

Aplicación de técnicas metaheurísticas a problemas de asignación de recursos

Memoria

Autor

Sergio Alejandro Pastor Pastor

serpaspa@ei.upv.es

Tutor

Federico Barber

fbarber@dsic.upv.es

Introducción

La asignación de recursos ha sido y es un problema complejo, pues existen innumerables soluciones posibles y formas para generarlas. Pero ninguna destaca por encima de todas las demás como técnica universal a aplicar. En su lugar existen diferentes aproximaciones que abordan el problema y se comportan mejor o peor según la casuística específica de cada uno. En este documento abordaremos la asignación de jornadas laborales a diferentes proyectos de desarrollo software dentro del área de la consultoría, donde los periodos de tiempo son muy acusados y el recurso, tiene un coste elevado. Trataremos el problema desde los aspectos más generales de la distribución de recursos hasta las peculiaridades de nuestro caso, haremos análisis de algunas técnicas y herramientas presentes hoy en día y qué aporta nuestra herramienta que no podíamos cubrir con el estado del arte actual, hablaremos de los detalles de su diseño e implementación y finalmente descubriremos la potencia que aportan las técnicas metaheurísticas a la resolución de este tipo de problemas en comparación con otras más tradicionales.

Índice

1. Problemática	4
1.1 Descripción general	4
1.2 Motivación.....	4
1.3 Tipología	5
1.4 Estado del arte	5
1.4.1 Técnicas existentes.....	5
1.4.2 Aplicaciones existentes	8
2. Problemática específica	10
2.1 Tareas a asignar.....	12
2.1.1 Proyectos.....	13
2.1.2 SATs	13
2.2 Recursos disponibles	13
2.2.1 Módulos	14
2.3 Esquema	14
2.4 Restricciones de factibilidad y criterios de optimalidad	15
2.4.1 Factibilidad	15
2.4.2 Optimalidad.....	15
2.5 Granularidad de las asignaciones.....	17
3. Análisis y especificación del problema a resolver	18
3.1 Estructuración de la información	18
3.1.1 Gestión de tareas	18
3.1.2 Gestión de recursos.....	20
3.1.3 Asignación de recursos.....	21
4. Diseño de la solución.....	22
4.1 Elección de la estrategia de resolución	23
4.2 Aplicación del algoritmo GRASP al problema de la planificación	24
4.2.1 Inicialización	24
4.2.2 Bucle de control	25
4.2.3 Proceso de creación	25
4.2.4 Proceso de mejora	26
4.2.5 Evaluación de asignaciones.....	28
4.2.6 Proceso de reconstrucción del calendario	29
5. Implementación	31
5.1 Lenguaje y entorno de desarrollo.	31
5.2 Implementación del planificador en .NET utilizando Visual Studio	32

5.2.1	Capa de acceso a datos	33
5.2.2	Capa de lógica de control	35
5.2.2	Capa de presentación	38
6.	Seguimiento de la aplicación	39
6.1	Comunicación con la aplicación <i>Solution Manager</i> de SAP.	39
6.2	Multithreading	39
6.3	Suministrar el servicio vía web	40
6.4	Planificación multimódulo	40
6.5	Consultores multimódulo	40
6.6	Sección de configuración	41
6.7	Control de versiones	41
7.	Evaluación de laboratorio	42
7.1	Equipo de pruebas	42
7.2	Experimentos realizados	42
7.2.1	Evaluación de la relación con el cliente	42
7.2.2	Evaluación de las cualificaciones requeridas	46
7.3	Evaluación del rendimiento	49
7.3.1	Resultados base	49
7.3.2	Incremento del tiempo en función del incremento de calendarios creados	49
7.3.3	Incremento del tiempo en función del incremento de iteraciones de mejora	50
7.3.3	Incremento del tiempo en función del incremento combinado	50
7.4	Consumo de recursos	51
8.	Evaluación de campo	52
8.1	Evolución de la calidad del calendario en función del número de iteraciones	53
8.1.1	Interpretación de los resultados	53
9.	Conclusiones	55
10.	Bibliografía	56
10.1	Desarrollo software y gestión de proyectos	56
10.2	SAP	56
10.3	Distribución de recursos	56

1. Problemática

1.1 Descripción general

El planteamiento de la situación es relativamente sencillo: Disponemos de una o más tareas que deben realizarse y, a la vez, tenemos uno o más recursos que se pueden hacer cargo de que la acción se lleve a cabo.

Existen cuatro recursos principales que podemos asignar a las tareas:

- **Humano:** Cuánta gente se va a dedicar al cumplimiento de la tarea.
- **Económico:** Cantidad de dinero que se va a invertir en la tarea.
- **Tecnológico:** Tecnología implicada en el desarrollo de la tarea.
- **Temporal:** Periodo de tiempo necesario para resolver la tarea.

Así pues, trataremos de asignar una porción de estos recursos a cada una de las tareas que tenemos que llevar a cabo, teniendo en cuenta que los recursos son limitados y emplearlos en una tarea en concreto implica no disponer de ellos para otras, al menos, simultáneamente.

Lo que a simple vista parece un problema sencillo, comprobaremos a lo largo del documento por qué dista mucho de serlo y por qué no existe una solución óptima y genérica para el mismo.

1.2 Motivación

Tradicionalmente, la persona encargada de la gestión del proyecto o de elaborar los calendarios de turnos de sus empleados, precisa un alto conocimiento de qué es capaz cada persona, así como de sus restricciones particulares (de tiempo o de habilidad) para poder distribuir la carga de trabajo. Este problema es admisible cuándo la cantidad de recursos de la que dispones es reducida y por lo tanto el abanico de soluciones al problema es más pequeño. Pero cuándo la cantidad de recursos y tareas aumenta linealmente, la cantidad de soluciones lo hace exponencialmente y esa persona deja de ser capaz de manejar la solución o, al menos, de dar una solución óptima.

Estas situaciones no eran abordables antes de que aparecieran los computadores por la gran casuística que los caracteriza. En cuanto aparecieron y se pudieron formalizar las restricciones y evaluar la calidad de las soluciones de forma programática, volvieron a estar bajo el foco de atención de la comunidad académica. Como veremos más adelante, existen herramientas que ayudan a la persona encargada de la planificación dándoles más visión sobre el problema y facilitándole la toma de decisiones, aunque el proceso de planificación sigue completamente en manos de esa persona. Otras, en las que nosotros estamos interesados, evalúan la situación del problema en cualquier fase en la que se encuentre y te proponen una solución al mismo.

Nuestro proyecto, el trabajo final de master, queda enmarcado dentro de las competencias de la inteligencia artificial. Nos valdremos de las técnicas aprendidas en el desarrollo del master para dar solución a un problema real de distribución de recursos en el ámbito del desarrollo de producto software.

La razón principal por la que este problema está siendo tan investigado es por el crecimiento derivado de la empresa o sociedad que hace una buena gestión de sus recursos. Existe una gran competitividad dentro de un mismo sector por acaparar una mayor porción del mercado. Una óptima gestión de los recursos permite incrementar la carga de trabajo sin incrementar el número de recursos, o dicho de otra forma, incrementar los beneficios sin incrementar el coste.

1.3 Tipología

Encontramos dos posibles soluciones al problema de la distribución de recursos:

- **Factibles:** Este tipo abarca el conjunto de soluciones que cumplen las restricciones definidas del problema. Es decir, todas las válidas y por lo tanto elegibles.
- **Optimizables:** Este tipo, es un subconjunto de las factibles, pues además de ser soluciones correctas al problema, responden a unos criterios de calidad definidos por el usuario (criterios de optimización)

La diferencia remite en que mientras el primero no intenta escatimar en recursos para poder llegar a una solución correcta porque no estamos seguros de que vayamos a encontrar al menos una, el segundo reorganiza los recursos para intentar repartirlos de una manera más eficiente.

En la planificación tradicional, una persona era capaz de encontrar un número reducido de soluciones factibles y de estas se quedaba con la que desde un punto de vista marcado por la subjetividad y la experiencia profesional opinaba que era la mejor.

Con los recursos disponibles en la actualidad, la bolsa de soluciones factibles que podemos crear es mucho mayor, además podemos evaluarlas y mejorarlas simultáneamente en busca no sólo de las soluciones óptimas, si no de la mejor que podemos encontrar en el tiempo disponible.

1.4 Estado del arte

En este punto procederemos a enumerar una relación de técnicas y aplicaciones que las implementan en la actualidad para dar solución al problema de la planificación.

1.4.1 Técnicas existentes

1.4.1.1 Heurísticas

La heurística es una de las técnicas usadas para la asignación de recursos usando el método más óptimo y con el que se consigue el mejor resultado posible. Puede que la solución obtenida no sea el caso más óptimo para nuestro problema por diferentes variables, pero si el mejor en un problema dado debido a ciertas limitaciones.

Ventajas

Los métodos heurísticos tienen como principal ventaja que siempre elegirán la solución más óptima posible. Insertándole unas variables concretas, por ejemplo en el caso de nuestra aplicación si los recursos son los consultores, deberíamos insertar fechas clave, relación con el cliente, áreas donde cada consultor destaca más...

A partir de ese punto con las técnicas heurísticas se debería buscar qué consultor es el que mejor se adapta a cada tarea.

Otra de las ventajas, es que se pueden saber qué dependencias tienen unos recursos y tareas con otros.

Inconvenientes

Como podemos deducir, aunque los métodos heurísticos elijan la mejor solución, puede darse el caso de que la solución obtenida no sea la solución que mejor se adapte a nuestro problema.

Algunos motivos pueden ser:

- Si tratamos de buscar todas las soluciones posibles, nuestro problema tiene una métrica de tipo np-completo y el tiempo de evaluación de todas las soluciones para determinados problemas, no es abordable ni para un computador.

- Si apodamos técnicas que nos permitan limitar el abanico de soluciones, como técnicas de poda o búsqueda local, podemos estar descartando soluciones válidas o ser incapaces de mejorar la que tenemos actualmente porque hemos encontrado un máximo local.

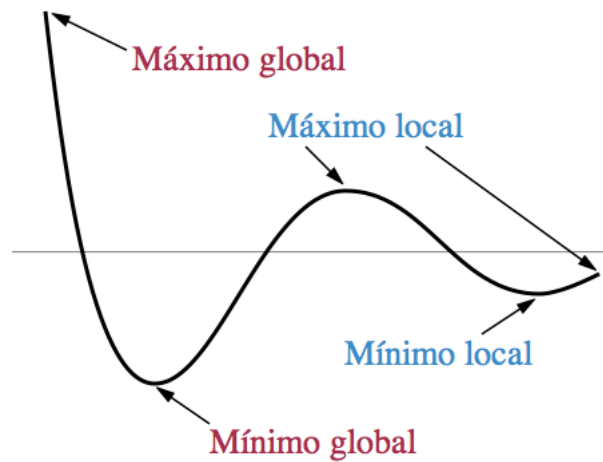
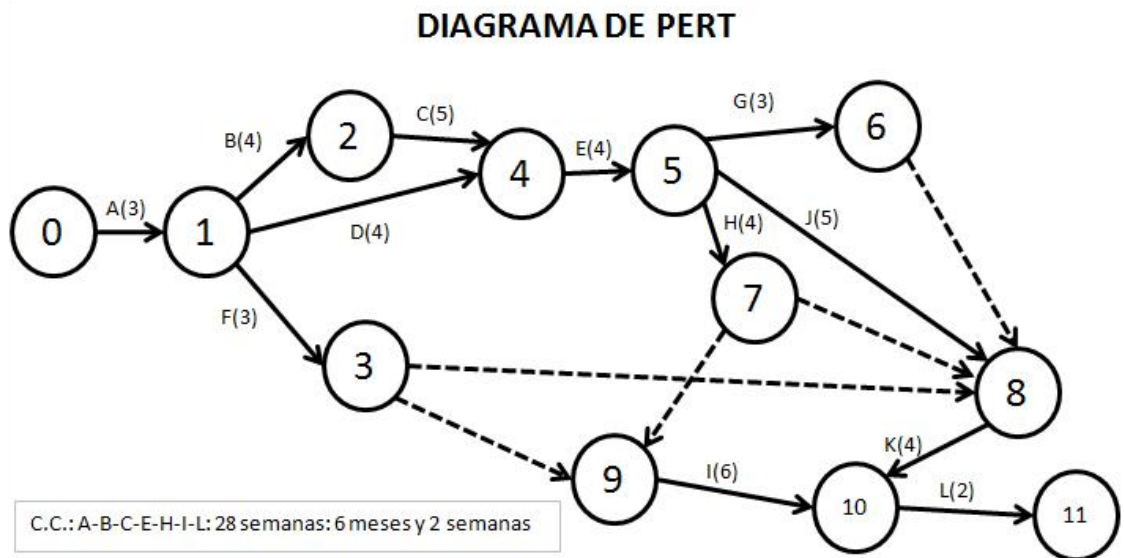


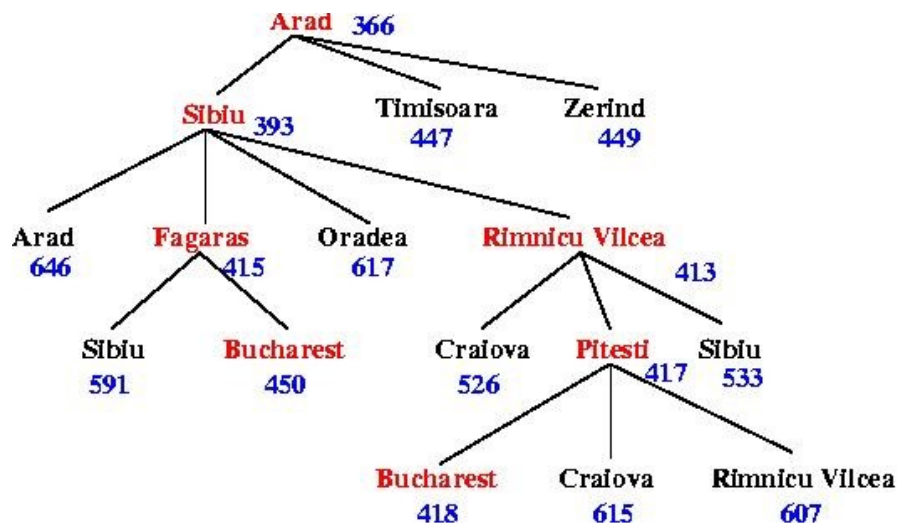
Figura 1. Evaluación de soluciones

Ejemplos

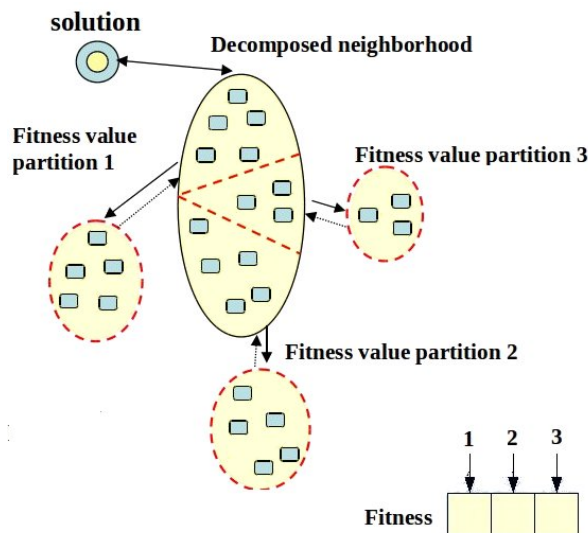
- **Diagramas Pert (CPM):** los diagramas de PERT o diagrama de estados, son una técnica utilizada en la gestión de proyectos para la correcta distribución de los recursos y ordenación en la que se han de realizar las tareas en un proyecto. Para la optimización de esta técnica se usa el método de la ruta crítica (Critical Path Method) que consiste en elegir el camino del diagrama cuya distribución requiera menos tiempo para el proyecto.



- **Diagrama A*:** consiste en la búsqueda del camino más corto desde un punto a otro, donde se examinan todos los caminos posibles, y aunque se elige el más óptimo su punto débil es el tener que hacer una búsqueda de todas las posibilidades.



- **Metaheurística:** es un método heurístico que sirve para solucionar problemas en los que no existe un algoritmo o un método ya definido para solventarlos. Por ejemplo, porque existe una restricción fuerte de tiempo y precisas de la mejor solución posible dentro de la ventana de tiempo definida. Al igual que los heurísticos, se utilizará siempre la mejor solución posible. Un ejemplo de metaheurística es GRASP, utilizado en nuestra aplicación, aunque hay una gran cantidad de metaheurísticas en la actualidad, algunos son algoritmos genéticos, colonia de hormigas o listas TABÚ.



1.4.1.2 Sistemas basados en reglas

Esta metodología se basa en el seguimiento de un conjunto de premisas o reglas para llevar a cabo la solución que se más óptima y sencilla. Cada una de las reglas se basa en la relación existente entre dos objetos donde se tiene en cuenta el problema que se quiere abordar y la solución a la que se quiere llegar. Estas reglas están definidas por un humano, quien toma el rol de experto.

Ventajas

Los sistemas expertos tienen como ventaja que trabajan rápidamente llegando a una buena solución con un buen nivel de fiabilidad, y son capaces de manejar grandes cantidades de datos. Además estas técnicas pueden ser usadas en diferentes áreas (no solo en el área de la computación) como economía, medicina, biología... Otra de las ventajas es su modularidad, es decir que podemos es añadiendo más reglas sin necesidad de realizar un cambio en el resto de las reglas existentes.

Inconvenientes

Los sistemas expertos necesitan que una persona experta en una determinada área “les enseñe”, puesto que previamente necesitan almacenar información sobre un tema para poder llegar a solucionarlo mediante reglas, por lo que tienen que ser modificados cada vez según la aplicación que se le vaya a dar, lo que supone un gran coste.

Ejemplos

- **CLIPS:** es un lenguaje y aplicación que se basa en la selección de las reglas por orden de prioridad realizando comparaciones con las demás reglas definidas.
- **RBS:** se basa en una representación del conocimiento mediante un conjunto de reglas. Consta de dos partes: la base del conocimiento y el motor de inferencia.

1.4.2 Aplicaciones existentes

- **Planificación de negocio:** actualmente existen un gran conjunto de aplicaciones de planificación de negocio y recursos. Estas son por lo tanto las aplicaciones con las que tenemos que compararnos para evaluar la calidad de nuestra solución.
 - o *Microsoft Project:* La aplicación crea calendarización de rutas críticas, además de cadenas críticas y metodología de eventos en cadena disponibles como add-ons de terceros. Los calendarios pueden ser secuenciados para una disponibilidad limitada de recursos, y las gráficas visualizadas en una Gráfica de Gantt. Adicionalmente, Project puede reconocer diferentes clases de usuarios, los cuales pueden contar con distintos niveles de acceso a proyectos, vistas y otros datos. Los objetos personalizables como calendarios, vistas, tablas, filtros y campos, son almacenados en un servidor que comparte la información con todos los usuarios.
 - o *IBM C-PLEX:* Se trata de un paquete software implementado originalmente en lenguaje C que soporta diversos tipos de optimización matemática. Resuelve problemas de programación numérica, problemas lineales o cuadráticos. Proporciona interfaces en C++, C#, Java o Python con las que interactuar. También se puede comunicar con Excel y Matlab.

- *Retain International*: es un software utilizado en grandes empresas internacionales que sirve para la organización de la empresa de forma global aunque esta tenga sedes en diferentes países. Utiliza métodos heurísticos para la planificación de las tareas eligiendo la mejor planificación posible. Está disponible en varias plataformas.
 - *HMS Software*: se encarga de la planificación de recursos a nivel de empresas con recursos humanos (*TimeControl*) y a nivel industrial con recursos materiales como equipamiento, pedidos o pagos (*TimeControl Industrial*).
 - *OnePMO*: es otro software de organización de tareas y recursos similar al Microsoft Project con planificación automática.
 - *OptaPlanner*: se encarga de la planificación de recursos en determinados escenarios (tanto humanos como materiales).
 - *Pyhop*: es un organizador de tareas y estados que elabora un proyecto de negocio basándose en tareas y reglas que se le pasan inicialmente como variables, a partir de donde decidirá el plan de trabajo.
- **Antivirus**: existen una gran cantidad de antivirus en la actualidad que se basan en análisis heurístico para la búsqueda de virus basándose en su firma en caso de no ser conocidos. Podemos destacar varios ejemplos: *Panda Security, Kaspersky, CCM, Norton Antivirus...*
 - **Análisis modal de fallos y efectos (AMFE)**: se encarga de la detección de fallos en un sistema para su prevención y la mejora de la calidad, como puede ser *PTC Windchill FMEA*, que se encarga de la identificación y clasificación de los fallos encontrados y planificación de una estrategia para solucionarlos. Otro ejemplo de aplicación software es *Visual Factory FMEA 7*, que se encarga de la administración de este tipo de documentos mediante gráficos con forma de árbol.

2. Problemática específica

La gestión de la planificación es especialmente importante en el área del desarrollo software, esto es debido a que pese al avance de la tecnología, esta es incapaz de formalizar un problema y describirlo en forma de código fuente. Por lo que la creación de productos software es una tarea principalmente artesanal, que lleva tiempo de analizar el problema, de escribir el código que solucione el problema y personas altamente cualificadas que lo hagan.

Un error de análisis puede suponer haber malgastado muchos recursos y volver a una etapa de desarrollo anterior. Incluso con el producto terminado y en uso, un fallo puede derivar en detener la producción completa de una fábrica. Las fechas de entrega tienen la misma importancia, porque en caso de no llegar a tiempo a un arranque es tiempo que el cliente no está generando beneficios.

Estamos por lo tanto ante un problema con altas exigencias de los cuatro tipos de recursos que mencionábamos anteriormente.

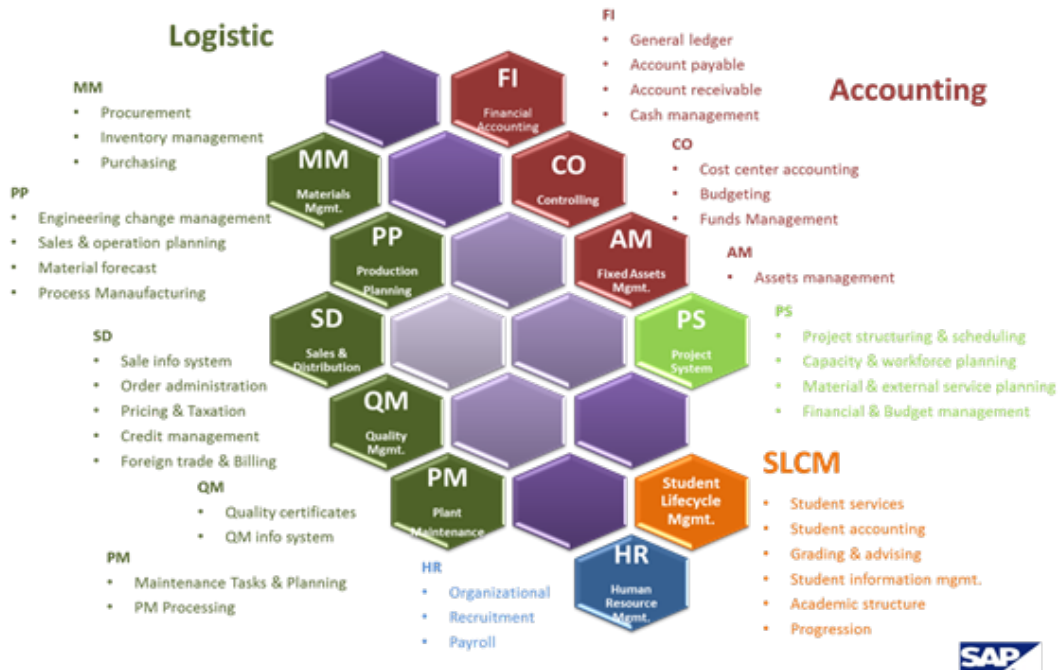
El ámbito de uso de la aplicación está dentro de los márgenes de la consultoría de desarrollo software, concretamente el desarrollo e implantación de sistemas SAP.



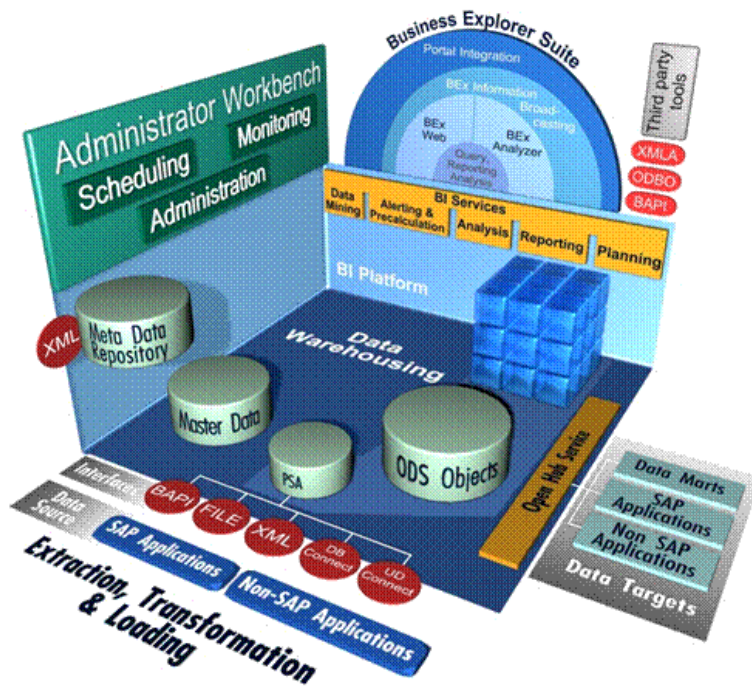
SAP AG es una empresa multinacional alemana dedicada al diseño de productos informáticos de gestión empresarial, tanto para empresas como para organizaciones y organismos públicos. La principal característica que la define es que el producto es genérico y se adapta en cada implantación a las necesidades específicas del cliente. Gracias a esto reutilizamos la experiencia obtenida de otras implantaciones para mejorar el producto estándar y facilitar próximos desarrollos.

Por este motivo la solución de SAP está separada en secciones con funcionalidad propia que llamamos módulos y que se implantan individualmente en el cliente. Algunos de estos módulos son: Sistemas (AS), Ventas (SD), Logística (LO), Costes (CO), Finanzas (FI), Control de la producción (PP), Gestión de stocks (IC),...

SAP ERP & SLCM (ECC 6.0) Implementation Service



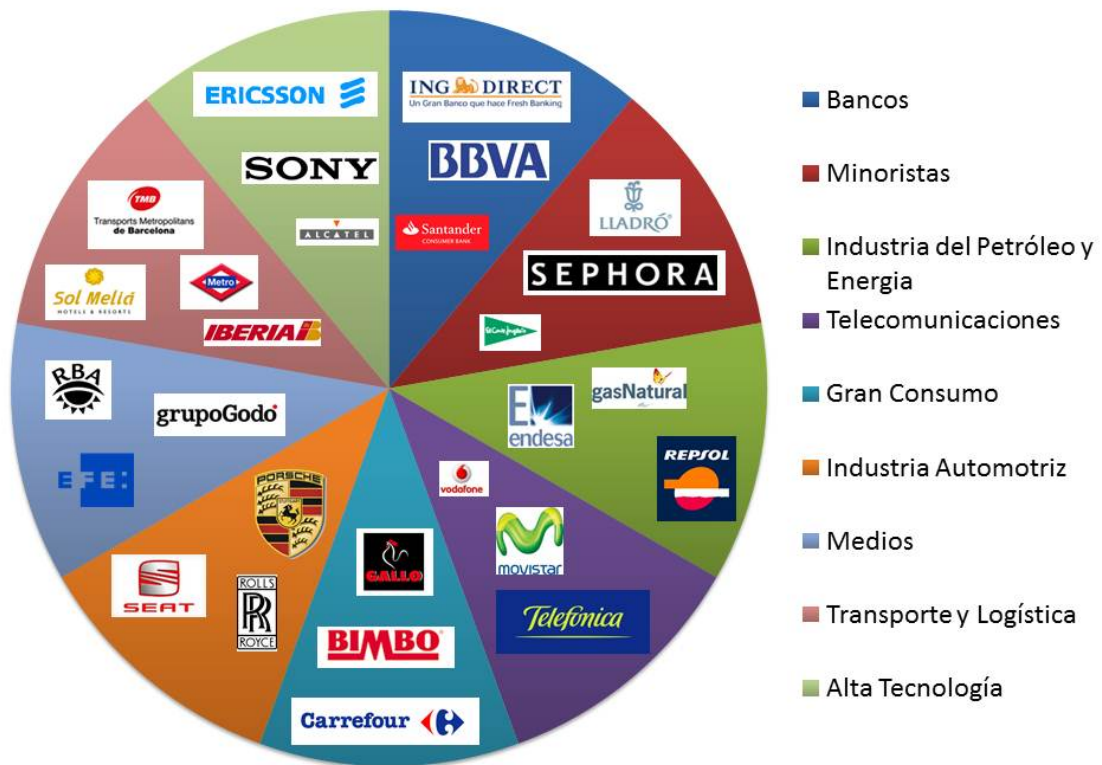
La aplicación se ha diseñado tomando como modelo el módulo BW o *Business Warehouse*, este módulo se encarga de almacenar, tratar y dar sentido a todos los datos de los que dispone la empresa (vengan de un sistema SAP o no) para facilitar la toma de decisiones de negocio. La potencia reside en la combinación de datos que eran independientes pero que al verlos juntos toman un nuevo significado, para ello, algunas técnicas utilizadas son los cuadros de mando o la minería de datos.



Gran parte de las empresas a nivel mundial confía a SAP la gestión de su empresa y por lo tanto existe un gran número de profesionales que se dedican a su gestión a cualquier nivel. Es por esto que nuestra aplicación a medida cobra relevancia, pues está ajustada a la forma de trabajar que emplea SAP y de este modo es utilizable no sólo por nuestra empresa, sino por cualquier consultora que se dedique a lo mismo que nosotros.

Algunos de los clientes que tienen SAP son:

Cientes de SAP en Diversas Áreas



2.1 Tareas a asignar

El trabajo a asignar, por lo tanto, son desarrollos software completos o parciales en cualquiera de sus etapas: preventa, análisis, desarrollo, implementación, pruebas técnicas y de validación o soporte.

Las tareas han sido modeladas de la siguiente forma:

2.1.1 Proyectos

Si estamos ante un desarrollo nuevo, ya sea una implantación desde cero o la creación de una funcionalidad que no existía previamente, entonces se trata de un proyecto, a su vez, este se desglosa en agrupaciones de tareas con un sentido lógico, que representan una funcionalidad completa y con fecha de entrega propia a los que llamamos hitos. Finalmente, los hitos están compuestos por tareas, cada tarea es una unidad de trabajo indivisible y tiene una o varias cualificaciones necesarias para poder completarla. Las tareas están asignadas a una persona durante uno o varios días hasta que se completan.

2.1.2 SATs

Un SAT es una incidencia, un error inesperado que ha aparecido en el sistema del cliente en un desarrollo que ya se había completado y validado, por lo tanto escapa de los términos del contrato acordado y es facturable. Estos errores tienen una escala de gravedad en función de la limitación que le supongan al cliente para funcionar correctamente. SAP se compromete a, que si la gravedad es lo suficientemente alta, (Por ejemplo, has tenido que parar la cadena de producción) a darte una solución en una ventana de tiempo de cuatro horas, independientemente de la parte del mundo en la que esté la empresa. Por la naturaleza del SAT, este engloba directamente a las tareas que deben realizarse para solventarlo. Tareas con la misma estructura que las de los proyectos.

2.2 Recursos disponibles

Analicemos en detalle de los tipos de recursos existentes, que peso tienen dentro de nuestro problema específico:

Humano: El principal recurso de una consultora (sea software o no) son las personas que trabajan en ella, el producto que venden son las horas de trabajo de sus consultores. Por lo tanto este es el más valioso de todos.

Económico: La cantidad de dinero invertida en el proyecto juega varios papeles porque: permite aumentar el número de consultores que puedes utilizar, mejorar el equipo con el que trabajas, tanto de prestaciones como de versión de SAP y finalmente esto implica una menor cantidad de tiempo invertido en el desarrollo.

Tecnológico: Estamos delante de un modelo de negocio que vende soluciones tecnológicas, una mayor calidad de la tecnología empleada implica unos tiempos de ejecución más rápidos, una interfaz más amigable y menos errores de la herramienta.

Tiempo: Recordemos que el producto que ofrecemos es el tiempo de trabajo de nuestros consultores, personas más capacitadas con mejores herramientas podrán hacer el mismo trabajo en menos tiempo. Abaratando en el proceso el coste final del desarrollo.

Resumiendo, tenemos cuatro parámetros para jugar con ellos, pero premiar o castigar uno tiene efectos sobre los demás. Por ejemplo, contratar consultores expertos en sus áreas y mejor equipo supone que el desarrollo se hará en menos tiempo y con menor probabilidad de error, pero tiene un fuerte impacto económico. Por otro lado puedes dedicar menos personal a la tarea, pero implica dilatar el desarrollo en el tiempo y tener bloqueado ese recurso durante más tiempo.

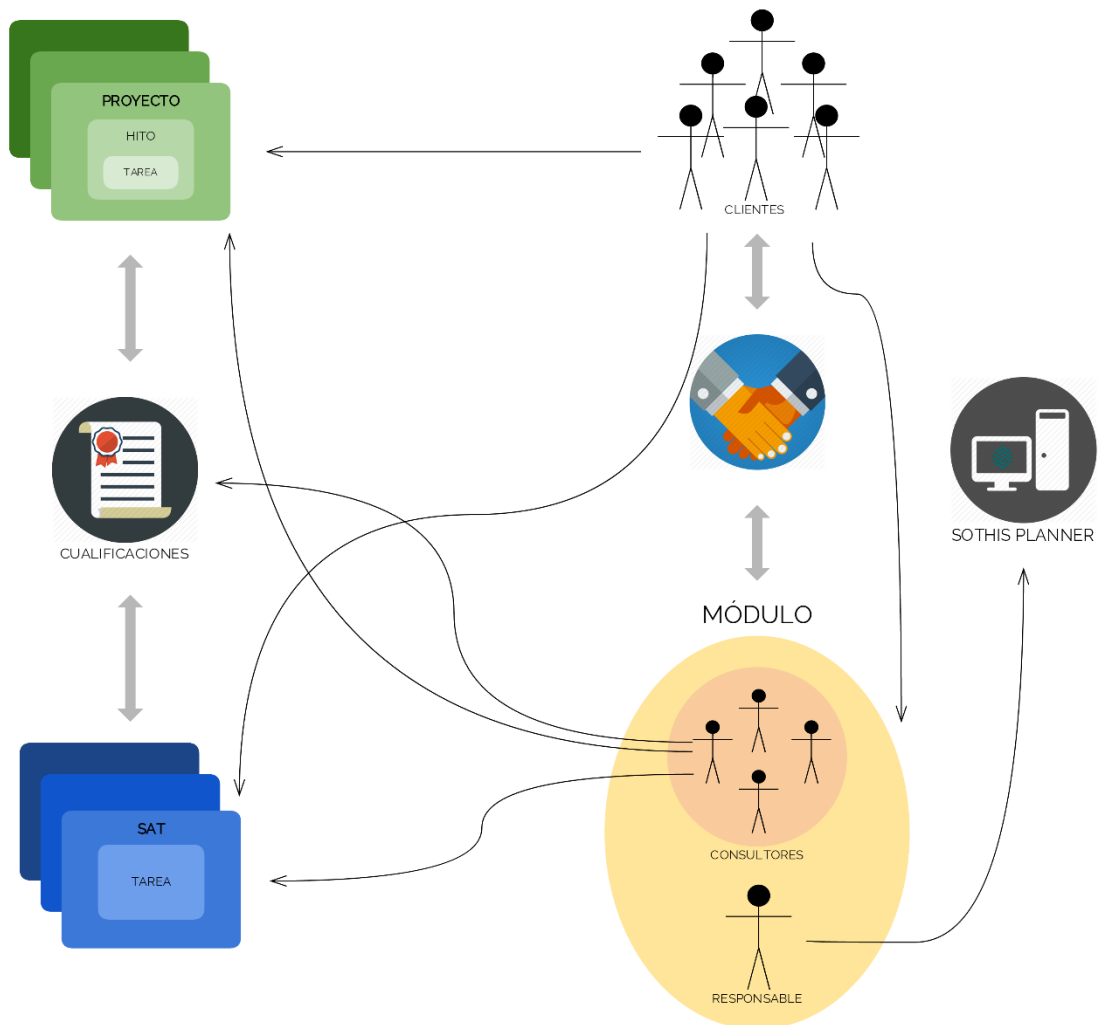
Los recursos han sido modelados de la siguiente forma:

2.2.1 Módulos

Los módulos son equipos de trabajo con organización propia, tienen el mismo nombre que los módulos de SAP porque los miembros que lo componen tienen conocimientos en ese área y pueden hacer implantaciones de ese módulo SAP. Cada miembro del equipo de trabajo es un consultor y representa nuestra unidad mínima de asignación a tareas. Los consultores pueden pertenecer a varios módulos según la variedad de conocimientos que posean.

Los módulos tienen un conjunto de cualificaciones del mismo tipo que las tareas, es decir, para que un consultor pueda pertenecer a un módulo de SAP tiene que poseer una serie de conocimientos sobre el funcionamiento del mismo. Todas las cualificaciones están asociadas al módulo y los consultores tienen un subconjunto de esas cualificaciones. Ese subconjunto puede ser la totalidad de cualificaciones del módulo si el consultor es un experto en el mismo.

2.3 Esquema



2.4 Restricciones de factibilidad y criterios de optimalidad

En los diversos escenarios de aplicación de este tipo de sistemas, existen restricciones muy particulares que tienen en cuenta la tipología concreta de los trabajadores, de las tareas o servicios a realizar, etc. En esta aplicación, se reflejan restricciones suficientemente generalistas y correspondientes a un escenario de tipo desarrollo software. Sin pérdida de generalidad, estas restricciones permiten el desarrollo de un sistema altamente complejo y realista. Ídem con los criterios de optimalidad.

La factibilidad se describe mediante restricciones, que son las condiciones que una solución factible debe cumplir necesariamente.

Los criterios de optimización son funciones que evalúan una solución factible y determinan su calidad (bondad, coste, etc.).

En los siguientes dos puntos definiremos las restricciones y los criterios.

2.4.1 Factibilidad

1. **Cualificaciones:** Para poder resolver una tarea, un consultor debe estar capacitado para ella, al menos parcialmente.
2. **Disponibilidad:** Las asignaciones de tareas solo se pueden hacer dentro de la jornada laboral del consultor y cada consultor tiene su propia jornada laboral.
3. **Repartición equitativa:** El calendario debe de estar balanceado, todos los consultores deben de tener tareas asignadas todos los días siempre que las haya, aunque no sean los más indicados.

2.4.2 Optimalidad

1. **Cualificaciones:** Esta restricción es también un criterio de optimalidad. La persona asignada DEBE saber hacer la tarea al menos parcialmente, pero la calidad de la asignación MEJORA cuantos más requerimientos de la tarea sepa hacer.

Este criterio intenta exprimir al máximo el potencial de un recurso, la mejor elección para una tarea es la persona cuyas cualificaciones son similares a los requisitos de la misma. Dejando a las personas más cualificadas para las tareas más exigentes y a las menos cualificadas las tareas más rutinarias.

- 2. Fechas de entrega:** Siempre que sea posible se debe de intentar asignar las tareas antes de la entrega. El motivo por el que este punto es un criterio de optimización y no de factibilidad es que la aplicación se va a utilizar como entorno de simulación, para ver si es posible aumentar la carga de trabajo del equipo o proporcionar fechas estimadas de entrega al cliente. También debemos de tener en cuenta los cambios de requisitos dentro del desarrollo software, un cambio de prioridad en la implementación puede suponer un cambio de fechas de entrega, por esto se involucra íntimamente al cliente en el desarrollo, para que sea conocedor de los avances de su proyecto y consciente de por qué las fechas de entrega son las que son.
- 3. Relación profesional:** Siempre que sea posible, debe intentar asignarse a la persona que tenga una mejor relación profesional con el solicitante. Un mismo consultor puede no estar disponible para dos clientes simultáneamente, pero de la colección de recursos elegibles, debemos tratar de maximizar los parámetros: conocimiento que el consultor tiene sobre el sistema del cliente y su forma de gestión, relación extraprofesional. Por razones personales puede que un consultor no pueda trabajar con un cliente o al revés, que dada la buena relación sea una buena opción repetir la asignación.

2.5 Granularidad de las asignaciones

Se hace necesario acordar de antemano con qué unidad de tiempo mediremos la duración de las tareas y el tiempo empleado por los consultores en su resolución. Así como a qué nivel de detalle queremos hacer la estimación de esta duración.

Comencemos por la duración de las tareas, en una primera aproximación parece lógico intentar medir la duración de cada tarea individualmente y cuando se la asignamos a un consultor dispone de una ventana de tiempo reservada igual a esa estimación para completarla. Sin embargo, cuando tratamos de hacer una planificación real, sólo valoramos las tareas más importantes y añadimos una cantidad extra de tiempo con la que creemos podemos terminar las menos relevantes. Por ejemplo, en un hito de instalación del sistema, sabes que es muy importante el tiempo consumido por el instalador, y que vas a tardar un poco más porque tienes que crear las conexiones con la base de datos y la configuración inicial, pero realmente lo que más has valorado es el tiempo de instalación.

Por este motivo y porque resulta muy pesado para el usuario de la aplicación bajar al nivel de detalle de la tarea, hemos acordado que es mejor estimar la duración del hito o del SAT (pues sabes aproximadamente cuánto tiempo va a llevar el proceso global) y hacer una distribución normal del tiempo en las tareas que dependen del hito o del SAT. Es importante tener en cuenta que habrá tareas con un tiempo estimado excesivo y otras insuficientes, pero con consultar el hito/SAT al que pertenecen sabes por donde continuar el desarrollo.

Hasta este punto sabemos qué queremos medir, pero ¿Cuál es la unidad de medida apropiada? De nuevo, una primera aproximación parece lógico utilizar las horas, al fin y al cabo un empleado dedica ocho horas diarias a su trabajo y podemos medir los hitos y SATs con la misma unidad. Sin embargo, cuando una tarea no encaja en las horas que tiene disponibles un consultor para ese día, entramos en un problema de fragmentación de tareas y es preciso una descripción precisa de cómo debe actual el sistema cuándo esto sucede. ¿Debe de continuar con la tarea que tenía empezada el próximo día?, ¿Debo de buscar una tarea que quepa en la ventana de tiempo disponible?, ¿Debo de asignar la misma tarea a dos consultores para que quepa el desarrollo sin necesidad de ser fragmentada?

Una segunda aproximación nos lleva al siguiente punto, el producto que vende la empresa son bolsas de jornadas de consultores, es decir, si contratas una bolsa de 10 jornadas quiere decir que 10 días laborables completos un consultor va a estar dedicado a ti en un régimen de tiempo acordado. Por lo tanto resulta un poco más lógico que la unidad de medida que utilicemos sea la jornada laboral y que en caso de terminar la tarea marcada para ese día, se continúe con la siguiente dentro del hito/SAT en el que se encontraba la anterior.

Por último remarcar que la asignación realizada sólo marca a la persona como responsable de la realización de la misma, al fin y al cabo el consultor trabaja dentro de un equipo y puede haber varios miembros de un equipo trabajando en la misma tarea. El calendario creado es solamente una guía recomendada.

3. Análisis y especificación del problema a resolver

Comprendida la problemática específica, pasamos a formalizar el análisis realizado.

3.1 Estructuración de la información

Dada la naturaleza del sistema, es preciso jerarquizar la información y almacenarla de forma organizada.

3.1.1 Gestión de tareas

3.1.1.1 Proyectos

Los proyectos están asignados a un cliente y a uno o varios módulos según el tipo de desarrollo. También tienen una fecha de entrega y dos ratios: el número de jornadas en las que se estima a priori que va a suponer el desarrollo y el número de jornadas que llevo consumidas hasta el momento, el cual es la suma de las jornadas consumidas en los hitos que posee.

Los hitos pertenecen tanto al proyecto que se está desarrollando cómo al módulo que se va a encargar de hacer el desarrollo. De esta forma el modelado queda abierto a cómo se quiera gestionar in situ. Puede haber varios módulos que se auto gestionen y dividan sus recursos sobre el proyecto, o bien, se puede hacer una gestión global del proyecto dando de alta los diferentes hitos de todos los módulos que participen. De nuevo, al igual que en el proyecto, encontramos una fecha fin del desarrollo y dos contadores, uno para las jornadas estimadas y otro para las jornadas empleadas.

Las tareas son las funcionalidades individuales que en conjunto suman un requerimiento del desarrollo. Por ejemplo, si un hito de mi proyecto es gestionar mis facturas, necesito: una interfaz gráfica para interactuar con ellas, una lógica de control de las mismas en función de mi modelo de negocio, una capa de permanencia para almacenarlas, comunicación con dispositivos de salida como impresoras, etcétera.

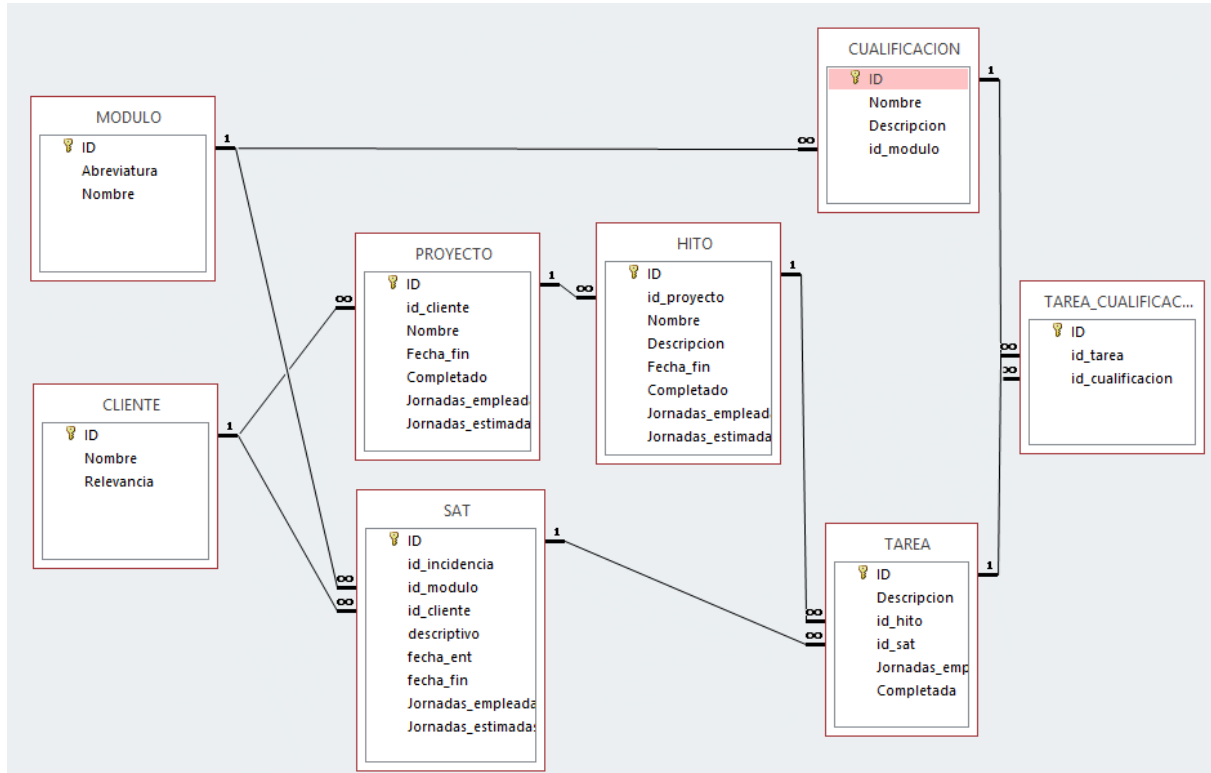
Finalmente cada tarea tiene una o más cualificaciones para poder llevarla a cabo. Por ejemplo, para poder realizar la tarea *Imprimir facturas*, debemos de saber cómo se comunica el sistema SAP con los dispositivos de salida, cómo enviarle la información en el formato que espera el dispositivo, cómo maquetar la factura o de donde obtener la información de la base de datos que necesito para componerla.

3.1.1.2 SATs

Los SATs guardan similitud con los proyectos, pero poseen un nivel jerárquico menos pues están compuestos directamente por las mismas tareas que los hitos.

Un SAT tiene un identificador propio de SAP para que el consultor pueda buscar información detallada acerca de él en el sistema, el módulo y cliente al que pertenece, una breve descripción de problema, dos fechas, una en la que entró el problema y otra en la que debe estar resuelto y los dos contadores de jornadas: las estimadas y las empleadas.

3.1.1.3 Diagrama



3.1.2 Gestión de recursos

3.1.2.1 Módulos

Los módulos son los equipos de desarrollo, estos se distribuyen la carga de trabajo de forma autónoma aunque se coordinan con el resto de módulos que participan en los proyectos comunes.

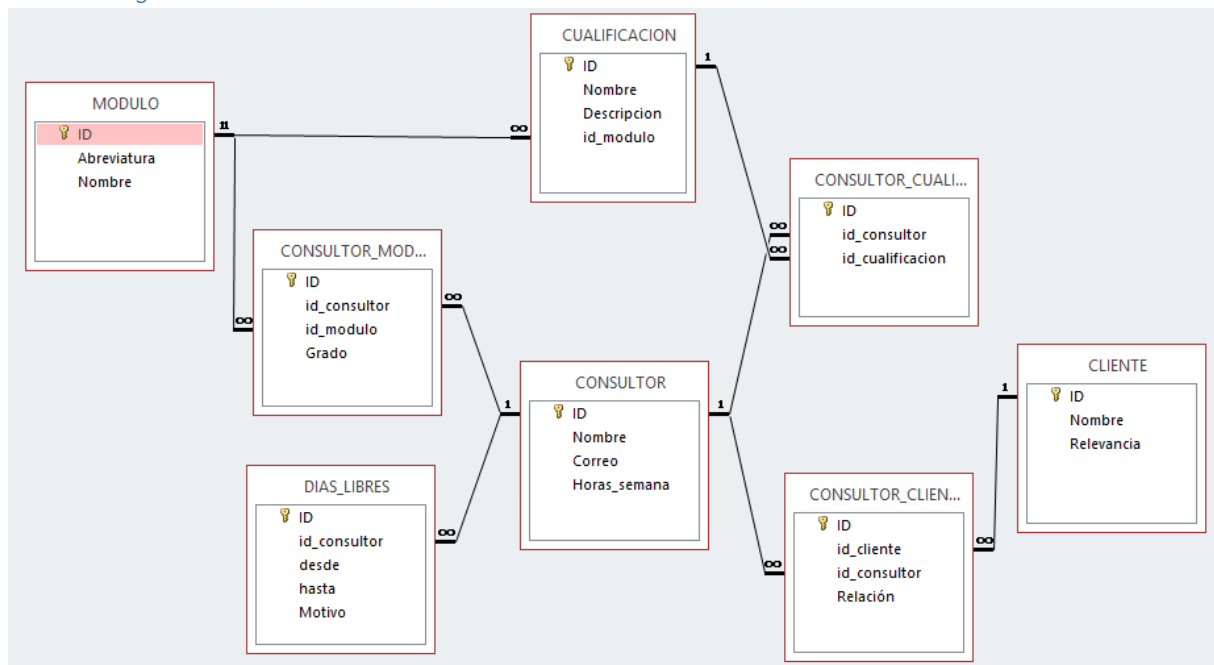
Un módulo tiene una relación de consultores que pertenecen a él y un conjunto de aptitudes requeridas para la implantación y gestión del mismo. Uno de los consultores es el coordinador del equipo y es nuestro usuario clave, quien va a utilizar la aplicación y distribuir la carga de trabajo en última estancia entre el resto de miembros.

3.1.2.2 Consultores

Los consultores son el recurso del que disponemos para asignar las tareas en última instancia. Estos pertenecen a uno o más módulos según la variedad de cualificaciones que posean y las cualificaciones siempre estarán relacionadas con el módulo. Típicamente, cuando un consultor es asignado a un proyecto, crea una relación estrecha con el cliente y adquiere conocimientos intrínsecos de su modelo de gestión, es necesario evaluar esta relación porque cuando aparece un SAT o un nuevo desarrollo de un cliente, este solicita que se encargue la misma persona que lo hizo previamente.

Los consultores como trabajadores que son, tienen una jornada laboral que cumplir y el tiempo que dedican diariamente al avance del proyecto o SAT es limitado, además es necesario saber la disponibilidad de la persona, pues puede estar de vacaciones, de baja médica o simplemente un día en concreto es festivo.

3.1.2.3 Diagrama

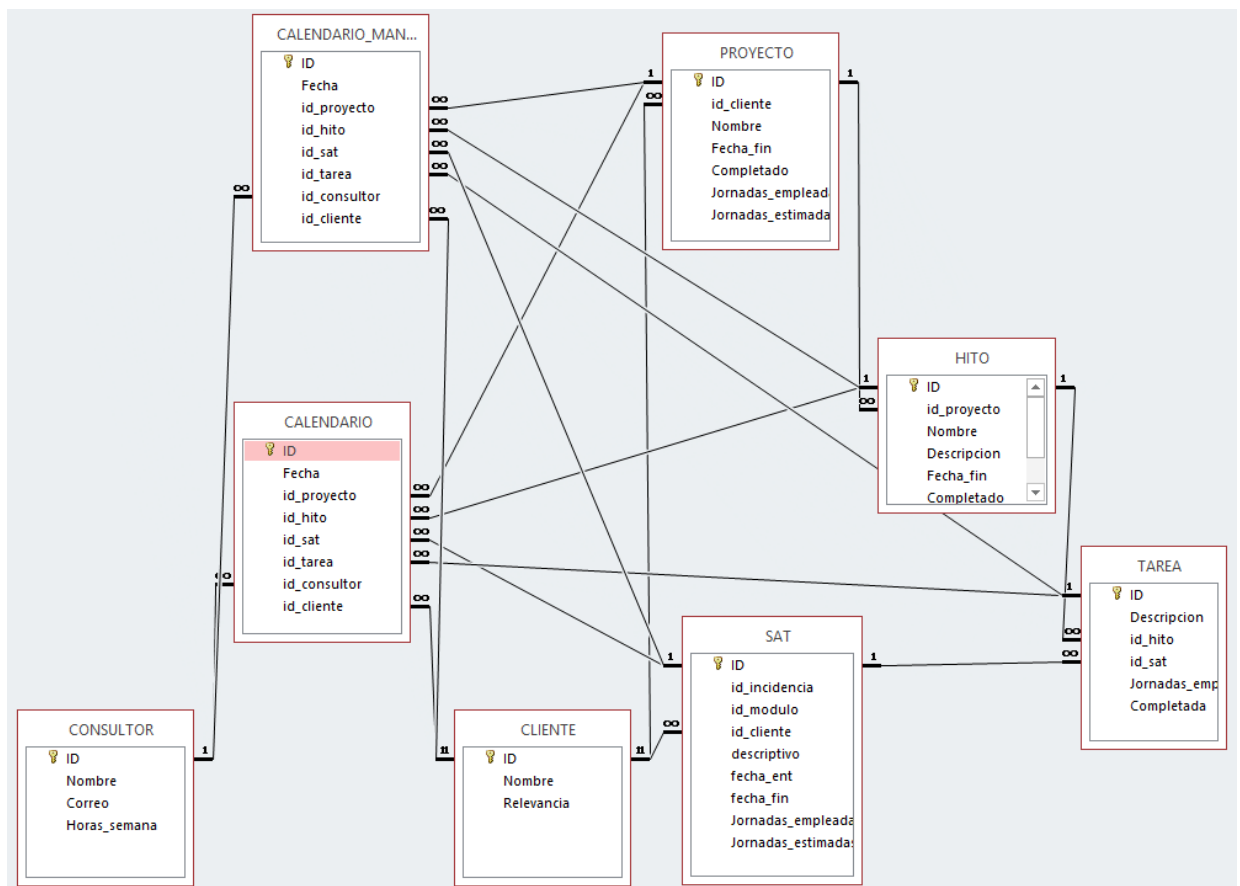


3.1.3 Asignación de recursos

El último paso es hacer una gestión a lo largo del tiempo del avance de los proyectos y los SATs.

Almacenaremos el calendario propuesto por el planificador en una estructura que nos relacione la tarea a realizar con la persona que lo va a hacer y cuándo lo va a hacer.

Es común dentro del mundo de la planificación que un experto tenga claras ciertas asignaciones que deben cumplirse, por ejemplo una preventa que sólo puede hacer el líder equipo o una exigencia concreta del cliente tanto de fecha cómo de consultor, pero que no tenga claras el resto de asignaciones de tareas porque simplemente no son tan relevantes. Por ello ponemos al alcance del usuario la opción de la planificación manual, es decir, podemos hacer entradas en el calendario que el planificador respetará a la hora de crear la planificación.



4.1 Elección de la estrategia de resolución

En la sección [1.4 Estado del arte](#) hemos visto una selección de técnicas y herramientas existentes que resuelven problemas de planificación. Tras estudiar los beneficios e inconvenientes de cada una, hemos llegado a las siguientes conclusiones:

Precisamos un desarrollo a medida de la aplicación:

- Muchas de las herramientas del mercado, aunque ayudan al usuario a definir la planificación, poco o nada hacen por automatizar la distribución de la carga de trabajo y la herramienta queda limitada a una interfaz de gestión manual.
- Del conjunto de herramientas que realmente aplican técnicas heurísticas, necesitan en cualquier caso una adaptación a nuestro problema en particular y esa genericidad creemos que puede llevar un lastre de rendimiento.
- Son herramientas caras de licenciar y de mantener, además que generan una dependencia con el fabricante a la hora de realizar modificaciones.
- Una herramienta a medida aplicada a este escenario en concreto puede ser reutilizada por otras consultoras, abriendo una posible vía de negocio en caso de comercializarse.

Descartadas las herramientas existentes, de las posibles técnicas, las metaheurísticas son las que mejor se aproximan a nuestra visión por los siguientes motivos:

- Es un problema de orden cúbico (todas las combinaciones de días a generar del calendario * todas las tareas que planificar * todos los consultores disponibles), por lo tanto una búsqueda en grafo tipo A* no sería factible por tiempo de ejecución y recursos del sistema.
- Se generan muchas soluciones válidas al problema, la cuestión es cuál de ellas es la más apropiada. Como no sabemos si la vamos a encontrar, debemos establecer un procedimiento que en un periodo de tiempo acotado nos devuelva la mejor opción válida.
- Los planificadores basados en reglas funcionan correctamente para crear soluciones válidas (definidas por las propias reglas), pero presentan problemas para evaluarlas y mejorarlas de forma incremental.

Una vez elegidas las metaheurísticas como técnica a seguir, falta definir cual se amoldaría mejor a la naturaleza del problema. De todas las posibles, dos destacan por encima de las demás:

- **Genéticos:** De usar esta técnica, crearíamos una primera generación de calendarios y luego los combinaríamos entre ellos, intercambiando las tareas de los consultores o la fecha a la que habían sido asignada. Sin embargo, esta técnica no es válida porque esos intercambios aleatorios podrían comprometer la consistencia del calendario al hacer asignaciones en días inadecuados o a consultores no cualificados para la resolución de la tarea.
- **GRASP:** Esta técnica supone la mejor elección para abordar el problema. En su proceso de creación montaríamos calendarios válidos basándonos en las restricciones del problema y en su proceso de mejora, al hacer los intercambios uno a uno, podemos comprobar si el cambio es a mejor y si es correcto llevarlo a cabo. Además el número de tareas asignadas a cada consultor permanecería constante, dejando el calendario balanceado de forma natural.

Por lo tanto creemos que GRASP ofrece la mejor aproximación al problema y la más similar a cómo un humano crearía la planificación.

4.2 Aplicación del algoritmo GRASP al problema de la planificación

En este punto describiremos el procedimiento a seguir a alto nivel, sin ceñirnos a ningún lenguaje en concreto, para adaptar el algoritmo GRASP a nuestro problema.

4.2.1 Inicialización

En este bloque se crean las estructuras de datos que contienen la información de las tareas y recursos y se preparan para su posterior manipulación.

Parámetros

- Fechas (desde – hasta): Intervalo de tiempo para el cual vamos a crear el calendario.
- Consultores: Personas a las que vamos a asignarles las tareas

Acciones

- Cargar los datos de los consultores. Incluyendo sus días libres.
- Cargar las tareas de proyectos y SATs abiertos de los módulos a los que pertenecen los consultores.
- Calcular el número de jornadas dedicadas a cada tarea.
 - o Para cada hito/SAT
 - Restarle al número de jornadas estimadas el número de jornadas empleadas.
 - Si es superior a cero
 - o Dividir el número de jornadas restante equitativamente entre todas las tareas activas del hito/SAT.
 - Si no
 - o Solicitar un incremento de jornadas estimadas
- Cargar las tareas de planificación manual que se encuentran dentro del intervalo.
- Borrar de la planificación automática las tareas de tipo planificación manual.
- Cargar las cualificaciones de cada consultor y las requeridas por cada tarea.
- Cargar los festivos.
- Cargar las fechas fin de hitos y SATs.
 - o Iteramos la colección
 - Guardamos el campo fecha fin en una nueva colección de fechas.
 - o Ordenamos la colección de fechas ascendentemente.

Notas

El motivo por el que eliminamos de la planificación las tareas de tipo manual es que asumimos que las rigen unas restricciones no controladas por la aplicación. Por lo tanto respetamos la asignación realizada por el usuario, pero desde el momento que asigna una tarea de forma manual, deberá hacerlo siempre para el resto de jornadas restantes de esa tarea, si las hubiera.

4.2.2 Bucle de control

Este es el método que define GRASP, en él aparecen los métodos principales involucrados en la generación del calendario.

Parámetros

- Nº de calendarios a crear. (*num_cal*)
- Nº de iteraciones del proceso de mejora de cada calendario. (*num_iter*)

Acciones

- Repetir *num_cal* veces: proceso de creación.
- Repetir *num_cal* veces: proceso de mejora con *num_iter* iteraciones de mejora en cada ejecución del proceso.

4.2.3 Proceso de creación

En esta parte debemos hacer una distribución de las tareas entre los consultores seleccionados atendiendo solamente a los criterios de factibilidad.

Parámetros

- Vacío

Acciones

- Mientras quedan tareas que asignar.
 - o Obtener una tarea aleatoria para un consultor en concreto.
 - Da igual si es de SAT o de proyecto, en caso que no queden de un tipo, devolver del otro.
 - Guardar la tarea en una colección de tareas que va a hacer ese consultor.
 - Obtener el resto de consultores que puede hacer esa tarea y guardarnos cuántas cualificaciones tienen de las totales requeridas para realizar la tarea.
- Para cada día dentro del intervalo de fechas.
 - o Si el día no es festivo.
 - Para cada consultor que planificar.
 - Si no tiene el día libre.
 - o Obtenemos una de las tareas de su colección de tareas aleatoria.
 - o Creamos la tripleta fecha-consultor-tarea y la guardamos en una colección tipo calendario.
- Guardamos el calendario en nuestra estructura de datos de calendarios creados.

Notas

Notemos que en esta etapa no se ha tenido en cuenta la duración de la tarea a la hora de crear el calendario. Esto es debido a que:

- Tras hacer varias pruebas haciendo tantas inserciones en el calendario de una misma tarea como jornadas asignadas tiene, nos dimos cuenta que las jornadas dedicadas a la misma tarea quedaban dispersas en el calendario y resultaba muy difícil y costoso computacionalmente reordenarlas en el proceso de mejora.
- Es conveniente limitar el número de frentes abiertos simultáneamente, el ser humano es más eficiente resolviendo problemas de uno en uno que varios a la vez.

4.2.4 Proceso de mejora

En esta sección, el objetivo es coger un calendario factible, creado en el proceso anterior y aplicar técnicas de mejora local para generar una solución más óptima.

Parámetros

- Vacío

Acciones

- Inicializar una variable donde guardaremos el calendario mejor valorado del proceso de mejora con el propio calendario creado.
- Evaluar el calendario creado y guardar la valoración. (Ver punto 4.2.5 Evaluación de asignaciones).
- Repetir *num_iter* veces
 - o Seleccionar aleatoriamente dos consultores diferentes que participan en el calendario.
 - o Obtener la lista de tareas de ambos consultores.
 - o Obtener una tarea aleatoria de cada lista.
 - o Evaluar la puntuación de ambas asignaciones si se intercambiaran las tareas los consultores (Ver punto 3.3.5 Evaluación de asignaciones).
 - o Si el cambio es una mejora o queremos hacer siempre el intercambio
 - Hacemos el intercambio de asignaciones.
 - Actualizamos la lista de tareas de cada consultor.
 - o Cada diez intercambios.
 - Intercalamos un mini-proceso de creación.
 - Para todos los consultores.
 - o Obtenemos su lista de tareas
 - Para cada tarea
 - Obtenemos la fecha de fin del hito/SAT al que pertenece.
 - o Reordenamos la lista de tareas en función de qué tareas tienen la fecha de fin más próxima.
 - o Cada dos intercambios.
 - Evaluamos el estado actual del calendario en proceso de mejora.
 - Si la puntuación obtenida supera la puntuación máxima del proceso.
 - Actualizamos el mejor calendario con el actual.
 - Actualizamos la mejor puntuación obtenida.
- Finalmente, guardamos el mejor calendario del proceso de mejora en una colección de calendarios que han sido mejorados junto con su valoración global.

Notas

Acerca del intercambio de tareas entre consultores

- El motivo por el cual las tareas seleccionadas a intercambio deben de pertenecer a dos consultores diferentes es que la única razón por la que valdría intercambiar dos tareas del mismo consultor sería para intentar que la tarea se hiciera antes de su plazo de entrega, pero este problema queda subsanado con el mini-proceso de creación intercalado.

Acerca de intercambiar siempre las asignaciones sin tener en cuenta la valoración:

- Tras evaluar un posible intercambio, parece lógico que si el cambio no es a mejor, no hacerlo y probar con un nuevo par de tareas. Sin embargo, la visibilidad del proceso de mejora sólo tiene en cuenta la iteración actual y puede darse el caso de que una decisión de intercambio tomada haya mejorado levemente la solución global, pero que impida que nuevos intercambios tengan lugar en las próximas iteraciones porque ninguno mejora la puntuación. Cuando esta situación se produce, quiere decir que hemos encontrado un máximo local, que no tiene por qué coincidir con el máximo global.
- Para evitar esta situación, podemos permitir intercambios que empeoren la solución temporalmente, esperando que un intercambio futuro mejore la puntuación final a cantidades superiores a las del máximo local.
- En cualquier caso tendríamos guardado el máximo local como resultado del proceso de mejora en caso de que no aparecieran mejores candidatos.

Acerca de los mini-procesos de creación.

- Tras varias pruebas de generación de calendarios, nos dimos cuenta de que a pesar de asignarle una gran importancia en la valoración del intercambio a que la tarea fuera planificada antes de su fecha de fin, muy pocas veces conseguíamos que una ejecución completa de GRASP nos devolviera las tareas ordenadas por fecha de entrega y encima se perdía calidad en otros factores como la relación con el cliente o el número de cualificaciones del consultor.
- Por ello y entendiendo como máxima global que debe realizarse primero lo que tiene una fecha de entrega más cercana, decidimos cada ciertas iteraciones del proceso de mejora incluir una reordenación por consultor de sus asignaciones por fecha. Gracias a esto podemos asignarle un mayor porcentaje de la valoración de la asignación a otros aspectos de esta, conservando dicha prioridad de entrega.

4.2.5 Evaluación de asignaciones

Es clave discernir entre las posibles soluciones al problema de clasificación, qué hace a una mejor que su vecina. Para ello, partiendo de los criterios de optimalidad mencionados previamente, se ha definido un proceso formal que permite evaluar y normalizar la calidad de una asignación y por consiguiente, la calidad del calendario completo.

Cada criterio es evaluado individualmente en base 1 y posteriormente escalado en función de la importancia definida por el usuario.

Evaluación de las cualificaciones

El modelado del problema dicta que las tareas tienen una serie de cualificaciones requeridas para poder completarlas. Al mismo tiempo, el consultor ha desarrollado otra serie de cualificaciones en el desempeño de su trabajo. Estas cualificaciones están atadas al módulo relacionado con el desarrollo actual. De este modo:

- Contamos el número de cualificaciones requeridas por la tarea
- Contamos cuantas cualificaciones requeridas por la tarea las dispone el consultor
- Devolvemos:
 - $\frac{\text{Nº de cualificaciones del consultor}}{\text{Nº de cualificaciones de la tarea}}$

Evaluación de la relación con el cliente

El grado de relación entre el consultor y el cliente se evalúa manualmente, con un escalár comprendido en el rango [0,3] cuyo significado es el siguiente:

Escalár	Tipo de relación
0	Mala relación
1	Sin relación
2	Relación regular
3	Buena relación

- Devolvemos:
 - $\frac{\text{Relación con el cliente}}{3}$

Evaluar si la tarea está antes de la fecha límite

Este criterio evalúa cómo de próxima es la fecha de entrega de la tarea (la de su hito/SAT) dentro de la lista de fechas de entrega generada en la inicialización.

Por este motivo comprobamos en qué posición de la lista se encuentra nuestra fecha.

- Si se encuentra dentro del 25% de fechas más próximas
 - Devolvemos 1
- Si se encuentra dentro del 25% de fechas más alejadas
 - Devolvemos 0
- Si no se cumple nada de lo anterior
 - Devolvemos 0.5

Evaluación global de una asignación

Este método, se encarga de hacer las evaluaciones individuales y ponderar el resultado. Para ello:

- Obtiene la valoración de las cualificaciones 'X'
- Obtiene la valoración de la relación con el cliente 'Y'
- Obtiene la valoración de la proximidad de fecha de entrega 'Z'
- Obtiene el peso que le da el usuario a cada criterio
 - o Devuelve:

$$(X * Coef(X)) + (Y * Coef(Y)) + (Z * Coef(Z))$$

- De este modo, por ejemplo:
 - o Para unas valoraciones:
 - X = 1 (Todas las cualificaciones)
 - Y = 0.33 (Sin relación)
 - Z = 1 (Fecha de entrega próxima)
 - o Para unos coeficientes
 - Coef(X) = 0.6
 - Coef(Y) = 0.3
 - Coef(Z) = 0.1
 - o Obtenemos
 - $1*0.6 + 0.33*0.3 + 1*0.1 = 0.79$

Nótese que para el correcto funcionamiento del sistema de evaluación, el peso de los coeficientes siempre debe sumar 1.

Evaluación global de un calendario

Valiéndonos del método anterior, podemos recorrer todas las asignaciones que tiene el calendario y evaluarlas individualmente. Tras cada evaluación acumulamos la puntuación obtenida en un contador.

Por lo tanto, la evaluación de un calendario es la suma de todas las evaluaciones de las asignaciones que lo componen.

En un calendario perfecto, el número de asignaciones corresponde con su puntuación, pues cada asignación tendría una puntuación igual a uno.

4.2.6 Proceso de reconstrucción del calendario

Llegados a este punto, tenemos una colección de pares [calendario mejorado, puntuación] de igual cantidad a los calendarios creados. Tras recorrerla, nos quedamos con el calendario con la puntuación más alta asociada, este es por lo tanto, la mejor solución que hemos podido encontrar con las restricciones que nos han puesto.

Sin embargo, recordemos que esta versión del calendario no incluye todas las jornadas destinadas a esa tarea y además, no hemos tenido en cuenta las tareas de tipo manual. Por lo tanto necesitamos amoldar la información que tenemos a un calendario real.

Para ello:

- Para cada día dentro del intervalo de fechas.
 - Si el día es una asignación manual.
 - Recuperamos la tarea y el consultor asociado y lo insertamos.
 - Saltar a la siguiente iteración.
 - Si el día no es festivo.
 - Para cada consultor que planificar.
 - Si no tiene el día libre.
 - Obtenemos la primera tarea de su lista.
 - Si le quedan jornadas.
 - Asignar esa tarea.
 - Si no
 - Coger la siguiente tarea.

Terminado este proceso estamos listos para presentar los resultados al usuario.

5. Implementación

Ahora que ya tenemos definido a alto nivel el proceso de generación del calendario mediante la metaheurística GRASP, debemos elegir con qué tecnología llevar a cabo el desarrollo de la aplicación. Los factores que se han tenido en cuenta antes de tomar la decisión son los siguientes:

- **Usuario clave:** La persona a la que va destinada la aplicación es un líder de equipo o un gestor de proyectos. Un consultor con potestad para distribuir la carga de trabajo con sus compañeros. El consultor dispone de equipo portátil propio.
- **Escenario de uso:** Aparecen posibles casos de uso de la aplicación, pero principalmente son escenarios de simulación. Algunos son:
 - o Aparición de un nuevo SAT con fecha entregable próxima. ¿Puedo desplazar mi planificación sin que el calendario se vea gravemente afectado?
 - o Aparición de un nuevo proyecto. ¿En cuántas jornadas estimo el desarrollo?, ¿Cuándo podría estar terminado?
 - o Posibilidad de incrementar la plantilla. ¿Cuánta más carga de trabajo puedo soportar teniendo en cuenta las cualidades del miembro? Ante la posibilidad de contratar varios, ¿Dadas las cualificaciones de los candidatos, cual me permitiría ser más eficiente?
 - o Disminución de plantilla. ¿Qué tareas no puedo atender?, ¿Por cuánto tiempo puedo seguir con la carga de trabajo que llevaba si no me reponen el recurso?
- **Recursos de la empresa:** Actualmente SAP funciona principalmente sobre entornos Windows y los equipos de la empresa funcionan sobre ese sistema operativo, además se dispone de una intranet (un *sharepoint*) y otras herramientas de gestión de facturación del trabajo realizado.
- **Sensibilidad de la información:** Los datos contenidos son privados y sólo conciernen al consultor y su cliente. Nadie más está autorizado a verlos.
- **Potencia de computación:** El algoritmo es de tipo voraz y por lo tanto consume muchos recursos del sistema.

Dadas estas restricciones, la decisión ha sido implementar una aplicación de escritorio, que el usuario clave pueda ejecutar en su equipo portátil con potencia suficiente y se guarde la base de datos en una de las unidades cifradas del equipo.

5.1 Lenguaje y entorno de desarrollo.

Definidos los casos de uso de la aplicación y visto que se va a ejecutar sobre una plataforma Windows, en un equipo con arquitectura x64, tres tecnologías destacan por encima de las demás:

- **Java:** Este lenguaje de programación tiene conectividad con cualquier origen de datos gracias a los drivers JDBC, la máquina virtual de Java permitiría ejecutar la aplicación sobre otros sistemas operativos no Windows y tener una portabilidad relativamente simple a Android. Además existen diversos IDEs (*Integrated Development Environment*) que trabajan con Java y permiten un *deploy* rápido, como pueden ser *eclipse* y *NetBeans*.

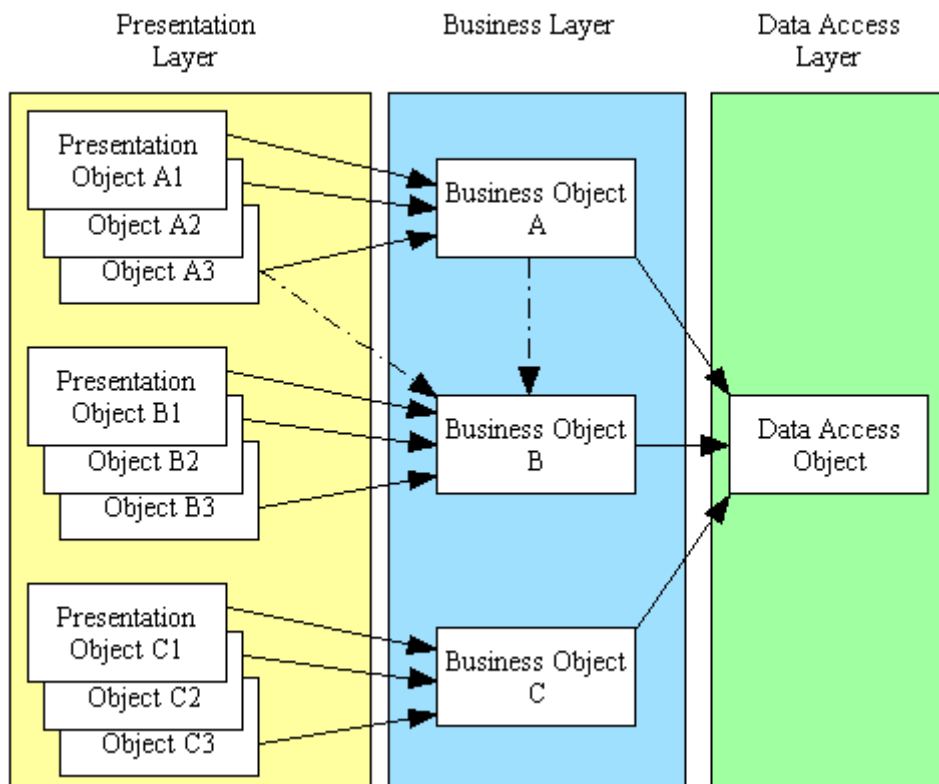
- **C/C++:** Estos lenguajes son dos de los más reconocidos mundialmente. Existe mucha documentación al respecto y permiten una gran eficiencia gracias a la gestión de la memoria. Además, existen librerías de altas prestaciones que permitirían operar con los elementos de datos que manejamos más rápidamente. Sin embargo, la creación de interfaces gráficas con las que interactuar con la lógica de la aplicación es un proceso tedioso y complicado.
- **C#:** Este lenguaje perteneciente al paquete .NET Framework de Windows, guarda similitudes con Java en lo referente a la máquina virtual que lo sostiene por debajo y la forma de acceder a métodos y propiedades de las clases. Además incorpora la potencia de Windows forms para la gestión de ventanas y de ADO.NET para el control de acceso a datos.

De las tres opciones, finalmente optamos por desarrollarlo en el entorno Visual Studio con el lenguaje C#, por las facilidades de gestión visual, acceso a datos, robustez y experiencia de desarrollo usando esa tecnología.

5.2 Implementación del planificador en .NET utilizando Visual Studio

A la hora de dar código a la aplicación, lo haremos utilizando una estructura basada en capas, donde cada capa cumple con una funcionalidad específica y está conectada con la adyacente.

Limitaremos el número de capas a tres, a saber: Acceso a datos, Lógica de control, Presentación



5.2.1 Capa de acceso a datos

Esta capa es la encargada de gestionar la permanencia de los datos cuando no se encuentran en memoria principal, debe de interaccionar con la base de datos y darle a la capa de lógica de control la información que solicite.

Elección de la base de datos

De acuerdo con el análisis de uso realizado, la aplicación va a ser ejecutada en un equipo portátil y debe de estar accesible si el usuario se desplaza y no tiene posibilidades de acceso a internet.

Estas restricciones nos impiden utilizar un motor de gestión de base de datos instalado en un servidor al que la aplicación cliente se conecte y le solicite la información. La base de datos debe encontrarse en el sistema del consultor, el cual no puede permitirse tener ejecutando en segundo plano el motor de la BD. La opción restante es utilizar el almacenamiento basado en ficheros y la mejor opción disponible es utilizar Microsoft Access.

En un fichero Access con extensión *mdb*, podemos declarar tablas, tipos de datos de columnas, esquemas de relación de datos y reglas de actualización de los mismos. Así como seguridad de acceso a la información.

Interacción con la base de datos

Visual Studio incorpora drivers de tipo OLE DB para interaccionar con bases de datos Access.

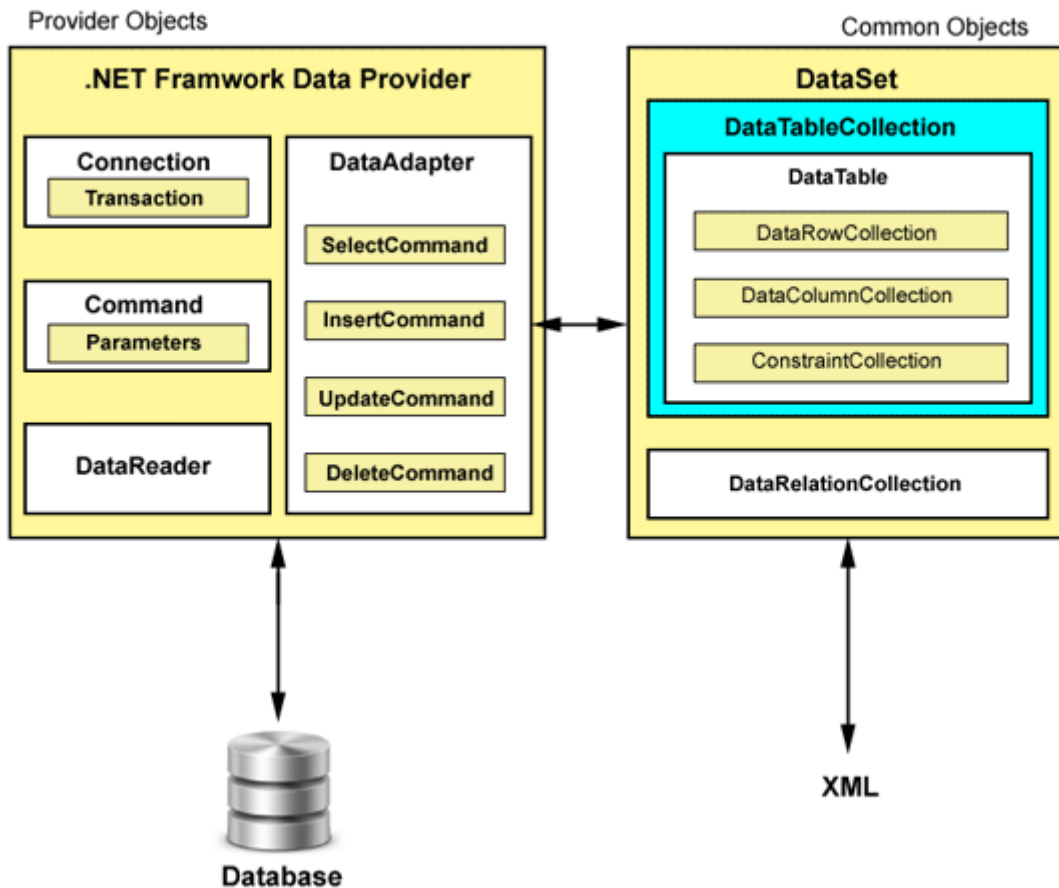
En la creación del proyecto, crearemos una nueva conexión tipo OLE DB contra el fichero Access que es nuestra base de datos y que almacenamos en un directorio de la solución.

Guardaremos la conexión en las propiedades de la aplicación (fichero estándar App.config) para reutilizarla cuando sea necesario y que en caso de que la ruta cambie, solo tengamos que modificarla en un lugar y sin recompilar la aplicación.

ADO.NET dispone de estructuras de datos para la gestión de la información de forma eficiente y robusta. Se agrupa de la siguiente forma:

- Conjunto de datos (DataSet): Un DataSet hace la función de una vista de la base de datos sólo que los datos se encuentran en memoria principal. Dentro del DataSet seleccionamos que tablas de la base de datos queremos utilizar y las relacionamos entre sí. Estas tablas tienen el nombre de DataTable.
- Adaptador de tablas (TableAdapter): Un TableAdapter es un objeto relacionado con el DataTable, este contiene los comandos de selección, actualización y borrado de la Tabla.
- Tabla de datos (DataTable): Un DataTable es una estructura de datos formada por columnas y filas, es una abstracción de una tabla de base de datos real y es el contenedor de los datos de dicha tabla, con la que se relaciona a través del TableAdapter.

ADO.NET Architecture



Forma de trabajo:

- Al inicio de la aplicación, le pediremos a los TableAdapter que carguen los datos de la base de datos es sus correspondientes DataTable.
- Presentamos los datos al usuario y le dejamos manipularlos, siempre sobre la copia contenida en los DataTable.
- Cuando el usuario haya terminado y decida propagar los cambios, detectaremos las diferencias entre la versión en memoria principal y la versión de la base de datos. Estas diferencias serán las aplicadas.

De esta forma conseguimos implantar una forma de trabajo que simula un escenario desconectado, donde el acceso a la base de datos se hace de forma puntual y evita los bloqueos de las tablas. Finalmente, dado que la aplicación está pensada para ser usada por sólo un usuario y que cada usuario dispondrá por lo tanto de su fichero Access, no existen problemas de inconsistencias a la hora de actualizar la base de datos, es decir, el valor que indique el DataTable siempre prevalece al que diga la base de datos.

5.2.2 Capa de lógica de control

También conocida como la capa de negocio, esta sección del software es la encargada de responder a los eventos introducidos por el usuario en la interfaz gráfica y procesarlos consecuentemente.

Es en esta sección donde implementaremos nuestro algoritmo GRASP aplicado a nuestro problema específico de asignación de recursos. Además añadiremos lógica de control para gestionar el correcto acceso a datos y asegurar su consistencia.

Siguiendo las máximas dictadas por la programación orientada a objetos, procederemos a encapsular el algoritmo en una clase donde definiremos sus atributos y métodos, algunos de los cuales serán accesibles desde otras partes de la aplicación para solicitar la planificación de un calendario.

GRASP.cs

Esta es nuestra clase destinada a la definición de la metaheurística, la acompaña la extensión cs que denota el lenguaje en el que está escrito, C#.

Encapsulados en cinco métodos diferentes, encontramos los cinco puntos descritos del diseño de la solución que explicamos en el punto 4.2 Aplicación del algoritmo GRASP al problema de la planificación

Inicialización y bucle de control

```
public GRASP(DateTime desde, DateTime hasta)
    : this()
{
    this.desde = desde;
    this.hasta = hasta;
}

public DB.dbDataSet.CALENDARIODataTable start(int id_modulo)
{
    dt_consultores = ta_consultores.GetData(id_modulo);
    dt_calendarioManual = ta_calendarioManual.GetData();

    dt_tareasProyectoCreacion =
ta_tareas.GetDataBy_tareasProyectoCompletadas(false, id_modulo);
    dt_tareasSATCreacion =
ta_tareas.GetDataBy_tareasSATCompletadas(false, id_modulo);

    borrarTareasManualesDePlanificacion();
    listaFechasFin = cargarFechasFin();
    consultores = obtenerListaConsultores();
    cargarNumJornadasPorTarea();

    for (int i = 0; i < Global.N_CALEDARIOS; i++)
    {
        crearCalendario();
        mejorarCalendario();
    }
    DB.dbDataSet.CALENDARIODataTable final = obtenerMejorCalendario();
}
```

Proceso de creación

```
private void crearCalendario()
{
    //Si no vamos a continuar con la tarea anterior, es necesario hacer una
    //distribución aleatoria en primer lugar.
    while(quedanTareas())
    {
        foreach (DataRow fila in consultores)
        {
            int consultor = (int)fila["Consultor_id"];
            //Cargar aleatoriamente tareas que el consultor está cualificado
            obtenerTareaAleatoria(consultor, out id_tarea, out id_hito, out
            id_proyecto, out id_SAT, out id_cliente);
        }
    }
    //Iteramos todos los dias del periodo
    for (DateTime fecha = desde; fecha.Date <= hasta.Date; fecha.AddDays(1))
    {
        if (!esFestivo(fecha))
        {
            //Iteramos todos los consultores que queremos planificar
            foreach (DataRow fila in consultores)
            {
                int consultor = (int)fila["Consultor_id"];
                if (!diaLibre(consultor, fecha) && hayTareasRestantes)
                {
                    //Recuperar una tarea de las seleccionadas aleatoriamente
                    obtenerTarea(consultor, out id_tarea, out id_hito, out id_proyecto,
                    out id_SAT, out id_cliente);
                    if (id_tarea > 0 && id_cliente > 0 && ((id_proyecto > 0 && id_hito
                    > 0) || id_SAT > 0))
                    {
                        DB.dbDataSet.CALENDARIORow filaCalendario =
                        calendarioTab.NewCALENDARIORow();
                        filaCalendario.id_tarea = id_tarea;
                        filaCalendario.id_hito = id_hito;
                        filaCalendario.id_proyecto = id_proyecto;
                        filaCalendario.id_sat = id_SAT;
                        filaCalendario.id_cliente = id_cliente;
                        filaCalendario.Fecha = fecha;
                        filaCalendario.id_consultor = consultor;
                        //Recupero la lista del diccionario
                        calendarioDic[consultor].Add(filaCalendario);
                    }
                }
            }
        }
    }
    //Guardamos el calendario en la cola de calendarios creados
    calendariosCreados.Enqueue(calendarioDic);
}
```

Proceso de mejora

```
private void mejorarCalendario()
{
while(calendariosCreados.Count() > 0)
{
//Recuperar el calendario creado
Dictionary<int, List<DB.dbDataSet.CALENDARIORow>> calendarioAdaptado =
calendariosCreados.Dequeue();
//Evaluamos el calendario creado para tomar la puntuación de referencia
double puntuacionMax = evaluarCalendario(calendarioAdaptado);
for (int i = 0; i < Global.N_ITERACIONES; i++)
{
//Obtener dos tareas aleatorias
List<DB.dbDataSet.CALENDARIORow> lista1, lista2;
DB.dbDataSet.CALENDARIORow asignacion1, asignacion2;
lista1 = calendarioAdaptado[consultor1];
lista2 = calendarioAdaptado[consultor2];
posicion1 = random.Next(0, lista1.Count());
posicion2 = random.Next(0, lista2.Count());
asignacion1 = lista1.ElementAt(posicion1);
asignacion2 = lista2.ElementAt(posicion2);
DB.dbDataSet.CALENDARIORow asignacion1prima, asignacion2prima;
//Las intercambiamos y evaluamos
double valoracion = evaluarCambio(asignacion1, asignacion2, out
asignacion1prima, out asignacion2prima);
//Si el cambio es a mejor o intercambiamos siempre, procedemos
if (Global.INTERCAMBIAR_SIEMPRE || (valoracion > 0))
{
lista1.ElementAt(posicion1).ItemArray = asignacion2prima.ItemArray;
lista2.ElementAt(posicion2).ItemArray = asignacion1prima.ItemArray;
}
//Meter en el diccionario
calendarioAdaptado[consultor1] = lista1;
calendarioAdaptado[consultor2] = lista2;
//Mini-proceso de creación en el proceso de mejora
if (i % 10 == 0)
{
Dictionary<int, List<DB.dbDataSet.CALENDARIORow>> calendarioOrdenado =
ordenarTareasConsultorPrioridad(ref calendarioAdaptado);
}
//Recalcular evaluacion
if (i % 2 == 0)
{
double puntuacion = evaluarCalendario(calendarioAdaptado);
if (puntuacion > puntuacionMax)
{
puntuacionMax = puntuacion;
mejorCalendario = new Dictionary<int,
List<DB.dbDataSet.CALENDARIORow>>(calendarioAdaptado);
}
}
}
//Guardamos el mejor calendario del proceso de mejora
calendariosMejorados.Enqueue(new CalendarioValoracion(mejorCalendario,
puntuacionMax));
}
```

Proceso de evaluación

El siguiente método se ejecuta para cada tarea perteneciente al calendario.

```
private double evaluarTarea(DB.dbDataSet.CALENDARIORow asignacion)
{
    //Método que evalúa la asignación de una tarea a un consultor en una
    fecha concreta
    return
    (
        evaluarRelacionCliente(asignacion.id_consultor,
        asignacion.id_cliente) * Global.PESO_RELACION_CLIENTE +
        evaluarCualificaciones(asignacion.id_consultor,
        asignacion.id_tarea) * Global.PESO_CUALIFICACIONES +
        evaluarTareaAntesFechaLimite(asignacion.id_proyecto,
        asignacion.id_tarea, asignacion.Fecha) * Global.PESO_FECHA_FIN
    ) / 3.0;
}
```

La aplicación está gobernada por los eventos que desencadena el usuario, tras indicarle a través de la interfaz gráfica qué módulo quiere planificar para qué grupo de fechas, esta le pasará la información al objeto GRASP que construirá el mejor calendario posible.

5.2.2 Capa de presentación

Esta es la sección más externa de la aplicación, tiene la función de interactuar con el usuario y de procesar las acciones emitidas con este, transformándolas en instrucciones comprensibles por la capa de negocio o de lógica de control.

El diseño acordado con el usuario está pensado para ser intuitivo, cada funcionalidad está separada por ventanas de las demás de forma que el usuario sea siempre consciente de donde se encuentra dentro de la aplicación.

La sección principal está principalmente ocupada por un lienzo sobre el que cargaremos los calendarios planificados, ya que toda la información gira en torno al calendario, hemos construido sobre él para que en ningún momento se pierda el objetivo de vista.

La gestión de los demás elementos es accesible a través del menú superior, claramente etiquetado con la funcionalidad asociada a los diferentes controles del menú.



Fecha	consultor1	consultor2	consultor3	consultor4	consultor5
03/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
12/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
13/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
14/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
15/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
16/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
19/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE
20/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE

Las formas de interactuar con la aplicación están definidas en detalle dentro del manual de usuario, adjunto a esta memoria en el **Anexo 1**.

6. Seguimiento de la aplicación

Para el desarrollo de este trabajo hemos definido, implementado y probado un conjunto de funcionalidades clave necesarias para la resolución del problema de clasificación. Lo que se corresponde con la primera iteración de un desarrollo software, pero el margen de mejora es muy amplio y hemos elaborado una lista de puntos de ampliación de la aplicación que podrían ser de interés para el usuario. Son los siguientes:

6.1 Comunicación con la aplicación *Solution Manager* de SAP.

La solución *Solution Manager* es un sistema centralizado de soporte proporcionado a los clientes de SAP como parte del acuerdo de licencia. Debido a la cantidad de productos software SAP y no SAP de los que puede disponer el cliente, SAP lanzó este entorno de gestión con el fin de reducir el coste de mantenimiento y centralizar los procesos de negocio.

La particularidad de este sistema que más nos interesa es que recoge la actividad real del consultor en el sistema del cliente, recoge el tiempo dedicado por cada consultor a cada tarea de un proyecto o SAT al nivel de detalle de horas.

De conectar nuestra aplicación con el *Solution*, podríamos mantener la información de nuestros consultores actualizada y saber el tiempo invertido real por cada uno, de forma que no tendríamos que actualizar manualmente el número de horas empleadas a cada tarea y podríamos tener información que explotar en nuevos encargos sobre el número de recursos necesarios y tiempo estimado de desarrollo.

Los problemas relacionados con esta conexión son claros, sería necesario incluir una capa de seguridad de acceso a los datos pues son confidenciales y además implicaría una serie de acuerdos legales entre la consultora y el cliente para poder acceder a la información.

6.2 Multithreading

Los sistemas de computación actuales disponen todos de varios núcleos en el procesador. Permitiendo obtener mayor potencia de cálculo paralelizando tareas.

La aplicación actualmente lanza el proceso GRASP en fondo, mientras que el hilo principal controla la aplicación, cuando GRASP termina notifica que lo ha hecho y devuelve el calendario que ha encontrado, el cual es presentado en la interfaz.

Pero el proceso GRASP podría acelerarse en gran medida y de una forma relativamente sencilla gracias a la separación semántica que lo caracteriza: el proceso de creación y el de mejora.

Para un desarrollo sostenible de esta mejora que no colapse el sistema del usuario, seguiríamos los siguientes pasos:

- Consultar mediante una llamada a sistema el número de hilos de los que dispone el procesador. De estos, emplearíamos un porcentaje de ellos definido por el usuario en una variable de configuración de la aplicación.
- Al inicializar la aplicación, crearíamos un *pool* de hilos igual en cantidad a la calculada.
- Del total de calendarios a crear, encargáramos a una porción de los hilos que creara una porción de ellos.

- Al mismo tiempo que los calendarios se producen, estos son encolados en una colección de calendarios a mejorar. La porción de hilos restantes rescataría de la cola el primer calendario y lo pasaría por el proceso de mejora. Al terminar, iría a por el siguiente, en caso de no encontrar ninguno, se quedaría bloqueado en el monitor que es la cola, a la espera de que entrara un nuevo calendario.
- Aunque el proceso de creación termine, no borramos los hilos, pues es probable que el usuario lance más planificaciones y la creación y destrucción de los hilos es una carga innecesaria.

6.3 Suministrar el servicio vía web

La empresa dispone de una red intranet donde diferentes artículos y recursos están disponibles para los usuarios. La web principal de esta intranet es un Microsoft Sharepoint.

El framework empleado en el desarrollo de la aplicación (.NET) está ampliamente interconectado entre los componentes de sus distribuciones. El motor de páginas web ASP.NET trabaja con el lenguaje C# con el que se ha implementado la aplicación. Bastaría con crear una web en C#, cuyos controles de visualización son homólogos a los de *Windows Forms* empleados y podríamos reutilizar el algoritmo C# dentro del sitio web.

Publicaríamos la web en un servidor *Windows Server*, con el *Internet Information Service* activo y podría ser accesible por cualquier persona conectada a la intranet de la empresa, además el algoritmo se beneficiaría de los recursos del servidor y no estaría limitado por el equipo portátil del consultor.

El motivo por el que esta solución no se ha llevado a cabo es que la aplicación está en periodo de pruebas dentro de un módulo piloto a la espera de resultados de su uso.

6.4 Planificación multimódulo

Aunque la arquitectura de la aplicación está diseñada para soportar proyectos que requieran la colaboración de varios módulos, actualmente sólo se está explotando para la auto-gestión de los equipos.

La problemática asociada a esta funcionalidad es que se deberían de planificar a los módulos completos y no sólo los proyectos en común, para asegurarnos de que todos los consultores tengan trabajo asignado. Supone un cambio en la interfaz de visualización para contener más información de forma organizada y un problema de recursos. Aumentar la casuística implica aumentar el número de calendarios a crear y de procesos de mejora para encontrar una solución óptima.

6.5 Consultores multimódulo

Algunos consultores poseen conocimientos de más de un módulo, o bien porque se han dedicado a él en el pasado y decidieron cambiar o porque están muy relacionados entre ellos.

Dada esta situación, un consultor solo sería elegible para el módulo que intentamos planificar en este momento, sin tener en cuenta que podría ser de más utilidad en otros SATs o proyectos.

Para alcanzar esta funcionalidad es necesario definir un criterio que determine bajo qué condiciones el consultor puede dedicarse a qué módulo. Por ejemplo, semanas alternas o sólo para momentos excepcionales de mucha carga de trabajo en un módulo en concreto.

6.6 Sección de configuración

Actualmente los parámetros que dirigen el comportamiento de la aplicación y el conjunto de días festivos se encuentran en dos ficheros XML en la ruta de instalación del programa. Estos ficheros contienen información relevante como el número de calendarios a crear o la importancia de cada criterio de optimización.

Resulta un poco pesado su edición manual y lo propicio es crear una sección de la interfaz gráfica para editar estas propiedades, y que en al terminar no sea preciso reiniciar la aplicación para que los cambios tomen efecto.

6.7 Control de versiones

Por la forma en la que se ha diseñado la aplicación, el calendario planificado sólo tiene relevancia en el momento en que es generado, para darle al usuario una visión ajustada del estado de desarrollo de sus proyectos y facilitarle en el proceso la toma de decisiones de negocio.

Sin embargo, puede darse el caso de que una versión del calendario tenga un interés particular y que el usuario decida guardarla de forma etiquetada para compararla con otras versiones en un futuro. Ahora mismo podemos hacer esto copiando el calendario sobre una hoja de cálculo con el clásico comando de Copiar-Pegar de Windows, pero esta funcionalidad permitiría centralizar la gestión de los calendarios generados.

7. Evaluación de laboratorio

Este apartado está destinado a detallar las pruebas de aceptación que se han realizado sobre el planificador con el fin de evaluar la calidad de la solución que nos ofrece así como el rendimiento del mismo.

7.1 Equipo de pruebas

La colección de experimentos se ha llevado a cabo sobre un equipo con las siguientes prestaciones:

Componente	Modelo
Procesador	Intel i7 3930K @3.8GHz
Memoria principal	12 GB DDR3 @1833MHz
Disco duro	SSD @550MB/s lectura secuencial

7.2 Experimentos realizados

Hemos acordado una colección de datos no reales que nos permitirán comprobar de un vistazo la calidad de la solución. Es decir, partimos de una solución esperada y le damos los datos al planificador para ver si es capaz de alcanzarla y en cuanto tiempo.

En esencia tenemos tres criterios de optimización, la relación con el cliente, las cualificaciones y las fechas de entrega. Hemos dispuesto dos tipos de experimentos para valorar los dos primeros y el tercero viene implícito en ambos. Siempre se tiene que intentar planificar las tareas antes que su fecha de finalización.

Por simplificación, las tareas a planificar se han agrupado únicamente en SATs, pues estos tienen relación directa con el solicitante y el proyecto únicamente añade una escala jerárquica más que no es relevante para la calidad de las asignaciones realizadas.

7.2.1 Evaluación de la relación con el cliente

Módulo

Módulo	Cualificaciones
BW	Cualificación1
	Cualificación11
	Cualificación111
	Cualificación2
	Cualificación22
	Cualificación222
	Cualificación3
	Cualificación33
	Cualificación333
	Cualificación4
	Cualificación44
	Cualificación444
	Cualificación5
	Cualificación55
	Cualificación555

Consultores y clientes

Para este experimento, todos los consultores tienen todas las cualificaciones del módulo. Pues queremos asegurar que el criterio que marca la diferencia es la relación con el cliente.

Esta disposición indica con que clientes tiene el consultor una buena relación, las combinaciones que no aparecen tienen el valor *no definidas*.

Módulo	Cualificaciones	Cliente
BW	Consultor1	Cliente1
		Cliente6
	Consultor2	Cliente2
		Cliente7
	Consultor3	Cliente3
		Cliente8
	Consultor4	Cliente4
		Cliente9
	Consultor5	Cliente5
		Cliente10

Tareas

Cada cliente tiene un SAT con una única tarea y esa tarea tiene tres cualificaciones. Los cinco primeros clientes esperan la finalización del desarrollo la primera semana de 2015 y los cinco últimos la segunda semana.

Además, el *Cliente1* tiene un desarrollo específico (*SATmanual*) para el que ha solicitado que sea llevado a cabo por el *Consultor1*. Tras discutirlo con el cliente, se ha decidido que las fechas acordadas para llevarlo a cabo son el 07/01/2015 y 08/01/2015.

Cliente	SAT	Fecha fin	Tarea	Cualificación	Fecha asignación
Cliente1	SAT1	09/01/2015	Tarea1	Cualificación1	
				Cualificación11	
				Cualificación111	
	SATmanual	23/01/2015	TareaManual	Cualificación1	07/01/2015 y 08/01/2015
Cliente2	SAT2	09/01/2015	Tarea2	Cualificación2	
				Cualificación22	
				Cualificación222	
Cliente3	SAT3	09/01/2015	Tarea3	Cualificación3	
				Cualificación33	
				Cualificación333	
Cliente4	SAT4	09/01/2015	Tarea4	Cualificación4	
				Cualificación44	
				Cualificación444	
Cliente5	SAT5	09/01/2015	Tarea5	Cualificación5	
				Cualificación55	
				Cualificación555	
Cliente6	SAT6	16/01/2015	Tarea6	Cualificación1	
				Cualificación11	
				Cualificación111	

Cliente7	SAT7	16/01/2015	Tarea7	Cualificación2
				Cualificación22
				Cualificación222
Cliente8	SAT8	16/01/2015	Tarea8	Cualificación3
				Cualificación33
				Cualificación333
Cliente9	SAT9	16/01/2015	Tarea9	Cualificación4
				Cualificación44
				Cualificación444
Cliente10	SAT10	16/01/2015	Tarea10	Cualificación5
				Cualificación55
				Cualificación555

Importancia de cada criterio de cualificación

Para la ocasión se han fijado los siguientes parámetros en el xml de configuración concernientes al porcentaje de la puntuación que aporta cada criterio:

- Relación con el cliente: 0.6
- Cualificaciones: 0.3
- Fecha fin: 0.1
- Nº de calendarios a crear: 10
- Nº de intercambios del proceso de mejora: 100

Solución esperada

El resultado esperado es que obviando los festivos, el planificador asigne a cada consultor los SATs de los clientes con los que mantiene una buena relación, ordenados por fecha de entrega y respetando las asignaciones manuales.

Resultado de la planificación

Se ha lanzado el planificador para los datos descritos desde el 01/01/2015 hasta la actualidad. Este es el calendario obtenido:

	Fecha ▲	consultor1	consultor2	consultor3	consultor4	consultor5
▶	02/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	12/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
	13/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
	14/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
	15/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
	16/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
	19/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE
	20/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE

Si nos desplazamos a las fechas correspondientes a las tareas manuales para el *Consultor1* y revisamos el bloque de *Vista detallada*, podemos comprobar que se han mantenido las asignaciones manuales y que el consultor puede continuar con la tarea que tenía asignada antes de las fechas de planificación manual.

Fecha	consultor1	consultor2	consultor3	consultor4	consultor5
02/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5

Vista detallada
Consultor: consultor1
SAT: (1), sat1
Tarea: tarea1

Fecha	consultor1	consultor2	consultor3	consultor4	consultor5
02/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5

Vista detallada
Consultor: consultor1
SAT: (11), Sat manual
Tarea: tareaManual

Fecha	consultor1	consultor2	consultor3	consultor4	consultor5
02/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5

Vista detallada
Consultor: consultor1
SAT: (11), Sat manual
Tarea: tareaManual

Fecha	consultor1	consultor2	consultor3	consultor4	consultor5
02/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
12/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10

Vista detallada
Consultor: consultor1
SAT: (1), sat1
Tarea: tarea1

Observaciones

Cumplimiento de las restricciones del problema:

- **Disponibilidad:** Se han respetado los días festivos propios de la época, incluyendo los fines de semana:

Enero 2015						
Lu	Ma	Mi	Ju	Vi	Sá	Do
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

- **Cualificaciones:** Todos los consultores están cualificados para todas las tareas, por lo tanto la restricción de cualificaciones se cumple siempre.
- **Repartición equitativa:** Por último, se aprecia que *Consultor1* tiene dos días más asignados debido a las tareas asignadas manualmente, pero el calendario está globalmente balanceado y todas las personas tienen la misma carga de trabajo.

Cumplimiento de los criterios de optimización del problema:

- **Relación con el cliente:** A cada consultor se le ha asignado las tareas correspondientes de los clientes con los que mantienen una buena relación.
- **Fechas de entrega:** Las tareas están planificadas antes de su fecha de entrega correspondiente.
- **Cualificaciones:** Todos los consultores están cualificados para todas las tareas, por lo tanto el criterio de cualificaciones se cumple siempre.

7.2.2 Evaluación de las cualificaciones requeridas

Módulo

Módulo	Cualificaciones
BW	Cualificación1
	Cualificación11
	Cualificación111
	Cualificación2
	Cualificación22
	Cualificación222
	Cualificación3
	Cualificación33
	Cualificación333
	Cualificación4
	Cualificación44
	Cualificación444
	Cualificación5
	Cualificación55
	Cualificación555

Consultores y cualificaciones

Como este experimento trata de medir la idoneidad de un consultor para realizar una tarea basándonos únicamente en los conocimientos de este sobre su desarrollo, pasaremos a seleccionar que cualificaciones del módulo domina cada consultor

Consultor	Cualificación
Consultor1	Cualificación1
	Cualificación11
	Cualificación111
Consultor2	Cualificación2
	Cualificación22
	Cualificación222
Consultor3	Cualificación3
	Cualificación33
	Cualificación333
Consultor4	Cualificación4
	Cualificación44
	Cualificación444
Consultor5	Cualificación5
	Cualificación55
	Cualificación555

Consultores y clientes

Hemos alterado con qué clientes mantienen los consultores una buena relación, de forma que las tareas que saben resolver son las de clientes con los que no mantienen contacto.

Módulo	Cualificaciones	Cliente
BW	Consultor1	Cliente2
		Cliente7
	Consultor2	Cliente3
		Cliente8
	Consultor3	Cliente4
		Cliente9
	Consultor4	Cliente5
		Cliente10
	Consultor5	Cliente1
		Cliente6

Tareas

Hemos mantenido las mismas tareas que en el experimento anterior.

Importancia de cada criterio de cualificación

Hemos intercambiado la importancia de relación con el cliente por la de las cualificaciones y aumentado la cantidad de calendarios para hallar la solución esperada.

- Relación con el cliente: 0.3
- Cualificaciones: 0.6
- Fecha fin: 0.1
- Nº de calendarios a crear: 30
- Nº de intercambios del proceso de mejora: 100

Solución esperada

El resultado esperado es el mismo que en el experimento anterior. El planificador debe de notar que la importancia recae sobre las cualificaciones y obviar el criterio de la relación con el cliente para obtener la mejor solución.

Resultado de la planificación

Se ha lanzado el planificador para los datos descritos desde el 01/01/2015 hasta la actualidad. Este es el calendario obtenido:

	Fecha ▲	consultor1	consultor2	consultor3	consultor4	consultor5
▶	02/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
	12/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
	13/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
	14/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
	15/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
	16/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
	19/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE
	20/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE

Observaciones

Cumplimiento de las restricciones del problema:

- **Disponibilidad:** Se han respetado los días festivos propios de la época, incluyendo los fines de semana
- **Cualificaciones:** Los consultores sólo tienen tareas asignadas para las cuales están cualificados.
- **Repartición equitativa:** Por último, se aprecia que *Consultor1* tiene dos días más asignados debido a las tareas asignadas manualmente, pero el calendario está globalmente balanceado y todas las personas tienen la misma carga de trabajo.

Cumplimiento de los criterios de optimización del problema:

- **Relación con el cliente:** Este criterio nunca se cumple porque hemos forzado al algoritmo a que así sea.
- **Fechas de entrega:** Las tareas están planificadas antes de su fecha de entrega correspondiente.
- **Cualificaciones:** Por la forma en la que se han declarado las tareas, sus cualificaciones requeridas coinciden plenamente con las de los consultores que pueden hacerlas. Por lo que, para una asignación con una restricción de cualificaciones correcta, la puntuación de este criterio siempre es la máxima.

7.3 Evaluación del rendimiento

Es un hecho que al aumentar la complejidad del problema, debemos de darle más opciones al algoritmo de encontrar una solución que nos sea práctica, y la única forma que tenemos de hacerlo es aumentando el número de calendarios aleatorios a crear y el número de intercambios que se realizarán en cada proceso de mejora. Sin embargo esto tiene un alto coste de computación asociado (al fin y al cabo GRASP es un algoritmo voraz) del cual el usuario debe ser consciente.

Hemos realizado varias planificaciones incrementando el número de calendarios y de iteraciones de mejora y anotado el tiempo empleado por el banco de pruebas en cada proceso de planificación.

7.3.1 Resultados base

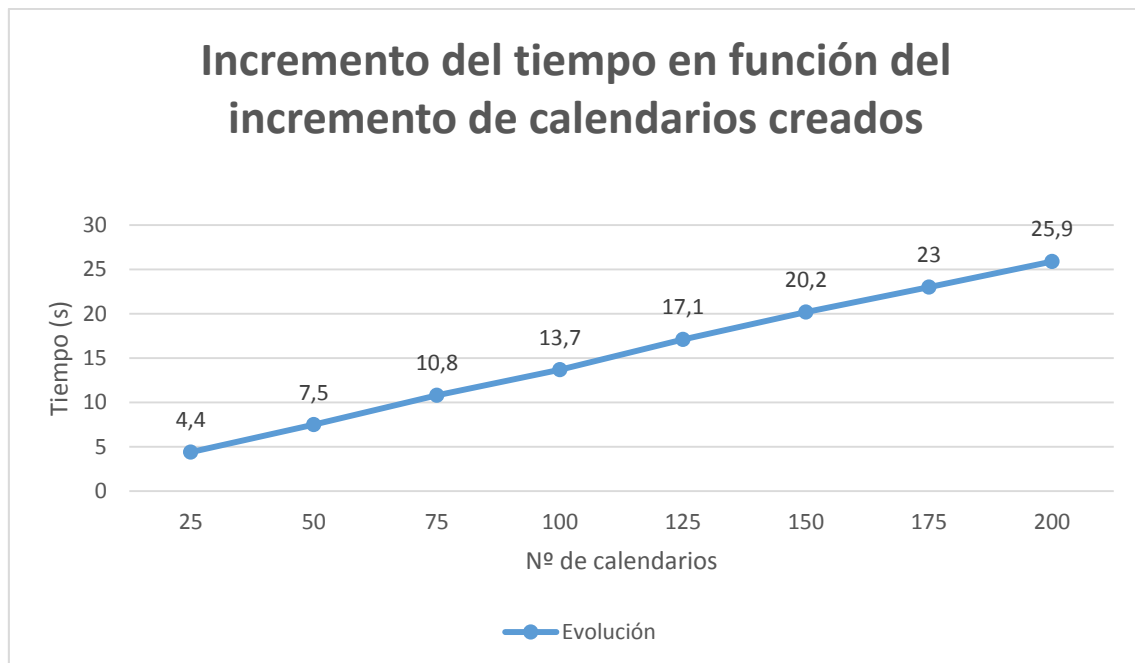
Por defecto, basándonos en la experiencia obtenida con la planificación de los calendarios de laboratorio, la aplicación viene suministrada con los siguientes valores para el número de calendarios y de iteraciones de mejora:

- Nº de calendarios: 30
- Nº de iteraciones de mejora: 100

Estos parámetros emplean una cantidad de 5s (incluyendo el proceso de adaptación de la información a la estructura de datos en la que visualizamos el calendario).

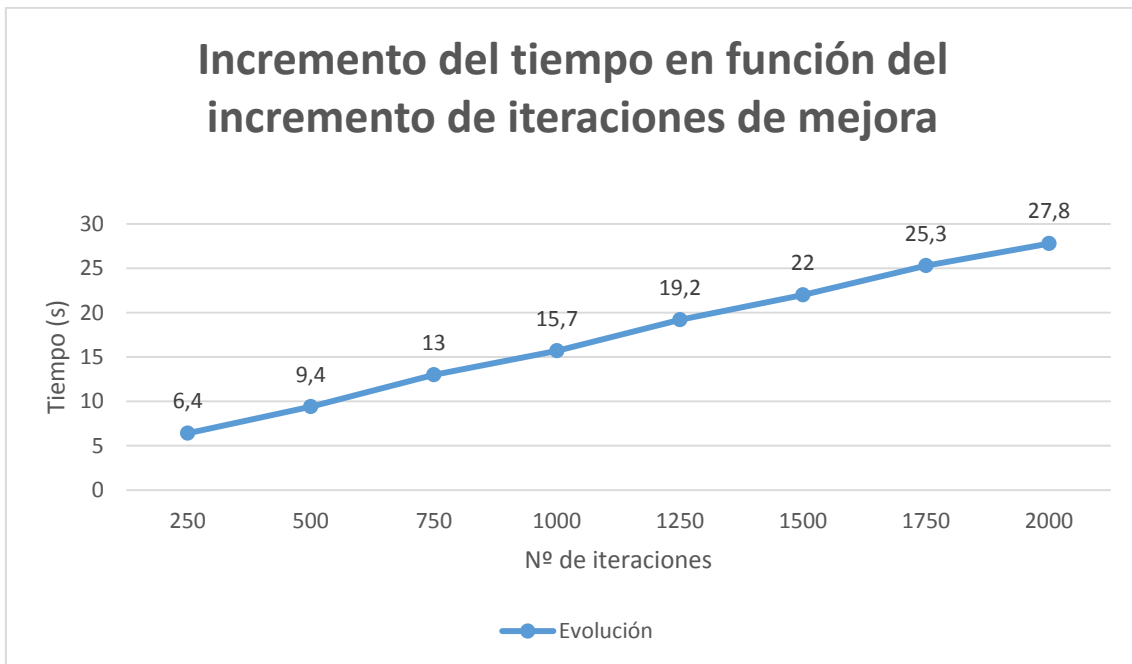
7.3.2 Incremento del tiempo en función del incremento de calendarios creados

Fijando el número de iteraciones de mejora en 100, el efecto de aumentar la cantidad de calendarios que queremos crear es el siguiente:



Por el gráfico observamos que el incremento del tiempo es lineal con el número de calendarios a crear, a razón de poco más de tres segundos por cada veinticinco calendarios más.

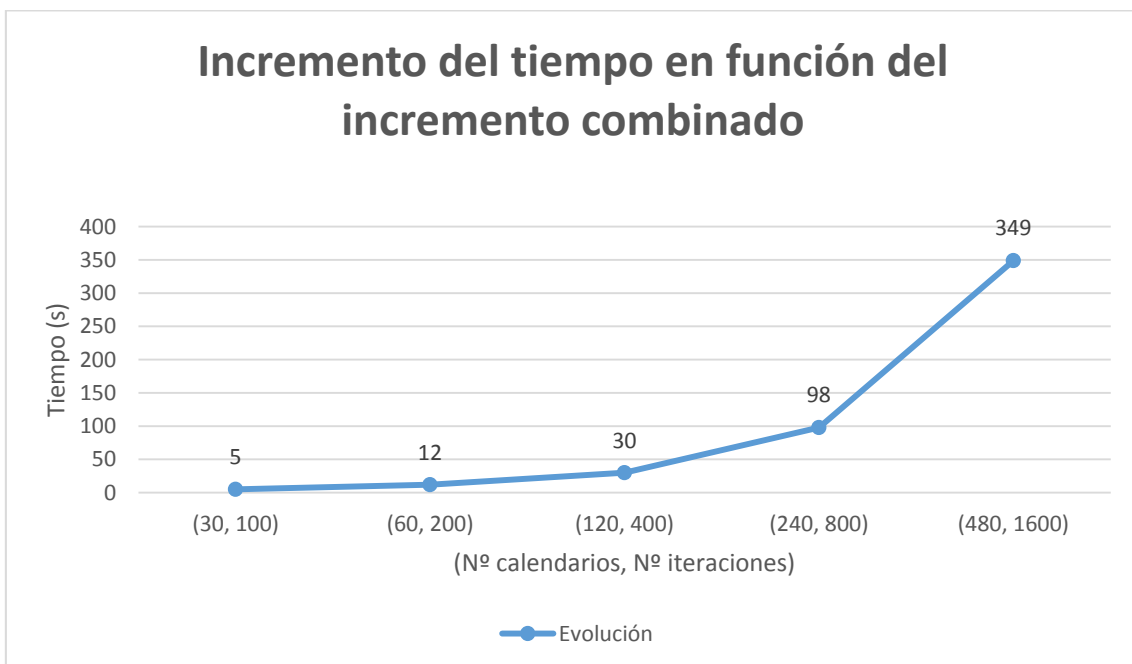
7.3.3 Incremento del tiempo en función del incremento de iteraciones de mejora
 Fijando el número de calendarios a crear en 30, el efecto de aumentar la cantidad de procesos de mejora por cada calendario es el siguiente:



De forma similar que con el número de calendarios, el incremento del tiempo es lineal con el número de iteraciones del proceso de mejora.

7.3.3 Incremento del tiempo en función del incremento combinado

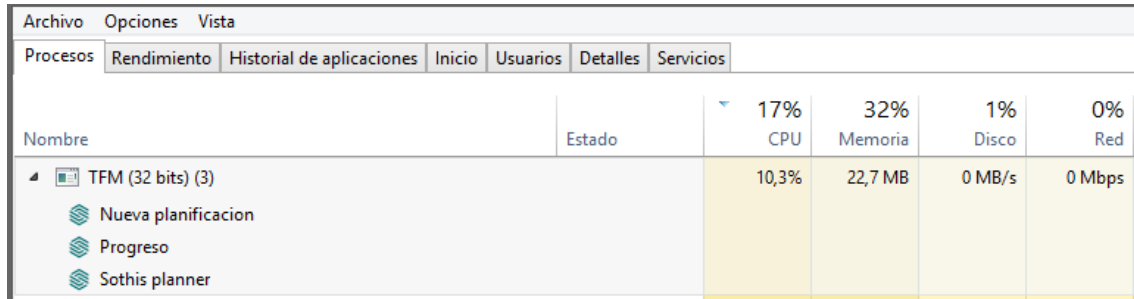
En caso de aumentar los dos parámetros simultáneamente, perdemos la relación lineal y entramos en un aumento del tipo nxm siendo n el nº de calendarios y m el nº de iteraciones de cada proceso de mejora.



7.4 Consumo de recursos

Partiendo de la descripción técnica del punto [7.1 Equipo de pruebas](#), el consumo de recursos del sistema por parte de nuestra aplicación es el siguiente:

- 10,3 % de tiempo de CPU
- 23MB de memoria principal
- Sin uso de disco, los datos necesarios están cargados en memoria para acelerar el proceso.



Procesos		Rendimiento	Historial de aplicaciones	Inicio	Usuarios	Detalles	Servicios
Nombre	Estado	17% CPU	32% Memoria	1% Disco	0% Red		
TFM (32 bits) (3)		10,3%	22,7 MB	0 MB/s	0 Mbps		
Nueva planificacion							
Progreso							
Sothis planner							

8. Evaluación de campo

La aplicación está siendo puesta a prueba en su ámbito de uso habitual, se le han cargado datos referentes a proyectos y SATs activos y estamos evaluando su comportamiento ante problemas reales.

Estos datos no están creados siguiendo un patrón para evaluar el correcto funcionamiento del sistema de una forma rápida y sencilla, sino que representan la situación actual del equipo, donde:

- Puede haber falta de recursos y pocas o ninguna de las tareas consiguen finalizarse a tiempo.
- Algunas cualificaciones no se poseen (por ser una tecnología nueva o simplemente un caso inusual) o las tienen uno o dos miembros del equipo.
- Existen dependencias entre módulos no contempladas en la versión inicial, por ejemplo, podemos estar a la espera de que el equipo de programación termine una implementación para continuar con nuestra parte de la faena.
- Aparecen SATs o errores que es preciso corregir cuando pensábamos que ese desarrollo había terminado y es preciso corregirlo para poder continuar.
- El cliente puede cambiar las prioridades de desarrollo y alterar completamente el orden de la planificación.

Ante todas estas situaciones, no está definido un protocolo de actuación claro, por lo que la mejor opción es incluir los cambios en la aplicación y replanificar, quizás solo a una o dos semanas vista y así ser conscientes de las implicaciones reales del cambio para poder negociar con el cliente o tratar de incorporar más recursos al equipo de desarrollo.

Previamente a esta aplicación, no constaba en una aplicación las tareas a seguir en un desarrollo, sino que basándonos en la experiencia e involucrando al cliente ampliamente en el proceso, se decidía a pocas semanas vista los próximos pasos a seguir, siempre y cuando el cliente no solicitara por contrato unas fechas claras de entrega y penalizaciones en caso de no cumplirse.

Por motivos de confidencialidad no podemos mostrar los datos incluidos en la aplicación, pero podemos describir en qué estado se encuentra la base de datos de la aplicación.

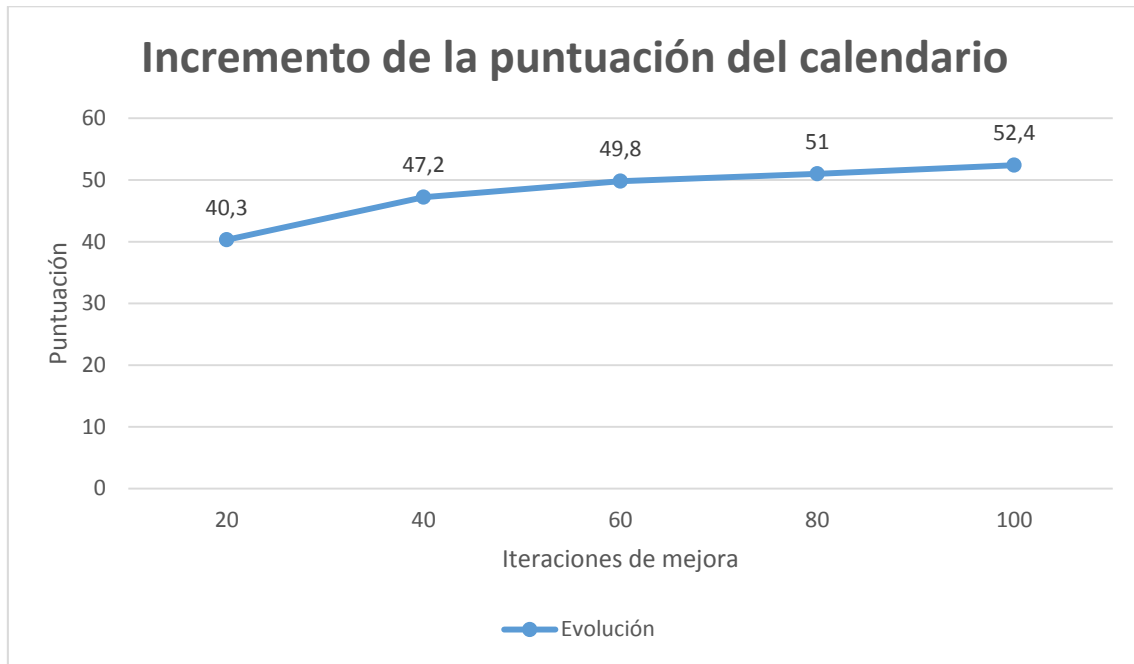
Elemento	Cantidad
Proyectos	1
Hitos	7
SATs	1
Tareas	67
Módulos	1
Consultores	5
Cualificaciones	41
Jornadas estimadas	444

8.1 Evolución de la calidad del calendario en función del número de iteraciones

Hemos realizado varias planificaciones con el estado actual de la planificación. Estos son la media de los resultados obtenidos para diez experimentos con treinta calendarios a crear y cien iteraciones de mejora para cada uno.

Los pesos de los criterios de optimalidad son:

- Relación con el cliente: 0.3
- Cualificaciones: 0.6
- Fecha fin: 0.1



8.1.1 Interpretación de los resultados

Algunas consideraciones a tener en cuenta sobre la lectura del gráfico son las siguientes:

- La aplicación se encuentra en una etapa de pruebas temprana, únicamente tiene cargadas un proyecto en el cual el cliente solicitó una planificación por entregables y un SAT.

El caso habitual es dedicar a un proyecto/SAT un consultor con experiencia, o dos si el segundo está aún aprendiendo o la fecha de entrega está próxima y no se cree que se pueda alcanzar con sólo una persona asignada.

Por lo tanto tenemos a cinco consultores realizando el trabajo de uno o dos.

- Se trata de un equipo reducido con una larga trayectoria y algunos con más de diez años de experiencia en el sector. Por lo tanto, todos están cualificados en prácticamente todos los aspectos del módulo y este aspecto representa el 60% de la puntuación total del calendario.

- Tenemos un total de 67 tareas que asignar, recordemos que una asignación perfecta implica una puntuación de uno por tarea. Hemos conseguido una media de 52,4 puntos de puntuación con una cantidad reducida de calendarios e iteraciones, o lo que es lo mismo, un 78% de asignaciones óptimas sobre la totalidad del problema.
- Podemos observar cómo la puntuación inicial es muy alta, esto es debido a la masa de puntuación de las cualificaciones y que a medida que avanza el tiempo, la puntuación se ve incrementada, pero que pierde empuje conforme se acerca a las iteraciones finales. Este fenómeno sucede porque las tareas seleccionadas para ser intercambiadas son elegidas aleatoriamente y el parámetro que permite intercambiarlas siempre está fijado a falso. Por lo tanto, conforme avanza el tiempo, la cantidad de intercambios que harían mejorar la puntuación desciende en comparación con la cantidad de intercambios posibles, haciendo menos probable que sean escogidos para su intercambio.

9. Conclusiones

Alcanzado este punto de la memoria, me gustaría hacer una pequeña recopilación de los puntos más importantes de este documento.

Hemos podido comprobar la problemática de la asignación de recursos desde un punto de vista global, pasando por un sector en concreto hasta la resolución de un caso particular a modo de ejemplo. Cómo las estrechas relaciones entre los recursos y tareas suponen un problema para la persona que las tiene que distribuir, pues es incapaz de contemplar toda la casuística que las rodea.

Fijando el sector del desarrollo de software a gran escala y el caso particular de las implantaciones de sistemas SAP, tras un análisis detallado de la forma de trabajar de las consultoras, ha sido necesario el desarrollo de una nueva técnica de planificación porque ninguna de las aplicaciones del mercado se adaptaba lo suficientemente bien a los requerimientos.

Creo que la aplicación no está tan refinada cómo las soluciones que proponen las herramientas descritas y que siempre existe margen de mejora, prueba de ello es la sección de seguimiento de la aplicación.

Sin embargo, dadas las condiciones en las que ha sido desarrollada la aplicación, sin un equipo de trabajo y sin la financiación de la que disponen el resto de productos, los resultados obtenidos tienen una alta valoración de optimalidad de acuerdo con los criterios definidos y de fiabilidad. Esto no afirma otra cosa que, las metaheurísticas tienen una gran potencia cómo técnica de distribución de recursos frente a métodos tradicionales y que cada vez se hace más notoria la necesidad de trasladar estas técnicas del mundo académico al cotidiano.

Personalmente, ha sido productiva para mí la decisión de un TFM con orientación profesional, me ha permitido establecer una conexión entre las técnicas y metodologías presentadas en el aula con los requerimientos del mundo laboral. Este nexo no sólo me ha permitido solventar un problema real, sino que deja la puerta abierta para seguir aplicando estas técnicas a futuras situaciones.

Sergio Alejandro Pastor Pastor

10. Bibliografía

10.1 Desarrollo software y gestión de proyectos

Humphrey, W. S. (1989). *Managing the software process* (Hardcover). Addison-Wesley Professional. Humphrey, WS, & Curtis, B. (1991). *Comments on a critical look [software capability evaluations]*. *Software, IEEE*, 8(4), 42-46.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). *Manifesto for agile software development*.

Cockburn, A. (2006). *Agile software development: the cooperative game*. Pearson Education.

Jorgensen, M., & Shepperd, M. (2007). A systematic review of software development cost estimation studies. *Software Engineering, IEEE Transactions on*, 33(1), 33-53.

Kerzner, H. R. (2013). *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

10.2 SAP

Acerca de SAP: <http://go.sap.com/about.html>

Proceso de negocio SAP: <http://searchsap.techtarget.com/definition/SAP>

Fu, H., & Fu, B. (2003). *Sap Bw: A Step-by-Step Guide*. Addison-Wesley Professional.

Egger, N., Fiechter, J. M. R., Rohlf, J., Rose, J., & Schruffer, O. (2006). *SAP BW Reporting And Analysis*. SAP PRESS.

10.3 Distribución de recursos

Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. 104, No. 9). Englewood Cliffs, NJ: Prentice-Hall.

Shen, C. C., & Tsai, W. H. (1985). A graph matching approach to optimal task assignment in distributed computing systems using a minimax criterion. *Computers, IEEE Transactions on*, 100(3), 197-203.

Lo, V. M. (1988). Heuristic algorithms for task assignment in distributed systems. *Computers, IEEE Transactions on*, 37(11), 1384-1397.

Shehory, O., & Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1), 165-200.

Marques-Silva, J. P., & Sakallah, K. (1999). GRASP: A search algorithm for propositional satisfiability. *Computers, IEEE Transactions on*, 48(5), 506-521.

Salman, A., Ahmad, I., & Al-Madani, S. (2002). Particle swarm optimization for task assignment problem. *Microprocessors and Microsystems*, 26(8), 363-371.

Piry, S., Alapetite, A., Cornuet, J. M., Paetkau, D., Baudouin, L., & Estoup, A. (2004). GENECLASS2: a software for genetic assignment and first-generation migrant detection. *Journal of heredity*, 95(6), 536-539.

Rajkumar, M., Asokan, P., Anilkumar, N., & Page, T. (2011). A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research*, 49(8), 2409-2423.

Teoh, C. K., Wibowo, A., & Ngadiman, M. S. (2013). Review of state of the art for metaheuristic techniques in Academic Scheduling Problems. *Artificial Intelligence Review*, 44(1), 1-21.

Anexo 1: Manual de usuario



GrupoSothis

MANUAL DE USUARIO

Sothis Planner

Centro Empresarial Lumiere, Nave 20
Ronda de Auguste y Louis Lumiere, 23
Parque Tecnológico de Paterna
46980 Paterna (Valencia) Spain
Phone. +34 902 88 35 33
Fax. +34 902 90 89 49
www.gruposothis.com



MANUAL DE USUARIO

Sothis Planner

ÍNDICE

1/ DESCRIPCIÓN GENERAL	3
1_1/ FUNCIONALIDADES	3
1_2/ ENTORNO DE USUARIO	3
2/ PLANIFICACIÓN	6
2_1/ NUEVA PLANIFICACIÓN	6
2_2/ PLANIFICACIÓN MANUAL	8
3/ MÓDULO	10
3_1/ NUEVO MÓDULO	10
3_2/ GESTIÓN DE MÓDULOS	11
4/ PROYECTO	13
4_1/ NUEVO PROYECTO	13
4_2/ GESTIÓN DE PROYECTOS	15
5/ SATs	16
5_1/ NUEVO SAT	16
5_2/ GESTIÓN DE SATs	17
6/ CLIENTES	19
6_1/ GESTIÓN DE CLIENTES	19
7/ CONFIGURACIÓN	20
7_1/ VARIABLES CONFIGURACIÓN	20
7_2/ DÍAS FESTIVOS	21





MANUAL DE USUARIO

SOTHIS PLANER

1/ DESCRIPCIÓN GENERAL

Nuestra aplicación sirve para la distribución de las diferentes tareas y recursos humanos en un proyecto/SAT dado. Para entender mejor su funcionamiento vamos a llevar a cabo la realización de un manual de usuario, donde explicaremos paso a paso como usar adecuadamente nuestra aplicación y sacarle el mejor provecho posible.

1_1/ FUNCIONALIDADES

En nuestra aplicación se pueden llevar a cabo las siguientes tareas, todas accesibles desde nuestro menú:

- **Planificación:** Nuestro programa puede llevar a cabo la planificación automática de los recursos y de las tareas desde una fecha dada que le imponemos o puede realizarse de forma manual, por lo que disponemos de dos opciones, crear una planificación nueva o a partir de un proyecto/SAT existente y hacer nosotros mismos la asignación.

- **Módulo:** Sothis está organizada en módulos, por lo que podemos crear un nuevo módulo e ir añadiendo todos los consultores del módulo. Tras esto definiremos las cualificaciones que caracterizan al módulo y cuales las poseen los consultores del mismo.

- **Proyecto:** Entendemos por proyecto un nuevo desarrollo, el proyecto se divide en hitos con sus propias fechas de entrega y dentro del hito encontramos las tareas a realizar. Al igual que en las demás funcionalidades, se puede gestionar un proyecto ya existente, variando todas las variables que hemos nombrado.

- **SATs:** Como sabemos, los SATs son incidencias puntuales, por lo que no necesitan la estructura propia de los proyectos. En este caso las tareas pertenecen directamente al SAT, el cual tiene la fecha de entrega y el identificador del *Solution Manager*.

- **Clientes:** Puesto que los proyectos/SATs son propuestos por nuestros clientes, necesitamos que una funcionalidad de la aplicación sea la organización de estos y su relación con nuestros consultores, para saber que consultor es mejor opción para un determinado cliente.

1_2/ ENTORNO DE USUARIO

Ahora que ya conocemos las funcionalidades, vamos a visualizar la interfaz de nuestra aplicación, de una forma general, para poder saber dónde se localiza cada aspecto.

Como hemos dicho en el punto anterior, todas las actividades están en (*Planificación, Módulo, Proyecto, SATs, Clientes*).

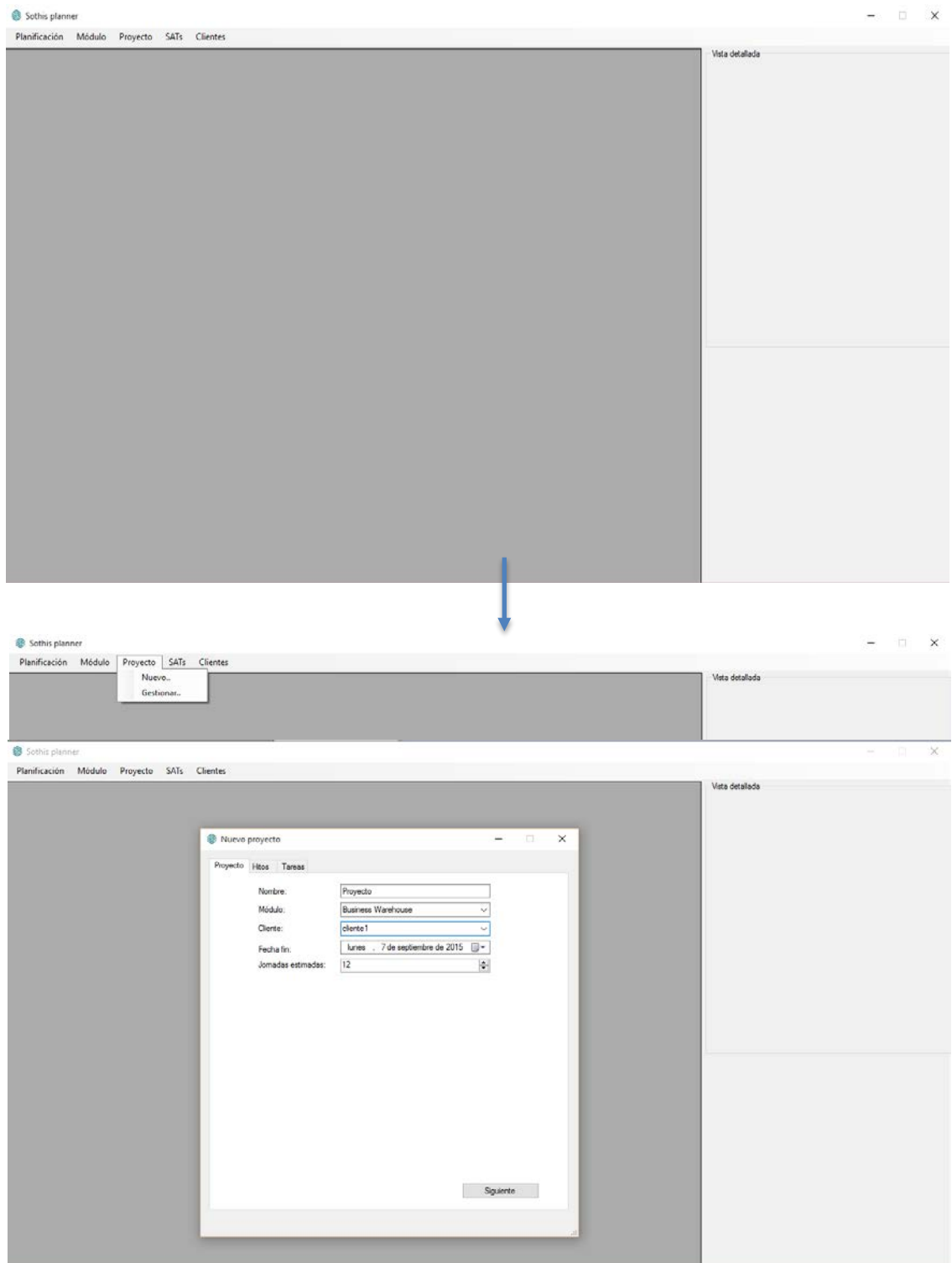


Figura 1. Demostración del funcionamiento de la interfaz



Nuestro programa principal es una ventana sin ningún tipo de menús interiores ni formularios, sino que para que nos aparezca cada ventana debemos darle a la opción a la que queremos acceder.

Como podemos ver en la **Figura 1**, si por ejemplo lo que queremos es acceder a la creación de un nuevo proyecto, tendríamos que hacer clic en **Proyecto > Nuevo...** y cómo podemos ver nos aparecería una nueva ventana a partir de la cual iríamos rellenando el formulario para realizar la acción deseada.

Cada una de estas funcionalidades que hemos nombrado primeramente se describirán en apartados independientes para entender en detalle su funcionamiento.



2/ PLANIFICACIÓN

Esta funcionalidad es la más importante de nuestra aplicación, puesto que se encarga de la distribución de las tareas de forma automática. Cuando planificamos elegimos para qué módulo de los que tenemos dados de alta queremos hacer la planificación y para qué periodo de tiempo.

2_1/ NUEVA PLANIFICACIÓN

Para realizar una nueva planificación debemos acceder en el menú Planificación > Nueva.. y nos aparecerá la ventana que se muestra en la **Figura 2** al realizar clic.

Tendremos como primer campo el módulo que queremos gestionar al que se le habrá asignado previamente un proyecto/SAT (se explicará en los apartados correspondientes). Luego tendremos que poner el periodo en el que queremos planificar dicho modulo, es decir, la fecha de inicio y la fecha de fin que elegiremos en un calendario que se abrirá al hacer clic sobre el recuadro de *Fecha*.

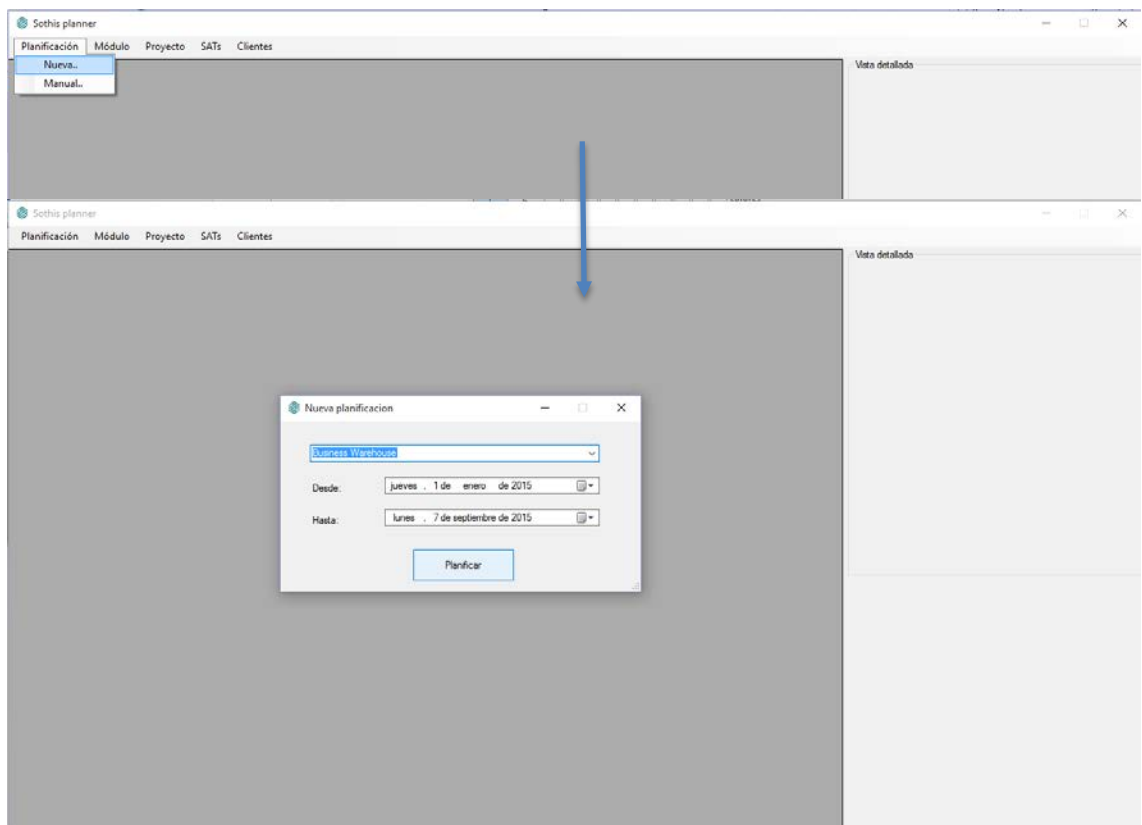


Figura 2. Creación de nueva planificación

Tras pulsar el botón *Planificar*, esperamos a que el proceso de creación del calendario termine, denotado porque la barra de carga se completará cómo se puede observar en la **Figura 3**.

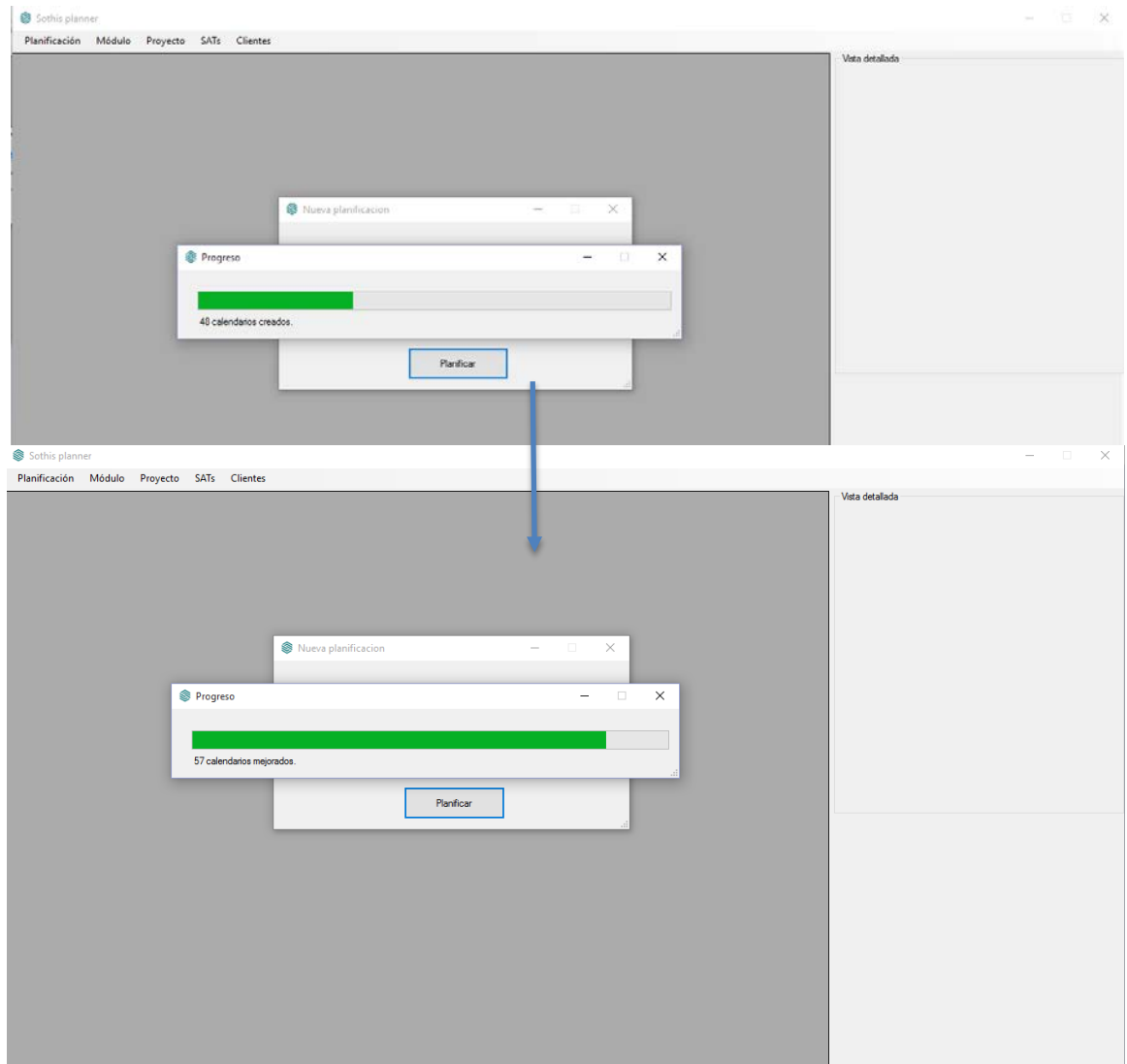


Figura 3. Carga y mejora de los calendarios

Una vez que haya acabado este dialogo donde se muestre que la barra esta completada **debemos cerrarlo**, entonces, el cuadro gris del fondo quedará relleno con la planificación generada.

Como podemos ver en la **Figura 4**, la planificación se mostrará en forma de tabla. Cada fila es un día laboral y cada columna representa a un consultor del módulo

Al hacer clic en cada celda denotada por el nombre del cliente, podremos ver en la zona de *Vista detallada* más información sobre la tarea, tales como hito/SAT y descripción de la misma.



Fecha	consultor1	consultor2	consultor3	consultor4	consultor5
02/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
05/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
07/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
08/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
09/01/2015	cliente1	cliente2	cliente3	cliente4	cliente5
12/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
13/01/2015	cliente1	cliente7	cliente8	cliente9	cliente10
14/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
15/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
16/01/2015	cliente6	cliente7	cliente8	cliente9	cliente10
19/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE
20/01/2015	cliente6	LIBRE	LIBRE	LIBRE	LIBRE

Figura 4. Nueva planificación

2_2/ PLANIFICACIÓN MANUAL

En el caso de que queramos asignar nosotros tareas ya existentes debemos realizar una planificación manual. Para ello accederemos a **Planificación > Manual..**

The image shows the 'Planificación' menu with 'Manual..' selected. Below it, the 'Planificación manual' dialog box is open, containing the following fields:

- Equipo: Fecha: Lunes, 7 de septiembre de 2015
- Módulo: Business Warehouse
- Consultor: consultor1
- Cliente: cliente1
- Proyecto: Proyecto
- SAT: SAT
- Nombre: Implantación BI/BO
- Hto: 1
- Tarea: 1

Buttons: Planificar, Guardar cambios

Figura 5. Inicio planificación manual

Estas tareas que vamos a llevar a cabo en la planificación manual deben ser tareas que han sido creadas manualmente, como veremos más adelante en el apartado de *Proyectos y SATs*.

Una vez hemos accedido en el menú, nos aparecerá una ventana como la que se muestra en la **Figura 5**. Al igual que en la creación de una nueva planificación, debemos asignar el módulo que tiene que trabajar con ese proyecto y la fecha.

Además debemos añadir otros campos tales como el consultor encargado del proyecto y el cliente que ha solicitado el desarrollo. Se ha de definir por tanto si la planificación que se va a llevar a cabo es una planificación de un SAT o de un proyecto.

Si la planificación es de un proyecto debemos marcar el opción *Proyecto* (es el marcado por defecto) y se activarán las opciones disponibles. Se ha de elegir un proyecto que ya haya creado en la sección de *Nombre*, y los hitos y las tareas en las secciones de *Hito* y *Tarea*.

En el caso de que sea un SAT y no un proyecto (**Figura 6**) tenemos que marcar la opción *SAT* ya que esta no es nuestra opción por defecto. Entonces se activarán las opciones de SAT, que serán las mismas que en los proyectos con la diferencia de que estos no tendrán hitos.

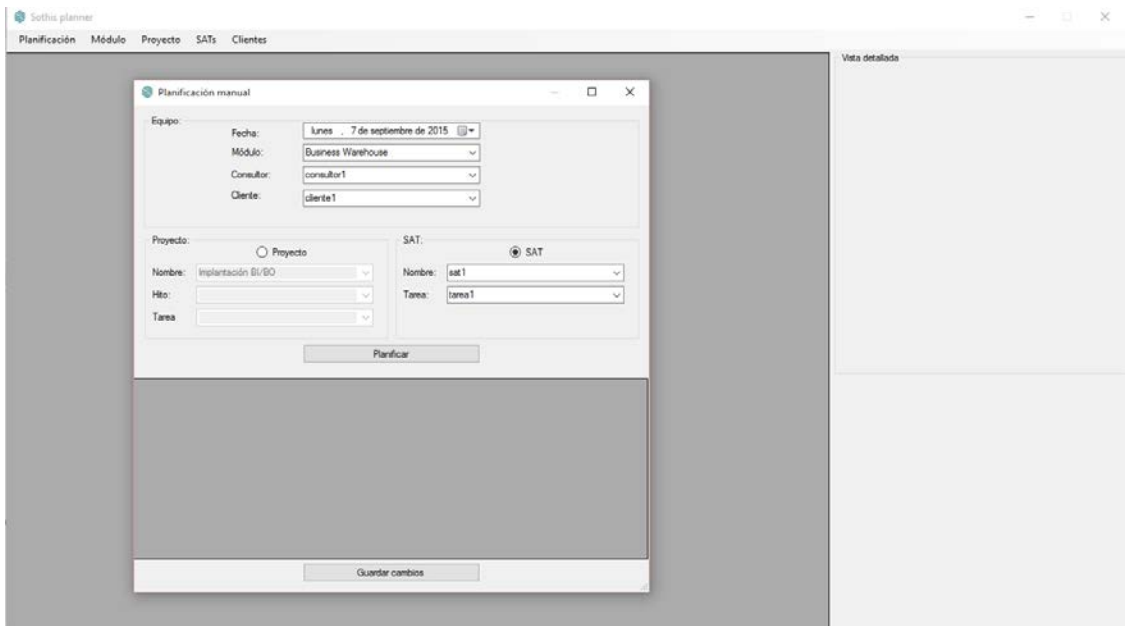


Figura 6. Planificación manual SAT.

Al pulsar *Planificar* creamos una entrada de tipo planificación manual, al terminar pulsamos *Guardar cambios* y propagaremos las modificaciones a la base de datos.

Nota Importante: Las asignaciones de tipo manual serán ignoradas por el planificador, este respetará las fechas para las cuales el recurso ha sido reservado, pero no tendrá en cuenta esta tarea para la asignación de recursos. Queda totalmente delegado en el usuario la asignación de esa tarea en el futuro.

3/ MÓDULO

Ahora vamos a pasar a la fase de módulos. En los módulos es donde organizamos nuestro equipo de trabajo, puesto que tenemos diferentes módulos que se dedican a diferentes áreas de SAP. Tendremos que organizar los diferentes consultores, teniendo en cuenta sus cualificaciones y sus vacaciones, y teniendo en cuenta también las cualificaciones que tiene en su haber cada consultor.

3_1/ NUEVO MÓDULO

Para la creación de un nuevo módulo debemos acceder a **Módulo > Nuevo...** A partir de aquí nos saldrá una nueva ventana como podemos ver en la **Figura 7**.

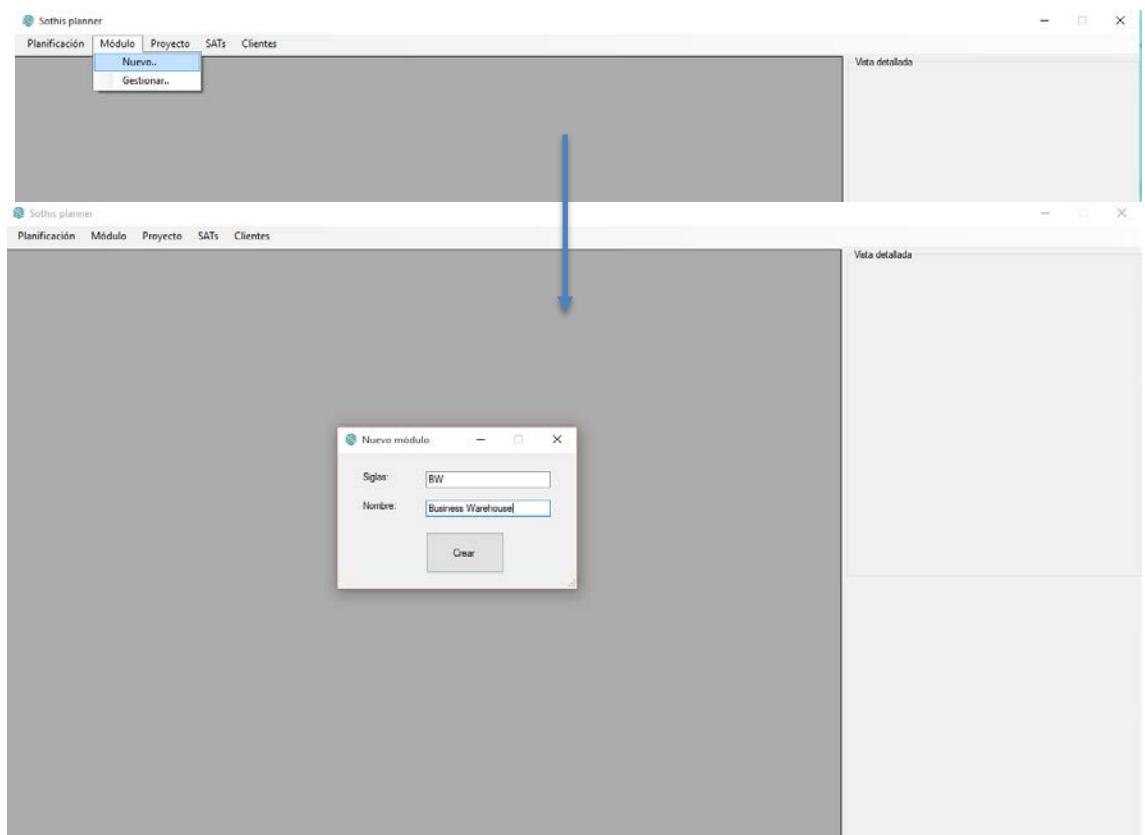


Figura 7. Creación nuevo módulo

En esta ventana nos aparecerán dos campos a rellenar, que serán las siglas que le daremos a nuestro módulo (SD, FI, IC, MM, BW, CO,...) y el nombre completo que le queremos asignar a nuestro módulo.

Una vez que tengamos esos campos completos le daremos al botón *Crear* y nos creará un módulo nuevo que podremos modificar y completar en nuestra funcionalidad de gestión de módulos.

3_2/ GESTIÓN DE MÓDULOS

Para la creación de un nuevo módulo debemos acceder a **Módulo > Gestionar...** Al hacer clic nos aparecerá una nueva ventana como la que se muestra en la **Figura 8**.

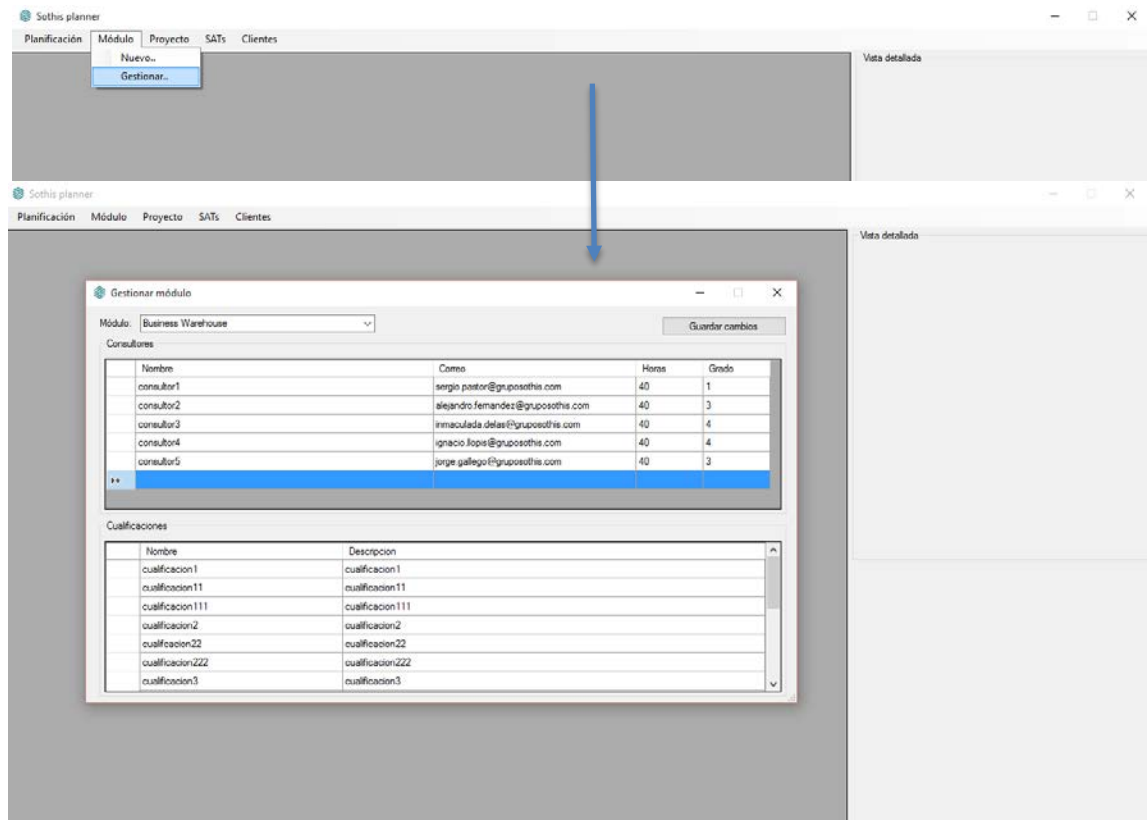


Figura 8. Gestión de módulos.

En esta ventana nos aparecerán una serie de opciones relacionadas con uno de los módulos existentes. Primero tenemos que elegir el módulo que queremos gestionar, haciendo uso del *ComboBox* superior.

Tras esto, nos saldrán debajo los consultores asignados a ese módulo y las cualificaciones necesarias del **MÓDULO**.

En la sección de Consultores si queremos añadir un nuevo consultor hacemos clic en la fila y añadimos los campos *Nombre*, *Correo*, *Horas* y *Grado*. Esas son las horas disponibles de su jornada laboral y su nivel (1.- Junior, 2.- Funcional, 3.-Senior, 4.- Experto).

Volviendo a la **Figura 8**, en el caso que queramos añadir más cualificaciones a nuestro módulo tendremos que seguir los mismos pasos que para añadir un nuevo consultor, se escribirá en una nueva fila la cualificación que caracteriza al módulo, poniendo el nombre de la cualificación en el campo *Nombre* y una descripción de esta cualificación en el campo *Descripción*.

Si hemos acabado de realizar los cambios correspondientes pulsaremos el botón *Guardar cambios* para actualizar la base de datos con los cambios que han sido realizados.

Si se quiere realizar una edición de los datos de un consultor debemos hacer doble clic sobre la fila donde se encuentran los datos de dicha persona sobre la flecha ▶ y aparecerá una ventana como se puede visualizar en la **Figura 9**.

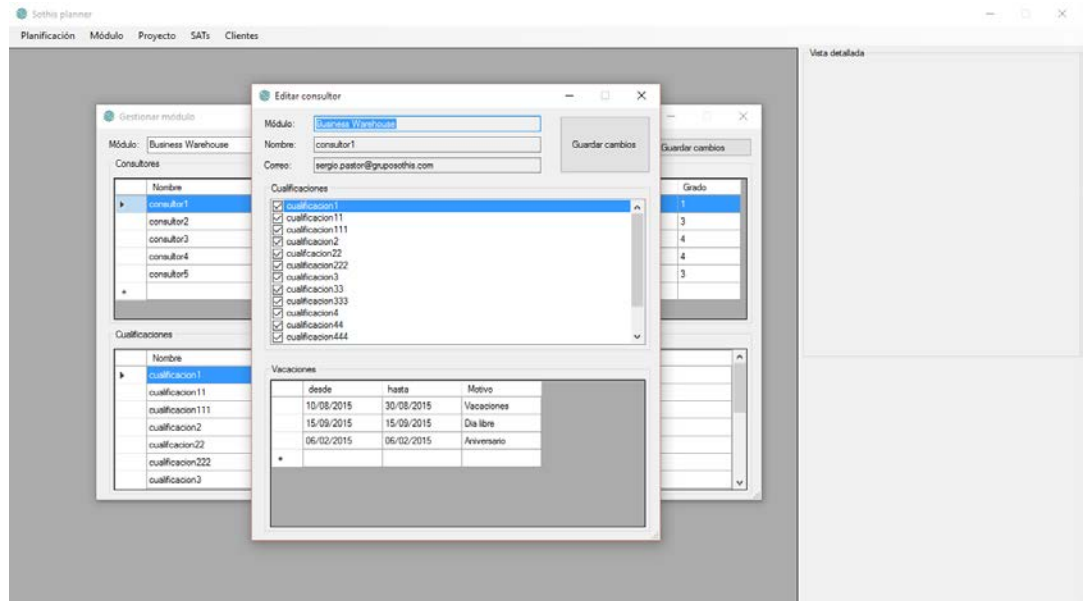


Figura 9. Modificación de los datos del consultor.

Cuando aparece esta ventana podemos editar las calificaciones del consultor y sus días de vacaciones.

Más abajo nos aparece una lista con las calificaciones del módulo, marcaremos aquellas que el consultor domine. Un consultor solo puede realizar aquellas tareas cuyas calificaciones sean iguales a las suyas (al menos una), por lo que cuantas más calificaciones, mayor diversidad de tareas puede hacer.

Por último en el apartado de vacaciones se marcará el periodo en el que cada consultor elegirá sus vacaciones, para saber que ese recurso no estará disponible en ese periodo de tiempo.

Al igual que en la ventana principal de la gestión de módulos, si hemos acabado de realizar los cambios en los consultores correspondientes pulsaremos el botón *Guardar cambios* para actualizar la base de datos con los cambios que han sido realizados.

4/ PROYECTO

En esta funcionalidad veremos cómo dar de alta y gestionar proyectos.

4_1/ NUEVO PROYECTO

Para la creación de un nuevo proyecto debemos hacer clic en **Proyecto > Nuevo..**

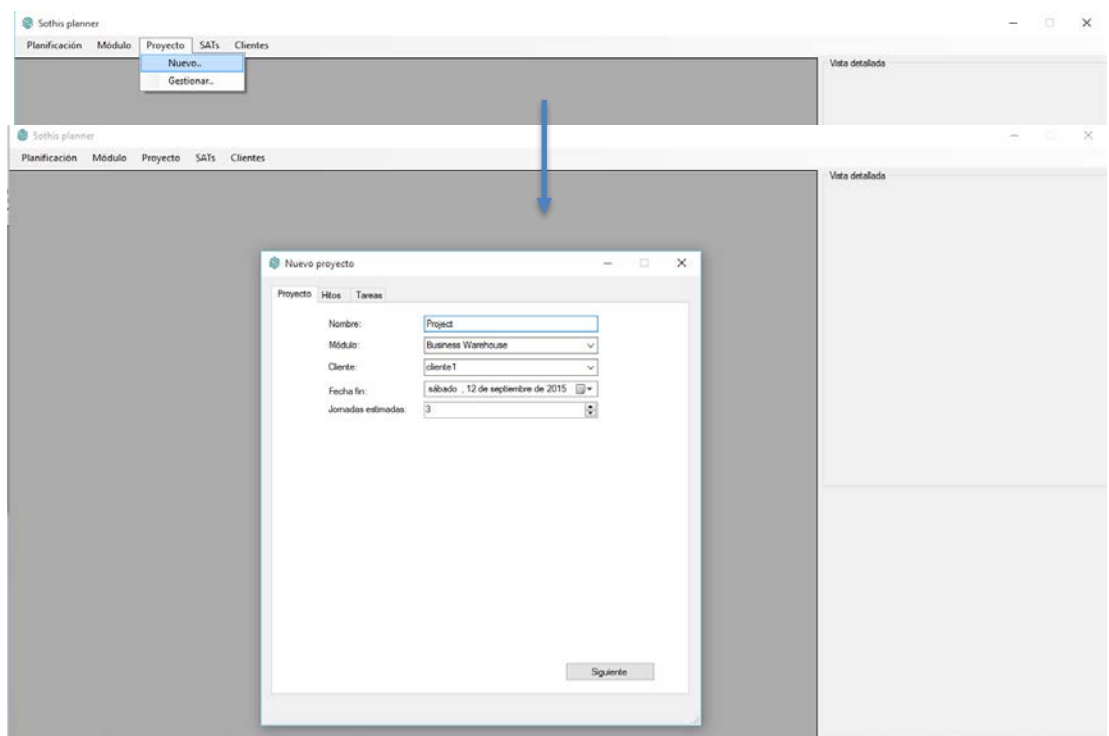


Figura 10. Creación de un nuevo proyecto

Como podemos ver se nos mostrará la siguiente ventana (**Figura 10**). Aquí tendremos que asignar un nombre a nuestro proyecto, elegir el módulo que lo va a llevar a cabo y el cliente que nos ha pedido este proyecto, así como el número de jornadas estimadas de desarrollo y la fecha en la que se ha pactado la finalización del proyecto.

Una vez que hayamos completado estos campos, podemos darle al botón *Siguiete* y nos aparecerá la ventana para crear los diferentes hitos del proyecto (**Figura 11**). En esta ventana debemos escribir el nombre, una breve descripción, las jornadas y la fecha en la que debe acabar dicho hito.

Si hemos dado de alta algún hito y queremos regresar a la ventana principal, nos aparecerá un mensaje advirtiéndonos que se borrarán los datos introducidos, por lo que es aconsejable revisar mejor la primer ventana de creación para que no sea necesario retroceder.

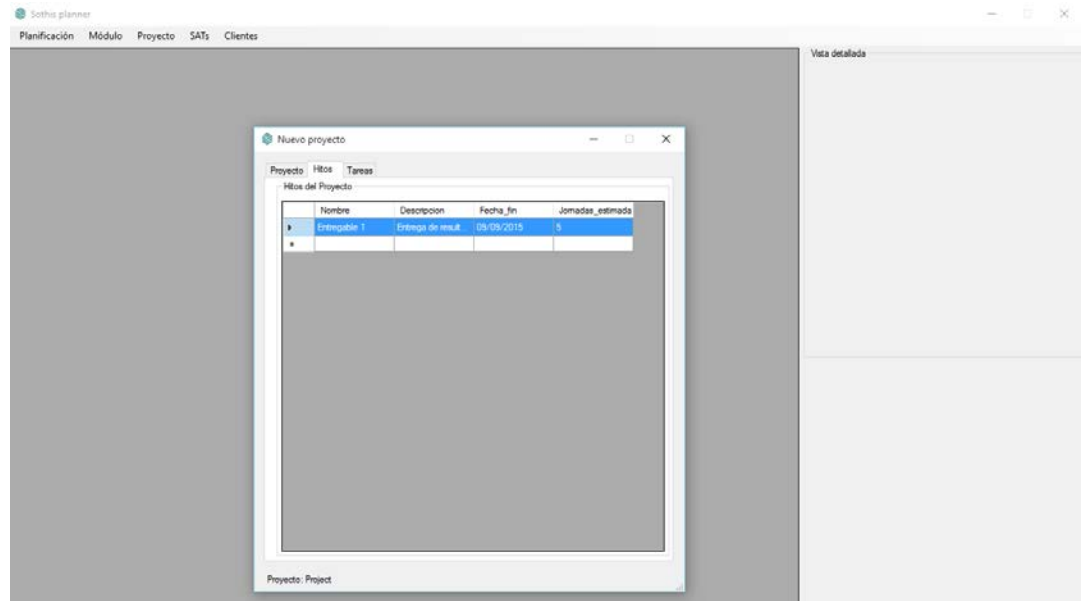


Figura 11. Creación Hito.

Una vez ya hemos creado nuestro hito, podemos hacer doble clic sobre él para poder añadir tareas a ese hito en concreto. Como podemos ver en la **Figura 12**, se nos mostrará la siguiente ventana donde podremos ir añadiendo una breve descripción de las tareas y sus cualificaciones necesarias.

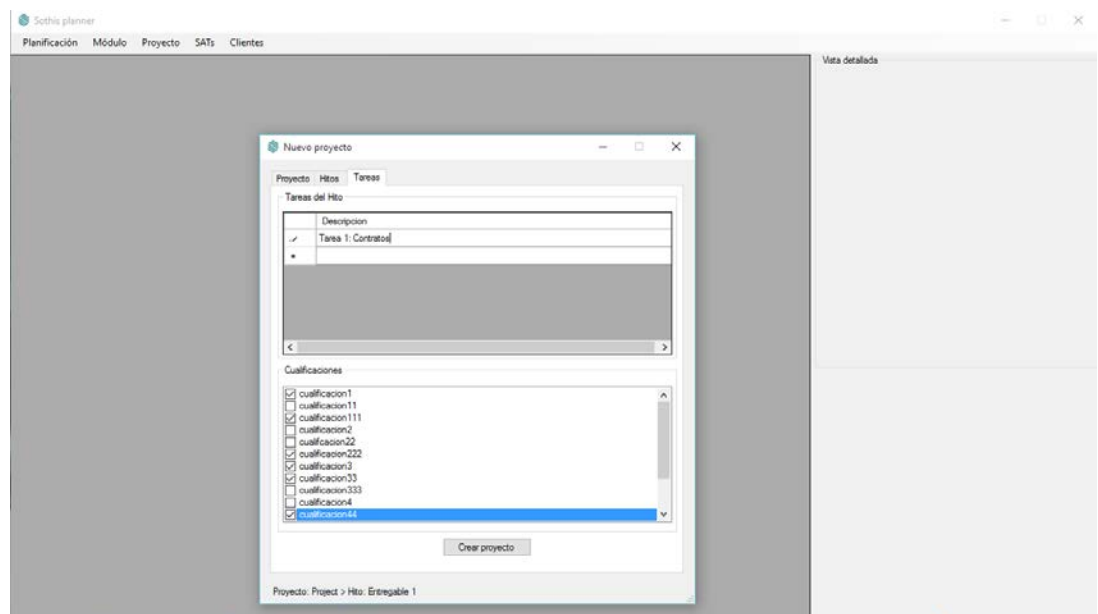


Figura 12. Creación tarea.

Quando hayamos finalizado la creación de nuestra tarea, podemos volver a la pestaña de Hitos para crear más hitos y tareas sin que se eliminen los datos. Cuando hayamos acabado de crear todas, le daremos al botón *Crear proyecto* y se mostrará un dialogo diciéndonos que el proyecto ha sido creado.

4_2/ GESTIÓN DE PROYECTOS

Para la gestión de nuestros proyectos pulsamos **Proyectos > Gestión..**

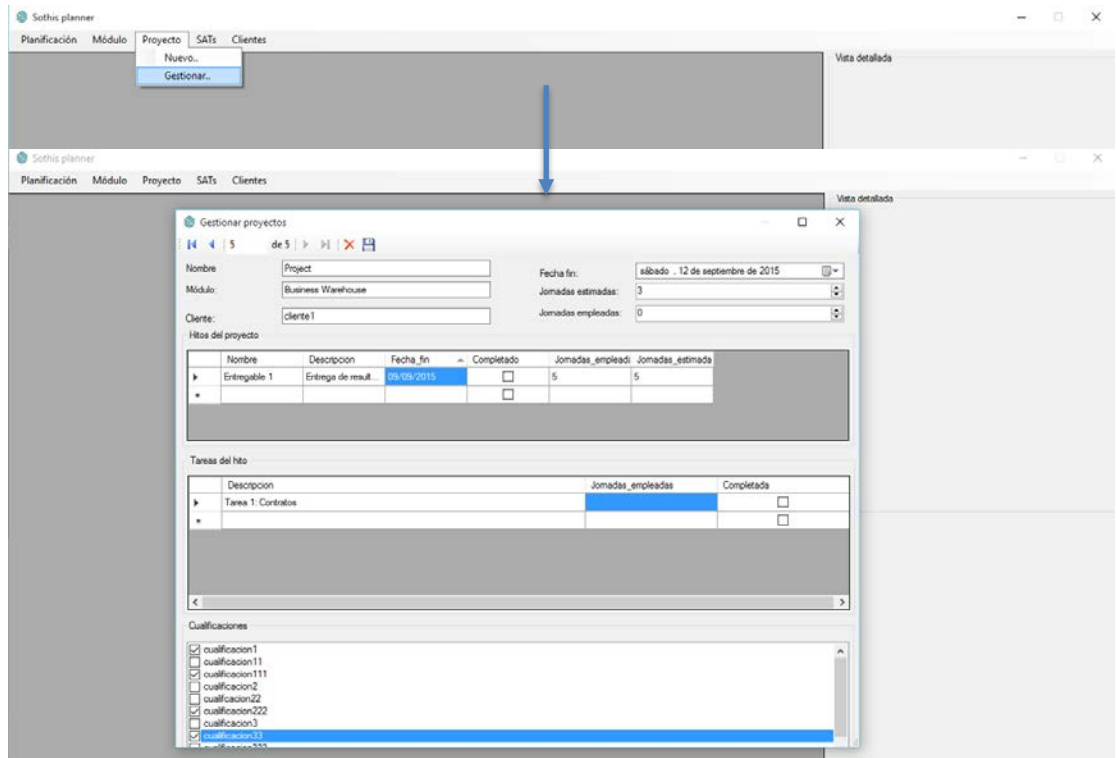


Figura 13. Gestión de proyectos.

Como podemos ver en la **Figura 13**, se nos abrirá la siguiente ventana donde podremos realizar una edición de nuestros proyectos ya creados. Si pulsamos en la flecha que se encuentra en la parte superior, podremos ir viendo los diferentes proyectos que tenemos creados y elegir uno para gestionar.

En esta ventana, podremos editar toda la información que hemos añadido previamente en la creación del proyecto.

Debajo tendremos dos apartados, uno será la edición de los hitos y otro la edición de las tareas. Podremos cambiar también todos los datos que hemos añadido en su creación (nombre, descripción, fecha, jornadas,...) y además podremos marcar tanto en la tarea como en el hito si estas han sido completadas.

Las tareas completadas serán ignoradas para la planificación, como es de suponer.

También podremos modificar las cualificaciones requeridas para cada tarea. Una vez hayamos realizado los cambios oportunos, pulsaremos el botón de guardado que se encuentra en la parte superior de la ventana.

5/ SATs

En esta parte vamos a llevar a cabo la creación y edición de nuestros SATs o incidencias.

5_1/ NUEVO SAT

Para la creación de un nuevo SAT debemos hacer clic en **SAT > Nuevo..**

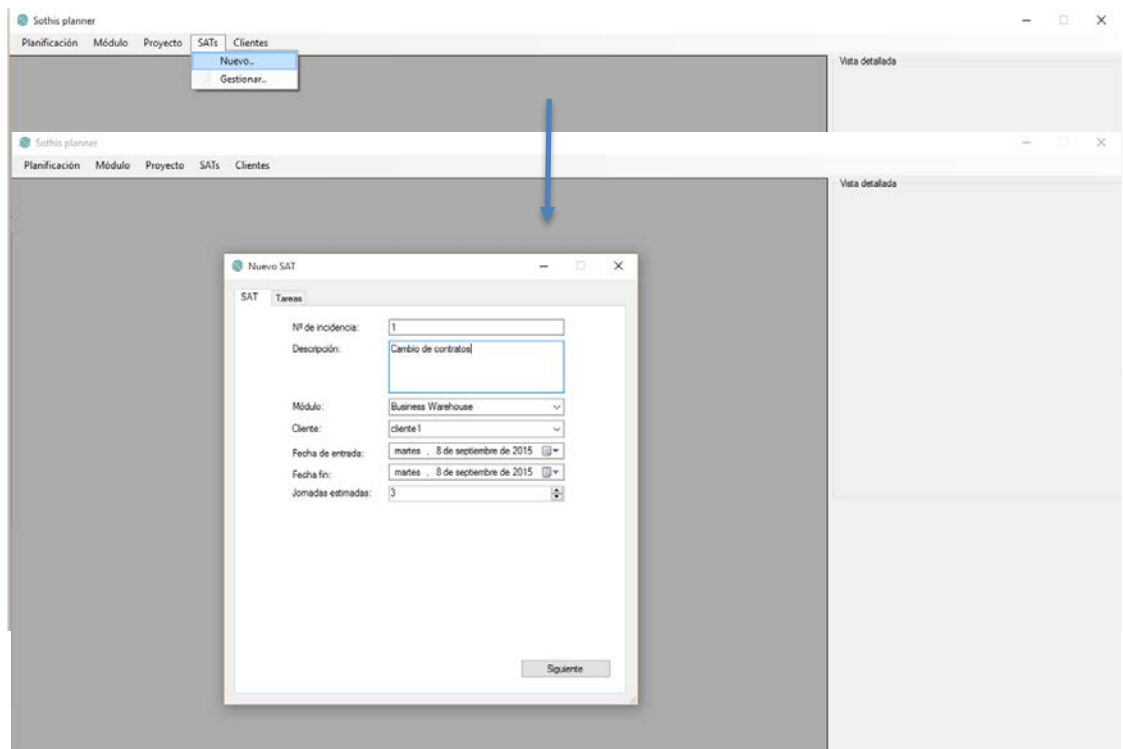


Figura 14. Creación de un nuevo SAT

Una vez hemos hecho clic, nos aparecerá una ventana como la que vemos en la **Figura 14**. Al igual que en la creación del proyecto, tendremos una primera ventana donde tendremos que poner el número de incidencia del *Solution Manager*, descripción, módulo correspondiente, el cliente, la fecha y el número de jornadas.

Como ya se ha dicho, es muy similar al proyecto, por lo que una vez que hayamos rellenado esos campos le podremos dar a *Siguiente* para rellenar las tareas.

Luego aparecerá una ventana como en la **Figura 15**, donde crearemos las correspondientes tareas añadiendo filas, y al pulsar la tarea deseada podremos añadir las cualificaciones necesarias que necesita esa tarea. Una vez acabado, pulsamos el botón *Crear SAT* para finalizar.

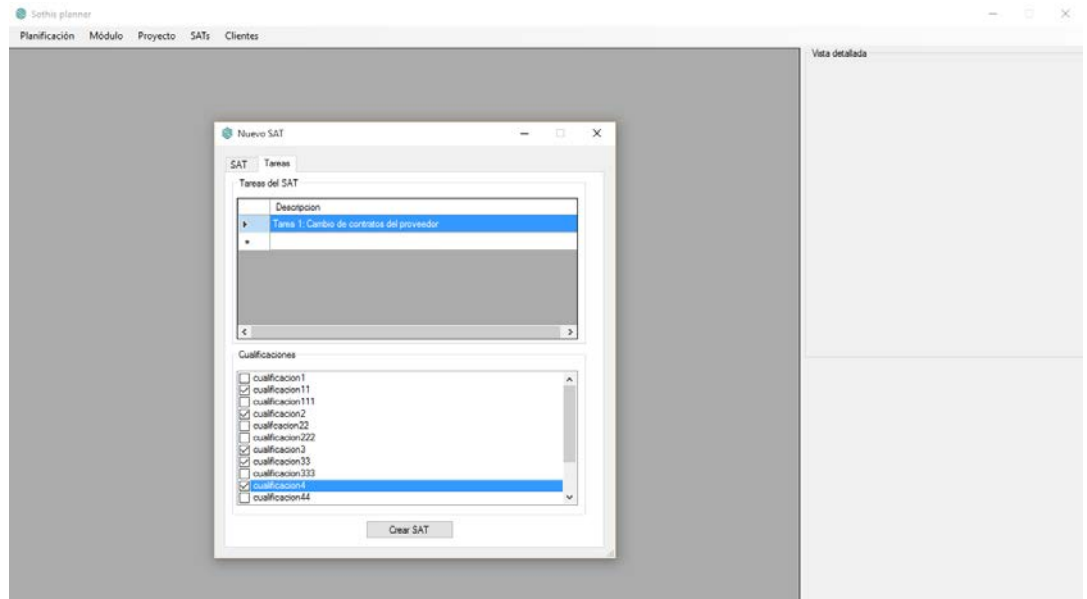


Figura 15. Creación de tareas.

5_2/ GESTIÓN DE SATs

Para la gestión de nuestros SATs pulsaremos **SATs > Gestión..**

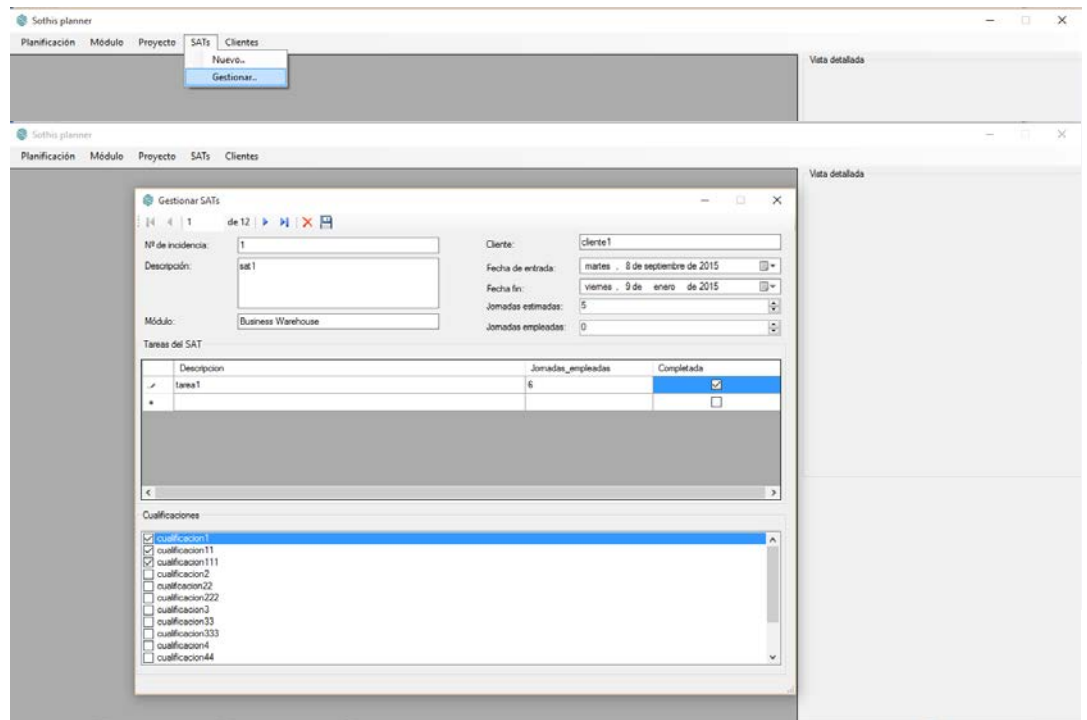


Figura 16. Gestión de SATs.



Como podemos ver en la **Figura 16**, se nos abrirá la siguiente ventana donde podremos realizar una edición de nuestros SATs. Si pulsamos en la flecha, al igual que en la gestión de nuestros proyectos, podremos ir viendo todos los SATs disponibles que hemos ido creando.

La vista será prácticamente la misma también que en los proyectos: tenemos el número de incidencia del SAT, su descripción, módulo, cliente, fechas y jornadas. Debajo de todos los datos principales tendremos una sección de tareas.

En la sección de tareas se encontrarán todas las tareas que hemos añadido manualmente en la creación de nuestro SAT, donde podremos editar su descripción, las jornadas y podremos marcar en este caso si la tarea ha sido completada.

De nuevo, de marcar una tarea como completada, será ignorada para la planificación.

Por otro lado, también podemos modificar las cualificaciones de la tarea, simplemente hacemos clic sobre la tarea deseada y se nos activará debajo en la sección de cualificaciones todas las cualificaciones, donde podremos añadir y quitar las que no queramos que tenga nuestra tarea.

Una vez que hemos acabado de realizar todos los cambios, le damos al botón de guardado.

6/ CLIENTES

En esta área vamos a hacer una gestión de nuestros clientes, pudiendo añadir e eliminar más clientes y gestionando la relación que mantiene con cada uno de nuestros consultores.

6_1/ GESTIÓN DE CLIENTES

Para la gestión de nuestros clientes accedemos a **Clientes > Gestionar...**

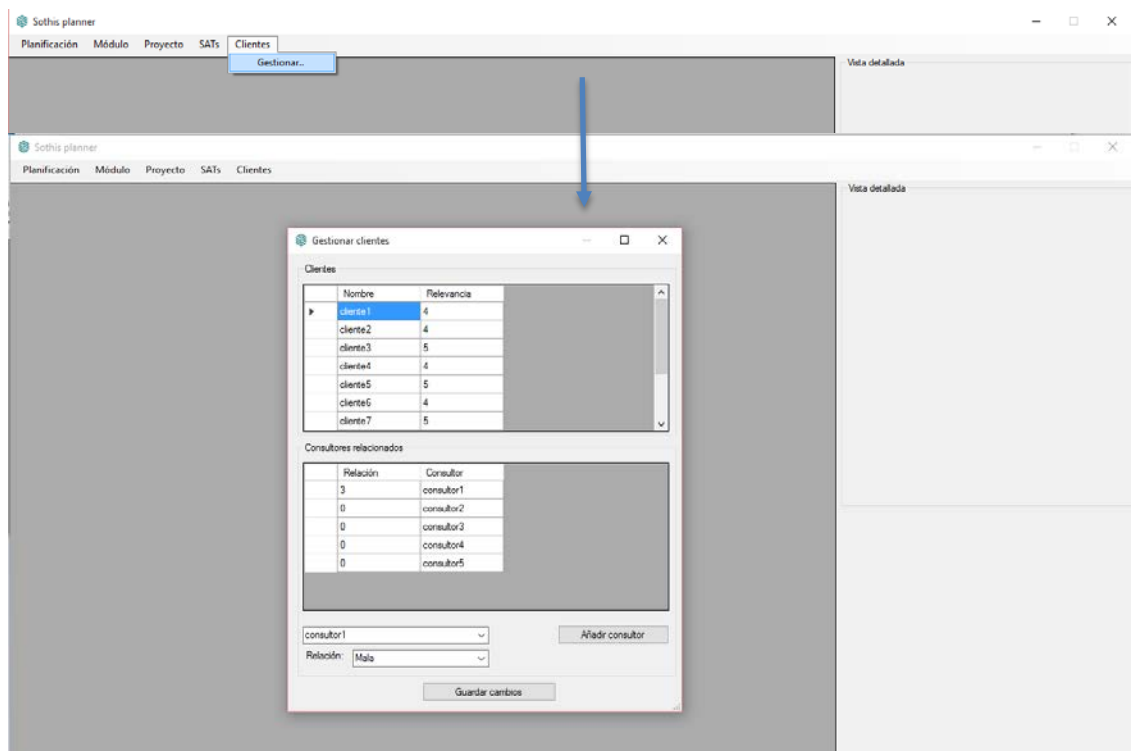


Figura 17. Gestión de clientes.

Una vez que hemos entrado, nos aparecerá una ventana cómo la de la **Figura 17**. En la parte superior se nos mostrará una ventana donde se encontrarán todos nuestros clientes. Podremos añadir más si pinchamos en una fila nueva vacía, donde podremos añadir el nombre del cliente y su relevancia para la empresa.

Cada vez que pulsemos en un cliente, la sección de debajo de consultores relacionados se modificará. Esto se debe a que cada cliente tiene unos consultores asignados, donde se puede añadir también su relación.

Si queremos añadir un consultor a un cliente en concreto, solo tenemos que pulsar el botón *Añadir consultor* estando marcado el cliente deseado y habiendo fijado previamente la relación que mantienen (0.- Mala, 1.- Sin relación, 2.- Regular, 3.- Buena).

Cuando hayamos finalizado de hacer todos los cambios pertinentes, pulsaremos el botón *Guardar cambios* para guardar todas las modificaciones.



7/ CONFIGURACIÓN

En este apartado vamos a mostrar cómo cambiar la configuración del programa, así como reorganizar los días festivos, puesto que algunos varían anualmente.

7_1/ CONFIGURACIÓN DE VARIABLES

Para modificar nuestras variables del programa, debemos editar el fichero XML de configuración del programa. Para ello, accederemos a la carpeta de instalación y una vez allí, accederemos a **XML > config.xml**.

Nota Importante: Para que las modificaciones surtan efecto, se deben hacer las modificaciones pertinentes antes de iniciar la aplicación.

El archivo tendrá la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuracion cantidad="6">
    <config name="n_calendarios" value="30" type="int"></config>
    <config name="n_iteraciones" value="100" type="int"></config>
    <config name="intercambiar_siempre" value="false" type="bool"></config>
    <config name="peso_relacion_cliente" value="0,3" type="double"></config>
    <config name="peso_cualificaciones" value="0,6" type="double"></config>
    <config name="peso_fecha_fin" value="0,1" type="double"></config>
</Configuracion>
```

Para los que no estén muy familiarizados con XML, tenemos 6 campos (como podemos ver en la cabecera, `cantidad="6"`) con 3 atributos en cada campo. Estos atributos son *name*, que se corresponde al nombre de nuestras variables de configuración, *value*, que es el valor que inicialmente le damos al atributo y *type* que es el tipo de variable (entero, carácter, decimal,...). Ahora vamos a pasar a la explicación de cada campo:

- **n_calendarios:** es el número de calendarios que crearán al hacer una nueva planificación. Cuántos más calendarios, mayor posibilidad de encontrar una solución óptima, sin embargo el tiempo de planificación aumenta también. Por defecto es 30.
- **n_iteraciones:** es el número de procesos de mejora a aplicar a cada calendario creado. Cuántas más iteraciones, mayor posibilidad de encontrar una solución óptima, sin embargo el tiempo de planificación aumenta también. Por defecto es 100.
- **intercambiar_siempre:** esta variable indica si los intercambios de tareas en el proceso de mejora deben hacer siempre, aunque sea para peor. (Ver más detalles en la memoria). Por defecto es falso.



Los siguientes tres parámetros están relacionados (valorados de 0 a 1 según nivel de importancia):

- **peso_relacion_cliente**: determina la importancia que le damos a que el consultor asignado a la tarea tenga una estrecha relación con el cliente, tanto de conocimientos del sistema como personal. Por defecto es 0,3.

- **peso_cualificaciones**: determina la importancia que le damos a que un consultor tenga el mayor número posible de las cualificaciones requeridas por la tarea. Por defecto es 0,6.

- **peso_fecha_fin**: determina la importancia que le damos a que la asignación de la tarea se haga antes de la fecha de finalización del hito/SAT. Por defecto es 0,1.

Estos tres parámetros **DEBEN** sumar en total 1, dándole más cantidad a aquel que le queramos dar más importancia (*peso_cualificaciones* en nuestro caso). Una vez realizadas las modificaciones, guardaremos el XML y habremos finalizado, pero hay que tener en cuenta que modificar nuestro archivo config.xml puede afectar tanto al rendimiento como a los resultados obtenidos por la aplicación.

7_2/ DÍAS FESTIVOS

Al igual que en la modificación de nuestro archivo **config.xml**, accederemos a la carpeta de instalación y luego a la carpeta **XML > festivos.xml**.

Nota Importante: al igual que en el apartado anterior, para que las modificaciones surtan efecto se deben realizar antes de iniciar la aplicación.

El archivo tiene una estructura similar al anterior:

```
<?xml version="1.0" encoding="utf-8" ?>

<Festivos cantidad="7">

    <Festivo name="Año nuevo" dia="01" mes="01"></Festivo>

    <Festivo name="Reyes" dia="06" mes="01"></Festivo>

    <Festivo name="San José" dia="19" mes="03"></Festivo>

    <Festivo name="Día de la Comunidad Valenciana" dia="09" mes="10"></Festivo>

    <Festivo name="Noche buena" dia="24" mes="12"></Festivo>

    <Festivo name="Navidad" dia="25" mes="12"></Festivo>

    <Festivo name="Noche vieja" dia="31" mes="12"></Festivo>

</Festivos>
```

Puesto que ya hemos realizado una explicación más detallada en el apartado anterior del funcionamiento de los XML, solo hablaremos del contenido.



El campo *name* es el nombre del día festivo, el campo *dia* el día correspondiente al festivo (si es menor de 10 se ha de escribir con el formato 0X) y el *mes* (el formato 0X se usa también en este caso).

Si queremos añadir algún día más en la empresa que queramos que sea festivo, utilizaremos el mismo formato que en los demás días:

```
<Festivo name="Nombre" dia="01" mes="01"></Festivo>
```

Esto implica que todos los días declarados en el fichero, para el año natural en curso, serán obviados por el planificador para todos los consultores.

Cuando hayamos acabado de realizar todos los cambios, guardaremos el archivo y a continuación ya podremos iniciar el programa.



GrupoSothis

CONFIDENCIALIDAD

Esta propuesta puede contener información confidencial y/o secretos industriales que pertenecen a Sothis Tecnologías de la Información, S.L. Esta información se entrega únicamente para permitir al destinatario poder valorar la oferta descrita en el presente documento.

Cuando se reciba el presente documento el destinatario se compromete a tratar esta información como confidencial y a no reproducir, ni divulgarla, exceptuando a personas directamente responsables de la propia evaluación del contenido de la misma, sin el consentimiento de Sothis Tecnologías de la Información, S.L. Sothis Tecnologías de la Información, S.L. se reserva el derecho sobre retomar las copias de esta oferta una vez terminada su evaluación.

Centro Empresarial Lumiere, Nave 20
Ronda de Auguste y Louis Lumiere, 23
Parque Tecnológico de Paterna
46980 Paterna (Valencia) Spain
Phone. +34 902 88 35 33
Fax. +34 902 90 89 49

www.gruposothis.com



GrupoSothis

www.gruposothis.com