



TCO

USO DE SOFIA2

COMO BACKEND

VS DESARROLLO A MEDIDA

SOBRE UNA BASE DATOS

RELACIONAL

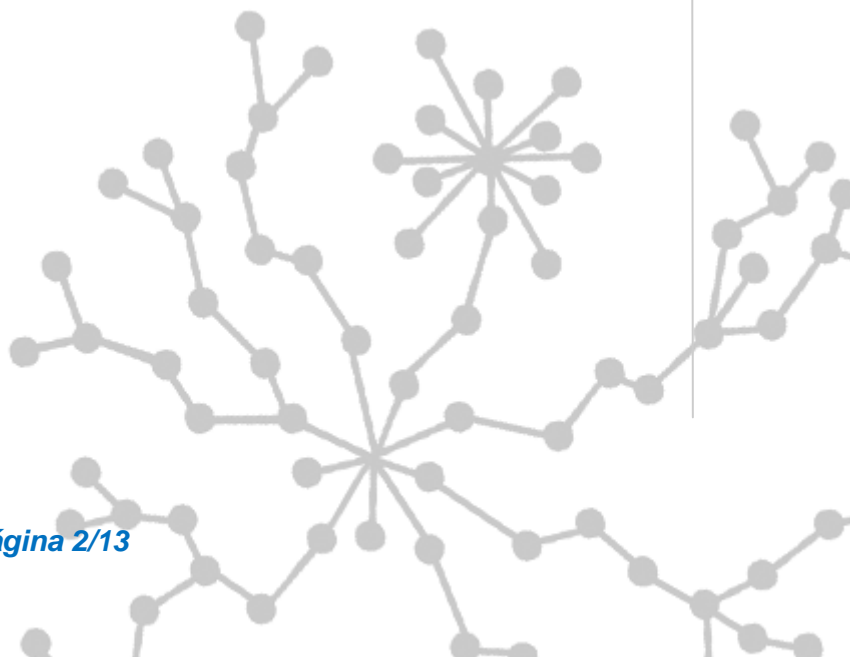
ABRIL 2015

Versión 1



1 INDICE

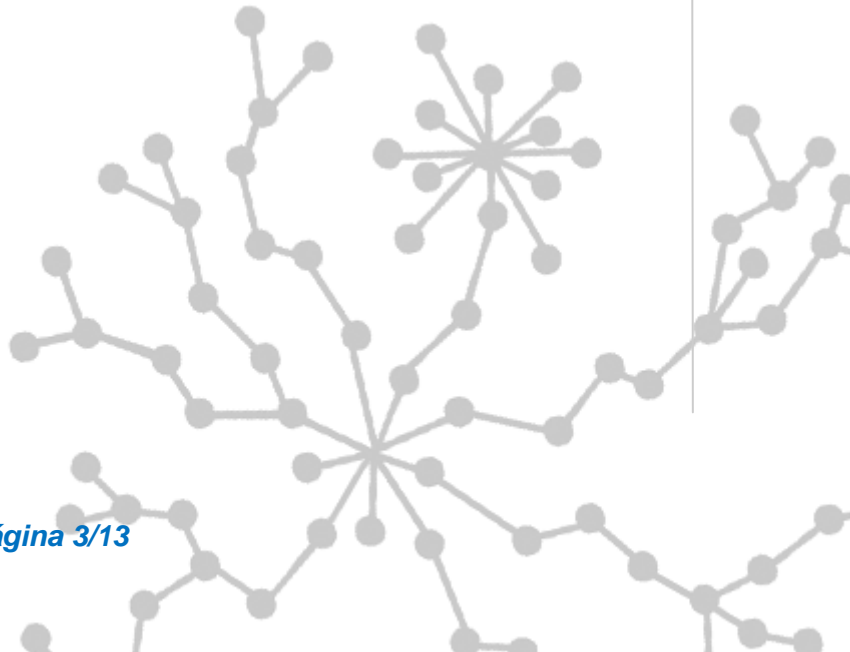
1	INDICE.....	2
2	OBJETIVOS Y ALCANCE DEL PRESENTE DOCUMENTO	3
3	QUÉ ES SOFIA2	4
3.1	CARACTERÍSTICAS DIFERENCIALES.....	4
4	COSTE TOTAL DE PROPIEDAD (TCO)	8
4.1	COSTES INICIALES.....	8
4.2	COSTES CORRIENTES.....	8
5	TCO ENTRE USO SOFIA2 COMO BACKEND	9
5.1	ESFUERZO DE DESARROLLO INICIAL.....	9
5.2	ESFUERZO ADMINISTRATIVO INICIAL	10
5.3	LICENCIAS DE SOFTWARE	11
5.4	HARDWARE DE SERVIDORES	11
5.5	HARDWARE DE ALMACENAMIENTO	12
5.6	ESFUERZO DE DESARROLLO CORRIENTE	12
5.7	ESFUERZO ADMINISTRATIVO CORRIENTE.....	12
5.8	MANTENIMIENTO Y SOPORTE TÉCNICO	12
5.9	OTRAS VENTAJAS DE SOFIA2	13



2 Objetivos y alcance del presente documento

El presente documento describe el razonamiento empresarial para proponer Sofia2 como Backend en lugar de hacer un desarrollo tradicional sobre una base de datos relacional.

Este informe hace una comparación entre el TCO (Coste Total de Propiedad) de la Plataforma Sofia2 y el de un desarrollo a medida considerando los costes iniciales y corrientes (software, hardware y personal).



3 Qué es Sofia2

SOFIA2 puede definirse como un middleware + repositorio capaz de procesar miles de eventos por segundo, con capacidades almacenamiento Big Data con reglas integradas, interfaces multiprotocolo y multilenguaje y todo operable desde una consola Web

Las características diferenciales de Sofia2 frente a otras Plataformas son:

- Entorno de Experimentación gratuito: CloudLab
- Operación y gestión de la Plataforma 100% web
- Enfoque Semántico (ligero)
- Independencia del protocolo de comunicación
- Modular, Extensible y personalizable
- Seguridad integrada
- Modular
- APIs multilenguaje
- API Manager integrado
- Motor de Reglas y Motor CEP integrado
- Capacidades Social Media integradas
- Implementación de referencia sobre software Open-Source
- Despliegue On-Premise y On-Cloud
- Integrada capacidades Big Data out-of-the-box
- Gestión de SW y configuración dispositivos centralizada

3.1 Características diferenciales

- **Entorno de Experimentación gratuito: CloudLab**
 - Dentro del Offering de la Plataforma se ofrece un Entorno de Experimentación en el que los desarrolladores de la Plataforma pueden probar todas las capacidades de la Plataforma sin coste. Esto permite poner en valor la Plataforma
 - Adicionalmente a través de CloudLab+ se ofrece la instalación de un entorno a medida y de horas de soporte a un precio asequible para el desarrollo de pilotos para mostrar la viabilidad de un proyecto.
- **Operación y gestión de la Plataforma 100% web**
 - Toda la Plataforma se gestiona, mantiene y opera desde una Consola Web, tanto la definición de entidades (ontologías), seguridad, creación de clientes, reglas,...

- Además se ofrece APIs REST para toda esta gestión, lo que permite integrar esta gestión en otras consolas de administración
- **Enfoque Semántico (ligero)**
 - La Plataforma ofrece un enfoque semántico, lo que permite aprovechar esta semántica a la hora de hacer consultas, procesos analíticos,...
 - Se propone un enfoque semántico ligero basado en JSON, apropiado para todo tipo de dispositivos, incluso los menos potentes.
 - Se ofrecen modelos creados para diversos dominios, como dominio en el ámbito Smart City, Smart Energy. Estos modelos pueden ampliarse.
 - Desde la consola Web se pueden crear las entidades conforme a los dominios creados.
- **Independencia del protocolo de comunicación**
 - Los mensajes intercambiados entre los clientes y la Plataforma son independientes de la Plataforma, se ofrecen out-of-the-box conectores para MQTT, REST, WebSockets, WebServices y JMS.
 - Los desarrolladores pueden crear nuevos conectores pluggables.
- **Modular, Extensible y personalizable**
 - La Plataforma está creada con un enfoque modular lo que permite sustituir la implementación de un módulo por otra contemplando los interfaces (por ejemplo uso de SAP HANA en lugar de base de datos MongoDB de Implementación de referencia).
 - La Plataforma está pensada para extenderse en el proyecto a través del concepto de plugin. Los plugins se despliegan como JARs y permiten crear nuevos conectores, definir el modelo de seguridad adecuado,...
 - La instalación puede personalizarse según las necesidades, instalando sólo los módulos adecuados, definiendo parametrización,...
- **Seguridad integrada**
 - Dentro de la Plataforma la seguridad es un concepto fundamental.
 - Se ofrece seguridad a nivel de comunicación (HTTPS, MQTTS,...) y también a nivel de permisos sobre las entidades, pudiendo ser entidades privadas, públicas o bien con acceso personalizado.
- **APIs multilenguaje**
 - Aparte del conector REST que puede usarse de forma sencilla desde cualquier lenguaje se ofrecen APIs multilenguaje cuando se necesitan protocolos más avanzados y eficientes.
 - Las APIs permiten comunicar de forma más sencilla con la plataforma.
 - Se ofrecen APIs Java, Javascript, C/C++, Python, Android, iOS, Node.js, Arduino,...
 - Todas estas APIs se ofrecen bajo licencia Apache y sin coste.
- **API Manager integrado**

- Las APIS como mecanismo de integración es ya una realidad como prueban grandes como Facebook, Twitter, Google, Bancos a nivel internacional,...
- Permite por un lado disponibilizar como APIs REST las entidades que maneja, permitiendo llevar al concepto Open Data el dinamismo de las APIs REST, también permite disponibilizar APIs Externas bajo un enfoque RESTful ofreciendo un acceso unificado.
- **Motor de Reglas y Motor CEP integrado**
 - La Plataforma incluye un motor de Reglas que permite definir en un lenguaje sencilla y a través de la consola Web las reglas a aplicar antes la llegada de un evento o dato o bien temporizada. Permite a los usuarios con permisos crear en un lenguaje de scripting nuevas lógicas reutilizables e invocables desde el motor (por ejemplo enviar SMS o mail,...)
 - Además se incluye un motor CEP que permite definir reglas en las que interviene el tiempo (por ejemplo que no ha llegado una cierta medida en 1 día). A los eventos generados por el motor CEP pueden suscribirse los clientes o servir como entrada al motor de Reglas.
- **Capacidades Social Media integradas**
 - Desde la propia consola pueden realizarse diversas búsquedas (perfiles, timeline, grupos, hashtags, tendencias) para las principales redes sociales.
 - Estas búsquedas se persisten sobre la plataforma para realizar procesos analíticos.
- **Implementación de referencia (RI) sobre software Open-Source**
 - La RI de la Plataforma se basa por completo en software open-source sin restricciones, esto permite que la plataforma no acarree costosas licencias en su implantación.
 - Además existen otras implementaciones creadas a medida en la que unas piezas son reemplazadas por otras (pj HANA en lugar de Mongo y Hadoop, Motor CEP Oracle en lugar del Motor CEP WSO2, ...)
- **Despliegue On-Premise y On-Cloud**
 - La Plataforma puede instalarse tanto On-Premise como On-Cloud, bien públicas o privadas.
 - Se soportan diversos modelos de operación y diversos tipos de soporte en función de las necesidades de la organización.
 - En cloud puede optarse por un modelo PaaS en el que se cobra por la infraestructura montada o bien SaaS en el que se cobra por el número de mensajes procesados o por TB usados.
- **Integrada capacidades Big Data out-of-the-box**
 - La Plataforma integra de forma nativa un enfoque Big Data, de modo que para cada entidad (ontología) se puede definir cuándo los datos pasan de la RTDB (base de datos Tiempo Real) a la HDB (base de datos histórica) (RI sobre Hadoop).

- La Plataforma integra capacidades para realizar consultas online tanto sobre la RTDB como sobre la HDB
- La Plataforma permite lanzar procesos analíticos sobre la BDH de forma sencilla e integrada.
- **Gestión de SW y configuración dispositivos centralizada**
 - Además de las APIs multilenguaje se ofrece una Infraestructura Java que permite construir aplicaciones para dispositivos embebidos autogestionada.
 - Desde la consola central se puede cargar el SW y la configuración que va a cada dispositivo o grupo de dispositivos.

IoT PLATFORM

Sofia2 BY INDRA

Sofia2 enables IoT interoperability with any device or system because of its multiplatform and multilanguage connectors. Sofia2 supports web modelling of entities, includes a real-time database and additional data storage for historical information with analysis capacities, dashboards and central management of all the platform's concepts via web Console.

LANGUAGES

- JAVA
- JAVASCRIPT
- C/C++
- NODE.JS
- ANDROID
- PYTHON
- ...

MESSAGING PROTOCOLS

- MQTT/MQTTS
- HTTP/HTTPS
- WEBSOCKETS
- TCP/IP

NATIVE CAPABILITIES

- Light Semantic
- Asset Management
- Central Management
- M2M Gateway
- Social Media analysis
- Analytics
- Real-time and historical storage
- Rules
- Integration

HOSTING OPTIONS

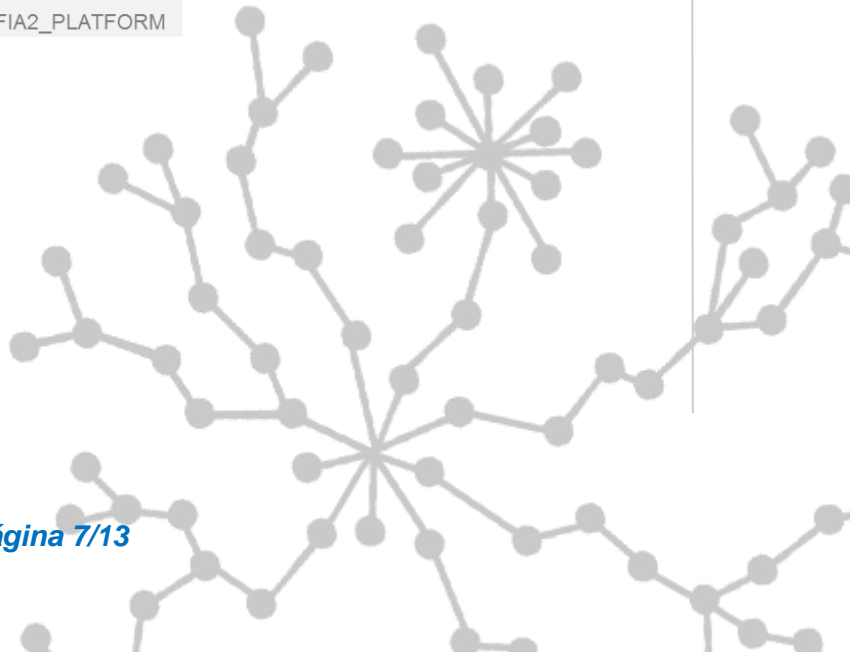
Cloud Hosted or On-Premise

TOP USE CASES

1. Smart Cities/Utilities/Grid
2. Home Automation/Smart Home
3. Smart Services Sector
4. Medical
5. Transportation
6. Wearables

WEBSITE sofia2.com

TWITTER @SOFIA2_PLATFORM



4 Coste Total de Propiedad (TCO)

4.1 Costes Iniciales

Los costes iniciales se componen de:

- **Esfuerzo de desarrollo inicial:** Coste de personal + Programación del desarrollador necesaria para la aplicación
- **Esfuerzo administrativo inicial:** Coste de personal + Administradores para instalar y configurar software, máquinas del clúster, particionado, ...
- **Licencias de software**
- **Hardware de servidores.** Servidores necesarios para ejecutar la base de datos (se excluye almacenamiento). Depende principalmente del número y tipo de procesadores y RAM. Otros costes incluyen recintos, conexiones de red, cableado y suministros de alimentación
- **Hardware de almacenamiento.** Almacenamiento necesario para almacenar los datos, varía en función de si se utiliza almacenamiento interno o compartido (SAN), de la cantidad de almacenamiento y de si se utilizan unidades de disco duro (HDD) o unidades de estado sólido (SSD).

4.2 Costes Corrientes

Los costes corrientes se componen de:

- **Esfuerzo de desarrollo corriente:** Personal + Programación necesaria para adaptar el almacén de datos a las necesidades del cliente, del mercado y empresariales
- **Esfuerzo administrativo corriente:** Personal + Esfuerzo administrativo necesario para mantener el funcionamiento y ejecución del almacén de datos
- **Mantenimiento y soporte técnico del software:** Mantenimiento: actualizaciones y soluciones de errores del software + Soporte técnico: asistencia para localizar y solucionar problemas técnicos en el software
- **Mantenimiento y soporte técnico del hardware:** Mantenimiento: actualizaciones y soluciones de errores del firmware y cualquier software que pueda incluir el hardware + Soporte técnico: asistencia telefónica para localizar y solucionar problemas técnicos en el hardware
- **Costes de despliegue diversos:** Otros costes necesarios para mantener la base de datos en funcionamiento. Incluye costes de nube/alojamiento/cubicación, costes de ancho de banda, tarifas eléctricas, etc.

5 CTO entre Uso Sofia2 como Backend

A continuación veremos cómo usar Sofia2 como Backend reduce los costes que componen el TCO de un sistema abordado como un desarrollo a medida sobre una base de datos relacional.

5.1 Esfuerzo de desarrollo inicial

El esfuerzo de desarrollo inicial se refiere al coste del tiempo dedicado por el desarrollador para conseguir que la aplicación y el almacén de datos trabajen juntos.

En el caso de un desarrollo sobre base de datos relacional, el esfuerzo de desarrollo inicial incluye tareas como definir el modelo de datos, crear una capa de mapeo objeto-relacional (ORM), escribir la lógica empresarial para la aplicación y hacer la capa de presentación para esta lógica.

Sofia2 está diseñado para que reducir los tiempos de desarrollo, de modo que un desarrollador en cualquier lenguaje pueda utilizar la Plataforma con facilidad.

Para eso a través de la Consola Sofia2 (Sofia2-Console) el desarrollador puede:

- Crear sus entidades (Ontologías en Sofia2, tablas en un SGBDR, colecciones en MongoDB)
- Definir sus reglas de negocio de forma sencilla y asistida
- Establecer seguridad en el acceso a sus entidades
- Acceso CRUD (consulta, inserción, borrado, actualización,...) a todas estas entidades a través de cualquier lenguaje (Java, Javascript, C, Android,...) lo que le permite desarrollar tanto aplicaciones Web MVC (API Java, Python, Node.js), aplicaciones HTML5 (API Javascript), aplicaciones móviles (API Android, iOS, Javascript...) o módulos de negocio (Java, Python, C,...)
- Capacidad de suscripción a eventos, consultas, reglas, ...de forma sencilla e independiente del protocolo de mensajería (JMS, MQTT, AMQP,...)
- Publicación asistida y web de APIS REST a partir de las entidades
- Capacidades GIS integradas
- Dashboards integrados
- Informes integrados
- Repositorio Big Data integrado

Por lo tanto, podemos decir que resulta mucho más rentable desarrollar con Sofia2 que hacer un desarrollo a medida sobre bases de datos relacionales.

Otra ventaja de productividad importante de Sofia2 es su diseño de Entidades (Ontologías) orientado a documentos y a los esquemas dinámicos. La forma en que almacena datos de la aplicación se corresponde con la tecnología y prácticas de desarrollo actuales, que han evolucionado considerablemente desde los comienzos de la industria de las bases de datos relacionales hace 30 años.

Algunos motivos que respaldan las ventajas de productividad de Sofia2 son:

- **Facilidad de uso:** Sofia2 es compatible con las metodologías de desarrollo actuales, permite a los desarrolladores realizar iteraciones de forma rápida y continua sobre el modelo de datos y todo desde un interfaz Web. En contraposición un desarrollo tradicional modelo relacional impone un estricto conjunto de limitaciones al desarrollo, tanto a nivel de modelo de datos, de creación de reglas, cambios,...
- **Modelo de datos.** Con Sofia2, el desarrollador solo tiene que crear el modelo de datos en un lugar: la Consola Web del propio producto. En un desarrollo los desarrolladores necesitan crear y mantener el modelo de datos en tres lugares mediante el uso de diferentes interfaces: la aplicación, la propia base de datos y la capa ORM.
- **Flexibilidad de datos.** A diferencia de una SGBDR, Sofia2 permite a los desarrolladores almacenar con facilidad datos polimórficos, así como datos semiestructurados y estructurados, en un almacén de datos individual.
- **Soporte JSON.** El almacenamiento en JSON, pilar básico de numerosas aplicaciones actuales, se realiza sin dificultades y no requiere conversión. Con una SGBDR, los desarrolladores necesitan “aplanar” y transformar JSON para almacenarlo en tablas relacionales, y más tarde tienen que recuperar las capas al realizar la extracción de la base de datos.

5.2 Esfuerzo administrativo inicial

La instalación y configuración de Sofia2 es económica y sencilla.

La Plataforma se compone de :

- **BDC (Base Datos Configuración) :** puede ser cualquier base de datos relacional. Por defecto funciona sobre una BD embebida MySQL.
- **BDTR (Base Datos Tiempo Real):** en la RI es un MongoDB lo que hace que el esfuerzo administrativo inicial sea bajo, un administrador solo debe tener en cuenta una variable: el número de nodos en el clúster. Solo existe un reducido conjunto de ajustes de configuración para poner el sistema en funcionamiento. Los administradores de MongoDB no necesitan integrar capas de memoria caché ni crear lógica de particionado horizontal personalizada para dirigir las consultas al nodo servidor correcto. En lugar de

esto, el almacenamiento en memoria, caché y el particionado horizontal son capacidades centrales de MongoDB.

- **BDH (Base Datos Histórica):** puede funcionar sobre MongoDB o Hadoop en función de las necesidades o preferencias.
- **SIB + Consola + Tools + API Manager + Process:** todos los módulos de negocio de la Plataforma están contruidos en Java, se despliegan como aplicaciones Web en cualquier servidor de aplicaciones JEE. El grueso de la configuración va en la BDC por lo que no es necesario crear ficheros de configuración complejos.

5.3 Licencias de software

Sofia2 es una Plataforma con una versión gratuita para la comunidad de código abierto (licencia Apache) y una edición para suscriptores comerciales que puede usarse en modo On Premise o en Modo Cloud.

Esta versión incluye soporte técnico en diferentes modalidades (desde 8x5 sin SLAS a 24x7 con SLAS estrictas), actualizaciones de software y soluciones de errores y algunas funciones adicionales.

La edición comercial de Sofia2 se factura de forma continua en lugar de puntualmente (esto es, una cuota anual por servidor).

5.4 Hardware de servidores

En general, los costes de servidores de Sofia2 son considerablemente inferiores a los de un desarrollo tradicional sobre BD relacional para cargas de trabajo y disponibilidad similar. Esto aplica a todos los componentes.

Sofia2 se diseña para utilizar hardware básico en arquitecturas escalables.

Los despliegues de Sofia2 normalmente utilizan servidores Linux básicos y económicos, que tienen un coste de tan solo 3.000 \$; incluso un sistema de baja energía y alto rendimiento puede costar tan solo 4.000 \$ (excluyendo almacenamiento).

5.5 Hardware de almacenamiento

La arquitectura escalable de Sofia2 permite reducir considerablemente los costes de almacenamiento.

Sofia2 puede utilizar el almacenamiento local económico y permite realizar un uso eficiente de las unidades de estado sólido (SSD).

5.6 Esfuerzo de desarrollo corriente

Las dinámicas del esfuerzo de desarrollo corriente son menores a las del esfuerzo de desarrollo inicial.

Con un desarrollo tradicional, el coste de realizar cambios en la aplicación es mayor, bien sean cambios en el esquema de una base de datos que ya se encuentre en producción (coste mayor que para una base de datos que aún no se ha entregado), como en el desarrollo de la lógica, reglas, seguridad, configuración).

Por ejemplo con Sofia2 resulta fácil para los desarrolladores agregar campos a las entidades, crear nuevas APIs, lo que se deriva en costes considerablemente inferiores y permite a los desarrolladores adaptar las aplicaciones a medida que evolucionen las demandas.

5.7 Esfuerzo administrativo corriente

El esfuerzo administrativo corriente incluye actividades que mantienen el sistema en buen estado de funcionamiento (por ejemplo, actualización del software y hardware, realización de copias de seguridad y recuperación de tiempos de interrupción inesperados).

Se requiere mucho menos tiempo y esfuerzo para administrar Sofia2 en comparación con un desarrollo tradicional.

La administración de un despliegue de Sofia2 implica principalmente administrar configuraciones de Linux y el propio hardware; solo es necesario conocer y administrar unos pocos parámetros.

5.8 Mantenimiento y soporte técnico

Las suscripciones de Sofia2 se facturan anualmente por core. Esto incluye el acceso al soporte técnico del producto, actualizaciones de software y soluciones de errores, así como ciertas funciones que solo se ofrecen en la edición de pago.

5.9 Otras ventajas de Sofia2

Resumiendo, además de los ahorros de costes tangibles, el modelo orientado a documentos y el esquema flexible de Sofia2 también aportan mayor agilidad y flexibilidad a las empresas, que a su vez proporcionan ventajas para generar ingresos.

Una vez implantada la Plataforma Sofia2 en una empresa esta puede utilizar la Plataforma (sin necesidad de montar nueva infraestructura) para hacer nuevos desarrollos y para integrar datos de otros sistemas de forma que los tenga centralizados en un repositorio común y con capacidades Big Data. Además puede desarrollar aplicaciones Sofia2 en cualquier tecnología y lenguaje.

