, que especifica el tipo de vehículo (objeto Vehicle) que puede realizar el trayecto en vacío, las duraciones mínima y máxima (objetos MinimumDuration y MaximunDuration) que puede tener el trayecto en vacío, y el coste de realizar el trayecto en

vacío (objeto FixedCost). Como ejemplo de producto vacío, al final de la Figura 3-4 se define un producto en vacío que puede ir desde MAD a BIO, y se puede asignar a tipos de vehículo 'Small' con un coste de 85, o a un tipo de vehículo 'Large' con un coste de 90. En ambos casos, la duración del trayecto sería de 45 minutos.

A continuación se va a mostrar el formato de la salida del servicio web que resuelve el FAP en VETU. La Tabla 3-6 muestra un ejemplo en XML, de la solución del FAP que genera el proceso de resolución. El elemento raíz que encapsula toda la información de la solución es FAP_Solution, que se compone de tres objetos: Result, que indica que el problema se ha resuelto correctamente, ObjetiveValue, que muestra los valores objetivo de la solución, y Routes, que es el conjunto de rutas de vehículos que ha generado la solución.

```
<?xml version="1.0" encoding="utf-8"?>
<FAP_Solution>
  <Result>
    <Status>Ok</Status>
    <Message Code ="FAP0">The FAP problem has been successfully solved</Message>
  </Result>
  <ObjectiveValue>
    <Cost>2</Cost>
    <VehicleUses>
      <VehicleUse
        <Name>Small</Name>
        <Amount>2</Amount>
      </VehicleUse>
      <VehicleUse>
        <Name>Large</Name>
        <Amount>1</Amount>
      </VehicleUse>
    </VehicleUses>
  </ObjectiveValue>
  <Rotations>
    <Rotation
      <AllocatedVehicle>Small</AllocatedVehicle>
      <Legs>
        <Leg>
          <Identifier>JG001</Identifier>
          <From>MAD</From>
          <To>BCN</To>
          <Departure>100</Departure>
          <Duration>50</Duration>
        </Leg>
        <Leg>
          <Identifier>JG003</Identifier>
          <From>BCN</From>
          <To>BIO</To>
          <Departure>500</Departure>
          <Duration>100</Duration>
        </Leg>
      </Legs>
    </Rotation>
    <Rotation>
      <AllocatedVehicle>Small</AllocatedVehicle>
      <Legs>
        <Leg>
          <Identifier>JG002</Identifier>
          <From>MAD</From>
          <To>BCN</To>
          <Departure>120</Departure>
```

```
          <Duration>50</Duration>
        </Leg>
        <Leg>
          <Identifier>JG004</Identifier>
          <From>BCN</From>
          <To>BIO</To>
          <Departure>400</Departure>
          <Duration>100</Duration>
        </Leg>
      </Legs>
    </Rotation>
    <Rotation>
      <AllocatedVehicle>Large</AllocatedVehicle>
      <Legs>
        <Leg>
          <Identifier>JG005</Identifier>
          <From>MAD</From>
          <To>BCN</To>
          <Departure>220</Departure>
          <Duration>50</Duration>
        </Leg>
        <Leg>
          <Identifier>JG006</Identifier>
          <From>BCN</From>
          <To>BIO</To>
          <Departure>450</Departure>
          <Duration>100</Duration>
        </Leg>
      </Legs>
    </Rotation>
  </Rotations>
</FAP_Solution>
```

Tabla 3-6. Ejemplo de solución del FAP

El objeto Result indica que el problema se ha resuelto correctamente, cuando el campo Status tiene el valor Ok y el mensaje es el de código FAP0. Si se ha producido algún error, el campo Status mostrará el valor Failed, y se mostrarán una serie de mensajes con los errores encontrados. La Tabla 3-7 muestra un ejemplo de solución con errores, donde el campo Status tiene el valor Failed, hay dos mensajes de error, y los objetos ObjetiveValue y Routes están vacíos.

```
<?xml version="1.0" encoding="utf-8"?>
< FAP_Solution>
  <Result>
    <Status>Failed</Status>
    <Message Code ="FAP10">Leg JG008 allocates undefined vehicle type
                          MF85</Message>
    <Message Code ="FAP11">Timewindow [90,60] in the departure of leg JG008 is
                          not valid.t</Message>
  </Result>
  <ObjectiveValue/>
  <Routes/>
</ FAP_Solution>
```

Tabla 3-7. Ejemplo de solución del FAP con errores

El objeto ObjectiveValue muestra los valores de las magnitudes de la solución que ha encontrado el proceso de cálculo. Este objeto se compone de los objetos Cost, que indica el coste total de la planificación calculada, y del objeto VehicleUses, que indica la cantidad de vehículos de cada tipo que son necesarios en la planificación calculada. El objeto VehicleUses es una lista de objetos VehicleUse. Este objeto VehicleType contiene dos campos:

- Name: Nombre del tipo de vehículo
- Amount: Cantidad de vehículos necesarios de este tipo para la planificación calculada.

En el ejemplo de fichero XML de salida de la Tabla 3-7, se utilizan 2 vehículo de tipo 'Small' y 1 vehículo de tipo 'Large', en la planificación que muestra el objeto Routes.

El objeto Rotations muestra la planificación de coste mínimo que ha calculado el algoritmo que ejecuta en la plataforma VETU, como la secuencia de trayectos que se asignan a cada tipo de vehículo. El objeto Rotations es una lista de objetos Rotation, que contiene la ruta asignada a un tipo de vehículo. El objeto Rotation se compone de dos objetos, el objeto AllocatedVehicle, que especifica el nombre del tipo de vehículo que hace la ruta calculada, y el objeto Legs, que expresa la ruta de un vehículo de este tipo. La ruta que contiene el objeto Legs es una lista de objetos Leg, que describe un trayecto. Un objeto Leg viene definido por los siguientes campos:

- Identifier: Identificador del trayecto
- From: Estación de salida del trayecto
- To: Estación de llegada del trayecto
- Departure: Hora de salida del trayecto
- Duration: Duración del trayecto

La hora de salida del trayecto es un valor que está dentro de la ventana de tiempo definida para la salida que se dio en la definición del trayecto (objetos MinimunDeparture y MaximunDeparture del fichero de entrada), y la duración del trayecto es un valor que está dentro de la ventana de tiempo definida para la duración del trayecto para el tipo de vehículo (objetos MinimunDuration y MaximunDuration del fichero de entrada)

3.3 Proceso de ejecución de un FAP mediante los servicios web

A la vista de los diversos estados que devuelve el servicio web de estado de resolución, y de los objetos que maneja el servicio web de resolución del FAP, el pseudocódigo de un programa que resuelve el FAP mediante llamada a los servicios web de VETU, es el que se muestra a continuación. En los ejemplos de uso bajo diversas plataformas de desarrollo que vienen a continuación, se mostrará una implementación detallada de este pseudocódigo.

```
FAP_Problem oEntrada
Definir los de vehículo en oEntrada
Definir trayectos en oEntrada
Request oResultado = wsFAP.solve(oEntrada)
If oResultado.Status = Failed
        Tratar errores (por ejemplo, mostrar errores al usuario)
Else If oResultado = Ok
        Bool bTerminado = false
        String strEstado
        While bSalir = false
                strEstado = wsProgress.checkStatus( oResultado.Ticket )
                If( strEstado = UNKNOWN OR strEstado = PROCESSED OR strEstado = ERROR)
                        bTerminado = true
                Else If(strEstado = ALLOCATED OR strEstado = IN_PROCESS)
                        Fijar timer de 1 minuto
                Else
```

```
                bTerminado = true;
            End If
        End While
        Switch strEstado
            Caso strEstado = PROCESSED
                FAP_Solution oSalida = wsFAP.getSolution( oResultado.Ticket )
                If oSalida.Result.Status = Ok
                    Almacenar oSalida en sistemas empresariales
                Else If oSalida.Result.Status = Failed
                    Mostar errores al usuario
                End If
            Caso strEstado = UNKNOWN
                Mostar error al usuario (el ticket no es válido)
            Caso strEstado = ERROR
                TratarErrores (por ejemplo, mostrarlos al usuario)
            Default
                Tratar error desconocido
        End
End If
```

Figura 3-5. Pseudocódigo de ejecución del FAP mediante los servicios web.

3.4 Ejemplo de uso desde .NET

En este apartado, se muestra un programa de ejemplo para resolver una instancia de FAP, mediante el sistema SaaS de VETU, escrito en C# (Microsoft .NET [1]) de Microsoft, bajo Visual Studio 2010[2].

Visual Studio crea todas las clases necesarias para acceder a servicios web, a partir de la definición WSDL del servicio. Para crear las clases de acceso al servicio web de resolución del FAP de VETU, se deben seguir los siguientes pasos[3]:

1. Pulsar con el botón derecho sobre el icono de la solución de Visual Studio
2. Pulsar sobre la opción de menú 'Add Service Reference …'
3. Pulsar el botón 'Advanced…'
4. Pulsar el botón 'Add Web Reference'
5. Escribir http://www.vetu.es/wsFAP/wsFAP?wsdl en la caja de edición URL
6. Pulsar el botón que está a la derecha de la caja de edición URL
7. Hacer click sobre la línea http://www.vetu.es/wsFAP/wsFAP?wsdl
8. Pulsar el botón 'Add Reference'

En este momento se han creado una serie de clases que permiten acceder al servicio web wsFAP, con acceso a las funciones solve y getSolution. En este momento se dispone de las clases necesarias para acceder al servicio web. Las clases principales que se han creado son las siguientes:

[1] Microsoft .NET es una marca registrada por Microsoft Corporation.
[2] Visual Studio .NET es una marca registrada por Microsoft Corporation.
[3] Esta secuencia de acciones podría cambiar según la versión de Visual Studio.

- wsFAPService: Es la clase de acceso al servicio web. Tiene las funciones solve y getSolution.
- fap Es la clase que encapsula la definición de la instancia de problema que se quiere resolver. Contiene a las clases vehicleType y product, que contienen la definición de los tipos de vehículo y de los productos, que permite definir la flota de vehículos y el plan comercial. La clase product contiene objetos assignment para definir los tipos de vehículo que pueden realizar un trayecto.
- fapSolution: Es la clase que encapsula la solución del problema que se ha resuelto. Contiene a las clases result, objectiveValue y rotations, que tiene la información de la resolución del problema, de los valores encontrados y las rotaciones definidas, respectivamente. Cada una de estas clases contiene a otras clases, según el formato definido para la solución en el apartado 3.2.2 de este capítulo.

Para crear las clases de acceso al servicio wsProgressService, se deben repetir los pasos anteriores, escribiendo en el punto 4.- la URL del servicio web de acceso al estado de resolución de problemas: http://www.vetu.es/wsProgress/wsProgress?wsdl. Una vez hecho, se habrá creado la clase wsProgressService, que es el acceso al servicio web, y tiene la función checkStatus.

El código fuente necesario para la resolución del FAP mediante los servicios web de VETU se muestra en la Figura 3-6. Este código fuente no sigue las mejores prácticas de programación, porque su objetivo es la simplicidad, para mostrar el acceso al servicio SaaS de VETU. Los números de línea de la columna de la izquierda se muestran solo para facilitar la siguiente explicación. Desde las líneas 1 a la 99 se definen los datos del problema a resolver, y desde la línea 102 hasta el final, es el acceso a los servicios web.

En la línea 2 se crea la instancia de la clase fap, que encapsula toda la información del problema, que son la flota de vehículos y los productos a planificar.

En la línea 5 se crea la flota de vehículos, que es un array de 2 vehículos. Antes de fijar los datos de cada tipo de vehículo, es necesario crear el objeto vehicleType (líneas 8 y 13). Una vez creado este objeto, se deben fijar todos los datos del tipo de vehículo. Desde la línea 8 a la 11 se crea un tipo de vehículo de nombre "Tipo1", con un tiempo mínimo de escala de 60 minutos y un coste fijo de 21.000 um. Desde la línea 13 a la 16 se crea un tipo de vehículo de nombre "Tipo2", con un tiempo mínimo de escala de 90 minutos y un coste fijo de 36.000 um.

En la línea 19 se crea la lista de 4 productos, de tipo product. Antes de fijar los datos de un producto, es necesario crearlo (en la línea 22 se crea el primer producto, y en las líneas 45, 62 y 78 se crean los otros tres productos). Una vez creado el producto, se especifica el valor de sus campos. Por una parte están los datos generales, por ejemplo desde la línea 23 a la 27 para el primer producto, donde se indica que se trata del trayecto JG000, que va de MAD a BCN y sale en el minuto 540 (09:00 horas). Los datos generales del segundo producto se definen entre las líneas 46 y 50, que indican que el trayecto JG001 va desde BCN a BIO, y puede salir entre el minuto 600 (10:00 horas) y el minuto 720 (12:00 horas). De forma similar, entre las líneas 63 y 67 se define los datos generales del tercer producto, y entre las líneas 80 y 84 se definen los datos generales de cuarto producto.

La definición de un producto requiere al menos un objeto assignment, que indica la información del trayecto que depende del tipo de vehículo. Antes de crear estos objetos assignment, se debe crear el objeto Assignment, que es un array de objetos assignment. Por ejemplo, en la línea 30 se crea un arrray para dos objetos assignment en el primer producto. De la misma forma, en la línea 53 se crea un array con dos objetos assignment para el segundo producto, en la línea 70 se crea un array con un objeto assignment para el tercer

producto, y en la línea 87 se crea un array con dos objetos assignment para el cuarto producto. Una vez creado el array de objeto assignment, se debe crear cada uno de los objetos assignment. Como ejemplo, en las líneas 32 a 36 se crea el primer objeto assignment del primer producto, y en las líneas 38 a 42 se crea el segundo objeto assignment del primer producto. Una vez creado cada objeto assigment, se indican el valor de sus campos. Como ejemplo, entre las líneas 32 y 36 se indica que el trayecto JG0000 lo puede realizar un vehículo de tipo Tipo1, con un coste de 4.000 um y una duración de trayecto entre 55 y 65 minutos, o entre las líneas 56 y 59 se indica que el trayecto JG0001 lo puede realizar un vehículo de tipo Tipo2 con un coste de 4.500 um y una duración fija de 60 minutos.

Una vez definido el problema, se debe enviar al servicio web: en la línea 102 se crea el acceso al servicio web, y en las líneas 105 y 106 se envía el problema al servicio web, con el usuario y la password proporcionada al contratar el servicio FAP en VETU. La llamada a la función solve devuelve un objeto request, que indica si la llamada al servicio web ha tenido algún problema. En la línea 109 se comprueba si ha existido algún error. Si no ha habido errores, se entra en la parte de búsqueda de la solución. En las líneas 116 y 117 se crea el acceso al servicio web que informa del estado de resolución del problema. Desde la línea 121 a la 134 se define un bucle con un timer que espera a que se resuelva el problema. Desde las líneas 136 a la 158, se comprueba el estado final de resolución. Si el estado es PROCESSED, se accede a la solución del problema (líneas 139 y 140), que viene encapsulada en el objeto fapSolution.

```
1   // Crea la instancia del problema
2   wsFAPReference.fap oInstancia = new wsFAPReference.fap();
3
4   // Crea la flota de vehículos
5   oInstancia.Fleet = new wsFAPReference.vehicleType[2];
6
7   // Crea los objetos Vehicle
8   oInstancia.Fleet[0] = new wsFAPReference.vehicleType();
9   oInstancia.Fleet[0].Name = "Tipo1";
10  oInstancia.Fleet[0].TimeSpam = 60;
11  oInstancia.Fleet[0].FixedCost = 21000;
12
13  oInstancia.Fleet[1] = new wsFAPReference.vehicleType();
14  oInstancia.Fleet[1].Name = "Tipo2";
15  oInstancia.Fleet[1].TimeSpam = 90;
16  oInstancia.Fleet[1].FixedCost = 36000;
17
18  // Crea los productos
19  oInstancia.Products = new wsFAPReference.product[4];
20
21  // Crea el primer producto
22  oInstancia.Products[0] = new wsFAPReference.product();
23  oInstancia.Products[0].Identifier = "JG0000";
24  oInstancia.Products[0].From = "MAD";
25  oInstancia.Products[0].To = "BCN";
26  oInstancia.Products[0].MinimunDeparture = 540;
27  oInstancia.Products[0].MaximunDeparture = 540;
28
29  // Crea el objeto Assignments del producto, y le añade dos objetos Assigment
30  oInstancia.Products[0].Assignments = new wsFAPReference.assignment[2];
31
32  oInstancia.Products[0].Assignments[0] = new wsFAPReference.assignment();
33  oInstancia.Products[0].Assignments[0].Vehicle = "Tipo1";
34  oInstancia.Products[0].Assignments[0].MinimunDuration = 55;
35  oInstancia.Products[0].Assignments[0].MaximunDuration = 65;
```

```
oInstancia.Products[0].Assignments[0].FixedCost = 4000;

oInstancia.Products[0].Assignments[1] = new wsFAPReference.assignment();
oInstancia.Products[0].Assignments[1].Vehicle = "Tipo2";
oInstancia.Products[0].Assignments[1].MinimunDuration = 60;
oInstancia.Products[0].Assignments[1].MaximunDuration = 60;
oInstancia.Products[0].Assignments[1].FixedCost = 4500;

// Crea el segundo producto
oInstancia.Products[1] = new wsFAPReference.product();
oInstancia.Products[1].Identifier = "JG0001";
oInstancia.Products[1].From = "BCN";
oInstancia.Products[1].To = "BIO";
oInstancia.Products[1].MinimunDeparture = 600;
oInstancia.Products[1].MaximunDeparture = 720;

// Crea el objeto Assignments del producto, y le añade un objeto Assigment
oInstancia.Products[1].Assignments = new wsFAPReference.assignment[1];

oInstancia.Products[1].Assignments[0] = new wsFAPReference.assignment();
oInstancia.Products[1].Assignments[0].Vehicle = "Tipo2";
oInstancia.Products[1].Assignments[0].MinimunDuration = 60;
oInstancia.Products[1].Assignments[0].MaximunDuration = 60;
oInstancia.Products[1].Assignments[0].FixedCost = 4500;

// Crea el tercer producto
oInstancia.Products[2] = new wsFAPReference.product();
oInstancia.Products[2].Identifier = "JG0002";
oInstancia.Products[2].From = "BCN";
oInstancia.Products[2].To = "BIO";
oInstancia.Products[2].MinimunDeparture = 660;
oInstancia.Products[2].MaximunDeparture = 780;

// Crea el objeto Assignments del producto, y le añade un objeto Assigment
oInstancia.Products[2].Assignments = new wsFAPReference.assignment[1];

oInstancia.Products[2].Assignments[0] = new wsFAPReference.assignment();
oInstancia.Products[2].Assignments[0].Vehicle = "Tipo1";
oInstancia.Products[2].Assignments[0].MinimunDuration = 50;
oInstancia.Products[2].Assignments[0].MaximunDuration = 60;
oInstancia.Products[2].Assignments[0].FixedCost = 4000;

// Crea el cuarto producto
oInstancia.Products[3] = new wsFAPReference.product();
oInstancia.Products[3].Identifier = "JG0003";
oInstancia.Products[3].From = "BIO";
oInstancia.Products[3].To = "MAD";
oInstancia.Products[3].MinimunDeparture = 780;
oInstancia.Products[3].MaximunDeparture = 780;

// Crea el objeto Assignments del producto, y le añade dos objetos Assigment
oInstancia.Products[3].Assignments = new wsFAPReference.assignment[2];

oInstancia.Products[3].Assignments[0] = new wsFAPReference.assignment();
oInstancia.Products[3].Assignments[0].Vehicle = "Tipo1";
oInstancia.Products[3].Assignments[0].MinimunDuration = 55;
oInstancia.Products[3].Assignments[0].MaximunDuration = 55;
oInstancia.Products[3].Assignments[0].FixedCost = 4000;

oInstancia.Products[3].Assignments[1] = new wsFAPReference.assignment();
```

```
oInstancia.Products[3].Assignments[1].Vehicle = "Tipo2";
oInstancia.Products[3].Assignments[1].MinimunDuration = 55;
oInstancia.Products[3].Assignments[1].MaximunDuration = 60;
oInstancia.Products[3].Assignments[1].FixedCost = 4500;

// Crea el acceso al servicio web
wsFAPReference.wsFAPService oCliente = new wsFAPReference.wsFAPService();

// Lanza la petición de resolución
wsFAPReference.request oResultado = oCliente.solve("usuario", "password",
oInstancia);

// Coprueba el resultado
if (oResultado.Status == "Failed")
{
   MessageBox.Show(oResultado.Message[0].Value, "Error en la llamada: ");
}
else if (oResultado.Status == "Ok")
{
    // Espera a que se ejecute el problema
    wsProgressReference.wsProgressService oProgessServiceClient =
                                    new wsProgressReference.wsProgressService();

    string strEstado = "";
    bool bTerminado = false;
    while (!bTerminado)
    {
        // Comprueba el estado
        strEstado = oProgessServiceClient.checkStatus("usuario", "password",
                                                      oResultado.Ticket);

        if (strEstado == "UNKNOWN" || strEstado == "ERROR" ||
            strEstado == "PROCESSED")
            break;
        else if (strEstado == "ALLOCATED" || strEstado == "IN_PROCESS")
            System.Threading.Thread.Sleep(1000);
        else // Estado desconocido
            break;
    }

    if (strEstado == "PROCESSED")
    {
        // Lee la solución
        wsFAPReference.fapSolution oSolucion =
             oCliente.getSolution("usuario", "password", oResultado.Ticket);

        // Tratar la solución
    }
    else if (strEstado == "UNKNOWN")
    {
        MessageBox.Show("El ticket de acceso a la solución no es válido.
                         Contacte con al administrador");
    }
    else if (strEstado == "ERROR")
    {
        MessageBox.Show("Se han producido errores en la planificación");
    }
    else
    {
        MessageBox.Show("Se ha producido un error desconocido.
```

```
156                         Contacte con al administrador");
157             }
158         }
159 }
```

Figura 3-6. Código fuente C# para la ejecución del FAP de VETU

3.5 Ejemplo de uso desde NetBeans

En este apartado, se muestra un programa de ejemplo para resolver una instancia de FAP, mediante el sistema SaaS de VETU, escrito en Java por medio del entorno de desarrollo NetBeans.

NetBeans crea todas las clases necesarias para acceder a servicios web, a partir de la definición WSDL del servicio. Para crear las clases de acceso al servicio web de resolución del FAP de VETU, se deben seguir los siguientes pasos:

1. Pulsar con el botón derecho sobre el proyecto, elegir "New" y en el desplegable que aparece pulsar sobre "Web Service Client".

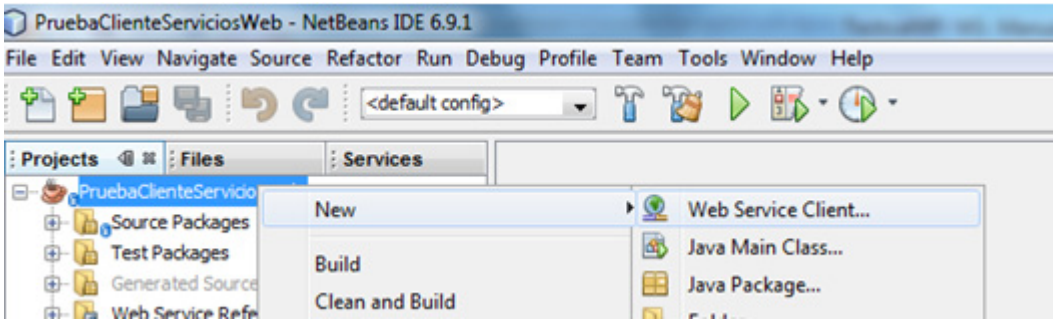

Figura 3-7. Paso 1 - Creación de cliente del servicio web NebBeans

Aparecerá una ventana emergente.

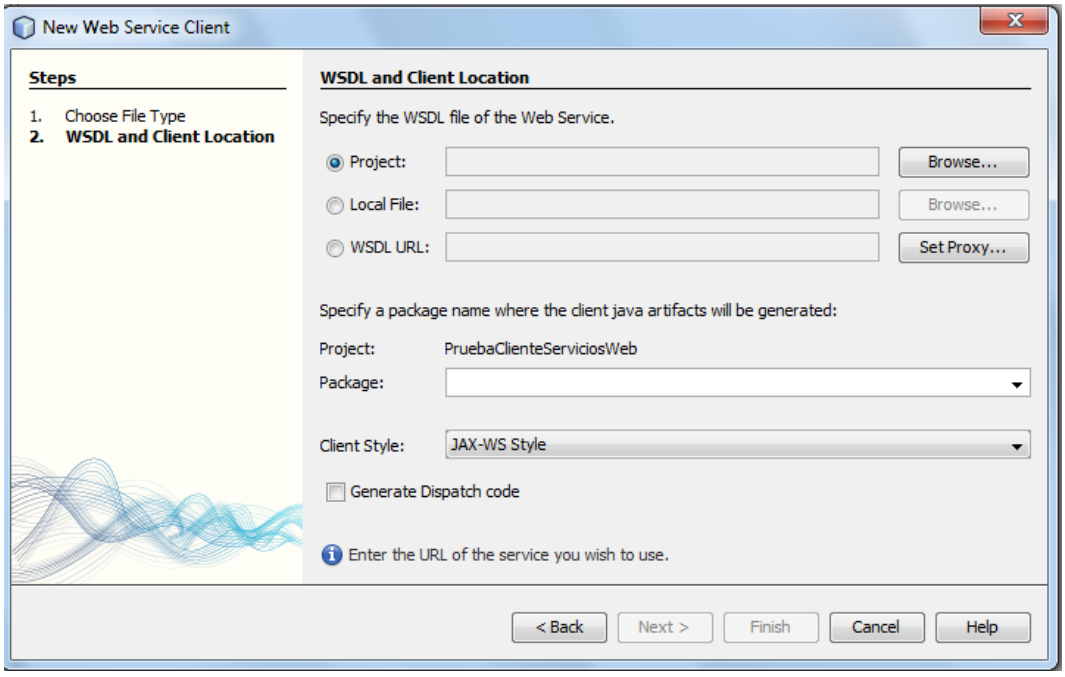

Figura 3-8. Paso 1 - Creación de cliente del servicio web NebBeans (2)

2. Marcar la opción "WSDL URL", escribir http://www.vetu.es/wsFAP/wsFAP?wsdl en la caja de edición y pulsar el botón "Finish".

En este momento se han creado una serie de clases que permiten acceder al servicio web wsFAP, con acceso a las funciones solve y getSolution. En este momento se dispone de las clases necesarias para acceder al servicio web. Las clases principales que se han creado son las siguientes:

- WsFAP: Es la clase de acceso al servicio web. Tiene las funciones solve y getSolution.
- Fap Es la clase que encapsula la definición de la instancia de problema que se quiere resolver. Contiene a las clases Fleet, Products, y EmptyProducts que contienen la definición de los tipos de vehículo y de los productos, que permiten definir la flota de vehículos y el plan comercial. La clase Fleet contiene la definición de los tipos de vehículos, la clase VehicleType. La clase Products contiene la definición de los objetos Product.
- FapSolution: Es la clase que encapsula la solución del problema que se ha resuelto. Contiene a las clases Result, ObjectiveValue y Rotations, que tiene la información de la resolución del problema, de los valores encontrados y de las rotaciones, respectivamente. Cada una de estas clases contiene a otras clases, según el formato definido para la solución en el apartado 3.2.2 de este capítulo.

Para crear las clases de acceso al servicio wsProgress, se deben repetir los pasos anteriores, escribiendo en el punto 2.- la URL del servicio web de acceso al estado de resolución de problemas: http://www.vetu.es/wsProgress/wsProgress?wsdl. Una vez hecho, se habrá creado la clase WsProgress, que es el acceso al servicio web, y tiene la función checkStatus.

El código fuente necesario para la resolución del FAP mediante los servicios web de VETU se muestra en la Figura 3-9. Este código fuente no sigue las mejores prácticas de programación, porque su objetivo es la simplicidad, para mostrar el acceso al servicio SaaS de VETU. Los números de línea de la columna de la izquierda se muestran solo para facilitar la siguiente explicación. Desde las líneas 1 a la 117 se definen los datos del problema a resolver, y desde la línea 120 hasta el final, es el acceso a los servicios web.

En la línea 2 se crea la instancia de la clase Fap, que encapsula toda la información del problema, que son la flota de vehículos y los productos a planificar.

En la línea 5 se crea la flota de vehículos, que es una colección de 2 vehículos. Antes de fijar los datos de cada tipo de vehículo, es necesario crear el objeto vehicleType (líneas 9 y 15). Una vez creado este objeto, se deben fijar todos los datos del tipo de vehículo. Desde la línea 9 a la 12 se crea un tipo de vehículo de nombre "Tipo1", con un tiempo mínimo de escala de 60 minutos y un coste fijo de 21.000 um. Desde la línea 15 a la 18 se crea un tipo de vehículo de nombre "Tipo2", con un tiempo mínimo de escala de 90 minutos y un coste fijo de 36.000 um.

En la línea 22 se crea la lista de productos, que es una colección de 4 productos de tipo Product. Antes de fijar los datos de un producto, es necesario crearlo (en la línea 26 se crea el primer producto, y en las líneas 53, 73 y 93 se crean los otros tres productos). Una vez creado el producto, se especifica el valor de sus campos. Por una parte están los datos generales, por ejemplo desde la línea 27 a la 31 para el primer producto, donde se indica que se trata del trayecto JG000, que va de MAD a BCN y sale en el minuto 540 (09:00 horas). Los datos generales del segundo producto se definen entre las líneas 54 y 58, que indican que el trayecto JG001 va desde BCN a BIO, y puede salir entre el minuto 600 (10:00 horas) y el

minuto 720 (12:00 horas). De forma similar, entre las líneas 74 y 78 se define los datos generales del tercer trayecto, y entre las líneas 94 y 98 se definen los datos generales de cuarto producto.

La definición de un productos requiere al menos un objeto assignment, que indica la información del trayecto que depende del tipo de vehículo. Antes de crear estos objetos assignment, se debe crear el objeto Assignments, que es una colección de objetos Assignment. Por ejemplo, en la línea 36 se crea la colección de objetos assignment en el primer producto, a la que se le añadirán dos objetos en las líneas 43 y 50. De la misma forma, en la línea 63 se crea la colección para los objetos assignment del segundo producto, a la que se le asignará un objeto en la líneas 70. En la línea 83 se crea la colección de objetos assignment para el tercer producto, a la que se le asignará el objeto en la línea 90. Por último, en la línea 103 se crea la colección de objetos assignment para el cuarto producto, a la que se le asignarán dos objetos en las líneas 110 y 117. Una vez creada la colección de objetos assignment, se debe crear cada uno de los objetos assignment. Como ejemplo, en la línea 38 se crea el primer objeto assignment del primer producto, y en la línea 45 se crea el segundo objeto assignment del primer producto. Una vez creado cada objeto assigment, se indica el valor de sus campos. Como ejemplo, entre las líneas 39 y 42 se indica que el trayecto JG0000 lo puede realizar un vehículo de tipo Tipo1, con un coste de 4.000 um y una duración de trayecto entre 55 y 65 minutos, o entre las líneas 66 y 69 se indica que el trayecto JG0001 lo puede realizar un vehículo de tipo Tipo2 con un coste de 4.500 um y una duración fija de 60 minutos.

Una vez definido el problema, se debe enviar al servicio web: en la línea 120 se crea el acceso al servicio web, y en las líneas 124 y 125 se envía el problema al servicio web, con el usuario y la password proporcionadas al contratar el servicio FAP en VETU. La llamada a la función solve devuelve un objeto request, que indica si la llamada al servicio web ha tenido algún problema. En la línea 127 se comprueba si ha existido algún error. Si no ha habido errores, se entra en la parte de búsqueda de la solución. En las líneas 132 a 135 se crea el acceso al servicio web que informa del estado de resolución del problema. Desde la línea 139 a la 153 se define un bucle con un timer que espera a que se resuelva el problema. Desde las líneas 156 a la 181, se comprueba el estado final de resolución. Si el estado es PROCESSED, se accede a la solución del problema (líneas 158 a 159), que viene encapsulada en el objeto fapSolution.

```
1  // Crea la instancia de problema
2  serviciowebfap.Fap fapProblem = new serviciowebfap.Fap();
3
4  // Crea la flota de vehículos
5  Fleet fleet = new Fleet();
6  fapProblem.setFleet(fleet);
7
8  //Crea los objetos vehicle
9  VehicleType vehicleType1 = new VehicleType();
10 vehicleType1.setName("Tipo1");
11 vehicleType1.setTimeSpam(60);
12 vehicleType1.setFixedCost(2100);
13 fleet.getVehicleType().add(vehicleType1);
14
15 VehicleType vehicleType2 = new VehicleType();
16 vehicleType2.setName("Tipo2");
17 vehicleType2.setTimeSpam(90);
18 vehicleType2.setFixedCost(3600);
19 fleet.getVehicleType().add(vehicleType2);
20
21 // Crea los productos
```

```
22  Products products = new Products();
23  fapProblem.setProducts(products);
24
25  // Crea el primer producto
26  Product producto1 = new Product();
27  producto1.setIdentifier("JG0000");
28  producto1.setFrom("MAD");
29  producto1.setTo("BCN");
30  producto1.setMinimunDeparture(540);
31  producto1.setMaximunDeparture(540);
32  products.getProduct().add(producto1);
33
34  // Crea el objeto Assignments del primer producto, y le añade dos objetos
35  Assigment
36  Assignments assignments1 = new Assignments();
37  producto1.setAssignments(assignments1);
38  Assignment assignment11 = new Assignment();
39  assignment11.setVehicle("Tipo1");
40  assignment11.setMinimunDuration(55);
41  assignment11.setMaximunDuration(65);
42  assignment11.setFixedCost(4000);
43  assignments1.getAssignment().add(assignment11);
44
45  Assignment assignment12 = new Assignment();
46  assignment12.setVehicle("Tipo2");
47  assignment12.setMinimunDuration(60);
48  assignment12.setMaximunDuration(60);
49  assignment12.setFixedCost(4500);
50  assignments1.getAssignment().add(assignment12);
51
52  // Crea el segundo producto
53  Product producto2 = new Product();
54  producto2.setIdentifier("JG0001");
55  producto2.setFrom("BCN");
56  producto2.setTo("BIO");
57  producto2.setMinimunDeparture(600);
58  producto2.setMaximunDeparture(720);
59  products.getProduct().add(producto2);
60
61  // Crea el objeto Assignments del segundo producto, y le añade un objeto
62  Assigment
63  Assignments assignments2 = new Assignments();
64  producto2.setAssignments(assignments2);
65  Assignment assignment21 = new Assignment();
66  assignment21.setVehicle("Tipo2");
67  assignment21.setMinimunDuration(60);
68  assignment21.setMaximunDuration(60);
69  assignment21.setFixedCost(4500);
70  assignments2.getAssignment().add(assignment21);
71
72  // Crea el tercer producto
73  Product producto3 = new Product();
74  producto3.setIdentifier("JG0002");
75  producto3.setFrom("BCN");
76  producto3.setTo("BIO");
77  producto3.setMinimunDeparture(660);
78  producto3.setMaximunDeparture(780);
79  products.getProduct().add(producto3);
80
81  // Crea el objeto Assignments del tercer producto, y le añade un objeto
```

```
Assigment
Assignments assignments3 = new Assignments();
producto3.setAssignments(assignments3);
Assignment assignment31 = new Assignment();
assignment31.setVehicle("Tipo1");
assignment31.setMinimunDuration(50);
assignment31.setMaximunDuration(60);
assignment31.setFixedCost(4000);
assignments3.getAssignment().add(assignment31);

// Crea el cuarto producto
Product producto4 = new Product();
producto4.setIdentifier("JG0003");
producto4.setFrom("BIO");
producto4.setTo("MAD");
producto4.setMinimunDeparture(780);
producto4.setMaximunDeparture(780);
products.getProduct().add(producto4);

// Crea el objeto Assignments del cuarto producto, y le añade dos objetos
Assigment
Assignments assignments4 = new Assignments();
producto4.setAssignments(assignments4);
Assignment assignment41 = new Assignment();
assignment41.setVehicle("Tipo1");
assignment41.setMinimunDuration(55);
assignment41.setMaximunDuration(55);
assignment41.setFixedCost(4000);
assignments4.getAssignment().add(assignment41);

Assignment assignment42 = new Assignment();
assignment42.setVehicle("Tipo2");
assignment42.setMinimunDuration(55);
assignment42.setMaximunDuration(60);
assignment42.setFixedCost(4500);
assignments4.getAssignment().add(assignment42);

// Crea el acceso al servicio web
WSFAPService service = new WSFAPService();
serviciowebfap.WSFAP port = service.getWSFAPPort();

// Lanza la petición de resolución
Request oResultado = port.solve("usuario", "password", fapProblem);

if (oResultado.getStatus().equals("Failed")) {
    System.out.println("Error en la llamada: " +
oResultado.getMessage().get(0).getValue());
} else if (oResultado.getStatus().equals("Ok")) {
    // Espera a que se ejecute el problema
    WSExecutionProgressService serviceExecutionProgress = new
WSExecutionProgressService();
    WSExecutionProgress portProgress =
serviceExecutionProgress.getWSExecutionProgressPort();

    String strEstado = "";
    boolean bTerminado = false;
    while (!bTerminado) {
        strEstado = portProgress.checkStatus("usuario", "password",
oResultado.getTicket());

```

```
142         if (strEstado.equals("UNKNOWN") || strEstado.equals("ERROR") ||
143 strEstado.equals("PROCESSED")) {
144             bTerminado = true;
145         } else if (strEstado.equals("ALLOCATED") ||
146 strEstado.equals("IN_PROCESS")) {
147             Thread.sleep(5000);
148         } else // Estado desconocido
149         {
150             bTerminado = true;
151         }
152     }
153 
154     // Lee el resultado del problema
155     if (strEstado.equals("PROCESSED")) {
156         // Lee la solución
157         FapSolution fapSolution = port.getSolution("usuario", "password",
158 oResultado.getTicket());
159         // Tratar la solución.
160         // Mostrar resultado
161         JAXBContext jc = JAXBContext.newInstance(String.class,
162 serviciowebfap.FapSolution.class);
163         JAXBIntrospector introspector = jc.createJAXBIntrospector();
164         Marshaller marshaller = jc.createMarshaller();
165         marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT,
166 Boolean.TRUE);
167         if (null == introspector.getElementName(fapSolution)) {
168             JAXBElement jaxbElement = new JAXBElement(new QName("ROOT"),
169 Object.class, fapSolution);
170             marshaller.marshal(jaxbElement, System.out);
171         } else {
172             marshaller.marshal(fapSolution, System.out);
173         }
174     } else if (strEstado.equals("UNKNOWN")) {
175         System.out.println("Ticket de acceso a la solución no válido");
176     } else if (strEstado.equals("ERROR")) {
177         System.out.println("Se han producido errores en la planificación");
178     } else {
179         System.out.println("Se ha producido un error desconocido");
180     }
181 }
182 
```

Figura 3-9. Código fuente Java en NetBeans para la ejecución del FAP de VETU

3.6 Ejemplo de uso desde Eclipse

En este apartado, se muestra un programa de ejemplo para resolver una instancia de FAP, mediante el sistema SaaS de VETU, escrito en Java por medio del entorno de desarrollo Eclipse.

Eclipse crea todas las clases necesarias para acceder a servicios web, a partir de la definición WSDL del servicio. Para crear las clases de acceso al servicio web de resolución del FAP de VETU, se debe tener instalado en Eclipse el módulo de Java EE y se deben seguir los siguientes pasos:

1. Pulsar con el botón derecho sobre el proyecto, elegir “New” y en el desplegable que aparece pulsar sobre “Other”.

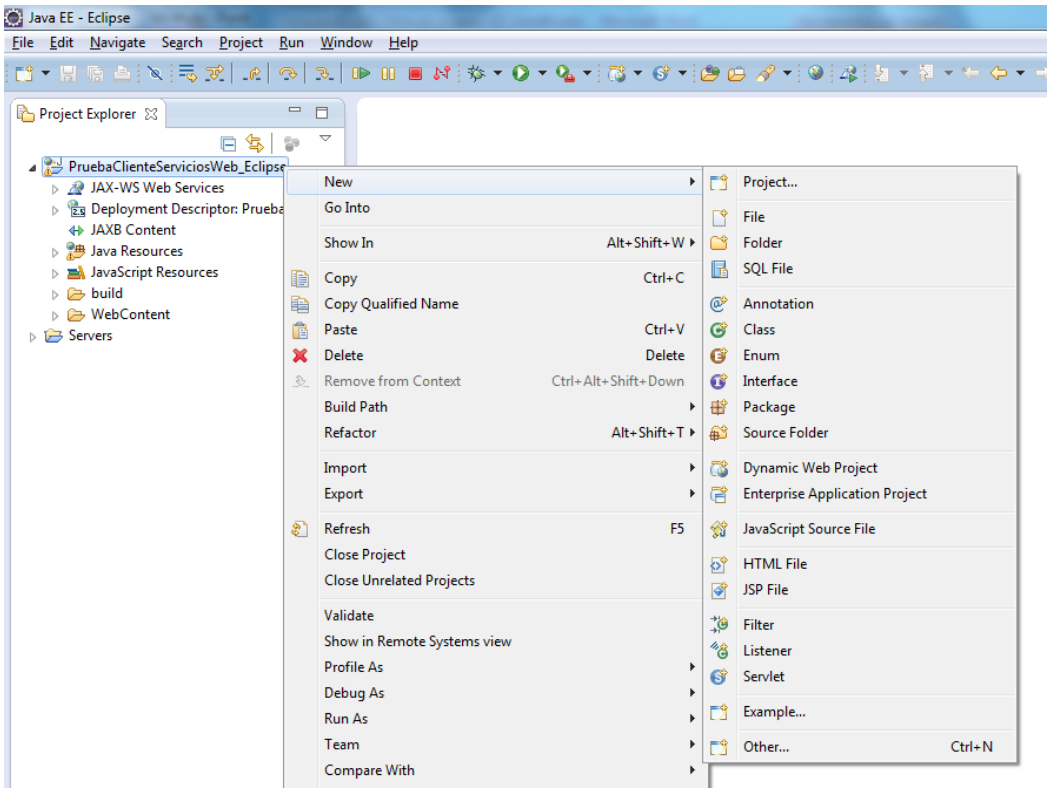

Figura 3-10. Paso 1 - Creación de cliente del servicio web Eclipse

Aparecerá una ventana emergente para seleccionar el tipo de fichero que se desea crear.

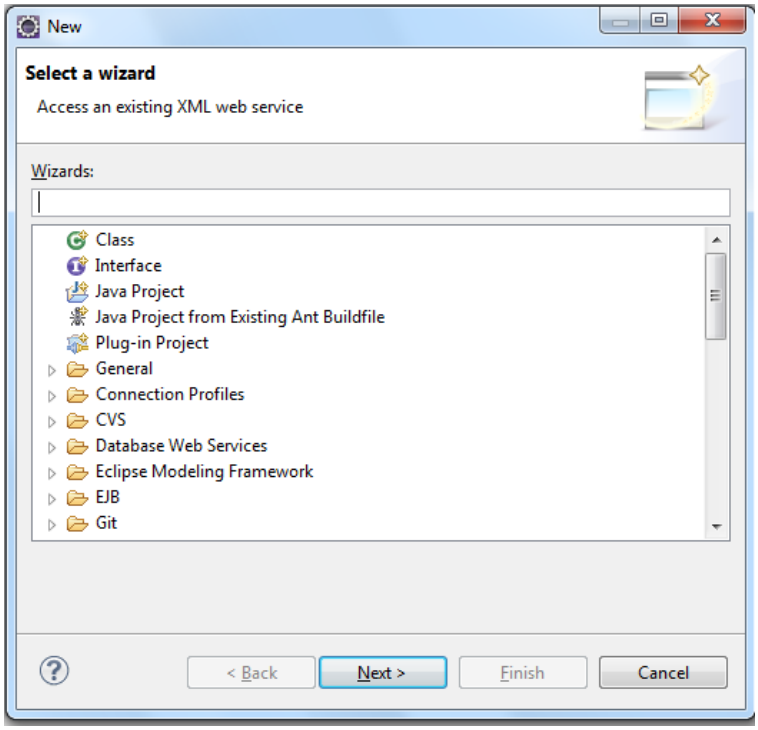

Figura 3-11. Paso 1 - Creación de cliente del servicio web Eclipse (2)

2. Se busca la carpeta de "Web Services" y se elige el tipo de fichero "Web Service Client", y se pulsará sobre el botón "Next".

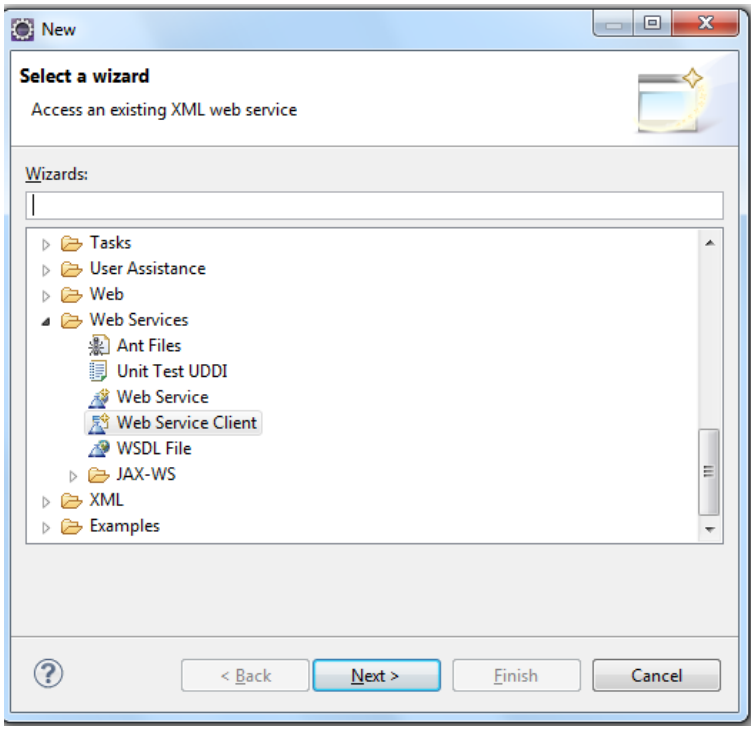

Figura 3-12. Paso 2 - Creación de cliente del servicio web Eclipse

Se mostrará una ventana para configurar el Cliente del Servicio Web.

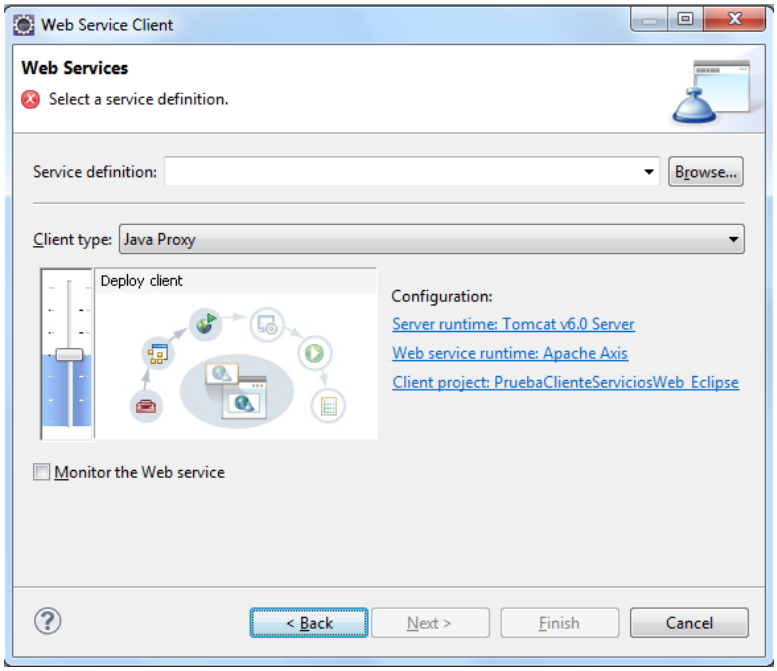

Figura 3-13. Paso 2 - Creación de cliente del servicio web Eclipse (2)

3. En la caja de texto "Service definition" se escribirá lo siguiente: http://www.vetu.es/wsFAP/wsFAP?wsdl. En caso de no tener configurados un servidor o un servicio web, se pulsará sobre el nombre y se elegirá uno de los mostrados en el

listado emergente. No hay que tocar el tipo de cliente ni el nivel, tiene que estar en la opción marcada por defecto: "Deploy client".

4. Una vez configurado se pulsará "Finish".

En este momento se han creado una serie de clases que permiten acceder al servicio web wsFAP, con acceso a las funciones solve y getSolution. En este momento se dispone de las clases necesarias para acceder al servicio web. Las clases principales que se han creado son las siguientes:

- WsFap: Es la clase de acceso al servicio web. Tiene las funciones solve y getSolution.
- Fap Es la clase que encapsula la definición de la instancia de problema que se quiere resolver. Contiene a las clases Fleet, Products, y EmptyProducts que contienen la definición de los tipos de vehículo y de los productos, que permiten definir la flota de vehículos y el plan comercial. La clase Fleet contiene la definición de los tipos de vehículos, la clase VehicleType. La clase Products contiene la definición de los objetos Product.
- FapSolution: Es la clase que encapsula la solución del problema que se ha resuelto. Contiene a las clases Result, ObjectiveValue y Rotations, que tiene la información de la resolución del problema, de los valores encontrados y de las rotaciones, respectivamente. Cada una de estas clases contiene a otras clases, según el formato definido para la solución en el apartado 3.2.2 de este capítulo.

Para crear las clases de acceso al servicio wsProgress, se deben repetir los pasos anteriores, escribiendo en el punto 3.- la URL del servicio web de acceso al estado de resolución de problemas: http://www.vetu.es/wsProgress/wsProgress?wsdl. Una vez hecho, se habrá creado la clase WsProgress, que es el acceso al servicio web, y tiene la función checkStatus.

El código fuente necesario para la resolución del FAP mediante los servicios web de VETU se muestra en la Figura 3-14. Los números de línea de la columna de la izquierda se muestran solo para facilitar la siguiente explicación. Desde las líneas 1 a la 98 se definen los datos del problema a resolver. En las líneas 2 y 3 se crea la instancia de la clase TacticalPlanning, que encapsula toda la información del problema.

En la línea 2 se crea la instancia de la clase Fap, que encapsula toda la información del problema, que son la flota de vehículos y los productos a planificar.

En la línea 5 se crea la flota de vehículos, que es un array de 2 vehículos, y se le asigna a la clase Fap. Antes de fijar los datos de cada tipo de vehículo, es necesario crear el objeto vehicleType (líneas 9 y 15). Una vez creado este objeto, se le asigna la posición correspondiente del array de vehículos (líneas 10 y 16), y se deben fijar todos los datos del tipo de vehículo. Desde la línea 11 a la 13 se crea un tipo de vehículo de nombre "Tipo1", con un tiempo mínimo de escala de 60 minutos y un coste fijo de 21.000 um. Desde la línea 17 a la 19 se crea un tipo de vehículo de nombre "Tipo2", con un tiempo mínimo de escala de 90 minutos y un coste fijo de 36.000 um.

En la línea 22 se crea la lista de 4 productos, de tipo Product, y se le asigna al objeto fap (línea 23). Antes de fijar los datos de un producto, es necesario crearlo (en la línea 26 se crea el primer productos, y en las líneas 53, 73 y 93 se crean los otros tres productos). Una vez creado el producto, se especifica el valor de sus campos. Por una parte están los datos generales, por ejemplo desde la línea 28 a la 32 para el primer producto, donde se indica que se trata del trayecto JG000, que va de MAD a BCN y sale en el minuto 540 (09:00 horas). Los

datos generales del segundo producto se definen entre las líneas 55 y 59, que indican que el trayecto JG001 va desde BCN a BIO, y puede salir entre el minuto 600 (10:00 horas) y el minuto 720 (12:00 horas). De forma similar, entre las líneas 75 y 79 se define los datos generales del tercer trayecto, y entre las líneas 95 y 99 se definen los datos generales de cuarto producto.

La definición de un productos requiere al menos un objeto assignment, que indica la información del trayecto que depende del tipo de vehículo. Antes de crear estos objetos assignment, se debe crear el objeto assignments, que es un array de objetos Assignment. Por ejemplo, en la línea 36 se crea un arrray para dos objetos assignment en el primer producto. De la misma forma, en la línea 63 se crea un array con dos objetos assignment para el segundo producto, en la línea 83 se crea un array con un objeto assignment para el tercer producto, y en la línea 403 se crea un array con dos objetos assignment para el cuarto producto. Una vez creado el array de objeto assignment, se debe crear cada uno de los objetos assignment. Como ejemplo, en las líneas 38 y 39 se crea el primer objeto assignment del primer producto, y en las líneas 45 y 46 se crea el segundo objeto assignment del primer producto. Una vez creado cada objeto assigment, se indican el valor de sus campos. Como ejemplo, entre las líneas 40 y 43 se indica que el trayecto JG0000 lo puede realizar un vehículo de tipo Tipo1, con un coste de 4.000 um y una duración de trayecto entre 55 y 65 minutos, o entre las líneas 67 y 70 se indica que el trayecto JG0001 lo puede realizar un vehículo de tipo Tipo2 con un coste de 4.500 um y una duración fija de 60 minutos.

Una vez definido el problema, se debe enviar al servicio web: en la línea 120 se crea el acceso al servicio web, y en las líneas 124 y 125 se envía el problema al servicio web, con el usuario y la password obtenidas al contratar el servicio FAP de VETU. La llamada a la función solve devuelve un objeto request, que indica si la llamada al servicio web ha tenido algún problema. En la línea 127 se comprueba si ha existido algún error. Si no ha habido errores, se entra en la parte de búsqueda de la solución. En las líneas 140 a 141 se crea el acceso al servicio web que informa del estado de resolución del problema. Desde la línea 139 a la 153 se define un bucle con un timer que espera a que se resuelva el problema. Desde las líneas 156 a la 181, se comprueba el estado final de resolución. Si el estado es PROCESSED, se accede a la solución del problema (líneas 158 y 159), que viene encapsulada en el objeto fapSolution.

```
1  // Crea la instancia de problema
2  servicioWebFAP.Fap fap = new servicioWebFAP.Fap();
3
4  // Crea la flota de vehículos
5  VehicleType[] fleet = new VehicleType[2];
6  fap.setFleet(fleet);
7
8  // Crea los objetos Vehicle
9  VehicleType vehicle1 = new VehicleType();
10 fleet[0] = vehicle1;
11 vehicle1.setName("Tipo1");
12 vehicle1.setTimeSpam(60);
13 vehicle1.setFixedCost(21000);
14
15 VehicleType vehicle2 = new VehicleType();
16 fleet[1] = vehicle2;
17 vehicle2.setName("Tipo2");
18 vehicle2.setTimeSpam(90);
19 vehicle2.setFixedCost(36000);
20
21 // Crea los productos
22 Product[] products = new Product[4];
23 fap.setProducts(products);
24
25 // Crea el primer producto
```

```
26  Product product1 = new Product();
27  products[0] = product1;
28  product1.setIdentifier("JG0000");
29  product1.setFrom("MAD");
30  product1.setTo("BCN");
31  product1.setMinimunDeparture(540);
32  product1.setMaximunDeparture(540);
33
34  // Crea el objeto Assignments del primer producto, y le añade dos objetos
35  Assigment
36  Assignment[] assignments1 = new Assignment[2];
37  product1.setAssignments(assignments1);
38  Assignment assignment11 = new Assignment();
39  assignments1[0] = assignment11;
40  assignment11.setVehicle("Tipo1");
41  assignment11.setMinimunDuration(55);
42  assignment11.setMaximunDuration(65);
43  assignment11.setFixedCost(4000);
44
45  Assignment assignment12 = new Assignment();
46  assignments1[1] = assignment12;
47  assignment12.setVehicle("Tipo2");
48  assignment12.setMinimunDuration(60);
49  assignment12.setMaximunDuration(60);
50  assignment12.setFixedCost(4500);
51
52  // Crea el segundo producto
53  Product product2 = new Product();
54  products[1] = product2;
55  product2.setIdentifier("JG0001");
56  product2.setFrom("BCN");
57  product2.setTo("BIO");
58  product2.setMinimunDeparture(600);
59  product2.setMaximunDeparture(720);
60
61  // Crea el objeto Assignments del segundo producto, y le añade un objeto
62  Assigment
63  Assignment[] assignments2 = new Assignment[1];
64  product2.setAssignments(assignments2);
65  Assignment assignment21 = new Assignment();
66  assignments2[0] = assignment21;
67  assignment21.setVehicle("Tipo2");
68  assignment21.setMinimunDuration(60);
69  assignment21.setMaximunDuration(60);
70  assignment21.setFixedCost(4500);
71
72  // Crea el tercer producto
73  Product product3 = new Product();
74  products[2] = product3;
75  product3.setIdentifier("JG0002");
76  product3.setFrom("BCN");
77  product3.setTo("BIO");
78  product3.setMinimunDeparture(660);
79  product3.setMaximunDeparture(780);
80
81  // Crea el objeto Assignments del tercer producto, y le añade un objeto
82  Assigment
83  Assignment[] assignments3 = new Assignment[1];
84  product3.setAssignments(assignments3);
85  Assignment assignment31 = new Assignment();
```

```
assignments3[0] = assignment31;
assignment31.setVehicle("Tipo1");
assignment31.setMinimunDuration(50);
assignment31.setMaximunDuration(60);
assignment31.setFixedCost(4000);

// Crea el cuarto producto
Product product4 = new Product();
products[3] = product4;
product4.setIdentifier("JG0003");
product4.setFrom("BIO");
product4.setTo("MAD");
product4.setMinimunDeparture(780);
product4.setMaximunDeparture(780);

// Crea el objeto Assignments del cuarto producto, y le añade dos objetos
Assigment
Assignment[] assignments4 = new Assignment[2];
product4.setAssignments(assignments4);
Assignment assignment41 = new Assignment();
assignments4[0] = assignment41;
assignment41.setVehicle("Tipo1");
assignment41.setMinimunDuration(55);
assignment41.setMaximunDuration(55);
assignment41.setFixedCost(4000);

Assignment assignment42 = new Assignment();
assignments4[1] = assignment42;
assignment42.setVehicle("Tipo2");
assignment42.setMinimunDuration(55);
assignment42.setMaximunDuration(60);
assignment42.setFixedCost(4500);

// Crea el acceso al servicio web
WSFAPService service = new WSFAPServiceLocator();
servicioWebFAP.WSFAP port = service.getWSFAPPort();

// Lanza la petición de resolución
Request oResultado = port.solve("usuario", "password", fap);

if (oResultado.getStatus().equals("Failed")) {
    System.out.println("Error en la llamada: " +
oResultado.getMessage(0).get_value());
} else if (oResultado.getStatus().equals("Ok")) {
    // Espera a que se ejecute el problema
    WSExecutionProgressService serviceExecutionProgress = new
WSExecutionProgressServiceLocator();
    WSExecutionProgress portProgress =
serviceExecutionProgress.getWSExecutionProgressPort();

    String strEstado = "";
    boolean bTerminado = false;
    while (!bTerminado) {
        strEstado = portProgress.checkStatus("usuario", "password",
oResultado.getTicket());

        if (strEstado.equals("UNKNOWN") || strEstado.equals("ERROR") ||
strEstado.equals("PROCESSED")) {
            bTerminado = true;
        } else if (strEstado.equals("ALLOCATED") ||
```

```
146 strEstado.equals("IN_PROCESS")) {
147             Thread.sleep(1000);
148         } else // Estado desconocido
149         {
150             bTerminado = true;
151         }
152     }
153 
154     // Lee el resultado del problema
155     if (strEstado.equals("PROCESSED")) {
156         // Lee la solución
157         FapSolution wFSolution = port.getSolution("usuario", "password",
158 oResultado.getTicket());
159         // Tratar la solución.
160         //Mostrar resultado
161         JAXBContext jc = JAXBContext.newInstance(String.class,
162 servicioWebFAP.FapSolution.class);
163         JAXBIntrospector introspector = jc.createJAXBIntrospector();
164         Marshaller marshaller = jc.createMarshaller();
165         marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT,
166 Boolean.TRUE);
167         if (null == introspector.getElementName(wFSolution)) {
168             JAXBElement jaxbElement = new JAXBElement(new QName("ROOT"),
169 Object.class, wFSolution);
170             marshaller.marshal(jaxbElement, System.out);
171         } else {
172             marshaller.marshal(wFSolution, System.out);
173         }
174     } else if (strEstado.equals("UNKNOWN")) {
175         System.out.println("Ticket de acceso a la solución no válido ");
176     } else if (strEstado.equals("ERROR")) {
177         System.out.println("Se han producido errores en la planificación");
178     } else {
179         System.out.println("Se ha producido un error desconocido ");
180     }
181 }
```

Figura 3-14. Código fuente Java en Eclipse para la ejecución del FAP de VETU

4 Mensajes

En este capítulo se muestran todos los mensajes que pueden devolver las funciones de los servicios web de VETU, para la resolución del FAP.

4.1 Mensajes de envío de problemas

La Tabla 4-1 muestra la lista de mensajes que puede devolver el objeto Request de la función solve() del servicio web wsFAP, cuando se envía una solicitud de resolución de una instancia del FAP.

Código	Descripción
SRV0	The problem has been received. It is ready to solve. El problema enviado es correcto y está listo para su resolución. Este proceso de resolución puede tardar varios minutos. Se puede comprobar el estado de resolución, y una vez terminado, se puede acceder a la solución.
SRV1	User authentication failed. El usuario no se ha podido autenticar correctamente.
SRV2	The problem does not have a valid format. El formato de la entrada del problema no corresponde con el definido por la plataforma VETU, y no se puede resolver.
SRV3	The user has not access to this problem. El usuario enviado con la función solve() no tiene contrato para resolver el problema de VETU.
SRV4	The contract has expired El contrato para acceder a resolución del problema de VETU ha caducado y no puede ser utilizado el servicio.

Tabla 4-1. Mensajes que devuelve la función solve() del servicio web wsFAP

4.2 Mensajes del FAP

La siguiente tabla muestra los mensajes de error que se pueden mostrar en la resolución de una instancia de problema de asignación de flotas, resuelto mediante los servicios web de VETU.

Código	Descripción
FAP0	The FAP problem has been successfully solved El problema se ha resuelto correctamente.
FAP1	There are bugs in the entry file. El formato del fichero de entrada del FAP no corresponde con el definido por la plataforma VETU, y el problema no se puede resolver.
FAP2	Error solving the problem. Please contact support team. Ha ocurrido un problema inesperado en la resolución del problema. El equipo técnico de VETU ya lo ha detectado y está estudiando el problema. Puede ponerse en contacto con el servicio técnico.
FAP3	The problem has no feasible solution. Please check problem data. El problema no tiene solución. Si después de revisar el problema, no encuentra la causa por la que no se puede resolver, contacte con el servicio técnico.
FAP4	Data entry has not defined the Fleet object. El objeto Fleet define los tipos de vehículos que se pueden asignar a los trayectos. Si no se definen los tipos de vehículo, no se puede resolver el problema.
FAP5	Data entry has not defined Products object. El objeto Products define los trayectos que se van a asignar a los tipos de vehículo. Si no se definen los trayectos, no se puede resolver el problema.
FAP6	There is not any VehicleType object. No se ha definido ningún tipo de vehículo para asignar trayectos. Al menos, se debe definir un tipo de vehículo.
FAP7	There is not any Product object. No se ha definido ningún producto. No se puede resolver un problema que no tiene productos definidos.
FAP8	Some vehicle type has not defined Name objet Algún objeto VehicleType no tiene definido el objeto Name. Se debe definir el objeto Name y darle un valor único que no tenga ningún otro objeto

	VehicleType.
FAP9	Some vehicle type has empty value for Name object. Algún objeto VehicleType no tiene definido un valor para el objeto Name. Se debe dar un valor al objeto Name, que debe ser un valor único que no tenga ningún otro objeto VehicleType.
FAP10	VehicleType xxx has not defined FixedCost objet. No se ha definido el objeto FixedCost correspondiente al tipo de vehículo xxx. Es necesario especificar el coste fijo del uso de cada tipo de vehículo.
FAP11	VehicleType xxx has empty value for FixedCost objet. No se ha definido el valor del objeto FixedCost correspondiente al tipo de vehículo xxx. Es necesario especificar el coste fijo del uso de cada tipo de vehículo.
FAP12	The value xxx for FixedCost objet for VehicleType called yyy is not valid. El valor xxx que se ha definido para el coste fijo del tipo de vehículo yyy no es válido por que no representa un número real. Se debe especificar un número real positivo.
FAP13	VehicleType xxx has not defined TimeSpam objet. No se ha definido el objeto TimeSpam correspondiente al tipo de vehículo xxx. Es necesario especificar el tiempo mínimo de escala de cada tipo de vehículo.
FAP14	VehicleType xxx has empty value for TimeSpam objet. No se ha definido el valor del objeto TimeSpam correspondiente al tipo de vehículo xxx. Es necesario especificar el tiempo mínimo de escala de cada tipo de vehículo. Debe ser un número entero positivo, expresado en minutos.
FAP15	The value xxx for TimeSpam objet for VehicleType called yyy is not valid. El valor xxx que se ha definido para el tiempo mínimo de escala del tipo de vehículo yyy no es válido por que no representa un número entero. Se debe especificar un número entero positivo.
FAP16	Some product has not defined Identifier objet Algún objeto Product no tiene definido el objeto Identifier. Se debe definir el objeto Identifier y darle un valor único que no tenga ningún otro objeto Product.
FAP17	Some product has empty value for Identifier object.

	Algún objeto Product no tiene definido un valor para el objeto Identifier. Se debe dar un valor al objeto Identifier, que debe ser un valor único que no tenga ningún otro objeto Product.
FAP18	Product xxx has not defined From objet. No se ha definido el objeto From correspondiente al producto de identificador xxx. Es necesario especificar la estación de salida de cada producto.
FAP19	Product xxx has empty value for From objet. No se ha definido el valor del objeto From correspondiente al producto de identificador xxx. Es necesario especificar la estación de salida de cada producto.
FAP20	Product xxx has not defined To objet. No se ha definido el objeto To correspondiente al producto de identificador xxx. Es necesario especificar la estación de llegada de cada producto.
FAP21	Product xxx has empty value for To objet. No se ha definido el valor del objeto To correspondiente al producto de identificador xxx. Es necesario especificar la estación de llegada de cada producto.
FAP22	Product xxx has not defined MinimunDeparture objet. No se ha definido el objeto MinimunDeparture correspondiente al producto de identificador xxx. Es necesario especificar la mínima posible hora de salida del producto.
FAP23	Product xxx has empty value for MinimunDeparture objet. No se ha definido el valor del objeto MinimunDeparture correspondiente al producto de identificador xxx. Es necesario especificar la mínima posible hora de salida del producto.
FAP24	The value xxx for MinimunDeparture objet, for product yyy is not valid. El valor xxx que se ha definido para la mínima posible salida del producto yyy no es válido. Debe ser un número entero comprendido entre los valores 0 (lunes a las 00:00) y 1080 (domingo a las 24:00).
FAP25	Product xxx has not defined MaximunDeparture objet. No se ha definido el objeto MaximunDeparture correspondiente al producto de identificador xxx. Es necesario especificar la máxima posible hora de salida del producto.

FAP26	Product xxx has empty value for MaximunDeparture objet. No se ha definido el valor del objeto MaximunDeparture correspondiente al producto de identificador xxx. Es necesario especificar la máxima posible hora de salida del producto.
FAP27	The value xxx for MaximunDeparture objet, for product yyy is not valid. El valor xxx que se ha definido para la máxima posible salida del producto yyy no es válido. Debe ser un número entero comprendido entre los valores 0 (lunes a las 00:00) y 1080 (domingo a las 24:00).
FAP28	There is not any Assignments object for product xxx No se ha definido el objeto Assignments para el producto xxx. Es necesario definir para cada producto, el objeto Assignments, y al menos un objeto Assignment.
FAP29	Some Assignment for product xxx has not defined Vehicle objet. No se ha definido el objeto Vehicle dentro de un campo Assignment, para el producto xxx. Todos los objetos Assignment deben tener definido e.
FAP30	Some assignment for product xxx has empty value for Vehicle objet. No se ha definido el valor de objeto Vehicle correspondiente a algún objeto Assignment del producto xxx. Es necesario especificar el vehículo de cada objeto Assigment de los productos.
FAP31	Product xxx has not defined MinimunDuration for some Assignment object. No se ha definido el objeto MinimunDuration en algún objeto Assignment del producto de identificador xxx. Es necesario especificar la mínima duración del producto en cada Assignment.
FAP32	Product xxx has empty value for MinimunDuration for some Assignment object. No se ha definido el valor del objeto MinimunDuration en algún objeto Assignment del producto de identificador xxx. Es necesario especificar la mínima duración como un número entero positivo.
FAP33	The value xxx for MinimunDuration objet, for some Assignment object of the product yyy is not valid. El valor xxx que se ha definido para la mínima duración de algún objeto Assigment del producto yyy no es válido. Debe ser un número entero positivo.
FAP34	Product xxx has not defined MaximunDuration for some Assignment object.

	No se ha definido el objeto MaximunDuration en algún objeto Assignment del producto de identificador xxx. Es necesario especificar la máxima duración del producto en cada Assignment.
FAP35	Product xxx has empty value for MaximunDuration for some Assignment object. No se ha definido el valor del objeto MaximunDuration en algún objeto Assignment del producto de identificador xxx. Es necesario especificar la máxima duración como un número entero positivo.
FAP36	The value xxx for MaximunDuration objet, for some Assignment object of the product yyy is not valid. El valor xxx que se ha definido para la máxima duración de algún objeto Assigment del producto yyy no es válido. Debe ser un número entero positivo.
FAP37	Product xxx has not defined FixedCos for some Assignment object. No se ha definido el objeto FixedCost en algún objeto Assignment del producto de identificador xxx. Es necesario especificar el objeto FixedCost de cada Assignment de los productos.
FAP38	Product xxx has empty value for FixedCost for some Assignment object. No se ha definido el valor del objeto FixedCost en algún objeto Assignment del producto de identificador xxx. Es necesario especificar el coste de cada Assignment, como un número mayor o igual que cero.
FAP39	The value xxx for FixedCost objet, for some Assignment object of the product yyy is not valid. El valor xxx que se ha definido para el coste fijo de algún objeto Assigment del producto yyy no es válido. Debe ser un número mayor o igual que cero.
FAP40	Product xxx has not defined any Assignment object. No se ha definido ningún objeto Assigment para el producto xxx. Cada producto debe definir al menos un objeto Assignment.
FAP41	The timespam xxx defined for vehicle type yyy has an invalid value. El valor xxx definido como tiempo mínimo de escala del tipo de vehículo yyy no es correcto. El tiempo mínimo de escala deber ser mayor que cero y menor de una semana.
FAP42	The time window in the departure (xxx, yyy) defined for product zzz is not valid.

	La ventana de tiempo en la salida que se ha definido para el producto zzz no es válida. La mínima posible salida debe ser menor o igual que la máxima posible salida del trayecto.
FAP43	The assignment given for product xxx with minimum duration of yyy and maximun duration of zzz is not valid. La asignación definida para el producto xxx que tiene una duración mínima de yyy y una duración máxima de zzz no es correcta. La mínima duración deber menor o igual que la máxima duración.
FAP44	Data entry has not defined EmptyProducts object No se definido el objeto EmptyProducts, con los productos en vacío. Si no de define productos en vacío, este objeto puede estar vacío, pero siempre debe aparecer.
FAP45	Some EmptyProduct has not defined From object. Algún objeto EmptyProduct no ha definido la estación de salida. Este campo es obligatorio para definir el producto en vacío.
FAP46	Some EmptyProduct has empty value for From object. El valor de la estación de salida de algún objeto EmptyProduct está vacío. Este campo no puede estar vacío.
FAP47	Some EmptyProduct has not defined To object. Algún objeto EmptyProduct no ha definido la estación de llegada. Este campo es obligatorio para definir el producto en vacío.
FAP48	Some EmptyProduct has empty value for To object. El valor de la estación de llegada de algún objeto EmptyProduct está vacío. Este campo no puede estar vacío.
FAP49	There is not any EmptyAssignments object for empty product from xxx to yyy El producto vacío que sale de la estación xxx y llega a la estación yyy no tiene definido el objeto EmptyAssignments, que es obligatorio.
FSAP50	Some assignment for empty product from xxx to yyy has not defined Vehicle object. No se ha definido el objeto Vehicle en el producto en vacío que sale de xxx y llega a yyy. Es necesario especificar el vehículo en cada objeto EmptyAssignment.

FAP51	Some assignment for empty product from xxx to yyy has empty value for Vehicle object. El objeto Vehicle de algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy, está vacío. Se debe especificar un valor para el vehículo.
FAP52	Empty product from xxx to yyy has not defined MinimunDuration for some EmptyAssignment object. No se ha especificado el objeto MinimunDuration en algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy. Es necesario especificar la duración mínima.
FAP53	Empty product from xxx to yyy has empty value for MinimunDuration for some EmptyAssignment object. El objeto MinimumDuration de algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy, tiene un valor vacío. Se debe especificar el valor de este objeto.
FAP54	The value vvv for MinimunDuration objet, for some Assignment object of the product from xxx to yyy is not valid. El objeto MinimumDuration de algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy, no tiene un valor válido. Se debe especificar un valor correcto para este objeto, que es un número entero.
FAP55	Empty product from xxx to yyy has not defined MaximunDuration for some Assignment objet. No se ha especificado el objeto MaximunDuration en algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy. Es necesario especificar la duración máxima.
FAP56	Empty product from xxx to yyy has empty value for MaximunDuration for some Assignment object. El objeto MaximumDuration de algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy, tiene un valor vacío. Se debe especificar el valor de este objeto.
FAP57	The value vvv for MaximunDuration object, for some Assignment object of the empty product from xxx to yyy is not valid El objeto MaximumDuration de algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy, no tiene un valor válido. Se debe especificar un valor correcto para este objeto, que es un número entero.
FAP58	Empty product from xxx to yyy has not defined FixedCost for some Assignment

	object. No se ha especificado el objeto FixedCost en algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy. Es necesario especificar el coste del trayecto.
FAP59	Empty product from xxx to yyy has empty value for FixedCost for some Assignment object. El objeto FixedCost de algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy, tiene un valor vacío. Se debe especificar el valor de este objeto.
FAP60	The value vvv for FixedCost object for some Assignment object of the empty product from xxx to yyy is not valid. El objeto FixedCost de algún objeto EmptyAssignment del producto vacío que sale de xxx y llega a yyy, no tiene un valor válido. Se debe especificar un valor correcto para este objeto, que es un número decimal.
FAP61	Empty product from xxx to yyy has not defined any EmptyAssignment object. El producto vacío que sale de xxx y llega a yyy no tiene definido ningún objeto EmptyAssignment. Cada producto vacío debe tener al menos un objeto EmptyAssignment.
FAP62	Vehicle named vvvv has not defined Amount for some Limitation object. Algún objeto Limitation del vehículo vvvv, no tiene definido el campo Amount.
FAP63	Vehicle named vvvv has defined an empty value for Amount for some Limitation object. Algún objeto Limitation del vehículo vvvv, no tiene definido el valor del campo Amount.
FAP64	The value vv for Amount object, for some Limitation object of the vehicle vvvv is not valid. It must be greater than 0. El valor vv especificado en algún objeto Limitation del vehículo vvvv no es válido. Debe ser un número mayor que 0.
FAP65	The value vv for Amount object, for some Limitation object of the vehicle vvvv is not valid. It must be an integer. El valor vv especificado en algún objeto Limitation del vehículo vvvv no es válido. Debe ser un número entero.
FAP66	Vehicle named vvvv has not defined Criterion for some Limitation object.

	Algun objeto Limitation del vehículo vvvv no ha definido el objeto Criterion.
FAP67	Vehicle named vvvvv has defined an empty value for Criterion for some Limitation object. Algun objeto Limitation del vehículo vvvv no ha definido el valor del objeto Criterion.
FAP68	The value vv for Criterion object, for some Limitation object of the vehicle vvvv is not valid. It must be Maximum, Equal or Minimum. El valor del objeto Criterion de algún objeto Limitation del vehículo vvvv no es válido. Debe tener estos valores: 'Maximum', 'Equal', 'Minimim'.

5 Próximas versiones

Este manual corresponde a la versión de la resolución del FAP que publica VETU actualmente. Aplicando la política de consolidación del sistema SaaS de VETU, y para ofrecer un mejor servicio a sus clientes, Infozara está trabajando continuamente en la mejora de sus productos y servicios. Estas mejoras para el FAP vienen dadas por la inclusión de nuevos condicionantes que permitan que el problema pueda reflejar más requerimientos de negocio, que permitan que el sistema pueda aplicarse en más empresas, y abrir las posibilidades de optimización, que lleven a la reducción de costes de la operativa de los clientes de Infozara.

En este capítulo se describen las nuevas funciones que tendrán las próximas versiones del servicio SaaS que resuelve el FAP en VETU. Si desea más información sobre estas nuevas funciones, puede ponerse en contacto con el servicio técnico de Infozara.

5.1 Múltiples ventanas de tiempo

En la versión actual del FAP que resuelve VETU, cada trayecto viene definido por una ventana de tiempo en la salida, que expresa el rango de posible para la hora de salida del trayecto. El algoritmo de optimización fija la hora de salida de cada trayecto dentro de su ventana de tiempo, de forma que pueda crear las rutas de menos coste. Una opción más flexible, y que permitiría optimizar más el uso de la flota de vehículos, sería poder definir varias ventanas de tiempo en la salida del trayecto. Por ejemplo, con dos ventanas de tiempo en la salida, se podría expresar el requerimiento 'crear un trayecto que puede salir de MAD a BCN el lunes por la tarde (entre las 17:00 y las 19:00) o el martes por la tarde (ente las 18:00 y las 19:00)'. El algoritmo que resuelve el FAP, debería fijar la salida del trayecto el lunes o el martes de forma que la planificación global sea la de menor coste.

5.2 Coste del trayecto en función de su duración

En la versión actual que resuelve VETU, el coste de un trayecto es un coste fijo, que depende del tipo de vehículo que se asigna al trayecto, la duración del trayecto es variable, y du duración viene definida por una ventana de tiempo. En realidad, el coste del trayecto es función de la duración del trayecto, porque el consumo de combustible depende fuertemente de la velocidad. Si el trayecto tiene menor duración, supone mayor velocidad, y mayor consumo. En la próxima versión del FAP que resuelve VETU, se podrá especificar el coste variable en función de la duración, para que el algoritmo de optimización pueda fijar la duración de los trayectos que lleve a un menor coste global.