

vVacation

Biblioteca de desarrollo para
el cálculo de vacaciones con igualdad social

Manual de usuario

Versión 1.2

**Control de versiones**

Versión	Contenido
1.0	Versión inicial de vVacation. Biblioteca en C++.
1.1	Versión compatible con .NET.
1.2	Versión componente ejecutable en cualquier plataforma.



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.....	5
1.1	En este manual.....	5
1.2	¿Quiénes somos?.....	5
1.3	Conocimientos necesarios.....	6
1.4	Nomenclatura.....	6
1.5	Soporte.....	6
1.6	Actualización de versiones.....	6
1.7	Más información.....	7
2	Asignación de vacaciones.....	8
2.1	Introducción.....	8
2.2	El proceso.....	8
2.3	Naturaleza del problema.....	9
3	Manual de usuario.....	13
3.1	Introducción.....	13
3.2	Contenido.....	13
3.3	Modelo conceptual de vVacation.....	13
3.4	Instalación.....	14
3.5	Licencias.....	14
3.6	Configuración de un proyecto C++.....	15
3.7	Configuración de un proyecto .NET.....	15
3.8	Ejemplo de programación en C++.....	15
3.9	Ejemplo de programación en .NET.....	17
3.10	Componente ejecutable.....	20
3.10.1	Uso del componente ejecutable.....	20
3.10.2	Características de objetos y campos del problema.....	21
4	Manual de referencia C++.....	23
4.1	vSolverVacaciones.....	23
4.2	vvTematica.....	24
4.3	vvEmpleado.....	25
4.4	vvGrupoTematicas.....	26
4.5	vvSolverVacaciones.....	27
5	Manual de referencia .NET.....	30
5.1	vvcppVacationSolver.....	30



1 Introducción

El reparto de las vacaciones en colectivos numerosos es una tarea complicada, porque la empresa debe garantizar el nivel de servicio, y los empleados quieren disfrutar las vacaciones en periodos elegidos por ellos. El problema del reparto de vacaciones consiste en encontrar un buen balance entre el nivel de servicio que ofrece la empresa y el cumplimiento de la demanda de vacaciones que realizan los empleados.

vVacation es una biblioteca de desarrollo que permite calcular el mejor reparto de vacaciones, basado en un sistema de puntuación que garantiza la igualdad social de todos los empleados, a la vez que mantiene el nivel de servicio que ofrece la empresa a sus clientes.

Este documento es el manual de usuario de la biblioteca vVacation, y presenta toda la información necesaria para poder utilizarla, con varios ejemplos. Este documento es también una guía de usuario que describe todas las clases y funciones de la biblioteca.

Este manual está dirigido a programadores que desarrollan aplicaciones de cálculo, con conocimientos básicos de C++ o .NET. vVacation también está disponible como componente ejecutable que se puede invocar desde cualquier plataforma de desarrollo.

1.1 En este manual

Este manual está estructurado en los siguientes capítulos:

- Asignación de vacaciones
En este capítulo se describe el modelo de problema de asignación de vacaciones a colectivos de trabajadores que resuelve vVacation.
- Biblioteca vVacation
Se ofrece una guía detallada para un correcto uso de la biblioteca vVacation.
- Guía de usuario
Se detallan todas las clases y funciones de la biblioteca vVacation.

1.2 ¿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.sprocs.es), LISBB (www.lisbb.es), QRP (qrp.infozara.es), AMBÚ (ambu.infozara.es) y SMOTY (www.smoty.es) del Plan nacional de I+D+i.

- Ha desarrollado productos y componentes industriales en estado precompetitivo en el marco de los proyectos anteriores o como desarrollo posterior ante demanda del mercado.
- Ha construido y está construyendo servicios en la Cloud y en el ámbito del Internet de las Cosas (IoT).
- Tiene una cartera de clientes a los que se les está ofreciendo servicios de valor añadido.

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

1.3 Conocimientos necesarios

Como vVacation es una biblioteca escrita para programadores en C++, este manual asume que el lector tiene experiencia en el desarrollo de programas en C++, y tiene conocimientos de uso de algún entorno de desarrollo en C++. VVacation también está disponible como ensamblado .NET, pero solo son necesarios unos mínimos conocimientos de la plataforma .NET para realizar aplicaciones que utilicen este ensamblado.

No son necesarios conocimientos de técnicas de optimización para el uso de vVacation.

1.4 Nomenclatura

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

1.5 Soporte

Si es usted ha adquirido licencias de vVacation, puede obtener soporte técnico sobre el uso de la biblioteca, poniéndose en contacto con el soporte técnico de Infozara.

Si usted no ha adquirido licencias de la biblioteca vVacation, y tiene cualquier duda o sugerencia sobre el producto, puede ponerse en contacto con el equipo de Infozara por los mecanismos descritos en el apartado 1.7.

1.6 Actualización de versiones

vVacation es una biblioteca software para la resolución del problema del reparto de vacaciones por igualdad social, manteniendo el nivel de servicio que ofrece la empresa, y evolucionará con nuevas funcionalidades. Si usted ha contratado el servicio de mantenimiento de vVacation, recibirá actualizaciones a medida que Infozara vaya publicando nuevas versiones. Los proyectos desarrollados con vVacation no se verán afectados por los cambios de versión, ya que el interface público de las clases de vVacation no cambia con las nuevas versiones.



1.7 Más información

Si desea más información sobre cualquier aspecto de la biblioteca vVacation, puede ponerse en contacto con Infozara, a través de los siguientes medios:

Teléfono: +34 976 25 43 76
Correo electrónico: informa@infozara.es
marketing@vetu.es

También puede consultar las siguientes direcciones web:

www.infozara.es
www.vetu.es
www.vetu.es/webvetu/vvacation.do



2 Asignación de vacaciones

2.1 Introducción

En empresas con colectivos grandes de personal, es habitual un proceso de negocio, que tiene el objetivo de asignar las vacaciones de los empleados de la mejor forma posible. En esta asignación de vacaciones aparecen dos aspectos importantes. Por un lado está el nivel de servicio que debe ofrecer la empresa a sus clientes mientras existe personal de vacaciones, y por otro lado están las preferencias de vacaciones de los empleados. El nivel de servicio se expresa como la cantidad de empleados que deben trabajar cada periodo de tiempo (o como la cantidad de trabajadores que pueden estar de vacaciones en un periodo de tiempo). Las preferencias de los empleados es una lista de periodos de vacaciones solicitados, ordenados de mayor a menor preferencia.

2.2 El proceso

En general, el proceso de asignación de vacaciones en empresas con grandes colectivos de personal, sigue los siguientes pasos:

1. La empresa y los empleados negocian las condiciones de asignación de vacaciones. En general, el proceso es anual.
2. La empresa envía las condiciones de asignación a los empleados, y les solicita que indiquen sus preferencias de vacaciones. En general se asignan dos periodos de dos semanas de vacaciones al año, y el empleado indica tres o cuatro periodos de vacaciones, ordenados por orden de preferencia.
3. La empresa calcula el nivel de servicio mínimo requerido para dar el servicio requerido a sus clientes. Este nivel de servicio se puede expresar de las siguientes formas:
 - Cantidad mínima de trabajadores por periodo que deben estar presentes para garantizar el nivel de servicio.
 - Cantidad máxima de trabajadores que pueden estar de vacaciones en cada periodo de tiempo, para garantizar el nivel de servicio.
4. Los empleados envían su solicitud de vacaciones a la empresa.
5. La empresa calcula el mejor reparto de vacaciones. El mejor reparto significa que los trabajadores disfrutan de sus vacaciones en los periodos que han solicitado.
6. La empresa notifica a cada empleado los periodos de vacaciones asignados.
7. Se abre un periodo de reclamaciones, y gestión de vacaciones, que dura todo el periodo.

El objetivo que se busca tras el proceso completo de asignación de vacaciones es la igualdad social colectiva y la satisfacción general de los empleados, sin que eso afecte al nivel de servicio de la empresa.

vVacation es un sistema de cálculo basado en técnicas de Investigación Operativa, que realiza de forma automática la asignación de vacaciones, maximizando la cantidad de periodos preferidos que se asignan a los empleados. El cálculo que realiza vVacation corresponde al



paso número 3 del proceso descrito anteriormente.

2.3 Naturaleza del problema

El problema de la asignación de vacaciones es un problema de optimización combinatoria, lo que significa que existen muchas formas diferentes de asignar las vacaciones a los empleados. Cada asignación de vacaciones es diferente y el objetivo es encontrar la mejor asignación de vacaciones posible. La mejor asignación de vacaciones es aquella que asigna el mayor número de periodos de vacaciones en los periodos elegidos por los empleados.

Como muestra de la naturaleza exponencial del problema, que lleva a una gran dificultad de resolución se va a plantear el pequeño ejemplo que muestra la Tabla 2-1. Este problema consiste en asignar cuatro semanas de vacaciones a cada trabajador. Las cuatro semanas de vacaciones se agrupan en dos periodos de dos semanas consecutivas. Las celdas de color rojo indican las semanas solicitadas por cada trabajador, y el número escrito en las celdas rojas indica el nivel de prioridad de esa semana para el trabajador. Por ejemplo, el trabajador T1 prefiere en primer lugar la semana 4, luego la semana 5, después la semana 9 y su última opción es la semana 11. De la misma forma, los trabajadores T0, T2 y T4 han elegido cuatro semanas de vacaciones, ordenadas por preferencia.

La fila de título 'Demanda' en la Tabla 2-1 muestra la cantidad de trabajadores que han solicitado vacaciones en cada una de las semanas. Esta fila muestra el concepto de demanda de vacaciones. Las filas de título 'Vacaciones' y 'Trabajo', expresan el nivel de servicio que debe ofrecer la empresa. El nivel de servicio se puede expresar como 'cantidad máxima de trabajadores que pueden estar de vacaciones cada semana' (fila 'Vacaciones'), o como 'cantidad mínima de trabajadores necesarios en el puesto de trabajo cada semana' (fila 'Trabajo'). Como ejemplo, en la semana 3, la empresa ha calculado que solo puede haber un trabajador de vacaciones, porque necesita tres trabajadores para poder realizar el trabajo en esa semana.

	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
T0		1	2			3	4					
T1					1	2				3		4
T2			3	4					1	2		
T3			1	2				3	4			
Demanda	0	1	3	2	1	2	1	1	2	2	0	1
Vacaciones	1	1	1	1	1	1	1	2	2	2	3	3
Trabajo	3	3	3	3	3	3	3	2	2	2	1	1

Tabla 2-1. Problema de asignación de vacaciones a cuatro empleados

A la vista de la información que refleja la Tabla 2-1, el problema de asignación de vacaciones consiste en asignar cuatro semanas de vacaciones a cada empleado de la mejor forma posible, de forma que se cumpla el nivel de servicio. La mejor forma posible será aquella en que se satisfaga en mayor medida a la comunidad de trabajadores. La mayor satisfacción de un trabajador se da si se le asignan las cuatro semanas que ha seleccionado. En caso de no asignarle las cuatro semanas que ha solicitado, el trabajador obtendrá más satisfacción si se le asignan las semanas que ha elegido en primer lugar. De esta forma se puede crear un sistema de puntuación que sea capaz de medir la satisfacción en la asignación de vacaciones



de un trabajador.

Un sistema de puntuación podría ser el siguiente: si a un empleado se le asigna una semana de prioridad 1, se asocian 5 puntos; si se asigna una semana de prioridad 2, se asocian 4 puntos; si se asigna una semana de prioridad 3, se asocian 3 puntos, si se asocia una semana de prioridad 4, se asocian 2 puntos; y si se asocia una semana que no estaba seleccionada por el trabajador, asocian 0 puntos. La Tabla 2-2 muestra el nivel de puntuación de cada nivel de prioridad. Como ejemplo de cálculo de puntuación, si al trabajador T0 se le asignan las semanas de vacaciones 1, 2, 5 y 6, tendrá $5 + 4 + 3 + 2 = 14$ puntos. En ese caso se obtiene la máxima puntuación, porque se le han asignado las cuatro semanas que había seleccionado en sus preferencias. Otro ejemplo sería, si al trabajador T3 se le asignan las semanas de vacaciones 3, 4, 8 y 9. En este caso la puntuación será de $4 + 0 + 2 + 0 = 6$ puntos.

Prioridad	Puntuación
Primera elección	5 puntos
Segunda elección	4 puntos
Tercera elección	3 puntos
Cuarta elección	2 puntos
No selección	0 puntos

Tabla 2-2. Nivel de puntuación

El problema global de asignación de vacaciones consiste en buscar la asignación de vacaciones a todos los trabajadores que lleva a la mayor puntuación global, cumpliendo con el nivel de servicio. Existen muchas formas diferentes de hacer la asignación de las vacaciones, y cada forma de asignar tiene una puntuación total diferente. El objetivo de vVacation es encontrar la asignación de mayor puntuación en poco tiempo de cálculo. Para hacerlo, vVacation utiliza técnicas de Investigación Operativa. La gran ventaja de uso de vVacation es que el programador que la utiliza, es capaz de encontrar el mejor reparto de vacaciones, sin necesidad de saber sobre esas técnicas específicas.

Para mostrar la naturaleza de crecimiento exponencial del número de soluciones que presenta este problema, a continuación se muestran un número muy reducido de soluciones, donde se observan diferentes puntuaciones, para mismo nivel de servicio. La Tabla 2-3 muestra una posible asignación de vacaciones donde se respeta el nivel de servicio porque las vacaciones asignadas cada semana son menores o iguales al número máximo de trabajadores que pueden estar de vacaciones. Las celdas con fondo de color azul representan las vacaciones asignadas. En esta asignación de vacaciones, al trabajador T0 se le asignan las dos primeras semanas de su elección (puntuación de $5+4=9$ puntos), al trabajador T1 se le asignan las primera y cuarta semanas de su elección (puntuación $5+2=7$ puntos), al trabajador T3 no se le asigna ninguna semana de las que había elegido (puntuación de 0 puntos), y al trabajador T4 se le asignan semanas que corresponden a su tercera y cuarta elección (puntuación de $3+2=5$ puntos). Si a un trabajador se le asigna una semana de vacaciones que no está entre sus prioridades, se obtiene una puntuación de 0 puntos en esa asignación. La puntuación total de la asignación de vacaciones es la suma de las puntuaciones de cada trabajador. De esta forma, la asignación de vacaciones que muestra la Tabla 2-3 tiene una puntuación total de $9+7+7=21$ puntos. Cuanto mayor sea la puntuación de una asignación de vacaciones, mejor será esa asignación, porque hay más trabajadores que van a disfrutar de sus vacaciones en los periodos que ha elegido. La forma de comparar la calidad de las diferentes asignaciones de vacaciones es comparar su puntuación; a mayor puntuación, mejor asignación de vacaciones. De esta forma, resolver el problema de asignación de vacaciones consiste en encontrar la asignación de vacaciones que maximice el valor de la puntuación total.



	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	Puntuación
T0		1	2			3	4						9
T1					1	2				3		4	7
T2			3	4					1	2			0
T3			1	2				3	4				5
Demanda	0	1	3	2	1	2	1	1	2	2	0	1	Puntuación total 21 puntos
Vacaciones	1	1	1	1	1	1	1	2	2	2	3	3	
Asignadas	0	1	1	1	1	1	1	2	2	0	3	3	

Tabla 2-3. Asignación de vacaciones con 21 puntos

	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	Puntuación
T0		1	2			3	4						7
T1					1	2				3		4	12
T2			3	4					1	2			9
T3			1	2				3	4				9
Demanda	0	1	3	2	1	2	1	1	2	2	0	1	Puntuación total 37 puntos
Vacaciones	1	1	1	1	1	1	1	2	2	2	3	3	
Asignadas	1	1	1	1	1	1	1	1	2	2	2	2	

Tabla 2-4. Asignación de vacaciones con 37 puntos

La Tabla 2-4 muestra otra posible asignación de vacaciones mejor que la asignación de Tabla 2-3, porque tiene mayor puntuación global. Dentro de la gran cantidad de asignaciones de vacaciones que se pueden crear, la mejor solución es la que muestra la Tabla 2-5, con 37 puntos, y la peor asignación de vacaciones sería la que muestra la Tabla 2-6, con cero puntos totales, porque ningún empleado disfruta de vacaciones en ningún empleado disfruta de vacaciones donde las seleccionó.

Como conclusión:

- Existen muchas formas posibles de asignar las vacaciones a los empleados, para dar el mismo nivel de servicio.
- La calidad de una asignación de vacaciones se puede medir por la puntuación que obtiene cada trabajador.
- El objetivo del problema de asignación de vacaciones es encontrar el reparto de vacaciones que obtiene la mayor puntuación total.



	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	Puntuación
T0		1	2			3	4						7
T1					1	2				3		4	12
T2			3	4					1	2			14
T3			1	2				3	4				5
Demanda	0	1	3	2	1	2	1	1	2	2	0	1	Puntuación total 38 puntos
Vacaciones	1	1	1	1	1	1	1	2	2	2	3	3	
Asignadas	1	1	1	1	1	1	1	2	2	2	2	2	

Tabla 2-5. Mejor solución. Puntuación de 37 puntos.

	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	Puntuación
T0		1	2			3	4						0
T1					1	2				3		4	0
T2			3	4					1	2			0
T3			1	2				3	4				0
Demanda	0	1	3	2	1	2	1	1	2	2	0	1	Puntuación total 0 puntos
Vacaciones	1	1	1	1	1	1	1	2	2	2	3	3	
Asignadas	1	1	1	1	1	1	1	2	2	2	2	2	

Tabla 2-6. Peor solución. Puntuación de 0 puntos.



3 Manual de usuario

3.1 Introducción

vVacation es una librería C++ para resolver problemas de asignación de vacaciones en colectivos de personal. vVacation también se entrega como un ensamblado .NET y como un componente ejecutable, de forma que puede integrarse de forma sencilla con los sistemas empresariales del cliente que la adquiera.

En este capítulo se explica todo lo necesario para poder utilizar vVacation. También se muestra el contenido de vVacation, y se explica su uso, mediante un ejemplo sencillo.

3.2 Contenido

Al adquirir una licencia de vVacation, Infozara envía al cliente la biblioteca de desarrollo vVacation, con todo lo necesario para desarrollar aplicaciones con procesos de asignación de vacaciones a colectivos de personal.

La biblioteca vVacation está escrita en ANSI C++, y no está concebida para ningún entorno de desarrollo ni sistema operativo en especial. El lenguaje C++ se puede compilar y ejecutar en todos los sistemas operativos. vVacation está escrita en C++ estándar, y solo el proceso de verificación de licencias hace llamadas al sistema operativo.

Desde la versión v1.1, vVacation es compatible con el desarrollo sobre la plataforma .NET de Microsoft. Desde la versión v1.2, vVacation se entrega como componente ejecutable que se puede invocar desde cualquier plataforma o sistema operativo. A lo largo de este capítulo se explicará el uso de vVacation con estas dos nuevas formas de entrega.

3.3 Modelo conceptual de vVacation

El modelo conceptual de la biblioteca vVacation es muy sencillo, debido a que tiene un carácter muy específico, y únicamente utiliza la información estrictamente necesaria de cara al usuario, aunque interiormente presente una cierta complejidad, de la que no debe tener conocimiento el usuario. El modelo conceptual se muestra en la Figura 3-1 y se lee de la siguiente forma:

‘Un empleado viene definido por su nombre, la cantidad de semanas seguida que disfruta en cada bloque de vacaciones, la cantidad de semanas de vacaciones que debe disfrutar, y está asociado a una temática. El empleado selecciona por orden de prioridad las semanas que quiere disfrutar de vacaciones. Un grupo de temáticas engloba a trabajadores de varias temáticas, y establece la cantidad máxima de trabajadores que pueden estar de vacaciones en cada semana. Hay un solver que resuelve el problema, que consiste en encontrar las semanas de vacaciones de cada trabajador, de forma que se asignan las preferencias de los trabajadores en la mayor medida posible, respetando las condiciones marcadas por los grupos de temáticas. vVacation proporciona el solver que encuentra la mejor asignación de vacaciones.’

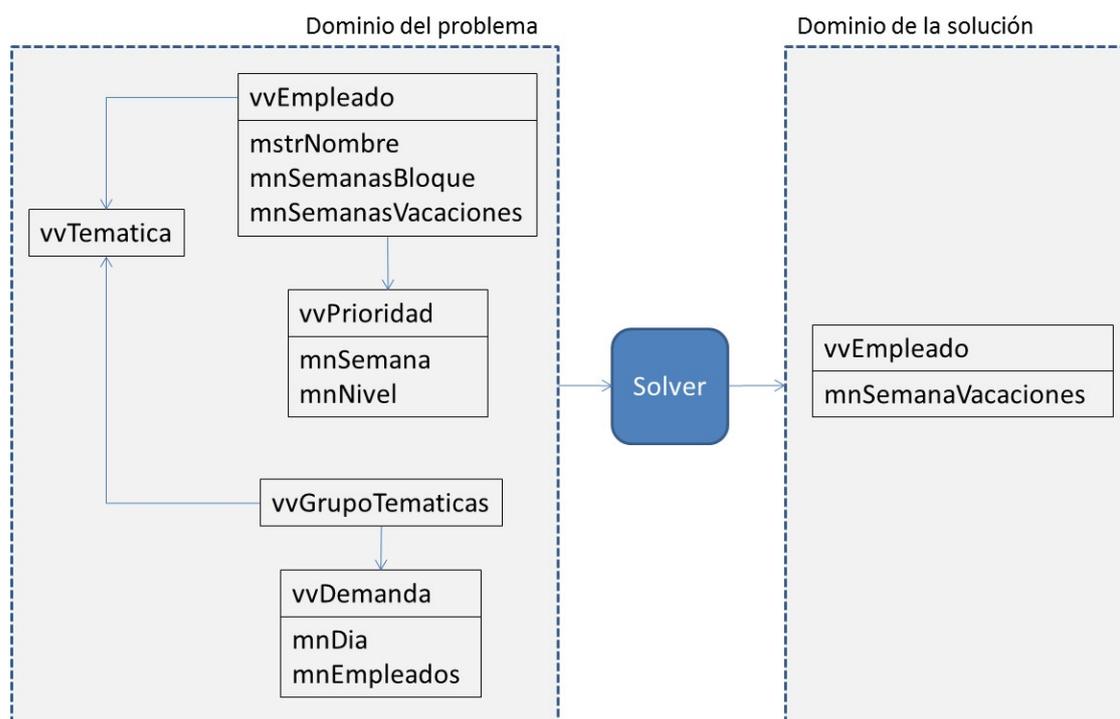


Figura 3-1. Modelo conceptual de vVacation

3.4 Instalación

La biblioteca vVacation se distribuye como un fichero comprimido con el código fuente y la documentación, según la siguiente estructura:

- \include Contiene los ficheros de cabecera .h.
- \lib Contiene le fichero de biblioteca vVacationLib.lib para desarrollar programas C++, el ensamblado vVacationWrapper.dll, para desarrollar programas sobre la plataforma .NET de Microsoft, y el ejecutable vVacationSolver.exe, que es el ejecutable que resuelve el problema de asignación de vacaciones. En este directorio también aparecen dll que consume el ejecutable.
- \doc Contiene el manual de usuario de vVacation.

La instalación consiste en descomprimir el fichero para obtener la estructura anterior, y copiar los ficheros al directorio de proyecto de desarrollo. Posteriormente, el programador que hace uso de vVacation podrá copiar los ficheros de la instalación a las estructura de proyecto que considere oportunos.

3.5 Licencias

vVacation se licencia por cada ordenador donde se utiliza la biblioteca. Las licencias van asociadas a la mac address de cada ordenador.

El proceso de validación de licencias es el siguiente:

- Identificar la MAC ADDRESS del ordenador donde se va a instalar la licencia. Para obtener la MAC ADDRESS en un ordenador son sistema operativo Windows, se debe



teclea `ipconfig /all` en la línea de comandos, y observar el valor 'Physical Address'.

- Enviar la MAC ADDRESS al servicio técnico de Infozara.
- El servicio técnico de Infozara enviará al cliente un fichero de licencia (.lic).
- Colocar el fichero de licencia en el directorio donde esté cada ejecutable que incluye a la librería vVacation.

3.6 Configuración de un proyecto C++

Para poder hacer uso de vVacation en un proyecto C++:

- Se deben añadir las directivas `#include` con los ficheros .h correspondientes a las clases de vVacation.
- Se debe añadir al proyecto la biblioteca `vVacationLib.lib`, y asegurar que el proyecto enlaza con ella.

3.7 Configuración de un proyecto .NET

Para poder hacer uso de vVacation en un proyecto .NET:

- Se debe añadir referencia al ensamblado `vVacationWrapper.dll`, que contiene clases escritas en C++ administrado, para su uso como cualquier ensamblado .NET.

Al crear la referencia al ensamblado, se tiene acceso al namespace `vVacationWrapper`.

3.8 Ejemplo de programación en C++

En este apartado se va a mostrar el uso de la biblioteca vVacation en una aplicación C++, a través de un sencillo ejemplo, que resuelve el problema planteado en el apartado 3.2.

En la línea 1 se crea el solver de vVacation, con 12 semanas de programación de vacaciones.

El primer paso para resolver un problema de asignación de vacaciones con vVacation es definir las temáticas, que son la relación entre la cantidad de vacaciones que se puede asignar y los empleados a tener en cuenta en esa restricción de cantidad de vacaciones. En el ejemplo de este apartado solo se crea una temática, para mostrar un ejemplo sencillo. La temática se crea en la línea 5.

El siguiente paso es definir un grupo de temáticas, que es una restricción de cantidad de trabajadores que pueden estar de vacaciones. En la línea 8 se crea el grupo de temáticas, y se establece que por defecto, existirá un máximo de un trabajador de vacaciones en cada semana del periodo de 12 semanas. En la línea 9 se indica que la restricción de cantidad de trabajadores de vacaciones se asocia a trabajadores de la temática creada en la línea 5. Entre las líneas 10 y 14 se modifica la cantidad de trabajadores de vacaciones por defecto, a diversos valores. Por ejemplo, en la línea 12 se especifica que en la semana 9 pueden estar de vacaciones un máximo de dos trabajadores.

El siguiente paso es definir los empleados, que se hace entre las líneas 17 y 20. Para definir un empleado son necesarios cuatro parámetros: el nombre, la temática a la que pertenece, los bloques de semanas de vacaciones que se le deben asignar, y la cantidad de semanas de vacaciones. Los cuatro empleados creados en este ejemplo pertenecen a la única temática



definida, fijan periodos de vacaciones de dos semanas, y debe tener un total de cuatro semanas de vacaciones dentro del periodo de doce semanas. Una vez creados los empleados, entre las líneas 23 y 41 se indican las semanas de vacaciones preferidas por cada empleado. Como ejemplo, en la línea 28 se indica que la semana preferida por el empleado **T1** es la semana número 4, y en la línea 35 se indica que la tercera mejor semana para el empleado **T2** es la semana número 2.

Una vez definidas las temáticas, los grupos de temáticas y los trabajadores, se puede resolver el problema, que se hace en la línea número 45. La variable `dObjValue` contiene el valor de la puntuación total una vez resuelto el problema. La variable `nStatus` indica si se ha resuelto correctamente el problema. Si el valor de esta variable es igual a `Optimal_Solution_Found`, significa que el algoritmo que implementa `vVacation` ha encontrado la solución óptima del problema de asignación de vacaciones.

En la línea 51 se escribe un fichero XML con la solución del problema, y entre las líneas 54 y 82 se muestra el resultado por pantalla. Para acceder a las vacaciones de cada empleado, se utiliza la función `vacaciones()` de la clase `vvEmpleado`, que se hace en las líneas 60, 66, 72 y 78, una vez por empleado.

```
1 // Crea el solver para el problema de asignación de vacaciones
2 vvSolverVacaciones oSolver(12);
3
4 // Crea una temática, que es la residencia de los empleados
5 vvTematica* pMadrid = oSolver.addTematica(string("Madrid"));
6
7 // Crea una grupo de temáticas para definir la cantidad de vacaciones
8 vvGrupoTematicas* pResidencia = oSolver.addGrupoTematicas( 1 );
9 pResidencia->addTematica(pMadrid);
10 pResidencia->setVacationDemand(7, 2);
11 pResidencia->setVacationDemand(8, 2);
12 pResidencia->setVacationDemand(9, 2);
13 pResidencia->setVacationDemand(10, 3);
14 pResidencia->setVacationDemand(11, 3);
15
16 // Crea los empleados
17 vvEmpleado* pT0 = oSolver.addEmpleado( "T0", pMadrid, 2, 4 );
18 vvEmpleado* pT1 = oSolver.addEmpleado( "T1", pMadrid, 2, 4 );
19 vvEmpleado* pT2 = oSolver.addEmpleado( "T2", pMadrid, 2, 4 );
20 vvEmpleado* pT3 = oSolver.addEmpleado( "T3", pMadrid, 2, 4 );
21
22 // Fija las preferencias de vacaciones de cada empleado
23 pT0->addPreferencia( 1, 1);
24 pT0->addPreferencia( 2, 2);
25 pT0->addPreferencia( 5, 3);
26 pT0->addPreferencia( 6, 4);
27
28 pT1->addPreferencia(4, 1);
29 pT1->addPreferencia(5, 2);
30 pT1->addPreferencia(9, 3);
31 pT1->addPreferencia(11, 4);
32
33 pT2->addPreferencia(8, 1);
34 pT2->addPreferencia(9, 2);
35 pT2->addPreferencia(2, 3);
36 pT2->addPreferencia(3, 4);
37
38 pT3->addPreferencia(2, 1);
39 pT3->addPreferencia(3, 2);
```



```
40 pT3->addPreferencia(7, 3);
41 pT3->addPreferencia(8, 4);
42
43 // Resuelve el problema
44 double dObjValue = 0;
45 int nStatus = oSolver.solve(dObjValue);
46
47 // Comprueba el estado de resolución
48 if (nStatus == Optimal_Solution_Found)
49 {
50     // Escribe la solución en un fichero XML
51     oSolver.escribirSolucion(dObjValue, string("SalidaVacaciones.xml"));
52
53     // Escribe la solución por pantalla
54     vector<int> vSemanas;
55     vector<int>::iterator its;
56
57     std::cout << std::endl << std::endl
58               << "Vacaciones de los empleados: ";
59
60     vSemanas = pT0->vacaciones();
61     std::cout << std::endl << "T0 -> Puntuacion: "
62               << pT0->puntuacion() << " Vacaciones: ";
63     for (its = vSemanas.begin(); its != vSemanas.end(); its++)
64         std::cout << (*its) << " ";
65
66     vSemanas = pT1->vacaciones();
67     std::cout << std::endl << "T1 -> Puntuacion: "
68               << pT1->puntuacion() << " Vacaciones: ";
69     for (its = vSemanas.begin(); its != vSemanas.end(); its++)
70         std::cout << (*its) << " ";
71
72     vSemanas = pT2->vacaciones();
73     std::cout << std::endl << "T2 -> Puntuacion: "
74               << pT2->puntuacion() << " Vacaciones: ";
75     for (its = vSemanas.begin(); its != vSemanas.end(); its++)
76         std::cout << (*its) << " ";
77
78     vSemanas = pT3->vacaciones();
79     std::cout << std::endl << "T3 -> Puntuacion: "
80               << pT3->puntuacion() << " Vacaciones: ";
81     for (its = vSemanas.begin(); its != vSemanas.end(); its++)
82         std::cout << (*its) << " ";
83 }
```

3.9 Ejemplo de programación en .NET

En este apartado se va a mostrar el uso de la biblioteca vVacation en una aplicación C#, a través de un sencillo ejemplo, que resuelve el problema planteado en el apartado 3.2.

Una vez creada la referencia al ensamblado vVacationWrapper.dll, se dispone de acceso al espacio de nombres vVacationWrapper. En este espacio de nombre existen las clases públicas, que se utilizan para resolver un problema de asignación de vacaciones.

En la línea 5 se crea el solver de vVacation, con 12 semanas de programación de vacaciones. El solver está encapsulado en la clase `vvcppVacationSolver`.

El primer paso para resolver un problema de asignación de vacaciones con vVacation es



definir las temáticas, que son la relación entre la cantidad de vacaciones que se puede asignar y los empleados a tener en cuenta en esa restricción de cantidad de vacaciones. En el ejemplo de este apartado solo se crea una temática, para mostrar un ejemplo sencillo. La temática se crea en la línea 8. Una temática viene definida por la clase `vvcppTematica`.

El siguiente paso es definir un grupo de temáticas, que es una restricción de cantidad de trabajadores que pueden estar de vacaciones. En la línea 8 se crea el grupo de temáticas, y se establece que por defecto, existirá un máximo de un trabajador de vacaciones en cada semana del periodo de 12 semanas. En la línea 11 se indica que la restricción de cantidad de trabajadores de vacaciones se asocia a trabajadores de la temática creada en la línea 9. Entre las líneas 13 y 17 se modifica la cantidad de trabajadores de vacaciones por defecto, a diversos valores. Por ejemplo, en la línea 15 se especifica que en la semana 9 pueden estar de vacaciones un máximo de dos trabajadores. Un grupo de temáticas viene definido por la clase `vvcppGrupoTematicas`.

El siguiente paso es definir los empleados, que se hace entre las líneas 20 y 23. Para definir un empleado son necesarios cuatro parámetros: el nombre, la temática a la que pertenece, los bloques de semanas de vacaciones que se le deben asignar, y la cantidad de semanas de vacaciones. Los cuatro empleados creados en este ejemplo pertenecen a la única temática definida, fijan periodos de vacaciones de dos semanas, y debe tener un total de cuatro semanas de vacaciones dentro del periodo de doce semanas. Una vez creados los empleados, entre las líneas 26 y 44 se indican las semanas de vacaciones preferidas por cada empleado. Como ejemplo, en la línea 31 se indica que la semana preferida por el empleado **T1** es la semana número 4, y en la línea 39 se indica que la tercera mejor semana para el empleado **T2** es la semana número 2.

Una vez definidas las temáticas, los grupos de temáticas y los trabajadores, se puede resolver el problema, que se hace en la línea número 48. La variable `dObjValue` contiene el valor de la puntuación total una vez resuelto el problema. La variable `nStatus` indica si se ha resuelto correctamente el problema. Si el valor de esta variable es igual a `Optimal_Solution_Found`, significa que el algoritmo que implementa `vVacation` ha encontrado la solución óptima del problema de asignación de vacaciones.

En la línea 54 se escribe un fichero XML con la solución del problema, y entre las líneas 59 y 824 se muestra el resultado por pantalla. Para acceder a las vacaciones de cada empleado, se utiliza la función `vacaciones()` de la clase `vvEmpleado`, que se hace en las líneas 59, 66, 73 y 77, una vez por empleado.

```
1 using vVacationWrapper;
2
3
4 // Crea el solver para el problema de asignación de vacaciones
5 vvcppVacationSolver oSolver = new vvcppVacationSolver(2);
6
7 // Crea una temática, que es la residencia de los empleados
8 vvcppTematica oMadrid = oSolver.addTematica("Madrid");
9
10 // Crea un grupo de temáticas para definir la cantidad de vacaciones
11 vvcppGrupoTematicas oResidencias = oSolver.addGrupoTematicas(1);
12 oResidencias.addTematica(oMadrid);
13 oResidencias.setVacationDemand(7, 2);
14 oResidencias.setVacationDemand(8, 2);
15 oResidencias.setVacationDemand(9, 2);
16 oResidencias.setVacationDemand(10, 3);
17 oResidencias.setVacationDemand(11, 3);
18
```



```
19 // Crea los empleados
20 vvcppEmpleado oT0 = oSolver.addEmpleado("T0", oMadrid, 2, 4);
21 vvcppEmpleado oT1 = oSolver.addEmpleado("T1", oMadrid, 2, 4);
22 vvcppEmpleado oT2 = oSolver.addEmpleado("T2", oMadrid, 2, 4);
23 vvcppEmpleado oT3 = oSolver.addEmpleado("T3", oMadrid, 2, 4);
24
25 // Fija las preferencias de vacaciones de cada empleado
26 oT0.addPreferencia(1, 1);
27 oT0.addPreferencia(2, 2);
28 oT0.addPreferencia(5, 3);
29 oT0.addPreferencia(6, 4);
30
31 oT1.addPreferencia(4, 1);
32 oT1.addPreferencia(5, 2);
33 oT1.addPreferencia(9, 3);
34 oT1.addPreferencia(11, 4);
35
36 oT2.addPreferencia(8, 1);
37 oT2.addPreferencia(9, 2);
38 oT2.addPreferencia(2, 3);
39 oT2.addPreferencia(3, 4);
40
41 oT3.addPreferencia(2, 1);
42 oT3.addPreferencia(3, 2);
43 oT3.addPreferencia(7, 3);
44 oT3.addPreferencia(8, 4);
45
46 // Resuelve el problema
47 double dObjValue = 0;
48 int nStatus = oSolver.solve( ref dObjValue);
49
50 // Comprueba el estado de resolución
51 if (nStatus == vvcppDefiniciones.Optimal_Solution)
52 {
53     // Escribe la solución en un fichero XML
54     oSolver.escribirSolucion(dObjValue, "SalidaVacaciones.xml");
55
56     // Escribe la solución por pantalla
57     List<int> lstSemanas = new List<int>();
58
59     oT0.vacaciones(ref lstSemanas);
60     Console.WriteLine("\nT0: Puntuacion" + oT0.puntuacion() + " Vacaciones: ");
61     foreach (int nSemana in lstSemanas)
62         Console.WriteLine(nSemana + " ");
63
64     lstSemanas.Clear();
65     oT1.vacaciones(ref lstSemanas);
66     Console.WriteLine("\nT1: Puntuacion" + oT1.puntuacion() + " Vacaciones: ");
67     foreach (int nSemana in lstSemanas)
68         Console.WriteLine(nSemana + " ");
69
70     lstSemanas.Clear();
71     oT2.vacaciones(ref lstSemanas);
72     Console.WriteLine("\nT2: Puntuacion" + oT2.puntuacion() + " Vacaciones: ");
73     foreach (int nSemana in lstSemanas)
74         Console.WriteLine(nSemana + " ");
75
76     lstSemanas.Clear();
77     oT3.vacaciones(ref lstSemanas);
78     Console.WriteLine("\nT3: Puntuacion" + oT3.puntuacion() + " Vacaciones: ");
```



```
79     foreach (int nSemana in lstSemanas)
80         Console.WriteLine(nSemana + " ");
81 }
```

3.10 Componente ejecutable

vVacation también está disponible como componente ejecutable que puede ser invocado desde cualquier aplicación. El componente es un programa ejecutable que recibe tres parámetros en la línea de comandos: el path del fichero con los datos del problema, el path del fichero donde debe dejar la solución, y el path de un fichero log que indica la evolución de la resolución del problema.

3.10.1 Uso del componente ejecutable

A continuación se muestra como ejemplo el código fuente escrito en .NET que invoca al componente ejecutable vVacationSolver.exe. En la siguiente figura se muestra el código fuente, y se comenta a continuación, haciendo referencia a los números de línea.

En la línea se incluye la referencia al espacio de nombre que contiene a las clases necesarias para la ejecución de una aplicación. En la línea 6 se define el path del fichero que contiene la definición del problema de asignación de vacaciones. En la línea 7 se crea el fichero. Típicamente, crearFicheroProblemaVacaciones es una función que accedes a alguna base de datos o sistema empresarial, para escribir toda la información del problema de asignación de vacaciones en el fichero de path strEntrada. En las líneas 11 y 12 se definen los path de los ficheros de solución y de logs, respectivamente. Entre las líneas 14 y 16 se crea la información de llamada al proceso de ejecución, que se encapsula en la clase .NET ProcessStartInfo. En concreto, en la línea 14 se crea la instancia de esta clase, en la línea 15 se indica el path del ejecutable que se va a lanzar, en este caso es el solver de vVacation, de nombre vVacationSolver.exe. En la línea 17 se especifican los parámetros que se van a pasar en la llamada al ejecutable. El argumento es la secuencia de los tres paths de los tres ficheros, separados por un espacio.

En línea número 24, se lanza el programa ejecutable y en la línea 26 se indica que se debe esperar a que termine su ejecución ejecutable. Cuando se ejecuta el código de la línea 28, ya se ha creado la solución, y lo que queda es leer el fichero de la solución y volcar su contenido en los sistemas empresariales.

```
1  using System.Diagnostics;
2
3
4
5  // Crea el fichero XML con el problema
6  string strEntrada = "vVacationProblem.xml"
7  crearFicheroProblemaVacaciones(strEntrada);
8
9  // Crea el path de los ficheros de resultados
10 string strSalida = "vVacationSolution.xml";
11 string strLog    = "vVacationLog.xml";
12
13 // Ejecuta el programa para resolver el problema
14 ProcessStartInfo psi= new ProcessStartInfo();
15 psi.FileName      = "vVacationSolver.exe";
16 psi.UseShellExecute = false;
17 psi.Arguments     = strEntrada + " " + strSalida + " " + strLog;
```



```
18
19 Process p = new System.Diagnostics.Process();
20 p.StartInfo = psi;
21
22 try
23 {
24     p.Start();
25     p.WaitForExit();
26
27     // Guarda la solución
28     guardarSolucionVacaciones( strSalida );
29 }
30 catch (Exception ex)
31 {
32     MessageBox.Show(ex.Message, "Error en la llamada al solver");
33 }
```

3.10.2 Características de objetos y campos del problema

La Tabla 3-8 muestra las características de cada uno de los objetos y campos de problema de asignación de vacaciones (objeto raíz `Vacation_Problem`).

Objeto o campo	Requerido	Vacío	Cardinalidad
<code>Vacation_Problem</code>	SI	NO	1
<code>Tematics</code>	SI	NO	1
<code>Tematic</code>	SI	NO	1 o más
<code>Staff</code>	SI	NO	1
<code>Employee</code>	SI	NO	1 o más
<code>Name</code>	SI	NO	1
<code>EmployeeTematic</code>	SI	NO	1
<code>WeeksPerBlock</code>	SI	NO	1
<code>WeeksBetweenBlocks</code>	SI	NO	1
<code>WeeksOnVacation</code>	SI	NO	1
<code>Options</code>	SI	NO	1
<code>Option</code>	SI	NO	1 o más
<code>WeekNumber</code>	SI	NO	1
<code>Level</code>	SI	NO	1
<code>TematicsGroups</code>	SI	NO	1
<code>TematicsGroup</code>	SI	NO	1 o más
<code>TematicsOnGroup</code>	SI	NO	1
<code>TematicOnGroup</code>	SI	NO	1 o más
<code>VacationsDemands</code>	SI	NO	1
<code>VacationDemand</code>	SI	NO	1 o más
<code>WeekNumber</code>	SI	NO	1
<code>MaxVacation</code>	SI	NO	1

Tabla 3-1. Características de objetos y campos del problema
Características de objetos y campos de la solución

La Tabla 3-9 muestra las características de cada uno de los objetos y campos de la solución del problema de asignación de vacaciones (objeto raíz `Vacation_Solution`).



Objeto o campo	Requerido	Vacío	Cardinalidad
Vacation_Solution	SI	NO	1
Result	SI	NO	1
Status	SI	NO	1
ObjectiveValue	SI	NO	1
SocialEquality	SI	NO	1
Vacation	SI	NO	1
EmployeeVacation	SI	NO	1 o más
Name	SI	NO	1
VacationWeeks	SI	NO	1
VacationWeek	SI	NO	1 o más
Score	SI	NO	1

Tabla 3-2. Características de objetos y campos de la solución



4 Manual de referencia C++

Este capítulo hace referencia a todas las clases públicas de la biblioteca C++ vVacation.

4.1 vSolverVacaciones

Categoría Clase de desarrollo.

Descripción `vvSolverVacaciones` es la clase que almacena toda la información que define al problema de asignación de vacaciones, y ejecuta el algoritmo que lo resuelve.

Fichero include

```
class vvSolverVacaciones
{
public:
vvSolverVacaciones( int nSemanas );
~vvSolverVacaciones();

vvTematica* addTematica( string& strNombre);
void addTematica(vvGrupoTematicas* pGrupo, string& strTematica);
vvEmpleado* addEmpleado( string strNombre, string& strTematica,
int nSemanasSeguidas, int nSemanasVacaciones);
vvEmpleado* addEmpleado( string strNombre, vvTematica* pTematica,
int nSemanasSeguidas, int nSemanasVacaciones );
vvGrupoTematicas* addGrupoTematicas(
int nMaximosTrabajadoresDeVacaciones);

int solve( double& dValue );

void escribirSolucion(int nIgualdadSocial, string& strFichero);
};
```

Constructores `vvSolverVacaciones(int nSemanas);`

El constructor recibe como parámetro el número de semanas de planificación.

Funciones miembro públicas

`vvTematica* addTematica(string& strNombre);`

Crea la temática cuyo nombre recibe como argumento.

`void addTematica(vvGrupoTematicas* pGrupo, string& strTematica);`

Añade una temática a un grupo de temáticas. La temática de nombre `strTematica` debe estar previamente definida.



```
vvEmpleado* addEmpleado( string strNombre, string& strTematica,  
                          int nSemanasSeguidas, int nSemanasVacaciones);
```

Añade un empleado, definido por su nombre o identificador, la temática asociada al empleado, el número de semanas de cada bloque de vacaciones que se asignan al empleado, y el número total de semanas de vacaciones que se deben asignar al empleado. La temática de nombre `strTematica` debe estar previamente definida.

```
vvEmpleado* addEmpleado( string strNombre, vvTematica* pTematica,  
                          int nSemanasSeguidas, int nSemanasVacaciones );
```

Añade un empleado, definido por su nombre o identificador, la temática asociada al empleado, el número de semanas de cada bloque de vacaciones que se asignan al empleado, y el número total de semanas de vacaciones que se deben asignar al empleado.

```
vvGrupoTematicas* addGrupoTematicas(  
                          int nMaximosTrabajadoresDeVacaciones);
```

Crea un grupo de temáticas, que definen una carga máxima de trabajadores que pueden estar de vacaciones en cada semana del periodo.

```
int solve( double& dValue );
```

Resuelve el problema de asignación de vacaciones.

```
void escribirSolucion(int nIgualdadSocial, string& strFichero);
```

Escribe en un fichero la solución del problema de asignación de vacaciones. `nIgualdadSocial` es la puntuación total obtenida en el problema de asignación de vacaciones, y `strFichero` es el path del fichero donde se escribe la solución del problema.

4.2 vvTematica

Categoría Clase de desarrollo.

Descripción `vvTematica` representa una temática dentro del problema de asignación de vacaciones.

Fichero include

```
class vvTematica  
{  
public:  
  
    vvTematica(string& strNombre);  
    virtual ~vvTematica()  
  
    string& nombre(void);
```



```
        bool es(vvTematica* pTematica);  
};
```

Constructores vvTematica(string& strNombre);

Una temática viene definida únicamente por su nombre.

Funciones miembro públicas string& nombre(void);

Devuelve una referencia al nombre de la temática.

```
bool es(vvTematica* pTematica);
```

Comprueba si la temática que recibe como argumento corresponde a la instancia de temática.

4.3 vvEmpleado

Categoría Clase de desarrollo.

Descripción vvEmpleado representa un empleado al que se asignan vacaciones.

Fichero include class vvEmpleado
{
public:

```
    vvEmpleado(string strNombre, vvTematica* pTematica,  
               int nSemanasSeguidas, int nSemanasVacaciones)  
  
    virtual ~vvEmpleado()  
  
    Preferencia* addPreferencia( int nSemana,int nNivelPrioridad );  
  
    string& nombre(void);  
  
    void saveSolution();  
  
    vector<int>& vacaciones(void);  
    int puntuacion(void) const;  
  
    void escribirSolucion(ofstream& os);  
};
```

Constructores vvEmpleado(string strNombre, vvTematica* pTematica,
int nSemanasSeguidas, int nSemanasVacaciones)

Un empleado viene definido por su nombre strNombre y está asociado a una temática pTematica. El parámetro nSemanasSeguidas indica la cantidad de semanas que tiene cada periodo de vacaciones que se va a fijar al



trabajador. El parámetro `nSemanasVacaciones` indica la cantidad total de semanas de vacaciones que se van a fijar al trabajador.

Funciones miembro públicas

`Preferencia*` `addPreferencia(int nSemana,int nNivelPrioridad);`

Añade una preferencia de semana de vacaciones al empleado. El parámetro `nSemana` indica la semana, y el parámetro `nNivelPrioridad` indica el nivel de prioridad de esa semana de vacaciones para el trabajador.

`string&` `nombre(void);`

Devuelve el nombre o identificador del trabajador.

`vector<int>&` `vacaciones(void);`

Devuelve un vector con las semanas de vacaciones asignadas tras resolver el problema de asignación de vacaciones.

`int` `puntuacion(void) const;`

Devuelve la puntuación que ha obtenido el trabajador en la asignación de vacaciones.

`void` `escribirSolucion(ofstream& os);`

Escribe en un fichero la asignación de vacaciones del empleado. La información se escribe en formato XML.

4.4 vvGrupoTematicas

Categoría Clase de desarrollo.

Descripción `vvGrupoTematicas` representa la restricción de cantidad de trabajadores de ciertas temáticas que pueden estar de vacaciones en cada semana.

Fichero include

```
class vvGrupoTematicas
{
public:

    vvGrupoTematicas(int nSemanas,
                    int nTrabajadoresDeVacacionesPorSemana );

    virtual ~vvGrupoTematicas();

    void addTematica( vvTematica* pTematica );

    void setVacationDemand(int nWeek, int nMaxOnVacation);

    void addEmpleado(vvEmpleado* pEmpleado);
};
```



Constructores `vvGrupoTematicas(int nSemanas,
int nTrabajadoresDeVacacionesPorSemana);`

Un grupo de temáticas se define para una cantidad de semanas, y define por defecto una cantidad máxima de trabajadores de vacaciones igual para todas las semanas.

Funciones miembro públicas `void addTematica(vvTematica* pTematica);`

Añade una temática al grupo de temáticas.

`void setVacationDemand(int nWeek, int nMaxOnVacation);`

Fija la cantidad `nMaxOnVacation` de trabajadores de vacaciones en la semana concreta `nWeek`.

4.5 vvSolverVacaciones

Categoría Clase de desarrollo.

Descripción `vvSolverVacaciones` calcula asignación de vacaciones de los empleados según sus preferencias y las restricciones marcadas por los grupos de temáticas. Esta clase contiene toda la información del problema de asignación de vacaciones. Esta clase crea las instancias de las clases `vvTematica`, `vvEmpleado`, y `vvGrupoTematicas`, y también es responsable de liberar la memoria reservada por las instancias de estas clases.

Fichero include

```
class vvSolverVacaciones
{
public:
    vvSolverVacaciones( int nSemanas );
    ~vvSolverVacaciones();

    vvTematica* addTematica( string& strNombre);

    void addTematica(vvGrupoTematicas* pGrupo,
                    string& strTematica);

    vvEmpleado* addEmpleado( string strNombre,
                             string& strTematica,
                             int nSemanasSeguidas,
                             int nSemanasVacaciones);

    vvEmpleado* addEmpleado(string strNombre,
                             vvTematica* pTematica,
                             int nSemanasSeguidas,
                             int nSemanasVacaciones );
```



```
vvGrupoTematicas* addGrupoTematicas(  
    int nMaximosTrabajadoresDeVacaciones);  
  
int solve( double& dValue );  
  
void escribirSolucion(int nIgualdadSocial, string& strFichero);  
};
```

Constructores

```
vvSolverVacaciones( int nSemanas );
```

El solver que resuelve el problema de asignación de vacaciones se define para una cantidad de semanas. Después se deben crear en el solver las temáticas, los empleados y los grupos de temáticas.

Funciones miembro públicas

```
vvTematica* addTematica( string& strNombre);
```

Crea una temática de nombre strNombre.

```
void addTematica( vvGrupoTematicas* pGrupo,  
    string& strTematica);
```

Añade la temática de nombre strTematica al grupo de temáticas pGrupo.

```
vvEmpleado* addEmpleado( string strNombre,  
    string& strTematica,  
    int nSemanasSeguidas,  
    int nSemanasVacaciones);
```

Crea el empleado definido por el nombre strNombre, asociado a la temática strTematica, que agrupa las vacaciones en bloque de nSemanasSeguidas semanas, y disfruta de nSemanasVacaciones semanas de vacaciones.

```
vvEmpleado* addEmpleado( string strNombre,  
    vvTematica* pTematica,  
    int nSemanasSeguidas,  
    int nSemanasVacaciones );
```

Crea el empleado definido por el nombre strNombre, asociado a la temática pTematica, que agrupa las vacaciones en bloque de nSemanasSeguidas semanas, y disfruta de nSemanasVacaciones semanas de vacaciones.

```
vvGrupoTematicas* addGrupoTematicas(  
    int nMaximosTrabajadoresDeVacaciones);
```

Crea un grupo de temáticas que define por defecto la cantidad de nMaximosTrabajadoresDeVacaciones para todas las semanas.

```
int solve( double& dValue );
```

Resuelve el problema de asignación de vacaciones. En el parámetro dValue se almacena el nivel de prioridad total alcanzado en la solución óptima.

Devuelve el estado de resolución del problema. Si devuelve el valor



`Optimal_Solution_Found`, indica que se ha encontrado la mejor solución al problema. Si devuelve -1 significa que el solver no ha encontrado la solución al problema de asignación de vacaciones.

```
void escribirSolucion(int nIgualdadSocial, string& strFichero);
```

Escribe la solución del problema de asignación de vacaciones en el fichero de path `strFichero`. El parámetro `nIgualdadSocial` es el valor de prioridad total de la asignación de vacaciones.



5 Manual de referencia .NET

Este capítulo hace referencia a todas las clases públicas de la biblioteca .NET vVacationWrapper.

5.1 vvcppVacationSolver

Categoría Clase de desarrollo.

Descripción vvcppVacation es la representación del problema de asignación de vacaciones. Calcula asignación de vacaciones de los empleados según sus preferencias y las restricciones marcadas por los grupos de temáticas. Esta clase contiene toda la información del problema de asignación de vacaciones. Esta clase crea las instancias de las clases [vvcppTematica](#), [vvcppEmpleado](#), y [vvcppGrupoTematicas](#), y también es responsable de liberar la memoria reservada por las instancias de estas clases

Esta clase dispone de todas las funciones necesarias para definir el problema, resolverlo y acceder a la solución.

Es una clase escrita en lenguaje C++ administrado, y es accesible desde cualquier lenguaje de desarrollo administrado de la plataforma .NET.

Fichero include

```
public ref class vvcppVacationSolver
{
public:

    vvcppVacationSolver(int nSemanas)

    virtual ~vvcppVacationSolver(void)

    vvcppTematica^ addTematica(String^ strNombre);

    vvcppEmpleado^ addEmpleado( String^ strNombre,
                                vvcppTematica^ pcppTematica,
                                int nSemanasSeguidas,
                                int nSemanasVacaciones);

    vvcppGrupoTematicas^ addGrupoTematicas(
                                int nMaximosTrabajadoresDeVacaciones );

    int solve(double% dObjValue);

    void escribirSolucion(double dObjValue, String^ strPath);
};
```

Constructores vvcppVacationSolver (int nSemanas);

El constructor se crea para un número de semanas de vacaciones.

**Funciones
miembro
públicas**

```
vcppTematica^ addTematica(String^ strNombre);
```

Añade una temática de nombre `strNombre`.

```
vcppEmpleado^ addEmpleado(String^ strNombre,  
                             vcppTematica^ pcppTematica,  
                             int nSemanasSeguidas,  
                             int nSemanasVacaciones);
```

Crea el empleado definido por el nombre `strNombre`, asociado a la temática, que agrupa las vacaciones en bloque de `nSemanasSeguidas` semanas, y disfruta de `nSemanasVacaciones` semanas de vacaciones.

```
vcppGrupoTematicas^ addGrupoTematicas(  
    int nMaximosTrabajadoresDeVacaciones );
```

Crea un grupo de temáticas que define por defecto la cantidad de `nMaximosTrabajadoresDeVacaciones` para todas las semanas.

```
int solve(double% dObjValue);
```

Resuelve el problema de asignación de vacaciones. En el parámetro `dValue` se almacena el nivel de prioridad total alcanzado en la solución óptima.

Devuelve el estado de resolución del problema. Si devuelve el valor `Optimal_Solution_Found`, indica que se ha encontrado la mejor solución al problema. Si devuelve `-1` significa que el solver no ha encontrado la solución al problema de asignación de vacaciones.

```
void escribirSolucion(double dObjValue, String^ strPath);
```

Escribe la solución del problema de asignación de vacaciones en el fichero de path `strFichero`. El parámetro `nIgualdadSocial` es el valor de prioridad total de la asignación de vacaciones.