

Hacia Fistera 2.0: aplicaciones de empresa para PYMES*

José María Casanova Crespo, Juan José Sánchez Penas
Alejandro García Castro, José Dapena Paz
Igalia: Ingeniería en Informática y Software Libre
Gutenberg 34B 2º, 15008, A Coruña, Galicia, España

8 de septiembre de 2003

©2003, Igalia S.L. ¹

Resumen

Fistera 2.0 constituye una evolución en las aplicaciones de gestión empresarial en software libre. Partiendo del trabajo realizado por **Igalia** en su versión 1.0, se ha rediseñado una nueva arquitectura multicapa para las aplicaciones empresariales que verá la luz en su versión estable a principios del año 2004.

En este artículo se describe que era **Fistera** en sus inicios y que supondrá la versión 2.0. También se realiza una evaluación sobre los distintos proyectos libres orientados al mundo de la gestión para detallar finalmente en que consistirá el nuevo **Fistera 2**.

1. Aportaciones

En este artículo se aporta con respecto a anteriores exposiciones similares, las novedades del proyecto **Fistera** y el estado de desarrollo de su versión 2.0. Este documento plasma con respecto a la presentación de **Fistera** en la IV Guadec celebrada en Dublín[16] una arquitectura definida para la construcción de aplicaciones de gestión basándose en la tecnología **Fistera 2**. También se incluyen nuevas conclusiones sobre la aplicación de **GNU/Linux** y **GNOME** a entornos empresariales y un breve estado del arte de los programas de gestión libres.

2. Introducción: necesidad de aplicaciones empresariales

En los últimos años, las plataformas basadas en software libre y su nuevo modelo de desarrollo cooperativo, y en particular el sistema **GNU/Linux** y sus aplicaciones asociadas, se han convertido en una alternativa sólida a cualquier otra solución tradicional, basada

*O proxecto **Fistera** é obxecto de axuda pola Xunta de Galicia a través da Consellería de Innovación, Industria e Comercio

¹Copyright Igalia, S.L. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

en software propietario cerrado. El nacimiento de decenas de nuevos proyectos cada mes, la adopción cada vez mayor del software libre por parte de administraciones públicas de todo el mundo, e incluso el apoyo explícito de grandes corporaciones como IBM, Sun o HP, son pruebas contundentes de la evolución de este movimiento que tiene apenas dos décadas y cuya explosión se está viviendo en los últimos cinco años.

El avance del software libre tiene que ver con distintos aspectos, que se pueden ver en la mayoría de los casos como complementarios: en ocasiones es decisiva la libertad que otorga al usuario el hecho de disponer del código fuente y de la posibilidad de analizarlo, modificarlo y redistribuirlo; en otras, cobra especial interés la reducción del TCO (coste total de propiedad) o la muchas veces mayor calidad técnica de las soluciones libres. En cualquier caso, miles de usuarios (empresas, administraciones o usuarios finales) están optando a cada vez mayor velocidad por soluciones innovadoras basadas en una nueva forma de entender el desarrollo de software.

Esta evolución del sistema va pareja a la aparición de soluciones (aplicaciones, distribuciones, servicios) para los distintos ámbitos y sectores de utilización de software libre. Cada vez son menos los huecos en los que hay que recurrir a sistemas cerrados porque el conjunto de sistemas operativos y aplicaciones libres no son capaces todavía de satisfacer las demandas de los usuarios. Si hace unos años las limitaciones de las soluciones de escritorio era una crítica habitual a los sistemas libres, en la actualidad esa carencia ya no existe, gracias a proyectos como KDE o Gnome, que han equiparado la calidad del escritorio GNU/Linux a la de cualquier sistema propietario. No obstante, todavía existen algunos ámbitos en los que faltarán unos años para que esta situación llegue; entre ellos están el del diseño gráfico, el de las herramientas de CAD, o, atendiendo a lo que más interesa para el presente artículo, el sector de las aplicaciones de gestión empresarial (ERP, CRM, contabilidad, etc.).

Existe un hueco por llenar, que se corresponde con las tecnologías, herramientas y aplicaciones adecuadas para construir sistemas de gestión eficientes, eficaces, flexibles, modernos y de calidad. **Fisterra** es un proyecto nacido con la vocación de colaborar en este trabajo, intentando que los sistemas libres, además de todos los sectores ya “conquistados”, puedan acceder a nuevos ámbitos en los que su aplicación traería mayor libertad para todos los participantes.

3. Aplicaciones de gestión en GNU/Linux

La necesidad de aplicaciones y tecnologías de desarrollo especializadas en la gestión empresarial es algo difícilmente cuestionable. GNU/Linux, por la herencia de UNIX, nace como un sistema con especial fuerza en la parte servidor, lo que conlleva que su mayor cuota de mercado durante muchos años haya sido los servidores de intranet y de Internet (el ejemplo más utilizado es el del servidor web Apache). Progresivamente, esta mayor capacidad para el “lado servidor” se ha ido desplazando al *desktop* (escritorio del usuario final), y esto abre nuevas posibilidades entre las que, en el ámbito empresarial y de gestión, es cada vez más evidente el de la aplicación de software libre para solucionar las necesidades que tradicionalmente cubrían (y todavía cubren en la mayoría de los casos) los ERP/CRM propietarios de las grandes empresas de desarrollo de software (SAP, Navision, Libra, etc.).

En los últimos años, han nacido iniciativas derivadas de la coincidencia por parte de distintos grupos de desarrolladores o empresas en la necesidad de software libre de gestión.

Algunas de estas iniciativas están en sus primeras fases todavía y cuentan con pocos apoyos (grupo de desarrolladores limitado y problemas de financiación), pero otras comienzan a vislumbrarse como proyectos de gran interés. Comentamos a continuación algunas de las iniciativas existentes:

- GNUe [1] (GNU Enterprise): metaproyecto que forma parte del proyecto GNU, y que tiene tres objetivos claros: un conjunto de herramientas, incluyendo interfaces de usuario, generadores de informes, módulos para la creación de aplicaciones cliente/servidor de gestión, etc.; un conjunto de paquetes que implementen, utilizando las herramientas anteriores, un ERP completo; y finalmente, la creación de una comunidad de usuarios alrededor de este proyecto. GNUe nace en marzo de 2000, aunque toma las riendas de varios proyectos anteriores que ya se habían iniciado en 1996. En estos momentos las herramientas (servidor de aplicaciones, generador de informes, generador de formularios, navegador, etc.) comienzan a ser utilizables, aunque sus versiones no son todavía muy estables. Los paquetes están todavía en las primeras fases de desarrollo, o incluso sin comenzar. No obstante, ya existen tres proyectos que utilizan las herramientas de GNUe para implementar directamente sus programas de gestión: Luca [10], GNUe Small Business [11] y Rent-Free [12]. El lenguaje escogido para el desarrollo es Python.
- GNUCash [2]: aunque con frecuencia no es citado como una herramienta de gestión empresarial, por estar claramente enfocado a la administración contable más doméstica, el proyecto tiene características interesantes que podrían ser utilizadas para una contabilidad más empresarial. Es un proyecto muy estable, con licencia GPL y desarrollado con tecnología GNOME.
- Compiere [3]: iniciado en 1999 por la empresa del mismo nombre, se trata de un ERP desarrollado en Java y con interfaz web. Es multiplataforma y está publicado con licencia Mozilla (MPL), en la actualidad está ya en una versión completamente estable. El servidor está basado en JBoss [13], con el desarrollo basado en servlets y EJBs. El proyecto tiene en la actualidad una gran actividad.
- G-CTB [5] (GNU ConTaBilidad): implementación de una aplicación de contabilidad clásica, con licencia GPL, que utiliza tecnología GNOME para la interfaz gráfica y el acceso a datos, está desarrollado en C, y se encuentra todavía en versiones iniciales, además de encontrarse el proyecto completamente parado en el último año.
- ASPL-Fact [4]: proyecto iniciado en el 2000 e impulsado por la empresa ASPL (Advanced Software Production Line) que pretende crear una arquitectura modular para el desarrollo de aplicaciones de gestión en GNU/Linux. Por el momento el proyecto se ha centrado en la definición de una arquitectura flexible basada en tecnología GNOME y en el protocolo BEEP para la comunicación cliente/servidor; una primera versión de esta arquitectura está prevista para agosto de 2003. Todo el desarrollo es bajo licencia GPL y con C como el lenguaje principal de programación.
- Bulmages [6]: aplicación de contabilidad para GNU/Linux, con licencia GPL, que utiliza tecnología KDE, desarrollado en C++, y se encuentra en la actualidad en una versión utilizable, con las funcionalidades básicas y implementadas. Su desarrollo

está siendo coordinado por miembros del grupo de usuarios de GNU/Linux de Mallorca. El proyecto inició su actividad en noviembre de 2002.

- Facturalux [7]: aplicación de gestión tipo ERP para GNU/Linux, con licencia GPL y tecnología KDE (QT, C++, Kugar), utiliza una arquitectura denominada A3D (Arquitectura Abierta de Aplicaciones Dinámicas), que almacena código y datos en la BD utilizando el estándar XML y ECMAScript; BEEP [14] es el protocolo de comunicación escogido. El proyecto está impulsado por InfoSIAL, desde principios del 2001, y mantiene una actividad alta en la versión actual del código, que todavía es un prototipo.
- GestIONG [8]: implementación de una aplicación de gestión de ONGs y sociedades sin ánimo de lucro, haciendo especial hincapié en la creación de un programa estándar de contabilidad. El proyecto dio comienzo mayo de 2003 y está disponible ya un prototipo con funcionalidad básica. El programa, con licencia GPL, utiliza también tecnología KDE para la implementación (QT, C++, Kugar) y presenta una actividad considerable actualmente.
- Gestión Libre [9] en Hispalinux: proyecto que pretende coordinar acciones (documentación, desarrollo, especificaciones) relacionadas con la introducción de software libre de gestión en empresas y administraciones públicas. El proyecto existe desde el año 1999.

En este contexto, en el que se ve claramente que están surgiendo iniciativas en los últimos 3 años para la creación de software de gestión empresarial, y también se puede observar que la mayor parte de los proyectos están todavía lejos de proporcionar una solución sólida al usuario, en el año 2003, nace el proyecto **Fisterra**. El proyecto es impulsado y desarrollado por **Igalia** y basado en tecnologías **GNOME**. En los siguientes apartados se detallan los aspectos más importantes del proyecto, tanto en su primera fase, que se detalla en el apartado 4, en la que un desarrollo GPL a medida es utilizado como punta de lanza para el inicio del camino hacia un ERP Open Source [15], como en la segunda parte, descrita en el apartado 5, en la que ya se plantea la arquitectura en la que se está trabajando actualmente en el proyecto, y que constituye un paso ya mucho más ambicioso para colocar a **Fisterra** definitivamente entre las alternativas más interesantes para la construcción de soluciones ERP/CRM en GNU/Linux.

4. **Fisterra 1.0**

Fisterra 1.0 es la primera versión estable del proyecto, nacida de una generalización de un desarrollo para el sector de la reparación de lunas de automóvil. Esta versión todavía no planteaba una solución global para construir ERPs sectoriales, pero fue un primer paso en esta dirección, y su publicación tuvo una gran repercusión, abriendo el desarrollo del proyecto. En los apartados posteriores se explica el origen del proyecto y el diseño y la arquitectura de esta versión, así como la situación actual de la misma y su futuro tras la aparición de **Fisterra 2.0**.

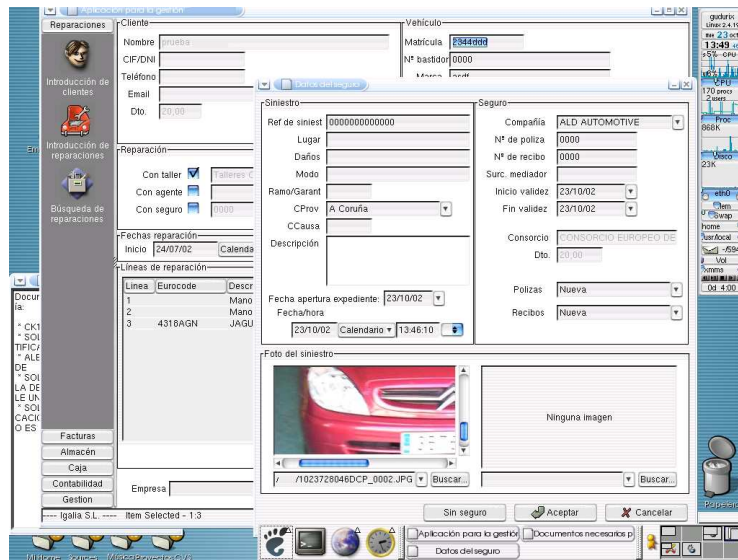


Figura 1: Captura de pantalla de **Fisterra 1**

4.1. Origen del proyecto: reparación de lunas de automóviles

En el año 2002, Igalia firmó un contrato con una empresa gallega dedicada a la reparación de lunas en el sector del automóvil llamada Auto Arte. La empresa posee actualmente ocho talleres de reparación y una sede central y otra sede secundaria, todos ellos separados geográficamente y sin conexión permanente a Internet. El contrato consistía, además de en el diseño de toda la arquitectura informática para la conexión y el funcionamiento diario de toda la empresa, en el desarrollo de un software de gestión empresarial que sustituyese totalmente al utilizado en aquel momento. Cuando se firmó el contrato, la empresa no poseía ningún tipo de informatización en los talleres, y utilizaba en las sedes centrales un software de gestión propietario empaquetado, con todos los ordenadores utilizando sistemas operativos cerrados. El compromiso era crear una solución totalmente libre, que funcionara sobre el sistema operativo GNU/Linux, y que fuese posteriormente publicada como primer paso para la creación de un proyecto de software libre abierto.

Durante varios meses, tras un intenso y exhaustivo análisis de las necesidades del cliente, un equipo de desarrolladores trabajaron con intensidad en el proyecto, y crearon una solución a medida basada completamente en tecnologías GNOME. La solución fue desplegada como un prototipo, los datos de la antigua aplicación migrados a la nueva estructura, y la aplicación fue puesta a prueba durante un tiempo, para pasar finalmente a explotación. Tras estar unos meses en funcionamiento constante y depurar los lógicos errores que fueron surgiendo, Igalia estableció un proceso de publicación que hizo oficial el día 13 de marzo de 2003, tras haberlo publicitado con anterioridad en varios foros de Internet. Igalia se comprometía a la creación del proyecto en sourceforge y creación de la web pública del proyecto, lista de correo, paquete debian y mantenedor/coordinador del proyecto, a la publicación de la versión 1.4. Apertura total a la colaboración de la comunidad, bajo licencia GPL; y a la publicación de la versión 1.0 del documento propuesta de arquitectura Igalia para aplicaciones de gestión sobre tecnologías Gnome 2. Todo ello

se cumplió con exactitud posteriormente.

El 1 de mayo de 2003, **Fisterra** se convirtió en un proyecto abierto y público de software libre. **Fisterra** sigue hasta el día de hoy funcionando como sistema global de gestión para la empresa Auto Arte.

4.2. Objetivos de **Fisterra** 1

Los objetivos del desarrollo propuesto eran los siguientes:

- Secciones de gestión a cubrir:
 - Clientes (CRM), gestión sencilla de clientes.
 - Ventas, gestión de la realización de reparaciones.
 - Compras, gestión de la adquisición de material.
 - Cadena de suministro, gestión de almacenes.
 - Exportación de información para la gestión y la relación con otras empresas colaboradoras.
 - Administración de la aplicación.
- TPVs² para los talleres y los almacenes.
- Gestión centralizada de la información de todos los talleres y almacenes. Para ello era necesaria la realización de la replicación de la información a través de todos los nodos de la red.
- Posibilidad de comunicaciones por líneas de bajo caudal, usando módems.
- Uso de licencia GPL y publicación de la aplicación para la creación de un proyecto libre.
- Utilización de tecnologías de software libre e interfaces gráficos. El uso de GNOME nos proporciona la solución a este requisito.

4.3. Diseño del sistema

El diseño del sistema final se realizó después de varias pruebas tecnológicas y teniendo en cuenta las necesidades de la planificación del proyecto. Esta solución debía permitir la continuación hacia una arquitectura más ambiciosa a la vez que cumplía las necesidades de tiempo para la entrega del proyecto.

En la figura 2 podemos ver la arquitectura planeada para la implementación de **Fisterra** 1. Se decidió utilizar únicamente tecnologías de la plataforma de desarrollo GNOME para la implementación; esta fue una decisión de compromiso con una tecnología que solucionaba la mayor parte de las necesidades del proyecto y cuyos planteamientos y expectativas eran muy interesantes. Las fechas límite para la finalización del proyecto nos hizo buscar una arquitectura de dos capas, más sencilla en la implementación, pero sin olvidar la necesidad futura de migración a tres capas.

²Terminales Punto de Venta

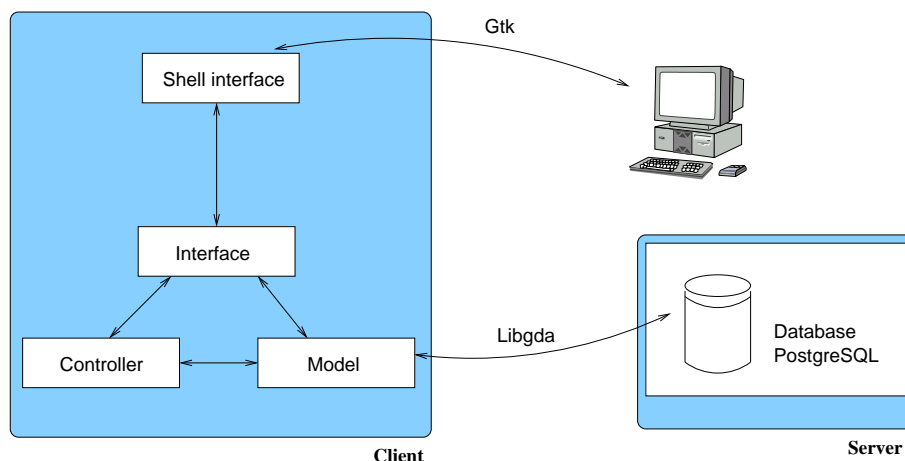


Figura 2: Arquitectura de **Fisterra 1**

Como vemos en la figura 2 la solución utilizada es el patrón de arquitectura MVC³. En el modelo introdujimos el acceso a datos y la lógica de negocio, al no disponer de una capa para la separación explícita de la misma. En el controlador se situaba la lógica de funcionamiento del interfaz. La vista se define mediante los ficheros XML de glade.

Como hemos dicho esta arquitectura planteada debía permitir la creación en un futuro de una estructura multicapa (cliente/servidor/base de datos), para ello se utilizaría CORBA para la comunicación entre modelos y controladores/interfaces.

4.4. Tecnologías utilizadas

Para la implementación se utilizó GNOME 1.4, y se utilizó C como lenguaje de programación. En ese momento todavía no se había publicado una versión estable de GNOME 2. El acceso a datos se realizó utilizando *libgda* (0.2.96). Esta era la última versión estable de la biblioteca para la versión de GNOME que estábamos utilizando; la aplicación usa *libgda* para aislar el acceso a la base de datos.

El entorno de desarrollo utilizado fue: *GNU/Emacs* y *glade*. Para la construcción de los interfaces en la ejecución se usó la biblioteca *libglade*, que nos permite cargar los XMLs definidos usando glade en tiempo de ejecución.

Uno de los requisitos de la aplicación era la gestión centralizada de los datos que se introducción en cada tienda. Después de evaluar algunas soluciones para al replicación de bases de datos se decidió el desarrollo de una solución propia sobre *PostgreSQL*. El sistema utiliza un log de sentencias SQL realizadas sobre cada base de datos. El nodo central tiene la responsabilidad de gestionar el intercambio de datos entre los nodos utilizando los logs disponibles en cada posición. Los envíos se realizan utilizando conexiones *SSH* para asegurar las conexiones y evitar problemas de seguridad.

4.5. Publicación y colaboración

Como resultado del proceso de preparación para la publicación que sufrió el proyecto realizado a medida para Auto Arte, surgió la web pública del proyecto (www.fisterra.org)

³*model-view-controller*

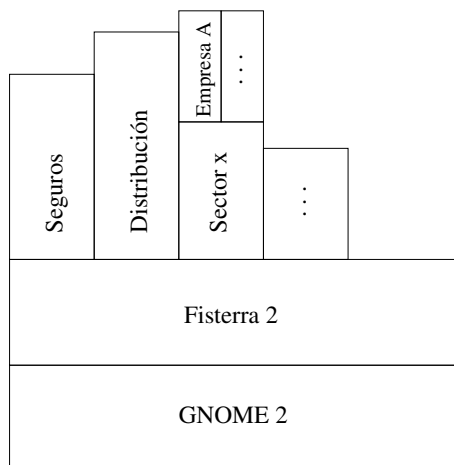


Figura 3: Estructura de **Fisterra 2.0**

y una entrada en SourceForge (www.sourceforge.net/projects/fisterra) en las que se centra el desarrollo de la aplicación.

Con la publicación se buscaba la mayor colaboración posible de la Comunidad de Software Libre, por lo que una de las primeras tareas a llevar a cabo fue la de dar soporte de internacionalización, ya incluida en la versión 1.5.90. Durante los meses de verano se ha trabajado en añadir nuevas funcionalidades a esta versión y en corregir errores que van surgiendo, fruto del uso continuado de la aplicación en entornos reales de explotación.

Como ejemplo del éxito de la publicación, ya desde el segundo mes de su aparición en sourceforge.net, **Fisterra** contaba con más de 20 desarrolladores implicados en la evolución del proyecto. Tras los cuatro primeros meses, las listas de correo tienen una actividad creciente, ha habido más de 25.000 visitas en la página web, más de 1.200 descargas de la documentación, y alrededor de 1.800 personas se han bajado el software para probarlo.

5. **Fisterra 2.0**

5.1. **Presente del proyecto: Distribución y Seguros**

Actualmente **Fisterra** es un proyecto vivo, desde el momento de su publicación hasta la fecha de hoy se ha trabajado mucho en el diseño y arquitectura de la nueva versión. Se ha invertido grandes esfuerzos en definir las necesidades de las aplicaciones de gestión empresarial tanto desde el punto de vista tecnológico como conceptual.

Fisterra 2 es hoy por hoy el núcleo sobre el que se apoyarán otros proyectos de **Igalia** orientados a desarrollos verticales para sectores específicos. En este momento existen varias adaptaciones en marcha: se está trabajando en una solución para el sector de las corredurías de seguros mediante un proyecto parcialmente financiado por la Xunta de Galicia mediante las ayudas del plan de Industria de la Consellería de Innovación, Industria e Comercio con la participación de la Unión Europea, a través del Fondo de Desarrollo Regional; además se está generalizando la solución de **Fisterra 1** para englobar las empresas de distribución, este proyecto se engloba en la adaptación y personalización de la aplicación para otra

empresa orientada a este tipo de negocio. Por último, **Igalia** está en conversaciones con empresas especializadas en la gestión de negocio para la colaboración en el enriquecimiento de la solución y para la distribución a mayor escala de sistemas basados en esta tecnología.

En la actualidad el desarrollo de **Fisterra** se realiza de forma pública mediante el servidor de CVS de *sourceforge*. Para la realización del trabajo en grupo se dispone de una página web, una lista de correo para los desarrolladores y un repositorio para la bajada de archivos.

Uno de los intereses principales en la actualidad es la creación de una comunidad de desarrolladores fuerte interesada en la creación de aplicaciones de gestión basadas en software libre; por ello estamos intentando la colaboración con otros proyectos, acordando arquitecturas comunes que nos puedan llevar en un futuro cercano a la creación de estándares de facto para la creación de este tipo de aplicaciones.

Intentando la generalización de los procesos de negocio de una empresa **Fisterra** se ha dividido en las siguientes partes principales:

- CRM⁴, proceso relacionado con la gestión de las relaciones con clientes.
- Entradas, proceso por el que se realizan compras de materiales por parte de la empresa, comienza con un presupuesto de una empresa y finaliza con una factura de cargo.
- Ventas, proceso por el que se realiza una venta por parte de la empresa, comienza con un presupuesto para un cliente y finaliza con una factura de abono.
- Cadena de suministro, proceso que se encarga de la gestión de material de toda la empresa.
- Negocio, encargado de realizar el análisis de la situación actual de la empresa.

5.2. Objetivos de **Fisterra 2**

Fisterra 2, después de la publicación de **Fisterra 1.0** como primera versión pública de **Fisterra**, nace como la evolución tecnológica en el desarrollo de aplicaciones de gestión empresarial de **Igalia**.

Tras la experiencia alcanzada tras el desarrollo de la primera versión se ha realizado una refactorización completa de la arquitectura para permitir desarrollos más escalables de sistemas de información para empresas.

Los objetivos de **Fisterra 2** son principalmente abstraerse de **Fisterra 1** y de autoarte para construir una solución más general y flexible que sea adaptable de forma sencilla a nuevos negocios.

Entre los objetivos para este nuevo desarrollo se encuentran:

- Incorporación de la tecnología GNOME2.
- Arquitectura multicapa.
- Adaptación a nuevos negocios.

⁴Customer Relationship Management

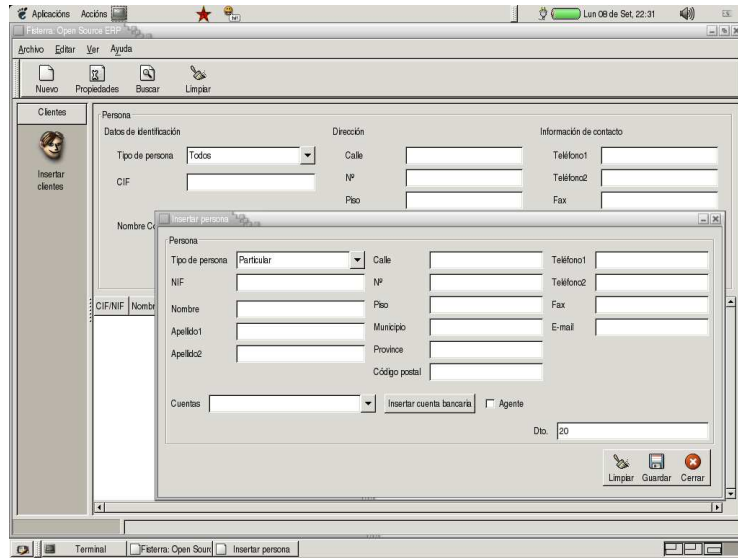


Figura 4: Captura de pantalla de Fisterra 2

5.3. Diseño del sistema

El diseño de Fisterra 2.0 se orienta hacia una arquitectura cliente-servidor. Este diseño es una refactorización más generalista del existente en la versión 1.0.

Fisterra 2.0 establece una separación entre cliente y servidor. El cliente tiene como funciones la interacción con el usuario final mediante la interfaz gráfica. Es necesario poder ejecutar instancias del cliente en múltiples equipos. Su implementación es totalmente independiente de la del servidor siempre que cumpla las interfaces de comunicación.

El servidor se encarga de gestionar el proceso de negocio de la aplicación. Realizar las operaciones que enviadas por los clientes, a la vez que gestiona el acceso a la base de datos, el control de accesos y las autorizaciones.

5.3.1. Cliente Fisterra 2.0

El cliente en Fisterra es la parte visible al usuario, la cual está controlada por un autómatas de estados que determina el flujo de la aplicación. La estructura interna del cliente está definida siguiendo el patrón de diseño MVC (*Model-View-Controller*) en dos niveles: a nivel de aplicación y a nivel de ventana.

La arquitectura a dos niveles nos posibilita definir claramente como se construye una ventana y posteriormente el flujo de las transiciones entre las mismas.

Cada ventana está formada por tres componentes:

Vista descrita mediante ficheros XML de glade.

Controlador rellena la ventana en base a su modelo, gestiona los eventos y callbacks.

Modelo almacena los objetos con los que se trabaja en la ventana.

Este mismo planteamiento se sigue cuando se describe la aplicación cliente en general, las partes son las mismas pero con distintas funciones.

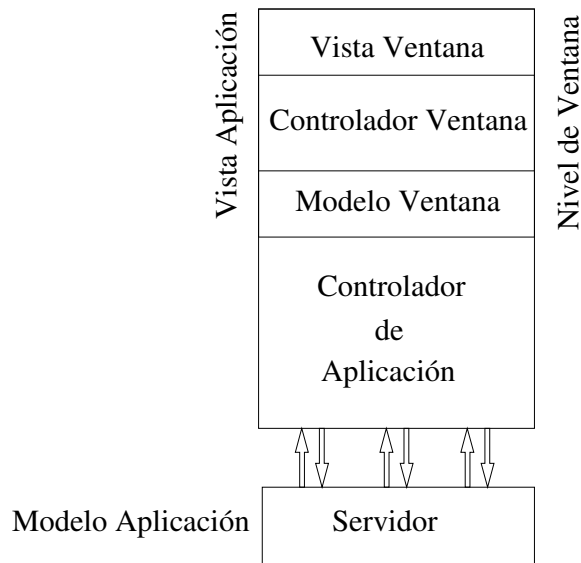


Figura 5: Diseño del Cliente de Fistera2

Vista representada por cada una de las ventanas del nivel anterior.

Controlador gestiona las transiciones y control entre ventanas, se implementa mediante un autómata de estados.

Modelo dispone de los servicios que se encuentran alojados en el servidor.

Esta separación en dos niveles supone una claridad en cuanto a implementación y seguimiento del flujo del cliente, y permite separar de forma clara los distintos elementos que intervienen en la ejecución de la aplicación.

5.3.2. Servidor Fistera

La novedad en **Fistera 2.0** se encuentra principalmente en que se separa del cliente todo el comportamiento de gestión de negocio, autenticación y acceso a datos en un servidor. Éste proporciona distintos servicios que son utilizados por el cliente para la ejecución de la aplicación.

El sistema servidor permite disponer de un único punto de acceso a la base de datos, de forma que todas las operaciones contra ella se ejecutarán desde un único punto. Esto además elimina la dependencia de la base de datos para la gestión de la concurrencia, tal y como ocurría en **Fistera**.

El servidor permite gestionar la operativa de distintos clientes y disponer de una gestión global de autorizaciones y control de las operaciones.

El servidor se encuentra descrito mediante los siguientes componentes y capas:

La capa de comunicaciones implementa las interfaces disponibles para su utilización en el cliente, se encarga de aceptar las peticiones en el cliente y redirigirlas al servicio correspondiente. La tecnología de comunicaciones se mantendrá abstracta al sistema, pero para las implementaciones iniciales se utilizará CORBA.

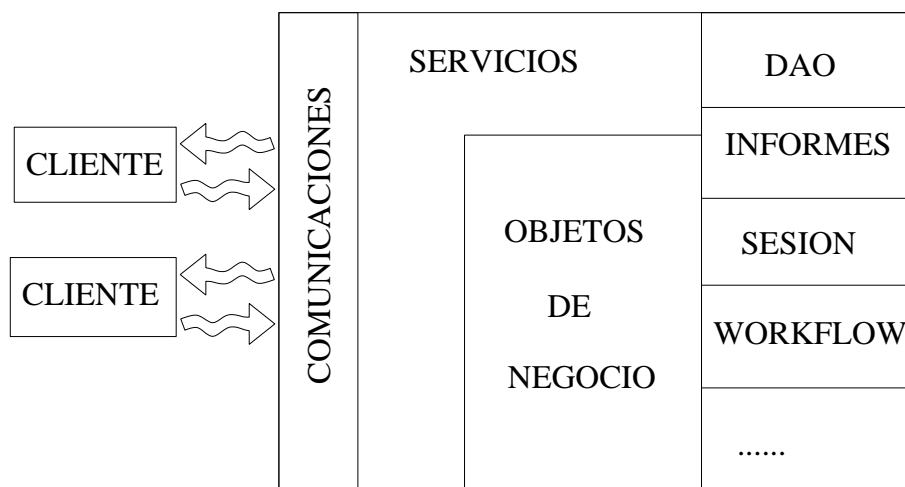


Figura 6: Diseño servidor **Fisterra** 2.0

En la capa de servicios se dividen en 2 tipos, servicios de utilidad y servicios de negocio. Entre los servicios utilidad se encuentran el acceso a datos, gestión de workflow, gestión de sesión y elaboración de informes. Los servicios de negocio implementan comportamientos del negocio soportado por la aplicación y se apoya sobre los servicios utilidad, de esta forma las personalizaciones del modelo de negocio consistirían en modificar o parametrizar estos servicios.

La nueva arquitectura está detallada en la figura 6

5.4. Tecnologías utilizadas

Fisterra utiliza como base el conjunto de tecnologías que conforman el Gnome SDK, para las interfaces de usuario y el servidor de aplicaciones. Sobre éstas, integra e introduce otras tecnologías, para dar soporte a persistencia, e implementar la comunicación, tal y como hemos podido ver en apartados anteriores. Sobre ellas, **Fisterra** constituye por si mismo un completo framework de desarrollo de aplicaciones.

A continuación detallamos las principales tecnologías que se utilizan en **Fisterra**.

5.4.1. Tecnologías GNOME

GTK Librería que proporciona los widgets y controles. En **Fisterra** se utilizan los estándares de desarrollo de interfaces del

escritorio GNOME, y se observa el respeto por los estándares que define, en especial las HIG⁵.

LibGlade Librería para la descripción visual de interfaces de usuario, y el uso de estas descripciones en aplicaciones. La mayor parte de las interfaces de usuario de **Fisterra** se desarrollan con esta herramienta.

⁵HIG (*Human Interface Guidelines*): Documento de normas y estándares sobre la implementación de interfaces de usuario en GNOME.

GObject Sistema que permite realizar orientación a objetos en C, usado por el proyecto GNOME. Los objetos de negocio y la persistencia se implementan utilizando GObject.

LibXML, libXSLT Librerías que implementan operaciones para manipulación de ficheros XML, y transformaciones XSL. Se utilizan en **Fisterra** como formato de intercambio.

Orbit Implementación ligera de CORBA, protocolo de invocación remota de métodos, y comunicación de objetos interoperable. Se utiliza para la comunicación entre los distintos servicios del sistema, y entre cliente y servidor.

libGDA Librería de acceso a bases de datos del proyecto Gnome. Se utiliza para implementar el acceso genérico a bases de datos relacionales, y proporcionar así persistencia a la capa de negocio.

5.4.2. Otras tecnologías

PostgreSQL Gestor de bases de datos relacional, que implementa un gran conjunto de funcionalidades de SQL, de forma eficiente. Es la base de datos elegida para el almacenamiento de información en **Fisterra**.

5.5. Publicación y colaboración

En el momento de elaboración de este artículo se encuentra publicada en la web la versión 0.0.2 de **Fisterra 2.0**, está formado por un subconjunto de las funcionalidades de **Fisterra** sobre las que se han realizado las pruebas tecnológicas sobre el entorno GNOME 2, consistentes en la migración de la interfaz de **Glade1** a **Glade2** y la actualización de la librería **libgda** a su versión 0.12 desde la 0.2.

Esta versión soporta internacionalización (i18n) y está disponible en castellano, inglés y euskera. La plataforma soportada por Igalia es Debian GNU/Linux y se está trabajando en facilitar el proceso de instalación en otras distribuciones. La instalación actualmente en Debian se limita a introducir un nuevo repositorio en **apt**.

Igalia está trabajando actualmente en buscar puntos de encuentro con otros proyectos para intercambiar experiencias y tratar de clarificar el futuro de las aplicaciones de gestión empresarial como Software Libre.

6. Conclusiones y trabajo futuro

En este artículo hemos presentado el pasado, presente y planes futuros de un nuevo proyecto de software libre llamado **Fisterra**. El desarrollo del ERP Open Source se encuentra en estos momentos inmerso en la segunda etapa de generalización, con la definición de una arquitectura que marca el rumbo cara a un framework que permita definir con facilidad nuevas aplicaciones de gestión empresarial para sectores distintos de los ya implementados. Con la versión 1 ya estabilizada y en funcionamiento, el proyecto y sus desarrolladores está embarcado en una segunda fase intensa y compleja de la que debería salir, si todo sigue como se prevé, una versión avanzada de la segunda versión, siguiendo las decisiones de diseño arquitectónico que se explican en el artículo.

Con estas líneas trazadas, se deja claro cuál es el enfoque actual del proyecto, que camina hacia un ERP abierto en dos niveles diferentes: por un lado los componentes básicos que forman el framework de desarrollo de aplicaciones basadas en **Fisterra** (solución horizontal común), y por otro las implementaciones de los distintos sectores verticales especializadas y posiblemente adaptables para las necesidades de una empresa en concreto. Tres sectores están siendo desarrollados ya, y sus partes comunes están siendo abstraídas, y dichas implementaciones formarán parte de las próximas versiones de **Fisterra 2**, completamente integrada con las más recientes tecnologías del proyecto **GNOME**.

A pesar de que los desarrolladores del proyecto se encuentran satisfechos con las decisiones tomadas, tanto tecnológicas como de diseño de arquitectura, es muy importante el intercambio y la colaboración con otros proyectos con objetivos similares, con lo que en estos momentos se están llevando a cabo iniciativas para contactar con los coordinadores de distintos proyectos de software de gestión empresarial libre. En el futuro, **Fisterra** buscará todavía más la sinergia con otras iniciativas, intentando ser una solución mejor y más completa para los usuarios de software libre.

7. Agradecimientos

Los autores del artículo quieren agradecer a Javier Vázquez Lamas, Xavier Castaño García, Alberto García González, José Juan González Alonso, Javier Fernandez García-Boente, Alejandro Piñeiro Iglesias y Xavier Rodríguez Calvar su colaboración en el desarrollo del proyecto **Fisterra**.

Referencias

- [1] GNU Enterprise: <http://www.gnenterprise.org>
- [2] GNU Cash - Open Source accounting Software: <http://www.gnucash.org>
- [3] Compiere, Smart Open Source ERP Software with integrated CRM Solutions: <http://www.compiere.org>
- [4] ASPL-Fact, facturación para Linux: <http://fact.aspl.es>
- [5] G-CTB: GNU ConTaBilidad: <http://www.tecnopro.net/gctb>
- [6] Bulmages, contabilidad para Linux: <http://bulmages.bulma.net>
- [7] FacturaLUX - Software ERP de código libre: <http://www.facturalux.org>
- [8] GestioNG, Software solidario para organizaciones solidarias: <http://gestiong.sourceforge.net>
- [9] Gestión Libre: <http://gestion-libre.hispalinux.es>
- [10] LUCA, programa de gestión: <http://www.vialibre.org.ar>
- [11] GNUe Small Business Edition: <http://www.gnu.org/software/gnue-sb>
- [12] Rent-Free: <http://savannah.nongnu.org/projects/rentfree>

- [13] JBoss, Professional Open Source: <http://www.jboss.org>
- [14] RoadRunner, Industrial Strength BEEP Toolkit: <http://rr.codefactory.se>
- [15] Open Source Initiative: <http://www.opensource.org>
- [16] Alejandro García Castro, José Dapena Paz. GNOME for business appliances: case study and architecture proposal. GUADEC IV Dublin,2003.