MATERIAL MARKET

TFG GRADO INFORMÁTICA - ÁREA JAVA J2EE

ESTUDIANTE: ANTONIO MARTÍN MORENO

CONSULTOR: ALBERT GRAU

FECHA 12 DE MARZO 2014

1

INTRODUCCIÓN

Apartados:

- Introducción
- Síntesis
- Funcionalidad básica

INTRODUCCIÓN

Este documento es el punto de partida del Trabajo de Fin de Grado de los estudios oficiales del Grado de Ingeniería Informática. Como tal, en este documento se aplicaran todos los conocimientos adquiridos durante los años de estudio en dicho grado. El proyecto que se va a desarrollar está basado en una aplicación Java J2EE.

Desde hace años, Java y mas concretamente en su versión J2EE, lleva dominando el mercado en cuanto al desarrollo de aplicaciones web se refiere. El despegue de internet en los últimos tiempos, la sencillez y la rápida aceptación de este lenguaje, han sido las claves para que Java en su versión empresarial sean líderes en el desarrollo de aplicaciones web.

Algunas de sus ventajas mas conocidas son aquellas que tienen que ver con que Java es un lenguaje multiplataforma: “write once and run everywhere”, es fácil de usar, es orientado a objetos, es robusto, es ampliamente soportado por la comunidad y está en constante evolución. Por lo tanto estamos en condiciones de asegurar que Java es buena opción a la hora de desarrollar un proyecto web con ciertas garantías de éxito.

SÍNTESIS

En lo que al proyecto se refiere, el proyecto está basado en un sistema que denominaremos Material Market a lo largo de todo el trabajo.

La idea que subyace detrás de este Market es un sistema que gestiona la compra-venta de materiales por parte de los usuarios del sistema. A grandes rasgos este sistema permitirá la publicación de ofertas y demandas de materiales por lo usuarios registrados en el sistema.

Gracias a este, los usuarios de esta plataforma podrán realizar actividades de comercio electrónico, con todos los beneficios que esto conlleva: deslocalización geográfica, ahorro en costes de sistemas y ahorro en tiempo.

El sistema permite poner en contacto a personas y permite gestionar sus las necesidades de comercio electrónico de una forma ágil y dinámica que de otra forma no sería posible.

Tal y como se ha comentado en el apartado anterior, el sistema se implementará utilizando la tecnología Java J2EE que tantos beneficios aporta.

FUNCIONALIDAD BÁSICA

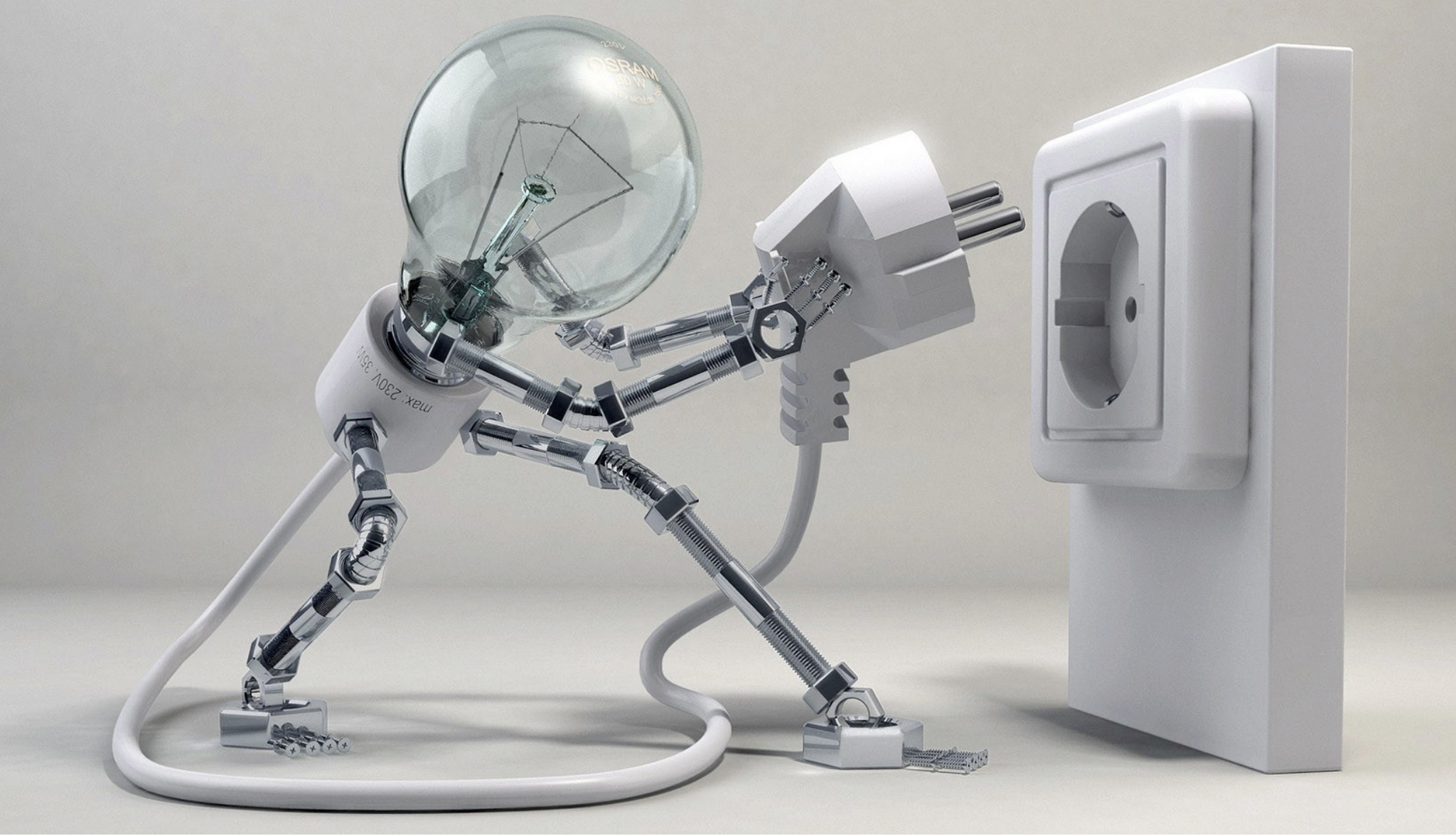
El sistema aportará las siguientes funcionalidades:

- Permitirá publicar demandas de materiales.
- Permitirá publicar ofertas de materiales.
- Permitirá buscar ofertas y demandas de materiales.
- Permitirá aplicar tanto a las ofertas como a las demandas de materiales.
- Permitirá gestionar la actividad de los usuarios: qué ofertas y demandas han publicado, y qué usuarios han aplicado a estas, etc.
- Existirá un administrador que controle el buen uso de la aplicación dado por los usuarios. Podrá eliminar ofertas y demandas si lo cree oportuno. Podrá buscar ofertas y demandas al igual que los usuarios.
- El sistema gestionará el alta y baja de los usuarios, la seguridad, mostrará sugerencias de ofertas y demandas, etc.

PRESENTACIÓN

Apartados:

- Justificación
- Objetivos
- Enfoque y metodología
- Planificación
- Productos obtenidos



JUSTIFICACIÓN DEL PROYECTO

En los últimos años la forma de hacer negocios ha cambiado drásticamente gracias al uso de las nuevas tecnologías de información. El gran culpable de este auge ha sido internet. De un tiempo atrás hasta la actualidad, internet ha pasado de ser algo casi inexistente, a ser el centro de nuestras vidas. Internet permite que todo y todos estemos conectados. En consecuencia han surgido nuevas formas de comunicación, como son las redes sociales, y nuevas formas de hacer negocios, como es el e-commerce o comercio electrónico.

Tiempo atrás, para hacer negocios era requisito indispensable tener una cartera de clientes para poder realizar una actividad profesional con un mínimo de garantías de éxito. Actualmente, gracias a internet y a las redes sociales, la cartera de clientes está implícita en internet. Todos los potenciales clientes están allí. Simplemente se necesita reunirlos en una plataforma donde converjan y se puedan poner en contacto para hacer negocios. Esta es la idea que subyace en nuestro sistema. Así pues, simplemente con dirigirse a nuestra plataforma, un usuario podrá realizar ofertas o demandas sobre sus necesidades actuales y hacer negocios con otros usuarios existentes.

OBJETIVOS

El objetivo del presente trabajo no es otro que el de construir un sistema software capaz de gestionar la información del modelo de negocio presentado anteriormente.

Ademas del objetivo anterior, que es el que persigue nuestro TFG, otros objetivos por los que se suele construir un sistema son aquellos relacionados con las oportunidades de negocio que existen en el mercado. Es decir, el objetivo es conseguir beneficios, hablando desde un punto de vista económico.

Para nuestro caso, implementar e implantar nuestro sistema, supondría una oportunidad de negocio en un segmento donde en la actualidad no hay muchas alternativas. Con un plan de empresa adecuado, con su estudio de viabilidad correspondiente, este proyecto podría ser el comienzo de un aventura profesional para una persona emprendedora, donde el objetivo es claramente obtener un beneficio económico.

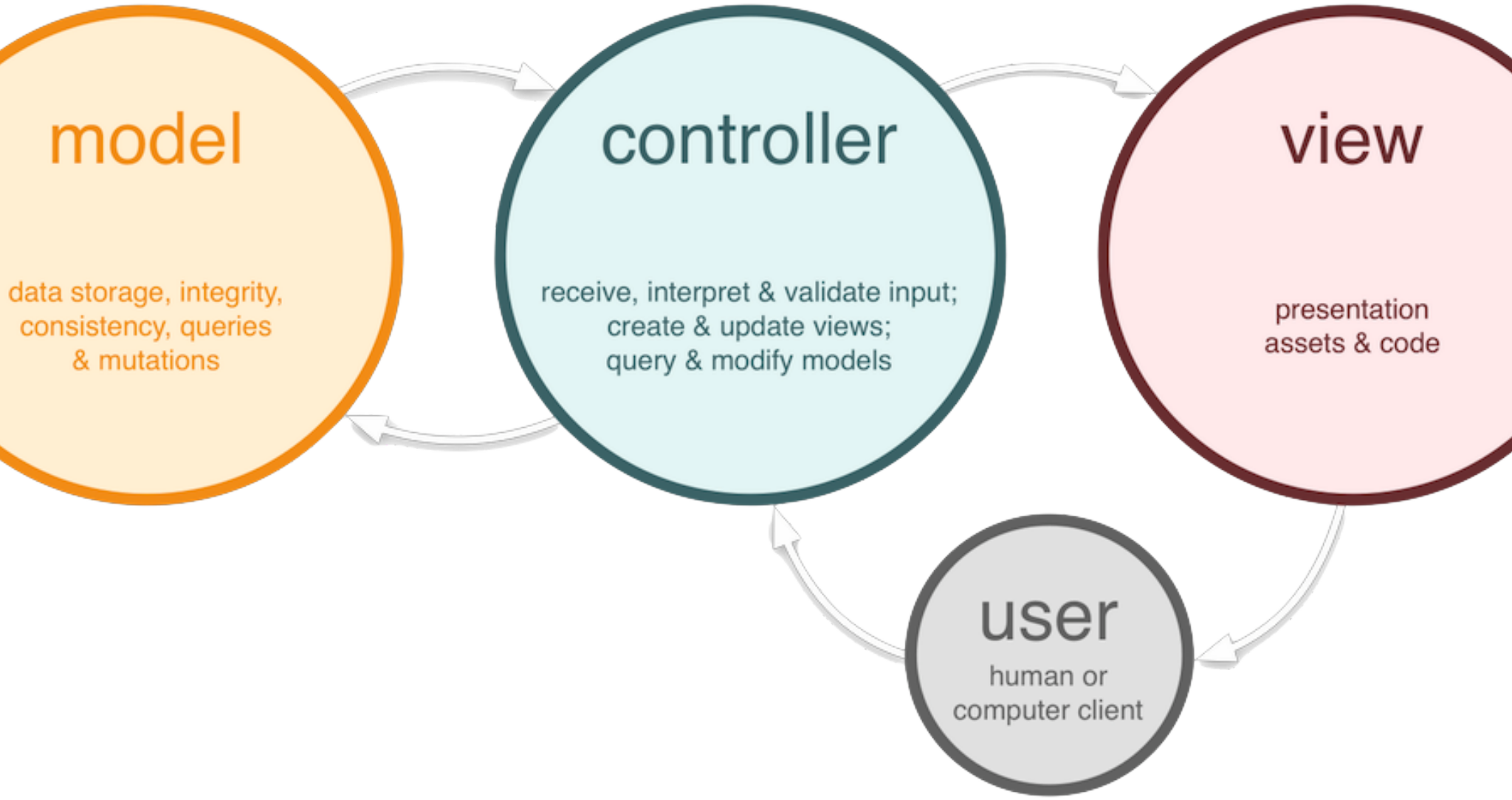
Visto del lado de los usuarios, el objetivo de esta aplicación es que los usuarios se aprovechen de las ventajas y beneficios que le proporciona esta plataforma de e-commerce. En consecuencia, también les reportaría beneficios económicos.

ENFOQUE Y MÉTODO SEGUIDO

En cuanto a la metodología de desarrollo aplicada, se seguirá un desarrollo basado en el ciclo de vida en cascada clásico, de tal forma que el inicio de capa etapa coincide con la finalización de la etapa anterior.

Etapas clásicas:

- Análisis de requisitos
- Diseño del sistema.
- Implementación del sistema software
- Pruebas
- Implantación y entrega del proyecto.
- Mantenimiento



ARQUITECTURA CLIENTE-SERVIDOR DE TRES CAPAS

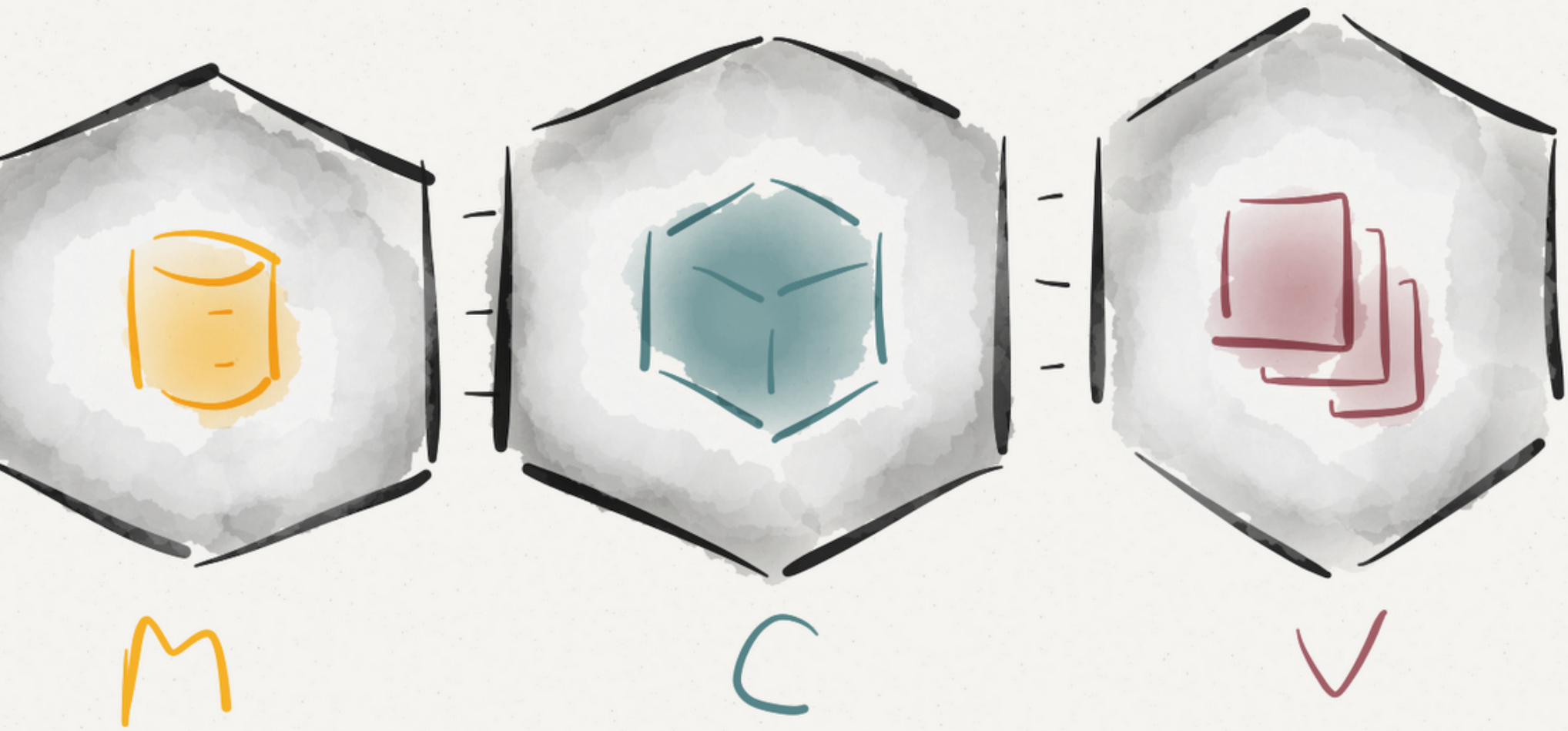
En cuanto al *enfoque* se refiere, la arquitectura que se utilizará en nuestra aplicación será una arquitectura cliente-servidor de tres capas:

- *Cliente*: navegador del cliente.
- *Servidor de aplicaciones J2EE*: procesamiento de la aplicación. Usaremos el servidor JBOSS en su versión 7.1.1
- *Servidor de base de datos*: administración de datos. Usaremos el SGBD PostgreSQL en su versión 9.2

MODELO VISTA CONTROLADOR

En cuanto a la representación de la arquitectura software, utilizaremos el patrón de estilo arquitectónico *MVC* para realizar la división lógica en capas: Modelo-Vista-Controlador. La división quedaría como sigue:

- *Capa presentación*: esta capa contendrá todos los elementos relacionados con la gestión de la presentación.
- *Capa de dominio*: esta capa contendrá todos los elementos relacionados con la lógica de negocio de la aplicación.
- *Capa de servicios técnicos:* esta capa contendrá todos los elementos relacionados con el modelo.



CAPA DE PRESENTACIÓN

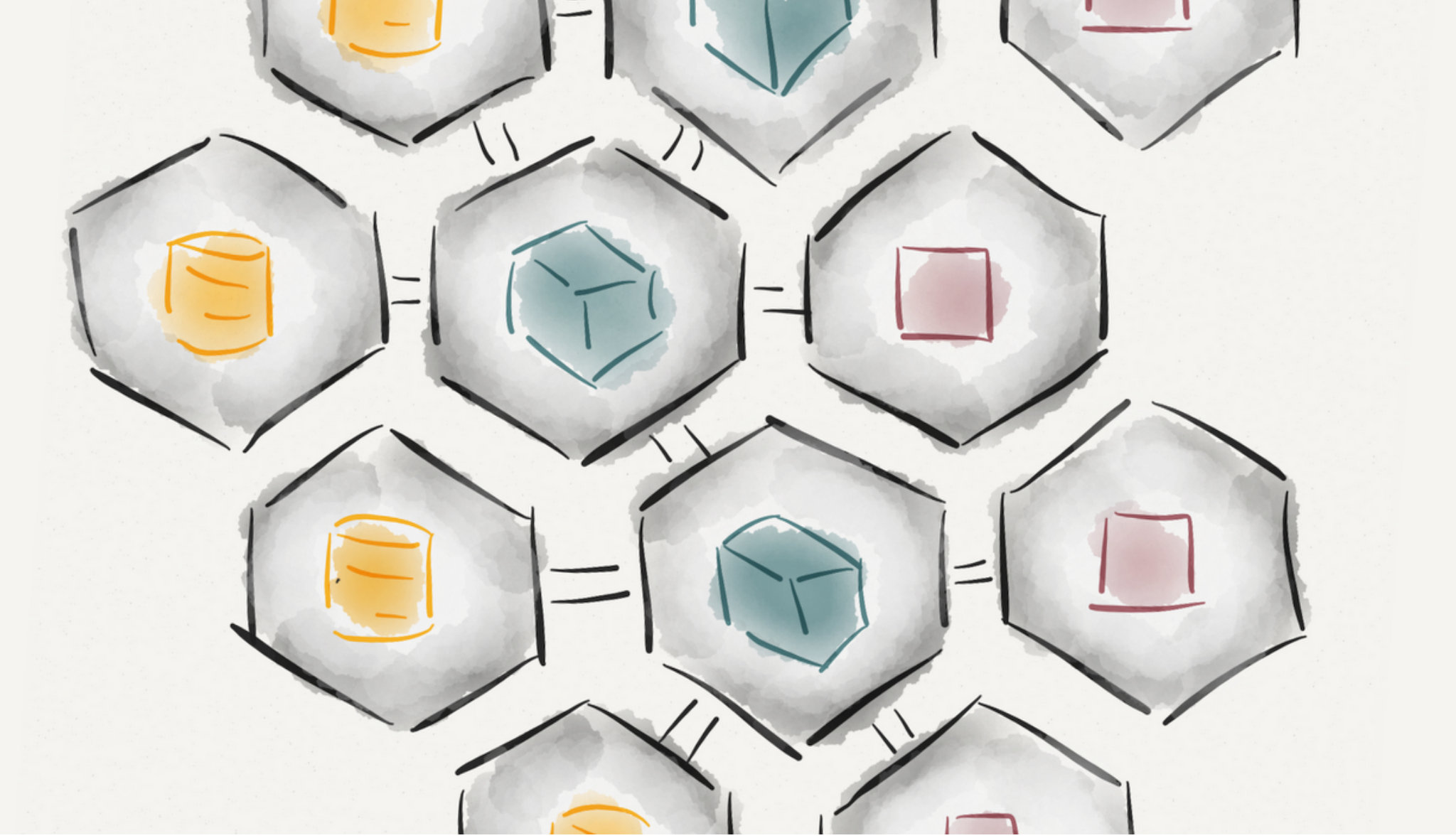
Como punto de partida y como decisión tecnológica, se usará el framework JSF + PrimeFaces que proporciona una implementación de un MVC. Además, este tándem, aporta un conjunto de componentes gráficos listo para usar como elementos ricos de interfaz gráfica en la capa de presentación.

Los elementos que participarán en la capa de presentación serán:

- *xhtml* para realizar el renderizado de las paginas web. Se usará Javascript para dotar de dinamismo a la paginas web generadas en el lado del cliente.
- *ManagedBean* para gestionar las acciones que ocurren en las paginas web. Actúa como controlador de la acciones..

En el caso de haber seleccionado Struts2 como MVC, los elementos que de forma equivalente participaran en la capa de presentación serían:

- *JSP's* para realizar el renderizado de las paginas web.
- *Actions* para gestionar las acciones que ocurren el paginas web. Actúa como controlador de la acciones.



CAPA DE DOMINIO

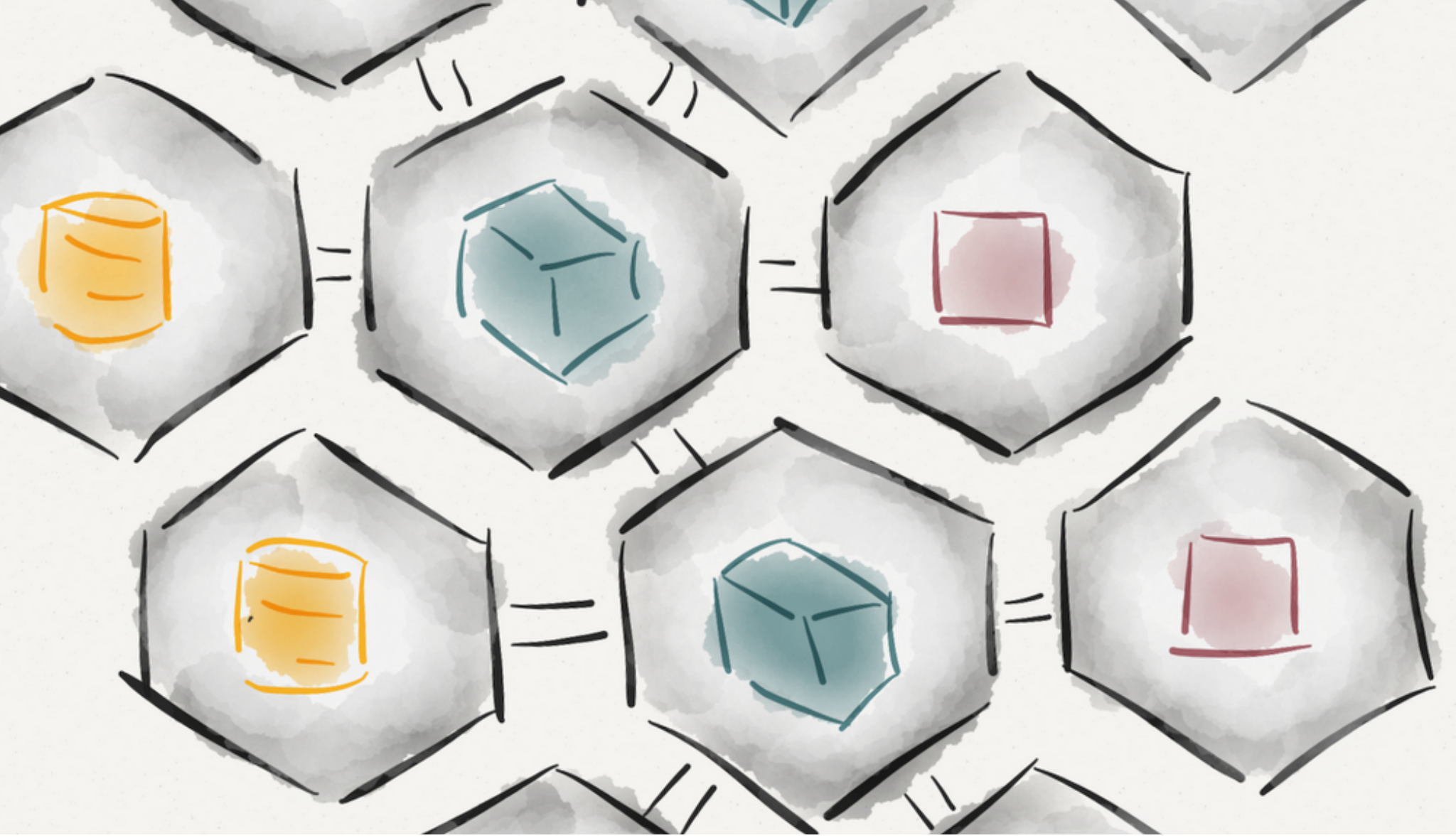
Como decisión tecnológica se utilizará la especificación EJB 3 para construir y gestionar los elementos de esta capa.

Algunas ventajas de esta tecnología:

- Desarrollo intuitivo y sencillo, ya que los EJB3 son simplemente objetos de java planos (POJOs).
- Buen rendimiento y soporte de ciclo de vida mejorados.
- Adoptan la gestión de la persistencia gracias al API de JPA 2.0.
- Gestión de transaccionalidad.
- Se introduce la inyección de dependencias.

Las elementos clave que participaran en la capa de dominio serán:

- *Los controladores de dominio o EJB's*. Estos serán los encargados de gestionar la lógica de negocio de la aplicación.



CAPA DE SERVICIOS TÉCNICOS

Como decisión tecnológica se utilizará la especificación JPA 2.0 para construir y gestionar los elementos de esta capa. Como implementación se usará Hibernate en su versión mas reciente 4.3.4. Para implementar esta capa se utilizará el API de JPA puro, sin usar directamente en el código clases de Hibernate. De esta forma se consigue que esta capa sea independiente de la implementación de la tecnología utilizada. Simplemente cambiando el descriptor de persistencia, donde se indica que el motor ORM es Hibernate, por otro motor como EclipseLink, nuestra aplicación seguiría funcionando sin tener que realizar ningún cambio en el código. Esta idea sigue el principio que Martin Fowler promueve: “Programa contra interfaces” y es una característica bastante deseable, consiguiendo un bajo acoplamiento entre la clases y menos dependencias.

Las elementos clave que participaran en la capa de servicios técnicos serán:

- *Los bean de entidad o EntityBeans.* Estos elementos manejaran el estado del modelo en todo momento. Tienen una relación directa con las tablas del modelo existentes en el servidor a las que representan.

HERRAMIENTAS

Se utilizarán las siguientes herramientas para realizar las diferentes tareas del proyecto

- Eclipse Juno J2EE como IDE principal de desarrollo, en su versión 4.3.2.
- Java Runtime Environment en su versión 1.6.
- Java EE en su versión 1.6 con sus correspondientes especificaciones para JSP's, Servlets, JPA, EJB, etc.
- Maven como sistema gestor del proyecto software, en su versión 3.0.3. Se usará para generar los entregables software y como gestor de dependencias.
- Magic Draw UML como herramienta de diseño.
- OpenProj como herramienta de planificación.
- iBooks Author, Pages y Keynotes como herramientas de creación de documentos.
- Gimp como herramienta de edición de imágenes.

RIESGOS

Los riegos principales que encontramos son:

- Riesgo en el uso de tecnologías desconocidas: desconocimiento en el desarrollo de aplicaciones web usando JSF-PrimeFaces.
- Riesgo en el ajuste de la planificación y que pueda ocurrir un desvío por causas como la comentada anteriormente.

CONTINGENCIAS

Debido a los riesgos inherentes en el desarrollo de la aplicación, en el supuesto caso de que no se pueda continuar con el desarrollo por cualquier problema ocurrido, se *optará* por continuar con el desarrollo aplicando la solución más factible y que guíe a la finalización del proyecto.

En ese caso, se documentará tal circunstancia con los cambios aplicados a lo largo del seguimiento del proyecto.

En el caso específico de que no se pueda desarrollar la aplicación usando JSF-PrimeFaces se optará por el uso de Struts2 como MVC.

PLANIFICACIÓN

El calendario vendrá condicionado por las fechas de los principales hitos requeridos por la asignatura. A saber:

- Entrega de PEC2: 17 de abril de 2014
- Entrega de PEC3: 02 de junio de 2014
- Entrega de PEC4: 16 de junio de 2014

Principalmente se dividen las tareas en bloques entre las fechas de los hitos. La composición de las tareas en el calendario resulta de la siguiente forma:

Tareas a realizar desde el 13 de marzo hasta el 17 de abril:

- Análisis funcional de la aplicación: definición de funcionalidades y diagramas de casos de uso.
- Diseño técnico de la aplicación: diagrama de clases y modelo entidad relación de la base de datos.
- Comienzo del desarrollo e implementación de la aplicación.

Tareas a realizar desde el 18 abril hasta el 02 de junio:

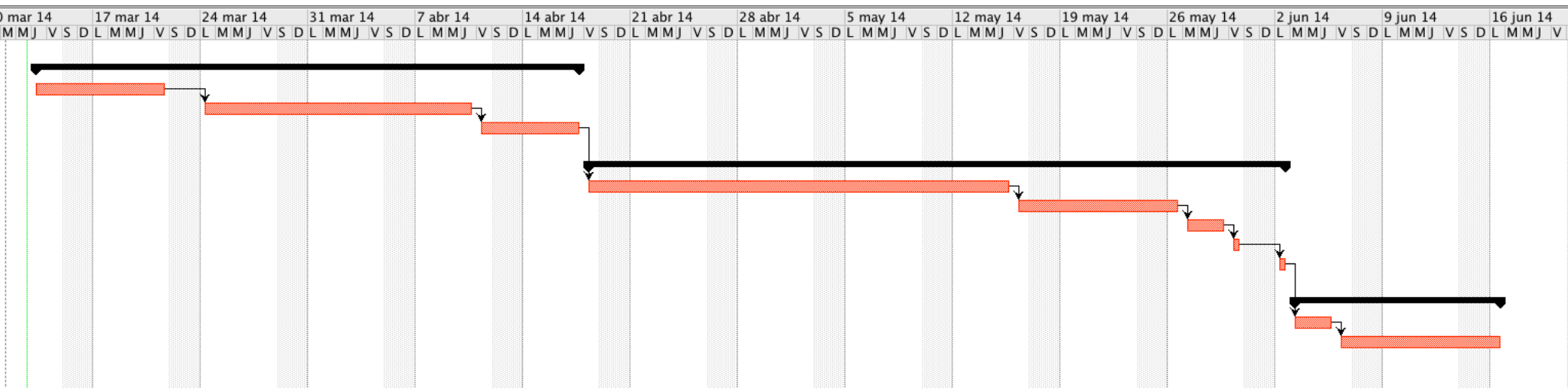
- Desarrollo e implementación la aplicación.
- Pruebas unitarias.
- Pruebas de usuario.
- Entrega del proyecto finalizado: artefacto software EAR + scripts de base de datos.
- Mantenimiento correctivo evolutivo del análisis funcional y del diseño técnico.

Tareas a realizar desde el 03 junio hasta el 16 de junio:

- Memoria del proyecto, con video en caso de que se requiera.
- Mantenimiento correctivo del software en el caso de que fuera necesario y entrega final: artefacto software EAR + scripts de base de datos

PEC2 entrega 17-04-2014	26 days?	**13/03/14 8:00**	**17/04/14 17:00**
analisis	7 days?	13/03/14 8:00	21/03/14 17:00
diseño técnico	14 days?	24/03/14 8:00	10/04/14 17:00
desarrollo	5 days	11/04/14 8:00	17/04/14 17:00
PEC3 entrega 02-06-2014	32 days?	**18/04/14 8:00**	**2/06/14 17:00**
desarrollo	20 days?	18/04/14 8:00	15/05/14 17:00
pruebas unitarias	7 days	16/05/14 8:00	26/05/14 17:00
pruebas de usuario	3 days?	27/05/14 8:00	29/05/14 17:00
empaquetamiento software	1 day?	30/05/14 8:00	30/05/14 17:00
mantenimiento correctivo documentación	1 day?	2/06/14 8:00	2/06/14 17:00
PEC4 entrega 16-06-2014	10 days?	**3/06/14 8:00**	**16/06/14 17:00**
mantenimiento correctivo software	3 days?	3/06/14 8:00	5/06/14 17:00
memoria del proyecto	7 days?	6/06/14 8:00	16/06/14 17:00

PLANIFICACIÓN EN DETALLE Y DIAGRAMA DE GANT



PRODUCTOS OBTENIDOS

DOCUMENTACIÓN A ENTREGAR

En cuanto a los entregables, se entregará una parte de documentación y otra parte con los artefactos software que constituyen el sistema.

En cuanto a la *documentación* se refiere, se entregará:

- El *análisis funcional* del sistema donde se detalla la funcionalidad del sistema implementado.
- El *diseño técnico* del sistema donde se explican de forma detallada como se ha implementado el sistema.
- Una *memoria* donde explica todo el proyecto en su conjunto.

ENTREGABLES SOFTWARE

En cuanto a los entregables software se refiere:

- se entregará un archivo tipo EAR que contendrá la implementación de la funcionalidad que proporciona el sistema. Este archivo se podrá desplegar en cualquier servidor de aplicaciones del mercado que cumpla con las especificaciones de *Java J2EE para la versión 1.6*, aunque tal y como se ha especificado previamente, el servidor de aplicaciones que se utilizará en el proyecto será el JBoss en su versión 7.1.1.

- se entregará un conjunto de scripts de SQL con toda la definición del modelo de datos del sistema para poder instalarlo en el sistema gestor de base de datos *PosgreSQL 9.2*. Este conjunto de scripts seguirá escrupulosamente el estándar de SQL, de tal forma que se pueda ejecutar en otros sistemas gestores de base de datos diferentes de PostgreSQL, por si en futuro se desea migrar a otro sistema gestores de base de datos.

GLOSARIO

Fuente: Wikipedia

GLOSARIO DE TÉRMINOS

AJAX - Asynchronous JavaScript And XML. es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

API - Interfaz de programación de aplicaciones, o Application Programming Interface, es el conjunto de funciones y métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

EAR - Un archivo EAR (Enterprise Archive) es un tipo de archivo que permite ejecutar aplicaciones J2EE escritas en el lenguaje Java. Los archivos EAR están comprimidos con el formato ZIP y cambiada su extensión a .jar. Existen tres operaciones básicas con este tipo de archivos: ver contenido, comprimir y descomprimir. Los archivos EAR contiene archivos WAR y JAR.

Eclipse IDE - es un IDE compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar aplicaciones Java.

EJB - Los EJB proporcionan un modelo de componentes distribuido estándar del lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (concurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

Framework - Marco de trabajo. Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Hibernate - es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo 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.

IDE - Entorno de Desarrollo Integrado o Integrated Development Environment. es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

HTML - Lenguaje de marcas de hipertexto (HyperText Markup Language), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc.

JAR - Un archivo JAR (por sus siglas en inglés, Java ARchive) es un tipo de archivo que permite ejecutar aplicaciones escritas en el lenguaje Java. Las siglas están deliberadamente escogidas para que coincidan con la palabra inglesa "jar" (tarro). Los archivos JAR están comprimidos con el formato ZIP y cambiada su extensión a .jar. Existen tres operaciones básicas con este tipo de archivos: ver contenido, comprimir y descomprimir.

Javascript - es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,3 basado en prototipos, imperativo, débilmente tipado y dinámico.

J2EE - Java Enterprise Edition. Es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JBOSS - Servidor de Aplicaciones J2EE. JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE, disponible en el mercado,

ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business

JRE - Java Runtime Environment. es un conjunto de utilidades que permite la ejecución de programas Java.

JSF - Java Server Faces. es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario).

JSP - Java Server Pages. es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML entre otros tipos de documentos. JSP es similar a PHP pero usa el lenguaje de programación Java.

Librería - conjunto de clases agrupadas por afinidad. Habitualmente se proporcionan en un archivo de tipo JAR.

Maven - es una herramienta de software para la gestión y construcción de proyectos Java.

MVC - El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

ORM - Object-Relational mapping o mapeo de objetos relacional. es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional, utilizando un motor de persistencia.

PostgreSQL - es un SGBD relacional orientado a objetos.

PrimeFaces - es un componente para JSF de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web.

POJO - Un POJO (acrónimo de Plain Old Java Object) es una sigla creada por Martin Fowler y utilizada por programadores Java para enfatizar el uso de clases simples y que no dependen de un framework en especial.

Programación por capas - La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

SGDB - Sistema Gestor de Base de datos. es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.

SQL - El lenguaje de consulta estructurado o SQL (Structured Query Language) es un

lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

Struts2 - Framework MVC. Struts se basa en el patrón de arquitectura de software Modelo-Vista-Controlador (MVC) el cual se utiliza ampliamente y es considerado de gran solidez. De acuerdo con este Framework, el procesamiento se separa en tres secciones diferenciadas llamadas el modelo, las vistas y el controlador.

TOMCAT - Servidor Web. funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP).

WAR - Un archivo WAR (Web Archive) es un tipo de archivo que permite ejecutar aplicaciones Web. Los archivos WAR están comprimidos con el formato ZIP y cambiada su extensión a .jar. Existen tres operaciones básicas con este tipo de archivos: ver contenido, comprimir y descomprimir. Los archivos WAR contienen archivos JAR.

Nota.- Se hecho uso de la Wikipedia para obtener todo el glosario de términos.

4

BIBLIOGRAFÍA

Fuente: UOC

BIBLIOGRAFÍA

Por citar algunos:

- Material oficial asignatura Ingeniería del Software del Grado de Ingeniería de la UOC.
- Material oficial asignatura Análisis y Diseño de Patrones del Grado de Ingeniería de la UOC.
- Material oficial asignatura Ingeniería del Software de Componentes distribuidos del Grado de Ingeniería de la UOC.
- Material oficial asignatura TFG para redactar documentos.