



**Trabajo Final de Graduación para optar por el título  
Bachiller en Ingeniería en Computación**

**Sistema Financiero-Contable**

**Avance**

**Alberto Chaves Villegas**

**Instituto Tecnológico de Costa Rica**

**Carrera Ingeniería en Computación**

**Sede San Carlos**

**Prof. Asesor: Oscar Víquez**

**22/03/2011**

## Resumen

El presente documento respalda un proceso de práctica de especialidad, que fue realizado en la empresa GBSYS ubicada en Curridabat, ciudad capital. En términos generales el proyecto consistió en la migración de un sistema construido en la tecnología Oracle Forms hacia la tecnología Java Web, por lo que se trata de un sistema que se despliega en un navegador, como cualquier otra página web. Específicamente la migración consistía en dos módulos, de los dieciséis en los que está constituido el sistema más antiguo, llamados presupuesto y proveeduría. Para completar estos módulos fue necesario un entrenamiento tanto en la herramienta a utilizar como también en entender la lógica del proceso en sí, para atacar este punto realizamos varias reuniones con algunos expertos en el tema, tanto de proveeduría como de presupuesto, subsanando las dudas que se presentaran en su momento.

El producto final de esta práctica, fue la conclusión de ambos módulos, estos corriendo sobre la plataforma que estaba utilizando el módulo de contabilidad, un sistema web construido con la tecnología Java2EE, cabe destacar que según la teoría la migración debe ser construir el sistema migrado exactamente igual al sistema anterior, sin embargo este proceso fue muy diferente, ya que, además de la migración también fue necesario programar ciertas funcionalidades extras que se encontraron en las reuniones, esto con el fin de que los módulos estuvieran más acordes a las necesidades actuales del usuario.

## Tabla de contenido

Resumen.....	2
Descripción del problema .....	4
Razón de ser de la empresa .....	4
Organigrama de la empresa.....	5
Departamento en el cual se desarrollará.....	5
Visión.....	6
Descripción.....	6
Productos Realizados .....	7
Personal Involucrado .....	7
Análisis de Riesgos .....	8
Objetivos del proyecto .....	9
Objetivo General .....	9
Objetivosespecíficos .....	9
Tecnologías/Herramientas.....	9
Solución Implementada .....	10
Arquitectura Conceptual de la Solución .....	10
Tecnologías/Herramientas.....	10
Modelos de subsistemas.....	10
Diagrama de Clases .....	11
Interfaces de usuario .....	18
Componentes y Servicios .....	19
Diseño de Bases de Datos .....	21
Sobre la aplicación .....	23
Conclusiones y Comentarios .....	26
Informes de Avance .....	<b>¡Error! Marcador no definido.</b>
Bibliografía .....	27

## Descripción del problema

### Contexto del proyecto

En un mercado cada vez más cambiante, que de la mano de la tecnología, avanza a pasos de gigante, en un campo donde las innovaciones tardan en quedar obsoletas lo que tarda la tierra en darle media vuelta al sol, en un campo laboral cada vez más competitivo, donde la creatividad, la innovación y la simplicidad son piezas claves para alcanzar el éxito, es necesario estar actualizado y saber cambiar.

El proyecto nace con esta visión. En la actualidad, gran cantidad de empresas usan sistemas de información para la administración de su negocio, desde el área contable, presupuesto, planillas, cuentas por pagar hasta para otros procesos menos cotidianos o comunes, que también podrían ser administrados mediante los sistemas.

Precisamente el objetivo del proyecto fue migrar un sistema que está construido en una tecnología vieja llamada Oracle Forms hacia una tecnología más innovadora, que utiliza la plataforma web para ejecutarse y que es fácil para el programador, llamada Java EE.

El sistema que se migró está orientado a la empresa pública de nuestro país, muchas de sus características están programadas respetando los lineamientos legales que hay en el país para la ejecución de estos procesos, que todas las empresas del estado deben realizar. Estos procesos podrían ser tanto la gestión de presupuesto, proveeduría, inventario, cuentas por pagar, cuentas por cobrar, etc. Y específicamente en nuestro proyecto se migraron dos módulos llamados: proveeduría y presupuesto.

### 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.

## Organigrama de la empresa

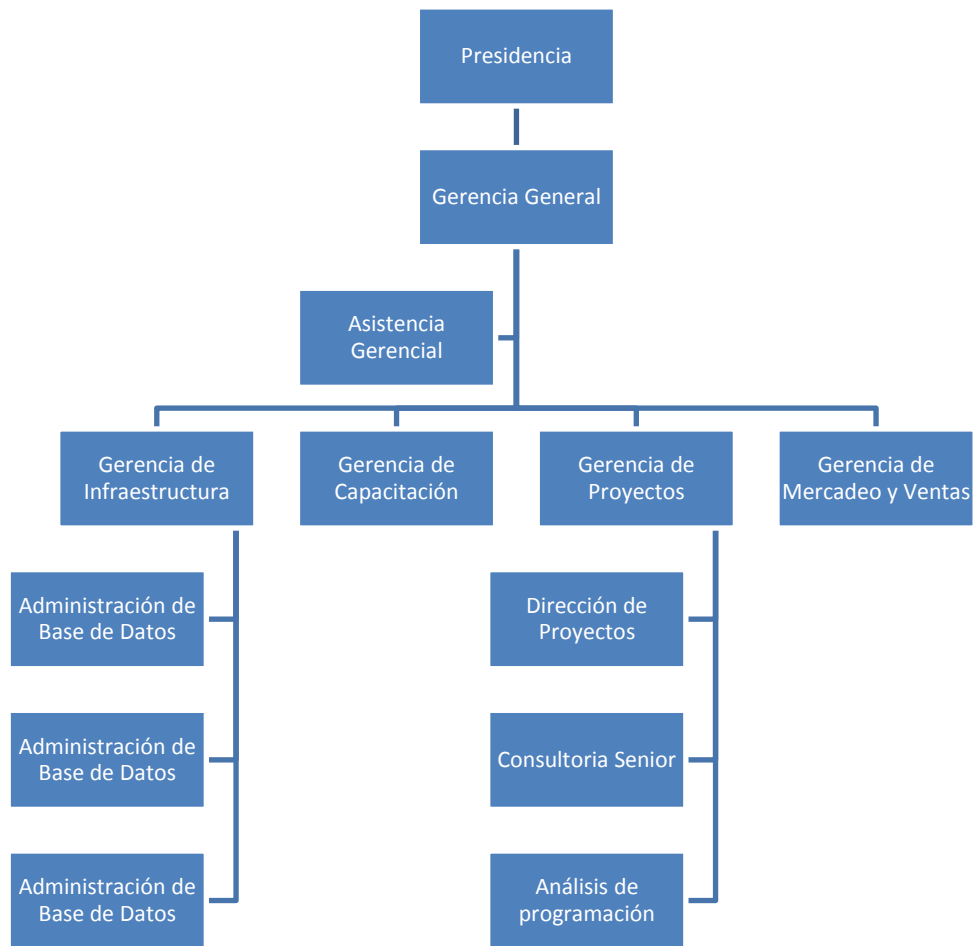


Ilustración 1 - Organigrama provisto por la gerencia de la empresa

### Departamento en el cual se desarrollará

La práctica supervisada se desarrolló 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. La cantidad de recursos humanos en el departamento de gerencia de proyectos es de 42 integrantes.

### Visión

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

### Descripción

El proyecto consiste, como vemos, en una migración desde una tecnología antigua como lo es Oracle Forms, a una tecnología más actual y que además es web, como lo es Java EE, más específicamente el proyecto abarcó la migración de los módulos de:

- Presupuesto
- Proveeduría

## Productos Realizados

Entre los productos realizados en esta práctica se encuentran los siguientes:

- Documentación de la arquitectura a utilizar.
- Módulo de Proveeduría Parte I desarrollado en J2EE (fuentes y ejecutables).
- Pruebas de funcionalidad y rendimiento de los módulos migrados.
- Manual de usuario y técnico de ambos módulos.
- Capacitación en el uso de los módulos migrados.

## Personal Involucrado

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

- Un Coordinador del proyecto con conocimientos del producto que apoyará en las fases del proyecto, la planeación de las actividades y el control del cronograma.
- Un líder técnico en arquitectura y desarrollo de aplicaciones WEB.
- Tres estudiantes de práctica que formarán parte del equipo desarrollador.

Este proyecto se dividió en varias fases: la primera fase fue la migración del módulo de contabilidad realizada también por estudiantes de práctica, la segunda fase, consta de los dos módulos antes mencionados presupuesto y proveeduría y se espera que una tercera fase sea realizada también por otros estudiantes de práctica.

La ventaja de incursionar en este proyecto en la segunda fase, fue que precisamente muchas funcionalidades llamémoslas base, ya estaban programadas, es por ello que la mayor cantidad de trabajo estaba orientada a desarrollar los procesos del negocio específicos de cada módulo.

## Análisis de Riesgos

Ante la conclusión del proyecto, se han enfrentado todos los riesgos que se establecieron al inicio, sin embargo, estos no afectaron tanto el desarrollo porque para la elaboración del proyecto se están utilizando automáticamente procedimientos que se ajustan a las estrategias de evasión o planes de contingencia de cada uno de los riesgos, así que los impactos han sido casi imperceptibles.

Tipo de riesgo	Descripción	Estrategia de evasión/mitigación	Plan de contingencia.
Técnico	El equipo de cómputo deja de funcionar por alguna razón desconocida. (Causa) Los equipos de cómputo están viejos y sin mantenimiento.	<ol style="list-style-type: none"> <li>1. Realizar mantenimientos preventivos periódicos a los equipos a nivel de software.</li> <li>2. Realizar limpieza del equipo a nivel de hardware.</li> <li>3. Realizar respaldos periódicos de la información más crítica.</li> </ol>	1. Contar con repuestos y herramientas básicas para la reparación de equipos, o en su defecto, contar con otro equipo de cómputo igual y restaurar la copia de seguridad.
Técnico	El servidor de base de datos se encuentra fuera de línea. (Causa) 1. Múltiples usuarios haciendo uso de las mismas máquinas virtuales. 2. Múltiples máquinas virtuales haciendo uso de la misma máquina host.	<ol style="list-style-type: none"> <li>1. Mantener constante comunicación con los encargados de administrar el centro de servidores.</li> <li>2. Mantener mayor independencia entre máquinas virtuales.</li> </ol>	1. Contar con una base de datos local, en la que se pueda seguir trabajando sin atrasos.
Humano	Alguna de las personas del grupo de trabajo se ausenta. (Causa) Enfermedad o dolencias físicas.	<ol style="list-style-type: none"> <li>1. Mantener un seguimiento de las actividades que está realizando cada miembro.</li> <li>2. Programar utilizando el Javadoc y demás documentación para facilitar entender el código.</li> </ol>	1. Si el riesgo presentara un atraso real, se debe contar con otro miembro que pueda continuar el proyecto.
Humano	Dedicación a otras actividades, que no son parte del proyecto, por órdenes de algún superior.	1. Asegurarse de contar con personal extra para estas actividades.	1. Valorar la posibilidad de extender la entrega del proyecto.

Con la conclusión del proyecto, se identifica claramente que la gestión de riesgos se debe realizar para cada proyecto en particular, debido a que todos los días se enfrenta algún riesgo y que mediante los planes de mitigación, estos se logran reducir; sin embargo, el carecer de una estrategia de mitigación podría llevar a causar atrasos significativos del proyecto.



## Objetivos del proyecto

### Objetivo General

- Migrar el Módulo de Proveeduría Parte I del producto Business Advance de la empresa GBSYS a una plataforma WEB utilizando J2EE.

### Objetivos específicos

- Capacitar al equipo desarrollador en la funcionalidad actual del Módulo de Proveeduría Parte I de Business Advance con el propósito de que conozcan el producto.
- 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.
- Migrar el módulo de Presupuesto Parte I<sup>1</sup> de Business Advance que están actualmente desarrollado con la herramienta Oracle Developer Suite 10g a plataforma WEB utilizando J2EE.
- Migrar el módulo de Proveeduría de Business Advance que están actualmente desarrollado con la herramienta Oracle Developer Suite 10g a plataforma WEB utilizando J2EE.

### Tecnologías/Herramientas

Entre las tecnologías que se van a utilizar para el desarrollo del proyecto se encuentran las siguientes:

- IDE NetBeans 6.8
- Framework de presentación Java ServePages (JSF) y ICEfaces
- Framework de persistencia de datos Hibernate
- Base de Datos Oracle 10g
- Servicios Web
- SubVersion (Control de versiones)
- Redmine (Control de proyectos)
- PlanningPoker (Estimación de historias de usuario) *[Cita 4]*

---

<sup>1</sup> La parte I de este módulo comprende los submódulos de Configuración, Solicitudes de Compra y sus respectivos reportes.

## Solución Implementada

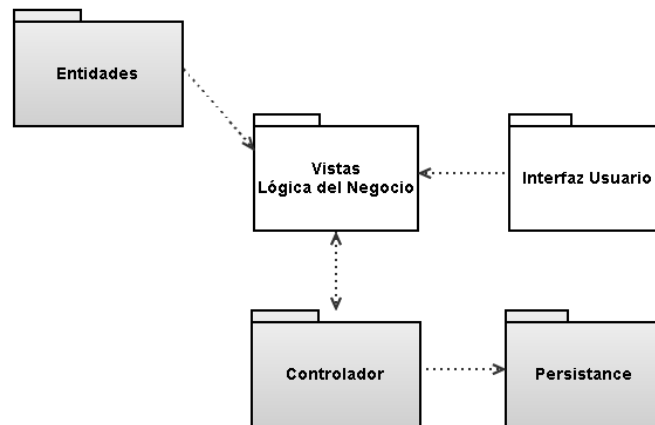
### Arquitectura Conceptual de la Solución

#### Tecnologías/Herramientas

A continuación se detallan las diferentes herramientas que se utilizaron para la generación del proyecto.

### Modelos de subsistemas

Estructura de los Paquetes



**Ilustración 2 - Muestra las diferentes partes que se utilizan en el desarrollo de alguna característica.**

Haciendo uso de la tecnología de persistencia de datos, el modelo de desarrollo que seguimos, para cada página web en la aplicación, se ve reflejada en la ilustración 1. Se crea una entidad que es lo mismo que una tabla en la base de datos, luego se crea una vista, las vistas contienen instancias de una o más entidades y también son clases de java. La vista\_es la que se encarga de enlazar las entidades a la interfaz del usuario, que en este caso son archivos “xhtml” con componentes Icefaces, mediante los cuales se crea la relación.

Cada entidad para que sea guardada en la base de datos, debe ser enviada a guardar en el controlador, en este punto la herramienta de persistencia de datos realiza la conversión de los campos de las clases y las almacena en la base de datos, que se tenga configurada en el archivo de persistencia.





## Diagrama del Modulo de Proveeduría 1:



Ilustración 5 - Muestra el diagrama de clases de proveeduría parte 1

Entre las principales entidades que se aprecian en el diagrama están: Solicitudes de Suministros, Empleados, Compra Institucional, Bancos y el Catálogo del PAO.

## Diagrama del Modulo de Proveeduría 1:



Ilustración 6 - Muestra el diagrama de clases de proveeduría parte 1



## Diagrama del Modulo de Proveduría 2:

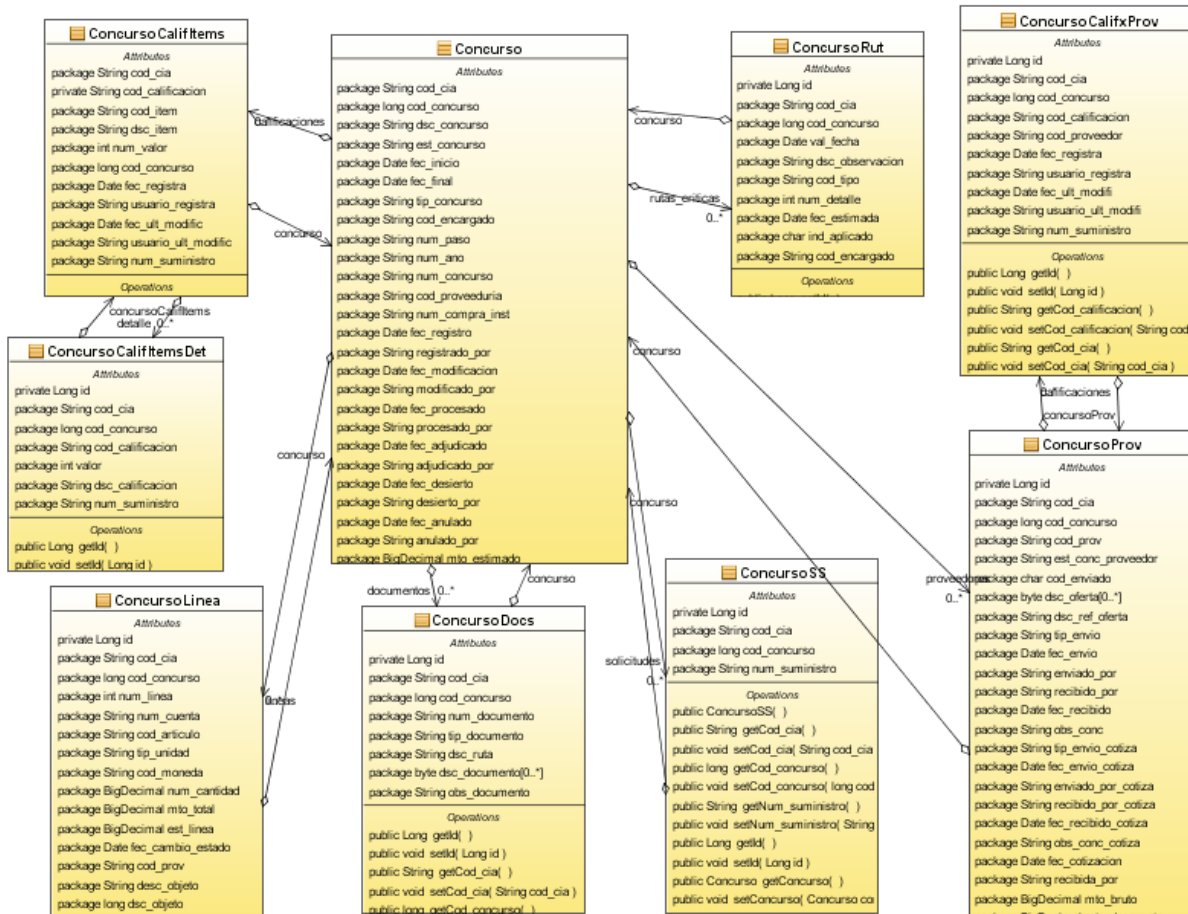


Ilustración 7 - Muestra el diagrama de clases del modulo de proveduría parte 2

En el diagrama se visualiza la entidad de Concursos de proveduría, la cual por cuestiones de espacio, no muestra todos sus campos, asociada a esta, están todas las sub listas de valores tales como: Calificación de Ítems, Líneas del detalle, Documentos, Solicitudes de Suministros asociadas, Proveedores, etc.

### Diagrama del Modulo de Proveeduría 3:

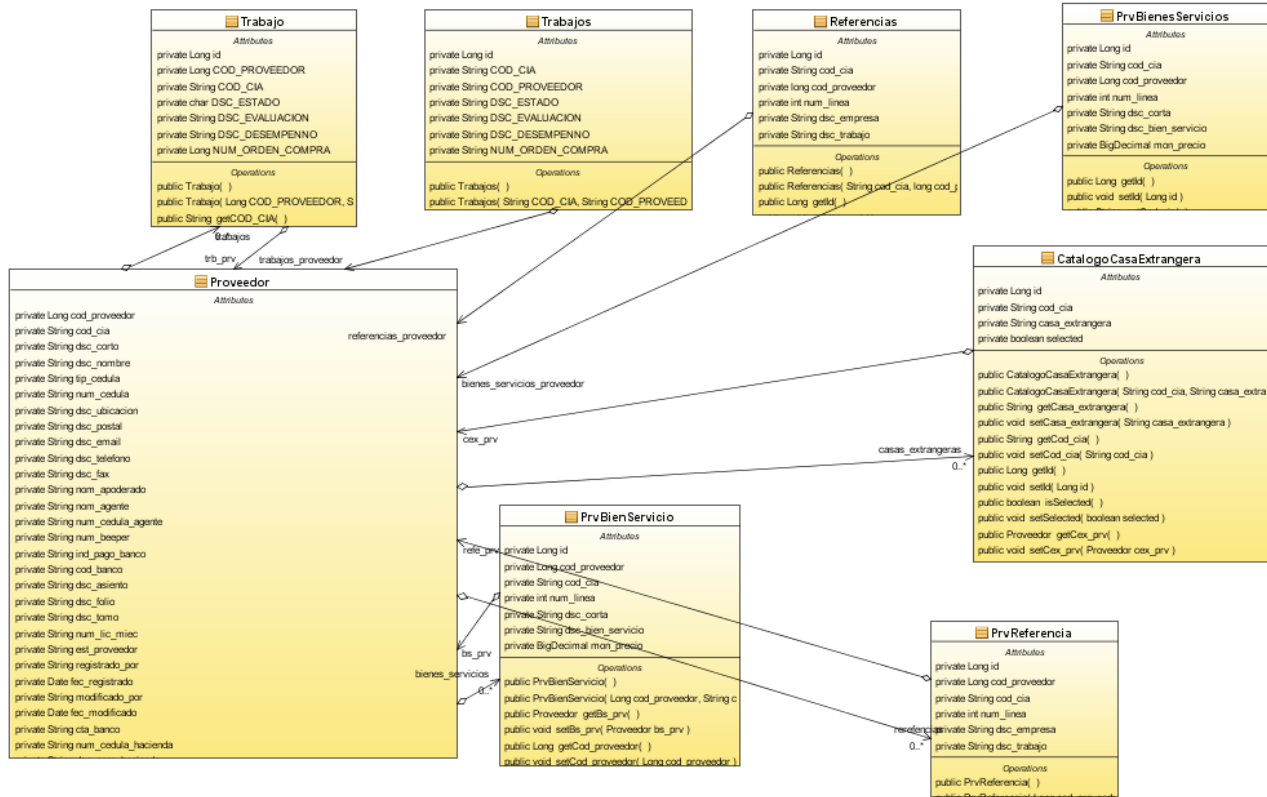


Ilustración 8 - Muestra el diagrama de clases del módulo de proveeduría parte 3

La principal entidad que se muestra en esta parte del programa es la de Proveedores, a su vez, se muestran las listas de valores asociadas al proveedor, tales como: Bienes Servicios, Referencias, Casas Extranjeras, Trabajos, etc.



## Diagrama del Modulo de Proveedorría 4:



Ilustración 9 - Muestra el diagrama de clases del modulo de proveedorría parte 4.

En el diagrama se muestran las entidades de Orden de Compra y sus sub listas de valores, Condiciones, Presupuesto y Detalles, Catalogo de Concursos y Artículos.

## Interfaces de usuario

- ICEfaces:

Es un framework de código abierto para construir aplicaciones web con AJAX tipo RIA (Rich Internet Application). [Cita 3]

Permite al programador incluir una serie de Ajax-tags en sus JSP o xhtml de tal manera que el código Ajax es generado por el propio framework automáticamente.

ICEFaces aísla completamente al desarrollador de AJAX. No hacen falta etiquetas especiales: se ponen los controles en la pantalla y ICEFaces se encarga de enviar sólo la información necesaria entre cliente y servidor.

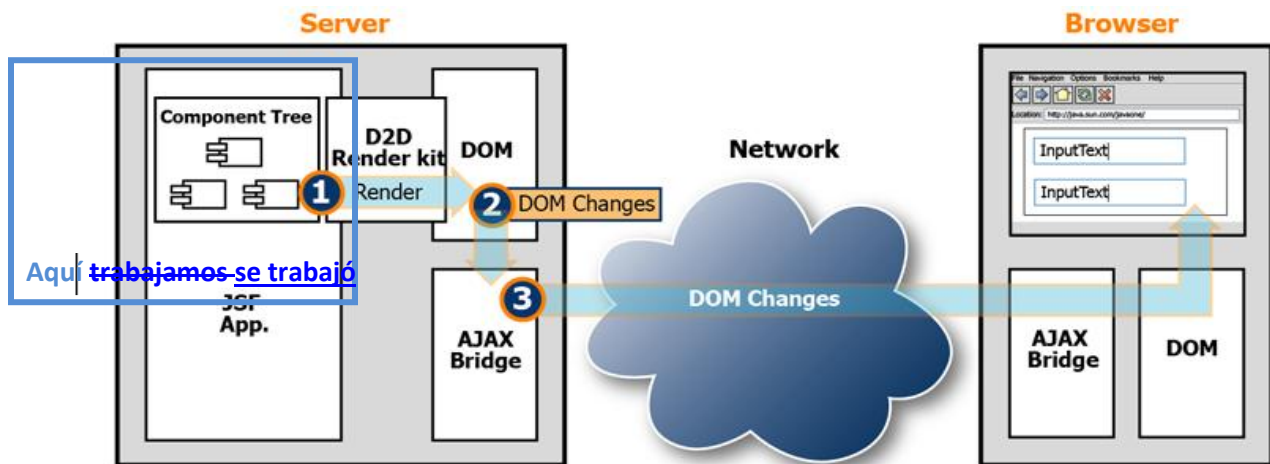
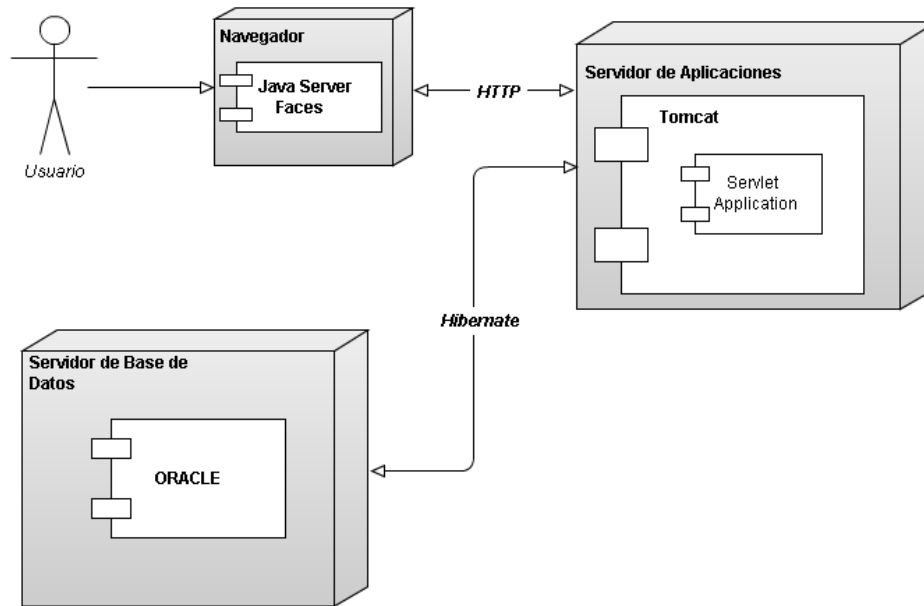


Ilustración 10 - Muestra como trabaja ICEFaces a bajo nivel

## Componentes y Servicios

### Deployment Diagram

---



**Ilustración 11 - Muestra los servicios y componentes utilizados en el proyecto**

La ilustración 5 muestra las diferentes capas que hace uso la aplicación cuando se recibe alguna petición del usuario. El navegador es el primero que envía las peticiones al servidor de aplicaciones, donde un servicio de apache las recibe y procesa, así mismo, en el servidor se ejecuta el sistema que esta enlazado a la base de datos, que para este caso en específico es de Oracle, vemos que la línea que conecta estos dos componentes se llama Hibernate, ya que, dentro de la aplicación es el encargado de realizar el mapeo de las estructuras de datos.

- SubVersion (Control de Versiones):

En el caso de desarrollo de software realizado por el equipo, queremos integrar nuestro código propio con el que finalmente va a ser lanzado. Para la gestión del desarrollo de todo el equipo, se utiliza esta herramienta la cual bajo un pequeño tutorial es relativamente sencilla de utilizar, además es muy versátil a la hora de registrar quien hizo qué cambios.

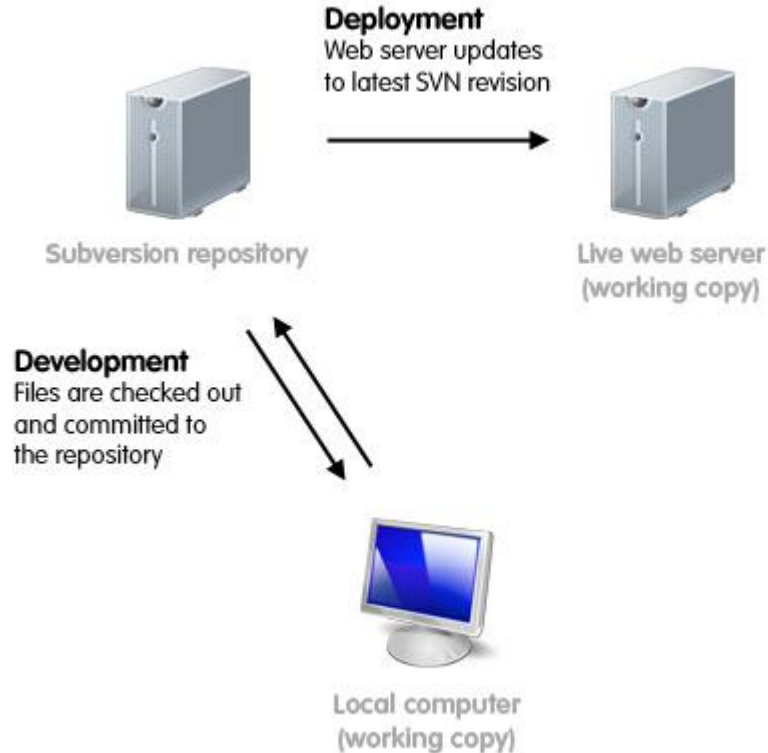


Ilustración 12 - Muestra como se logra la integración continua con un servidor de Subversión

- Maven:

Maven es una herramienta de gestión y comprensión de proyectos de desarrollo de software, normalmente en Java, que nos permite compilar, desplegar, documentar y gestionar las dependencias de código de forma sencilla y extensible.

Maven es una herramienta abierta y libre (pertenece al Proyecto Apache) claramente orientada al trabajo en equipo, esta es una de las principales razones por las que la elegimos como herramienta ideal para metodologías ágiles de programación y posterior Integración Continua con otras herramientas software. [Cita 1]

Para hacer uso de las funcionalidades de Maven dentro del proyecto, es necesario integrar la herramienta al IDE NetBeans. Maven debe tener configurado un servidor donde se almacenan las librerías, podría ser local o en internet, al momento que el proyecto necesite alguna librería y esta no se encuentre localmente, Maven se encarga de buscar en los repositorios y la integra al proyecto. Esta integración se realiza cuando se ejecuta un "Clean&Build" del proyecto.

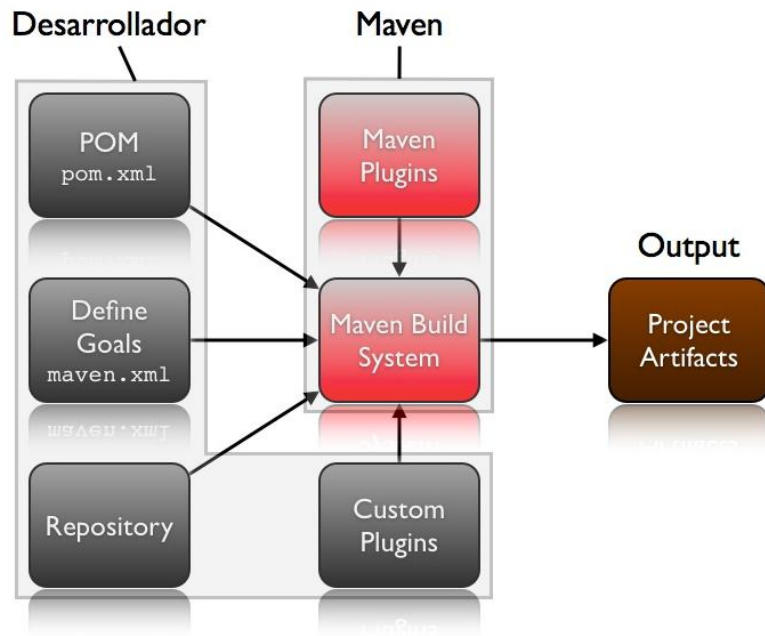


Ilustración 13 - Muestra la interacción de Maven con el proyecto

## Diseño de Bases de Datos

Hibernate: Mediante la utilización de la herramienta hibernate, se realiza la conversión de las clases (Modelo de clases) en tablas hacia cualquier base de datos, esta conversión la realiza, transparente al programador, por lo tanto el diseño de la base de datos es el diagrama de clases también.

Cada clase que cuente con la anotación “@Entity” o entidad de Hibernate, es mapeada en una tabla, a su vez los atributos de la clase serán mapeados como campos con su correspondiente tipo.

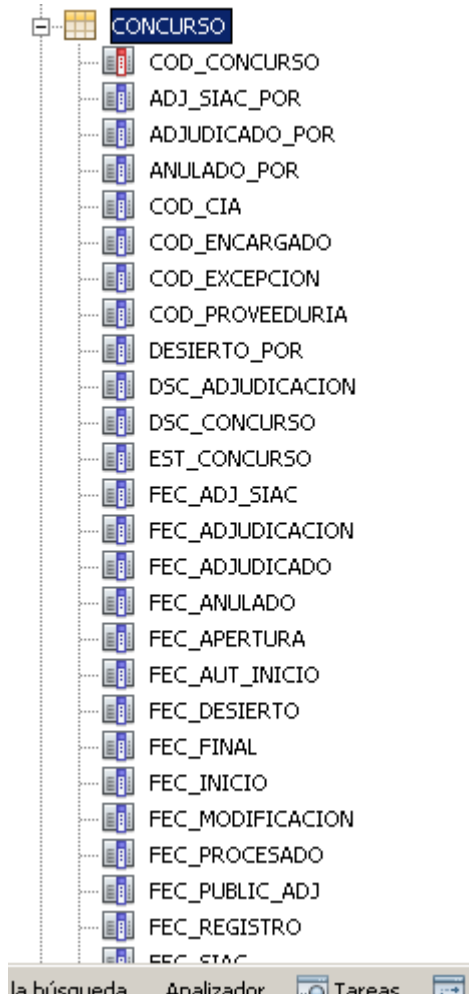
Existen también otras anotaciones por ejemplo: @Id que se utiliza para asignar una clave única a la tabla, @Temporal(DATE) para definir campos de fecha, @Cascade para definir la forma en que se eliminarán las listas de valores que contiene alguna entidad, @OneToMany utilizada para definir relaciones entre clases del tipo entidad. [Cita 5]

### Ejemplo:

```
23  /**
24   * @author Alberto Chaves V
25   * @version 1.0
26   * Clase que representa un concurso de proveeduría
27   */
28  @Entity
29  public class Concurso implements Serializable {
30
31      String cod_cia;
32      @Id
33      @GeneratedValue(strategy=GenerationType.AUTO)
34      long cod_concurso;
```

Ilustración 14 - Anotación Entidad utilizada en una clase.

La clase anterior se vería mapeada en una tabla como en la siguiente imagen:



La imagen muestra la tabla una vez mapeada, nótese que el atributo `cod_concurso`, se muestra es rojo, esto sucede porque en la clase fue declarado como campo `@Id`, el equivalente a un primarykey en el modelo relacional.

También se muestran el resto de campos mapeados.

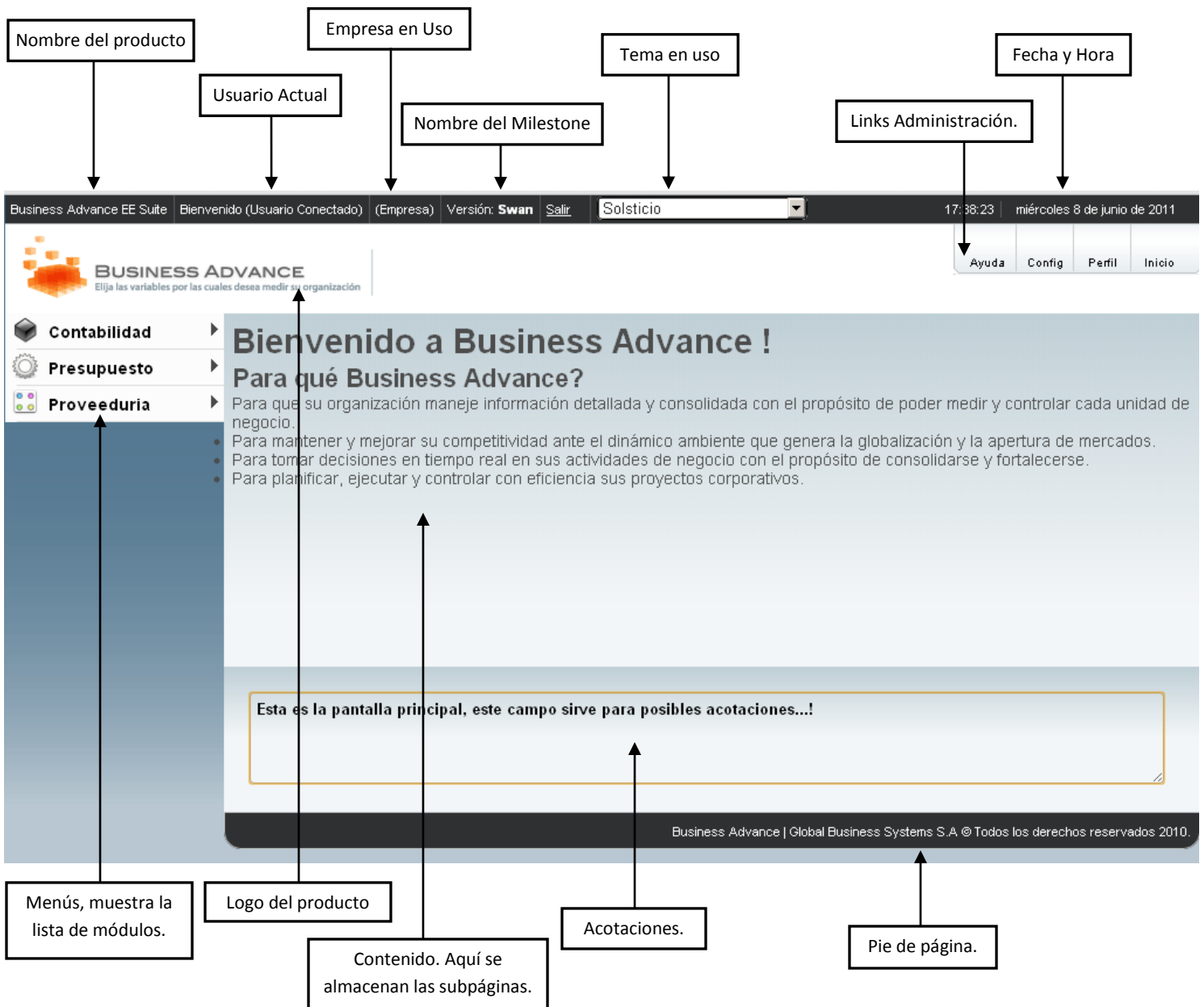
Ilustración 15 - Clase mapeada en una tabla con Hibernate

## Sobre la aplicación

La aplicación implementada consta de tres módulos, contabilidad, presupuesto y proveeduría, estos tres módulos quedaron integrados en un sistema, por las características de la metodología de trabajo que utilizamos llamada Scrum, se puede considerar un demo. Recordemos con Scrum [Cita 2] en cada cierre de sprint se realizan una serie de funcionalidades de la aplicación completas las cuales se pueden agrupar y tener un sistema estable, esta agrupación, en nuestro proyecto, se realizó con la herramienta de integración continua sub-version.

Algo que hay que señalar, es que como en todo sistema no hay secciones huérfanas sino que más bien es integral, en los módulos aquí desarrollados se necesitaron implementar algunas tablas y características que pertenecían a otros módulos del sistema original necesarias para el correcto funcionamiento de los de algunas características de los módulos en desarrollo, por lo que también se dejaron varias otras estructuras previstas para los módulos que se seguirán desarrollando, ya que la idea es completar la migración completa del sistema a una plataforma web.

A continuación se muestra y detalla la estructura de la pantalla base del sistema:



Además de las funcionalidades que se migraron también se integraron secciones nuevas a la aplicación, una sección que se tardó alrededor de dos semanas en desarrollarse fue el catálogo de servicios, para explicar mejor lo que hace el catálogo de servicios, un artículo 'x' puede pertenecer a una familia determinada a su vez el artículo puede pertenecer a una sub familia determinada y por último a una clase, así estaba en la aplicación anterior, pero uno de los requisitos nuevos fue que se pudieran tener n cantidad de niveles, es decir, una familia con



una subfamilia con una clase y una sub clase, etc. En resumen un catalogo de servicios dinámico, esta sección yo decidí desarrollarla y al final quedo muy presentable y después de una corrección que era agregar un campo de código quedó funcionando. La misma a nivel de datos se almacena en una tabla y el ordenamiento se realiza a nivel lógico mediante el uso de algunos índices, estas y otras secciones de la aplicación se muestran a continuación:



Ilustración 16 - Muestra la pantalla de catalogo de servicios dinámicos, en el tercer nivel.

Adicionalmente esta pantalla muestra la cuenta de activos y la cuenta de gastos asociada al elemento cuando se posiciona el cursor encima de algún ítem.

## Conclusiones y Comentarios

Bajo la metodología de trabajo de Scrum y después de un breve proceso de entrenamiento en las tecnologías usadas por la empresa, hemos logrado concluir con los objetivos que teníamos para este proyecto, parte del éxito radicó en la metodología de trabajo usada lo que permite un rápido desarrollo de las aplicaciones a costa de un mayor esfuerzo por parte del programador. Como bien es sabido, ninguna metodología puede asegurar el éxito de ningún proyecto, eso depende de la organización de los equipos de trabajo que estén comprometidos en su desarrollo.

Como producto final, se implementó un sistema web, que se ejecuta en un servidor de Apache, haciendo uso de una base de datos Oracle, lo cual en este ámbito, es transparente para el programador, ya que se trabaja con tecnología de persistencia de datos. El sistema posee los módulos de presupuesto y proveeduría, además del modulo de contabilidad que ya estaba hecho, estos módulos, aun no estarán en producción, hasta que no sean revisados exhaustivamente, por una empresa subcontratada, especializada en pruebas de calidad, funcionamiento y control lógico del proceso, después de este proceso el producto vuelve a desarrollo para que sean corregidas las pulgas y demás incoherencias que puedan aparecer, para luego ser mostrado a los gerentes y a los clientes, que serán lo que den el visto bueno, sobre si puede ser puesto en producción.

Como vemos, el desarrollo de un producto, no es cosa de un día, todo el tiempo hay mucha gente comprometida e involucrada en el proyecto, muchas ideas que retomar, funcionalidades que son deseables encontrar en el producto, adaptación a entandares preestablecidos para las aplicaciones dentro del ámbito y demás opiniones de los involucrados, que alimentan el producto final.

El aprendizaje que se pueda adquirir va a depender de la actitud de cada persona y de la actitud de hacer las cosas cada día mejor, no basta con estar en un ambiente laboral en el que se habla sobre muchos temas, desde el ámbito tecnológico hasta el ámbito deportivo y personal, también se debe estar abierto a aprender cosas nuevas y por supuesto, dar algo de cada quién.

## Bibliografía

1. Venera. "Integración Continua y Maven" [en línea]. 04 de mayo de 2011. Disponible en la web: <http://www.venera7.com/category/productividad/>
2. Wikipedia. "Scrum" [en línea]. 13 de abril de 2011. Disponible en la Web: <http://es.wikipedia.org/wiki/Scrum>
3. IceFaces "Página Principal del proyecto" [en línea] 17 de junio de 2011. Disponible en la web: <http://www.icefaces.org>
4. Planning Poker "Pagina Principal" [en línea] 17 de junio de 2011. Disponible en la web: <http://www.planningpoker.com/>
5. Hibernate "Manual de Referencia" [en línea] 17 jun. 11 Disponible en la web: <http://mundogeek.net/archivos/2007/01/27/hibernate/>