



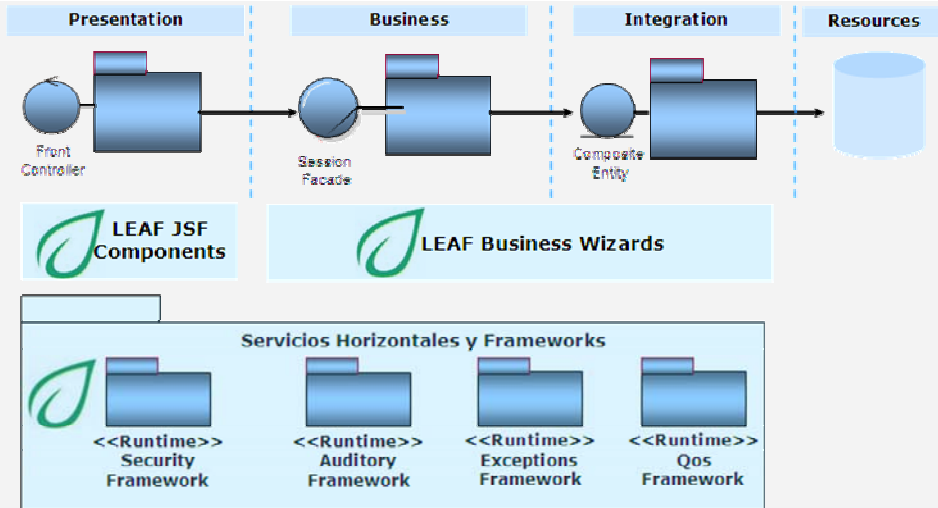
PLATAFORMA EMPRESARIAL PARA ARQUITECTURA, DISEÑO Y CONSTRUCCIÓN DE PROYECTOS CON JAVA ENTERPRISE EDITION



www.lucasian.com

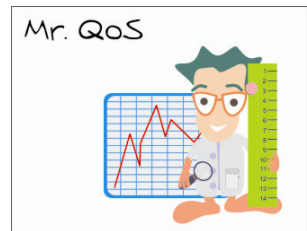
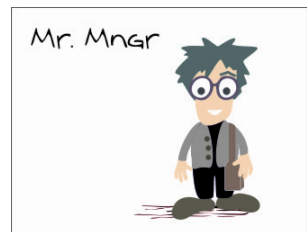
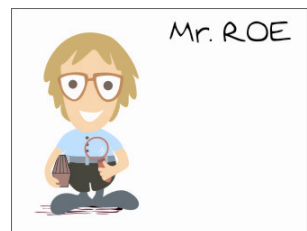
Lucasian Labs Latinoamerica
Revisión 2006 – Q1, Marzo de 2006, Ver. 2.0.4

Lucasian Enterprise Application Framework

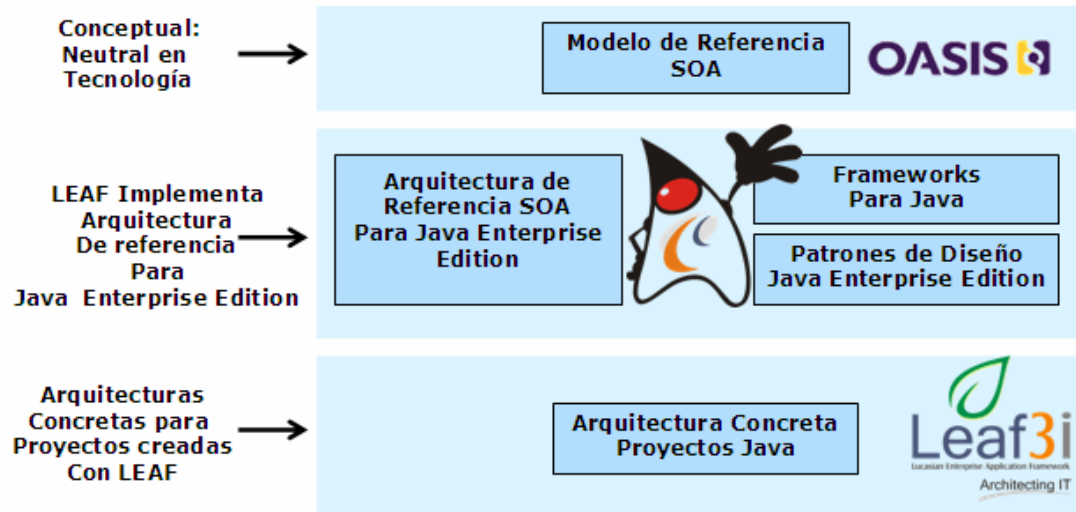


LEAF 3i es una plataforma de servicios empresariales, la cual permite aumentar la productividad y calidad del desarrollo de proyectos Java Enterprise Edition, a través de la utilización de su arquitectura de referencia SOA, soportada sobre los siguientes artefactos:

- ❖ **Core J2EE Design Patterns.** Framework estándar de patrones de diseño J2EE definido por los servicios de consultoría de Sun Microsystems. Referencia: <http://java.sun.com>
- ❖ **Lucasian Design Patterns (LDP).** Patrones de diseño J2EE de Lucasian Labs Ltda complementarios a los patrones propuestos por Sun Microsystems.
- ❖ **LEAF Business Wizards (LBW).** Los asistentes están implementados como plugins sobre la plataforma Eclipse 3.x, y generan el código fuente de alto rendimiento, portables entre servidores de aplicaciones, para los patrones de diseño de la arquitectura de referencia para Java Enterprise Edition.
- ❖ **LEAF Horizontal Frameworks (LHF).** LEAF provee un conjunto de frameworks probados en proyectos en producción, documentados y listos para reutilizar en la implementación de proyectos Java Enterprise Edition:
 - LEAF Security Framework
 - LEAF Exception Framework
 - LEAF Validations Framework
 - LEAF QoS Framework
 - LEAF Scheduler Framework
- ❖ **LEAF Operacional Visibility Enterprise Services (LOVE Services).** Consolas de Administración para configuración y visibilidad operacional de aplicaciones Java Enterprise Edition puestas en producción:
 - LEAF QoS Console
 - LEAF Scheduler Console
 - LEAF Security Console
- ❖ **LEAF Components.** A través de la implementación de proyectos J2EE, se han recopilado componentes reutilizables entre proyectos, los cuales están probados, documentados y satisfacen requerimientos comunes de infraestructura en Java Enterprise Edition.



Arquitectura SOA y LEAF



LEAF implementa los lineamientos del modelo de referencia SOA de OASIS y la arquitectura de referencia para Java Enterprise Edition, soportado por frameworks y patrones de diseño.

La arquitectura de referencia para Java ha sido madurada y probada por Lucasian Labs a través de la ejecución de más de 30 proyectos exitosos en la industria en sectores financiero, telecomunicaciones y gobierno. El modelo propuesto de LEAF con respecto a SOA es el siguiente:

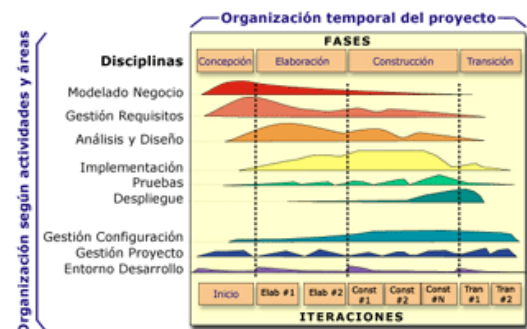
Proceso de Desarrollo Unificado (UP) y LEAF

En la fase de elaboración, LEAF aporta las bases arquitectónicas y lineamientos de la arquitectura de referencia SOA para disminuir el esfuerzo y la complejidad de la definición de la arquitectura línea base de un proyecto Java Enterprise Edition.

En la fase de construcción, LEAF aporta los siguientes artefactos conceptuales y de software:

- ❖ Guías de diseño.
- ❖ Mejores prácticas de programación.
- ❖ Guía de empaquetamiento y publicación de aplicaciones empresariales Java.
- ❖ Asistentes de generación de código basados en patrones de diseño (LEAF Business Wizards).
- ❖ Frameworks de Servicios Horizontales (LEAF Horizontal Services).
- ❖ Componentes reutilizables Java Server Faces.
- ❖ Consolas de producción para seguridad, procesos y calidad de servicio (LOVE Services).

Reduciendo significativamente el tiempo de desarrollo, pruebas y estabilización de cada iteración, tomando como base los artefactos producidos en la disciplina de diseño.



Matriz de Compatibilidad de LEAF

LEAF está focalizado en la arquitectura, diseño y desarrollo de proyectos empresariales de misión crítica con altos requerimientos de servicio. A continuación se muestra, la matriz tecnológica de compatibilidad de LEAF, la soporta los productos de los fabricantes líderes en la industria de software para construcción y puesta en producción de proyectos Java Enterprise Edition:



Valor Agregado de LEAF

La propuesta de valor de LEAF está fundamentada sobre los siguientes beneficios importantes:

- ❖ Aumentar la productividad de proyectos Java Enterprise Edition de misión crítica, hasta en un 70% en servicios de negocio, lo cual permite construir aplicaciones en corto tiempo para entrar en producción.
- ❖ Disminuir significativamente costos en tareas repetitivas de programación.
- ❖ Aumentar la calidad de los componentes Java desarrollados en un proyecto.
- ❖ Disminuir riesgos a través de la adherencia a estándares y mejores prácticas de arquitectura, diseño y programación, probadas en la industria.
- ❖ Portabilidad de sistemas Java Enterprise Edition entre servidores de aplicaciones y bases de datos comerciales y opensource líderes en la industria. De igual forma, el código generado por LEAF se puede utilizar con cualquier entorno de desarrollo para Java.

Quien utiliza LEAF?

LEAF se ha utilizado en proyectos Java Enterprise Edition de compañías prestigiosas del sector financiero (bancos, fondos de pensiones, aseguradoras) en Colombia, logrado resultados importantes en productividad, calidad de servicio y reducción de costos:

- ❖ Sistemas Corporativos - Grupo AVAL
- ❖ Porvenir S.A.
- ❖ Banco AV Villas
- ❖ Opencard S.A.
- ❖ Delima - Marsh

LEAF Business Wizards

LEAF 3i provee servicios de generación de código fuente Java, basado en patrones de diseño y portable entre servidores de aplicaciones, a través de Wizards (asistentes paso a paso) implementados como plugins para la plataforma Eclipse 3.x. LEAF no es una herramienta de desarrollo, por lo tanto el código Java generado se debe utilizar con una herramienta de desarrollo (IDE) para Java en proyectos empresariales de desarrollo de software.

Los asistentes de generación componentes Java de LEAF 3i, proveen aproximadamente un 70% de automatización de desarrollo de servicios de negocio e integración con base de datos. La versión 3i de LEAF incluye los siguientes asistentes de generación de código basado en patrones de diseño:

- ❖ Data Access Objects (DAO), Value Objects (VOs) & Data Validations Wizard.
- ❖ Read-Only Entity (ROE) y Data Transfer Objects (DTO) Wizard.
- ❖ Composite Entities (CE) y Data Transfer Objects (DTO) Wizard.
- ❖ EJB Façade Wizard.

Asistentes de Generación de Código para los Patrones de Diseño J2EE

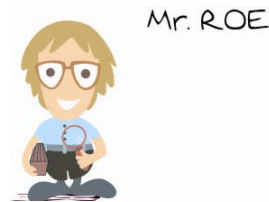


Mr DAO es nuestro asistente de generación de Data Access Objects, Value Objects y Data Validation Objects para integración con base de datos. Cada uno de los DAOs provee 18 servicios comunes para administrar (creación, consulta, actualización y borrado de registros) una tabla de base de datos. Los Value Objects generados encapsulan la información de la BD aislando la aplicación Java de la fuente de datos. Los Meta Value Objects, encapsulan la información necesaria para validar la información contenida en runtime de los Value Objects.



Mr. CE es nuestro asistente de generación de Composite Entities (CE) y Data Transfer Objects (DTO). Los CEs proveen los servicios para mapeo objeto relacional entre aplicaciones Java modeladas con objetos las cuales requieren interactuar con un repositorio de base de datos relacional.

Para arquitecturas orientadas a servicios, LEAF permite exponer los Composite Entities como Web Services para consumo directo desde subsistemas de negocios o aplicaciones cliente.



Mr. ROE es nuestro asistente de generación de entidades de negocio de solo lectura. Estas entidades encapsulan todos los servicios de consulta, para acceso de alto desempeño a información que se encuentra distribuida en varias tablas de una base de datos relacional.

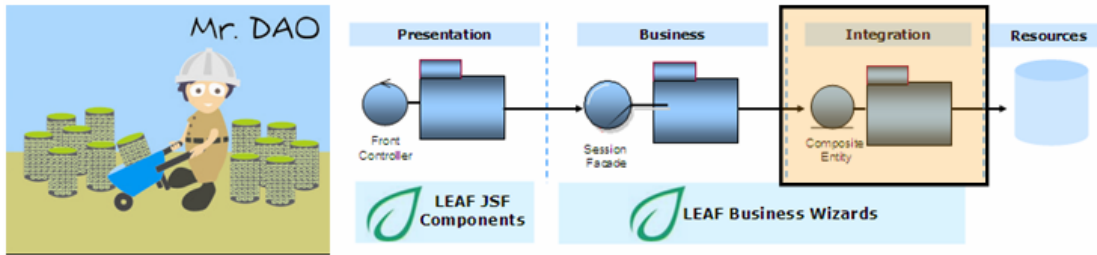
Para arquitecturas orientadas a servicios, LEAF permite exponer los Read-Only Entities como Web Services para consumo directo desde subsistemas de negocios o aplicaciones cliente.



Mr. Mngr es nuestro asistente de generación de componentes de negocio para Java Enterprise Edition. Estos componentes implementan el patrón EJB Façade, en conjunto con los patrones Business Delegate y Service Locator, para exposición de servicios de lógica de negocios de alto desempeño de los subsistemas que conforman una aplicación Java Enterprise Edition.

Para arquitecturas orientadas a servicios, LEAF permite exponer los EJB Façades como Web Services para aplicaciones cliente o procesos de negocio orquestados.

Mr DAO - Data Access Objects (DAO), Value Objects (VOs) & Data Validations Wizard



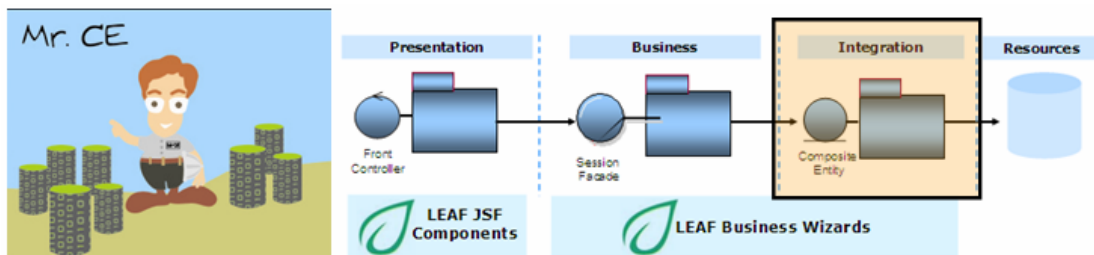
Los DAOs proveen los servicios de acceso a fuentes de datos, desacoplando los componentes de negocios de los detalles de un proveedor de datos específico, incrementando la mantenibilidad y extensibilidad de una aplicación Java.

Mr DAO genera una interfaz y una clase de implementación de DAOs, Value Objects y Meta-Value Objects, para cada tabla que se requiera administrar en bases de datos relacionales. Los DAOs son independientes del servidor de aplicaciones Java y exponen servicios para inserciones individuales y masivas, consultas de alto rendimiento con paginación, ordenamiento y filtros, actualizaciones parciales o totales y eliminación de información.

Los Meta-Value Objects se pueden utilizar para implementar validaciones de la información contenida en los atributos de Value Objects en capas de presentación, aumentando el tiempo de respuesta a los usuarios cliente.

El código fuente generado puede ser utilizado en cualquier IDE para Java.

Mr CE – Composite Entity (CE) & Data Transfer Object (DTO) Wizard



Mr. CE es nuestro asistente de generación de Composite Entities (CE) y Data Transfer Objects (DTO). Los CEs proveen los servicios para mapeo objeto relacional entre aplicaciones Java modeladas con objetos las cuales requieren interactuar con un repositorio de base de datos relacional.

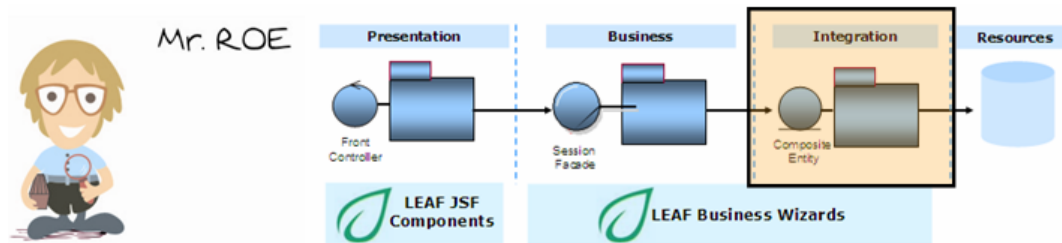
Los servicios de negocios que requieren administrar una entidad de negocios fragmentada en un conjunto de tablas de base de datos, deben utilizar los servicios de los Composite Entities. Si entidad del dominio de negocio, es trivial y está almacenada en una sola tabla, se recomienda utilizar una invocación directa a un DAO.

La información que manejan los Composite Entities, se encapsula en un Data Transfer Object (DTO), en cual es un objeto plano Java, compuesto por varios Value Objects. Los DTOs optimizan el transporte de datos a nivel de red entre tiers lógicos de una aplicación Java y provee la facilidad de manejar la información almacenada en tablas relacionales como objetos a través de mecanismos de herencia y composición.

Para arquitecturas orientadas a servicios, LEAF permite exponer los Composite Entities como Web Services para consumo directo desde subsistemas de negocios o aplicaciones cliente.

El código fuente generado puede ser utilizado en cualquier IDE para Java.

Mr ROE – Read-Only Entity Wizard (ROE) & Data Transfer Object (DTO) Wizard



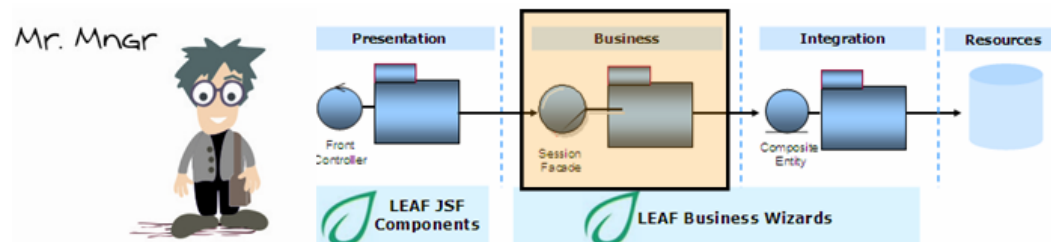
Mr. ROE es nuestro asistente de generación de entidades de negocio de solo lectura, denominadas Read-Only Entities (ROE). Estas entidades encapsulan todos los servicios de consulta, para acceso de alto desempeño a información que se encuentra distribuida en varias tablas de una base de datos relacional, evitando el consumo extra de recursos y problemas de desempeño que surgen en las aplicaciones Java cuando el mapeo de información se realiza en el servidor de aplicaciones, en lugar de la base de datos.

El patrón de Read-Only Entities permite realizar consultas SQL de complejidad media y alta, sobre varias tablas de un modelo relacional, complementando el patrón de los DAOs y los Composite Entities. Este patrón se utiliza en conjunto con Data Transfer Objects (DTOs).

Para arquitecturas orientadas a servicios, LEAF permite exponer los Read-Only Entities como Web Services para consumo directo desde subsistemas de negocios o aplicaciones cliente.

El código fuente generado puede ser utilizado en cualquier IDE para Java.

Mr Mngr – EJB Façade Wizard



Mr. Mngr es nuestro asistente de generación de componentes de negocio para Java Enterprise Edition. Estos componentes implementan el patrón EJB Façade, en conjunto con los patrones Business Delegate y Service Locator, para exposición de servicios de lógica de negocios de alto desempeño de los subsistemas que conforman una aplicación Java Enterprise Edition.

El patrón EJB Façade expone los servicios de negocio de un subsistema, a través un único componente transaccional, el cual realiza un uso óptimo del tráfico de red. Cada invocación a un método del Façade, ejecuta una operación completa de negocios. La generación de las interfaces y clases de los EJB Facades se realiza con base en una interfase Java de negocio suministrada por un desarrollador.

Los componentes generados son los siguientes: Interfaz Remota, Interfaz Home, Clase de Implementación del EJB, Business Delegate y Localizador de Servicios (Service Locator). Estos componentes son 100% portables entre servidores: OC4J, BEA WebLogic Server y JBOSS.

Para arquitecturas orientadas a servicios, LEAF permite exponer los EJB Facades como Web Services para aplicaciones cliente o procesos de negocio orquestados.

El código fuente generado puede ser utilizado en cualquier IDE para Java.

LEAF Enterprise Services

LEAF Horizontal Frameworks (LHF)

LEAF provee un conjunto de frameworks probados en proyectos en producción, documentados y listos para reutilizar en la implementación de proyectos Java Enterprise Edition:

Framework	Descripción
LEAF Security Framework	Framework para autenticación y autorización de usuarios con Directorios LDAP o Base de Datos. Servicios de auditoria para seguimiento de operaciones de negocios y técnicas.
LEAF Exception Framework	Framework para Manejo de Excepciones de Usuario y Sistema. Soporta Internacionalización (i18n).
LEAF Validations Framework	Framework para validaciones de datos locales: datos requeridos, formatos, longitudes de Value Objects. Forward caching de validaciones para interfaces Web. Framework para validación de datos remotos almacenados en un Servidor de Base de Datos.
LEAF QoS Framework	Framework de Calidad de Servicio para monitoreo en producción de los tiempos de respuesta de componentes de negocio e integración con BD. SLAs por componentes individuales.
LEAF Scheduler Framework	Calendarizador de procesos batch Java.

LEAF Operations & Visibility Enterprise Services (LOVE Services)

Las consolas de Administración proveen interfaces Web las cuales permiten la configuración de seguridad de aplicaciones empresariales Java, configuración de procesos batch y permiten monitorear la actividad de un sistema empresarial funcionando en un ambiente de producción incrementando la visibilidad operacional. A continuación se describen las consolas de LEAF disponibles en la versión 3i:

Consola Operacional	Descripción
LEAF Security Console	Consola Web de configuración de seguridad y rastreo de operaciones de negocios y técnicas para aplicaciones Java Enterprise Edition.
LEAF QoS Console	Consola Web de monitoreo para aplicaciones Java Enterprise Edition para visibilidad operacional en ambientes de producción. Configuración de SLAs por componentes de aplicación.
LEAF Scheduler Console	Consola Web para administración delegada de procesos batch Java.

Información de Contacto

Si requiere algún servicio de consultoría, auditoría, desarrollo o educación, por favor comuníquese con nosotros:

Contacto Online

E-mail :
services@lucasian.com

WebSite :
<http://www.lucasian.com>

Lucasian Labs Ltda.

Centro Empresarial,
Ave. Roosevelt, Calle 39. Oficina
503, Cali. Colombia – South
America.

Teléfonos: +57 - (311) 222.7528

TeleFax : +57 - (2) 554.8526

Lucasian Labs Mexico S.C.

Colorines Lt62 Mz 10. Colonia San Clemente
C.P. 01740. México D.F.

Teléfonos: 044 55 2129 2208

TeleFax : +(52) 56351805
