



*Facultad
de
Ciencias*

**DISEÑO Y DESARROLLO DE
PLANIFICADOR DE RECURSOS
MICROSOFT DYNAMICS NAV**
(Design and development of Microsoft
Dynamics NAV resource planner)

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Jairo López Piñeiro

Director: Andrés Iglesias Prieto

Co-Director: Jose Manuel Fernández Casado

Junio – 2017

Resumen

Microsoft Dynamics NAV es una solución de planificación de recursos empresariales (ERP) y una herramienta adaptable capaz de ayudar a las pequeñas y medianas empresas a automatizar y conectar sus ventas, compras, operaciones, contabilidad y administración de inventarios.

En el presente proyecto se pretende desarrollar e integrar una herramienta en el entorno anteriormente descrito, creando un planificador gráfico de recursos y para facilitar la gestión tanto de tareas como de recursos.

El objetivo principal del proyecto es proveer a los usuarios de una herramienta gráfica que les facilite la tarea de gestión y administración de su negocio.

El presente proyecto se ha realizado en conjunto con el equipo de desarrollo de la gerencia Pyme de la empresa Consulting Informatico Cantabria (CIC). La herramienta desarrollada de la que es objeto el presente trabajo de fin de grado es un proyecto real llevado a cabo por CIC para el grupo Linde España e Italia como por ejemplo: Ibermicar, Talleres Lozano, Maicar, Talleres Miquel, Reybesa, Ventocar, Serma, Vidal Leon Diesel, OMC, Qualift, etc., el cual primero será implementado y desarrollado en modo beta, para posteriormente implementarlo de manera definitiva.

El lenguaje usado internamente por Microsoft Dynamics NAV es C/AL; en este caso concreto el lenguaje nativo propio de la herramienta se han utilizado lenguajes de programación WEB como son JavaScript, JQuery, HTML5 y CSS.

Cabe descartar que, pese a que el proyecto está enfocado a un cliente concreto con unas necesidades propias, se pretende que el programa tenga la versatilidad de poder adaptarse a cualquier tipo de usuario con otras características y necesidades.

A través de la aplicación tanto web como local, el usuario podrá consultar los pedidos de servicio, consultar y administrar la carga de trabajo de cada recurso, asignar pedidos a recursos, redimensionar el tiempo de trabajo dedicado por cada recurso a cada pedido de servicio entre otras funcionalidades y gestionar todo lo anterior. La gran ventaja que ofrece la herramienta es que lo que anteriormente requería varios pasos ahora se consigue en uno solo y de manera gráfica.

Palabras clave

Microsoft Dynamics NAV, Planificador Grafico, Aplicación Web, JavaScript.

Abstract

Currently, Microsoft Dynamics NAV is a solution of business resources planner and a customizable tool that can help to small and medium-size enterprises to automating and connecting their purchases, operations, contability and inventory management. In this Project we develop and integrate a tool in the environment indicated above, creating a graphic planner of resources and to ease the management of both tasks and services.

The main objective of the Project is to provide the users with a graphical tool to simplify the management of their business.

The present Project has been done jointly with the development team of the SME Management of the Company Consulting Informatico Cantabria (CIC). The tool developed is not just a proof of concept, but a real full Project developed by CIC for the Linde Group of Spain and Italy like: Ibermicar, Tallere Lozano, Maicar, Talleres Miquel, Reybesa, Ventocar, Serma, Vidal Leon Diesel, OMS, Qualift, etc.. The tool will firstly be implemented and developed in beta mode, and then implemented in final versión.

The language used internally for Microsoft Dynamics NAV is C/AL; in this case, in addition to the native language of the tool, we have also used some web programming languages such as JavaScript, JQuery, HTML5 and CSS.

Notably, although the Project is focused on a particular client with his own needs, the intention is that the program has the versatility to be able to adapt to any type or user with other characteristics and needs.

Through the application, both web and local, the user can check the services orders, check and manage the workload of each resource, to assign orders to resources, resize the work time dedicateed for each resource to each order of service among other functionalities and managing all of the above. The great advantage that offers the tool is that previously required several steps now is achieved with only one and graphically.

Keywords

Microsoft Dynamics NAV, Graphic Planner, Web Application, JavaScript

Tabla de contenidos

1.	Introducción	6
1.1.	FullCalendar.....	6
1.2.	Objetivos y motivación del proyecto	6
1.3.	Estructuración de la memoria.....	7
2.	Herramientas utilizadas.....	8
2.1.	Herramientas Server.....	8
2.1.1.	Microsoft Dynamics NAV Server.....	8
2.1.2.	Microsoft SQL Server 2016.....	8
2.2.	Microsoft Dynamics NAV	9
2.3.	Otra herramientas.....	11
2.3.1.	Team Viewer 10	11
2.3.2.	MOPA.....	11
2.3.3.	JIRA	12
2.3.4.	Microsoft Office 2016.....	14
2.3.5.	Microsoft Visual Studio Profesional 2013.....	14
2.3.6.	Google Chrome	15
2.3.7.	Forti Client	16
2.3.8.	Windows Server 2016.....	17
3.	Análisis de requisitos	18
3.1.	Análisis de requisitos comunes al Grupo Linde	18
3.2.	Análisis de requisitos específico de clientes	18
4.	FullCalendar Scheduler.....	20
4.1.	Clases y estilos	20
4.2.	Adaptación a los requisitos	21
4.3.	Cambios de apariencia	33
4.4.	Creación e integración Control Add-in	34
4.5.	Integración Mapa Leaflet	36
4.6.	Adaptación grupo Italiano	37
5.	Adaptación Microsoft Dynamics NAV	39
6.	Paso y tratamiento de datos.....	42
6.1.	Escritura de datos	42
6.2.	Lectura de datos	45
6.3.	Tratamiento de datos	46
7.	Conclusiones	48
	Bibliografía.....	49
	Anexo I	50

1. Introducción

1.1. FullCalendar

FullCalendar es la base donde se cimienta todo el proyecto, más concretamente su rama de scheduler. Esta herramienta nos permite realizar las modificaciones que el cliente nos indica cómo se podrá ver en el capítulo encargado de describir análisis de requisitos. Tras realizar un estudio de aplicaciones similares se toma la decisión de utilizar la base de esta aplicación de código abierto.

Cuenta con varios tipos de licencia:

- Licencia de uso comercial (usada en nuestro proyecto)
- Licencia de uso no comercial o creativa
- Licencia de GPL

1.2. Objetivos y motivación del proyecto

La motivación del proyecto es crear una herramienta para sustituir los numerosos pasos que se tenían que realizar para asignar y modificar cualquier asignación de pedido a recurso, ya que gestiona tareas que cualquier empresa tiene que realizar cada día, por lo que, facilitar la realización de estas es algo de mucha importancia y demandado. Es por esto que se ha optado por una representación gráfica de la carga de trabajo de cada recurso ya que parece la manera más intuitiva de realizar esta tarea, lo que hará que se mejore la producción de los clientes que cuenten con dicha aplicación.

El objetivo principal de este proyecto es crear una herramienta gráfica desde la cual se puedan gestionar tanto los recursos como los pedidos de una manera sencilla y rápida a la par que intuitiva para el cliente, ya que, durante el proceso de creación, al ser una herramienta real como ya se ha indicado, se le hace participe en todo momento para diseñarla y configurarla acorde a sus necesidades.

En particular este proyecto pretender abordar los siguientes objetivos:

- Análisis de requisitos de los clientes pertenecientes al Grupo Linde.
- Adaptación de la herramienta FullCalendar para cumplir los requisitos anteriormente descritos.
- Modificación de la herramienta Microsoft Dynamics NAV para la integración de la herramienta.
- Comunicación de datos en ambos sentidos por medio de XML y funciones.
- Implementación e integración en cliente final.

1.3. Estructuración de la memoria

La memoria está organizada de la siguiente manera:

- En el capítulo 2 se describen todas las herramientas que se han usado para poder llevar a cabo este proyecto.
- En el capítulo 3 se establece el análisis de requisitos tanto común a todo el grupo como específico por cliente.
- En el capítulo 4 se describen los cambios realizados sobre FullCalendar para adaptarlo al análisis de requisitos.
- En el capítulo 5 se describen las modificaciones y adaptaciones sobre la herramienta Microsoft Dynamics NAV para el funcionamiento de la nueva aplicación diseñada en el presente proyecto.
- En el capítulo 6 se describe como se establece la comunicación y el tratamiento de datos entre la aplicación y el programa Microsoft Dinmycs NAV.
- Por último, en el capítulo 7 se repasarán los objetivos y los resultados obtenidos.

2. Herramientas utilizadas

2.1 . Herramientas Server

2.1.1. Microsoft Dynamics NAV Server

Herramienta localizada en el servidor la cual está conectada a la base de datos de SQL, esta aplicación funciona exclusivamente con SQL Server, usa el sistema de comunicaciones de Windows FrameWork y el protocolo de Clientes de Roles y Web Services. Permite la respuesta en paralelo de varias peticiones de clientes. La lógica empresarial se ejecuta íntegramente en el servidor.

Microsoft Dynamics NAV Server proporciona una capa adicional de seguridad entre los clientes y la base de datos. Aprovecha las características de autenticación de Windows Communications Framework para proporcionar otra capa de autenticación de usuario y utiliza suplantación para garantizar que la lógica empresarial se ejecuta en un proceso que ha sido instanciado por el usuario que envió la solicitud. Esto garantiza que todas las funciones de autenticación de Windows y de Microsoft Dynamics NAV y los permisos que se hayan concedido al usuario sean correctos. También asegura la auditoría a nivel lógico de negocio. En la *Figura 1* se muestra un esquema de la estructura que sigue Microsoft Dynamics NAV Server.

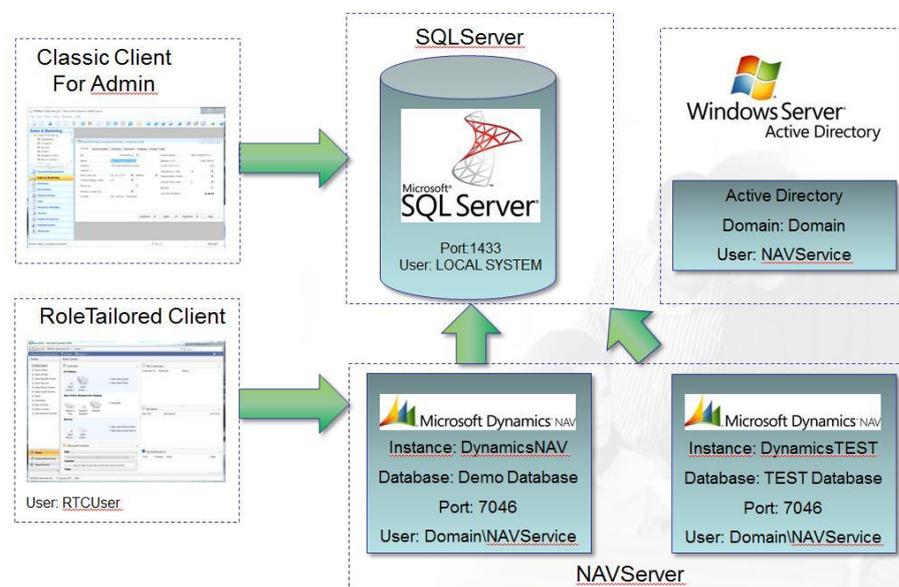


Figura 1. Esquema Microsoft Dynamics NAV Server

2.1.2. Microsoft SQL Server 2016

Programa encargado de gestionar las bases de datos encargadas de gestionar los usuarios y sus permisos, y los datos de la empresa contenidos dentro de las bases de datos.



Figura 2. Logo SQL Server 2016

2.2. Microsoft Dynamics NAV

Microsoft Dynamics NAV es una solución de planificación de recursos empresariales (ERP), es decir, una herramienta adaptable capaz de ayudar a las pequeñas y medianas empresas a automatizar y conectar sus ventas, compras, operaciones, contabilidad y administración de inventarios.



Figura 3. Logo Microsoft Dynamics NAV

Entre sus principales características se encuentran:

- **Contabilidad y administración financiera.**
 - Administración de efectivo, activos y cuenta bancaria.
- **Cadena de suministro, fabricación y operaciones.**
 - Realizar el seguimiento y administración de la producción, el inventario, los pedidos y los proveedores.
- **Ventas y servicios.**
 - Administración de los contactos, las oportunidades de ventas y los contratos de servicio.
- **Administración de proyectos.**
 - Creación de estimaciones, realiza seguimiento de proyectos y administra la capacidad de cada proyecto.
- **Informes e inteligencia empresarial.**
 - Permite una visión integral de su empresa para tomar decisiones informadas.
- **Soporte para monedas internacionales.**
 - Compete a nivel mundial con distintas monedas y en diferentes idiomas.
- **Implementación flexible.**
 - Realiza implementaciones locales o en la nube, según el modelo más adecuado para su empresa.

La herramienta cuenta con dos aplicaciones:

- **Desarrollo (Figura 4):** Permite acceder a los componentes de la herramienta para poder personalizarlo a las necesidades de cada usuario. Se utiliza el lenguaje de programación propietario de Microsoft Dynamics NAV, denotado como C/AL.

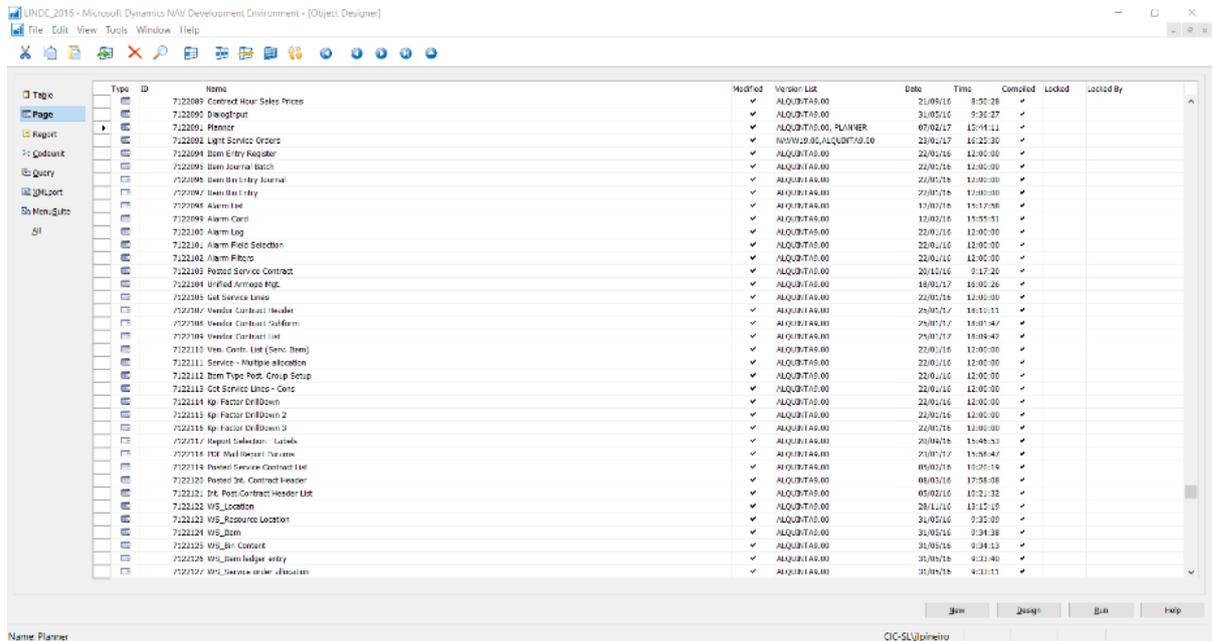


Figura 4. Entorno de desarrollo de Microsoft Dynamics NAV

- **Cliente (Figura 5):** Es la aplicación que se ejecuta en el ordenador cliente y que contacta con la herramienta en el servidor o nube, la cual es la encargada de tener todos los objetos.

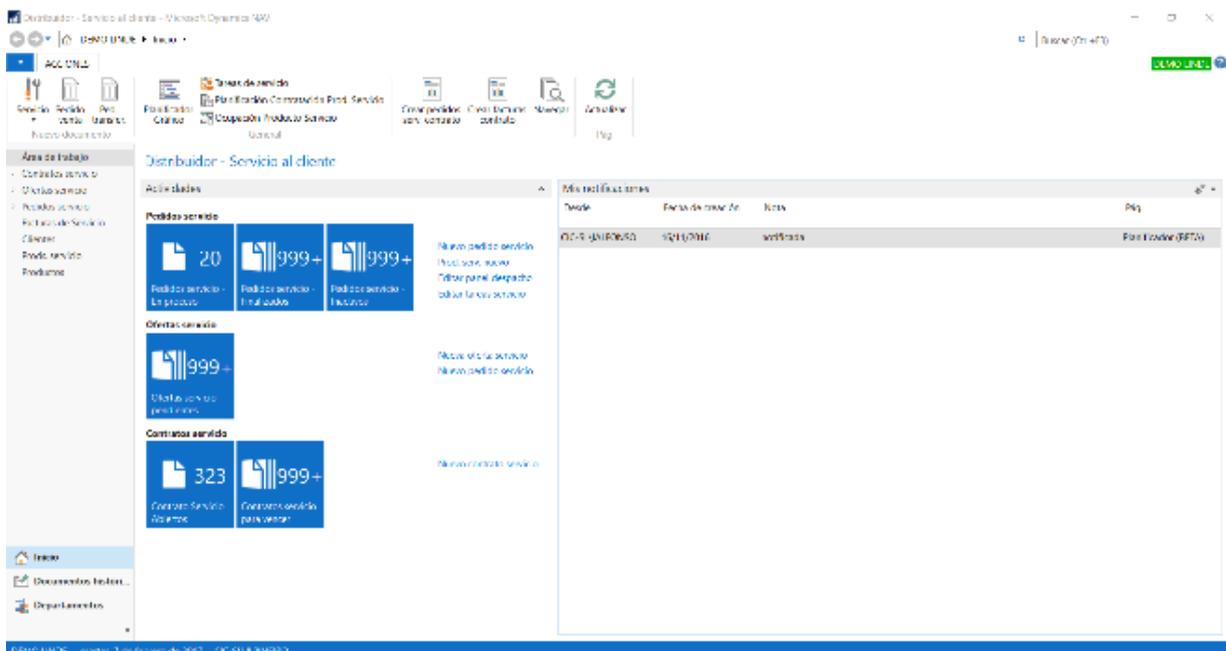


Figura 5. Entorno Cliente Microsoft Dynamics NAV

2.3. Otras de herramientas

2.3.1. Team Viewer 10

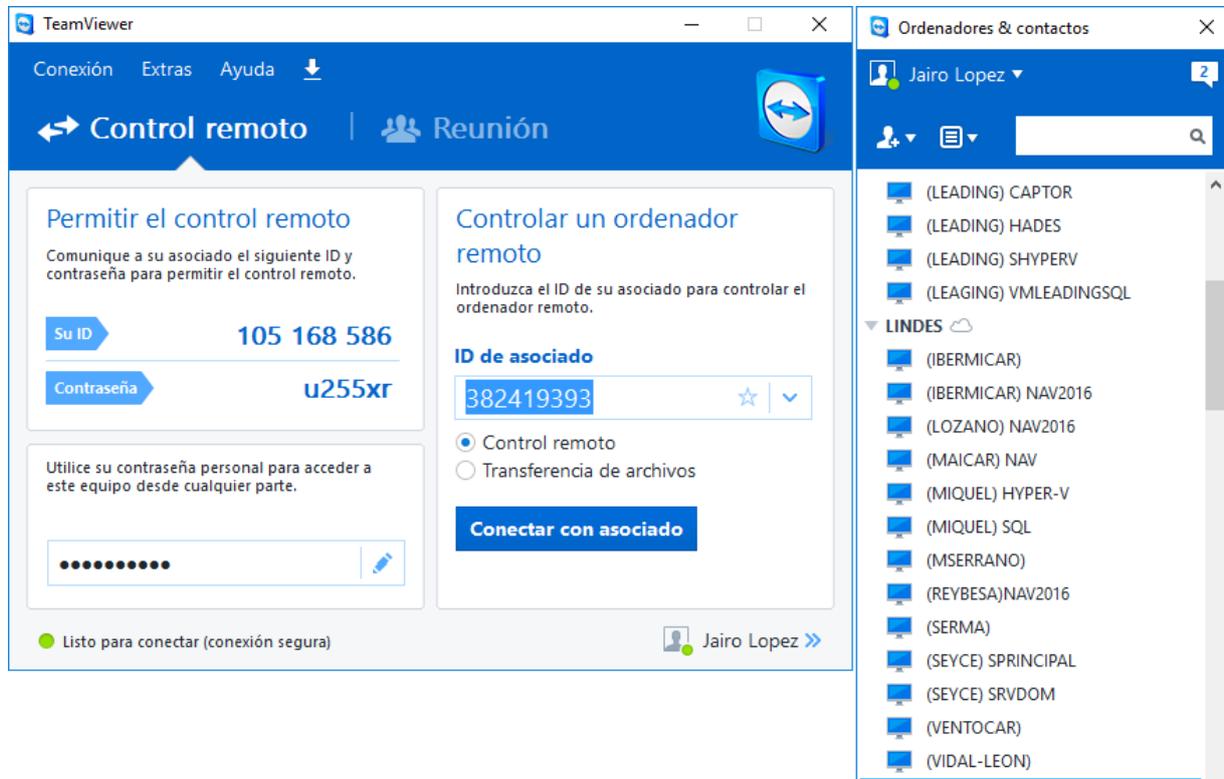


Figura 6. Entorno de trabajo de Team Viewer 10

Software informático privado que permite conectarse remotamente a otro equipo siempre y cuando tenga la aplicación instalada. Esta aplicación puede ser usada para realizar planes de mantenimiento en los servidores, realizar implantaciones o modificaciones de algunos servicios. En la *Figura 6* se muestra un ejemplo del entorno de trabajo de Team Viewer.

Entre sus funciones podemos encontrar:

- Compartir y controlar escritorios.
- Reuniones en línea.
- Videoconferencias.
- Transferencia de archivos entre ordenadores.

2.3.2. MOPA

El programa está perfectamente integrado con el Active Directory de la empresa lo que hace que nos podamos loguear con nuestro usuario de dominio como se muestra en la *Figura 7*. Permite registrar y administrar las horas que dedica un recurso a un proyecto en el cual está trabajando.

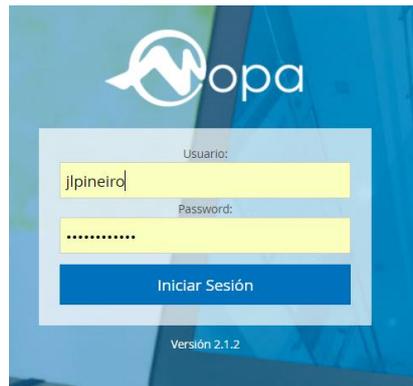


Figura 7. Ventana de acceso a MOPA

En la *Figura 8*, se muestra En rojo, las horas que son imputadas directamente desde NAV 2013 usado en la empresa para la gestión de los pedidos de servicio y contratos. En amarillo, las horas imputadas pero aún pendientes de asignar. En verde, las horas imputadas y registradas dentro del parte de trabajo de cada empleado.

Fecha	TTra	TCon	Proyecto	Fase	Gerencia	Horas	Descripción
01/02/2017 3,75 h.	PRD	NOR	PDYD15/053	1000	100	2	LINDEEVOLUT-152
	PRD	NOR	PDYD15/053	1000	100	2,5	LINDEEVOLUT-181
	PRD	NOR			100	2,75	PS17/000950-Vista Periodica Febrero 2017
	FOI	NOR			100	1	
02/02/2017 4,5 h.	PRD	NOR	PDYD15/053	1000	100	1,5	LINDEEVOLUT-177
	PRD	NOR	PDYD15/053	1000	100	1,75	LINDEEVOLUT-180
	PRD	NOR	PDYD15/053	1000	100	0,5	LINDEEVOLUT-178
	PRD	NOR			100	1	PS17/000838- Informe Balance y suma de saldos
	PRD	NOR			100	1,75	PS17/000950-Vista Periodica Febrero 2017
	PRD	NOR			100	1,75	PS17/000950-Vista Periodica Febrero 2017

Figura 8. Entorno de trabajo MOPA

Cabe destacar que esta herramienta ha sido desarrollada por el personal de CIC lo que hace que se pueda integrar perfectamente con el resto de herramientas usadas por la empresa como son el JIRA y NAV13. Se muestra un ejemplo en la *Figura 8*.

2.3.3. JIRA

JIRA es una aplicación basada en web para la gestión operativa de proyectos. Tiene las siguientes características principales:

- **Planifica**

- Crea historias de usuario e incidencias, planifica sprints y distribuye tareas entre tu equipo de software.
- **Supervisa**
 - Prioriza y analiza el trabajo de tu equipo en su contexto y con total visibilidad.
- **Seguridad**
 - La información del estado en el que se encuentra el proyecto siempre actualizada.
- **Crea Informes**
 - Mejora el rendimiento del equipo con datos visuales en tiempo real de gran utilidad.

Al igual que las anteriores herramientas, el acceso a la aplicación se realiza mediante el usuario y contraseña de dominio como se puede ver en la *Figura 9*.



Figura 9. Ventana de acceso a JIRA

Cada usuario del equipo de trabajo tiene su propio panel de trabajo en el cual tiene proyectos asignados por el project manager. Cada proyecto cuenta con tareas y subtareas que son asignadas a desarrolladores.

Cada tarea tiene su propia descripción de requisitos y el número de horas que tiene planificada para su elaboración. En las *Figuras 10 y 11* se muestra un ejemplo del entorno de trabajo.

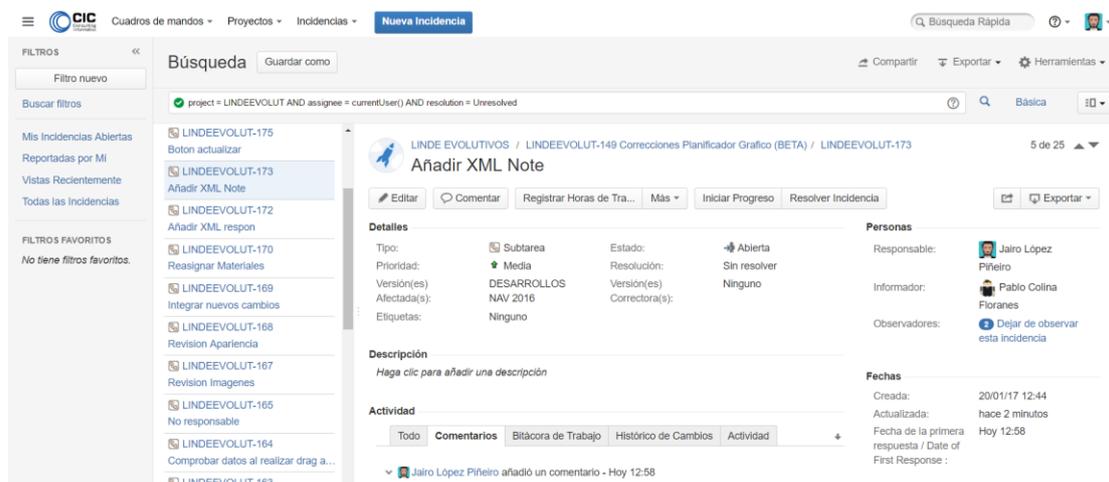


Figura 10. Entorno de trabajo de JIRA



Figura 11. Panel de tareas (izquierda) y seguimiento de tiempo (derecha)

Esta herramienta se encuentra perfectamente integrada con la herramienta MOPA descrita anteriormente para la gestión de los recursos.

2.3.4. Microsoft Office 2016

Microsoft Office 2016 es una suite de aplicaciones ofimáticas de productividad que incluye Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Microsoft Outlook y Skype Empresarial.



Figura 12. Logo Microsoft Office 2016

2.3.5. Microsoft Visual Studio Profesional 2013

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, etc., al igual que entornos de desarrollo

web, como ASP.NET MVC, Django, etc. En la *Figura 13* se muestra un ejemplo del entorno de trabajo de Microsoft Visual Studio

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros.

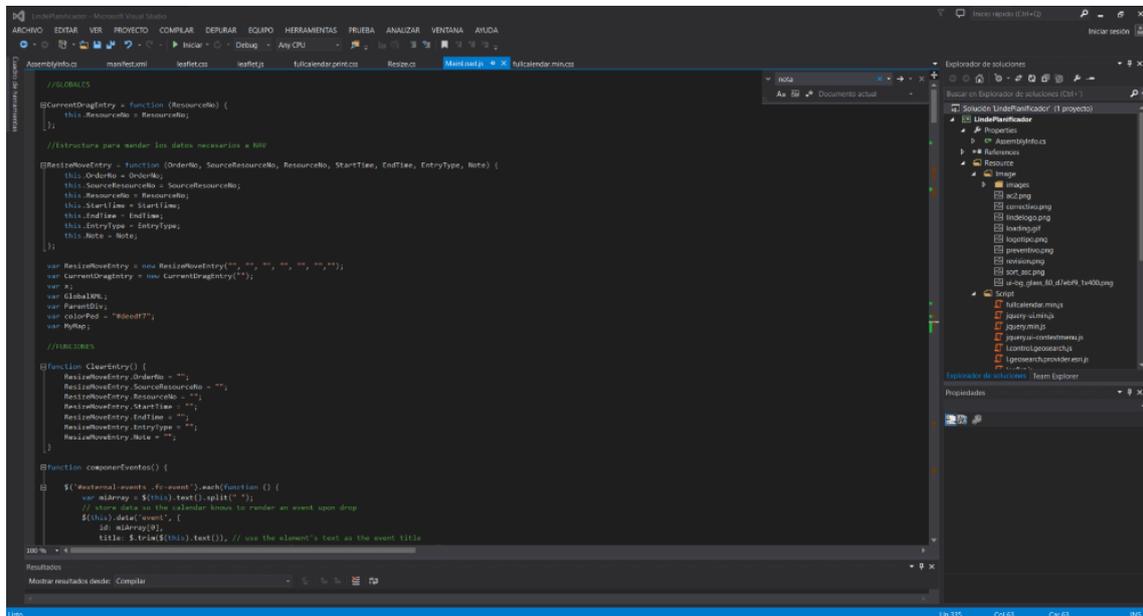


Figura 13. Entorno de trabajo de Microsoft Visual Studio Professional 2013

En este proyecto, esta herramienta es utilizada para realizar el código web para el proyecto como por ejemplo JavaScript, HTML5, CSS, etc. También permite administrar el directorio de recursos del proyecto donde puedes añadir las clases, imágenes y estilos.

2.3.6. Google Chrome

Ejecutando esta herramienta para desarrolladores que nos facilita Google Chrome podemos revisar el código fuente de la página actual, donde puedes comprobar la construcción del sitio web e incluso modificar el contenido en tu navegador. Permite realizar modificaciones al vuelo para ver cómo se presentan estilos distintos o comprobar los errores arrojados en la consola que nos facilita. En la *Figura 14* se muestra un ejemplo del entorno.

2.3.8. Windows Server 2016

Windows Server es una marca que abarca una línea de productos de servidor de Microsoft Corporation y consiste en un sistema operativo diseñado para servidores de Microsoft y una gama de tipos de productos dirigidos al mercado más amplio de negocios; Windows Server incluye, por ejemplo, interfaz de usuario, el administrador de tareas, el IP address management, active directory, escritorio remoto, entre otros.

Sobre dicho sistema operativo se monta Microsoft Dynamics NAV descrito anteriormente, que integra sus servicios y usa el conjunto de servicios de IIS para el acceso WEB para la aplicación.



Figura 16. Icono de Windows Server 2016

3. Análisis de requisitos

3.1. Análisis de requisitos comunes al Grupo Linde

Tras varias reuniones tanto presenciales como telefónicas o por videoconferencia, se establecen el siguiente análisis de requisitos genéricos para todo el grupo de clientes Linde España e Italia:

- Asignación por medio de drag and drop
- Eliminar asignación por medio de botón derecho.
- Nota común.
- Nota de pedido de servicio por medio del botón derecho.
- Función de añadir en el mapa dirección de pedidos asignados por función en botón derecho.
- Filtro de fechas para los pedidos sin asignar.
- Filtro para pedidos sin asignar.
- ToolTip para mostrar información del pedido de servicio.
- Redimensión del pedido.
- Eliminación de recurso del planificador.
- Configuración de colores por estados del pedido.
- Página de configuración del planificador.
- Mostrar pedido de servicio.
- Ordenación por prioridad de los pedidos pendientes de asignar.
- Reasignación de materiales.
- Franja horaria de 5:00 a 23:00

3.2. Análisis de requisitos específico de clientes

Tras la implementación de la versión beta se realizan reuniones y contactos periódicos con varios clientes para realizar un segundo análisis de requisitos en este caso, específicos, para cada uno de ellos. Por simplificar la memoria del proyecto, se muestran a continuación en el siguiente listado:

- Corrección de caracteres especiales.
- Personalización de los colores por estado de los pedidos.
- Colores por cliente.
- Diferenciado por borde en azul de segunda asignación.
- Color para identificar gráficamente si se está cumpliendo la estimación del tiempo inicial.
- Adaptaciones visuales.

- Orden de apoyo para pedidos que no se han podido completar por la primera asignación.
- Horas Disponibles de cada recurso.
- Horas Asignadas de cada recurso.
- Personalización de datos mostrados en ToolTip
- Borde rojo en caso de que el pedido contenga una nota.

4. FullCalendar Scheduler

FullCalendar nos proporciona una aplicación básica que consta de un calendario en el cual se puede arrastrar por medio de eventos de JavaScript elementos externos a la parte interior del calendario, redimensionar los eventos asignados, y moverlos entre los recursos. A continuación, en la *Figura 17* se muestra un ejemplo de la base utilizada para el proyecto.

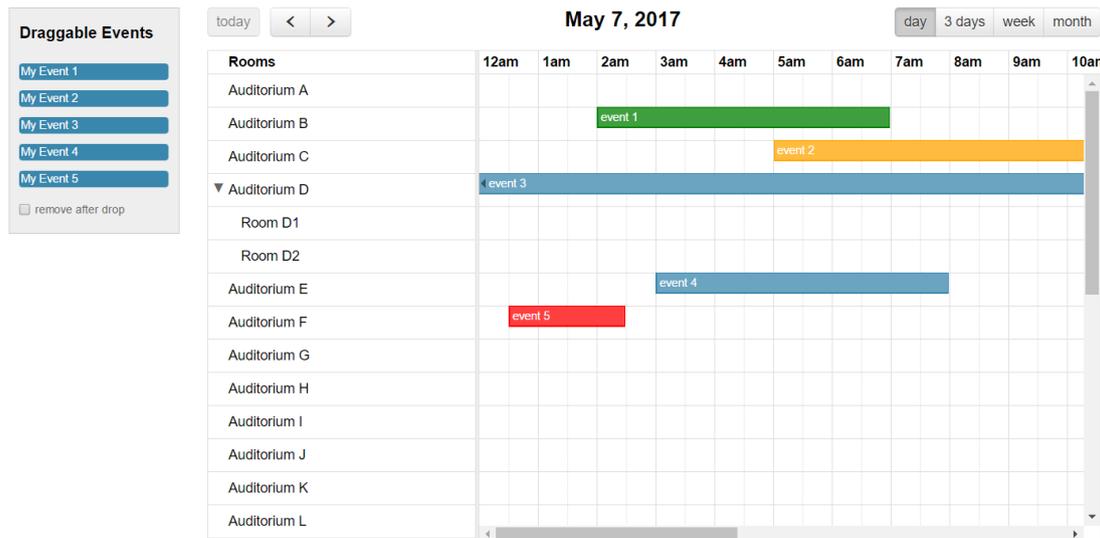


Figura 17. Estado inicial FullCalendar

4.1. Clases y estilos

Después de evaluar todas las clases y estilos que integra FullCalendar se ha tomado la decisión de quedarse con las clases que se puede ver en la *Figura 18*.



Figura 18. Clases de JavaScript y CSS elegidas de FullCalendar

El resto de clases y estilos no son necesarios para este proyecto.

A continuación, se procede a describir las clases y estilos más significativos:

- FullCalendar.min.js: Clase encargada de contener toda la lógica interna de las funciones implementadas.
- JQuery-ui.min.js: Clase principal para el uso de cualquier instrucción JQuery.

- JQuery.min.js: Al igual que la anterior, permite realizar el uso de cualquier instrucción JQuery.
- Scheduler.min.js: Clase encargada de contener toda la lógica interna de la parte scheduler del FullCalendar.
- FullCalendar.min.css: Contiene todos los estilos de FullCalendar.
- JQuery-ui.min.css: Permite dotar de estilos gráficos a las funciones JQuery.
- Scheduler.min.css: Añade estilos necesarios a FullCalendar.min.css para la parte de scheduler

En el presente proyecto se ha tomado la decisión de usar todos los “.min” ya que el tamaño del fichero se ve claramente reducido por la ausencia de comentarios y espacios innecesarios.

4.2. Adaptación a los requisitos

La clase principal del proyecto reside en el documento MainLoad.js. Este documento es el encargado de gestionar la mayor parte de la toma de decisiones sobre el calendario y los datos del mismo, este se ha modificado añadiéndole varias funciones, tratamiento de eventos, estructura de datos, etc., que pasaremos a describir en más profundidad en el presente apartado.

La **estructura de datos** que se ha elegido es la siguiente:

```

ResizeMoveEntry = function (OrderNo, SourceResourceNo, ResourceNo, StartTime,
EndTime, EntryType, Note) {
  this.OrderNo = OrderNo;
  this.SourceResourceNo = SourceResourceNo;
  this.ResourceNo = ResourceNo;
  this.StartTime = StartTime;
  this.EndTime = EndTime;
  this.EntryType = EntryType;
  this.Note = Note;
};

```

En ella alojamos todos los datos necesarios para cualquiera de los casos que podemos tener; en algunos casos puede ser que algún campo se rellene con blanco, debido a que no se usa en ese momento.

En el proyecto se ha decidido aprovechar la modularidad que nos presenta JavaScript para realizar todo en funciones. Procedemos a describir el orden de llamada a cada una de ellas y explicarlas a continuación.

Función principal que llama al resto:

```

function LoadResources(data)
{
  var xmlNavData = decodeURIComponent(data);
  var xmlDoc = $.parseXML(xmlNavData);

```

```
GlobalXML = xmlDoc;

var LoadTpl = "";
LoadTpl = GetTpl(xmlDoc, "0");
CreateParentDiv();
CreateTextDiv();
CreateDivFilter();
CreateDivOrders(LoadTpl);
CreateDivMap();
FinishParentDiv();
MakeMap();
ComponerCalendario(data);
}
```

La función GetTpl es la encargada de componer un Div por cada uno de los pedidos de servicio recibidos por medio del XML. Esta función también es reutilizada para recomponer todos los div en función del parámetro pasado por el filtro de pedidos.

Para ello se realizan varias evaluaciones, la primera de ellas comprueba si es la composición inicial de todos los pedidos o si por el contrario, es la composición por filtrado.

Lo siguiente que realiza, mediante el método de la burbuja, es ordenar por orden de prioridad todos los pedidos de servicio para cumplir uno de los requisitos del cliente. Posteriormente se evalúa si el cliente tiene configurado los colores por estado; en caso de tenerlo, se establece el color; si, por el contrario, no cuenta con la configuración, se establece un color predeterminado establecido en una variable global.

Variable global:

```
var colorPed = "#deedf7";
```

Por último, se compone el div con la información que el cliente nos ha indicado.t

Función GetTpl:

```
function GetTpl(xmlDoc, aux)
{
  var tpl = "";

  $xml = $(xmlDoc);
  $xml.sort;
  if (aux == "1" && x != "")//Buscar
  {
    var n = "Pend[ResDesc*=" + x + "]";

    var pedidos = $xml.find(n);

    pedidos.sort(function (a, b) {
      return (parseInt($(b).attr('Priority')) - parseInt($(a).attr('Priority')));
    });

    pedidos.each(function () {
      //Se pasara el color que se necesita para ese pedido de servicio.
    });
  }
}
```

```

    if ($(this).attr("Color") != undefined)
    { //Si no tiene color se pone el color predefinido de la variable global
      colorPed = $(this).attr("Color");
    }
    tpl += '<div class="fc-event" style="background-color:' + colorPed + '" id="' +
$(this).attr("ResID") + '>' + $(this).attr("ResDesc") + '</div>';
  });
} else {

  var pedidos = $xml.find("Pend");

  pedidos.sort(function (a, b) {
    return (parseInt($(b).attr('Priority')) - parseInt($(a).attr('Priority'))); //metodo de
la burbuja
  });

  pedidos.each(function () { //Ordenar por prioridad + numero + prioridad
    //Donde se muestran los pedidos disponibles.
    //Se pasara el color que se necesita para ese pedido de servicio.
    if ($(this).attr("Color")!=undefined)
    {
      colorPed = $(this).attr("Color");
    }
    tpl += '<div class="fc-event" style="background-color:' + colorPed + '" id="' +
$(this).attr("ResID") + '>' + $(this).attr("ResDesc") + '</div>';
  });
}
return (tpl)
}

```

La siguiente **función CreateParentDiv()** crea el Div principal:

```

function CreateParentDiv()
{
  //Div Padre del que cuelgan el resto.
  ParentDiv = document.createElement("div");
  ParentDiv.id = "ParentDiv";
  ParentDiv.style.position = "absolute";
  ParentDiv.style.right = 0;
  ParentDiv.style.width = "29%";
  ParentDiv.style.height = "99%";
}

```

La **función CreateTextDiv()** crea el div para añadir la nota y se deja oculto, este se activará cuando el usuario pulse sobre el botón derecho añadir nota.

```

function CreateTextDiv()
{
  //Div para añadir la nota
  textdiv = document.createElement("div");
  textdiv.id = "textdiv";
  textdiv.className = "effect1";
  textdiv.style.position = "absolute";
  textdiv.style.display = 'none';
}

```

```

textdiv.innerHTML = '<table
width="100%"><tbody><tr><h2>NOTA</h2></tr><tr><textarea rows="10" cols="60"
id="userInput" placeholder="Escriba su nota aquí..."></textarea></tr><tr><td
width="50%" align="center"><input type="button" value="Aceptar"
onclick="CreateNote()"></td><td width="50%" align="center"><input type="button"
value="Cancelar" onclick="HiddeNote()"></td></tr></tbody></table>';
navControl.append(textdiv.outerHTML);
}

```

CreateDivFilter() crea el div superior de la columna de la derecha, el cual incluye un botón de actualizar, y una caja de texto para establecer el filtro. Aquí se maneja el evento onkeyup, que llama a la función ReadFilter() que se encarga de pasar el contenido de la caja de texto a la función anteriormente descrita para recomponer los div de los pedidos en función al contenido pasado. Este evento realiza una llamada cada vez que se pulsa una tecla dentro de la caja de texto, lo que hace que el filtrado sea dinámico según se recibe texto.

```

function CreateDivFilter()
{
// Filtro
var DivFilterContent = document.createElement("div");
DivFilterContent.id = "fil";
DivFilterContent.style.height = "8%";
DivFilterContent.innerHTML = '<table width="100%"><tbody><tr><td width="10%"
align="center"><div id="rect0" onclick="actualizar()" style="height: 50px;width:
50px;background-image: url(' +
Microsoft.Dynamics.NAV.GetImageResource('ac2.png') + ');background-size:
cover;"></div></td><td width="40%" align="center"><p style="font-family: verdana;
font-size:12pt; font-style:italic">Filtro de Pedidos: </p></td><td width="50%"><input
type="text" id="filtro" onkeyup="ReadFilter()" size="20"></td></tr></tbody></table>';
ParentDiv.appendChild(DivFilterContent);
}

```

CreateDivOrders(tpl1) se encarga de generar un div para almacenar todos los pedidos de servicio y varios div anidados en él, uno por cada pedido de servicio que se tenga.

```

function CreateDivOrders(tpl)
{
// Pedidos
var DivOrders = document.createElement("div");
DivOrders.id = "external-events";
DivOrders.style.right = 0;
DivOrders.style.height = "50%";
DivOrders.innerHTML = tpl;
ParentDiv.appendChild(DivOrders);
}

```

CreateDivMap() compone el div que contiene el mapa generado, del que se detallará la creación posteriormente en el capítulo 4.5.

```
function CreateDivMap()
{
  //Mapa
  var DivMap = document.createElement("div");
  DivMap.style.textAlign = "center";
  DivMap.style.height = "41%";
  DivMap.id = "map";
  ParentDiv.appendChild(DivMap);
}
```

FinishParentDiv() se encarga de representar todo lo anteriormente descrito una vez que esté todo cargado y listo para ser mostrado.

```
function FinishParentDiv()
{
  navControl.append(ParentDiv.outerHTML);
}
```

MakeMap() está descrito en el capítulo 4.5, por lo cual no se entrará más en detalle en este apartado.

ComponerCalendario(data) es la parte que se encarga de realizar toda la carga del calendario. Por la extensión del, se pasará a describir de manera modular para su mejor comprensión.

Lo primero que se realiza en esta función es componer dos JSON con todos los recursos y pedidos de servicio asignados, que se reciben del XML por parte de la herramienta Microsoft Dynamics NAV. Para ello se lee del XML todo aquello que contenga Resource o ResAllocation y se añade a los JSON creados anteriormente. En caso de tener nota se pone el borde de color rojo para cumplir un requisito indicado por el cliente.

```
// Componer JSON recursos
var JsonResources = [];
var JsonAllocs = [];
$xml.find("Resource").each(function () {
  JsonResources.push({
    id: $(this).attr("No"),
    title: $(this).attr("Name"), horas: $(this).attr("Hours"), horasA:
$(this).attr("HoursA"),
    building: $(this).attr("Family")
  });
});

$xml.find("ResAllocation").each(function () {
  if ($(this).attr("Note") != "") //Si tiene nota
  {
    JsonAllocs.push({
      id: $(this).attr("ResAllocDoc"), resourceId: $(this).attr("ResResNo"),
      start: $(this).attr("ResAllocStart"), end: $(this).attr("ResAllocEnd"),

```

```

        title: $(this).attr("ResAllocDoc"), Desc: $(this).attr("Description"),
        ShipCode: $(this).attr("ShipToCode"), Alias: $(this).attr("Alias"),
        color: $(this).attr("Color"), status: $(this).attr("Status"),
        note: $(this).attr("Note"), borderColor: "red"
    });
}
else //Si no tiene nota
{
    JsonAllocs.push({
        id: $(this).attr("ResAllocDoc"), resourceId: $(this).attr("ResResNo"),
        start: $(this).attr("ResAllocStart"), end: $(this).attr("ResAllocEnd"),
        title: $(this).attr("ResAllocDoc"), Desc: $(this).attr("Description"),
        ShipCode: $(this).attr("ShipToCode"), Alias: $(this).attr("Alias"),
        color: $(this).attr("Color"), status: $(this).attr("Status"),
        MapAddress: $(this).attr("MapAddress"), note: " - "
    });
}
});

```

Lo siguiente que se establece son las opciones de FullCalendar. Como se ve para el desarrollo hemos decidido usar la licencia de código abierto, que será modificada una vez se realice la implantación definitiva. Se establecen las opciones de lenguaje que serán descritas en el capítulo 4.6 como que se puede modificar para clientes de distintos países. También se establece la zona horaria y la franja horaria solicitada por el cliente en los requisitos, así como que cada slot sea de treinta minutos y que el evento tenga por defecto una duración estimada inicial de una hora. Se establece que al ejecutar el calendario la hora que se vea por defecto sea las 8:00 y que no se puedan superponer varios eventos en un único recurso. Otra parte fundamental del código es la parte en la que indicamos que cuando un evento hace drop desde fuera del calendario tiene que ser eliminado de la lista exterior.

Función FullCalendar():

```

$('#calendar').FullCalendar({
    schedulerLicenseKey: 'GPL-My-Project-Is-Open-Source', //Licencia GPL
    lang: 'es',
    locale: 'es',
    timezone: 'Europe/Madrid',
    theme: true,
    now: ActualDate,
    eventLimitText: 'más',
    allDayText: 'Todo el día',
    minTime: "05:00:00", //Establecer el tiempo que queremos que se muestre en
pantalla
    maxTime: "23:00:00",
    editable: true, // enable draggable events
    droppable: true,
    drop: function () { $(this).remove();},
    aspectRatio: 1.8, //Controla el tamaño del calendario.
    slotDuration: '00:30:00',
    defaultTimedEventDuration: "01:00:00",
    forceEventDuration: true,
    scrollTime: '08:00:00', // Por defecto empieza en esta hora

```

```
eventOverlap: false, // Permite poner mas de un evento al mismo recurso en la
misma hora
eventLimit: true,
timeFormat: 'H(:mm)',
```

Modificación del header del calendario, se eliminan los botones de semana y mes a petición del cliente, ya que no son útiles para la versión en la que trabajamos. A petición del cliente se añaden dos columnas adicionales que constan de las horas asignadas de cada técnico y las horas disponibles. También se establece el agrupamiento por familias estableciendo resourceGroupField al valor de building que tiene el valor extraído del XML de cada recurso.

Modificación del header del calendario:

```
header: {
  left: 'today prev,next',
  center: 'title',
  right: 'timelineDay,timelineThreeDays' //,agendaWeek,month Para eliminar
de los botones el de mes y semana ya que en este cliente no es necesario
},
defaultView: 'timelineDay',
views: {
  timelineDay: {
    buttonText: 'Día',
    slotDuration: '00:30'
  },
  timelineThreeDays: {
    type: 'timeline',
    buttonText: '3 Días',
    duration: { days: 3 },
    titleFormat: '[Tres Dias]'
  },
  //agendaWeek: {
    //buttonText: 'Semana' Se elimina semana y mes del cliente.
  //},
  //month: {
    //buttonText: 'Mes',
    // slotLabelFormat: 'HH:mm'
  //},
},
buttonText: {
  today: 'Hoy',
  prev: '<',
  next: '>'
},
//Mas de una columna
resourceGroupField: 'building',
resourceAreaWidth: '25%',
resourceColumns: [
  {
    labelText: 'Recursos',
    field: 'title'
  },
  {

```

```

        labelText: 'H Asi',
        field: 'horas',
        width: '13%'
    },
    {
        labelText: 'H Dis',
        field: 'horasA',
        width: '13%',
        aling: 'center'
    }
],
//resourceLabelText: 'Recursos',
resources: JsonResources,
events: JsonAllocs,

```

A continuación se nombran todos los eventos que se pueden realizar dentro del calendario, pero simplemente se describirá uno solo por simplicidad.

Los eventos con los que cuenta el calendario son:

- Redimensionar un evento
- Arrastrar al calendario
- Cambio de recurso
- Ratón encima de recurso
- Ratón fuera de recursos
- Botón derecho
- Click

En el evento de “arrastrar al calendario” lo primero que se realiza es la asignación de recurso, hora inicio, hora fin y el tipo 2. Se establece tipo 2 ya que se tiene un enumerado que transforma el integer al tipo de evento que se quiera realizar para establecer la comunicación con el programa Microsoft Dynamics NAV. Posteriormente al evento interno del calendario se establecen los valores que tiene que contener extrayéndolo del XML por medio del id del evento, como son por ejemplo: Color por estado, descripción del pedido, estado, cliente, etc. Por último, se actualiza esta información dentro del calendario por medio de la llamada `updateEvent`

Evento `eventReceive`:

```

eventReceive: function (event) {
    // Arrastrar al calendario
    ClearEntry();
    ResizeMoveEntry.OrderNo = event.id;
    ResizeMoveEntry.ResourceNo = event.resourceId;
    ResizeMoveEntry.StartTime = event.start;
    ResizeMoveEntry.EndTime = event.end;
    ResizeMoveEntry.EntryType = 2;
    Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('ResizeMoveEvent',
[ResizeMoveEntry]);
    // Fijar y poner el color
    $xml.find("Pend[ResID*=" + event.id + "]").each(function () {
        event.color = $(this).attr("Color");
        event.Alias = $(this).attr("Alias");
    });
}

```

```

    event.ShipCode = $(this).attr("ShipCode");
    event.Desc = $(this).attr("Description");
    event.status = $(this).attr("Status");
  });
  $('#calendar').FullCalendar('updateEvent', event); //Actualizar el calendario.
},

```

Otro requisito del cliente es que en cada pedido se muestre información previamente solicitada en la fase de análisis de requisitos. Para ello tras realizar el estudio de posibilidades, se estableció en conjunto con el cliente el uso de un Tooltip que se muestre al pasar el ratón por encima y se oculte al quitar el ratón. En el evento Mouseover se establece en qué posición, por medio de coordenadas, queremos que se muestre y la información que tiene que mostrar del pedido.

Eventos Mouseover y Mouseout para Tooltip:

```

//Tooltip
eventMouseover: function (event, jsEvent)
{
    var tooltip = '<div class="tooltipevent">' + '<b>Nº Pedido: </b>' + event.id + '
<br>' + '<b>Nombre Cliente: </b>' + event.Alias + ' <br>' + '<b>Codigo Envio: </b>'
+ event.ShipCode + ' <br>' + '<b>Descripcion: </b>' + event.Desc + ' <br>' +
'<b>Estado: </b>' + event.status + ' <br>' + '<b>Nota: </b>' + event.note + '</div>';
    var $tooltip = $(tooltip).appendTo('body');

    $(this).mouseover(function (e) {
        $(this).css('z-index', 10000);
        $tooltip.fadeIn('700');
        $tooltip.fadeTo('50', 2.9);
    }).mousemove(function (e) {
        $tooltip.css('top', e.pageY - 110);
        $tooltip.css('left', e.pageX - 370);
    });
},

eventMouseout: function (calEvent, jsEvent) {
    $(this).css('z-index', 8);
    $('.tooltipevent').remove();
},

```

Se añade la opción de poder realizar click con el botón derecho sobre cualquier pedido para desplegar una serie de opciones que pueden ser consultadas en el Anexo I. Para poder realizar este botón se ha tenido que añadir el siguiente código.

Contextmenu():

```
$(document).contextmenu({
  delegate: ".hasmenu",
  preventContextMenuForPopup: true,
  preventSelect: true,
  menu: [
    {
      title: "Ver Pedido", action: function(event, ui) {
        var eventID = ui.target.closest('a').attr("data-event-id");
        // Permite ver el pedido en NAV , se envia el numero de pedido.
        ClearEntry();
        ResizeMoveEntry.EntryType = 5;
        ResizeMoveEntry.OrderNo = eventID;
        Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('ResizeMoveEvent',
[ResizeMoveEntry]);
      }
    },
    {
      title: "Crear Nota", action: function (event, ui) {
        var eventID = ui.target.closest('a').attr("data-event-id");
        // Permite agregar nota a un pedido en NAV, se envia el numero de
pedido.
        MostrarTextDiv(eventID);
      }
    },
    {
      title: "Eliminar Asignacion", action: function (event, ui) {
        var eventID = ui.target.closest('a').attr("data-event-id");
        // Permite eliminar la asignación de un pedido
        if (confirm('¿ Seguro que quieres desasignar ' + eventID + '?')) {
          ClearEntry();
          ResizeMoveEntry.EntryType = 4;
          ResizeMoveEntry.OrderNo = eventID;
          Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('ResizeMoveEvent',
[ResizeMoveEntry]);
          $('#calendar').FullCalendar('removeEvents', eventID);
        }
      }
    },
    {
      title: "Crear Orden de Apoyo", action: function (event, ui) {
        var eventID = ui.target.closest('a').attr("data-event-id");
        //Permite crear una orden de apoyo de un pedido
        ClearEntry();
        ResizeMoveEntry.EntryType = 7;
        ResizeMoveEntry.OrderNo = eventID;
        Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('ResizeMoveEvent',
[ResizeMoveEntry]);
      }
    },
    {
      title: "Añadir a Mapa", action: function (event, ui) {
        var eventID = ui.target.closest('a').attr("data-event-id");
```

```

    $xml.find("ResAllocation[ResAllocDoc*=" + eventID + "']").each(function ()
{
    MyMap.geosearch($(this).attr("MapAddress"));
});

    $xml.find("Pend[ResID*=" + eventID + "']").each(function () {
    MyMap.geosearch($(this).attr("MapAddress"));
});
}
},
beforeOpen: function (event, ui)
{
    ui.menu.zIndex($(event.target).zIndex() + 1);
}
});

```

Para que lo anterior funcione se tiene que añadir a los estilos CSS las siguientes líneas de código:

```

.menu{
    width: 100px;
    background: #000;
    color: #fff;
    position: absolute;
    z-index: 999999;
    display: none;
    box-shadow: 0 0 10px #713C3C;
}
.menu ul{
    list-style: none;
    padding: 0;
    margin: 0;
}
.menu ul a{
    text-decoration: none;
}
.menu ul li{
    width: 88%;
    padding: 6%;
    background-color: #F04D44;
    color: #fff;
}
.menu ul li: hover{
    background-color: #F7BA4B;
    color: #444343;
}
}

```

Por último, cabe destacar los cambios que se han realizado para adaptar visualmente el planificador a los colores corporativos y darle un estilo más atractivo para el usuario con algunos efectos.

Efecto tres dimensiones con el sombreado al pasar el ratón por encima de los pedidos de servicio:

```
.fc-event
{
  opacity: 0.8;
}

.fc-event:hover /*Efecto de sombra para cuando el raton pasa por encima de los
eventos*/
{
  opacity: 1;
  -moz-box-shadow: 0px 0px 18px black;
  -webkit-box-shadow: 0px 0px 18px black;
  box-shadow: 0px 0px 18px black;
}
```

Efecto de transparencia para el div de la nota, con esto se consigue que cuando el ratón este fuera del div, el contenido sea semitransparente y una vez que se detecta el evento de ratón encima del div, se convierte en opaco.

```
.fc-event
{
  opacity: 0.8;
}

.fc-event:hover /*Efecto de sombra para cuando el ratón pasa por encima de los
eventos*/
{
  opacity: 1;
  -moz-box-shadow: 0px 0px 18px black;
  -webkit-box-shadow: 0px 0px 18px black;
  box-shadow: 0px 0px 18px black;
}
```

4.3. Cambios de apariencia

Este apartado pretende mostrar el resultado de todos los cambios anteriormente descritos de manera gráfica. En la *Figura 19* se muestra la apariencia final de la aplicación donde se pueden apreciar las diferencias que hemos realizado en el capítulo anterior respecto a la versión inicial *Figura 20*.

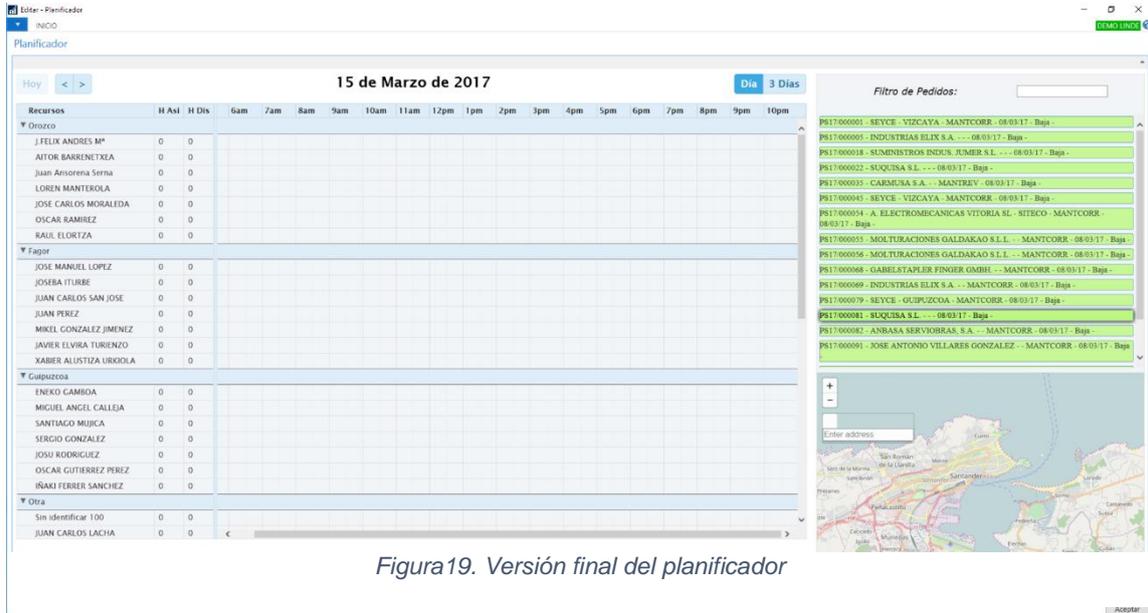


Figura 19. Versión final del planificador

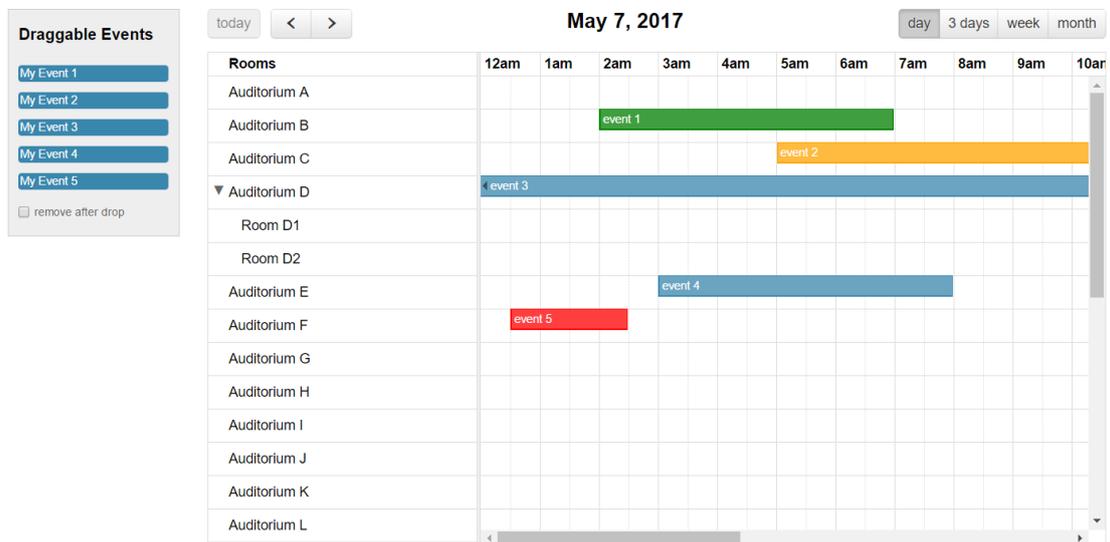


Figura 20. Versión inicial FullCalendar

4.4. Creación e integración Control Add-in

Para poder añadir una extensión a la herramienta Microsoft Dynamics NAV se han tenido que seguir los siguientes pasos:

- 1º. En el proyecto de Visual Studio se ha tenido que crear una nueva clase C# en la cual se ha de que invocar a la librería "Microsoft.Dynamics.Framework.UI.Extensibility.dll". Esta librería se encarga de la comunicación con Microsoft Dynamics NAV.
- 2º. Crear manifest.xml donde se referencian todos los recursos del proyecto para poder ser utilizados en el código web, como son las imágenes, los JavaScript y los CSS.
- 3º. Crear un nuevo fichero C# donde se establece una clase pública que se encargará de especificar que parte de los datos pueden ser leídos y escritos por medio de getters and setters. Se hablará de ello más detalladamente en la parte de paso de datos ya que es una de las partes fundamentales de la comunicación entre esta aplicación y la herramienta Microsoft Dynamics NAV.
- 4º. La información de ensamblado se establece en la clase assemblyInfo.cs donde se establece el nombre del control add-in que en el caso de este proyecto es LindePlanificador, también la versión de nuestro proyecto.

```
[assembly: AssemblyTitle("LindePlanificador")]  
[assembly: AssemblyProduct("LindePlanificador")]  
[assembly: AssemblyCopyright("Copyright © 2016")]  
[assembly: AssemblyVersion("1.1.0.0")]  
[assembly: AssemblyFileVersion("1.1.0.0")]
```

- 5º. El siguiente paso es añadir el proyecto a Microsoft Dynamics NAV. Para ello se ha de compilar la solución y posteriormente generar el proyecto, lo que hace que se cree una public key.
- 6º. Generar un zip en el cual se integren todos los recursos y el manifest.xml como se observa en la *Figura 21*.

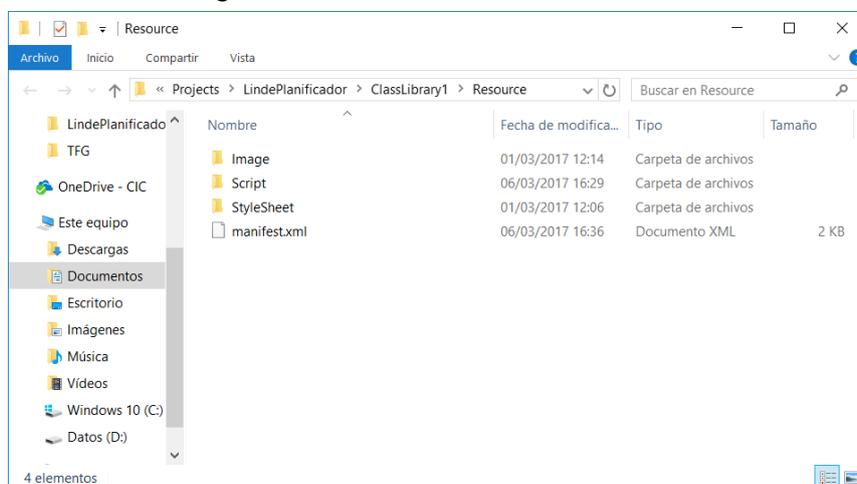


Figura 21. Recursos para añadir al ZIP

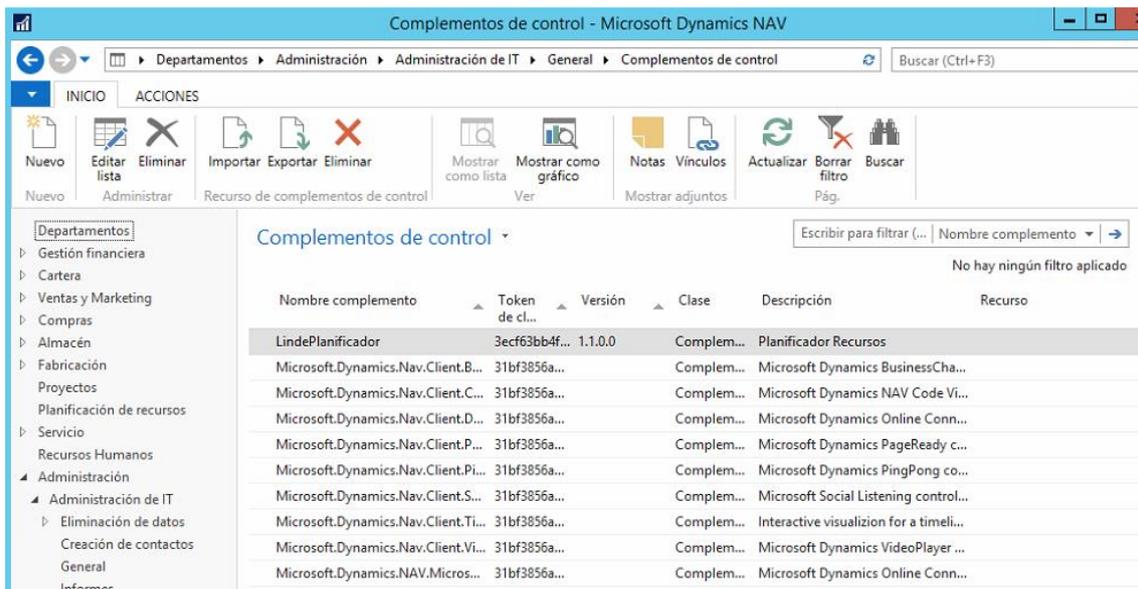


Figura 22. Complementos de Control Microsoft Dynamics NAV

Se añade a los complementos de control el Control add-in con la public key generada y la versión que hemos establecido, el resultado se puede ver en la Figura 22. Posteriormente, en importar se tiene que integrar el zip que se ha generado en el paso anterior con todos los recursos necesarios.

Posteriormente, se debe añadir la DLL del proyecto a la carpeta de Control add-in del servidor, para ello se tiene que ir a la siguiente ruta: C://Archivos de Programa/Microsoft Dynamicis NAV/90/Service/Add-ins/ e incluir la DLL en el directorio. Ejemplo DLL copiada en directorio en la Figura 23.

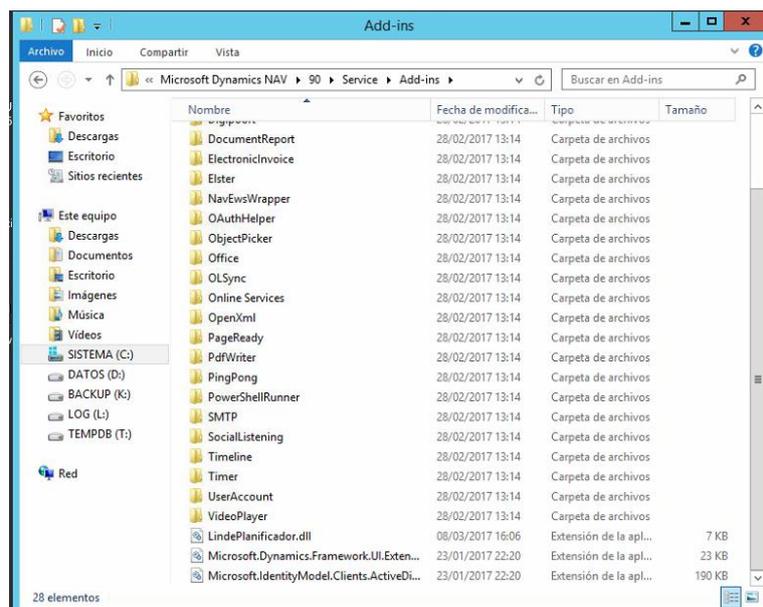


Figura 23. DLL en directorio C://Archivos de Programa/Microsoft Dynamicis NAV/90/Service/Add-ins/

4.5. Integración Mapa Leaflet

En el proceso de cumplir el requisito de disponer de un mapa en el cual se puedan añadir las localizaciones de cada uno de los pedidos de servicio asignados por medio del botón derecho, se realizó un estudio de todas las alternativas disponibles en el mercado, evaluando la opción de Google pero descartándola por el número limitado gratuito de localizaciones al mes. Finalmente, se optó por la opción de open-source Leaflet, una librería de código abierto de javascript para interactuar con los mapas.

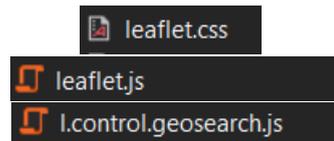


Figura 23. Librerías Leaflet

Tras integrar el mapa se presentó el problema de la geolocalización ya que por parte de Leaflet se establece que la búsqueda se realiza por medio de coordenadas de Latitud y Longitud. La manera que entrega la información la aplicación de Microsoft Dynamics NAV es por medio de una dirección.

Para solucionar este problema se añadió la siguiente librería, que se encarga de hacer la traducción de una dirección a las coordenadas de latitud y longitud de dicha dirección.

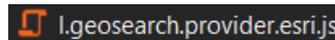


Figura 24. Librería Geolocalización.

Posteriormente, añadiendo una nueva función que invoca la clase anteriormente nombrada con los parámetros necesarios proporciona la latitud y longitud que posteriormente se pasa al mapa para realizar la localización del pedido de servicio.

Función MakeMap

```
function MakeMap()
{
  var osmTileUrl = 'http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png';
  var basemap = new L.TileLayer(osmTileUrl, { maxZoom: 18 });
  map = new L.Map('map', {
    layers: [basemap],
    center: new L.LatLng(43.462, -3.80), zoom: 12
  });
  MyMap = new L.Control.GeoSearch({
    provider: new L.GeoSearch.Provider.Esri(),
    showMarker: true,
  }).addTo(map);
}
```

A petición del cliente se establece que el centro del mapa sea configurable, por lo cual añadiendo la Latitud y Longitud de la ubicación de cada uno de nuestros usuarios se puede establecer que la visión principal sea la ubicación de la empresa.

4.6. Adaptación grupo Italiano

Como se ha comentado el presente documento se ha realizado para un cliente español y uno italiano por lo que un requisito es que sea adaptable a los dos idiomas.

A continuación, se describirá todos los pasos realizados para poder adaptar el idioma:

Para poder modificar el idioma de los menús superiores del calendario es necesario añadir una librería nueva que compone todo el calendario con el idioma indicado en la construcción.

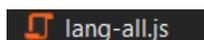


Figura 25. Librería Lenguajes

```
lang: 'it',  
locale: 'it',  
timezone: 'Europe/Roma',
```

Posteriormente se tiene que realizar una traducción en el código para adaptarlo al nuevo idioma, para ello se realiza una traducción de todas las etiquetas y se modifica en el código, por simplicidad solo se muestra el ejemplo de la modificación de una de ellas. Se muestran en la *Figura 26*.

Español	Italiano
Recursos	Risorse
Hoy	Oggi
Día	Giorno
3 Días	3 Giorni
Filtro Pedido	Filtro Ordine
Nº Pedido	Nr. Ordine
Nombre Cliente	Nome Cliente
Código Envío	Cod. Spedizione
Descripción	Descrizione
Estado	Stato
Nota	Nota
Ver Pedido	Mostra Ordine
Crear Nota	Creare Nota
Eliminar Asignación	Cancellare Asegnazione
Crear Orden de Apoyo	Creare Ordine appoggio
Escriba su nota aquí	Scriba la sua nota qui
Aceptar	Accettare
Cancelar	Cancellare
Horas Disponible	Ore Disponibile
Horas Asignadas	Ore Assegnate
Seguro que quieres desasignar	Sei sicuro di volere cancellare l'asegnazione?
Seguro que quieres dejar de ver el recurso del Planificador	Sei sicuro di nascondere il Risorso dal Pianificatore?

Figura 26. Tabla de traducciones.

Ejemplo de modificación del código para adaptación al nuevo idioma:

```
//Tooltip
eventMouseover: function (event, jsEvent)
{
    var tooltip = '<div class="tooltipevent">' + '<b>Nr. Ordine: </b>' + event.id + '
<br>' + '<b>Nome Cliente: </b>' + event.Alias + ' <br>' + '<b>Cod. Spedizione:
</b>' + event.ShipCode + ' <br>' + '<b>Descrizione: </b>' + event.Desc + ' <br>' +
'<b>Stato: </b>' + event.status + ' <br>' + '<b>Nota: </b>' + event.note + '</div>';
    var $tooltip = $(tooltip).appendTo('body');
```

Por último, se modifica en los ficheros de Microsoft Dynamics NAV los caption para añadirles la opción italiana. Al igual que en el ejemplo anterior se muestra un único ejemplo de modificación por simplicidad.

Language	Value
Inglés (Estados Unidos)	Unassigned Order Calculation
Español (España, tradicional)	Formula Cálculo Pedidos Sin Asignar
Italiano (Italia)	Gli ordini di calcolo Formula non assegnati

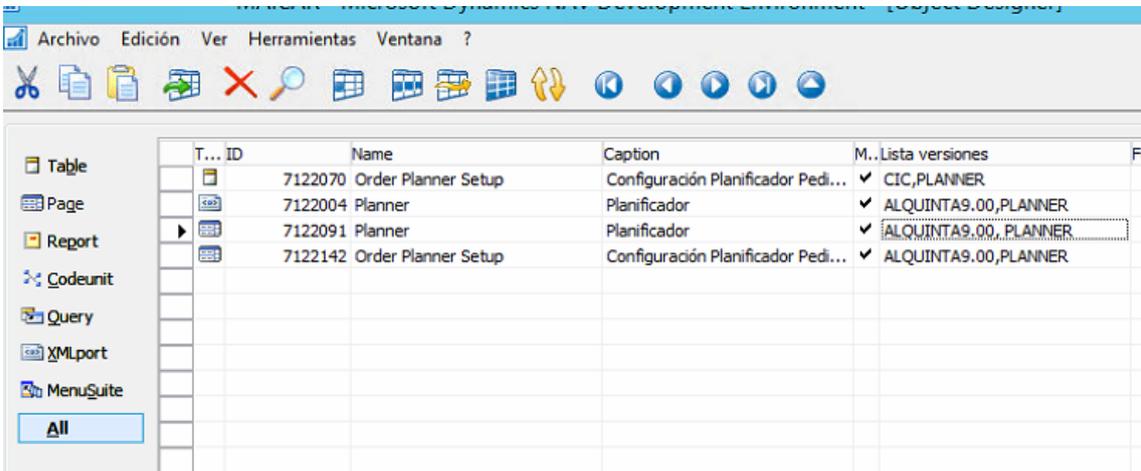
Figura 27. Caption de Campo Microsoft Dynamics NAV

5. Adaptación Microsoft Dynamics NAV

Microsoft Dynamics NAV se divide en 7 tipos de estructuras:

- **Tablas.** Conjunto de variables que se almacenan en la base de datos.
- **Paginas.** Interfaces que son mostradas al usuario para que interactúe, suelen ir asociadas a tablas. Son la manera que el usuario tiene de añadir datos a las tablas.
- **Informes.** Representación gráfica de datos por medio de Layouts creados por Visual Studio.
- **Codeunit.** Procesos de tratamiento de datos que son ejecutadas sobre la base de datos, al igual que las **Query**. Se usan para realizar modificaciones o consulta de datos.
- **XMLPort.** Método por el cual se pueden escribir y leer datos de aplicaciones externas a Microsoft Dynamics NAV.
- **MenuSuite.** Parte donde se personaliza que mostrar en el menú de búsqueda.

Para este proyecto se han tenido que usar 4 de ellas: Tabla, Paginas, XMLPort y MenuSuite, la *Figura 28* muestra cada una de ellas.



The screenshot shows the Microsoft Dynamics NAV Development Environment interface. On the left, there is a navigation pane with icons for Table, Page, Report, Codeunit, Query, XMLport, and MenuSuite. The main area displays a table with the following data:

T... ID	Name	Caption	M...Lista versiones
7122070	Order Planner Setup	Configuración Planificador Peda...	✓ CIC,PLANNER
7122004	Planner	Planificador	✓ ALQUINTA9.00,PLANNER
7122091	Planner	Planificador	✓ ALQUINTA9.00, PLANNER
7122142	Order Planner Setup	Configuración Planificador Peda...	✓ ALQUINTA9.00,PLANNER

Figura 28. Elementos creados en Microsoft Dynamics NAV

Para cumplir los requisitos de configuración de colores por estado y filtro de fecha de pedidos se crean dos tipos de estructuras:

- **Tabla Order Planner Setup:** La cual cuenta con todos los campos que son configurables por el usuario. Es donde se almacenan los datos introducidos por el usuario. En la *Figura 29* se muestra un ejemplo de los campos de la tabla.

E..	Field No.	Field Name	Data Type	Length	Description
▶	✓ 1	Primary Key	Code	10	
	✓ 2	Unassigned Order Calculation	DateFor...		

Figura 29. Campos Tabla Order Planner Setup

- Página Order Planner Setup:** Interfaz gráfica donde se solicita al cliente que introduzca qué rango de fechas quiere ver dentro del planificador gráfico y un botón que les lleva a la configuración de los colores por los estados personalizados que tiene cada uno de ellos. Como se ve en la *Figura 30* se menciona la misma variable que se declaró en la tabla anterior.

E..	Type	SubType	SourceExpr	Name	Caption
	Container	ContentArea		<Control2>	<<Control2>>
	Group	Group		<Control1000...	Parámetros
▶	Field		Unassigned Order Calculation	<Unassigned ...	<Formula Cálculo Pedidos ...

Figura 30. Variables y grupos de la página de configuración.

En la parte de las propiedades de la página de configuración por medio de SourceTable se establece la relación con la tabla anteriormente descrita para almacenar el dato que el cliente pueda introducir, como se muestra en *Figura 31*.

Property	Value
ID	
Name	Order Planner Setup
Caption	Configuración Planificador Pedidos
CaptionML	ENU=Order Planner Setup;ESP=Configuración Planificador Pedidos
Editable	<Yes>
Description	<>
Permissions	<Undefined>
PageType	<Card>
InstructionalTextML	<Undefined>
CardPageID	<Undefined>
DataCaptionExpr	<Undefined>
RefreshOnActivate	<No>
PromotedActionCategoriesML	<Undefined>
SourceTable	Order Planner Setup
SourceTableView	<Undefined>
InsertAllowed	No
ModifyAllowed	<Yes>
DeleteAllowed	No
DelayedInsert	<No>
ShowFilter	<Yes>
MultipleNewLines	<No>
SaveValues	<No>
AutoSplitKey	<No>
DataCaptionFields	<Undefined>
SourceTableTemporary	<No>
LinksAllowed	<Yes>
PopulateAllFields	<Undefined>

Figura 31. Propiedades página de configuración.

Después, se ha de añadir otra estructura denominada XMLPort, en la que se configurará qué datos tiene que entregar la herramienta Microsoft Dynamics NAV para ser tratada dentro del aplicativo. Esto se describirá más detalladamente en el siguiente capítulo, ya que forma parte del paso de datos.

Por último se tiene que crear una página denominada Planner, que es la encargada de tener las mismas estructuras que se han creado en la parte del aplicativo, la cual tratará los datos devueltos por el aplicativo. Al igual que lo anterior, se detallará en el próximo capítulo, ya que forma parte del paso y tratamiento de los datos.

6. Paso y tratamiento de datos

En este capítulo se mostrará la manera que tiene Microsoft Dynamics NAV de comunicar y leer los datos por parte de la aplicación de la que es objeto el proyecto.

6.1. Escritura de datos

El paso de mensajes se realiza como hemos indicado anteriormente por medio de la estructura de datos XMLPort, por lo que en este proyecto se ha procedido a crear esta estructura y configurarla para las necesidades del proyecto.

Se establecen datos de 4 tablas distintas que son necesarios para el funcionamiento de la aplicación:

- **Tabla de Params:** Tabla de la cual se usa la variable date para realizar el filtro de los pedidos que hemos definido anteriormente.
- **Tabla Resource:** Tabla donde se encuentran todos los recursos, de ellos se necesitan su clave primaria y única, número (No), Nombre, Familia, Horas Disponibles y Horas asignadas.
- **Tabla ResAllocation:** Tabla de la cual se extraen todos los pedidos asignados. Entre sus parámetros se encuentran Numero de recurso que es propietario de ese pedido, hora de inicio, hora de fin, prioridad, descripción del pedido, alias, color por estado, dirección para ser añadida al mapa y nota.
- **Tabla Unassigned:** Tabla de la que se extraen todos los pedidos que están registrados pero pendientes de asignación. Entre sus elementos se encuentran Número de identificación del pedido, descripción, color del estado en el que se encuentra, estado, alias, y dirección para ser añadida al mapa.

Node Name	Prefix	Node Type	Source Type	Data Source
		Element	Text	<root>
Params		Element	Table	<Integer>(Integer)
Date		Attribute	Text	CurrentDate
Resource		Element	Table	<Resource>(Resource)
No		Attribute	Field	Resource:No.
Name		Attribute	Field	Resource:Name
Family		Attribute	Text	<Family>
Hours		Attribute	Text	<Hours>
HoursA		Attribute	Text	<HoursA>
ResAllocation		Element	Table	ResAllocation(Service Ord...
ResResNo		Attribute	Field	ResAllocation:Resource No.
ResAllocDoc		Attribute	Field	ResAllocation:Document No.
ResAllocStart		Attribute	Text	<ResAllocStart>
ResAllocEnd		Attribute	Text	<ResAllocEnd>
ResAllocDate		Attribute	Field	ResAllocation:Allocation Date
Priority		Attribute	Text	<Priority>
Description		Attribute	Text	<Description>
ShipToCode		Attribute	Text	<ShipToCode>
MapAddress		Attribute	Text	MapAddress
Alias		Attribute	Text	<Alias>
Color		Attribute	Text	ResColor
Status		Attribute	Text	<Status>
Note		Attribute	Text	<Note>
Pend		Element	Table	UnassignedAllocations(Se...
ResID		Attribute	Field	UnassignedAllocations:Docum...
ResDesc		Attribute	Text	UnassignedDesc
Priority		Attribute	Text	PriorityNum
Color		Attribute	Text	UnColor
MapAddress		Attribute	Text	MapAddress
Description		Attribute	Text	Description2
ShipToCode		Attribute	Text	ShipToCode2
Alias		Attribute	Text	Alias2
Status		Attribute	Text	Status2

Figura 32 Conjunto de variables página Planner

A continuación, se muestran ejemplos de cómo se realizan los filtros para la entrega de datos en las Figuras 33 y 35:

- Primero se pasa la fecha indicada al formato del planificador para poder añadirlo, esto se realiza tanto para la hora de inicio como para la hora de fin.
- Después de la tabla de Service Header, en los datos donde su tipo de documento es order y su número de documento coincide con el de nuestro pedido sin asignar, se extraen los datos necesarios: Descripción, código de envío, alias y nota.
- Para asegurarse de que la dirección está correctamente formada por un formato predeterminado, se evalúa si tiene algún tipo de dato y se forma con el formato internacional: dirección, ciudad y provincia, en caso de que no tenga dirección de envío se establece la dirección de la empresa extraída de la cabecera formándola de la misma manera que la anterior.
- Se extrae la prioridad
- De la tabla ServItemLine se obtiene cualquier pedido en el cual coincida que su tipo de documento es order y su número de documento corresponda con el del pedido que se tiene sin asignar y su número de línea es coincidente con el del pedido sin asignar. Cuando se cumplen todas las condiciones anteriormente descritas, se extrae el estado y el color por estado.

```

91 ResAllocation - Export::OnAfterGetRecord()
92 TempdateTime := CREATEDATETIME(ResAllocation."Allocation Date",ResAllocation."Starting Time" + 7200000);
93 ResAllocStart := FORMAT(TempdateTime,0,9);
94 ResAllocStart := Text;
95 TempdateTime := CREATEDATETIME(ResAllocation."Allocation Date",ResAllocation."Finishing Time"+ 7200000);
96 ResAllocEnd := FORMAT(TempdateTime,0,9);
97
98 IF ServHeader.GET(ServHeader."Document Type"::Order,ResAllocation."Document No.") THEN BEGIN
99 Description := ServHeader.Description;
100 ShipToCode := ServHeader."Ship-to Code";
101 Alias := ServHeader.Name;
102 Note := ServHeader.Note;
103
104 IF ServHeader."Ship-to Code" <> '' THEN BEGIN
105 MapAddress := ServHeader."Ship-to Address" + ',' + ServHeader."Ship-to City";
106 END ELSE BEGIN
107 MapAddress := ServHeader.Address + ',' + ServHeader.City;
108 END;
109
110 END;
111 Priority := FORMAT(ServHeader.Priority);
112
113
114 IF ServItemLine.GET(ServHeader."Document Type"::Order,ResAllocation."Document No.",ResAllocation."Service Item Line No.") THEN BEGIN
115 IF RepairStatus.GET(ServItemLine."Repair Status Code") THEN BEGIN
116 Status := RepairStatus.Code;
117 ResColor := RepairStatus."Planner Color";
118 END;
119 END;
120

```

Figura 33. Código función pedidos asignados

Al igual que en el caso anterior caso, se repiten los patrones de filtrado y extracción de datos con la única diferencia de que se forma una descripción con una concatenación de datos añadiendo un espacio en blanco entre cada uno de los valores. También se añade una última línea en la cual, cuando no se encuentre ya ningún pedido de servicio sin asignar, se realiza un currXMLpor.SKIP. Esto hace que se termine la ejecución.

```

184 UnassignedAllocations - Export::OnAfterGetRecord()
185 UnassignedDesc := UnassignedAllocations."Document No.";
186 IF ServHeader.GET(ServHeader."Document Type"::Order,UnassignedAllocations."Document No.") THEN BEGIN
187
188 IF ServHeader.Status = ServHeader.Status::Finished THEN
189 currXMLport.SKIP;
190
191 // UnassignedDesc := UnassignedAllocations."Document No.";
192
193 IF ServHeader."Ship-to Code" <> '' THEN BEGIN
194 MapAddress := ServHeader."Ship-to Address" + ',' + ServHeader."Ship-to City";
195 END ELSE BEGIN
196 MapAddress := ServHeader.Address + ',' + ServHeader.City;
197 END;
198
199 Description2 := ServHeader.Description;
200 ShipToCode2 := ServHeader."Ship-to Code";
201 Alias2 := ServHeader.Name;
202
203 CLEAR(ServItemLine);
204 IF ServItemLine.GET(ServItemLine."Document Type"::Order,UnassignedAllocations."Document No.",UnassignedAllocations."Service Item
205 IF RepairStatus.GET(ServItemLine."Repair Status Code") THEN BEGIN
206 Status2 := RepairStatus.Code;
207 UnColor := RepairStatus."Planner Color";
208 END;
209 END;
210
211 UnassignedDesc += //txtSeparator + ServHeader."Customer No." +
212 txtSeparator + ServHeader.Name +
213 txtSeparator + ServHeader."Ship-to Code" +
214 txtSeparator + ServHeader."Service Order Type" +
215 txtSeparator + FORMAT(ServHeader."Order Date") +
216 txtSeparator + FORMAT(ServHeader.Priority) +
217 txtSeparator + ServItemLine."Service Zone Code";
218
219 PriorityInt := ServHeader.Priority;
220 PriorityNum := FORMAT(PriorityInt);
221 //Counter += 1;
222 //IF Counter > 100 THEN
223 // currXMLport.BREAK;
224 END ELSE BEGIN
225 currXMLport.SKIP;
226 END;

```

Figura 34. Código función pedidos sin asignar

6.2. Lectura de datos

La lectura de datos se realiza por medio de la página Planner.

Al realizar cualquier evento se invoca a:

```
Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('ResizeMoveEvent',
[ResizeMoveEntry]);
```

Esto es una llamada a las funciones que se encuentran establecidas en la página Planner. Al ejecutar la aplicación el programa Microsoft Dynamics NAV se queda en segundo plano escuchando si se efectúa cualquiera de las llamadas establecidas en la página encargada de invocar a este aplicativo. De esta manera si se efectúa cualquier cambio en los datos y se invoca desde el código web, se pasan los datos que se encuentran dentro de la estructura que hemos descrito en los capítulos anteriores y se pasan los datos necesarios, dejando en blanco los que no hacen falta.

6.3. Tratamiento de datos

Una vez que se han leído los datos, el último paso es realizar las operaciones oportunas con ellos, dependiendo de cada uno de los casos que se planteen se tendrá que escribir los datos en una u otra tabla.

A continuación, por simplicidad, se muestran dos de los casos de cómo se realiza el tratamiento de datos que se reciben de la aplicación.

La primera parte del código se efectúa si el tipo de llamada que se ha recibido contiene la variable de tipo con un Resize; en caso de que sea así, lo que se hace es sobre una variable temporal de la tabla de OrderAllocation almacenar todos los datos que se han recibido de la aplicación para posteriormente, con la instrucción ServOrderAllocation.MODIFY, realizar la escritura definitiva de datos en la tabla. El segundo caso no se va a describir tan detalladamente porque es igual.

```
IF Entry.EntryType.Equals(EntryTypes.Resize) THEN BEGIN
    // Localizar la asignacion y modificar su hora inicio y fin
    CLEAR(ServOrderAllocation);
    ServOrderAllocation.SetHideDialog(TRUE);
    ServOrderAllocation.SETRANGE("Document
Type",ServOrderAllocation."Document Type"::Order);
    ServOrderAllocation.SETRANGE(Status,ServOrderAllocation.Status::Active);
    ServOrderAllocation.SETRANGE("Document No.",Entry.OrderNo);
    ServOrderAllocation.SETRANGE("Resource No.",Entry.ResourceNo);
    IF ServOrderAllocation.FINDFIRST THEN BEGIN
        ServOrderAllocation."Starting Time" := 0T;
        ServOrderAllocation."Finishing Time" := 0T;
        ServOrderAllocation.VALIDATE("Starting Time",GetTime(Entry.StartTime));
        ServOrderAllocation.VALIDATE("Finishing Time",GetTime(Entry.EndTime));
        ServOrderAllocation.VALIDATE("Allocation Date",GetDate(Entry.StartTime));
        ServOrderAllocation.VALIDATE(ServOrderAllocation."Allocated Hours",
            (ServOrderAllocation."Finishing Time" -
ServOrderAllocation."Starting Time") /3600000);

        ServOrderAllocation.MODIFY;
    END ELSE BEGIN
        ERROR(txtAllocationUnfound,Entry.OrderNo);
    END;
    END ELSE IF Entry.EntryType.Equals(EntryTypes.MoveSameResource) THEN
BEGIN
    // Arrastrar asignación.
    CLEAR(ServItemLine);
    ServItemLine.SETRANGE(ServItemLine."Document
Type",ServItemLine."Document Type"::Order);
    ServItemLine.SETRANGE(ServItemLine."Document No.",Entry.OrderNo);
    IF ServItemLine.FINDFIRST THEN
        LineNo := ServItemLine."Line No.";

    // Localizar la asignación y modificar su hora inicio y fin
    CLEAR(ServOrderAllocation);
```

```

ServOrderAllocation.SetHideDialog(TRUE);
ServOrderAllocation.SETRANGE("Document
Type",ServOrderAllocation."Document Type"::Order);
ServOrderAllocation.SETRANGE("Document No.",Entry.OrderNo);
ServOrderAllocation.SETRANGE("Service Item Line No.",LineNo);
IF ServOrderAllocation.FINDFIRST THEN BEGIN
  ServOrderAllocation.SetHideDialog(TRUE);
  ServOrderAllocation.VALIDATE("Resource No.",Entry.ResourceNo);
  ServOrderAllocation."Starting Time" := 0T;
  ServOrderAllocation."Finishing Time" := 0T;
  ServOrderAllocation.VALIDATE("Starting Time",GetTime(Entry.StartTime));
  ServOrderAllocation.VALIDATE("Finishing Time",GetTime(Entry.EndTime));
  ServOrderAllocation.VALIDATE("Allocated Hours",
    (ServOrderAllocation."Finishing Time" -
ServOrderAllocation."Starting Time") /3600000);
  ServOrderAllocation.VALIDATE("Allocation Date",GetDate(Entry.StartTime));
  ServOrderAllocation.MODIFY(TRUE);
END;

```

7. Conclusiones

Los objetivos a cumplir por la aplicación quedaron definidos en el apartado 1.2. A continuación, se hará un repaso de ellos y de si han sido cumplidos:

- Se ha efectuado un análisis de requisitos que posteriormente han sido cumplidos por orden de prioridad.
- Se ha efectuado la adaptación de la herramienta FullCalendar para cumplir los requisitos establecidos.
- La aplicación del presente proyecto está completamente integrada con la herramienta Microsoft Dynamics NAV.
- Se ha establecido la comunicación de datos en ambos sentidos por medio de XML y llamadas a funciones.
- Se ha realizado la implementación e integración en cliente final, siendo usada por este de manera efectiva y correcta de forma diaria.

Tanto actualmente como en el futuro se seguirá trabajando en este aplicativo para adaptarlo a las necesidades de clientes que ya cuentan con él, como de futuros clientes que están interesados en integrarlo a sus herramientas Microsoft Dynamics NAV.

Bibliografía

- [1] Microsoft Dynamics NAV - <https://www.microsoft.com/es-xl/dynamics365/nav-overview>
- [2] SQL Server - <https://www.microsoft.com/es-es/sql-server/sql-server-2016>
- [3] Team Viewer 10 - <https://www.teamviewer.com/es/features/>
- [4] Microsoft Dynamics NAV Server - <http://www.lemursolution.com/node/91>
- [5] JIRA - <https://es.wikipedia.org/wiki/JIRA>
- [6] JIRA - <https://es.atlassian.com/software/jira>
- [7] Microsoft Office 2016 - <http://searchdatacenter.techtarget.com/es/definicion/Microsoft-Office-2016>
- [8] Microsoft Office - <https://products.office.com/es-es/>
- [9] Microsoft Visual Studio - https://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- [10] Fortigate - <http://forticlient.com/>
- [11] FortiClient - <https://bizagi.zendesk.com/hc/es/articles/204453160--Qu%C3%A9-es-y-c%C3%B3mo-usar-la-VPN-FortiClient->
- [12] Windows Server 2016 - <https://iiemd.com/windows-server/que-es-windows-server>
- [13] Leaflet - <http://leafletjs.com/>
- [14] GeoSearch - <https://github.com/smeijer/leaflet-geosearch>
- [15] StackOverFlow - <https://es.stackoverflow.com/>
- [16] FullCalendar - <https://FullCalendar.io/>
- [17] CIC - <http://www.cic.es/>

Anexo I

A continuación, se añade la instrucción técnica creada para el uso de la aplicación, en ella se describe detalladamente las instrucciones para realizar cualquier operación. Este manual proporciona una idea general del funcionamiento del planificador desarrollado en el presente proyecto.

Planificador Grafico de Pedidos de Servicio

Manual de configuración y uso.

Este documento contiene una guía para el usuario para el correcto uso y funcionamiento del planificador de Pedidos de Servicio en el Vertical ALQUINTA sobre Dynamics NAV 2016.

Autor: JLP
Versión: 1.0

Gerencia: PYME

Santander a 01/03/2016

1	INSTALACIÓN DEL PLANIFICADOR	1
2	CONFIGURACIONES NECESARIAS	2
2.1	PARÁMETROS DE CONFIGURACIÓN DEL PLANIFICADOR	2
2.2	CONFIGURACIÓN DE COLORES POR ESTADOS DE REPARACIÓN	3
3	USO DEL PLANIFICADOR	5
3.1	APARIENCIA Y ELEMENTOS	6
3.1.1	Lista de Pedidos Pendientes de Asignar	6
3.1.2	Mapa	7
3.1.3	Área de Trabajo	8
3.2	ASIGNACIÓN DE PEDIDOS DE SERVICIO	10
3.3	OPCIONES ADICIONALES	10
3.3.1	Ampliación/Reducción tiempo asignación Pedido de Servicio.	11
3.3.2	Cambio de asignación de Pedido de Servicio.	11
3.3.3	Tooltip con información adicional	11
3.3.4	Acciones con Pedido de Servicio desde el planificador	12
3.3.5	Notificaciones Globales	14
4	RESULTADO Y CONCLUSIONES	15

INTRODUCCIÓN

Planificador para la gestión gráfica de recursos en Microsoft Dynamics NAV. Este manual muestra las principales características de dicho control gráfico, menús, configuraciones, modificaciones gráficas, y usos básicos.

1 Instalación del Planificador

El Planificador Gráfico de Pedidos de Servicio se encuentra instalado por defecto en cada una de las migraciones a la versión de Dynamics NAV 2016 realizada en cada uno de los concesionarios de Linde Material Handling S.A.

Para poder hacer uso sin problemas del Planificador, será necesario tener instalado la última versión de JAVA, algo que por lo general todos los equipos con Windows tienen. En caso de no encontrarse en el equipo, podrá ser descargado gratuitamente desde la web de Oracle en: <https://www.java.com/es/download/>

También será necesario que el equipo tenga instalado una versión de Internet Explorer 9 o superior.

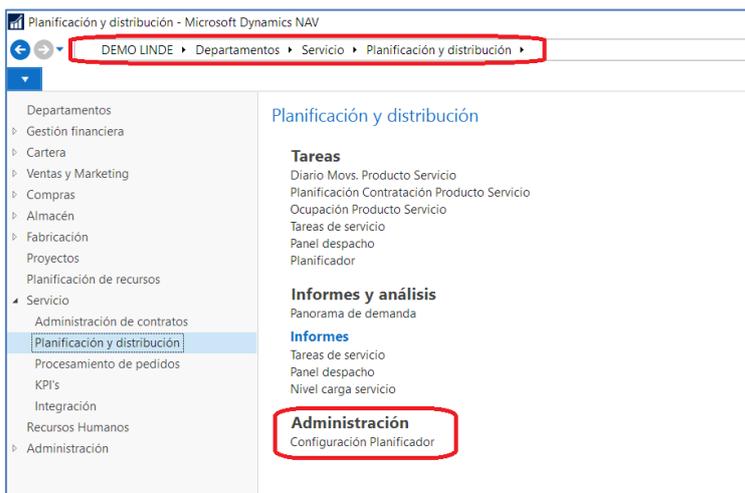
2 Configuraciones necesarias

El Planificador trabaja con muy pocas configuraciones necesarias. Aunque estas configuraciones irán aumentando a medida que se vayan incrementando las funcionalidades del propio planificador, todas ellas se encontrarán en la página de Configuración del Planificador.

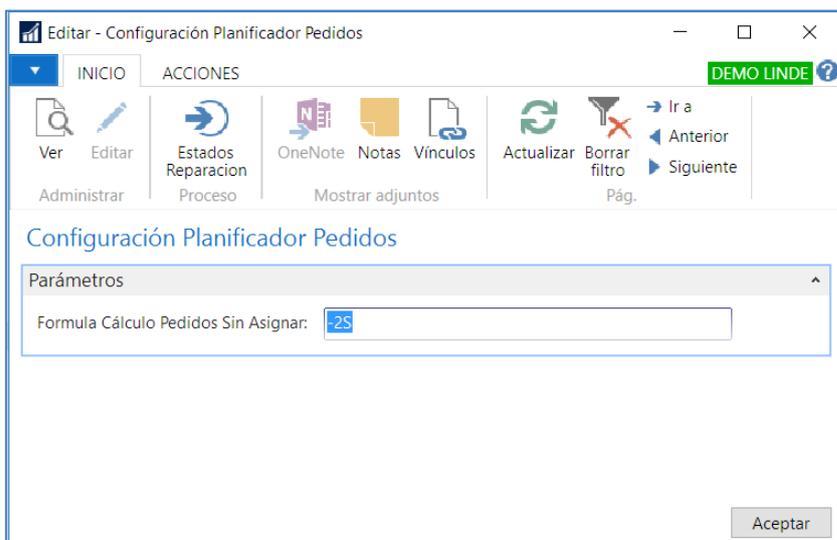
2.1 Parámetros de Configuración del Planificador

Como se indicaba anteriormente, el planificador cuenta con una página de configuración en la cual se podrán establecer los parámetros necesarios.

Esta página se encuentra dentro de NAV 2016 en: Departamentos → Servicio → Planificación y Distribución → Administración → Configuración Planificador, tal y como se aprecia en la siguiente imagen.



Al abrir esta página, se nos mostrará una similar a esta:



Actualmente esta página solo cuenta con un campo, denominado “Formula Calculo Pedidos sin Asignar”. En él, se deberá introducir la fórmula de fecha que se utilizará posteriormente para obtener la lista de pedidos de servicio pendientes de asignar. Esta fórmula deberá ser siempre **negativa**.

Esta fórmula se puede establecer por Día (D), por semana (S), por año (A) y si por ejemplo queremos hacer que se muestren los pedidos con fecha de hace 3 semanas tendríamos que poner: -3S.

Desde esta página podremos acceder también a la página de configuración de estados de reparación.

2.2 Configuración de Colores por Estados de Reparación

La siguiente configuración del Planificador, nos llevará a definir los diferentes colores que queremos que se nos muestre en la ventana principal. Estos colores se definirán para cada uno de los estados de reparación.

La página de configuración de estados es accesible desde la página de Configuración Planificador, o la podremos encontrar en: Departamentos → Servicio → Procesamiento de Pedidos → Configuración → Estado → Config. Estado Reparación.

Esto nos abrirá la página siguiente:

Código	Descripción	Estado ped. servicio	Prioridad	Indice AMPLUS	Color Planificador	Inicial	En proceso	Terminada	Remitido	Esperando al cliente	Oferta terminada	Registro permitido	Estado pendiente permitido	Estado en proceso permitido	Estado terminado permitido	Estado en espera permitido	No permitir insertar nuevas líneas de servicio
A FACTURAR	A facturar	Terminado	Baja	-1	#819FF7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EN ESPERA	En espera	En espera	Media baja	2	#BDBDBD	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
EN PROCESO	Servicio en proceso y asignado	En proceso	Alta	1	#FA5882	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
EN TALLER	En taller	Pendiente	Media alta	-1	#FA58DD	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ESPERA REC	En espera recambio	En espera	Media baja	7		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
ESPERA T	En espera transportista	En espera	Media baja	6	#BDBDBD	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
FINALIZADO	Servicio finalizado	Terminado	Baja	3	#D358F7	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
INICIAL	Estado inicial rep.	Pendiente	Media alta	-1	#8258FA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PARTE ENTR	Parte entregado Adm. PV	Terminado	Baja	-1	#5858FA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PDTE. FIRM	Pendiente Firma	Pendiente	Media alta	4	#2EFEF7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PED OFI	pendiente oficina	Terminado	Baja	-1	#2EFEF7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
PEND PARTE	Pendiente entrega parte mecani	En espera	Media baja	-1	#2EFE2E	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PEND.NP	Pendiente Numero Pedido	Terminado	Baja	-1	#FE2E2E	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PENDIENTE	pendiente	Pendiente	Media alta	0	#5858FA	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
PRESUPUEST	Pendiente de Aceptación Presup	En espera	Media baja	-1	#FE9A2E	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PTE MATERI	Pendiente Materiales	En espera	Media baja	-1	#D7DF01	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
REMITIDO	Remitido	Pendiente	Media alta	-1	#BEF781	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

En la columna resaltada en rojo, se podrá configurar el color que deberán tomar los Pedidos de Servicio según el estado en el que se encuentren. El color se deberá especificar utilizando el código Hexadecimal de cualquier color RGB.

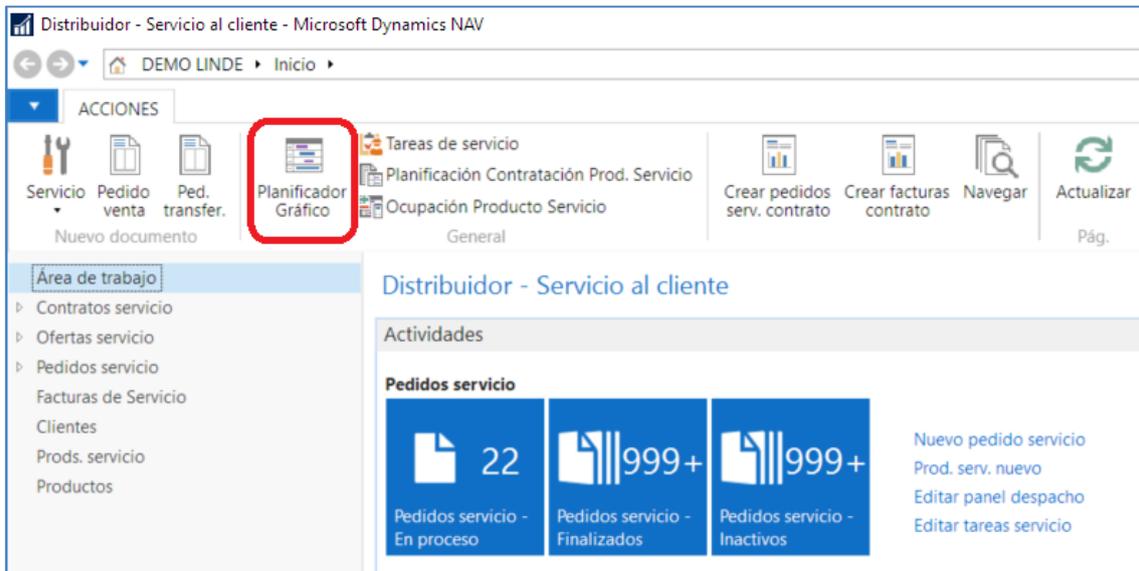
A continuación incluimos una serie de colores sugeridos para los principales estados de reparación, pero cada concesionario podrá configurar sus colores.

Cód. Estado	Código HEX	RGB	Muestra
INICIAL	#BEF781	190,247,129	
REMITIDO	#D7DF01	215,223,1	
EN PROCESO	#2EFE2E	46,254,46	
EN ESPERA	#BDBDBD	189,189,189	
PTE MATERI	#FE2E2E	254,46,46	
EN TALLER	#819FF7	129,159,247	
PDTE. FIRM	#FE9A2E	254,154,46	
FINALIZADO	#2EFEF7	46,254,247	
A FACTURAR	#5858FA	88,88,250	

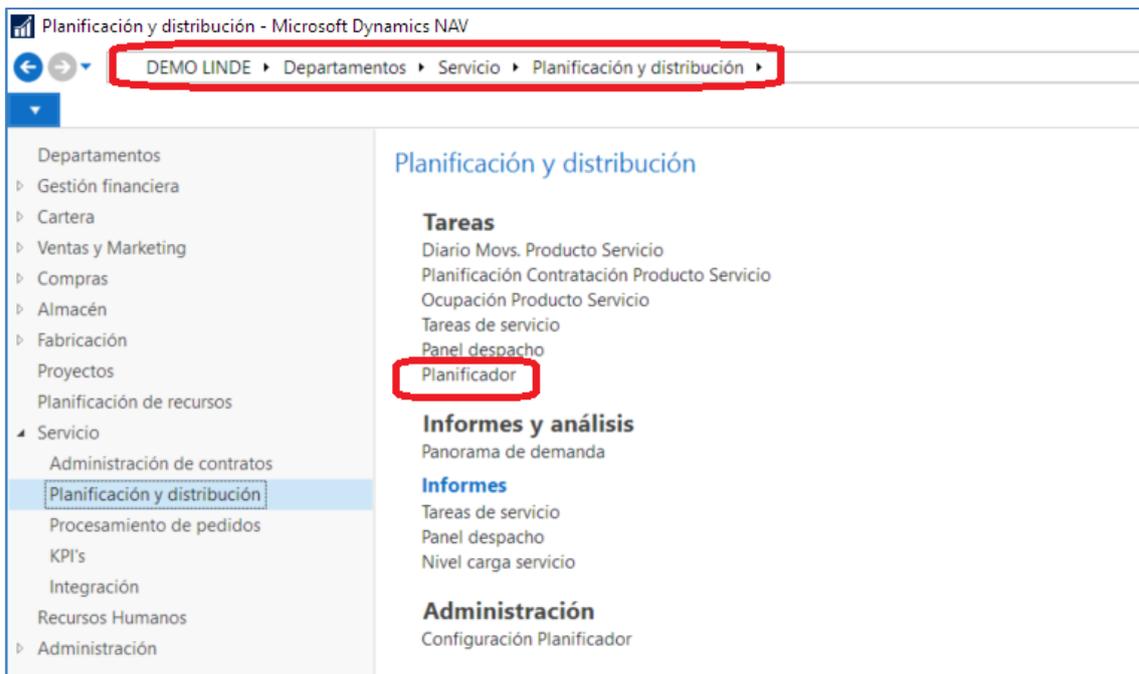
3 Uso del Planificador

Una vez hemos terminado con las configuraciones necesarias para la utilización del planificador, deberemos buscar la pagina donde se encuentra embebido.

Esta pagina podrá ser encontrada en el Area de Trabajo de los usuarios configurados como "Distribuidor de Servicio":

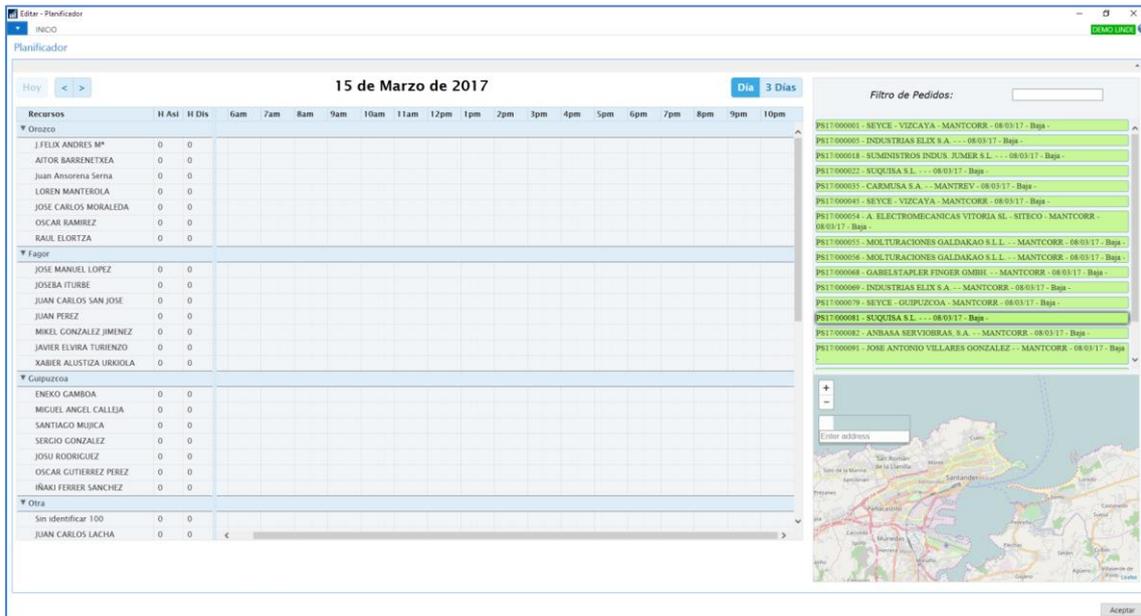


También podrá ser localizada en Departamentos → Servicio → Planificación y Distribución → Planificador.



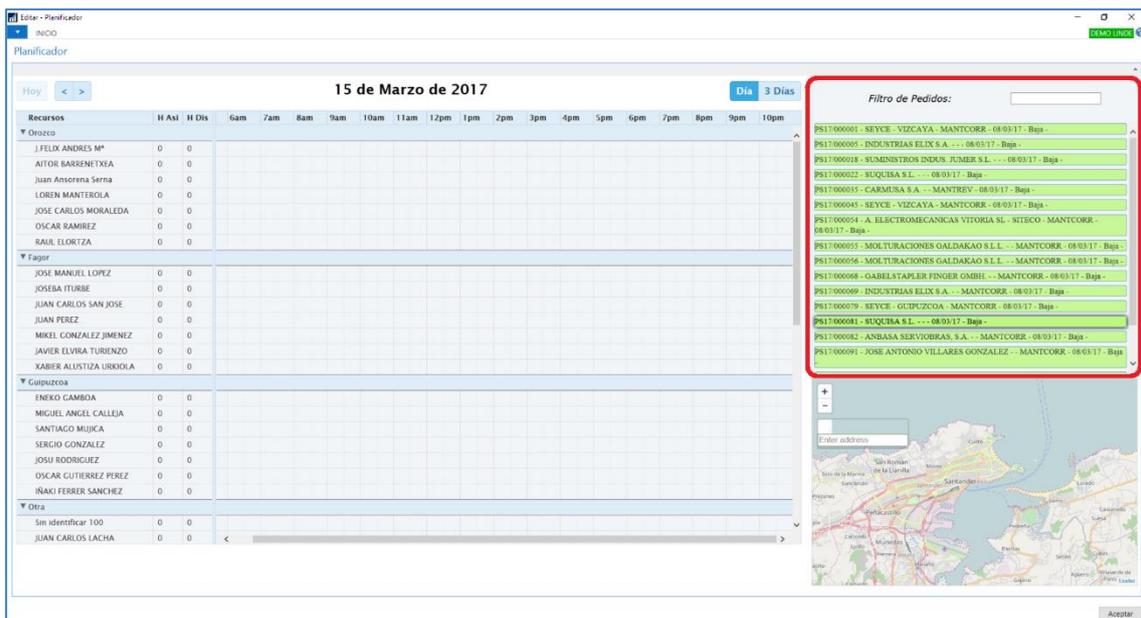
3.1 Apariencia y Elementos

Cuando abramos el planificador, el sistema nos mostrará la siguiente estructura de datos.



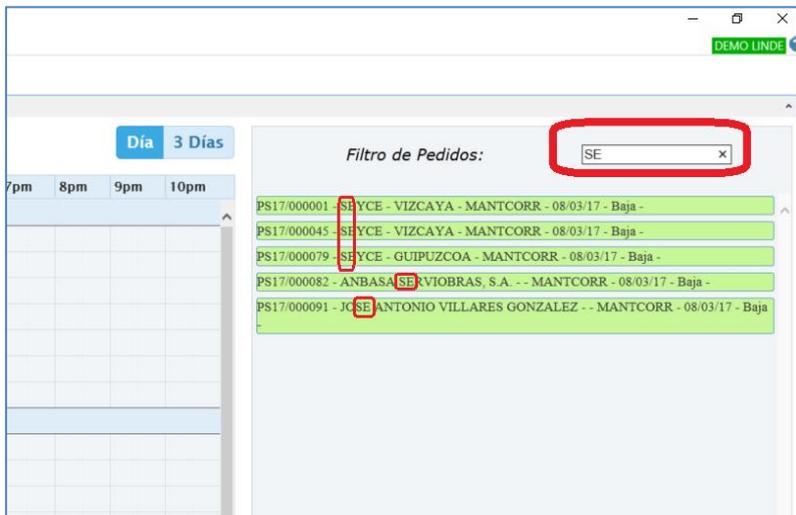
3.1.1 Lista de Pedidos Pendientes de Asignar

En la parte derecha del planificador, lo primero que se nos mostrará serán los Pedidos de Servicio pendientes de asignar. En función de la configuración de colores de los estados de reparación.



En esta lista, podremos hacer uno de la propia barra de desplazamiento para movernos por ella, también podremos filtrar la información utilizando el campo de "Filtro de Pedidos".

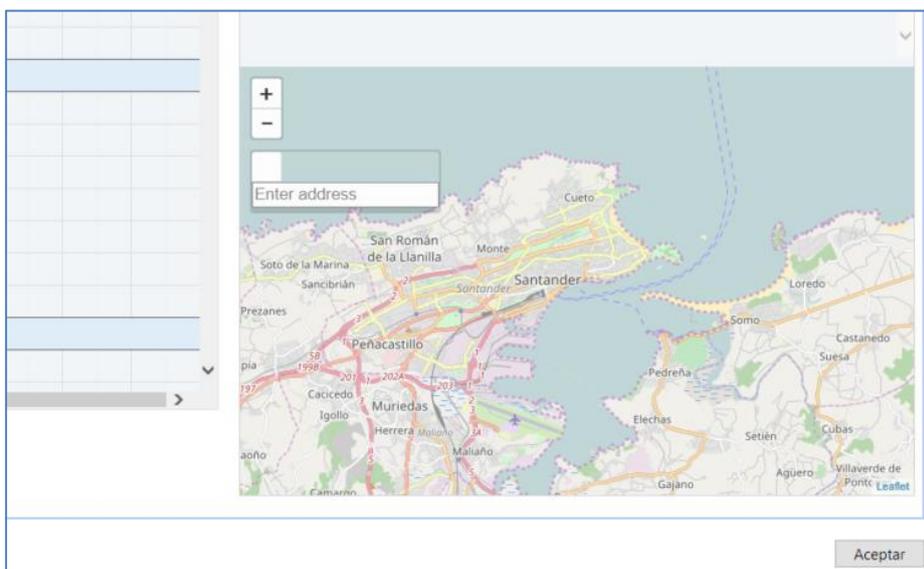
Este campo de filtro buscará, respetando Mayúsculas y Minúsculas, por la información mostrada en la lista de Pedidos de Servicio Pendientes. El filtro se irá aplicando a medida que escribimos.



3.1.2 Mapa

En la parte Inferior derecha del planificador, justo debajo de la lista de Pedidos de Servicio Pendientes, se mostrará un elemento de mapa geográfico.

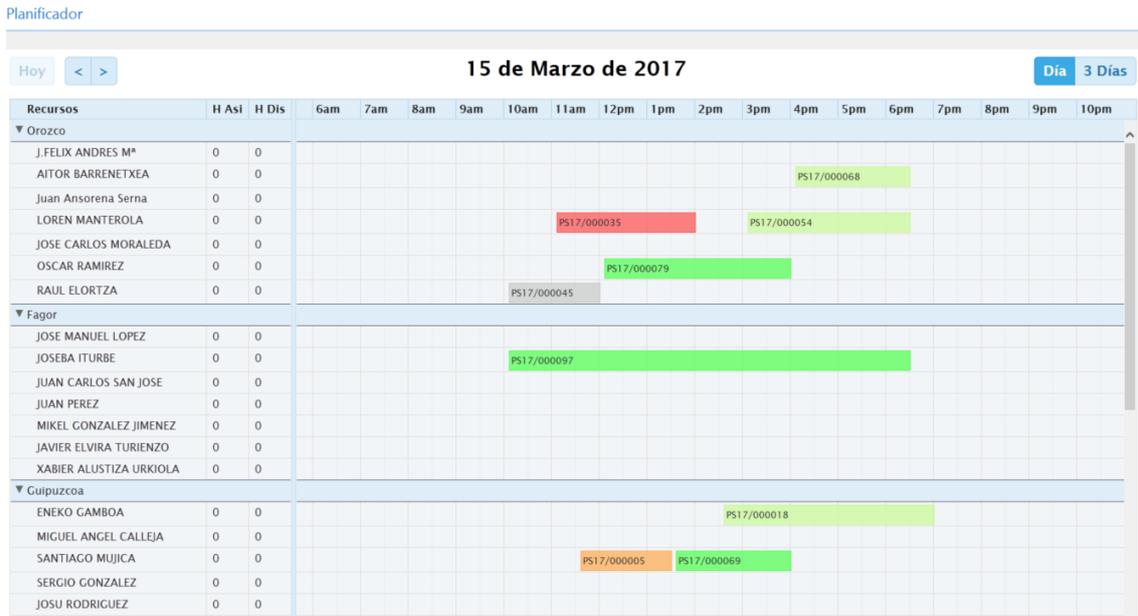
Nada más abrir el planificador, el mapa se nos situará en su ubicación por defecto que se encuentra en Santander. Esto podrá ser modificado en evolutivos posteriores.



Posteriormente, utilizando la funcionalidad “Ubicar en Mapa” que se explicará más adelante, el mapa nos mostrará la ubicación indicada en el Pedido de Servicio.

3.1.3 Área de Trabajo

El área principal del planificador es el que se encuentra en la parte central-izquierda del mismo. Esta área nos mostrará tanto los diferentes recursos, como los Pedidos de Servicio asignados a cada uno.

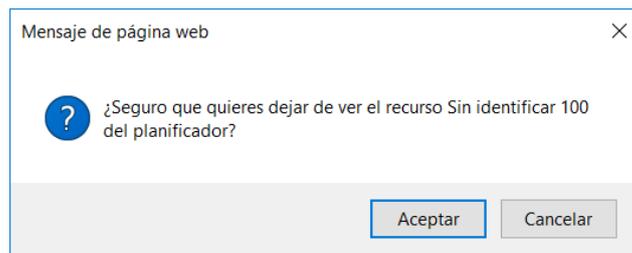


3.1.3.1 Recursos y Familias de Recursos

En la parte izquierda del Área de Trabajo del planificador se muestran los Recursos de tipo “Persona” que no se encuentren “Bloqueados”, agrupados según su “Grupo Familia Recurso” asignado.

Las familias de recursos pueden colapsarse para mostrar únicamente los recursos que deseemos.

De igual manera, si pichamos sobre cualquier recurso, el sistema nos preguntará si queremos dejar de ver dicho recurso. Esta acción solo nos elimina el recurso del planificador, en ningún caso modifica el registro en Dynamics NAV.



Junto a cada recurso, se muestran dos columnas donde aparecerán las horas asignadas y horas disponibles de dicho recurso. Esta funcionalidad no está implementada, ya que se encuentra pendiente de definir la lógica de los datos a mostrar.

3.1.3.2 Pedidos de Servicio

Finalmente, en el área central del área de Trabajo, se nos mostrarán los diferentes Pedidos de Servicio asignados a cada recurso en función de las fechas y horas de cada asignación.



Hay que destacar, que el planificador hace uso de dos campos nuevos aparecidos en Dynamics NAV 2016 y que solo son utilizados por el planificador: Hora Inicio y Hora Fin.

Editar - Asignaciones recurso - PS17/000045

INICIO ACCIONES NAVEGAR

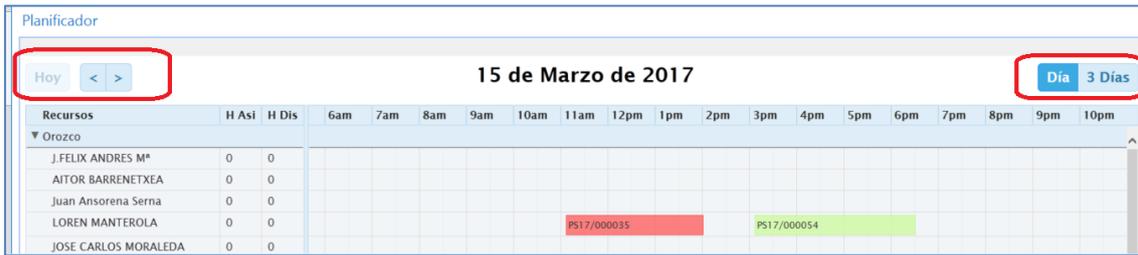
Nuevo Ver lista Editar lista Eliminar Mostrar como lista Mostrar como gráfico OneNote Notas Vínculos Actualizar Borrar filtro Buscar

Asignaciones recurso

Estado	Nº prod. servicio	Descripción prod. servicio	Nº recurso	Fecha asignaci...	Horas asignadas	Res...	Hora inicial	Hora finalización
Activo	000000	MAQUINA SEYCE PARA POST VEN...	135	15/03/2017	2,0	<input checked="" type="checkbox"/>	9:00:00	11:00:00

Por ello, los pedidos de servicio asignados de forma "Tradicional", no completarán estos campos, salvo que el propio usuario lo haga manualmente, y por tanto, dichos pedidos de servicio no podrán ser representados en el Planificador. En estos casos, el usuario deberá manualmente introducir los campos arriba señalados.

En el área de trabajo, se muestra de forma predeterminada el día actual, dividido en columnas por cada franja horaria. Esta vista puede ser modificada a un intervalo de 3 días, utilizando la opción correspondiente disponible en la parte superior derecha del área de trabajo.

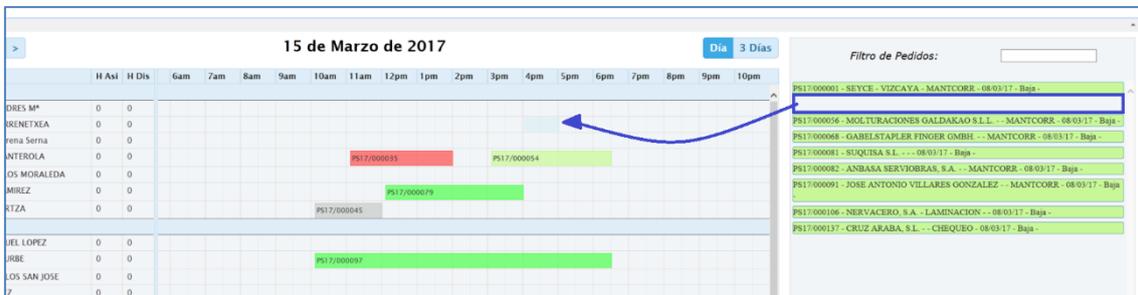


Para movernos entre días o periodo de 3 días, podremos utilizar las flechas que se pueden apreciar en la parte superior izquierda del área de trabajo.

Para regresar rápidamente al día o periodo actual, podremos utilizar la opción “Hoy” situada en la parte superior izquierda.

3.2 Asignación de Pedidos de Servicio

Una vez ya conocemos las diferentes áreas del planificador, podemos pasar a explicar su uso. Para realizar una asignación de un pedido a un técnico bastará con arrastrar el pedido de servicio deseado, desde la lista de pedidos pendientes, al técnico a quien que queramos asignárselo, haciendo coincidir el recuadro con la hora de inicio deseada.



En la imagen anterior, se puede ver una asignación de un PS para que sea iniciado a las 16:00.

Por defecto, el sistema asignará una duración estándar de 1 hora para cada Pedido de Servicio asignado utilizando el planificador. Esto, posteriormente, será configurable por cada concesionario.

3.3 Opciones Adicionales

Por supuesto, el planificador gráfico dispone de algunas opciones adicionales, que poco a poco serán ampliadas. Por ahora, el sistema cuenta con las siguientes:

3.3.1 Ampliación/Reducción tiempo asignación Pedido de Servicio.

Como se ha comentado anteriormente, los pedidos de servicio asignados mediante el planificador, tendrán un tiempo asignado de 1 hora.

Para ampliar o reducir este tiempo, bastará con situar el cursor del ratón sobre el borde del pedido de servicio que queramos modificar, y cuando dicho cursor cambie a la imagen de redimensión (como el estándar de Windows) haremos clic izquierdo y arrastraremos dicho borde.

3.3.2 Cambio de asignación de Pedido de Servicio.

Para reasignar un Pedido de Servicio, bastará con "pinchar" sobre cualquier pedido de servicio, y arrastrarlo hasta el recurso deseado, haciendo siempre coincidir la parte izquierda del recuadro del Pedido de servicio con la hora de inicio deseada.

Esta reasignación, realizará todos los pasos necesarios para indicarle a FIELDEAS-TRANSFER el cambio de asignación en caso de ser necesario.

3.3.3 Tooltip con información adicional

Una vez asignemos el pedido de servicio, en el área de trabajo solo podremos identificar el N° del Pedido de Servicio.

Si queremos mostrar más información, bastará con situar el cursor por encima, lo que nos hará aparecer un "tooltip" o mensaje emergente con dicha información.

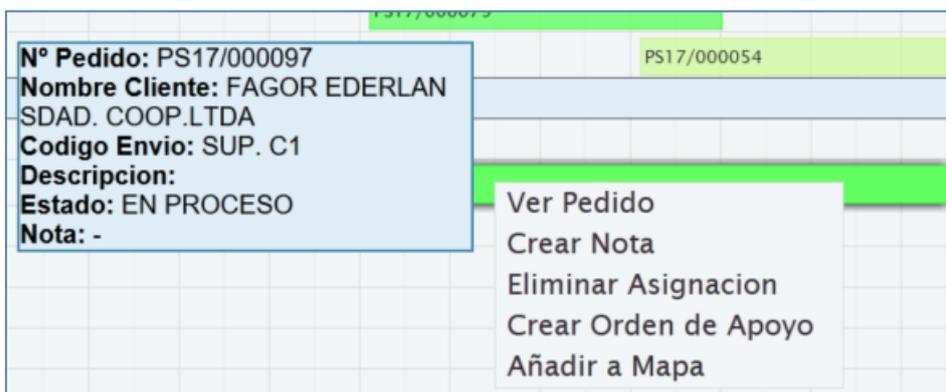


En dicho mensaje podremos encontrar:

- N° de Pedido de Servicio
- Nombre Cliente
- Cód. Dirección Envío Cliente
- Descripción del Pedido de Servicio
- Estado de Reparación
- Nota del Planificador: Esta nota será un mensaje que se podrá completar tanto desde el propio pedido, como desde el planificador. Ver punto 3.3.4.1 Crear Nota

3.3.4 Acciones con Pedido de Servicio desde el planificador

De igual forma, haciendo clic derecho sobre un pedido de servicio asignado, se nos presentarán algunas opciones adicionales.

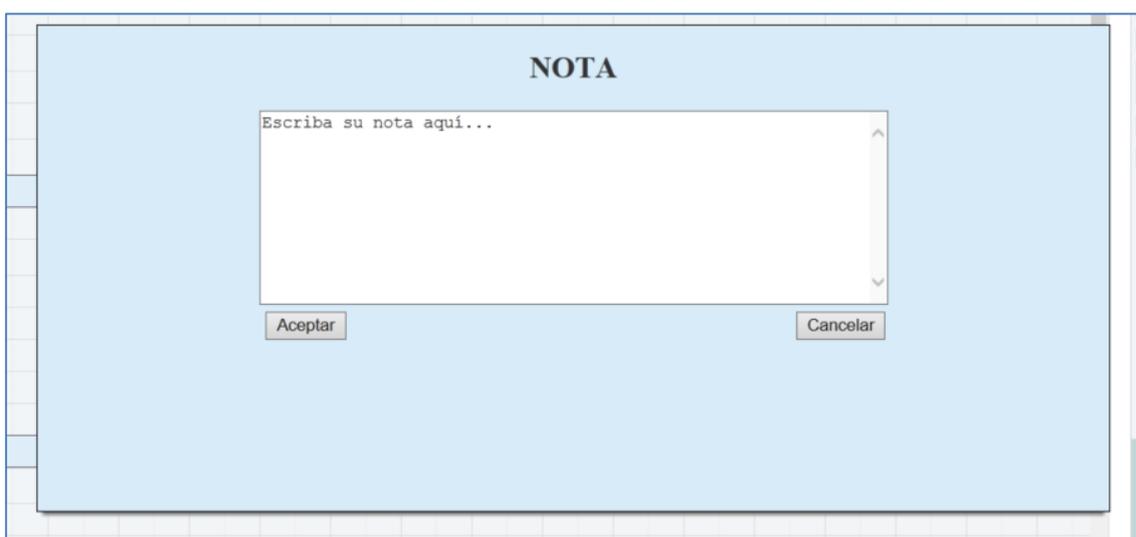


3.3.4.1 Ver pedido

Esta opción, nos abrirá la página del pedido de servicio indicado, donde podremos operar y/o consultar cualquier dato como si proviniésemos de la lista de Pedidos de Servicio o Panel de Tareas de Servicio.

3.3.4.2 Crear Nota

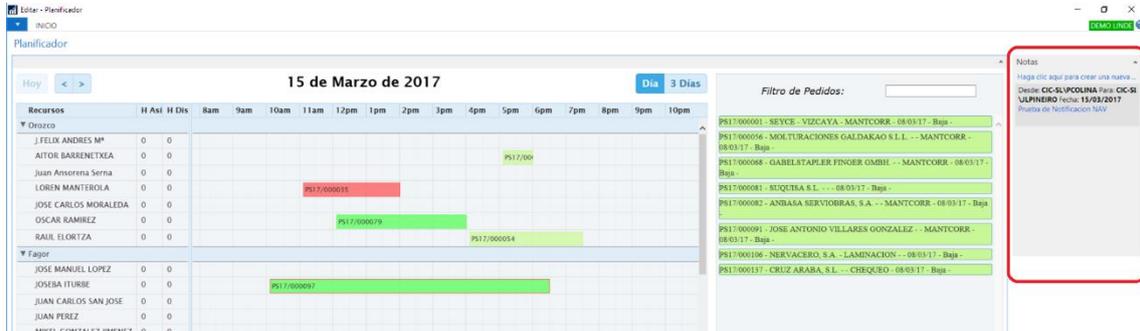
Si seleccionamos crear nota se muestra una ventana emergente en la cual podemos introducir una nota:



Esta nota, será posteriormente visible en el grupo "General" de la ficha del pedido de servicio.

3.3.5 Notificaciones Globales

Como en cualquier página de Dynamics NAV 2016, el usuario podrá utilizar también el sistema de notificaciones estándar ubicado en la página lateral. Esta funcionalidad es completamente estándar.



4 Resultado y Conclusiones

Esta nueva funcionalidad del Planificador Gráfico, añade una opción muy valiosa a cualquier empresa que utilice el Vertical de Maquinaria ALQUINTA sobre Dynamics NAV 2016. Aunque esta funcionalidad se encuentra ya disponible y de modo estable en todos y cada uno de los concesionarios de Linde Material Handling S.A. que dispongan de la versión de Dynamics NAV 2016, este producto seguirá siendo evolucionado para aumentar sus funcionalidades y capacidades.