



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

“Tercer Informe de Práctica”

Juan Diego Morera Chaves

Carrera Ingeniería en Computación

Instituto Tecnológico de Costa Rica

Prof. Asesor: Oscar Mario Víquez Acuña

Sede San Carlos

Junio, 2011

Tabla de contenido

Resumen Ejecutivo	4
1. Contexto del Proyecto.....	5
1.1. Datos Generales sobre La Empresa	5
1.2. Ubicación del departamento donde se va a realizar la Práctica de especialidad	9
1.3. Cantidad de Recurso Humano del Departamento	9
1.4. Descripción general del proyecto.....	9
2. Descripción del Problema	11
2.1. Enunciado del Problema	11
2.2. Enunciado de la Solución.....	11
2.3. Resumen de Necesidades y Expectativas	12
2.4. Perspectiva, supuestos y dependencias del producto	12
2.5. Supuestos.....	13
2.6. Requerimientos no funcionales	13
2.7. Tabla de Análisis de los Riesgos	13
3. Objetivos y Alcances del sistema.	14
3.1. Objetivo General.....	14
3.2. Objetivos Específicos.....	14
3.3. Alcances del proyecto.....	15
4. Modelo de Diseño.....	16
4.1. Arquitectura conceptual de la solución.	16
4.2. Modelos de Subsistemas.	16
4.3. Diagrama de Clases y de Base de Datos.....	18
4.4. Interfaces de Usuario.....	19
4.5. Componentes y servicios.	21
5. Referencias.	26
6. Conclusiones y Recomendaciones.....	26
7. Informes de Avance	¡Error! Marcador no definido.

Tabla de ilustraciones

Ilustración 1 (Organigrama GBSYS) Ver Original	7
Ilustración 2 (Tabla de necesidades y Expectativas) Ver Original	12
Ilustración 3 (Tabla de Análisis de Riesgos) Ver Original.....	13
Ilustración 4 (Diagrama General del Sistema) Ver Original.....	16
Ilustración 5 (Modelo de Subsistemas) Ver Original	17
Ilustración 6 (Diagrama de Clases Presupuesto) Ver Original.....	18
Ilustración 7 (Diagrama de Clases Proveeduría) Ver Original	19
Ilustración 8 (Página Principal Proyecto Avance) Ver Original.....	20
Ilustración 9 (Página de Gestión de Proveedores) Ver Original	21
Ilustración 10 (Ejemplo Utilización de Apache Maven) Ver Original	22
Ilustración 11 (Ejemplo uso de código JAVA) Ver Original	23
Ilustración 12 (Ejemplo uso de JSF) Ver Original.....	24
Ilustración 13 (Ejemplo salida información en interfaz) Ver Original	24
Ilustración 14 (Ejemplo relaciones Hibernate) Ver Original.....	25

Resumen Ejecutivo

El presente documento pretende, mediante un conjunto de secciones organizadas, informar del estado final del proyecto y de los productos generados en este proceso, haciendo un repaso por la descripción del problema, la cual presenta las características principales del lugar donde se va a realizar la práctica de especialidad y sobre algunos datos generales del proyecto a realizar, como por ejemplo antecedentes, tecnologías a utilizar, alcances y otros varios aspectos que representan el marco en el que se va a realizar el proyecto y que dan una idea del producto deseado y de la importancia que tiene el mismo para los intereses de la empresa.

Luego de esto se continua con la descripción de la solución que se implementó para poder dar una respuesta al ya citado problema, con lo cual se presenta el trabajo realizado, descrito en lenguaje de modelado UML, detallado y con todos los elementos necesarios para que se vea reflejado el trabajo que se realizó para poder proveer la solución esperada por parte del ente que la necesita, que en este caso sería la empresa en donde se va a realizar la el ya mencionado proceso de práctica de especialidad.

Para finalizar como un elemento de retroalimentación, el documento presenta un conjunto de conclusiones y recomendaciones que sirvan base para ayudar a mejorar el proceso de práctica de especialidad, el cual debe estar guiado a ser cada vez más competente y ayudar al estudiante a finalizar una etapa de sus estudios convertido en un profesional listo para afrontar la vida laboral de manera profesional.

1. Contexto del Proyecto

1.1. Datos Generales sobre La Empresa

La empresa donde se desarrolló la práctica es Global Business System S.A (GBSYS).

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 ó 506-2234-2756.

Facsímil: (506)2280-7542.

Principales actividades: Desarrollo de sistemas, Administración de Bases de datos, Consultorías informáticas.

1.1.1. Visión de la Empresa.

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

1.1.2. 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."

1.1.3. 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.

1.1.4. **Objetivos específicos**

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

- a) 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.
- b) Brindar entrenamiento en el manejo de equipo y paquetes de software en instituciones y empresas, por medio de seminarios, conferencias, charlas y cursos cortos.
- c) Brindar asesoría técnica en la selección y compra de equipo de cómputo.
- d) Desarrollar paquetes de software generalizados bajo estándares de calidad mundial.
- e) Distribuir software especializado de casas extranjeras de amplio reconocimiento en la industria.

1.1.5. Organigrama.

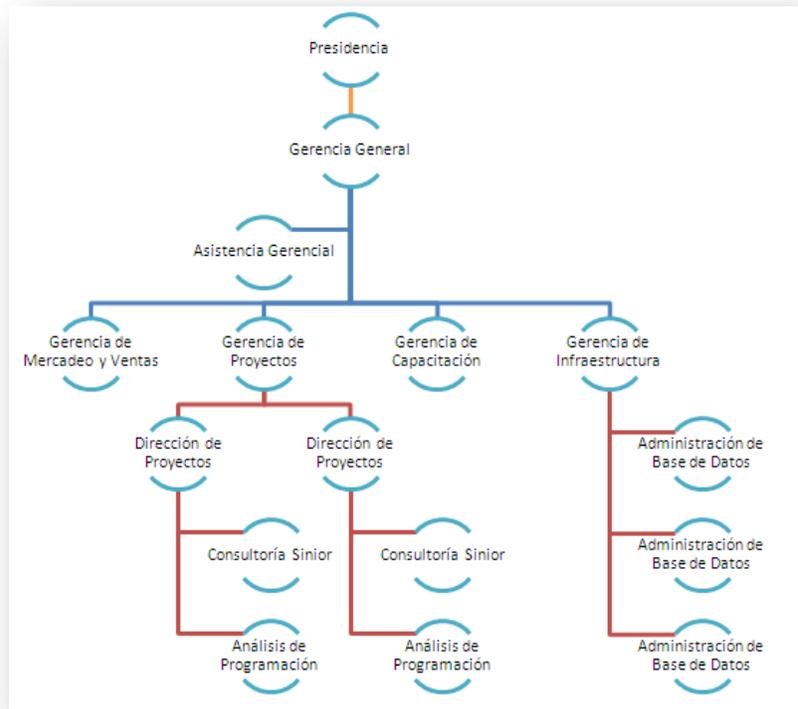


Ilustración 1 (Organigrama GBSYS) [Ver Original](#)

1.1.6. Estructura organizacional de la empresa.

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

- a) Presidente
- b) Gerente General

1.1.7. Departamentos.

- a) Asistente Administrativa
- b) Gerente de Ventas
- c) Gerente de Proyectos
- d) Gerente de Infraestructura
- e) Gerente de Capacitación

1.1.8. Descripción de los departamentos.

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, y tienen bajo su cargo a consultores sénior 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.

1.1.9. 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.

1.2. **Ubicación del departamento donde se va a realizar la Práctica de especialidad**

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.

1.3. **Cantidad de Recurso Humano del Departamento**

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

1.4. **Descripción general del proyecto**

En la actualidad existe un producto (ERP) de la empresa denominado Avance con 4 instalaciones en diferentes clientes.

Avance es una solución financiero administrativo orientada a optimizar la gestión de las unidades de negocio de la organización.

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

- a) Seguridad
- b) Cuentas por Pagar
- c) Administración
- d) Activos Fijos
- e) Contabilidad
- f) Inventarios
- g) Presupuesto
- h) Proveeduría
- i) Control Bancario
- j) Custodia de Valores
- k) Caja Chica
- l) Inversiones
- m) Emisión de pagos
- n) Facturación

- o) Cuentas por Cobrar
- p) Configuración
- q) Transportes

1.4.1 Los objetivos principales de Avance para la organización son:

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

1.4.1. Antecedentes del Proyecto

El actual producto Avance 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, los clientes han mostrado gran interés en tener la funcionalidad de este producto en plataforma WEB utilizando J2EE, logrando de este modo que la aplicación sea más portable a otros servidores de aplicaciones a parte del OAS.

En años anteriores se inició con la migración de este proyecto con lo cual otros estudiantes de practica ya han realizado algunos de los módulos de dicho sistema.

2. Descripción del Problema

2.1. Enunciado del Problema

Como parte de la migración del Avance la Empresa necesita que se continúe la migración con los módulos de Proveduría y Presupuesto lo cual sería el problema a solucionar para el caso de la presente práctica de especialidad. Dichos módulos vendrían a sustentar las necesidades del departamento de proveeduría de las empresas, así como movimientos presupuestarios que se generarían de dicho proceso, elemento vital para la mayoría de empresas públicas a las cuales está dirigida la herramienta.

2.2. Enunciado de la Solución

Para el final de la práctica se entregue una migración de los módulos de presupuestos y proveeduría que se integrarán al Sistema Avance que ya se ha venido migrando, sumado a algunas mejoras en los módulos anteriores que se habían realizado.

2.2.1. Tecnologías.

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

- a) IDE NetBeans 6.5.
- b) Framework de presentación Java Server Pages (JSF) e ICEfaces.
- c) Framework de persistencia de datos Hibernate.
- d) Base de Datos Oracle 10g.
- e) Servicios Web.

2.2.2. Descripción de los patrocinadores (“Stackeholders”).

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

- Luis Emilio Ramírez: Director de proyectos y coordinador del presente proyecto con conocimientos del producto que apoyará en las fases del proyecto, la planeación de las actividades y el control del cronograma.
- Pablo Peraza: líder técnico en arquitectura y desarrollo de aplicaciones WEB.
- Alberto Chaves: Estudiante que será parte del equipo desarrollador.

- Gustavo A. Rodríguez Vargas: Estudiante que será parte del equipo desarrollador.
- Juan Diego Morera Chaves: Estudiante que será parte del equipo desarrollador.

2.3. Resumen de Necesidades y Expectativas

Necesidad	Prioridad	Problema	Solución Actual	Solución Propuesta
Cambiar el lenguaje de programación del producto	Alta	El lenguaje es PL ya que el sistema fue hecho en SQLDeveloper y no es para nada portable porque necesita de las capacidades de Oracle para funcionar.	Se está migrando el producto a lenguaje JAVA el cual es multiplataforma.	Se seguirá utilizando el lenguaje JAVA ya que este ha demostrado ser el que se ajusta a las necesidades y capacidades que presenta el proyecto.
Desligar el proyecto de una base de datos específica	Alta	Al estar el producto generado para Oracle PL se desperdician posibilidades de acaparar mercado con opciones de utilizar distintos gestores de bases de datos	Desde Java se utiliza una conexión la cual puede ser configurada para trabajar con distintos tipos de base de datos, además se utiliza hibernate para no trabajar directamente en Hibernate y así poder hacer las consultas lenguaje JPQL el cual es un lenguaje de consulta independiente de plataforma orientado a objetos definidos como parte de la especificación Java Persistence API	Se seguirá trabajando con la metodología actual, pero aplicando un tipo de consulta más dinámica mediante el elemento llamado NamedQuery el cual es una propiedad de Java Persistence y permite realizar consultas específicas en lenguaje JPQL en Hibernate.

Ilustración 2 (Tabla de necesidades y Expectativas) [Ver Original](#)

2.4. Perspectiva, supuestos y dependencias del producto

Del producto se espera que sea un punto importante en lo referente a programación en la empresa dado que se van a implementar nuevas técnicas y metodologías que pretenden sean un referente para sus próximos proyectos, además la herramienta es muy esperada por sus clientes por lo que se espera que quede lo mejor desarrollada posible.

2.5. Supuestos.

- a) Servidor de Base de datos en funcionalidad, que contenga datos de ejemplo dada la aplicación pasada del sistema
- b) Servidor de Subversión con la parte funcional de los avances que se habían realizado anteriormente por los practicantes que habían testado previamente realizando la práctica de especialidad.
- c) Infraestructura (computadores, conexión a red, y cubículos), necesaria para desempeñar el trabajo.

2.6. Requerimientos no funcionales

- a) Tiempos de respuesta aceptables (menores a 10s.)
- b) Interfaz coherente con la anteriormente realizada en versiones previas de la migración.

2.7. Tabla de Análisis de los Riesgos

Nombre	Categoría	Posible Causa	Impacto (I)	Prob. Ocurrencia (P)	Exposición al riesgo (I*P)	Estrategia Evasión	Estrategia Mitigación	Estrategia Contingencia
Perdida de conexión con la base de datos	Tecnológico	Múltiple uso de los servidores por varias personas	8	2	16	Tener una copia de la base de datos en una BD local	Usar la copia de la base de datos en una BD local	Tener una copia de la base de datos en una BD local
Perdida de conexión con servidor de Subversión	Tecnológico	Múltiple uso de los servidores por varias personas	5	2	10		Actualizar los archivos a pie en las terminales de los compañeros	Crear un servidor Local de Subversión
Atrasos en la entrega del producto			10	1	10		Pasar el trabajo al siguiente Sprint	Trabajar con estimaciones de tiempo un poco pesimistas
Falta de seguimiento del proyecto por parte de la empresa	Personas	Se trata de una compañía con un volumen de trabajo alto y varias tareas diarias para cada persona	8	1	8		Pedir un mayor seguimiento de los avances que se van realizando.	Tener buena comunicación con los <i>Stakeholders</i> en general y una actitud positiva.

Ilustración 3 (Tabla de Análisis de Riesgos) [Ver Original](#)

3. Objetivos y Alcances del sistema.

3.1. Objetivo General

Se Migrarán dos módulos (Presupuesto y Proveeduría) desde un sistema antiguo, llamado Avance el cual estaba creado en Oracle Forms, con lo que se espera que al final quede un sistema nuevo elaborado en lenguaje JAVA.

3.2. Objetivos Específicos.

Para implementar tanto el módulo de presupuesto como el de proveeduría, se dividirá el módulo por características, para todas las características del sistema se realizará lo siguiente:

- a) Darle al programador un conocimiento medio en las metodologías usadas para construir el sistema.
- b) Enseñar al programador adaptación en el ambiente de desarrollo usado para construir el sistema.
- c) Darle al programador conceptos y definiciones contables básicas referentes al presupuesto y proveeduría de una empresa.
- d) Enseñar al programador los conceptos y definiciones contables básicas referentes al presupuesto y proveeduría de una empresa.
- e) Que el desarrollador investigue las diferentes funcionalidades que presentan las características y las documente según se necesite.
- f) Que el desarrollador verifique las diferentes validaciones que realizan las características a nivel de lógica del negocio.
- g) Que el desarrollador genere las estructuras de datos (Clases, relaciones, etc.) que utilice las características.
- h) Que el sistema posea una interfaz gráfica que permita el uso de las características.

Los demás objetivos se despliegan seguidamente en el orden en que se realizaran.

- a) Contar con una característica que permita la gestión de Formularios Presupuestarios.
- b) Contar con una característica que permita la gestión de Documentos Presupuestarios.
- c) Contar con una característica que permita la gestión de Movimientos Presupuestarios.
- d) Contar con una característica que permita la gestión de Modificaciones al Presupuesto.
- e) Contar con una característica que permita la gestión Proveedores.
- f) Contar con una característica que permita la gestión Solicitudes de Suministros.
- g) Contar con una característica que permita la gestión de concursos.
- h) Contar con una característica que permita la gestión Órdenes de Compra.

3.3. Alcances del proyecto

3.3.1. Módulo de Presupuestos

- a) Gestión de Formularios de presupuesto.
- b) Gestión de documentos Presupuestarios.
- c) Gestión de movimientos Presupuestarios.
- d) Gestión de modificaciones a los presupuestos.

3.3.2. Módulo de Proveduría

- a) -Gestión de proveedores.
- b) -Gestión de solicitudes de Suministros.
- c) -Gestión de concursos.
- d) -Gestión de órdenes de compra.

4. Modelo de Diseño

4.1. Arquitectura conceptual de la solución.

La arquitectura del programa sigue la metodología de Modelo Vista Controlador, en la cual se tiene la lógica del negocio por un lado, la parte visual aparte y la parte de la conexión a la base de datos también por aparte, teniendo más ordenada la interacción entre los componentes del proyecto y facilitando al programador el trabajo y desarrollo de sus funciones.

Para el presente proyecto se ha dividido esto en Clases tipo Vista (con la lógica del negocio), Clases Controladoras (Con la conexión a base de datos), y la interfaz HTML.

Seguidamente se muestra el diagrama general del sistema que muestra la interacción entre los componentes del sistema y el usuario.

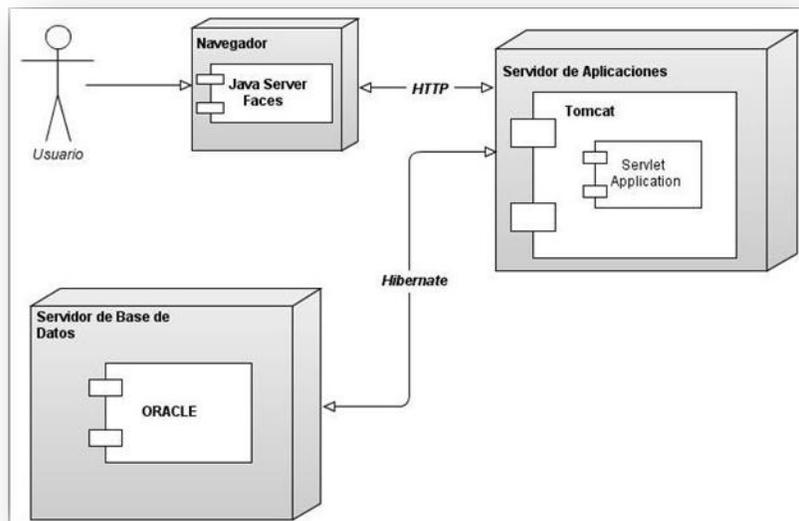


Ilustración 4 (Diagrama General del Sistema) [Ver Original](#)

4.2. Modelos de Subsistemas.

El modelo de subsistemas está conformado por los componentes lógicos que le dan forma a la aplicación y que interactúan entre sí para dar funcionalidad al sistema, en este caso los componentes son:

- Entidades
- Lógica del negocio (Vistas)
- Interfaz de usuario
- Clases controladoras
- Componentes de persistencia (mapeo con la base de datos)

Seguidamente se ejemplifica el modelo.

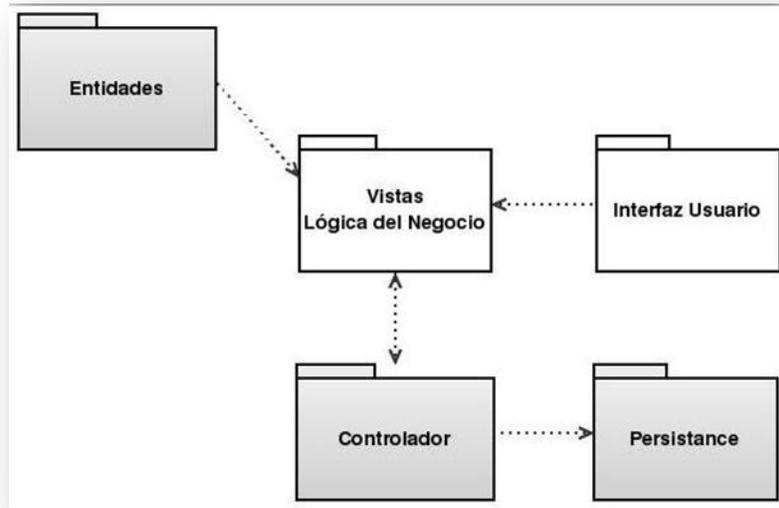


Ilustración 5 (Modelo de Subsistemas) [Ver Original](#)

Todos estos elementos que conforman el modelo de subsistemas están constantemente ligados y de su debida configuración y utilización depende el funcionamiento de la aplicación.

Descripción: en este diagrama se detallan las clases que se han creado para la el módulo de presupuestos, sus relaciones, atributos y funciones, los cuales están inmersos en la aplicación de Hibernate que se implementa en las clases del proyecto.

A continuación se muestra el Diagrama de clases del módulo de Proveeduría para el Sistema Avance.

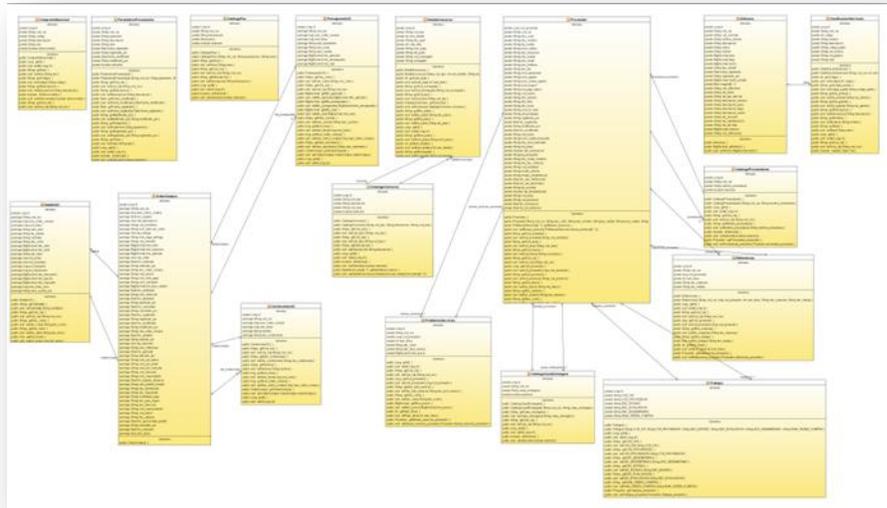


Ilustración 7 (Diagrama de Clases Proveeduría) [Ver Original](#)

Descripción: en este diagrama se detallan las clases que se han creado para la el módulo de Proveeduría, sus relaciones, atributos y funciones al igual que en el diagrama anterior, este presenta la funcionalidad de Hibernate en el mapeo de las clases hacia la base de datos.

4.4. **Interfaces de Usuario.**

La interfaz de usuario está basada en elementos de Java Server Faces y paso entre pantallas es manejado por Ajax por lo que siempre se trabaja sobre una sola página base, la cual sirve de modelo a seguir en todas las demás pantallas. Esta contiene un menú que permite recorrer todos los módulos y secciones del proyecto. Además presenta una barra de estado en la parte superior que informa sobre datos generales, como el nombre de la aplicación, versión de la aplicación, nombre de la empresa, menú de tema para la interfaz, hora y fecha del servidor, etc.

La sección de trabajo está en la parte central de la interfaz y se crea dinámicamente aprovechando las capacidades y configuración del monitor en el que se ejecute la aplicación.

Para terminar de aclarar los anteriores aspectos de interfaz seguidamente se muestra una imagen de la página principal que se presenta al usuario.



Ilustración 8 (Página Principal Proyecto Avance) [Ver Original](#)

Seguidamente se muestra una imagen del formato de página que se presenta al usuario en los distintos módulos que presenta el proyecto el cual consiste principalmente de formularios para ingreso de datos y tablas para el despliegue de información guardada

en base de datos.

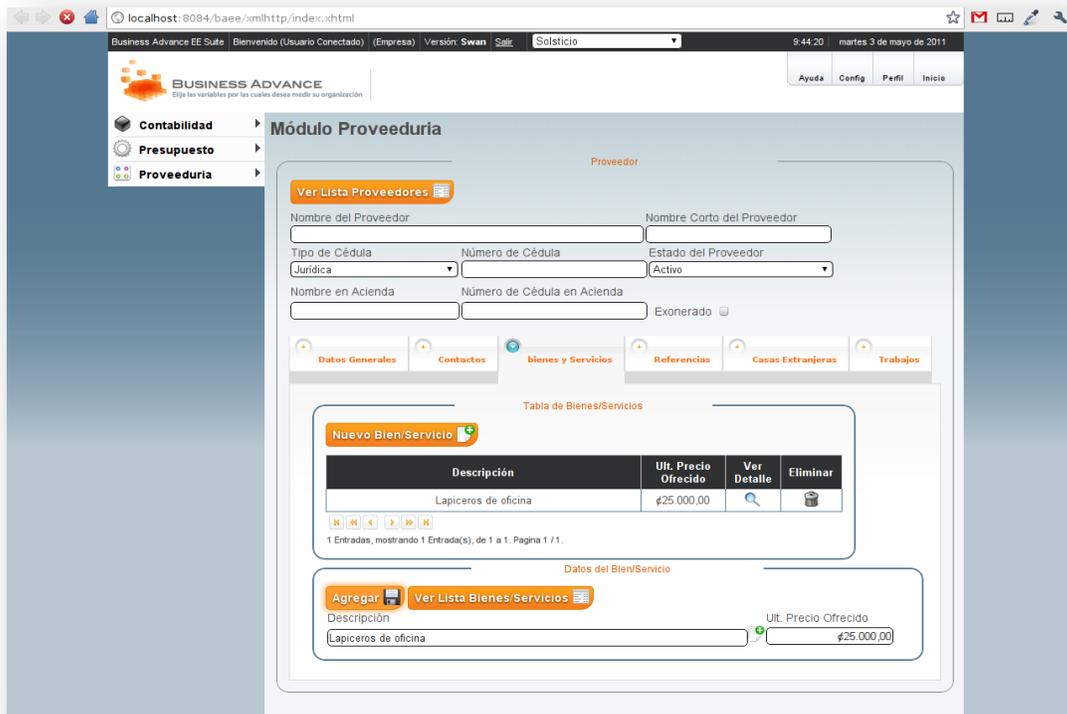


Ilustración 9 (Página de Gestión de Proveedores) [Ver Original](#)

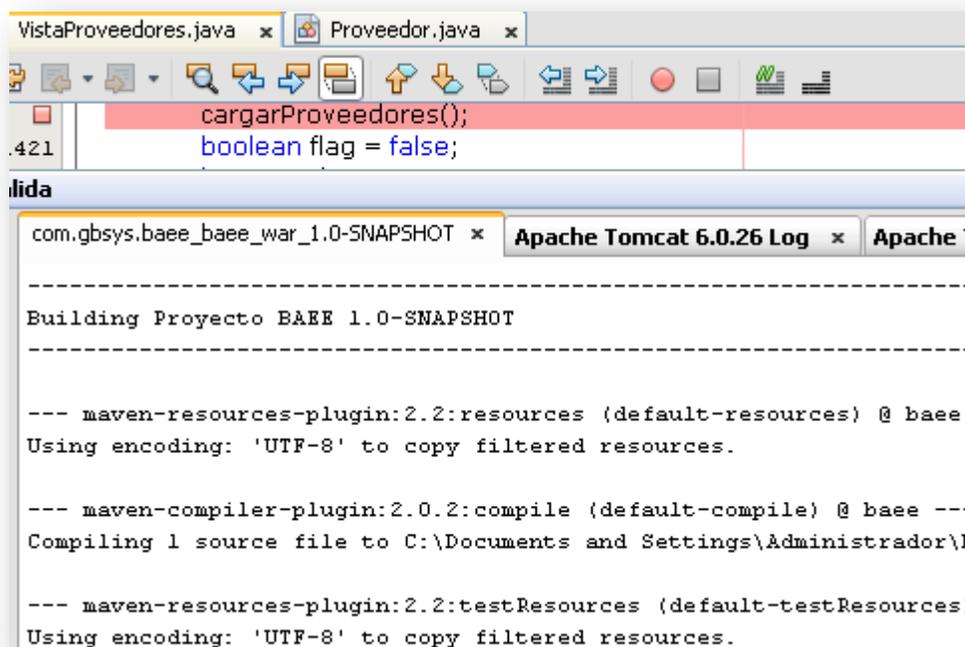
4.5. Componentes y servicios.

- Apache Maven
- Java
- Java Server Faces
- Hibernate
- ORACLE

Seguidamente se indica cómo se utilizarán las ya mencionadas tecnologías en la realización del proyecto.

Primero esta Apache Maven sobre el cual se monta el proyecto, Maven es una herramienta de software para la gestión y construcción de proyectos Java, utiliza Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Dinámicamente descarga los plugins que se necesitan desde un repositorio, desde el cual se provee acceso a muchas versiones de diferentes proyectos Open Source en Java. [\[1\]](#)

Para terminar de aclarar los anteriores aspectos del funcionamiento de maven en el proyecto seguidamente se muestra una imagen de la aplicación corriendo y algunos de los procesos que realiza maven, como la aplicación de plugins que el mismo baja y aloja en el servidor para la utilización en la corrida del proyecto.



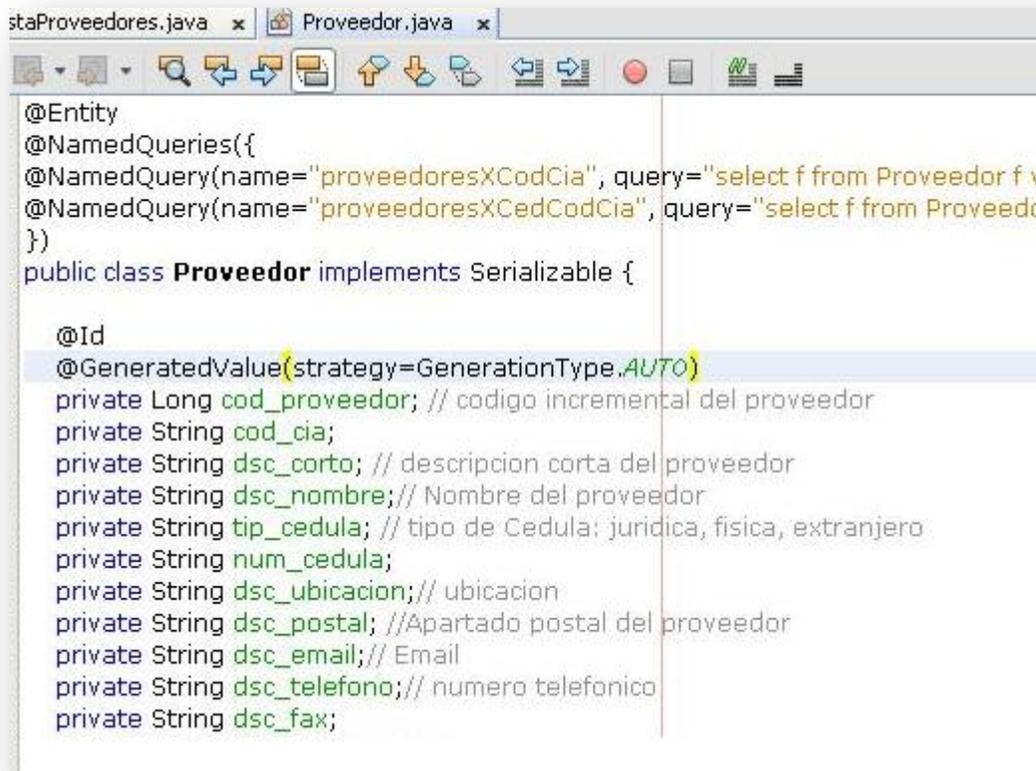
The screenshot shows an IDE window with two tabs: 'VistaProveedores.java' and 'Proveedor.java'. The code in 'Proveedor.java' is highlighted, showing the method 'cargarProveedores()' and a line 'boolean flag = false;'. Below the code is a console window titled 'com.gbsys.baee_baee_war_1.0-SNAPSHOT' and 'Apache Tomcat 6.0.26 Log'. The log output shows the following text:

```
-----  
Building Proyecto BAEE 1.0-SNAPSHOT  
-----  
  
--- maven-resources-plugin:2.2:resources (default-resources) @ baee ---  
Using encoding: 'UTF-8' to copy filtered resources.  
  
--- maven-compiler-plugin:2.0.2:compile (default-compile) @ baee ---  
Compiling 1 source file to C:\Documents and Settings\Administrador\I  
  
--- maven-resources-plugin:2.2:testResources (default-testResources) @ baee ---  
Using encoding: 'UTF-8' to copy filtered resources.
```

Ilustración 10 (Ejemplo Utilización de Apache Maven) [Ver Original](#)

Para la creación de la lógica del negocio se utilizará lenguaje JAVA el cual además de ser muy portable tiene muchas variantes en cuanto a su funcionalidad e interoperabilidad con diversas herramientas de programación como lo son Hibernate y Java Server Faces.

Como ejemplo se muestra parcialmente una entidad creada en código JAVA la cual por medio de Hibernate se va a poder mapear luego dentro de la base de datos como una tabla.



```
staProveedores.java x Proveedor.java x
@Entity
@NamedQueries({
@NamedQuery(name="proveedoresXCodCia", query="select f from Proveedor f v
@NamedQuery(name="proveedoresXCedCodCia", query="select f from Proveedo
})
public class Proveedor implements Serializable {

@Id
@GeneratedValue(strategy=GenerationType.AUTO)
private Long cod_proveedor; // codigo incremental del proveedor
private String cod_cia;
private String dsc_corto; // descripcion corta del proveedor
private String dsc_nombre; // Nombre del proveedor
private String tip_cedula; // tipo de Cedula: juridica, fisica, extranjero
private String num_cedula;
private String dsc_ubicacion; // ubicacion
private String dsc_postal; // Apartado postal del proveedor
private String dsc_email; // Email
private String dsc_telefono; // numero telefonico
private String dsc_fax;
```

Ilustración 11 (Ejemplo uso de código JAVA) [Ver Original](#)

Para la manipulación de interfaz se utiliza Java Server Faces el cual es un framework Ajax para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones por la gran cantidad de funcionalidades que este lenguaje acapara con sus componentes enriquecidos. Para la realización del proyecto JSF nos ha facilitado mucho la conexión de la interfaz con las entidades ya que se pueden cargar datos directamente en los campos que se presentan en interfaz, tal y como se presenta en el siguiente ejemplo grafico. [\[2\]](#)

```
es.java x Proveedor.java x catalogo_proveedores.xhtml x
<ice:dataTable
  id="tabla_proveedores"
  var="proveedor"
  value="#{VistaProveedores.proveedores}"
  rows="10">

  <ice:rowSelector id="selected"
    multiple="true"
    selectionListener="#{VistaVisualizarFuncionalidades.seleccion}"
    preStyleOnSelection="true"/>
  <ice:column style="width: 250px">
    <f:facet name="header">
      <ice:outputText value="Nombre Proveedor" style="width: 250px"/>
    </f:facet>
    <ice:outputText value="#{proveedor.dsc_nombre}" style="width: 250px"/>
  </ice:column>
```

Ilustración 12 (Ejemplo uso de JSF) [Ver Original](#)

Este fragmento de código toma los valores asignados para la entidad como la que se ve en (Figura #7), y se transforma en información que automáticamente se muestra en pantalla como se mostrara a continuación

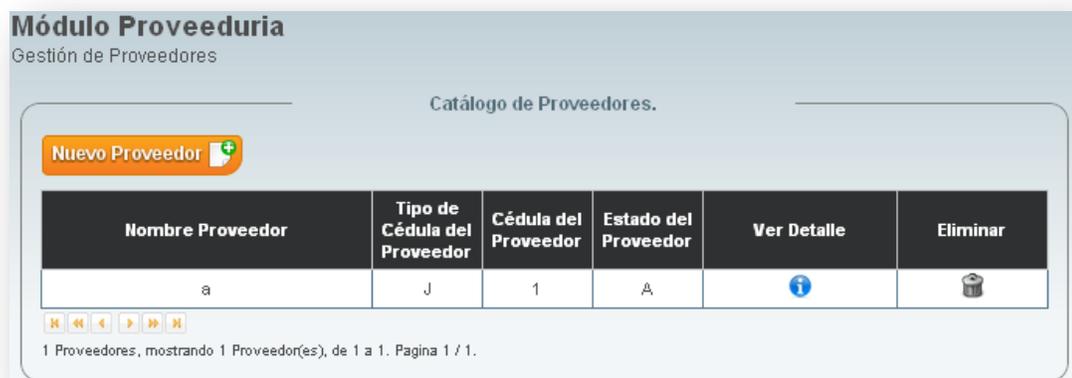
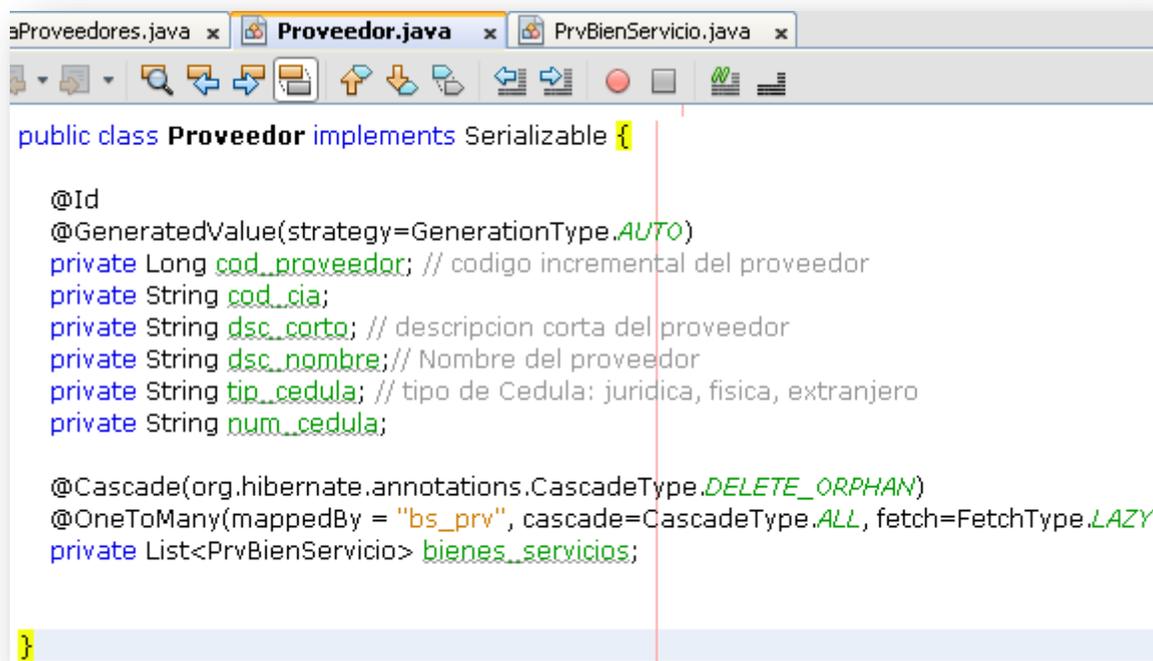


Ilustración 13 (Ejemplo salida información en interfaz) [Ver Original](#)

Finalmente, para la manipulación de datos de la base de datos se utiliza Hibernate Sobre una base de datos ORACLE, en el cual crea la base de datos a partir del modelo de

clases entidades que contenga el proyecto, con lo cual la manipulación es más directa y se facilitan los procesos de mantenimiento necesarios para el sistema. [\[3\]](#)

Además se crean también auto incrementales para los ID de las entidades y los tipos de relaciones entre entidades, así como reglas de cascada, los cuales se verán reflejados en la manipulación y manejo de la base de datos. Dichos tipos de relaciones como “oneToMany” o sea uno a muchos se ejemplifican seguidamente.



```
public class Proveedor implements Serializable {  
  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private Long cod_proveedor; // codigo incremental del proveedor  
    private String cod_cia;  
    private String dsc_corto; // descripcion corta del proveedor  
    private String dsc_nombre; // Nombre del proveedor  
    private String tip_cedula; // tipo de Cedula: juridica, fisica, extranjero  
    private String num_cedula;  
  
    @Cascade(org.hibernate.annotations.CascadeType.DELETE_ORPHAN)  
    @OneToMany(mappedBy = "bs_prv", cascade=CascadeType.ALL, fetch=FetchType.LAZY)  
    private List<PrvBienServicio> bienes_servicios;  
  
}
```

Ilustración 14 (Ejemplo relaciones Hibernate) [Ver Original](#)

5. Conclusiones y Recomendaciones

La realización de la práctica de especialidad finaliza con éxito lográndose completar los objetivos propuestos y mejorando características en los avances anteriores del ERP en el que se estaba trabajando lo cual deja en todo el equipo de trabajo mucha tranquilidad, tanto por la realización de los objetivos así como por el conocimiento adquirido.

También es de recalcar que la ayuda continua, el control y las buenas prácticas de introducción a las herramientas y tecnologías que fueron utilizadas en la realización del proyecto, fueron una herramienta clave para reducir la curva de aprendizaje y facilitar el desenvolvimiento por parte de nosotros los programadores.

Por otra parte es muy importante el señalar que el uso de metodologías ágiles para la realización del proyecto, impulsó el proyecto de manera positiva ya que ayudo a aplicar elementos como el seguimiento, control, adaptabilidad al cambio, y otros que ayudaron al equipo de trabajo a llevar el proceso de desarrollo de manera fluida.

Creo que sería una buena idea modificar el protocolo de papeleo y recopilación de información en el formato que se pide actualmente, por parte de la universidad, esto para que el proceso de práctica de especialidad no se vea afectado al ejecutar proyectos utilizando metodologías ágiles. Aun así ha sido muy grata la manera en que la escuela de computación ha facilitado mecanismos para sobrellevar estos cambios.

Además creo que la realización de la práctica de especialidad es el eslabón que termina de conectar nuestra educación con el mundo empresarial donde ya teniendo el conocimiento, lo que se termina de pulir es la parte de experiencia laboral, que al fin y al cabo es lo que más nos hace falta en este momento de nuestra carrera profesional.

6. Referencias.

[1] Maven, Wikipedia. Consultado el 17 de Mayo de 2011 Disponible en:

<http://es.wikipedia.org/wiki/Maven>

[2] Java Server Faces, Wikipedia. Consultado el 17 de Mayo de 2011, Disponible en:

http://es.wikipedia.org/wiki/JavaServer_Faces

[3] Hibernate, Wikipedia. Consultado el 17 de Mayo de 2011 Disponible en:

<http://es.wikipedia.org/wiki/Hibernate>