

Universidad de las Ciencias Informáticas



Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas

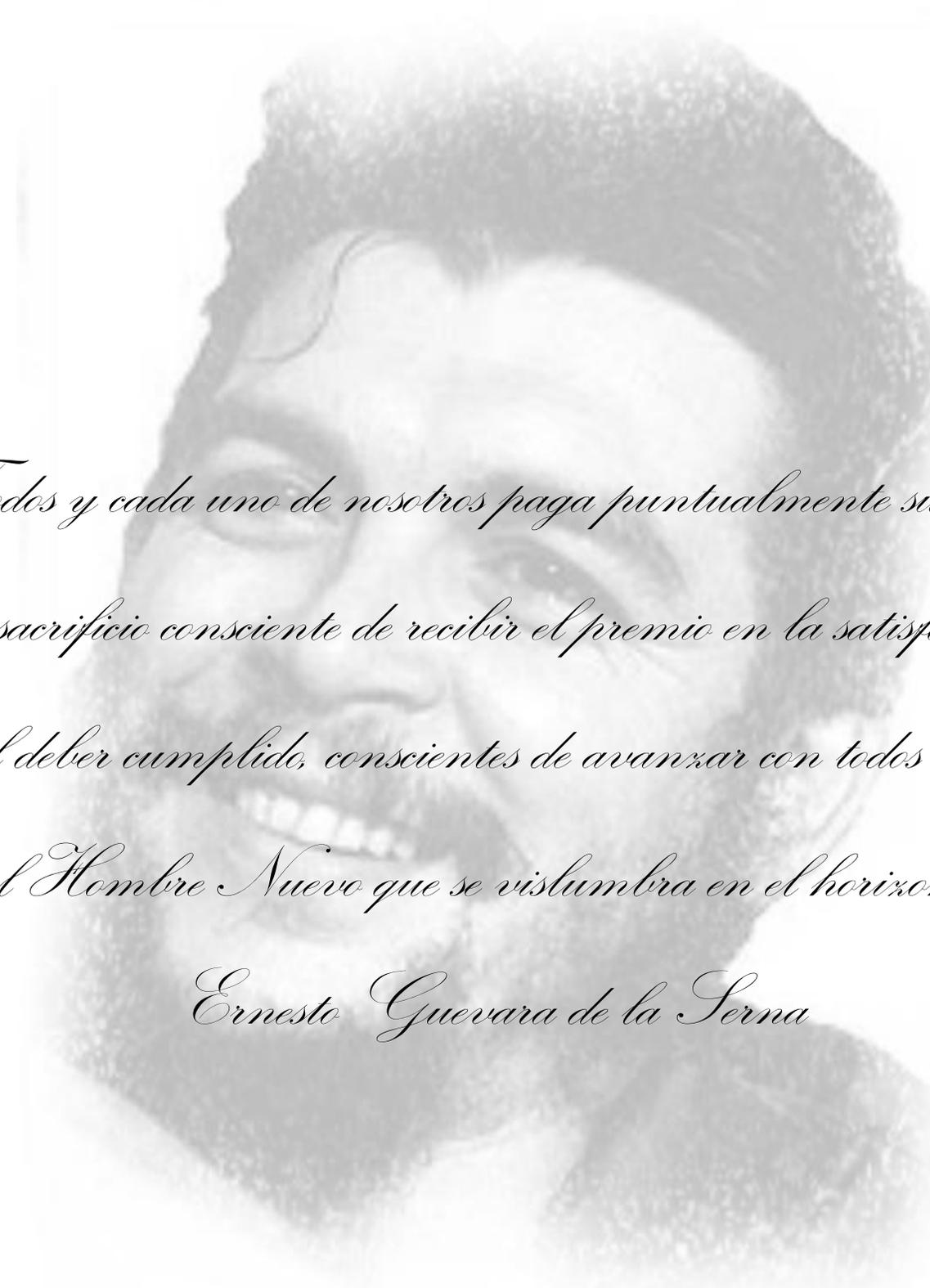
Título: Propuesta de integración de un motor de flujos de trabajo para el portal
UCIBioSoft.

Autor: Daniel Chávez Cruz

Tutor: Ing. Yuniesky Armentero Moreno

La Habana, Cuba

Junio, 2011



*Todos y cada uno de nosotros paga puntualmente su cuota
de sacrificio consciente de recibir el premio en la satisfacción
del deber cumplido, conscientes de avanzar con todos hacia
el Hombre Nuevo que se vislumbra en el horizonte.*

Ernesto Guevara de la Serna

Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de diciembre del año 2011

Daniel Chávez Cruz

Ing. Yuniesky Armentero Moreno

Firma del tutor



Agradecimientos

A mi mamá, por ser la mejor del mundo.

A mi papá, porque sé que esto es un sueño para él también.

Al Tutu, mi hermano, por ser universitario de la vida, y graduarse con honores.

A mi chiquitica linda, por ser un pedazote de mi.

A mi familia en general, mi abuelita, tías, primas y primos. A Kevin, Adrián, Delenis y al próximo que está por venir.

A mi hermanita Hera, por todo su amor.

A mis hermanitos de siempre; Dayli, Osvaldo, Carlitos, Yaniel y Quintero.

A la gente del café, por tantas noches de café y tesis, sin ustedes no habría llegado hasta aquí.

A las personas que me han ayudado en este último año, Freddy, Jorge Luis, Yaikiel; gracias por apoyarme y confiar en mí.

A los gordos JJ y Roberto Carlos, por compartir tantos momentos de alegría.

A mis amistades que me han acompañado durante estos cinco años.

Daniel Chávez Cruz



Dedicatoria

A mi madre.

A mi familia y amigos.

Daniel Chávez Cruz



Resumen

Resumen.

El Departamento de Bioinformática de la Universidad de las Ciencias Informáticas cuenta con el portal web UCIBioSoft el cual brinda servicios bioinformáticos a la comunidad científica de la UCI contando con varios módulos como alineación de secuencias, visualización de estructuras moleculares y construcción de filogenias. Surge la necesidad de incorporarle un módulo de gestión de flujos de trabajo, para llevar a cabo de manera automática los análisis de grandes set de datos y las búsquedas que involucren más de una base de datos o programas. Este trabajo tiene como objetivo proponer un motor de flujos de trabajo al portal UCIBioSoft para la ejecución de los flujos de trabajo. Proponiendo después de haber realizado un profundo estudio del arte, los materiales y métodos para la realización del módulo, entre los que se encuentran el motor de flujos de trabajo y demás herramientas que lo componen, la metodología a seguir y la representación arquitectónica del módulo de gestión de flujos de trabajo. Para validar la propuesta de desarrollo del módulo se utilizó el método de consulta a especialistas, Delphi, obteniendo resultados satisfactorios.

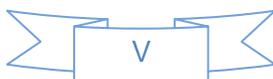
PALABRAS CLAVES

Flujos de trabajo, workflow, servicios web, bioinformática, UCIBioSoft.

Índice General

Índice general

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 UCIBioSoft	4
1.1.1 Prototipo de funcionamiento del portal UCIBioSoft	5
1.2 Flujos de trabajo	5
1.3 Aplicaciones manejadoras de flujos de trabajo	6
1.3.1 Arquitectura de las aplicaciones manejadoras de flujos de trabajo	7
1.3.2 Ejemplos de aplicaciones manejadoras de flujos de trabajo.....	9
1.4 Selección del motor de flujos de trabajo	12
CAPÍTULO 2: MATERIALES Y MÉTODOS	15
2.1 Materiales	15
2.1.1 Entorno de desarrollo integrado.....	15
2.1.2 NetBeans	15
2.1.3 Framework.....	15
2.1.4 Open-Jacob Draw2D	16
2.1.5 Framework de Interfaz de usuario	16
2.1.6 Direct Web Remoting.....	18
2.1.8 Portlet	20
2.1.9 LifeRay.....	21
2.2 Métodos	21
2.2.1 Metodología de desarrollo.....	21
2.2.2 Extreme Programming.....	23
2.2.3 Propuesta de Arquitectura del módulo de gestión de flujos de trabajo en el portal UCIBioSoft.....	23



Índice General

CAPÍTULO 3: VALIDACIÓN	27
3.1 Método Delphi	27
3.2.1 Elección de Expertos	31
3.2.2 Desarrollo práctico y explotación de resultados	34
CONCLUSIONES	44
RECOMENDACIONES	46
GLOSARIO DE TÉRMINOS	47
BIBLIOGRAFÍA	49

Índice General

Índice de Figuras

Figura 1. Prototipo general del portal UCIBioSoft.....	5
Figura 2. Arquitectura de las aplicaciones manejadoras de flujos de trabajo.....	7
Figura 3. Esquema de funcionamiento de Direct Web Remoting.	19
Figura 4. Coeficiente de competencia.	34
Figura 5. Frecuencias acumuladas.	38
Figura 6: Gráfica de puntos de corte.	40
Figura 7: Resultados generales de la encuesta.	42

Índice de Tablas

Tabla 1. Características de interés para el portal UCIBioSoft presentes en las diferentes aplicaciones de flujos de trabajo.	13
Tabla 2. Comparación entre metodologías ágiles y robustas.	22
Tabla 3. Grado de experiencia del posible experto.	31
Tabla 4. Fuentes de Argumentación.	32
Tabla 5. Cálculo del coeficiente de argumentación.	32
Tabla 6. Ejemplo de selección de uno de los expertos encuestados.....	33
Tabla 7. Encuesta a los expertos.	35
Tabla 8. Frecuencias acumuladas.	37
Tabla 9. Frecuencias absolutas Acumuladas.	38
Tabla 10. Frecuencias Relativas Acumulativas.	39
Tabla 11: Puntos de corte.....	39

Introducción

INTRODUCCIÓN

Durante el siglo XX con el incremento de la complejidad de técnicas de investigación biológicas y la capacidad de cálculo de las computadoras, se comenzaron a desarrollar investigaciones referentes a las ciencias biológicas usando para esto las tecnologías computacionales. Por lo que se fueron fundiendo la rama de la biología y la informática surgiendo la disciplina científica llamada Bioinformática (1). Esta comprende diferentes áreas del conocimiento tales como la biología, la química, la física y la informática lo cual le da un enfoque interdisciplinario.

En la actualidad a pesar de ser todavía un área relativamente nueva del conocimiento, ha habido un aumento en los volúmenes de datos biológicos y de herramientas informáticas para el desarrollo de nuevos estudios. La bioinformática cuenta con una gran diversidad de recursos, tales como: bases de datos, servidores, herramientas de software y algoritmos que involucran un sin número de datos con formatos heterogéneos. La reconocida revista *Nucleic Acids Research* dedica cada año sus primeros números a la publicación del listado actualizado de bases de datos y servidores de aplicaciones bioinformáticas disponibles, en los números de enero y julio del año 2010, se reportan una selección cuidadosa de 1230 bases de datos que cubren varios aspectos de la biología celular y molecular (2) y 1500 vínculos a servidores web de herramientas bioinformáticas (3) respectivamente. Sin embargo, gran parte de la información que se procesa, no está lo suficientemente cohesionada a pesar de estar relacionada entre sí y los servidores web no son suficientes para lograr la automatización *in silico*¹ de muchos procedimientos complejos en bioinformática.

La integración de datos heterogéneos es imprescindible para lograr una mejor y más amplia visión de toda la información disponible, para llevar a cabo de manera automática los análisis de grandes *set* de datos y las búsquedas que involucren más de una base de datos o programas. Finalmente, ante este creciente flujo de información, solo una estrecha integración entre los datos y las herramientas de análisis permitirá lograr una minería de datos real. En este contexto, se requiere el uso de técnicas innovadoras y de un sistema capaz de mejorar la accesibilidad de la información. Una solución a esta problemática, desde el campo de la bioinformática, ha sido la incorporación de vínculos semánticos entre los problemas biológicos y los recursos bioinformáticos a través de ontologías (4), así como la utilización de Sistemas Manejadores de Flujos de Trabajo (en inglés, *Workflows Management Systems*)

¹ In Silico: Término que hace referencia a los experimentos realizados mediante el uso de computadoras.



Introducción

unido a los servidores web para la integración y análisis de datos y herramientas heterogéneas. Aún no existe un ambiente de alto nivel capaz de clasificar las herramientas y proveer un acceso amigable a las aplicaciones a través de la web. En ese caso los usuarios podrían concentrarse en la lógica de la aplicación (en los aspectos biológicos) dejando a la plataforma el trabajo de componer las aplicaciones, el formato de entrada de los datos, proporcionar las opciones y coleccionar los resultados. Otro aspecto importante es la accesibilidad a la plataforma a través de un portal web. Un portal para bioinformática que, además de permitir el acceso a las herramientas y bases de datos existentes, debe permitir la formulación, composición y ejecución de problemas; así como la visualización y anotación de los resultados.

En la Universidad de las Ciencias Informáticas (UCI), el Departamento de Bioinformática, tiene como objetivo realizar el portal web UCIBioSoft que ofrezca en su primera versión algunos servicios básicos para el análisis de secuencias y estructuras proteicas. Para lograr su objetivo, pretende incorporar las últimas tendencias tecnológicas en el campo de la bioinformática, y que a su vez trate de ocultar los detalles técnicos y la complejidad de su funcionamiento a los usuarios finales. Para ello es necesario la incorporación de un módulo que permita la realización de flujos de trabajo con el objetivo de automatizar la secuencia de acciones, actividades o tareas necesarias para la ejecución de un experimento *in silico*, hecho que conlleva al siguiente **problema científico**: ¿Cómo ejecutar flujos de trabajo en el portal UCIBioSoft?

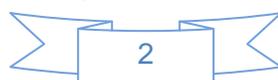
De lo anterior se define como **objeto de estudio**: los sistemas manejadores de flujos de trabajo, delimitando el **campo de acción**: los motores de flujos de trabajo de los sistemas manejadores de flujos de trabajo.

Se traza como **objetivo general** de la investigación proponer un motor de flujos de trabajo al portal UCIBioSoft para la ejecución de los flujos de trabajo.

Para cumplir con el objetivo propuesto se han trazado los siguientes **objetivos específicos**:

1. Seleccionar un motor de flujos de trabajo para ejecutar los mismos en el servidor del portal.
2. Definir herramientas y metodologías necesarias para la integración del motor de flujos de trabajo en la propuesta.
3. Validar la propuesta a partir del método Delphi.

Para lograr las metas planteadas se le dará cumplimiento a las siguientes **tareas**:



Introducción

1. Revisión bibliográfica sobre los sistemas manejadores de flujos de trabajo.
2. Selección del motor para ejecutar los flujos de trabajo en el servidor del portal.
3. Selección de las herramientas para la integración del motor de flujos de trabajo al portal UCIBioSoft.
4. Propuesta de representación arquitectónica del módulo de gestión de flujos de trabajo.
5. Validación de la propuesta a través del método Delphi.

El presente trabajo está estructurado en Resumen, Introducción, tres capítulos como cuerpo fundamental, Conclusiones, Recomendaciones y Bibliografía. Los capítulos corresponden a los diferentes aspectos temáticos:

Capítulo 1-Fundamentación Teórica:

Brinda una descripción general de las tendencias actuales para el desarrollo de portales web para bioinformática. Se realiza además un análisis de las diferentes aplicaciones existentes que permiten la integración de datos para la composición y ejecución de flujos de trabajo en bioinformática; enfatizando en aquellas de mayor impacto en la comunidad científica.

Capítulo 2- Materiales y Métodos:

Se propone la arquitectura conceptual del módulo de gestión de flujos de trabajo, así como las herramientas y metodología a utilizar para la validación de la propuesta de selección.

Capítulo 3-Validación:

Se valida la propuesta de selección del motor de flujos de trabajo y de la arquitectura conceptual del módulo de gestión de flujos de trabajo a través del método Delphi, mostrando el resultado del mismo.

Capítulo 1: Fundamentación Teórica

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el desarrollo del capítulo se tratarán los conceptos más importantes concernientes al estado del arte. Partiendo desde el objeto de estudio los sistemas manejadores de flujos de trabajo hasta el campo de acción los motores de flujos de trabajo de los sistemas manejadores de flujos de trabajo, proporcionando una descripción de las diferentes aplicaciones existentes en el mundo capaces de realizar flujos de trabajo. Dando respuesta así a las tareas: revisión bibliográfica sobre los sistemas manejadores de flujos de trabajo; selección del motor para ejecutar los flujos de trabajo en el servidor del portal.

1.1 UCIBioSoft

UCIBioSoft es un portal web dirigido a especialistas y científicos en bioinformática, desarrollado por el Departamento de Bioinformática de la UCI. Dicho portal cuenta con foros, blogs, bibliotecas de documentos, galería de imágenes y diferentes servicios para el trabajo científico. Además presenta un módulo de autenticación que le permite al usuario personalizar su sesión de trabajo y un módulo de análisis de secuencias que permite realizar alineaciones múltiples de secuencias escogiendo el programa para ello, que puede ser clustalw (5), t-cofee (6) o muscle (7), utilizando la plataforma de cálculo distribuida T-Arenal (8) para ejecutar los mismos.

El portal está desarrollado sobre el Sistema Manejador de Contenidos (CMS por sus siglas en inglés) y contenedor de portlets Liferay, aprovechando la capacidad de soporte de portlets de este. Se incorporó el framework Spring Portlet MVC, ajustándose al patrón de diseño Modelo-Vista-Controlador y como lenguaje de programación, fue escogido Java. El Apache Tomcat, será el servidor de aplicaciones donde estará ejecutándose la aplicación.

UCIBioSoft pretende ser un portal web de referencia para los biólogos, que incorpore las últimas tendencias tecnológicas para el desarrollo de experimentos bioinformáticos, utilizando para esto, nuevas herramientas, aplicaciones y servicios, siendo esta una vía para la socialización del conocimiento.

Capítulo 1: Fundamentación Teórica

1.1.1 Prototipo de funcionamiento del portal UCIBioSoft

El funcionamiento del portal UCIBioSoft comienza con una petición de usuario a través del protocolo HTTP al servidor Apache Tomcat (9). Este gestiona dicha petición y la envía al contenedor de portlet Liferay que, según la petición del usuario, muestra el portlet solicitado. Posteriormente en dependencia de la complejidad del cálculo que se vaya a realizar, si este requiere de gran capacidad de cómputo para su solución se utiliza la plataforma de cálculo distribuido T-Arenal. La información referente a los perfiles de usuario es gestionada en una base de datos de PostgreSQL permitiendo la persistencia de los datos.

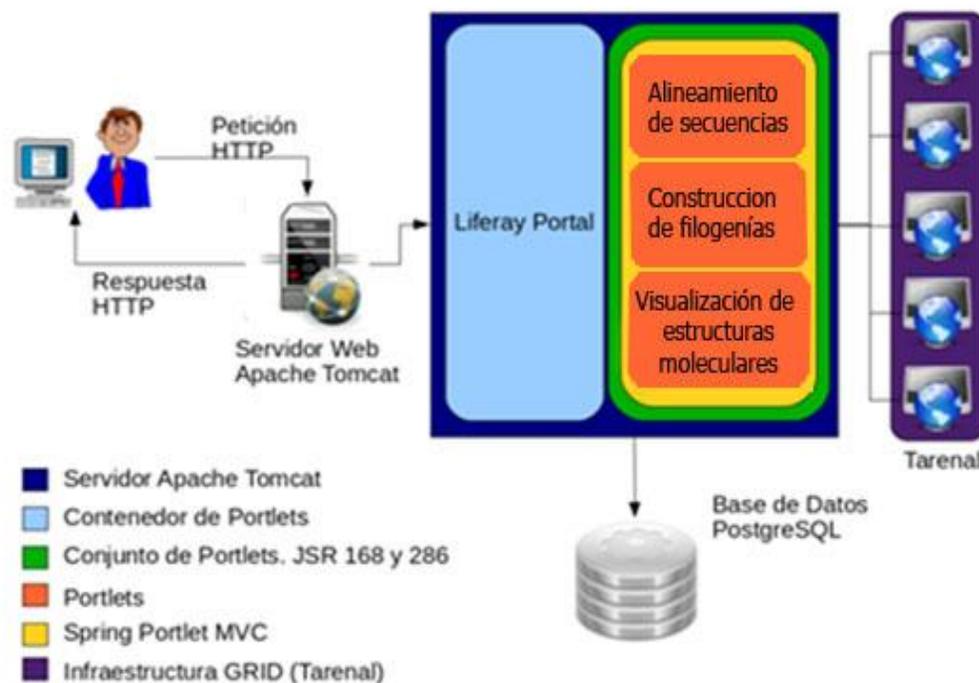


Figura 1. Prototipo general del portal UCIBioSoft.

1.2 Flujos de trabajo

Los flujos de trabajo (en inglés workflow) (10) son una tecnología que comenzó alrededor de los años 90 por la cual surge la organización *Workflow Management Coalition* (WfMC) (11), entidad reguladora de los estándares referentes a este tema. A partir de la creación de la organización WfMC, aumentaron el

Capítulo 1: Fundamentación Teórica

desarrollo y la investigación de manera acelerada, por las posibilidades que brinda para mejorar el rendimiento de las organizaciones y comunidades científicas.

Un flujo de trabajo consiste en una secuencia de tareas de acuerdo con un conjunto de reglas y procedimientos: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de estas. De manera que se ejecuten las tareas de forma automatizada obteniendo mayor agilidad en la ejecución de las secuencias de acciones.

En un flujo de trabajo hay un número de actividades finitas las cuales siempre se llevan a cabo de la misma manera. Por cada actividad o tarea hay un conjunto de datos de entrada, que al ejecutarse dicha tarea, son procesados a través del evento definido para esta actividad, devolviendo datos de salida, los cuales pasan de actividad en actividad y se obtienen por consiguiente los resultados finales.

En el mundo se han desarrollado diferentes aplicaciones manejadoras de flujos de trabajo tanto científicas como empresariales, teniendo como objetivo lograr mayor agilidad y automatización de los procesos.

1.3 Aplicaciones manejadoras de flujos de trabajo

Para poder implementar un flujo de trabajo se necesita de un sistema o aplicación manejadora de flujos de trabajo (10), los cuales automatizan la secuencia de acciones, actividades o tareas utilizadas para la ejecución del proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo.

Algunas de estas aplicaciones se limitan a su área en particular y otras permiten la comunicación con otras aplicaciones externas. Estas aplicaciones se caracterizan principalmente, por una adecuada integración con los sistemas de información actuales: bases de datos, gestión documental, mensajería, ERP (12), entre otros; permitiendo desde la ampliación de un simple proceso, hasta la integración de varios procesos interrelacionados.

Las aplicaciones de gestión de flujos de trabajo se caracterizan por las siguientes funcionalidades:

- Modelado y representación de los procesos de flujos de trabajo a través de iconos y líneas uniendo

Capítulo 1: Fundamentación Teórica

iconos.

- Selección de flujos de trabajo predefinidos en respuesta a una actividad específica.
- Programación de las actividades a realizar en cada tarea.
- Control de ejecución, monitorización y recuperación ante fallos.

1.3.1 Arquitectura de las aplicaciones manejadoras de flujos de trabajo

Las aplicaciones manejadoras de flujos de trabajo presentan una arquitectura propia de cada una de ellas, pero siempre guiada por la organización *Workflow Management Coalition* la cual desarrolló un estándar de arquitectura (13) para lograr una estructura genérica en el desarrollo de aplicaciones de este tipo. Dicha arquitectura se muestra en la figura 2.

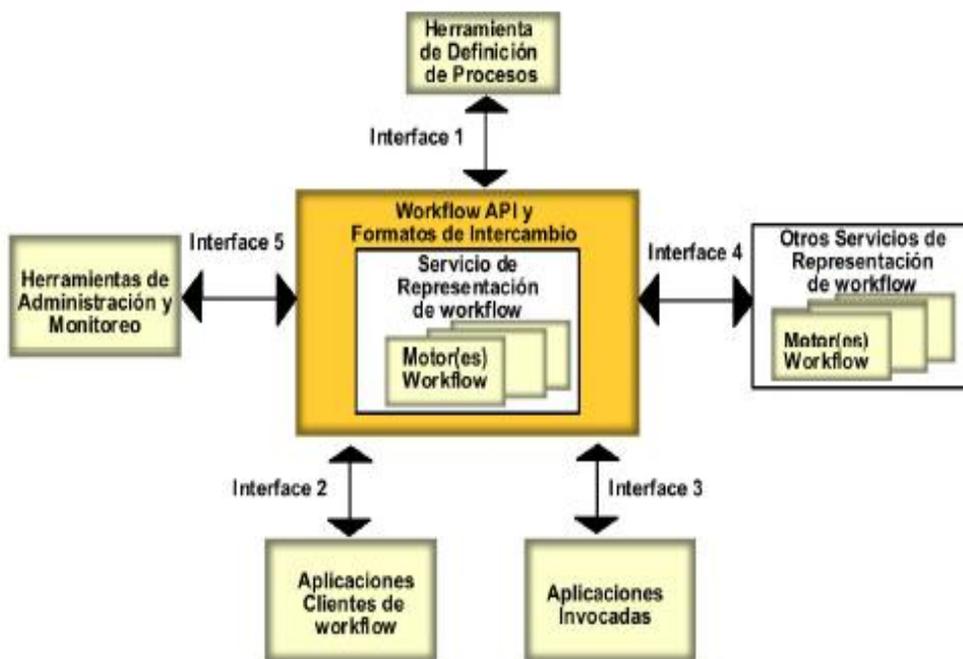


Figura 2. Arquitectura de las aplicaciones manejadoras de flujos de trabajo.

Esta arquitectura de referencia cuenta con distintos componentes que interactúan entre sí, a través de

Capítulo 1: Fundamentación Teórica

interfaces que permiten el intercambio de datos entre dichos componentes. A continuación, una explicación de cada uno de estos componentes:

- **Motor de flujos de trabajo**

Elemento de la arquitectura de los sistemas manejadores de flujos de trabajo más importante. Provee facilidades para la interpretación de los procesos, control de cada una de las instancias de los procesos y del ambiente de ejecución del flujo de trabajo, así como la navegación entre aplicaciones y la invocación de aplicaciones externas.

- **Servicio de representación de flujos de trabajo**

Se encarga de crear, gestionar y ejecutar cada una de las instancias del modelo de flujos de trabajo. En este componente es donde se encuentra alojado el motor de flujos de trabajo. En caso de que se encuentre en un entorno de ejecución distribuido, puede contener más de un motor que controlen por separado distintas partes de la ejecución del proceso.

- **Flujos de trabajo API e intercambio de formatos**

Es un conjunto de Interfaces de Programación de Aplicaciones (API por sus siglas en inglés, *Application Programming Interface*) y funciones de intercambio que soporta el servicio de representación de flujos de trabajo. Las API permiten la interacción del servicio de representación de flujos de trabajo con otros recursos y aplicaciones.

- **Herramientas de definición de procesos (interface 1)**

Son todas las herramientas utilizadas para modelar y describir un determinado flujo de trabajo. La salida de este proceso debe ser interpretada en tiempo de ejecución por el motor de flujos de trabajo.

- **Aplicaciones Clientes de flujos de trabajo (interface 2)**

Representa las entidades de software utilizadas que requieran participación humana para su realización. Dichas aplicaciones clientes de flujos de trabajo deben interactuar con el motor de flujos de trabajo de manera que sean flexibles en cuanto a identificadores de procesos y actividades, estructuras de datos y diferentes alternativas de comunicación.

Capítulo 1: Fundamentación Teórica

- **Aplicaciones invocadas (interface 3)**

Representa las aplicaciones que un sistema de flujos de trabajo, puede utilizar para realizar ciertas actividades que deben interactuar directamente con el motor de flujos de trabajo. La aplicación que se invoque puede ser local, es decir, estar en la misma plataforma o encontrarse en otra plataforma dentro de una red, en este caso la definición del proceso debe contener la información necesaria para poder encontrar dicha aplicación en la red.

- **Otros Servicios de representación de flujos de trabajo (interface 4)**

Permite la interoperabilidad entre diferentes sistemas de flujos de trabajo en la que se pueden realizar, desde actividades simples hasta un completo intercambio de definición de procesos y datos en tiempo de ejecución.

- **Herramientas de administración y monitoreo (interface 5)**

Permite que distintos servicios de ejecución de flujos de trabajo compartan las mismas funciones de administración y monitorización del sistema, que pueden ser el control de los recursos y la supervisión de estados del proceso.

1.3.2 Ejemplos de aplicaciones manejadoras de flujos de trabajo

A continuación las aplicaciones manejadoras de flujos de trabajo que responden al modelo de referencia de arquitectura, exponiendo el funcionamiento y características de cada una de ellas.

- **Wildfire**

Wildfire (14) es una aplicación de escritorio en la que se puede diseñar visualmente y ejecutar los flujos de trabajo de propósito bioinformático. Para la ejecución de una secuencia de tareas, la aplicación Wildfire exporta el diagrama en un lenguaje llamado GEL (*Grid Execution Language*) que es enviado a un intérprete del lenguaje GEL para su ejecución. El intérprete GEL puede estar ubicado en la misma computadora donde se encuentra ejecutándose la aplicación Wildfire o puede estar alojada en un servidor. En caso de que se encuentre alojado en un servidor, los archivos necesarios para la

Capítulo 1: Fundamentación Teórica

ejecución son enviados a través del protocolo seguro SSH (15). El intérprete de GEL puede ejecutar el flujo de trabajo localmente o puede ejecutarlo utilizando tecnología Grid a través de Condor o en clúster con plataforma LSF, PBS o SGE (16), para distribuir los procesos en distintas computadoras. Wildfire al igual que GEL está desarrollado en el lenguaje de programación Java, es multiplataforma y se distribuye bajo licencia GPL (17), mientras que GEL solo está disponible en plataformas UNIX y es gratuito para uso no comercial.

- **Kepler**

Kepler (18) es una aplicación de escritorio, orientada para dar soporte a los flujos de trabajo de propósito general y en la que se pueden diseñar y ejecutar los flujos de trabajo. Es una herramienta potente en la ejecución e interacción en tiempo real, pues proporciona el acceso a datos de manera local o remota y utiliza la potencia de cálculo de las tecnologías de red como son: Globus (19), la web y los servicios Soaplab (20). Proporciona facilidades para monitorizar los flujos de trabajo como son la recuperación ante fallos y el control de origen de los servicios. Soporta múltiples fuentes de datos independientemente de su estructura o tamaño, pueden ser datos tabulados en hojas de cálculo, textos planos, binarios o matrices y permite el acceso a disímiles bases de datos.

Kepler contiene una gran gama de módulos por defecto que puede ser ampliada mediante la búsqueda en los repositorios disponibles, y al estar programado en el lenguaje de programación Java, es extensible permitiendo desarrollar nuevos módulos y adicionárselos. Es independiente de plataforma, de código abierto y se distribuye bajo la licencia BSD (21).

- **Taverna**

Taverna (22) es una potente herramienta para el diseño, desarrollo y ejecución de experimentos y servicios de diversa índole en el plano científico; estos pueden ser tanto biológicos como químicos o pertenecer a otras ramas de la ciencia. La herramienta Taverna simplifica las tareas siguiendo un esquema de flujos de trabajo, combinando diferentes recursos cuidadosamente ordenados, donde cada recurso procesa ciertos datos y genera resultados. Estos recursos pueden ser repositorios de información como las bases de datos *Extensible Binary Meta Language* (EBML) (23) y *Swiss-Prot* (24), las cuales contienen información relacionada con la ciencia de la vida, o herramientas computacionales analíticas como *ClustalW* o *BLAST* (25). Taverna es un sistema de gestión de flujos

Capítulo 1: Fundamentación Teórica

de trabajo independiente de plataformas y de código abierto.

Las características fundamentales de Taverna son:

- Posee todas las funcionalidades de un sistema de gestión de flujos de trabajo escalable y extensible.
- Posee acceso a recursos locales y remotos, así como a herramienta de análisis.
- No es restringido a servicios predeterminados, lo que posibilita una rápida incorporación de nuevos servicios.
- Arquitectura extensible mediante plug-ins para la incorporación de nuevos servicios.
- Soporte para hojas de cálculo Excel y Comma-Separated Values (CSV) (26).
- Acceso seguro a recursos en la Web.
- Servicios de empaquetado para permitir la configuración y compartición de conjuntos de servicios.
- Diseñador gráfico que permite el arrastre y colocación de componentes en un flujo de trabajo.
- Validación de flujos de trabajo durante la etapa de diseño, y aplicación de valores intermedios a los mismos para su depuración.
- Multiplataforma (programado en Java), código abierto y con licencia LGPL.

Taverna engloba un conjunto de componentes para el diseño, desarrollo y ejecución de flujos de trabajo científicos. Dicha aplicación está compuesto por:

- Taverna Engine: un motor de flujos de trabajo.
- Taverna Workbench: un cliente de escritorio para el diseño de flujos de trabajo.
- Taverna Server: un servidor para la ejecución de flujos de trabajo remotos.

- **Mobyle**

Mobyle (27) es una aplicación web para la integración de los programas bioinformáticos con las bases de datos existentes, que provee una amplia gama de servicios como son los servicios web, las ejecuciones remotas y los flujos de datos. El sistema Mobyle está distribuido bajo la licencia GNU GPLv2 y su servidor está sobre un conjunto de módulos Python que proporcionan su seguridad.

Capítulo 1: Fundamentación Teórica

En su interfaz de usuario cuenta con un buscador de programas, dándole una amplia descripción de estos. Al ejecutar uno de estos programas y mostrar los resultados que retorna, expone además los programas que se pueden ejecutar para su posterior análisis. Otra de las características es que permite la ejecución de los programas que están disponibles en diferentes servidores Mobyly en un único portal. Esta característica facilita el acceso de usuarios a los servicios físicamente distribuidos en diferentes servidores, dentro de un entorno integrado que permita reutilizar los datos. Por otra parte esta aplicación cuenta con un sistema llamado PlayMOBY el cual permite la publicación de los servicios web BioMoby.

Esta aplicación incorpora un motor de flujos de trabajo orientado a los flujos de datos, lo que permite el encadenamiento de tareas sucesivas o paralelas, pero a su vez no permite la edición de flujos de trabajo.

- **Galaxy**

Galaxy (28) es una aplicación web de propósitos bioinformáticos desarrollada en el lenguaje Python y de código abierto, la cual cuenta con un amplio conjunto de herramientas para el análisis de genomas. Esta aplicación permite el diseño o edición de flujos de trabajo, los cuales pueden ser guardados para su posterior reutilización o para compartirlos con otros usuarios del sistema, teniendo además esta aplicación, soporte para la gestión de usuarios.

Galaxy presenta una interfaz amigable y fácil de usar. Entre las características más significativas que presenta la aplicación, es que utiliza para la ejecución de los flujos de trabajo y de las herramientas disponibles las tecnologías *cluster* y *grid*. Otra de las características que presenta es el soporte de ficheros.

Entre las desventajas que presenta se encuentra que no utiliza servicios web para agregarle funcionalidades a la aplicación y además no cuenta con una manera de representar los flujos de trabajo en algún formato de marcado pues estos se guardan directamente en la base de datos de flujos de trabajo del usuario.

1.4 Selección del motor de flujos de trabajo

Todas las aplicaciones de flujos de trabajo analizadas en el epígrafe anterior cuentan con características propias, siendo estas las que determinan el uso o no por parte de los usuarios según sus necesidades.

Capítulo 1: Fundamentación Teórica

Algunas de las que fueron tomadas en cuenta para la selección del motor de flujos de trabajo a utilizar en el portal UCIBioSoft, son: la utilización de servicios web para agregar nuevas funcionalidades al flujo de trabajo; que posean una licencia libre y sea de código abierto para poder utilizar, modificar y distribuir sin restricciones; que permita el diseño de flujos de trabajo a los usuarios; y que genere un lenguaje de marcado a partir del diseño del flujo de trabajo.

Como se muestra en la Tabla 1 todas las aplicaciones son de software libre, permitiendo la utilización de las mismas sin restricciones. La mayoría de estas aplicaciones tienen propósitos bioinformáticos, excluyéndose Kepler que es de propósito general. Las aplicaciones que permiten el diseño de flujos de trabajo son Wildfire, Kepler, Taverna y Galaxy, excluyéndose Mobyly que no presenta esta característica. Además Wildfire, Kepler, Taverna y Mobyly, presentan la funcionalidad de adicionar servicios web. El lenguaje de marcado lo utilizan las aplicaciones de escritorio, Wildfire, Kepler y Taverna.

Tabla 1. Características de interés para el portal UCIBioSoft presentes en las diferentes aplicaciones de flujos de trabajo.

	Wildfire	Kepler	Taverna	Mobyly	Galaxy
Servicios web	-	X	X	X	-
Diseño de flujos de trabajo	X	X	X	-	X
Software libre	X	X	X	X	X
Lenguaje de marcado	X	X	X	-	-
Propósito bioinformáticos	X	-	X	X	X

Entre las aplicaciones de flujos de trabajo antes expuestas y los principales criterios que se tuvieron en cuenta para la selección del motor de flujos de trabajo, se optó por Taverna. De las características necesarias para el portal UCIBioSoft, Taverna es una aplicación que se distribuye bajo licencia libre y de propósito bioinformático, que permite el uso de servicios web y el diseño de flujos de trabajo a través de ontologías, y utiliza además, lenguaje de marcado. Es una potente herramienta en el campo de la

Capítulo 1: Fundamentación Teórica

bioinformática, entre las más utilizadas mundialmente por las facilidades y funcionalidades que presenta, existiendo abundante documentación sobre la misma. Aunque cabe destacar como desventajas, que no cuenta con una aplicación web y no posee gestión de usuarios.

No se seleccionaron ninguna de las aplicaciones restantes por las siguientes razones:

- Mobylye: no permite el diseño de flujos de trabajo y por lo tanto tampoco genera un lenguaje de marcado.
- Galaxy: no utiliza servicios y no genera a partir del diseño del flujo de trabajo, un lenguaje de marcado.
- Wildfire: no permite la utilización de servicios web.
- Kepler: no es una aplicación con fines bioinformáticos exclusivamente.

Conclusiones

Después de haber realizado un riguroso estudio del arte, de los principales conceptos concernientes al tema, las características del portal UCIBioSoft y las aplicaciones más importantes a nivel mundial capaces de realizar flujos de trabajo; se seleccionó la aplicación Taverna, como motor de flujos de trabajo idóneo, para utilizarlo en el portal UCIBioSoft. Se expusieron los argumentos que dieron lugar a dicha selección, cumpliendo así con el objetivo específico: selección de un motor de flujos de trabajo para ejecutar los mismos en el servidor del portal. Y con las tareas: revisión bibliográfica sobre los sistemas manejadores de flujos de trabajo y selección del motor para ejecutar los flujos de trabajo en el servidor del portal.

Capítulo 2: Materiales y Métodos

CAPÍTULO 2: MATERIALES Y MÉTODOS

Introducción

En el capítulo se abordará lo referente a las definiciones y características principales de los materiales y métodos que incluyen las herramientas y metodología propuestas. Los cuales se utilizarán en el módulo de gestión de flujos de trabajo del portal UCIBioSoft.

2.1 Materiales

2.1.1 Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés de Integrated Development Environment) es un entorno de programación, que proporciona al programador un conjunto de herramientas útiles tales como: un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Un IDE provee un marco de trabajo amigable e interactivo para uno o varios lenguajes de programación.

2.1.2 NetBeans

Netbeans (29) es un IDE libre, gratuito y de código abierto patrocinado por Sun Microsystems. Desarrollado principalmente para soportar lenguaje Java aunque en la actualidad soporta múltiples lenguajes. Se puede ejecutar en plataformas Unix y Windows, por lo que es un software multiplataforma.

Esta herramienta es un entorno de desarrollo para que los programadores puedan escribir, compilar, depurar y ejecutar programas, tanto aplicaciones de escritorio como aplicaciones web. Todas las funciones de este IDE son provistas por módulos, donde cada una proporciona una función bien definida para el desarrollo de aplicaciones Java. Por lo tanto, Netbeans es el IDE de desarrollo propuesto para programar el módulo de gestión de flujos de trabajo del portal UCIBioSoft.

2.1.3 Framework

Un *framework* (30) es una estructura conceptual y tecnológica bien definida, compuesta por componentes o artefactos personalizables e intercambiables en el cual un proyecto de software es organizado y

Capítulo 2: Materiales y Métodos

desarrollado. Un *framework* puede incluir un lenguaje interpretado, bibliotecas y soporte de programas para lograr con esto la unión de todos los componentes del proyecto.

Algunas de las ventajas de los *framework* es que son inmediatamente productivos durante el comienzo de su uso al estar enmarcados sobre un dominio dado, obteniendo el programador mayor agilidad en el trabajo. Permiten que los arquitectos se centren en la toma de decisiones imprescindibles, en vez de gastar tiempo en hacer cumplir las mejores prácticas y propician que se escriba menos código; reduciendo considerablemente el tiempo de desarrollo.

2.1.4 Open-Jacob Draw2D

Open-Jacob Draw2D (31) es una librería Java Script que permite crear gráficos y diagramas en la web. Es el componente de dibujo de Open-Jacob, un editor de flujos de trabajo que permite dibujar directamente en el navegador, sin necesidad de instalar ningún software adicional. No existe abundante documentación acerca de su uso, lo cual pudiera representar una desventaja; aunque cabe destacar que las herramientas similares analizadas en el estudio realizado, eran privativas.

2.1.5 Framework de Interfaz de usuario

- **Interfaz de usuario Yahoo**

Interfaz de usuario Yahoo (YUI por sus siglas en inglés Yahoo User Interface) es un *framework* o librería desarrollada por la compañía Yahoo. Esta librería escrita en JavaScript facilita el desarrollo de aplicaciones web del lado del cliente y vincula para esto diferentes elementos visuales e interactivos que incluyen CSS.

Entre las características de este *framework* se encuentran la utilización de diferentes tecnologías tales como el manejo del árbol de objetos DOM que incluye la colocación de elementos y estilos CSS, además de la utilización de Ajax, puede crear efectos de movimiento y objetos que se pueden arrastrar y soltar. Otra de sus características es, que está liberada bajo la licencia BSD la cual incluye la utilización y modificación incluso para aplicaciones comerciales.

- **Dojo**

Capítulo 2: Materiales y Métodos

Dojo (32) es un framework de código abierto construido en JavaScript y con soporte para Ajax que permite desarrollar aplicaciones web del lado del cliente mejorando la apariencia de las mismas. Entre sus características permite la manipulación de eventos y del árbol de objetos DOM. Proporciona un conjunto de API para la comunicación asíncrona entre el cliente y el servidor, además permite la validación del lado del cliente de los formularios, admite arrastrar y soltar objetos y crear gráficas de diversos tipos. Una característica que lo identifica de los demás frameworks, es que está compuesto por un conjunto de componentes reutilizables (widget) y presenta doble licencia, BSD y AFL.

- **ExtJs**

El framework ExtJs (33) comenzó por ser una extensión de la popular librería Interfaz YUI. Luego sus desarrolladores crearon una comunidad de código abierto, uniendo los conocimientos de varios programadores Web, al punto que en la actualidad es una de las librerías más poderosas para el desarrollo de interfaces de usuarios Web en todo el mundo. Entre las principales características de ExtJs, se encuentran:

- Provee funcionalidades sencillas de utilizar para hacer páginas con apariencia de sistemas de escritorio, que pueden incluir ventanas, tablas y vistas muy divertidas capaces de interactuar en diferentes navegadores.
- Interactúa con el usuario y el navegador, vía el “Manejador de Eventos”, respondiendo a las órdenes del usuario a través del teclado y el ratón.
- Capacidad de comunicación con el servidor paralelamente, sin necesidad de tener que refrescar la página, usando la tecnología Ajax.
- Permite crear y manipular DataGrids, teniendo ventajas en este aspecto con otros frameworks.

- **Propuesta de selección de framework para la interfaz de usuario**

De los frameworks propuestos a utilizar en el módulo de gestión de flujos de trabajo, se determinó utilizar YUI. Este framework presenta todas las cualidades para mejorar las interfaces de usuarios de las aplicaciones web, además es el framework por defecto del CMS Liferay. Fue utilizado para el desarrollo

Capítulo 2: Materiales y Métodos

del portal UCIBioSoft y sus módulos, por lo que el equipo de desarrollo posee conocimientos suficientes y está familiarizado con su uso.

2.1.6 Direct Web Remoting

Direct Web Remoting (34) (DWR) es una librería de código abierto que ayuda a incluir tecnología Ajax en aplicaciones web. Esta librería permite ejecutar código Java en un servidor de aplicaciones, como si este código estuviera en el navegador. DWR permite invocar objetos Java alojados en un servidor desde el JavaScript de un navegador como si fueran objetos locales. DWR además, proporciona integración con el framework Spring. Esta librería cuenta con dos partes principales:

- DWR genera dinámicamente el código JavaScript que actuará como proxy de las clases Java. Este código será el encargado de utilizar las clases Java del servidor como si fueran objetos del cliente.
- Java es fundamentalmente síncrono mientras que Ajax es asíncrono, así que cuando se invoca un método remoto se debe proporcionar una función JavaScript de callback que será invocada cuando llegue la respuesta. El siguiente diagrama muestra como DWR puede cambiar el contenido de un combo como resultado de un evento (por ejemplo, “onclick”).

Capítulo 2: Materiales y Métodos

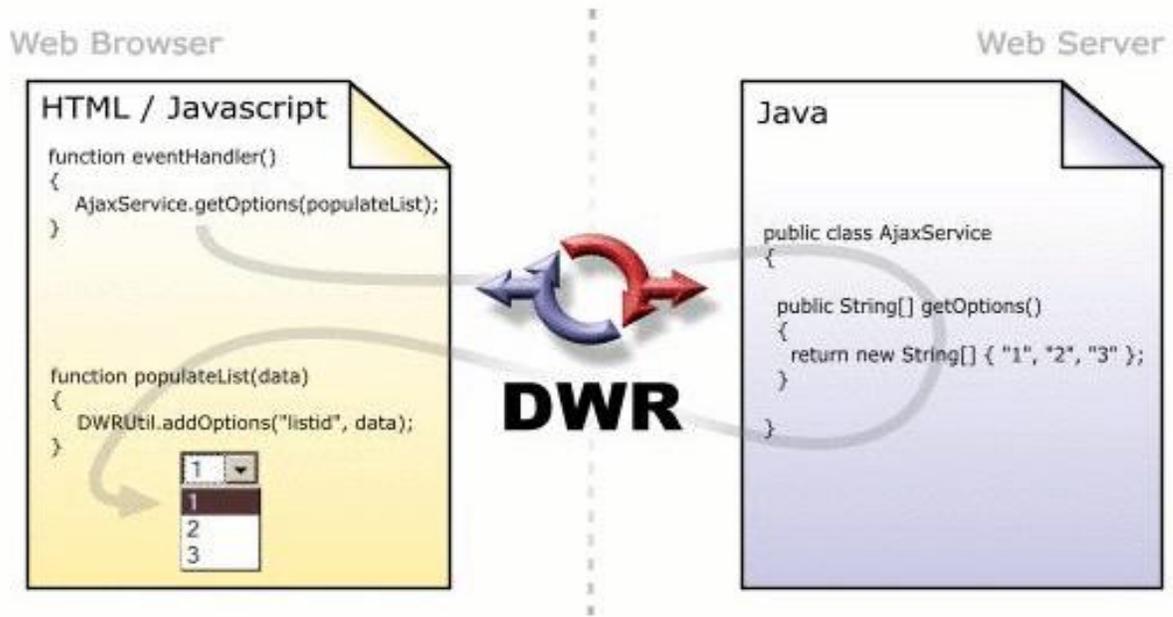


Figura 3. Esquema de funcionamiento de Direct Web Remoting.

En este caso, DWR genera dinámicamente una versión JavaScript de la clase Java `AjaxService`. Ésta es llamada por la función `eventHandler` y DWR gestiona toda la interacción remota con el servidor incluyendo la conversión de tipos de datos de los parámetros entre Java y JavaScript. Cuando llega la respuesta, se invoca la función callback `populateList` la cual utiliza una utilidad de DWR para actualizar el contenido del combo de la página web.

2.1.7 Spring

Spring (35) es un *framework* de código abierto para desarrollo de aplicaciones Java, existiendo una versión también para la plataforma .NET. Está basado en la programación orientada a aspectos proporcionando la reutilización de código y evitando la duplicación de este. Entre sus características principales se encuentra que, permite mejorar la práctica de programación reduciendo la utilización de los ficheros de configuración y facilita la utilización de objetos que usen EJB. Una ventaja de Spring es su modularidad, pudiendo usar algunos de los módulos sin comprometerse con el uso del resto. Entre sus

Capítulo 2: Materiales y Métodos

módulos para la web, se encuentra Spring Portlet MVC el cual se utilizará para el desarrollo del módulo de gestión de flujos de trabajo ya que proporciona los estándares JSP 168 y 286, y una arquitectura en capas.

2.1.8 Portlet

Un portlet (36) es un componente web gestionado por un contenedor que, tras la petición de un usuario, genera y presenta dinámicamente componentes de contenidos en la interfaz de usuario de un portal web. Estas aplicaciones web interactivas se integran en los portales añadiendo cada vez más funcionalidades. También permiten la personalización, presentación y gestión de la seguridad.

El contenido generado por los portlets se denomina fragmento. Los fragmentos agregados resultantes de la operación de varios portlets constituyen un documento que se traduce en la interfaz del portal.

La especificación de Portlets Java 1.0, Java Specification Request (JSR) 168, fue lanzado en octubre de 2003 proporcionando las normas capaces de construir portlets que se pueden ejecutar en los portales. La especificación Java Portlet consigue la interoperabilidad entre portlets y portales definiendo las API de portlets. En febrero de 2006 se comenzó a trabajar en la especificación de Java Portlets 2.0 (JSR 286), aprobándose la versión final en junio de 2008.

2.1.9 Sistema de Gestión de Contenidos

Sistema manejador de contenidos conocido por sus siglas en inglés como CMS (Content Management System), “es un software que se utiliza principalmente para facilitar la gestión de webs” (37) o sea, posibilita administrar el contenido de una web de forma automática, permitiendo publicar, editar, borrar, otorgar permisos de acceso o establecer los módulos visibles para los visitantes de la web. Aporta herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones.

Un CMS es una herramienta que permiten crear y mantener un web con facilidad, encargándose de los trabajos más tediosos que con anterioridad, ocupaban el tiempo de los administradores de las webs.

Capítulo 2: Materiales y Métodos

2.1.9 LifeRay

LifeRay (38) es un gestor de contenidos de código abierto escrito en Java. Este CMS tiene la capacidad de crear blogs, chats, foros de discusión y wikis, los cuales están gestionados por un sistema de roles y permisos con la capacidad de que cada usuario pueda gestionar sus propias páginas, a través de los portlets que desee.

El CMS LifeRay ofrece un entorno atractivo y fácilmente gestionable, que utiliza disímiles tecnologías web, estándares, lenguajes de script, soporte de varios idiomas, siendo independiente de plataforma y puede utilizar la mayoría de las bases de datos existentes.

2.2 Métodos

2.2.1 Metodología de desarrollo

“Una metodología de ingeniería de software es un proceso para la producción organizada del Software, empleando para ello una colección de técnicas predefinidas y convencionales en las notaciones. Una metodología se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada paso. Los pasos de la producción del software se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo” (39).

Las metodologías de desarrollo describen los pasos a seguir para desarrollar un producto de software, definen qué es lo que se debe hacer, cómo y quién debe hacerlo. Precisan el orden de las tareas, los artefactos que van a ser generados, quiénes son los responsables y cómo deben hacerse cada una de estas tareas durante el ciclo de desarrollo del proyecto. De esta manera, se obtiene un producto de alta calidad, desarrollado con rapidez y eficacia. La selección de la metodología a utilizar se hace en base a las características del equipo y las necesidades específicas de la situación.

Maddison, otro de los estudiosos del tema, define la metodología como el “conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información” (40).

Existen dos grandes grupos de metodologías de desarrollo de software, las robustas y las ágiles. A

Capítulo 2: Materiales y Métodos

continuación, una comparación entre ellas.

Tabla 2. Comparación entre metodologías ágiles y robustas.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Capítulo 2: Materiales y Métodos

2.2.2 Extreme Programming

Extreme Programming (XP) (41) es una metodología ágil para el desarrollo de software centrada en las relaciones interpersonales como clave para el éxito. Esta metodología promueve el trabajo en equipo, preocupándose por el aprendizaje de los programadores y manteniendo un buen clima de trabajo. XP se basa en la retroalimentación continua ente el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Se define además como, especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico.

XP define cuatro variables para sus proyectos de software: coste, tiempo, calidad y alcance. De estas cuatro variables, sólo tres de ellas podrán ser fijadas por las fuerzas externas al proyecto (clientes y jefes de proyecto), mientras que el valor de la cuarta variable será establecido por el equipo de desarrollo, en función de los valores de las otras tres.

Como muchas otras metodologías, XP cuenta con roles, los cuales definen quienes se encargan de realizar las actividades del proyecto. Siendo estos roles: programador, cliente, encargado de pruebas (Tester), encargado de seguimiento (Tracker), entrenador (Coach), consultor y gestor.

Para la realización de la propuesta del módulo de gestión de flujos de trabajo se utilizará la metodología XP. Se determinó la utilización de la misma, pues está centrada en la programación evitando la generación de gran cantidad de artefactos, además es propicia para desarrollar aplicaciones propensas a cambios, se ha utilizado para el desarrollo del portal UCIBioSoft y de los demás módulos con los que ya cuenta.

2.2.3 Propuesta de Arquitectura del módulo de gestión de flujos de trabajo en el portal UCIBioSoft

- **Prototipo general del portal UCIBioSoft**

El portal UCIBioSoft desarrollado con el CMS Liferay, cuenta con un conjunto de portlet. Cada uno de ellos responde a un módulo en específico, proporcionándole funcionalidades al portal. Por lo que se incorpora a este conjunto de portlet, como se percibe en la Figura 3, el módulo de gestión de flujos de trabajo. El cual responde a los estándares JSR 168 y JSR 286 como el resto de los módulos y se ajusta al patrón de

Capítulo 2: Materiales y Métodos

diseño MVC, proporcionado por el framework Spring Portlet MVC.

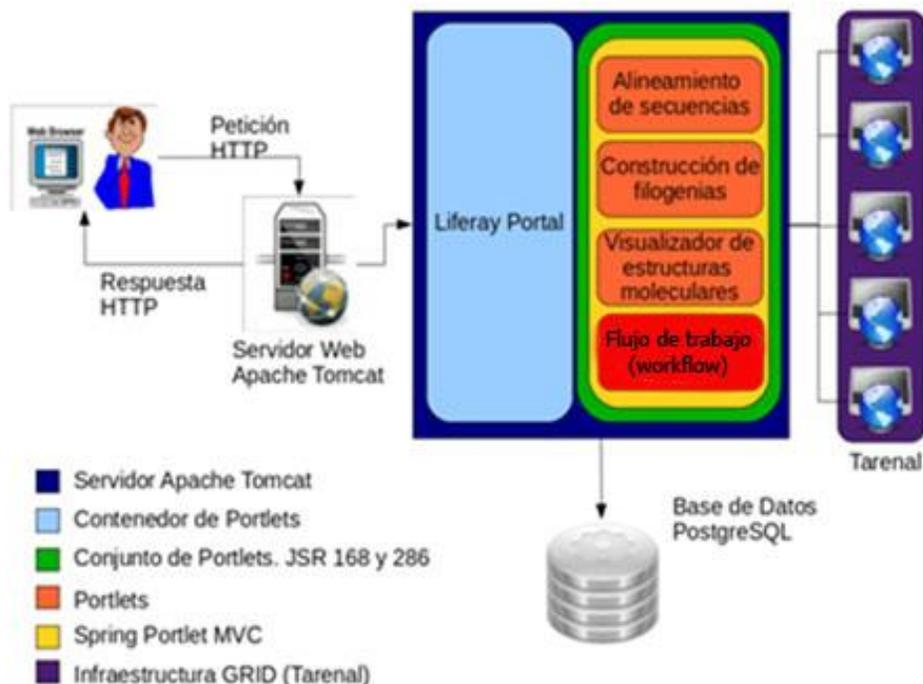


Figura 3. Módulo de gestión de flujos de trabajo en el prototipo general del portal UCIBioSoft.

- **Representación arquitectónica del módulo de gestión de flujos de trabajo**

En la Figura 4 se muestra la propuesta de representación arquitectónica del módulo de gestión de flujos de trabajo. En esta propuesta de diseño los usuarios acceden al portlet a través de sus perfiles de usuario. Estos pueden editar los flujos de trabajo en la interfaz gestor de flujos de trabajo o accediendo a la base de datos con el objetivo de cargar en el gestor, los flujos de trabajo anteriormente guardados. Después de editados los flujos de trabajo, se puede guardar dicha edición en la base de datos o puede ejecutar los flujos de trabajo utilizando el motor de la aplicación manejadora de flujos de trabajo Taverna, devolviendo los resultados intermedios y finales.

Capítulo 2: Materiales y Métodos

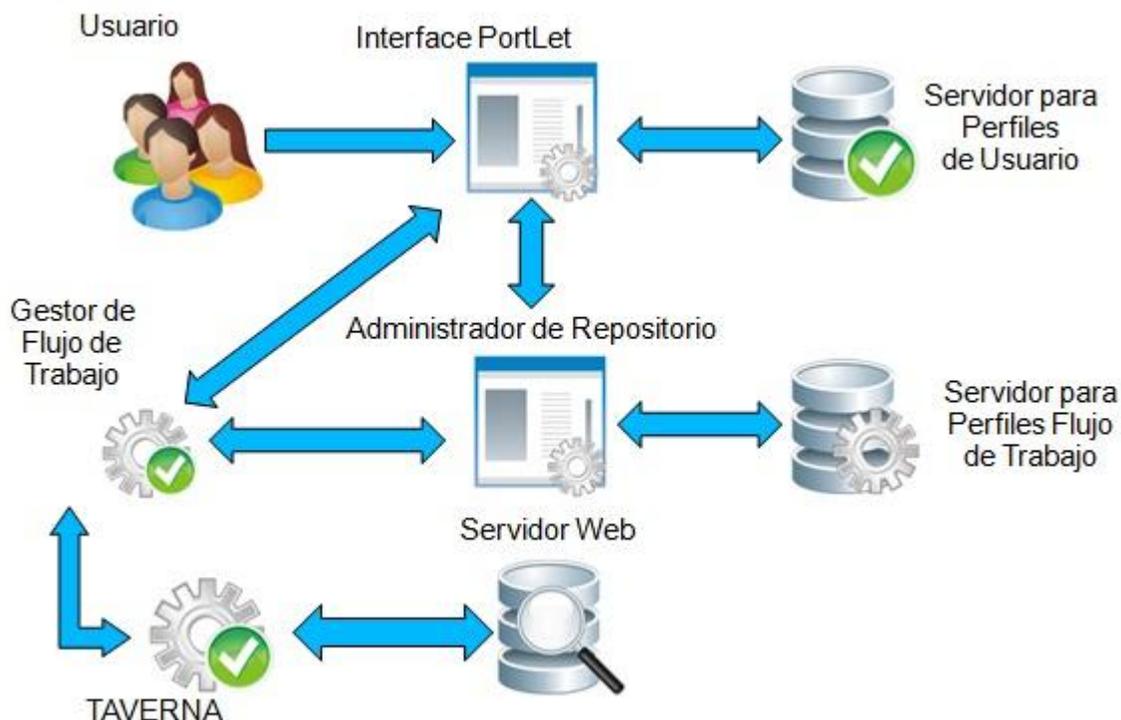


Figura 4. Representación arquitectónica del sistema de gestión de flujos de trabajo.

La representación arquitectónica a seguir es el patrón Modelo-Vista-Controlador pues proporciona que se separe la lógica del negocio de la interfaz de usuario, permitiendo el desarrollo por separado de cada una de estas y propiciando la reutilización y la flexibilidad de la aplicación. En la vista se encuentran las interfaces de usuario, las cuales según la propuesta, serán realizadas con el framework YUI proporcionándole a la aplicación un entorno agradable y Open Jacob Draw2D se utilizará para el diseño de los flujos de trabajo. En el modelo se encuentran las clases pertenecientes al motor de flujos de trabajo de Taverna y las clases de acceso a la base de datos de perfiles de usuarios y de flujos de trabajo. La capa controladora como intermediaria entre las capas de presentación y modelo, proporciona la comunicación entre estas últimas, respondiendo a eventos o acciones del usuario e invocando peticiones al modelo.

La acción de ejecución de flujos de trabajo, comienza al crear la secuencia de tareas como un objeto DataFlow. Este contiene una lista de Processor que representan cada una de las actividades a realizar

Capítulo 2: Materiales y Métodos

durante su ejecución. El Dataflow que no es más que el flujo del trabajo que, al ejecutarlo se envía a la clase WorkflowInstanceFacadeImpl y esta se encarga de ejecutar cada una de las acciones definidas en los Processors y devuelve los valores intermedios y finales en un arreglo de tipo ResultListener.

Conclusiones

Como resultado de la investigación en el capítulo se seleccionaron las herramientas y la metodología a utilizar durante el desarrollo de la propuesta del módulo de gestión de flujos de trabajo para el portal UCIBioSoft. Seleccionando como herramientas IDE, NetBeans, para el modelado la herramienta CASE Visual Paradigm, el CMS Liferay para vincularle el portlet módulo de gestión de flujos de trabajo al portal UCIBioSoft, el framework YUI para realizar la interfaz de usuario y Open-Jacob Draw2D para crear los diseños de flujos de trabajo en el portal.

Entre los métodos a utilizar para el desarrollo de la propuesta del módulo de gestión de flujos de trabajo en el portal, se seleccionó como guía para el desarrollo la metodología XP y se propuso la arquitectura conceptual del módulo.

De esta manera se le cumplió el objetivo específico: definir herramientas y metodologías necesarias para la integración del motor de flujos de trabajo en la propuesta. Y a la tarea: selección de las herramientas para la integración del motor de flujos de trabajo al portal UCIBioSoft.

Capítulo 3: Validación

CAPÍTULO 3: VALIDACIÓN

Introducción

A veces se hace necesario tener el criterio de otras personas con el conocimiento necesario para saber si lo investigado es realmente cercano a la realidad. En el presente capítulo se realiza la validación de la propuesta de integración de un motor de flujos de trabajo para el portal uciBioSoft. Con este objetivo se utilizó el método de consulta a especialistas (Delphi), el cual consiste en la selección de varios especialistas, que den su criterio sobre la utilidad de la propuesta presentada.

3.1 Método Delphi

El método Delphi es un procedimiento eficaz y sistemático que tiene como objeto la recopilación de opiniones de expertos sobre un tema particular con el fin de incorporar dichos juicios en la configuración de un cuestionario y conseguir un consenso a través de la convergencia de las opiniones de expertos.

El primer estudio de Delphi fue realizado en 1950 por la Rand Corporation para la fuerza aérea de Estados Unidos, y se le dio el nombre de Proyecto Delphi. El método Delphi utiliza como fuente de información un grupo de personas a las que se supone un conocimiento elevado de la materia que se va a tratar.

La calidad de los resultados depende:

- De la elaboración de los cuestionarios.
- La elección de los expertos consultados.

Es uno de los métodos de pronosticación más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de un grupo de expertos en el tema tratado. Permite rebasar el marco de las condiciones actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos que pueden estar basadas en un análisis lógico, como en su experiencia intuitiva.

Capítulo 3: Validación

El método presenta 4 características principales:

- **Anonimato:** Ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:
 - Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
 - Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
 - El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- **Iteración y realimentación controlada:** La iteración se consigue al presentar varias veces el mismo cuestionario. Como además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.
- **Respuesta del grupo en forma estadística:** La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.
- **Heterogeneidad:** Pueden participar expertos de determinadas ramas de actividad sobre las mismas bases.

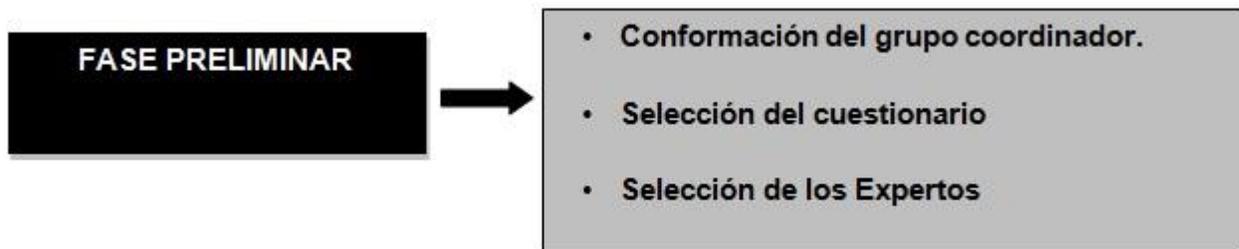
Algunas de las ventajas que presenta el método Delphi son:

- Permite obtener información de puntos de vista sobre temas muy amplios o muy específicos. Los ejercicios Delphi son considerados “holísticos”, cubriendo una variedad muy amplia de campos.
- El horizonte de análisis puede ser variado.
- Permite la participación de un gran número de personas, sin que se forme el caos.

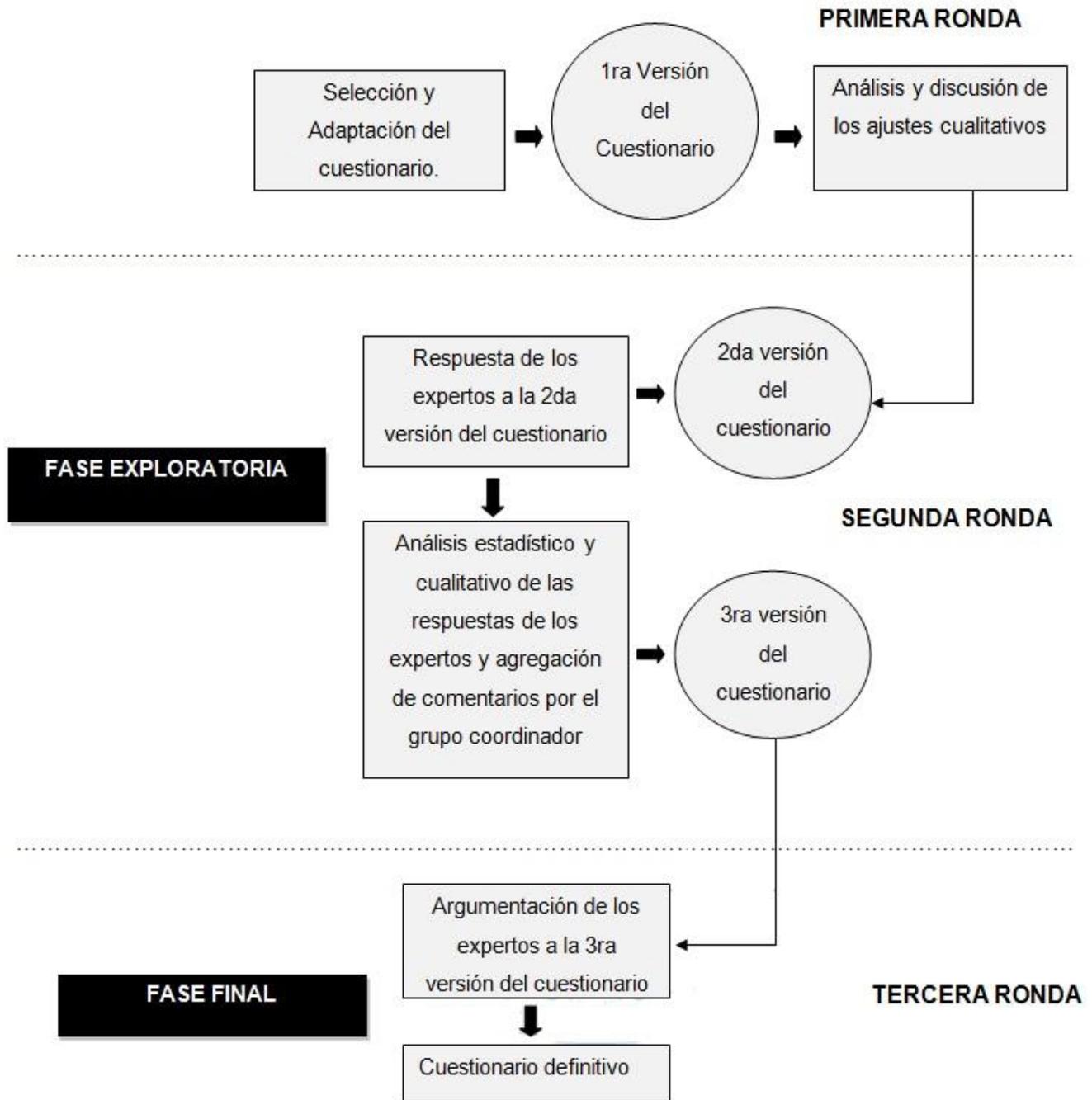
Capítulo 3: Validación

- Ayuda a explorar de forma sistemática y objetiva problemas que requieren la concurrencia y opinión cualificada.
- Elimina o aminora los efectos negativos de las reuniones de grupo “Cara-Cara”.

Debido a lo anterior es que se ha decidido el uso del método Delphi. Para aplicar el método se siguieron tres etapas fundamentales.



Capítulo 3: Validación



Capítulo 3: Validación

3.2.1 Elección de Expertos

Se definen como criterios de selección de expertos:

- Graduado del Nivel Superior.
- Vinculación al desarrollo de proyectos productivos.
- Arquitecto de aplicaciones web
- Conocimientos sobre aplicaciones manejadoras de flujos de trabajo.

Para la selección de los expertos finales se hace necesario primeramente conocer el grado de conocimiento del experto en cuestión, la misma se realiza con la ayuda del Coeficiente de competencia. Este coeficiente se determina mediante la fórmula: $K = \frac{1}{2} (k_c + k_a)$, donde k_c es el coeficiente de conocimientos y k_a es el coeficiente de argumentación. El coeficiente de conocimientos se obtiene de la siguiente tabla que recoge una autoevaluación del posible experto.

Tabla 3. Grado de experiencia del posible experto.

								X		
0	1	2	3	4	5	6	7	8	9	10

El experto marcará en la casilla enumerada, según su criterio acerca de la capacidad que él tiene sobre el tema que se la ha sometido a su consideración, en una escala del 0 al 10 y que después para ajustarla a la teoría de las probabilidades se multiplicará por 0,1; de esta forma, la evaluación "0" indica que el experto no tiene absolutamente ningún conocimiento de la problemática correspondiente, mientras que la evaluación "10" significa que el experto tiene pleno conocimiento de la problemática tratada. Entre estas dos evaluaciones extremas hay nueve intermedias. En la tabla considerada, $k_c = 0.8$.

Para calcular el **coeficiente de argumentación** se procede de la siguiente forma:

Capítulo 3: Validación

Tabla 4. Fuentes de Argumentación.

Fuentes de Argumentación	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (Alto)	M (Medio)	B (Bajo)
Experiencia.			
Trabajos nacionales.			
Trabajos internacionales.			
Conocimiento del tema.			

En esta tabla el experto debe marcar, según su criterio, su grado de competencia sobre los aspectos sometidos a consideración. Las marcas de los expertos se traducen a puntos, según la siguiente escala:

Tabla 5. Cálculo del coeficiente de argumentación.

Fuentes de Argumentación	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (Alto)	M (Medio)	B (Bajo)
Experiencia.	0.8	0.6	0.3
Trabajos nacionales.	0.06	0.06	0.06
Trabajos internacionales.	0.07	0.07	0.07

Capítulo 3: Validación

Conocimiento del tema.	0.07	0.07	0.07
Totales	1.0	0.8	0.5

Con estos elementos es suficiente para obtener el coeficiente de competencia K. En la siguiente tabla se muestra la selección de uno de los expertos encuestados.

Tabla 6. Ejemplo de selección de uno de los expertos encuestados.

	A (Alto)	M (Medio)	B (Bajo)
Experiencia.		x	
Trabajos nacionales.		x	
Trabajos internacionales.		x	
Conocimiento del tema.		x	

Entonces $k_a = 0.6 + 0.06 + 0.07 + 0.07 = 0.8$

El código para la interpretación de tales coeficientes de competencia es el siguiente:

Si $0.8 < k < 1.0$, el coeficiente de competencia es alto.

Si $0.5 < k < 0.8$, el coeficiente de competencia es medio.

Si $k < 0.5$ el coeficiente de competencia es bajo.

Observación: Como a la categoría de “bajo” se le otorgaron puntos, siempre el coeficiente de competencia quedará comprendido entre:

Capítulo 3: Validación

$$\frac{0 + 0.5}{2} \leq K \leq \frac{1 + 1}{2} \Leftrightarrow 0.25 \leq K \leq 1$$

Como puede apreciarse el coeficiente de competencia del experto analizado es medio. La forma descrita con anterioridad nos permite seleccionar la competencia de nuestros expertos. Se lograron seleccionar de acuerdo a este análisis realizado un total de 7 expertos. En la siguiente figura se representa el resultado de acuerdo al coeficiente de competencia en general:

Coeficiente de Competencia

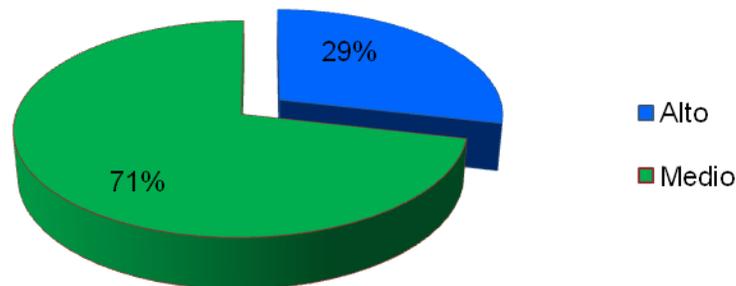


Figura 4. Coeficiente de competencia.

3.2.2 Desarrollo práctico y explotación de resultados

A continuación se ilustran 10 elementos sobre la propuesta de solución de integración del módulo de gestión de flujos de trabajo al portal UCIBiosoft realizada a 7 expertos. Se pidió que evaluaran los elementos en las categorías de Si, No, Tal vez. Se confeccionan tablas para ir recogiendo los resultados aportados por los expertos. Para ello es necesario auxiliarse del programa Microsoft Excel 2007. Los resultados se recogen en una tabla de doble entrada como la siguiente:

Capítulo 3: Validación

Tabla 7. Encuesta a los expertos.

No	Elementos	Si	Tal vez	No
1	¿Cree usted que la propuesta del módulo de gestión de flujos de trabajo satisface las necesidades del portal UCIBiosoft?			
2	¿Cree Ud., que dentro de la solución propuesta, la aplicación Taverna facilita la gestión de los flujos de trabajo?			
3	Partiendo de que la creación del módulo no genera gran cantidad de información, que el equipo de desarrollo es pequeño, estable y cuenta con la presencia del cliente, ¿sería factible guiar todo el proceso de desarrollo con una metodología ágil?			
4	Conociendo que XP es una metodología ágil, flexible ante cambios y dirigida fundamentalmente a la programación. ¿Sería factible la utilización de la misma como guía para el desarrollo del módulo?			
5	Taverna es una aplicación manejadora de flujo de trabajo de propósito bioinformático realizada en el lenguaje de programación Java, que permite el uso de servicios web y el diseño de flujos de trabajo a través de ontologías. ¿Cree			

Capítulo 3: Validación

	que este motor sea el adecuado para la ejecución de los flujos de trabajo en el portal?			
6	El framework de presentación YUI, de licencia BSD, que utiliza diferentes tecnologías como árbol de objeto DOM y Ajax e incluye la colocación de elementos y estilos CSS. ¿Sería acertado la utilización de YUI en la implementación del módulo de gestión de flujos de trabajo?			
7	Conociendo que en la actualidad el IDE NetBeans es de código abierto, gratuito, multiplataforma y permite la implementación de aplicaciones web. ¿Sería factible el uso del mismo como IDE para el desarrollo del módulo?			
8	Teniendo en cuenta que los usuarios finales de la aplicación son profesionales en la rama de la bioinformática y no necesariamente tienen experiencias en el uso de aplicaciones de gestión de flujos de trabajo. ¿Considera que la propuesta de solución permite al usuario gestionar los flujos de trabajo de una manera simple e interactiva?			
9	¿Considera que la arquitectura definida presenta la robustez necesaria para soportar la gestión de flujos de trabajo?			
10	Si lo considera necesario proponga otras			

Capítulo 3: Validación

	vías de solución.	
--	-------------------	--

Luego de realizar la encuesta a los expertos seleccionados se muestran los resultados en la siguiente tabla, donde A_i corresponde a las preguntas de la encuesta y C_i a las posibles respuestas.

C1: Si C2: Tal vez C3: No

Tabla 8. Frecuencias acumuladas.

Tabla de Frecuencias Acumuladas					
No	Elementos	C1	C2	C3	Total
1	A1	7	0	0	7
2	A2	6	1	0	7
3	A3	7	0	0	7
4	A4	6	1	0	7
5	A5	7	0	0	7
6	A6	1	5	1	7
7	A7	5	2	0	7
8	A8	5	2	0	7
9	A9	4	3	0	7

La tabla de frecuencias acumuladas quedaría representada gráficamente como lo demuestra la figura:

Capítulo 3: Validación

Frecuencias Acumuladas

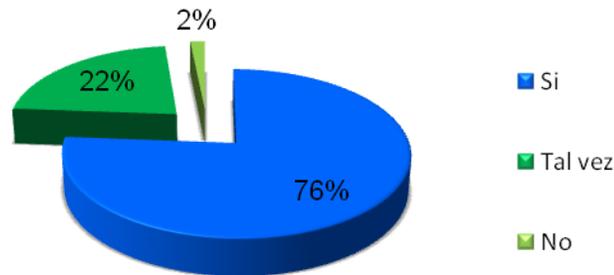


Figura 5. Frecuencias acumuladas.

Tabulados los datos, se realizan los siguientes pasos para obtener los resultados deseados:

Primer paso: Se construye una tabla de frecuencias acumuladas. Esto es, cada número en la fila, excepto el primero se obtiene sumándole el anterior.

Tabla 9. Frecuencias absolutas Acumuladas.

Tabla de Frecuencias Absolutas Acumuladas				
No	Elementos	C1	C2	C3
1	A1	7	7	7
2	A2	6	7	7
3	A3	7	7	7
4	A4	6	7	7
5	A5	7	7	7
6	A6	1	6	7
7	A7	5	7	7
8	A8	5	7	7
9	A9	4	7	7

Segundo paso: Se copia la tabla anterior y se borran los resultados numéricos. Ahora, en esta nueva tabla, se construye la tabla de frecuencias relativas acumulativas. Esta tabla se logra dividiendo por 7 (número total de expertos) cada uno de los números de la tabla anterior.

Capítulo 3: Validación

Tabla 10. Frecuencias Relativas Acumulativas.

Tabla de Frecuencias Relativas Acumulativas				
No	Elementos	C1	C2	C3
1	A1	0,9999	0,9999	0,9999
2	A2	0,85714286	0,9999	0,9999
3	A3	0,9999	0,9999	0,9999
4	A4	0,85714286	0,9999	0,9999
5	A5	0,9999	0,9999	0,9999
6	A6	0,14285714	0,85714286	0,9999
7	A7	0,71428571	0,9999	0,9999
8	A8	0,71428571	0,9999	0,9999
9	A9	0,57142857	0,9999	0,9999

Tercer paso: Buscar las imágenes de los elementos de la tabla anterior por medio de la función (Dist. Normal. Standard Inv). La siguiente tabla se muestra con los resultados obtenidos de los pasos anteriores, donde se agregan tres nuevas columnas y una fila para colocar los valores de la suma de las columnas y de las filas (**Suma**); el promedio de las filas (**P**); el valor de **N** (se obtiene al dividir la suma de las sumas entre 27, este 27 se ha obtenido de multiplicar el número de categorías (3) por el número de preguntas); el valor **N-P** da el valor promedio que otorgan los expertos consultados a cada pregunta.

Tabla 11: Puntos de corte

Puntos de Corte						N =	2,88
No	Elementos	C1	C2	C3	Suma	P	N-P
1	A1	3,72	3,72	3,72	11,16	3,72	-0,84
2	A2	1,07	3,72	3,72	8,51	2,84	0,05
3	A3	3,72	3,72	3,72	11,16	3,72	-0,84
4	A4	1,07	3,72	3,72	8,51	2,84	0,05
5	A5	3,72	3,72	3,72	11,16	3,72	-0,84
6	A6	-1,07	1,07	3,72	3,72	1,24	1,64
7	A7	0,57	3,72	3,72	8,00	2,67	0,21
8	A8	0,57	3,72	3,72	8,00	2,67	0,21
9	A9	0,18	3,72	3,72	7,62	2,54	0,34
Suma		13,54	30,82	33,47	77,83		
Puntos de Corte		1,50	3,42	3,72			

Capítulo 3: Validación

Las sumas obtenidas en las tres primeras columnas nos dan los puntos de cortes: 1.50, 3.42, 3.72. Los puntos de corte nos sirven para determinar la categoría o grado de adecuación de la propuesta según la opinión de los expertos consultados. La tabla que se muestra a continuación representa los puntos de cortes por las preguntas realizadas a los expertos, evidenciando en qué rango se encuentra la pregunta con respecto al punto de corte para determinar “sí” son factibles, “no” son factibles o “tal vez” es factible las propuestas realizadas.

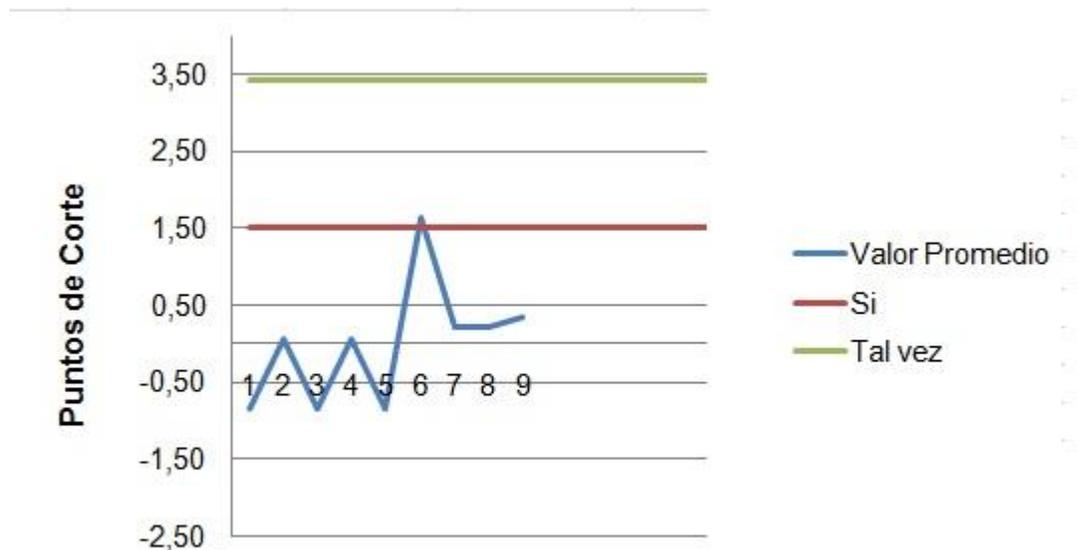


Figura 6: Gráfica de puntos de corte.

Capítulo 3: Validación

Las preguntas que resultarían “si” serían las que poseen valores menores que 1.50; los valores “tal vez” entre (1.51, 3.42); y con más de 3.42 sería “no”. De acuerdo a los valores de N-P se tiene: Menos 1.50; (1.51, 3.42); más de 3.42.

Se obtiene finalmente que:

Afirmaciones	Categoría
1	Si
2	Si
3	Si
4	Si
5	Tal vez
6	Si
7	Si
8	Si
9	Si

De acuerdo a los resultados obtenidos mostrados en la tabla anterior se puede dar por concluida la validación en cuanto a su elaboración teórica, ya que los resultados arrojados fueron satisfactorios. Se presenta a continuación un resumen con los resultados obtenidos.

Capítulo 3: Validación

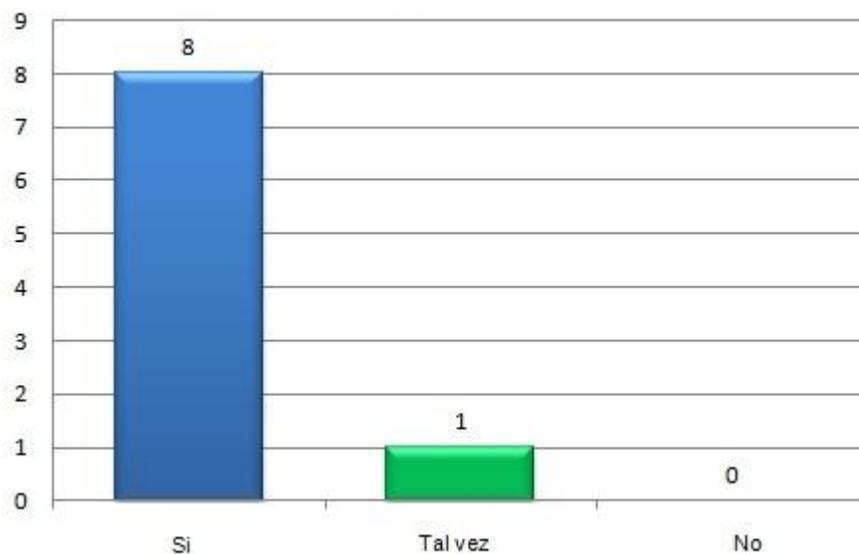


Figura 7: Resultados generales de la encuesta.

Conclusiones

En este capítulo se validó la solución propuesta para desarrollar el módulo de gestión de flujos de trabajo para el portal UCIBioSoft, usando para esto el método de validación Delphi. Se seleccionaron 7 expertos y se validaron 9 preguntas relacionadas con la propuesta planteada, obteniendo resultados satisfactorios evaluados en “Si” es factible y “Tal vez” es factible. Cumpliendo de esta manera con el objetivo específico y la tarea Validar la propuesta a partir del método Delphi.

Capítulo 3: Validación

Conclusiones

CONCLUSIONES

Al finalizar el presente trabajo de diploma se arriba a las siguientes conclusiones:

1. Se seleccionó Taverna como motor de flujos de trabajo para el portal UCIBioSoft.
2. Para la integración del motor de flujos de trabajo en la propuesta, se definieron; como *framework* de interfaz de usuario: YUI, Liferay como CMS, NetBeans fue escogido como IDE de desarrollo, como librería para crear gráficos y diagramas en la web fue seleccionado Open-Jacob Draw2D y Spring como *framework* de desarrollo de aplicaciones; guiados por la metodología XP.
3. La propuesta fue satisfactoriamente validada a partir del método Delphi.

Conclusiones

Recomendaciones

RECOMENDACIONES

- Aplicar las experiencias obtenidas en el desarrollo de investigaciones similares.
- Realizar la implementación de la propuesta realizada.

Glosario de términos

GLOSARIO DE TÉRMINOS

API: Una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

ERP: Conjunto de sistemas de información gerencial que permite la integración de ciertas operaciones de una empresa, especialmente las que tienen que ver con la producción, la logística, el inventario, los envíos y la contabilidad.

Java: Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

JSP: JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

MVC: (Model View Controller, MVC por sus siglas en inglés) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

Ontología: Estructura jerárquica que define formalmente las relaciones semánticas de un conjunto de conceptos. Se usa para crear vocabularios controlados/estructurados para la recuperación o el intercambio de información.

Servicio Web: Es una colección de protocolos y estándares, cuya función es el intercambio de datos entre aplicaciones a través de Internet. La interacción de aplicaciones de distintos sistemas operativos permite que programas de muy diversa concepción se combinen y proporcionen servicios integrados. Ésta es la nueva Internet, centrada en las aplicaciones y basada en los servicios.

Glosario de términos

Servlets: Los servlets son objetos que corren dentro del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones (ej: OC4J Oracle), que, además de contenedor para servlet, tendrá contenedor para objetos más avanzados, como son los EJB (Tomcat sólo es un contenedor de servlets).

Widget: Componente gráfico con el cual el usuario interactúa ejemplo, una ventana, una barra de tareas o una caja de texto.

BIBLIOGRAFÍA

1. **Rolando Gutiérrez Rosales, Dayana Cabrera Fleites.** Adaptación de OpenUp/Basic para el Polo Productivo de BioInformática. [En línea] 2009. [Citado el: 9 de 2 de 2011.] http://bibliodoc.uci.cu/TD/TD_2534_09.pdf.
2. **Guy R. Cochrane, Michael Y. Galperin.** Nucleic Acids Research. [En línea] [Citado el: 10 de 2 de 2011.] http://nar.oxfordjournals.org/content/38/suppl_1/D1.full.
3. **Michelle D. Brazas, Joseph T. Yamada, B. F. Francis Ouellette.** Nucleic Acids Research. [En línea] [Citado el: 10 de 2 de 2011.] http://nar.oxfordjournals.org/content/38/suppl_2/W3.abstract.
4. **Esmeralda Ramos, Haydemar Núñez, Roberto Casañas.** Esquema para evaluar ontologías únicas para un dominio de conocimiento. [En línea] Enero-Abril de 2009. [Citado el: 11 de 2 de 2011.] <http://www.scielo.org.ve/pdf/enl/v6n1/art05.pdf>.
5. ncbi. [En línea] [Citado el: 16 de 2 de 2011.] <http://www.ncbi.nlm.nih.gov/pubmed/7984417>.
6. ncbi. [En línea] [Citado el: 29 de 05 de 2011.] <http://www.ncbi.nlm.nih.gov/pubmed/10964570>.
7. **Edgar, Robert C.** Nucleic Acids Research. [En línea] [Citado el: 29 de 05 de 2011.] <http://nar.oxfordjournals.org/content/32/5/1792.full>.
8. **Maikel Zúñiga Suárez, Keiler Arnaldo González Torres.** Integración de la plataforma Tarenal con el middleware Globus Toolkit 4. [En línea] 6 de 2008. [Citado el: 29 de 05 de 2011.] http://bibliodoc.uci.cu/TD/TD_1743_08.pdf.
9. Oracle Sun Developer Network. [En línea] [Citado el: 8 de 6 de 2011.] <http://java.sun.com/products/jsp/tomcat/>.
10. Pixelware. [En línea] [Citado el: 12 de 2 de 2011.] <http://www.pixelware.com/workflow-flujo-trabajo.htm>.
11. Workflow Management Coalition. [En línea] [Citado el: 16 de 2 de 2011.] <http://www.wfmc.org/>.

Bibliografía

12. **Chiesa, Florencia.** METODOLOGÍA PARA SELECCIÓN DE SISTEMAS ERP. [En línea] [Citado el: 16 de 2 de 2011.] <http://www.ucla.edu.ve/dac/departamentos/informatica-II/metodologia-para-seleccion-de-sistemas-erp.PDF>.
13. Workflow Management Coalition. [En línea] [Citado el: 19 de 2 de 2011.] <http://www.wfmc.org/reference-model.html>.
14. Wildfire. [En línea] [Citado el: 12 de 2 de 2011.] <http://wildfire.bii.a-star.edu.sg/index.php>.
15. Massachusetts Institute of Technology . [En línea] 2005. [Citado el: 16 de 2 de 2011.] <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>.
16. **J.L. Vázquez-Poletti, Eduardo Huedo Cuesta, Rubén Santiago Montero, Ignacio Martín Llorente.** Centro de Estudios Superiores Felipe II. [En línea] [Citado el: 17 de 2 de 2011.] <http://www.cesfelipesecondo.com/revista/Articulos2004b/Articulo4.pdf>.
17. GNU Operating System. [En línea] [Citado el: 17 de 2 de 2011.] <http://www.gnu.org/licenses/licenses.es.html>.
18. Kepler. [En línea] [Citado el: 12 de 2 de 2011.] <https://kepler-project.org/>.
19. **Albert Osiris Sofia, Sandra Isabel Casas.** Survey de Tecnologías Grid . [En línea] [Citado el: 17 de 2 de 2011.] <http://www.cyta.com.ar/ta0704/v7n4a1.htm>.
20. EMBL-EBI. [En línea] [Citado el: 17 de 2 de 2011.] <http://www.ebi.ac.uk/Tools/webservices/soaplab/overview>.
21. Principio Legal. [En línea] [Citado el: 17 de 2 de 2011.] <http://www.principiolegal.com/licencias.php>.
22. Taverna. [En línea] [Citado el: 12 de 2 de 2011.] <http://www.taverna.org.uk/>.
23. EMBL Nucleotide Sequence Database. [En línea] [Citado el: 13 de 2 de 2011.] <http://www.ebi.ac.uk/embl/>.
24. ExPASy Proteomics Server. [En línea] [Citado el: 16 de 2 de 2011.] <http://expasy.org/sprot/>.

Bibliografía

25. EMBL. [En línea] [Citado el: 16 de 2 de 2011.] <http://www.ebi.ac.uk/imgt/hla/blast.html>.
26. ASERCA. [En línea] [Citado el: 16 de 2 de 2011.] http://www.aserca.gob.mx/artman/publish/article_1471.asp.
27. **Bertrand Néron, Hervé Ménager, Corinne Maufrais, Nicolas Joly, Julien Maupetit, Sébastien Letort, Sébastien Carrere, Pierre Tuffery, Catherine Letondal.** Mobyte: a new full web bioinformatics framework. [En línea] 30 de 6 de 2009. [Citado el: 18 de 2 de 2011.] <http://bioinformatics.oxfordjournals.org/content/25/22/3005.full.pdf>.
28. bitbucket. [En línea] 18 de 01 de 2011. [Citado el: 20 de 02 de 2011.] <https://bitbucket.org/galaxy/galaxy-central/wiki/Home>.
29. Netbeans. [En línea] [Citado el: 12 de 2 de 2011.] <http://netbeans.org/community/releases/69/>.
30. **Leynier Viquillón Lavorí, Pedro Luis Rojas Lemus.** Framework para la generación de interfaces gráficas para la plataforma educativa Dolphin. [En línea] 2 de 2011. [Citado el: 13 de 2 de 2011.] http://bibliodoc.uci.cu/TD/TD_02916_10.pdf.
31. **Fernández, Alma.** web master libre. [En línea] 23 de 11 de 2006. [Citado el: 13 de 2 de 2011.] <http://www.webmasterlibre.com/2006/11/23/open-jacob-draw2d-javascript-para-crear-graficos-y-diagramas/>.
32. Dojo toolkit. [En línea] [Citado el: 8 de 6 de 2011.] <http://dojotoolkit.org/>.
33. Sencha. [En línea] [Citado el: 8 de 6 de 2011.] <http://www.sencha.com/products/extjs/>.
34. Junta de Andalucía. [En línea] [Citado el: 15 de 2 de 2011.] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/DWR>.
35. **Johnson, Rod.** SpringSource Community. [En línea] 2010. [Citado el: 8 de 6 de 2011.] <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/index.html>.
36. **Gothe, Deepak.** Oracle Sun Developer Network. [En línea] Enero de 2010. [Citado el: 15 de 6 de 2011.] <http://developers.sun.com/portalservlet/reference/techart/jsr286/jsr286.html>.

Bibliografía

37. **Cuerda, Xavier García.** Mosaic. [En línea] 29 de 11 de 2004. [Citado el: 19 de 02 de 2011.] <http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/>.
38. Liferay. [En línea] [Citado el: 13 de 2 de 2011.] <http://www.liferay.com/es/products/liferay-portal/overview>.
39. **Jacobson, Ivar, Rumbaugh, James y Booch, Grady.** *El lenguaje Unificado de Modelado. Manual de Referencia.(Segunda Edición)*. Madrid : Addison Wesley, 2007.
40. **VALIENTE, Martín Alejo.** MODELO DE PROTOCOLO PARA ANÁLISIS Y EVALUACIÓN DEL SOFTWARE DESARROLLADO EN LAS PYMES ARGENTINAS. [En línea] 2008. [Citado el: 29 de 15 de 2011.] <http://posgrado.frba.utn.edu.ar/investigacion/tesis/MIS-2009-Valiente.pdf>.
41. **Maite Rodríguez Corbea, Meylin Ordóñez Pérez.** LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO EN CUBA. [En línea] 6 de 2007. [Citado el: 29 de 5 de 2011.] http://bibliodoc.uci.cu/TD/TD_0837_07.pdf.