

**PROYECTO DE GENERACIÓN AUTOMÁTICA DE CÓDIGO PARA EL
DESARROLLO DE SOFTWARE DE GESTIÓN DE NEGOCIOS, VÍA
WEB UTILIZANDO EL ESTANDAR TECNOFACTOR**

FELIPE LOPERA VALENCIA

JUAN PABLO ESCOBAR CANO

MARCO TULIO RUIZ FLOREZ

**INSTITUCION UNIVERSITARIA DE ENVIGADO
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS
ENVIGADO
2011**

**PROYECTO DE GENERACIÓN AUTOMÁTICA DE CÓDIGO PARA EL
DESARROLLO DE SOFTWARE DE GESTIÓN DE NEGOCIOS, VÍA
WEB UTILIZANDO EL ESTANDAR TECNOFACTOR**

FELIPE LOPERA VALENCIA
JUAN PABLO ESCOBAR CANO
MARCO TULIO RUIZ FLOREZ

Trabajo de grado presentado para optar al título de Ingeniero en Sistemas

Francisco Montoya González
Asesor

**INSTITUCION UNIVERSITARIA DE ENVIGADO
FACULTAD DE INGENIERÍAS
INGENIERÍA DE SISTEMAS
ENVIGADO
2011**

DEDICATORIA

Dedicamos este gran logro especialmente a nuestras familias, ya que, gracias a ellos fue posible iniciar nuestro proceso de formación profesional por factores económicos y morales.

AGRADECIMIENTOS

Agradecemos a nuestras familias por el apoyo incondicional durante nuestro proceso de formación profesional, a todos nuestros compañeros, docentes, asesor, y demás personas, que de una u otra forma estuvieron acompañándonos en todo momento.

Muchas gracias a Tecnofactor, por permitirnos iniciar nuestro proceso laboral y por el apoyo brindado en este proyecto de práctica empresarial.

CONTENIDO

INTRODUCCIÓN	12
1. INFORME ASPECTOS GENERALES LA DE PRÁCTICA EMPRESARIAL en Tecnofactor S.A	13
1.1. ASPECTOS GENERALES DE LA PRÁCTICA.....	13
1.2. CENTRO DE PRÁCTICA.....	13
1.2.1. Razón social:.....	13
1.2.2. MISIÓN:	14
1.2.3. VISIÓN:.....	14
1.2.4. Valores corporativos	14
1.3. OBJETIVOS DE PRÁCTICA	15
1.4. JUSTIFICACIÓN.....	15
2. ESPECIFICACIONES DE ACUERDO CON LA MODALIDAD (ÁMBITO) DE PRÁCTICA.....	15
2.1. PLANTEAMIENTO DEL PROBLEMA.	17
2.2. EQUIPO DE TRABAJO.....	17
2.3. METODOLOGÍA DE TRABAJO.....	19
2.3.1. Especificación de un problema (llamado Bug).....	19
2.3.2. Diseño de solución.....	19
2.3.3. Desarrollo.....	19
2.3.4. Pruebas.....	20
2.3.5. Despliegue.....	20
2.4. RESULTADOS ESPERADOS.....	22
3. SISTEMATIZACIÓN DE LA INFORMACIÓN DE ACUERDO CON LA MODALIDAD DE LA PRÁCTICA.....	22

3.1.	Proceso de capacitación.....	22
3.2.	Metodología de trabajo.....	23
3.3.	Inconvenientes generales.....	23
3.3.1.	Estimación de tiempos y recursos.....	23
3.3.2.	Despliegues en diferentes ambientes.....	24
3.4.	Interacción entre roles.....	24
4.	PROYECTO DE GENERACIÓN AUTOMÁTICA DE CÓDIGO PARA EL DESARROLLO DE SOFTWARE DE GESTIÓN DE NEGOCIOS, VÍA WEB UTILIZANDO EL ESTANDAR TECNOFACTOR	26
4.1.	PLANTEAMIENTO DEL PROBLEMA DEL PROYECTO DE PRÁCTICA.	26
4.2.	JUSTIFICACIÓN DEL PROYECTO DE PRÁCTICA.....	26
4.3.	OBJETIVOS DEL PROYECTO DE PRÁCTICA	27
4.3.1.	Objetivo general	27
4.3.2.	Objetivos Específicos	27
4.4.	DISEÑO METODOLÓGICO DEL PROYECTO DE PRÁCTICA.	27
4.4.1.	Enfoque de investigación	27
4.4.2.	Fuentes de Información.....	27
4.4.3.	Tratamiento de la Información	28
4.4.4.	Población	28
4.4.5.	Etapas del proyecto	28
4.5.	CRONOGRAMA DE ACTIVIDADES DEL PROYECTO DE PRÁCTICA 28	
4.6.	PRESUPUESTO.....	30
4.6.1.	PRESUPUESTO GLOBAL DEL PROYECTO.....	30
4.6.2.	DESCRIPCIÓN DE LOS GASTOS DE PERSONAL.....	32

4.6.3. DESCRIPCIÓN DE MATERIALES Y SUMINISTROS.....	33
5. Desarrollo del proyecto de práctica.....	33
5.1.1. Características de la arquitectura del proyecto base de Tecnofactor. 33	
5.1.2. Requisitos del proyecto generador de código, para el desarrollo de proyectos de software de gestión de negocios, vía web.....	36
5.1.3. Adaptaciones sobre el proyecto base de Tecnofactor.	37
5.1.4. Restricciones, atributos y validaciones	40
5.1.5. Diseño del proyecto de generación de código	41
6. Inconvenientes en el desarrollo de la práctica	41
7. Conclusiones	41
8. Recomendaciones	42
9. Anexos.....	43
Referencias bibliográficas	44

LISTA DE TABLAS

	Pág.
Tabla 1. Aspectos generales de la práctica empresarial	13
Tabla 2. Actividades organizacionales	16
Tabla 3. Equipo de trabajo	18
Tabla 4. Cronograma de actividades.	27
Tabla 5. Presupuesto global del proyecto	30
Tabla 6. Descripción de los gastos de personal.	31
Tabla 7. Descripción de materiales y suministros.	32
Tabla 8. Necesidades.	35
Tabla 9. Características.	36

LISTA DE FIGURAS

	Pág.
Figura 1. Gestión de producción de Tecnofactor.	20
Figura 2. Proceso de mantenimiento a los productos de software.	24
Figura 3. Cronograma de actividades.	29
Figura 4. Arquitectura utilizada en Tecnofactor.	33
Figura 5. Framework Spring.	34

GLOSARIO

CRUD: Es el acrónimo de Crear, Obtener, Actualizar y Borrar.

DAO: acrónimo de *Objetos de acceso a datos* son una serie de objetos que le permiten tener acceso y manipular datos mediante programación en bases de datos locales o remotas.

MAESTRO: Formulario que permite realizar las operaciones básicas(CRUD).

SQL: El Lenguaje de consulta estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

FRAMEWORK: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

PATRONES ARQUITECTÓNICOS: son los que definen la estructura de un sistema software, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema.

SPRING: Un framework contenedor liviano basado en la técnica **Inversión de Control** (IoC) y una implementación de desarrollo según el paradigma de **Orientación a Aspectos** (AOP).

MVC: es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

BUG: En inglés significa error. También llamado incidencia. En Tecnofactor, representa un error en una aplicación o un requisito nuevo para la misma.

RESUMEN

En este informe se trata un problema identificado en la empresa Tecnofactor, el cual se basa en la necesidad de un proyecto, generador de código automático, que permita agilizar los procesos de desarrollo de software y la utilización de los estándares arquitectónicos definidos en la empresa. Con el objeto de dar un aporte a la solución del problema planteado, se pretende definir y diseñar un proyecto de software con las características antes mencionadas, que automatice el proceso de creación de código con base en código ya existente, simulando el proceso de creación de código utilizado por los desarrolladores en la creación de vistas u opciones poco complejas.

ABSTRACT

This report addresses a problem identified in the company Tecnofactor, which generates the need for a project, called automatic code generator. It facilitates the speed up of software development processes and the use for architectonic standards defined by the organization. In order to make a contribution to the problem's solution, it is intended to define and design a software project with the above characteristics, which automates the process of creating code based on an existing one, simulating the creating code process used by developers in creating views or non-complex options.

INTRODUCCIÓN

En este informe, se especifica la modalidad de práctica empresarial seleccionada, todas sus características y funciones principales adquiridas dentro del centro de práctica (Tecnofactor). Además se define el diseño de un proyecto para la empresa, el cual tiene como objetivo principal, la generación de código automático de acuerdo a la arquitectura de desarrollo utilizada en la misma empresa.

El desarrollo de un producto de software a la medida, tiende a cambiar esta percepción de producto de software a prestación de servicios de desarrollo de software, una vez terminada la fase inicial del producto, pues constantemente surgen nuevas modificaciones a estos productos, de acuerdo a las necesidades del cliente. Un generador de código automático, en proyectos de software a la medida, debe seguir la arquitectura definida para el proyecto y además, debe permitir realizar modificaciones al código generado automáticamente.

1. INFORME ASPECTOS GENERALES LA DE PRÁCTICA EMPRESARIAL en Tecnofactor S.A

1.1. ASPECTOS GENERALES DE LA PRÁCTICA

Tabla 1. Aspectos generales de la práctica empresarial

Nombre de Estudiante	Juan Pablo Escobar Cano Marco Tulio Ruiz Flórez Felipe Lopera Valencia
Programa Académico	Ingeniería de Sistemas
Nombre de la AGENCIA O CENTRO DE PRÁCTICA	Tecnofactor S.A
NIT.	900063400-8
Dirección	Diagonal 32 B # 29 sur 24
Teléfono	444 0080
Dependencia o Área	Análisis y desarrollo de software
Nombre Completo del Jefe del estudiante	Andrés Felipe Saldarriaga
Cargo	Gerente de tecnología
Labor que desempeña el estudiante	Analista desarrollador Analista desarrollador Analista desarrollador
Nombre del Asesor de práctica	Francisco Montoya González
Fecha de inicio de la práctica	Junio 20 de 2011
Fecha de finalización de la práctica	Noviembre 22 de 2011

Autoría.

1.2. CENTRO DE PRÁCTICA

1.2.1. Razón social: Tecnofactor S.A

TECNOFACTOR S.A. es una empresa orientada a la prestación de servicios de desarrollo e integración de soluciones en tecnologías de información y comunicación. Específicamente la Empresa ofrece soluciones como el desarrollo de software a la medida, capacitación especializada y consultoría. Aseguramos la calidad de nuestros servicios. Trabajamos con metodologías ágiles, flexibles y de alto rendimiento; que facilitan un acompañamiento

profesional, con personal capacitado, comprometido y en constante desarrollo. En **TECNOFACTOR S.A.** estamos siempre dispuestos a conocer del negocio de nuestros clientes para ser un nuevo integrante en su estrategia de negocio.

1.2.2. MISIÓN:

En **TECNOFACTOR S.A.** ofrecemos apoyo en tecnologías informáticas para las empresas que tengan conformado este departamento o que posean conciencia de ello, satisfaciendo sus necesidades de automatización de procesos, creando sistemas a la medida o adaptando los ya existentes, que permitan marcar la diferencia con sus competidores. Brindamos altos estándares de calidad, metodologías ágiles, flexibles, de alto rendimiento y un acompañamiento profesional con personal capacitado, comprometido y en constante aprendizaje, dispuestos a conocer del negocio de nuestros clientes. Aseguramos el crecimiento de nuestra organización con rentabilidad y desarrollo personal y profesional.

1.2.3. VISIÓN:

En el 2011 ser una empresa sólida y con un portafolio de clientes diversificado que nos reconocen por la responsabilidad y calidad de nuestros servicios de tecnología informática, con personal competente, comprometido con nuestros clientes y a la vanguardia en tecnología.

1.2.4. Valores corporativos

- Responsabilidad
Cumplimos adecuada y confiablemente con nuestros compromisos organizacionales.
- Alegría
Trabajamos con entusiasmo, actitud positiva y amor por lo que hacemos.
- Perseverancia
Somos decididos, firmes y constantes en la ejecución de nuestros propósitos y solución de problemas.
- Solidaridad
Tenemos disposición para compartir y trabajar en las tareas y metas comunes de la organización.
- Respeto
Reconocemos, comprendemos y apreciamos a los demás, ya sea por sus ideas, experiencias y valor como personas.
- Ética

Obramos correcta y acertadamente en beneficio de la organización y sus clientes.¹

1.3. OBJETIVOS DE PRÁCTICA

- Diseñar e implementar estrategias para dar solución a problema.
- Implementar acciones y estrategias de mejoramiento continuo de los procesos que usan con frecuencia.
- Identificar oportunidades de negocio (clientes actuales y potenciales) y trasmitirla a los roles que corresponda haciendo la gestión pertinente.
- Asistir y participar activamente de los programas, proyectos e iniciativas corporativas en las que se vea involucrado.

1.4. JUSTIFICACIÓN

Tecnofactor, como todas las empresas de desarrollo de software, debe estar a la vanguardia en cuanto a nuevas tecnologías y métodos que facilitan y mejoran los productos de software. Además, debe hacer lo posible por estar un paso delante de la competencia, tratando de mantener sus clientes y darse a conocer en diferentes mercados.

Gracias a los conocimientos adquiridos durante el proceso de formación profesional, se tiene la capacidad de brindar apoyo en cada uno de los problemas y requerimientos solicitados a la empresa, buscando la solución adecuada, para dar cumplimiento a las necesidades del cliente.

2. ESPECIFICACIONES DE ACUERDO CON LA MODALIDAD (ÁMBITO) DE PRÁCTICA

Análisis y desarrollo de software

El propósito principal de un analista desarrollador, en Tecnofactor, es analizar y construir productos de software que cumplan con altos estándares de calidad y con todos los requerimientos del cliente. Dentro de este perfil, se pueden resaltar las siguientes actividades con sus respectivas áreas:

¹ TECNOFACTOR S.A. Información Corporativa, Envigado [on line]

Tabla 2. Actividades organizacionales

AREA DE RESULTADOS	ACTIVIDADES
<p align="center">GESTIÓN</p> <p>Cumplimiento de las fechas y tiempos estimados para la ejecución de las actividades de manera oportuna y eficiente.</p>	<ul style="list-style-type: none"> Participar en la reunión de seguimiento y control de los proyectos.
	<ul style="list-style-type: none"> Asegurar que los desarrollos efectuados cumplan con las etapas de ciclo de vida del proceso de desarrollo garantizando la calidad y completitud de los requisitos, actividades o tareas asignadas.
	<ul style="list-style-type: none"> Mantener la trazabilidad de los requisitos con el producto terminado.
	<ul style="list-style-type: none"> Identificar e informar sobre oportunidades de mejora de los componentes de los sistemas sobre los que trabaja.
	<ul style="list-style-type: none"> Participar en la gestión de cambios del producto.
	<ul style="list-style-type: none"> Analizar y comprender los requisitos del sistema.
<p align="center">PRODUCCIÓN</p> <p>Trabajar bajo estándares de calidad y completitud que permitan satisfacer las necesidades de los clientes.</p>	<ul style="list-style-type: none"> Apoyar en la elicitación, modelamiento y verificación de requisitos
	<ul style="list-style-type: none"> Definir las estructuras, vistas y funcionalidades bajo la arquitectura que suplirán los requisitos del cliente.
	<ul style="list-style-type: none"> Realizar la construcción de acuerdo con el diseño definido.
	<ul style="list-style-type: none"> Realizar pruebas unitarias de las funcionalidades.
	<ul style="list-style-type: none"> Ejecutar correcciones a problema detectados en las funcionalidades.
	<ul style="list-style-type: none"> Apoyar en el montaje de la aplicación de los diferentes ambientes.
	<ul style="list-style-type: none"> Apoya para el mantenimiento de la documentación del producto (EF y DC).
	<ul style="list-style-type: none"> Apoyar en el montaje de todos los componentes necesarios para el correcto funcionamiento de los aplicativos.
	<ul style="list-style-type: none"> Apoya la construcción del manual de usuario.
<ul style="list-style-type: none"> Asegurar el control de la configuración de los productos de trabajo e ítems de configuración requeridos para el producto. 	
<ul style="list-style-type: none"> Asegurar el control de la configuración de los productos de trabajo e ítems de configuración requeridos para el producto. 	

Tomado de Documentación interna Tecnofactor.

2.1. PLANTEAMIENTO DEL PROBLEMA.

La dinámica de trabajo de **TECNOFACTOR S.A.** está fundamentada en mejores prácticas y procedimientos estándar que han sido adaptados a partir del Modelo de Capacidad y Madurez - CMMI. Este modelo identifica prácticas generales y específicas de ingeniería de software, que pueden ser aplicadas a grandes, medianas y pequeñas empresas y pueden ser fácilmente adaptadas a distintos dominios de la industria de software. Fue desarrollado inicialmente para los procesos relativos al software por la Universidad Carnegie-Mellon para el SEI (Software Engineering Institute).

Los diseños arquitectónicos empresariales están basados principalmente en el principio fundamental del diseño que es la reutilización para la disminución de costos y en la integración para la agilidad y visión en el cambio; los diseños arquitectónicos son alineados específicamente a las metas organizacionales y a los diferentes puntos de vista que tiene la organización.

Adicionalmente, el desarrollo de las arquitecturas definidas se hace a partir de tecnologías de última generación que permiten asegurar la personalización, reutilización, ajuste a procesos de negocio, automatización y seguridad en todas las aplicaciones que desarrollamos o los servicios que prestamos.

2.2. EQUIPO DE TRABAJO.

Tecnofactor, está dividido en varios equipos de trabajo llamados EAD (Equipos de Alto Desempeño), estos equipos, manejan diferentes proyectos para diferentes clientes de la empresa. Cada equipo está compuesto de la siguiente manera:

- Líder Gestor

Se encargan de coordinar todas las actividades, tiempos, entregas, interacciones y demás funciones que facilitan la relación con los clientes y usuarios finales de nuestros servicios.

- Líder técnico

Se encargan de asegurar los productos de trabajo mediante una serie de pruebas que van desde la verificación del código hasta la estabilización en los diferentes ambientes del producto.

- Analista desarrollador

Es la persona que realiza las funciones de un analista técnico y de un programador; es decir, parte de una información previa recibida del analista funcional, en función de la cual desarrolla las aplicaciones y organiza los datos.

Además de los roles mencionados anteriormente, existen otros 2 roles superiores transversales a cada equipo de trabajo en la empresa:

- Arquitecto de soluciones

Sus labores son, básicamente, gestionar y optimizar los procesos y productos de la compañía de acuerdo a la estrategia de negocio, para analizar y tomar decisiones basadas en la relación costo-beneficio.

- Gerente de proyectos

Ejecutar una gestión integral de los proyectos y procesos definidos para la elaboración de un producto o la prestación de un servicio, el mejoramiento continuo y la implementación de mejores prácticas para la optimización de los recursos; manteniendo un desempeño exitoso de los equipos de alto desempeño y la satisfacción de los clientes.

Cada uno de nosotros se encuentra en un equipo diferente, por lo tanto en proyectos diferentes. Para el proyecto de práctica, el equipo de trabajo está conformado así:

Tabla 3. Equipo de trabajo

Nombres	Cargo
Juan Pablo Escobar Cano	Analista desarrollador
Marco Ruiz Florez	Analista desarrollador
Felipe Lopera Valencia	Analista desarrollador
Andrés Felipe Saldarriaga	Jefe de práctica

Autoría.

2.3. METODOLOGÍA DE TRABAJO.

Características de la metodología de trabajo utilizada en Tecnofactor:

- Sigue y utiliza los procedimientos, procesos, estándares, herramientas de la compañía para cumplir con las necesidades del cliente.
- Dirige sus esfuerzos para cumplir con calidad, cumplimiento y completitud los requerimientos del proyecto.
- Se preocupa por conocer las generalidades del proyecto (satisfacción del cliente, porcentaje de cumplimiento).
- Conoce sus fortalezas y aspectos por mejorar lo que le permite implementar una metodología de trabajo para cumplir con las actividades asignadas.
- Canaliza su energía al cumplimiento de los objetivos diarios (Concentración).
- Conoce la integridad del producto y proyecto, donde el resultado exitoso es considerado cuando responde a todas las necesidades definidas.²

La metodología de trabajo está compuesta por los siguientes puntos:

2.3.1. Especificación de un problema (llamado Bug)

El proceso inicia con la especificación de un problema, ya sea, un error en una aplicación, o un cambio nuevo para la misma. Esta especificación, se realiza sobre una herramienta de software, llamada Mantis, la cual facilita la gestión de cada uno de los proyectos en la empresa.

2.3.2. Diseño de solución

Una vez especificado el problema, y comprendido a cabalidad, por el analista desarrollador, se procede a realizar un diseño de la posible solución al problema enunciado. Con base en este diseño, se realiza una estimación del tiempo que demora la implementación de la solución planteada.

2.3.3. Desarrollo

En esta etapa, con base en el diseño realizado, se procede a desarrollar el código necesario para dar solución al problema especificado en la etapa inicial.

² TECNOFACTOR S.A. Información Corporativa, Documentación Perfil analista. Envigado.

2.3.4. Pruebas

Una vez realizado el desarrollo correspondiente a la solución del bug, se procede a realizar pruebas a la nueva funcionalidad desarrollada; estas pruebas se hacen en un ambiente de pruebas, el cual se asemeja al ambiente de producción de cada uno de los proyectos de Tecnofactor.

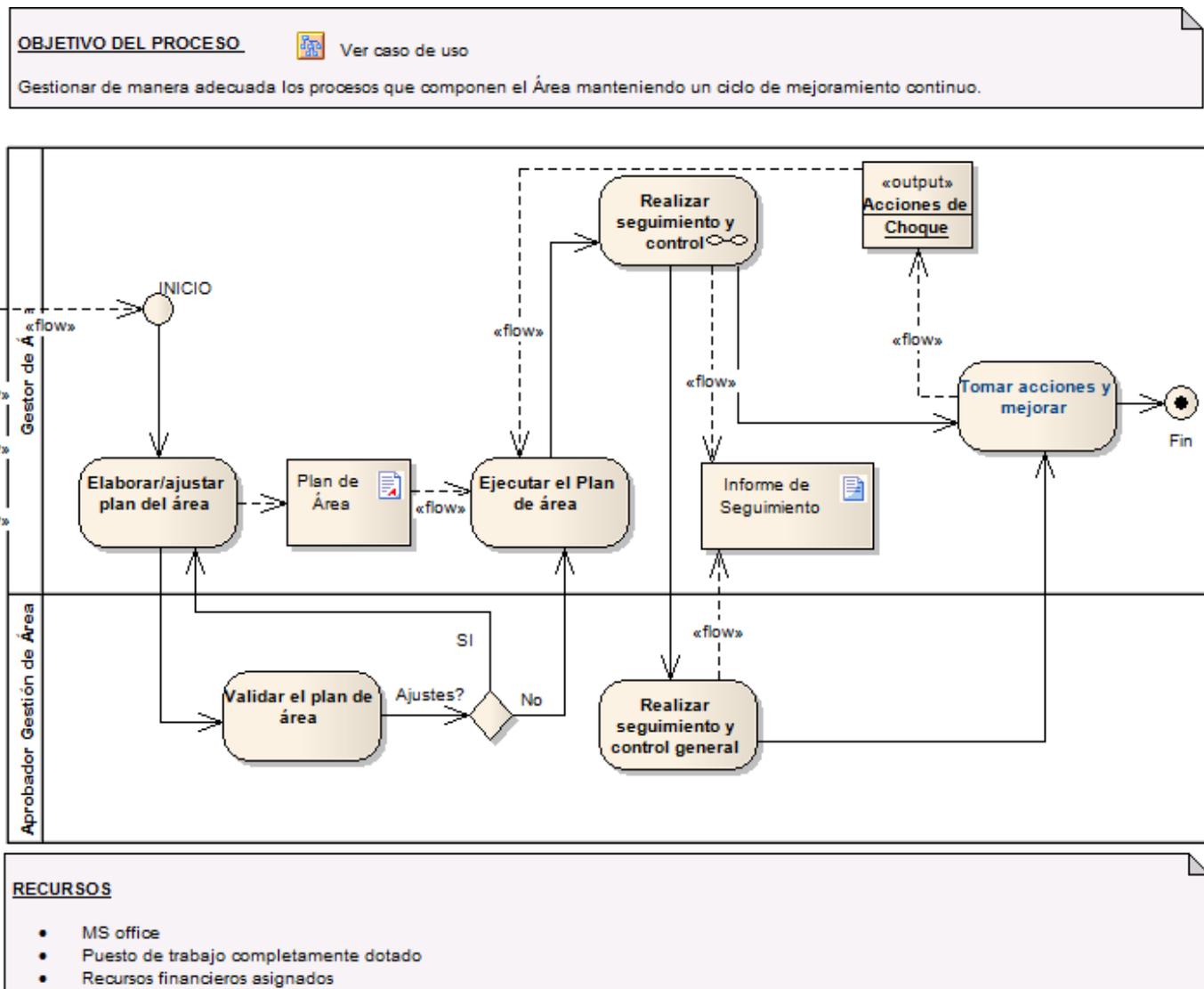
En caso de que las pruebas no entreguen los resultados esperados, el proceso se debe iniciar nuevamente desde el paso 2.3.2, hasta que la solución desarrollada sea correcta.

2.3.5. Despliegue

Una vez superada la etapa de pruebas, se genera una nueva versión de la aplicación modificada y se realiza el despliegue en ambiente de producción.

En la siguiente figura se detalla con mayor profundidad la metodología de trabajo utilizada en Tecnofactor:

Figura 1. Gestión de producción de Tecnofactor.



Tomado de Documentación interna Tecnofactor.

2.4. RESULTADOS ESPERADOS.

- Cumplir con todas las expectativas de los clientes, desarrollando soluciones de calidad para los requerimientos solicitados.
- Adquirir cada vez más conocimientos con el paso del tiempo, para crecer como profesional y poder brindar un mejor apoyo en los procesos de la empresa.
- Tener la capacidad para diseñar estrategias que ayuden al desarrollo de los productos en menor tiempo.
- Participar en todo el ciclo de vida de software, desde su etapa inicial hasta su etapa final, brindando el apoyo técnico, en cada una de ellas, afianzando conocimientos y apoyando el crecimiento de la empresa.
- Realizar especificaciones de software, que ayuden a los clientes a comprender de una forma fácil y con un lenguaje más técnico, la solución a sus necesidades planteadas.

3. SISTEMATIZACIÓN DE LA INFORMACIÓN DE ACUERDO CON LA MODALIDAD DE LA PRÁCTICA.

3.1. Proceso de capacitación

En Tecnofactor, para la gestión de sus proyectos, manejan diferentes herramientas de software diseñadas para este fin. En el periodo de inducción y capacitación en la empresa, una de las actividades comunes durante este periodo, es la capacitación sobre el manejo de cada una de estas herramientas.

El resto de actividades durante el periodo de capacitación, se basa en la adaptación del empleado, a la forma de trabajo empleada en Tecnofactor; esta actividad, consta de los siguientes puntos:

- Capacitación CMMI

En esta actividad, se capacita al empleado acerca de los procesos y certificaciones que tiene la empresa a nivel de CMMI; además, se incentiva al empleado a generar progresos para la mejora de los

procesos establecidos, con el fin de avanzar a los siguientes niveles de certificación CMMI.

- Capacitación acerca de la arquitectura utilizada

Con el acompañamiento del arquitecto, se da a conocer la forma de trabajo y la arquitectura general para los proyectos de software desarrollados en Tecnofactor.

3.2. Metodología de trabajo

Como ya se especificó, la metodología de trabajo utilizada en Tecnofactor, consta, básicamente de 5 etapas. En cada una de estas etapas, es común encontrar diferentes problemas que aumentan el tiempo de solución al bug planteado, lo cual produce un impacto en las estimaciones temporales, retrasando la fecha de entrega de los productos de software desarrollados. Para dar solución a estas situaciones, Tecnofactor cuenta con una estructura organizacional, la cual permite escalar cada uno de los problemas presentados de acuerdo a su complejidad, de la siguiente manera:

Cuando el nivel de complejidad de la situación, supera el alcance del analista desarrollador, ya sea a nivel de negocio del cliente, o técnico, existen 2 roles superiores a los cuales se puede acudir en estos casos:

- Líder técnico

Esta persona, cuenta con los conocimientos técnicos y de negocio necesarios para apoyar situaciones complejas que puede encontrar un analista, en cualquiera de las etapas de la metodología de trabajo de la empresa. Además apoya

- Líder gestor

Es la persona con mayor conocimiento del negocio del cliente. El líder gestor es el puente de comunicación entre el cliente (usuario final) y el grupo de analistas, por lo tanto, es la persona indicada para apoyar en la solución de dudas a nivel de negocio encontradas por el analista desarrollador.

3.3. Inconvenientes generales

3.3.1. Estimación de tiempos y recursos

Uno de los problemas más comunes para el analista desarrollador, es realizar una buena estimación de tiempos y recursos para el desarrollo de una solución

a un bug asignado; los 2 roles mencionados anteriormente, con base en los conocimientos específicos de cada uno, apoyan el proceso de estimación, tratando de mitigar el riesgo de fallo en el mismo.

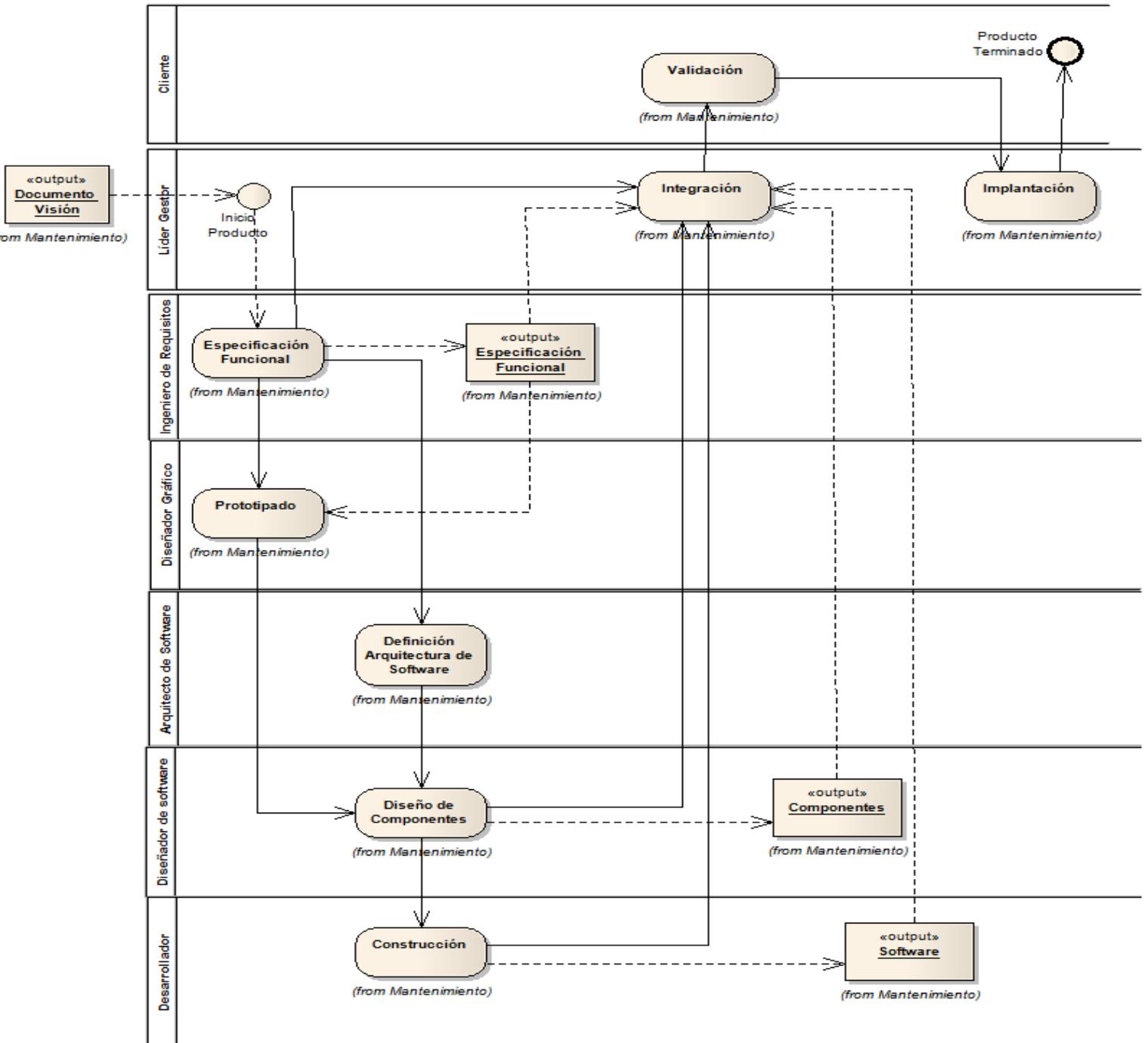
3.3.2. Despliegues en diferentes ambientes

Para el despliegue de aplicaciones en los diferentes ambientes manejados en Tecnofactor, es muy común encontrar problemas que impiden el correcto funcionamiento de la aplicación

3.4. Interacción entre roles

La siguiente figura, muestra en detalle el proceso de interacción entre los diferentes roles para la solución de los requerimientos de los clientes:

Figura 2. Proceso de mantenimiento a los productos de software.



Tomado de Documentación interna Tecnofactor.

4. PROYECTO DE GENERACIÓN AUTOMÁTICA DE CÓDIGO PARA EL DESARROLLO DE SOFTWARE DE GESTIÓN DE NEGOCIOS, VÍA WEB UTILIZANDO EL ESTANDAR TECNOFACTOR

4.1. PLANTEAMIENTO DEL PROBLEMA DEL PROYECTO DE PRÁCTICA.

En los proyectos de software en la etapa de codificación es común, entre otras, la creación de una vista que permita realizar operaciones básicas de inserción, actualización, búsqueda y borrado con base en sus atributos.

Actualmente se encuentra gran variedad de herramientas CASE (Computer Assisted Software Engineering), las cuales permiten generar de forma automática formularios con cada una de las operaciones básicas antes mencionadas. Estas herramientas generan su propia arquitectura.

Por otra parte, el framework de desarrollo Spring, proporciona una funcionalidad que permite la creación automática de un formulario con todas sus acciones básicas, pero encapsula todo su código. La utilización de las funcionalidades de estas herramientas en ocasiones no resulta muy útil, cuando es necesario seguir una arquitectura de software específica diferente, o cuando es necesario realizar modificaciones a las vistas generadas automáticamente, y más aun, cuando la empresa que desarrolla software, presta servicios de software a la medida, cambiando, mejorando o adaptando sus productos, según la necesidad de sus clientes.

Es un hecho destacado por numerosos investigadores y profesionales que los sistemas informáticos deben evolucionar para adecuarse a los siempre cambiantes requisitos del entorno.³

Debido a estas implicaciones ha surgido la inquietud de:

¿Será posible diseñar un proyecto generador de código automático con base en la arquitectura utilizada en Tecnofactor?

4.2. JUSTIFICACIÓN DEL PROYECTO DE PRÁCTICA.

El diseñar un proyecto con generación automática de código java, con una arquitectura basada en Spring MVC (utilizada por Tecnofactor en sus proyectos), traerá como beneficio a Tecnofactor, en sus proyectos de desarrollo Web, disminuir el tiempo de desarrollo de cada proyecto, garantizando la calidad del software y la utilización de su propia arquitectura, la cual no es fácil encontrar en herramientas generadoras de código, como las herramientas CASE. Utilizar la arquitectura manejada en Tecnofactor en el proyecto a diseñar, permitirá, además, su utilización no sólo en proyectos nuevos, sino también, en nuevos cambios o adaptaciones a proyectos existentes, pues seguirán conservando su arquitectura inicial.

³ J. A. Carsí, I. Un Generador Automático de Planes de Migración de Datos. Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia – España. [on line]

4.3. OBJETIVOS DEL PROYECTO DE PRÁCTICA

4.3.1. Objetivo general

Diseñar un proyecto base que permita la generación automática de código java, para el desarrollo de proyectos de software de gestión de negocios, vía web, utilizando el lenguaje de programación java, basado en el patrón arquitectónico Modelo Vista Controlador con el estándar de Tecnofactor.

4.3.2. Objetivos Específicos

- Especificar las características de la arquitectura utilizada en el proyecto base de Tecnofactor.
- Definir los requisitos del proyecto generador de código, para el desarrollo de proyectos de software de gestión de negocios, vía web.
- Definir adaptaciones necesarias sobre el proyecto base de Tecnofactor para el diseño del nuevo proyecto de generación de código.
- Definir restricciones, atributos, validaciones y operaciones posibles a utilizar en el diseño del proyecto generador de código automático.
- Diseñar el proyecto generador de código automático java, basado en la arquitectura del proyecto base de Tecnofactor.

4.4. DISEÑO METODOLÓGICO DEL PROYECTO DE PRÁCTICA.

4.4.1. Enfoque de investigación

Este trabajo de investigación tiene un enfoque cuantitativo, ya que, es un proyecto de desarrollo tecnológico. El enfoque cuantitativo se dedica a recoger, procesar y analizar datos cuantitativos o numéricos sobre variables previamente determinadas.⁴

4.4.2. Fuentes de Información

La principal fuente de información utilizada es la documentación interna alojada en la intranet de Tecnofactor. Además de esto, se utiliza, como medio virtual, internet. También se utilizan como fuente algunas personas conocedoras del tema como líderes técnicos y líderes gestores de la empresa.

⁴ SARDUY DOMINGUEZ, Yanetsys. El análisis de información y las investigaciones cuantitativo y cualitativo, Escuela Nacional de Salud Pública [on line].

4.4.3. Tratamiento de la Información

El tratamiento de la información se hace en diferentes etapas:

- Lectura de la información.
- Análisis de la información.
- Comunicación de resultados.

4.4.4. Población

Analista desarrollador, líder técnico, líder gestor, Gerente de proyectos.

4.4.5. Etapas del proyecto

El proyecto, básicamente, está dividido en 5 etapas correspondientes a los 5 objetivos específicos definidos para el mismo, los cuales se desarrollan más adelante.

4.5. CRONOGRAMA DE ACTIVIDADES DEL PROYECTO DE PRÁCTICA

Tabla 4. Cronograma de actividades.

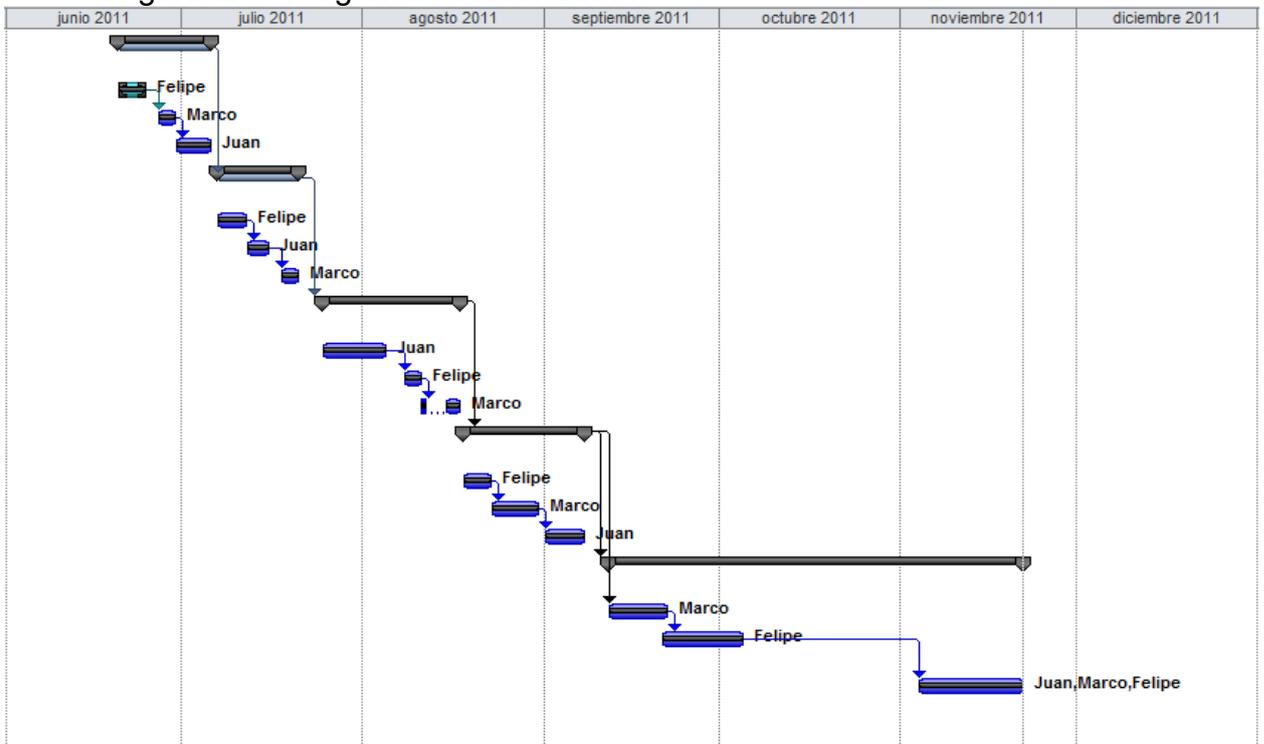
Id	Nombre de tarea	Duración	Comienzo	Fin	% completado
1	Especificar las características de la arquitectura utilizada en el proyecto base de Tecnofactor.	12 días	lun 6/20/11	mar 7/5/11	100%
2	Indagar sobre la arquitectura modelo vista controlador	5 días	lun 6/20/11	vie 6/24/11	100%
3	Investigar sobre el framework Spring	3 días	lun 6/27/11	mié 6/29/11	100%
4	Investigar la integración del framework Spring con la arquitectura MVC	4 días	jue 6/30/11	mar 7/5/11	100%
5	Definir los requisitos del proyecto generador de código, para el desarrollo de proyectos de software de gestión de negocios, vía web.	10 días	jue 7/7/11	mié 7/20/11	100%
6	Determinar las características básicas del proyecto	3 días	jue 7/7/11	lun 7/11/11	100%

7	Investigar las necesidades a solucionar con el proyecto	4 días	mar 7/12/11	vie 7/15/11	100%
8	Fijar cuales necesidades hacen referencia a cuales características	3 días	lun 7/18/11	mié 7/20/11	100%
9	Modificar el proyecto base de Tecnofactor de acuerdo a los requisitos especificados.	17.75 días	lun 7/25/11	mié 8/17/11	100%
10	Definir la implementacion de mejoras, para la generación de codigo automatico	9 días	lun 7/25/11	jue 8/4/11	100%
11	Especificar la integración del framework	3 días	lun 8/8/11	mié 8/10/11	100%
12	Definir las capas por las cuales estará dividido el proyecto	3 días	jue 8/11/11	mié 8/17/11	100%
13	Definir restricciones, atributos, validaciones y operaciones posibles a utilizar en el diseño del proyecto generador de código automático.	15 días	jue 8/18/11	mié 9/7/11	100%
14	Definir validaciones y restricciones permitidas para cada uno de los atributos	3 días	jue 8/18/11	lun 8/22/11	100%
15	Definir estructuras de los atributos y clases	6 días	mar 8/23/11	mar 8/30/11	100%
16	Definir la estructura SQL	5 días	jue 9/1/11	mié 9/7/11	100%
17	Diseñar el proyecto generador de código automático java, basado en la arquitectura del proyecto base de Tecnofactor.	51 días	lun 9/12/11	lun 11/21/11	100%
18	Definir estructura archivos bases	8 días	lun 9/12/11	mié 9/21/11	100%

19	Definir una plantilla en Excel para cargar los atributos que se asociarán a una vista para generar su código	10 días	mié 9/21/11	mar 10/4/11	100%
20	Definir la forma de lectura y adaptación de archivos bases, para las nuevas vistas generadas automáticamente	12 días	vie 11/4/11	lun 11/21/11	100%

Autoría.

Figura 3. Cronograma de actividades.



Autoría.

4.6. PRESUPUESTO

4.6.1. PRESUPUESTO GLOBAL DEL PROYECTO

Tabla 5. Presupuesto global del proyecto

PRESUPUESTO GLOBAL DEL PROYECTO

RUBROS	FUENTES		TOTAL
	Estudiantes	TECNOFACTOR	
Personal (Nómina)	\$ 0	\$ 2.400.000	\$ 2.400.000
Material y suministros	\$ 0	\$ 2.000.000	\$ 2.000.000
Salidas de campo	\$ 0	\$ 0	\$ 0
Bibliografía	\$ 0	\$ 0	\$ 0
Equipos	\$ 2.500.000	\$ 2.500.000	\$ 5.000.000
TOTAL	\$ 2.500.000	\$ 6.900.000	\$ 9.400.000

Autoría.

4.6.2. DESCRIPCIÓN DE LOS GASTOS DE PERSONAL

Tabla 6. Descripción de los gastos de personal.

DESCRIPCIÓN DE LOS GASTOS DE PERSONAL					
Nombre Del personal	FUNCIÓN EN EL TRABAJO	DEDICACIÓN (horas)	FUENTES		TOTAL
			Estudiantes	TECNOFACTOR	
Juan Pablo Escobar Cano	Analista desarrollador	180	\$ 0	\$ 4444,44444	\$ 800.000
Felipe Lopera Valencia	Analista desarrollador	180	\$ 0	\$ 4444,44444	\$ 800.000
Marco Tulio Ruiz Flórez	Analista desarrollador	180	\$ 0	\$ 4444,44444	\$ 800.000
TOTAL		540		\$ 13333,3333	\$ 2.400.000

Autoría.

4.6.3. DESCRIPCIÓN DE MATERIALES Y SUMINISTROS

Tabla 7. Descripción de materiales y suministros.

DESCRIPCIÓN DE MATERIALES Y SUMINISTROS			
Material	FUENTES		TOTAL
	Estudiantes	TECNOFACTOR	
Papelería	\$ 0	\$ 500.000	\$ 500.000
Servicios públicos e internet	\$ 0	\$ 1.500.000	\$ 1.500.000
TOTAL	\$ 0	\$ 2.000.000	\$ 2.000.000

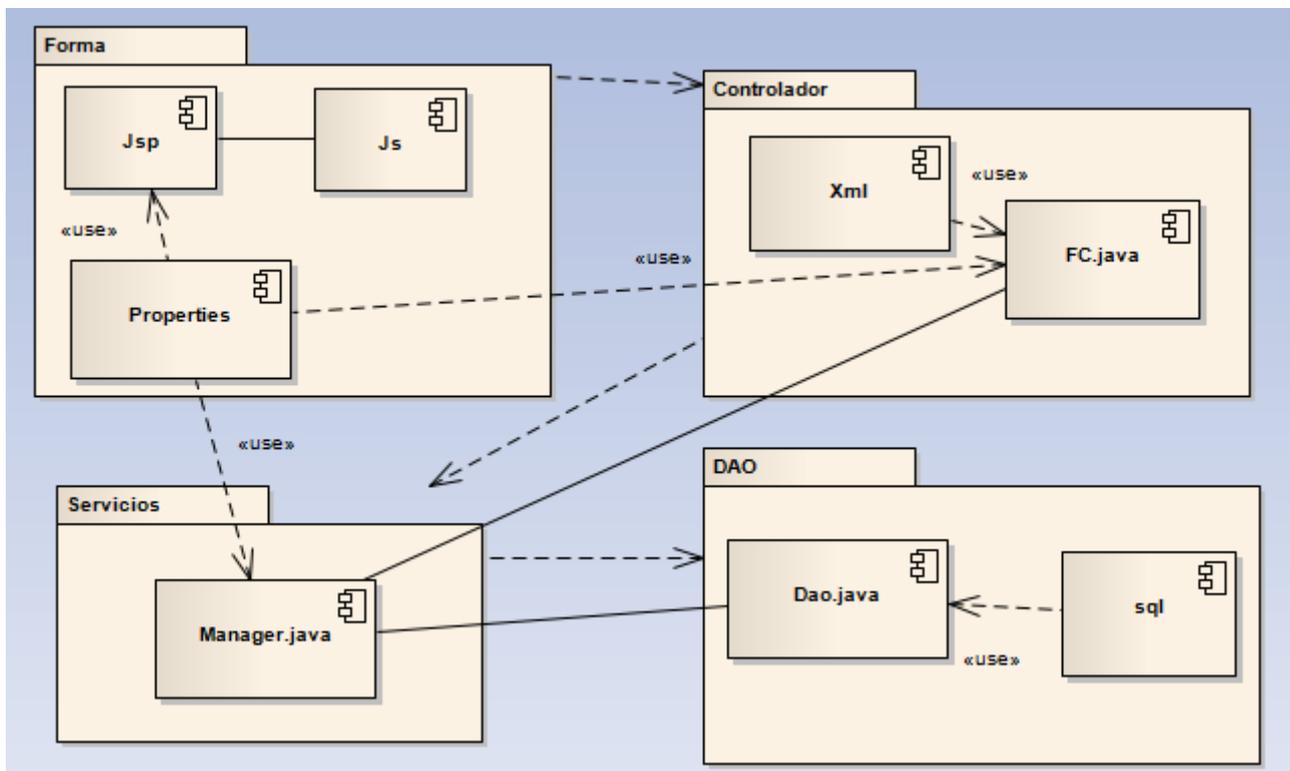
Autoría.

5. Desarrollo del proyecto de práctica

5.1.1. Características de la arquitectura del proyecto base de Tecnofactor.

La arquitectura utilizada en Tecnofactor se basa en el patrón arquitectónico MVC, conjuntamente apoyado con un framework que permite configurar y desarrollar todo el potencial de la arquitectura (Spring).

Figura 4. Arquitectura utilizada en Tecnofactor.



Autoría.

Nota:

La mayoría de proyectos en Tecnofactor dependen, para su correcto funcionamiento, del sistema de seguridad (desarrollado en Tecnofactor), el cual permite la autenticación a la aplicación, a través de un usuario y contraseña. Además, en este, se deben registrar cada uno de los recursos utilizados en la aplicación. (Entiéndase por recurso, el controlador asociado a una vista en el sistema).

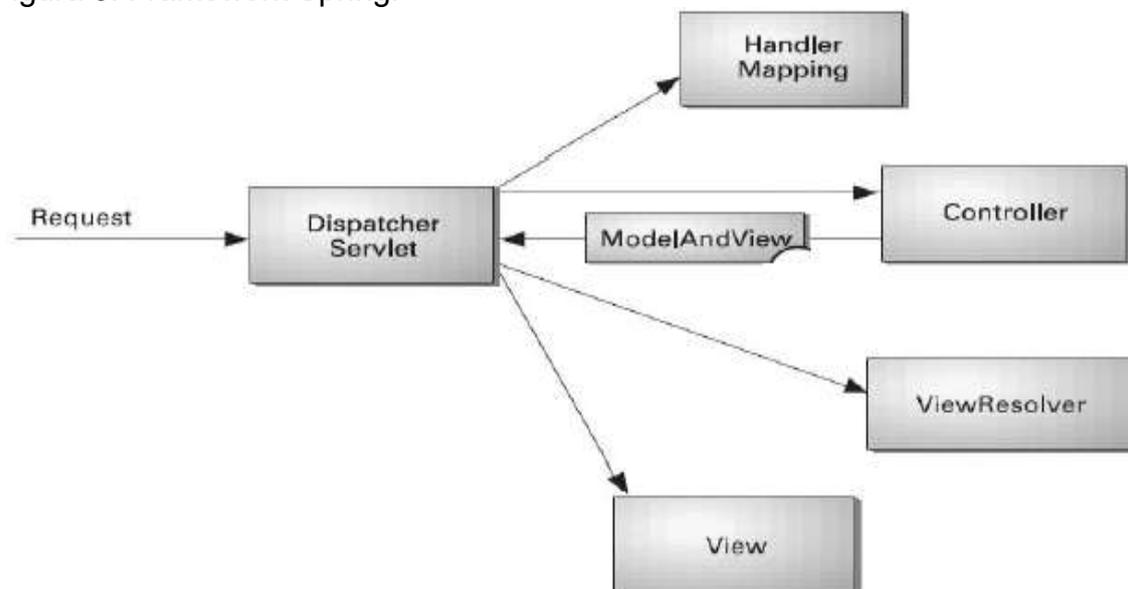
Spring framework

Spring Framework es un marco de trabajo opensource para el desarrollo de aplicaciones bajo la plataforma Java, aunque existen algunas versiones del framework para la plataforma .NET (Spring.net).

Dentro de las características más destacadas de Spring, esta la aplicabilidad y compatibilidad que tiene para implementar y desarrollar aplicaciones con el patrón MVC, para lo cual introduce unos elementos importantes que permiten al desarrollador crear sistemas desacoplados y fáciles de mantener en el tiempo.

En la siguiente figura2 se ilustra la forma como interactúan esos elementos para crear una aplicación con base en el Spring-MVC Framework.

Figura 5. Framework Spring.



Tomado de. Springframework.org

Dispatcher Servlet: Es un servlet implementado por Spring que realiza las tareas de Front-Controller lo que significa que cada una de las peticiones del usuario pasan a través de el, y es responsabilidad del Dispatcher Servlet delegar esa petición del usuario al controlador idóneo para esa tarea en específico.

Handler Mapping: Para obtener el nombre del Controller que va a encargarse de gestionar la petición, Spring incorpora un Handler Mapping el cual le indica al Dispatcher Servlet cual sera el Controller que debiera recibir la petición de usuario. Dentro de Spring existen varios Handler Mapping que tiene características y capacidades diferentes para el mapeo a los Controllers.

Controller: El controller es el encargado de gestionar la petición que viene del usuario, la cual le ha transferido el Dispatcher Servlet. Para poder implementar un Controller con Spring, es necesario heredar de los Controller ya implementados por Spring. Spring trae implementados una serie de Controller que tienen un propósito específico. Uno de los Controllers que mas se usan en las aplicaciones de Tecnofactor, es el controlador "MultiAction" el cual permite implementar una serie de acciones que tengan una lógica similar.

ModelAndView: Luego que el Controller recibe la petición desde el Dispatcher Servlet, se debe crear un objeto que se denomina ModelAndView, el cual tiene la responsabilidad de entregar el nombre de la vista (En este caso una página **jsp**) que se quiere mostrar, y además de esto, los datos que serán desplegados en ella.

ViewResolver: El ViewResolver es el encargado de hacer el mapeo entre el nombre lógico de las vistas y las jsp. Spring trae diferentes ViewResolver para configurar las Vistas de la aplicación. El utilizado en la Plantilla Web es el “ResourceBundleViewResolver” el cual define el mapeo entre los nombre lógicos y las vistas a través de un archivo de propiedades (.properties).

View: Las vistas en este caso serían las paginas jsp que son las que le comunican y capturan información del usuario. Es bueno aclarar que para aumentar la funcionalidad de las aplicaciones se usan una serie de tags de propósito general para las páginas jsp. La librería usada para este propósito es **JSTL**.

JSTL

JSTL es una librería de etiquetas estándar para java, usada en las páginas JSP para cumplir con el desacoplamiento de los componentes de una aplicación y concordar con las especificaciones del modelo MVC.

Los tags JSTL no contiene código Java, lo que hace que no tengamos que compilar las jsp cada vez que hagamos un cambio en ellas, mejorando, tanto el rendimiento, como el tiempo en que nos demoramos en bajar el servidor y reiniciarlo.⁵

5.1.2. Requisitos del proyecto generador de código, para el desarrollo de proyectos de software de gestión de negocios, vía web.

5.1.2.1. Necesidad

Tabla 8. Necesidades.

Necesidad	Prioridad	Interesado	Solución propuesta
N1: Optimizar el tiempo y proceso de desarrollo de aplicaciones en la empresa.	Alta	Director de proyectos. Analista desarrollador.	Diseñar un proyecto que permita la generación automática del código necesario para las operaciones básicas permitidas en un formulario (CRUD).

Autoría.

⁵ TECNOFACTOR S.A. Información Corporativa, Manual Plantilla web. Envigado.

5.1.2.2. Características

Tabla 9. Características.

CARACTERISTICA	NECESIDAD
EL sistema deberá permitir descargar una plantilla de Excel con una estructura definida para la carga de los atributos con los cuales se generará el formulario.	N1
El sistema permitirá cargar un archivo de Excel, con una estructura definida, en el cual, se especifican los atributos para la generación del formulario.	N1
El sistema permitirá registrar la ubicación, en el disco, del proyecto base sobre el cual se realizará la generación de código.	N1
El sistema, para la generación de un formulario deberá gestionar los siguientes tipos de archivos: *.js, *.jsp, *.java, *.sql, *.xls, *.xml, *.properties	N1

Autoría.

5.1.3. Adaptaciones sobre el proyecto base de Tecnofactor.

La base principal del proyecto es la manipulación de los archivos del proyecto base, por ello, se deben registrar ciertos identificadores claves, dentro de los mismos, para que el sistema identifique los fragmentos de código que debe modificar para cada uno de los formularios a generar.

Para cada uno de los archivos a generar, se definió una estructura específica, sobre la cual se debe trabajar:

- Archivos *.sql:

Se manejarán 4 tipos de archivos sql:

- Insert

Para los insert, se manejará un archivo base con la siguiente estructura:

```
INSERT INTO TABLA
(
    CAMPOS
)
VALUES
(
    ?:PARAMETROS
```

)

Para este caso, los identificadores (palabras claves) son TABLA, CAMPOS y PARAMETROS, los cuales deben ser reemplazados por cada uno de los nombres de atributos del formulario que se creará.

- Update

```
UPDATE TABLA  
SET
```

```
    CAMPO = ?:PARAMETRO
```

```
WHERE
```

```
    CAMPO_PK = ?:PARAMETRO_PK
```

En este caso los identificadores claves, son CAMPO, PARAMETRO, CAMPO_PK y PARAMETRO_PK; CAMPO_PK y PARAMETRO_PK, corresponden al nombre del campo clave primaria de la tabla y el valor clave primaria que se envía como parámetro para ejecutar la sentencia UPDATE en la base de datos.

- SelectPk

Este tipo de select, corresponde a la consulta a una tabla en la base de datos, la cual se filtra por el campo clave primaria de la misma. La estructura base para esta consulta es la siguiente:

```
SELECT
```

```
    CAMPO
```

```
FROM TABLA
```

```
WHERE
```

```
    CAMPO_PK = ?:PARAMETRO_PK
```

- SelectList

Este tipo de consulta, devuelve todos los registros de una tabla en la base de datos. La estructura base de esta consulta es la siguiente:

```
SELECT
```

```
    CAMPO
```

```
FROM TABLA
```

Nota:

En caso de tener más de un atributo, se debe registrar cada atributo, separado por comas (,).

Archivos *.java

De acuerdo a la arquitectura especificada anteriormente, para la creación de estos tipos de archivos, se manejarán 3 formatos. Dentro de este trabajo, sólo se definirá, la forma de manejar un método (insert) en la capa de acceso a datos, en la capa de servicios, y en la capa de controlador, junto con la clase del modelo.

- Modelo

Para la creación de clases en el modelo, se deben registrar cada uno de los atributos cargados al sistema, con sus tipos de datos correspondientes y con un modificador de ámbito privado. Además, se deben generar los métodos set y get para cada uno de los atributos registrados, con base en la siguiente estructura:

- DAO

Para esta capa, se definirán interfaces y sus implementaciones.

- Definición de un método, en una interfaz DAO:

```
public void operacion(ObjetoModelo objetoModelo) throws  
SQLException;
```

En este caso, los identificadores claves son ObjetoModelo, el cual, será modificado por el nombre de la clase del modelo a la que corresponde la clase DAO a crear, y operación, que es el método a crear, ya sea insert, update, delete y select.

- Definición de un método insert, en la clase que implementa la interfaz del DAO:

```
public void insert(ObjetoModelo objetoModelo) throws SQLException {  
    PreparedStatement pstmt = null;  
    Vector parametros = new Vector();  
    try {  
        //Esta línea se debe repetir para cada uno de los atributos de la  
        clase modelo.  
        parametros.add(new ParametroQuery("CAMPO",  
objetoModelo.getCdCampo() ));  
        pstmt = helperBD.prepareFromFile(Conexion.getRutaSQL() +  
"dao/insert_ObjetoModelo.sql", parametros);  
        pstmt.execute();  
    }finally {  
        HelperBD.closeCursor(pstmt);  
    }  
}
```

En este método, como identificadores claves, se tiene, las palabras ObjetoModelo, CAMPO. La línea: `parametros.add(new ParametroQuery("CAMPO", objetoModelo.getCdCampo()));` se debe repetir para cada uno de los atributos de la clase modelo, reemplazando la palabra CAMPO, por el nombre del campo de la tabla, asociado al atributo de la clase modelo.

5.1.4. Restricciones, atributos y validaciones

En la construcción de código automático, siempre se deben tener en cuenta los atributos (variables), estos son los parámetros con los cuales se va a trabajar, a los cuales hay que pensar si se necesitan validaciones, y más aún, si con estos se van a realizar operaciones. Al momento de definir los atributos hay que tener muy presente las características principales de estos, por ejemplo, el tipo de dato, el tamaño máximo, su encapsulación y su valor inicial; el Excel que se debe cargar en el sistema debe cumplir con todas estas opciones para poder definir las clases y no tener atributos redundantes en los proyectos de Tecnofactor, para esto, la estructura definida para el archivo es la siguiente:

- Nombre atributo
- Nombre base de datos
- Tipo de dato clase
- Tipo de dato base de datos
- Tamaño base de datos
- Permite Null (Si, no)
- Es clave primaria (Si, no)
- Es secuencia (Si, no)
 - Nombre de la secuencia

Las validaciones se harán en el archivo *.js de la siguiente manera: En el proyecto base de Tecnofactor existe un método recursivo que nos ayuda a validar los atributos en una jsp, valida, la máscara (tipo de dato de que se quiere validar) y la obligatoriedad del atributo; esta función hace el proceso de validación recursivamente, se le pueden pasar todos los atributos de un formulario (jsp), con la máscara respectiva para cada atributo, una entrada true o false que indica si es o no obligatoria y un mensaje, el cual se debe mostrar en pantalla si alguno de los atributos no es correcto; esta función retorna un booleano el cual indica si, todos los atributos están correctos o no.

La función validarForma, es la siguiente:

```
//uso validarForma( forma , [ campo, mascara, esObligatorio,
mensaje] )
function validarForma( ){
var forma = validarForma.arguments[0];
var retorno = false;
for ( contador=1; contador < validarForma.arguments.length ;
contador++){
campo = eval ( "forma."+ (validarForma.arguments[contador]) );
```

```

if ( !campo ){
alert(Campo+' '+ validarForma.arguments[contador] + ' '+noExiste);
retorno = false;
break;
}else{
mascara = validarForma.arguments[++contador] ;
esObligatorio = validarForma.arguments[++contador] ;
mensajeRetorno = validarForma.arguments[++contador] ;
retorno = validarCampoForma( campo, mascara, esObligatorio ,
mensajeRetorno);

if ( !retorno ){
if ( campo.type != 'hidden' && !eval(campo.length) ){
campo.focus();
}
break;
}
}
}
return retorno;
}

```

5.1.5. Diseño del proyecto de generación de código

Ver anexos 9.1 y 9.2

6. Inconvenientes en el desarrollo de la práctica

Durante el proceso de desarrollo de la práctica empresarial no se presentaron inconvenientes, el proceso fue bastante bueno, con un acompañamiento continuo, por parte de todas las áreas de la empresa, apoyando el crecimiento profesional.

7. Conclusiones

- El diseño de un proyecto que ayuda a aumentar la productividad en los tiempos de desarrollo en la empresa, aplicando los conocimientos adquiridos durante nuestra formación profesional, y los adquiridos durante el proceso de práctica, nos permite crecer en nuestro proceso laboral, y demostrar nuestras capacidades y conocimientos.
- El desarrollo de proyectos de software, es una responsabilidad muy grande, ya que está en juego el nombre de la empresa, y el no

cumplimiento de los requerimientos del cliente, puede generar pérdidas millonarias a esta. Tecnofactor, al darnos la confianza de poder cumplir estos requerimientos y retribuirle esa confianza, nos ayuda en nuestro proceso de formación, tanto profesional como personal, gracias al conocimiento adquirido en cada uno de los proyectos y la realización de trabajo en equipo, para sacarlos adelante.

- La realización del proceso de práctica, nos ayudó a conocer nuevas tecnologías, que facilitan el proceso de análisis de desarrollo de software, realizando aplicaciones más estables y seguras, las cuales cumplen con las necesidades del cliente.
- De forma más general pensamos que el desarrollo de la práctica empresarial juega un papel muy importante para el crecimiento profesional de un ingeniero, por lo tanto consideramos que nuestra labor en la empresa, a nivel personal, fue gratificante por la experiencia alcanzada y la habilidad lograda para la solución y pronta respuesta a solicitudes que se presentan a diario en el medio laboral. Además, pertenecer a una empresa tan ambiciosa y prometedora como es Tecnofactor representa un gran punto a favor para nuestro desarrollo metodológico en Arquitectura de Software.

8. Recomendaciones

- Invertir más recursos humanos y/o recursos temporales para lograr el desarrollo completo del proyecto planteado.
- Determinar la mejor forma de desarrollar el proyecto teniendo en cuenta posibles cambios a futuro, lo cual facilitaría el mantenimiento.
- Implementar mejoras que permitan la adición de nuevas funcionalidades al proyecto, con operaciones más complejas dentro de los formularios como por ejemplo:
 - Buscadores con más criterios de filtros
 - Objetos manejados por sesiones
 - Manejo de relaciones entre objetos
- Cuando se realicen cambios en la arquitectura de Tecnofactor, también se debería ver afectada la plantilla base para que el proyecto se conserve a lo largo de la trayectoria de la empresa.
- Una vez desarrollado el producto de software, generador de código, desarrollar programas de capacitación dirigidos a los usuarios que van a interactuar con la herramienta, para lograr, de esta forma, un manejo fácil y rápido de todas sus funcionalidades. Todo esto con el fin de permitir una mayor productividad, estar a la altura de los tiempos, una agradable experiencia para el usuario y generar más competitividad.

9. Anexos

- 9.1. Documento de visión y alcance.
- 9.2. Documento de especificación de requisitos.
- 9.3. Plantilla Excel definida para la carga de atributos.
- 9.4. Diseño del proyecto (EAP)
- 9.5. Artículo

Referencias bibliográficas

- J. A. Carsí, I. Ramos, J. Silva, J. Pérez, V. Anaya, Un Generador Automático de Planes de Migración de Datos. Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia – España. Fecha publicación 1, Julio 2002. Fecha consulta. 15 de septiembre de 2011. Disponible en: <http://issi.dsic.upv.es/publications/archives/f-1050406290873/No1Art02.pdf>
- Andriano, Natalia Valeria, Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitación. Universidad Blas Pascal – Universidad Nacional de la Plata. Argentina. Fecha publicación: Agosto 2006. Fecha consulta Septiembre 28 de 2011. Disponible en: <http://postgrado.info.unlp.edu.ar/Carrera/Magister/Ingenieria%20de%20Software/Tesis/Andriano.pdf>.
- SARDUY DOMINGUEZ, Yanetsys. El análisis de información y las investigaciones cuantitativo y cualitativo, Escuela Nacional de Salud Pública [on line]. La Habana, 24 de Octubre del 2006 [Consultado 21 de Septiembre del 2010]. Disponible en: http://bvs.sld.cu/revistas/spu/vol33_3_07/spu20207.htm
- Tecnofactor S.A. Información corporativa. Envigado. [On line]. Consultado 13 de Octubre de 2011. Disponible en: <http://www.tecnofactor.com/>
- Tecnofactor S.A. Información Corporativa, Documentación Perfil analista. Envigado. Consultado 20 de Octubre de 2011. Disponible en la intranet de la empresa.
- Tecnofactor S.A. Información Corporativa, Manual Plantilla web. Envigado. Consultado 30 de Octubre de 2011. Disponible en la intranet de la empresa.