

Memoria

Trabajo Final de Carrera en Tecnología .NET

Consultor: David Gañan Jiménez

13/01/2014

Alberto Ruiz Domínguez
albertoruizdominguez@uoc.edu



Universitat Oberta
de Catalunya

Índice de contenido

1	Introducción	4
2	Descripción del proyecto	6
2.1	Justificación del proyecto	6
2.1.1	¿Por qué el proyecto?	6
2.1.2	Estudio de mercado	7
2.1.3	Descripción del proyecto	11
2.2	Objetivos del proyecto	11
2.2.1	Generales	11
2.2.2	Específicos	13
2.3	Resumen de funcionalidades a implementar	14
2.4	Resultados esperados	15
2.5	Productos obtenidos	15
2.6	Retorno de la inversión	15
2.7	Planificación inicial vs planificación final	16
3	Análisis	19
3.1	Requerimientos funcionales / no funcionales	19
3.2	Diagrama de casos de uso	26
3.3	Modelo conceptual	28
4	Diseño	30
4.1	Diagrama de arquitectura: de aplicación / SW / HW	30
4.2	Diseño de la BD / Diagrama E-R	36
4.3	Modelo de clases	38
4.4	Diagrama de secuencia	41
4.5	Diagrama de estado	41
4.6	Diseño de la interfaz de usuario	42
4.6.1	Apariencia y estilo (look & feel)	42
4.6.2	Estructura de la vista	42
4.6.3	Navegación	43
4.6.4	Jerarquía de vistas	44
4.6.5	Descripción de vistas	45
5	Implementación	51
5.1	Software utilizado	51
5.2	Capas de la aplicación	53
6	Evaluación de costes	55
7	Trabajo futuro	56
7.1	Evolución funcional	56
7.2	Evolución tecnológica	58
8	Conclusiones	60
9	Bibliografía	62

Índice de ilustraciones

Ilustración 1 - Diagrama de Gantt	18
Ilustración 2 - Diagrama de casos de uso	26
Ilustración 3- Modelo conceptual	28
Ilustración 4 - Concepto cliente-servidor	30
Ilustración 5 - Diagrama conceptual del modelo lógico de aplicación multicapa	31
Ilustración 6 - Diagrama de capas de la solución	33
Ilustración 7 - Diagrama de componentes de la solución	34
Ilustración 8 - Diagrama de implementación	35
Ilustración 9 - Diagrama E-R Liquidación de gastos.....	36
Ilustración 10 - Diagrama de clases de entidades	38
Ilustración 11 - Diagrama de clases de repositorios de acceso a la base de datos.....	39
Ilustración 12 - Diagrama de clases de repositorios de acceso a Servicios Web	40
Ilustración 13 - Diagrama de clases de la interfaz de usuario (View-ViewModel).....	41
Ilustración 14 - Diagrama de estado para para la liquidación.....	41
Ilustración 15 - Esquema general de diseño de una vista	43
Ilustración 16 - Detalle de jerarquía de vistas de la aplicación	44
Ilustración 17 - Detalle de la vista Login.....	45
Ilustración 18 - Detalle de la vista Lista de liquidaciones	45
Ilustración 19 - Detalle de la vista Lista de autorizaciones.....	46
Ilustración 20 - Detalle de la vista Liquidación	46
Ilustración 21 - Detalle de la vista Gasto	47
Ilustración 22 - Detalle de la vista Gasto (Comprobante)	47
Ilustración 23 - Detalle de la vista Lista de departamentos	48
Ilustración 24 - Detalle de la vista Departamento.....	48
Ilustración 25 - Detalle de la vista Lista de usuarios.....	49
Ilustración 26 - Detalle de la vista Usuario	49
Ilustración 27 - Detalle de la vista Gestión de calendarios	50
Ilustración 28 - Diagrama del modelo físico.....	53

Índice de tablas

Tabla 1 - Resumen de valoración de la alternativa ERP	7
Tabla 2 - Resumen de valoración de la alternativa Ofimática.....	8
Tabla 3 - Resumen de valoración de la alternativa Desarrollo a medida.....	9
Tabla 4 – Resumen de comparación entre soluciones <i>software</i>	10
Tabla 5 - Costes del procedimiento actual de liquidación de gastos	15
Tabla 6 - Costes del nuevo procedimiento de liquidación de gastos	16
Tabla 7- Descomposición Estructural de Actividades (WBS).....	17
Tabla 8 - Duración estimada de actividades.....	17
Tabla 9- Tabla de espacio de nombres.....	32
Tabla 10 - Tabla de descripción de tablas de base la base de datos.....	37
Tabla 11 - Precios de perfiles.....	55
Tabla 12 - Presupuesto de desarrollo del proyecto <i>software</i> de liquidación de gastos	55

1 Introducción

Dotnet Consulting, S.A. es una empresa de ámbito internacional con sede en Madrid (España) que presta servicios de Auditoría y Consultoría Financiera por todo el mundo. Su plantilla actual está compuesta por aproximadamente 5.000 empleados, donde el 20% es personal de estructura y el 80% restante está formado por personal operativo que ofrece servicio a sus clientes.

La empresa realiza al año 152.000 liquidaciones de gastos derivadas de su actividad mercantil, cuyo importe total asciende a 6.000.000 €, lo que supone una media diaria de 690 liquidaciones con un gasto medio de 40 €.

Recientemente Dotnet Consulting ha encargado un estudio para la implantación de un *software* que les permita realizar la liquidación de gastos de todos sus empleados, tanto los que prestan servicio dentro como fuera de la organización. Los objetivos principales que la empresa pretende conseguir con la implantación de este *software* son los siguientes:

- Trasladar el abono de los gastos de liquidación de la caja a la nómina mensual.
- Reducir los costes (directos e indirectos) del procedimiento de liquidación.
- Reducir del tiempo dedicado por todos los intervinientes en el proceso.
- Reducir el número de errores u omisiones de datos, autorizaciones y documentos requeridos en el proceso.

En la actualidad, los gastos se liquidan mediante el siguiente procedimiento que se realiza de forma manual:

1. Un empleado rellena una hoja de liquidación donde detalla todos los gastos que le serán reembolsados en el momento de la presentación de los gastos.
2. El empleado debe adjuntar a la hoja de liquidación todos los tiques, facturas y/o resguardos que la normativa de la compañía establece según el tipo de gasto.
3. El empleado debe enviar la hoja de liquidación junto con todos los justificantes al responsable de su departamento para que autorice la operación de liquidación de gastos.
4. En caso de que el responsable de departamento NO autorice la liquidación, este informa verbalmente al empleado solicitante de los motivos y le devuelve la hoja para que la rectifique. Una vez que el empleado recibe la hoja y los motivos de rechazo debe comenzar el procedimiento de nuevo volviendo al paso 1 o 2.

5. En caso de que el responsable de departamento SI autorice la liquidación, debe firmar la hoja y devolverla al empleado junto con todos los justificantes requeridos que previamente recibió.
6. El empleado debe dirigirse personalmente a la caja de la compañía para presentar la hoja de liquidación autorizada.
7. El personal de caja recibe la documentación, revisa que es correcta, que no falta ninguno de los comprobantes de pago requeridos y que la hoja ha sido autorizada por el responsable del departamento de la persona que solicita la liquidación de gastos.
8. Si alguna de las validaciones anteriores no es correcta, se le informa al empleado de la incidencia detectada y se le devuelve toda la documentación para que pueda subsanar el problema. El empleado debe comenzar el procedimiento de nuevo volviendo al paso 1 o 2.
9. Si todas las validaciones son correctas, el personal de ventanilla archiva toda la documentación y abona en efectivo al empleado el montante total de la hoja de liquidación.

Los gastos susceptibles de ser liquidados son: billetes de tren, avión, barco o autobús; alquiler de coche, taxi, gastos de representación, formación, combustible, uso y alquiler de teléfono, hotel, kilometraje en vehículo propio, dietas diarias, aparcamiento, peaje, portes, u otros gastos.

Los perfiles de empleados que realizan gastos de liquidación son: comerciales, jefes de proyectos, consultores, directivos, etc.

2 Descripción del proyecto

2.1 Justificación del proyecto

2.1.1 ¿Por qué el proyecto?

Dotnet Consulting ha identificado que su procedimiento de liquidación de gastos de personal es excesivamente caro e ineficiente. Asimismo, la compañía ha advertido en él errores y omisiones que provocan un sobre coste adicional. La siguiente es una lista de algunos de los problemas e inconvenientes que se producen diariamente:

- En la gestión de cada tipo de gasto existen diferentes criterios de validación: en los restaurantes se exige una factura, en el taxi un recibo con NIF, en el metro el título de viaje. Las diferentes tipologías inducen, en ocasiones, a error al personal administrativo que debe gestionarlo.
- El personal que trabaja fuera de la oficina por largos periodos de tiempo suele tener problemas para liquidar sus gastos hasta que no regresa a la oficina del viaje. Esta situación se agrava cuando los viajes son por periodos superiores a un mes y los gastos liquidables comienzan a sumar una cantidad significativa.
- Las autorizaciones del pago de liquidaciones suponen un problema para los autorizadores por el esfuerzo y tiempo que requiere su verificación.
- El departamento de administración que gestiona las liquidaciones se encuentra en ocasiones desbordado por el tiempo dedicado a esta actividad cuando coinciden un número elevado de ellas, como por ejemplo en el cierre de ejercicios fiscales.
- La gestión manual y la cantidad de papel que se emplea durante todo el procedimiento supone un coste del tiempo del personal administrativo y de materiales que podría reducirse mediante la tecnología y la automatización.

Por otro lado, la compañía encargó un estudio de mercado con el objetivo de conocer las posibles alternativas para sustituir el procedimiento actual mediante la aplicación de las tecnologías de la información. En este estudio se confirmó que existían opciones de mejora asequibles para la empresa.

En conclusión, el cliente desea normalizar el procedimiento de liquidación y utilizar un *software* que proporcione soporte a esta actividad haciéndola más eficiente y beneficiosa para la empresa y sus empleados.

2.1.2 Estudio de mercado

En el estudio de mercado encargado por Dotnet Consulting se analizaron las siguientes propuestas, que a continuación se exponen de forma resumida:

- ERP (Enterprise Resource Planning)
- Ofimática
- Desarrollo a medida

ERP (Enterprise Resource Planning)

Entra las distintas soluciones de mercado, el ERP de mayor penetración entre las grandes empresas españolas es SAP (SAP, 2013).

La solución de SAP para la liquidación de gastos (SAP Gastos, 2013) requiere la implantación y parametrización de los módulos básicos del ERP que interactúan con la Liquidación de gastos de viaje, estos son:

- Cálculo de nómina (HR) para el reembolso de los importes de gastos de desplazamiento al empleado.
- Gestión Financiera (FI)
- Business Workflow (BW)
- Adicionalmente el módulo Employee Self-Service (ESS) permite realizar la entrada descentralizada de los gastos de viajes por los mismos viajeros.

El siguiente cuadro, resume la valoración de esta alternativa:

Tabla 1 - Resumen de valoración de la alternativa ERP

Variable	Valor
Coste de la solución:	Precio a medida bajo solicitud en función del volumen de facturación de la empresa. Existe una configuración para compañías con facturación inferior a 20.000.000,00 de euros, que supone un coste mínimo aproximado de 300.000,00 € en licencias más coste de implantación del equipo de consultoría.
Periodo de implantación (mínimo):	Más de 6 meses
Ventajas:	<ul style="list-style-type: none"> • Solución integrada que permite recapitalizar la inversión anterior en esta tecnología con la implantación de nuevos módulos, ya que existe una interconexión entre los datos de los diferentes módulos que imprime mayor productividad a la solución cuanto mayor es su integración y más áreas empresariales están cubiertas.

	<ul style="list-style-type: none"> • Modelos funcionales estándar predefinidos y ajustados a las normativas de cada país. • Multi-moneda. • Escalable vertical y horizontalmente. • Ofrece soporte técnico y funcional (a través de compañía integradora). • Integración con el sistema actual de nóminas mediante un pequeño desarrollo adicional.
Inconvenientes:	<ul style="list-style-type: none"> • No se puede trabajar en modo desconectado. • Duración de implantación muy grande. • Coste de licencia elevado. • Coste de mantenimiento elevado. • Coste de escalado elevado (en la mayoría de los casos es mayor que lineal). • Si la primera instalación no incluye las áreas críticas financieras la solución pierde mucho valor y la relación coste licencia / funcionalidad se incrementa exponencialmente. • Tecnología propietaria, aunque admite integración con desarrollos .NET.

Ofimática

La solución ofimática que Dotnet Consulting tiene homologada a nivel corporativo para todos sus empleados es la *suite* Microsoft Office 2010. Aprovechando la versatilidad de la hoja de cálculo de este paquete, es factible plantear una solución basada en plantillas de dicha hoja de cálculo para rellenar los datos de liquidación e incluso adjuntar las imágenes dentro la propia hoja.

Existen plantillas ya creadas como la que propone la empresa Cash Trainers para la liquidación de gastos de viajes con Excel (Cash Trainers, 2013). El siguiente cuadro, resume la valoración de esta alternativa:

Tabla 2 - Resumen de valoración de la alternativa Ofimática

Variable	Valor
Coste de la solución:	Menos de 150,00 €, sin tener en cuenta costes de distribución por estación de trabajo que puede ser automatizado con un coste inferior a 1.000,00 €.
Periodo de implantación (mínimo):	1 mes
Ventajas:	<ul style="list-style-type: none"> • Solución que capitaliza las herramientas ofimáticas existentes de la compañía (aprovecha los activos de licencias actuales). • La hoja de cálculo es ampliamente conocida por los

	<p>empleados de la compañía. La curva de tiempo de aprendizaje es prácticamente nula.</p> <ul style="list-style-type: none"> • Se puede trabajar en modo desconectado. • Solo soporte técnico. • Duración de la implantación muy corta. • Coste muy bajo.
Inconvenientes:	<ul style="list-style-type: none"> • No tiene en cuenta el aspecto multi-moneda, debería desarrollarse un complemento adicional que dificultaría su distribución y mantenimiento. • No guarda correlación entre los usuarios. • No se asegura el almacenamiento unificado de todos los documentos. • No permite consultar información histórica ni relacionarla entre los usuarios, ubicaciones u otros criterios. • No escala de ningún modo. • No permite la explotación conjunta de todos los datos. • No se integra con el sistema actual de nóminas.

Desarrollo a medida

El proyecto de desarrollo a medida del *software* de liquidación de gastos presentado en este documento, es una alternativa adicional a las ya presentadas que cubre el 100% de los requisitos expresados por el cliente. Su realización se proyecta sobre tecnologías de la plataforma .NET que ya disponen de los elementos necesarios y probados para generar un producto con todas las garantías.

El siguiente cuadro, resume la valoración de esta alternativa:

Tabla 3 - Resumen de valoración de la alternativa Desarrollo a medida

Variable	Valor
Coste de la solución:	12.000,00 € (sin incluir costes de mantenimiento ni explotación).
Periodo de implantación (mínimo):	4 mes
Ventajas:	<ul style="list-style-type: none"> • <i>Software</i> adaptado al 100% a las necesidades funcionales y técnicas de la compañía y de la problemática exigida. • Propiedad intelectual del desarrollo es de la compañía. • Integración personalizada el sistema actual de nóminas. • Se puede trabajar en modo desconectado. • Ofrece soporte técnico y funcional. • Duración del desarrollo e implantación corto.

	<ul style="list-style-type: none"> • Multi-moneda • Consolidación de la información • Explotación futura de los datos históricos. • Coste bajo a moderado si se incluyen costes estimados de explotación y mantenimiento. • Escalable vertical y horizontalmente.
Inconvenientes:	<ul style="list-style-type: none"> • Es un producto a medida y no goza de la evolución y nivel de madurez que puede tener un producto estándar. • El mantenimiento debe ser asumido por la compañía.

Conclusiones

La siguiente tabla muestra el resumen de las características de cada una de las soluciones estudiadas y ofrece la base de comparación para determinar las conclusiones de acuerdo con los requisitos y necesidades expresadas por el cliente.

Tabla 4 – Resumen de comparación entre soluciones *software*

Característica	ERP	Ofimática	Desarrollo
Coste	Elevado	Muy asequible	Asequible
Periodo implantación	> 6 meses	1 mes	< 4 meses
Cobertura de los requisitos	> 80%	25%	100%
Cumple con normativas estándar	Sí	No	No (pero existe opción de adaptación si se requiere)
Multi-moneda	Sí	No	Sí
Escalable vertical y horizontalmente	Sí	No	Sí
Soporte técnico	Sí	Sí	Sí
Soporte funcional	Sí (a través de integrador)	No	Sí
Integración con sistema de nóminas	Sí (con desarrollo adicional)	No	Sí
Modo desconectado	No	Sí	Sí
Almacenamiento de datos y <i>backup</i>	Sí	No	Sí
Correlación automática con usuarios	Sí	No	Sí
Vista de estados de liquidación	Sí	No	Sí
Explotación multidimensional	Sí	No	Sí
Control de evolución del <i>software</i>	No	Sí	Sí

En conclusión, la opción del proyecto de **Desarrollo a medida** parece la más acertada y que presenta mejor equilibrio desde los puntos de vista funcional, económico, temporal y técnico, así como la mejor integración y horizonte de futuro para los planes tecnológicos de la compañía.

2.1.3 Descripción del proyecto

Marco de referencia

El proyecto de Liquidación de Gastos para Dotnet Consulting es una iniciativa de construcción de *software*, que tiene por objetivo principal la mejora de la eficiencia del citado proceso.

El presente documento establece una estimación de costes y planificación temporal del proyecto, a partir de los datos entregados por la compañía en septiembre de 2013 como base de estudio. En este documento también se definen los objetivos completos, el alcance, la funcionalidad y otros aspectos del proyecto que forman el marco de referencia de este documento.

Plazos establecidos

La planificación de este proyecto, así como la estimación del esfuerzo, quedan recogidas en los plazos establecidos, de modo que se tomará, como fecha inicial del proyecto, el 19-09-2013, y se prevé que el nuevo sistema pueda entrar en fase de explotación el 13-01-2014.

Ciclo de vida y herramientas para la estimación de costes y planificación

Para conseguir una estimación ajustada del volumen de trabajo y una planificación lo más realista posible, se ha considerado conveniente utilizar el ciclo de vida en cascada con prototipado con objeto de desarrollar el nuevo sistema. La utilización de este ciclo de vida, la capacidad de análisis de nuestro equipo y la experiencia en las herramientas utilizadas permiten que las estimaciones sean muy ajustadas a la realidad. Para la estimación de costes y la planificación se utilizará Microsoft Project 2010.

2.2 Objetivos del proyecto

2.2.1 Generales

Trasladar el abono de los gastos de liquidación de la caja a la nómina mensual

El pago de los gastos de liquidación en nómina a los empleados supone algunas ventajas para la empresa respecto al pago en caja:

1. El pago a través de nómina supone un abono mediante transferencia bancaria de la compañía a sus empleados en unas condiciones económicas muy ventajosas,

pactadas previamente con la entidad bancaria que realiza las transferencias. Adicionalmente el riesgo por robo es prácticamente nulo, y se reduce la gestión diaria del efectivo para el pago en caja que se precisa para hacer frente a las cantidades requeridas, así como la contabilización de los distintos flujos de forma manual, lo que añade un coste adicional.

2. Los flujos salientes de caja por el pago de liquidaciones de gastos se ven retrasados a final de mes en el caso del pago por nómina, mientras en el pago por caja estos flujos se producen en el momento de la liquidación. En el primer caso la empresa obtiene un beneficio financiero frente al segundo caso.

Reducir los costes (directos e indirectos) del procedimiento de liquidación

Los costes directos son aquellos que se atribuyen de forma directa al objetivo del coste, en este caso a la liquidación de gastos, mientras que los costes indirectos no tienen una relación directa con el objetivo atribuible y son prorrateados mediante algún método entre diferentes objetivos. De forma somera se identifican los siguientes:

- Costes directos: Los importes de las liquidaciones y del coste del personal dedicado a esta actividad si su única función es la liquidación de costes.
- Costes indirectos: La gestión del efectivo, el coste del papel, del equipo informático, de los suministros, de la custodia y archivo de la documentación, de los servicios en envío y recepción de documentación, del tiempo dedicado por los recursos en horas laborables a realizar la liquidación, etc.

La cuantificación de todos y cada uno de estos costes es una actividad muy complicada de llevar a cabo para ser realizada a la perfección. Sin embargo, se puede hacer una aproximación por las partidas de costes más elevadas para tener una orientación de los costes reales.

Manteniendo fijos los importes de las liquidaciones (que podrían ser mejorados mediante políticas de gastos) podemos tomar como gastos directos el coste de los tres empleados que la compañía tiene asignados a tiempo completo para esta actividad. El coste total es de 135.000,00 € anuales.

Para los costes indirectos tomaremos la partida anual destinada a guarda y custodia de los documentos de liquidación que asciende a 38.000,00 €, más otros gastos estructurales que suman la cantidad anual de 3.570,00 €.

Finalmente obtenemos un montante final de 176.570,00 € por los costes derivados de la liquidación de gastos. Si al año se producen 152.000 liquidaciones podemos obtener una ratio por liquidación de 1,16, es decir, cada liquidación cuesta 1,16 €. El objetivo de este punto será por tanto reducir la ratio haciendo que tienda a 0.

Reducir del tiempo dedicado por todos los intervinientes en el proceso

Derivado del punto anterior y considerando un coste indirecto la dedicación del resto de personal de la organización: empleados que liquidan, autorizadores, personal de mensajería, etc.; la reducción del tiempo dedicado a este proceso supone un ahorro significativo para la compañía, o al menos, la disponibilidad de mayor tiempo por parte de los intervinientes para realizar otras actividades.

2.2.2 Específicos

Reducir el número de errores u omisiones de datos

Uno de los condicionantes en el proceso de liquidación de gastos que aumentan el tiempo de los intervinientes y el coste general, es el derivado de los errores humanos producidos. Por ejemplo, no rellenar adecuadamente campos del formulario, equivocarse en el departamento perteneciente, presentar la liquidación sin la correspondiente firma del autorizador, omitir o perder los comprobantes de pago, son solo algunas de las incidencias más habituales. Sin embargo, en todos los casos y después de evidenciar el problema hay que volver a repetir el ciclo, multiplicando la dedicación de los intervinientes.

Las solución *software*, debe ayudar a reducir los errores detectando de forma prematura todos las incidencias posibles (al menos las más habituales) e informando a los usuarios para que actúen en consecuencia.

Realizar liquidaciones en remoto

Actualmente los empleados deben dirigirse a la ventanilla de la empresa para poder realizar su liquidación de gastos. Sin embargo, la ventanilla solo funciona durante un horario determinado, al igual que el personal de mensajería que trabaja en horario de oficina, o incluso peor, el personal autorizador puede encontrarse también de viaje, por lo que el tratamiento de la liquidación se va demorando entre cada paso del procedimiento hasta que confluyen en el tiempo y lugar los intervinientes requeridos.

Esta situación se complica especialmente por el hecho de que el 80% de la plantilla viaja con mucha frecuencia.

Por lo tanto, asistida por Internet, la aplicación *software* debe permitir la comunicación desde cualquier punto en el mundo donde el usuario pueda conectarse a este medio.

Eliminar las limitaciones horarias para realizar la liquidación de gastos

Al plantear una solución de funcionamiento de ámbito mundial, se presenta otro inconveniente en el tratamiento de las liquidaciones derivado de la coincidencia horaria entre los intervinientes.

Para resolver esta problemática, la solución *software* debe permitir la comunicación asíncrona o en diferido entre los intervinientes, guardando las informaciones y acciones tomadas en cada paso del procedimiento hasta que los usuarios puedan continuar con el proceso.

Catalogar y consolidar la información de los gastos producidos

Una de las consecuencias positivas que tiene para la empresa la implantación de un sistema *software* para la gestión de liquidaciones es que la información de todas las transacciones permanece almacenada y se encuentra en formato digital. Estas circunstancias propician un escenario ideal para aplicar técnicas de análisis de datos que permitirá a la empresa en el futuro poder explotar la información almacenada, que permitirá descubrir entre otras informaciones: tendencias de gastos, evaluar el promedio de los gastos liquidables, diferencias de costes por zonas geográficas, etc.

2.3 Resumen de funcionalidades a implementar

1. Definición departamental y carga de los datos de los usuarios de la organización y sus roles en la aplicación.
2. Identificación de usuario en los procesos relacionados con la liquidación de gastos.
3. Captura, consulta y modificación (cuando sea permitido) de los datos relacionados de liquidación de gastos, tanto en modo conectado (al servidor de datos) como desconectado.
4. Seguimiento del estado cada una de las liquidaciones de gastos y notificación al usuario vía interfaz y correo electrónico de las acciones le son requeridas o que le afectan.
5. Validación de los datos e informaciones, de acuerdo con la normativa empresarial y con los requisitos funcionales detallados, que ayuden al usuario a clasificar cada tipo de gasto y requerirle los datos asociados al citado gasto.
6. Gestionar los calendarios laborales y establecer los importes establecidos para dietas entre otros parámetros del sistema.

7. Abono masivo de liquidaciones mediante registro en nómina de los importes asociados a las liquidaciones. Esta funcionalidad la completa el programa de nóminas, por lo que el alcance del presente proyecto solo incluye la integración con el *software* citado vía fichero de texto.
8. Proporcionar un historial y seguimiento de las liquidaciones tanto a los usuarios como a la organización.

2.4 Resultados esperados

Los requisitos expresados en este documento están dirigidos al cumplimiento de los resultados esperados de este proyecto en todos los ámbitos: operativo, económico, funcional, etc.

La verificación de su cumplimiento será medida a través de los resultados de las pruebas de aceptación de usuario, que serán contrastados con los requisitos tanto funcionales como no funcionales para determinar el grado de adecuación y calidad del *software* entregado.

2.5 Productos obtenidos

La ejecución del proyecto generará los siguientes productos o entregables:

- Aplicación de liquidación de gastos
- Software de instalación
- Manual de usuario

2.6 Retorno de la inversión

La empresa soporta en la actualidad los siguientes costes anuales derivados del procedimiento de liquidación de gastos:

Tabla 5 - Costes del procedimiento actual de liquidación de gastos

Concepto	Importe
Tres recursos humanos destinados al 100% a la tramitación y liquidación de gastos, incluyendo la gestión diaria de efectivo y el tratamiento documental. Coste empresarial por recurso: 45.000,00 €.	135.000,00 €
152.000 liquidaciones anuales con un coste por guarda y custodia de expediente (hoja de liquidación más comprobantes de gastos) de 0,25 €.	38.000,00 €

Concepto	Importe
Otros gastos fijos derivados de la liquidación de gastos.	3.570,00 €
Total	176.570,00 €

La inversión requerida para sustituir el procedimiento actual por el nuevo más los costes de mantener el sistema funcionando el primer año supone las siguientes partidas:

Tabla 6 - Costes del nuevo procedimiento de liquidación de gastos

Concepto	Importe
Desarrollo del <i>software</i>	11.988,00 €
Costes de infraestructura de red, equipos informáticos, licencias de productos y consumos relacionados (1 año).	20.000,00 €
Costes de administración de la nueva aplicación en modalidad de costes compartidos (1 año).	4.250,00 €
Total	36.238,00 €

Por lo tanto el cálculo del ROI con un plazo de retorno de un año se realiza según la siguiente fórmula:

$$ROI = \frac{\text{Retorno de la inversión} - \text{Inversión inicial}}{\text{Inversión inicial} \times 100} = \frac{(176.570 - 36.238) - 36.238}{36.238 \times 100} = 287,25 \%$$

2.7 Planificación inicial vs planificación final

La siguiente tabla describe cada una de las actividades y su descomposición (WBS), con la asignación del perfil de proyecto:

Tabla 7- Descomposición Estructural de Actividades (WBS)

Código	Actividad	Perfil
01	Inicio de proyecto	
02	Plan de trabajo	Jefe de proyecto
03	Análisis	
03.01	Análisis funcional	Analista
04	Diseño	
04.01	Diseño técnico	Analista programador
04.02	Prototipo	Analista programador
05	Implementación	
05.01	Implementación cliente	Analista programador
05.02	Implementación servidor	Analista programador
06	Pruebas	
06.01	Pruebas de integración	Analista programador
06.02	Pruebas de sistema	Analista
07	Documentación	
07.01	Manual de instalación	Analista programador
07.02	Manual de usuario	Analista
07.03	Memoria	Jefe de proyecto
07.04	Presentación Virtual	Jefe de proyecto
08	Final de proyecto	

Estimación de tiempos inicial

El proyecto se estimó con una dedicación continuada de tres horas por jornada de lunes a domingo y no se consideraban los días festivos como tales:

Tabla 8 - Duración estimada de actividades

Actividad	Duración (días)	Inicio	Finalización
Plan de trabajo	12	19/09/2013	30/09/2013
Análisis	8	01/10/2013	08/10/2013
Diseño	8	09/10/2013	16/10/2013
Prototipo	12	17/10/2013	28/10/2013
Implementación de aplicación cliente	12	29/10/2013	10/11/2013
Implementación de aplicación servidora	15	11/11/2013	25/11/2013
Pruebas de integración	6	26/11/2013	01/12/2013
Pruebas de sistema	8	02/12/2013	09/12/2013
Manual de instalación	2	10/12/2013	11/12/2013
Manual de usuario	5	12/12/2013	16/12/2013
Memoria	15	17/12/2013	17/12/2013
Presentación virtual	13	01/01/2014	13/01/2014

Diagrama de Gantt inicial

El siguiente diagrama muestra la estructura temporal del proyecto así como la asignación de perfiles a cada actividad detallada anteriormente:

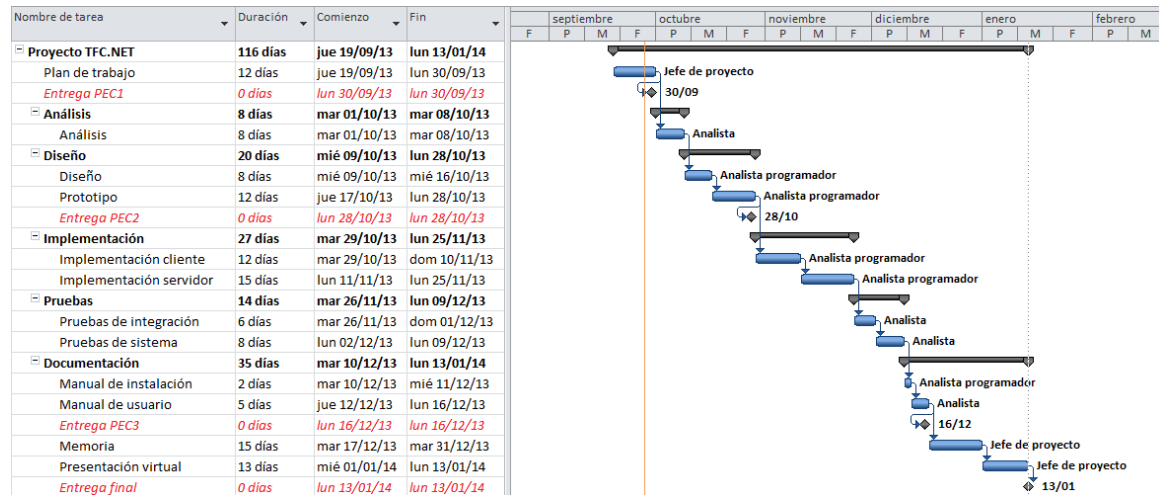


Ilustración 1 - Diagrama de Gantt

Desviación respecto a la planificación inicial

Se han producido desviaciones especialmente relevantes en la implementación de la aplicación y cliente y servidora. Las razones fundamentales han sido las siguientes:

- Dedicación excesiva a labores de investigación y desarrollo de componentes para la capa de presentación (WPF) que permitirán aplicar el patrón MVVM sin pérdida de funcionalidad.
- Dedicación excesiva de investigación y prototipado del algoritmo de sincronización de datos entre cliente y servidor. Entre otros problemas encontrados, se detalla el que provocó la adaptación de las entidades POCO y su adaptación al diseño con una considerable carga de tiempo:

La utilización de Entity Framework (EF), al menos hasta la versión 6.0.2 que es la utilizada en el proyecto, muestra algunas incompatibilidades cuando se utiliza junto con WCF para la serialización/deserialización en entidades POCO (Plain Old CLR Objects). Entre ellas las entidades POCO no pueden presentar propiedades marcadas con la palabra reservada *virtual*, ya que el proxy generado por Entity Framework para realizar el seguimiento de cambios es dinámico y WCF no puede serializar este tipo de objetos, por lo que es necesario eliminar esta característica. En consecuencia también se deja de utilizar la característica *Lazy loading*, que permite la carga de datos diferida de las propiedades de las entidades POCO, y se requiere la adaptación del modelo de programación para cargar las propiedades relacionadas. Adicionalmente la serialización de tipos de objetos con referencias a objetos de su mismo tipo (relaciones reflexivas), generan ciclos de iteración que EF tampoco puede manejar.

3 Análisis

3.1 Requerimientos funcionales / no funcionales

Los siguientes apartados contienen el catálogo de requerimientos tanto funcionales como no funcionales que el *software* desarrollado en este proyecto cubrirá. Cada requisito está identificado unívocamente por un identificador, tiene una descripción que resume el requisito, y está clasificado según las siguientes categorías:

- **Necesario:** Indica que es obligatorio que el requisito esté cubierto por el *software* en desarrollo.
- **Opcional:** Indica que el *software* puede ser liberado sin la implementación del requisito.
- **Deseable:** Indica que siempre que sea posible, se tratará de implementar el requisito, pero no será requerido para la liberación del producto final.

Requerimientos funcionales

Req. Nº	Categoría	Prioridad	Descripción
RF.01.01	01. Departamento	1. Necesario	El sistema almacenará obligatoriamente el nombre de los departamentos que se definan en el sistema.
RF.01.02	01. Departamento	1. Necesario	El sistema podrá almacenar opcionalmente los siguientes datos de un departamento: - Usuario responsable del departamento - Departamento del que depende
RF.01.03	01. Departamento	1. Necesario	El sistema permitirá crear nuevos departamentos.
RF.01.04	01. Departamento	1. Necesario	El sistema permitirá modificar los datos de cada departamento.
RF.01.05	01. Departamento	1. Necesario	El sistema permitirá eliminar un departamento si no existen otros que dependan de él.
RF.01.06	01. Departamento	1. Necesario	El sistema permitirá localizar un departamento realizando una búsqueda incremental con los caracteres introducidos.

Req. Nº	Categoría	Prioridad	Descripción
RF.02.01	02. Usuario	1. Necesario	El sistema almacenará obligatoriamente los siguientes datos de un usuario: <ul style="list-style-type: none"> - Identificador (login) - Contraseña - Login habilitado (si/no). Si, por defecto. - Nombre completo - Departamento al que pertenece - Dirección de email - Notificar por email (si/no). No, por defecto.
RF.02.02	02. Usuario	1. Necesario	El sistema solo permitirá a un usuario pertenecer a un solo departamento en un momento dado.
RF.02.03	02. Usuario	1. Necesario	El sistema permitirá que un usuario pueda cambiar la pertenencia a otro departamento.
RF.02.04	02. Usuario	1. Necesario	El sistema permitirá habilitar o deshabilitar un perfil de administrador de cualquier usuario creado.
RF.02.05	02. Usuario	1. Necesario	El sistema asignará automáticamente el perfil de autorizador a cualquier usuario que resulte de su asignación como responsable a un departamento.
RF.02.06	02. Usuario	1. Necesario	El autorizador de un usuario es el responsable del departamento al que pertenece el usuario.
RF.02.06	02. Usuario	1. Necesario	El sistema no permitirá que un usuario sea eliminado si ya ha realizado alguna liquidación.
RF.03.01	03. Liquidación	1. Necesario	El sistema almacenará obligatoriamente los siguientes datos de una liquidación: <ul style="list-style-type: none"> - Nombre del empleado. - Mes y año de la liquidación. - Estado de la liquidación.
RF.03.02	03. Liquidación	1. Necesario	Una liquidación podrá incluir tantos gastos como sean necesarios.
RF.03.03	03. Liquidación	1. Necesario	Los gastos incluidos en una liquidación pueden haberse realizado en cualquier momento, y no tienen por qué coincidir con el mes en el que se realiza la liquidación.
RF.03.04	03. Liquidación	1. Necesario	El sistema creará automáticamente una liquidación para el usuario en el mes en curso si no tiene ninguna abierta.
RF.03.05	03. Liquidación	1. Necesario	El sistema no permitirá a un usuario tener más de una liquidación abierta al mismo tiempo.

Req. Nº	Categoría	Prioridad	Descripción
RF.03.06	03. Liquidación	1. Necesario	El sistema marcará una liquidación en estado "Abierta" cuando se haya creado y no se haya solicitado aún autorización (estado "Esperando autorización").
RF.03.07	03. Liquidación	1. Necesario	El sistema marcará una liquidación en estado "Abierta" cuando esté en estado "Rechazada" o "Autorizada" y el usuario realiza de nuevo cambios sobre ella.
RF.03.08	03. Liquidación	1. Necesario	El sistema marcará una liquidación en estado "Esperando autorización" cuando ha sido solicitada la autorización por el usuario al que pertenece.
RF.03.09	03. Liquidación	1. Necesario	El sistema marcará una liquidación en estado "Rechazada" cuando el autorizador la ha rechazado.
RF.03.10	03. Liquidación	1. Necesario	El autorizador deberá proporcionar una breve descripción de las razones de rechazo en caso de que rechace una autorización.
RF.03.11	03. Liquidación	1. Necesario	El sistema marcará una liquidación en estado "Autorizada" cuando el autorizador la autorice.
RF.03.12	03. Liquidación	1. Necesario	El sistema pasará a estado "Pagada" las liquidaciones que se encuentren en proceso de transferencia al sistema de nóminas.
RF.03.13	03. Liquidación	1. Necesario	El sistema marcará una liquidación en estado "Cerrada" cuando el sistema ya la haya transferido al sistema de nóminas.
RF.03.14	03. Liquidación	1. Necesario	El sistema solo permitirá consultar sus liquidaciones al usuario al que pertenecen y a su autorizador.
RF.03.15	03. Liquidación	1. Necesario	El sistema permitirá al usuario modificar una liquidación existente si no se encuentra en estado "Pagada" o "Cerrada".
RF.03.15	03. Liquidación	1. Necesario	El usuario podrá solicitar una autorización de pago a su autorizador desde el formulario de liquidación para realizar su abono a través de la nómina.
RF.03.17	03. Liquidación	1. Necesario	El sistema transferirá de forma automática las cantidades a abonar al sistema de nóminas el día 25 de cada mes, volcando un fichero de transferencias a nóminas de todas aquellas liquidaciones en estado "Autorizada".

Req. Nº	Categoría	Prioridad	Descripción
RF.03.18	03. Liquidación	1. Necesario	Cuando un usuario se encuentra trabajando en estado desconectado, es decir, sin conexión al servidor central, no podrá modificar tampoco las liquidaciones en estado "Autorizado" para evitar problemas de sincronización durante el proceso de transferencias al sistema de nóminas.
RF.04.01	04. Gasto	1. Necesario	El sistema almacenará obligatoriamente los siguientes datos para cada gasto: <ul style="list-style-type: none"> - Tipo de gasto - Fecha del gasto - Importe total del gasto (incluyendo impuestos) - Descripción del gasto (causa o motivo) - Moneda (por defecto la que el usuario tenga seleccionada)
RF.04.02	04. Gasto	1. Necesario	El sistema permitirá almacenar opcionalmente la imagen digital del comprobante del gasto.
RF.04.03	04. Gasto	1. Necesario	El sistema solo contemplará los siguientes tipos de gastos para clasificar la introducción de estos: <ul style="list-style-type: none"> - billetes de tren, avión, barco o autobús - alquiler de coche - taxi - gastos de representación - formación - combustible - uso y alquiler de teléfono - hotel - kilometraje en vehículo propio - dietas diarias - aparcamiento - peaje - portes - otros gastos
RF.04.04	04. Gasto	1. Necesario	El sistema determinará el importe de la dieta que se debe abonar cuando un gasto sea de tipo "dietas diarias".
RF.04.05	04. Gasto	1. Necesario	El cálculo de la dieta se realizará comprobando si la fecha del gasto corresponde con un día festivo del calendario de días festivos que tenga asignado el usuario. Entonces la dieta será ordinaria (no es día festivo) ó extraordinaria (es día festivo).

Req. Nº	Categoría	Prioridad	Descripción
RF.05.01	05. Calendario	1. Necesario	El sistema permitirá crear calendarios de días festivos.
RF.05.02	05. Calendario	1. Necesario	El sistema almacenará obligatoriamente el nombre de los calendarios de días festivos y la lista de días festivos.
RF.05.03	05. Calendario	1. Necesario	El sistema creará los calendarios de días festivos, incluyendo por defecto como días festivos los sábados y domingos de cada fin de semana desde el 1 de enero hasta el 31 de diciembre.
RF.05.04	05. Calendario	1. Necesario	El administrador podrá marcar o desmarcar un día festivo a través de un calendario que identifique si el día es festivo o no.
RF.05.05	05. Calendario	1. Necesario	El sistema permitirá duplicar (hacer una copia) de un calendario de días festivos.
RF.05.06	05. Calendario	1. Necesario	El sistema permitirá reestablecer los días festivos por defecto en un calendario, que serán los fines de semana.
RF.05.07	05. Calendario	1. Necesario	El administrador podrá hacer cambios sobre un calendario existente y guardarlos.
RF.05.08	05. Calendario	1. Necesario	El administrador no podrá eliminar calendarios que ya hayan sido utilizados para realizar cálculos de liquidaciones.
RF.06.01	06. Seguridad	1. Necesario	La aplicación cliente siempre pedirá al usuario una identificación y una contraseña para comenzar a operar. El usuario y la contraseña deben haber sido previamente parametrizados por un administrador.
RF.06.02	06. Seguridad	1. Necesario	La aplicación cliente ofrecerá un procedimiento en la interfaz de usuario para cerrar la sesión actual y volver al proceso de identificación.
RF.06.03	06. Seguridad	1. Necesario	Cuando la aplicación cliente se inicia, esta intentará conectar al servidor para poder validar en línea las credenciales del usuario. Si no es posible conectarse, se activa el modo “desconectado” e intenta autenticar al usuario a través de la base de datos local.
RF.06.04	06. Seguridad	1. Necesario	La aplicación cliente ofrecerá un mecanismo en la interfaz de usuario de conectar o desconectar las comunicaciones con el servidor a voluntad del usuario.

Req. Nº	Categoría	Prioridad	Descripción
RF.06.05	06. Seguridad	2. Opcional	Pasados 5 minutos de inactividad, la aplicación cliente se bloqueará automáticamente y solicitará de nuevo las credenciales al usuario para continuar trabajando.
RF.06.06	06. Seguridad	3. Deseable	La aplicación cliente ofrecerá un mecanismo en la interfaz de usuario para indicar si se desea "reconexión automática" (en intervalos de 5 minutos).
RF.06.07	06. Seguridad	3. Deseable	Si la aplicación cliente se encuentra en estado de desconexión y la opción de reconexión automática está activada, esta intentará automáticamente conectarse al servidor en intervalos de 5 minutos.
RF.06.08	06. Seguridad	3. Deseable	El sistema permitirá al usuario parametrizar el tiempo de "reconexión automática".
RF.06.09	06. Seguridad	3. Deseable	El sistema permitirá al usuario parametrizar el tiempo de "bloqueo automático".
RF.07.01	07. Notificación	1. Necesario	El sistema emitirá las notificaciones de aviso como notas emergentes a través de la interfaz de usuario, cuando esté conectado.
RF.07.02	07. Notificación	3. Deseable	El sistema emitirá las notificaciones de aviso a través de correos electrónicos cuando el usuario ó el autorizador así lo tengan configurado.
RF.07.03	07. Notificación	1. Necesario	El sistema avisará al usuario y al autorizador implicado cuando las solicitudes del primero cambien de estado.
RF.07.04	07. Notificación	1. Necesario	El sistema avisará al autorizador cuando tenga solicitudes pendientes de autorización.
RF.07.05	07. Notificación	1. Necesario	La aplicación cliente deberá notificar al sistema de posibles caídas de sistema y de su causa siempre que sea posible.
RF.07.06	07. Notificación	1. Necesario	La aplicación servidora deberá notificar al sistema de posibles caídas de sistema y de su causa siempre que sea posible.
RF.08.01	08. Integración	1. Necesario	El sistema sincronizará automáticamente los datos del usuario, las liquidaciones y las autorizaciones pendientes cada vez que se conecten el cliente y el servidor.
RF.08.02	08. Integración	1. Necesario	La interfaz con el software de gestión de nóminas estará basada en un fichero CSV o delimitado por comas.

Req. Nº	Categoría	Prioridad	Descripción
RF.08.03	08. Integración	1. Necesario	El formato del fichero CSV delimitado por comas contendrá los siguientes campos: DNI, Nombre del tipo de gasto, Importe, Moneda.
RN.09.01	09. Interfaz	1. Necesario	La aplicación cliente deberá seguir un diseño intuitivo y sencillo que indique al cliente en todo momento que opciones puede realizar y como llevarlas a cabo.
RN.10.01	10. Mantenibilidad	1. Necesario	La aplicación dispondrá de un sistema de instalación automático tanto para el software de servidor como para el software de cliente.
RN.11.01	11. Capacidad	1. Necesario	El sistema debe ser capaz de escalar tanto vertical como horizontalmente de acuerdo a los requisitos físicos se precisen en cada momento para dar soporte a las necesidades de almacenamiento, procesamiento y concurrencia.
RN.12.01	12. Comunicación	3. Deseable	Siempre que sea posible se utilizaran estándares abiertos de comunicación entre la aplicación cliente y servidora, tales como servicios web por HTTP.
RN.13.01	13. Plataforma	1. Necesario	El sistema se desarrollará se realizará con tecnología Microsoft .NET.

Requerimientos no funcionales

Req. Nº	Categoría	Prioridad	Descripción
RN.09.01	09. Interfaz	1. Necesario	La aplicación cliente deberá seguir un diseño intuitivo y sencillo que indique al cliente en todo momento que opciones puede realizar y como llevarlas a cabo.
RN.10.01	10. Mantenibilidad	1. Necesario	La aplicación dispondrá de un sistema de instalación automático tanto para el software de servidor como para el software de cliente.
RN.11.01	11. Capacidad	1. Necesario	El sistema debe ser capaz de escalar tanto vertical como horizontalmente de acuerdo a los requisitos físicos se precisen en cada momento para dar soporte a las necesidades de almacenamiento, procesamiento y concurrencia.
RN.12.01	12. Comunicación	3. Deseable	Siempre que sea posible se utilizaran estándares abiertos de comunicación entre la aplicación cliente y servidora, tales como servicios web por HTTP.

RN.13.01	13. Plataforma	1. Necesario	El sistema se desarrollará se realizará con tecnología Microsoft .NET.
----------	----------------	--------------	--

3.2 Diagrama de casos de uso

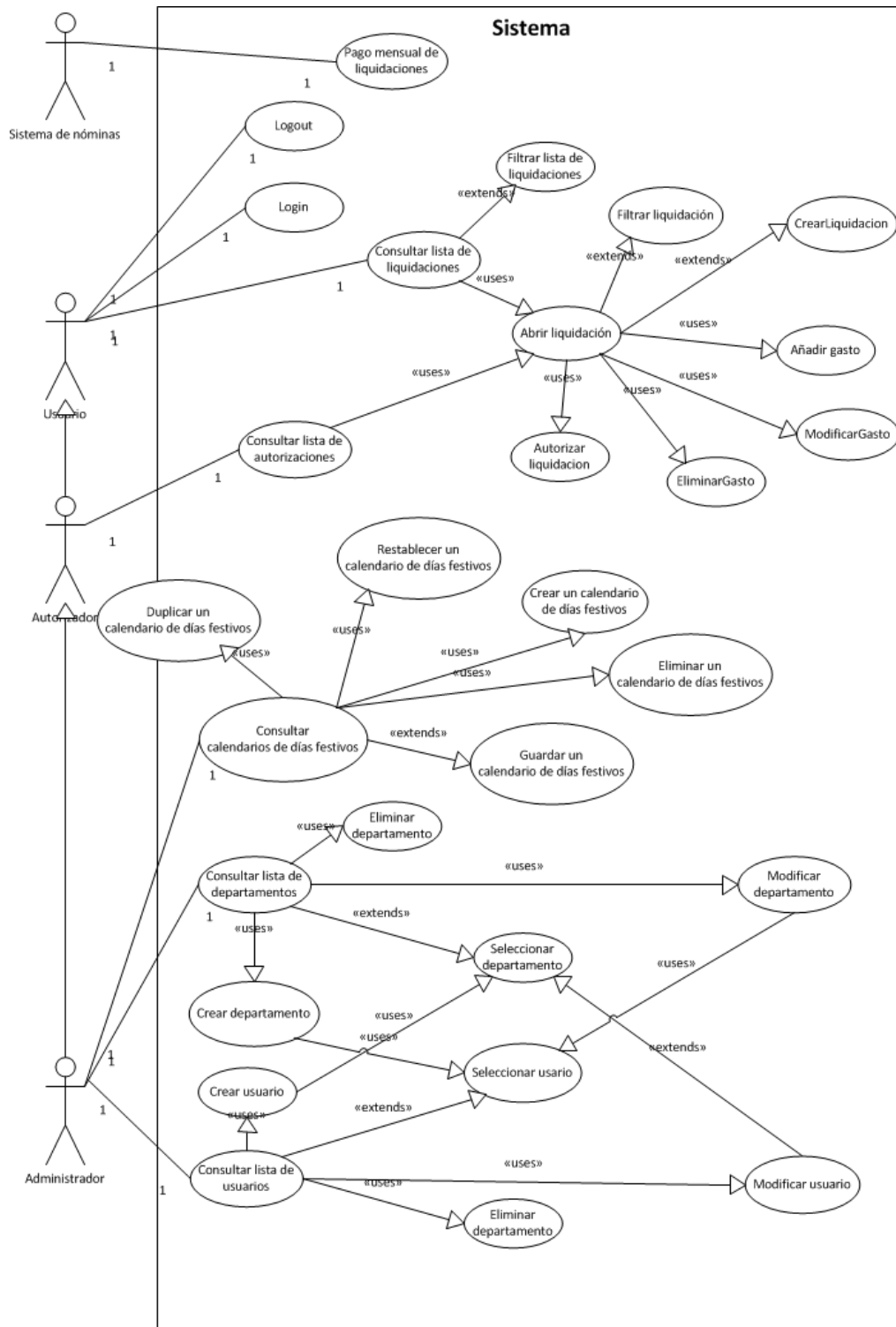


Ilustración 2 - Diagrama de casos de uso

Actores

- **Usuario:** Es el que tiene menor nivel de derechos de operación sobre el sistema. Puede realizar los procesos de identificación, creación, consulta y modificación de sus propias liquidaciones de gastos.
- **Autorizador:** Dispone de los mismos derechos de operación que el usuario, pero adicionalmente puede autorizar o rechazar las solicitudes de aprobación de las liquidaciones de los usuarios. Su función es asignada automáticamente por el sistema cuando un usuario es asignado como responsable de un departamento.
- **Administrador:** Dispone de derechos de operación para configurar la estructura organizativa de la aplicación sobre la compañía: departamentos y usuarios, así como para establecer los calendarios de días festivos para realizar liquidaciones.
- **Sistema de pago de nóminas:** Sistema externo que recibe la relaciones de liquidaciones de usuarios que deben ser incluidas en la nómina mensual.

Los procesos de alto nivel que son iniciados por los actores anteriormente relacionados son: login (autenticación), consulta de liquidaciones, consulta de autorizaciones, consulta de calendarios de días festivos, consulta de departamentos y consulta de usuarios. A partir de estos casos de uso primarios se especifican o extienden casos de uso más específicos que refinan cada uno de los procesos. A modo de ejemplo se copia a continuación el caso de login:

Nombre del caso de uso	Login
Identificación del caso de uso	CU.06.01
Descripción (resumen)	Solicita las credenciales a un usuario y las valida.
Actores	Usuario
Casos de uso relacionados	Logout (CU.06.02)
Precondición	El sistema muestra la ventana de login para recuperar las credenciales del usuario.
Postcondición	El sistema muestra las opciones del menú que el usuario puede utilizar en función de su perfil.
Proceso normal	<ol style="list-style-type: none"> 1. El sistema muestra la ventana de login. 2. El usuario introduce el nombre de usuario, la contraseña y pulsa el botón "login". 3. El sistema cierra la ventana de login y configura el entorno para el perfil del usuario
Alternativas de proceso y	3b. Los datos introducidos por el usuario no son

excepciones correctos (no existe el usuario o no coincide la contraseña).

3b1. El sistema muestra un mensaje indicando que las credenciales no son válidas.

3b2. El sistema vuelve al paso 1.

3.3 Modelo conceptual

A partir del análisis y estudio del problema se han identificado las siguientes relaciones y estructuras de datos que permiten modelar el diseño de la solución software. Para representar este modelo conceptual se utiliza el diagrama entidad-relación siguiente:

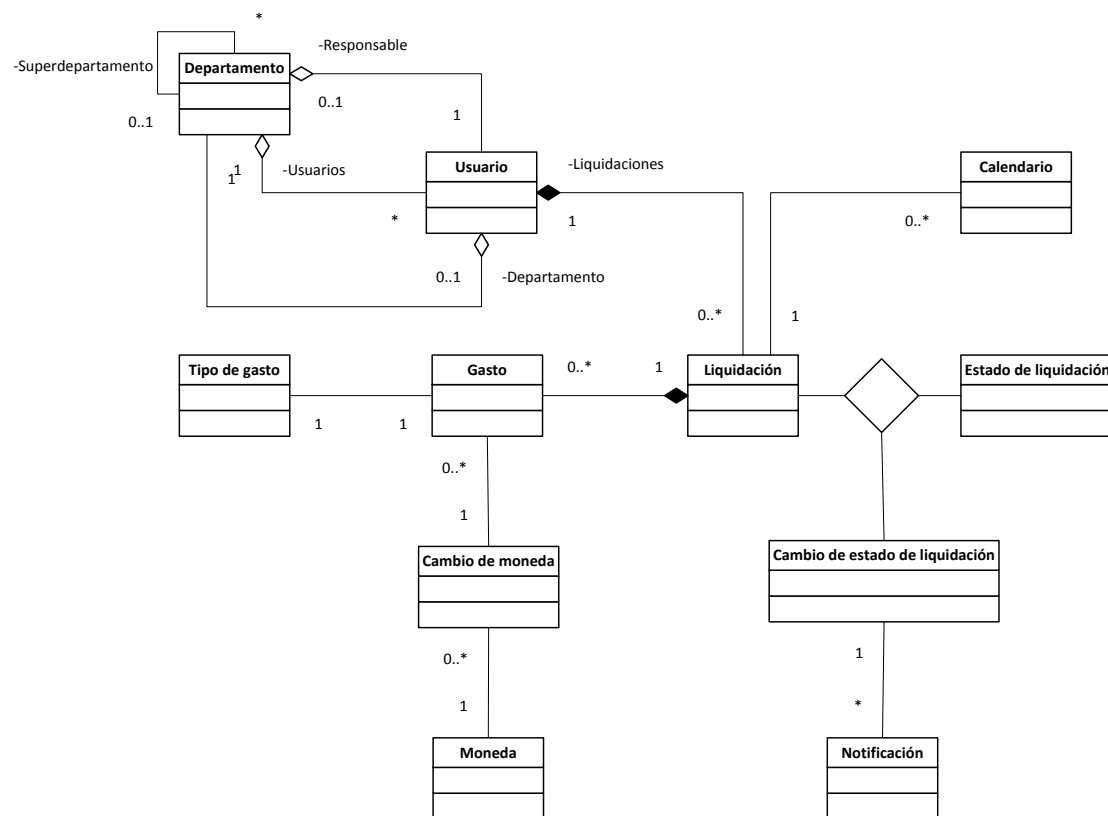


Ilustración 3- Modelo conceptual

Las entidades son unidades de información que asocian información relativa a objetos, personas o hechos relevantes dentro del dominio de tratamiento de la solución software.

El núcleo del modelo conceptual se fundamenta en la entidad *Liquidación*, que agrupa todos los gastos comprendidos durante un periodo de tiempo (típicamente un mes). La liquidación comenzará con un *Estado de liquidación* asociado que tipificará la situación dentro del proceso de liquidación en la que se encuentra (abierta, aprobada, rechazada, cerrada, etc). Cada cambio en el estado de liquidación se asocia con una entidad *Cambio de estado de liquidación* que ofrece un mecanismo de trazabilidad para conocer el momento y

condiciones en que se produce el cambio de estado. Adicionalmente, un cambio de liquidación estará asociado a una o varias entidades *Notificación* que permitirán registrar y gestionar los avisos necesarios que el sistema debe ofrecer al usuario al que pertenece la liquidación.

Por otro lado una liquidación podrá estar compuesta por un número variable de entidades *Gasto*, que informará de su naturaleza, importe y condiciones en las que se produce. Para facilitar el tratamiento de cada gasto se clasifica su tipo con una entidad asociada *Tipo de gasto*, que indicará su ámbito de aplicación (dietas, hoteles, taxis, etc).

Las liquidaciones computan sus costes siempre en euros, por lo que desde el punto de vista de control de costes es necesario unificar el importe de cada uno de los gastos desde la moneda o divisa en la que se produce el abono del gasto. Para cumplir con esta exigencia, se define una entidad *Moneda* que actúa como divisa asociada al gasto, y se relaciona con el gasto a través de la entidad *Cambio de moneda*, que mantiene asociada la relación del cambio de divisa con el Euro, en función de las variaciones de cotización que las divisas experimenten y se registren en el sistema.

Desde el punto de vista de seguridad y organización, se definen las entidades *Usuario* y *Departamento*. Cada usuario tendrá asociada un número variables de liquidaciones, y de la relación de pertenencia entre los propios usuarios con el departamento se generan los roles de *Usuario* y *Responsable* de departamento (o *Autorizador*). Finalmente, la jerarquía de la organización entre los usuarios quedará definida por la dependencia entre departamentos, donde un departamento podrá tener definido un departamento superior o ninguno (posición más alta de la jerarquía), y a su vez un departamento podrá tener un numero variable de subdepartamentos o departamentos dependientes.

4 Diseño

4.1 Diagrama de arquitectura: de aplicación / SW / HW

Debido a los requisitos indicados en la fase de análisis, el producto final debe contar con un elemento o aplicación que opere en el ordenador del usuario aun cuando éste no disponga de conexión a las oficinas centrales de la compañía para consultar u operar con sus liquidaciones de gastos.

La arquitectura más adecuada para este tipo de distribución lógica responde a una arquitectura “cliente-servidor” (Wikipedia, 2013), en la que la aplicación cliente, también conocida como cliente, solicita datos o acciones al servidor mediante peticiones. El servidor entonces realiza la acción o recupera los datos solicitados, devolviendo el resultado como respuesta. El esquema conceptual de este tipo de arquitectura se representa en la figura siguiente:

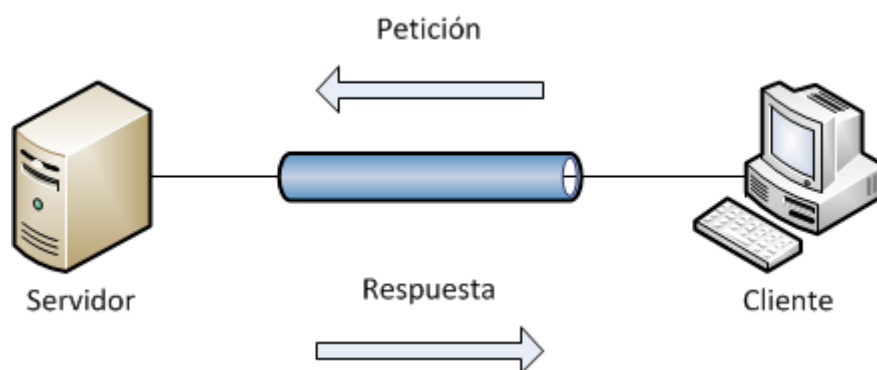


Ilustración 4 - Concepto cliente-servidor

La arquitectura “cliente-servidor” se distingue de otras arquitecturas parecidas en el hecho de que tanto el cliente como el servidor se reparten cierta capacidad de procesamiento y/o almacenamiento, precisamente la característica necesaria para que el cliente pueda seguir operando cuando no existe conexión con el servidor.

Modelo lógico

La arquitectura lógica de la aplicación servidora y cliente cuentan con un diseño tradicional multicapa como se detalla en el diagrama de bloques en la figura siguiente:

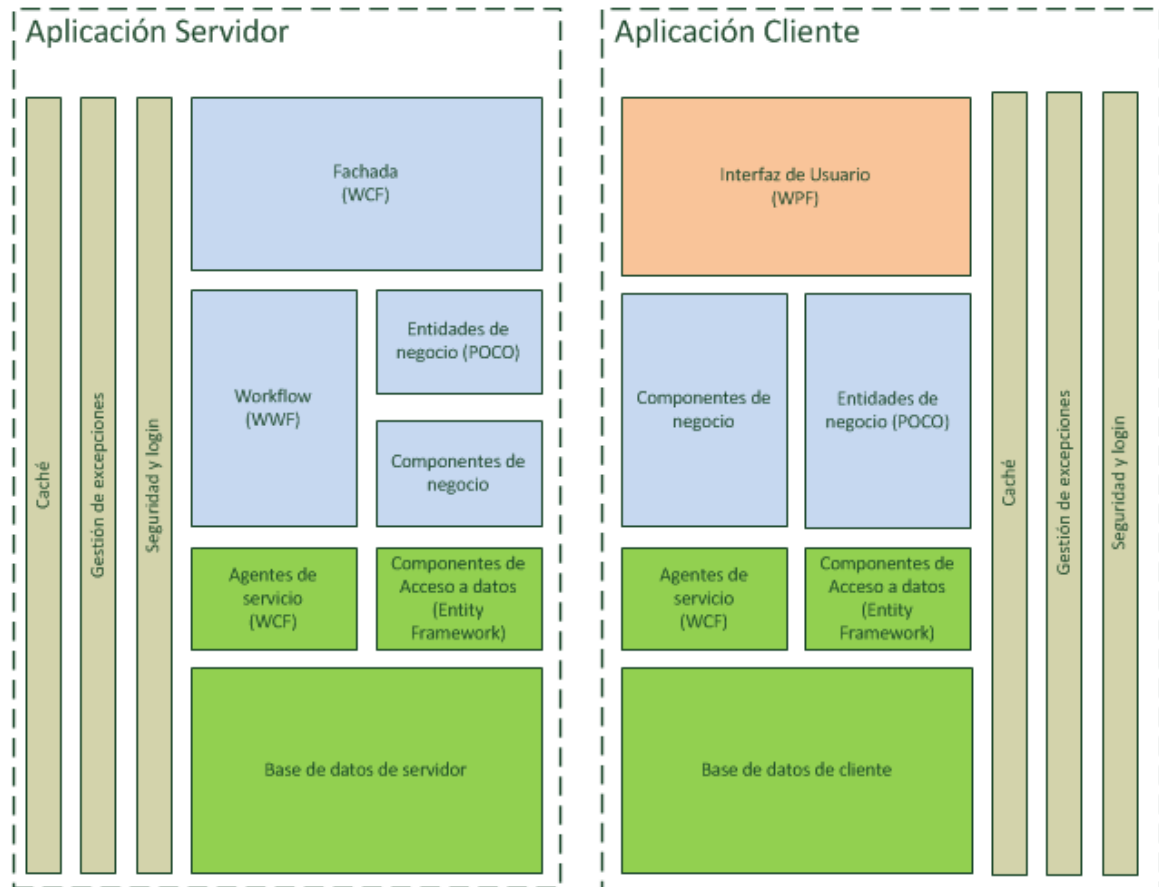


Ilustración 5 - Diagrama conceptual del modelo lógico de aplicación multicapa

En el diseño tradicional multicapa, se realizan tres grandes separaciones: interfaz de usuario (de color naranja), lógica de negocio (de color azul) y persistencia de datos (de color verde). Adicionalmente en el modelo pueden existir servicios transversales (utilizados por todas las capas anteriores) a la arquitectura de la aplicación.

Entre los objetivos principales del diseño multicapa se busca desacoplar los componentes existentes entre ellas, para obtener un mayor nivel de abstracción y ganar en escalabilidad, facilidad de implementación y mantenimiento.

Los elementos comunes a ambas arquitecturas son los componentes y las entidades de negocio que regulan el funcionamiento de la aplicación. Los agentes de servicio y acceso a datos permiten abstraer a la aplicación de sus respectivas implementaciones.

Sin embargo la aplicación servidora a diferencia de la aplicación cliente no ofrece una interfaz de usuario ya que la solución siempre será explotada a través del cliente. En su lugar presenta una interfaz de comunicación también denominada fachada (Fowler, 2013) en referencia al patrón de diseño.

En algunos componentes se ha introducido una referencia a las tecnologías que operan a ese nivel.

Espacios de nombres

Los espacios de nombres constituyen una herramienta muy importante en la arquitectura de la aplicación, y tiene por objetivos fundamentales los dos siguientes:

- Crear un sistema de organización de clases más fácil de entender, utilizar y mantener.
- Controlar el ámbito de nombres de clases y métodos.

A partir de estas máximas se han definido los siguientes espacios de nombres que actuarán como contenedores de las clases en cada uno de sus niveles e implementarán los patrones de diseño indicados:

Tabla 9- Tabla de espacio de nombres

Variable	Valor
LiquidacionGastos	Espacio de nombres elemental de la solución software. En él se encuentra la clase <i>App</i> de aplicación.
LiquidacionGastos.Framework	<p>Contiene las clases de uso básico y transversal a todas las capas de aplicación. Fundamentalmente son clases base que implementan los patrones:</p> <ul style="list-style-type: none"> • <i>Command</i> (Wikipedia, 2013) • <i>Observer</i> (Fowler, Observer Pattern, 2013) • <i>ServiceLocator</i> (Microsoft MSDN, 2013) • <i>State Machine</i> (Fowler, State Machine, 2013) <p>y otra clases de soporte y excepciones relacionadas.</p>
LiquidacionGastos.Services	Agrupar clases de servicios de aplicación tanto de naturaleza técnica como funcional (relativa al dominio de la solución).
LiquidacionGastos.Synch	Contiene las clases que realizan la sincronización de datos entre la aplicación cliente y servidora.
LiquidacionGastos.Views	Comprende los elementos de interfaz de usuario denominados Vistas (<i>Views</i>) relativos al patrón <i>MVVM</i> .
LiquidacionGastos.ViewModels	Comprende los elementos de control de las Vistas (<i>Views</i>) y coordinación de datos relativos al patrón <i>MVVM</i> .
LiquidacionGastos.WebServerRepositories	Incluye las clases de consumo de servicios web siguiendo un patrón <i>Repository</i> .
LiquidacionGastos.Business	Engloba las clases de lógica de negocio del dominio de la aplicación. Típicamente son clases que ofrecen operaciones (API) de alto nivel para los procesos del negocio.
LiquidacionGastos.Models	Contiene las clases <i>POCO</i> (Plain Old CLR Object) que representan las entidades de datos de la aplicación.

LiquidacionGastos.Repositories	Incluye las clases que abstraen el acceso a los orígenes de datos a través de la implementación del patrón <i>Repository</i> (Microsoft MSDN, 2013).
LiquidacionGastos.UnitsOfWork	Comprende las clases que abstraen las operaciones de acceso a datos coordinadas, tales como ejecución transaccional, acceso distribuido y/o acceso compartido a fuentes de datos. Estas clases implementan el patrón <i>UnitOfWork</i> (Fowler, Unit Of Work Pattern, 2013).
LiquidacionGastos.Security	Agrupar las clases relacionadas con aspectos de seguridad de la aplicación tales como Autenticación, gestión de roles y del objeto Principal (Contexto de seguridad de ejecución del código) ó seguridad de clave de acceso basada en PBKDF2-SHA1 (Wikipedia, 2013).

Finalmente las capas de la aplicación quedarán definidas según el siguiente diagrama:

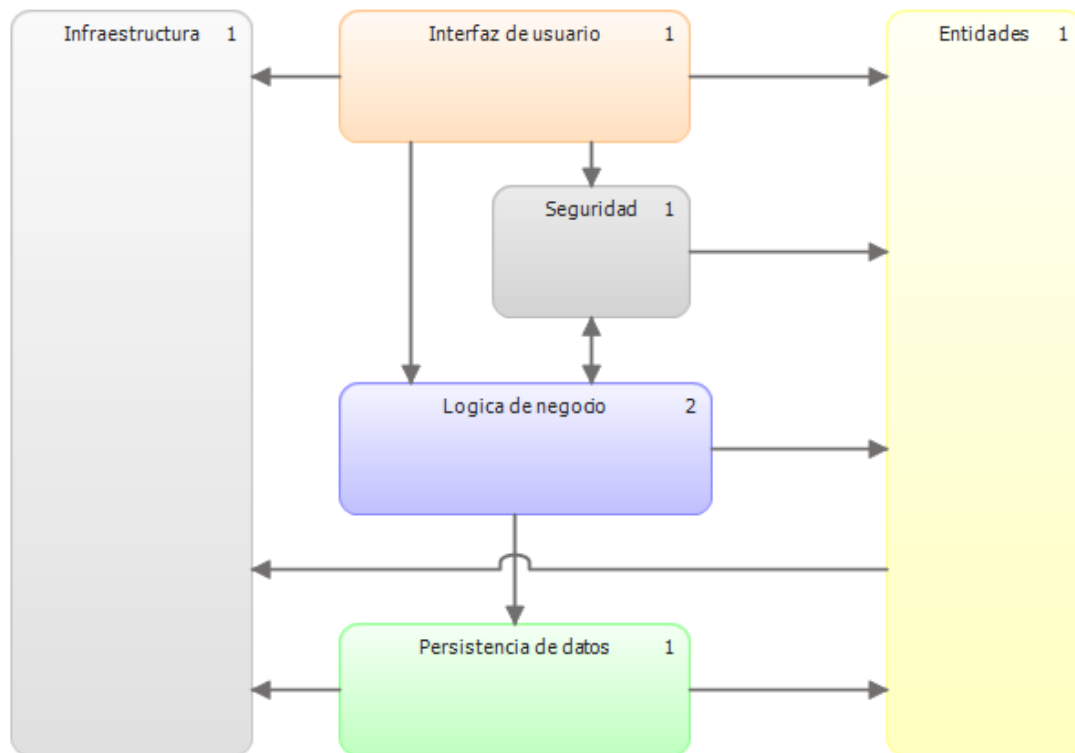


Ilustración 6 - Diagrama de capas de la solución

Por último se ofrece el modelo de componentes relacionados con cada una de las capas de la figura anterior, donde cada uno de los nombres de los componentes coinciden con el ámbito del espacio de nombres de la tabla anterior:

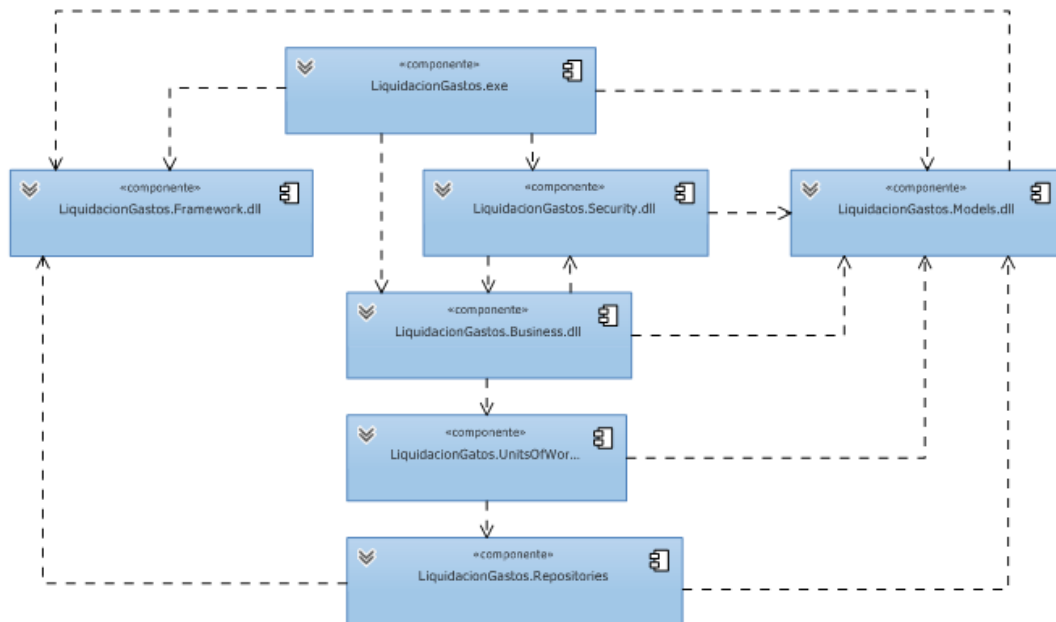


Ilustración 7 - Diagrama de componentes de la solución

Modelo físico

El la arquitectura física de la solución se divide en dos grupos atendiendo a las necesidades el programa cliente y del programa servidor.

El programa cliente requiere una interfaz de usuario rica, es decir, con múltiples métodos de entrada y salida de información, intuitivos y con un tiempo de respuesta rápido. Puesto que la implementación se realiza a través de WPF el producto más adecuado es un ensamblado ejecutable (.exe).

Por otro lado el programa servidor, debe exponer Servicios Web, y no precisa la incorporación de una interfaz de usuario para su gestión u operación, ya que su cometido principal es proveer servicios de datos y sincronización al programa cliente. Con este planteamiento, se pueden llevar a cabo diferentes aproximaciones.

Si bien, la que ofrece mayor nivel de escalabilidad es la implementación a través de IIS (Internet Information Server), se implementará como servicio Windows ya que esta última aproximación reduce los costes de licenciamiento del servidor web y facilita el despliegue.

A continuación se presenta el diagrama UML de despliegue:

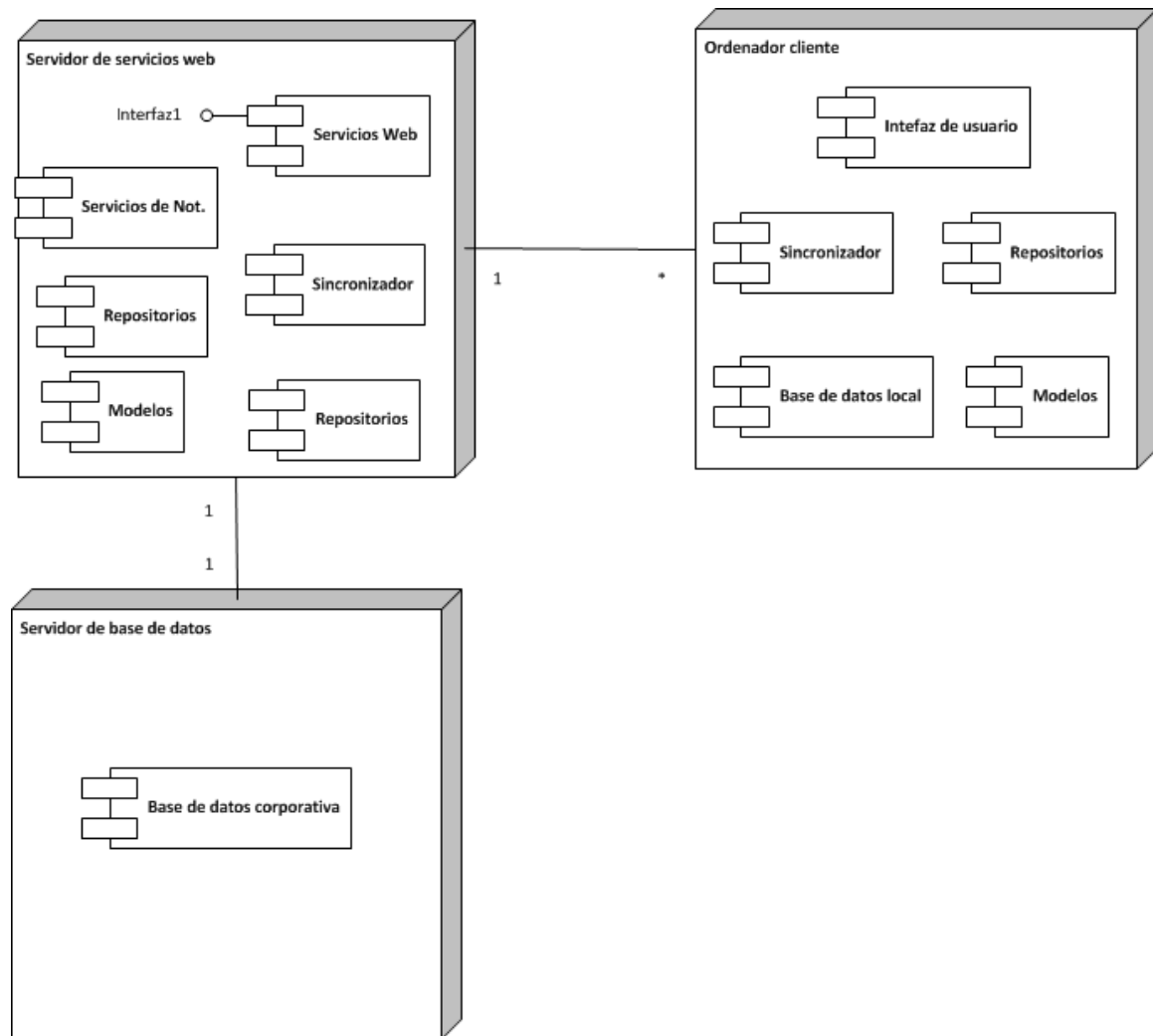


Ilustración 8 - Diagrama de implementación

Cabe señalar que en la arquitectura “cliente-servidor”, el servidor no tiene porqué ser un único servidor físico o un único programa, sino que pueden ser diferentes servidores o programas trabajando como una única unidad lógica. Esta idea introduce los conceptos de escalado horizontal y vertical (Wikipedia, 2013) para ajustar las capacidades del modelo físico a los requisitos reales del sistema.

En su dimensión más elemental la arquitectura física de la solución está compuesta por un servidor y un ordenador personal que actuará de cliente. En el apartado de implementación se describe los detalles de los componentes que formarán la solución y su ubicación en los posibles nodos físicos.

4.2 Diseño de la BD / Diagrama E-R

La solución software incorpora dos bases de datos: para el programa servidor y para el programa cliente. En ambos casos la tecnología utilizada para la persistencia de datos es SQL Server 2012 en cualquiera de sus ediciones comprendidas entre la Express hasta la Enterprise para el servidor. No obstante, se recomienda la edición Express para la aplicación cliente por motivos de licenciamiento y recursos mínimos requeridos.

En cuanto al modelo de datos lógico cabe señalar que será compartido por ambos programas cliente y servidor en sus respectivos motores de base datos mencionados anteriormente.

El siguiente diagrama muestra el modelo entidad-relación diseñado para realizar la persistencia de datos:

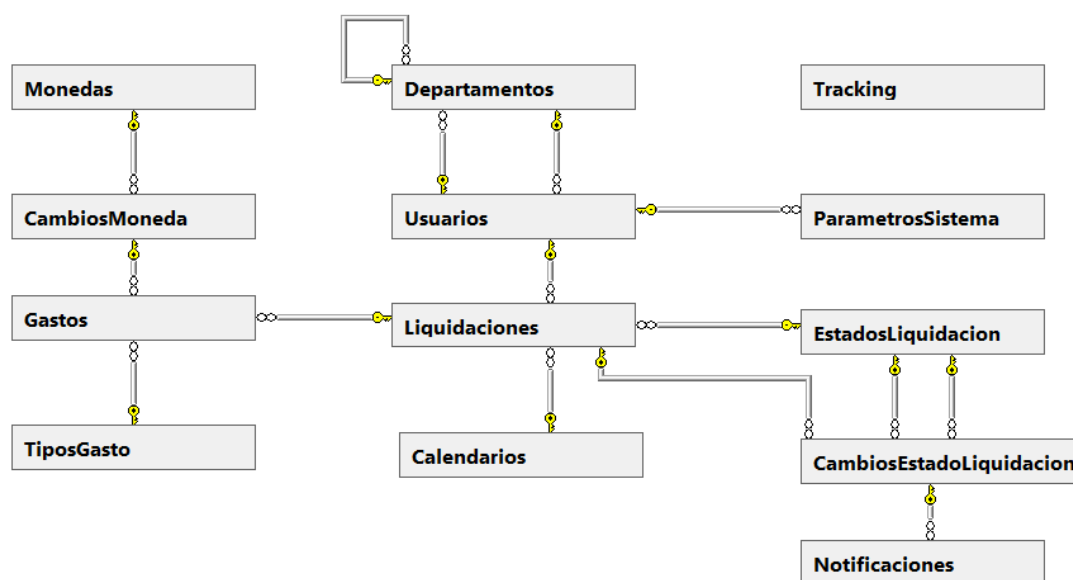


Ilustración 9 - Diagrama E-R Liquidación de gastos

De forma general el modelo de base de datos se encuentra normalizado al menos en tercera forma normal (3NF), con la excepción de algunos valores de la tabla *ParametrosSistema* que guardan identificadores (claves primarias) de *Calendarios* y *Monedas*. En este caso la relación de integridad referencial se delega a la lógica de la aplicación para facilitar el mantenimiento y parametrización del administrador, ya que sus valores son evaluados en tiempo de ejecución y admite valores no existentes, es decir, si por ejemplo una moneda ya no existe se toma la predefinida en el sistema (tabla *ParametrosSistema* con valor *IdUsuario* igual a NULL).

Respecto al tipo de claves primarias utilizadas, se emplean dos tipos diferentes: integer y GUID (Global Unique Identifier).

Las tablas maestras (*Monedas*, *TiposGasto*, *ParametrosSistema* y *EstadosLiquidación*) emplean el primer tipo, ya que sus valores están prefijados, el volumen de datos asociados

es pequeño y existe una correspondencia bidireccional con tipos enumerados en las capas lógicas superiores.

El resto de tablas utilizan un tipo GUID, que a pesar de reducir el rendimiento de los índices respecto al tipo anterior, asegura la carencia de colisiones en la base de datos cuando las claves primarias son generadas desde la capa de negocio y en distintos ordenadores, para ser posteriormente consolidadas en las bases de datos del cliente y del servidor.

Ésta última aproximación tiene la ventaja de evitar una consulta adicional al motor de base de datos por cada nuevo registro para obtener el identificador generado por la misma base de datos, y permite inserciones relacionadas con las claves primarias reales desde la capa de lógica de negocio que posteriormente son actualizadas a la base de datos de forma masiva. Con este esquema se eliminan en gran medida las esperas del usuario en los procesos donde intervienen inserciones y se difieren a un momento en el que el proceso ya ha acabado y/o puede ser realizado de forma asíncrona.

En cuanto a la descripción del modelo E-R, éste se corresponde casi directamente con el modelo conceptual identificado en el análisis, si bien la tabla *Tracking* requiere una mención específica por su función, ya que actúa como una meta-tabla que contiene el registro de seguimiento de cambios del resto de entidades del modelo. Posteriormente durante el proceso de sincronización entre el cliente y servidor ésta información es utilizada para actualizar convenientemente los datos de ambos extremos de la solución.

El siguiente cuadro resumen explica la información contenida en cada tabla del modelo de la base de datos:

Tabla 10 - Tabla de descripción de tablas de base la base de datos

Tabla	Descripción
TiposGasto	Almacena la información relacionada con el tipo de gasto que clasifica a un gasto de la tabla Gasto.
Gastos	Almacena la información relacionada con los gastos del usuario incluidos en una liquidación.
Liquidaciones	Almacena la información sobre las liquidaciones de los usuarios en el sistema.
EstadosLiquidacion	Almacena la información relacionada con los estados de liquidación. Los identificadores de esta tabla son valores enteros que coinciden con valores constantes en la lógica de la aplicación.
ParametrosSistema	Almacena la información relativa a valores parametrizables del sistema. La clave primaria es un valor entero que corresponde con valores constantes del código fuente.
CambiosMoneda	Almacena la información relativa a los cambios de valor de las monedas ó divisas registradas en el sistema.

Usuarios	Almacena la información relativa a los usuarios.
CambiosEstadoLiquidacion	Almacena la información de los cambios de estado que sufren las liquidaciones.
Monedas	Almacena la información de las monedas o divisas con las que opera el sistema.
Departamentos	Almacena la información de los departamentos registrados en el sistema.
Notificaciones	Almacena la información de las notificaciones que genera el sistema a usuarios.
Calendarios	Almacena la información de los calendarios de días festivos.
Tracking	Almacena la información relativa al seguimiento de cambio de las entidades del modelo.

4.3 Modelo de clases

Diagrama de clases de entidades

Una entidad es una clase POCO que representa a una unidad de información de forma autónoma dentro del dominio de la solución. En este proyecto las entidades corresponden en una relación 1:1 con las definidas en el modelo E-R.

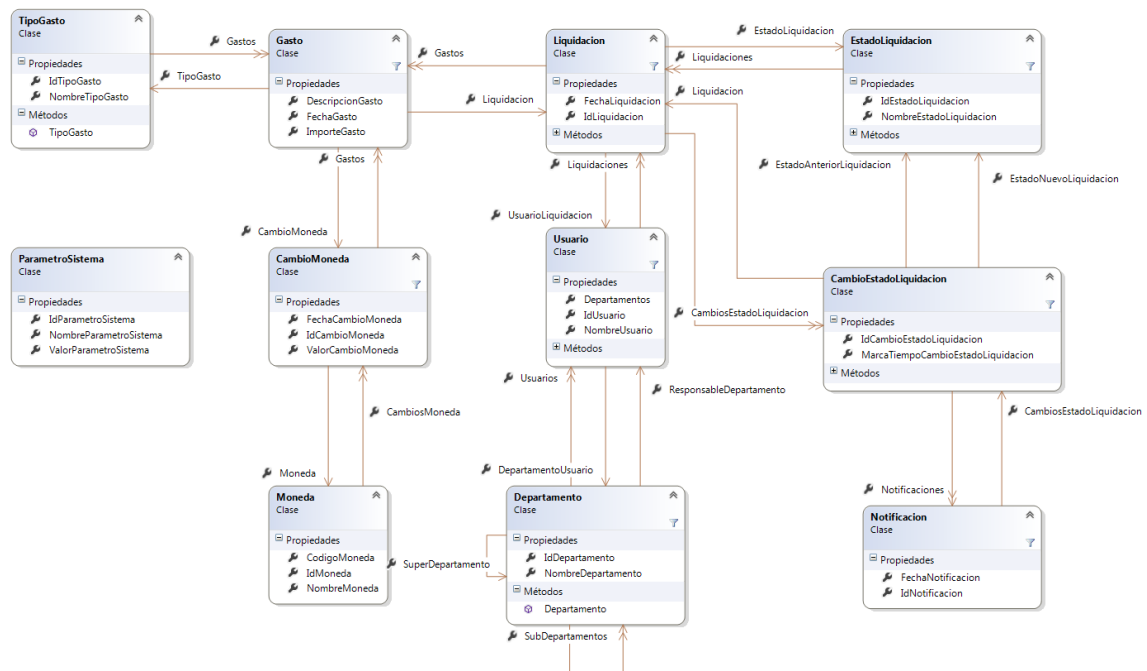


Ilustración 10 - Diagrama de clases de entidades

Diagrama de clases de repositorios de acceso a la base de datos

Los repositorios de datos que proporcionan acceso a la base de datos SQL SERVER, heredan de la clase base *EntityFrameworkRepository* cuya principal responsabilidad es abstraer la funcionalidad de gestión del contexto de la tecnología *Entity Framework*. A partir de este nivel, aparecen dos clases adicionales:

- *TrackingEntryRepository*: tiene la responsabilidad de gestionar las entradas en la tabla de seguimiento de entidades (tracking).
- *EntityFrameworkTrackedRepository*: cuya principal responsabilidad es proveer las funciones de acceso a datos interconectándolas con el sistema de seguimiento (tracking), para registrar los cambios de cada entidad.

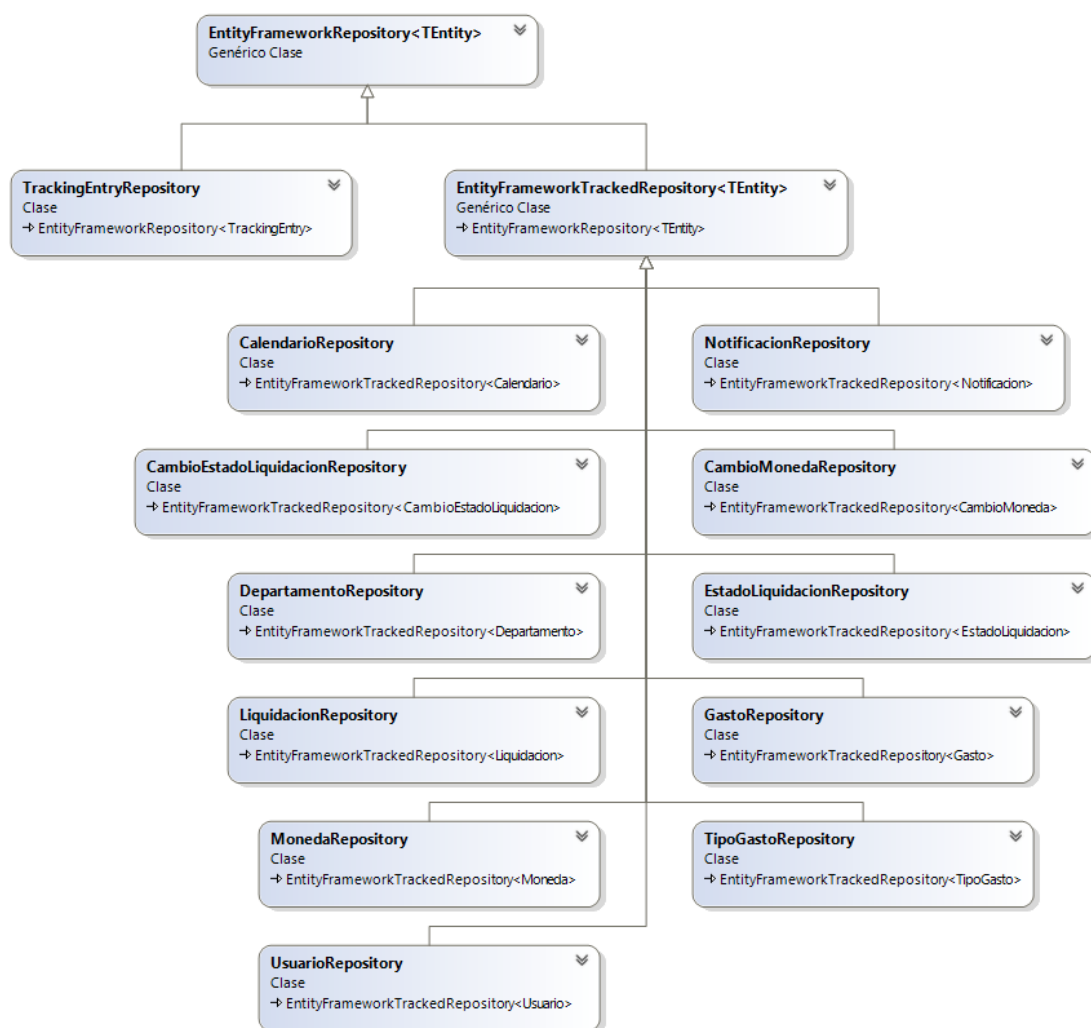


Ilustración 11 - Diagrama de clases de repositorios de acceso a la base de datos

El resto de clases que heredan de *EntityFrameworkTrackedRepository* son implementaciones concretas del patrón *Repository* para cada una de las entidades del modelo.

Diagrama de clases de repositorios de acceso a Servicios Web

Este tipo de repositorios proporcionan a la aplicación acceso a servicios Web ocultando los detalles concretos de su consumo a través de tecnología WCF. Todos los repositorios de esta categoría heredan de la clase *WebServiceRepository*, que incorpora la estructura básica de funcionamiento de todos ellos. El resto de clases heredadas proporcionan servicios concretos del patrón *Repository* para la sincronización de datos, sincronización de fecha y hora, sincronización de entradas de seguimiento de entidades y autenticación de usuarios.

Adicionalmente se incluye en este diagrama las clases *WebServicesUnitOfWork* y *CompleteSyncUnitOfWork* que respectivamente actúan como clase base y concreta del patrón *UnitOfWork* en las operaciones de comunicación vía *Web Service* y sincronización.

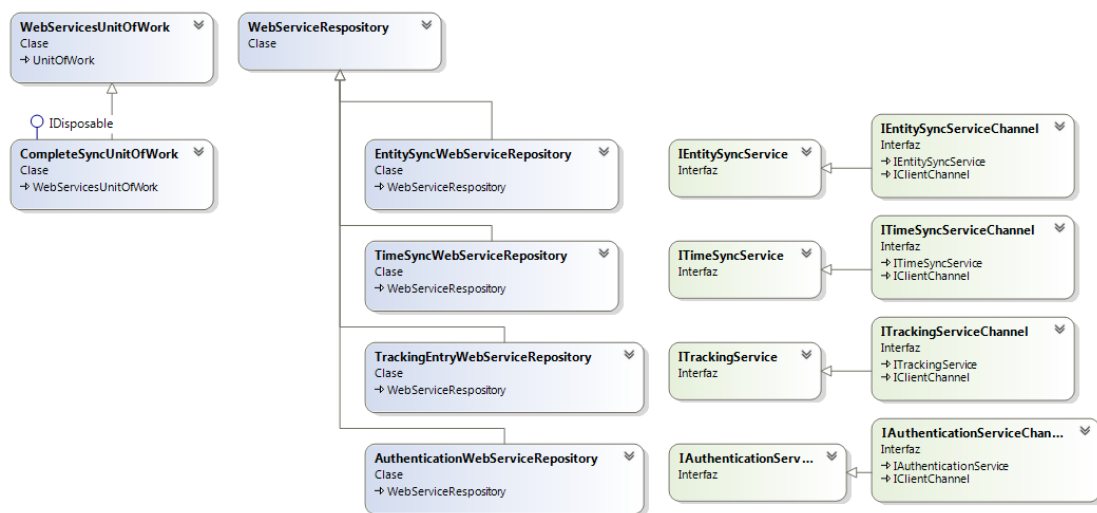


Ilustración 12 - Diagrama de clases de repositorios de acceso a Servicios Web

Las interfaces representadas junto a cada repositorio indican el contrato de comunicación del *Web Service* que implementan las clases *ServiceChannel*, y que a su vez actúan como puente de comunicación con cada Repositorio.

Diagrama de clases de la interfaz de usuario (View-ViewModel)

Por último se detallan las clases de control y coordinación (ViewModel) entre el modelo (lógica de negocio) y la vista (presentación) que responden al modelo de implementación MVVM, junto con las clases de servicio de aplicación de interfaz de usuario que permiten realizar la gestión de vistas en esta capa:

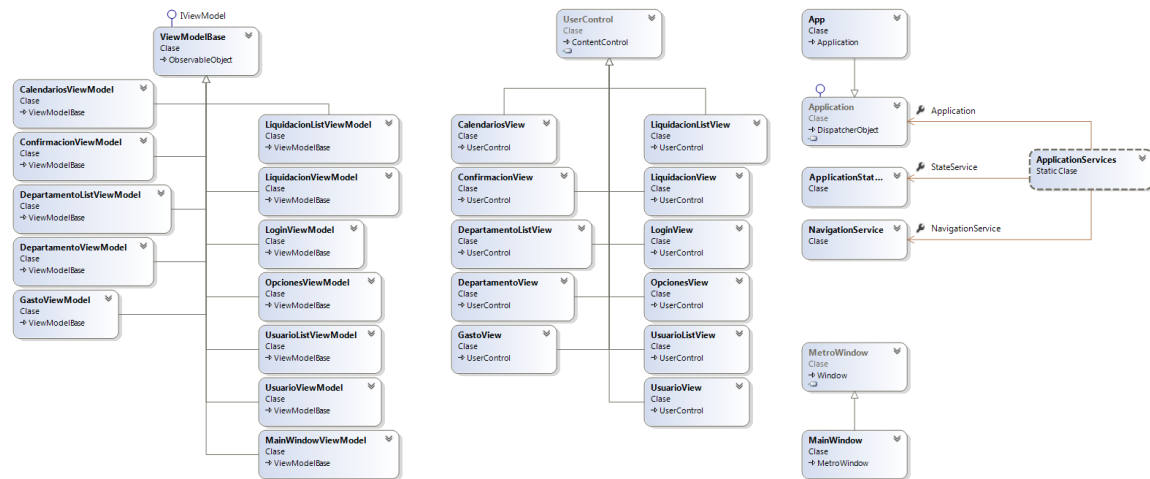


Ilustración 13 - Diagrama de clases de la interfaz de usuario (View-ViewModel)

4.4 Diagrama de secuencia

No se han definido.

4.5 Diagrama de estado

Estado de liquidación

Define los posibles estados en los que se puede encontrar una liquidación y las acciones que provocan la transición entre los citados estados.

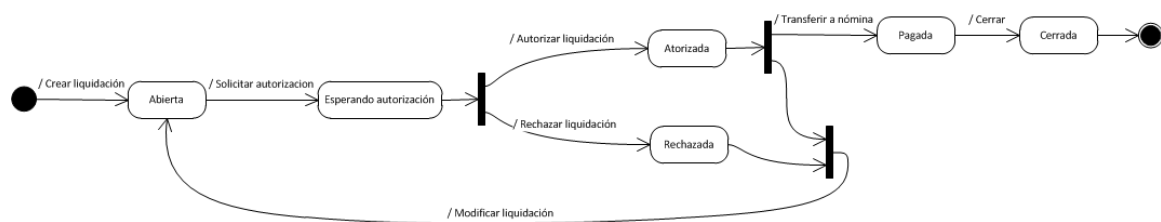


Ilustración 14 - Diagrama de estado para para la liquidación

Descripción de estados:

- **Abierta:** Es el estado en el que se encuentra una liquidación cuando se crea o cuando ha sido modificada por el usuario.
- **Esperando autorización:** Es el estado en el que se encuentra una liquidación cuando se solicita su autorización para proceder al envío a nómina.
- **Rechazada:** Cuando el autorizador rechaza el envío a nómina de una liquidación queda en este estado.
- **Autorizada:** Es el estado en el que queda una liquidación cuando el envío a nómina es autorizado por el autorizador.

- **Pagada:** Es un estado transitorio que utiliza el sistema para marcar las liquidaciones que ya han sido incluidas en el pago a nómina.
- **Cerrada:** Es el estado final al que pasa una liquidación cuando ya ha sido enviada al sistema de nóminas para su liquidación y se ha completado este proceso.

Descripción de transiciones:

- **Crear liquidación:** Se produce toda vez que el usuario consulta la lista de liquidaciones y no existe ninguna en estado abierta.
- **Solicitar autorización:** Se produce cuando el usuario solicita la autorización de la liquidación al responsable de su departamento para que el sistema pueda transferir los importes de los gastos a la nómina del usuario.
- **Rechazar liquidación:** Se produce cuando el autorizador rechaza la liquidación y no la autoriza por algún motivo.
- **Modificar liquidación:** Se produce toda vez que el usuario realiza cambios en la liquidación.
- **Autorizar liquidación:** Se produce cuando el autorizador aprueba la liquidación para su abono en nómina.
- **Transferir a nómina:** Se produce cuando el sistema lleva a cabo el proceso automático de notificación de gastos autorizados al sistema de nóminas.
- **Cerrar:** Se produce una vez que el sistema ha realizado con éxito la transferencia de nóminas.

4.6 Diseño de la interfaz de usuario

4.6.1 Apariencia y estilo (look & feel)

El estilo y diseño visual de la aplicación está basado en las Guías de Diseño de Interfaz de Usuario de Windows 8 (Microsoft GUX 8, 2013). Para lograr la apariencia de los controles de Windows 8, se utiliza una librería de uso libre denominada Mahapps.com (Mahapps.com, 2013).

4.6.2 Estructura de la vista

De forma general el diseño de una vista de la aplicación está compuesta por cuatro áreas principales:

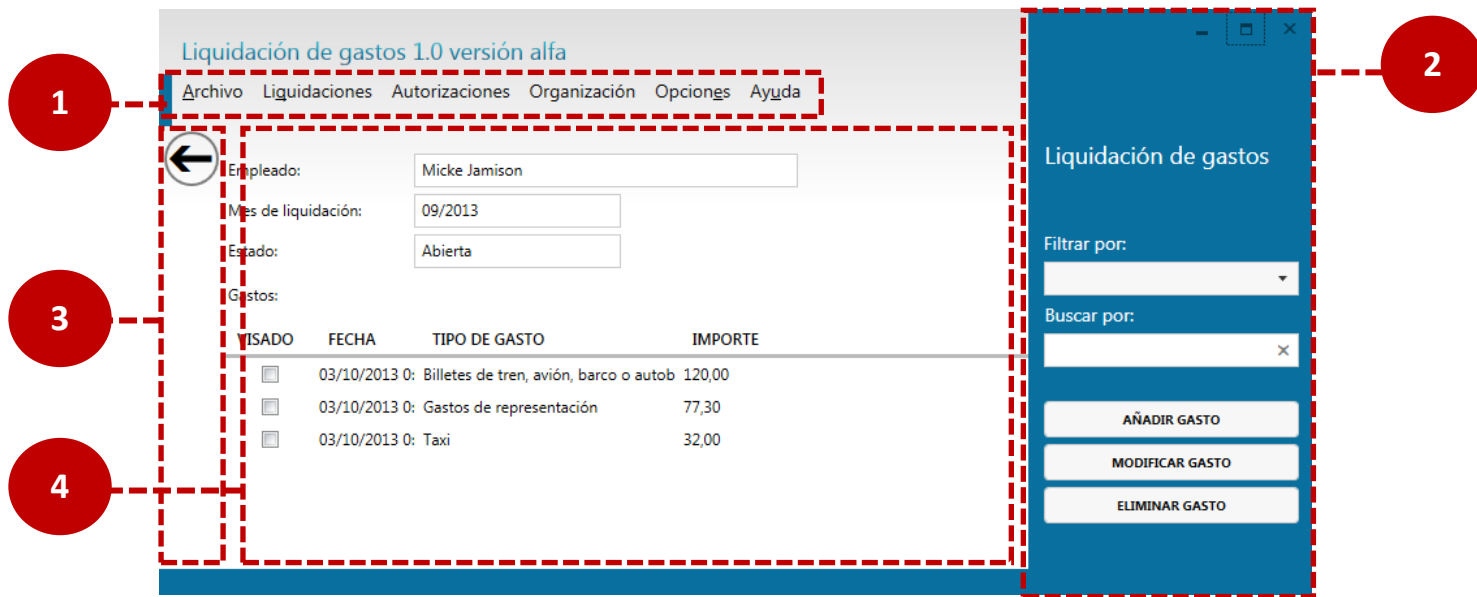


Ilustración 15 - Esquema general de diseño de una vista

1. **Área de comandos de aplicación:** Es la zona de la ventana principal donde se muestra el menú de la aplicación que proporciona acceso a las diferentes funcionalidades. Su apariencia varía en función del perfil de usuario y del punto de navegación en el que se encuentre. De las cuatro áreas es la única que está ligada a la ventana principal, el resto están ligadas a la vista que se muestra en cada momento.

2. **Área de comandos de contexto:** Es la zona de operación principal del usuario dentro de una vista. Ofrece comandos de filtrado y búsqueda de información donde es requerido, así como las acciones en forma de botón que pueden ser aplicadas.

3. **Área de cancelación:** Es una zona común al que ya están acostumbrados los usuarios de aplicaciones Windows 8. Su función principal es volver a la vista anterior.

4. **Área de trabajo:** Es el área donde se muestran los datos de operación de cada vista y se muestran los controles oportunos de acuerdo al objetivo de la vista.

4.6.3 Navegación

La interfaz de usuario debe ofrecer al usuario de un modo intuitivo de localizar la información y las acciones que desee realizar.

Las vistas mantienen una relación de jerarquía entre ellas, ya que no es posible acceder a un detalle de un gasto sin haber seleccionado previamente una liquidación de la lista de liquidaciones. Por lo tanto, para el diseño de la navegación se va a utilizar el patrón Jerárquico (Microsoft Nav Patt, 2013) que permite cumplir con los requisitos anteriormente expuestos en este documento.

4.6.4 Jerarquía de vistas

El siguiente diagrama jerárquico muestra los enlaces de navegación entre vistas y que justifican el diseño del sistema jerárquico implementado.

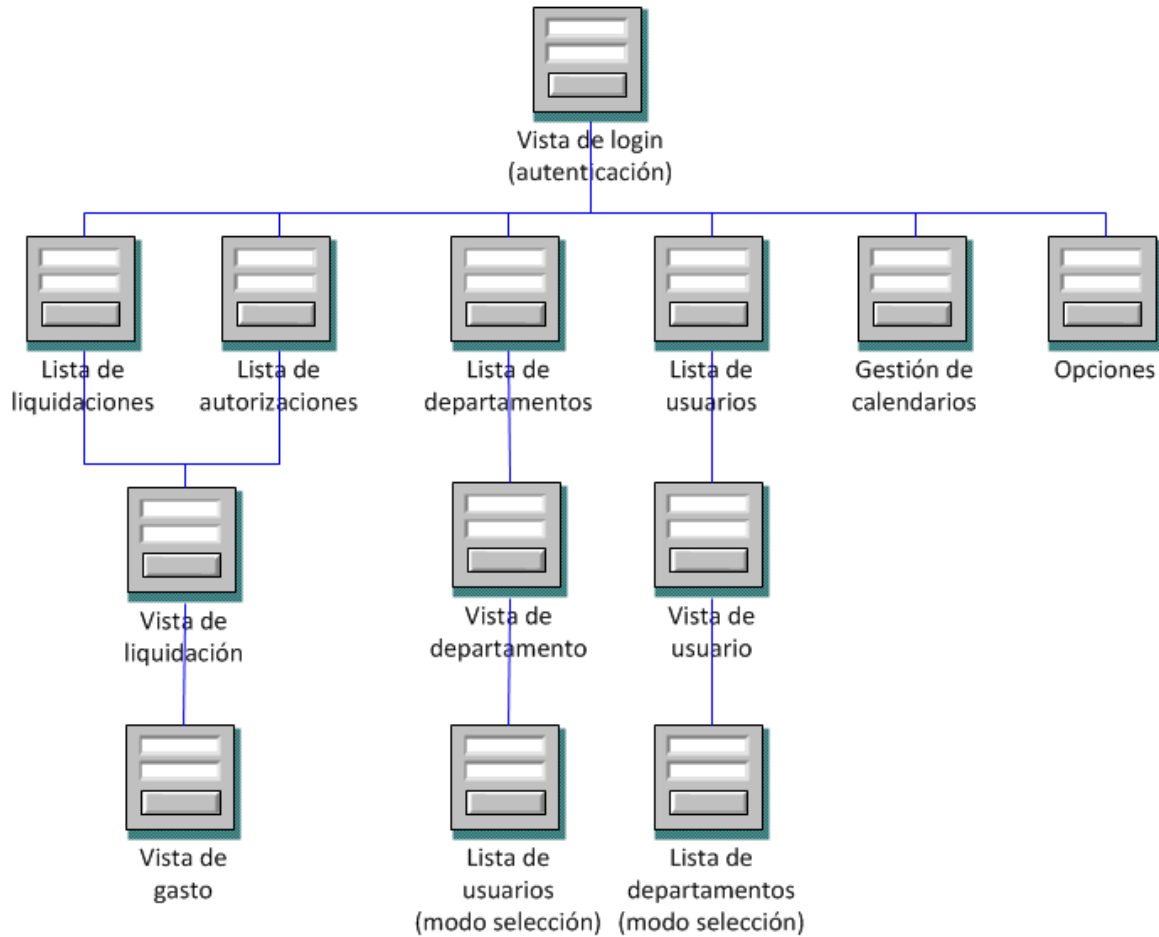


Ilustración 16 - Detalle de jerarquía de vistas de la aplicación

4.6.5 Descripción de vistas

Vista Login

La vista del proceso Login permite al usuario identificarse ante el sistema. Es un requisito necesario e indispensable antes de comenzar a operar.

Identificación

Identificador de usuario:

Contraseña:

identificar

Ilustración 17 - Detalle de la vista Login

La vista está compuesta por dos campos editables donde el usuario/autorizador/administrador introduce sus credenciales. Tras pulsar el botón “identificar” el sistema inicia el proceso de validación.

Vista Lista de liquidaciones

La vista ofrece una lista de todas las liquidaciones que pertenecen al usuario.

	AUTORIZADA	FECHA	ESTADO	IMPORTE TOTAL
<input type="checkbox"/>		01/01/2013 0:	Cerrada	
<input type="checkbox"/>		01/02/2013 0:	Cerrada	
<input type="checkbox"/>		01/03/2013 0:	Cerrada	
<input type="checkbox"/>		01/11/2013 0:	Abierta	

Lista de liquidaciones de gastos

Filtrar por:

Buscar por:

ABRIR

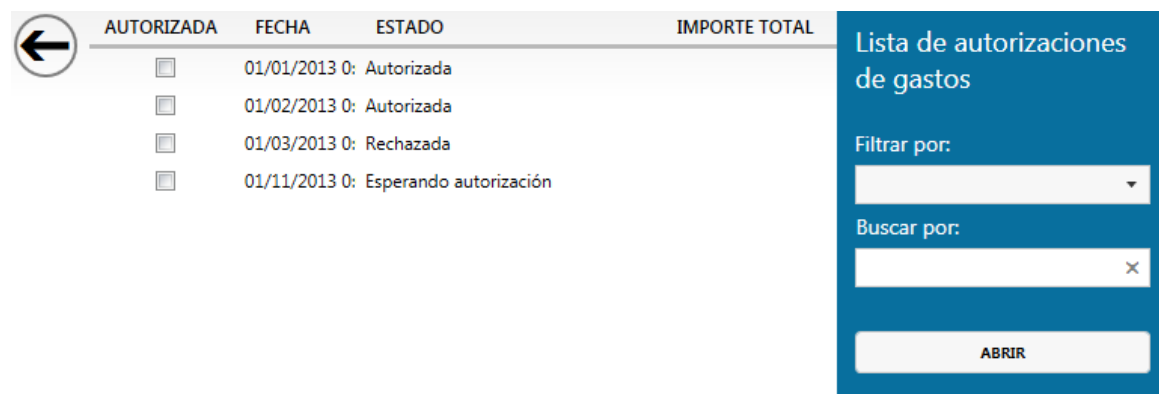
Ilustración 18 - Detalle de la vista Lista de liquidaciones

Los datos se muestran en formato tabular, de forma que el usuario puede navegar por la lista con rapidez. La lista ofrece funciones adicionales tales como la ordenación por columnas con solo seleccionar la que es de interés.

En el lado derecho se muestra una lista desplegable que permite filtrar de entre todas las liquidaciones solo aquellas en las que el estado es el seleccionado. También es posible filtrar liquidaciones que contengan conceptos o parte de conceptos que el usuario puede teclear en el cuadro “Buscar por:”.

Finalmente a través del botón “ABRIR”, el usuario puede acceder al detalle de la liquidación seleccionada.

Vista Lista de autorizaciones



AUTORIZADA	FECHA	ESTADO	IMPORTE TOTAL
<input type="checkbox"/>	01/01/2013 0:	Autorizada	
<input type="checkbox"/>	01/02/2013 0:	Autorizada	
<input type="checkbox"/>	01/03/2013 0:	Rechazada	
<input type="checkbox"/>	01/11/2013 0:	Esperando autorización	

Ilustración 19 - Detalle de la vista Lista de autorizaciones

La vista de lista de autorizaciones solo es accesible para el usuario Autorizador, y muestra un listado de todas las liquidaciones de los usuarios que dependen de los departamentos de los que el Autorizador es responsable. Desde esta vista solo es posible realizar filtrados de las liquidaciones y acceder a cualquiera de ellas a través del botón “ABRIR”.

Vista Liquidación



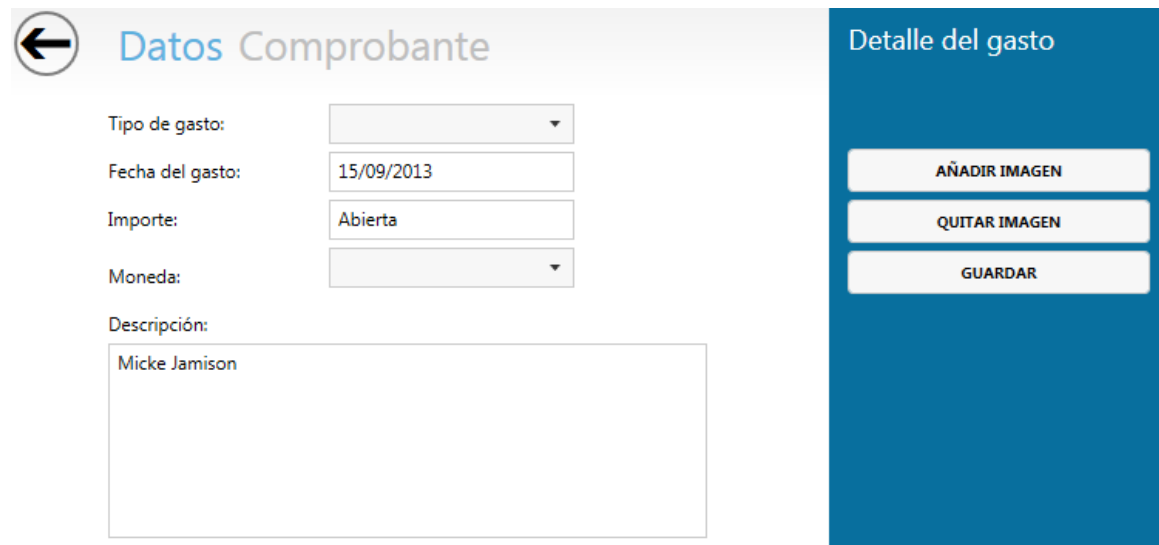
VISADO	FECHA	TIPO DE GASTO	IMPORTE
<input type="checkbox"/>	03/10/2013 0:	Billetes de tren, avión, barco o autob	120,00
<input type="checkbox"/>	03/10/2013 0:	Gastos de representación	77,30
<input type="checkbox"/>	03/10/2013 0:	Taxi	32,00

Ilustración 20 - Detalle de la vista Liquidación

La vista de liquidación ofrece el detalle de la liquidación de gastos de un usuario concreto, está compuesta de una cabecera informativa, un listado de los gastos incluidos en ella y los botones de operación y filtrado a la derecha.

Vista Gasto

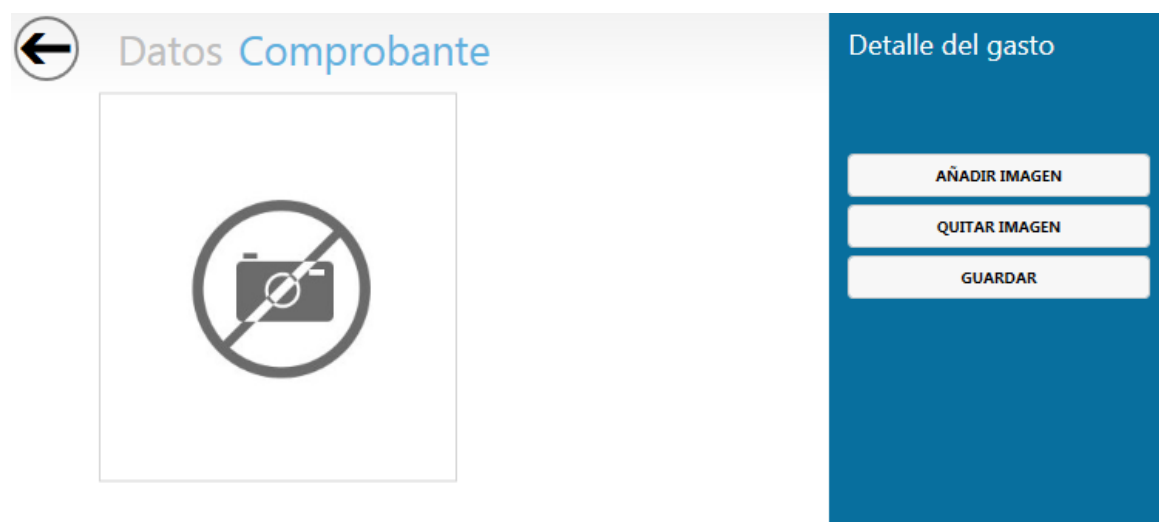
Accesible a través de la edición o creación de un gasto en la vista de liquidación, permite consultar o modificar sus datos.



The screenshot shows a web interface with two main sections. On the left, under the heading 'Datos Comprobante', there are several input fields: 'Tipo de gasto' (a dropdown menu), 'Fecha del gasto' (containing '15/09/2013'), 'Importe' (containing 'Abierta'), 'Moneda' (a dropdown menu), and 'Descripción' (a text area containing 'Micke Jamison'). On the right, under the heading 'Detalle del gasto', there are three buttons: 'AÑADIR IMAGEN', 'QUITAR IMAGEN', and 'GUARDAR'.

Ilustración 21 - Detalle de la vista Gasto

Su composición tiene dos áreas: la primera permite editar el tipo de gasto, fecha en el que se ha producido, el importe y divisa en que ha sido abonado y finalmente un texto libre. El segundo área permite la visualización e inserción de la imagen del comprobante del gasto a través de los botones de operación que figuran en el lado derecho como muestra la imagen:



This screenshot is similar to the previous one, but the 'Descripción' text area is replaced by a large square placeholder containing a camera icon with a diagonal slash through it, indicating that an image of the receipt should be uploaded. The 'Detalle del gasto' buttons remain on the right.

Ilustración 22 - Detalle de la vista Gasto (Comprobante)

Vista Lista de departamentos

Presenta una vista jerárquica de los departamentos definidos en la aplicación para el control y gestión de gastos.

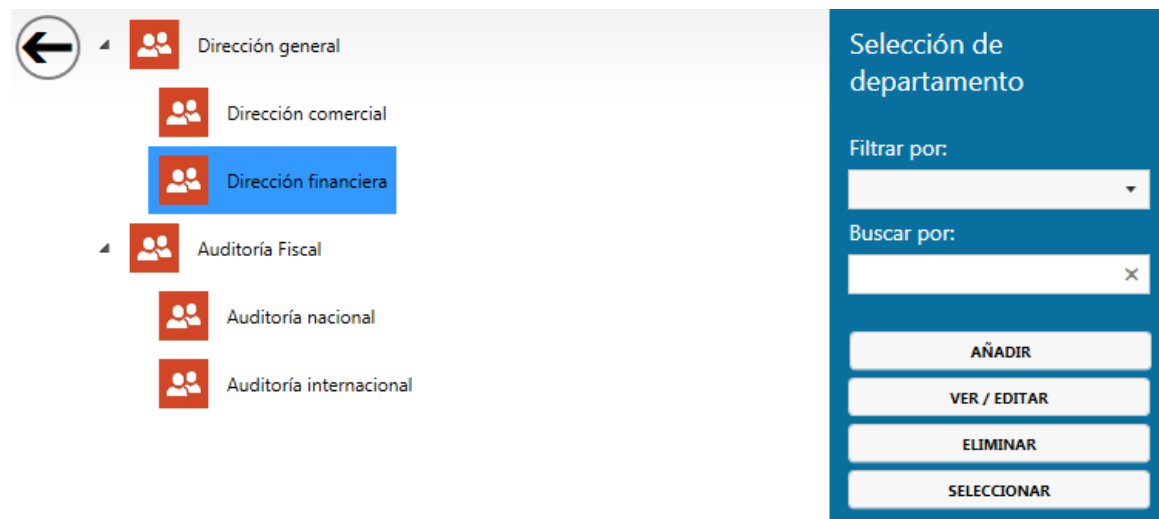


Ilustración 23 - Detalle de la vista Lista de departamentos

La vista mostrada en el detalle corresponde al modo de selección, donde se hace visible el botón “Seleccionar”. Esta operación no estará disponible en la vista cuando no opere en modo selección.

Vista Departamento

Se accede a través de la creación o edición de un departamento desde la vista anterior, y permite editar los detalles del mismo.

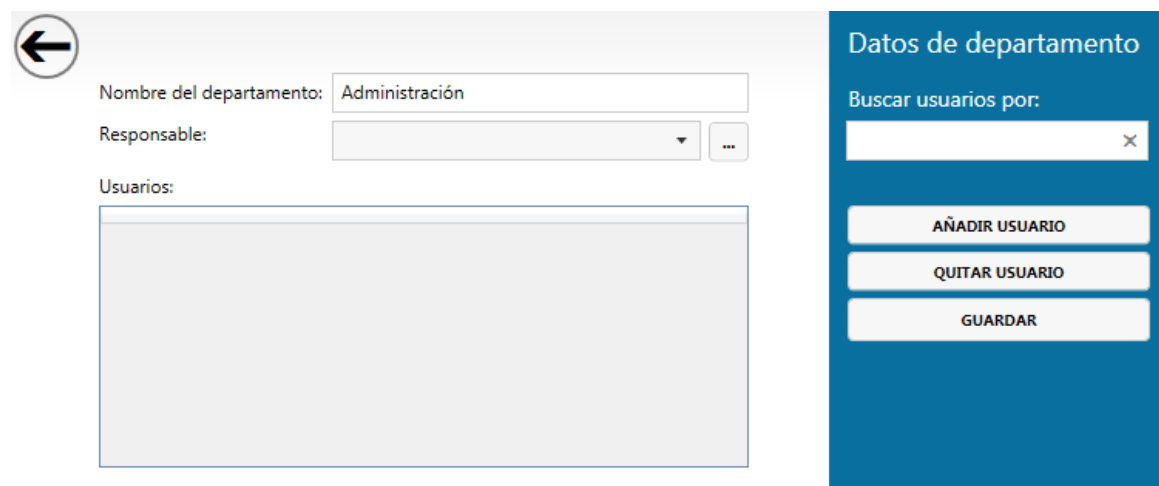


Ilustración 24 - Detalle de la vista Departamento

Vista Lista de usuarios

Similar a la vista de departamentos la siguiente vista permite realizar el mantenimiento de los usuarios de la aplicación a través de los controles de operación y filtrado en el lado derecho de la vista.

NOMBRE COMPLETO	DEPARTAMENTO
Jamie Oliver	Cocina
Tim Johanson	Administración
Luis Fernández	Logística
Michael Lewis	Internacional
Alberto Ruiz	Consultoría

Selección de usuario

Filtrar por:

Buscar por:

CREAR

VER/EDITAR

ELIMINAR

SELECCIONAR

Ilustración 25 - Detalle de la vista Lista de usuarios

La vista mostrada en el detalle corresponde al modo de selección, donde se hace visible el botón “Seleccionar”. Esta operación no estará disponible en la vista cuando no opere en modo selección.

Vista Usuario

Permite editar los datos relativos a un usuario, y es accesible a través de las operaciones de creación o edición desde la vista de lista de usuarios.

Identificador:

Contraseña:

Contraseña (repetir):

Nombre completo:

Departamento:

Dirección de email:

Notificar por email:

Datos del usuario

ACEPTAR

Ilustración 26 - Detalle de la vista Usuario

Vista Gestión de calendarios

La vista permite realizar el mantenimiento de calendarios de días festivos, que posteriormente la solución utilizará para el cálculo de liquidación de dietas.



Ilustración 27 - Detalle de la vista Gestión de calendarios

Las funciones que la vista ofrece al administrador son:

- Crear: Establecer un nuevo calendario que solamente marca como días festivos los fines de semana.
- Eliminar: Borrar de forma permanente un calendario previamente creado. Esta operación no se podrá realizar si un calendario ya ha sido asignado a un usuario o ha sido utilizado para realizar el cálculo de dietas de una liquidación.
- Guardar: Salva de forma permanente los cambios realizados en un calendario.
- Restablecer: Establecer como laborales todos los días del año del calendario, excepto los fines de semana.
- Duplicar: Crear una copia de un calendario.

5 Implementación

5.1 Software utilizado

El proyecto encargado por el cliente ha sido desarrollado según sus indicaciones¹ sobre la plataforma .NET, por lo que al menos se han seleccionado para su aplicación las siguientes tecnologías descritas:

Windows Presentation Foundation (WPF)

Windows Presentation Foundation (Microsoft, 2013) es un sistema de presentación de la última generación, para crear aplicaciones cliente de Windows que proporcionan una experiencia impactante para el usuario desde el punto de vista visual. Con WPF, es posible crear una amplia gama de aplicaciones independientes y hospedadas en navegador.

El núcleo de WPF es un motor de representación basado en vectores e independiente de la resolución que se crea para sacar partido del hardware de gráficos moderno. WPF extiende el núcleo con un conjunto completo de características de desarrollo de aplicaciones que incluye Lenguaje XAML, controles, enlace de datos, diseño, gráficos 2D y 3D, animación, estilos, plantillas, documentos, multimedia, texto y tipografía. WPF se incluye en Microsoft .NET Framework, de modo que es posible compilar aplicaciones que incorporen otros elementos de la biblioteca de clases de .NET Framework.

Language-Integrated Query (LINQ)

LINQ (Microsoft, 2013) es una innovación introducida en .NET Framework versión 3.5 que elimina la distancia que separa el mundo de los objetos y el mundo de los datos.

Tradicionalmente, las consultas con datos se expresan como cadenas sencillas, sin comprobación de tipos en tiempo de compilación. Además, es necesario aprender un lenguaje de consultas diferente para cada tipo de origen de datos: bases de datos SQL, documentos XML, servicios Web diversos, etc.

LINQ convierte una consulta en una construcción de lenguaje de primera clase en C# y Visual Basic. Las consultas se escriben para colecciones de objetos fuertemente “tipadas”, utilizando palabras clave del lenguaje y operadores con los que se está familiarizado.

LINQ permite escribir consultas en Visual Basic o en C# con bases de datos SQL Server, documentos XML, conjuntos de datos ADO.NET y cualquier colección de objetos que admita la interfaz IEnumerable o la interfaz genérica IEnumerable<T>. También se ha previsto la compatibilidad de LINQ con ADO.NET Entity Framework, por lo que existen fabricantes que se encuentran escribiendo proveedores LINQ para muchos servicios Web y otras implementaciones de bases de datos.

El único requisito para la utilización de LINQ que el proyecto precisa es que esté orientado a .NET Framework 3.5 o posterior.

¹ Ver Requisito no funcional RN004

Entity Framework (EF)

Entity Framework (Microsoft, 2013) es un sistema de asignación objeto-relacional que permite a los desarrolladores de .NET trabajar con datos relacionales usando objetos específicos del dominio. Elimina la necesidad de la mayor parte del código de acceso a datos que los desarrolladores suelen tener que escribir, acelerando la implementación de la solución.

En la presente solución la aplicación de EF permite crear elementos de acceso a datos con un reducido número de líneas de código, lo cual facilitará el mantenimiento y su utilización.

SQL Server

SQL Server (Microsoft, 2013) es un motor de base de datos propiedad de Microsoft que proporciona servicios (entre otros) de almacenamiento de modelos relacionales, multidimensionales y dispone de un sistema de escalado que facilita la gestión de volúmenes de datos desde dispositivos móviles hasta sistemas basados en la nube (Cloud Computing).

En el proyecto actual la persistencia de datos se lleva a cabo con servidores SQL Server, por su perfecta integración con la tecnología .NET a través de Entity Framework y LINQ, y su sistema de escalabilidad.

Windows Communication Foundation (WCF)

Windows Communication Foundation (Microsoft, 2013) es un marco de trabajo para la creación de aplicaciones orientadas a servicios. Con WCF, es posible enviar datos como mensajes asíncronos de un extremo de servicio a otro. Un extremo de servicio puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación. Un extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio. Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios.

La aplicación de WCF en el proyecto permite flexibilizar la fachada de la solución servidora al tiempo que proporciona canales de comunicación con el cliente.

Simple Mail Transfer Protocol (SMTP)

El SMTP (Wikipedia, 2013) o Protocolo para la Transferencia Simple de Correo electrónico, es un protocolo de la capa de aplicación. Protocolo de red basado en texto, utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA, teléfonos móviles, etc.). Está definido en el RFC 2821 y es un estándar oficial de Internet.

El funcionamiento de este protocolo se da en línea, de manera que opera en los servicios de correo electrónico. Sin embargo, este protocolo posee algunas limitaciones en cuanto a la recepción de mensajes en el servidor de destino (cola de mensajes recibidos). Como

alternativa a esta limitación normalmente se asocia este protocolo con otros, como el POP o IMAP, otorgando a SMTP la tarea específica de enviar correo, y el de recibirlos a los otros protocolos antes mencionados (POP O IMAP).

En la presente solución *software* se utilizar el protocolo SMTP para el envío de correo electrónico a un servidor de correo remoto, sin necesidad de recibir correo, por lo tanto las limitaciones del protocolo SMTP no son aplicables en esta situación.

Para poder operar con las funciones de SMTP se requiere .NET Framework 2.0 o posterior.

5.2 Capas de la aplicación

El siguiente diagrama representa el despliegue de componentes por capas sobre la infraestructura hardware de la solución de liquidación de gastos:

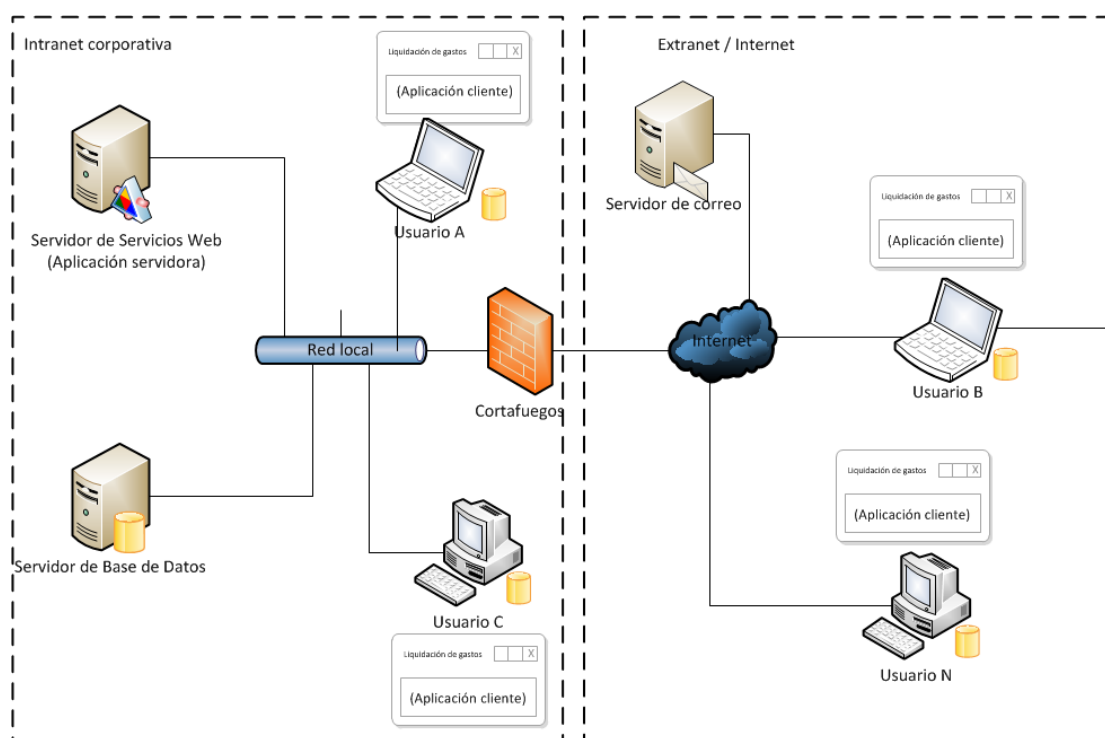


Ilustración 28 - Diagrama del modelo físico

El detalle asignación de componentes se recoge en la siguiente tabla:

Capa lógica	Computadora	Ensamblado	Descripción
Interfaz de usuario	Usuario A, B, C..N	LiquidacionGastos.exe	Programa cliente
Lógica de negocio	Usuario A, B, C..N	LiquidacionGastosBusiness.dll	Reglas de negocio
Lógica de negocio	Usuario A, B, C..N	LiquidacionGastos.UnitsOfWork.dll	Transacciones
Persistencia	Usuario A, B, C..N	LiquidacionGastos.Repositories.dll	Acceso a datos
Persistencia	Usuario A, B, C..N	Instancia de SQL Sever	Modelo de datos

Capa lógica	Computadora	Ensamblado	Descripción
Seguridad	Usuario A, B, C..N	LiquidacionGastos.Security.dll	Seguridad de usuario y procesos
Entidades	Usuario A, B, C..N	LiquidacionGastos.Models.dll	Entidades POCO
Infraestructura	Usuario A, B, C..N	LiquidacionGastos.Framework.dll	Funciones básicas del sistema
Interfaz de usuario	Servidor de Servicios Web	LiquidacionGastosService.exe	Programa Servicio Windows
Lógica de negocio	Servidor de Servicios Web	LiquidacionGastosBusiness.dll	Reglas de negocio
Lógica de negocio	Servidor de Servicios Web	LiquidacionGastos.UnitsOfWork.dll	Transacciones
Persistencia	Servidor de Servicios Web	LiquidacionGastos.Repositories.dll	Acceso a datos
Persistencia	Servidor de Base de Datos	Instancia de SQL Sever	Modelo de datos
Seguridad	Servidor de Servicios Web	LiquidacionGastos.Security.dll	Seguridad de usuario y procesos
Entidades	Servidor de Servicios Web	LiquidacionGastos.Models.dll	Entidades POCO
Infraestructura	Servidor de Servicios Web	LiquidacionGastos.Framework.dll	Funciones básicas del sistema

6 Evaluación de costes

Para la evaluación económica del presente proyecto sólo se estiman los costes de recursos humanos, ya que tanto los costes de licencias como del hardware necesario serán soportados directamente por el cliente.

Para calcular los costes de los recursos humanos, se toman como referencia los costes de las tarifas de mercado para esta tecnología y tipo de proyecto, se establecen los precios de los perfiles y se aplican en la tabla de presupuesto.

Dado que el proyecto tiene una dedicación continuada de tres horas por jornada de lunes a domingo, se realiza un prorrateo de tres horas en la columna de tarifas.

Tabla 11 - Precios de perfiles

Perfil	Coste / hora	Coste / jornada	Coste / jornada (3h)
Jefe de proyecto	48 €	384 €	144 €
Analista	36 €	288 €	108 €
Analista programador	24 €	192 €	72 €

Tabla 12 - Presupuesto de desarrollo del proyecto *software* de liquidación de gastos

Cód.	Actividad	Estimación (jornadas)	Perfil	Coste (euros)
01	Inicio de proyecto	0		
02	Plan de trabajo	12	Jefe de proyecto	1.718,00 €
03	Análisis			
03.01	Análisis Funcional	8	Analista	864,00 €
04	Diseño			
04.01	Diseño Técnico	8	Analista programador	576,00 €
04.02	Prototipo	12	Analista programador	864,00 €
05	Implementación			
05.01	Implementación cliente	12	Analista programador	864,00 €
05.02	Implementación servidor	15	Analista programador	1.080,00 €
06	Pruebas			
06.01	Pruebas de integración	6	Analista programador	432,00 €
06.02	Pruebas de sistema	8	Analista	864,00 €
07	Documentación			
07.01	Manual de instalación	2	Analista programador	144,00 €
07.02	Manual de usuario	5	Analista	540,00 €
07.03	Memoria	15	Jefe de proyecto	2.160,00 €
07.04	Presentación Virtual	13	Jefe de proyecto	1.872,00 €
08	Final de proyecto			
Total		116		11.988,00 €

Por lo tanto el coste total del desarrollo del proyecto es de **11.988,00 €**.

7 Trabajo futuro

Se diferencian dos áreas de evolución del proyecto en función de su naturaleza, a saber:

- Evolución funcional: Relacionada con los aspectos directamente relacionados con la liquidación de gastos
- Evolución tecnológica: Aspectos puramente técnicos que pueden mejorar el funcionamiento, mantenimiento y aseguramiento de la inversión.

7.1 Evolución funcional

Plantillas de liquidación

Durante las pruebas del producto se ha comprobado que en la operativa normal, existen casos de liquidación de gastos que se repiten en el tiempo. Por ejemplo: un consultor que viaja semanalmente a Holanda durante un proyecto de seis meses liquida durante el proyecto más de un 80% de gastos similares o equivalentes –billetes de avión, taxis, dietas, descripciones, etc– por esta razón y en la misma proporción podrá reducir el tiempo de creación de sus liquidaciones si el sistema le permite reutilizar datos de liquidaciones anteriores.

Entre las opciones que parecen más adecuadas se encuentran:

- Crear plantillas de liquidaciones ó gastos a partir de una liquidación existente, de tal forma que la plantilla pueda utilizar para introducir los nuevos gastos que ya figuran definidos en la plantilla.
- Copiar liquidaciones ó gastos directamente en una nueva liquidación. Esta variación permitiría al usuario crear un gasto o una liquidación copiando los datos de una previamente seleccionada.
- Utilizar un repositorio de plantillas corporativo, que evite al usuario introducir determinados datos (destinos, tipos de gastos, moneda utilizada, etc) cuando estos quedan concretados por el tipo de o finalidad de la liquidación.

Reglas de liquidación de gastos

Del mismo modo que en la realización de liquidaciones de gastos, el autorizador debe supervisar en la actualidad todas las informaciones introducidas por sus empleados y verificar que se cumplan las limitaciones que deba o considere imponer.

Si determinadas restricciones o cumplimientos son requeridos por un autorizador pero no lo son por la normativa de la compañía el autorizador deberá realizar estas comprobaciones manualmente cada vez. Con el objetivo de reducir su tiempo de dedicación a esta actividad, el sistema podría facilitar el establecimiento de normativas o reglas de validación para un autorizador, departamento ó usuario concreto, tales como fijación de importes máximos por

gasto, liquidación ó actividad, intervalos de fecha permitidos para realizar liquidaciones, asignación de costes compartidos entre diferentes departamentos, etc.

En el caso más avanzado, el propio sistema podría autorizar o rechazar de forma automática determinadas liquidaciones si se cumplen o no las reglas de autorización establecidas.

Integración con agencias de viajes u organizaciones de gestión de los viajes o gastos

Es habitual que los viajes corporativos se reserven y planifiquen a través de un departamento, agencia ó incluso empresa externa. Si esta situación ocurre de forma habitual, sería razonable considerar la posibilidad de integrar el sistema de liquidación con el software de la agencia de viajes, de tal forma que fuese posible planificar los gastos con anticipación asociándolos a las reservas de vuelo, coche, hotel, etc. Una vez que los gastos han sido realizados, la aplicación permitiría realizar la liquidación introduciendo automáticamente los gastos y costes previamente planificados.

Integración con entidades de información económica

La aplicación utiliza los cambios de moneda o cotizaciones para realizar los cálculos de cada liquidación en el momento que se produjo el gasto. Mantener actualizada esta información supone un trabajo diario que mermará los beneficios económicos obtenidos por reducción de actividades en la liquidación de gastos. Por lo tanto, se sugiere una integración con servicios online existentes para actualizar diariamente desde el servidor las cotizaciones de las divisas utilizadas por el sistema.

Informes estadísticos de gastos

La información recogida por el sistema a lo largo del tiempo, puede proveer información de valor para la empresa tales como: áreas de mayor gasto, tipos de transporte más utilizados, destinos o áreas geográficas de mayor servicio prestado, etc.

Estas informaciones pueden ser extraídas con técnicas de análisis de datos (cubos multidimensionales) disponibles en el sistema de base de datos SQL Server, e implementar en la aplicación informes que permitan a la compañía realizar un gasto más óptimo en los desplazamientos de sus empleados o ayudar a planificar con mayor acierto los presupuestos relacionados con estas actividades en los siguientes años.

Interfaz Multi-idioma

Aunque el cliente no ha solicitado una versión multi-idioma para la presente versión software, parece razonable plantear una evolución multi-idioma en una compañía que opera en todo el mundo y que es susceptible de captar empleados que no conozcan la lengua española.

7.2 Evolución tecnológica

Interfaz web, móvil y Windows 8.x

El proyecto se ha implementado con una solución para la navegación entre vistas siguiendo las líneas de diseño del patrón MVVM. Esta aproximación se ha realizado pensando en una posible evolución de la capa de interfaz de usuario a otros entornos, por esta razón, se ha llevado a cabo con una implementación a medida no bloqueante, ya que el bloqueo de ventanas es una característica habitualmente que no está disponibles de forma nativa en otras plataformas como en el entorno de móviles o web.

Sincronización de datos

La versión actual de la solución incorpora un algoritmo de sincronización bidireccional de datos entre el cliente y el servidor personalizado. Aunque existen técnicas muy avanzadas para realizar esta actividad como las características de SQL SERVER, la casuística que se plantea en los requisitos de la solución *software* sugiere la aplicación de técnicas ligeras y estándares de comunicación, tales como los Servicios Web, que ha motivado el diseño particularizado de esta función entre el cliente y el servidor.

En el diseño actualmente implementado las entradas generadas en la tabla de seguimiento (*tracking*) crecen exponencialmente con cada instancia de entidad, ya que por cada modificación de una entidad se añade una entrada de seguimiento para la misma, resultando después de varias modificaciones, diferentes entradas de seguimiento para la misma instancia de entidad, lo que provoca una caída de rendimiento en la operación de sincronización a medida que las entidades van siendo modificadas además del elevado volumen que la tabla puede alcanzar.

Puesto que el objetivo final es realizar una sincronización de los datos actuales –no históricos– el algoritmo solo debería mantener un único registro de seguimiento (*tracking*) de cada instancia de entidad, de modo que exista una relación 1:1. Por lo tanto en cada operación de actualización de la entidad el registro de seguimiento también será actualizado en lugar de realizar otra inserción de otro registro de seguimiento.

Introducción del motor de Workflow WWF (Windows WorkFlow Foundation)

Un flujo de trabajo es un conjunto de unidades elementales llamadas actividades que están almacenadas como un modelo que describe un proceso real. Los flujos de trabajo proporcionan una manera de describir el orden de ejecución y las relaciones de dependencia entre las partes de trabajo de ejecución corta o prolongada.

La integración en el presente proyecto de WWF (Microsoft, 2013) puede prestar un entorno de trabajo para dar soporte al flujo de trabajo para la autorización de las hojas de liquidaciones, que aumentaría la escalabilidad del sistema y la robustez en un futuro.

Mejoras de seguridad

Tanto los despliegues como las comunicaciones vía Web pueden ser mejorados en aras de evitar fugas de información o intromisiones por parte de agentes o entidades no deseadas. En este sentido, se recomienda la introducción de certificados digitales en las comunicaciones del servidor que ofrece los Servicios Web.

Idealmente el software cliente que sea desplegado en formato de software de instalación también puede utilizar un certificado de autenticidad que evitará una posible impostura del software de liquidación de gastos desplegado.

Integración con directorio LDAP Corporativo

El sistema de liquidación de gastos incorpora un sistema de mantenimiento de usuarios que incluye la gestión de sus credenciales con la aplicación. Sin embargo, cada vez es más común que las empresas cuenten con un directorio LDAP (Wikipedia, 2013) que permite centralizar las credenciales en un único punto, reforzando la seguridad y abaratando el mantenimiento de las credenciales de usuarios corporativos.

8 Conclusiones

La elaboración del presente proyecto ha permitido obtener una experiencia completa en el desarrollo de un producto software para dar solución a una problemática empresarial determinada como es la gestión y liquidación de gastos de personal de una compañía.

El proyecto se ha realizado siguiendo una metodología en cascada con prototipado. Puesto que el tamaño del software puede ser considerado pequeño, los riesgos identificados y su corrección ha sido posible, en algunos casos, con un esfuerzo y tiempo reducido. Sin embargo, se ha constatado que otras aproximaciones metodológicas tales como los modelos incrementales y los modelos en espiral ofrecen una ventaja adicional en la identificación temprana de riesgos y su tratamiento.

Respecto a las técnicas de análisis y diseño, se verifica que una aproximación orientada a objetos permite realizar un software con índices de reutilización y mantenibilidad mucho mayores que los que cabrían esperar frente a técnicas estructuradas y procedurales. Probablemente esta diferenciación ha resultado más notoria en la implementación de la primera versión del software, donde se ha precisado realizar el desarrollo de un elevado número de componentes para obtener un primer prototipo con pocas funciones. A modo de ejemplo se cita que la implementación del proceso de autenticación (login), requiere la implementación de: navegación y estructura de la interfaz de usuario, del modelo de entidades, modelo de datos, sistema de consulta etc. No obstante, el elevado volumen de desarrollo inicial ha supuesto obtener mayores tasas de reutilización y rapidez en la implementación de funciones posteriores, debido a que ya existían otras funciones y componentes elementales requeridos.

Relacionado con las actividades de diseño e implementación cabe destacar la aplicación y beneficios obtenidos de patrones. Estas especificaciones han supuesto una importante guía para la resolución de problemas conocidos en la industria de la ingeniería del software, al tiempo que como se ha comprobado son una garantía de éxito, pues en la mayoría de los casos constituyen soluciones probadas y maduras a problemas concretos. A modo de ejemplo se menciona por su relevancia en este proyecto, el patrón *MVVM*, *Repository*, *Observer*, *Factory* o el patrón de arquitectura Multicapa (*Multi-layer*) entre muchos otros.

Otro punto a reseñar es el relacionado con el tratamiento y sincronización de datos. Desde el enfoque tecnológico la utilización de elementos de comunicación como los Servicios Web y el modelo *RESTful* constituyen unas potentes herramientas para el intercambio de información entre aplicaciones, ya sean tanto homogéneas como heterogéneas, y sus beneficios quedan patentes en este proyecto frente al modelo de comunicación basado en RPC. De hecho, uno de los aspectos más destacables, tal y como indica Aaron Skonnard (Skonnard, 2008), es que incluso el diseño de la signature de las funciones de un Servicio Web RESTful sigue la idea de Interfaz Uniforme (*Uniform Interface*), una filosofía muy diferente al modelo basado en componentes RPC cuyos nuevos componentes suelen introducir nuevas interfaces – y conjunto de métodos – para el acceso a sus funciones.

Desde el punto de vista funcional la sincronización de datos obliga a tener presente el tiempo de vida del dato y su calidad entre otros factores, tales como el momento de la sincronización, el orden de transferencia, los conflictos de identidades, etc. En el proyecto de Liquidación de gastos se ha desarrollado un modelo de sincronización de datos basado en el seguimiento de cambios y edad de cada entidad para establecer la validez frente a colisiones entre el sistema cliente y servidor. Aunque en el proyecto se ha incorporado una primera aproximación de este algoritmo de sincronización de datos, se ha sugerido una evolución del procedimiento para mejorar el volumen de datos gestionados, la cantidad de datos transferidos, la velocidad de sincronización y la carga añadida para el sistema por la ejecución de este proceso. Se considera relevante mencionar el papel desempeñado en este apartado las claves primarias de tipo GUID (*Global Unique Identifier*) (Wikipedia, 2013), que asegurando un nivel de colisiones prácticamente inexistente permiten agilizar la creación de instancias de nuevas entidades de forma descentralizada y su posterior sincronización.

En conclusión, el proyecto Liquidación de gastos desarrollado en este Trabajo Fin de Carrera ha sido en sí mismo una vasta fuente de experiencias e información que ha permitido explorar muchas de las situaciones que se producen habitualmente en los proyectos de ingeniería de *software*, y ha permitido entrenar las técnicas y prácticas habituales para su gestión, partiendo desde las labores de gestión, valoración y seguimiento de proyectos hasta las actividades más técnicas de programación pasando por el análisis y diseño, todo ello en un marco temporal reducido pero enriquecedor.

9 Bibliografía

Cash Trainers. (2013). *Liquidación gastos de viaje con excel*. Recuperado el 25 de septiembre de 2013, de <http://cashtrainers.com/project/liquidacion-gastos-de-viaje-en-excel>

Fernández, J. M. (27 de julio de 2010). *Adminstración de riesgos*. Recuperado el 29 de septiembre de 2013, de http://www.uv.mx/personal/jfernandez/files/2012/03/3_Riesgos2012.pdf

Fowler, M. (2013). *Facade*. Recuperado el 12 de octubre de 2013, de P of EAA Catalog: <http://martinfowler.com/eaCatalog/remoteFacade.html>

Fowler, M. (2013). *Observer Pattern*. Recuperado el 12 de octubre de 2013, de <http://martinfowler.com/eaDev/MediatedSynchronization.html>

Fowler, M. (2013). *State Machine*. Recuperado el 8 de octubre de 2013, de <http://martinfowler.com/dslCatalog/stateMachine.html>

Fowler, M. (2013). *Unit Of Work Pattern*. Recuperado el 9 de octubre de 2013, de <http://www.martinfowler.com/eaCatalog/unitOfWork.html>

Karyelin, E. (18 de diciembre de 2011). *Análisis de riesgos de un proyecto*. Recuperado el 29 de septiembre de 2013, de <http://www.eoi.es/blogs/estefanykaryelindeaza/2011/12/18/analisis-de-riesgo-de-un-proyecto/>

Mahapps.com. (2013). *Mahapps.com*. Recuperado el 2013 de octubre de 11, de <http://mahapps.com/MahApps.Metro/>

Microsoft. (2013). *Entity Framework*. Recuperado el 24 de septiembre de 2013, de <http://msdn.microsoft.com/es-es/data/ef.aspx>

Microsoft. (2013). *LINQ (Language-Integrated Query)*. Recuperado el 25 de septiembre de 2013, de <http://msdn.microsoft.com/es-es/library/bb397926.aspx>

Microsoft. (2013). *SQL Server*. Recuperado el 25 de septiembre de 2013, de <http://www.microsoft.com/es-es/sqlserver/product-info/overview-capabilities.aspx>

Microsoft. (2013). *Windows Communication Foundation*. Recuperado el 24 de septiembre de 2013, de <http://msdn.microsoft.com/es-es/library/dd456779.aspx>

Microsoft. (2013). *Windows Presentation Foundation*. Recuperado el 21 de septiembre de 2013, de <http://msdn.microsoft.com/es-es/library/ms754130.aspx>

Microsoft. (2013). *Windows Workflow Foundation*. Recuperado el 22 de septiembre de 2013, de <http://msdn.microsoft.com/es-es/library/dd489441.aspx>

Microsoft GUX 8. (2013). *Guías de Diseño de Interfaz de Usuario de Windows 8*. Recuperado el 2013 de octubre de 11, de <http://msdn.microsoft.com/en-us/library/windows/apps/hh465424.aspx>

Microsoft MSDN. (2013). Recuperado el 8 de octubre de 2013, de Repository Pattern: <http://msdn.microsoft.com/en-us/library/ff649690.aspx>

- Microsoft MSDN. (2013). *Service Locator*. Recuperado el 8 de octubre de 2013, de <http://msdn.microsoft.com/en-us/library/ff648968.aspx>
- Microsoft Nav Patt. (2013). *Windows Store Apps - Navigation patterns*. Recuperado el 19 de octubre de 2013, de <http://msdn.microsoft.com/library/windows/apps/hh761500.aspx>
- SAP. (2013). *SAP*. Recuperado el 20 de septiembre de 2013, de <http://www.sap.com/>
- SAP Gastos. (2013). *SAP Gastos*. Recuperado el 25 de septiembre de 2013, de http://help.sap.com/saphelp_470/helpdata/es/b6/cd976f49e911d189060000e8322f96/content.htm
- Skonnard, A. (2008). *A guide to designing and building RESTful Web Services with WCF 3.5*. Recuperado el 13 de noviembre de 2013, de <http://msdn.microsoft.com/en-us/library/dd203052.aspx>
- Smith, J. (Febrero de 2009). *Aplicaciones WPF con el patrón de diseño Model-View-ViewModel*. Recuperado el 23 de septiembre de 2013
- Wiki EOI. (2013). *Definición de costes directos y costes indirectos*. Recuperado el 27 de septiembre de 2013, de http://www.eoi.es/wiki/index.php/Costes_directos_e_indirectos_en_Finanzas#Definici.C3.B3n_de_costes_directos_y_costes_indirectos
- Wikipedia. (2013). *Arquitectura cliente-servidor*. Recuperado el 27 de septiembre de 2013, de <http://es.wikipedia.org/wiki/Cliente-servidor>
- Wikipedia. (20 de agosto de 2013). *Escalabilidad*. Recuperado el 29 de septiembre de 2013, de http://es.wikipedia.org/wiki/Escalabilidad#Escalada_vertical_vs._horizontal
- Wikipedia. (2013). *Global unique identifier*. Recuperado el 2 de octubre de 2013, de http://es.wikipedia.org/wiki/Globally_unique_identifier
- Wikipedia. (2013). *LDAP*. Recuperado el 15 de diciembre de 2013, de <http://es.wikipedia.org/wiki/LDAP>
- Wikipedia. (2013). *Patrón Command*. Recuperado el 8 de octubre de 2013, de [http://es.wikipedia.org/wiki/Command_\(patr%C3%B3n_de_dise%C3%B1o\)](http://es.wikipedia.org/wiki/Command_(patr%C3%B3n_de_dise%C3%B1o))
- Wikipedia. (2013). *PBKDF2-SHA1*. Recuperado el 15 de octubre de 2013, de <http://en.wikipedia.org/wiki/PBKDF2>
- Wikipedia. (2013). *Simple Mail Transfer Protocol*. Recuperado el 26 de septiembre de 2013, de http://es.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol