



Sistema SaaS para la resolución del  
problema de planificación de producción

Manual de usuario

Versión 1.0

[infozara.es](http://infozara.es)

© 2016 Infozara

## 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

La información que contiene este documento está legamente protegida y es confidencial a Infozara, sus clientes, y a quienes Infozara lo entregue expresamente con el propósito de evaluar el sistema VETU. No se puede reproducir este documento de ninguna forma mecánica ni electrónica, incluyendo archivos electrónicos, sin el consentimiento expreso de Infozara.

## ÍNDICE

1	Introducción.....	4
1.1	En este manual.....	4
1.2	SaaS.....	4
1.3	VETU.....	5
1.4	¿Quiénes somos?.....	5
1.5	Planificación de producción.....	6
1.6	Nomenclatura.....	6
1.7	Soporte.....	6
1.8	Actualización de versiones.....	7
1.9	Más información.....	7
2	Problema de planificación de producción.....	8
2.1	Modelo conceptual del problema.....	8
2.2	El problema al detalle.....	8
3	Acceso a los servicios de VETU.....	11
3.1	Arquitectura general de todo el sistema.....	11
3.2	Servicios Web.....	11
3.2.1	Servicio web de comprobación de estado de problemas.....	12
3.2.2	Servicio web de resolución del PPP.....	12
3.2.3	Características de objetos y campos.....	18
3.3	Proceso de ejecución de un PPP mediante los servicios web.....	19
3.4	Ejemplo de uso desde .NET.....	20
3.5	Ejemplo de uso desde NetBeans.....	25
3.6	Ejemplo de uso desde Eclipse.....	30
4	Listas de mensajes.....	38
4.1	Mensajes de envío de problemas.....	38
4.2	Mensajes del PPP.....	38
5	Próximas versiones.....	44
5.1	Solape de cambios de utillaje.....	44
5.2	Minimizar coste.....	44
5.2.1	Penalizaciones por retraso.....	44

## 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 planificación de producción, que calcula la secuencia de pedidos que procesa cada máquina de una fábrica, en el menor tiempo posible.

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

Es un documento dirigido tanto al gestor de recursos de la empresa, que se encarga de la planificación de producción, 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 en 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 planificación de producción**  
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 planificación de producción, 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 planificación de producción en el sistema SaaS de VETU.
- **Próxima versión**  
Se describen las nuevas funcionalidades previstas para la próxima versión del sistema de planificación de producción 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 planificación de producción, ofreciendo como SaaS el acceso a un algoritmo que minimiza el tiempo total de proceso de los pedidos de los clientes.

El acceso al servicio SaaS de planificación de producción de VETU, se proporciona mediante servicios web tal y como se describe en el capítulo 3 de 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 la planificación de producción, pero publica más, como la planificación operativa de personal. Para ver todos los problemas que publica VETU, consulte la web [www.vetu.es](http://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](http://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.spoce.es](http://www.spoce.es)), LISBB ([www.lisbb.es](http://www.lisbb.es)), QRP ([qrp.infozara.es](http://qrp.infozara.es)), AMBÚ ([ambu.infozara.es](http://ambu.infozara.es)) y SMOTY ([www.smoty.es](http://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 Planificación de producción

En este manual de usuario definimos el problema de planificación de producción (PPP), como el problema de planificar la secuencia de fabricación de pedidos de las máquinas de una fábrica para acabar en el menor tiempo posible.

Las entradas del problema son:

- Líneas de fabricación o máquinas  
Máquinas que fabrican o procesan productos. Cada máquina puede fabricar ciertos productos.
- Pedidos de los clientes  
Pedidos que realizan los clientes y debe procesar la fábrica.

El objetivo es encontrar la secuencia de pedidos que realiza cada máquina de forma que se termine toda la producción en el menor tiempo posible.

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

Si el problema de planificación de producción de su empresa difiere del aquí presentado, o requiere alguna característica no recogida en el PPP 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.  |
| PPP  | Problema de planificación de producción, introducido en apartado 1.5 y descrito en mayor detalle en el capítulo 2. |
| UT   | Unidad de tiempo. En general la unidad de tiempo será el minuto o la hora.   |

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 `SolutionPPP`.

Las direcciones web o direcciones de correo electrónico que se referencian en este documento, se muestran en color azul, como [www.vetu.es](http://www.vetu.es).

## 1.7 Soporte

Si es usted cliente de la plataforma VETU de Infozara, puede obtener soporte técnico sobre el PPP según los mecanismos que especifica su contrato con 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 PPP, 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 PPP de VETU.

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

## 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](mailto:info@vetu.es)

[ppp@vetu.es](mailto:ppp@vetu.es)

También puede consultar las siguientes direcciones web:

[www.infozara.es](http://www.infozara.es)

[www.vetu.es](http://www.vetu.es)

[www.vetu.es/webvetu/saas.do](http://www.vetu.es/webvetu/saas.do)

[www.vetu.es/webvetu/planificacionProduccion.do](http://www.vetu.es/webvetu/planificacionProduccion.do)

## 2 Problema de planificación de producción

### 2.1 Modelo conceptual del problema

La Figura 2.1 muestra el modelo conceptual del PPP. El diagrama se lee de la siguiente forma:

‘El problema de planificación de producción (PPP) viene definido por una serie de líneas de producción (o máquinas) y una serie de pedidos.

Cada línea de producción puede realizar una serie de productos, a una tasa de producción por unidad de tiempo. Cada vez que una línea cambia de productos, es necesario un cambio de utillaje, que se mide en unidades de tiempo. Cada línea de producción define un tiempo mínimo de fabricación para cada producto

Los pedidos vienen definidos por un identificador, una demanda de un producto y una fecha máxima de entrega, que es opcional.

El problema consiste en calcular la secuencia de pedidos que procesa cada máquina, de forma que se fabriquen todos los pedidos de productos en el menor tiempo posible.’

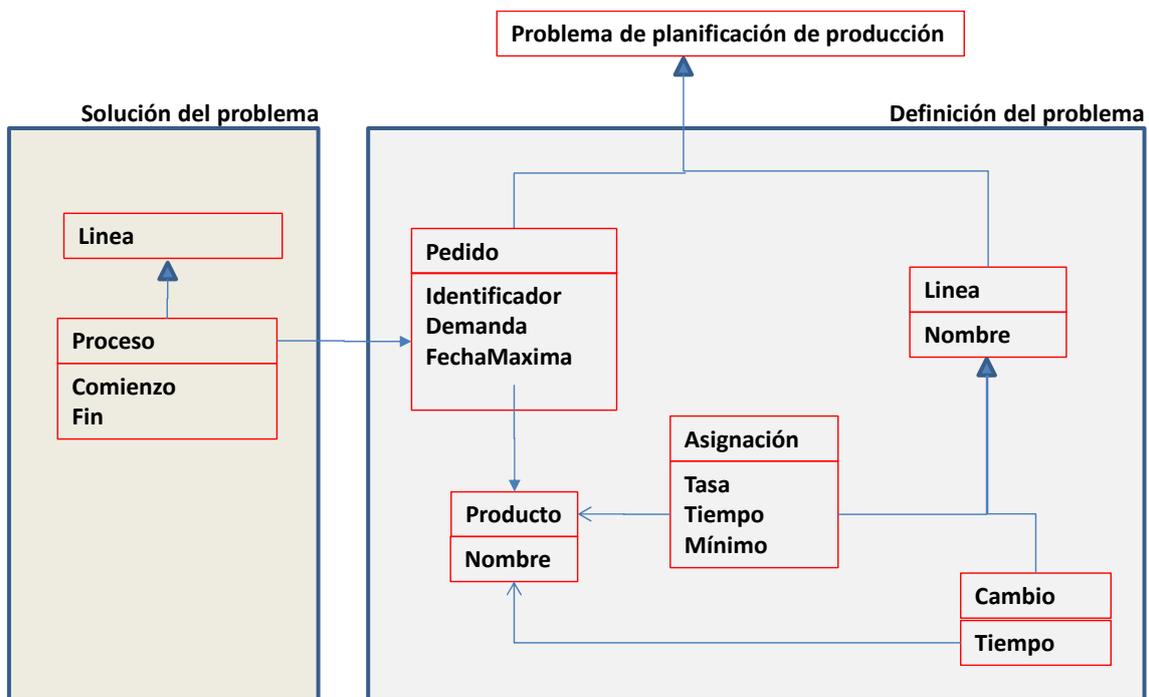


Figura 2-1. Modelo conceptual del PPP

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

### 2.2 El problema al detalle

El objetivo que busca el PPP es calcular la secuencia de pedidos que procesa cada máquina de la fábrica. La secuencia de pedidos es una lista ordenada en el tiempo de los productos que fabrica cada línea. Los productos que fabrica cada máquina vienen definidos por los pedidos de los clientes, es decir cada pedido de cliente significa la fabricación de un producto. Cada asignación de un producto a una máquina se define por un comienzo y un final. Un pedido se

puede fabricar en varias máquinas. Si un pedido se fabrica en varias máquinas, y tiene una fecha máxima de entrega, la fabricación del producto en cada máquina debe cumplir con la condición de la fecha máxima de entrega del pedido. La Tabla 2-1 muestra los parámetros que definen un producto que se fabrica en la fábrica, y la Tabla 2-2 muestra los parámetros que definen un pedido de un producto, que se va a fabricar.

Parámetro	Descripción	Tipo
Nombre	Definición única del producto	Texto

**Tabla 2-1. Definición de los parámetros de un producto**

Parámetro	Descripción	Tipo
Identificador	Identificador único del pedido	Texto
Producto	Identificador del producto que contiene este pedido	Texto
Demanda	Cantidad de productos que se deben fabricar para este pedido	Double
Fecha entrega	Fecha de entrega opcional del pedido, expresado en días	Entero

**Tabla 2-2. Definición de los parámetros de un pedido**

Cada máquina puede fabricar unos ciertos tipos de productos. En general, hay varias máquinas que pueden fabricar un cierto producto, y un producto se puede fabricar en varias máquinas, pero no en todas. Cada máquina que puede fabricar o procesar un cierto producto, lo hace a una tasa de fabricación diferente. La tasa de fabricación de una máquina es la cantidad de productos que procesa por unidad de tiempo. Cuando una máquina termina de procesar un producto, es habitual que se deban realizar cambios de utillaje o ajustes, antes de comenzar a procesar el siguiente producto. Estos cambios de utillaje o ajustes consumen una cierta cantidad de tiempo en el que la máquina no está fabricando. Estos cambios de utillaje son distintos en cada máquina, y habitualmente son distintos para cada secuencia de productos. La Tabla 2-3 muestra la definición de los parámetros de una línea.

Parámetro	Descripción	Tipo
Nombre	Identificador único de la línea	Texto
Asignaciones	Lista de asignaciones de productos que fabrica la línea	Lista
Asignación	Características de fabricación del producto en la línea	Objeto
Producto	Identificador del producto	Texto
Tiempo Mínimo	Tiempo mínimo de fabricación de este producto en la línea. Se expresa en horas.	Double
Tasa Fabricación	Cantidad de productos por hora que se fabrican por minuto en esta línea	Double
Cambios Utillaje	Lista de cambios de utillaje al cambiar entre productos seguidos que se fabrican en la línea	Lista
Cambio Utillaje	Características del cambio de utillaje	Objeto
Producto Anterior	Identificador del producto que se fabrica en la línea antes de hacer el cambio de utillaje	Texto
Producto Posterior	Identificador del producto que se fabrica en la línea después de hacer el cambio de utillaje.	Texto
Tiempo	Tiempo de cambio de utillaje en la línea cuando se pasa de fabricar Producto Anterior a fabricar Producto Posterior. Se expresa en horas.	Double

**Tabla 2-3. Definición de los parámetros de una línea.**

En función de los pedidos que se asignen a cada máquina, y de cómo se ordenen en el tiempo

los pedidos de cada máquina, se obtiene un tiempo diferente de fabricación en cada máquina, y un tiempo diferente de fabricación global, que es el mayor tiempo de fabricación de todas las máquinas. El objetivo del PPP es encontrar la planificación de producción que minimiza el tiempo de fabricación global.

Las opciones de optimización que presenta este problema de planificación de producción son:

- Asignación de un pedido a varias máquinas.
- Tiempo de fabricación de un producto en cada máquina.
- Cambios de utillaje.
- Secuencia de fabricación de pedidos en cada máquina.

La definición de un producto es muy sencilla, ya que viene dada únicamente por un nombre o identificador único. Las opciones de optimización relativas a un producto vienen dadas por su relación con las líneas de producción.

La definición de una máquina o línea de producción es muy sencilla, ya que viene dada únicamente por un nombre o identificador único. Las opciones de optimización de una línea de producción vienen dadas por su relación con la fabricación de productos.

Si una determinada máquina puede fabricar un cierto producto, es necesario conocer la tasa de fabricación de ese producto en la máquina, que es la cantidad de productos que fabrica la máquina por unidad de tiempo. Por ejemplo, si la máquina FJ234 fabrica el producto P001 a una tasa de 20 unidades/hora, significa que cada hora que esté funcionando la máquina FJ234, fabrica 20 unidades de producto P001. Si un pedido de un cliente tiene una demanda de 600 unidades de producto P001, la máquina FJ234 tardará  $600/20 = 30$  horas en fabricar los productos P001 correspondientes a ese pedido. El tiempo mínimo de fabricación de un producto en una línea significa que no se puede asignar a esa línea, la fabricación de un pedido o la parte de un pedido, cuya duración sea menor que ese tiempo mínimo.

Si una determinada máquina puede realizar dos productos, es necesario especificar el cambio de utillaje o de preparación que es necesario realizar en la máquina cada vez que cambia el producto. Por ejemplo si la máquina FJ234 puede fabricar los productos P002 y P007, se debe conocer la duración del tiempo de preparación de la máquina FJ234 cuando deja de fabricarse el producto P002 y va a fabricarse inmediatamente después el producto P007, y también se debe conocer el tiempo de preparación de la máquina FJ234 cuando deja de fabricarse el producto P007 y va a fabricarse inmediatamente después el producto P002.

El problema de planificación de producción que publica Vetu no requiere especificar las unidades de tiempo de los diferentes conceptos que se miden como unidades temporales. Es responsabilidad del cliente del sistema mantener homogeneidad en las todas las magnitudes temporales del problema. Por ejemplo, si las tasas de fabricación de los productos en las líneas se expresan por horas, las fechas máximas de entrega de los pedidos también se deben especificar por horas, y el resultado del tiempo de fabricación debe entenderse que viene expresado en horas, y los comienzo y fin del proceso de cada pedido en cada máquina también se debe interpretar en horas.

### 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 planificación de producción (PPP).

#### 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 determinar la secuencia de pedidos que procesa cada máquina de sus fábricas. 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.

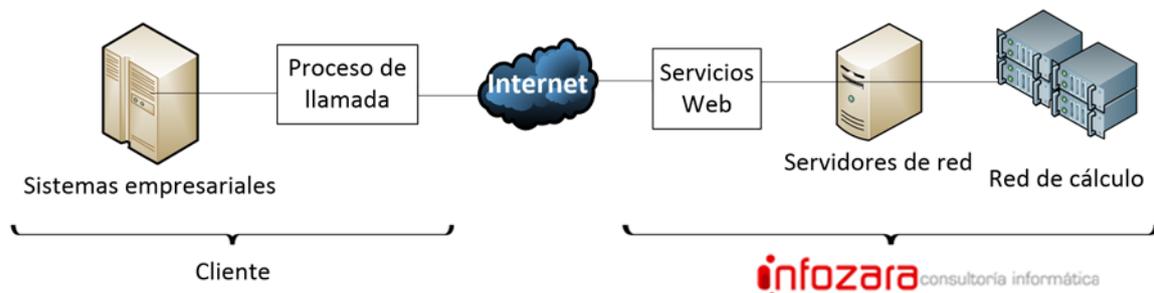


Figura 3-1. Arquitectura del sistema SaaS

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 de producción, 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 PPP no se resuelve de forma instantánea, sino que puede requerir de varios minutos, dado que es un problema complejo de optimización combinatoria. 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 PPP se llama `wsPPP` 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( int 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 PPP

El servicio web que permite la resolución del PPP se llama `wsPPP`, y tiene dos funciones

```
Request solve( PPP_Problem oProblem )
```

```
PPP_Solution getSolution( int nTicket )
```

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

<http://www.vetu.es/wsPPP/wsPPP?wsdl>

La función `solve()` recibe la clase `PPP_Problem`, que encapsula la instancia del PPP a resolver, y devuelve una 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 un 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 **PPP\_Problem** encapsula la información del problema que se quiere resolver. La Figura 3-4 muestra la versión XML de la definición de un problema PPP. El objeto raíz **PPP\_Problem** se compone de tres objetos: **Products**, que define los productos que se fabrican, **Lines**, que es una lista de objetos **Line**, que definen las máquinas o líneas de fabricación, y **Orders**, que es la lista de todos los pedidos que se deben fabricar.

```
<PPP_Problem>
  <Products>
    <Product>
      <Name>PC01</Name>
    </Product>
    <Product>
      <Name>PC23</Name>
    </Product>
    <Product>
      <Name>QZ145</Name>
    </Product>
    <Product>
      <Name>QZ457</Name>
    </Product>
  </Products>
  <Lines>
    <Line>
      <Name>FJ2106</Name>
      <Assignments>
        <Assignment>
          <Product>PC01</Product>
          <Rate>10</Rate>
          <MinTime>2</MinTime>
        </Assignment>
        <Assignment>
          <Product>PC23</Product>
          <Rate>12</Rate>
          <MinTime>2</MinTime>
        </Assignment>
      </Assignments>
    </Line>
  </Lines>
</PPP_Problem>
```

```

        </Assignment>
    </Assignments>
    <ChangeOvers>
        <ChangeOver>
            <ProductFrom>PC01</ProductFrom>
            <ProductTo>PC23</ProductTo>
            <Time>3</Time>
        </ChangeOver>
        <ChangeOver>
            <ProductFrom>PC23</ProductFrom>
            <ProductTo>PC01</ProductTo>
            <Time>3</Time>
        </ChangeOver>
    </ChangeOvers>
</Line>
<Line>
    <Name>JA1107</Name>
    <Assignments>
        <Assignment>
            <Product>QZ145</Product>
            <Rate>12</Rate>
            <MinTime>2</MinTime>
        </Assignment>
        <Assignment>
            <Product>QZ457</Product>
            <Rate>6</Rate>
            <MinTime>2</MinTime>
        </Assignment>
    </Assignments>
    <ChangeOvers>
        <ChangeOver>
            <ProductFrom>QZ145</ProductFrom>
            <ProductTo>QZ457</ProductTo>
            <Time>3</Time>
        </ChangeOver>
        <ChangeOver>
            <ProductFrom>QZ457</ProductFrom>
            <ProductTo>QZ145</ProductTo>
            <Time>4</Time>
        </ChangeOver>
    </ChangeOvers>
</Line>
<Line>
    <Name>GU0401</Name>
    <Assignments>
        <Assignment>
            <Product>PC23</Product>
            <Rate>9</Rate>
            <MinTime>2</MinTime>
        </Assignment>
        <Assignment>
            <Product>QZ457</Product>
            <Rate>8</Rate>
            <MinTime>2</MinTime>
        </Assignment>
    </Assignments>
    <ChangeOvers>
        <ChangeOver>
            <ProductFrom>PC23</ProductFrom>
            <ProductTo>QZ457</ProductTo>
    
```

```

        <Time>3</Time>
    </ChangeOver>
    <ChangeOver>
        <ProductFrom>QZ457</ProductFrom>
        <ProductTo>PC23</ProductTo>
        <Time>4</Time>
    </ChangeOver>
</ChangeOvers>
</Line>
</Lines>
<Orders>
    <Order>
        <Identifier>OP003</Identifier>
        <Product>PC01</Product>
        <Demand>5000</Demand>
        <LimitDate/>
    </Order>
    <Order>
        <Identifier>OP103</Identifier>
        <Product>PC01</Product>
        <Demand>2500</Demand>
        <LimitDate/>
    </Order>
    <Order>
        <Identifier>OP403</Identifier>
        <Product>PC23</Product>
        <Demand>4500</Demand>
        <LimitDate/>
    </Order>
    <Order>
        <Identifier>OP309</Identifier>
        <Product>QZ145</Product>
        <Demand>3000</Demand>
        <LimitDate/>
    </Order>
    <Order>
        <Identifier>OP634</Identifier>
        <Product>QZ457</Product>
        <Demand>3500</Demand>
        <LimitDate/>
    </Order>
</Orders>
</PPP_Problem>
    
```

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

El objeto **Product** contiene la definición de un producto que se procesa o fabrica, y viene dada únicamente por el siguiente campo:

- Name: Nombre o referencia del producto.

En el ejemplo de problema PPP que muestra la Figura 3-4, se definen cuatro productos, de nombres PC01, PC23, QA145 y QZ457.

El objeto **Line** define una máquina o línea de proceso o línea de fabricación. La definición de una línea se hace por tres campos principales:

- Name: Nombre o identificador único de la línea.
- Assignments: Define las características de asignación de un producto a una línea.
- ChangeOvers: Lista de objetos ChangeOver, que definen los cambios de utillaje.

Cada **Assignment** define que un producto se puede fabricar en la línea. Los campos que define

una asignación de un producto a una línea son los siguientes:

- **Product:** Nombre del producto que se fabrica en la línea.
- **Rate:** Tasa de fabricación del producto en la línea. Se expresa como cantidad de producto que se procesa o fabrica por unidad de tiempo.
- **MinTime:** Tiempo mínimo que se debe asignar este producto a esta línea. No están permitidos tiempos de proceso menores que esta cantidad. Se expresa en unidades de tiempo.

Cada **ChangeOver** define el cambio de utillaje o de preparación de una línea cuando cambia el producto que se fabrica. Este cambio se define por los siguientes cambios:

- **ProductFrom:** Producto que está fabricando actualmente la línea.
- **ProductTo:** Producto que va a fabricar la línea después de ProductFrom.
- **Time:** Tiempo necesario para adaptar la línea antes de que pueda comenzar a procesar el producto ProductTo, después de haber estado procesando el producto ProductFrom

En el ejemplo de problema PPP que muestra la Figura 3-4, se definen tres líneas, de nombres FJ2106, JA1107 y GU401. La línea FJ2106 puede fabricar el producto PC01 a una tasa de 10 unidades de producto por unidad de tiempo, y el producto PC23 a una tasa de 12 unidades por unidad de tiempo. El tiempo mínimo de fabricación de pedidos de estos productos es de 2 UT en la línea FJ2106. La tercera línea, de nombre GU0401 fabrica los productos PC23 y QZ457, y los tiempos de preparación o cambio de utillaje son de 3 UT, cuando se pasa de fabricar el producto PC23 a fabricar el producto QZ457, y 4 UT cuando se pasa de fabricar el producto QZ457 a fabricar el producto PC23.

El objeto **Orders** es una lista de objetos **Order**, que define un pedido de un cliente sobre un producto. El objeto **Order** se compone de los siguientes campos:

- **Identifier:** Identificador único del pedido.
- **Product:** Nombre del producto que se demanda en el pedido.
- **Demand:** Cantidad de producto que se demanda en el pedido.
- **LimitDate:** Fecha máxima de entrega del pedido, expresada en UT. Esta información es opcional. Si un pedido tiene fecha máxima de entrega, el proceso de asignación lo debe planificar de forma que el total de la fabricación del pedido esté realizada antes o en la fecha máxima.

A continuación se va a mostrar el formato de la salida del servicio web que resuelve el PPP en VETU. La Figura 3-5 muestra un ejemplo en XML, de la solución del PPP que genera el proceso de resolución, que corresponde al problema definido en la Figura 3-4. El objeto raíz que encapsula toda la información de la solución es **PPP\_Solution**, que se compone de tres objetos: **Result**, que indica si el problema se ha resuelto correctamente, **ObjectiveValue**, que indica el valor de tiempo de planificación, y **LineAllocations**, que muestra la secuencia de fabricación de los pedidos en las líneas.

```

<PPP_Solution>
  <Result>
    <Status>Ok</Status>
    <Message Code ="PPP0">The problem has been successfully solved</Message>
  </Result>
  <ObjectiveValue>
    <Value>750</Value>
  </ObjectiveValue>
  <LineAllocations>
    <AllocatedLine>
      <Name>FJ2106</Name>
    </AllocatedLine>
  </LineAllocations>
</PPP_Solution>
    
```

```

<Allocations>
  <Allocation>
    <Order>OP003</Order>
    <Begin>0</Begin>
    <End>500</End>
  </Allocation>
  <Allocation>
    <Order>OP103</Order>
    <Begin>500</Begin>
    <End>750</End>
  </Allocation>
</Allocations>
</AllocatedLine>
<AllocatedLine>
  <Name>JA1107</Name>
  <Allocations>
    <Allocation>
      <Order>OP309</Order>
      <Begin>0</Begin>
      <End>250</End>
    </Allocation>
    <Allocation>
      <Order>OP634</Order>
      <Begin>250</Begin>
      <End>750</End>
    </Allocation>
  </Allocations>
</AllocatedLine>
<AllocatedLine>
  <Name>GU0401</Name>
  <Allocations>
    <Allocation>
      <Order>OP403</Order>
      <Begin>0</Begin>
      <End>500</End>
    </Allocation>
    <Allocation>
      <Order>OP634</Order>
      <Begin>500</Begin>
      <End>562.5</End>
    </Allocation>
  </Allocations>
</AllocatedLine>
</LineAllocations>
</PPP_Solution>
    
```

Figura 3-5. Ejemplo de solución del PPP

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 **PPP0**. 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 Figura 3-6 muestra un ejemplo de solución con errores, donde el campo **Status** tiene el valor **Failed**, hay dos mensajes de error, y los objetos **ObjectiveValue** y **LineAllocations** están vacíos.

Un mensaje específico de error a tener en cuenta es el mensaje 'PPP3: The problem has no feasible solution. Please check problem data'. Este mensaje significa que el problema que se ha planteado no tiene solución. Esta situación puede darse si la fechas máximas de entrega de los pedidos no se puede satisfacer.

```

<PPP_Solution>
  <Result>
    <Status>Failed</Status>
    <Message Code ="PPP6">Data entry has not defined Orders object.</Message>
    <Message Code ="PPP16">Order SD334504 has not defined Demand
      object</Message>
  </Result>
  <ObjectiveValue/>
  <Allocations/>
</PPP_Solution>
    
```

Figura 3-6. Ejemplo de solución del PPP con errores

El objeto **LineAllocations** muestra la secuencia de pedidos que realiza cada línea, es decir, es el plan de producción. El objeto **LineAllocations** es una lista de objetos **AllocatedLine**, que describen la secuencia de trabajo ordenada en el tiempo de cada línea. Este objeto **AllocatedLine** se compone de los objetos **Name**, que especifica el nombre de la línea, y **Allocations**, que es una lista de objetos **Allocation**. Este objeto **Allocation** representa un proceso de fabricación de un pedido en la línea, y se compone de los siguientes campos:

- Order: Identificador del pedido que fabrica la línea.
- Begin: Comienzo del proceso de fabricación del pedido en la línea.
- End: Fin del proceso de fabricación del pedido en la línea.

Como ejemplo, en la solución del PPP que se muestra en la Figura 3-5, la línea FJ2106 se fabrica el pedido OP003 desde el instante 0 al 500, y se fabrica el pedido OP003 desde el instante 500 al 750. Cabe destacar que un pedido se puede asignar a varias máquinas, como ocurre por ejemplo con el pedido OP634, que en solución de la Figura 3-5, se asigna a las líneas JA1107 y GU0401.

### 3.2.3 Características de objetos y campos

La Tabla 3-2 muestra las características de cada uno de los objetos y campos del problema PPP (objeto **PPP\_Problem**).

Objeto o campo	Requerido	Vacío	Cardinalidad
Products	SI	NO	1
Product	SI	NO	1 o más
Name	SI	NO	1
Lines	SI	NO	1
Line	SI	NO	1 o más
Name	SI	NO	1
Assignments	SI	NO	1
Assignment	SI	NO	1 o más
Product	SI	NO	1
Rate	SI	NO	1
MinTime	SI	NO	1
ChangeOvers	SI	NO	1
ChangeOver	SI	SI	0 o más

ProductFrom	SI	NO	1
ProductTo	SI	NO	1
Time	SI	NO	1
Orders	SI	NO	1
Order	SI	NO	1 o más
Identifier	SI	NO	1
Product	SI	NO	1
Demand	SI	NO	1
LimitDate	SI	SI	1

**Tabla 3-2. Características de objetos y campos del problema PPP**

La Tabla 3-3 muestra las características de cada uno de los objetos y campos de la solución del problema PPP (objeto **PPP\_Solution**).

Objeto o campo	Requerido	Vacío	Cardinalidad
Result	SI	NO	1
Status	SI	NO	1
Message	SI	NO	1 o más
ObjectiveValue	SI	NO	1
Value	SI	NO	1
LineAllocations	SI	NO	1
AllocatedLine	SI	NO	1 o más
Name	SI	NO	1
Allocations	SI	SI	1
Allocation	SI	NO	0 o más
Order	SI	NO	1
Begin	SI	NO	1
End	SI	NO	1

**Tabla 3-3. Características de objetos y campos de la solución del PPP**

### 3.3 Proceso de ejecución de un PPP 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 PPP, el pseudocódigo de un programa que resuelve el PPP 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.

**PPP\_Problem** oEntrada  
 Definir productos en oEntrada  
 Definir líneas en oEntrada

```

Definir pedidos en oEntrada
Request oResultado = wsPPP.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
            PPP_Solution oSalida = wsPPP.getSolution( oResultado.Ticket )
            If oSalida.Result.Status = Ok
                Almacenar oSalida en sistemas empresariales
            Else If oSalida.Result.Status = Failed
                Mostrar errores al usuario
            End If
        Caso strEstado = UNKNOWN
            Mostrar 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 3-7. Pseudocódigo de ejecución del PPP mediante los servicios web.

### 3.4 Ejemplo de uso desde .NET

En este apartado, se muestra un programa de ejemplo para resolver una instancia del PPP, mediante el sistema SaaS de VETU, escrito en C# (Microsoft .NET <sup>1</sup>(1)) de Microsoft, bajo Visual Studio 2010<sup>2</sup>.

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 CAP de VETU, se deben seguir los siguientes pasos<sup>3</sup>:

<sup>1</sup> Microsoft .NET es una marca registrada por Microsoft Corporation.

<sup>2</sup> Visual Studio .NET es una marca registrada por Microsoft Corporation.

<sup>3</sup> Esta secuencia de acciones podría cambiar según la versión de Visual Studio.

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/wsPPP/wsPPP?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/wsPPP/wsPPP?wsdl>
8. Pulsar el botón 'Add Reference'

En este momento se han creado una serie de clases que permiten acceder al servicio web **wsPPP**, 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:

- **WsPPPService**: Es la clase de acceso al servicio web. Tiene las funciones **solve** y **getSolution**.
- **PPP** Es la clase que encapsula la definición de la instancia de problema que se quiere resolver. Contiene a las clases **Products**, **Lines** y **Orders**, que contienen la definición de los tipos de los productos que se fabrica en la fábrica, las definiciones de las líneas de fabricación y los pedidos, respectivamente.
- **PPPSolution**: Es la clase que encapsula la solución del problema que se ha resuelto. Contiene a las clases **result**, **objectiveValue** y **allocatedLines**, que tiene la información de la resolución del problema, de los valores encontrados y las asignaciones de productos, 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 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 PPP mediante los servicios web de VETU se muestra en la Figura 3-8. 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 94 se definen los datos del problema a resolver, y desde la línea 97 hasta el final, es el acceso a los servicios web.

En la línea 2 se crea la instancia de la clase **pppProblem**, que encapsula toda la información del problema, que son los productos, las líneas y los pedidos.

En la línea 5 se crean los productos que fabrica la empresa, que es un array de 3 productos. En las líneas 7, 10 y 13 se crean cada uno de los tres productos. El producto se define por la clase **product**, y el único parámetro del objeto **product** es el nombre del producto. En las líneas 8, 11 y 14 se fija el nombre de cada producto.

En la línea 17 se crea un array de 2 líneas de fabricación. Las líneas se define con el objeto **line**. La primera línea de producción se crea en la línea de código fuente número 20, y la segunda se crea en la línea de código fuente número 48. El objeto **line** tiene tres objetos principales: **Name**, **Assignments** y **ChangeOvers**, que contiene el nombre, los productos que puede procesar la

línea, y los cambios de utillaje por cambios de producto en la línea, respectivamente. Los nombres de las líneas se fijan en las líneas de código fuente número 21 y 49. El objeto **Assignments** es un array de objetos **Assignment**, que contiene los parámetros de trabajo de la línea para fabricar un producto. En la línea de código fuente número 23 se indica que la línea "A04" puede fabricar dichos productos. Entre las líneas de código fuente 25 y 28 se indica que la línea "A04" puede fabricar el producto "P001" (línea de código fuente 32), a una tasa de 20 unidades de producto por hora (línea de código fuente 28), y que el tiempo mínimo de fabricación de este producto en esta línea es de 4 horas (línea de código fuente 26). De la misma forma entre las líneas 30 y 33 del código fuente, se indican los parámetros de funcionamiento del proceso de del producto "P104" en la línea "A04". El objeto **ChangeOvers** es un array de objetos **ChangeOver**, que define el tiempo de cambio de utillaje al cambiar el producto de fabricación en la línea, después de fabricar anteriormente otro producto. Como ejemplo, en la línea de código fuente número 37 se crea un objeto **ChangeOver**, que indica que si en la línea "A04" se está fabricando el producto "P001" (línea de código fuente 38), y después se va a fabricar el producto "P104" (línea de código fuente 39), serán necesarias 4 horas para realizar el cambio de utillaje (línea de código fuente 40). De la misma manera, entre las líneas de código fuente 42 y 45, se definen el segundo cambio de utillaje de la línea "A04", entre los productos "P104" y "P101".

En la línea 76 se crea un array de 3 pedidos. Un pedido se define con el objeto **order**. Como ejemplo, en la línea de código fuente número 78 se crea una orden para fabricar 1000 unidades (línea de código fuente 81), del producto "P001". El identificador del pedido es "Ord002" (línea de código fuente 79), y no tiene fecha de entrega (línea de código fuente 82). De la misma forma, entre las líneas de código fuente 84 y 88 se define el segundo pedido, y entre las líneas de código fuente 90 y 94, se define el tercer pedido.

Una vez definido el problema, se debe enviar al servicio web: en la línea 97 se crea el acceso al servicio web, y en las líneas 100 y 101 se envía el problema al servicio web. 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 104 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 111 y 112 se crea el acceso al servicio web que informa del estado de resolución del problema. Desde la línea 116 a la 129 se define un bucle con un timer que espera a que se resuelva el problema. Desde las líneas 131 a la 153, se comprueba el estado final de resolución. Si el estado es PROCESSED, se accede a la solución del problema (líneas 134 y 135), que viene encapsulada en el objeto **pppSolution**.

```

1 // Crea la instancia del problema
2 wsPPReference.pppProblem oProblem = new wsPPReference.pppProblem();
3
4 // Crea los productos que se fabrican
5 oProblem.Products = new wsPPReference.product[3];
6
7 oProblem.Products[0] = new wsPPReference.product();
8 oProblem.Products[0].Name = "P001";
9
10 oProblem.Products[1] = new wsPPReference.product();
11 oProblem.Products[1].Name = "P104";
12
13 oProblem.Products[2] = new wsPPReference.product();
14 oProblem.Products[2].Name = "P267";
15
16 // Crea dos líneas de producción
17 oProblem.Lines = new wsPPReference.line[2];
18
19 // Crea la primera línea y la define
20 oProblem.Lines[0] = new wsPPReference.line();
    
```

```

21 oProblem.Lines[0].Name = "A04";
22
23 oProblem.Lines[0].Assignments = new wsPPReference.assignment[2];
24
25 oProblem.Lines[0].Assignments[0] = new wsPPReference.assignment();
26 oProblem.Lines[0].Assignments[0].MinTime = 4;
27 oProblem.Lines[0].Assignments[0].Product = "P001";
28 oProblem.Lines[0].Assignments[0].Rate = 20;
29
30 oProblem.Lines[0].Assignments[1] = new wsPPReference.assignment();
31 oProblem.Lines[0].Assignments[1].MinTime = 4;
32 oProblem.Lines[0].Assignments[1].Product = "P104";
33 oProblem.Lines[0].Assignments[1].Rate = 10;
34
35 oProblem.Lines[0].ChangeOvers = new wsPPReference.changeOver[2];
36
37 oProblem.Lines[0].ChangeOvers[0] = new wsPPReference.changeOver();
38 oProblem.Lines[0].ChangeOvers[0].ProductFrom = "P001";
39 oProblem.Lines[0].ChangeOvers[0].ProductTo = "P104";
40 oProblem.Lines[0].ChangeOvers[0].Time = 4;
41
42 oProblem.Lines[0].ChangeOvers[1] = new wsPPReference.changeOver();
43 oProblem.Lines[0].ChangeOvers[1].ProductFrom = "P104";
44 oProblem.Lines[0].ChangeOvers[1].ProductTo = "P001";
45 oProblem.Lines[0].ChangeOvers[1].Time = 2;
46
47 // Crea la segunda línea y la define
48 oProblem.Lines[1] = new wsPPReference.line();
49 oProblem.Lines[1].Name = "B07";
50
51 oProblem.Lines[1].Assignments = new wsPPReference.assignment[2];
52
53 oProblem.Lines[1].Assignments[0] = new wsPPReference.assignment();
54 oProblem.Lines[1].Assignments[0].MinTime = 4;
55 oProblem.Lines[1].Assignments[0].Product = "P001";
56 oProblem.Lines[1].Assignments[0].Rate = 25;
57
58 oProblem.Lines[1].Assignments[1] = new wsPPReference.assignment();
59 oProblem.Lines[1].Assignments[1].MinTime = 4;
60 oProblem.Lines[1].Assignments[1].Product = "P267";
61 oProblem.Lines[1].Assignments[1].Rate = 12;
62
63 oProblem.Lines[1].ChangeOvers = new wsPPReference.changeOver[2];
64
65 oProblem.Lines[1].ChangeOvers[0] = new wsPPReference.changeOver();
66 oProblem.Lines[1].ChangeOvers[0].ProductFrom = "P001";
67 oProblem.Lines[1].ChangeOvers[0].ProductTo = "P267";
68 oProblem.Lines[1].ChangeOvers[0].Time = 6;
69
70 oProblem.Lines[1].ChangeOvers[1] = new wsPPReference.changeOver();
71 oProblem.Lines[1].ChangeOvers[1].ProductFrom = "P267";
72 oProblem.Lines[1].ChangeOvers[1].ProductTo = "P001";
73 oProblem.Lines[1].ChangeOvers[1].Time = 4;
74
75 // Crea los pedidos
76 oProblem.Orders = new wsPPReference.order[3];
77
78 oProblem.Orders[0] = new wsPPReference.order();
79 oProblem.Orders[0].Identifier = "Ord001";
80 oProblem.Orders[0].Product = "P001";
    
```

```

81  oProblem.Orders[0].Demand      = 1000;
82  oProblem.Orders[0].LimitDate   = "";
83
84  oProblem.Orders[1] = new wsPPReference.order();
85  oProblem.Orders[1].Identifier   = "Ord002";
86  oProblem.Orders[1].Product     = "P267";
87  oProblem.Orders[1].Demand      = 1200;
88  oProblem.Orders[1].LimitDate   = "";
89
90  oProblem.Orders[2] = new wsPPReference.order();
91  oProblem.Orders[2].Identifier   = "Ord003";
92  oProblem.Orders[2].Product     = "P104";
93  oProblem.Orders[2].Demand      = 1600;
94  oProblem.Orders[2].LimitDate   = "";
95
96  // Crea el acceso al servicio web
97  wsPPReference.wsPPService oCliente = new wsPPReference.wsPPService();
98
99  // Lanza la petición de resolución
100 wsPPReference.request oResultado = oCliente.solve("usuario", "password",
101  oProblem);
102
103 // Comprueba el resultado
104 if (oResultado.Status == "Failed")
105 {
106     MessageBox.Show(oResultado.Message[0].Value, "Error en la llamada: ");
107 }
108 else if (oResultado.Status == "Ok")
109 {
110     // Espera a que se ejecute el problema
111     wsProgressReference.wsProgressService oProgressServiceClient =
112         new wsProgressReference.wsProgressService();
113
114     string strEstado = "";
115     bool bTerminado = false;
116     while (!bTerminado)
117     {
118         // Comprueba el estado
119         strEstado = oProgressServiceClient.checkStatus("usuario", "password",
120             oResultado.Ticket);
121
122         if (strEstado == "UNKNOWN" || strEstado == "ERROR" ||
123             strEstado == "PROCESSED")
124             break;
125         else if (strEstado == "ALLOCATED" || strEstado == "IN_PROCESS")
126             System.Threading.Thread.Sleep(1000);
127         else // Estado desconocido
128             break;
129     }
130
131     if (strEstado == "PROCESSED")
132     {
133         // Lee la solución
134         wsPPReference.pppSolution oSolucion =
135             oCliente.getSolution("usuario", "password", oResultado.Ticket);
136
137         // Tratar la solución
138     }
139     else if (strEstado == "UNKNOWN")
140     {

```

```

141     MessageBox.Show("El ticket de acceso a la solución no es válido.
142     Contacte con al administrador");
143     }
144     else if (strEstado == "ERROR")
145     {
146         MessageBox.Show("Se han producido errores en la planificación");
147     }
148     else
149     {
150         MessageBox.Show("Se ha producido un error desconocido.
151         Contacte con al administrador");
152     }
153 }
154 }
    
```

Figura 3-8. Código fuente C# para la ejecución del CAP de VETU

### 3.5 Ejemplo de uso desde NetBeans

En este apartado, se muestra un programa de ejemplo para resolver una instancia del PPP, 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 PPP 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".

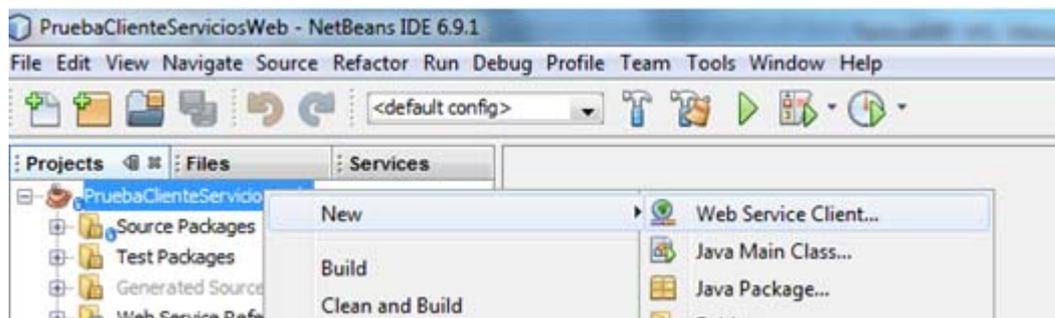


Figura 3-7. Paso 1 - Creación de cliente del servicio web NebBeans

Aparecerá una ventana emergente.

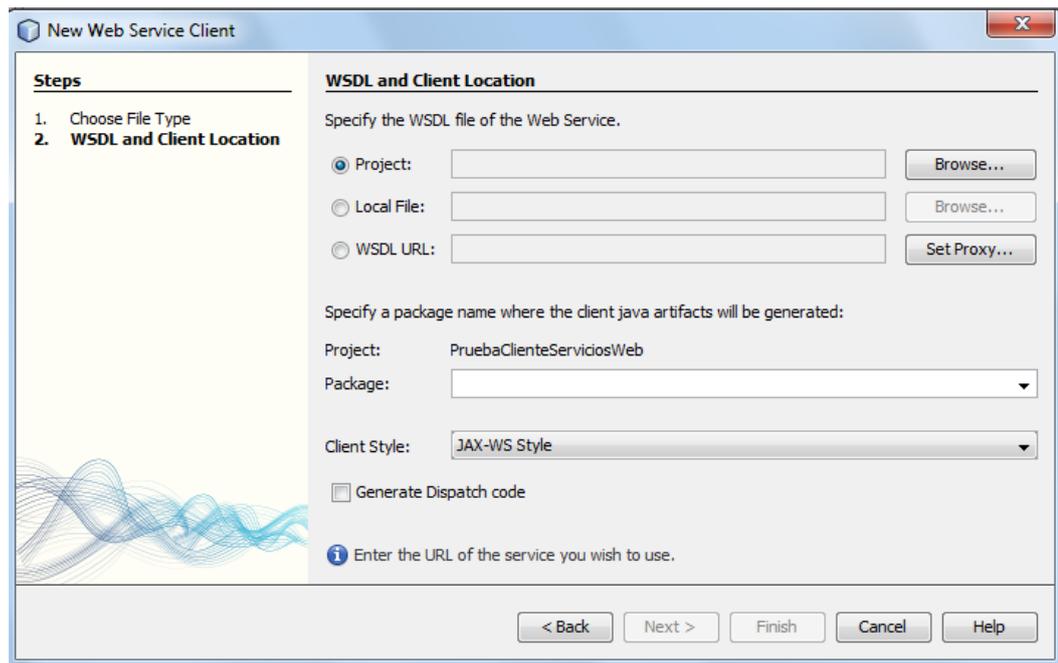


Figura 3-8. Paso 1 - Creación de cliente del servicio web NetBeans (2)

2. Marcar la opción "WSDL URL", escribir <http://www.vetu.es/wsPPP/wsPPP?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 **wsPPP**, 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:

- **WsPPPService**: Es la clase de acceso al servicio web. Tiene las funciones **solve** y **getSolution**.
- **PppProblem** Es la clase que encapsula la definición de la instancia de problema que se quiere resolver. Contiene a las clases **Products**, **Lines** y **Orders**, que contienen la definición de los tipos de los productos que se fabrica en la fábrica, las definiciones de las líneas de fabricación y los pedidos, respectivamente.
- **PppSolution**: Es la clase que encapsula la solución del problema que se ha resuelto. Contiene a las clases **Result**, **ObjectiveValue** y **LineAllocations**, que tiene la información de la resolución del problema, de los valores encontrados y las asignaciones de productos, 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 **WsProgressService**, que es el acceso al servicio web, y tiene la función **checkStatus**.

El código fuente necesario para la resolución del PPP mediante los servicios web de VETU se muestra en la Figura 3-8. 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 4 a la 117 se definen los datos del problema a resolver, y desde la línea 120 hasta el final, se implementa el acceso a los servicios web.

En la línea 2 se crea la instancia de la clase **PppProblem**, que encapsula toda la información del problema, que son los productos, las líneas y los pedidos.

En la línea 5 se inicia la creación de los productos que fabrica la empresa, para ello se instancia la clase **Products** que contiene una lista de productos. En las líneas 8, 12 y 16 se crean cada uno de los tres productos. El producto se define por la clase **Product**, y su único parámetro es el nombre del producto. En las líneas 9, 13 y 17 se fija el nombre de cada producto.

En la línea 21 se crea un objeto de tipo **Lines** que contiene la lista de líneas de fabricación. Cada una de las líneas se define a través de la clase **Line**. La primera línea de producción se crea en la línea de código fuente número 25, y la segunda se crea en la línea de código fuente número 60. La clase **Line** tiene tres atributos principales: **Name**, **Assignments** y **ChangeOvers**, que contienen el nombre, los productos que puede procesar la línea, y los cambios de utillaje por cambios de producto en la línea, respectivamente. Los nombres de las líneas se fijan en las líneas de código fuente número 27 y 62. La clase **Assignments** contiene una lista de objetos **Assignment**, que contiene los parámetros de trabajo de la línea para fabricar un producto. Entre las líneas de código fuente 32 y 36 se indica que la línea "A04" puede fabricar el producto "P001" (línea de código fuente 34), a una tasa de 20 unidades de producto por hora (línea de código fuente 35), y que el tiempo mínimo de fabricación de este producto en esta línea es de 4 horas (línea de código fuente 33). De la misma forma entre las líneas 38 y 42 del código fuente, se indican los parámetros de funcionamiento del proceso del producto "P104" en la línea "A04". La clase **ChangeOvers** contiene una lista de objetos **ChangeOver**, que define el tiempo de cambio de utillaje al cambiar el producto de fabricación en la línea, después de fabricar anteriormente otro producto. Como ejemplo, en la línea de código fuente número 47 se crea un objeto de la clase **ChangeOver**, que indica que si en la línea "A04" se está fabricando el producto "P001" (línea de código fuente 48), y después se va a fabricar el producto "P104" (línea de código fuente 49), serán necesarias 4 horas para realizar el cambio de utillaje (línea de código fuente 50). De la misma manera, entre las líneas de código fuente 53 y 57, se definen el segundo cambio de utillaje de la línea "A04", entre los productos "P104" y "P101".

En la línea 95 se crea un objeto de tipo **Orders** que contiene una lista de pedidos. Un pedido se define a través de un objeto de la clase **Order**. Como ejemplo, en la línea de código fuente número 98 se crea una orden para fabricar 1000 unidades (línea de código fuente 101), del producto "P001". El identificador del pedido es "Ord001" (línea de código fuente 99), y no tiene fecha de entrega (línea de código fuente 102). De la misma forma, entre las líneas de código fuente 105 y 110 se define el segundo pedido, y entre las líneas de código fuente 112 y 117, se define el tercer pedido.

Una vez definido el problema, se debe enviar al servicio web: en la línea 120 se crea el acceso al servicio web, y en la línea 124 se envía el problema al servicio web. La llamada a la función **solve** devuelve un objeto de tipo **Request**, que indica si la llamada al servicio web ha tenido algún problema. En la línea 126 se comprueba si ha existido algún error. Si no ha habido errores, se entra en la parte de búsqueda de la solución. En la línea 131 se crea el acceso al servicio web que informa del estado de resolución del problema. Desde la línea 136 a la 149 se define un bucle con un timer que espera a que se resuelva el problema. Desde las líneas 152 a la 163, se comprueba el estado final de resolución. Si el estado es PROCESSED, se accede a la solución del problema (líneas 154 y 155), que viene encapsulada en un objeto de tipo **PppSolution**.

```

1 // Crea la instancia de problema
2 serviciowebppp.PppProblem oProblem = new serviciowebppp.PppProblem();
3
4 // Crea los productos que se fabrican
5 Products products = new Products();
6 oProblem.setProducts(products);
7
8 Product product1 = new Product();
9 product1.setName("P001");
10 products.getProduct().add(product1);
11
12 Product product2 = new Product();
13 product2.setName("P104");
14 products.getProduct().add(product2);
15
16 Product product3 = new Product();
17 product3.setName("P267");
18 products.getProduct().add(product3);
19
20 // Crea dos líneas de producción
21 Lines lines = new Lines();
22 oProblem.setLines(lines);
23
24 // Crea la primera línea y la define
25 Line line1 = new Line();
26 lines.getLine().add(line1);
27 line1.setName("A04");
28
29 Assignments assignments1 = new Assignments();
30 line1.setAssignments(assignments1);
31
32 Assignment assignment11 = new Assignment();
33 assignment11.setMinTime(4.0);
34 assignment11.setProduct("P001");
35 assignment11.setRate(20.0);
36 assignments1.getAssignment().add(assignment11);
37
38 Assignment assignment12 = new Assignment();
39 assignment12.setMinTime(4.0);
40 assignment12.setProduct("P104");
41 assignment12.setRate(10.0);
42 assignments1.getAssignment().add(assignment12);
43
44 ChangeOvers changeOvers1 = new ChangeOvers();
45 line1.setChangeOvers(changeOvers1);
46
47 ChangeOver changeOver11 = new ChangeOver();
48 changeOver11.setProductFrom("P001");
49 changeOver11.setProductTo("P104");
50 changeOver11.setTime(4.0);
51 changeOvers1.getChangeOver().add(changeOver11);
52
53 ChangeOver changeOver12 = new ChangeOver();
54 changeOver12.setProductFrom("P104");
55 changeOver12.setProductTo("P001");
56 changeOver12.setTime(2.0);
57 changeOvers1.getChangeOver().add(changeOver12);
58
59 // Crea la segunda línea y la define
60 Line line2 = new Line();
    
```

```

61 lines.getLine().add(line2);
62 line2.setName("B07");
63
64 Assignments assignments2 = new Assignments();
65 line2.setAssignments(assignments2);
66
67 Assignment assignment21 = new Assignment();
68 assignment21.setMinTime(4.0);
69 assignment21.setProduct("P001");
70 assignment21.setRate(25.0);
71 assignments2.getAssignment().add(assignment21);
72
73 Assignment assignment22 = new Assignment();
74 assignment22.setMinTime(4.0);
75 assignment22.setProduct("P267");
76 assignment22.setRate(12.0);
77 assignments2.getAssignment().add(assignment22);
78
79 ChangeOvers changeOvers2 = new ChangeOvers();
80 line2.setChangeOvers(changeOvers2);
81
82 ChangeOver changeOver21 = new ChangeOver();
83 changeOver21.setProductFrom("P001");
84 changeOver21.setProductTo("P267");
85 changeOver21.setTime(6.0);
86 changeOvers2.getChangeOver().add(changeOver21);
87
88 ChangeOver changeOver22 = new ChangeOver();
89 changeOver22.setProductFrom("P267");
90 changeOver22.setProductTo("P001");
91 changeOver22.setTime(4.0);
92 changeOvers2.getChangeOver().add(changeOver22);
93
94 // Crea los pedidos
95 Orders orders = new Orders();
96 oProblem.setOrders(orders);
97
98 Order order1 = new Order();
99 order1.setIdentifier("Ord001");
100 order1.setProduct("P001");
101 order1.setDemand(1000);
102 order1.setLimitDate("");
103 orders.getOrder().add(order1);
104
105 Order order2 = new Order();
106 order2.setIdentifier("Ord002");
107 order2.setProduct("P267");
108 order2.setDemand(1200);
109 order2.setLimitDate("");
110 orders.getOrder().add(order2);
111
112 Order order3 = new Order();
113 order3.setIdentifier("Ord003");
114 order3.setProduct("P104");
115 order3.setDemand(1600);
116 order3.setLimitDate("");
117 orders.getOrder().add(order3);
118
119 // Crea el acceso al servicio web
120 WsPPPService service = new WsPPPService();
    
```

```

121 serviciowebppp.WsPPP oCliente = service.getWsPPPPort();
122
123 // Lanza la petición de resolución
124 Request oResultado = port.solve("usuario", "password", oProblem);
125
126 if (oResultado.getStatus().equals("Failed")) {
127     System.out.println("Error en la llamada: " +
128         oResultado.getMessage().get(0).getValue());
129 } else if (oResultado.getStatus().equals("Ok")) {
130     // Espera a que se ejecute el problema
131     WsProgressService serviceExecutionProgress = new WsProgressService();
132     WsProgress portProgress = serviceExecutionProgress.
133     getWsProgressPort();
134
135     String strEstado = "";
136     boolean bTerminado = false;
137     while (!bTerminado) {
138         strEstado = portProgress.checkStatus("usuario", "password",
139             oResultado.getTicket());
140
141         if (strEstado.equals("UNKNOWN") || strEstado.equals("ERROR") ||
142             strEstado.equals("PROCESSED")) {
143             bTerminado = true;
144         } else if (strEstado.equals("ALLOCATED") ||
145             strEstado.equals("IN_PROCESS")) {
146             Thread.sleep(5000);
147         } else { // Estado desconocido
148             bTerminado = true;
149         }
150     }
151
152     // Lee el resultado del problema
153     if (strEstado.equals("PROCESSED")) {
154         // Lee la solución
155         OfapSolution oSolution = oCliente.getSolution("usuario",
156             "password", oResultado.getTicket());
157         // Tratar la solución.
158     } else if (strEstado.equals("UNKNOWN")) {
159         System.out.println("Ticket de acceso a la solución no válido.");
160     } else if (strEstado.equals("ERROR")) {
161         System.out.println("Se han producido errores en la resolución.");
162     } else {
163         System.out.println("Se ha producido un error desconocido.");
164     }
165 }

```

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

### 3.6 Ejemplo de uso desde Eclipse

En este apartado, se muestra un programa de ejemplo para resolver una instancia del PPP, 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 PPP 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".

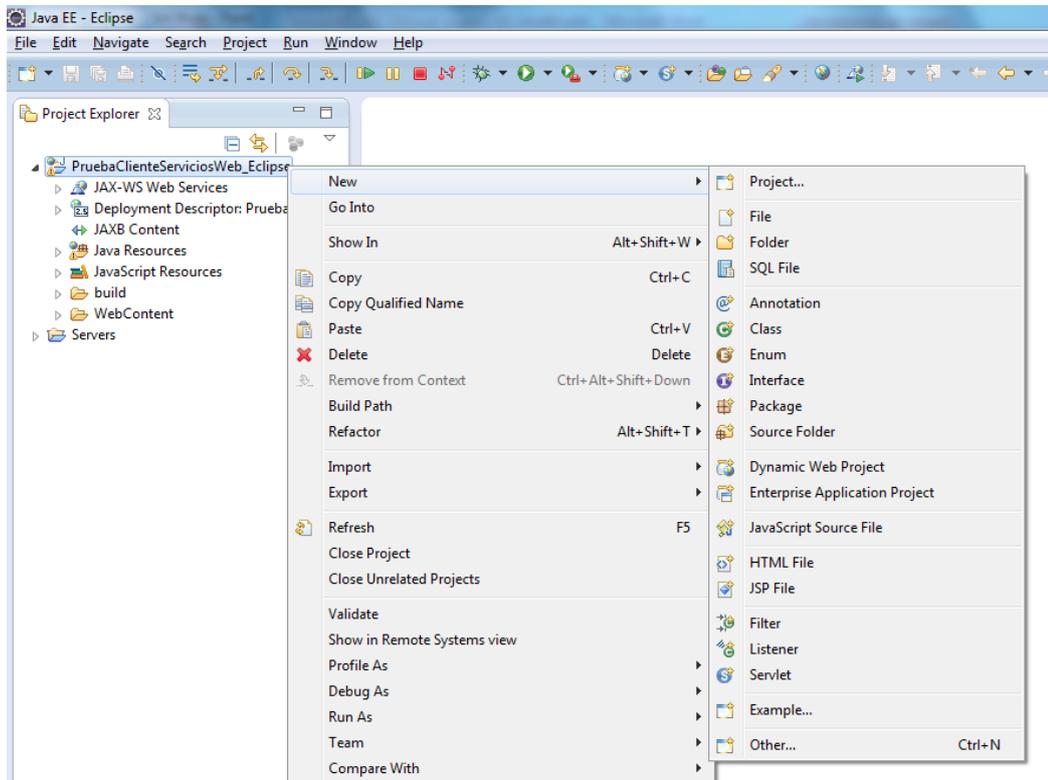


Figura 3-10. Paso 1 - Creación de cliente del servicio web Eclipse

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

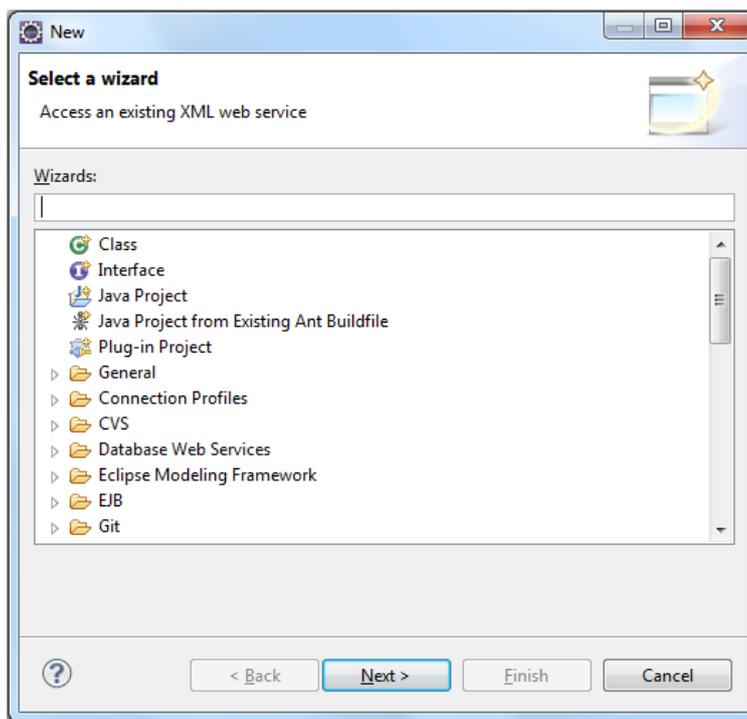


Figura 3-11. Paso 1 - Creación de cliente del servicio web Eclipse (2)

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

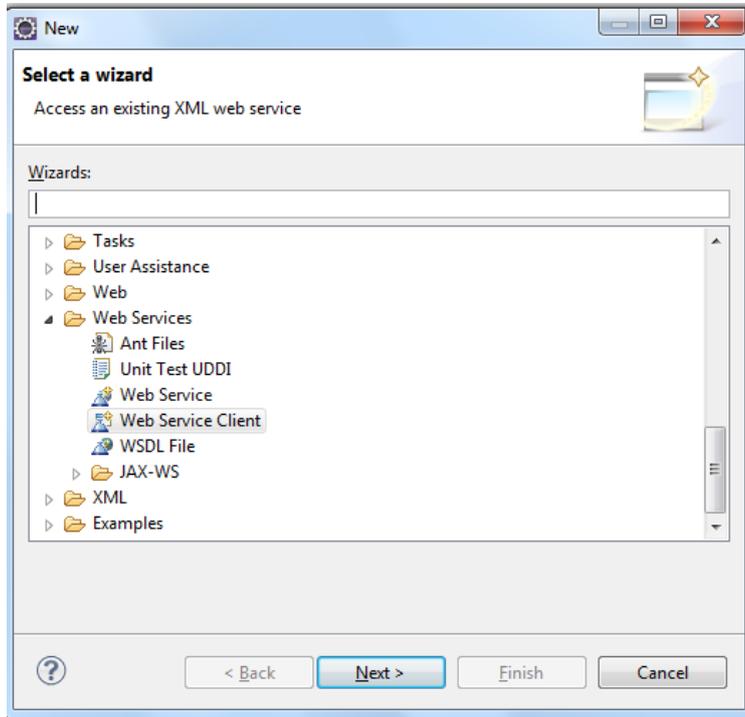


Figura 3-12. Paso 2 - Creación de cliente del servicio web Eclipse

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

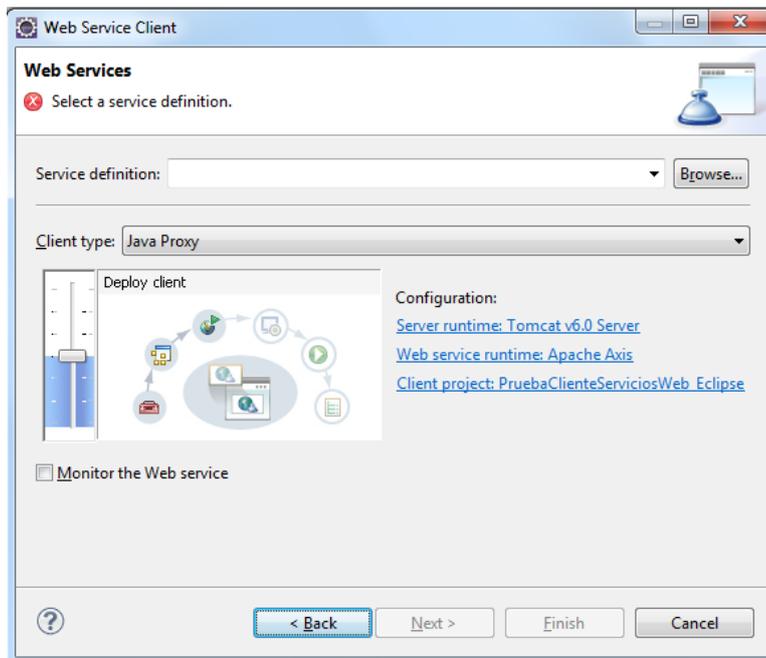


Figura 3-13. Paso 2 - Creación de cliente del servicio web Eclipse (2)

- En la caja de texto "Service definition" se escribirá lo siguiente:

<http://www.vetu.es/wsPPP/wsPPP?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 **wsPPP**, 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:

- **WsPPPSERVICE**: Es la clase de acceso al servicio web. Tiene las funciones **solve** y **getSolution**.
- **PppProblem** Es la clase que encapsula la definición de la instancia de problema que se quiere resolver. Contiene a las clases **Products**, **Lines** y **Orders**, que contienen la definición de los tipos de los productos que se fabrica en la fábrica, las definiciones de las líneas de fabricación y los pedidos, respectivamente.
- **PppSolution**: Es la clase que encapsula la solución del problema que se ha resuelto. Contiene a las clases **Result**, **ObjectiveValue** y **AllocatedLines**, que tiene la información de la resolución del problema, de los valores encontrados y las asignaciones de productos, 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 **WsProgressService**, que es el acceso al servicio web, y tiene la función **checkStatus**.

El código fuente necesario para la resolución del PPP mediante los servicios web de VETU se muestra en la Figura 3-8. 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 5 a la 123 se definen los datos del problema a resolver, y desde la línea 126 hasta el final, se implementa el acceso a los servicios web.

En la línea 2 se crea la instancia de la clase **PppProblem**, que encapsula toda la información del problema, que son los productos, las líneas y los pedidos.

En la línea 5 se crean los productos que fabrica la empresa, que es un array de 3 productos. En las líneas 8, 12 y 16 se crean cada uno de los tres productos. Un producto se define por la clase **Product**, y el único parámetro del objeto **Product** es el nombre del producto. En las líneas 9, 13 y 17 se fija el nombre de cada producto.

En la línea 21 se crea un array de 2 líneas de fabricación. Las líneas se definen con creando un objeto de la clase **Line**. La primera línea de producción se crea en la línea de código fuente número 25, y la segunda se crea en la línea de código fuente número 60. La clase **Line** tiene tres atributos principales: **Name**, **Assignments** y **ChangeOvers**, que contienen el nombre, los productos que puede procesar la línea, y los cambios de utillaje por cambios de producto en la línea, respectivamente. Los nombres de las líneas se fijan en las líneas de código fuente número 27 y 62. El objeto de tipo **Assignments** es un array de objetos de tipo **Assignment**, que contiene los parámetros de trabajo de la línea para fabricar un producto. En la línea de código fuente número 29 se indica que la línea "A04" puede fabricar dichos productos. Entre las líneas de

código fuente 32 y 36 se indica que la línea "A04" puede fabricar el producto "P001" (línea de código fuente 34), a una tasa de 20 unidades de producto por hora (línea de código fuente 35), y que el tiempo mínimo de fabricación de este producto en esta línea es de 4 horas (línea de código fuente 33). De la misma forma entre las líneas 38 y 42 del código fuente, se indican los parámetros de funcionamiento del proceso de del producto "P104" en la línea "A04". El objeto de tipo **ChangeOvers** es un array de objetos de tipo **ChangeOver**, que define el tiempo de cambio de utillaje al cambiar el producto de fabricación en la línea, después de fabricar anteriormente otro producto. Como ejemplo, en la línea de código fuente número 47 se crea un objeto de la clase **ChangeOver**, que indica que si en la línea "A04" se está fabricando el producto "P001" (línea de código fuente 48), y después se va a fabricar el producto "P104" (línea de código fuente 49), serán necesarias 4 horas para realizar el cambio de utillaje (línea de código fuente 50). De la misma manera, entre las líneas de código fuente 53 y 57, se definen el segundo cambio de utillaje de la línea "A04", entre los productos "P104" y "P101".

En la línea 95 se crea un array de 3 pedidos. Un pedido se define con el objeto de tipo **Order**. Como ejemplo, en la línea de código fuente número 98 se crea una orden para fabricar 1000 unidades (línea de código fuente 101), del producto "P001". El identificador del pedido es "Ord002" (línea de código 99), y no tiene fecha de entrega (línea de código fuente 102). De la misma forma, entre las líneas de código fuente 105 y 110 se define el segundo pedido, y entre las líneas de código fuente 112 y 117, se define el tercer pedido.

Una vez definido el problema, se debe enviar al servicio web: en la línea 120 se crea el acceso al servicio web, y en la línea 124 se envía el problema al servicio web. La llamada a la función **solve** devuelve un objeto de tipo **Request**, que indica si la llamada al servicio web ha tenido algún problema. En la línea 126 se comprueba si ha existido algún error. Si no ha habido errores, se entra en la parte de búsqueda de la solución. En la línea 131 se crea el acceso al servicio web que informa del estado de resolución del problema. Desde la línea 137 a la 150 se define un bucle con un timer que espera a que se resuelva el problema. Desde las líneas 153 a la 164, se comprueba el estado final de resolución. Si el estado es PROCESSED, se accede a la solución del problema (líneas 155 y 156), que viene encapsulada en un objeto de tipo **PppSolution**.

```

1 // Crea la instancia de problema
2 servicioWebPPP.PppProblem oProblem = new servicioWebPPP.PppProblem();
3
4 // Crea los productos que se fabrican
5 Product[] products = new Product[3];
6 oProblem.setProducts(products);
7
8 Product product1 = new Product();
9 product1.setName("P001");
10 products[0] = product1;
11
12 Product product2 = new Product();
13 product2.setName("P104");
14 products[1] = product2;
15
16 Product product3 = new Product();
17 product3.setName("P267");
18 products[2] = product3;
19
20 // Crea dos líneas de producción
21 Line[] lines = new Line[2];
22 oProblem.setLines(lines);
23
24 // Crea la primera línea y la define
25 Line line1 = new Line();
26
    
```

```

27 line1[0] = line1;
28 line1.setName("A04");
29
30 Assignment[] assignments1 = new Assignment[2];
31 line1.setAssignments(assignments1);
32
33 Assignment assignment11 = new Assignment();
34 assignment11.setMinTime(4.0);
35 assignment11.setProduct("P001");
36 assignment11.setRate(20.0);
37 assignments1[0] = assignment11;
38
39 Assignment assignment12 = new Assignment();
40 assignment12.setMinTime(4.0);
41 assignment12.setProduct("P104");
42 assignment12.setRate(10.0);
43 assignments1[1] = assignment12;
44
45 ChangeOver[] changeOvers1 = new ChangeOver[2];
46 line1.setChangeOvers(changeOvers1);
47
48 ChangeOver changeOver11 = new ChangeOver();
49 changeOver11.setProductFrom("P001");
50 changeOver11.setProductTo("P104");
51 changeOver11.setTime(4.0);
52 changeOvers1[0] = changeOver11;
53
54 ChangeOver changeOver12 = new ChangeOver();
55 changeOver12.setProductFrom("P104");
56 changeOver12.setProductTo("P001");
57 changeOver12.setTime(2.0);
58 changeOvers1[1] = changeOver12;
59
60 // Crea la segunda línea y la define
61 Line line2 = new Line();
62 lines[1] = line2;
63 line2.setName("B07");
64
65 Assignment[] assignments2 = new Assignment[2];
66 line2.setAssignments(assignments2);
67
68 Assignment assignment21 = new Assignment();
69 assignment21.setMinTime(4.0);
70 assignment21.setProduct("P001");
71 assignment21.setRate(25.0);
72 assignments2[0] = assignment21;
73
74 Assignment assignment22 = new Assignment();
75 assignment22.setMinTime(4.0);
76 assignment22.setProduct("P267");
77 assignment22.setRate(12.0);
78 assignments2[1] = assignment22;
79
80 ChangeOver[] changeOvers2 = new ChangeOver[2];
81 line2.setChangeOvers(changeOvers2);
82
83 ChangeOver changeOver21 = new ChangeOver();
84 changeOver21.setProductFrom("P001");
85 changeOver21.setProductTo("P267");
86 changeOver21.setTime(6.0);
    
```

```

87  changeOvers2[0] = changeOver21;
88
89  ChangeOver changeOver22 = new ChangeOver();
90  changeOver22.setProductFrom("P267");
91  changeOver22.setProductTo("P001");
92  changeOver22.setTime(4.0);
93  changeOvers2[1] = changeOver22;
94
95  // Crea los pedidos
96  Order[] orders = new Order[3];
97  oProblem.setOrders(orders);
98
99  Order order1 = new Order();
100 order1.setIdentifier("Ord001");
101 order1.setProduct("P001");
102 order1.setDemand(1000);
103 order1.setLimitDate("");
104 orders[0] = order1;
105
106 Order order2 = new Order();
107 order2.setIdentifier("Ord002");
108 order2.setProduct("P267");
109 order2.setDemand(1200);
110 order2.setLimitDate("");
111 orders[1] = order2;
112
113 Order order3 = new Order();
114 order3.setIdentifier("Ord003");
115 order3.setProduct("P104");
116 order3.setDemand(1600);
117 order3.setLimitDate("");
118 orders[2] = order3;
119
120 // Crea el acceso al servicio web
121 WsPPPService service = new WsPPPServiceLocator();
122 servicioWebPPP.WsPPP oCliente = service.getwsPPPPort();
123
124 // Lanza la petición de resolución
125 Request oResultado = port.solve("usuario ", "password", oProblem);
126
127 if (oResultado.getStatus().equals("Failed")) {
128     System.out.println("Error en la llamada: " +
129         oResultado.getMessage(0).get_value());
130 } else if (oResultado.getStatus().equals("Ok")) {
131     // Espera a que se ejecute el problema
132     WsProgressService serviceExecutionProgress = new
133     WsProgressServiceLocator();
134     WsProgress portProgress = serviceExecutionProgress.getwsProgressPort();
135
136     String strEstado = "";
137     boolean bTerminado = false;
138     while (!bTerminado) {
139         strEstado = portProgress.checkStatus("usuario", "password",
140             oResultado.getTicket());
141
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     bTerminado = true;
150   }
151 }
152
153 // Lee el resultado del problema
154 if (strEstado.equals("PROCESSED")) {
155   // Lee la solución
156   CapSolution oSolution = oCliente.getSolution("usuario", "password",
157   oResultado.getTicket());
158   // Tratar la solución.
159 } else if (strEstado.equals("UNKNOWN")) {
160   System.out.println("Ticket de acceso a la solución no válido.");
161 } else if (strEstado.equals("ERROR")) {
162   System.out.println("Se han producido errores en la resolución.");
163 } else {
164   System.out.println("Se ha producido un error desconocido.");
165 }
}
```

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

## 4 Listas de mensajes

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

### 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 **wsPPP**, cuando se envía una solicitud de resolución de una instancia del PPP.

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 <b>solve()</b> no tiene contrato para resolver el problema de VETU.
SRV4	The contract has expired  El contrato para acceder a la 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 wsPPP**

### 4.2 Mensajes del PPP

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

Código	Descripción
PPO	<p>The PPP problem has been successfully solved.</p> <p>El problema se ha resuelto correctamente.</p>
PPP1	<p>There are bugs in the entry file.</p> <p>El formato del fichero de entrada del PPP no corresponde con el definido por la plataforma VETU, y el problema no se puede resolver.</p>
PPP2	<p>Error solving the problem. Please contact support team.</p> <p>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.</p>
PPP3	<p>The problem has no feasible solution. Please check problem data.</p> <p>El problema no tiene solución.</p> <p>Si después de revisar el problema, no encuentra la causa por la que no se puede resolver, contacte con el servicio técnico.</p>
PPP4	<p>Data entry has not defined Products object.</p> <p>El objeto <b>Products</b> define los productos que se fabrican en la fábrica. Si no se definen los productos, no se puede resolver el problema.</p>
PPP5	<p>Data entry has not defined Lines object</p> <p>El objeto <b>Lines</b> define las líneas o máquinas que fabrican Si no se especifican las líneas, no se puede resolver el problema.</p>
PPP6	<p>Data entry has not defined Orders object</p> <p>El objeto <b>Orders</b> define los pedidos que se deben fabricar. Si no se especifican los pedidos, no se puede resolver el problema.</p>
PPP7	<p>There is not any Product object</p> <p>No se ha definido ningún producto. Si no se especifican productos, no se puede resolver el problema.</p>
PPP8	<p>There is not any Line object.</p> <p>No se ha definido ninguna línea o máquina para fabricar. Si no se define ninguna línea, no se puede resolver el problema.</p>

PPP9	<p>There is not any Order object.</p> <p>No se ha definido ningún pedido. Si no se especifican pedidos, no se puede resolver el problema.</p>
PPP10	<p>Some Product has not defined Name object.</p> <p>La definición de algún producto no contiene el campo <b>Name</b>. Este campo es necesario para poder identificar cada producto.</p>
PPP11	<p>Some Product has empty value for Name object.</p> <p>La definición de algún producto no ha definido el valor del campo <b>Name</b>. Este campo es necesario para poder identificar cada producto.</p>
PPP12	<p>Some Order has not defined Identifier object.</p> <p>La definición de algún pedido no contiene el campo <b>Identifier</b>. Este campo es necesario para poder identificar cada pedido.</p>
PPP13	<p>Some Order has empty value for Identifier object.</p> <p>La definición de algún pedido no ha definido el valor del campo <b>Identifier</b>. Este campo es necesario para poder identificar cada pedido.</p>
PPP14	<p>Order xxx has not defined Product object.</p> <p>El pedido de identificador xxx no ha definido el campo <b>Product</b>. Es necesario especificar el producto a que se refiere el pedido, para poder asignar el pedido a una línea.</p>
PPP15	<p>Order xxx has empty value for Product object.</p> <p>El pedido de identificador xxx no ha definido el valor del campo <b>Product</b>. Es necesario especificar el producto a que se refiere el pedido, para poder asignar el pedido a una línea.</p>
PPP16	<p>Order xxx has not defined Demand object.</p> <p>El pedido de identificador xxx no ha definido el campo <b>Demand</b>. Es necesario especificar cantidad de producto del pedido, para poder asignar el pedido a una línea.</p>
PPP17	<p>Order xxx has empty value for Demand objet.</p> <p>El pedido de identificador xxx no ha definido el valor del campo <b>Demand</b>. Es necesario especificar cantidad de producto del pedido, para poder asignar el pedido a una línea.</p>

PPP18	<p>The value xxx for Demand objet for Order ooo is not valid.</p> <p>El valor xxx que se ha especificado como demanda del pedido de identificador ooo no es válido. La demanda debe ser un número (double) positivo.</p>
PPP19	<p>Order xxx has not defined LimitDate object.</p> <p>El pedido de identificador xxx no define el campo <b>LimitDate</b>. Este campo es obligatorio de especificar para definir un pedido.</p>
PPP20	<p>The value xxx for LimitDate objet for Order ooo is not valid.</p> <p>El pedido de identificador xxx no define un valor para el campo <b>LimitDate</b>. Este campo es obligatorio de especificar para definir un pedido. Si el pedido xxx no tiene fecha tope de entrega, se debe especificar el valor "-1"</p>
PPP21	<p>Some Line has not defined Name object.</p> <p>Alguna línea no ha definido el campo <b>Name</b>. El valor de este campo es necesario para poder identificar cada línea.</p>
PPP22	<p>Some Line has empty value for Name object.</p> <p>Alguna línea no ha definido el valor del campo <b>Name</b>. El valor de este campo es necesario para poder identificar cada línea.</p>
PPP23	<p>There is not any Assignments object for Line lll.</p> <p>La línea de nombre lll no ha definido el campo <b>Assignments</b>. Es necesario especificar los productos que puede fabricar la línea, para poder resolver el problema.</p>
PPP24	<p>The line lll has not define any Assignment.</p> <p>La línea de nombre lll no ha definido ningún campo <b>Assignment</b>. Es necesario especificar los productos que puede fabricar la línea, para poder resolver el problema.</p>
PPP25	<p>Some assignment for line lll has not defined Product object.</p> <p>Alguna asignación de producto de la línea lll no ha definido el campo <b>Product</b>. Es necesario identificar el producto que puede fabricar la línea.</p>
PPP26	<p>Some assignment for line lll has empty value for Product object.</p> <p>Alguna asignación de producto de la línea lll no ha definido el valor del campo <b>Product</b>. Es necesario identificar el producto que puede fabricar la línea.</p>
PPP27	<p>Some assignment for line lll has not defined Rate object.</p>

	<p>Alguna asignación de producto de la línea III no ha definido el campo <b>Rate</b>. Es necesario especificar la tasa de producción de cada producto en la línea.</p>
PPP28	<p><b>Some assignment for line lll has empty value for Rate object.</b></p> <p>Alguna asignación de producto de la línea III no ha definido el valor del campo <b>Rate</b>. Es necesario especificar la tasa de producción de cada producto en la línea.</p>
PPP29	<p><b>The value vvv for Rate object for some assignment of some order for line lll is not valid.</b></p> <p>El valor vvv que se ha especificado como tasa de producción en alguna asignación de producto de la línea III no es válido. La tasa de producción es un número (double) positivo.</p>
PPP30	<p><b>Some assignment for line lll has empty value for MinTime object.</b></p> <p>Alguna asignación de producto de la línea III no ha definido el campo <b>MinTime</b>. Es necesario especificar el tiempo mínimo de fabricación de cada producto en la línea.</p>
PPP31	<p><b>Some assignment for line lll has empty value for MinTime object.</b></p> <p>Alguna asignación de producto de la línea III no ha definido el valor del campo <b>MinTime</b>. Es necesario especificar el tiempo mínimo de fabricación de cada producto en la línea.</p>
PPP32	<p><b>The value vvv for MinTime objet for some assignment of some order for line lll is not valid.</b></p> <p>El valor vvv que se ha especificado como tiempo mínimo de fabricación de un producto en la línea III no es válido. El tiempo mínimo de fabricación de un producto en una línea es un número (double) positivo.</p>
PPP33	<p><b>There is not any ChangeOvers object for Line lll.</b></p> <p>No se ha definido el campo <b>ChangeOvers</b> de la línea III. Es necesario especificar los tiempos de cambio de utillaje de cada línea, para poder resolver el problema. Si no se especifica ningún cambio de utillaje, este campo puede estar vacío.</p>
PPP34	<p><b>Some changeover for line lll has not defined ProductFrom object.</b></p> <p>Algún campo <b>ChangeOver</b> de la línea III no ha definido el campo <b>ProductFrom</b>. Es necesario especificar el valor de los dos productos que conllevan cambio de utillaje.</p>
PPP35	<p><b>Some changeover for line lll has empty value for ProductFrom object.</b></p> <p>Algún campo <b>ChangeOver</b> de la línea III no ha definido el valor del campo <b>ProductFrom</b>. Es necesario especificar el valor de los dos productos que conllevan</p>

	cambio de utillaje.
PPP36	<p>Some changeover for line lll has not defined ProductTo object.</p> <p>Algún campo <b>ChangeOver</b> de la línea lll no ha definido el campo <b>ProductTo</b>. Es necesario especificar el valor de los dos productos que conllevan cambio de utillaje.</p>
PPP37	<p>Some changeover for line lll has empty value for ProductTo object.</p> <p>Algún campo <b>ChangeOver</b> de la línea lll no ha definido el valor del campo <b>ProductTo</b>. Es necesario especificar el valor de los dos productos que conllevan cambio de utillaje.</p>
PPP38	<p>Some changeover for line lll has not defined Time object.</p> <p>Algún objeto <b>ChangeOver</b> definido para la línea lll no ha definido el campo <b>Time</b>. El tiempo de cambio es necesario para especificar una operación de cambio de utillaje.</p>
PPP39	<p>Some changeover for line lll has empty value for Time objet.</p> <p>Algún objeto <b>ChangeOver</b> definido para la línea lll no ha definido el valor del campo <b>Time</b>. El tiempo de cambio es necesario para especificar una operación de cambio de utillaje.</p>
PPP40	<p>The value vvv for Time objet for some changeover for line vv is not valid.</p> <p>El valor vvv definido para el campo <b>Time</b> de algún objeto <b>ChangeOver</b> no es valido. El tiempo de cambio de utillaje debe ser ubn número (double) positivo.</p>

## 5 Próximas versiones

Este manual corresponde a la versión de la resolución del PPP 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 PPP 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 PPP en VETU. Si desea más información sobre estas nuevas funciones, puede ponerse en contacto con el servicio técnico de Infozara.

### 5.1 Solape de cambios de utillaje

Al cambiar el producto que fabrica una máquina o línea de producción, es habitual realizar un cambio de utillaje. Estos cambios los realizan operarios especializados, que tienen una capacidad de trabajo limitada. Esto significa que en la fábrica se puede hacer un limitado número de cambios de utillaje simultáneos por unidad de tiempo. En la próxima versión del PPP que publica Vetu se podrá especificar la carga de trabajo máxima que puede aplicarse a cambios de utillaje por unidad de tiempo.

### 5.2 Minimizar coste

El criterio de optimización del PPP actual que publica Vetu es minimizar el tiempo global de producción. Si cada máquina tiene un coste de producción diferente, otro criterio de optimización podría ser minimizar el coste total de producción de los pedidos. La planificación más rápida no será necesariamente la de menor coste, y viceversa. La próxima versión del PPP que publique Vetu permitirá seleccionar el criterio de optimización en la producción.

#### 5.2.1 Penalizaciones por retraso

Si el criterio de optimización de la producción es minimizar el coste total, se pueden tener en cuenta posibles penalizaciones en el retraso de entrega de ciertos pedidos. Esto significa que la empresa puede estar dispuesta a pagar ciertas penalizaciones en el retraso, si eso lleva a planificaciones globales de menor coste. La próxima versión del PPP que publique Vetu permitirá especificar penalizaciones en la entrega de pedidos, cuando el criterio de optimización sea minimizar el coste total de la producción.