



Facultad 5

Centro de Consultoría y Desarrollo de Arquitecturas Empresariales

**ARQUITECTURA DE INTEGRACION PARA EL MONITOREO Y ANÁLISIS DEL
PROCESO DE FERMENTACIÓN EN EL CENTRO DE INMUNOLOGÍA MOLECULAR**

Tesis presentada en opción al título de Máster en Informática Aplicada

Autora: Ing. Claudia Beatriz Larramendi Ferrás

Tutores: DrC. Orestes Febles Díaz

DrC. Ernesto Chico Veliz

La Habana

2015

DECLARACIÓN JURADA DE AUTORÍA

Yo Claudia Beatriz Larramendi Ferrás, con carnet de identidad 90012126717, declaro que soy la autora principal del resultado que expongo en el presente trabajo titulado “Arquitectura de integración para el monitoreo y análisis del proceso de fermentación en el Centro de Inmunología Molecular”, para optar por el título de Máster en Informática Aplicada.

Declaro que todo lo expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de la Habana a los ____ días del mes de ____ del año _____.

Firma del Maestrante

ÍNDICE

CAPITULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	13
1.1. Centro de Inmunología Molecular	13
1.1.1. Estado actual del CIM.....	13
1.1.2. Proceso de fermentación en el CIM.....	14
1.2. Sistemas informáticos para monitoreo de procesos industriales	16
1.2.1. Funcionalidades.....	16
1.2.2. Componentes arquitectónicos.....	17
1.2.3. Comunicación con otros sistemas.....	19
1.3. Sistemas informáticos para el análisis de procesos industriales	19
1.3.1. Funcionalidades.....	20
1.3.2. Componentes arquitectónicos.....	20
1.3.3. Comunicación con otros sistemas.....	21
1.4. Integración de sistemas	22
1.4.1. Patrones de integración	24
1.4.2. Estilos de integración.....	29
1.5. Estándares para la interoperabilidad.....	30
1.6. Análisis de las herramientas existentes para el monitoreo y análisis de procesos industriales	32
1.7. Conclusiones parciales	33
CAPITULO 2. ARQUITECTURA DE INTEGRACIÓN PARA EL MONITOREO Y ANÁLISIS	34
2.1. Arquitectura de integración propuesta.....	34
2.2. Patrón de integración.....	38
2.2.1. Patrones de diseño SOA.....	40
2.3. Estilo de integración.....	42
2.4. Patrón arquitectónico	44
2.5. Herramientas libres.....	49
2.6. Frameworks de aplicaciones.....	50
2.7. Patrones de diseño	50

2.8.	Protocolos de comunicación con servidores	52
2.9.	Premisas para el uso de la arquitectura propuesta	53
2.9.1.	Escalabilidad.....	53
2.9.2.	Principios de diseño de servicios en SOA.....	54
2.9.3.	Rendimiento.....	55
2.9.1.	Conclusiones parciales	55
CAPITULO 3. VALIDACIÓN DE LA PROPUESTA.....		56
3.1.	Validación por expertos de la arquitectura aplicando la Escala Likert	56
3.2.	Validación de satisfacción de usuarios utilizando la Técnica V.A ladov	59
3.3.	Implementación de la arquitectura de integración	60
3.3.1.	Arquitectura de la solución.....	61
3.3.2.	Despliegue de la solución	63
3.4.	Pruebas a la solución.....	64
3.4.1.	Ambiente de prueba.....	66
3.5.	Diseño experimental	67
3.1.	Conclusiones parciales	69
CONCLUSIONES		70
RECOMENDACIONES		71
BIBLIOGRAFÍA		72
ANEXOS.....		75

ÍNDICE DE FIGURAS

Figura 1 - Arquitectura tecnológica actual del CIM.	14
Figura 2- Mapa de procesos de nivel 1 del macro-proceso IFA.....	15
Figura 3 - Actividades del proceso de Fermentación.....	15
Figura 4 - Arquitectura General de un SCADA.....	17
Figura 5 - Arquitectura General de un DCS.....	18
Figura 6 - Arquitectura general de un sistema de análisis de procesos industriales	21
Figura 9 - Patrón de integración Information Portal	25
Figura 10 - Patrón de integración Data Replication	26
Figura 11 - Patrón de integración Shared Business Function.....	26
Figura 12 - Patrón de integración Service Oriented Architecture	27
Figura 13 - Patrón de integración Distributed Business Process	28
Figura 14 - Patrón de integración Business to Business Integration.....	28
Figura 15 - Arquitectura de integración para el análisis y monitoreo	35
Figura 16 - SOA componentes y relaciones	38
Figura 17 - Patrón SOA Service Bus.....	41
Figura 18 - Patrón SOA Orchestration.....	41
Figura 19 - Patrón SOA Workflowdize.....	42
Figura 20 – Estilo de integración File Transfer	44
Figura 21 - Diagrama general de la arquitectura de integración	46
Figura 22 - Diagrama orientado a componentes de la arquitectura de integración	46
Figura 23 - Diagrama orientado a dominio de la arquitectura de integración.....	47
Figura 24 - Valoración de los expertos sobre la usabilidad de la arquitectura propuesta.....	59
Figura 25 - Nivel de satisfacción de usuarios del SIMAFI CIM.	60
Figura 26 - Arquitectura del “SIMAFI CIM”.	61
Figura 27 - Diagrama de despliegue de la solución implementada.....	64
Figura 28 – Base de datos del proceso de fermentación.....	65
Figura 29 - Gráficas de área mostrando las variables medidas por el SCADA en el CIM.....	66

Figura 30 - Graficas de series de tiempo mostrando los resultados del análisis de muestras	66
Figura 31 - Comportamiento del tiempo en el experimento.	69

ÍNDICE DE TABLAS

Tabla 1 - Comparación y análisis de las arquitecturas de los sistemas similares a SIMAFI CIM en el mercado	32
Tabla 2 - Características de los expertos involucrados en la validación	58
Tabla 3 - Descripción de la función initView	61
Tabla 4 - Descripción de la función getFermentation.....	62
Tabla 5 - Descripción de la función FermentationDAOImplementation.....	62
Tabla 6 - Descripción de la función CSVToDatabaseTask	63
Tabla 7 - Resultados del experimento	68

SÍNTESIS

El Centro de Inmunología Molecular tiene el reto de mejorar su productividad para incrementar los ingresos por concepto de exportación y además lograr un mayor impacto en la red de Salud Pública cubana. En aras de alcanzar esta meta, es fundamental perfeccionar el uso de las capacidades productivas de las que dispone dicha organización, siendo una excelente vía para lograrlo el aprovechamiento y la explotación de las Tecnologías de la Información y las Comunicaciones. Actualmente aunque algunas de las operaciones productivas ejecutadas en el CIM son gestionadas desde una plataforma web, aún existen procesos industriales que requieren integrarse a la informatización que se está llevando a cabo allí. Uno de ellos es la fermentación a pequeña y gran escala; donde se realizan de forma manual muchas de las actividades claves que tributan a la obtención del Ingrediente Farmacéutico Activo.

Partiendo de esta situación, se hace necesaria una arquitectura de integración que pueda ser utilizada para implementar un sistema informático que permita el monitoreo y análisis del proceso de fermentación industrial. Esto es posible aprovechando la existencia de un Sistema de Supervisión, Control y Adquisición de Datos conectado a los fermentadores; potenciando así componentes arquitectónicos tales como los servicios para la obtención de las variables medidas por este SCADA, una base de datos con los registros de los resultados de los análisis de muestreo realizados por los operadores de la planta ANTYTER y una serie de gráficas que combinan toda esta información.

Como validación funcional, se implementó un sistema basado en la arquitectura propuesta. Se probó la adquisición y persistencia satisfactorias de los datos publicados por el SCADA y se realizaron un conjunto de gráficas experimentales con información de una corrida de fermentación y de varios análisis de muestras, que permitieron comprobar la mejora en los tiempos y el aumento de las posibilidades durante el análisis del proceso de fermentación.

Palabras clave: análisis, fermentación, monitoreo, proceso, sistema.

INTRODUCCIÓN

Los avances científico-técnicos marcan la evolución de la sociedad actual. El surgimiento de las Tecnologías de la Información y las Comunicaciones (TIC), así como su vinculación a innumerables esferas sociales, han conllevado a un constante y vertiginoso progreso de las diferentes ciencias. Dicho avance también está dado por la fusión de algunas de ellas, tal es el caso de la bioinformática.

Según la definición del Centro Nacional para la Información Biotecnológica o National Center for Biotechnology Information (NCBI por sus siglas en inglés): es una disciplina científica emergente que utiliza las tecnologías de la información para organizar, analizar y distribuir información biológica con la finalidad de responder preguntas complejas en los dominios de la biología. Es un área de investigación multidisciplinaria, que puede ser ampliamente definida como la unión de dos ciencias: Biología y Computación, impulsada por la incógnita del genoma humano y la promesa de una nueva era en la cual la investigación genómica puede ayudar dramáticamente a mejorar la condición y calidad de vida humana. [García 2009]

Otra forma de conceptualizarla pudiera ser: la disciplina que aplica las TIC para gestionar información biológica. En esta área del conocimiento las computadoras son utilizadas para recopilar, almacenar, analizar e integrar información biológica y genética; la cual pueda ser utilizada para el descubrimiento y producción de medicamentos basados en genes.

También llamada biocomputación, se dedica fundamentalmente a la obtención de conocimiento a partir del análisis computacional de datos biológicos. Este es un campo relativamente reciente que presenta un desarrollo acelerado, basado fundamentalmente en los extraordinarios avances hechos en secuenciación genómica y en las técnicas para la elaboración de mapas genéticos, además de en otras disciplinas biológicas y biotecnológicas. [Michael Nilges 2014]

La bioinformática aporta las herramientas computacionales para modelar, estudiar y comprender los sistemas biológicos. Debido a ello constituye un soporte fundamental para el análisis de los datos generados por los procesos industriales que emplean microorganismos y células para obtener o modificar un producto, comprendidos dentro de la biotecnología.

La definición expuesta en 1992 durante la “Convención sobre diversidad biológica” hace referencia a la biotecnología como: toda aplicación tecnológica que utilice sistemas biológicos y organismos vivos o sus derivados para la creación o modificación de productos o procesos para usos específicos. [Biotecnología 2014]

Según la Asociación Americana para el Avance de la Ciencia o American Association for the Advancement of Science (AAAS, por sus siglas en inglés) biotecnología es: el uso de organismos y procesos biológicos para proporcionar alimentos, productos químicos y servicios para satisfacer las necesidades de los seres humanos. [AAAS 2009]

De esta forma se evidencia como la biotecnología, apoyada en la bioinformática, constituye una vía estratégica para aumentar el bienestar social. El progreso en ambas áreas pretende potenciar la aplicación de las nuevas tecnologías y de los avances genéticos para el beneficio de la salud.

En Cuba desde 1981, se creó el Frente Biológico para la coordinación y jerarquización de las actividades en la esfera de las biociencias, con el objetivo de impulsar el desarrollo y la aplicación de la biotecnología en el país. Los principales centros vinculados a este sector son el Instituto Carlos J. Finlay, el Centro de Ingeniería Genética y Biotecnología (CIGB), Centro de Inmunoensayo (CIE), Centro Nacional de Investigaciones Científicas (CNIC), Instituto de Medicina Tropical Pedro Kourí (IPK), Centro Nacional de Biopreparados (BIOCEN) y el Centro de Inmunología Molecular (CIM). [Cruz 2010]

En el caso específico del CIM, es una empresa de alta tecnología que pertenece al grupo de la Industria Biotecnológica y Farmacéutica (BIOCUBAFARMA). Esta entidad se dedica fundamentalmente a la investigación, desarrollo y fabricación de productos a partir del cultivo de células de mamíferos. Tiene como misión la comercialización de medicamentos biológicos para el tratamiento del cáncer y otras enfermedades. [CIM 2012]

Actualmente dicho centro dispone de varias plantas productivas, entre ellas: Biotech Pharma, EPOVAC, LABEX y ANTYTER. En esta última se desarrollan varios procesos productivos dirigidos a la obtención del Ingrediente Farmacéutico Activo (IFA) de anticuerpos monoclonales, componente fundamental para vacunas y otros biomedicamentos.

Dicha planta opera de manera continua, realizando un ciclo de operaciones productivas que abarca: la preparación de medios de alimentación para células, la descongelación y expansión de células, la fermentación de las mismas y su cosecha.

Teniendo en cuenta que la operación fundamental es la fermentación a pequeña y gran escala de las células que producen el IFA, se requiere prestar especial atención a dicho proceso para garantizar un alto estándar de calidad de sus producciones.

La fermentación se realiza en corridas de hasta 150 días, dentro de bioreactores de 2000 litros de capacidad, equipados para el cultivo de células animales. El contenido de estos tanques es monitoreado permanentemente por un Sistema de Supervisión, Control y Adquisición de Datos o Supervisory Control and Data Acquisition (SCADA, por sus siglas en inglés). A diario los operadores y directivos de la planta deben revisar el estado de las variables medidas por el SCADA (temperatura, nivel de oxígeno, presión, volumen, flujo de alimentación y de cosecha, entre otras).

Hoy en día, el estado de las variables es medido automáticamente y tiempo real, pero el análisis del progreso y/o las variaciones de dichas variables es realizado manualmente. Esto

dificulta la toma de decisiones y dificulta en extremo la realización de análisis de comportamientos y tendencias del proceso de fermentación.

Ante estas dificultades la dirección del CIM solicita la ayuda de la Universidad de las Ciencias Informáticas (UCI) para realizar un diagnóstico de su arquitectura empresarial actual y diseñar un estado deseado de la misma que solucionara los problemas existentes. Así surge el proyecto Diagnóstico de la Arquitectura Empresarial del Dominio IFA (DAEIFA), llevado a cabo por el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE).

En una de las sesiones de trabajo de dicho proyecto surge por parte del cliente la solicitud de un sistema para informatizar el registro de datos del proceso de fermentación que actualmente no son medidos por el SCADA y que además permitiera combinarlos con las variables que si son registradas por dicho SCADA. También se requiere que el sistema informático tenga un módulo de reportes que facilite el análisis mediante una visualización coherente de gráficos que muestren la evolución algunas variables relevantes para el proceso de fermentación.

Las entrevistas a profundidad realizadas a directivos, especialistas y tecnólogos de la entidad evidencian que a pesar de que la fermentación en el CIM, un proceso realizado con tecnologías modernas y regido por principios de las actuales buenas prácticas de producción, sufre varias deficiencias:

- Algunas de sus actividades no están informatizadas.
- El SCADA de los fermentadores registra variables enfocadas al control del equipamiento, no al monitoreo del proceso productivo.
- Los análisis diarios del comportamiento de las células dentro de los fermentadores, son realizados manualmente.
- Es muy complejo realizar un análisis histórico del comportamiento de toda una corrida de fermentación, pues se deben recopilar todos los registros en formato duro y hacer el estudio de cada uno de ellos de forma manual.

A partir de la situación problemática previamente expuesta se identificó el siguiente **problema científico**: ¿Cómo realizar el monitoreo y análisis del proceso de fermentación en el CIM de forma más eficiente?

Para dar respuesta a esta interrogante se plantea como **objeto de estudio**: Arquitecturas de Software de los sistemas informáticos para el monitoreo y análisis de procesos industriales, enmarcado en el **campo de acción**: el proceso de fermentación en el CIM. Para dar solución al problema científico se trazó como **objetivo general** de esta investigación: definir una arquitectura de integración para el monitoreo y análisis del proceso de fermentación en el CIM.

Se plantea como **hipótesis** que si se define una arquitectura de integración y se utiliza en el desarrollo de un sistema informático para el monitoreo y análisis del proceso de fermentación en el CIM, entonces se logrará aumentar la eficiencia del mismo.

Se presentan como **objetivos específicos**:

- Construir el marco teórico referencial de la investigación
- Diagnosticar el estado actual del proceso de fermentación en el CIM, aplicando métodos científicos.
- Definir una arquitectura de integración para el sistema informático de monitoreo y análisis del proceso de fermentación en el CIM.
- Validar la arquitectura de integración definida mediante el desarrollo y puesta en práctica del sistema para el monitoreo y análisis del proceso de fermentación en el CIM.

Para el desarrollo de la investigación se utilizarán los siguientes **Métodos científicos**:

Métodos teóricos:

Analítico-sintético: para construir el marco teórico de la investigación.

Hipotético-deductivo: para elaborar y demostrar la hipótesis de la investigación.

Análisis Histórico-lógico: para conocer el estado del arte y situación actual de las arquitecturas de software para sistemas informáticos empleados en el monitoreo y análisis de procesos industriales.

Métodos empíricos:

Entrevista: para realizar el diagnóstico de la situación actual del proceso de fermentación en el CIM.

Experimento: para validar la arquitectura de software definida, mediante la implantación en el CIM del sistema informático desarrollado utilizando dicha arquitectura.

Observación: para registrar los resultados reales de la incorporación del sistema informático al proceso de fermentación en el CIM.

Medición: para obtener datos cuantitativos del proceso de fermentación en el CIM.

Encuesta: para validar el sistema informático cualitativamente.

Resultados esperados:

Como **aporte práctico** se plantea la realización de un sistema informático que automatice las actividades del proceso de fermentación en el CIM que aún se realizan manualmente.

Como **aporte social** se establece que con el desarrollo de un sistema informático para el monitoreo y análisis del proceso de fermentación en el CIM se obtendrán mejoras en dicho proceso que es clave para la obtención del IFA.

El **aporte tecnológico** reside en el aumento de los reportes alcanzables mediante el desarrollo de un módulo gráfico que permita al usuario seleccionar las variables medidas por el SCADA que desea visualizar en una misma gráfica, facilitando el análisis de comportamientos y tendencias de una corrida del proceso de fermentación en el CIM

Estructura de la tesis

La tesis consta de introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y un cuerpo de anexos, que permite abundar en los temas tratados en la misma.

Capítulo 1: Se presenta el estado actual del CIM y se expone un estudio del estado del arte de los sistemas informáticos para el control y análisis de procesos industriales, así como su arquitectura. También se brindará un estudio de los sistemas de este tipo que hayan sido desarrollados en nuestro país y el tratamiento que realizan los mismos a la arquitectura de software. También se plasman los indicadores relevantes en el área de los procesos biotecnológicos.

Capítulo 2: Se definen los componentes de la arquitectura de software para el sistema de análisis y control del proceso de fermentación en el CIM. Se presentan los estilos arquitectónicos utilizados para lograr que dicha definición sea todo lo flexible que se necesita para dar solución al problema planteado.

Capítulo 3: Se valida la arquitectura funcionalmente utilizándola para el desarrollo de un sistema informático para el análisis y control del proceso de fermentación en el CIM; a su vez se valida cualitativamente utilizando métodos científicos para obtener los resultados de la puesta en práctica de dicho sistema.

CAPITULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En este capítulo queda plasmada la fundamentación del marco teórico. Se incluye aquí el estado del arte del objeto de estudio y la situación actual del campo de acción. También se presenta una panorámica de resultados obtenidos por otros investigadores en el área de los sistemas informáticos para el monitoreo y análisis de procesos industriales. Se plantean los puntos de vista de la autora acerca de dichas temáticas enunciando el punto de partida hacia la solución a desarrollar para el monitoreo y análisis del proceso de fermentación en el Centro de Inmunología Molecular.

1.1. Centro de Inmunología Molecular

El Centro de Inmunología Molecular es una institución biotecnológica cubana dedicada a la investigación, desarrollo y fabricación de productos a partir del cultivo de células de mamíferos cumpliendo con las regulaciones de las actuales Buenas Prácticas de Manufactura (BPM). Esta empresa de alta tecnología pertenece al grupo de la Industria Biotecnológica y Farmacéutica conocido como BIOCUBAFARMA.

Esta importante entidad tiene como misión la investigación, desarrollo, fabricación en gran escala y comercialización de productos biofarmacéuticos en el mercado nacional e internacional. Fundamentalmente trabaja con anticuerpos monoclonales y otras proteínas recombinantes, aplicables al diagnóstico y tratamiento del cáncer y otras enfermedades relacionadas con el sistema inmune.

Por otra parte su visión es lograr impacto a nivel mundial en el tratamiento del cáncer y otras enfermedades crónicas no transmisibles con productos novedosos. Además de desarrollar un crecimiento sostenible de las exportaciones, accediendo a los mercados de los países industrializados. Ha de convertirse en una empresa Biotecnológica de referencia internacional desarrollando una cultura empresarial de excelencia sobre la base de la innovación, diversificación, y la consagración al trabajo.[CIM 2012]

1.1.1. Estado actual del CIM

Actualmente este centro cuenta con más de 1100 trabajadores y cuatro plantas productivas dedicadas al desarrollo de productos biológicos: EPOVAC para la elaboración de proteínas recombinantes, ANTYTER para la producción de anticuerpos monoclonales recombinantes, LABEX para la manufactura de reactivos biológicos para uso en laboratorios y Biotech Pharmaceutical Co. Ltd. para la fabricación de anticuerpos monoclonales.[CIM 2012]

El CIM es una entidad con altos índices de productividad cuyos resultados se convierten en aportes importantes a la economía del país siendo un exportador de excelencia que compite con éxito en el mercado mundial. Comercializa sus productos con 112 países y abastece la red

cubana de salud pública, evidenciándose su impacto económico y social a nivel nacional e internacional.

La fabricación de productos biotecnológicos a partir del cultivo de células vivas requiere de una atención sistemática pues esta actividad se desarrolla dentro de un sector de mercado muy competitivo. Para lograr esto, el CIM ha diseñado una estrategia sobre la base de la preparación profesional y técnica de sus trabajadores, la constante renovación de su tecnología, el incremento de la capacidad productiva de acuerdo a las exigencias de los mercados y el mantenimiento sistemático a las instalaciones auxiliares que apoyan la fabricación. También es importante resaltar que posee un eficiente Sistema de Gestión de la Calidad, sobre la base de una mejora continua como factor elemental para lograr mayor eficiencia y eficacia en todos los procesos productivos y en la calidad del producto final.[CIM 2012]

Actualmente la infraestructura tecnológica disponible es la que se muestra en la siguiente figura:

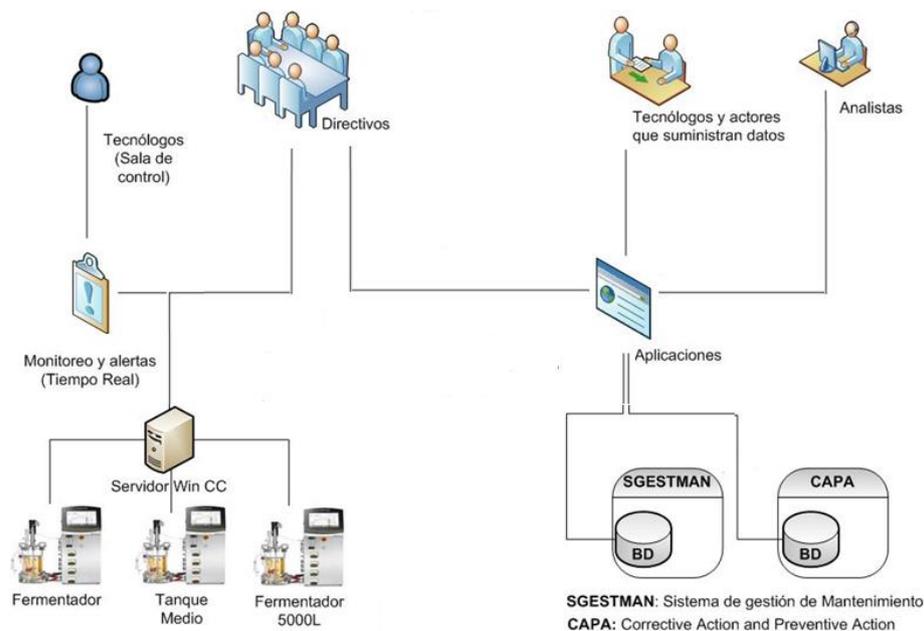


Figura 1 - Arquitectura tecnológica actual del CIM. (Fuente: elaboración propia)

1.1.2. Proceso de fermentación en el CIM

Una de las plantas productivas del CIM es ANTYTER, dedicada fundamentalmente a la fabricación del Ingrediente Farmacéutico Activo (IFA) de anticuerpos monoclonales. La misma abarca un área de 2700 m², dispone de líneas de distribución de aire comprimido libre de aceite y gases industriales así como otras facilidades auxiliares (lavandería, autoclaves, máquinas fregadoras y cámaras frías de almacenamiento) que complementan los servicios de apoyo a la producción.[CIM 2012]

Esta área opera de manera continua, desarrollando diferentes procesos industriales que se aprecian en la siguiente figura:

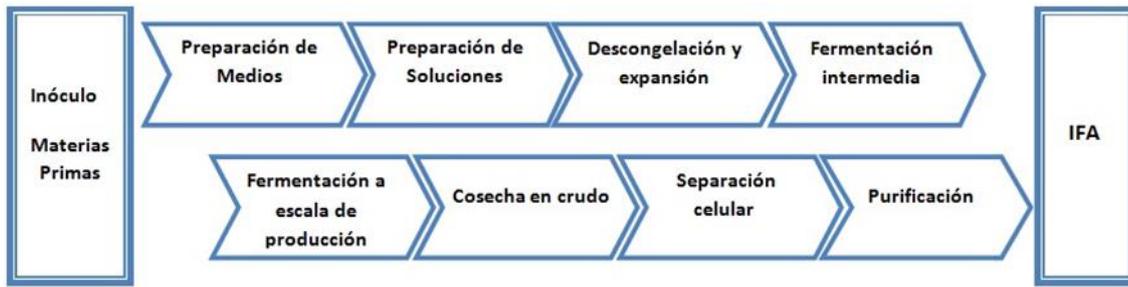


Figura 2- Mapa de procesos de nivel 1 del macro-proceso IFA. (Fuente: elaboración propia)

Dentro del macro-proceso IFA se encuentra el proceso de fermentación, para el cual están planteados los siguientes objetivos: [Llerena 2013]

- Lograr un proceso de fermentación en modo perfusión, a altas concentraciones celulares y de manera ininterrumpida hasta alcanzar el tiempo máximo permitido regulatoriamente (150 días).
- Lograr un proceso de incorporación continua de conocimiento al proceso de fermentación, que constituye la base tecnológica del CIM.

Para el cumplimiento de estos objetivos se ejecuta el proceso de fermentación en sus dos versiones: intermedia e industrial. Se utilizan fermentadores de 2000 L de capacidad, equipados para el cultivo de células animales en modo continuo y perfusión, los cuales permiten alcanzar capacidades de producción de hasta 45 kg de IFAs de anticuerpos monoclonales.

Los tanques dentro de los cuales se realiza la fermentación están equipados con sistemas automatizados que miden variables como temperatura, presión, nivel de oxígeno y controlan las válvulas que regulan el volumen del contenido del fermentador.

La fermentación tiene como subprocesos: preparación del fermentador, alimentación de medio y muestreo de variables. A continuación se representa la secuencia de actividades de la fermentación:

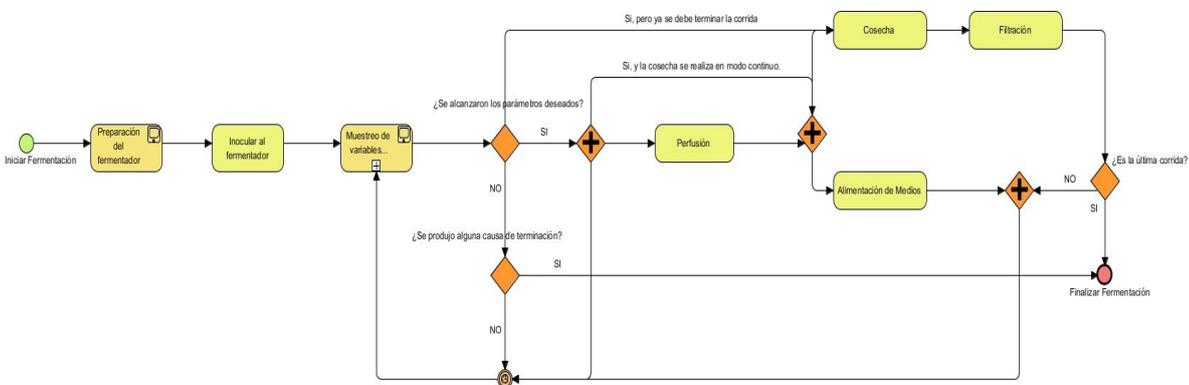


Figura 3 - Actividades del proceso de Fermentación (Fuente: elaboración propia)

De manera general la fermentación se realiza durante corridas que duran hasta 150 días, aunque el proceso se termina antes en caso de ocurrir un fallo mecánico o al detectar contaminación.

Este proceso es clave para la obtención de IFA, debido a ello se debe realizar manteniendo los parámetros de calidad definidos por el Departamento de Control de Procesos. En aras de cumplir con dichos parámetros la fermentación es monitoreada a diario, se verifica manualmente la concentración de anticuerpos y células, la pureza del contenido del fermentador y se registran los cambios ocurridos dentro del fermentador.

Estos registros diarios se realizan de forma manual y en formato duro, por lo que se dificulta la realización del análisis del progreso, variaciones y tendencias de las variables de toda una corrida de fermentación. Actualmente para realizar este tipo de estudios se debe procesar un alto volumen de documentos físicos, ello dificulta la toma de decisiones y la obtención de conocimiento científico que pueda ser utilizado para incrementar los niveles de productividad del CIM.

1.2. Sistemas informáticos para monitoreo de procesos industriales

La ejecución de un proceso industrial requiere no sólo administrar la mano de obra, la materia prima y la maquinaria de la planta; también es fundamental disponer de los datos referentes a la evolución y el estado actual de dicho proceso. El dominio de dichos datos es imprescindible para la oportuna toma de decisiones en aras de mejorar la calidad del producto, incrementar la eficiencia en la producción y conservar la inversión de capital realizada en la planta.

Una estrategia factible para alcanzar un ambiente de producción industrial a alta escala y con la calidad requerida para mantener la competitividad en el mercado internacional actual, consiste en incorporar herramientas informáticas que permitan visualizar, gestionar y almacenar la información del proceso industrial en cuestión.

1.2.1. Funcionalidades

Dentro de las soluciones informáticas aplicables al control de procesos industriales existen dos variantes fundamentales: los Sistemas de Control Distribuido (DCS por sus siglas del inglés *Distributed Control System*) y los Sistemas de Control y Adquisición de Datos (SCADA por sus siglas del inglés *Supervisory Control and Data Acquisition*). En ambos casos las funcionalidades principales son:

- Adquisición y almacenado de datos: para obtener, procesar y almacenar la información recibida de los sensores o actuadores; de manera continua y confiable. [A. Daneels 2009]
- Ejecución de acciones de control: para modificar la evolución del proceso, actuando sobre los reguladores autónomos básicos (consignas, alarmas, eventos, etc.) o incluso directamente sobre el proceso mediante las salidas conectadas.

- Supervisión de variables de proceso: para monitorear la realidad de los procesos se realiza una representación gráfica y animada del mismo, permitiendo a los operadores una interrelación de los equipos físicos de la planta con los equipos virtuales de las interfaces gráficas de usuario (GUI por sus siglas del inglés *Graphic User Interface*). [García 2014]

Existen características subyacentes que tributan a la correcta ejecución de las funcionalidades previamente expuestas y que son igualmente importantes: [Whitt 2012]

- Arquitectura de software abierta y flexible, con capacidad de ampliación y adaptación.
- Conectividad con otras aplicaciones y bases de datos, locales o distribuidas en redes de comunicación.
- Módulo para la explotación de los datos adquiridos para gestión de la calidad, control estadístico, gestión de la producción y gestión administrativa y financiera.

1.2.2. Componentes arquitectónicos

En aras de identificar componentes arquitectónicos para incluirlos en la arquitectura desarrollada, se estudiaron las generalidades de los sistemas SCADA y DCS respectivamente. Las siguientes imágenes muestran ambas arquitecturas:

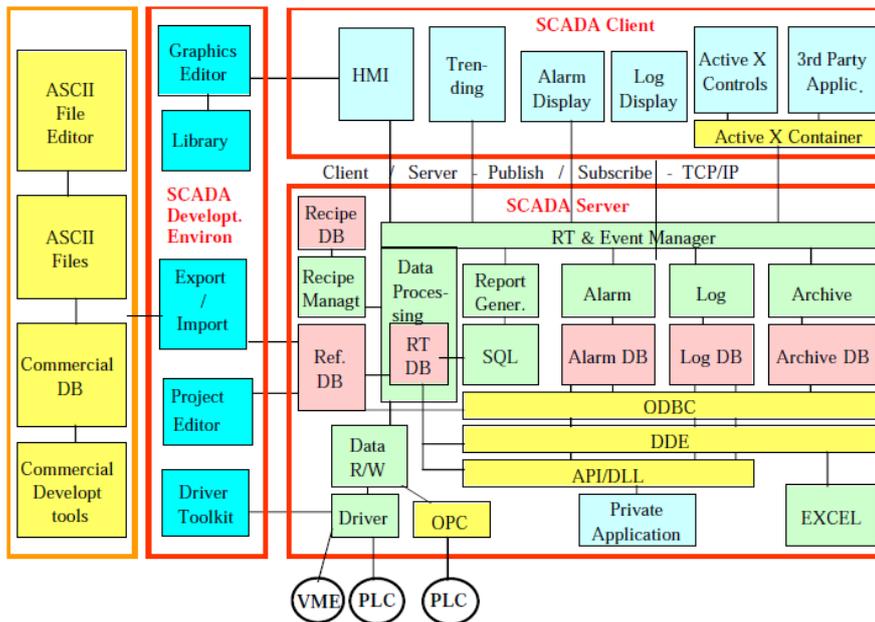


Figura 4 - Arquitectura General de un SCADA (Fuente: [A. Daneels 2009])

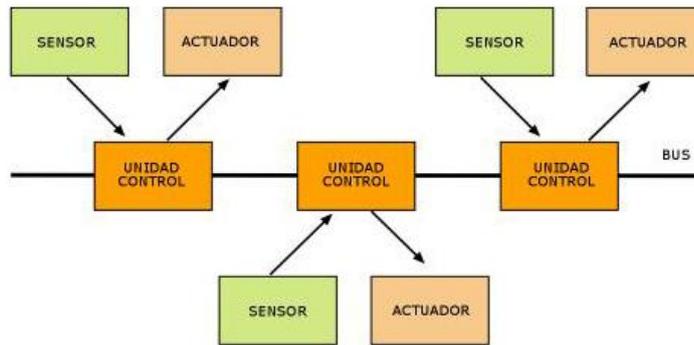


Figura 5 - Arquitectura General de un DCS

A continuación se exponen los componentes que fueron seleccionados por su relevancia para el monitoreo de procesos industriales y por ser aplicables al entorno actual del proceso de fermentación en el CIM:

- Dispositivos de campo. Sensores que permanentemente monitorean el entorno del proceso industrial; con el objeto de generar un evento que será gestionado por los actuadores. Estos a su vez son elementos de salida capaces de recibir una orden del controlador y realizar una acción (encendido/apagado, subida/bajada de persiana, apertura/cierre de electroválvula). [Román 2013]
- Base de datos. Considerada el núcleo lógico, contiene los valores de las variables relativas al contexto del proceso industrial, almacenados sistemáticamente para su posterior uso. La gestión de datos debe realizarse con bajos tiempos de acceso. Se suele utilizar Conectividad Abierta de Base de Datos (ODBC por sus siglas del inglés *Open Database Connectivity*); como interfaz estratégica para acceder a los datos desde cualquier aplicación, sin importar qué Sistema de Gestión de Bases de Datos (DBMS por sus siglas del inglés *Database Management System*) los almacene. [Deza 2011]
- Interfaz humano-máquina (HMI por sus siglas del inglés *Human-Machine Interface*). Se debe considerar que en particular las interfaces para el monitoreo y control de procesos industriales poseen características distintivas, muchas de ellas críticas, que impactan fuertemente en la comunicación que se debe propiciar. Cuestiones importantes son la cantidad de información a monitorear y el tipo de responsabilidad de control que se va a delegar en el operador. [Sergio Martig 2014]

Muchas aplicaciones de esta naturaleza se diseñaron e implementaron con una vista monolítica, o sea, son aplicaciones que no están preparadas para interactuar con funcionalidades externas. Existen variantes para lograr una integración intrínseca e independiente de la tecnología subyacente. En la literatura relacionada al objeto de estudio de esta investigación, se muestran escenarios donde se modifican las aplicaciones analizadas para aumentar sus capacidades comunicativas o se implanta una arquitectura que soporte determinado nivel de integración y al

mismo tiempo permita una flexibilidad en la ejecución de los procesos claves de la organización que la utiliza.

1.2.3. Comunicación con otros sistemas

Los retos que enfrentan las industrias de procesos de hoy han aumentado debido al estado inestable de la economía mundial. En este tipo de entorno, las demandas del sistema de control de procesos también han crecido, llegando más allá de simples funciones de regulación. Los sistemas de control se han convertido en una herramienta para manejar las plantas industriales, recortar el consumo de energía, reducir costos, disminuir las emisiones y proporcionar información rápida y precisa del piso de campo para apoyar la producción y la toma de decisiones de negocios.

En un escenario industrial donde se utilicen herramientas informáticas para el monitoreo y control de los procesos, normalmente coexisten más de un sistema. Es necesario que la información llegue a todos ellos de forma transparente, desde los que están en el piso de operaciones de la planta hasta los de inteligencia empresarial que se encuentran en las computadoras de los directivos y analistas.

Es recomendable también que sea posible el acceso a la planta y los datos del proceso desde fuera de la planta, mediante clientes web y dispositivos móviles inteligentes. Esto es necesario para personas que tienen que acceder a la planta en tiempo real y procesar información desde lugares remotos.

Para interactuar con otros sistemas los SCADA normalmente utilizan un servidor OPC (por sus siglas del inglés *OLE for Process Control*), el cual dará soporte de comunicación entre todos los elementos relacionados en el entorno de la red local. OPC es una arquitectura de comunicación para intercambio de datos en tiempo real de diferentes aplicaciones y dispositivos de control de diferentes fabricantes. Así será posible acceder a datos del proceso desde aplicaciones externas al sistema SCADA, a través de la red LAN (por sus siglas del inglés *Local Area Network*), estableciendo los procedimientos de seguridad necesarios en el servidor OPC. [Gómez 2015]

1.3. Sistemas informáticos para el análisis de procesos industriales

El análisis de procesos juega un papel fundamental en el éxito de su negocio para infinidad de industrias. Los analizadores de procesos realizan un aporte esencial a la optimización del proceso y la calidad con que se realiza el mismo, además permiten una producción ecológica y sostenible en prácticamente cada segmento de la industria, en cualquier parte del mundo.

También pretende eliminar las deficiencias existentes en el proceso objeto, a partir de representaciones gráficas lo cual aporta una mejor visualización de la distribución tanto de la maquinaria, equipo y del área de trabajo dentro de la planta productiva.

1.3.1. Funcionalidades

Los constantes cambios en las tecnologías de información han ocasionado gran impacto en las empresas industriales, tanto que han llegado a determinar la ventaja competitiva de muchas, adoptando estrategias basadas en los sistemas informáticos, para afrontar el exigente entorno de un mercado internacional cada vez más competitivo. Según el grado de avance de automatización alcanzado en determinada empresa como bien pudiera ser el Centro de Inmunología Molecular, hace que el sistema de información inherente al proceso productivo, represente una herramienta inductora de eficiencia y, por ende, de productividad.

En este entorno un sistema de información se refiere a un conjunto de componentes interrelacionados que capturan, almacenan, procesan y distribuyen la información para apoyar la toma de decisiones, el control, el análisis y visión en los procesos de negocio de una institución. Se debe resaltar la importancia clave de las herramientas informáticas en la generación e intercambio de información entre procesos clave y de apoyo. El sistema de información debe recopilar datos que generados por las distintas actividades, son necesarios para el funcionamiento de otras. El propio sistema también debe ser capaz de distribuir dicha información a cada actividad, por lo que juega un rol decisivo para la coordinación de actividades primarias entre sí.

En la coordinación de actividades del proceso productivo, los sistemas de procesamiento de operaciones (SPO), mediante las tecnologías de información, realizan y registran las operaciones, permiten el enlace oportuno con proveedores de insumos, y además son los únicos medios de generación de información para la evaluación del desempeño. Para inducir la eficiencia en el proceso productivo debe existir una coordinación tal de actividades inherentes, que se garantice el adecuado manejo y uso de insumos y demás recursos, que son elementos claves, no sólo para lograr la calidad y especificaciones del producto, sino también por su incidencia en el costo del mismo. [Rodríguez Medina 2012]

1.3.2. Componentes arquitectónicos

Las alternativas de sistemas de información aplicables al análisis del proceso productivo de cualquier empresa crecen cada día, pero la decisión de adoptar alguna debe apoyarse en la habilidad para determinar que la elección de una determinada tecnología tenga un impacto estratégico; que promueva cambios para obtener ventaja competitiva; y además, que sea susceptible de adaptación siguiendo el enfoque de la cadena de valor, para garantizar la coordinación de las actividades involucradas en el cambio, lo cual induce a la eficiencia. Es por ello que un sistema informático para el análisis de procesos industriales del sector biotecnológico tal como la fermentación industrial en el CIM tiene entre sus componentes:

- Fuentes u orígenes de datos.
- Receptores de datos.

- Contenedores o almacenes de datos.
- Motores de análisis de datos.
- Reportadores de resultados de análisis de datos.

La siguiente imagen muestra funcionamiento general y los componentes arquitectónicos de este tipo de sistemas:

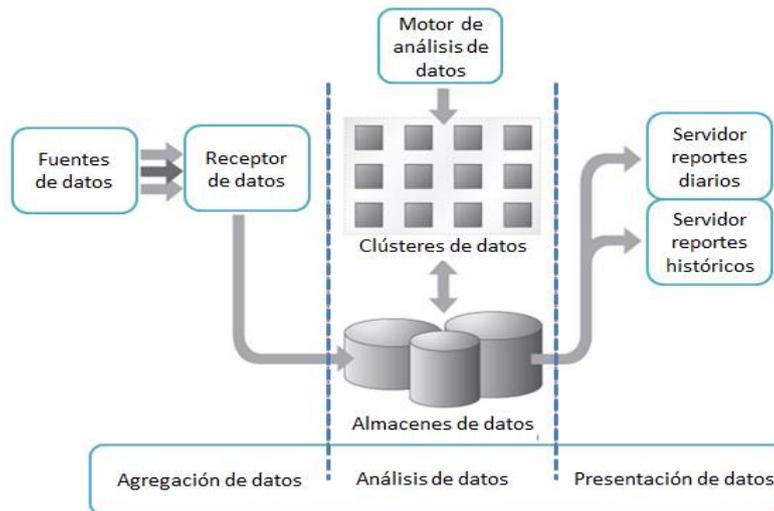


Figura 6 - Arquitectura general de un sistema de análisis de procesos industriales (Fuente: elaboración propia)

Es válido mencionar que la información pasa por tres etapas diferentes de procesamiento dentro de estos sistemas: agregación, análisis y presentación. En la primera etapa se obtienen los datos del proceso industrial, posteriormente en una segunda etapa se persisten internamente y se analizan mediante disímiles mecanismos, finalmente en la tercera fase se procede a mostrar los resultados de los análisis realizados.

1.3.3. Comunicación con otros sistemas

Las organizaciones cuentan con muchos sistemas informáticos que trabajan a los diferentes niveles y funciones. Se tienen los sistemas de soporte a ejecutivos (SSE) al nivel estratégico; sistemas de información para la administración (SIA) y sistemas de soporte para la toma de decisiones (SSD), al nivel de administración o gerencial; sistemas de trabajo del conocimiento (STC) y sistemas de automatización de oficina (SAO) al nivel de conocimientos; y sistemas de procesamiento de operaciones (SPO) al nivel operativo. [Salvendy 2005]

Para el intercambio entre estos sistemas se plantea un enfoque de “usar información de estado para accionar en relación a un recurso”; donde el rol de los sistemas de información modernos, mediante tecnologías avanzadas en el área de manufactura y producción, tiene que ver precisamente con proveer información acerca del estado de los insumos, mediante el uso de los demás recursos, y apoyar la coordinación de actividades.

El estado de un recurso, corresponde a los cambios que experimentan los insumos en el proceso de producción de un bien, para lo cual pasan por distintas etapas que podemos identificar como: inicial,

intermedia y final. En la etapa inicial las actividades de logística interna deben coordinarse de tal forma que los insumos se reciban, almacenen y se lleven a producción en forma adecuada y oportunamente, y al mismo tiempo se genere información para el control de inventarios de materiales y planificación de nuevas compras. La etapa intermedia está referida al proceso mismo de transformación de los insumos en la forma final del producto; aquí las actividades primarias corresponden a la categoría de operaciones, que también requieren de intercambio de información acerca de los distintos estados de cambio de los insumos, lo cual garantice el logro del estado final deseado del producto. En la etapa final, la coordinación de actividades de logística externa, aspira el adecuado manejo del producto para su posterior distribución a los compradores, así como para el control de inventarios de productos terminados; tal coordinación depende del intercambio de información de actividades y el incremento de la eficiencia derivada, la cual se debe a la reducción de costos y tiempo en el manejo del tipo de inventario mencionado.

La razón funcional del proceso productivo se materializa cuando a partir del insumo, sometido a distintas fases de transformación, se obtiene el producto, por ello es vital hacer un seguimiento coordinado de todo el proceso de transformación, de manera que se pueden evitar al máximo situaciones no deseadas que ocasionen incrementos en costos y tiempo, como son el desperdicio, el reproceso y, en especial, aquellas que debilitan la imagen del producto por desmejoras de la calidad. Todo lo anterior solo es posible de lograr con calidad utilizando de manera coordinada e integrada todos los sistemas de información que gestionan y analizan las diferentes etapas de los procesos industriales.

1.4. Integración de sistemas

La infraestructura tecnológica de las empresas están típicamente formada por disímiles aplicaciones que pueden ser construidas a la medida, adquiridos a un tercero, parte de un sistema heredado o una combinación de los mismos; que operan en múltiples niveles de diferentes plataformas de sistemas operativos. Es común encontrar una empresa que cuenta con sitios web diferentes, varios casos de aplicaciones para gestión de procesos y diversas soluciones departamentales.

En este fenómeno intervienen varios factores: [Hohpe 2003]

- El desarrollo de aplicaciones de negocio no es trivial. La creación de una única gran aplicación, para dirigir un negocio completo es casi imposible. Los proveedores de sistemas de planificación de recursos empresariales ERP (por sus siglas del inglés *Enterprise Resource Planning*) han tenido cierto éxito en la creación de grandes aplicaciones de negocios que sirven como punto de integración en las empresas. La realidad, sin embargo, es que hasta las grandes compañías desarrolladoras como SAP, Oracle, Peoplesoft y similares sólo realizan una fracción de las funciones necesarias en un empresa típica.
- La difusión de las funciones de negocio a través de múltiples aplicaciones ofrece la flexibilidad para seleccionar “el mejor” paquete de contabilidad, “el mejor” gestor de clientes o el sistema de procesamiento de pedidos que mejor se adapte a las necesidades

del negocio. Una ventanilla única para las aplicaciones empresariales por lo general no es lo que las organizaciones tecnológicas están interesadas en desarrollar, ni es posible dado el número requisitos empresariales individuales.

Los desarrolladores han aprendido a atender a esta preferencia y ofrecer aplicaciones centradas alrededor de una función básica específica. Sin embargo, la siempre presente necesidad de añadir nuevas funcionalidades a los paquetes de software existentes ha causado algunos efectos colaterales. Por ejemplo, muchos sistemas de facturación han comenzado a incorporar atención al cliente y la funcionalidad de contabilidad. Asimismo, el fabricante de software de atención al cliente toma una parte en la implementación de funciones de facturación simples, tales como disputas o ajustes. Mantener una separación funcional clara entre los sistemas es difícil, más aún cuando a pesar de que la petición fue procesada por tres sistemas informáticos diferentes, para el cliente es una transacción única que debe ser completada con rapidez, seguridad y calidad.

Con el fin de apoyar los procesos de negocio comunes y el intercambio de datos entre aplicaciones, estas aplicaciones deben integrarse. La integración de aplicaciones tiene que proporcionar el intercambio de datos eficiente, confiable y seguro entre múltiples sistemas empresariales.

Desafortunadamente la integración empresarial a nivel de sistemas informáticos no es tarea fácil. Por definición, la integración de la empresa tiene que lidiar con muchas aplicaciones que se ejecutan en plataformas múltiples en diferentes lugares. Los proveedores de software ofrecen suites para la integración de aplicaciones empresariales EAI (por sus siglas del inglés *enterprise application integration*) que proporcionan plataformas de integración entre lenguajes. Sin embargo, esta técnica abarca sólo una pequeña parte de las complejidades de integración. Los verdaderos desafíos de la integración incluyen temas de negocios y tecnologías. A continuación se exponen algunos de los principales retos de la integración de sistemas: [Hohpe 2003]

- Las aplicaciones de negocios están centradas en un área funcional específica, como la gestión de relaciones con los clientes, facturación, finanzas, entre otras. Por esta razón, muchos grupos de desarrollo de TI se organizan en alineación con estas áreas funcionales. Para una integración empresarial con éxito no sólo se necesitan para establecer una comunicación entre varios sistemas informáticos, sino también entre las unidades de negocio y departamentos que ya no intercambiarían con solamente con una aplicación específica, ya que cada aplicación es ahora parte de un flujo global de aplicaciones integradas y servicios.
- Debido a su amplio alcance, los esfuerzos de integración suelen tener implicaciones de largo alcance en el negocio. Una vez que el procesamiento de las funciones de negocio más críticos se incorpora en una solución de integración, el buen funcionamiento de esa

solución se convierte en vital para el negocio. Un defecto o mal comportamiento de la solución de integración pueden costar la pérdida monetaria considerable en pedidos perdidos, pagos mal encaminados y clientes descontentos.

- Una limitación importante del desarrollo de soluciones de integración es el control limitado que los desarrolladores de integración suelen tener sobre las aplicaciones participantes. En la mayoría de los casos, las aplicaciones son sistemas legados o paquetes de aplicaciones que no se pueden cambiar sólo para ser conectadas a una solución de integración. Esto a menudo deja a los desarrolladores de integración en una situación en la que tienen que compensar las deficiencias dentro de las aplicaciones o las diferencias entre las aplicaciones.
- A pesar de la necesidad de ampliar el margen para soluciones de integración, a pocos estándares se han establecido en este dominio. La llegada de XML, XSL y servicios Web sin duda marcan el avance más significativo de características basadas en estándares en una solución de integración. Es válido recordar que la falta de interoperabilidad entre los productos de "normas conformes" fue uno de los principales obstáculos para CORBA, que ofrecía una solución técnica sofisticada para la integración de sistemas.
- Además, las normas existentes de servicios Web XML abordan sólo una fracción de los desafíos de la integración. La estandarización de todo el intercambio de datos a XML se puede comparar a la escritura de todos los documentos que utilizan un alfabeto común. A pesar de que el alfabeto es común, todavía se utiliza para representar a muchos idiomas y dialectos, que no pueden ser entendidos fácilmente por todos los lectores. Lo mismo es aplicable a la integración empresarial. La existencia de una presentación común (por ejemplo, XML) no implica semántica común. La resolución de diferencias semánticas entre sistemas resulta ser una tarea particularmente difícil y requiere mucho tiempo.
- El desarrollo de una solución EAI es un reto en sí mismo, la operación y mantenimiento de una solución de este tipo puede ser aún más desalentador. La combinación de tecnologías y la naturaleza distribuida de soluciones EAI hacen de la implementación y el monitoreo tareas complejas, además solución de fallos que requieren una combinación de habilidades que en muchos casos, no existen dentro de las operaciones de TI o se distribuyen en muchas personas diferentes.

1.4.1. Patrones de integración

No hay respuestas simples para la integración empresarial. A pesar de que la integración es un tema muy amplio y difícil, se han resuelto bastantes problemas de integración definiendo diseños comunes en el desarrollo de funcionalidades relacionadas con la integración de aplicaciones, esto se conoce como patrones de integración. Los patrones especifican una manera estándar de

realizar ciertas tareas y ayudan a conocer con un lenguaje común determinadas cosas que se desarrollan habitualmente, estos han evolucionado a través del tiempo por ensayo y error o de otros arquitectos de integración con experiencia

Para la autora de esta investigación integración significa la interconexión de los sistemas informáticos, procesos de negocio y personas. Para la comprensión de los patrones de integración es útil estudiar algunos de los escenarios de integración más comunes: [Hohpe 2003]

- Information Portals
- Data Replication
- Shared Business Functions
- Service-Oriented Architectures
- Distributed Business Processes
- Business-to-Business Integration

Esta lista no es de ninguna manera una taxonomía completa de todos los elementos que influyen en la integración pero sí ayuda a ilustrar el tipo de soluciones que los arquitectos de integración construyen. Muchos proyectos de integración consisten en una combinación de varios tipos de integración. A continuación se detallan los seis patrones de integración más utilizados:

Information Portal



Figura 7 - Patrón de integración Information Portal (Fuente: [Hohpe 2003])

Muchos usuarios de negocio tienen que acceder a más de un sistema para responder a una pregunta específica o para realizar una sola función empresarial. Por ejemplo, para verificar el estado de un pedido, un representante de servicio al cliente puede tener acceso al sistema de gestión de pedidos en el mainframe más iniciar sesión en el sistema que gestiona los pedidos realizados a través de Internet.

El patrón Portal de Información muestra información agregada de múltiples fuentes en una sola pantalla para evitar que los de acceso de usuario múltiples sistemas informáticos. En su implementación más simple divide la pantalla en varias zonas, cada una de las cuales muestra la información de un sistema diferente. En versiones más sofisticadas proporciona una interacción

limitada entre zonas, por ejemplo, cuando un usuario selecciona un elemento de una lista en la zona A, la zona B se actualiza con información detallada sobre el elemento seleccionado.

Data Replication

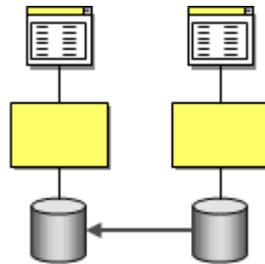


Figura 8 - Patrón de integración Data Replication (Fuente: [Hohpe 2003])

Los diferentes sistemas de negocios necesitan tener acceso a los mismos datos. Por ejemplo, la dirección de un cliente puede ser utilizado en el sistema de atención al cliente (cuando el cliente llama para cambiar el pedido), el sistema de contabilidad (para calcular el impuesto de ventas), el sistema de envío (a etiquetar el envío) y el sistema de facturación (para enviar una factura). Muchos de estos sistemas van a tener sus propios almacenes de datos para guardar la información relacionada con el cliente. Cuando un cliente llama para cambiar su dirección entonces todos estos sistemas tienen que cambiar su copia de este dato del cliente. Esto se puede lograr mediante la aplicación de una estrategia de integración basada en la replicación de datos.

Hay muchas maneras diferentes de implementar la replicación de datos. Por ejemplo, algunos proveedores de bases de datos se acumulan funciones de replicación en la base de datos, podemos exportar datos en archivos y volver a importar en el otro sistema, o podemos usar middleware orientado a mensajes para el transporte de registros de datos dentro de los mensajes.

Shared Business Function

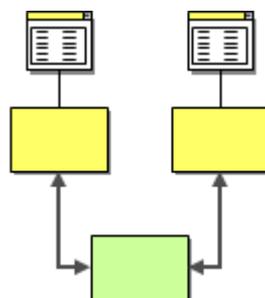


Figura 9 - Patrón de integración Shared Business Function (Fuente: [Hohpe 2003])

De la misma manera que muchas aplicaciones de negocios almacenan datos redundantes, sino que también tienden a implementar la funcionalidad redundante. Múltiples sistemas pueden necesitar para comprobar si un número de seguro social es válido, si la dirección coincide con el código postal especificada o si un elemento en particular está en stock. Tiene sentido empresarial para exponer estas funciones en función empresarial compartida que se implementa una vez y está disponible como un servicio a otros sistemas.

Una función de negocio compartido puede abordar algunas de las mismas necesidades que la replicación de datos. Por ejemplo, podríamos implementar una función denominada "Obtener dirección del Cliente que podría permitir a otros sistemas para solicitar la dirección del cliente cuando es necesario y no siempre guardar una copia redundante. La decisión entre estos dos enfoques es impulsado por una serie de criterios, tales como la cantidad de control que tenemos sobre los sistemas (llamar a una función compartida suele ser más intrusivo que cargar datos en la base de datos) o la tasa de cambio (una dirección puede ser necesario cambiar con frecuencia, pero con muy poca frecuencia).

Service Oriented Architecture

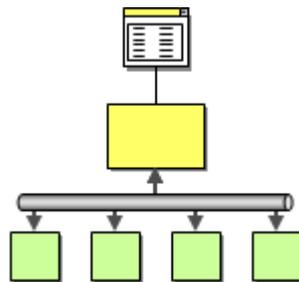


Figura 10 - Patrón de integración Service Oriented Architecture (Fuente: [Hohpe 2003])

Este patrón está basado en los servicios que implementan para compartir funciones de negocios. Un servicio es una función bien definida que es universalmente disponible y responde a las peticiones de "consumidores de servicios". Una vez que una empresa reúne una colección de servicios útiles, la gestión de los servicios se convierte en una función importante. En primer lugar, las aplicaciones necesitan alguna forma de directorio de servicios, una lista centralizada de todos los servicios disponibles. En segundo lugar, cada servicio necesita describir su interfaz de tal manera que una aplicación puede "negociar" un contrato de comunicaciones con el servicio. Estas dos funciones, de descubrimiento de servicios y la negociación, son los elementos clave que conforman una arquitectura orientada a servicios.

Las Arquitecturas Orientadas a Servicios (SOA) desdibujan la línea entre la integración y aplicaciones distribuidas. Una nueva aplicación se puede desarrollar utilizando los servicios existentes que a su vez pueden ser proporcionados por otras aplicaciones remotas. Por lo tanto,

llamar a un servicio se puede considerar la integración entre las dos aplicaciones. Por otro lado una arquitectura orientada a servicios por lo general proporciona herramientas que hacen que pidan un servicio externo casi tan simple como llamar a un método local. Debido a que todos los servicios están disponibles de manera consistente, SOA se refiere a veces como "arquitecturas de bus de servicios".

Distributed Business Process

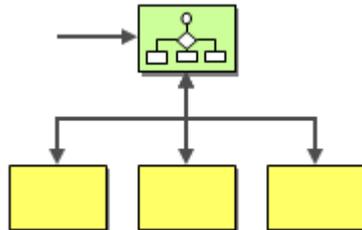


Figura 11 - Patrón de integración Distributed Business Process (Fuente: [Hohpe 2003])

Uno de los principales impulsores de la integración es el hecho de que una sola transacción de negocios a menudo se propaga a través de muchos sistemas diferentes. En la mayoría de los casos, todas las funciones relevantes se incorporan dentro de las aplicaciones existentes. Lo que falta es la coordinación entre las aplicaciones. Por lo tanto, este patrón añade un componente de gestión de procesos de negocio que gestiona la ejecución de una función de negocio a través de múltiples sistemas existentes.

Los límites entre una arquitectura orientada a servicios y una empresa distribuida pueden desdibujarse. Por ejemplo, se podría exponer a todas las funciones empresariales relevantes como el servicio y luego codificar los procesos de negocio dentro de una aplicación que tiene acceso a todos los servicios a través de una SOA.

Business-to-Business Integration

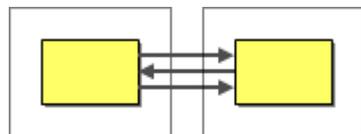


Figura 12 - Patrón de integración Business to Business Integration (Fuente: [Hohpe 2003])

Los patrones anteriores han considerado principalmente la interacción entre las aplicaciones y funciones de negocios dentro de una empresa. En muchos casos, las funciones de negocio pueden estar disponibles de proveedores externos o socios comerciales. Por ejemplo, la compañía naviera puede proporcionar un servicio para los clientes para calcular los envíos de costes de envío o de pista. O una empresa puede utilizar un proveedor externo para calcular las

tasas de impuestos sobre las ventas. Del mismo modo, la integración se produce con frecuencia entre los socios comerciales. Un cliente puede ponerse en contacto con un distribuidor para preguntar sobre el precio y la disponibilidad de un elemento. En respuesta, el minorista puede solicitar al proveedor para el estado de un envío que contenga el elemento fuera de stock. Muchas de las consideraciones anteriores se aplican igualmente a la integración business-to-business. Sin embargo, la comunicación a través de Internet o cualquier otra red general plantea nuevas cuestiones relacionadas con los protocolos de transporte y seguridad.

Luego de este análisis se puede plantear que en un escenario industrial si utilizan adecuadamente los patrones de integración pueden ayudar a llenar la amplia brecha entre la visión de alto nivel de integración y la implementación real del sistema. Dadas las condiciones actuales del entorno tecnológico del Centro de Inmunología Molecular, se determina utilizar en esta investigación el patrón de integración SOA para implantar en esta entidad los componentes de una Arquitectura Orientada a Servicios que comuniquen las aplicaciones informáticas existentes y otras que puedan desarrollarse.

1.4.2. Estilos de integración

Desde los inicios de la era computacional hasta la actualidad la necesidad de comunicación entre los sistemas ha permanecido latente en desarrolladores, arquitectos y decisores. Existe un grupo importante de vías, procedimientos y estándares que deben ser tenidos en cuenta para implementar iniciativas que integren sistemas de información. Para el logro de una integración eficiente muchos elementos han evolucionado desde el punto de vista arquitectónico, técnico y metodológico. Aun así todas las soluciones de integración implementadas deben enfrentar los siguientes retos:

- Las redes son poco fiables: las soluciones de integración tienen que transportar datos de un ordenador a otro a través de las redes. En comparación con un proceso en ejecución en un solo equipo, los sistemas distribuidos tienen que estar preparados para hacer frente a un conjunto mucho más amplio de posibles problemas. Muchas veces, dos sistemas a integrar están separados por continentes y los datos entre ellos tienen que viajar a través de líneas telefónicas, segmentos de LAN, enrutadores, conmutadores, redes públicas y enlaces satelitales. Cada uno de estos pasos puede causar retrasos o interrupciones.
- Las redes son lentas: el envío de datos a través de una red es mucho más lento que hacer una llamada a un método local. El diseño de una solución ampliamente distribuida podría tener consecuencias negativas en el rendimiento.
- Cualquier pareja de aplicaciones es diferente: las soluciones de integración necesitan para transmitir información entre los sistemas que utilizan diferentes lenguajes de programación,

plataformas operativas y formatos de datos. Una solución de integración tiene que ser capaz de interactuar con todas estas diferentes tecnologías.

- El cambio es inevitable: las aplicaciones cambian con el tiempo. Una solución de integración tiene que seguir el ritmo de los cambios en las aplicaciones que conecta. Las soluciones de integración pueden quedar fácilmente atrapadas en un efecto avalancha de cambios, es decir si un sistema cambia, todos los otros sistemas pueden verse afectados. Una solución de integración debe minimizar las dependencias de un sistema a otro mediante el uso de la articulación flexible entre aplicaciones.

Con el tiempo, los desarrolladores han superado estos retos con cuatro enfoques principales:[Hohpe 2003]

1. File Transfer: Una aplicación escribe un archivo que otra posteriormente lee. Las aplicaciones tienen que ponerse de acuerdo sobre el nombre del archivo y la ubicación, el formato del archivo, el momento de cuándo se escribe y se lee, y quién va a eliminar el archivo.
2. Shared Database: Múltiples aplicaciones comparten el mismo esquema de datos, que se encuentra en una sola base de datos física. Como no hay almacenamiento de datos duplicados, no hay datos que deban ser transferidos de una aplicación a la otra.
3. Remote Procedure Invocation: Una aplicación expone algunas de sus funciones de modo que se puede acceder de forma remota por otras aplicaciones como un procedimiento remoto. La comunicación se produce en tiempo real y de forma sincronizada.
4. Messaging: Una aplicación publica un mensaje a un canal de mensajes común. Otras aplicaciones pueden leer el mensaje desde el canal en un momento posterior. Las aplicaciones deben estar de acuerdo en un canal, así como el formato del mensaje. La comunicación es asíncrona.

Mientras que todos los cuatro enfoques esencialmente resolver el mismo problema, cada estilo tiene sus ventajas y desventajas distintas. De hecho, las aplicaciones pueden integrar el uso de múltiples estilos de forma que cada punto de integración se aprovecha del estilo que se adapte lo mejor.

1.5. Estándares para la interoperabilidad

Es importante utilizar los estándares aceptados a nivel global para no afectar la interoperabilidad. Los estándares y normas son descripciones técnicas detalladas, elaboradas con el fin de garantizar la interoperabilidad entre elementos construidos independientemente, así como la capacidad de replicar un mismo elemento de manera sistemática.

Según la Organización Internacional para la Estandarización (ISO), uno de los principales organismos internacionales desarrolladores de estándares, la normalización es la actividad que tiene por objeto establecer, ante problemas reales o potenciales, disposiciones destinadas a usos comunes y repetidos, con el fin de obtener un nivel de ordenamiento óptimo en un contexto dado, que puede ser tecnológico, político o económico.

Un estándar, tal como lo define la ISO "son acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías o definiciones de características para asegurar que los materiales, productos, procesos y servicios cumplan con su propósito". Por lo tanto un estándar de telecomunicaciones "es un conjunto de normas y recomendaciones técnicas que regulan la transmisión en los sistemas de comunicaciones". Queda bien claro que los estándares deberán estar documentados, es decir escritos en papel, con objeto que sean difundidos y captados de igual manera por las entidades o personas que los vayan a utilizar. [ISO 2015]

La interoperabilidad se basa en que las comunicaciones obedezcan a normas claras y unívocas. Estos documentos técnicos definen las exigencias o requisitos operativos, que a su vez pueden ir acompañadas de recomendaciones operativas. Si la norma está bien escrita, dos sistemas que las satisfagan han de poder dialogar sin problemas. También podrán evolucionar libremente sin riesgo de alterar la comunicación, siempre que respeten la norma que define sus interfaces.

La norma puede definir elementos como:

- los formatos de datos intercambiados en un determinado contexto, que describen las secuencias de información o de comandos que debe enviar el sistema,
- el protocolo de comunicación, que es el modo en que deben responder sus corresponsales
- las tensiones y corrientes a emplear
- los tipos de cables y conectores.

En el ámbito de los sistemas informáticos una decisión importante con los archivos es el formato a utilizar. Muy rara vez la salida de una aplicación es exactamente lo que se necesita para otra solución, por lo que se tendrá que hacer un poco de transformación de los archivos en el camino. Como resultado, los formatos de archivo estándar han crecido con el tiempo. Los sistemas mainframe suelen utilizar fuentes de datos a partir de los formatos del sistema de archivos de COBOL, mientras que los sistemas Unix utilizan archivos basados en texto. La moda actual es utilizar XML para el intercambio de datos en forma de mensajes. Una industria de lectores, escritores, y herramientas de transformación se ha construido en torno a cada uno de estos formatos. Algunos ejemplos de estándares abiertos son: XML, XHTML, PNG, Vorbis, FLAC, entre

otros. Algunos de ellos han logrado el estatuto de norma internacional, sin perder la cualidad de ser una norma abierta.

1.6. Análisis de las herramientas existentes para el monitoreo y análisis de procesos industriales

Para llevar a cabo la gestión eficaz de un proceso industrial se necesitan herramientas que permitan guiar el desarrollo exitoso del mismo. Teniendo en cuenta que cada vez son más las soluciones de software desarrolladas y aplicadas a las diferentes áreas sociales; debe intentarse la adquisición de una solución ya existente que cumpla los requerimientos para resolver el problema planteado. Con este objetivo se analizan soluciones que realicen la automatización de las actividades de monitoreo y análisis de los procesos industriales. Un problema común identificado en las herramientas estudiadas es que no se encontraron indicios de aplicación de las variables asociadas a los procesos industriales en la rama de la bioinformática, menos aún de los indicadores relevantes para el proceso de fermentación en el CIM.

La totalidad de sistemas estudiados junto con todas sus características se muestran en el Anexo 1 de esta investigación. A continuación la Tabla 1 muestra un estudio comparativo de tres de las herramientas analizadas:

Tabla 1 - Comparación y análisis de las arquitecturas de los sistemas similares a SIMAFI CIM en el mercado

Criterios analizados	SPMAP	DABin	SIMAFI CIM
Soporte sobre la web	✓	✓	✓
Tipo de licencia	Propietaria	GPL	Open Source
Comunicación con otros sistemas	✓	✓	✓
Visualización de variables online	✓	✓	✓
Visualización de variables offline	---	---	✓
Indicadores compuestos	---	---	✓
Reportes	✓	✓	✓
Histórico	✓	---	✓
Persistencia de datos	✓	---	✓
Orientada a servicios	---	---	✓
Orientada a eventos	✓	✓	✓
Arquitectura escalable	✓	---	✓
Gestión de usuarios	---	---	✓
Control de acceso	---	---	✓

1.7. Conclusiones parciales

Como parte del marco teórico de la investigación se estudiaron los aspectos relevantes de los sistemas para el monitoreo procesos industriales, principalmente relacionados con la comunicación a todos los niveles y los componentes arquitectónicos que se manejan en estos sistemas. Con ello se determinó que componentes y funcionalidades se aplicarían en esta investigación.

El estudio de los sistemas de análisis de procesos que más se ajustan al tema de la investigación reafirmó las ventajas que estos ofrecen para la toma de decisiones en escenarios de automatización industrial y control de procesos claves como la fermentación en el Centro de Inmunología Molecular.

La revisión de las características y funciones principales de los sistemas de análisis de datos permitió tener una visión general de las múltiples posibilidades para el procesamiento del alto volumen de información generada en el CIM.

Las arquitecturas estudiadas resultan muy útiles para la automatización de las actividades de monitoreo y análisis de los procesos industriales; pero muchas no son aplicables de las variables asociadas a la bioinformática. Además presentan limitaciones en cuanto a la interoperabilidad entre los diferentes sistemas, por lo que se hace necesario desarrollar una arquitectura propia que permita la integración de los componentes tecnológicos existentes en el CIM y de otros que puedan ser incorporados. También es preciso desarrollar un sistema informático que utilice los componentes de dicha arquitectura para automatizar las actividades del proceso de fermentación que aún se realizan manualmente.

CAPITULO 2. ARQUITECTURA DE INTEGRACIÓN PARA EL MONITOREO Y ANÁLISIS

En este capítulo se presenta la solución al problema planteado, una arquitectura de integración para el monitoreo y análisis del proceso de fermentación en el CIM. Se explica el diseño de esta solución con diagramas que van de lo general a lo particular, de esta forma se expone cada uno de sus componentes arquitectónicos y se realiza además una descripción detallada de los mismos, así como las relaciones entre ellos.

2.1. Arquitectura de integración propuesta

Hay muchas opiniones en cuanto a lo que la arquitectura de software es. Una de las más aceptadas es la descripción de la IEEE de la arquitectura de software como los “conceptos o propiedades de un sistema en su entorno enfocado en sus elementos, relaciones fundamentales y en los principios de su diseño y evolución”. [IEEE 2015]

Definiciones como la que aparece en el libro “The Rational Unified Process” hacen hincapié en la arquitectura como un conjunto de decisiones importantes sobre la organización de un sistema de software, la selección de los elementos estructurales y las interfaces por el cual el sistema se compone, junto con su comportamiento que se especifica en las colaboraciones entre esos elementos, la composición estructural y los elementos de comportamiento en subsistemas progresivamente más grandes, y el estilo arquitectónico que guía esta organización de elementos y sus interfaces, sus colaboraciones, y su composición. [Kruchten 2003]

Otra definición un poco más descriptiva aparece en el libro “SOA Patterns”, enuncia que arquitectura de software es el conjunto de decisiones fundamentales acerca de un producto de software o solución diseñada para satisfacer los atributos de calidad del proyecto (los requisitos arquitectónicos). Incluye los componentes principales, sus principales atributos y sus colaboraciones (interacciones y comportamientos) para cumplir con los atributos de calidad. Puede y por lo general debe, ser expresado en varios niveles de abstracción, donde el número de niveles depende del tamaño y la complejidad del proyecto. [Rotem-Gal-Oz 2012]

En el marco de esta investigación la autora define como arquitectura de integración al conjunto de tecnologías presentes en el CIM, incluyendo como componentes los trabajadores de la entidad, los sistemas informáticos existentes y otros que puedan ser incorporados a la infraestructura tecnológica; y se tienen en cuenta también las relaciones entre ellos. La siguiente figura muestra la arquitectura de integración para el monitoreo y análisis del proceso de Fermentación en el CIM:

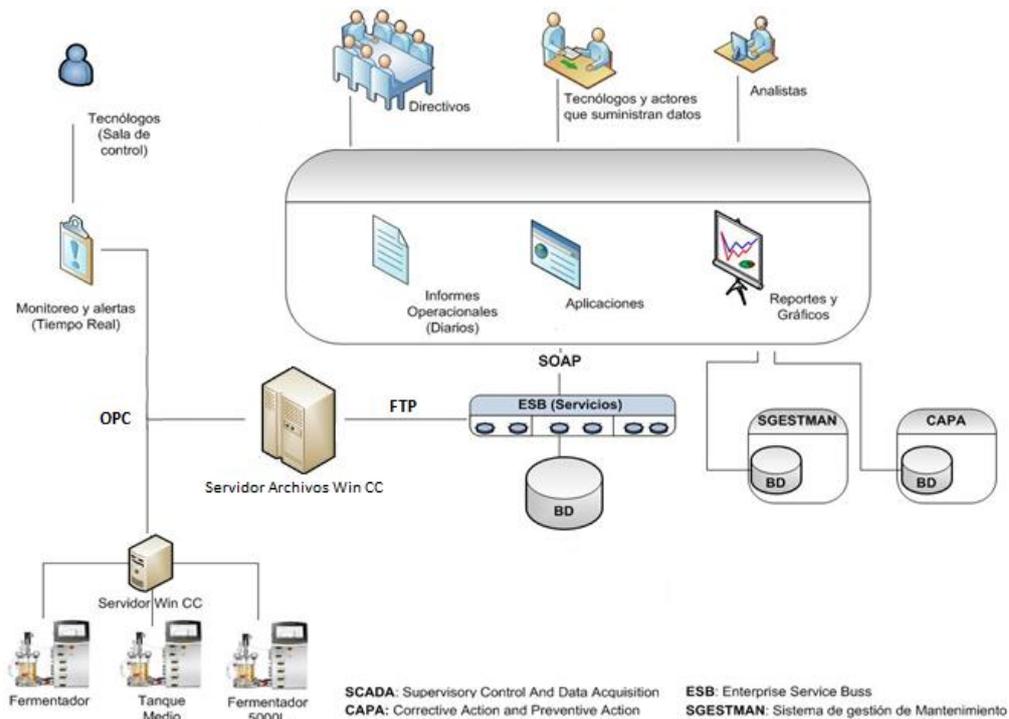


Figura 13 - Arquitectura de integración para el análisis y monitoreo (Fuente: elaboración propia)

Una descripción más detallada de los componentes actuales de la arquitectura propuesta sería:

1. Roles generales

Actores que suministran datos: son fundamentalmente aquellos trabajadores que analizan muestras, realizan diferentes pruebas (HPLC, DO, ELISA) y a partir de los resultados obtenidos reflejan los datos en los registros que se generan dentro del proceso de fermentación.

Analistas: forman parte del personal calificado encargado de construir conocimiento (detectar patrones de comportamiento partiendo del estudio de la data histórica, los cuales han de quedar registrados en reportes y gráficos) a partir del análisis de la información recopilada.

Tecnólogos de la sala de control y los **Directivos:** deben poder monitorear permanentemente el comportamiento del proceso de fermentación, así como recibir las alarmas en tiempo real y los informes operacionales diarios.

2. Recursos tecnológicos

SCADA WinCC: es el acrónimo de “Supervisory Control And Data Acquisition”, que puede traducirse como “Control, Supervisión y Adquisición de Datos”. Los sistemas SCADA automatizan el monitoreo y control de procesos industriales, son parte integral de la mayoría de los ambientes complejos o geográficamente dispersos, ya que pueden recoger información de una gran cantidad de fuentes muy rápidamente y presentarla en una forma entendible permitiendo tomar decisiones operacionales apropiadas. Sus funciones principales dentro del escenario del CIM son:

- Supervisión remota de instalaciones y equipos
- Control de instalaciones y equipos

- Procesamiento de datos
- Visualización gráfica dinámica
- Generación de reportes
- Representación de señales de alarmas
- Almacenamiento de información histórica
- Programación de eventos

Permite el aprovechamiento de información obtenida en tiempo real, a fin de proveer realimentación sobre el desarrollo del proceso que estará bajo control. Para el cliente CIM aporta facilidades como: los tecnólogos de la sala de control podrán recibir mensajes o alarmas cuando alguna de las variables críticas del proceso de fermentación esté fuera de rango, los operadores que trabajan directamente con los tanques en las plantas de fermentación conseguirán modificar los parámetros de producción y los directivos obtendrán informes operacionales diarios creados a partir de la última data disponible. El monitoreo constante será posible para ambos niveles.

ESB: Un Enterprise Service Bus o Bus de Servicios Empresariales actúa como punto central donde se registran los servicios expuestos por las aplicaciones de un entorno empresarial y sobre el cual se puede construir aplicaciones que re-aprovechen todas estas funcionalidades ya implementadas. Sus principales beneficios son:

- Permite la construcción de aplicaciones y procesos de negocio débilmente acoplados, en base a unos servicios ya existentes.
- Conecta de manera uniforme diferentes plataformas como por ejemplo: SAP, Siebel, Alfresco, Documentum y FileNet.
- A partir de un mensaje de entrada se puede inspeccionar su contenido y llamar a un servicio u otro dependiendo de los valores que lleve dentro.
- Facilita controlar las excepciones de todos los servicios de manera uniforme, independientemente del protocolo o plataforma, permitiendo transformarlas a un formato común para todas las aplicaciones de la organización.
- Admite resolver dinámicamente los puntos de publicación de los servicios, siendo fácil cambiar, por ejemplo, de un entorno de pre-producción a producción de manera transparente para los clientes.
- Permite auditar los servicios y generar sus trazas, para detectar el tipo de información más consultada, o el servicio más utilizado, sus tiempos de respuesta, entre otros indicadores.
- Posibilita proteger o desproteger servicios, a nivel de transporte o del propio servicio.

Las aplicaciones empresariales consumirán servicios publicados en un ESB los cuales a su vez tendrán como fuente principal una base de datos. Dentro de este ambiente se utilizará para

conectar, mediar, y controlar la interacción entre diversas aplicaciones y servicios a lo largo de un entorno altamente distribuido y heterogéneo.

Debe proporcionar a dicha organización por un lado un sistema de mensajería robusto de elevada disponibilidad y altamente escalable que permita garantizar la comunicación entre los distintos servicios y aplicaciones de la organización independientemente de su localización geográfica; y por otro lado un mecanismo que permita fácilmente la definición de nuevos procesos o su posterior modificación orquestando los servicios existentes en la organización.

SGESTMAN: Sistema utilizado para la gestión del mantenimiento.

CAPA: Aplicación empleada para el registro de no conformidades detectadas durante el proceso de fermentación

3. Relaciones entre los componentes de la arquitectura

SOAP: Simple Object Access Protocol o Protocolo Simple de Acceso a Objetos, está basado en XML (Extensible Markup Language o Lenguaje de Etiquetado Extensible) y se utiliza para invocar procedimientos en forma remota. Utiliza cualquier protocolo que permita transportar mensajes de texto, principalmente HTTP. Consta de tres partes: una envoltura (envelope), la cual define qué hay en el mensaje y cómo procesarlo; un conjunto de reglas de codificación para expresar instancias de tipos de datos; y una convención para representar llamadas a procedimientos y respuestas. El protocolo SOAP tiene tres características principales:

- Extensibilidad (permite la seguridad y enrutamiento de servicios web).
- Neutralidad (SOAP puede ser utilizado sobre cualquier protocolo de transporte como SMTP, TCP o JMS, como así también sobre HTTP)
- Independencia (utiliza cualquier modelo de programación).

En el marco de trabajo del CIM se utilizará para el envío de mensajes entre los servicios publicados en el ESB y las aplicaciones dentro del Portal de Gestión Integrada de Información de Operaciones Biotecnológicas.

FTP (*File Transfer Protocol* o Protocolo de Transferencia de Archivos): se utiliza para la transferencia de archivos entre sistemas que están conectados a una red TCP (por sus siglas del inglés *Transmission Control Protocol* o Protocolo de Control de Transmisión), se basa en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo. [Indiana 2014] En el ámbito de esta investigación se utiliza para obtener el fichero de valores separados por coma CSV (por sus siglas del inglés *Comma-Separated Values*), publicado por el SCADA WinCC de los fermentadores del CIM.

2.2. Patrón de integración

La construcción de sistemas distribuidos e integrados sigue siendo un problema difícil de resolver. En primer lugar, se requiere una comprensión sólida de los componentes individuales que se conectarán. A continuación, se requiere conectar estos componentes de una forma que equilibre el acoplamiento débil frente a los requisitos de todo el sistema, como la latencia y la seguridad. Por último, pero no menos importante, el sistema resultante tiene que ser controlado y gestionado. Con el tiempo, una serie de enfoques se han propuesto para resolver estos retos: componentes distribuidos, mensajería EAI, y más recientemente las arquitecturas orientadas a servicios.

De los patrones de integración previamente mencionados, se determina utilizar SOA por elementos decisivos como son las características de los servicios como unidades con interfaces bien definidas que permiten la interoperabilidad entre los diferentes sistemas independientemente de la tecnología o los marcos de trabajo utilizados en su desarrollo.

Arquitectura orientada a servicios es definida para los sistemas basados en las interacciones de los componentes débilmente acoplados y autónomos llamados servicios. Cada servicio expone procesos y comportamientos a través de contratos, que se componen de mensajes a direcciones descubribles llamados Endpoint o puntos finales. El comportamiento de un servicio se rige por las políticas que son externas al propio servicio. Los contratos y los mensajes son utilizados por los componentes externos llamados consumidores de servicios. [Rotem-Gal-Oz 2012]

La siguiente figura ilustra de SOA sus componentes y sus relaciones:

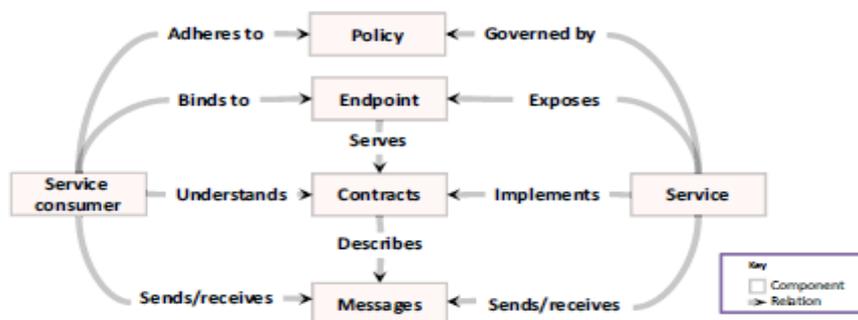


Figura 14 - SOA componentes y relaciones (Fuente: [Rotem-Gal-Oz 2012])

Para facilitar la comprensión del patrón de integración utilizado en la arquitectura propuesta en esta investigación se exponen algunos elementos de los componentes SOA:

Servicio

Es el componente central de SOA. En mi opinión, un servicio debe proporcionar una función de negocio y debe ser una parte importante de la lógica. Además, un servicio debe aplicar todas las

funcionalidades incluidas en los contratos que expone. Una de las características deseables de los servicios es su autonomía, lo que significa que el servicio debe ser principalmente autosuficiente.

Contrato

La colección de todos los mensajes soportados por el servicio es conocido como contrato. Puede ser unilateral, lo que significa que ofrece un conjunto cerrado de mensajes que siguen un flujo en una dirección. Alternativamente, un contrato puede ser bilateral, con el servicio el intercambio de mensajes con un grupo predefinido de componentes. El contrato de servicio es análogo a la interfaz de un objeto en el diseño orientado a objetos.

Endpoint

Es un identificador universal del recurso URI (por sus siglas del inglés *Universal Resource Identifier*), expresa la dirección o lugar específico donde se puede encontrar el servicio. Un contrato específico puede ser expuesto en un endpoint específico.

Mensaje

La unidad de la comunicación en SOA es el mensaje. Los mensajes pueden venir en diferentes formas, tales como:

- HTTP GET: en la transferencia de estado representacional (REST).
- SOAP: protocolo simple de acceso a objetos
- JMS: mensajes de servicios Java
- SMTP: protocolo para transferencia simple de correo

La diferencia entre un mensaje y otras formas de comunicación, tales como una llamada a procedimiento remoto RPC (por sus siglas del inglés *Remote Procedure Call*), no es sutil. Un RPC requiere a menudo la aplicación que realiza la llamada a tenga conocimiento de los detalles de implementación del otro sistema. Con la mensajería, este no es el caso. Los mensajes tienen una cabecera y un cuerpo (la carga útil). La cabecera es generalmente genérica y puede ser entendido por los componentes de infraestructura y marco sin conocer los detalles de implementación. Esto reduce dependencias y acoplamiento.

Política

Un diferenciador importante entre SOA y diseño orientado a objetos es la existencia de políticas. Así como una interfaz o contrato separa especificaciones de las implementaciones, las políticas separan especificaciones dinámicas de las especificaciones estáticas o semánticas. Una política define los términos y condiciones para hacer un servicio disponible para el consumidor de servicios. Los aspectos únicos de las políticas son que pueden ser actualizados en tiempo de ejecución y que están externalizados de la lógica de negocio. Una política específica propiedades dinámicas, tales como la seguridad (encriptación, autenticación, autorización), auditoría, acuerdos de nivel de servicios SLA (por sus siglas del inglés *Service Level Agreement*), y así sucesivamente.

Consumidor de servicios

Un servicio sólo tiene sentido si otro componente de software lo utiliza. Los consumidores de servicios son los componentes de software que interactúan con un servicio a través de mensajería. Los consumidores pueden ser aplicaciones cliente u otros servicios; el único requisito es que se adhieren a la SOA en la que intervienen.

2.2.1. Patrones de diseño SOA

Si bien estos enfoques y herramientas han sido de gran ayuda, aún no existe una vía fácil para equilibrar los requisitos potencialmente opuestos en una solución coherente. Es por esto que los patrones de diseño son un recurso tan fundamental para la construcción de soluciones SOA con éxito. Los patrones codifican el conocimiento y la experiencia de una manera que se puede aplicar en una variedad de contextos y tecnologías. De los patrones de diseño SOA se aplican a esta investigación: [Rotem-Gal-Oz 2012]

Service Bus

Implementar el patrón Service Bus o Bus de Servicios, implica utilizar una infraestructura de mensajería unificada para la transformación de mensajes, la mediación, el enrutamiento y la invocación. La esencia está en abstraer las comunicaciones entre servicios. Para lograr esto, el patrón Service Bus combina varios patrones de diseño para la integración empresarial, incluyendo:

- Message bus: conecta los diferentes servicios
- Message router: determina qué mensaje enviar y a dónde
- Channel adaptor: convierte formatos y protocolos

El patrón Bus de Servicios se compone de tres funciones principales:

- Service registration: El bus tiene que saber dónde encontrar los servicios de modo que pueda invocarlos. También debe proporcionar un mecanismo para permitir que los servicios puedan configurar y exponer los endpoint adicionales de otros servicios que se pueden consumir.
- Message handling: El bus de servicio proporciona capacidades para invocar los servicios registrados utilizando el criterio de valoración que han definido. También envía mensajes de rutas al registro de servicio. Además transforma protocolos o mensajes para asegurarse de que el servicio dirigido puede manejar los mensajes.
- Publish/subscribe: El bus incorpora servicios de suscripción (que pueden ser considerados como un tipo de registro). Entonces, cuando los servicios deseen publicar mensajes, el bus de servicio puede utilizar el enrutamiento, transformaciones, y la invocación para llamar a los servicios suscritos.

Un diagrama de la arquitectura de una solución que utiliza el patrón Service Bus suele mostrar varios servicios con una entidad central que conecta a todos. Esto se aprecia en la siguiente figura:

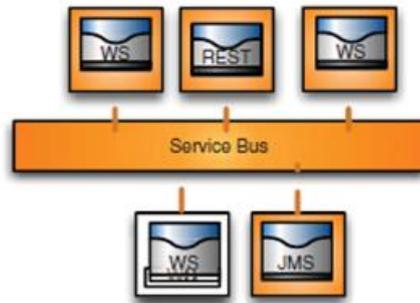


Figura 15 - Patrón SOA Service Bus (Fuente: [Rotem-Gal-Oz 2012])

Orchestration

Se implementa el patrón Orquestación para externalizar los procesos de negocio como servicios y permitir que estos se rijan, sean controlados, y se adapten a los cambios de forma dinámica. El patrón de Orquestación, tal y como se ilustra en la Figura 17, es relativamente simple, en esencia se trata de la adición de un motor de flujo de trabajo que es externo a los servicios. Se modelan los diferentes procesos de negocio como los flujos de interacciones de servicios y el motor de ejecución se encargará de monitorearlos y administrarlos para llevar a cabo el proceso.

El componente principal es el motor de flujo de trabajo, este gestiona los flujos de trabajo, también alberga las instancias de flujo de trabajo y supervisa su progreso. Cada proceso se instancia como una instancia de flujo de trabajo que puede programar y administrar el proceso en sí. La instancia de flujo de trabajo es capaz de bifurcación (el envío de solicitudes de paralelo), unión (a la espera de respuestas o reacciones de los múltiples servicios) y manipulación de fracasos. Un flujo de trabajo puede ser un proceso de corta duración, pero en la mayoría de los casos será un proceso en ejecución por más tiempo.

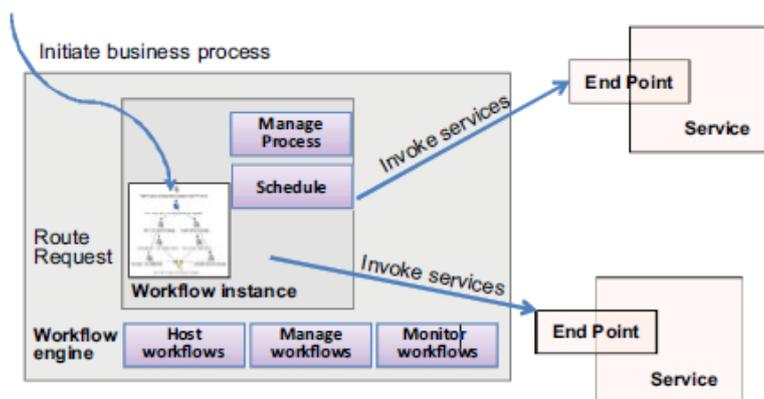


Figura 16 - Patrón SOA Orchestration (Fuente: [Rotem-Gal-Oz 2012])

En cierto sentido, el patrón de orquestación es una implementación especial del patrón de Saga, donde el coordinador es externo a todos los participantes. El patrón Saga es más genérico, ya que también se puede implementar sin un componente central que sabe qué pasos es necesario hacer y cuándo debe completar el proceso de negocio. Esto tiene la ventaja de mantener los servicios

más autónomos, con procesos flexibles, pero el costo es la falta de claridad en cuanto a lo que constituye un proceso de negocio y dando como resultado dificultades en el seguimiento y la comprensión de la situación actual de los negocios procesos.

Workflowdize

Introducir un motor de workflow dentro del servicio para manejar el entorno volátil y cambiante procesos y orquestar la lógica estable. El patrón Workflowdize, tal como se representa en la Figura 18, se basa en la adición de un flujo de trabajo motor al servicio para impulsar los procesos de negocio. Los flujos de trabajo pueden llegar a ser bastante complejos, manejando procesos de larga ejecución con varios puntos de entrada, cuando las solicitudes y las respuestas llegan de servicios externos.

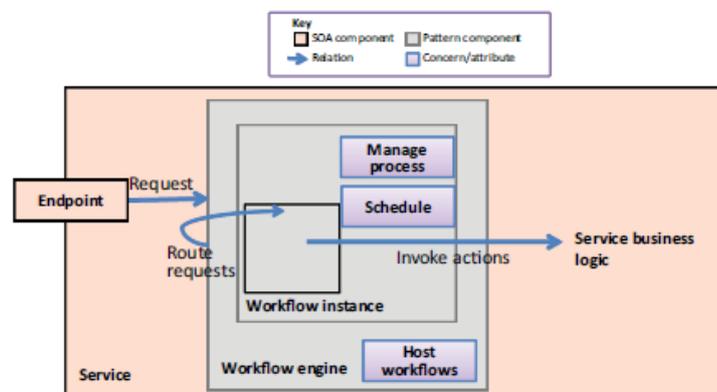


Figura 17 - Patrón SOA Workflowdize (Fuente: [Rotem-Gal-Oz 2012])

La ventaja de utilizar los flujos de trabajo es que permitirá ordenar y reorganizar las actividades en los procesos de una manera muy flexible. Se modela el proceso de negocio como un flujo de actividades que se producen a medida que lleguen los mensajes. Debido a que cada actividad se puede probar de forma individual, la reutilización de ellos requiere menos pruebas en general. Al reordenar las actividades, puede responder rápidamente a las cambiantes necesidades del negocio con menos riesgo.

2.3. Estilo de integración

Si las necesidades de integración fueran siempre las mismas, existiría un único estilo de integración. Sin embargo, al igual que cualquier esfuerzo tecnológico complejo, la integración de aplicaciones consiste en una serie de consideraciones y consecuencias que se deben tener en cuenta para cualquier oportunidad de integración.

El primer criterio es la integración de aplicaciones en sí. Se puede desarrollar una aplicación única, independiente que no necesita colaboración con ninguna otra aplicación, así se evitaría todo el problema de integración. La realidad, sin embargo, es que incluso una empresa sencilla tiene múltiples aplicaciones, aplicaciones que necesitan trabajar juntas para proporcionar una

experiencia unificada para empleados de la empresa, socios y clientes. Los otros criterios de decisión son: [Fisher 2012]

- Acoplamiento de aplicaciones: incluso las aplicaciones integradas debe minimizar sus dependencias entre sí de manera que cada uno puede evolucionar sin causar problemas a los demás. Los sistemas rígidos hacen numerosas suposiciones acerca de cómo funcionan las otras aplicaciones; cuando dichas aplicaciones cambian esos supuestos, se afecta la integración. La interfaz para la integración de aplicaciones debe ser lo suficientemente específico para implementar la funcionalidad útil, pero en general lo suficiente flexible para permitir que la aplicación para cambiar según sea necesario.
- Simplicidad de integración: cuando se integra una aplicación en una empresa, los desarrolladores deben esforzarse por reducir al mínimo el cambio de la aplicación y minimizar la cantidad de código de integración necesario.
- Tecnología de integración: diferentes técnicas de integración requieren amplias cantidades variables de software especializado y hardware. Estas herramientas especiales son costosas, esto puede conducir a la dependencia de un proveedor y consecuentemente a aumentar la carga sobre los desarrolladores para entender cómo usar las herramientas para integrar aplicaciones.
- Formato de datos: las aplicaciones integradas deben ponerse de acuerdo sobre el formato de los datos que intercambian o deben tener un traductor intermedio para unificar aplicaciones que insisten en diferentes formatos de datos.
- Puntualidad de datos: la integración debe reducir al mínimo la cantidad de tiempo entre el momento en que una sola aplicación decide compartir algunos datos y otras aplicaciones tienen esos datos. Los datos deben ser intercambiados con frecuencia en pequeños trozos, en lugar de esperar para el intercambio de un gran conjunto de elementos no relacionados. La latencia en el intercambio de datos ha de tenerse en cuenta en el diseño de la integración.
- Datos o funcionalidades: las aplicaciones integradas pueden no querer simplemente compartir datos, puede que deseen compartir la funcionalidad de tal manera que cada aplicación puede invocar la funcionalidad en los otros. La invocación remota de funcionalidades puede ser difícil de conseguir, no es lo mismo que invocar la funcionalidad local y tiene consecuencias significativas en el funcionamiento de integración.
- Asíncronía: Es posible que el subprocedimiento está disponible cuando el procedimiento superior quiere invocarlo. Sin embargo, un procedimiento puede no querer esperar a que el subprocedimiento se ejecute; puede que desee invocar el subprocedimiento de forma asíncrona, comenzando el subprocedimiento pero luego dejar que se ejecute en segundo plano. Esto es especialmente cierto en las aplicaciones integradas, cuando la aplicación

remota no se esté ejecutando o la red no esté disponible, la aplicación de origen puede desear simplemente hacer que los datos compartidos estén disponibles o ingrese una solicitud de una llamada subprocedimiento, pero luego ir a otro trabajo confiando en que la aplicación remota actuará en algún momento posterior.

Como se puede ver, hay varios criterios diferentes que deben ser considerados al momento de elegir y diseñar un enfoque de integración. Hay más de un enfoque para la integración de aplicaciones, cada uno aborda algunos de los criterios de integración mejores que otros. De los cuatro principales estilos de integración en esta investigación se utiliza:

File Transfer: las aplicaciones producen archivos de datos compartidos para que otros puedan consumirlos y a su vez consumen archivos que otros han producido.

Para una empresa como el CIM que tiene múltiples aplicaciones que se están construyendo de forma independiente, con diferentes lenguajes y plataformas, la integración para que dichos sistemas funcionen juntos y puedan intercambiar información se alcanza aplicando este enfoque.

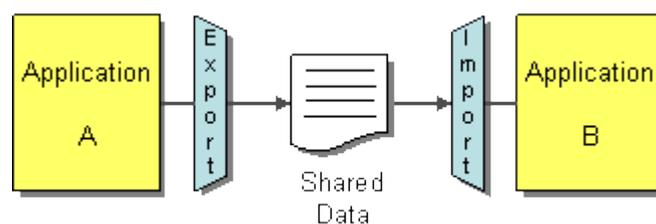


Figura 18 – Estilo de integración File Transfer (Fuente: [Hohpe 2003])

El SCADA WinCC es el encargado de producir el archivo que contienen información que otras aplicaciones necesitan consumir. El ESB como agente integradores es el encargado de asumir la responsabilidad de transformar los archivos en diferentes formatos y de exponerlos como servicios de datos accesibles a otros sistemas. En este entorno se producen los archivos a intervalos regulares de acuerdo con la naturaleza del proceso de fermentación.

2.4. Patrón arquitectónico

En el desarrollo de aplicaciones para el monitoreo y análisis de procesos industriales se propone el uso del patrón arquitectónico Modelo-Vista-Presentador, altamente recomendado para los entornos de ejecución de dichas aplicaciones que deben separar la interfaz de usuario (UI por sus siglas del inglés *User Interface*) de la lógica del negocio.

La característica principal de este patrón es que la clase Presentador (contiene la lógica de la interfaz de usuario) hace de intermediario entre la Vista (la interfaz gráfica de usuario) y el Modelo de datos (implementa la lógica de negocio). La Vista tiene métodos que reciben los datos que

debe pintar ya "procesados", entonces solamente debe mostrar esos datos en los componentes gráficos. Por su parte el Presentador hará de enlace entre el Modelo y la Vista, sus funciones son recibir las interfaces que debe implementar el Modelo y enviar los datos procesados a la Vista.[MSDN 2007]

Algunas de las ventajas de utilizar este patrón son:

- Mejoras en las posibilidades de mantenimiento, debido a que cada estructura es independiente de la otra, los cambios o actualizaciones pueden ser realizados sin afectar la aplicación como un todo.
- Escalabilidad, ya que las estructuras están basadas en diferentes componentes físicos y lógicos, el escalamiento de la aplicación de forma horizontal es razonablemente sencillo.
- Flexibilidad, como cada estructura puede ser manejada y escalada de forma independiente, la flexibilidad se incrementa.
- Disponibilidad, las aplicaciones pueden aprovechar la arquitectura modular de los sistemas implicados usando componentes que persistan los datos fácilmente lo que incrementa la disponibilidad.

Las estructuras que se proponen son:

- 1- Vista: es la única que interactúa directamente con el usuario presentándole los componentes de interfaz del sistema, permitiendo el intercambio de información entre ambos. Contiene varias interfaces gráficas que posibilitan capturar los datos insertados por los usuarios, además de mostrarles los resultados de sus peticiones y los estados de la aplicación. Esta estructura puede implementarse para interactuar con el Presentador y el Modelo, o sólo con el primero de estos, según la aplicación lo requiera.
- 2- Presentador: actúa como mediador entre la Vista y el Modelo. Puede ser utilizado en diferentes variantes: "Controlador Supervisado" o "Vista Pasiva". En su primera variante el Presentador no gestiona la forma en que la información es mostrada en la Vista, es ella quien define cómo la información es formateada y mostrada en la pantalla a partir de los componentes que contiene. Mientras que en la segunda variante la Vista no implementa ningún tipo de lógica, su única función es mostrar la información que recibe del Presentador, él es quien maneja los eventos de los componentes de interfaz y a su vez gestiona el intercambio de datos con el Modelo. [Fowler 2006]
- 3- Modelo: ignora cómo la información es mostrada al usuario. Contiene toda la lógica de las aplicaciones, en ella es donde se efectúa la mayor parte del procesamiento de la información del contexto de la aplicación, se realizan cálculos sobre los datos almacenados y se ejecutan los algoritmos específicos del programa. Además es la única

que interactúa directamente con las entidades del dominio, es la encargada de intercambiar con otros sistemas, solo mediante ella se puede acceder a los recursos obtenidos de terceras aplicaciones. Contiene el gestor de base de datos para lenguaje SQL. Esta estructura interactúa con el Presentador, recibe sus solicitudes de almacenamiento o recuperación de información, y envía la respuesta a las peticiones.

Con el objetivo de exponer claramente el funcionamiento de la propuesta arquitectónica de integración para el monitoreo y análisis del proceso de fermentación en el CIM, se presentan los siguientes diagramas que muestran la estructura de la arquitectura, los elementos que la componen y la forma que interactúan entre ellos:

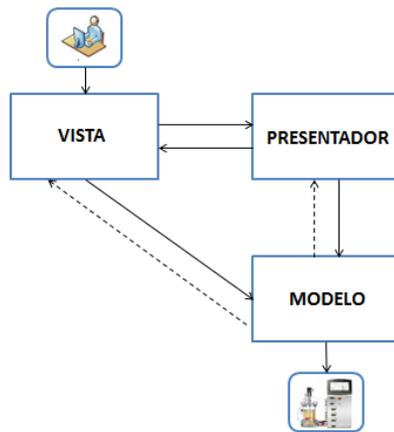


Figura 19 - Diagrama general de la arquitectura de integración (Fuente: elaboración propia)

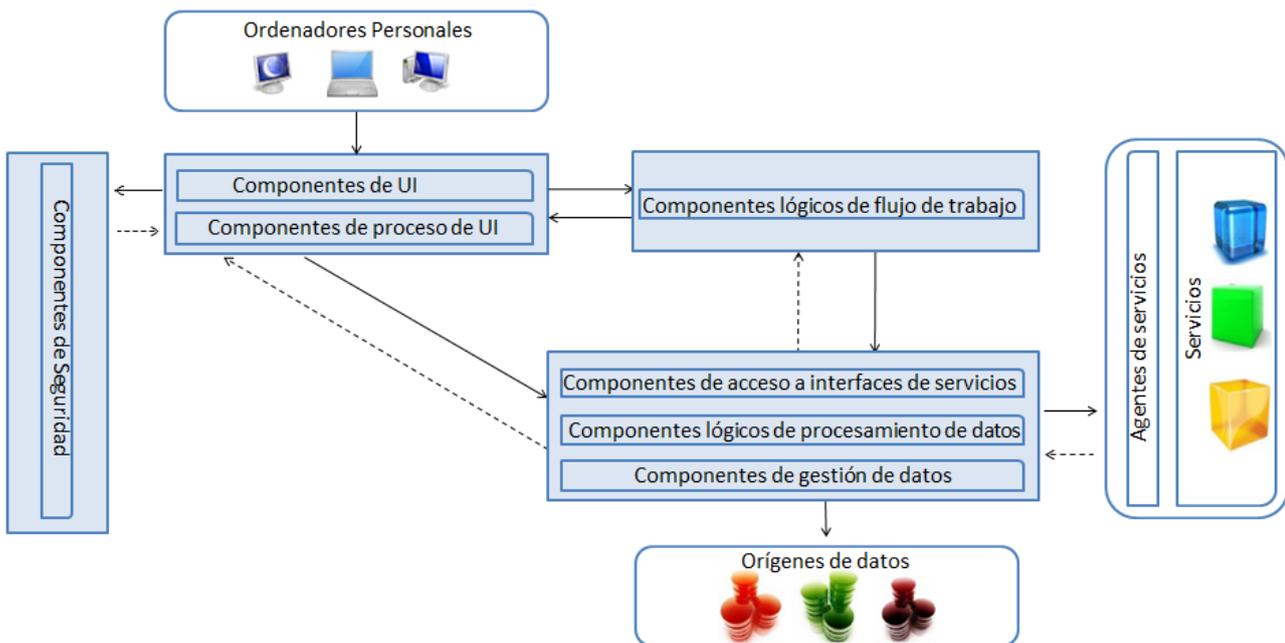


Figura 20 - Diagrama orientado a componentes de la arquitectura de integración (Fuente: elaboración propia)

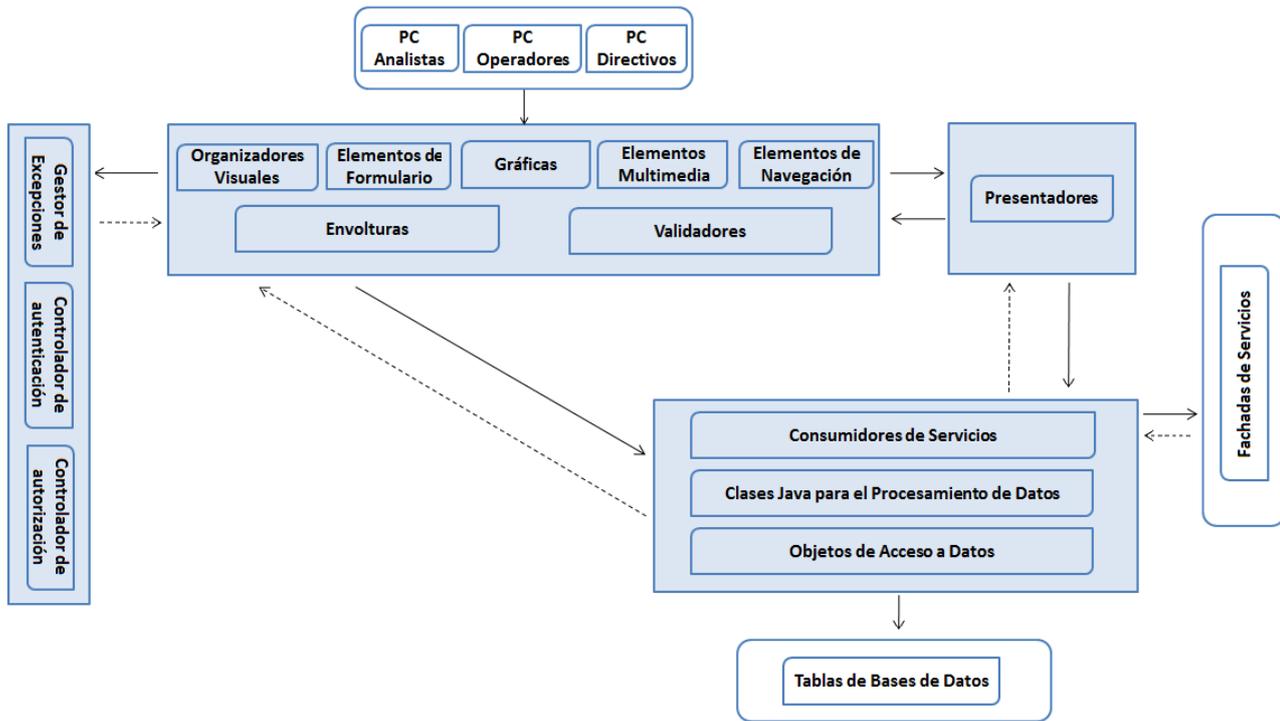


Figura 21 - Diagrama orientado a dominio de la arquitectura de integración (Fuente: elaboración propia)

A continuación se describen los componentes de cada estructura:

1) Vista

a) Componentes de interfaz de usuario

- i) Organizadores visuales: son contenedores de componentes de presentación, se encargan de distribuir los elementos de interfaz de usuario dentro del diseño de las diferentes vistas del sistema.
- ii) Elementos de formularios: tienen diseños nativos del framework Vaadin, concebidos para mejorar la usabilidad de la aplicación. Aquí se incluyen etiquetas, botones, paneles, selectores múltiples, calendarios, menú contextual, contenedores de texto plano y de contraseñas.
- iii) Gráficas: utilizan los diseños de la librería JFreeChart, que presenta una amplia gama de tipos de gráficos. Entre ellos se utilizan para esta solución los de barras, de área, de línea y de serie; muestran el progreso de las variables que rigen el proceso de fermentación.
- iv) Elementos multimedia: permiten reproducir audio y visualizar vídeos e imágenes en el navegador cliente.
- v) Elementos de navegación: posibilitan embeber sitios web externos, para incorporarlos y tratarlos como parte de la aplicación desarrollada.

b) Componentes de proceso de interfaz de usuario

- i) **Envolturas:** son la base sobre la cual se muestran las gráficas. En los navegadores los gráficos se muestran como gráficos vectoriales redimensionables SVG (por sus siglas del inglés *Scalable Vector Graphics*), por lo cual el usuario puede ver los gráficos tanto estáticos como dinámicos sin pérdidas en la calidad de la imagen. [Vaadin 2010] Dentro de esta propuesta su función es a partir de la selección del usuario gestionar la gráfica que se debe mostrar.
- ii) **Validadores:** se encargan de procesar los datos insertados por los usuarios, verifican que estén completos y en el formato adecuado.

2) Presentador

a) Componentes lógicos de flujo de trabajo

- i) **Presentadores:** sirven de enlace para el intercambio de solicitudes y respuestas de las clases de acceso a datos y de los componentes de las vistas.

3) Modelo

a) Componentes de acceso a interfaces de servicios

- i) **Consumidores de servicios:** encapsulan los datos de la petición para consumir determinado servicio (nombre, dirección de publicación, métodos), y obtiene la respuesta de la fachada para luego ser procesada por la aplicación.

b) Componentes lógicos de procesamiento de datos

- i) **Clases Java para el procesamiento de datos:** se encargan de ejecutar las tareas propias de la aplicación, en ellas se realizan cálculos como el promedio de los datos de las variables de la fermentación.

c) Componentes de gestión de datos

- i) **Objetos de acceso a datos:** implementan una interfaz abstracta para el intercambio con la base de datos.

Otros elementos serían:

1) Seguridad

- a) **Gestor de excepciones:** maneja los problemas ocurridos en tiempo de ejecución, no permite al usuario interactuar directamente con los errores del sistema.
- b) **Controlador de autenticación:** evita el acceso a la aplicación por parte de personal no autorizado, se aplica mediante la inserción de la combinación usuario y contraseña para entrar al sistema.

- c) Controlador de autorización: restringe el acceso a determinados escenarios e información, según el rol del usuario que se ha autenticado en el sistema.

2) Servicios

- a) Fachadas de servicios: es lo que publican los servicios para que las aplicaciones puedan consumir datos de ellos.

2.5.Herramientas libres

Los desarrolladores de sistemas para el monitoreo y análisis de procesos industriales pueden utilizar diversas herramientas libres, ya sean Entornos de Desarrollo Integrado (IDEs por sus siglas del inglés *Integrated Development Environment*) o plataformas integradas que permitan implementar un Arquitectura Orientada a Servicios (SOA por sus siglas del inglés *Service Oriented Architecture*) [WSO2 2015]. Algunos ejemplos son:

Eclipse: Entorno de desarrollo multiplataforma. Permite utilizar varios lenguajes de programación como C/C++, PHP, Python y Java. Contiene una amplia gama de herramientas que permiten construir interfaces gráficas de usuario, modelar, graficar, reportar y probar sistemas. Consume algunos recursos del sistema en tiempo de ejecución. [Foundation 2015]

IntelliJ IDEA: Entorno de desarrollo multiplataforma. Muy útil en la implementación de sistemas empresariales, orientados a la web y/o para dispositivos móviles. Posibilita el desarrollo con Java, Scala y Groovy. Incluye un asistente avanzado para el completamiento de código que soporta JavaScript, HTML y CSS entre otras tecnologías. [JetBrains 2015]

WSO2 ESB: El Bus de Servicios Empresariales de WSO2 es un producto ligero, de alto rendimiento y prácticamente sin latencia. Provee soporte para tecnologías como SOAP, servicios web y servicios REST; también es compatible con todos los patrones de integración empresarial. Es completamente configurable desde su interfaz de usuario y sus capacidades son extensibles debido a la gran cantidad de conectores que incorpora. [WSO2 2015] Hasta el momento son más de 100 los conectores implementados para integrar soluciones de terceros; ya sean de contabilidad (ej: Billiving, Paypal), comunicaciones (ej:Twilio, Gmail), desarrollo de aplicaciones (ej: Github, LDAP), administración de recursos humanos (ej: Zoho People, PeopleHR), entre otros sectores. [Infante 2015]

WSO2 AS: El Servidor de Aplicaciones de WSO2 se utiliza para el despliegue de componentes entiéndase servicios de datos, servicios web desarrollados en axis2, jax-ws, spring-ws; también sirve para desplegar aplicaciones web tradicionales y servicios RESTful. Las aplicaciones web desplegadas en esta plataforma pueden compartir la lógica de negocio, datos y procesos a través de todo el ecosistema de herramientas. Reúne las mejores tecnologías de código abierto para

aplicaciones web, habilitando extensiones de monitoreo, clusterización y otras muchas prestaciones. [WSO2 2015]

2.6. Frameworks de aplicaciones

Construir aplicaciones utilizando frameworks otorga al programador ventajas tales como el ahorro de código, la agilización y la simplificación del proceso de desarrollo. [Maldonado 2008] Los frameworks pueden incluir soporte para sistemas, librerías y lenguajes de programación, algunos ejemplos para la implementación de soluciones destinadas al monitoreo y análisis de procesos industriales son:

Vaadin: es un marco de trabajo Java para la creación de aplicaciones web modernas, con un aspecto atractivo, y de excelentes prestaciones. Sus prioridades principales son la productividad del desarrollador, la experiencia del usuario final y su escalabilidad. Con una combinación única de construcción de JVM y HTML5, ayuda a reducir el tiempo de desarrollo y la cantidad de errores durante este proceso. Es de código abierto, se distribuye bajo licencia Apache y cuenta con una comunidad internacional de más de 150.000 desarrolladores. Provee más de 400 componentes de interfaz de usuario, orientados a la usabilidad en las aplicaciones enriquecidas de internet (RIA por sus siglas de inglés *Rich Internet Applications*) [Vaadin 2015]

Spring: proporciona una infraestructura que actúa de soporte para desarrollar aplicaciones Java. Es un contenedor de inversión de control, de código abierto. Sus características pueden ser empleadas en cualquier solución desarrollada en Java, además contiene variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. No impone ningún modelo de programación, por ello este framework se ha vuelto popular en la comunidad al ser considerado una alternativa, sustituto e incluso un complemento al modelo EJB. [Pivotal 2015]

Hibernate: es una herramienta de mapeo objeto-relacional para .Net y Java que facilita la interrelación de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia GNU LGPL. [Hibernate 2015]

2.7. Patrones de diseño

La adopción de un enfoque ingenieril con la calidad adecuada para el desarrollo de sistemas que permitan el monitoreo y análisis de procesos industriales, incluye como aspectos clave la gestión del conocimiento y la compartición de buenas prácticas. Esto a su vez conlleva a la cuestión de cómo capturar, representar y compartir ese conocimiento, un mecanismo para conseguir este objetivo son los patrones de diseño. [Díaz 2007]

Un patrón de diseño nombra, motiva y explica de forma sistemática un diseño general que afronta un problema de diseño recurrente en los sistemas orientados a objetos. Describe el problema, la

solución, cuándo aplicar la solución y sus consecuencias. [GoF 2004] Se puede plantear entonces que un patrón de diseño expresa la relación entre tres componentes esenciales: un problema a resolver, una solución y un determinado contexto donde se repite una situación que vincula a los dos primeros componentes.

Un conjunto importante de patrones de diseño son los publicados en el libro "Design Patterns", comúnmente se conocen como patrones GOF (por sus siglas del inglés *Gang of Four* formada por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides). Son 23 patrones aplicados usualmente para programas de software orientado a objetos.

Siguiendo el libro de GOF los patrones se clasifican según el propósito para el que han sido definidos, del total se seleccionaron aquellos que se adecuan al entorno de desarrollo de aplicaciones para el monitoreo y análisis de procesos industriales: [GoF 2004]

1. Creacionales: solucionan problemas de creación de instancias. Ayudan a encapsular y abstraer dicha instanciación.
 - a. Prototype (Prototipo): especifica los tipos de objetos para crear usando una instancia prototípica, y crea nuevos objetos copiando este prototipo. En el entorno de esta solución es utilizado para crear el prototipo Fermentation que luego será utilizado para instanciar los diferentes objetos correspondientes a las distintas iteraciones del proceso de fermentación.
2. Estructurales: solucionan problemas de composición de clases y objetos.
 - a. Adapter (Adaptador): convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Permite que cooperen clases que de otra manera no podrían por tener interfaces incompatibles. Dentro de la arquitectura propuesta se emplea para hacer compatibles los prototipos de los objetos del negocio con los componentes de interfaz de las vistas, permitiendo que los datos lleguen al usuario.
 - b. Decorator (Decorador): añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. Para el entorno de este trabajo, se utiliza de conjunto con el patrón Adaptador, mientras aquel crea los elementos, este le asigna los atributos correspondientes; por ejemplo cada vez que se selecciona una o más variables del SCADA de los fermentadores del CIM, el decorador indica qué mostrar en cada gráfica.
3. De comportamiento: aportan soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.
 - a. Command (Comando): encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con distintas peticiones, encolar o llevar un registro de las peticiones y poder deshacer las operaciones. En combinación con la tecnología

WSO2, este patrón podrá ser utilizado para encapsular las peticiones a las fachadas de los servicios que consumirán los sistemas para el monitoreo y análisis del proceso industrial fermentación.

- b. Observer (Observador): define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos. Para este caso, se evidencian las bondades de este patrón cuando al modificar la información en las tuplas de la base de datos, todas las variables dependientes cambian de forma automática.

Otro de los patrones de diseño utilizado en esta arquitectura es el Objeto de Acceso a Datos (DAO por sus siglas del inglés *Data Access Object*) uno de los más importantes de la J2EE. [Johnson 2004] Proporciona una interfaz a algún tipo de mecanismo de persistencia, mediante el mapeo de las peticiones del Modelo, realizando las operaciones necesarias para acceder a datos específicos sin exponer a los detalles de la base de datos. En este caso se utiliza una interfaz abstracta para separar la declaración de la implementación de los DAO, utilizados para mapear la información de la base de datos del proceso de fermentación.

2.8. Protocolos de comunicación con servidores

Los programas en el entorno industrial constantemente intercambian datos con sus servidores de correo, de archivos y de aplicaciones. Para definir en cada caso la forma en que se realizarán las peticiones al servidor y en que se enviarán las respuestas a las solicitudes de los clientes, se establecen protocolos de comunicación. [Black 2007] Entre los protocolos recomendados para los sistemas para el monitoreo y análisis de procesos industriales se encuentran:

IMAP (*Internet Message Access Protocol* o Protocolo de Acceso a Mensajes de Internet): ofrece una comunicación bidireccional entre las cuentas de correo electrónico y el cliente web local, esto significa que las acciones ejecutadas en los clientes locales se reflejarán automáticamente en los servidores de correo; esta sincronización evita la pérdida de los nuevos mensajes y la descarga repetida del mismo mensaje. [Crispin 2008]

HTTPS (*Hypertext Transfer Protocol Secure* o Protocolo Seguro de Transferencia de Hipertexto): crea un canal cifrado para el tráfico de información sensible. De este modo se consigue que la información sea transmitida de forma segura y no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, pues lo único que obtendrá será un flujo de datos cifrados. [Rescorla 2000] En esta solución se emplea para comunicar de forma segura los navegadores en las computadoras de los usuarios con los servidores de aplicaciones web.

2.9. Premisas para el uso de la arquitectura propuesta

Por definición, SOA trae muchos beneficios arquitectónicos a un sistema de software distribuido, aporta muchos atributos deseables tales como: reusabilidad, adaptabilidad y facilidades en el mantenimiento de las aplicaciones. En este acápite se plantean algunas proposiciones para aumentar la calidad de las aplicaciones a integrar, procurando mantener buenas prácticas durante el uso de la arquitectura presentada en este trabajo. [Bondi 2010]

2.9.1. Escalabilidad

En ingeniería informática y telecomunicaciones, la escalabilidad es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. En general, también se podría definir como la capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes. [Bondi 2010]

La escalabilidad debe formar parte del proceso de diseño de software puesto que no es una característica separada que se pueda agregar posteriormente. Al igual que con otras funciones de aplicación, las decisiones que se tomen durante las primeras fases de diseño y codificación determinarán en gran medida la escalabilidad de la aplicación. Requiere una pertenencia equilibrada entre dos dominios distintos, software y hardware.

Puesto que la escalabilidad no es un problema de diseño de las aplicaciones independientes, sino más bien se evidencia en las aplicaciones distribuidas. Las aplicaciones distribuidas están también un paso más allá de las tradicionales aplicaciones de cliente-servidor. Las aplicaciones distribuidas son aplicaciones que están diseñadas como aplicaciones de n niveles. La arquitectura de estas aplicaciones distribuidas favorece el diseño de aplicaciones escalables compartiendo recursos, como bases de datos y componentes empresariales.

La escalabilidad supone un factor crítico en el crecimiento de una arquitectura de sistemas integrados. Si un sistema tiene como objetivo crecer en el número de usuarios manteniendo su rendimiento actual, tiene que evaluar dos posibles opciones: con hardware de mayor potencia o con una mejor combinación de hardware y software.

Se pueden distinguir dos tipos de escalabilidad, vertical y horizontal:

- El escalar verticalmente o escalar hacia arriba, significa el añadir más recursos a un solo nodo en particular dentro de un sistema, tal como el añadir memoria o un disco duro más rápido a una computadora.
- La escalabilidad horizontal, significa agregar más nodos a un sistema, tal como añadir una computadora nueva a un programa de aplicación para espejo.

2.9.2. Principios de diseño de servicios en SOA

En la arquitectura orientada a servicios existen un conjunto definido de elementos que se aplican colectivamente para alcanzar los objetivos de las empresas. Dentro de dichos elementos están los principios de diseños de servicios, que son buenas prácticas para proponer una forma de lograr algo basadas en las experiencias pasadas o en la aceptación de toda la industria. Cuando se trata de soluciones de integración, un principio de diseño representa una muy recomendable guía para la configuración de la lógica de la solución de una determinada manera y con ciertos objetivos en mente.

A continuación se proponen algunos principios de diseño que contribuirán al aumento de la interoperabilidad intrínseca de las aplicaciones a integrar, así como al incremento de las opciones de proveedores de contenidos para dichas aplicaciones y la mejoría respecto a la agilidad organizacional: [Erl 2007]

- **Contratos de Servicios (estandarización y diseño):** son la expresión consistente de las capacidades y el propósito general del servicio; establecen las condiciones de contratación, proporcionando las limitaciones y requisitos técnicos, así como cualquier información semántica el propietario del servicio desea hacer pública.
- **Acoplamiento de Servicios (intra-servicio y dependencias del consumidor):** enfatiza en la comprensión de cómo medir y asignar niveles apropiados de acoplamiento de entre las partes de una solución orientada a servicios.
- **Abstracción de Servicios (ocultación de información y tipos de meta-abstracción):** evita la proliferación innecesaria de la información del servicio, los datos correspondientes a la lógica del servicio y su implementación no se publican pues protegen la integridad de la unión formada entre él y sus futuros consumidores.
- **Reutilización de Servicios (diseño comercial y agnóstico):** aboga por la reutilización repetida, se esfuerza por obtener el máximo valor posible de cada pieza de software.
- **Autonomía de Servicios (procesamiento y control de fronteras):** representa la capacidad de autogobernarse. Algo que es autónomo tiene la libertad y el control para tomar sus propias decisiones sin necesidad de aprobación externa. Por lo tanto, el nivel en que un servicio es autónomo representa la medida en que es capaz de actuar independientemente.
- **Descubrimiento de Servicios (Interpretación y comunicación):** ayuda a determinar si la automatización de los requisitos que se necesita cumplir ya existe dentro de un inventario de servicios.
- **Componibilidad de Servicios (diseño de miembros de composición y composiciones complejas):** introduce nuevas consideraciones de diseño que aseguran que los servicios son capaces de participar en múltiples composiciones para resolver varios problemas

mayores; el ensamblaje de las capacidades de diferentes fuentes para resolver un problema más grande es la base de la computación distribuida.

2.9.3. Rendimiento

En el mundo del desarrollo de software de escritorio para un único usuario, los requisitos de rendimiento están claramente definidos. En ese contexto, es importante que el usuario no advierta retrasos significativos al realizar las tareas más habituales. Sin embargo, a medida que se crearon más aplicaciones empresariales en el entorno web, los requisitos de rendimiento de las aplicaciones web comenzaron a rivalizar con las de las aplicaciones para escritorio de un único usuario. Hoy en día, cuando las tecnologías de comunicaciones más rápidas prevalecen, los usuarios exigen un mayor rendimiento de los sitios.

De manera general existen algunos mecanismos para mejorar el rendimiento de las aplicaciones ya sean web o de escritorio:

- Utilizar la memoria caché: esta solución también puede funcionar para aplicaciones de lectura-escritura si se utilizan eventos para expirar los caché cuando sea necesario.
- No almacenar estados volátiles permanentemente: no es necesario persistir absolutamente todos los datos de la aplicación. Como regla general, sólo hay que almacenar en medios permanentes (almacenes operacionales, base de datos, entre otros) información de negocio críticos, necesarios y accionables.
- Ubicar los datos: si para obtener datos es necesario pasar por un balanceador de carga, un servidor web, un servidor de aplicaciones y luego llegar base de datos (en vez de por un balanceador de carga y un servidor web), la escalabilidad y el rendimiento se van a ver afectados. Utilizando SOA con un ESB se elimina este problema pues se garantiza con los endpoint de los servicios el acceso directo a la información que ellos publican.
- Restringir el acceso concurrente a recursos limitados: no sólo aplica a operaciones de lectura, sino también a operaciones transaccionales (aplicando el proceso de "escribir después", como ser usando un mecanismo asíncronico de mensajería).

2.9.1. Conclusiones parciales

En el presente capítulo se describió la arquitectura de integración y el sistema informático que automatiza el flujo de datos del proceso de fermentación en el CIM y las actividades como el registro de los resultados del muestreo. Se detalló el patrón y estilo de integración de la arquitectura; así como las herramientas de desarrollo, patrón arquitectónico y de diseño de la aplicación.

CAPITULO 3. VALIDACIÓN DE LA PROPUESTA

En esta etapa se procede a la validación de la arquitectura de integración para el monitoreo y análisis del proceso de fermentación en el CIM. Se aplican estrategias de evaluación como la escala de Likert fundamentada en la opinión de expertos y la técnica de ladov para comprobar la satisfacción de usuarios. También se implementa un sistema informático basado en los componentes de la arquitectura de integración propuesta para validarla funcionalmente, se realizan pruebas a dicha solución; se despliega la misma en un ambiente experimental en el Centro de Inmunología Molecular y posteriormente se analizan los principales resultados.

3.1. Validación por expertos de la arquitectura aplicando la Escala Likert

Según la ISO 9142:2003 la usabilidad es el grado en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico. [Montero 2009] Otros autores exponen que la usabilidad se podría definir como la facilidad de uso. Específicamente, hace referencia a la rapidez con que se puede aprender a utilizar algo, la eficiencia al utilizarlo, cuál es su grado de propensión al error y cuánto le gusta a los usuarios. [Nielsen 2012]

A partir de las definiciones anteriores, para la presente investigación se especifican los siguientes atributos relativos a la usabilidad de la arquitectura de integración:

- **Aplicabilidad:** capacidad de la arquitectura para permitirle al usuario aprender su aplicación. Facilidad con la que los usuarios desarrollan una interacción efectiva con el mismo. Debe expresarse con la suficiente claridad y síntesis para que sea posible su implementación por otras personas.
- **Factibilidad:** posibilidad real de su utilización y de los recursos que requiere. Facilidad con la que el usuario hace uso del proceso de monitoreo y control con menos pasos o más naturales a su formación específica. Capacidad de la arquitectura para permitirle al usuario entender si es idónea y cómo puede usarse para las tareas y condiciones de uso particulares.
- **Flexibilidad:** relativa a la variedad de posibilidades con las que el usuario y la arquitectura pueden intercambiar información, similitud con tareas anteriores y adaptabilidad. Generalizable y extensible a otros contextos. Capacidad de la arquitectura para permitirle al usuario utilizarlo y controlarlo.
- **Conformidad:** capacidad de la arquitectura para adherirse a las normas, guías o regulaciones relativas al tema. Que tenga pertinencia por su importancia, por su valor social y las necesidades a que da respuesta. Que logre los objetivos para los cuales fue concebida.

Para validar la usabilidad de la arquitectura de integración se confeccionó un cuestionario que puede ser consultado en el Anexo 2. Este fue aplicado a especialistas con conocimientos y/o experiencias en arquitectura de software, integración de sistemas, monitoreo y análisis de procesos industriales. Los expertos podían seleccionar para cada planteamiento alguna de las siguientes respuestas: definitivamente sí (DS), probablemente sí (PS), indeciso (I), probablemente no (PN) y definitivamente no (DN).

La valoración por parte de expertos tuvo en cuenta a 12 profesionales de diferentes entornos que aportan a la industria, tales como: UCI, CIM, ISPJAE, DESOFT y ETECSA. El nivel de competencia de los mismos, se comprobó a partir de la metodología implementada por el Comité Estatal para la Ciencia y la Técnica de la extinta URSS para la elaboración de pronósticos científico-técnicos, la cual se detalla y referencia en [Cruz Ramírez 2012]. En dicha metodología la competencia se determina mediante la siguiente fórmula:

$$K = \frac{K_c + K_a}{2}$$

Donde las variables significan:

K: Coeficiente de competencia.

K_c: Coeficiente de conocimiento o información que tiene el experto acerca del problema, calculado sobre la valoración del propio experto en una escala de 0 a 10 y multiplicado por 0,1.

K_a: Coeficiente de argumentación o fundamentación de los criterios del experto, obtenido como resultado de la suma de los puntos de acuerdo a la escala que muestra el Anexo 3. Esta se incorpora a la encuesta aplicada a los expertos.

A partir de estos datos se especifica el grado de coeficiente de competencia de acuerdo con: si $K > 0,8$ la competencia del experto es Alta; si $0,5 \leq K \leq 0,8$ la competencia del experto es Media; si $K < 0,5$ la competencia del experto es Baja.

El Anexo 4 evidencia el nivel de competencia de los expertos en el objeto de investigación. A partir de estos datos se concluye que el 77 % posee un grado de coeficiente de competencia alto y un 23 % un grado medio, confirmándose que la mayoría de los expertos seleccionados tiene un grado de coeficiente de competencia alto. El número de expertos es mayor que el mínimo (siete) a tener en cuenta [Ramos Blanco 2013], lo cual disminuye el error de la decisión que se tome y mejora la previsión. La Tabla 2 refleja las características que componen a los expertos involucrados en la validación.

Tabla 2 - Características de los expertos involucrados en la validación

Perfil de trabajo	Cantidad
Especialista en arquitectura de software	5
Especialista en monitoreo y control de procesos industriales	6
Especialista en integración de sistemas informáticos	5
Especialista en bases de datos	4
Años de experiencia en la industria del software	
Entre 5 y 10 años	8
Más de 10 años	5
Categoría científica	
Máster	5
Doctor	4
Institución	
UCI	9
CIM	7
ISPJAE	1
SITRANS	1
DESOFT	1
ETECSA	1

Para procesar los resultados de la encuesta realizada fue empleada la escala de Likert [Ávila Baray 2006], donde se otorga una puntuación entre 1 y 5 a cada ítem. Con esta técnica se calculan los porcentajes de concordancia de los expertos con cada una de las posibles respuestas para los planteamientos formulados. Luego se calcula un índice porcentual (*IP*) que integra en un solo valor la aceptación de cada planteamiento por los evaluadores mediante la siguiente fórmula:

$$IP = \frac{5(\% \text{ de } DS) + 4(\% \text{ de } PS) + 3(\% \text{ de } I) + 2(\% \text{ de } PN) + 1(\% \text{ de } DN)}{5}$$

En la siguiente Figura se observa que el índice porcentual relacionado con la valoración de los expertos sobre la usabilidad del marco de trabajo, es mayor que 90 en casi todos los casos.

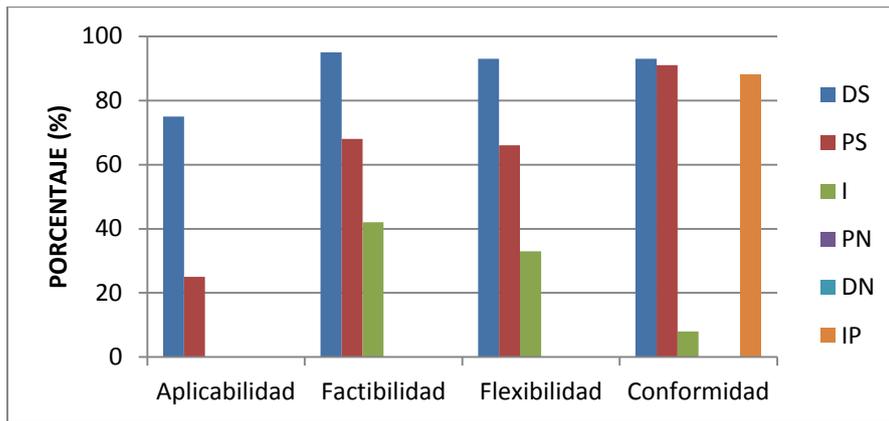


Figura 22 - Valoración de los expertos sobre la usabilidad de la arquitectura propuesta (Elaboración propia)

Esto evidencia la alta valoración por los expertos sobre: la claridad y síntesis de la propuesta arquitectónica; posibilidad de su uso en las organizaciones de la industria según sus elementos y condiciones de uso; su flexibilidad y adherencia a las buenas prácticas de consenso internacional.

3.2. Validación de satisfacción de usuarios utilizando la Técnica V.A Iadov

Se seleccionó esta técnica para la validación cualitativa del sistema informático desarrollado en la presente investigación pues constituye una vía indirecta para el estudio de la satisfacción de un grupo de personas a la hora de desempeñar ciertas actividades. El índice de satisfacción grupal obtenido luego de procesar todas las respuestas de los entrevistados, es una de las bondades de esta práctica. Se basa en el análisis de un cuestionario que tiene una estructura interna determinada que desconoce el entrevistado. Se ha modificado el cuestionario para que pueda ser aplicado en el campo de los Sistemas para el Monitoreo y Análisis de Procesos Industriales, las preguntas pueden verse en el Anexo 5.

Iadov está conformada por cinco preguntas: tres cerradas y dos abiertas. La relación entre las preguntas cerradas se establece a través del denominado "Cuadro Lógico de Iadov", indicando la posición de cada persona en la escala de satisfacción.[Fabre 2014] En el Anexo 6 se presenta el cuadro lógico modificado con las preguntas cerradas para la valoración por parte de los usuarios.

El número resultante de la interrelación de las tres preguntas indica la posición individual de cada usuario en la siguiente escala de satisfacción: clara satisfacción (A), más satisfecho que insatisfecho (B), no definida (C), más insatisfecho que satisfecho (D), clara insatisfacción (E) y contradictoria (C). La cantidad de respuestas por cada categoría es utilizada para calcular el Índice de Satisfacción Grupal (ISG) mediante la fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

Donde N representa el número total de usuarios encuestados.

El valor del ISG permite reconocer las siguientes categorías grupales:

- Insatisfacción: desde (- 1) hasta (- 0,5).
- Contradicción: desde (- 0,49) hasta (+ 0,49).
- Satisfacción: desde (0,5) hasta (1).

Como parte del trabajo de tesis doctoral del DrC. Orestes Febles Díaz, se realizó un programa informático que permite de forma dinámica y sencilla la aplicación del método de ladov. Al utilizar esta aplicación con los datos obtenidos de un grupo de 15 usuarios que fueron encuestados, se obtuvo un $ISG = 0.7911765$.

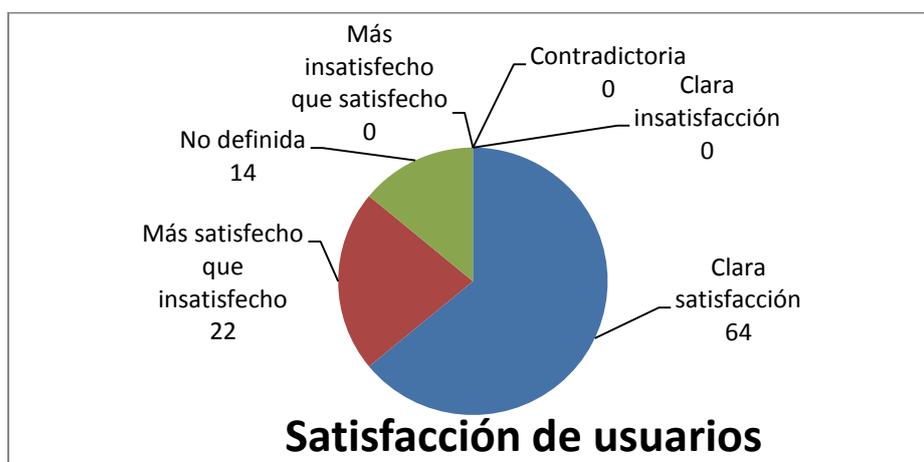


Figura 23 - Nivel de satisfacción de usuarios del SIMAFI CIM. (Elaboración propia)

La técnica de ladov contempla además dos preguntas complementarias de carácter abierto. Estas permiten profundizar en las causas que originan los diferentes niveles de satisfacción. Las respuestas dadas plantearon sugerencias de utilidad para la presente y futuras investigaciones entre las que se destacan:

- Utilizar herramientas informáticas para gestionar rápidamente el elevado volumen de datos que generan la operación industrial de fermentación en el CIM.
- Incluir en la base de datos del sistema toda la información referente a los resultados de los diferentes ensayos clínicos realizados durante el análisis de muestras del contenido del fermentador.

3.3. Implementación de la arquitectura de integración

Se realizó la implementación de un sistema informático basado en la arquitectura de integración propuesta para validar el correcto funcionamiento de la misma. Para ello se utilizaron los componentes arquitectónicos definidos: IntelliJ IDEA como entorno de desarrollo integrado, el lenguaje de programación JAVA, los frameworks Vaadin, Spring e Hibernate, el ESB de WSO2

como plataforma para la orquestación de servicios y como sistema gestor de base de datos se utilizó PostgreSQL.

3.3.1. Arquitectura de la solución

Como parte de esta investigación se implementó en Sistema Informático para el Monitoreo y Análisis de la Fermentación Industrial en el Centro de Inmunología Molecular “SIMAFI CIM”. Como patrón arquitectónico se empleó el Modelo-Vista-Presentador. A continuación se muestra la arquitectura del sistema implementado, el nodo SCADA no forma parte de la solución implementada, pero constituye el elemento proveedor de la información referente al proceso de fermentación en el CIM.

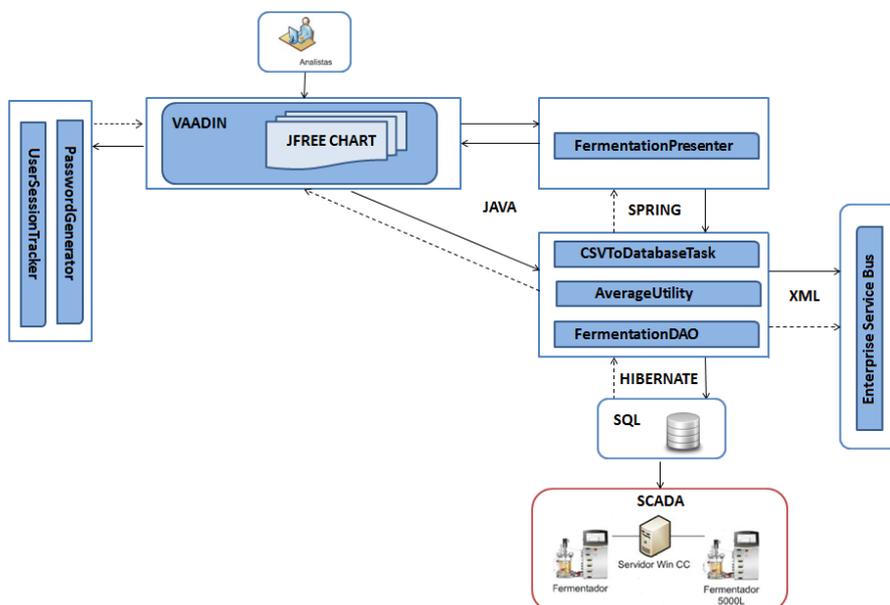


Figura 24 - Arquitectura del “SIMAFI CIM”. (Elaboración propia)

A continuación se exponen los componentes fundamentales de cada estructura dentro del patrón MVP implementado que intervienen en una secuencia completa de funcionamiento del sistema. Los mismo se definen a partir de las funciones que se relacionan a continuación, de las cuales se brindan detalles relacionados con los parámetros que reciben, el valor de retorno, las clases u operaciones con las que colaboran y una breve descripción (ver tablas 3, 4,5).

- Vista

Tabla 3 - Descripción de la función initView

initView	
Parámetros	Ninguno
Valor de retorno	Ninguno

Colaboraciones	<p>En la clase ChartsView realiza una llamada al método buildChartsForm() que se encarga de añadir todos los componentes de interfaz correspondientes al formulario donde el usuario selecciona las gráficas que desea visualizar. Para esto cada selector se mantiene escuchando los cambios que ocurren en la vista con el objetivo de mostrar oportunamente cada gráfico a medida que sean marcados.</p> <p>Por otra parte en la clase FermentationView realiza una llamada al método buildFermentationTable() que trabaja de conjunto con un contenedor personalizado que permite seleccionar las columnas adecuadas según las variables cuyos datos históricos se deseen visualizar.</p>
Descripción	Es la funcionalidad que inicializa la Vista, es la encargada de solicitar al Presentador todos los datos disponibles de las variables SCADA del proceso de fermentación y además gestiona la visualización de dicha información.

- Presentador

Tabla 4 - Descripción de la función getFermentation

getFermentation	
Parámetros	Ninguno
Valor de retorno	Una lista de objetos de tipo Fermentation.
Colaboraciones	Trabaja en conjunto con todas las vistas y todas las implementaciones de los objetos de acceso a datos relativos a la clase de dominio Fermentation.
Descripción	Es la funcionalidad que responde a las peticiones de las vistas enviándoles los objetos con la información de las variables de fermentación obtenidas a partir de la lectura completa de la base de datos del sistema.

- Modelo

Tabla 5 - Descripción de la función FermentationDAOImplementation

FermentationDAOImplementation	
Parámetros	Ninguno

Valor de retorno	Ninguno
Colaboraciones	Responde a las solicitudes transmitidas por los presentadores ChartsPresenter y FermentationPresenter permitiéndoles el acceso a los objetos con los datos de las variables del proceso de fermentación.
Descripción	Su función principal es el mapeo de las columnas de la base de datos con los atributos de la clase de dominio Fermentation, habilitando la creación de objetos que contengan la data de dicho proceso de negocio. Estos objetos son utilizados como fuente de información para los gráficos de progreso y la tabla histórica.

Por otra parte los componentes del bus de servicios de WSO2 para el acceso a la fuente original de los datos y su persistencia:

- Servicios

Tabla 6 - Descripción de la función CSVToDatabaseTask

CSVToDatabaseTask	
Parámetros	Dirección de la fuente de datos, en este caso el archivo de valores separados por coma exportado por el SCADA WinCC (csvDataSourcePath), el carácter utilizado como separador en el archivo especificado (csvColumnSeparator) y la fila donde se debe comenzar la lectura de dicho archivo (startingRow).
Valor de retorno	Ninguno
Colaboraciones	Trabaja en conjunto con las clases CSVConfig y JDBCStoreManager que contienen las configuraciones del archivo .csv y de la base de datos relacional respectivamente.
Descripción	Su función principal es la ejecución de una tarea programada en el bus de servicios de WSO2. El objetivo fundamental es persistir en una base de datos relacional la información contenida en el fichero exportado directamente por el SCADA de los fermentadores del CIM. Se ejecuta en un intervalo especificado por el usuario.

3.3.2. Despliegue de la solución

Para mostrar los diferentes nodos físicos de la red en los que se distribuyen los componentes de la solución implementada teniendo como base la arquitectura propuesta, se desarrolla el diagrama de despliegue tal como describe la siguiente Figura:

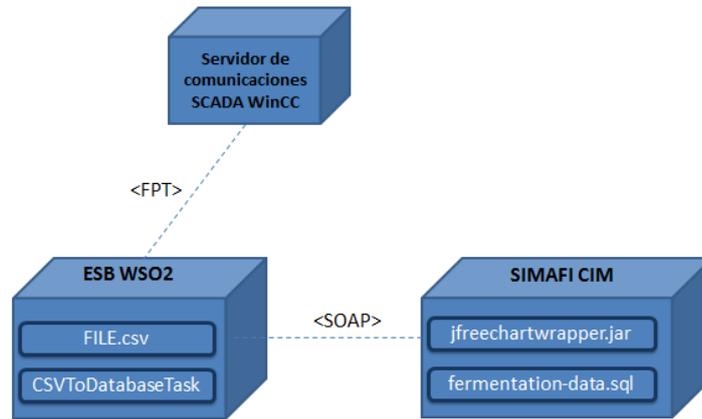


Figura 25 - Diagrama de despliegue de la solución implementada (Fuente: elaboración propia)

Se propone utilizar tres nodos de ejecución:

- Servidor de Comunicaciones SCADA WinCC: se trata del servidor dedicado al componente de comunicación del SCADA, que permite el intercambio con otros sistemas y aunque no forma parte de la solución implementada es necesario para obtener la información que este publica.
- Bus de Servicios WSO2: nodo donde se ejecuta la tarea programada que se encarga de la lectura del archivo de valores separados por coma y de su persistencia en una base de datos relacional. Además a él accede el sistema implementado para hacer consultas a dicha fuente de información y/o crear una copia local de la misma según se considere pertinente.
- SIMAFI CIM: aquí se encuentra el sistema desarrollado para el monitoreo y análisis del proceso de fermentación. En él se grafica el comportamiento de las variables medidas por el SCADA, se muestran los datos históricos de las iteraciones de dicho proceso y se registran los resultados del análisis de muestras del contenido de los fermentadores.

La cantidad de nodos físicos puede variar en dependencia de los recursos con que se cuente. El ESB WSO2 se puede ejecutar en la misma computadora que el SIMAFI CIM.

3.4. Pruebas a la solución

Para verificar el correcto funcionamiento de la arquitectura de integración implementada se realizaron un conjunto de pruebas de software. Entre estas pruebas se encuentran las de unidad, de integración, de sistema y de desempeño. El objetivo fundamental de las pruebas de unidad es centrar el proceso de verificación en la menor unidad de diseño de software y probar que cada una de las partes individuales funciona de forma correcta. [Beck 2003]. Las pruebas de integración son aquellas en las cuales los distintos módulos que conforman el software ya han sido probados

individualmente y se combinan para ser probados en grupos, validándose la comunicación entre ellos, además de otras funcionalidades específicas. Las pruebas de integración se realizan después de las pruebas de unidad y antes de las pruebas de sistema. [Lewis 2004] Las pruebas de sistema se realizan teniendo en cuenta el sistema forma integral, para evaluar el cumplimiento con los requerimientos especificados. Es el momento en que se prueban todas las funcionalidades del sistema y se deben corregir los errores o incongruencias detectadas en cuanto al funcionamiento global del sistema.

Existen distintas técnicas de pruebas que proporcionan criterios para generar casos de pruebas que provoquen fallos en los programas, estas técnicas se agrupan en: técnicas de caja blanca o estructurales y técnicas de caja negra o funcionales, la primera se basa en un minucioso examen de los detalles procedimentales a evaluar, por lo que es necesario conocer la lógica del programa, sin embargo la segunda se basa en la realización de pruebas sobre la interfaz del programa, entiéndase por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. [Pressman 2005]

Al sistema desarrollado se le realizaron pruebas de caja negra con el objetivo de medir la funcionalidad operativa del software. Se probó por separado cada uno de los componentes para validar sus funcionalidades específicas descritas anteriormente. Se validó también la comunicación e integración de los mismos mediante el paso de mensajes entre ellos. Finalmente se le aplicaron un conjunto de pruebas al sistema de forma integral, los resultados corroboraron que cada una de las funcionalidades previstas para cada componente tiene correspondencia con los requisitos especificados y funcionan correctamente, pudiéndose comprobar que:

1- La tarea programada en el ESB WSO2 utiliza como fuente de información el archivo de valores separados por coma publicado por el SCADA WinCC y almacena en una base de datos relacional las variables registradas en los fermentadores de ANTYTER.

csv_id [PK] integer	data_time timestamp without time zone	flujo_de_oxigeno_valu double precision	flujo_de_aire_valu double precision	ph_valu double precision	do_valu double precision	rpm_valu double precision	presion_valu double precision	temperatura_valu double precision
1	2015-08-26 00:00:01	1.68584597110748	0.943405270576477	6.9541335105896	47.8213348388672	40.291805267334	0.306645542383194	36.990234375
2	2015-08-26 00:00:07	1.65501344203949	0.946806013584137	6.95462799072266	47.7507514953613	40.3594169616699	0.30944836139679	36.9755821228027
3	2015-08-26 00:00:13	1.64287412166595	0.943891167640686	6.95578145980835	47.8544502258301	40.284610748291	0.305223882198334	36.9829063415527
4	2015-08-26 00:00:19	1.5938333272934	0.941462278366089	6.95372152328491	47.8709983825684	40.3709297180176	0.308270424604416	36.991455078125
5	2015-08-26 00:00:25	1.6040301322937	0.942433893680573	6.95619344711304	47.7882270812988	40.2860450744629	0.306361228227615	36.99267578125
6	2015-08-26 00:00:31	1.64845812320709	0.94486266374588	6.95438098907471	47.7681884765625	40.3953819274902	0.307579845190048	36.991455078125
7	2015-08-26 00:00:37	1.65865516662598	0.941219329833984	6.95677042007446	47.7246284484863	40.4025840759277	0.304898858070374	36.9853515625
8	2015-08-26 00:00:43	1.70721065998077	0.945105612277985	6.95644092559814	47.7978096008301	40.3953895568848	0.307214230298996	36.9841270446777
9	2015-08-26 00:00:49	1.65986883640289	0.941462218761444	6.95372152328491	47.7612113952637	40.3824424743652	0.306523710489273	36.9890098571777
10	2015-08-26 00:00:55	1.67467844486237	0.94486278295517	6.95619344711304	47.7315940856934	40.438549041748	0.307254880666733	36.9731407165527
11	2015-08-26 00:01:01	1.67783415317535	0.943891227245331	6.95586395263672	47.7821311950694	40.2601470947266	0.304858237504959	36.9853477478027
12	2015-08-26 00:01:07	1.66739499568939	0.94461989402771	6.95446300506592	47.7455368041992	40.2659034729004	0.308757841587067	36.986572265625
13	2015-08-26 00:01:13	1.64287447929382	0.943405389785767	6.95322704315186	47.8413772583008	40.3378372192383	0.306848704814911	37.006103515625
14	2015-08-26 00:01:19	1.60475850105286	0.944619834423065	6.95347452163696	47.7891044616699	40.3738059997559	0.30477699637413	36.986572265625
15	2015-08-26 00:01:25	1.62685072422028	0.942190885543823	6.95330953598022	47.8919258117676	40.4155311584473	0.307011127471924	37.0000038146973
16	2015-08-26 00:01:31	1.65549898147593	0.94049072265625	6.95454549789429	47.755126953125	40.3766899108887	0.308311015367508	36.9792442321777
17	2015-08-26 00:01:37	1.63024973869324	0.942676663398743	6.95380401611328	47.745532989502	40.4140892028809	0.304817616939545	36.9780235290527
18	2015-08-26 00:01:43	1.66326797008514	0.946077227592468	6.95421600341797	47.7786483764648	40.3881912231445	0.308107942342758	36.984130859375
19	2015-08-26 00:01:49	1.66011166572571	0.946077108383179	6.95479297637939	47.7490348815918	40.4169654846191	0.30676743388176	36.990234375
20	2015-08-26 00:01:55	1.6698225736618	0.942919552326202	6.95495748519897	47.8509674072266	40.3953895568848	0.307011157274246	36.996337890625

Figura 26 – Base de datos del proceso de fermentación (Fuente: elaboración propia)

2- El Sistema Informático para el Monitoreo y Análisis de la Fermentación Industrial en el Centro de Inmunología Molecular grafica los datos históricos de las diferentes variables medidas por el SCADA durante toda una corrida del proceso, entre ellas flujo de oxígeno, presión y temperatura.

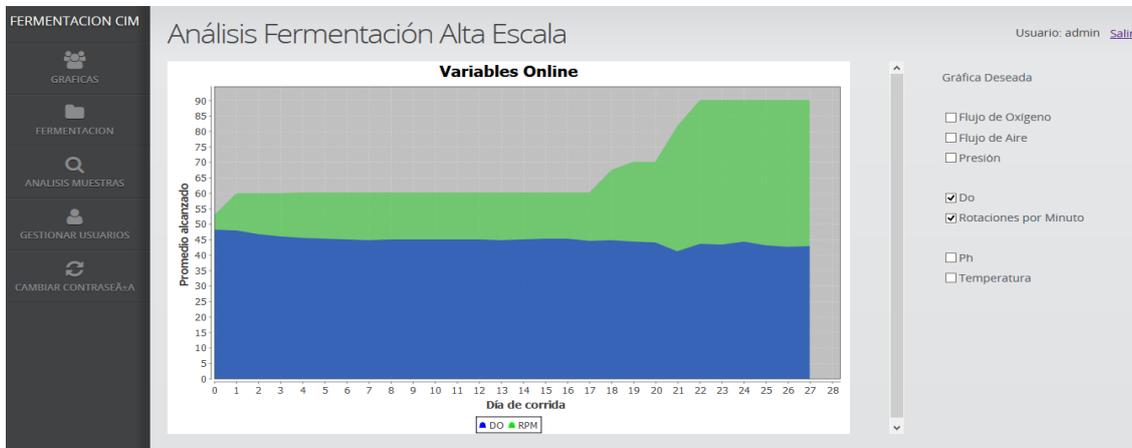


Figura 27 - Gráficas de área mostrando las variables medidas por el SCADA en el CIM (Elaboración propia)

3- El SIMAFI CIM registra los resultados de análisis de muestras tomadas al contenido de los fermentadores, automatizando actividades del proceso de fermentación que actualmente se realizan de forma manual.

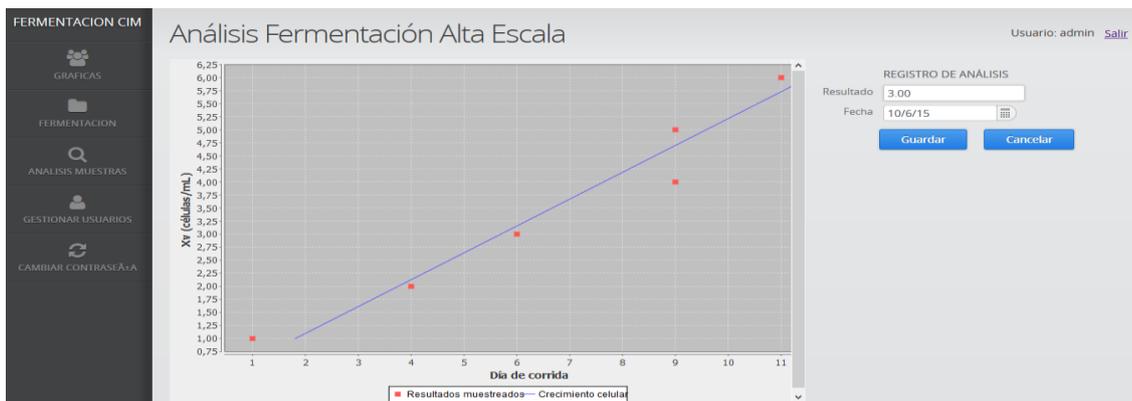


Figura 28 - Graficas de series de tiempo mostrando los resultados del análisis de muestras (Elaboración propia)

De esta forma es posible afirmar que con la implementación y puesta en marcha del sistema informático para el monitoreo y análisis del proceso de fermentación en el CIM, se aumenta la eficiencia del mismo en cuanto al tiempo en que se realiza.

3.4.1. Ambiente de prueba.

Para la correcta realización de las pruebas de sistema, se prepararon dos escenarios que permitieron validar el funcionamiento en conjunto de los componentes SCADA WinCC, ESB WSO2 y SIMAFI CIM, y por tanto la integración funcional de la solución. Para ello se contó con una serie de recursos que facilitaron el trabajo de ejecutar cada caso de prueba. Este apartado describe los detalles sobre el ambiente en que desarrollaron las pruebas para validar la solución, especificando los elementos físicos y lógicos.

Recursos físicos

Se utilizaron tres computadoras con las siguientes características:

- Computadora # 1:
 - Procesador Intel(R) Core (TM) i3-2120 CPU @ 3.30GHz
 - Memoria 4GB de RAM
 - Tarjeta de red a 100Mbit/s

- Computadora # 2:
 - Procesador Intel(R) Core (TM) i3-3120 CPU @ 2.50 GHz
 - Memoria de 4GB de RAM
 - Tarjeta de red a 100Mbit/s

- Computadora # 2:
 - Procesador Intel(R) Pentium(R) quad core CPU @ 2.58 GHz
 - Memoria de 4GB de RAM
 - Tarjeta de red a 100Mbit/s

Recursos lógicos

A continuación se hace referencia a los elementos de software que se utilizaron para llevar a cabo las pruebas:

1. Archivo 3232TA1407.csv compartido vía ftp.
2. Driver JDBC PostgreSQL para la interacción del WSO2 ESB y el SIMAFIN con la base de datos.
3. Códigos de prueba implementados en Java que simulan componentes del HMI del SCADA.

3.5. Diseño experimental

Para comprobar la efectividad de la propuesta se diseñaron y aplicaron varias pruebas experimentales que permitieron realizar una valoración acerca de la inmediatez con la que se manejan y transmiten grandes cantidades de datos entre los componentes distribuidos.

En la realización del experimento intervienen tres componentes fundamentales: el módulo de comunicación del SCADA, el bus de servicios de WSO2 y las interfaces de comunicación implementadas como parte de la solución propuesta.

La prueba consiste en medir el tiempo en milisegundos que demora el proceso de envío y recepción desde un publicador hacia varios suscriptores ejecutándose remotamente, y realizando 10 iteraciones para las cantidades 51200, 102500, 205100 y 410200 tuplas leídas de las variables configuradas en el SCADA. La prueba se realizó en tres computadoras con las características

explicadas en el epígrafe anterior, y aunque los resultados fueron favorables, pudiera afirmarse que de haberse utilizado un ambiente ideal para sistemas SCADA, en referencia tanto a las características físicas de las computadoras como a las de red, entonces los resultados hubieran sido superiores.

Durante las pruebas se logró evaluar tanto el desempeño de la propuesta desarrollada, permitiendo finalmente validar la solución propuesta y llegar a un grupo de conclusiones a partir de comparar los resultados obtenidos con respecto al tiempo que demora actualmente realizar todo este proceso de forma manual.

Para cada uno de los escenarios se ejecutaron 100 corridas y se obtuvieron los tiempos promedios en cada uno de ellos como se muestra en la siguiente Tabla 10 donde se detallan los resultados del desempeño de la solución de integración desarrollada:

Tabla 7 - Resultados del experimento

Escenario	Cantidad de suscriptores	Cantidad de variables	Cantidad de tuplas leídas	Tiempo promedio (ms)
1	1	1	1	0.302
2	1	5	51200	1513
3	2	5	102500	3027
4	2	8	205100	6053
5	10	10	410200	12108

Teniendo en cuenta que el proceso que realizan los tecnólogos, analistas y directivos cuando deben realizar el análisis y monitoreo de una corrida del proceso de fermentación consiste en bajar al piso de la planta operacional, exportar manualmente el archivo .csv con los datos disponibles hasta el momento, copiar el archivo en un medio de almacenamiento portátil, trasladarse al piso de trabajo, procesar el .csv hasta obtener una hoja de cálculos excel, incorporar manualmente a este último los resultados del análisis de muestras, graficas en el excel el progreso de la iteración en curso de la fermentación y finalmente imprimir varias copias de estos gráficos para que de conjunto analistas y directivos puedan estudiarlas y tomar las decisiones operacionales pertinentes, el tiempo mínimo que se podría demorar el más rápido de los operadores siempre sería mayor que un minuto, sin embargo con la arquitectura de integración, esta actividad se realiza en un tiempo máximo de aproximadamente 12.108 segundos, lo que

significa que a través de la arquitectura el despacho de los datos ocurre aproximadamente 5 veces más rápido.

A medida que van aumentando la cantidad de tuplas simultáneas que se generan en el SCADA y la cantidad de variables a analizar y responsables a notificar, el tiempo en que se realizan estas notificaciones a través la plataforma va aumentando como se muestra en la siguiente Figura. Sin embargo, si es el operador el encargado de realizar la notificación en el mismo escenario, con la misma cantidad de variables, el tiempo en el que las realiza sería considerablemente mayor.

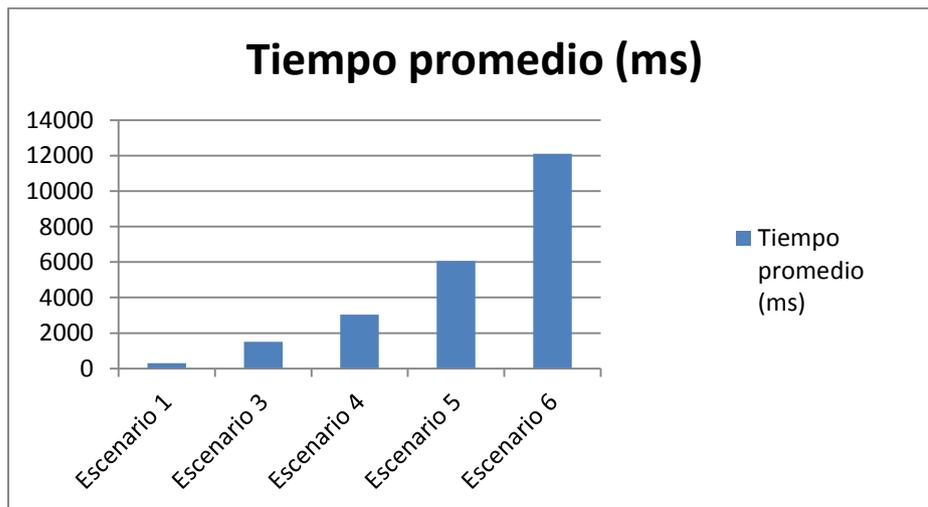


Figura 29 - Comportamiento del tiempo en el experimento. (Elaboración propia)

Nótese que para graficar 410276 tuplas de variables leídas, lo cual constituye una cantidad bastante severa, utilizadas simultáneamente por 10 suscriptores, se obtuvo un tiempo promedio aproximado de 12 segundos, lo cual sería imposible de lograr por un operador. Por tanto mientras mayor sea el número de alarmas y responsables a notificar, más grande será la diferencia entre el proceso manual y el proceso automatizado propuesto en esta investigación.

3.1. Conclusiones parciales

En el presente capítulo se describió la implementación de la arquitectura de integración a través del desarrollo del SIMAFI CIM y se validaron las funcionalidades propuestas para el mismo. Las pruebas aplicadas con el objetivo de comprobar el correcto funcionamiento del sistema en conjunto con el SCADA WinCC y el ESB WSO2 fueron satisfactorias, demostrando la efectividad del mismo.

Los resultados de las pruebas experimentales que se muestran en la Figura 31 y la tabla 7, así como los análisis derivados de las mismas, muestran que la arquitectura de integración propuesta y el sistema informático desarrollado en la presente investigación contribuyen a aumentar la eficiencia en el proceso de fermentación del CIM con respecto al mecanismo actual.

CONCLUSIONES

Al término de la presente investigación se arriba a las siguientes conclusiones:

- El análisis de las características de las aplicaciones para monitoreo y análisis de procesos industriales y el estudio de los aspectos necesarios para definir una arquitectura de integración de sistemas, permitió construir el marco teórico referencial de la investigación y obtener elementos relevantes para elaborar la propuesta arquitectónica de integración y desarrollar el sistema informático asociado a la fermentación en el CIM.
- Al completar una propuesta arquitectónica para la integración de las aplicaciones que ya existen en el CIM y de otras que puedan ser desarrolladas, se incluyeron varios componentes que facilitarán el monitoreo y análisis del proceso de fermentación.
- La validación cualitativa de la arquitectura de integración y del sistema informático desarrollado, se realizó a partir del criterio de un grupo de expertos y usuarios respectivamente, permitiendo evaluar la usabilidad de ambas propuestas.
- Mediante un diseño experimental que integró el sistema desarrollado al escenario tecnológico del CIM, se pudo afirmar que la solución propuesta disminuye el tiempo de transferencia de los datos entre los componentes distribuidos hasta 5 veces, respecto al desempeño previo que muestra el intercambio, persistencia y análisis mediante gráficos que actualmente es realizado de forma manual.

RECOMENDACIONES

Al concluir la presente investigación se proponen, a manera de recomendaciones, una serie de tareas para ampliar y dar continuidad al trabajo realizado:

- Independizar el módulo de registro de análisis de muestras y desarrollarlo como una aplicación independiente que abarque todas las variantes y aspectos de este subproceso de la fermentación. Por supuesto esta aplicación se incorporaría a la arquitectura de integración propuesta.
- Enriquecer la arquitectura de integración incorporando componentes tales como un almacén de datos operacional y un almacén de datos históricos, que utilicen como fuente primaria y común la base de datos desarrollada en esta investigación.
- Desarrollar un portal corporativo con tecnología Liferay de forma que se puedan incluir en él como portlets la aplicación desarrollada en este trabajo y otras ya existentes en el CIM que también fueron implementadas con Vaadin.

BIBLIOGRAFÍA

- A. DANEELS. WHAT IS SCADA? *International Conference on Accelerator and Large Experimental Physics Control Systems* [Type of Work]. 2009.
- AAAS. Definitions of biotechnology [online]. [New Zeland]: 2009. Available from World Wide Web:<<http://ehrweb.aaas.org/ehr/books/glossary.html#biotechnology>>.
- ÁVILA BARAY, H.L. Introducción a la metodología de la investigación [online]. [Chihuahua, México]: 2006.
- BECK, K. Test-Driven Development By Example [online]. 2003.
- BIOTECNOLOGÍA, C.D. ¿Qué es la biotecnología? In U.D. CONCEPCIÓN. Chile, 2014.
- BLACK, U. *Redes de computadores: protocolos, normas e interfases*. edited by ALFAOMEGA. Edtion ed. México, 2007.
- BONDI, A. Characteristics of scalability and their impact on performance. In *Proceedings of the international workshop on Software and performance*. 2010.
- CIM. Capacidad productiva. In. Habana: Centro de Inmunología Molecular, 2012.
- CIM. Capacidad Productiva para la producción de Anticuerpos Monoclonales (ANTYTER). In. Habana: Centro de Inmunología Molecular, 2012.
- CIM. Centro de Inmunología Molecular. In., 2012.
- CIM. Control y Aseguramiento de la Calidad. In. Habana: Centro de Inmunología Molecular, 2012.
- CRISPIN, M. Internet Message Access Protocol - SORT and THREAD Extensions. *Carnegie Mellon University* [Type of Work]. 2008. Available from Internet:<<http://www.rfc-editor.org/info/rfc5256>>.
- CRUZ, M.B.A. Biotecnología en Cuba: origen y resultados alcanzados. *Centro de Estudios de la Economía Cubana* [Type of Work]. 2010. Available from Internet:<https://www.nodo50.org/cubasigloXXI/pensamiento/anaya_300410.pdf>.
- CRUZ RAMÍREZ, M. Perfeccionamiento de un instrumento para la selección de expertos en las investigaciones educativas. . *REDIE. Revista electrónica de investigación educativa* [Type of Work]. 2012, vol. 14, pp. 167-179. ISSN 1607-4041.
- DEZA, J.P. Gestión de base de datos con scada para el control automatizado de una válvula de control proporcional. *Revista de la Facultad de Ingeniería Industrial* [Type of Work]. 2011, vol. 14. ISSN 1560-9146.
- DÍAZ, P. *Ingeniería de la web y patrones de diseño*. edited by P. HALL. Edtion ed., 2007. ISBN 84-2054609-7.
- ERL, T. *SOA: Principles of Service Design*. Edtion ed., 2007. ISBN 007-6092043232.
- FABRE, A.F.D.C. Validación mediante criterio de usuarios del sistema de indicadores para prever, diseñar y medir el impacto en los proyectos de investigación del sector agropecuario. *Revista Ciencias Técnicas Agropecuarias* [Type of Work]. 2014, vol. 23. ISSN 2071-0054.

FISHER, M. *Spring Integration in Action*. Edtion ed., 2012. ISBN 978-1935182436.

FOUNDATION, E. Eclipse desktop & web IDEs. In., 2015.

FOWLER, M. The evolution of MVC and other UI architectures. 2006. Available from Internet:<<http://martinfowler.com/eaDev/uiArchs.html>>.

GARCÍA, E.F. Diseño e implementación de un sistema de control y monitoreo basado en HMI-PLC para un pozo de agua potable. *Ingeniería Investigación y Tecnología* [Type of Work]. 2014, vol. XV. ISSN 1405-7743.

GARCÍA, I.O.Z. Bioinformática - ¿Qué es bioinformática? In S. CIENCIA. Universidad Ibero Americana 2009.

GOF *Design Patterns: Elements of Reusable Object-Oriented Software*. Edtion ed., 2004. ISBN 0-201-63361-2.

GÓMEZ, F. Control avanzado en estaciones depuradoras de aguas residuales. *Universidad de Huelva* [Type of Work]. 2015. Available from Internet:<<http://www.uhu.es/fernando.gomez/tecoagua/TecoAgua/Presentaciones.html>>.

HIBERNATE. Hibernate. Everything data. In., 2015.

HOHPE, G. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Edtion ed., 2003. ISBN 978-0321200686.

IEEE. Defining architecture. *Systems and software engineering — Architecture description* [Type of Work]. 2015. Available from Internet:<<http://www.iso-architecture.org/ieee-1471/defining-architecture.html>>.

INDIANA, U.O. What is FTP, and how do I use it to transfer files? 2014. Available from Internet:<<https://kb.iu.edu/d/aerg>>.

INFANTE, J. WSO2 ESB: Almacén de conectores a la carta. In *Desarrollo SOA en la Comunidad WSO2-Latinoamérica*. 2015.

ISO. Standards - ISO. 2015. Available from Internet:<<http://www.iso.org/iso/home/standards.htm>>.

JETBRAINS. IntelliJ IDEA, The Most Intelligent Java IDE. In., 2015.

JOHNSON, R. *J2EE Development without EJB*. Edtion ed., 2004. ISBN 0-7645-5831-5.

KRUCHTEN, P. *The Rational Unified Process: An Introduction* Edtion ed., 2003. ISBN 978-0321197702.

LEWIS, W.E. Software Testing and Continuous Quality Improvement [online]. 2004.

LLERENA, M.R. Estado Actual de la Arquitectura Empresarial del Dominio IFA del CIM. In D.D.P. DAEIFA. Habana, 2013, p. 29-40.

MALDONADO, D.M. Utilizando Framework de programación en las Empresas. In., 2008.

MICHAEL NILGES, J.P.L. *Bioinformatics definition in Centre d'Informatique pour la Biologie*. edited by I.P. UNITÉ DE BIO-INFORMATIQUE STRUCTURALE. Edtion ed., 2014.

MONTERO, H. *Informe APEI sobre usabilidad*. edited by A.P.D.E.E. INFORMACIÓN. Edtion ed. España: Gijón, 2009. ISBN 978-84-692-3782-3.

MSDN. MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java. 2007. Available from Internet:<<https://msdn.microsoft.com/en-us/library/ff649571.aspx>>.

NIELSEN, J. Usability 101: Introduction to Usability 2012. Available from Internet:<<http://www.nngroup.com/articles/usability-101-introduction-to-usability/>>.

PIVOTAL, S. Spring. In., 2015.

PRESSMAN, R. Ingeniería de software. Un enfoque práctico. [online]. 6ta. 2005.

RAMOS BLANCO, K. Proceso Base de Ingeniería de Requisitos para las pequeñas y medianas empresas de desarrollo de software. . In. La Habana Universidad de las Ciencias Informáticas, 2013, vol. Tesis en opción al título de Máster en Calidad de Software.

RESCORLA, E. HTTP Over TLS. 2000. Available from Internet:<<https://tools.ietf.org/html/rfc2818>>.

RODRÍGUEZ MEDINA, G. Análisis estratégico del proceso productivo en el sector industrial. *Revista de Ciencias Sociales* [Type of Work]. 2012, vol. 8, pp. 135-156. ISSN 1315-9518.

ROMÁN, J. Sistemas de Control Distribuidos y Controladores Lógicos Programables. *Automatización de Plantas Industriales* [Type of Work]. 2013. Available from Internet:<<http://www.abb.com/>>.

ROTEM-GAL-OZ, A. *SOA Patterns*. Edtion ed., 2012. ISBN 978-1933988269.

SALVENDY, G. *Handbook of Industrial Engineering: Technology and Operations*. Edtion ed.: Wiley-Interscience, 2005. ISBN 0-471-33057-4.

SERGIO MARTIG, S.C. *Interfaces para el Monitoreo y Control de Procesos Industriales*. Edtion ed. Bahía Blanca, Argentina: Instituto de Investigación en Ciencia y Tecnología Informática (IICyTI), 2014.

VAADIN. JFreeChart wrapper for Vaadin. 2010. Available from Internet:<<https://vaadin.com/directory/-/directory/addon/jfreechart-wrapper-for-vaadin>>.

VAADIN. Introduction - vaadin.com. In., 2015.

WHITT, M.D. *Successful Instrumentation and Control Systems Design*. Edtion ed.: International Society of Automation, 2012. ISBN 978-1-936007-45-5.

WSO2. WSO2 - lean . enterprise . middleware. In., 2015.

WSO2. WSO2 Application Server - 100% Open Source and Cloud Native. In., 2015.

WSO2. WSO2 Enterprise Service Bus - Only 100% Open Source ESB In., 2015.

ANEXOS

Anexo 1

Comparación y análisis de las arquitecturas de las herramientas similares al SIMAFI CIM

No.	Herramientas	1. Soporte para la web	4. Tipo de licencia	6. Comunicación con otros sistemas	7. Visualización de variables online	8. Visualización de variables offline	9. Indicadores compuestos	10. Reportes	11. Histórico	12. Persistencia de datos	13. Orientada a servicios	14. Orientada a eventos	15. Arquitectura escalable	16. Gestión de usuarios	17. Control de acceso
1	Aimax	Si	Propietaria	Si	No	Si	No	Si	Si	Si	No	Si	No	No	Si
2	BAM WSO2	Si	Open source	Si	Si	No	No	Si	Si	Si	Si	Si	No	No	Si
3	CUBE	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	No	Si	Si
4	DABin	Si	Propietaria	Si	Si	No	No	Si	Si	Si	No	Si	No	No	No
5	DotProject	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	No	No	Si
6	FIX	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	Si	Si	Si
7	IFIX	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	Si	Si	Si
8	Lookout	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	Si	Si	Si
9	Monitor Pro	Si	GPL	Si	Si	No	No	Si	Si	Si	No	Si	No	No	Si
10	SCADA InTouch	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	Si	Si	Si
11	SCADA GALBA	Si	Open source	Si	Si	Si	No	Si	Si	Si	No	Si	Si	Si	Si
12	SCADA Movicon	Si	Propietaria	Si	No	Si	No	Si	Si	No	No	Si	No	No	Si
13	SCADA Oasys	Si	Propietaria	Si	No	Si	No	Si	Si	No	No	Si	Si	Si	Si
14	SCADA WinCC	Si	GPL	Si	Si	Si	No	Si	Si	Si	No	No	No	No	Si
15	SCADA Yokogawa	Si	Propietaria	Si	Si	No	No	Si	Si	Si	No	Si	Si	Si	Si
16	SPMAP	Si	Propietaria	Si	Si	No	No	Si	No	No	No	Si	No	No	No
17	SYSMAC SCS	Si	GPL	Si	Si	Si	No	Si	No	Si	No	No	No	Si	Si
18	WizAAM	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	No	No	Si
19	Wizcon Supervisor	Si	Propietaria	Si	Si	Si	No	No	Si	Si	No	Si	Si	Si	Si
20	Zenon Supervisor	Si	Propietaria	Si	Si	Si	No	Si	Si	Si	No	Si	Si	Si	Si

Anexo 2

Encuesta de usabilidad de la arquitectura de integración:

Terminología

- **Aplicabilidad:** capacidad de la arquitectura para permitirle al usuario aprender su aplicación. Facilidad con la que los usuarios desarrollan una interacción efectiva con el mismo. Debe expresarse con la suficiente claridad y síntesis para que sea posible su implementación por otras personas.
- **Factibilidad:** posibilidad real de su utilización y de los recursos que requiere. Facilidad con la que el usuario hace uso del proceso de monitoreo y control con menos pasos o más naturales a su formación específica. Capacidad de la arquitectura para permitirle al usuario entender si es idónea y cómo puede usarse para las tareas y condiciones de uso particulares.
- **Flexibilidad:** relativa a la variedad de posibilidades con las que el usuario y la arquitectura pueden intercambiar información, similitud con tareas anteriores y adaptabilidad. Generalizable y extensible a otros contextos. Capacidad de la arquitectura para permitirle al usuario utilizarlo y controlarlo.
- **Conformidad:** capacidad de la arquitectura para adherirse a las normas, guías o regulaciones relativas al tema. Que tenga pertinencia por su importancia, por su valor social y las necesidades a que da respuesta. Que logre los objetivos para los cuales fue concebida.

1) Según su criterio cómo describiría la *aplicabilidad* del proceso:

Alta

Media

Baja

Otro criterio (Argumente):

2) Según su criterio cómo describiría la *factibilidad* del proceso:

Alta

Media

Baja

Otro criterio (Argumente):

1) Según su criterio cómo describiría la *flexibilidad* del proceso:

Alta

Media

Baja

Otro criterio (Argumente):

2) Según su criterio cómo describiría la *conformidad* del proceso:

Alta

Media

Baja

Otro criterio (Argumente):

Anexo 3

Escala de puntuación para calcular coeficiente de argumentación

No.	Fuentes de argumentación	Grado de influencia de cada una de las fuentes en sus criterios		
		Alto	Medio	Bajo
1	Estudios teóricos realizados por usted	0.31	0.23	0.1
2	Experiencia adquirida durante su vida profesional	0.51	0.43	0.3
3	Conocimiento de investigaciones y/o publicaciones internacionales	0.05	0.04	0.03
4	Conocimiento de investigaciones y/o publicaciones nacionales	0.04	0.03	0.02
5	Su propio conocimiento sobre el tema de investigación	0.05	0.04	0.03
6	Intuición	0.04	0.03	0.02
	Total	1.0	0,8	0,5

Anexo 4

Nivel de competencia de los expertos

No.	Kc	Ka						K	Grado	
		1	2	3	4	5	6			Valor
1	1	0,31	0,51	0,05	0,04	0,05	0,04	1,0	1	Alto
2	1	0,31	0,51	0,05	0,04	0,05	0,04	1,0	1	Alto
3	1	0,31	0,51	0,05	0,04	0,05	0,04	1,0	1	Alto
4	1	0,31	0,51	0,05	0,04	0,05	0,04	1,0	1	Alto
5	0,8	0,23	0,43	0,04	0,03	0,04	0,03	0,08	0,8	Medio
6	1	0,31	0,43	0,05	0,04	0,04	0,03	0,9	0,95	Alto
7	1	0,31	0,43	0,05	0,04	0,04	0,03	0,9	0,95	Alto
8	0,8	0,23	0,43	0,04	0,03	0,04	0,03	0,08	0,8	Medio
9	1	0,23	0,51	0,04	0,04	0,05	0,04	0,91	0,96	Alto
10	1	0,23	0,51	0,04	0,04	0,05	0,04	0,91	0,96	Alto
11	1	0,23	0,51	0,04	0,03	0,05	0,04	0,9	0,95	Alto
12	1	0,23	0,51	0,04	0,03	0,05	0,04	0,9	0,95	Alto
13	0,8	0,23	0,43	0,04	0,03	0,04	0,03	0,08	0,8	Medio

Anexo 5

Encuesta para validar satisfacción de usuarios del SIMAFI CIM

Nombre de la persona: _____

Área donde labora: _____

Rol: () Operador () Tecnólogo () Analista () Jefe de Proyecto () Director () Otro

1) ¿Considera usted posible realizar el monitoreo y análisis de procesos industriales sin utilizar un sistema informático?

2) ¿Si usted decidiera emplear un sistema informático para el monitoreo y análisis de procesos industriales utilizaría esta solución?

3) ¿Le satisface este sistema informático para apoyar el monitoreo y análisis de procesos industriales en el Centro de Inmunología Molecular?

4) ¿Qué elementos considera positivos de este sistema?

5) ¿Cuáles son sus sugerencias para la continuidad del desarrollo e implantación de este sistema informático en la infraestructura CIM?

Anexo 6

Cuadro lógico de ladov para medir el nivel de satisfacción de usuarios del SIMAFI CIM

	1. ¿Considera usted posible realizar el monitoreo y análisis de procesos industriales sin utilizar un sistema informático?								
	No			No Sé			Sí		
3. ¿Le satisface este sistema informático para apoyar el monitoreo y análisis de procesos industriales en el CIM?	2. ¿Si usted decidiera emplear un sistema informático para el monitoreo y análisis de procesos industriales utilizaría esta solución?								
	Sí	No Sé	No	Sí	No Sé	No	Sí	No Sé	No
Me satisface mucho	1	2	6	2	2	6	6	6	6
No me satisface tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me satisface	6	3	6	3	4	4	3	4	4
No me satisface nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4