



Trabajo Final de Graduación para optar por el título

Bachiller en Ingeniería en Computación

Informe Final

Gustavo Adolfo Rodríguez Vargas

Carrera Ingeniería en Computación

Instituto Tecnológico de Costa Rica

Prof. Asesor: Oscar Víquez

Sede San Carlos

3 de Junio de 2011

Resumen Ejecutivo

Business Advance es un sistema financiero contable desarrollado en el año 2004 por la compañía GBSYS, convirtiéndose en uno de los principales productos de la compañía.

Durante los últimos años y tratando de satisfacer las necesidades de los clientes la empresa busca mejorar este sistema, migrando el producto a una plataforma más portable y que se pueda personalizar de acuerdo a los requerimientos de cada cliente.

Cabe destacar que dentro de los cambios realizados al producto y que lo hace más llamativo a los ojos de los consumidores, es el hecho de ser adaptable a los distintos servidores de aplicaciones y plataformas de bases de datos.

Actualmente éste se encuentra en una etapa inicial de migración, en la que se han desarrollado cuatro módulos: contabilidad, sistema, presupuesto y proveeduría.

Para el desarrollo de este proyecto se cuenta con la participación del Director de Proyectos, el Ing. Luis Emilio Ramírez Espinoza, un Coordinador Técnico en Arquitectura y Desarrollo de Aplicaciones WEB, el Ing. Pablo Peraza Vargas y un Equipo de Desarrollo constituido por estudiantes de práctica del Instituto Tecnológico de Costa Rica.

Es importante mencionar la experiencia tan significativa que se obtuvo durante éste proceso, ya que el hecho de poner en práctica los conocimientos adquiridos beneficia en la formación profesional y personal de cada uno de los estudiantes.

Tabla de Contenido

Resumen Ejecutivo	2
1. Contexto del proyecto.....	56
1.1 Descripción de la empresa.....	56
1.2. Visión de la empresa.....	67
1.3. Misión de la empresa.....	67
1.4. Objetivos de la empresa.....	67
1.5. Organigrama	78
1.6. Razón de ser de la empresa.....	89
1.7 Ubicación del departamento donde se realiza la práctica.....	89
1.7.1 Razón de ser del departamento	89
1.7.2 Cantidad de recurso humano en el departamento.....	89
1.8 Antecedentes del Proyecto.....	89
Descripción general del proyecto	940
Antecedentes del proyecto.....	940
2. Descripción del problema.....	1044
2.1 Personal involucrado	1044
2.2 Necesidades y Expectativas.....	1112
2.3 Requerimientos no funcionales	1213
2.4 Componentes y servicios	1314
3. Análisis de Riesgos.....	1617
4. Objetivos y alcances del Sistema	1819
4.1 Objetivo General.....	1819
4.2 Objetivos específicos.....	1819
4.3 Alcances del proyecto.....	1920
5. Solución Implementada	2021
5.1 Arquitectura conceptual de la solución.....	2021
5.1.1 Capa de Acceso a Datos.....	2021
5.1.2 Lógica del negocio.....	2223
5.1.3 Capa de presentación	2223
5.2 Los modelos de subsistemas.....	2425
5.3 Diagrama de Clases	2526
5.3.1 Diagrama de clases para Presupuesto	2526
5.3.2 Diagrama de clases para Proveeduría.....	2627
5.4 Interfaces de usuario	2728
5.5 Diseño de base de datos.....	2930

6. Conclusiones y comentarios	<u>3031</u>
Módulo Presupuesto.	<u>3031</u>
Módulo Proveduría.	<u>3031</u>
7. Referencias	<u>3233</u>

1. Contexto del proyecto

1.1 Descripción de la empresa

La empresa donde se desarrollará la práctica es Global Business System S.A (GBSYS). En la siguiente tabla se muestra la descripción de la naturaleza de la empresa GBSYS:

Nombre	GBSYS S.A
Cédula Jurídica	3-101-088600-22
Inscripción	Registro Público, Sección Mercantil, tomo 509, folio 186, asiento 159
Apartado Postal	Apartado 61-1002, San José, Costa Rica
Dirección	400 mts sur de la Pops de Curridabat, anexo edificio Trébol
Central telefónica	(506)2234-7876 (506)2234-2756
Facsímil	(506)2280-7542
Principales actividades	Desarrollo de sistemas. Administración de Bases de datos. Consultorías informáticas.

1.2. Visión de la empresa

La visión de la empresa es "ser reconocidos mundialmente como una corporación especializada en tecnología informática."

1.3. Misión de la empresa

La misión de la empresa es "ofrecer productos y servicios de la más alta calidad en tecnología informática que mejoren la eficacia, eficiencia y rentabilidad de nuestros clientes."

1.4. Objetivos de la empresa

Objetivo general

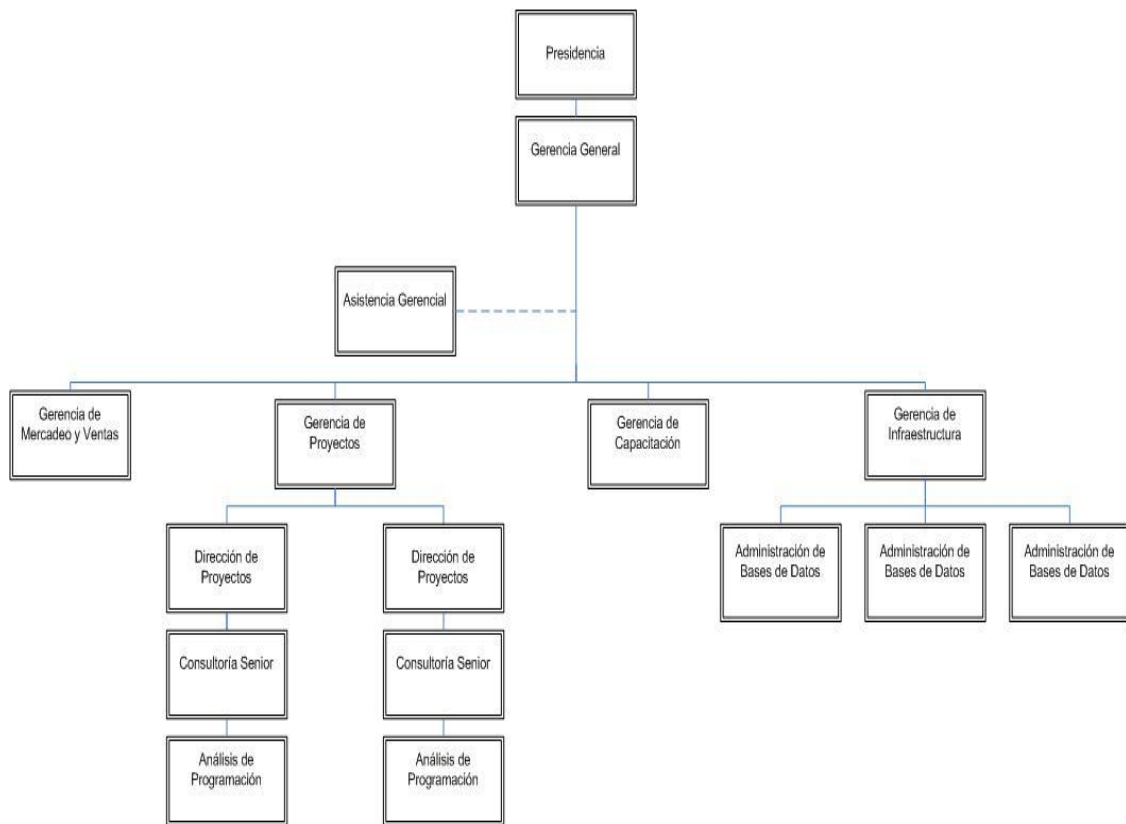
Ofrecer a los clientes de GBSYS, productos y servicios de alta calidad, que cumplan con sus necesidades y que se encuentren al alcance de su negocio.

Objetivos específicos

Entre los objetivos que se han establecido dentro de esta visión empresarial se destacan los siguientes:

- Desarrollar y dar mantenimiento a sistemas de cómputo de aplicación específica ajustados a los distintos niveles organizacionales y las condiciones particulares en que se desempeña cada usuario.
- Brindar entrenamiento en el manejo de equipo y paquetes de software en instituciones y empresas, por medio de seminarios, conferencias, charlas y cursos cortos.
- Brindar asesoría técnica en la selección y compra de equipo de cómputo.
- Desarrollar paquetes de software generalizados bajo estándares de calidad mundial.
- Distribuir software especializado de casas extranjeras de amplio reconocimiento en la industria.

1.5. Organigrama



Estructura organizacional

En cuanto a la estructura organizacional de la compañía se podría decir que es una organización de tipo funcional compuesta de la siguiente manera:

- Presidente
- Gerente General

Departamentos

- Asistente Administrativa
- Gerente de Ventas
- Gerente de Proyectos
- Gerente de Infraestructura
- Gerente de Capacitación

Actualmente la Asistente Administrativa y el Gerente de Ventas, conforman cada uno un departamento.

El Gerente de Proyectos tiene a cargo ocho Directores de Proyectos, según el organigrama. Estos a su vez tienen bajo su cargo a consultores seniors y a analistas programadores.

El Gerente de Infraestructura, tiene a su cargo a los Desarrolladores de Bases de Datos, que según lo muestra el organigrama son actualmente tres personas.

El Gerente de Capacitación, tiene a su cargo la conformación, elaboración y programación de los cursos de capacitación para clientes y empleados de la empresa. Los cursos de capacitación son referentes a las tecnologías que la empresa utiliza en los proyectos.

1.6. Razón de ser de la empresa

La razón de la empresa es ser una empresa especializada en Bases de Datos, y Desarrollo de Sistemas en ambientes ORACLE, Microsoft y Java. Adicionalmente se cuenta con una línea de distribución de productos de software especializados, incluyendo productos propios y otros ofrecidos por casas extranjeras reconocidas mundialmente.

También ofrecer servicios mediante la modalidad de outsourcing ya sea en soporte técnico para administración de sistemas y bases de datos, como para desarrollo de aplicaciones.

1.7 Ubicación del departamento donde se realiza la práctica

1.7.1 Razón de ser del departamento

La práctica supervisada se desarrollará en el departamento de gerencia de proyectos. La razón de ser del departamento es el de desarrollo, mantenimiento y consultoría de sistemas de información para el sector público y privado. Este departamento se centra en tecnologías Oracle, Java y Microsoft.

1.7.2 Cantidad de recurso humano en el departamento

La cantidad de recursos humanos en el departamento de gerencia de proyectos es de 42 integrantes.

1.8 Antecedentes del Proyecto

Descripción general del proyecto

En la actualidad existe un producto (ERP) de la empresa denominado *Business Advance* (Sistema Administrativo Financiero) con 4 instalaciones en diferentes clientes. *Business Advance* es una solución financiero administrativo orientada a optimizar la gestión de las unidades de negocio de la organización.

Business Advance es un ERP que está compuesto e integrado por los siguientes módulos:

- Seguridad
- Administración
- Contabilidad
- Presupuesto
- Control Bancario
- Caja Chica
- Emisión de pagos
- Cuentas por Cobrar
- Transportes
- Cuentas por Pagar
- Activos Fijos
- Inventarios
- Proveeduría
- Custodia de Valores
- Inversiones
- Facturación

Los objetivos principales de *Business Advance* para la **organización** son:

- Manejo de la información detallada y consolidada con el propósito de poder medir y controlar cada unidad de negocio.
- Mantener y mejorar su competitividad ante el dinámico ambiente que genera la globalización y la apertura de mercados.
- Tomar decisiones en tiempo real en sus actividades de negocio con el propósito de consolidarse y fortalecerse.
- Planificar, ejecutar y controlar con eficiencia sus proyectos corporativos.

Antecedentes del proyecto

El actual producto *Business Advance*, fue desarrollado hace aproximadamente 6 años utilizando la herramienta Oracle Developer Suite 9i. Posteriormente fue migrado utilizando Oracle Developer Suite 10g, debido a las necesidades y cambios tecnológicos, así como también a factores económicos, y requisitos de los clientes. Actualmente los clientes han mostrado gran interés en tener la funcionalidad de este producto en plataforma WEB utilizando J2EE. Con esto se pretende lograr que la aplicación sea más portable a otros servidores de aplicaciones a parte del OAS.

2. Descripción del problema

El producto Business Advance se encuentra desarrollado bajo un servidor de aplicaciones ORACLE. Los productos ORACLE, a pesar de ser herramientas líderes a nivel mundial para la gestión de Bases de Datos, y de brindar gran robustez y fiabilidad en sus sistemas, requiere de un alto costo para su obtención. Los elevados precios para la adquisición de licencias, han llevado a muchas compañías a optar por aplicaciones que trabajen sobre plataformas de menor costo.

Se requiere que el producto sea migrado nuevamente a un servidor de aplicaciones más portable, utilizando tecnologías actuales, esto con el fin de independizar la lógica del negocio con el servidor de Base de Datos, lo que permite así, que el producto se pueda ofrecer en distintos precios dependiendo de sus características y reduciendo costos. Esto Facilita además, que sean más las compañías, que puedan adquirir el producto

2.1 Personal involucrado

Entre las personas involucradas en el proyecto se encuentran las siguientes:

El Ing. Luis Emilio Ramírez Espinoza, Coordinador del Proyecto con conocimientos del producto que apoyará en las fases del proyecto, la planeación de las actividades y el control del cronograma.

El Ing. Pablo Peraza Vargas coordinador técnico en arquitectura y desarrollo de aplicaciones WEB.

Equipo de desarrollo conformado por tres estudiantes de práctica, Gustavo A. Rodríguez Vargas, Juan Diego Morera y Alberto Chaves que tendrán a cargo los dos módulos como objetivo común.

2.2 Necesidades y Expectativas

Necesidad	Problema que conlleva	Solución Actual	Solución que se propone
Independizar la lógica del negocio del sistema gestor de base de datos.	Migrar la totalidad de la lógica del negocio almacenada en procedimientos de base de datos y pasarlos al lenguaje al que se desea migrar.	Procedimientos almacenados encargados de gran parte de la lógica del negocio	Procedimientos elaborados en un lenguaje de alto nivel (java) encargados de la totalidad de la lógica del negocio
Migrar a tecnologías de desarrollo orientada a objetos como java, para agilizar el proceso de desarrollo del módulo actual y posterior.	Transcribir la lógica de los procedimientos en un lenguaje distinto		Utilizar un lenguaje de alto nivel, orientado a objetos como java, facilitando el desarrollo del mismo.
Reducir costos en adquisición de licencias, utilizando tecnologías de menor costo	Ajustar las características del producto actual, para que sea perfectamente adaptables a tecnologías de menor costo	El producto se encuentra elaborado bajo un servidor de aplicaciones ORACLE, utilizando además ORACLE Forms para la interfaz gráfica del cliente	Utilizar tecnologías como Java server faces, hibernate y java, que son tecnologías de código abierto, con soporte garantizando.

2.3 Requerimientos no funcionales

Escalabilidad

- El diseño debe contemplar el uso óptimo de recursos tales como conexiones a la base de datos, o Contemplar en el diseño la clara partición entre datos, recursos y aplicaciones para optimizar la escalabilidad del sistema.
- Debe contemplar requerimientos de crecimiento para usuarios tanto internos como externos.

Disponibilidad

- Debe contemplar requerimientos de confiabilidad y consistencia de los componentes de negocio ante recuperaciones. En caso de fallas de algún componente, no debe haber pérdida de información.
- Debe contemplar requerimientos de consistencia transaccional. Ante la falla del aplicativo, se debe contar con mecanismos que contemplen la interrupción de transacciones para que estas finalicen de manera correcta.

Mantenibilidad

- Se debe estructurar el código de una manera consistente y predecible.
- Para objetos que son frecuentemente manejados en la lógica del negocio, implementar las respectivas interfaces que aseguren su fácil implementación en el sistema.
- Asegurar que el diseño de las interfaces contemplen el que las propiedades públicas y los parámetros de los métodos sean de un tipo común (estandarizados).
- El sistema debe ser construido e implantado de tal manera que un cambio en los parámetros de negocio no obligue a la generación de una nueva versión del módulo.

Desempeño:

- La aplicación debe ofrecer un buen desempeño del sistema ante una alta demanda acorde a los requerimientos funcionales y no funcionales de la solución.

2.4 Componentes y servicios

Maven

Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant (y en menor medida a PEAR de PHP y CPAN de Perl) pero tiene un modelo de configuración de construcción más simple, basado en un formato XML [1]. Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos, componentes externos y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado [1].

Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) 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 [3].

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

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 Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL. [4]

Apache Tomcat

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache [5].

Base de Datos ORACLE

Oracle es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation [6].

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones
- Estabilidad
- Escalabilidad
- Soporte multiplataforma.

CSS

Las hojas de estilo en cascada (en inglés *Cascading Style Sheets*), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores [9].

La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un documento de su *presentación*

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local, que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

JQuery

jQuery es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC [10].

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

3.Análisis de Riesgos

Riesgo	Categoría	Posible causa	Impacto	Probabilidad ocurrencia	Exposición	Estrategia de mitigación	Estrategia de contingencia
Subestimación en las tareas que se establecen para el Spring semanal	Diseño	No se tienen requerimientos claros a la hora de establecer las tareas de la historia de usuario correspondiente	Atraso en el Spring semanal y por ende en el desarrollo del proyecto en general	Media		Para la mitigación de este riesgo, se hace un análisis semanal del esfuerzo de tareas anteriores y se estima el esfuerzo de la nueva tarea	Si la tarea se ha subestimado se aplica más esfuerzo a dicha tarea para finalizar en el Spring acordado, esto quiere decir que la tarea que antes le correspondía a un solo desarrollador ahora pasaría a resolverse por dos desarrolladores
Perdida de la conexión del servidor de base de datos del sistema	Tecnológico	El servidor en el que se encuentra la base de datos del sistema se encuentra en otro departamento de la empresa que no es el departamento de desarrollo (soporte) y se utiliza para otra pruebas	Atraso en el Spring semanal y por ende en el desarrollo del proyecto en general	Media		Para la mitigación de este riesgo, se deben de establecer tareas que no precisen de la base de datos del sistema, por lo tanto ante cualquier caída de la conexión se atacan este tipo de tareas.	En el caso de no contar con el servidor de base de datos, se atacan las tareas que no requieren de esta conexión, como la creación de interfaz de usuario, etc.
Tareas pendientes en el cierre del Spring semanal	Tecnológico	No se tienen requerimientos claros a la hora de establecer las tareas de la historia de usuario correspondiente	Atraso en el Spring semanal y por ende en el desarrollo del proyecto en general	Baja		Para la mitigación de este riesgo, se hace un análisis semanal del esfuerzo de tareas anteriores y se estima el esfuerzo de la	Se establece en el siguiente Spring las tareas que quedaron pendientes con una mayor prioridad ya que correspondían a

						nueva tarea	tareas pasadas
Aparición de nuevas tareas no establecidas y que son requisito para la tarea que se está resolviendo	Tecnológico y diseño	No se tienen requerimientos claros a la hora de establecer las tareas de la	Atraso en el Spring semanal y por ende en el desarrollo del proyecto en	Baja		Cada semana se hace un estudio de los requisito en general, para verificar la dependencia de nuevas tareas con otras posibles a futuro	Se establece en el siguiente Spring las tareas de este tipo con una mayor prioridad ya que muchas dependen de estas
Atraso en la reuniones de capacitación con el coordinador del proyecto	Personas		Atraso en el desarrollo del producto y en el establecimiento de requisitos	Media			Se atacan los requisitos e historias de usuario que se tienen claras al momento y se dejan pendientes las que necesiten capacitación por parte del coordinador

4. Objetivos y alcances del Sistema

4.1 Objetivo General

- Migrar los Módulos de Presupuesto y Proveeduría del producto *Business Advance* de la empresa GBSYS a una plataforma WEB.

4.2 Objetivos específicos

Los objetivos son enumerados siguiendo un orden cronológico.

1. Capacitar al equipo desarrollador en las herramientas que se utilizarán en el desarrollo del sistema por medio de tutoriales, libros y otros medios para el uso correcto de las mismas.
2. Capacitar al equipo desarrollador en la funcionalidad actual del Módulo de Presupuesto y Proveeduría de *Business Advance* con el propósito de que conozcan el producto.
3. Migrar el módulo de Presupuesto de Business Advance que están actualmente desarrollado con la herramienta Oracle Developer Suite 10g a plataforma WEB utilizando J2EE.
 1. Contar con una característica que permita la gestión de Formularios Presupuestarios.
 2. Contar con una característica que permita la gestión de Documentos Presupuestarios.
 3. Contar con una característica que permita la gestión de Movimientos Presupuestarios.
 4. Contar con una característica que permita la gestión de Modificaciones al Presupuesto.
4. Migrar el módulo de Proveeduría de Business Advance que están actualmente desarrollado con la herramienta Oracle Developer Suite 10g a plataforma WEB utilizando J2EE.
 1. Contar con una característica que permita la gestión de solicitudes de suministros
 2. Contar con una característica que permita la gestión de control de concursos del modulo de proveeduría
 3. Contar con una característica que permita la gestión de órdenes de compra
 4. Contar con una característica que permita la gestión de proveedores.

4.3 Alcances del proyecto

- **Presupuesto**
 - El sistema permitirá el registro de nuevos formularios presupuestarios para el próximo año en proceso, además de permitir aprobar y aplicar los mismos.
 - El sistema permitirá el ingreso de documentos presupuestarios, que modificarán el presupuesto dentro del año en proceso.
 - El sistema podrá generar movimientos presupuestarios a partir de documentos y formularios aplicados, y que se verán reflejados en las cuentas presupuestarias.
 - El sistema permitirá realizar modificaciones del presupuesto, modificando las cuentas presupuestarias.

- **Proveeduría**
 - El sistema podrá registrar solicitudes de suministro para la compañía.
 - El sistema permitirá registrar y gestionar el control de concursos.
 - El sistema podrá registrar y emitir nuevas órdenes de compra.
 - El sistema podrá llevar un control de proveedores, con sus características, y ayudar en la toma de decisiones a la hora de seleccionar un proveedor en particular.

5. Solución Implementada

5.1 Arquitectura conceptual de la solución

El nuevo producto Business Advance está construido bajo una plataforma de J2EE [7], la cual es una plataforma de programación, parte de la Plataforma Java [2], para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares, ejecutándose sobre un servidor de aplicaciones.

El hecho de estar diseñado bajo esta plataforma constituye una gran ventaja para el producto, ya que Java EE incluye varias especificaciones de API, tales como: JDBC, RMI, e-mail, JMS, Servicios Web, XML que pueden tomarse como punto de partida para el desarrollo del proyecto además de permitir una escalabilidad del producto.

El proyecto se encuentra dividido en 3 capas:

- Capa de acceso a datos
- Lógica del negocio
- Capa de Presentación

5.1.1 Capa de Acceso a Datos

La capa de acceso a datos para Business Advance, a diferencia de otros productos de la empresa GBSYS, no utiliza "jdbc" para el acceso a base de datos. Se optó por implementar Hibernate [3], una nueva opción que la compañía está utilizando en algunos proyectos y que ha resultado exitoso para nuevos productos que se están desarrollando actualmente.

Hibernate [3] 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 (Clases en Java encargadas de conectar la capa lógica con la capa de presentación) de las entidades que permiten establecer estas relaciones.

En otras palabras, el modelo de base de datos es representado por el modelo de objetos; únicamente el desarrollador se limita a indicar cuál será la relación de

los objetos entity (entidades del modelo) Java e Hibernate se encargará de interpretarlos e integrarlos a la base de datos, asignando las relaciones entre tablas de una manera automática.

La clase “ControlGenerico” se encarga de llamar a un archivo de configuración “persistence” el cual le indica el datasource de la base de datos a la que se desea conectar. Es ahí cuando las clases Entity definidas por el desarrollador son interpretadas por Hibernate y éste las configurará como una tabla dentro de la base de datos indicada.

Ejemplo de una Clase Entity:

```
/**
 *
 * @author Gustavo A. Rodríguez Vargas
 */
@Entity
public class CatalogoConcurso implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String cod_tipo;
    private String descripcion;
    private String cod_siac;
    // private List

    @Transient
    private boolean selected;
    @OneToMany(mappedBy = "concurso", cascade=CascadeType.ALL, fetch=FetchType.EAGER)
    private List<DetalleConcurso> detalleConcursos;

    public CatalogoConcurso() {
        this.detalleConcursos = new ArrayList<DetalleConcurso>();
    }

    public CatalogoConcurso(String cod_tipo, String descripcion, String cod_siac) {
        this.cod_tipo = cod_tipo;
        this.descripcion = descripcion;
        this.cod_siac = cod_siac;
        this.detalleConcursos = new ArrayList<DetalleConcurso>();
    }
}
```

El objeto al cual se relaciona

Se establece la relacion 1:N, N:1, N:N, 1:1

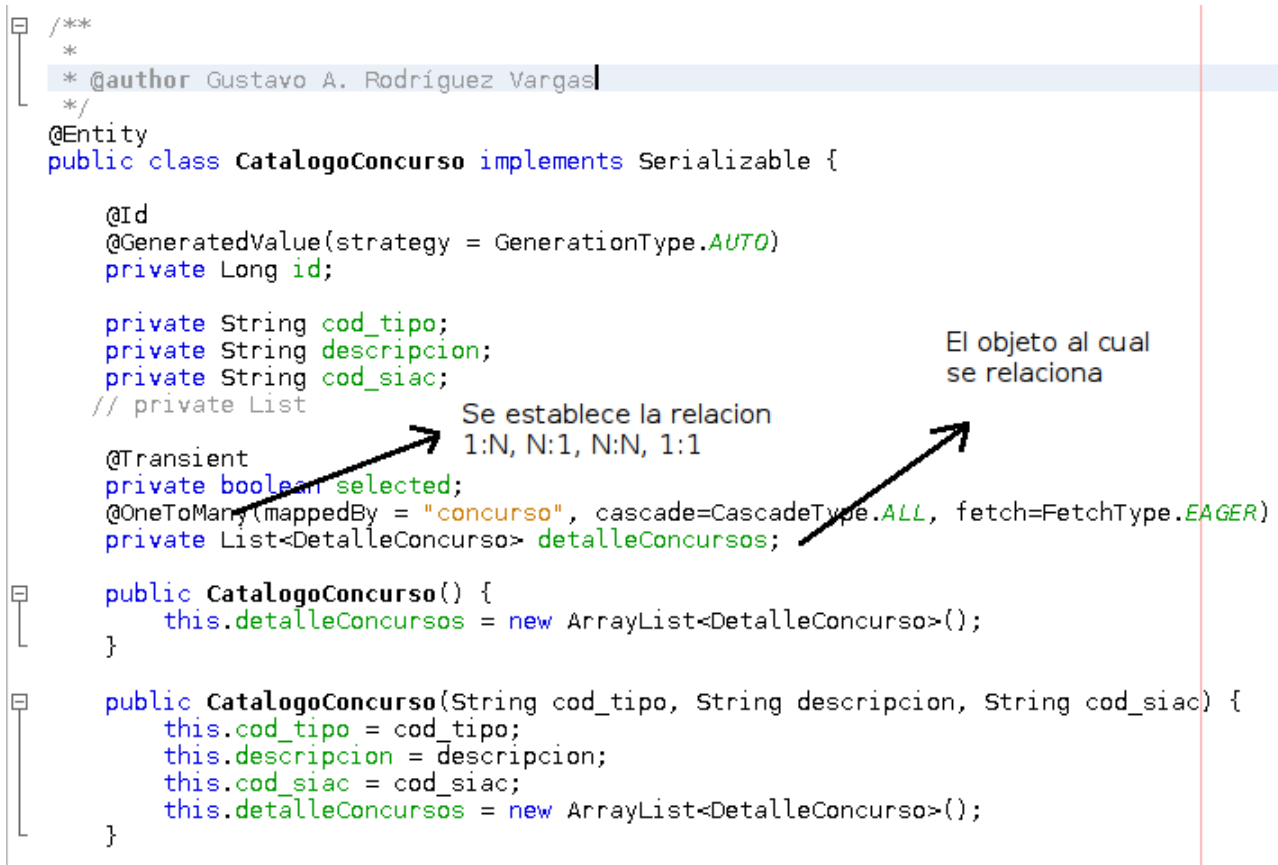


Ilustración 1: Ejemplo de Entity

5.1.2 Lógica del negocio

La lógica del negocio se encuentra desarrollada dentro de los paquetes de vistas del proyecto. Los llamados “Beans” [4, JSF] quienes son los que contendrán los métodos y algoritmos del proyecto. Estos “Beans” (archivos .java) son sumamente sencillos de implementar y conectar con la capa de presentación. El objetivo de la utilización de Java Server Faces, es permitir esa facilidad de enviar y recibir mensajes captados por la interfaz de usuario, y agilizar el proceso de desarrollo.

5.1.3 Capa de presentación

Por último tenemos la capa de presentación que está constituida por archivos XHTML y el framework Java Server Faces para la implementación de la Interfaz de Usuario, junto con componentes como “CSS” [9] y JQuery [10] para efectos y eventos del diseño de interfaz.

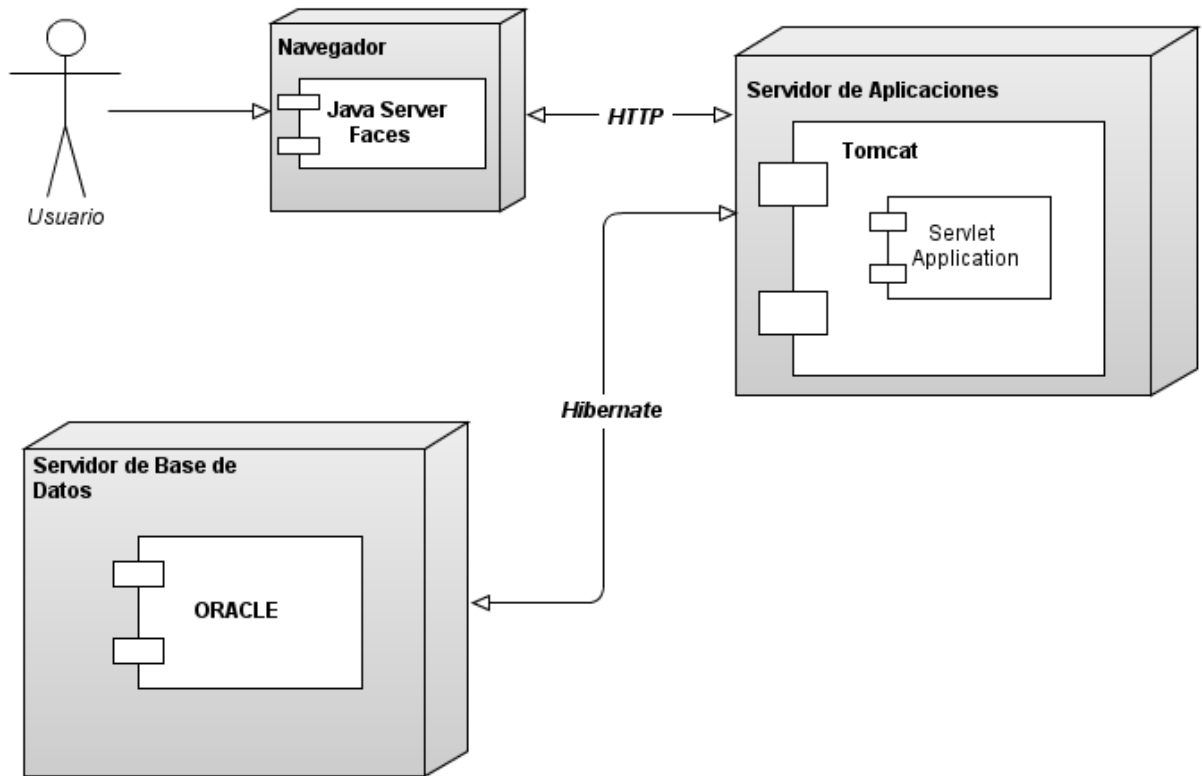


Ilustración2 Diagrama General del Proyecto

5.2 Los modelos de subsistemas

Estructura de los Paquetes

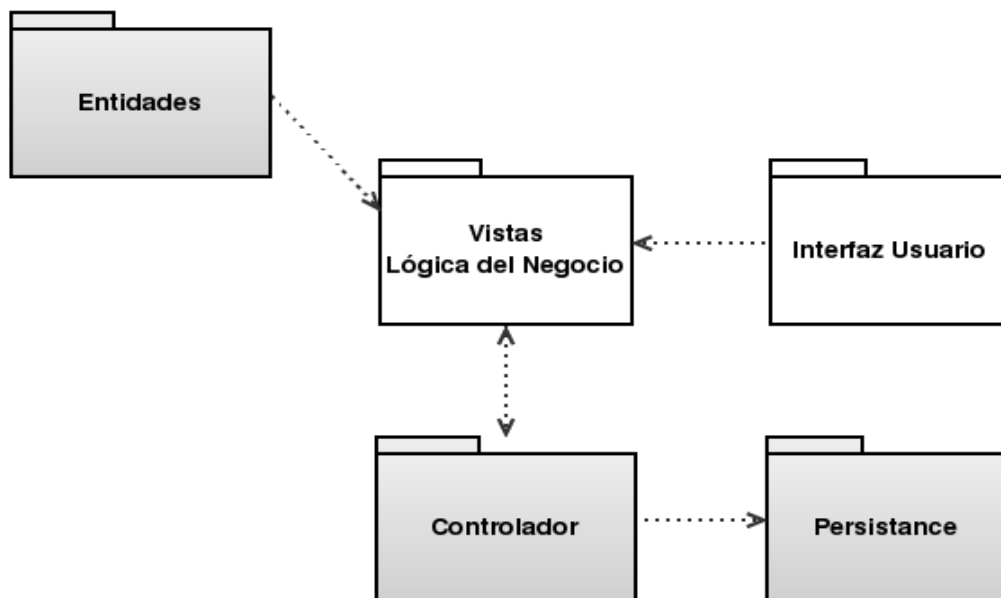


Ilustración3 Estructura de los Paquetes

Entidades

Son el modelo de Objetos que al final serán representados en el servidor de base de datos con sus respectivas relaciones entre ellas.

Vistas

Clases en .Java encargada de procesar la información proveniente de la capa de presentación y de enviar una respuesta. Las vistas hacen llamadas a las entidades para operar con ellas y la capa de presentación hace llamadas a estas vistas.

Controlador

El controlador es el encargado de llamar al Persistencee para todos los procesos que tengan que ver con llamadas a base de datos.

Persistencee

Archivo XML encargado de amarrar la aplicación con la base de datos a través de Hibernate [3].

Interfaz de Usuario

Plantillas XHTML (Interfaz de Usuario) de la capa de presentación.

5.4 Interfaces de usuario

Java Server Faces(JSF) es una tecnología y framework para aplicaciones Javabasadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa Java Server Pages (JSP)como la tecnología que permite hacer el despliegue de las páginas [4]

Todas las plantillas de interfaz están desarrolladas bajo este Framework, ya que aporta gran cantidad de componentes y facilidades para los desarrolladores, además de ser sencillo y estructurado.

El diseño de la interfaz esta básicamente constituido por una plantilla base, que a su vez contiene los encabezados que debe llevar todas las páginas, el menú respectivo de opciones que presta el producto y un área de tronco.

Es en el área del tronco donde se encuentran contenidas todas las distintas páginas del Business Advance. Únicamente esta porción de la interfaz es lo que cambia durante la navegación entre páginas.

Cada página hereda la plantilla base, con su encabezado y menú.



Ilustración 6: Plantilla Base



Ilustración7 Pagina Estándar

Todas las páginas son archivos .XHTML que únicamente se mostrarán en el área del tronco de la plantilla. El usuario no sentirá una recarga de la página ya que solamente el área indicada será la que cambie.

Nota: El diseño global de la interfaz actual, solo es una versión previa, ya que el diseño final (estilos, CSS) son aportados por un diseñador aparte cuando todos los módulos del producto se encuentren implementados. El diseño actual es únicamente una base para el desarrollo del producto.

5.5 Diseño de base de datos

Como se menciona en el punto [2.1.1](#); Hibernate [3] se encarga de convertir el modelo de objetos en un modelo de base de datos. Por lo tanto, será el encargado de asignar las llaves necesarias entre las tablas de una manera automática según las relaciones que existen entre los objetos Entidad.

El objetivo de la utilización de este servicio es independizar en su totalidad la lógica del negocio con la base de datos. Al contener el diseño en el modelo de objetos no es imprescindible un tipo específico de servidor de base de datos, ya que Hibernate [3] se adapta fácilmente a la gran mayoría de gestores de bases de datos comerciales actualmente: MySQL, Postgres, SQL, Oracle, otros.

La solución final hasta el momento, después de finalizada la práctica de especialidad, está constituida por cuatro módulos: contabilidad, sistema, proveeduría y presupuesto. A partir de esta solución, los siguientes desarrolladores se encargarán de ir implementando los módulos restantes, según el patrón de diseño y las tecnologías que se han utilizado para desarrollar el producto.

6. Conclusiones y comentarios

Módulo Presupuesto.

Para el producto desarrollado del módulo presupuesto, se logró implementar toda la funcionalidad requerida para el manejo de formularios presupuestarios. Éste es uno de los puntos claves del modelo, ya que es a través de estos formularios que se maneja el movimiento de fondos presupuestarios. El proyecto es capaz de registrar formularios presupuestarios ordinarios o extraordinarios, además de presentar un mantenimiento de los mismos (edición y borrado).

Además de la funcionalidad elaborada para los formularios presupuestarios, se integró el manejo de documentos y modificaciones presupuestarias exitosamente. Éstos son elementos importantes para el manejo del presupuesto extraordinario, ya que los mismos se encargan de gestionar el presupuesto de una manera distinta del flujo normal. Por ejemplo: si es necesario un cambio de fondos del presupuesto entre las cuentas durante el año en proceso, el producto cuenta con las herramientas necesarias para manejar estos tipos de casos.

Por último, todo formulario, modificación o documento genera movimientos presupuestarios. A través de estos movimientos, es como se mueve el saldo entre las cuentas y el sistema desarrollado cuenta con esta funcionalidad.

Módulo Proveeduría.

El módulo de proveeduría es uno de los más extensos del sistema. Al inicio de la práctica se estableció que dentro de las características más prioritarias para éste, era la capacidad de gestionar solicitudes de suministro, información de proveedores; además de llevar un control de calificaciones y órdenes de compra

Los objetivos logrados dentro de este módulo son:

- La capacidad de registrar solicitudes de suministro.
- La gestión del catálogo de proveedurías, casas extranjeras, Objetivos del Plan Anual Operativo, clasificación de servicios y calificaciones. Estos se encontraban fuera de los objetivos previamente establecidos pero que se lograron desarrollar para mejorar la funcionalidad del sistema.
- El módulo cuenta con las características de gestionar la información de los proveedores.
- La capacidad de llevar un control de concursos, según los requerimientos del sistema.
- La capacidad de registrar nuevas órdenes de compra.

Como se detalló anteriormente, todos los objetivos planteados para el módulo de presupuesto durante el inicio de la práctica de especialidad, se concluyeron de manera satisfactoria.

Dentro del módulo de proveeduría, los objetivos fueron concluidos casi en su totalidad. Ya que hicieron falta pequeños procesos o funcionalidades que se solicitaron para el nuevo sistema y que no se encontraban en el producto anterior. Esto se debió principalmente a que la prioridad era conseguir las mismas características del producto anterior con la migración. Concluido esto, se procedería a integrar los nuevos cambios o requerimientos en la versión nueva.

Todo el proyecto de desarrollo (fuentes), fue entregado a la empresa. Este proyecto será continuado por otros estudiantes, que se encargarán de continuar con los módulos restantes.

Por último, uno de los puntos más importantes que se logró durante el tiempo de trabajo en la práctica de especialidad, fue la experiencia y el conocimiento obtenido no sólo en las tecnologías que se utilizaron, sino también en el aprendizaje sobre el manejo del presupuesto y la proveeduría de una compañía.

La metodología utilizada durante el proceso de desarrollo llamada SCRUM, permitió conocer y aprender sobre el trabajo de equipo en una compañía y la importancia que tiene para con la misma. Esta técnica permitió además, obtener las bases sobre la administración de proyectos y la autogestión en un proceso de desarrollo. Por otro lado, reflejo la importancia de una buena organización en un equipo de trabajo.

7. Referencias

- [1] Maven. Wikipedia, consultado el 17 de marzo de 2011. Disponible en: <http://es.wikipedia.org/wiki/Maven>
- [2] ¿Qué es la tecnología Java y por qué lo necesito? Consultado el 17 de marzo de 2011. Disponible en: http://www.java.com/es/download/faq/whatis_java.xml
- [3] Hibernate. Wikipedia, consultado el 17 de marzo de 2011. Disponible en: <http://es.wikipedia.org/wiki/Hibernate>
- [4] JSF. Wikipedia, consultado el 17 de marzo de 2011. Disponible en: http://es.wikipedia.org/wiki/JavaServer_Faces
- [5] Tomcat. Wikipedia, consultado el 4 de Mayo de 2011. Disponible en: <http://es.wikipedia.org/wiki/Tomcat>
- [6] Base de Datos Oracle. Wikipedia, consultado el 4 de Mayo de 2011. Disponible en <http://es.wikipedia.org/wiki/Oracle>
- [7] Java EE. Wikipedia, consultado el 4 de Mayo de 2011. Disponible en: http://es.wikipedia.org/wiki/Java_EE
- [8] Diseño de Sistemas. Wikipedia, consultado el 4 de Mayo de 2011. Disponible en: http://es.wikipedia.org/wiki/Dise%C3%B1o_de_sistemas
- [9] Hojas de estilo en cascada. Wikipedia, consultado el 4 de Mayo de 2011. Disponible en: http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada
- [10] JQuery. Wikipedia, consultado el 4 de Mayo de 2011. Disponible en: <http://es.wikipedia.org/wiki/JQuery>